Informatica® Cloud Data Integration

# Amazon Redshift コネクタ

Informatica Cloud Data Integration Amazon Redshift コネクタ
2022 年 10 月

© 著作権 Informatica LLC 2019, 2022

本ソフトウェアおよびマニュアルは、使用および開示の制限を定めた個別の使用許諾契約のもとでのみ提供されています。本マニュアルのいかなる部分も、いかなる手段（電子的複写、写真複写、録音など）によっても、Informatica LLC の事前の承諾なしに複製または転載することは禁じられています。

米政府の権利プログラム、ソフトウェア、データベース、および関連文書や技術データは、米国政府の顧客に配信され、「商用コンピュータソフトウェア」または「商業技術データ」は、該当する連邦政府の取得規制と代理店固有の補足規定に基づきます。このように、使用、複製、開示、変更、および適応は、適用される政府の契約に規定されている制限およびライセンス条項に従うものとし、政府契約の条項によって適当な範囲において、FAR 52.227-19、商用コンピュータソフトウェアライセンスの追加権利を規定します。

Informatica、Informatica ロゴ、Informatica Cloud、および PowerCenter は、米国およびその他の国における Informatica LLC の商標または登録商標です。Informatica の商標の最新リストは、Web（https://www.informatica.com/trademarks.html ）にあります。その他の企業名および製品名は、それぞれの企業の商標または登録商標です。

本ソフトウェアまたはドキュメンテーション（あるいはその両方）の一部は、第三者が保有する著作権の対象となります。必要な第三者の通知は、製品に含まれています。

特許については、https://www.informatica.com/legal/patents.html を参照してください。

免責: 本文書は、一切の保証を伴わない「現状渡し」で提供されるものとし、Informatica LLC は他社の権利の非侵害、市場性および特定の目的への適合性の黙示の保証などを含めて、一切の明示的および黙示的保証の責任を負いません。Informatica LLC では、本ソフトウェアまたはドキュメントに誤りのないことを保証していません。本ソフトウェアまたはドキュメントに記載されている情報には、技術的に不正確な記述や誤植が含まれる場合があります。本ソフトウェアまたはドキュメントの情報は、予告なしに変更されることがあります。

NOTICES

この Informatica 製品（以下「ソフトウェア」）には、Progress Software Corporation（以下「DataDirect」）の事業子会社である DataDirect Technologies からの特定のドライバ（以下「DataDirect ドライバ」）が含まれています。DataDirect ドライバには、次の用語および条件が適用されます。

1. DataDirect ドライバは、特定物として現存するままの状態で提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。
2. DataDirect または第三者は、予見の有無を問わず発生した ODBC ドライバの使用に関するいかなる直接的、間接的、偶発的、特別、あるいは結果的損害に対して責任を負わないものとします。本制限事項は、すべての訴訟原因に適用されます。訴訟原因には、契約違反、保証違反、過失、厳格責任、詐称、その他の不法行為を含みますが、これらに限るものではありません。

本マニュアルの情報は、予告なしに変更されることがあります。このドキュメントで問題が見つかった場合は、infa_documentation@informatica.com までご報告ください。

Informatica 製品は、それらが提供される契約の条件に従って保証されます。Informatica は、商品性、特定目的への適合性、非侵害性の保証等を含めて、明示的または黙示的ないかなる種類の保証をせず、本マニュアルの情報を「現状のまま」提供するものとします。

発行日: 2022-12-15

# 目次


**序文** . . . . . 8

Informatica のリソース . . . . . 8

Informatica マニュアル . . . . . 8

Informatica Intelligent Cloud Services Web サイト . . . . . 8

Informatica Intelligent Cloud Services コミュニティ . . . . . 8

Informatica Intelligent Cloud Services マーケットプレース . . . . . 9

データ統合コネクタのドキュメント . . . . . 9

Informatica ナレッジベース . . . . . 9

Informatica Intelligent Cloud Services Trust Center . . . . . 9

Informatica グローバルカスタマサポート . . . . . 9

**第 I 部 : Amazon Redshift コネクタの概要** . . . . . 10

**第 1 章 : Amazon Redshift コネクタの概要** . . . . . 11

Secure Agent と Amazon Redshift の統合 . . . . . 12

**第 2 章 : コネクタ比較** . . . . . 13

マッピング機能 . . . . . 13

ソース機能 . . . . . 14

ターゲット機能 . . . . . 15

**第 3 章 : タスクの例** . . . . . 17

同期タスクの使用例 . . . . . 17

マッピングおよびマッピングタスクの使用例 . . . . . 17

Oracle CDC ソースを使用したマッピングタスクの使用例 . . . . . 17

詳細モードのマッピングの使用例 . . . . . 18

**第 II 部 : Amazon Redshift V2 コネクタを使用したデータ統合** . . . . . 19

**第 4 章 : Amazon Redshift V2 コネクタの概要** . . . . . 20

Amazon Redshift V2 コネクタアセット . . . . . 20

Amazon Redshift の概要 . . . . . 21

Amazon Redshift Spectrum の概要 . . . . . 21

外部スキーマと外部テーブル . . . . . 21

Amazon Redshift V2 コネクタの管理 . . . . . 22

SSL 用に Amazon Redshift を設定 . . . . . 23

サーバーレスランタイム環境でクライアントサイド暗号化を設定するための前提条件 . . . . . 25

詳細モードのマッピングに SSE-KMS 暗号化を設定するための前提条件 . . . . . 26

最小限の Amazon IAM ポリシーの作成 . . . . . 26

IAM 認証 . . . . . 27

AssumeRole を使用した一時的なセキュリティ資格情報. . . . . 27
Amazon Redshift Spectrum の前提条件タスク. . . . . 30
マッピングのステージングパフォーマンスの最適化. . . . . 31
Amazon Redshift とのプライベート通信. . . . . 32

第 5 章 : Amazon Redshift V2 接続. . . . . 33
Amazon Redshift V2 接続のプロパティ. . . . . 33
プロキシの設定. . . . . 35
Windows でのプロキシ設定. . . . . 36
Linux でのプロキシ設定. . . . . 37

第 6 章 : Amazon Redshift ソースとターゲット. . . . . 38
Amazon Redshift V2 ソース. . . . . 38
Amazon Redshift V2 ソースの Amazon Redshift ステージングディレクトリ. . . . . 38
Amazon Redshift V2 ソースのデータ暗号化. . . . . 38
Unload コマンド. . . . . 39
ソースパーティション. . . . . 40
Amazon Redshift V2 ターゲット. . . . . 41
Amazon Redshift V2 ターゲットの Amazon Redshift ステージングディレクトリ. . . . . 42
Amazon Redshift V2 ターゲットのデータ暗号化. . . . . 42
Copy コマンド. . . . . 43
ターゲットテーブルの分析. . . . . 44
ステージングファイルの保持. . . . . 44
テーブルのクリーンアップ. . . . . 45
リカバリおよびリスタート処理. . . . . 45
書き込み時にレコードの順序を維持. . . . . 46
DELIMITER および QUOTE としての 8 進数値. . . . . 47
成功ファイルおよびエラーファイル. . . . . 47

第 7 章 : Amazon Redshift によるマッピングとマッピングタスク. . . . . 49

第 8 章 : マッピングにおける Amazon Redshift V2 オブジェクトのルールおよびガイドライン. . . . . 50
マッピングの Amazon Redshift V2 ソース. . . . . 51
SQL クエリの設定のルールおよびガイドライン. . . . . 53
複数のソースオブジェクトの追加. . . . . 54
複数のソースオブジェクトの追加のルールおよびガイドライン. . . . . 56
マッピングでの Amazon Redshift V2 ターゲット. . . . . 57
ターゲットの作成に関するルールとガイドライン. . . . . 62
ターゲットクエリをオーバーライドするためのルールとガイドライン. . . . . 63
マッピングの Amazon Redshift V2 ルックアップ. . . . . 64
接続されていないルックアップトランスフォーメーション. . . . . 67
接続されていないルックアップトランスフォーメーションの設定. . . . . 68

ルックアップキャッシングの有効化. . . . . 70
ルックアップトランスフォーメーションの設定のルールおよびガイドライン. . . . . 71
Oracle CDC ソースを使用したマッピングタスクの例. . . . . 71
詳細モードのマッピングの例. . . . . 72
SQL トランスフォーメーションを使用した SQL クエリの処理. . . . . 73
マッピングに対する動的スキーマ処理. . . . . 76
キー範囲パーティションの設定. . . . . 77
書き込み操作の一括処理. . . . . 77

**第 9 章 : マッピングの移行. . . . . 78**

移行されたマッピングへの同じオブジェクトパスの使用. . . . . 78
移行されたマッピングへの別のオブジェクトパスの使用. . . . . 78
移行オプション. . . . . 79
マッピングの移行のルールとガイドライン. . . . . 80

**第 10 章 : 接続タイプのアップグレード. . . . . 81**

接続切り替えの例. . . . . 82
切り替え後に保持される詳細プロパティ. . . . . 84
ルールおよびガイドライン. . . . . 85

**第 11 章 : Amazon Redshift のプッシュダウンの最適化. . . . . 86**

プッシュダウンの最適化のタイプ. . . . . 86
ソースのプッシュダウンの最適化と完全なプッシュダウンの最適化を使用したデータ統合の動作. . . . . 87
プッシュダウンの最適化のシナリオ. . . . . 87
Amazon Redshift V2 マッピングタスクに対する最適化の設定. . . . . 88
Amazon Redshift V2 接続を使用したプッシュダウンの最適化. . . . . 89
プッシュダウン互換性. . . . . 89
複数のターゲットに対する完全なプッシュダウンの最適化. . . . . 96
プッシュダウンの最適化のプレビュー. . . . . 97
Amazon Redshift V2 マッピングタスクに対する最適化の設定. . . . . 97
プッシュダウンの最適化ジョブのクリーンな停止. . . . . 97
プッシュダウンの最適化に関するルールおよびガイドライン. . . . . 98
Amazon Redshift ODBC 接続を使用したプッシュダウンの最適化. . . . . 104
Amazon Redshift ODBC 接続の設定. . . . . 104
ODBC 接続の作成. . . . . 108
接続されていないルックアップトランスフォーメーションの設定. . . . . 110
クロススキーマプッシュダウンの最適化. . . . . 112
Amazon Redshift ソースオブジェクトに対するカスタムクエリまたは SQL オーバーライドの設定. . . . . 113
Amazon Redshift マッピングでサポートされる関数および演算子. . . . . 113
Amazon Redshift ODBC マッピングでサポートされているトランスフォーメーション. . . . . 116
プッシュダウンの最適化に関するルールおよびガイドライン. . . . . 116

第 12 章 : データ型リファレンス. . . . . 118
Amazon Redshift とトランスフォーメーションのデータ型. . . . . 118
データ型のルールとガイドライン. . . . . 119

第 13 章 : トラブルシューティング概要. . . . . 121
Amazon Redshift V2 コネクタのトラブルシューティング. . . . . 121

第 III 部 : Amazon Redshift コネクタを使用したデータ統合. . . . . 123

第 14 章 : Amazon Redshift コネクタの概要. . . . . 124
Amazon Redshift コネクタのアセット. . . . . 124
Amazon Redshift の概要. . . . . 125
Amazon Redshift コネクタの例. . . . . 125
Amazon Redshift コネクタの管理. . . . . 125
SSL 用の Amazon Redshift の設定. . . . . 126
最小の Amazon IAM ポリシーの作成. . . . . 126
IAM 認証. . . . . 127

第 15 章 : Amazon Redshift 接続の概要. . . . . 129
Amazon Redshift 接続のプロパティ. . . . . 129
プロキシの設定. . . . . 130
Windows でのプロキシ設定. . . . . 130
Linux でのプロキシ設定. . . . . 131

第 16 章 : Amazon Redshift ソースとターゲット. . . . . 132
Amazon Redshift ソース. . . . . 132
Amazon Redshift ソースの Amazon Redshift ステージングディレクトリ. . . . . 132
Amazon Redshift ソースのサーバーサイド暗号化. . . . . 133
Amazon Redshift ソースのクライアントサイド暗号化. . . . . 133
Unload コマンド. . . . . 134
パーティション化. . . . . 135
Amazon Redshift ターゲット. . . . . 135
Amazon Redshift ターゲットの Amazon Redshift ステージングディレクトリ. . . . . 136
ターゲットテーブルの分析. . . . . 136
Amazon Redshift ターゲットのデータ暗号化. . . . . 137
ステージングファイルの保持. . . . . 137
Copy コマンド. . . . . 138
フィールドマッピング. . . . . 139
テーブルのクリーンアップ. . . . . 139
大きいテーブルに関する作業. . . . . 140
DELIMITER および QUOTE としての 8 進数値. . . . . 140
成功ファイルおよびエラーファイル. . . . . 141

第 17 章 : Amazon Redshift による同期タスク. . . . . 143
同期タスクの Amazon Redshift ソース. . . . . 143
同期タスクの Amazon Redshift ターゲット. . . . . 145
同期タスクの Amazon Redshift ルックアップ. . . . . 148
同期タスクのルールおよびガイドライン. . . . . 148
同期タスクの例. . . . . 149

第 18 章 : Amazon Redshift によるマッピングとマッピングタスク. . . . 150
マッピングの Amazon Redshift ソース. . . . . 150
キー範囲パーティション化の設定. . . . . 152
マッピングの Amazon Redshift ターゲット. . . . . 153
マッピングの Amazon Redshift ルックアップ. . . . . 156
テンプレートベースのマッピングタスクの Amazon Redshift オブジェクト. . . . . 157
マッピングタスクの Amazon Redshift ソース. . . . . 157
マッピングタスクの Amazon Redshift ターゲット. . . . . 158

第 19 章 : Amazon Redshift のプッシュダウンの最適化. . . . . 162
Amazon Redshift マッピングでサポートされる関数および演算子. . . . . 162
Amazon Redshift ODBC 接続の設定. . . . . 165
Windows 上の Amazon Redshift ODBC 接続の設定. . . . . 165
Linux 上の Amazon Redshift ODBC 接続の設定. . . . . 167
ODBC 接続の作成. . . . . 168
クロススキーマプッシュダウンの最適化. . . . . 170
Amazon Redshift マッピングタスクに対するクロススキーマの最適化の設定. . . . . 170
プッシュダウンの最適化の関数に関するルールおよびガイドライン. . . . . 171

第 20 章 : データ型リファレンスの概要. . . . . 172
Amazon Redshift とトランスフォーメーションのデータ型. . . . . 172

第 21 章 : トラブルシューティング概要. . . . . 174
Amazon Redshift コネクタのトラブルシューティング. . . . . 174
Amazon Redshift 接続のトラブルシューティング. . . . . 174

索引. . . . . 176

# 序文

*Amazon Redshift コネクタ*を使用して、*Amazon Redshift* と *Amazon Redshift V2 コネクタ*について学びます。パート I を使用して、概要と Amazon Redshift V2 コネクタと Amazon Redshift コネクタ間の機能比較を確認します。パート II とパート III を使用して、Amazon Redshift コネクタで利用できる機能について学習します。

# Informatica のリソース

Informatica は、Informatica Network やその他のオンラインポータルを通じてさまざまな製品リソースを提供しています。リソースを使用して Informatica 製品とソリューションを最大限に活用し、その他の Informatica ユーザーや各分野の専門家から知見を得ることができます。

## Informatica マニュアル

Informatica マニュアルポータルでは、最新および最近の製品リリースに関するドキュメントの膨大なライブラリを参照できます。マニュアルポータルを利用するには、https://docs.informatica.com にアクセスしてください。

製品マニュアルに関する質問、コメント、ご意見については、Informatica マニュアルチーム（infa_documentation@informatica.com）までご連絡ください。

## Informatica Intelligent Cloud Services Web サイト

Informatica Intelligent Cloud Services Web サイト（http://www.informatica.com/cloud）にアクセスできます。このサイトには、Informatica Cloud 統合サービスに関する情報が含まれます。

## Informatica Intelligent Cloud Services コミュニティ

Informatica Intelligent Cloud Services コミュニティを使用して、技術的な問題について議論し、解決します。また、技術的なヒント、マニュアルの更新情報、FAQ（よくある質問）への答えを得ることもできます。

次の Informatica Intelligent Cloud Services コミュニティにアクセスします。

https://network.informatica.com/community/informatica-network/products/cloud-integration

開発者は、次の Cloud 開発者コミュニティで詳細情報を確認したり、ヒントを共有したりできます。

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers

## Informatica Intelligent Cloud Services マーケットプレース

Informatica マーケットプレイスにアクセスすると、データ統合コネクタ、テンプレート、およびマップレットを試用したり購入したりできます。

https://marketplace.informatica.com/

## データ統合コネクタのドキュメント

データ統合コネクタのドキュメントには、マニュアルポータルからアクセスできます。マニュアルポータルを利用するには、https://docs.informatica.com にアクセスしてください。

## Informatica ナレッジベース

Informatica ナレッジベースを使用して、ハウツー記事、ベストプラクティス、よくある質問に対する回答など、製品リソースを見つけることができます。

ナレッジベースを検索するには、https://search.informatica.com にアクセスしてください。ナレッジベースに関する質問、コメント、ご意見の連絡先は、Informatica ナレッジベースチーム（KB_Feedback@informatica.com）です。

## Informatica Intelligent Cloud Services Trust Center

Informatica Intelligent Cloud Services Trust Center は、Informatica のセキュリティポリシーおよびリアルタイムでのシステムの可用性について情報を提供します。

Trust Center（https://www.informatica.com/trust-center.html）にアクセスします。

Informatica Intelligent Cloud Services Trust Center にサブスクライブして、アップグレード、メンテナンス、およびインシデントの通知を受信します。Informatica Intelligent Cloud Services Status ページには、すべての Informatica Cloud 製品の実稼働ステータスが表示されます。メンテナンスの更新はすべてこのページに送信され、停止中は最新の情報が表示されます。更新と停止の通知がされるようにするには、Informatica Intelligent Cloud Services の 1 つのコンポーネントまたはすべてのコンポーネントについて更新の受信をサブスクライブします。すべてのコンポーネントにサブスクライブするのが、更新を逃さないようにするための最良の方法です。

登録するには、https://status.informatica.com/に移動し、**[更新を購読登録]** をクリックします。その後、電子メール、SMS テキストメッセージ、Webhook、RSS フィードとして、またはこの 4 つを任意に組み合わせて送信された通知を受信することを選択ができます。

## Informatica グローバルカスタマサポート

電話またはオンラインでカスタマサポートセンターに連絡できます。

オンラインサポートについては、Informatica Intelligent Cloud Services の **[サポート要求の送信]** をクリックしてください。またオンラインサポートを使用して問題を記録することもできます。オンラインサポートを利用するには、ログインが必要です。https://network.informatica.com/welcome でログイン要求できます。

Informatica グローバルカスタマサポートの電話番号は、Informatica の Web サイト https://www.informatica.com/services-and-training/support-services/contact-us.html に掲載されています。

# パート I: Amazon Redshift コネクタの概要

この部には、以下の章があります。


- Amazon Redshift コネクタの概要, 11 ページ
- コネクタ比較, 13 ページ
- タスクの例, 17 ページ

# 第 1 章

# Amazon Redshift コネクタの概要

Amazon Redshift コネクタを使用して、Amazon Redshift に対してデータの読み取りおよび書き込みを実施できます。コネクタを使用して、Amazon Redshift のレコードを表すソースとターゲットを作成します。。

Amazon Redshift コネクタを使用してデータ統合タスクを作成および実行する場合、Secure Agent は、タスクフローと Amazon Redshift 接続設定を基にして Amazon Redshift に対してデータの読み取りおよび書き込みを行います。Secure Agent は、TCP/IP ネットワークを介して Amazon Simple Storage Service（Amazon S3）に接続し、データに対して読み取りと書き込みを行います。Secure Agent は Amazon ドライバを使用して Amazon Redshift と通信します。

任意のデータソースから Amazon Redshift にデータを移動できます。

次のコネクタを使用して、接続を作成し、Amazon Redshift に対してデータを統合します。

Amazon Redshift V2 **コネクタ**

これは Amazon Redshift の接続に推奨されるコネクタです。Amazon Redshift V2 コネクタを使用して、マッピングまたはマッピングタスクを作成します。

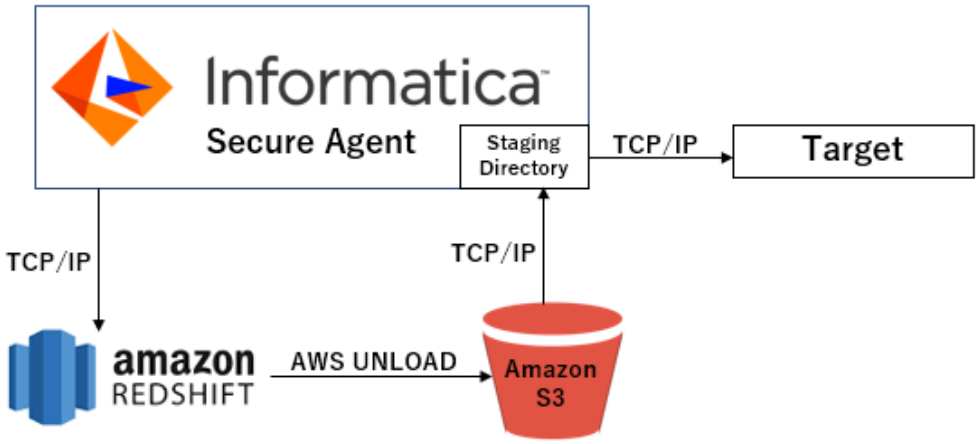
Amazon Redshift **コネクタ**

これは Amazon Redshift コネクタの古いバージョンです。同期タスクを使用するときにのみ、Amazon Redshift コネクタを使用してデータの読み取りと書き込みを行うことをお勧めします。

# Secure Agent と Amazon Redshift の統合

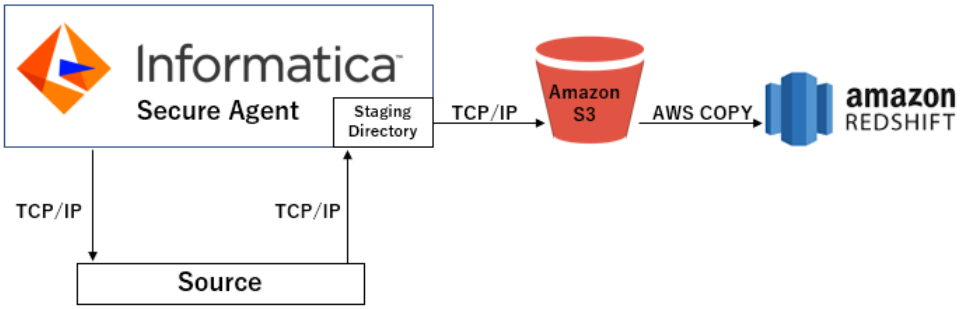
Secure Agent は、Amazon Redshift 接続を使用して Amazon Redshift に接続します。

次の図は、Secure Agent が Amazon Redshift に接続してデータを読み取る方法を示しています。




Secure Agent は、Amazon Redshift に接続して UNLOAD コマンドを発行し、Amazon Redshift から接続プロパティに指定された Amazon Simple Storage Service（Amazon S3）バケットにデータを読み取ります。その後、Secure Agent は TCP/IP ネットワークを経由して Secure Agent システム上でステージングディレクトリにデータを保存します。

次の図は、Secure Agent が Amazon Redshift に接続してデータを書き込む方法を示しています。




Secure Agent は、Secure Agent システムのステージングディレクトリからデータを読み取り、TCP/IP ネットワークを経由して Amazon Simple Storage Service（Amazon S3）にデータを書き込みます。Secure Agent は、Amazon S3 から Amazon Redshift のターゲットテーブルにデータを書き込む COPY コマンドを発行します。

# 第 2 章

# コネクタ比較

データを統合または取り込むための要件に基づいて、Amazon Redshift コネクタまたは Amazon Redshift V2 コネクタのいずれかを使用して、データ統合タスクを作成できます。

統合タスクの作成と読み取りおよび書き込み操作を設定するための機能は両方のコネクタで異なります。Amazon Redshift V2 コネクタに新機能と改良点が提供されているため、Amazon Redshift V2 コネクタを使用することをお勧めします。

## マッピング機能

次の表に、Amazon Redshift コネクタによってサポートされるマッピング機能の比較を示します。

| **マッピング機能** | Amazon Redshift **コネクタ** | Amazon Redshift V2 **コネクタ** |
|---|---|---|
| Hosted Agent | ○ | はい |
| プロキシサーバー | ○ | はい |
| 同期タスク | はい | いいえ |
| マッピングタスク | ○ | はい（優先） |
| 詳細モードのマッピング | いいえ | はい |
| Oracle CDC ソースからの読み取り | いいえ | はい |

| マッピング機能 | Amazon Redshift コネクタ | Amazon Redshift V2 コネクタ |
|---|---|---|
| プッシュダウンの最適化 | ODBC Subtype=Redshift の ODBC 接続を使用して、Amazon Redshift ソースとターゲット間のソースまたは完全なプッシュダウンの最適化を有効にします。 | - Amazon S3 V3 ソース接続および Amazon Redshift V2 ターゲット接続を使用して、Amazon S3 ソースと Amazon Redshift ターゲット間の完全なプッシュダウンの最適化を有効にします。<br>- ODBC Subtype=Redshift の ODBC 接続を、Amazon Redshift ソースとターゲット間のプッシュダウンの最適化に使用します。<br>**注**: プッシュダウンの最適化は、詳細モードのマッピングには適用されません。 |
| Lookup トランスフォーメーション | キャッシュ、キャッシュなし、および接続済 | キャッシュ、キャッシュなし、接続済、および未接続<br>**注**: キャッシュされていないルックアップは、詳細モードのマッピングには適用されません。 |

次の表に、Amazon Redshift コネクタによってサポートされる Amazon Redshift 機能を一覧表示します。

| Amazon Redshift 機能 | Amazon Redshift コネクタ | Amazon Redshift V2 コネクタ |
|---|---|---|
| VPC エンドポイント | ○ | はい |
| Redshift Spectrum | いいえ | ○ |

# ソース機能

Amazon Redshift からオブジェクトをインポートしてデータを読み取るときに、詳細ソースのプロパティを設定して読み取り操作の動作を決定することができます。例えば、暗号化形式でデータを読み取ったり、最適なパフォーマンスにパーティション化を設定できます。

次の表に、Amazon Redshift コネクタと Amazon Redshift V2 コネクタ用に Amazon Redshift ソースからデータを読み取るときに使用できるソース機能を一覧表示します。

| 特徴 | Amazon Redshift コネクタ | Amazon Redshift V2 コネクタ |
|---|---|---|
| ステージングディレクトリ [1] | はい | はい |
| サーバーサイド暗号化 | ○ | はい |
| KMS を使用したサーバーサイド暗号化 | ○ | はい |
| クライアントサイド暗号化 [1] | はい | はい |
| Unload コマンド | ○ | はい |
| パーティション化 [1] | はい | はい |
| 大きいテーブルに関する作業 | ○ | はい |

| 特徴 | Amazon Redshift コネクタ | Amazon Redshift V2 コネクタ |
|---|---|---|
| DELIMITER および QUOTE としての 8 進数値 | ○ | はい |
| 成功ファイルおよびエラーファイル | ○ | はい |
| 異なるスキーマからオブジェクトをインポート | いいえ | はい |
| [1] 詳細モードのマッピングには適用されません。 | | |

# ターゲット機能

Amazon Redshift からオブジェクトをインポートしてデータを書き込むときに、詳細ターゲットのプロパティを設定して書き込み操作の動作を決定することができます。例えば、暗号化形式でデータを書き込んだり、書き込み操作の完了後に Amazon S3 ステージングファイルを保持できます。

次の表に、データを Amazon Redshift ターゲットに書き込むときに使用できるターゲット機能を一覧表示します。

| ターゲット機能 | Amazon Redshift コネクタ | Amazon Redshift V2 コネクタ |
|---|---|---|
| ステージングディレクトリ [1] | はい | はい |
| サーバーサイド暗号化 | ○ | はい |
| クライアントサイド暗号化 [1] | はい | はい |
| ターゲットテーブルの分析 | ○ | はい |
| ステージングファイルの保持 | ○ | はい |
| Copy コマンド | ○ | はい |
| テーブルのクリーンアップ | ○ | はい |
| リカバリおよびリスタート処理 [1] | いいえ | はい |
| 書き込み時にレコードの順序を維持 [1] | いいえ | はい |
| 大きいテーブルに関する作業 | ○ | はい |
| DELIMITER および QUOTE としての 8 進数値 | ○ | はい |
| 成功ファイルおよびエラーファイル [1] | はい | はい |

| **ターゲット機能** | Amazon Redshift **コネクタ** | Amazon Redshift V2 **コネクタ** |
| --- | --- | --- |
| 異なるスキーマからオブジェクトをインポート | いいえ | はい |
| [1] 詳細モードのマッピングには適用されません。 | | |

# 第 3 章

# タスクの例

このセクションでは、Amazon Redshift および Amazon Redshift V2 コネクタがサポートするすべてのタスクの例を示します。

## 同期タスクの使用例

MySQL データベースで受注詳細を保存する電子商取引の組織に勤めているとします。組織は、MySQL データベースから Amazon Redshift ターゲットにデータを移動する必要があります。

Amazon Redshift コネクタを使用して、同期タスクを作成し、Amazon Redshift ターゲットに書き込みます。

## マッピングおよびマッピングタスクの使用例

顧客 ID、品目コード、品目数量など、注文書の詳細をオンプレミスの MySQL データベースに保存する組織で作業しています。注文書の詳細を分析して、特定の状態で注文された品目を確認し、入手可能なクラウドベース環境で状態の優れているターゲットテーブルにオンプレミスの MySQL データベースからデータを移動する必要があります。

Amazon Redshift V2 コネクタを使用して、パラメータ化されたマッピングを作成し、MySQL データベースから注文書を優れた状態で読み取り、複数の Amazon Redshift ターゲットに書き込み、すべての状態に対して今後のマーケティングキャンペーンを準備します。

## Oracle CDC ソースを使用したマッピングタスクの使用例

組織では、ミッションクリティカルな Oracle の本番システムからリアルタイムの変更データをレプリケートして、重大ではない作業の介入（オフラインのレポートシステムや分析処理システムなど）を最小限に抑える必要があります。

Amazon Redshift V2 コネクタを使用して、Oracle CDC ソースから変更データをキャプチャして Amazon Redshift のターゲットテーブルに書き込みます。Oracle CDC ソースをマッピングに追加して、関連付けられたマッピングタスクを実行すると、変更データがターゲットに書き込まれます。

# 詳細モードのマッピングの使用例

顧客 ID、品目コード、および品目数量などの大量の注文書の詳細を Amazon S3 に保存している組織で働いているとします。注文書の詳細を迅速に分析して将来の収益を増加させるために、Amazon S3 から別のクラウドベースの環境にデータを移植する必要があります。

Amazon Redshift V2 コネクタを使用して、Amazon S3 からすべての注文レコードを読み取って Amazon Redshift ターゲットに書き込む場合のパフォーマンスを向上させるために、詳細モードでマッピングを作成します。

# パート II: Amazon Redshift V2 コネクタを使用したデータ統合

この部には、以下の章があります。


- Amazon Redshift V2 コネクタの概要, 20 ページ
- Amazon Redshift V2 接続, 33 ページ
- Amazon Redshift ソースとターゲット, 38 ページ
- Amazon Redshift によるマッピングとマッピングタスク, 49 ページ
- マッピングにおける Amazon Redshift V2 オブジェクトのルールおよびガイドライン, 50 ページ
- マッピングの移行, 78 ページ
- 接続タイプのアップグレード, 81 ページ
- Amazon Redshift のプッシュダウンの最適化, 86 ページ
- データ型リファレンス, 118 ページ
- トラブルシューティング概要, 121 ページ

# 第 4 章

# Amazon Redshift V2 コネクタの概要

Amazon Redshift V2 コネクタを使用して、Amazon Redshift に対してデータの読み取りおよび書き込みを安全に実施できます。Amazon Redshift V2 のソースとターゲットは、Amazon Redshift のレコードとして表されます。

Amazon Redshift Serverless クラスタに接続して、データの読み取りまたは書き込みを行うこともできます。任意のデータソースから Amazon Redshift にデータを移動できます。データ統合は Amazon ドライバを使用して Amazon Redshift と通信します。

Amazon Redshift V2 接続を作成して、マッピング、およびマッピングタスクで使用できます。マッピングを詳細モードに切り替えて、高度な機能を有効にするためのトランスフォーメーションと関数を含めることができます。詳細クラスタは、セルフサービスクラスタ、ローカルクラスタ、または Amazon Web Services でホストされている場合があります。

マッピングタスクを作成して、マッピングまたは統合テンプレートで定義されているデータフローロジックに基づきデータを処理します。Oracle CDC ソースから変更データをキャプチャして Amazon Redshift のターゲットテーブルに書き込むマッピングタスクを作成することもできます。

Amazon Redshift V2 のマッピング、またはマッピングタスクを実行する場合、Secure Agent はワークフローと Amazon Redshift V2 の接続設定に基づいてデータを Amazon Redshift に書き込みます。Secure Agent は、TCP/IP ネットワークを介して Amazon Simple Storage Service（Amazon S3）に接続し、データを書き込みます。Amazon S3 は、ソースからデータをコピーするのと同時にそのデータを Amazon Redshift クラスタに移動できるストレージサービスです。Secure Agent は、Amazon S3 から Amazon Redshift のターゲットテーブルにデータをコピーする copy コマンドを発行します。

**注:** ローカルクラスタのサポートはプレビューで利用できます。評価目的でのプレビュー機能はサポートされていますが、保証されておらず、本番環境には対応していません。非本番環境でのみ使用することをお勧めします。Informatica は、本番環境用の今後のリリースにプレビュー機能を含める予定ですが、市場や技術的な状況の変化に応じて導入を行わない場合もあります。詳細については、Informatica グローバルカスタマサポートにお問い合わせください。

## Amazon Redshift V2 コネクタアセット

データ統合でアセットを作成し、Amazon Redshift V2 コネクタを使用してデータを統合します。

Amazon Redshift V2 コネクタを使用する場合は、次のようなデータ統合アセットを含めることができます。

- データ転送タスク

- 動的マッピングタスク
- マッピング
- マッピングタスク

アセットとトランスフォーメーションの設定に関する詳細については、データ統合のドキュメントの「*マッピング*」、「*トランスフォーメーション*」、および「*タスク*」を参照してください。

# Amazon Redshift の概要

Amazon Redshift は、組織がデータの分析と保存に使用できるクラウドベースのペタバイト規模のデータウェアハウスサービスです。

Amazon Redshift は、カラムナデータストレージ、並列処理、およびデータ圧縮を使用し、データを保存して高速なクエリ実行を実現しています。Amazon Redshift は、リーダーノードと計算ノードで構成されたクラスタベースのアーキテクチャを使用します。リーダーノードは計算ノードを管理し、外部クライアントプログラムと通信します。リーダーノードはクライアントアプリケーションと対話し、計算ノードと通信します。計算ノードはデータを保存し、リーダーノードのクエリを実行します。PostgreSQL ドライバを使用するすべてのクライアントは、Amazon Redshift と通信できます。

# Amazon Redshift Spectrum の概要

Amazon Redshift Spectrum により、Amazon S3 に格納されたさまざまな形式の大量のデータに対して、複雑な Amazon Redshift SQL クエリを実行できます。Amazon Redshift Spectrum では、データをロードまたは変換することなく、Amazon S3 データファイルを読み取るためのクエリを直接実行できます。

指定した Amazon Redshift クラスタを拡張することなく、大量の Amazon S3 データファイルのクエリを実行できます。

Amazon Redshift Spectrum は、Amazon Redshift クラスタとは独立した Amazon Redshift サーバーに存在します。Amazon Redshift Spectrum を使用してクエリを実行すると、計算が大量に必要なすべてのタスクを Amazon Redshift Spectrum が Amazon Redshift Spectrum レイヤにプッシュするため、クエリがより高速に実行され、使用される Amazon Redshift クラスタの処理能力が低減されます。

## 外部スキーマと外部テーブル

Amazon Redshift Spectrum を使用するには、外部データカタログ内のデータベースを参照する外部スキーマ内に外部テーブルを作成する必要があります。Avro、ORC、Parquet、RCFile、SequenceFIile、および Textfile のファイル形式の外部テーブルを作成できます。

外部データベースと外部テーブルのメタデータは、外部データカタログに格納されます。Amazon S3 のデータカタログとデータファイルにアクセスするには、Amazon Redshift の権限を付与する必要があります。

Amazon Redshift で外部データベースを作成できます。単一の外部テーブル、複数の外部テーブル、または外部テーブルに結合された標準の Amazon Redshift テーブルからデータを読み取ることができます。

複数の Amazon Redshift クラスタに複数の外部テーブルを含めることができます。同じリージョンの Amazon Redshift クラスタから、Amazon S3 上の同じデータのクエリを実行できます。Amazon S3 でデータを更新すると、すべての Amazon Redshift クラスタでそのデータがすぐに使用可能になります。

外部テーブルを作成するときは、データを読み取る Amazon S3 の場所を指定する必要があります。外部テーブルを作成するには、Amazon S3 データファイルの構造を定義し、外部データカタログに外部テーブルを登録します。その後、クエリの実行や、外部テーブルの結合を行うことができます。

外部テーブルをソースとして追加してマッピングを作成すると、**[ソースオブジェクトの選択]** ダイアログボックスに `spectrum_schemaname` 形式で外部テーブル名が表示されます。

Athena または Glue のデータカタログを使用して外部テーブルを作成する際には、使用するデータ型が Amazon Redshift V2 コネクタでサポートされていることを確認してください。

外部テーブルを作成するときに Amazon Redshift V2 コネクタがサポートしているデータ型を以下に示します。

- Bigint（INT8）
- Boolean（BOOL）
- Char（CHARACTER）
- Date

  **注:** ORC、Parquet、および Textfile のファイル形式の外部テーブルを作成するときに適用できます。
- Decimal（NUMERIC）
- Double Precision（FLOAT8）
- Integer（INT、INT4）
- Real（FLOAT4）
- Smallint（INT2）
- タイムスタンプ
- Varchar（CHARACTER VARYING）

**外部テーブルのルールおよびガイドライン**

外部テーブルについては、次のルールとガイドラインを考慮してください。

- Amazon Redshift Spectrum 外部テーブルからはデータの読み取りのみができます。Amazon Redshift Spectrum 外部テーブルに対してデータの挿入や更新を行うことはできません。
- Secure Agent では、ターゲットトランスフォーメーション内で使用できるターゲットオブジェクトのリストから外部テーブル名が削除されることはありません。
- 外部テーブルで pre-SQL および post-SQL コマンドを使用してターゲット操作を実行することはできません。

外部テーブルを作成する方法の詳細については、AWS のマニュアルを参照してください。

# Amazon Redshift V2 コネクタの管理

ユーザーが Amazon Redshift V2 コネクタを使用するには、成功ファイルおよびエラーファイルが格納された Secure Agent ディレクトリに対するアクセス権がユーザーに与えられていることを組織の管理者が確認する必要があります。ディレクトリパスは、各 Secure Agent マシンのランタイム環境で同じである必要があります。組織の管理者は次のタスクを実行するも必要があります。

- Amazon Redshift JDBC URL を取得します。

- 認証を管理します。次の 2 つの方法のいずれかを使用します。
  - アクセスキー ID とシークレットアクセスキーを作成する。
    Amazon Redshift V2 接続を設定する際に、そのアクセスキー ID とシークレットアクセスキーの値を指定します。アクセスキー ID とシークレットアクセスキーの作成の詳細については、AWS のマニュアルを参照してください。
  - AWS Identity and Access Management（IAM）認証を設定してセキュリティを強化する。
    IAM 認証を使用する場合は、アクセスキー ID とシークレットアクセスキーを Amazon Redshift V2 接続で明示的に指定しないでください。代わりに、Redshift ロールの Amazon リソース名（ARN）を作成し、最小の Amazon IAM ポリシーを Redshift ロールの ARN に追加して、Redshift ロールの ARN を Redshift クラスタに追加する必要があります。

    タスクを作成する際に、UNLOAD コマンドと COPY コマンドの AWS_IAM_ROLE オプションに Redshift ロールの ARN を指定します。

    接続プロパティのアクセスキー ID とシークレットアクセスキーおよび UNLOAD コマンドと COPY コマンドの AWS_IAM_ROLE の両方を指定した場合は、AWS_IAM_ROLE が優先されます。

    IAM 認証と、顧客マスタキーを使用したサーバーサイド暗号化を使用する場合は、IAM EC2 ロールと IAM Redshift ロールを顧客マスタキーに追加する必要があります。

    Hosted Agent は IAM 認証をサポートしていません。Amazon Redshift V2 コネクタの IAM 認証を設定する方法については、「IAM 認証 」 (ページ 27)を参照してください。
- SSL 接続をサポートする場合は、Amazon Redshift を SSL 用に設定します。
- クライアントサイド暗号化を有効にする場合は、マスタ対称キーを作成します。
- サーバーサイド暗号化を有効にする場合は、AWS Key Management Service（AWS KMS）で管理された顧客マスタキーを作成します。
- Amazon Redshift V2 コネクタ用に最小の Amazon IAM ポリシーを作成します。
- Amazon Redshift Spectrum を使用して Amazon S3 のデータカタログとデータファイルにアクセスするには、Amazon Redshift クラスタに必要な権限があることを確認します。
- CDC ソースから変更データをキャプチャするマッピングを作成し、関連付けられたマッピングタスクを実行して変更データを Amazon Redshift ターゲットに書き込む場合は、CDC ソースを設定します。
  CDC ソースとのマッピングを作成するには PowerExchangeClient と CDC のライセンスが必要です。
- 詳細モードで実行するようにマッピングを設定する場合は、Redshift クラスタと詳細クラスタが同じ Virtual Private Cloud（VPC）にあることを確認してください。
- サーバーレスランタイム環境を使用する際に、サーバーレスランタイム環境と Amazon Redshift クラスタが異なる VPC にある場合は、VPC ピアリングを設定する必要があります。詳細については、How-To ライブラリの記事
  「https://docs.informatica.com/integration-cloud/cloud-data-integration-connectors/h2l/1678-configure-vpc-peering-between-the-amazon-redshift-cluster-a/abstract.html」を参照してください。
- アカウント間のアクセスに Amazon リソースネーム（ARN）を使用するには、AWS によって説明されている前提条件に従っていることを確認してください。詳細については、AWS のマニュアルを参照してください。

## SSL 用に Amazon Redshift を設定

Amazon Redshift への SSL 接続をサポートするように Secure Agent を設定できます。

1. 次の場所から Amazon Redshift 証明書をダウンロードします。
   https://s3.amazonaws.com/redshift-downloads/redshift-ssl-ca-cert.pem
2. 次のコマンドを実行してキーストアに証明書ファイルを追加します。`${JAVA_HOME}/bin/keytool -keystore {JAVA_HOME}/lib/security/cacerts -import -alias <string_value> -file <certificate_filepath>`

3. 管理者で、**［ランタイム環境］**を選択します。
4. Secure Agent のリストから Secure Agent を選択します。
5. 右上隅の**［編集］**をクリックします。
6. **［システム構成の詳細］**セクションで、**［タイプ］**を**［DTM］**に変更します。
7. **［JVMOption1］**の横にある**［エージェント設定の編集］**アイコンをクリックし、次のコマンドを入力します。`- Djavax.net.ssl.trustStore=<keystore_name>`
8. **［JVMOption2］**の横にある**［エージェント設定の編集］**アイコンをクリックし、次のコマンドを追加します。`- Djavax.net.ssl.trustStorePassword=<password>`
9. Amazon Redshift V2 接続のプロパティで指定した JDBC URL に次のパラメータを追加します。`ssl=true`。例: `jdbc:redshift://mycluster.xyz789.us-west- 2.redshift.amazonaws.com:5439/dev?ssl=true`
10. **［OK］**をクリックして変更を保存します。

## サーバーレスランタイム環境で SSL を設定するための前提条件

Amazon Redshift V2 コネクタでサーバーレスランタイム環境を使用して、SSL 対応の Amazon Redshift データベースに接続できます。

サーバーレスランタイム環境を使用して安全な Amazon Redshift V2 接続を設定する前に、次のタスクを実行する必要があります。

- Amazon　S3 バケットに SSL 証明書を追加する。
- .yml サーバーレス構成ファイルを設定する。
- サーバーレス環境を設定する。
- SSL を使用するように接続プロパティを設定する。

### Amazon　S3 バケットに SSL 証明書を追加する

サーバーレスランタイム環境で SSL 接続を設定するには、次の手順を実行します。

1. AWS のサーバーレスエージェント設定用に次の構造を作成します: `<Supplementary file location>/serverless_agent_config`
2. AWS アカウントの次の場所にある Amazon S3 バケットに証明書名とソースパスを追加します: `<Supplementary file location>/serverless_agent_config/SSL`

### .yml サーバーレス構成ファイルを設定する

次の手順を実行して、サーバーレスランタイム環境で.yml サーバーレス構成ファイルを設定し、証明書名とパスエントリを追加して、Amazon Redshift V2 コネクタが SSL を使用できるようにします。

1. 次のコードスニペットをテキストエディタにコピーします。

```
version: 1
agent:
  agentAutoApply:
    general:
      sslStore:
        - fileCopy:
            sourcePath: SSL/<cert_name>
        - importCerts:
            certName: <cert_name>
            alias: <alias name of the certificate>
```

   ここで、ソースパスは AWS の証明書ファイルのディレクトリパスです。

2. 構文とインデントが有効であることを確認してから、ファイルを `serverlessUserAgentConfig.yml` として次の AWS の場所に保存します: `<Supplementary file location>/serverless_agent_config`
   .yml ファイルが実行されると、SSL 証明書が AWS の場所からサーバーレスエージェントディレクトリにコピーされます。

**サーバーレス環境を設定する**

サーバーレスランタイム環境で JVMOption1 と JVMOption2 を設定します。

1. サーバーレスランタイム環境のプロパティに移動し、**[編集]** をクリックします。
2. **[ランタイム設定のプロパティ]** タブで **[JVMoption1]** をクリックし、次のプロパティを追加します。
   `-Djavax.net.ssl.trustStore=/home/cldagnt/SystemAgent/jdk/jre/lib/security/cacerts`
3. **[JVMoption2]** をクリックして、次のプロパティを追加します。
   `-Djavax.net.ssl.trustStorePassword=changeit`
4. **[保存]** をクリックします。
5. ランタイム環境を再デプロイします。

SSL **を使用するように接続プロパティを設定する**

サーバーレスランタイム環境でランタイムプロパティを設定した後、**[JDBC URL]** 接続プロパティで `ssl=true` と指定します。

例: `jdbc:redshift://mycluster.xyz789.us-west- 2.redshift.amazonaws.com:5439/dev?ssl=true`

# サーバーレスランタイム環境でクライアントサイド暗号化を設定するための前提条件

Amazon Redshift V2 コネクタでサーバーレスランタイム環境を使用して、クライアントサイド暗号化を設定できます。

サーバーレスランタイム環境を使用してクライアントサイド暗号化を設定する前に、.yml サーバーレス構成ファイルを設定する必要があります。

.yml **サーバーレス構成ファイルを設定する**

次の手順を実行して、サーバーレスランタイム環境で.yml サーバーレス構成ファイルを設定し、Amazon Redshift V2 コネクタがクライアントサイド暗号化を使用できるようにします。

1. 次のコードスニペットをテキストエディタにコピーします。

```
version: 1
agent:
  agentAutoApply:
    general:
    jdk:
         security:
         policyJars:
           - local_policy.jar
        - US_export_policy.jar
```

2. 構文とインデントが有効であることを確認してから、ファイルに `serverlessUserAgentConfig.yml` という名前を付け、次の AWS の場所に保存します:
   `<Supplementary file location>/serverless_agent_config`

   .yml ファイルが実行されると、ポリシー jar が AWS の場所からサーバーレスエージェントディレクトリにコピーされます。
3. .yml 構成ファイルを更新した後に、サーバーレスランタイム環境を再デプロイします。

接続プロパティでマスタ対称キーを指定し、ソースおよびターゲットの詳細プロパティでクライアントサイド暗号化のタイプを指定します。

## 詳細モードのマッピングに SSE-KMS 暗号化を設定するための前提条件

詳細モードのマッピングに SSE-KMS 暗号化を使用するには、次のいずれかのタスクを実行する必要があります。

- ~/.aws/credentials の場所にある資格情報を使用するには、AWS でマスターインスタンスプロファイルとワーカーインスタンスプロファイルを作成し、KMS ポリシーをワーカープロファイルにアタッチして、クラスタ設定でプロファイルを指定する必要があります。
- Amazon EC2 で Secure Agent を使用し、AWS でマスターインスタンスプロファイルとワーカーインスタンスプロファイルを作成して、KMS ポリシーをワーカープロファイルにアタッチします。
- Amazon EC2 で Secure Agent を使用し、デフォルトの IAM ロールを使用して、KMS ポリシーを Secure Agent ロールにアタッチします。

## 最小限の Amazon IAM ポリシーの作成

Amazon IAM ポリシーを作成して、Amazon Redshift との間でデータの読み取りと書き込みを行う際に Amazon S3 でデータのステージングを実行するために必要なアクセス権限を定義します。

Amazon S3 でデータのステージングを実行するには、次の必要な最小の権限を使用します。

- PutObject
- GetObject
- DeleteObject
- ListBucket
- ListBucketMultipartUploads.詳細モードのマッピングにのみ適用されます。

次のサンプル Amazon IAM ポリシーを使用できます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:ListBucket",
                                                                "s3:ListBucketMultipartUploads"
                        ],
            "Resource": [
                "arn:aws:s3:::<bucket_name>/*",
                "arn:aws:s3:::<bucket_name>"
            ]
        }
    ]
}
```

詳細モードのマッピングの場合、同じ AWS リージョン内で異なる AWS アカウントを使用できます。これらのマッピングで使用する AWS アカウントへのアクセスが Amazon IAM ポリシーで承認されていることを確認します。

**注: ［テスト接続］**はユーザーに割り当てられた IAM ポリシーを検証しません。Amazon S3 バケット名は、ソースおよびターゲットの詳細プロパティで利用できます。

## IAM 認証

オプション。Secure Agent が Amazon Elastic Compute Cloud（EC2）システムにインストールされている場合は IAM 認証を設定できます。マッピングおよびマッピングタスクを実行する場合に、Amazon S3 ステージングバケットへのアクセスを安全に制御するために IAM 認証を使用します。

EC2 システムにインストールされた Secure Agent でマッピングおよびマッピングタスクを実行する場合、IAM 認証を実行します。

IAM 認証を設定するには、次の手順を実行します。

1. 最小の Amazon IAM ポリシーを作成します。詳細については、「最小限の Amazon IAM ポリシーの作成」(ページ 26)を参照してください。
2. ロールの作成。
   - Amazon EC2 ロールを作成します。EC2 ロールの作成中に、最小の Amazon IAM ポリシーを関連付けます。Amazon EC2 ロールは、EC2 インスタンスを作成または起動する場合に使用されます。Amazon EC2 ロールの作成の詳細については、AWS のマニュアルを参照してください。
   - Amazon Redshift リソースへの安全なアクセスのために、Amazon Redshift ロールの ARN を作成します。Amazon Redshift ロールの作成中に、最小の Amazon IAM ポリシーを関連付けます。Amazon Redshift ロールの ARN の作成の詳細については、AWS のマニュアルを参照してください。

     **注:** UNLOAD と COPY コマンドで、同じ Amazon Redshift ロールの ARN を使用します。
3. 手順 2 で作成した Amazon EC2 ロールを EC2 インスタンスに割り当てます。
4. UNLOAD コマンドと COPY コマンドを使用して正常に読み取りと書き込み操作を実行するには、Amazon Redshift ロールの ARN を Amazon Redshift クラスタに割り当てます。Amazon Redshift クラスタへの Amazon Redshift ロールの ARN の追加の詳細については、AWS のマニュアルを参照してください。
5. EC2 インスタンスに Secure Agent をインストールします。

**注:** サーバーレスランタイム環境を使用する場合、IAM 認証を設定することはできません。

## AssumeRole を使用した一時的なセキュリティ資格情報

AssumeRole を使用した一時的なセキュリティ資格情報を使用すると、同じ AWS アカウントまたは異なる AWS アカウントから Amazon S3 ステージングバケットにアクセスできます。

**sts:AssumeRole** 権限を持っていて、AWS アカウント内に一時的なセキュリティ資格情報を使用するための信頼関係が構築されていることを確認します。信頼関係は、IAM ロールを作成するときに、IAM ロールの信頼ポリシーで定義されます。IAM ロールは、IAM ユーザーを信頼されたエンティティとして追加し、IAM ユーザーに一時的なセキュリティ資格情報の使用と AWS アカウントへのアクセスを許可します。信頼関係を構築する方法の詳細については、AWS のマニュアルを参照してください。

信頼された IAM ユーザーが一時的なセキュリティ資格情報を要求すると、AWS Security Token Service（AWS STS）が、指定した期間有効な一時的なセキュリティ資格情報を動的に生成し、その資格情報を信頼された IAM ユーザーに提供します。一時的なセキュリティ資格情報は、アクセスキー ID、シークレットアクセスキー、シークレットトークンで構成されます。

動的に生成された一時的なセキュリティ資格情報を使用するには、Amazon Redshift V2 接続を作成するときに［**IAM ロール ARN**］接続プロパティの値を入力します。IAM ロール ARN は、AWS リソースを一意に識別します。次に、［**一時的な資格情報の期間**］詳細ソースプロパティおよびターゲットプロパティで、一時的なセキュリティ資格情報を使用できる期間を秒単位で指定します。

## 外部 ID

Amazon S3 バケットが別の AWS アカウントにある場合に、Amazon S3 バケットへのより安全なアカウント間アクセスのために外部 ID を指定できます。

オプションで、AWS Security Token Service（STS）への AssumeRole 要求で外部 ID を指定できます。

外部 ID は文字列である必要があります。

次のサンプルは、想定される IAM ロールの信頼ポリシーの外部 ID 条件を示しています。

```
"Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AWS_Account_ID : user/user_name"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "dummy_external_id"
        }
      }
    }
  ]
```

## 一時的なセキュリティ資格情報のポリシー

一時的なセキュリティ資格情報を使用して Amazon S3 ステージングバケットにアクセスするには、IAM ユーザーと IAM ロールにポリシーが必要です。

次のセクションに、IAM ユーザーと IAM ロールに必要なポリシーを示します。

IAM **ユーザー**

IAM ユーザーは、同じ AWS アカウントまたは異なる AWS アカウントで一時的なセキュリティ資格情報を使用するために、`sts:AssumeRole` ポリシーを持っている必要があります。

次のサンプルポリシーでは、IAM ユーザーに AWS アカウントで一時的なセキュリティ資格情報を使用することを許可しています。

```
{
"Version":"2012-10-17", "Statement":{ "Effect":"Allow", "Action":"sts:AssumeRole",
"Resource":"arn:aws:iam::<ACCOUNT-HYPHENS>:role/<ROLE-NAME>" }
}
```

次のサンプルポリシーでは、中国地域の IAM ユーザーに AWS アカウントで一時的なセキュリティ資格情報を使用することを許可しています。

```
{
"Version":"2012-10-17", "Statement":{ "Effect":"Allow", "Action":"sts:AssumeRole", "Resource":"arn:aws-
cn:iam::<ACCOUNT-HYPHENS>:role/<ROLE-NAME>" }
}
```

IAM **ロール**

IAM ロールには、IAM ユーザーに対して一時的なセキュリティ資格情報を使用した Amazon S3 バケットへのアクセスを許可するために、`sts:AssumeRole` ポリシーと IAM ロールにアタッチされた信頼ポリシーが必要です。このポリシーは、IAM ユーザーがアクセスできる Amazon S3 バケットと、IAM ユーザーが実行できるアクションを指定します。信頼ポリシーは、Amazon S3 バケットにアクセスできる AWS アカウントの IAM ユーザーを指定します。

次のポリシーは、サンプルの信頼ポリシーです。

```
{
"Version":"2012-10-17", "Statement":[{ "Effect":"Allow", "Principal":{ "AWS":"arn:aws:iam::AWS-account-
```

```
ID:root" },
"Action":"sts:AssumeRole" }
 ]
  }
   }
```

ここで、Principal 属性には、IAM ユーザーの ARN を指定することもできます。ARN はそれ以上のアクセスを制限します。IAM ユーザーは、動的に生成された一時的なセキュリティ資格情報を使用できます。以下に例を示します。

```
"Principal" : { "AWS" : "arn:aws:iam:: AWS-account-ID :user/ user-name " }
```

AWS Key Management Service（AWS KMS）で管理された顧客マスタキーを使用する一時的なセキュリティ資格情報を使用し、KMS を使用した暗号化を有効にするには、KMS ポリシーを作成する必要があります。

次の操作を実行すると、一時的なセキュリティ資格情報を使用し、KMS を使用した暗号化を有効にすることができます。

- GenerateDataKey
- DescribeKey
- 暗号化
- 復号化
- ReEncrypt

サンプルポリシー:

```
{
"Version":"2012-10-17", "Statement":[{ "Effect":"Allow", "Action":[ "kms:GenerateDataKey", "kms:DescribeKey",
"kms:Encrypt", "kms:Decrypt", "kms:ReEncrypt*" ], "Resource": ["arn:aws:kms:region:account:key/<KMS_key>"]
    }
  ]
   }
```

KMS を設定し、中国地域の Amazon S3 エンドポイントにアクセスする場合は、次のサンプルポリシーを使用します。

```
{
"Version":"2012-10-17", "Statement":[{ "Effect":"Allow", "Action":[ "kms:GenerateDataKey", "kms:DescribeKey",
"kms:Encrypt", "kms:Decrypt", "kms:ReEncrypt*" ], "Resource": ["arn:aws-cn:kms:region:account:key/<KMS_key>"]
    }
  ]
   }
```

## AssumeRole for EC2 を使用した一時的なセキュリティ認証情報

AssumeRole を使用して一時的なセキュリティ認証情報を使用し、Amazon EC2 ロールが同じ AWS アカウントまたは異なる AWS アカウントから Amazon S3 ステージングバケットにアクセスすることができます。

Amazon EC2 ロールは、永続的なアクセスキーと秘密鍵を必要とせずに、同じまたは異なる AWS アカウントから別の IAM ロールを引き受けることができます。Amazon EC2 ロールは、別のリージョンから別の IAM ロールを引き受けることもできます。

AssumeRole for EC2 を使用して一時セキュリティ資格情報を使用する場合は、次の前提条件を考慮してください。

- AssumeRole for EC2 を使用して一時的なセキュリティ認証情報を使用するには、Amazon EC2 などの AWS サービスに Secure Agent をインストールします。
- AWS EC2 サービスにアタッチされた EC2 ロールは、Amazon S3 へのアクセス権を持っていてはいけませんが、別の IAM ロールを引き受ける権限が必要です。

- EC2 ロールが引き受ける必要のある IAM ロールには、アクセス許可ポリシーと信頼ポリシーがアタッチされている必要があります。
  EC2 ロールおよび IAM ロールのポリシーの詳細については、「一時的なセキュリティ資格情報のポリシー」(ページ 28)を参照してください。

**[IAM ロール ARN]** 接続プロパティで提供される IAM ロールを引き受けるように EC2 ロールを設定するには、接続プロパティで **[ロールの引き受けに EC2 ロールを使用]** チェックボックスを選択します。

## 一時セキュリティ資格情報の使用に関するルールおよびガイドライン

一時セキュリティ資格情報の使用については、次のガイドラインを参照してください。

- 一時的なセキュリティ認証情報を要求する IAM ユーザーまたは IAM ロールは、AWS リソースへのアクセス権を持っていてはいけません。
- 認証済み IAM ユーザーまたは IAM ロールのみ、AWS Security Token Service（AWS STS）から一時セキュリティ資格情報を要求できます。
- タスクを実行する前に、一時セキュリティ資格情報の有効期間が、そのタスクを実行するのに十分な長さであることを確認してください。実行中のタスクに対して、一時セキュリティ資格情報の有効期間を延長することはできません。例えば、Amazon Redshift との間で読み取りと書き込みを行い、一時的なセキュリティ認証情報の有効期限が切れた場合、一時的なセキュリティ認証情報の期間を延長することはできないので、タスクが失敗する原因となります。
- 一時セキュリティ資格情報の有効期限が切れると、AWS は、その資格情報を使ったリソースへのアクセスを IAM ユーザーまたは IAM ロールに許可しません。マッピングで現在使用している一時セキュリティ資格情報の有効期限が切れる前に、新しい一時セキュリティ資格情報を要求する必要があります。
- 詳細モードのマッピングの場合、一時セキュリティ資格情報は、**[一時資格情報の期間]** の詳細ソースプロパティおよびターゲットプロパティで設定された時間が経過した後でも期限切れになりません。
- 一時セキュリティ資格情報を使用するのに、AWS アカウントのルートユーザー資格情報を使用しないでください。一時セキュリティ資格情報を使用するには、IAM ユーザーの資格情報を使用する必要があります。
- IAM ロール ARN を使用して Amazon Redshift V2 接続を作成し、SSE-KMS 暗号化を使用する場合、Amazon Redshift V2 の詳細ソースプロパティでアンロードオプションとして AWS_IAM_ROLE を指定する必要があります。
- マッピング内のソースとターゲットの両方が同じ Amazon S3 バケットを指している場合は、ソーストランスフォーメーションとターゲットトランスフォーメーションで同じ Amazon S3 接続が使用されます。異なる 2 つの Amazon S3 接続を使用する場合は、両方の接続に対して接続プロパティで同じ値を設定します。
- マッピング内のソースとターゲットが異なる Amazon S3 バケットを指している場合、異なる 2 つの Amazon S3 接続を使用できます。
  両方の接続に対して、接続プロパティで異なる値を設定できます。ただし、接続プロパティで **[ロールの引き受けに EC2 ロールを使用]** チェックボックスをオンにする必要があります。また、ソースプロパティとターゲットプロパティの **[一時的な資格情報の期間]** フィールドにも同じ値を指定する必要があります。

# Amazon Redshift Spectrum の前提条件タスク

Amazon Redshift Spectrum の外部テーブルからデータを読み取るには、Amazon S3 内のデータカタログとデータファイルにアクセスするために、Amazon Redshift クラスタに必要な権限を与える必要があります。

1. AWS Identity and Access Management（IAM）ロールを作成して、Amazon S3 内の外部データカタログとデータファイルへの Amazon Redshift クラスタのアクセスを承認します。
2. 指定した Amazon Redshift クラスタに IAM ロールを関連付けます。
3. 外部スキーマを作成します。
4. 外部スキーマ内で、IAM ロールに Amazon Redshift ロールの ARN を指定します。

5. 外部スキーマ内に外部テーブルを作成し、データの読み取り元に Amazon S3 の場所を指定します。外部テーブルの作成の詳細については、AWS のマニュアルを参照してください。

**注:** Amazon Redshift クラスタとデータファイルを含む Amazon S3 バケットは、同じリージョンに属している必要があります。Amazon Redshift クラスタは、バージョン 1.0.1294 以降である必要があります。

# マッピングのステージングパフォーマンスの最適化

データ統合は、デフォルトで、Amazon Redshift から読み書きする前にデータのステージング用として一時フォルダにフラットファイルをローカルに作成します。ステージングのパフォーマンスを最適化するようにデータ統合を設定できます。

ステージングプロパティを設定しない場合、データ統合は最適化された設定なしでステージングを実行しますが、これがタスクのパフォーマンスに影響する場合があります。

ステージングプロパティを有効にする場合は、次のルールを考慮してください。

- プッシュダウンの最適化が有効になっているマッピングを実行すると、プッシュダウンの最適化なしでマッピングが実行されます。
- 読み取り操作の場合、ステージングはソーストランスフォーメーションにのみ適用され、ルックアップトランスフォーメーションには適用されません。
- データにタイムゾーン付きの Timestamp データ型が含まれている場合、ローカルフラットファイルでデータをステージングせずにジョブが実行されます。
- マッピングにソースとして Oracle CDC が含まれ、ターゲットとして Amazon Redshift が含まれている場合、ローカルフラットファイルでデータをステージングせずにジョブが実行されます。

## ステージングパフォーマンスを最適化するための Amazon Redshift Connector の有効化

次のタスクを実行して、読み取り操作用の `INFA_DTM_RDR_STAGING_ENABLED_CONNECTORS`、および Secure Agent プロパティの書き込み操作用の `INFA_DTM_STAGING_ENABLED_CONNECTORS` というステージングプロパティを設定します。

1. Administrator で、**[ランタイム環境]** をクリックします。
2. このプロパティを設定する Secure Agent を編集します。
3. **[システム構成の詳細]** セクションで、**[データ統合サーバー]** に **[サービス]** を選択し、タイプに **[Tomcat]** を選択します。

4. Amazon Redshift コネクタのプラグイン ID に、読み取り操作用の Tomcat プロパティ *INFA_DTM_RDR_STAGING_ENABLED_CONNECTORS* と書き込み操作用の *INFA_DTM_STAGING_ENABLED_CONNECTORS* の値を設定します。
プラグイン ID は、次のディレクトリのマニフェストファイルの中にあります: `<Secure Agent のインストールディレクトリ>/<AmazonRedshift パッケージ>/CCIManifest`

次の図は、読み取り操作のために Secure Agent に設定されたステージングプロパティを示しています。

| | | |
|---|---|---|
| Tomcat | INFA_DTM_RDR_STAGING_ENABLED_CONNECTORS | '451600' |

マッピングを実行すると、マシンの次のディレクトリにフラットファイルが作成されます: `C:\Windows\Temp\AmazonRedshift\stage\<AmazonRedshift_Target.txt>`

次の図は、書き込み操作用に Secure Agent で設定されたステージングプロパティを示しています。

| | | |
|---|---|---|
| Tomcat | INFA_DTM_STAGING_ENABLED_CONNECTORS | '605501' |

セッションログを確認します。フラットファイルが正常に作成されると、データ統合は次のメッセージをセッションログに記録します。INFA_DTM_STAGING が正常に有効化され、フラットファイルを使用してローカルステージングファイルが作成されます。また、INFA_DTM_STAGING が、ソースインスタンス[Source]のコネクタに対して有効になります。

# Amazon Redshift とのプライベート通信

Amazon Redshift とのプライベート通信は、AWS コンソールでゲートウェイエンドポイントを設定することで有効にできます。

トラフィックをパブリックインターネットに公開せずに Amazon Redshift とのプライベート通信を確立するように Amazon Redshift V2 コネクタを設定できます。Amazon Redshift とのプライベート接続を確立するには、Secure Agent が AWS Virtual Private Cloud（VPC）のサブネットの一部であることを確認します。ゲートウェイエンドポイントを作成し、Amazon S3 データを Amazon Redshift にステージングできます。

Amazon Redshift に接続するようにプライベート通信を設定するには、クラスタサブネットグループを作成し、Redshift が管理する VPC エンドポイントを作成し、ゲートウェイエンドポイントを設定してから、Amazon Redshift V2 接続プロパティでゲートウェイエンドポイントを設定します。

詳細については、
Configuring private communication with Amazon Redshift using the Amazon Redshift V2 Connector を参照してください。

# 第 5 章

# Amazon Redshift V2 接続

Amazon Redshift V2 接続を使用すると、Amazon Redshift とのデータの読み取りまたは書き込みを行うことができます。Amazon Redshift V2 接続を使用すると、マッピングおよびマッピングタスクでソースまたはターゲットを指定できます。

AWS Identity and Access Management（IAM）認証を使用すると、Amazon S3 リソースへのアクセスを安全に制御できます。有効な AWS 資格情報があり、IAM 認証を使用する場合は、Amazon Redshift V2 接続を作成するときに、アクセスキーとシークレットキーを指定する必要はありません。

**[接続]** ページで Amazon Redshift V2 接続を作成し、マッピングまたはマッピングタスクに関連付けます。Amazon Redshift に対してデータの読み取りまたは書き込みを行うために、ソースおよびターゲットのプロパティを定義します。

**注:** HTTP プロキシと SOCKS プロキシの両方を有効にすると、デフォルトでは SOCKS プロキシが使用されます。SOCKS プロキシの代わりに HTTP プロキシを使用する場合は、システムプロパティで **[DisableSocksProxy]** プロパティの値を true に設定します。

## Amazon Redshift V2 接続のプロパティ

Amazon Redshift V2 接続をセットアップする際には、接続プロパティを設定します。

次の表に、Amazon Redshift V2 接続プロパティを示します。

| プロパティ | 説明 |
|---|---|
| 接続名 | 接続の名前。<br>各接続名は組織内で一意である必要があります。接続名には、英数字、スペース、および次の特殊文字を含めることができます。_ .+ -,<br>最大長は 255 文字です。 |
| 説明 | 接続の説明。最大長は 4000 文字です。 |
| タイプ | Amazon Redshift V2 接続タイプ。 |
| ランタイム環境 | タスクを実行するランタイム環境の名前。<br>Secure Agent、Hosted Agent、またはサーバーレスランタイム環境を選択します。<br>**注**: Hosted Agent は、詳細クラスタで実行されるマッピングには適用されません。 |
| ユーザー名 | Amazon Redshift アカウントのユーザー名。 |

| プロパティ | 説明 |
| --- | --- |
| パスワード | Amazon Redshift アカウントのパスワード。 |
| アクセスキー ID | Amazon S3 ステージングバケットにアクセスするためのアクセスキー。<br>次の認証方法に基づいてアクセスキー値を入力します。<br>- 基本認証。実際のアクセスキー値を入力します。<br>- IAM 認証。アクセスキー値は入力しないでください。<br>- ロールの引き受けを使用した一時的なセキュリティ資格情報。Amazon S3 ステージングバケットへのアクセス権なしで、IAM ユーザーのアクセスキーを入力します。<br>- EC2 用のロールの引き受け。アクセスキー値は入力しないでください。 |
| シークレットアクセスキー | Amazon S3 ステージングバケットにアクセスするためのシークレットアクセスキー。<br>秘密鍵はアクセスキーに関連付けられており、アカウントを一意に識別します。<br>次の認証方法に基づいてアクセスキー値を入力します。<br>- 基本認証。実際のアクセスシークレット値を入力します。<br>- IAM 認証。アクセスシークレット値は入力しないでください。<br>- ロールの引き受けを使用した一時的なセキュリティ資格情報。Amazon S3 ステージングバケットへのアクセス権なしで、IAM ユーザーのアクセスシークレットを入力します。<br>- EC2 用のロールの引き受け。アクセスシークレット値は入力しないでください。 |
| IAM ロール ARN | 動的に生成された一時的なセキュリティ資格情報を使用するためにユーザーに引き継がれた IAM ロールの Amazon Resource Number（ARN）。<br>一時的なセキュリティ資格情報を使用して Amazon S3 ステージングバケットにアクセスする場合はこのプロパティの値を設定します。<br>IAM ロールの ARN の取得方法の詳細については、AWS のマニュアルを参照してください。 |
| 外部 ID | Amazon S3 ステージングバケットが別の AWS アカウントにある場合に、Amazon S3 バケットへのより安全なアクセスを確立するための外部 ID。 |
| ロールの引き受けに EC2 ロールを使用 | オプション。チェックボックスを選択すると、EC2 ロールが IAM ロール ARN オプションで指定された別の IAM ロールを引き受けることができます。<br>**注**: EC2 ロールには、同じアカウントまたは異なるアカウントから IAM ロールを引き受けるためのアクセス許可がアタッチされたポリシーが必要です。 |
| マスタ対称キー [1] | クライアントサイド暗号化を有効にする場合の、Base64 形式で示す 256 ビットの AES 暗号化キー。暗号化キーは、サードパーティ製ツールを使用して生成できます。 |
| JDBC URL | Amazon Redshift V2 接続の URL。<br>JDBC URL は次の形式で入力します。<br>`jdbc:redshift://<amazon_redshift_host>:<port_number>/<database_name>` |

| プロパティ | 説明 |
| --- | --- |
| クラスタリージョン | アクセスするバケットが存在する AWS クラスタリージョンです。<br>［JDBC URL］接続プロパティで指定したカスタム JDBC URL にクラスタリージョン名が含まれていない場合にクラスタリージョンを選択します。<br>**［クラスタリージョン］**と［JDBC URL］の両方の接続プロパティでクラスタリージョンを選択した場合、［JDBC URL］接続プロパティで指定したクラスタリージョンは無視されます。<br>［JDBC URL］接続プロパティで指定したクラスタリージョン名を使用するには、このプロパティでクラスタリージョンとして**［なし］**を選択します。<br>AWS SDK によってサポートされるクラスタリージョンに対してのみ、データの読み取りと書き込みを行うことができます。<br>次のいずれかのクラスタリージョンを選択します。<br>- なし<br>- アジアパシフィック（ムンバイ）<br>- アジアパシフィック（ソウル）<br>- アジアパシフィック（シンガポール）<br>- アジアパシフィック（シドニー）<br>- アジアパシフィック（東京）<br>- アジアパシフィック（香港）<br>- AWS GovCloud（米国）<br>- AWS GovCloud（米国東部）<br>- カナダ（中部）<br>- 中国（北京）<br>- 中国（寧夏）<br>- 欧州（アイルランド）<br>- 欧州（フランクフルト）<br>- EU（パリ）<br>- EU（ストックホルム）<br>- 南米（サンパウロ）<br>- 中東（バーレーン）<br>- 米国東部（バージニア北部）<br>- 米国東部（オハイオ）<br>- 米国西部（北カリフォルニア）<br>- 米国西部（オレゴン）<br>デフォルトは**［なし］**です。 |
| 顧客マスタキー ID | AWS Key Management Service（AWS KMS）によって生成された顧客マスタキー ID、またはアカウント間アクセス用のカスタムキーの ARN。<br>Amazon S3 ステージングバケットが存在するリージョンのカスタママスタキー ID を生成する必要があります。顧客が生成した顧客マスタキー ID またはデフォルトの顧客マスタキー ID を入力できます。<br>詳細モードでマッピングを実行すると、同じリージョンに対してクロスアカウント KMS キーを使用できます。 |
| [1] 詳細モードのマッピングには適用されません。 | |

# プロキシの設定

組織で送信プロキシサーバーを使用してインターネットに接続する場合、Secure Agent は、そのプロキシサーバー経由で Informatica Intelligent Cloud Services に接続します。HTTP プロキシと SOCKS プロキシの両方を有効にすると、デフォルトでは SOCKS プロキシが使用されます。SOCKS プロキシの代わりに HTTP プロキシを使用する場合は、システムプロパティで**［DisableSocksProxy］**プロパティの値を true に設定します。

プロキシサーバーを設定するときは、次のルールを考慮してください。

- 認証されていないプロキシサーバーのみを使用して Informatica Intelligent Cloud Services に接続できます。
- サーバーレスランタイム環境を使用する場合、プロキシサーバーを使用して Informatica Intelligent Cloud Services に接続することはできません。
- プロキシ設定は、詳細モードのマッピングには適用されません。

# Windows でのプロキシ設定

Windows マシンで Secure Agent のプロキシサーバーの設定を行うには、Secure Agent を介してプロキシサーバー設定を構成します。

1. **［開始］** > **［すべてのプログラム］** > **［Informatica Cloud Secure Agent］** > **［Informatica Cloud Secure Agent］** をクリックして、Secure Agent Manager を起動します。

   Secure Agent Manager に Secure Agent のステータスが表示されます。
2. ［Secure Agent Manager］ページで **［プロキシ］** をクリックします。
3. **［プロキシサーバーを使用］** をクリックして、プロキシサーバーの設定を入力します。
4. 次のプロキシサーバーの詳細を設定します。

| フィールド | 説明 |
|---|---|
| プロキシホスト | 必須。Secure Agent が使用する送信プロキシサーバーのホスト名。 |
| プロキシポート | 必須。送信プロキシサーバーのポート番号。 |

5. **［OK］** をクリックします。

   Secure Agent Manager によって Secure Agent が再起動され、設定が適用されます。

## Secure Agent Manager を使用したプロキシ設定

Secure Agent Manager を使用してプロキシサーバーを設定するには、次の手順を実行します。

1. **［開始］** > **［すべてのプログラム］** > **［Informatica Cloud Secure Agent］** > **［Informatica Cloud Secure Agent］** をクリックして、Secure Agent Manager を起動します。

   Secure Agent Manager に Secure Agent のステータスが表示されます。
2. ［Secure Agent Manager］ページで **［プロキシ］** をクリックします。
3. **［プロキシサーバーを使用］** をクリックして、プロキシサーバーの設定を入力します。
4. 次のプロキシサーバーの詳細を設定します。

| フィールド | 説明 |
|---|---|
| プロキシホスト | 必須。Secure Agent が使用する送信プロキシサーバーのホスト名。 |
| プロキシポート | 必須。送信プロキシサーバーのポート番号。 |

5. **［OK］** をクリックします。

   Secure Agent Manager によって Secure Agent が再起動され、設定が適用されます。

## JVMOption を使用したプロキシ設定

Windows または Linux マシンの Secure Agent の JVMOptions を使用して、プロキシサーバー設定を構成できます。

1. Informatica Intelligent Cloud Services にログインします。
2. ［管理］を開いて［**ランタイム環境**］を選択します。
3. プロキシサーバーを設定する Secure Agent を選択します。
4. ページの右上隅にある［**編集**］をクリックします。
5. ［**システム構成の詳細**］セクションで、データ統合サービスの［**タイプ**］に［**DTM**］を選択します。
   - **JVMOption** フィールドに次のパラメータを追加し、各パラメータに適した値を指定します。

| パラメータ | 説明 |
|---|---|
| -Dhttps.proxyHost= | 送信 HTTPS プロキシサーバーのホスト名。 |
| -Dhttps.proxyPort= | 送信 HTTPS プロキシサーバーのポート番号。 |

   以下に例を示します。

   ```
   JVMOption1=-Dhttps.proxyHost=<proxy_server_hostname>
   JVMOption2=-Dhttps.proxyPort=<proxy_port>
   ```

6. ［**保存**］をクリックします。

   Secure Agent が再起動して設定が適用されます。

# Linux でのプロキシ設定

Secure Agent のインストーラにより、ブラウザで設定されている設定項目に基づいて Secure Agent のプロキシサーバー設定が設定されます。Secure Agent に定義されているプロキシサーバーの設定は、コマンドラインから更新できます。

Linux マシンで Secure Agent のプロキシサーバーを設定するには、`proxy.ini` ファイルを更新するシェルコマンドを使用します。ネットワーク管理者に問い合わせて、プロキシの設定項目を決めてください。

1. 次のディレクトリに移動します。

   ```
   <Secure Agent のインストールディレクトリ>/apps/agentcore
   ```

2. `proxy.ini` ファイルを更新します。
   - 認証されていないプロキシの `proxy.ini` ファイルを更新するには、次のコマンドを入力します。

     ```
     ./consoleAgentManager.sh configureProxy <proxy host> <proxy port>
     ```

3. Secure Agent を再起動します。

# 第 6 章

# Amazon Redshift ソースとターゲット

Amazon Redshift 接続のソースとターゲットのプロパティを割り当てることができます。

## Amazon Redshift V2 ソース

マッピングのソースとして Amazon Redshift V2 オブジェクトを使用できます。テーブル、ビュー、またはマテリアライズビューを Amazon Redshift V2 ソースとして使用できます。

詳細ソースプロパティを設定するとき、Amazon Redshift V2 固有のプロパティを設定します。データを暗号化し、Amazon S3 にステージングファイルを保持して、Amazon Redshift でファイルにクエリの結果を安全にアンロードできます。

### Amazon Redshift V2 ソースの Amazon Redshift ステージングディレクトリ

Secure Agent は、ソースプロパティで指定したディレクトリにステージングファイルを作成します。Secure Agent は、Amazon Redshift V2 からデータを読み取り、そのデータを Amazon S3 に書き込んで、次にローカルステージングディレクトリに書き込みます。次に、Secure Agent は、ステージングされたファイルをステージングディレクトリから削除します。

マッピングプロパティで、処理するデータの容量に応じた適切な容量のディスク領域を持つステージングディレクトリを指定します。Secure Agent をホストするマシン上のディレクトリを指定します。

### Amazon Redshift V2 ソースのデータ暗号化

データを保護するために、ソースからデータを読み取るときにデータを暗号化することができます。

［**スケジュール**］ページの Amazon Redshift V2 の詳細ソースプロパティにある［**暗号化タイプ**］フィールドで、暗号化のタイプを選択します。Unload コマンドは、AWS が管理する暗号化キーと AWS Key Management Service キーを使用して、サーバーサイド暗号化用に Amazon S3 に一時ファイルを作成します。

サーバーサイド暗号化用に Unload コマンドで AWS Key Management Service によって生成された顧客マスタキー ID を使用します。

以下のタイプの暗号化を選択できます。

**なし**

データは暗号化されません。

SSE-S3

**[SSE-S3]** 暗号化タイプを選択した場合、Unload コマンドは Amazon S3 バケットに一時ファイルを作成し、Amazon S3 はサーバーサイド暗号化用の AWS が管理する暗号化キーを使用してファイルを暗号化します。

SSE-KMS

**[SSE-KMS]** 暗号化タイプを選択した場合、Unload コマンドは Amazon S3 バケットにステージングファイルを作成し、Amazon S3 はサーバーサイド暗号化用の AWS KMS が管理する顧客マスタキーまたは Amazon リソースネーム（ARN）を使用してファイルを暗号化します。

接続プロパティで指定した AWS KMS が管理する顧客マスタキーまたは ARN は、Amazon S3 をホストする地域と同じ地域に属する必要があります。

例えば、Amazon S3 が **[米国西部（オレゴン)]** 地域でホストされている場合、**[SSE-KMS]** 暗号化タイプを選択すると、同じ地域で有効にされた AWS KMS が管理する顧客マスタキーを使用する必要があります。

CSE-SMK

**[CSE-SMK]** 暗号化タイプを選択した場合、Amazon Redshift はマスタ対称キーを使用して Amazon S3 サーバーにデータをアップロードし、[暗号化] オプションを設定した copy コマンドおよび追加のセキュリティ用のプライベート暗号化キーを使用して、データをロードします。

**[CSE-SMK]** 暗号化タイプを有効にするには、接続プロパティでマスタ対称キー ID を指定する必要があります。

**注:** Amazon Redshift V2 コネクタは、マスタ対称キーを使用するサーバーサイド暗号化と、顧客マスタキーを使用するクライアントサイド暗号化をサポートしていません。

# Unload コマンド

Unload コマンドを使用すると、Amazon Redshift からデータを抽出し、Amazon S3 でステージングファイルを作成できます。Unload コマンドはセキュアな接続を使用して、Amazon S3 上の 1 つ以上のファイルにデータをロードします。

Unload コマンドオプションを **[Unload オプション]** フィールドで直接指定できます。オプションは大文字で入力し、オプションの区切りにはセミコロンを使用します。以下に例を示します。

```
DELIMITER = \036;ESCAPE = OFF;NULL=text;PARALLEL = ON;AWS_IAM_ROLE=arn;aws;iam;;<account ID>;role/<role-name>
```

**注:** NULL Unload コマンドオプションは、詳細モードのマッピングには適用されません。

印刷できない文字の 8 進数の表現を DELIMITER として使用することを推奨します。

pre-SQL コマンドまたは post-SQL コマンドとして Unload コマンドを実行する場合は、`ALLOWOVERWRITE` オプションを指定して既存のオブジェクトを上書きします。

デフォルトでは、UNLOAD プロパティフィールドは空です。

## Unload コマンドのオプション

Unload コマンドのオプションは Amazon Redshift からデータを抽出し、Amazon S3 のステージングファイルに特定の形式でデータをロードします。特定の文字を使用してデータを区切ったり、複数のファイルにデータを並行して読み込んだりできます。

Unload コマンドにオプションを追加するには、**[Unload オプション]** オプションを使用します。

以下のオプションを設定できます。

DELIMITER

入力ファイル内のフィールドを区切るための単一の ASCII 文字。パイプ（|）、チルダ（~）、タブ（\t）などの文字を使用できます。指定した区切り文字をデータに含めることはできません。区切り文字がデータに含まれる場合、ESCAPE を使用すると、区切り文字が通常の文字として読み取られます。デフォルト値は\036 で、印刷できない 8 進数のレコード区切り文字です。

ESCAPE

アンロードされたデータの区切り文字が指定される前に、区切りアンロードファイルに CHAR カラムと VARCHAR カラムのエスケープ文字を追加できます。デフォルトでは、この ESCAPE オプションは **ON** になっています。ESCAPE オプションを無効にするには、ESCAPE オプションの値として **OFF** を指定します。例えば、`ESCAPE = OFF` と指定します。

**注:** DTM ステージングプロパティを有効にする場合は、Unload コマンドオプションで `ESCAPE=OFF` を使用して、読み取り DTM ステージングを有効にする必要があります。`ESCAPE = OFF` を指定しない場合、マッピングでは DTM ステージングプロパティが考慮されず、ステージングパフォーマンスの最適化を使用せずに実行されます。

REGION

Amazon S3 ステージングバケットがクラスタ領域と同じ領域にない場合は、REGION 属性を使用できます。Amazon Redshift が米国東部（バージニア北部）にある場合は、アジアパシフィック（ムンバイ）領域にある Amazon S3 バケットを使用してステージングファイルを作成することができます。例： `REGION = ap-south-1`

PARALLEL

Unload コマンドは、クラスタ内のスライス数に従ってデータを複数のファイルに並行して書き込みます。デフォルトはオンです。並行オプションをオフにすると、Unload コマンドはデータを順番に書き込みます。データファイルの最大サイズは 6.5 GB です。

NULL

NULL Unload コマンドオプションを使用することによって、Amazon Redshift ソーステーブルの NULL 値を、NULL Unload コマンドオプションを使用して指定した文字列と置き換えることができます。

NULL Unload コマンドオプションの値を次の形式で入力します: `NULL=text` 文字列値の入力時に、スペースを追加しないでください。NULL Unload コマンドの詳細については、AWS のマニュアルを参照してください。

**注:** NULL Unload コマンドオプションは、詳細モードのマッピングには適用されません。

AWS_IAM_ROLE

Amazon EC2 システムにインストールされた Secure Agent でマッピングを実行する Amazon Redshift ロールリソース名（ARN）を次の形式で指定します。`AWS_IAM_ROLE=arn:aws:iam::<account ID>:role/<role-name>`

例: `arn:aws:iam::123123456789:role/redshift_read`

ADDQUOTES

ADDQUOTES は、デフォルトでは UNLOAD コマンドで実行されます。Unload コマンドは各データフィールドに引用符を追加します。引用符を追加すると、UNLOAD コマンドは区切り文字を含むデータ値を読み取ることができます。二重引用符（"）がデータに含まれる場合、ESCAPE を使用すると、区切り文字が通常の文字として読み取られます。

## ソースパーティション

Amazon Redshift からデータを読み取る際には、パーティション化を設定して実行時のマッピングのパフォーマンスを最適化できます。パーティションタイプによって、Secure Agent がパーティションポイントでパーテ

ィションにデータを分散する方法を制御します。詳細モードでマッピングを実行する場合、ソースのパーティショニングを設定することはできません。

キー範囲パーティション化などのパーティションタイプを定義できます。1 つまたは複数のフィールドの値に基づいて Amazon Redshift データをパーティション化するようにキー範囲パーティション化を設定します。キー範囲パーティション化により、Secure Agent はパーティションキーとして定義したフィールドに基づいてソースデータ行を分散します。Secure Agent は、フィールド値を各パーティションの範囲値と比較して、適切なパーティションに行を送信します。

データ値が均等に分散されているカラムには、キー範囲パーティション化を使用します。そうしないと、パーティションのサイズが等しくなくなる可能性があります。例えば、あるカラムのキー値 1 から 1000 の間には 10 行あり、キー値 1001 から 2000 の間には 999 行があるというようになります。

キー範囲パーティション化では、あるパーティションのクエリが別のパーティションよりも先に行を返す場合があります。または、あるパーティションは行を返し、他のパーティションは行を返さない場合もあります。テーブル内の行がキー範囲と同じ順序である場合にこのようなことが起こります。あるクエリは行を読み取って行を返し、他のクエリは同じ行を読み取ってフィルタすることもあります。

**注:** データ型が Integer や String のパーティションキーのみを設定できます。

マッピングで複数のパーティションを設定する場合、Secure Agent は、最初のパーティションの開始範囲および最後のパーティションの終了範囲で指定した値を無視します。Secure Agent は、最初のパーティションの開始範囲で 10 未満の値、および最後のパーティションの終了範囲で最後のパーティションに指定した値を超える値を使用します。

例えば、マッピングで 3 つのパーティションを設定し、最初のパーティションの開始範囲の値に 5、最後のパーティションの終了範囲の値に 90 を指定した場合、マッピングは正常に実行されます。ただし、Secure Agent は指定した値を無視し、最初のパーティションの開始範囲の値で 10 未満の値、および最後のパーティションの終了範囲の値で 90 を超える値を使用します。

**注:** ソースのキー範囲パーティション化を設定してターゲットを作成すると、マッピングタスクは失敗します。

ソースのパーティション化を使用する場合、フィルタを設定することはできません。

# Amazon Redshift V2 ターゲット

マッピングまたはマッピングタスクのターゲットとして Amazon Redshift V2 オブジェクトを使用できます。入力ソースに基づいて Amazon Redshift V2 ターゲットを作成することもできます。

詳細ターゲットプロパティを設定するとき、Amazon Redshift V2 固有のプロパティを設定します。データを暗号化したり、クエリ効率向上のためにデータベーステーブルの統計メタデータを更新したり、Amazon S3 バケットのフラットファイルから Amazon Redshift にデータをロードしたり、ディスク領域の回復とテーブル内の行のソートのためにテーブルのクリーンアップを使用したりできます。

**注:** ターゲットテーブルの配布キーカラムに NULL 値が含まれていて、同じターゲットテーブルの更新/挿入操作を使用してタスクを設定すると、タスクにより重複する行が作成される場合があります。重複する行の作成を防ぐには、次のいずれかのタスクを実行する必要があります。

- データをロードするときに、NULL 値を NULL 以外の値に置き換えます。
- 配布キーカラム内に NULL 値を想定する場合は、カラムを配布キーとして設定しないようにします。
- データをロードするときに、ターゲットテーブルから配布キーカラムを一時的に削除します。Pre-SQL と Post-SQL のプロパティを使用して、ターゲットテーブルの配布キーカラムを削除してから、追加することができます。

# Amazon Redshift V2 ターゲットの Amazon Redshift ステージングディレクトリ

Secure Agent は、ターゲットプロパティで指定したディレクトリにステージングファイルを作成します。Secure Agent はデータを Amazon Redshift に書き込む前に、ステージングディレクトリに書き込みます。

Secure Agent は、データを Amazon S3 に書き込んだ後、ステージングされたファイルをステージングディレクトリから削除します。処理するデータのボリュームに最適なディスク領域を使用して、マッピングプロパティのステージングディレクトリを指定します。Secure Agent をホストするマシン上のディレクトリを指定します。

Secure Agent は、ステージングディレクトリにサブディレクトリを作成します。サブディレクトリでは次の命名規則を使用します。`<staging directory>/infaRedShiftStaging<MMddHHmmssSSS+xyz>`

# Amazon Redshift V2 ターゲットのデータ暗号化

データを保護するために、サーバーサイド暗号化またはクライアントサイド暗号化を有効にして、Amazon Redshift に挿入するデータを暗号化できます。

Amazon Redshift ターゲットでサーバーサイドとクライアントサイドの両方の暗号化を有効にすると、クライアントサイト暗号化がデータのロードに使用されます。

## Amazon Redshift V2 ターゲットのサーバーサイド暗号化

.csv ファイルを Amazon S3 にアップロードしてステージングするときに Amazon Redshift でデータを暗号化する場合は、サーバーサイド暗号化を有効にする必要があります。

サーバーサイド暗号化を有効にするには、詳細ターゲットプロパティで［**S3 サーバー側の暗号化**］を選択し、接続プロパティで［**顧客マスタキー ID**］を指定します。

サーバーサイド暗号化用に接続プロパティで AWS Key Management Service（AWS KMS）によって生成された顧客マスタキー ID を設定できます。IAM 認証と、顧客マスタキーを使用したサーバーサイド暗号化を使用する場合は、IAM EC2 ロールと IAM Redshift ロールを顧客マスタキーに追加する必要があります。

詳細ターゲットプロパティでサーバーサイド暗号化を選択し、接続プロパティで顧客マスタキー ID を指定しない場合、データの暗号化には Amazon S3 が管理する暗号化キーが使用されます。

## Amazon Redshift V2 ターゲットのクライアントサイド暗号化

クライアントサイド暗号化は、データを Amazon Redshift サーバーに送信する前にデータを暗号化する方法です。

Amazon Redshift V2 ターゲットに対してクライアントサイド暗号化を有効にすると、Secure Agent はソースからデータを取得し、そのデータをステージングディレクトリに書き込み、そのデータを暗号化してから Amazon S3 バケットに書き込みます。次に、Amazon S3 バケットは Amazon Redshift にデータを書き込みます。
**注:** サーバーレスランタイム環境を使用する場合、Amazon Redshift V2 ターゲットのクライアント側の暗号化を設定することはできません。 Amazon Redshift V2 ターゲットでサーバーサイドおよびクライアントサイドの両方の暗号化を有効にすると、クライアントサイト暗号化がデータのロードに使用されます。

クライアントサイド暗号化を有効にするには、接続プロパティでマスタ対称キーを指定し、詳細ターゲットプロパティで**[S3 クライアント側の暗号化]**を選択する必要があります。

Secure Agent は、マスタ対称キーを使用してデータを暗号化します。マスタ対称キーは Base64 形式の 256 ビット AES 暗号化キーです。Amazon Redshift V2 コネクタはマスタ対称キーを使用して Amazon S3 サーバーにデータをアップロードし、［暗号化］オプションを設定した copy コマンドと追加のセキュリティ用のプライベート暗号化キーを使用して、Amazon Redshift にデータをロードします。

# Copy コマンド

Copy コマンドを使用すると、テーブルにデータを追加できます。Copy コマンドはセキュアな接続を使用して、Amazon S3 バケット内のフラットファイルから Amazon Redshift にデータをロードします。

Copy コマンドオプションを［**コピーオプション**］フィールドで直接指定できます。オプションは大文字で入力し、オプションの区切りにはセミコロンを使用します。以下に例を示します。

```
DELIMITER = \036;ACCEPTINVCHARS = #;QUOTE = \037;COMPUPDATE = ON;AWS_IAM_ROLE=arn;aws;iam;;<account ID>;role/
<role-name>
```

印刷できない文字の 8 進数の表現を DELIMITER および QUOTE として使用することを推奨します。

## Copy コマンドのオプション

Copy コマンドオプションは、Amazon S3 からデータを読み取り、特定の形式で Amazon Redshift にデータを書き込みます。テーブル内のデータに圧縮を適用したり、特定の文字でデータを区切ることができます。

Copy コマンドにオプションを追加するには、［**CopyOptions プロパティファイル**］オプションを使用します。

以下のオプションを設定できます。

DELIMITER

入力ファイル内のフィールドを区切るための単一の ASCII 文字。パイプ（|）、チルダ（~）、タブ（\t）などの文字を使用できます。区切り文字をデータに含めることはできません。デフォルトは\036 で、印刷できない 8 進数のレコード区切り文字です。

ACCEPTINVCHARS

データに無効な UTF-8 文字が含まれていても、VARCHAR 列にデータをロードします。ACCEPTINCHARS を指定すると、Secure Agent は、ACCEPTINVCHARS で指定された文字からなる同じ長さの文字で無効な UTF-8 文字を置き換えます。ACCEPTINVCHARS で'|'を指定した場合、Secure Agent は 3 バイトの UTF-8 文字を'|||'に置き換えます。

ACCEPTINVCHARS を指定しない場合、COPY コマンドは、無効な UTF-8 文字が検出されるとエラーを返します。VARCHAR 列で ACCEPTINVCHARS オプションを使用できます。デフォルト疑問符（?）です。

QUOTE

カンマ区切り値で使用する引用符文字を指定します。デフォルトは\037 で、印刷できない 8 進数のユニット区切り文字です。

REGION

Amazon S3 ステージングバケットがクラスタ領域と同じ領域にない場合は、REGION 属性を使用できます。Amazon Redshift が米国東部（バージニア北部）にある場合は、アジアパシフィック（ムンバイ）領域にある Amazon S3 バケットを使用してステージングファイルを作成することができます。例: `REGION = ap-south-1`

COMPUPDATE

現在の圧縮エンコーディングをオーバーライドし、空のテーブルに圧縮を適用します。テーブル内の行が 10 万を超える場合は挿入操作で COMPUPDATE オプションを使用します。COMPUPDATE の動作は設定方法に応じて異なります。

- COMPUPDATE を指定しない場合、ターゲットテーブルが空で、テーブル内のすべての列で RAW またはエンコーディングが指定されないと、COPY コマンドは圧縮を適用します。
- COMPUPDATE を［オン］に指定した場合、ターゲットテーブルが空で、テーブル内の列で RAW 以外のエンコーディングが指定されると、COPY コマンドは既存のエンコーディングを置き換えます。
- COMPUPDATE を［オフ］に指定した場合、COPY コマンドは圧縮を適用しません。

デフォルトは OFF です。

TRUNCATECOLUMN

データをターゲットに書き込む前に、VARCHAR データ型および CHAR データ型のカラムのデータを切り詰めます。ターゲットに書き込むデータのサイズがターゲットカラムのサイズより大きい場合、Secure Agent はデータをターゲットカラムに書き込む前にデータを切り詰めます。

デフォルトでは、この TRUNCATECOLUMNS オプションは OFF になっています。TRUNCATECOLUMNS オプションを有効にするには、TRUNCATECOLUMNS オプションの値として ON を指定します。例えば、TRUNCATECOLUMNS=ON と指定します。

AWS_IAM_ROLE

Amazon EC2 システムにインストールされた Secure Agent でタスクを実行する Amazon Redshift ロールリソース名（ARN）を次の形式で指定します。`AWS_IAM_ROLE=arn:aws:iam::<account ID>:role/<role-name>`

例: `arn:aws:iam::123123456789:role/redshift_write`

## ターゲットテーブルの分析

クエリのパフォーマンスを最適化するには、ターゲットテーブルを分析するためのタスクを設定できます。ターゲットテーブル分析は、データベーステーブルの統計メタデータを更新します。

**［ターゲットテーブルの分析］**オプションを使用すると、テーブルからサンプル行を抽出し、サンプルを分析して、列の統計情報を保存できます。Amazon Redshift は、統計メタデータを使用してクエリプランナーを更新します。クエリプランナーは、統計メタデータを使用して、クエリ効率の向上に最適な計画を作成して選択します。

Copy コマンドを使用して既存のテーブルにデータをロードしたら、**［ターゲットテーブルの分析］**オプションを実行できます。新しいテーブルにデータをロードした場合、Copy コマンドはデフォルトで分析を実行します。

## ステージングファイルの保持

Secure Agent がターゲットにデータに書き込むと、Amazon S3 でステージングファイルを保持できます。ファイルを保持すると、Amazon S3 で組織データのデータレイクを作成できます。保存したファイルはデータのバックアップとしても使用できます。

ターゲット接続を作成する場合、ステージングファイルの保存用にファイルプレフィックスまたはディレクトリプレフィックスを設定できます。プレフィックスを指定すると、Secure Agent は、ターゲットで指定した Amazon S3 の場所にあるディレクトリ内にファイルを作成します。**［S3 でステージングファイルを保持するためのプレフィックス］**プロパティのいずれかのオプションを設定します。

- ディレクトリプレフィックスおよびファイルプレフィックスを指定します。例えば、`backup_dir/backup_file` です。Secure Agent は以下のディレクトリとファイルを作成します。

  - `backup_dir_<year>_<month>_<date>_<timestamp_inLong>`

  - `backup_file.batch_<batch_number>.csv.<file_number>.<encryption_if_applicable>`
- ファイルプレフィックスを指定します。例えば、backup_file です。Secure Agent は以下のディレクトリとファイルを作成します。

  - `<year>_<month>_<date>_<timestamp_inLong><3 digit of random number>00<ProcessID><PartitionId>`

  - `backup_file.batch_<batch_number>.csv.<file_number>.<encryption_if_applicable>`
- プレフィックスを指定しないでください。Secure Agent はステージングファイルを保存しません。

## テーブルのクリーンアップ

テーブルのクリーンアップを使用してディスク領域を回復し、指定したテーブルまたはデータベース内のすべてのテーブルの行をソートします。

削除やロードなどの一括操作を実行した後、または差分更新を実行した後、データベーステーブルをクリーンアップしてディスク領域を回復し、Amazon Redshift のクエリパフォーマンスを向上する必要があります。行の削除や更新を実行しても、Amazon Redshift は空き領域を再利用しません。

データベースまたはテーブルをクリーンアップすると、多くの場合、クエリパフォーマンスの一貫性を維持できます。データベース全体またはデータベース内の個々のテーブルのディスク領域を回復できます。データベースまたは指定されたデータベース管理スケジュールで最小限の操作を求める場合は、クリーンアップを実行する必要があります。クリーンアップを長く実行すると、データベース操作に影響を与える場合があります。大規模な未ソートの領域では、クリーンアップが長くなることが多いため、クリーンアップを実行します。

詳細ターゲットプロパティを設定する場合、テーブルのクリーンアップオプションを有効にできます。

次のリカバリオプションを選択できます。

**なし**

行のソートまたはディスク領域の回復は行わないでください。

**フル**

指定したテーブルまたはデータベース内のすべてのテーブルをソートし、前回の更新操作や削除操作で削除のマークが付けられた行によって使用されたディスク領域を回復します。

**ソートのみ**

削除行から解放された領域を回復せずに、指定したテーブルまたはデータベース内のすべてのテーブルをソートします。

**削除のみ**

前回の更新操作や削除操作で削除のマークが付けられた行によって使用されたディスク領域を回復し、テーブルを圧縮して使用領域を解放します。

**再インデックス**

インターリーブソートキーカラムで値の分布を分析し、テーブルのクリーンアップ操作全体を設定してパフォーマンスを改善します。

## リカバリおよびリスタート処理

Amazon Redshift V2 コネクタでは、CDC ソースから変更データをキャプチャして Amazon Redshift のターゲットテーブルに書き込むマッピングタスクを実行する際に、リカバリおよびリスタート処理がサポートされます。詳細モードのマッピングを実行する場合、リカバリおよびリスタート処理は使用できません。

マッピングタスクが失敗したり、完了前に停止されたりした場合、Secure Agent は、ターゲットシステムの `infa_recovery_table` テーブルに格納されているリカバリ情報を使用して、変更データの抽出を中断箇所から再開します。これにより、変更データの喪失や、ソースとターゲットの不整合が防止されます。

リカバリおよびリスタート処理を有効にするには、マッピングタスクを作成または編集する際に、**[スケジュール]** ページで詳細セッションプロパティの **[リカバリストラテジ]** を **[最終チェックポイントから再開]** に設定します。このように設定すると、マッピングタスクで変更データの処理を中断箇所から再開できるようになります。

特別な状況では、マッピングタスクのリスタートポイントを指定することもできます。例えば、マッピングタスクを初めて開始するときには、スキップされる変更レコードが生じないように、ターゲットの作成時に対応するリスタートポイントを指定するのが一般的です。デフォルトのリスタートポイントはログの末尾（EOL）です。これは、ログの現在の CDC 処理に相当します。ログで利用可能な最も早いレコードや特定の日時から抽

出処理が開始されるようにリスタートポイントを指定することもできます。時間ベースのリスタートポイントを使用すると、最初の作業単位（UOW）の終了時刻がそのリスタート時刻より後になっているログで抽出処理が開始されます。

リスタートポイントを指定する際には次のことに注意してください。

- リスタートポイントは、そのマッピングタスクに関連付けられているマッピングのすべてのソースに適用されます。
- 設定したリスタートポイントが早すぎると、対応するログファイルが期限切れになっていることがあります。その場合、そのリスタートポイントの値は、利用可能なログファイルの中で最も早いレコードと見なされます。
- リスタートポイントをログファイルの最新のレコードより後に設定すると、エラーメッセージが表示されます。

**注:** リスタート情報はマッピングタスク（特定のソースとターゲットの組み合わせ）に関連付けられます。マッピングのソースオブジェクトを変更した場合は、そのマッピングの新しいマッピングタスクを作成するか、既存のマッピングタスクのリスタートリビジョン番号を増やす必要があります。リスタートリビジョン番号を増やすには、マッピングタスクの［**CDC ランタイム**］ページに移動して、［**再起動ポイントの選択**］ダイアログボックスを開き、［**OK**］をクリックします。このいずれかの操作を行わないと、マッピングタスクが次回の実行時に失敗します。

## 書き込み時にレコードの順序を維持

CDC ソースからターゲットテーブルに変更レコードをキャプチャする際に、変更レコードの注文番号を維持することができます。このプロパティを使用すると、CDC ソースとターゲットの不整合を防止できます。詳細モードのマッピングを実行する場合、［書き込み時にレコードの順序を保持］プロパティは使用できません。

CDC ソースの 1 つの行で 1 つのレコードを複数回変更する際に、詳細ターゲットプロパティの［**書き込み時にレコードの順序を維持**］オプションを有効にすると、変更レコードがターゲットテーブルに書き込まれるときに変更レコードの注文番号が維持されます。

例えば、以下の CDC ソーステーブルのレコードで複数の操作を実行したとします。

| Emp ID | Emp Name | Emp Description | RowType | RowID |
|---|---|---|---|---|
| 1 | John | L1 | 挿入 | 1 |
| 1 | John | L2 | 更新 | 2 |
| 1 | John | L3 | 更新 | 3 |

ここで、`RowID` は CDC ソーステーブルの変更レコードの順序を示すものとします。

この場合、Secure Agent は、次の変更レコードを注文番号とともにターゲットテーブルに書き込みます。

| Emp ID | Emp Name | Emp Description | RowType | RowID |
|---|---|---|---|---|
| 1 | John | L3 | 更新 | 3 |

## DELIMITER および QUOTE としての 8 進数値

印刷できる ASCII 文字に加えて、印刷できる ASCII 文字と印刷できない ASCII 文字の 8 進数値を DELIMITER および QUOTE として使用できます。

印刷できる文字を DELIMITER または QUOTE として使用するには、ASCII 文字またはそれぞれの 8 進数値を指定します。ただし、印刷できない文字を DELIMITER または QUOTE として使用するには、それぞれの 8 進数値を指定する必要があります。

印刷できる文字の例:

`DELIMITER=#`または `DELIMITER=\043`

印刷できない文字の例、ファイルセパレータ:

`QUOTE=\034`

8 進数値 000-037 と 177 は印刷できない文字を表し、040-176 は印刷できる文字を表します。次の表に、Amazon Redshift によりサポートされる、Copy コマンドの QUOTE と DELIMITER、および Unload コマンドの DELIMITER の推奨 8 進数値を示します。

| コマンドオプション | 推奨 8 進数値 |
|---|---|
| COPY QUOTE | 001-010、016-037、041-054、057、073-100、133、135-140、173-177 |
| COPY DELIMITER | 001-011、013、014、016、017、020-046、050-054、057、073-133、135-177 |
| UNLOAD DELIMITER | 001-011、013、014、016、017、020-041、043-045、050-054、056-133、135-177 |

## 成功ファイルおよびエラーファイル

Secure Agent は、マッピングの実行後に成功ファイルとエラーファイルを生成します。成功ファイルとエラーファイルは、行レベルの詳細を含む.csv ファイルです。詳細モードでマッピングを実行する場合、成功ファイルとエラーファイルは使用できません。

Secure Agent は、マッピングの実行後に成功ファイルとエラーファイルを生成します。成功ファイルには、Amazon Redshift に正常に書き込まれた各レコードのエントリが含まれます。各エントリには、レコードのすべてのフィールドに対して書き込まれた値が含まれます。このファイルを使用すると、Secure Agent が Amazon S3 バケットと Amazon Redshift ターゲットに書き込んだデータを把握できます。

エラーファイルには、各データエラーのエントリが含まれます。ファイルの各エントリには、レコードのすべてのフィールドの値とエラーメッセージが含まれます。エラーファイルを使用すると、Secure Agent が Amazon Redshift ターゲットにデータを書き込まない理由を把握できます。

Secure Agent は成功ファイルまたはエラーファイルを上書きしません。エラー行ファイルと成功行ファイルが生成されたディレクトリからこれらのファイルに直接アクセスします。不要になったファイルは手動で削除できます。

成功ファイルのマッピングプロパティを設定する場合は、次のガイドラインを考慮してください。

- Secure Agent が成功行ファイルを生成する場合のファイルパスを指定する必要があります。
- 成功行ファイルでは次の命名規則が使用されます。`<timestamp>success`

エラーファイルのマッピングプロパティを設定する場合は、次のガイドラインを考慮してください。

- Secure Agent がエラー行ファイルを生成する場合のファイルパスを指定する必要があります。
- 成功行ファイルでは、次の命名規則が使用されます。`<timestamp>error`

  **注:** insert タスクと upsert タスクのエラー行ファイルは同じ命名規則に従います。

- エラーファイルディレクトリを定義する場合は、変数$PMBadFileDir を使用します。変数$PMBadFileDir を使用すると、アプリケーションがファイルを次の Secure Agent ディレクトリに書き込みます。<Secure Agent インストールディレクトリ>/apps/Data_Integration_Server/data/error。

# 第 7 章

# Amazon Redshift によるマッピングとマッピングタスク

マッピングを作成するときに、ソースまたはターゲットのトランスフォーメーションを Amazon Redshift V2 オブジェクトを表すように設定できます。

マッピングタスクを作成して、マッピングまたは統合テンプレートで定義されているデータフローロジックに基づきデータを処理します。Oracle CDC ソースから変更データをキャプチャして Amazon Redshift のターゲットテーブルに書き込むマッピングタスクを作成することもできます。マッピングを詳細モードに切り替えて、高度な機能を有効にするためのトランスフォーメーションと関数を含めることができます。

**注意:** ソース、ターゲット、およびルックアップトランスフォーメーションで設定された pre-SQL および post-SQL コマンドは、実行時には検証されません。安全でない可能性があるコンテンツがコマンド構文に含まれていないこと、および接続オブジェクトで設定したデータベースユーザーに適切な権限があることを確認してください。安全なネットワーク通信のために、TLS 対応のデータベース接続を使用することをお勧めします。

# 第 8 章

# マッピングにおける Amazon Redshift V2 オブジェクトのルールおよびガイドライン

マッピングで Amazon Redshift V2 オブジェクトを設定する場合は、次のルールとガイドラインを考慮してください。

- マッピングで簡易フィルタを使用する場合は、`YYYY-MM-DD HH24:MI:SS.MS` 形式でフィルタ条件を指定する必要があります。詳細フィルタを使用する場合は、`date_time_fix.f_timestamp < to_date('2012-05-24 09:13:57','YYYY-MM-DD HH24:MI:SS.MS')`形式でフィルタ条件を指定する必要があります。
- ソースまたはターゲットのクエリを定義し、テーブル名またはカラム名に特殊文字が含まれている場合は、テーブル名またはカラム名を二重引用符で囲む必要があります。
- ソースカラム名の特殊文字をターゲットに書き込むには、マッピングでターゲットトランスフォーメーションを設定して、実行時に新しいターゲットを作成します。［実行時に新規作成］ウィンドウで［ソースフィールド名をターゲットでそのまま使用］プロパティを有効にします。既存のオブジェクトを選択した場合、このチェックボックスは表示されません。この機能は、詳細モードのマッピングには適用されません。
- boolean データ型を含む Amazon Redshift V2 オブジェクトを選択してデータをプレビューする場合、Secure Agent は boolean データ型の値を切り詰めて、boolean 値の最初の文字のみを表示します。
- フィールドに日付または日時のデータ型が含まれている場合、**［データプレビュー］** タブには、+4 時間のオフセットでフィールド値が表示されます。
- 詳細モードのマッピングを使用してデータを読み書きする場合、ソーステーブル名に Unicode 文字または特殊文字を含めることはできません。

# マッピングの Amazon Redshift V2 ソース

マッピングでは、ソースのトランスフォーメーションを Amazon Redshift V2 ソースを表すように設定できます。

以下の表に、ソーストランスフォーメーションで設定できる Amazon Redshift V2 ソースプロパティを示します。

| プロパティ | 説明 |
|---|---|
| 接続 | ソース接続の名前。ソース接続を選択するか、**[新しいパラメータ]** をクリックしてソース接続に新しいパラメータを定義します。 |
| ソースタイプ | ソースオブジェクトのタイプ。<br>次のソースオブジェクトのいずれかを選択します。<br>- 単一オブジェクト<br>- 複数のオブジェクト。複数のオブジェクトとの暗黙の結合および詳細リレーションを使用できます。<br>- クエリ [1]。ソースタイプにクエリを選択する場合は、**[フィールドマッピング]** タブのクエリで選択したすべてのフィールドをマップする必要があります。<br>- パラメータ<br>**注**: マッピングのパラメータファイルを使用して、ランタイム時にソースクエリオブジェクトと複数のオブジェクトを上書きすることはできません。<br>ソースタイプにクエリを選択した場合、ブール値はターゲットに 0 または false として書き込まれます。<br>クエリを指定する場合は、末尾にセミコロン（;）を使用しないようにしてください。 |
| オブジェクト | ソースオブジェクトの名前。<br>単一または複数のソースオブジェクトを選択できます。 |
| パラメータ | ソースオブジェクトに既存のパラメータを選択するか、**[新しいパラメータ]** をクリックしてソースオブジェクトに新しいパラメータを定義します。パラメータをソースタイプとして選択する場合にのみ、**パラメータ**プロパティが表示されます。ランタイム時にパラメータを上書きする場合、**[パラメータを上書き]** オプションを選択します。 |
| フィルタ | フィルタ条件に基づいてレコードをフィルタリングします。<br>簡易フィルタまたは詳細フィルタを指定できます。 |
| ソート | 指定した条件に基づいてレコードをソートします。以下のソート条件を指定できます。<br>- パラメータを使用しません。使用するフィールドとソートのタイプを選択します。<br>- パラメータを使用します。ソートオプションを指定するには、パラメータを使用します。 |

[1] 詳細モードのマッピングには適用されません。

以下の表に、ソーストランスフォーメーションで設定できる Amazon Redshift V2 の詳細ソースプロパティを示します。

| プロパティ | 説明 |
|---|---|
| S3 バケット名 | データのステージングに使用する Amazon S3 バケットの名前。<br>バケット名をフォルダパスで指定することもできます。Amazon Redshift クラスタと異なる領域にある Amazon S3 バケット名を指定する場合、Unload コマンドオプションに REGION 属性を設定する必要があります。 |
| 圧縮を有効にする | Amazon S3 ステージングディレクトリのステージングファイルを圧縮します。<br>Secure Agent がステージングファイルを圧縮すると、タスクのパフォーマンスが向上します。デフォルトではオンに設定されています。 |
| ステージングディレクトリの場所 [1] | ローカルステージングディレクトリの場所。<br>Secure Agent ランタイム環境でタスクを実行する場合は、ランタイム環境内の対応する Secure Agent マシンで利用できるディレクトリパスを指定します。<br>次の形式でディレクトリパスを指定します: `<staging directory>`<br>例えば、`C:\Temp` です。このディレクトリに対する書き込み権限があることを確認してください。 |
| Unload オプション | Unload コマンドのオプション。<br>Unload コマンドにオプションを追加します。このコマンドは、Amazon Redshift からデータを抽出し、Amazon S3 でステージングファイルを作成します。Amazon Redshift ロールの Amazon リソース名（ARN）を指定します。<br>次のオプションを追加できます。<br>- DELIMITER<br>- ESCAPE<br>- PARALLEL<br>- NULL[1]<br>- AWS_IAM_ROLE<br>- REGION<br>- ADDQUOTES<br>例: DELIMITER = \036;ESCAPE = OFF;NULL=text;PARALLEL = ON;AWS_IAM_ROLE=arn;aws;iam;;<account ID>;role/<role-name>;REGION = ap-south-1<br>Secure Agent をホストするマシン上のディレクトリを指定します。<br>**注**: Unload コマンドに手動でオプションを追加しない場合、Secure Agent はデフォルト値を使用します。 |
| NULL 値を NULL として扱う | Amazon Redshift からデータを読み取る際に NULL 値を保持します。 |
| 暗号化タイプ | Amazon S3 ステージングディレクトリのデータを暗号化します。<br>以下の暗号化タイプを選択できます。<br>- なし<br>- SSE-S3<br>- SSE-KMS<br>- CSE-SMK[1]<br>デフォルトは None です。 |
| S3 ファイルの分割ダウンロード [1] | 大きな Amazon S3 オブジェクトを複数パーツに分けてダウンロードします。<br>Amazon S3 オブジェクトのファイルサイズが 8MB を超える場合は、オブジェクトを複数パーツに分けて同時にダウンロードできます。デフォルトは 5MB です。 |

| プロパティ | 説明 |
| --- | --- |
| マルチパートダウンロードしきい値サイズ [1] | Amazon S3 オブジェクトを複数のパートに分けてダウンロードするための最大しきい値サイズ。<br>デフォルトは 5 MB です。 |
| スキーマ名 | デフォルトのスキーマ名をオーバーライドします。<br>**注**: スキーマ名を使用する場合、カスタムクエリを設定することはできません。 |
| ソーステーブル名 | デフォルトのソーステーブル名をオーバーライドします。<br>**注**: ソースタイプに［**複数のオブジェクト**］または［**クエリ**］を選択した場合、［**ソーステーブル名**］オプションは使用できません。 |
| Pre-SQL | Amazon Redshift からデータを読み取る前にクエリを実行するための Pre-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| Post-SQL | Amazon Redshift にデータを書き込んだ後にクエリを実行するための Post-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| 個別選択 | 一意の値を選択します。<br>このオプションを選択すると、Secure Agent に `SELECT DISTINCT` 文が含まれます。Amazon Redshift は末尾のスペースを無視します。したがって、予期されるより少ない行が Secure Agent で抽出される可能性があります。<br>**注**: ソースタイプをクエリとして選択した場合や、［SQL **クエリ**］プロパティを使用して［**個別選択**］オプションを選択した場合、Secure Agent は［**個別選択**］オプションを無視します。 |
| SQL クエリ | デフォルトの SQL クエリをオーバーライドします。<br>カラム名を二重引用符で囲みます。SQL クエリでは大文字と小文字が区別されます。Amazon Redshift データベースでサポートされている SQL 文を指定します。<br>SQL クエリでカラムを指定するときに、クエリのカラム名がマッピングのソースカラム名と一致するようにします。 |
| 一時的な資格情報の期間 | IAM ユーザーが動的に生成された一時的な資格情報を使用して AWS リソースにアクセスできる期間。期間を秒単位で入力します。<br>デフォルトは 900 秒です。<br>900 秒以上が必要な場合、AWS コンソールで期間を最大 12 時間まで設定できます。設定した後で、同じ期間をこのプロパティに入力してください。 |
| トレースレベル | 詳細トレースレベルを使用して、ソーストランスフォーメーションのログに表示される情報の詳細度を取得します。 |
| [1] 詳細モードのマッピングには適用されません。 | |

## SQL クエリの設定のルールおよびガイドライン

SQL クエリを設定するときは、次のルールおよびガイドラインを考慮してください。

- ソースタイプに［**クエリ**］を選択した場合、デフォルトの SQL クエリをオーバーライドすることはできません。
- ［**SQL クエリ**］プロパティで日付とタイムスタンプの列を入力する場合は、日付とタイムスタンプのカラムを次の形式で設定する必要があります。

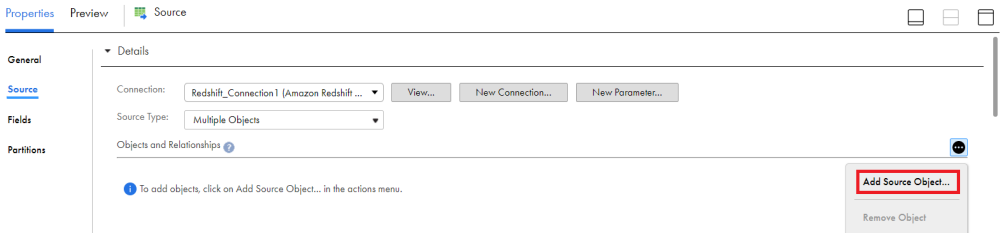
  `YYYY-MM-DD HH24:MI:SS.MS`

- ソースの SQL クエリを指定すると、ブール値はターゲットに 0 または false として書き込まれます。

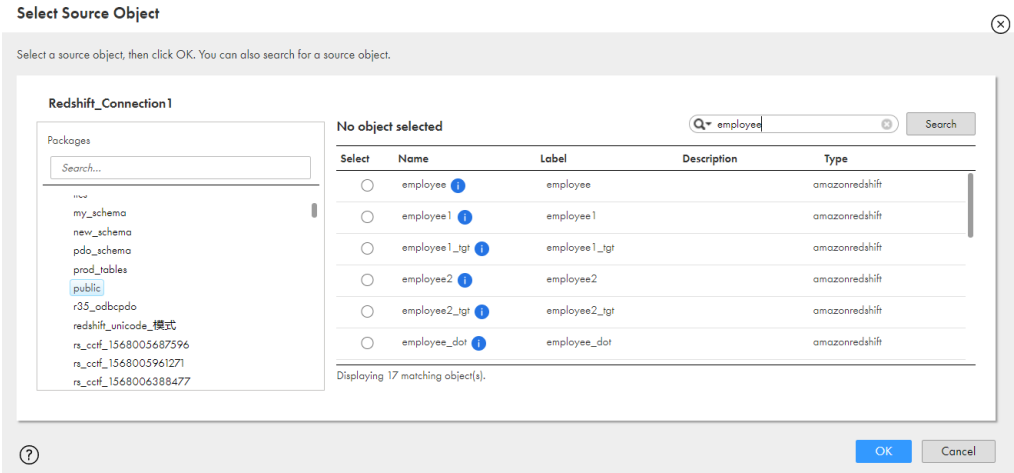
# 複数のソースオブジェクトの追加

ソーストランスフォーメーションを作成するときに、ソースタイプとして複数の Amazon Redshift V2 オブジェクトを選択してから、結合を設定してテーブルを結合できます。リレーション条件またはクエリを定義して、テーブルを結合できます。

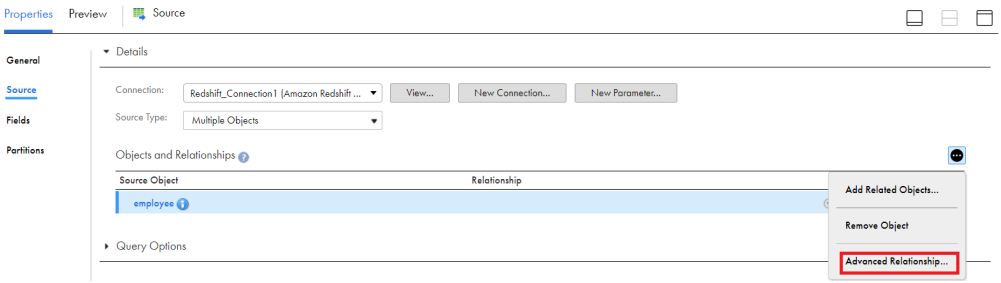
1. ソーストランスフォーメーションで、**[ソースタイプ]** を **[複数のオブジェクト]** としてクリックします。
2. **[アクション]** メニューから、**[ソースオブジェクトの追加]** をクリックします。



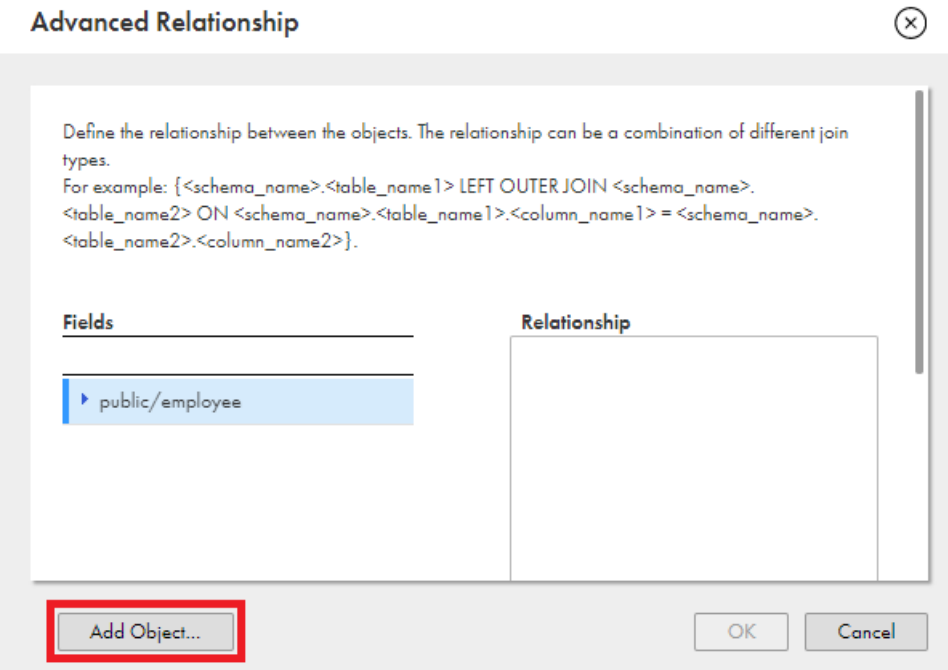

3. 表示されたリストから、追加するソースオブジェクトを選択します。



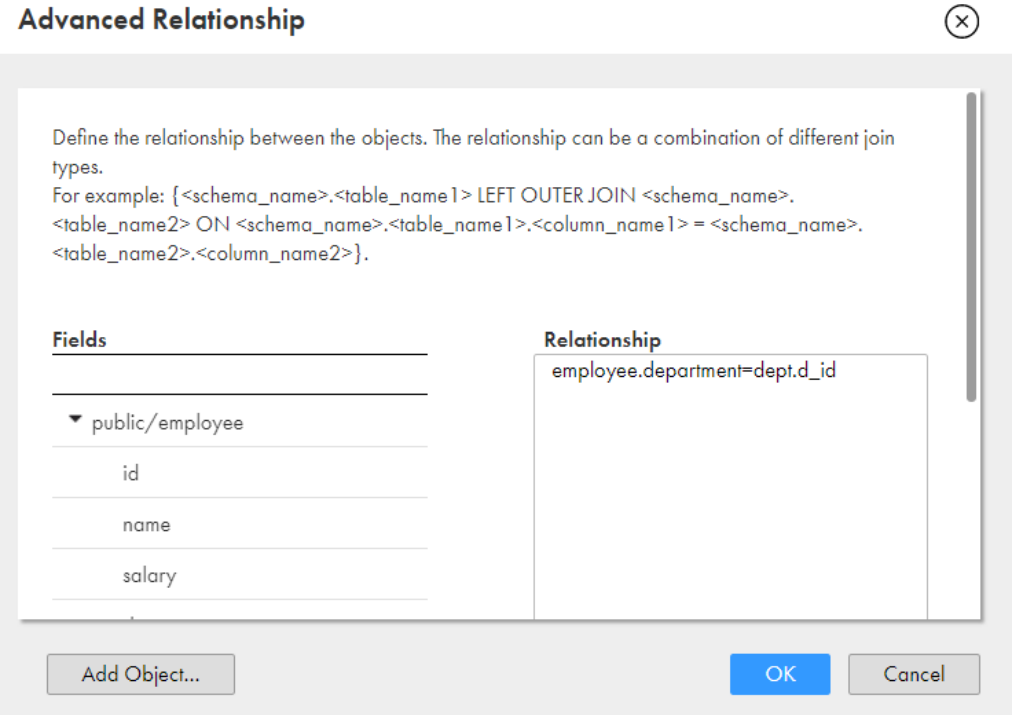

4. **[OK]** をクリックします。
5. **[関連オブジェクトのアクション]** メニューから、**[詳細リレーション]** を選択します。




6. **[詳細リレーション]** ウィンドウで、**[オブジェクトの追加]** をクリックしてさらにオブジェクトを追加できます。



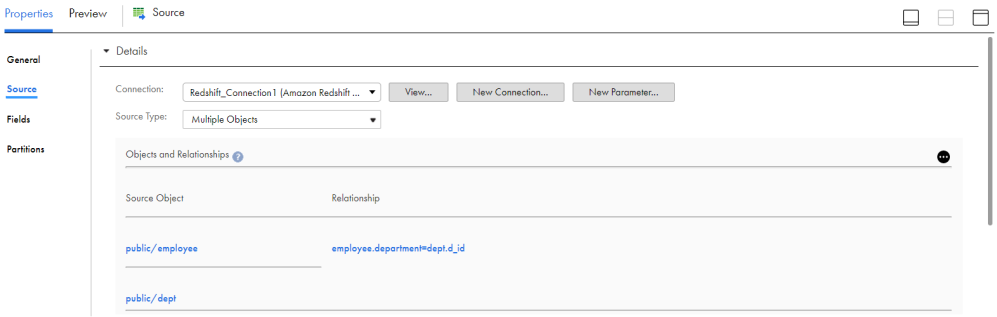

7. **[OK]** をクリックします。
8. 独自の条件を設定するか、クエリを指定して、テーブル間のリレーションを定義します。




**注:** 結合式を設定するときは、フィールドを選択して、結合条件またはクエリ構文を定義します。

9. **[OK]** をクリックします。

次の図は、Amazon Redshift V2 テーブル間に定義された詳細結合条件の例を示しています。




## 複数のソースオブジェクトの追加のルールおよびガイドライン

複数のソースオブジェクトを追加するときは、次のルールとガイドラインを考慮してください。

- 詳細リレーションオプションを使用する場合、パーティション化を設定することはできません。
- テーブル名とカラム名に予約語を使用する場合、テーブル名を二重引用符で囲む必要があります。
- 複数のソースオブジェクトを追加する場合、自己結合を使用することはできません。
- 詳細リレーションのカラム名に特殊文字を使用すると、クエリが正しく形成されず、マッピングタスクが失敗します。
- **[詳細リレーション]** ウィンドウで **[オブジェクトの追加]** をクリックしてオブジェクトをさらに追加すると、テーブルがロードされないか、ロードに時間がかかる場合があります。その後で、オブジェクトをもう一度インポートしてください。
- 関連するソースオブジェクトを追加するときにスキーマを検索することはできません。下にスクロールして、スキーマを手動で選択する必要があります。
- 完全外部結合条件は、=、,、および AND 演算子でのみ使用できます。
- 詳細リレーションオプションを使用し、スキーマ名のオーバーライドを使用して結合クエリを指定すると、マッピングは失敗します。マッピングが失敗して、次の結合クエリが表示されます。

  - インナージョイン

  - 左外部結合

  - 右外部結合

  - 完全外部結合

  - クロス結合
- スキーマ名を上書きして、関連するソースオブジェクトに詳細フィルタを設定する場合は、フィルタ条件でスキーマ名を指定する必要があります。フィルタ条件には、次の構文を使用します。
  `public.alldatatypes_src.f_smallint=1`
- スキーマ名をオーバーライドして、関連するソースオブジェクトに詳細フィルタを設定すると、Secure Agent は、関連するソースオブジェクトではなく、親オブジェクトにのみ詳細フィルタを適用します。
- プライマリキーと外部キーのリレーションを持つ親オブジェクトと子オブジェクトを選択し、関連オブジェクトの外部キーがテーブルのプライマリキーでもある場合、ターゲットの作成時にマッピングタスクが失敗します。

- **［複数のオブジェクト］**ソースタイプを選択し、emp などのソースオブジェクトを追加し、emp.id と dept.d_id などの異なるカラムでプライマリキーと外部キーのリレーションを定義すると、マッピングが失敗して、次のエラーが表示されます。
  [FATAL] Unload/Copy command failed with error: Invalid operation: column emp.d_id does not exist.

  **［関連オブジェクトの選択］**リストには、emp テーブルに d_id カラムがない場合でも dept 関連オブジェクトの結合条件が emp.d_id=dept.d_id として表示されます。

# マッピングでの Amazon Redshift V2 ターゲット

データを Amazon Redshift に書き込むには、マッピングで Amazon Redshift V2 オブジェクトをターゲットとして設定します。

ソースパーティションを有効にした場合、Secure Agent はパススルーパーティション化を使用して Amazon Redshift にデータを書き込み、実行時のマッピングのパフォーマンスを最適化します。Amazon Redshift V2 ターゲットの名前と説明を指定します。ターゲットオブジェクトのターゲットプロパティと詳細プロパティを設定します。

以下の表に、ターゲットトランスフォーメーションで設定できるターゲットプロパティを示します。

| プロパティ | 説明 |
|---|---|
| 接続 | ターゲット接続の名前。ターゲット接続を選択するか、**［新しいパラメータ］**をクリックしてターゲット接続に新しいパラメータを定義します。 |
| ターゲットタイプ | ターゲットオブジェクトのタイプ。<br>［単一オブジェクト］または［パラメータ］を選択します。 |
| オブジェクト | ターゲットオブジェクトの名前。<br>実行時に、既存のターゲットオブジェクトを使用するか、新しいターゲットオブジェクトを作成するかを選択できます。実行時に新しいターゲットを作成するには、［実行時に新規作成］オプションを選択します<br>新しいターゲットを、指定したテーブルタイプとパスに基づいて実行時に作成します。<br>オブジェクト名、テーブルタイプ、およびパスを指定して、実行時に新しいオブジェクトを作成します。 |
| 実行時に新規作成 | ［実行時に新規作成］オプションに適用されます。<br>ソースと同じフィールド名で実行時にターゲットオブジェクトを作成できるかどうかを指定します。 |
| ソースフィールド名をターゲットでそのまま使用 | ［実行時に新規作成］オプションに適用されます。このオプションは、既存のターゲットには表示されません。<br>すべてのソースフィールド名（特殊文字を含む）をソースとまったく同じようにターゲットに保持する場合に選択します。このオプションを無効にすると、ソースの特殊文字は、ターゲットでアンダースコア文字に置換されます。<br>デフォルトでは無効になっています。 |

| プロパティ | 説明 |
| --- | --- |
| パラメータ | ターゲットオブジェクトの既存のパラメータを選択するか、［**新しいパラメータ**］をクリックしてターゲットオブジェクトの新しいパラメータを定義します。<br>ターゲットタイプとして［パラメータ］を選択する場合にのみ［**パラメータ**］プロパティが表示されます。ランタイム時にパラメータを上書きする場合、［**パラメータを上書き**］オプションを選択します。 |
| 操作 | ターゲット操作のタイプ。<br>次のいずれかの操作を選択します。<br>- Insert<br>- Update<br>- Upsert<br>- Delete<br>- Data Driven[1]<br>CDC ソースから変更データをキャプチャするマッピングを作成する場合は［Data Driven］を選択します。 |
| データ依存条件 | 行に挿入、更新、削除、または拒否操作のフラグを設定する式を定義できます。<br>非 CDC ソースに対するデータ依存条件を指定する必要があります。CDC ソースに対しては、CDC ソーステーブルの行がすでに操作タイプでマークされているため、このフィールドは空白のままにしておく必要があります。<br>**注**: 操作タイプとして［Data Driven］を選択した場合にのみ表示されます。 |
| 更新カラム | ターゲットで Update、Upsert、Delete の各操作を実行するための論理プライマリキーとして使用するカラムを選択します。<br>**注**: ターゲットテーブルにすでにプライマリキーが含まれている場合は必要ありません。 |
| ターゲットの作成 | 新しいターゲットを作成します<br>新しいターゲットを作成するときに、次のフィールドの値を入力します。<br>- **名前**: ターゲットオブジェクトの名前を入力してください。<br>- **パス**: スキーマ名を指定して、スキーマ内にターゲットテーブルを作成します。デフォルトでは、このフィールドは空です。 |
| [1] 詳細モードのマッピングには適用されません。 | |

次の表に、Amazon Redshift V2 の詳細なターゲットプロパティを示します。

| プロパティ | 説明 |
| --- | --- |
| S3 バケット名 | ファイルを Amazon Redshift ターゲットに書き込む Amazon S3 バケットの名前。<br>バケット名をフォルダパスで指定することもできます。Amazon Redshift クラスタと異なる領域にある Amazon S3 バケット名を指定する場合、Copy コマンドオプションに REGION 属性を設定する必要があります。 |
| 圧縮を有効にする | ステージングファイルを Amazon Redshift に書き込む前に圧縮します。<br>Secure Agent がステージングファイルを圧縮すると、タスクのパフォーマンスが向上します。デフォルトではオンに設定されています。 |

| プロパティ | 説明 |
| --- | --- |
| ステージングディレクトリの場所 [1] | ローカルステージングディレクトリの場所。<br>Secure Agent ランタイム環境でタスクを実行する場合は、ランタイム環境内の対応する Secure Agent マシンで利用できるディレクトリパスを指定します。<br>次の形式でディレクトリパスを指定します: `<staging directory>`<br>例えば、`C:\Temp` です。このディレクトリに対する書き込み権限があることを確認してください。 |
| バッチサイズ [1] | バッチ内の最小行数。<br>0 より大きい値を入力します。デフォルトは 2000000 です。 |
| INSERT 用のアップロードバッチごとの最大エラー数 | INSERT 用のアップロードバッチが失敗になるエラー行の数。<br>正の整数値を入力します。デフォルトは 1 です。<br>エラー数がプロパティ値以上になると、Secure Agent はバッチ全体をエラーファイルに書き込みます。 |
| データロード前にターゲットテーブルをトランケート | 新しいデータをロードする前に Amazon Redshift のターゲットテーブルにある既存データをすべて削除します。 |
| Char および Varchar に Null 値が必要 | データ型が Char および Varchar の Amazon Redshift のカラムにデータを書き込む際に文字列値を NULL に置換します。<br>デフォルトは空の文字列です。<br>**注**: データ型が Int、Bigint、numeric、real、または double の 1 つのカラムのみを含むテーブルに NULL 値を書き込むマッピングを実行すると、マッピングが失敗します。この場合は、[Char **および** Varchar **に** Null **値が必要**] プロパティにデフォルト値以外の値を指定する必要があります。 |
| S3 ファイルの整合性のための待ち時間（秒）[1] | Secure Agent がステージングファイルと Amazon S3 で利用可能なファイルのリストを一致させるまで待機する時間（秒数）。<br>デフォルトは 0 です。 |
| コピーオプション | Copy コマンドのオプション。<br>デフォルトの区切り文字カンマ（,）または二重引用符（"）がデータで使用されている場合に、Amazon S3 から Amazon Redshift ターゲットにデータを書き込むための Copy コマンドにオプションを追加します。Amazon Redshift ロールの Amazon リソース名（ARN）を指定します。<br>次のオプションを追加できます。<br>- DELIMITER<br>- ACCEPTINVCHARS<br>- QUOTE<br>- COMPUPDATE<br>- AWS_IAM_ROLE<br>- REGION<br>以下に例を示します。<br>`DELIMITER = \036;ACCEPTINVCHARS = #;QUOTE = \037 COMPUPDATE = ON;AWS_IAM_ROLE=arn:aws:iam::<account ID>:role/<role-name>;REGION = ap-south-1`<br>Secure Agent をホストするマシン上のディレクトリを指定します。<br>**注**: Copy コマンドに手動でオプションを追加しない場合、Secure Agent はデフォルト値を使用します。 |

| プロパティ | 説明 |
| --- | --- |
| S3 サーバー側の暗号化 | Amazon S3 がアップロード時にデータを暗号化することを示します。<br>このプロパティを有効にするには、接続プロパティで顧客マスタキー ID を指定します。デフォルトでは選択されていません。 |
| S3 クライアント側の暗号化 [1] | Secure Agent がプライベートキーを使用してデータを暗号化することを示します。<br>このプロパティを有効にするには、接続プロパティでマスタ対称キー ID を指定します。サーバーサイドおよびクライアントサイドの両方の暗号化を有効にすると、サーバーサイド暗号化は無視されます。 |
| ターゲットテーブルの分析 | ターゲットテーブルで ANALYZE コマンドを実行します。<br>Amazon Redshift のクエリプランナーは、統計メタデータを更新して、クエリ効率の向上に最適な計画を作成して選択します。 |
| ターゲットテーブルのクリーンアップ | ディスク領域を回復し、指定したテーブルまたはデータベース内のすべてのテーブルの行をソートします。<br>次のリカバリオプションを選択できます。<br>- なし<br>- フル<br>- ソートのみ<br>- 削除のみ<br>- 再インデックス<br>デフォルトは［なし］です。 |
| S3 でステージングファイルを保持するプレフィックス | Amazon S3 でステージングファイルを保持します。<br>Amazon S3 でステージングファイルを保持するには、ディレクトリプレフィックスとファイルプレフィックスの両方をスラッシュ（/）で区切って指定するか、ファイルプレフィックスのみを指定します。例えば、`backup_dir/backup_file` や `backup_file` です。 |
| 成功ファイルディレクトリ [1] | Amazon Redshift の成功ファイルのディレクトリ。<br>Secure Agent をホストするマシン上のディレクトリを指定します。 |
| エラーファイルディレクトリ [1] | Amazon Redshift のエラーファイルのディレクトリ。<br>Secure Agent をホストするマシン上のディレクトリを指定します。 |

| プロパティ | 説明 |
| --- | --- |
| ソース行の扱い | デフォルトのターゲット操作をオーバーライドします。<br>デフォルトは［INSERT］です。<br>次のいずれかのオーバーライドオプションを選択します。<br>**なし**<br>デフォルトでは、なしが有効になっています。Secure Agent は、**［操作］**ターゲットプロパティで選択されたタスク操作と見なします。<br>INSERT<br>挿入操作を実行します。このオプションが有効になっている場合、Secure Agent は、挿入のフラグが設定された行をすべて挿入します。このオプションが無効になっている場合、Secure Agent は、挿入のフラグが設定された行を拒否します。<br>DELETE<br>削除操作を実行します。このオプションが有効になっている場合、Secure Agent は、削除のフラグが設定された行をすべて削除します。このオプションが無効になっている場合、Secure Agent は、削除のフラグが設定されたすべての行を拒否します。<br>UPDATE **と** UPSERT<br>更新操作と更新/挿入操作を実行します。更新操作を実行するには、プライマリキー列と、プライマリキー列以外の列を 1 つ以上マップする必要があります。次のデータオブジェクトの操作属性を選択できます。<br>- ［更新時に更新］: Secure Agent は更新時にすべての行を更新します。<br>- ［更新または挿入］: Secure Agent は既存の行を更新したり、挿入のマークが付けられているように他の行を挿入したりします。<br>詳細については、「Amazon Redshift V2 コネクタのトラブルシューティング」 (ページ 121)を参照してください。<br>Amazon Redshift V2 コネクタは、アップグレードストラテジトランスフォーメーションでの更新/挿入操作をサポートしていません。アップデートストラテジトランスフォーメーションを使用して Amazon Redshift ターゲットにデータを書き込むには、**［ソース行の扱い］**を**［なし］**として選択する必要があります。<br>デフォルトでは、Secure Agent は**［操作］**ターゲットプロパティに指定された値に基づいてタスク操作を実行します。ただし、**［ソース行の扱い］**プロパティでオプションを指定した場合、Secure Agent は**［操作］**ターゲットプロパティまたはアップデートストラテジトランスフォーメーションで指定された値を無視します。 |
| ターゲットクエリのオーバーライド [1] | Secure Agent が更新操作用に生成するデフォルトの更新クエリを、指定した更新クエリでオーバーライドします。 |
| TransferManager スレッドプールサイズ [1] | データを並行して書き込むスレッドの数。<br>デフォルトは 10 です。 |
| Pre-SQL | Amazon Redshift からデータを読み取る前にクエリを実行するための Pre-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| Post-SQL | Amazon Redshift にデータを書き込んだ後にクエリを実行するための Post-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |

| プロパティ | 説明 |
| --- | --- |
| 書き込み時にレコードの順序を維持 [1] | CDC ソースからデータを読み取って Amazon Redshift ターゲットに書き込む際にレコードの順序を維持します。<br>このプロパティは、CDC ソースから変更レコードをキャプチャするマッピングを作成する場合に使用します。このプロパティを使用すると、CDC ソースとターゲットの不整合を防止できます。 |
| 最小アップロードパートサイズ [1] | オブジェクトをアップロードするための Amazon Redshift オブジェクトの最小サイズ。<br>デフォルトは 5 MB です。 |
| バッチごとのファイル数 [1] | バッチごとの一時ファイルの数を計算します。<br>ファイルの数を指定しない場合は、Amazon Redshift V2 コネクタにより一時ファイルの数が計算されます。 |
| スキーマ名 | デフォルトのスキーマ名をオーバーライドします。 |
| ターゲットテーブル名 | デフォルトのターゲットテーブル名を上書きします。 |
| リカバリスキーマ名 [1] | ターゲットシステムの `infa_recovery_table` テーブルに格納されているリカバリ情報を含むスキーマ。この情報は、変更データの抽出を前回のチェックポイントから再開するために使用されます。 |
| 一時的な資格情報の期間 | IAM ユーザーが動的に生成された一時的な資格情報を使用して AWS リソースにアクセスできる期間。期間を秒単位で入力します。<br>デフォルトは 900 秒です。<br>900 秒以上が必要な場合、AWS コンソールで期間を最大 12 時間まで設定できます。設定した後で、同じ期間をこのプロパティに入力してください。 |
| 拒否された行の転送 | このプロパティは Amazon Redshift V2 コネクタに適用できません。 |
| [1] 詳細モードのマッピングには適用されません。 | |

## ターゲットの作成に関するルールとガイドライン

Secure Agent は、［**ターゲットの作成**］フィールドで指定されたターゲットテーブル名を小文字に変換します。

ターゲットを作成する際には、［**ターゲットフィールド**］タブでターゲットオブジェクトのメタデータを表示および編集できます。ターゲットオブジェクトのカラムのデータ型や精度を編集したり、プライマリキーを定義したりできます。メタデータを編集するには、［**ターゲットフィールド**］タブで［**オプション**］ > ［**メタデータの編集**］をクリックします。

ターゲットを作成するときは、次のルールとガイドラインを考慮してください。

- ターゲットを作成する際に、ソーステーブルに Text データ型のカラムが含まれていると、次のようなエラーメッセージが表示されます。

  ```
  Unsupported datatype - 'text' for column
  'LONGTXTAREA__C'
  ```

  この場合、ソーストランスフォーメーションでソースカラムのデータ型を編集する必要があります。
- ランタイム時にターゲットをパラメータ化することはできません。
- ターゲットを作成して大文字または大文字と小文字が混在するアルファベットでテーブル名またはカラム名を指定すると、Secure Agent でターゲットのテーブル名またはカラム名が小文字のアルファベットに変換

され、マッピングタスクが正常に実行されます。ただし、Tomcat ログに次のエラーメッセージが表示されます。
`An unexpected exception occured while fetching metadata:[The following object is not found].`

- Time データ型のターゲットを作成すると、新しいターゲットのデータ型は Timestamp になります。
- 実行前 SQL または実行後 SQL クエリを使用して vaccuum コマンドでマッピングタスクを実行すると、マッピングは失敗します。

## ターゲットクエリをオーバーライドするためのルールとガイドライン

ターゲットクエリのオーバーライドを指定して、Secure Agent が更新操作に対して生成する更新クエリをオーバーライドできます。

Amazon Redshift ターゲットにオーバーライドターゲットクエリプロパティを使用する場合は、次のルールとガイドラインを考慮してください。

- 詳細ターゲットプロパティの［**ソース行の扱い**］プロパティで［**更新**］を選択します。
- 更新/挿入操作にオーバーライドターゲットクエリを使用することはできません。
- オーバーライドターゲットクエリを次の形式で指定します。

```
UPDATE <Target Schema Name here>.<Target TABLE NAME here>
  SET <column1> = {{TU}}.<column1>,
      <column2> = {{TU}}.<column2>,
      <column> = {{TU}}.<column>
  FROM {{TU}}
  WHERE <Target Schema Name>.<Target TABLE NAME here> .<update column1> = {{TU}}. <update column1>
  AND <Target Schema Name>.<Target TABLE NAME here> .<update column2> = {{TU}}. <update column2>
  AND ... <Target Schema Name>.<Target TABLE NAME here> .<update column> = {{TU}}. <update column>
```

- :TU のカラム名はターゲットテーブルのカラム名と一致している必要があります。
- WHERE カラムはこのクエリで必須です。
- オーバーライドターゲットクエリステートメントの FROM カラムで、左、右、または完全外部結合を使用することはできません。
- クエリ内のすべてのカラム名は修飾名である必要があり、テーブル名とスキーマ名はカラム名に関連付けられている必要があります。
  注: SET カラムの場合、Amazon Redshift ではテーブルを含むカラム名を使用できません。SET カラムにはカラム名のみを含める必要があります。
- Amazon Redshift V2 コネクタは、:TU を一時テーブル名に置き換え、更新クエリを検証しないため、有効な SQL 構文でオーバーライドターゲットクエリを指定する必要があります。
- オーバーライドターゲットクエリを使用して、カラムのマッピングの順序を変更することはできません。
- 更新操作に複数のオーバーライドターゲットクエリを指定することはできません。
- オーバーライドターゲットクエリオプションを使用する場合は、［**スキーマ名**］フィールドと［**ターゲットテーブル名**］フィールドからスキーマ名とテーブル名のオーバーライドを追加で設定しないでください。

# マッピングの Amazon Redshift V2 ルックアップ

Amazon Redshift V2 マッピングでオブジェクトのルックアップを作成できます。指定したルックアップ条件に基づいて、Amazon Redshift V2 ルックアップオブジェクトからデータを取得できます。

Amazon Redshift V2 ルックアップトランスフォーメーションを使用して、Amazon Redshift オブジェクトのデータをルックアップします。例えば、顧客コードが格納されているソーステーブルに対して、ターゲットテーブルに顧客名が含まれるようにすると、要約データが読みやすくなります。Amazon Redshift V2 ルックアップトランスフォーメーションを使用すると、別の Amazon Redshift オブジェクト内にある顧客名をルックアップできます。

マッピングに、以下のルックアップを追加することができます。

- キャッシュされた状態での接続
- キャッシュされていない状態での接続。マッピングのみに適用されます。
- キャッシュされた状態での未接続
- 動的ルックアップキャッシュ。マッピングのみに適用されます。

接続プロパティで指定した JDBC URL を使用して、ルックアップを作成します。

以下の表に、ルックアップトランスフォーメーションで設定できる Amazon Redshift V2 ルックアップオブジェクトのプロパティを示します。

| プロパティ | 説明 |
|---|---|
| 接続 | ルックアップ接続の名前。<br>既存の接続の選択、新しい接続の作成、またはルックアップ接続プロパティのパラメータ値の定義ができます。<br>実行時にルックアップ接続プロパティを上書きする場合は、**[実行時にパラメータのオーバーライドを許可する]** オプションを選択します。 |
| ソースタイプ | ソースオブジェクトのタイプ［単一オブジェクト］、［クエリ］、または［パラメータ］を選択します。<br>**注**: ソースタイプをクエリとして選択した場合、キャッシュされていないルックアップを設定することはできません。 |
| パラメータ | 値を定義するパラメータファイル。この値はタスクを編集せずに更新されます。<br>ルックアップオブジェクトに既存のパラメータを選択するか、**[新しいパラメータ]** をクリックしてルックアップオブジェクトに新しいパラメータを定義します。<br>パラメータをルックアップタイプとして選択する場合にのみ、**パラメータ**プロパティが表示されます。<br>実行時にパラメータを上書きする場合は、**[実行時にパラメータのオーバーライドを許可する]** オプションを選択します。<br>タスクが実行されると、Secure Agent は詳細セッションプロパティで指定されたファイルのこのパラメータを使用します。 |
| ルックアップオブジェクト | マッピングのルックアップオブジェクトに名前を付けます。 |

| プロパティ | 説明 |
| --- | --- |
| 複数一致 | ルックアップ条件で複数の一致が返されるときの動作。すべての行、任意の行、最初の行、最後の行、またはエラーを返すようにすることができます。<br>ルックアップオブジェクトのプロパティで次のオプションから選択して、動作を決定します。<br>- 最初の行を返す<br>- 最後の行を返す<br>- 任意の行を返す<br>- すべての行を返す<br>- エラーを報告 |
| フィルタ | 適用できません |
| ソート | 適用できません |

以下の表に、ルックアップトランスフォーメーションで設定できる Amazon Redshift V2 の詳細ルックアッププロパティを示します。

| プロパティ | 説明 |
| --- | --- |
| S3 バケット名 [1] | データのステージングに使用する Amazon S3 バケットの名前。<br>バケット名をフォルダパスで指定することもできます。Amazon Redshift クラスタと異なる領域にある Amazon S3 バケット名を指定する場合、Unload コマンドオプションに REGION 属性を設定する必要があります。 |
| 圧縮を有効にする [1] | Amazon S3 ステージングディレクトリのステージングファイルを圧縮します。<br>Secure Agent がステージングファイルを圧縮すると、タスクのパフォーマンスが向上します。デフォルトではオンに設定されています。 |
| ステージングディレクトリの場所 [1] | ローカルステージングディレクトリの場所。<br>Secure Agent ランタイム環境でタスクを実行する場合は、ランタイム環境内の対応する Secure Agent マシンで利用できるディレクトリパスを指定します。<br>次の形式でディレクトリパスを指定します: `<staging directory>`<br>例えば、`C:\Temp` です。このディレクトリに対する書き込み権限があることを確認してください。<br>詳細モードのマッピングには適用されません。 |

| プロパティ | 説明 |
| --- | --- |
| Unload オプション [1] | Unload コマンドのオプション。<br>Unload コマンドにオプションを追加します。このコマンドは、Amazon Redshift からデータを抽出し、Amazon S3 でステージングファイルを作成します。Amazon Redshift ロールの Amazon リソース名（ARN）を指定します。<br>次のオプションを追加できます。<br>- DELIMITER<br>- ESCAPE<br>- PARALLEL<br>- NULL<br>- AWS_IAM_ROLE<br>- REGION<br>- ADDQUOTES<br>例: DELIMITER = \036;ESCAPE = OFF;NULL=text;PARALLEL = ON;AWS_IAM_ROLE=arn;aws;iam;;<account ID>;role/<role-name>;REGION = ap-south-1<br>詳細モードのマッピングでは、NULL オプションは使用できません。<br>Secure Agent をホストするマシン上のディレクトリを指定します。<br>**注**: Unload コマンドに手動でオプションを追加しない場合、Secure Agent はデフォルト値を使用します。 |
| NULL 値を NULL として扱う [1] | Amazon Redshift からデータを読み取る際に NULL 値を保持します。 |
| 暗号化タイプ [1] | Amazon S3 ステージングディレクトリのデータを暗号化します。<br>以下の暗号化タイプを選択できます。<br>- なし<br>- SSE-S3<br>- SSE-KMS<br>- CSE-SMK<br>詳細モードで設定されたマッピングでは、SSE-S3 暗号化のみを使用できます。<br>デフォルトは None です。 |
| S3 ファイルの分割ダウンロード [1] | 大きな Amazon S3 オブジェクトを複数パーツに分けてダウンロードします。<br>Amazon S3 オブジェクトのファイルサイズが 8MB を超える場合は、オブジェクトを複数パーツに分けて同時にダウンロードできます。デフォルトは 5MB です。<br>詳細モードのマッピングには適用されません。 |
| マルチパートダウンロードしきい値サイズ [1] | Amazon S3 オブジェクトを複数のパートに分けてダウンロードするための最大しきい値サイズ。<br>デフォルトは 5 MB です。<br>詳細モードのマッピングには適用されません。 |
| スキーマ名 | デフォルトのスキーマ名をオーバーライドします。<br>**注**: スキーマ名を使用する場合、カスタムクエリを設定することはできません。 |
| ソーステーブル名 | デフォルトのソーステーブル名をオーバーライドします。<br>**注**: ソースタイプに［**複数のオブジェクト**］または［**クエリ**］を選択した場合、［**ソーステーブル名**］オプションは使用できません。 |
| Pre-SQL | Amazon Redshift からデータを読み取る前にクエリを実行するための Pre-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |

| プロパティ | 説明 |
| --- | --- |
| Post-SQL | Amazon Redshift にデータを書き込んだ後にクエリを実行するための Post-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| 個別選択 [1] | 一意の値を選択します。<br>このオプションを選択すると、Secure Agent に `SELECT DISTINCT` 文が含まれます。Amazon Redshift は末尾のスペースを無視します。したがって、予期されるより少ない行が Secure Agent で抽出される可能性があります。<br>**注**: ソースタイプをクエリとして選択した場合や、［SQL **クエリ**］プロパティを使用して［**個別選択**］オプションを選択した場合、Secure Agent は［**個別選択**］オプションを無視します。 |
| SQL クエリ [1] | デフォルトの SQL クエリをオーバーライドします。<br>カラム名を二重引用符で囲みます。SQL クエリでは大文字と小文字が区別されます。Amazon Redshift データベースでサポートされている SQL 文を指定します。<br>SQL クエリでカラムを指定するときに、クエリのカラム名がマッピングのソースカラム名と一致するようにします。 |
| 一時資格情報の期間 [1] | IAM ユーザーが動的に生成された一時的な資格情報を使用して AWS リソースにアクセスできる期間。期間を秒単位で入力します。<br>デフォルトは 900 秒です。<br>900 秒以上が必要な場合、AWS コンソールで期間を最大 12 時間まで設定できます。設定した後で、同じ期間をこのプロパティに入力してください。 |
| トレースレベル | 詳細トレースレベルを使用して、ソーストランスフォーメーションのログに表示される情報の詳細度を取得します。 |
| [1] キャッシュされていないルックアップには適用されません。 | |

ルックアップトランスフォーメーションの詳細については、『*トランスフォーメーション*』を参照してください。

## 接続されていないルックアップトランスフォーメーション

マッピングで Amazon Redshift ソースに対して未接続のルックアップトランスフォーメーションを設定できます。ルックアップトランスフォーメーションを使用して、指定したルックアップ条件に基づいて Amazon Redshift からデータを取得します。

未接続のルックアップトランスフォーメーションは、パイプライン内のいずれのソース、ターゲット、またはトランスフォーメーションにも接続していないルックアップトランスフォーメーションです。

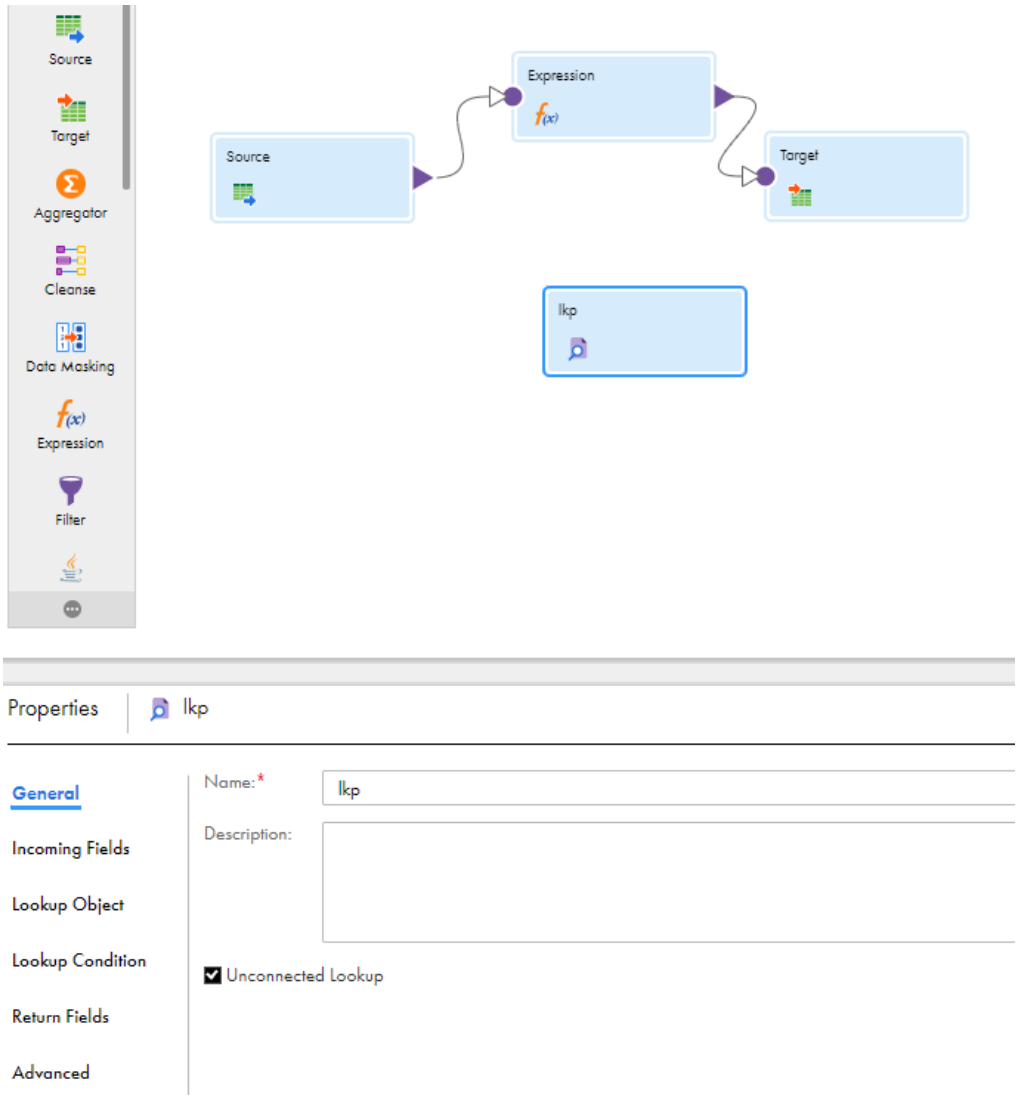
未接続のルックアップトランスフォーメーションは、別のトランスフォーメーションの:LKP 式の結果から入力値を受け取ります。統合サービスは、ルックアップトランスフォーメーションのルックアップポートと条件に基づいてルックアップソースにクエリを送信し、返された値を:LKP 式が含まれるポートに渡します。:LKP 式を使用することで、ルックアップの結果を別のトランスフォーメーション内の式に渡すことができます。

ルックアップトランスフォーメーションの詳細については、「*トランスフォーメーション*」を参照してください。

## 接続されていないルックアップトランスフォーメーションの設定

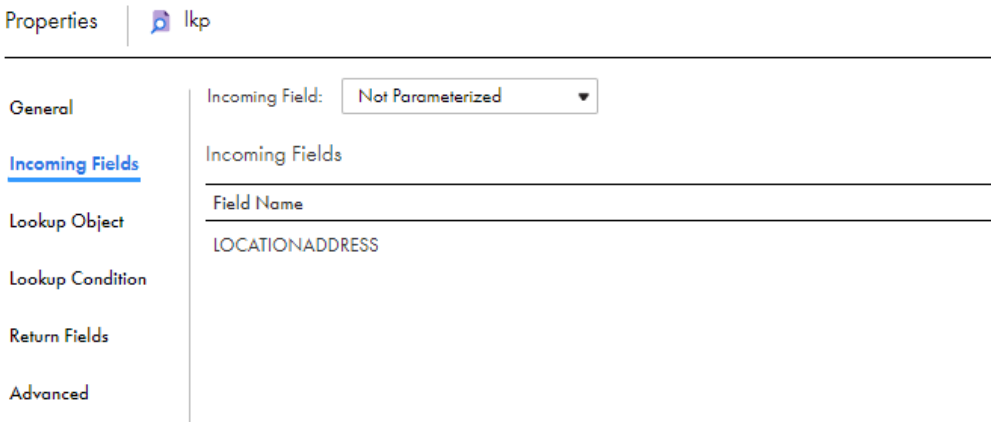
接続されていないルックアップトランスフォーメーションを設定するには、**[接続されていないルックアップ]** オプションを選択して受信フィールドを追加し、ルックアップ条件を設定してから戻り値を指定します。次に、別のトランスフォーメーションでのルックアップ式を設定します。

1. マッピングにルックアップトランスフォーメーションを追加します。
2. ルックアップトランスフォーメーションの **[全般]** タブで、**[接続されていないルックアップ]** オプションを有効にします。

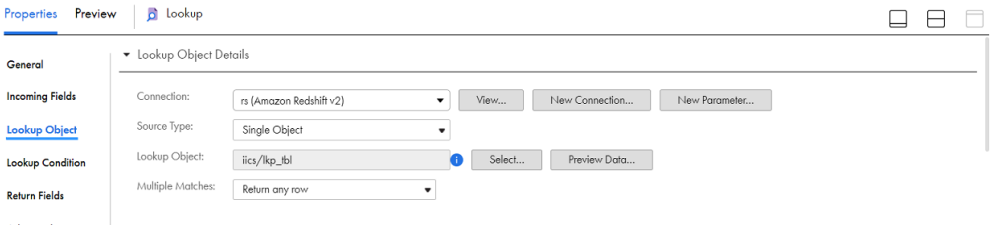



3. ルックアップトランスフォーメーションの **[受信フィールド]** タブで、:LKP 式の各引数に対して受信フィールドを作成します。

作成するルックアップ条件それぞれについて、ルックアップトランスフォーメーションに受信フィールドを追加します。



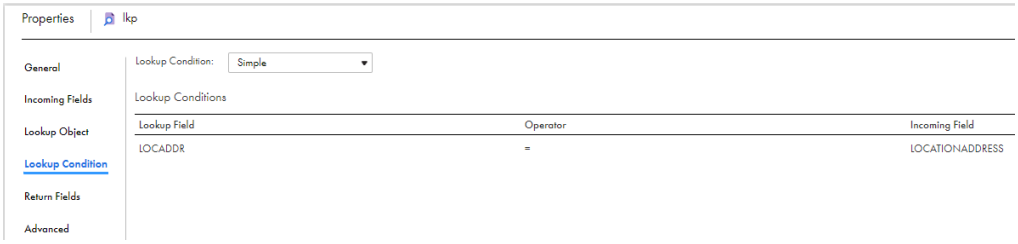

4. **［ルックアップオブジェクト］**タブでルックアップオブジェクトをインポートします。




未接続ルックアップで**［複数一致］**プロパティを値**［すべての行を返す］**に設定することはできません。

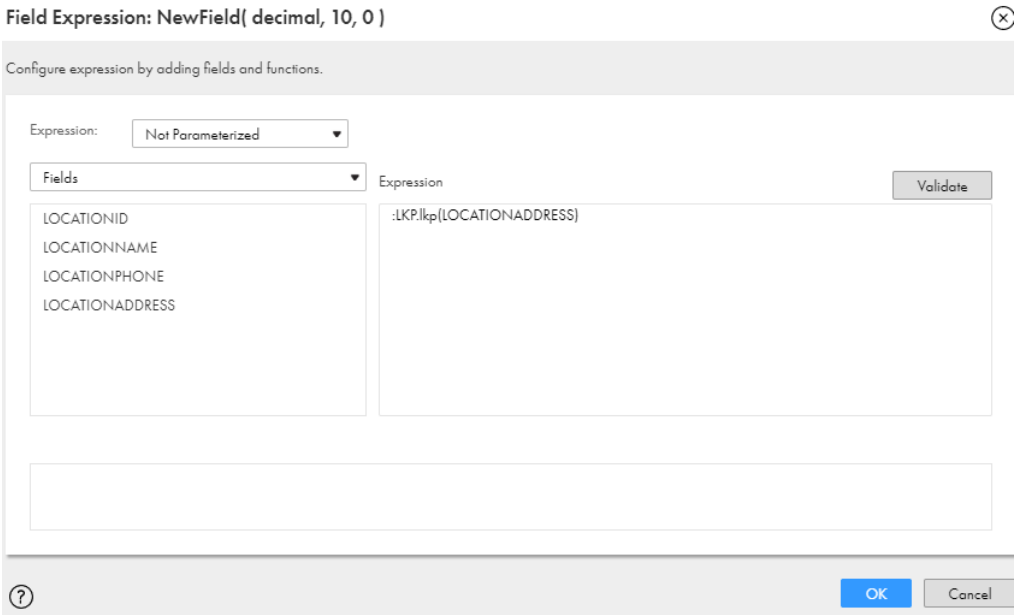
5. 戻り値を指定します。

   ルックアップトランスフォーメーションに複数の入力値を渡し、データのカラムを 1 つ返すことができます。データ統合は、ルックアップクエリから 1 つの値を返すことができます。戻りフィールドを使用して、戻り値を指定します。




6. 別のトランスフォーメーションのルックアップ式を設定します。

式トランスフォーメーションを使用するトランスフォーメーションの:LKP 式から、未接続のルックアップトランスフォーメーションの入力値を指定します。引数は、ルックアップ条件に使用されているルックアップトランスフォーメーションのローカル入力フィールドと一致するローカル入力ポートです。




7. フィールドをターゲットにマッピングします。

## ルックアップキャッシングの有効化

マッピングでルックアップトランスフォーメーションを設定すると、ランタイムセッション中にルックアップデータをキャッシュできます。

**[ルックアップキャッシングの有効化]** を選択すると、データ統合はルックアップソースを一度クエリし、セッション中に使用する値をキャッシュします。これにより、パフォーマンスを向上できます。キャッシュされたルックアップを保存するディレクトリを指定できます。

**ルックアップキャッシュの永続性**

ルックアップキャッシュの永続性を使用してルックアップキャッシュファイルを保存し、キャッシュを使用するように設定されたルックアップトランスフォーメーションを次回データ統合が処理するときにこのファイルが再利用されるようにします。

**[キャッシュファイル名のプレフィックス]** フィールドで、永続ルックアップキャッシュファイルで使用するファイル名プレフィックスを指定できます。

ルックアップテーブルが変更された場合は、**[ルックアップソースからの再キャッシュ]** プロパティを有効にして、ルックアップキャッシュを再構築できます。

**動的ルックアップキャッシュ**

動的ルックアップキャッシュを使用してルックアップキャッシュがターゲットに同期し続けるようにします。デフォルトでは、動的ルックアップキャッシュは無効になっており、静的キャッシュを表します。

キャッシュが静的である場合、ルックアップキャッシュ内のデータは、マッピングタスクの実行時に変更されません。

タスクがキャッシュを複数回使用する場合、タスクは同じデータを使用します。キャッシュが動的である場合、タスクはタスク内のアクションに基づいてキャッシュを更新します。そのため、タスクがルックアップを複数回使用する場合、ダウンストリームトランスフォーメーションでは更新されたデータが使用されます。

ルックアップキャッシングの詳細については、データ統合のマニュアルにある「*トランスフォーメーション*」を参照してください。

## ルックアップトランスフォーメーションの設定のルールおよびガイドライン

Amazon Redshift ルックアップトランスフォーメーションを設定する場合は、次のルールとガイドラインを考慮してください。

- 詳細モードでは、マッピングに接続されていないルックアップは使用できません。
- 詳細モードでは、マッピングの動的ルックアップ キャッシュは設定できません。
- キャッシュされていないルックアップに SQL クエリプロパティを使用することはできません。
- **［複数一致］**プロパティが［エラーを報告］に設定されているルックアップトランスフォーメーションは、詳細モードのマッピングでエラーメッセージが表示されることなく正常に実行されます。

## Oracle CDC ソースを使用したマッピングタスクの例

組織では、ミッションクリティカルな Oracle の本番システムからリアルタイムの変更データをレプリケートして、重大ではない作業の介入（オフラインのレポートシステムや分析処理システムなど）を最小限に抑える必要があります。Amazon Redshift V2 コネクタを使用すると、Oracle CDC ソースから変更データをキャプチャして Amazon Redshift のターゲットテーブルに書き込むことができます。Oracle CDC ソースをマッピングに追加して、関連付けられたマッピングタスクを実行すると、変更データがターゲットに書き込まれます。

1. データ統合で、**［新規］** > **［マッピング］** > **［作成］**をクリックします。

   **［新しいマッピング］**ダイアログボックスが表示されます。
2. マッピングの名前と説明を入力します。
3. ソーストランスフォーメーションの全般プロパティで、名前と説明を指定します。
4. **［ソース］**タブで、設定済みの Oracle CDC 接続を選択し、必要なソースプロパティを指定します。
5. ターゲットトランスフォーメーションの全般プロパティで、名前と説明を指定します。
6. **［ターゲット］**タブで、次の手順を実行してターゲットプロパティを設定します。
   a. **［接続］**フィールドで、Amazon Redshift V2 接続を選択します。
   b. **［ターゲットの種類］**フィールドで、ターゲットオブジェクトのタイプを選択します。
   c. **［オブジェクト］**フィールドで、必要なターゲットオブジェクトを選択します。
   d. **［操作］**フィールドで、ソースレコードの挿入、更新、削除が正しく処理されるように**［Data Driven］**を選択します。
   e. **［データ依存条件］**フィールドは空白のままにします。
   f. **［詳細プロパティ］**セクションで、必要なターゲットプロパティの値を指定します。**［書き込み時にレコードの順序を維持］**チェックボックスをオンにし、**［リカバリスキーマ名］**プロパティの値を入力する必要があります。
7. **［フィールドマッピング］**タブで、受信フィールドをターゲットフィールドにマップします。手動でマップすることも、フィールド名に基づいて自動的にマップすることもできます。
8. **［アクション］**メニューで、**［新しいマッピングタスク］**をクリックします。

   **［新しいマッピングタスク］**ページが表示されます。
9. **［定義］**タブで、タスク名を入力し、設定済みのマッピングを選択します。
10. **［CDC ランタイム］**タブで、必要なプロパティを指定します。

    **CDC ランタイム**プロパティの詳細については、Oracle CDC コネクタのヘルプを参照してください。

11. ［**スケジュール**］タブの［**詳細セッションプロパティ**］セクションで、次のプロパティを指定します。
    a. ［**ファイル終端（EOF）でコミット**］フィールドで、このプロパティの値として［**いいえ**］を選択します。
    b. ［**コミットタイプ**］フィールドで、このプロパティの値として［**ソース**］を選択します。
    c. ［**リカバリストラテジ**］フィールドで、このプロパティの値として［**最終チェックポイントから再開**］を選択します。
12. ［**保存**］ > ［**実行**］をクリックして、マッピングを保存して実行します。

    スケジュールを作成して、手動で操作しなくてもマッピングタスクが定期的に実行されるようにすることもできます。マッピングタスクの実行間隔ができるだけ短くなるようにスケジュールを定義できます。

**モニタ**で、タスクの実行後にログのステータスを監視できます。

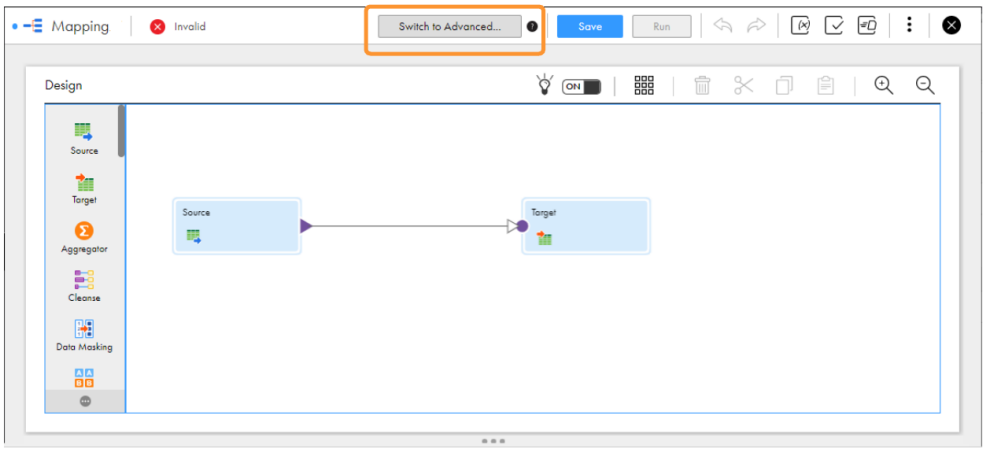
## 詳細モードのマッピングの例

顧客 ID、品目コード、および品目数量などの大量の注文書の詳細を Amazon S3 に保存している組織で働いているとします。注文書の詳細を分析して将来の収益を増加させるために、Amazon S3 から別のクラウドベースの環境にデータを移植する必要があります。

Amazon S3 からすべての注文レコードを読み取って Amazon Redshift ターゲットに書き込む場合のパフォーマンスを向上させるために、詳細モードで実行されるマッピングを作成します。

1. データ統合で、［**新規**］ > ［**マッピング**］ > ［**マッピング**］をクリックします。
2. Mapping Designer で、［**詳細に切り替え**］をクリックします。

   次の図は、Mapping Designer の［**詳細に切り替え**］ボタンを示しています。




3. ［**詳細に切り替え**］ダイアログボックスで、［**詳細に切り替え**］をクリックします。

   Mapping Designer でマッピングキャンバスが更新され、詳細モードで使用できるトランスフォーメーションと関数が表示されます。
4. マッピングの名前、場所、説明を入力します。
5. ソーストランスフォーメーションを追加して、全般プロパティで名前と説明を指定します。
6. Amazon S3 ソースからデータを読み取るには、[**ソース**]タブで、ソース詳細を指定する次の手順を実行します。
   a. ［**接続**］フィールドで、Amazon S3 V2 ソース接続を選択します。

b. **［ソースタイプ］**フィールドで、ソースのタイプを選択します。

c. **［オブジェクト］**フィールドで、必要なオブジェクトを選択します。

d. **[詳細プロパティ]**セクションで、適切な値を指定します。

7. **［フィールド］**タブで、Amazon S3 ソースフィールドをターゲットフィールドにマップします。
8. ターゲットトランスフォーメーションの全般プロパティで、名前と説明を指定します。
9. **[ターゲット]**タブで次の手順を実行して、Amazon Redshift ターゲットにデータを書き込むためのターゲットの詳細を指定します。

   a. **［接続］**フィールドで、Amazon Redshift V2 ターゲット接続を選択します。

   b. **［ターゲットタイプ］**フィールドで、ターゲットのタイプを選択します。

   c. **［オブジェクト］**フィールドで、必要なオブジェクトを選択します。

   d. **［操作］**フィールドで、必要な操作を選択します。

   e. **［詳細プロパティ］**セクションで、詳細ターゲットプロパティに適切な値を指定します。
10. Amazon S3 ソースおよび Amazon Redshift ターゲットをマップします。
11. **［保存］** > **［実行］**をクリックして、マッピングを検証します。

    ［監視］で、タスクの実行後にログのステータスを監視できます。

# SQL トランスフォーメーションを使用した SQL クエリの処理

Amazon Redshift V2 マッピングのミッドストリームで SQL クエリを処理するように SQL トランスフォーメーションを設定できます。詳細モードでは、マッピングに SQL トランスフォーメーションは設定できません。

SQL トランスフォーメーションをマッピングに追加する場合、［SQL］タブで、接続とトランスフォーメーションが処理する SQL のタイプを定義します。

## SQL トランスフォーメーションを使用した SQL クエリとストアドプロシージャの処理

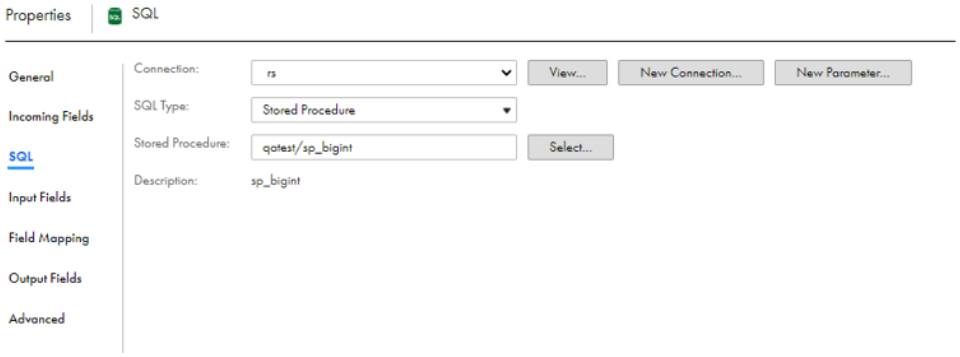
SQL トランスフォーメーションをマッピングに追加する場合は、［SQL］タブで、データベース接続およびトランスフォーメーションが処理する SQL のタイプを定義します。

SQL トランスフォーメーションは、次のタイプの SQL 文を処理できます。

**ストアドプロシージャ**

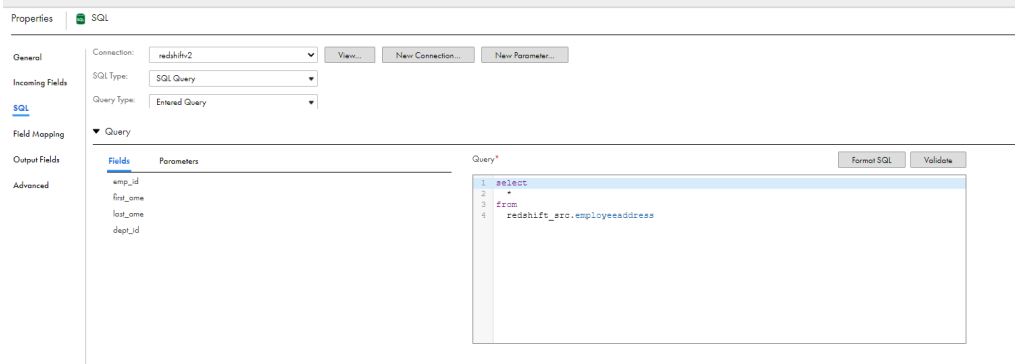
Amazon Redshift のストアドプロシージャを呼び出すように SQL トランスフォーメーションを設定できます。SQL トランスフォーメーションを作成する場合は、ストアドプロシージャが Amazon Redshift データベースに存在している必要があります。SQL トランスフォーメーションは、ストアドプロシージャを

処理する場合、入力パラメータをストアドプロシージャに渡します。ストアドプロシージャは戻り値をトランスフォーメーションの出力フィールドに渡します。




SQL **クエリ**

SQL エディタで定義した入力済みクエリを処理するように SQL トランスフォーメーションを設定できます。SQL トランスフォーメーションはクエリを処理し、行を返します。SQL トランスフォーメーションは、基になるデータベースからエラーが発生した場合、またはユーザー構文にエラーがある場合でもエラーを返します。



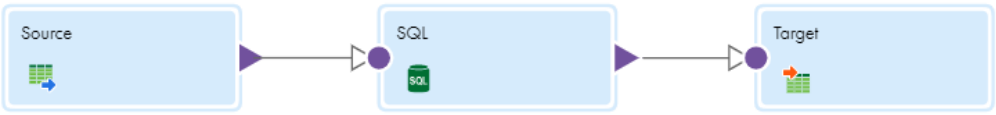

**注:** クエリを 1 つだけ指定します。複数の SQL クエリを指定することはできません。 保存済みクエリタイプは使用できません。

SQL クエリとストアドプロシージャの詳細については、データ統合のヘルプの「*トランスフォーメーション*」を参照してください。

## ストアドプロシージャの設定

SQL トランスフォーメーションの［**SQL**］タブで、ストアドプロシージャを処理するように SQL トランスフォーメーションを設定できます。

Amazon Redshift マッピングに SQL トランスフォーメーションを追加します。




SQL トランスフォーメーションで次のタスクを実行します。

1. SQL トランスフォーメーションの名前と説明を入力します。

2. ［**受信フィールド**］タブで、トランスフォーメーションに含めるデータを決定するフィールドルールを定義します。
3. SQL トランスフォーメーションの［**プロパティ**］パネルで［**SQL**］タブをクリックします。
4. ［**SQL**］タブで、次のタスクを実行します。
   a. データベースへの接続を選択します。

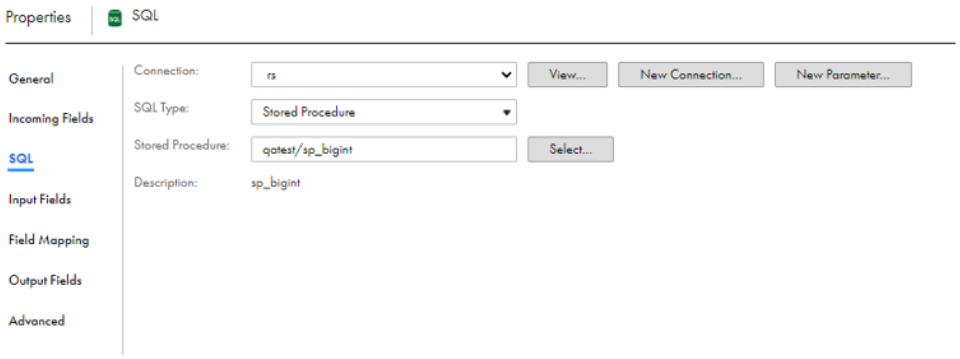
      接続を選択するか、パラメータを使用することができます。

   b. SQL タイプを［**ストアドプロシージャ**］に設定します。
   c. ［**選択**］をクリックしてデータベースからストアドプロシージャを選択するか、呼び出すストアドプロシージャの正確な名前を入力します。

      ストアドプロシージャ名では大文字と小文字が区別されます。

      **注:** マッピングが開いている状態で新しいストアドプロシージャをデータベースに追加すると、新しいストアドプロシージャは使用可能なストアドプロシージャのリストに表示されません。リストを更新するには、マッピングを閉じてから再度開きます。

   次の図に、設定済みの SQL トランスフォーメーションプロパティを示します。




5. ［**フィールドマッピング**］タブで、選択したストアドプロシージャの受信フィールドに入力フィールドをマッピングする方法を指定します。
6. 要件に応じて、トランスフォーメーションに詳細プロパティを定義します。

## ストアドプロシージャのルールとガイドライン

ストアドプロシージャについては、次のルールとガイドラインを考慮してください。

- ストアドプロシージャで使用されるテーブルと同じカラム名をソーステーブルで使用すると、マッピングタスクは失敗します。
- 例えば、ストアドプロシージャのカラムに対して `numeric(10,2)`などの数値データ型を指定すると、マッピングタスクは失敗します。
- マッピングでストアドプロシージャを作成する場合、出力ポートに refcursor データ型を含めることはできません。
- ストアドプロシージャに指定した入力パラメータが異なる場合でも、同じ名前の複数のストアドプロシージャから読み取ることはできません。

# マッピングに対する動的スキーマ処理

ユーザーがデータオブジェクトスキーマに加えた変更をデータ統合がどのように処理するかを選択できます。マッピングタスクを実行するたびにスキーマが更新されるようにするには、タスクで動的スキーマ処理を有効にします。

スキーマの変更には、データオブジェクトに対する次の 1 つ以上の変更が含まれます。

- フィールドの追加。
- フィールドのデータ型、精度、またはスケールの更新。

タスクの設定時に、［**スケジュール**］ページでスキーマ変更処理を設定します。

次の表に、スキーマ変更処理のオプションを示します。

| オプション | 説明 |
|---|---|
| 非同期 | デフォルト。データ統合は、マッピングまたはマッピングタスクの編集時および Informatica Intelligent Cloud Services のアップグレード時にスキーマを更新します。 |
| 動的 | データ統合は、タスクが実行されるたびにスキーマを更新します。<br>スキーマの更新では、次のオプションを選択できます。<br>- **変更して変更を適用する。** データ統合はターゲットスキーマを変更し、ソースから新しいフィールドを追加します。<br>- **現在のもの削除してから再作成する。**既存のターゲットテーブルを削除し、ソースからのすべての受信メタデータフィールドを使用して、実行時にターゲットテーブルを再作成します。<br>- DDL **の変更を適用しない。**データ統合はスキーマの変更をターゲットに適用しません。 |

詳細については、データ統合のヘルプの「*タスク*」にある「スキーマ変更処理」のトピックを参照してください。

## 動的スキーマ処理のルールとガイドラインマッピング

動的スキーマ変更処理を有効にする場合は、次のルールとガイドラインを考慮してください。

- データ型、精度、およびスケールの更新は適用されません。
- Varchar データ型の精度の低下を伴うスキーマ更新は適用されません。
- ターゲットテーブルのカラムで特殊文字を使用しており、［**変更して変更を適用する**］を選択した場合、エラーが発生してマッピングタスクは失敗します。
- ソーストランスフォーメーションのフィールドマッピングで完全にパラメータ化されたフィルタを使用しており、［**動的**］オプションを選択した場合、［**スケジュール**］ページには、スキーマを更新するためのオプションが表示されません。
- ソーストランスフォーメーションで SQL オーバーライドを使用しており、［**変更を変更して適用する**］を選択した場合、エラーが発生してマッピングタスクは失敗します。
- ソースにプライマリキーとしてフィールドを追加し、［**変更して変更を適用する**］を選択した場合、マッピングタスクが正常に実行され、フィールドがターゲットにプロパゲートされます。ただし、プライマリキーは考慮されません。
- ソーステーブル名またはスキーマ名をオーバーライドして、［**変更して変更を適用する**］スキーマ変更処理オプションを選択し、オーバーライドされたソーステーブルに新しいフィールドが追加されると、マッピングは正常に実行されます。ただし、Secure Agent は、オーバーライドされたソーステーブルから新しく追加されたフィールドを取得しません。

## キー範囲パーティションの設定

フィールド値に基づいて Amazon Redshift データをパーティション化するようにキー範囲パーティションを設定します。

1. ［**ソースのプロパティ**］で、［**パーティション**］タブをクリックします。
2. リストから必要な［**パーティションキー**］を選択します。
3. ［**新しいキー範囲の追加**］をクリックしてパーティションを追加します。
4. ［**開始範囲**］および［**終了範囲**］を指定します。

## 書き込み操作の一括処理

一括処理を有効にして、Amazon Redshift に大量のデータを書き込むことができます。一括処理では、最小限の API 呼び出しを使用し、書き込み操作のパフォーマンスが向上します。

一括処理を有効にするには、Secure Agent プロパティでプロパティ -*DENABLE_WRITER_BULK_PROCESSING=true* を指定します。

マッピングを実行する前に、以下の手順を実行して一括処理を設定します。

1. Administrator で、［**ランタイム環境**］タブにリストされている Secure Agent を選択します。
2. ［**編集**］をクリックします。
3. ［**システム構成の詳細**］セクションで、サービスとして［**データ統合サービス**］を選択し、タイプとして［**DTM**］を選択します。
4. **JVMOption1** プロパティを編集し、*-DENABLE_WRITER_BULK_PROCESSING=true* と入力します。
5. ［**保存**］をクリックします。

**注:** プッシュダウンの最適化を使用して設定したマッピングタスクには適用されません。

# 第 9 章

# マッピングの移行

ある環境で接続とマッピングを設定してから別の環境にマッピングを移行して実行すると、データ統合は以前の環境で設定されたランタイム属性を使用して、新しい環境でマッピングを正常に実行します。

移行後は、管理者サービスから接続プロパティを変更できますが、マッピングを変更する必要はありません。

開発組織（組織 1）でマッピングを開発し、次にプロダクション組織（組織 2）でマッピングを移行して実行するシナリオについて考えてみます。移行後に、組織 2 では同じまたは異なる接続エンドポイントあるいはオブジェクトパスを使用したい場合があるでしょう。要件に応じて、移行を計画する前に、この節のガイドラインを考慮してください。

## 移行されたマッピングへの同じオブジェクトパスの使用

組織 2 に移行されたマッピングで組織 1 と同じオブジェクトパスを使用する場合は、組織 2 の Amazon Redshift アカウントで同じデータベース、スキーマ、およびテーブルを使用する必要があります。

例えば、組織 1 に使用されるアカウント 1 と組織 2 に使用されるアカウント 2 という 2 つの異なるアカウントがある場合、スキーマおよびテーブル名のオブジェクトパスは、次のように両方のアカウントで同じである必要があります。

アカウント 1: SCHEMA1/TABLE1

アカウント 2: SCHEMA1/TABLE1

このシナリオでは、詳細プロパティのスキーマとテーブルをオーバーライドする必要はありません。

## 移行されたマッピングへの別のオブジェクトパスの使用

マッピングを移行した後に、別のオブジェクトパスを使用して、新しい環境からマッピングを実行できます。

このシナリオでは、マッピングを移行する前に、オブジェクトメタデータまたはランタイム属性を変更して、移行された環境にオブジェクトパスを反映させることができます。新しい環境でマッピングを編集または更新する必要はありません。

原則として、詳細プロパティまたはオブジェクトプロパティでスキーマとテーブルを指定すると、データ統合は次の優先順位で属性を使用します。

1. **ランタイムの詳細属性**。マッピングのソース、ターゲット、またはルックアップトランスフォーメーションのスキーマ名やテーブル名などの詳細プロパティ。
2. **オブジェクトメタデータ**。マッピングのソース、ターゲット、またはルックアップトランスフォーメーションで選択したオブジェクトタイプ。

# 移行オプション

移行を行うときは、次のオプションのいずれかを選択して、オブジェクトパスを更新できます。

## オプション 1: 接続プロパティを更新して、新しいオブジェクトを参照する

マッピングを組織 2 にインポートすると、**[接続の確認]** セクションで既存の接続を変更して、組織 2 の指定されたスキーマおよびテーブルにアクセス可能な接続にマッピングできます。

## オプション 2: 詳細プロパティからプロパティをオーバーライドする

移行の前に、組織 1 のマッピングの詳細プロパティで、組織 2 のオブジェクトに必要なスキーマとテーブル名を指定します。

移行後にマッピングを実行すると、Secure Agent は設定された詳細パラメータを使用して、指定されたオブジェクトを組織 1 からインポートされたマッピングでオーバーライドします。

## オプション 3: マッピングのプロパティをパラメータ化する

移行前に、スキーマやテーブル名などの詳細属性をパラメータ化するという選択ができます。マッピングでは、入力パラメータ、入出力パラメータ、およびパラメータファイルを設定できます。

マッピングを移行した後は、マッピングを編集または更新しないでください。入出力パラメータを使用した場合は、パラメータファイルを使用してスキーマとテーブルの属性を変更し、タスクの実行時に変更が適用されるようにすることができます。

**マッピング内のオブジェクトではなく、詳細プロパティのみをパラメータ化する**

詳細プロパティのみをパラメータ化し、実行時にそれらのプロパティを使用する場合は、マッピングのオブジェクトプロパティでプレースホルダオブジェクトを選択してから、詳細プロパティからこのプレースホルダオブジェクトへのオーバーライドを指定します。プレースホルダオブジェクトに、オーバーライドとして指定した対応するテーブルと同じメタデータが含まれていることを確認してください。マッピングを実行すると、詳細プロパティで指定した値でプレースホルダオブジェクトがオーバーライドされます。

**オブジェクトと詳細プロパティをパラメータ化する**

Amazon Redshift オブジェクトタイプと詳細フィールドをパラメータ化したままにする場合は、パラメータの追加時に入力パラメータウィンドウで **[実行時にパラメータのオーバーライドを許可]** オプションが選択されていない状態で、必要なオブジェクトをタスクレベルで選択します。タスクを実行すると、詳細プロパティで指定した値が優先されます。

## パラメータ化ルール

オブジェクトと詳細プロパティをパラメータ化する場合は、次のルールを考慮してください。

- パラメータ化は、**[ターゲットの作成]** オプションを使用するマッピングには適用されません。
- 移行の使用例では、入力パラメータのみが詳細モードのマッピングに適用されます。
- マッピングに複数のパイプラインが設定されている場合は、Amazon Redshift オブジェクトをパラメータ化しないでください。移行する前に、マッピングの作成時にプレースホルダオブジェクトを選択する必要があります。

## マッピングの移行のルールとガイドライン

移行されたマッピングに同じオブジェクトパスまたは異なるオブジェクトパスを使用する場合は、次のルールとガイドラインを考慮してください。

- 次の表に、トランスフォーメーション、オブジェクトタイプ、および新しい環境に移行したときに保持されるマッピングの詳細プロパティのフィールドを示します。

| トランスフォーメーション | オブジェクトタイプ | 詳細フィールド |
|---|---|---|
| ソース | 単一オブジェクト、複数のオブジェクト | スキーマとテーブル名 |
| ルックアップ | 単一オブジェクト<br>**注**: 未接続または接続済みの、キャッシュされたオブジェクトおよびキャッシュされていないオブジェクトに適用されます。 | |
| ターゲット | 単一オブジェクト | |

- マッピングを組織 2 に移行する前に、組織 1 の接続 1 を組織 2 にマッピングします。これにより、組織 1 の詳細プロパティで設定されたスキーマとテーブル名にアクセスできるようになります。
- マッピングを組織 2 に移行した後は、マッピングを編集しないでください。
- マッピングでスキーマとテーブル名をオーバーライドすると、組織 2 の接続で使用されているスキーマが無効な場合、エラーが発生してマッピングは失敗します。マッピングが失敗して、次のエラーが表示されます:
  `An unexpected exception occured while fetching metadata:[The following schema is not found - invalidschema]`
- 実行時にデータオブジェクトスキーマを動的に更新することはできません。スキーマ変更処理は適用されないため、ソース、ターゲット、またはルックアップトランスフォーメーションで選択されたテーブルと対応する詳細フィールドのオーバーライドに対して同じメタデータを使用する必要があります。
- マッピングで SQL オーバーライドを指定しないでください。指定した場合、マッピングの実行が失敗します。
- 複数のソースがある場合は、次のガイドラインを検討してください。
  - マッピングに複数のソースが含まれている場合は、次の例のように、ソースオブジェクトが同じスキーマを持つ場合にのみ、詳細プロパティのスキーマをオーバーライドできます。
    オブジェクト 1: `<schema1.object1>`

    オブジェクト 2: `<schema1.object2>`

    オブジェクト 1 のスキーマが`<schema1.object1>`で、オブジェクト 2 が`<schema2.object2>`などのようにスキーマが異なる場合、移行は機能しません。
  - マッピングに複数のオブジェクトが含まれている場合は、詳細プロパティからテーブルをオーバーライドしないでください。

# 第 10 章

# 接続タイプのアップグレード

Amazon Redshift V1 接続または Amazon Redshift ODBC 接続を使用して Amazon Redshift にアクセスしている場合は、新しい Amazon Redshift V2 コネクタにアップグレードできます。Amazon Redshift 接続または Amazon Redshift ODBC 接続を使用する既存のマッピングおよびマッピングタスクのソースまたはターゲット接続タイプを Amazon Redshift V2 接続に置き換えることができます。

既存のマッピングで接続を置き換えると、以前に選択したオブジェクトは保持されません。Amazon Redshift オブジェクトを再インポートする必要があります。2 つのコネクタ間で共通のフィールドに設定されたソース、ターゲット、およびルックアップの詳細プロパティは、新しいコネクタに保持されます。また、古いコネクタからの設定済みのフィールドマッピングを保持するためのオプションもあります。

古いコネクタの設定済みの値を使用して、マッピングを正常に実行できます。さらに、拡張された Amazon Redshift V2 コネクタが提供する機能を設定できます。

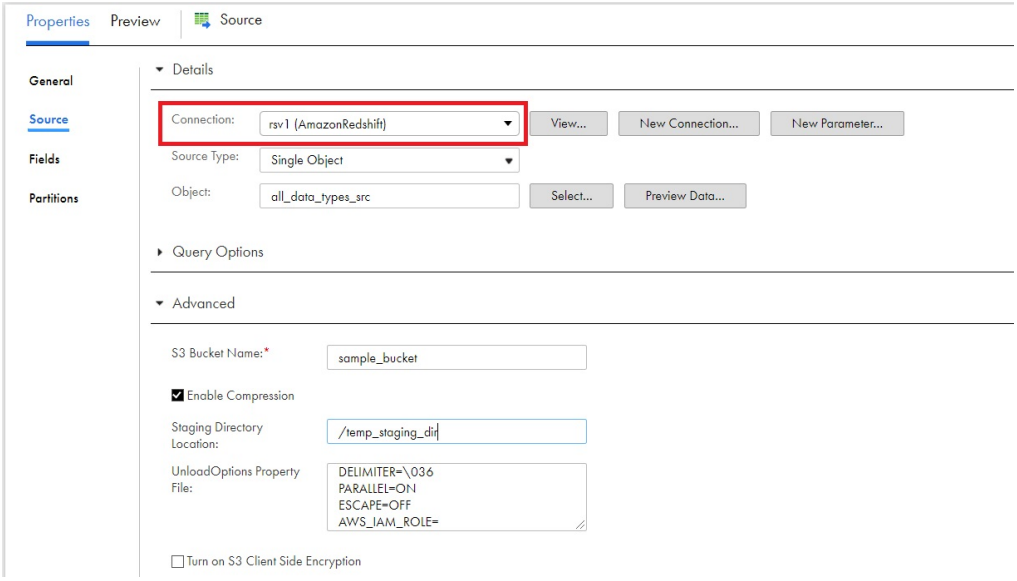
**注:** マッピングで Amazon Redshift V1 接続を使用して Amazon Redshift との間でデータの読み取りまたは書き込みを行う場合は、Amazon Redshift V2 接続を使用して、拡張コネクタが提供する機能を利用することをお勧めします。Amazon Redshift V2 コネクタのライセンスを取得するには、グローバルカスタマサポートにお問い合わせください。

# 接続切り替えの例

Amazon Redshift V1 接続を使用する既存の Amazon Redshift マッピングを Amazon Redshift V2 接続にアップグレードするとします。

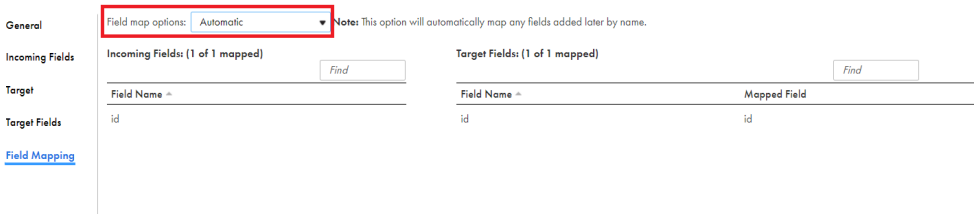
1. Amazon Redshift V2 にアップグレードする既存の Amazon Redshift V1 マッピングを開きます。

   次の図に、Amazon Redshift V1 接続を使用する、ソーストランスフォーメーションで設定された詳細プロパティを含んだ既存のマッピングを示します。




2. 接続を切り替えたときにフィールドマッピングからのマッピングされたフィールドを保持するには、**［フィールドマッピング］**タブで、Amazon Redshift V1 マッピングの次の**［フィールドマップオプション］**メニューから選択を行います。

   - 切り替え後に、自動的にマッピングされたフィールドを保持するには、**［自動］**を選択します。




   - 切り替え後に、保持されたフィールドを手動でマッピングするには、**［手動］**を選択します。

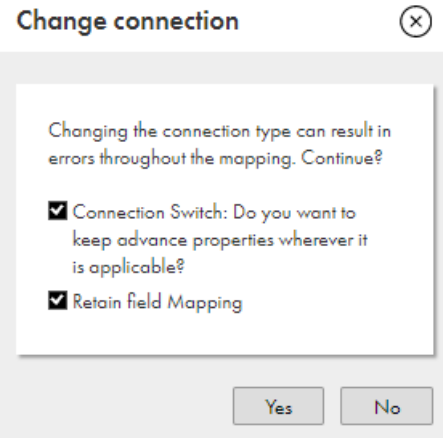
     **注:** 手動を選択した場合は、接続を切り替えた後に、保持されたフィールドを以前のマッピングを使用して自動的にマッピングするオプションがあります。

3. 接続を切り替えるには、**［接続］**フィールドで、接続を Amazon Redshift V1 から Amazon Redshift V2 に変更します。

4. **［接続の変更］**ダイアログボックスで**［接続の切り替え］**プロパティを選択し、**［はい］**をクリックします。

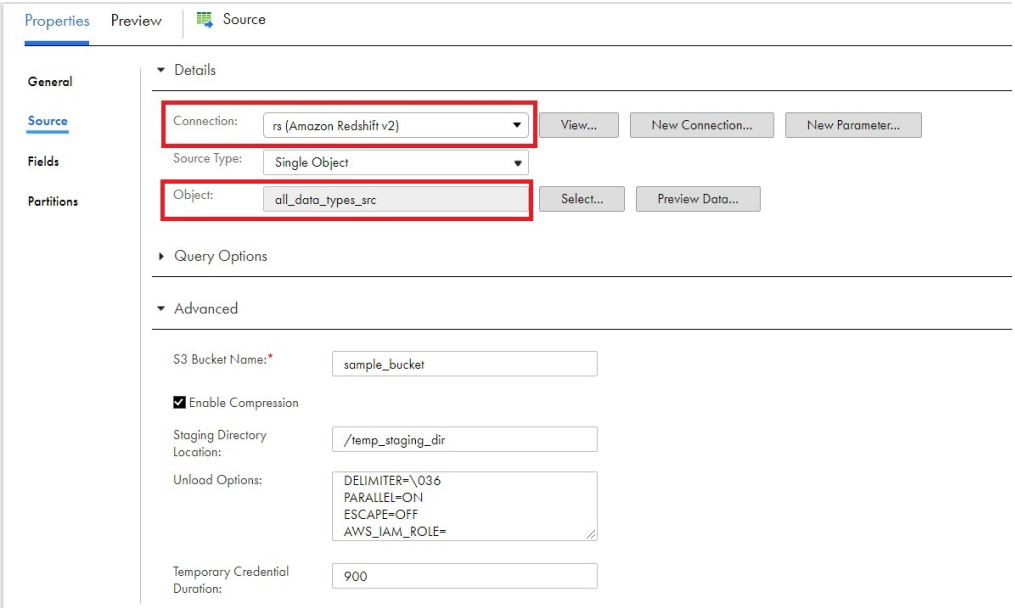
   - **接続の切り替え**。選択した接続に切り替えます。
   - **フィールドマッピングの保持**。Amazon Redshift V1 からの設定済みのフィールドマッピングを保持します。

次の図に、選択する必要のあるオプションを示します。



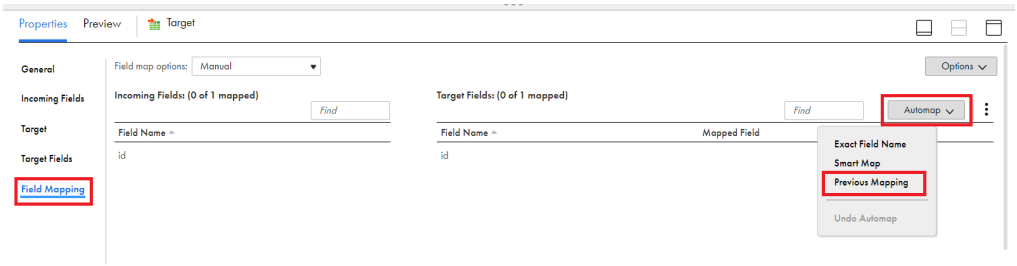

5. マッピングでは、Amazon Redshift V1 と同じオブジェクトパスを使用します。

   次の図に、Amazon Redshift V1 と同じオブジェクトパスを使用して切り替えた接続を示します。




Amazon Redshift V1 マッピングから設定されたソースの詳細プロパティが、ソーストランスフォーメーションに反映されています。

6. Amazon Redshift V1 マッピングの［**フィールドマッピング**］タブで［**手動**］を選択し、Amazon Redshift V2 のフィールドマッピングを反映するには、［**フィールドマッピング**］タブで［**自動マップ**］を選択し、［**以前のマッピング**］を選択します。




**注:** Amazon Redshift V1 で［**自動**］を選択した場合、このタスクを実行する必要はありません。

7. ［**保存**］をクリックします。

# 切り替え後に保持される詳細プロパティ

Amazon Redshift V1 接続または Amazon Redshift ODBC 接続を使用する既存のマッピングまたはマッピングタスクのソースまたはターゲット接続タイプを Amazon S3 V2 接続に置き換える場合に、設定された詳細プロパティを保持するためのオプションがあります。

次の表に、Amazon Redshift V2 マッピングに保持される Amazon Redshift ODBC からの設定済みのソース、ルックアップ、およびターゲットの詳細プロパティを示します。

| **プロパティ** | Amazon Redshift V2 **に保持される** Amazon Redshift ODBC **プロパティ** |
|---|---|
| ソース | - Pre SQL<br>- Post SQL<br>- SQL オーバーライド |
| ターゲット | - Pre SQL<br>- Post SQL |

次の表に、Amazon Redshift V2 マッピングに保持される Amazon Redshift V1 の設定済みのソース、ルックアップ、およびターゲットの詳細プロパティを示します。

| **プロパティ** | Amazon Redshift V2 **に保持される** Amazon Redshift V1 **のプロパティ** |
|---|---|
| ソースとルックアップ | - S3 バケット名<br>- 圧縮を有効にする<br>- ステージングディレクトリの場所<br>- Unload オプション<br>- 暗号化タイプ<br>- S3 ファイルのマルチパートダウンロード<br>- ソーステーブル名<br>- Pre-SQL<br>- Post-SQL<br>- SQL クエリ<br>- 個別選択 |
| ターゲット | - S3 バケット名<br>- 圧縮を有効にする<br>- ステージングディレクトリの場所<br>- S3 クライアント側の暗号化<br>- S3 サーバー側の暗号化<br>- Pre-SQL<br>- Post-SQL<br>- バッチサイズ<br>- バッチアップロード挿入ごとの最大エラー数<br>- データアップロード前のターゲットテーブルの切り詰め<br>- Char および Varchar に NULL 値が必要<br>- S3 ファイルの整合性のための待ち時間（秒）<br>- コピーオプション<br>- ターゲットテーブルのクリーンアップ<br>- ターゲットテーブルの分析<br>- S3 でステージングファイルを保持するプレフィックス<br>- ターゲットテーブル名<br>- 最小アップロードパートサイズ<br>- 成功ファイルディレクトリ<br>- エラーファイルのディレクトリ<br>- TransferManager スレッドプールサイズ |

# ルールおよびガイドライン

既存のマッピングの Amazon Redshift ODBC 接続または Amazon Redshift V1 接続を Amazon Redshift V2 接続に置き換える場合は、次のルールを考慮してください。

- Amazon Redshift のソースプロパティとターゲットプロパティで Pre-SQL、Post-SQL、および SQL クエリコマンドを指定し、Amazon Redshift V2 への接続をアップグレードすると、クエリは切り詰められます。この問題は、Amazon Redshift ODBC 接続または Amazon Redshift V1 接続のクエリの文字数が 65535 文字を超えると発生します。
- 接続を Amazon Redshift V2 接続に置き換えた後も、オブジェクトフィールドには、以前の接続で選択したオブジェクトが表示されます。Amazon Redshift オブジェクトを再インポートし、マッピングのフィールドを再マッピングする必要があります。

# 第 11 章

# Amazon Redshift のプッシュダウンの最適化

プッシュダウンの最適化を使用して、トランスフォーメーションロジックをターゲットデータベースにプッシュできます。データベースにプッシュできるトランスフォーメーションロジックの量は、データベース、トランスフォーメーションロジック、およびタスク設定によって異なります。Secure Agent では、Secure Agent がデータベースにプッシュできないすべてのトランスフォーメーションロジックが処理されます。詳細モードでマッピングを実行した場合、プッシュダウンの最適化は使用できません。

## プッシュダウンの最適化のタイプ

プッシュダウンの最適化を適用すると、タスクのプロパティで指定した最適化のタイプに基づいて、タスクは、トランスフォーメーションロジックをソースデータベースまたはターゲットデータベースにプッシュします。データ統合は、トランスフォーメーションロジックを SQL クエリに変換するか、Amazon Redshift コマンドを Amazon Redshift データベースに変換します。データベースでは、その SQL クエリまたは Amazon Redshift コマンドを実行してトランスフォーメーションを処理します。

マッピングでは、次のタイプのプッシュダウンの最適化を設定できます。

**なし**

このタスクは、トランスフォーメーションロジックを Amazon Redshift データベースにプッシュダウンしません。

Full

このタスクは、ターゲットデータベースで処理するために可能な限り多くのトランスフォーメーションロジックをプッシュします。トランスフォーメーションがマッピングでサポートされていない場合、タスクは、プッシュダウンの最適化のためにトランスフォーメーションがサポートされるポイントまでマッピングロジックを部分的にプッシュダウンします。

**ソース**

このタスクは、ソースデータベースで処理するために可能な限り多くのトランスフォーメーションロジックをプッシュダウンします。ソースのプッシュダウンの最適化を選択すると、タスクは、ターゲットトランスフォーメーションに到達するまで、設定されたすべてのトランスフォーメーションのトランスフォーメーションロジックをマッピングのダウンストリームにプッシュします。また、ターゲットトランスフォーメーションロジックを省略し、残りのマッピングロジックをソースデータベースにプッシュダウンします。

## ソースのプッシュダウンの最適化と完全なプッシュダウンの最適化を使用したデータ統合の動作

Amazon Redshift に対する読み書きを行うマッピングに対して完全なプッシュダウンの最適化またはソースのプッシュダウンの最適化を選択すると、データ統合は、マッピングがソースからターゲットまで到達するのかダウンストリームトランスフォーメーションまで到達するのかを分析し、処理用のマッピングロジックの一部のみをデータベースにプッシュダウンするか全体をプッシュダウンするかを決定します。

マッピング内のすべてのトランスフォーメーションがプッシュダウンの最適化でサポートされている場合、タスクはマッピングロジック全体をデータベースにプッシュダウンします。

トランスフォーメーションがマッピングでサポートされていない場合、タスクは、プッシュダウンの最適化のためにトランスフォーメーションがサポートされるポイントまでマッピングロジックを部分的にプッシュダウンします。データ統合は、プッシュダウンする必要のあるトランスフォーメーションロジックの SELECT ステートメントを生成して実行します。次に、データ統合サービスはこの SQL クエリの結果が読み込み、マッピングの残りのトランスフォーメーションを処理します。

# プッシュダウンの最適化のシナリオ

マッピングで Amazon Redshift V2 コネクタまたは Amazon Redshift ODBC コネクタを使用してプッシュダウンの最適化を設定できます。

### Amazon Redshift V2 コネクタ

マッピングで Amazon Redshift V2 コネクタを使用する場合は、次のようなシナリオでプッシュダウンの最適化の設定を使用できます。

| ソースエンドポイントとターゲットエンドポイント | マッピングでサポートされているプッシュダウンシナリオ | プッシュダウンの最適化のタイプ |
|---|---|---|
| Amazon S3 ソース<br>Amazon Redshift ターゲット | Amazon S3 から読み取り、Amazon Redshift V2 接続を使用して Amazon Redshift に書き込みます。<br>データをソースからターゲットに読み取るときに、この接続は AWS コマンドを使用します。 | 完全 |
| Amazon Redshift ソース<br>Amazon Redshift ターゲット | Amazon Redshift V2 接続を使用して、Amazon Redshift から読み取り、Amazon Redshift に書き込みます。<br>ソースとターゲットは、同じクラスタ領域または異なるクラスタ領域に存在することができます。 | ソース、完全<br>**注:** Secure Agent は、処理用のマッピングロジック全体、または必要に応じて部分的なマッピングロジックのみを Amazon Redshift にプッシュします。 |
| Amazon Redshift ソース | Amazon Redshift V2 接続を使用して Amazon Redshift から読み取り、他のターゲットに書き込みます。 | ソース |
| **注:** Secure Agent または Hosted Agent を使用して、プッシュダウンの最適化が設定されたマッピングを実行できます。 | | |

### ODBC コネクタ

**重要:** マッピングで Amazon Redshift V2 接続を使用して、プッシュダウンの最適化を設定することをお勧めします。Amazon Redshift V2 接続を使用して特定のトランスフォーメーションロジックをプッシュダウンできない場合は、Amazon Redshift ODBC 接続を使用したプッシュダウンの最適化の設定を試すことができます。

Amazon Redshift ODBC 接続を使用して Amazon Redshift に対してデータの読み取りおよび書き込みを行うマッピングのプッシュダウンを設定できます。

| ソースエンドポイントとターゲットエンドポイント | マッピングでサポートされているプッシュダウンシナリオ | プッシュダウンの最適化のタイプ |
|---|---|---|
| Amazon Redshift ソース<br>Amazon Redshift ターゲット | Redshift ODBC サブタイプで Amazon Redshift ODBC 接続を使用して、Amazon Redshift に対する読み取りおよび書き込みを行います。<br>データをソースからターゲットに読み取るときに、この接続はデータベースコマンドを使用します。 | ソース、完全 |
| **注**: Secure Agent を使用して、プッシュダウンの最適化が設定されたマッピングを実行できます。 | | |

**注:** 詳細モードのマッピングに基づくマッピングタスクにプッシュダウンの最適化を設定することはできません。

# Amazon Redshift V2 マッピングタスクに対する最適化の設定

次の手順を実行して、Amazon Redshift V2 マッピングタスクに対するプッシュダウンの最適化を設定します。

1. 要件に応じてマッピングを作成します。
   - Amazon S3 から読み取りを行って Amazon Redshift に書き込むには、ソーストランスフォーメーションで Amazon S3 V2 接続を使用し、ターゲットトランスフォーメーションで Amazon Redshift V2 接続を使用します。
   - Amazon Redshift との間で読み取りまたは書き込みを行うには、ソーストランスフォーメーションとターゲットトランスフォーメーションで Amazon Redshift V2 接続を使用します。
2. マッピングタスクを作成します。
   a. 設定されたマッピングを選択します。
   b. **［スケジュール］**タブの**［プッシュダウンの最適化］**セクションで、プッシュダウンの最適化の値を**［完全］**または**［ソースへ］**に設定します。

      ソースプッシュダウンの最適化は、Amazon Redshift ソースから読み取りを行う場合にのみ使用します。

      **注:** 最適化コンテキストタイプオプションは、Amazon Redshift V2 マッピングタスクには適用されません。
   c. タスクを保存し、**［終了］**をクリックします。

マッピングタスクの実行時に、トランスフォーメーションロジックが Amazon Redshift データベースにプッシュされます。プッシュダウンの最適化の実行を確認するには、ジョブのセッションログをチェックします。**［マイジョブ］**ページで、開始したジョブを監視することができます。

# Amazon Redshift V2 接続を使用したプッシュダウンの最適化

Amazon Redshift V2 接続を含むマッピングのプッシュダウンの最適化を設定できます。プッシュダウンの最適化により、マッピングのパフォーマンスが向上します。

プッシュダウンの最適化用に設定されたタスクを実行すると、トランスフォーメーションロジックが Amazon Redshift クエリに変換されます。タスクでクエリが Amazon Redshift に送信され、Amazon Redshift データベースでマッピングロジックが処理されます。

マッピングでプッシュダウンの最適化を設定して、Amazon Redshift V2 接続を使用して Amazon Redshift から読み取ることができる場合は、テーブル、外部テーブル、ビュー、およびマテリアライズ済みのビューからデータを読み取ることができます。

**例**

急成長を遂げているデータサイエンス組織で働いているとします。この組織は、財務分析を行うソフトウェア製品を開発して、個人のプロフィール、企業、職務、広告主、および出版社をつなぐ財務グラフを作成しています。組織では、Amazon Web サービスに基づいたインフラストラクチャを使用して、Amazon S3 にそのデータを保存します。組織では、可視化を作成してリアルタイム分析を実現するためにビジネスインテリジェンスサービスを実装する予定です。AWS コマンドを使用してトランスフォーメーションを設定することによって Amazon S3 から Amazon Redshift までデータをロードし、適切なデータウェアハウスモデルと使用要件をサポートできます。

Amazon S3 V2 接続を作成して、Amazon S3 ソースからデータを読み取ります。Amazon Redshift V2 接続を作成し、プッシュダウン最適化を使用して Amazon Redshift ターゲットにデータを書き込みます。プッシュダウン最適化で Amazon Redshift V2 接続を使用すると、パフォーマンスが向上し、関連するコストを削減できます。

## プッシュダウン互換性

トランスフォーメーション、関数、および演算子をデータベースにプッシュするようにタスクを設定できます。

プッシュダウンの最適化を使用した場合、Secure Agent はデータベース内で同等の演算子、関数を特定することで、トランスフォーメーションの式を変換します。対応する演算子および関数が存在しない場合、Secure Agent はトランスフォーメーションロジックを処理します。

### Amazon Redshift V2 マッピングでサポートされている関数

プッシュダウンの最適化を使用する場合、Secure Agent はデータベース内で同等の関数を特定することで、トランスフォーメーションの式を変換します。同等の関数がない場合、Secure Agent はトランスフォーメーションロジックを処理します。

次の表に、Amazon Redshift データベースでのプッシュダウン関数の可用性を示します。X の付いたカラムは、関数を Amazon Redshift データベースにプッシュできることを示します。ダッシュ（-）記号の付いたカラムは、関数をデータベースにプッシュできないことを示しています。

| 機能 | プッシュダウン | 機能 | プッシュダウン | 機能 | プッシュダウン |
|---|---|---|---|---|---|
| ABORT() | - | INITCAP() | X | REG_MATCH() | - |
| ABS() | X | INSTR() | X | REG_REPLACE | - |

| 機能 | プッシュダウン | 機能 | プッシュダウン | 機能 | プッシュダウン |
|---|---|---|---|---|---|
| ADD_TO_DATE() | X | IS_DATE() | - | REPLACECHR() | X |
| AES_DECRYPT() | - | IS_NUMBER() | X | REPLACESTR() | X |
| AES_ENCRYPT() | - | IS_SPACES() | X | REVERSE() | - |
| ASCII() | - | ISNULL() | - | ROUND(DATE) | - |
| AVG() | X | LAST() | - | ROUND(NUMBER) | X |
| CEIL() | X | LAST_DAY() | X | RPAD() | X |
| CHOOSE() | - | LEAST() | - | RTRIM() | X |
| CHR() | X | LENGTH() | X | SET_DATE_PART() | - |
| CHRCODE() | - | LN() | X | SIGN() | X |
| COMPRESS() | - | LOG() | X | SIN() | X |
| CONCAT() | X | LOOKUP() | X | SINH() | - |
| COS() | X | LOWER() | X | SOUNDEX() | - |
| COSH() | - | LPAD() | X | SQRT() | X |
| COUNT() | X | LTRIM() | X | STDDEV() | X |
| CRC32() | - | MAKE_DATE_TIME() | - | SUBSTR() | X |
| CUME() | - | MAX() | X | SUM() | X |
| DATE_COMPARE() | X | MD5() | X | SYSTIMESTAMP() | X |
| DATE_DIFF() | X | MEDIAN() | X | TAN() | X |
| DECODE() | X | METAPHONE() | - | TANH() | - |
| DECODE_BASE64() | - | MIN() | X | TO_BIGINT | X |
| DECOMPRESS() | - | MOD() | X | TO_CHAR(DATE) | X |
| ENCODE_BASE64() | - | MOVINGAVG() | - | TO_CHAR(NUMBER) | X |
| EXP() | X | MOVINGSUM() | - | TO_DATE() | X |
| FIRST() | - | NPER() | - | TO_DECIMAL() | X |
| FLOOR() | X | PERCENTILE() | - | TO_FLOAT() | X |
| FV() | - | PMT() | - | TO_INTEGER() | X |
| GET_DATE_PART() | X | POWER() | X | TRUNC(DATE) | X |

| 機能 | プッシュダウン | 機能 | プッシュダウン | 機能 | プッシュダウン |
|---|---|---|---|---|---|
| GREATEST() | - | PV() | - | TRUNC(NUMBER) | X |
| IIF() | X | RAND() | - | UPPER() | X |
| IN() | - | RATE() | - | VARIANCE() | X |
| INDEXOF() | - | REG_EXTRACT() | - | | |

## Amazon Redshift V2 マッピングでサポートされている演算子

プッシュダウンの最適化を使用する場合、Secure Agent はデータベース内で同等の演算子を特定することで、トランスフォーメーションの式を変換します。同等の演算子がない場合、Secure Agent はトランスフォーメーションロジックを処理します。

次の表に、Amazon Redshift データベースで使用できるプッシュダウン演算子を示します。X マークの付いたカラムは、完全なプッシュダウンの最適化を使用して、演算子を Amazon Redshift データベースにプッシュできることを示しています。

| 演算子 | プッシュダウン |
|---|---|
| + | X |
| - | X |
| * | X |
| / | X |
| % | X |
| \|\| | X |
| > | X |
| < | X |
| = | X |
| >= | X |
| <= | X |
| ! = | X |
| AND | X |
| OR | X |
| NOT | X |

## Amazon Redshift V2 マッピングでサポートされているトランスフォーメーション

プッシュダウンの最適化を設定すると、Secure Agent は、Amazon Redshift に対する設定済みのトランスフォーメーションのプッシュを試行します。

次の表に、Amazon Redshift にプッシュダウンできるトランスフォーメーションの利用可否を示します。X の付いたカラムは、トランスフォーメーションを Amazon Redshift にプッシュできることを示しています。

| トランスフォーメーション | プッシュダウン | ガイドライン |
|---|---|---|
| Aggregator | X<br>次の集計を実行できます。<br>- AVG<br>- COUNT<br>- MAX<br>- MIN<br>- MEDIAN<br>- SUM<br>- VARIANCE | アグリゲータトランスフォーメーションを設定し、入力フィールドが集計関数またはグループ化フィールドで使用されていない場合、ANY_VALUE（）関数はフィールドからの値を返すため、出力は確定的ではありません。 |
| Expression | X | - |
| フィルタ | X | - |
| Joiner | X | 2 つの Amazon S3 ソースまたは 2 つの Amazon Redshift ソース間でジョイナトランスフォーメーションを設定できます。 |
| Lookup | 接続済みのルックアップと未接続のルックアップの両方が適用可能です。 | マッピングでルックアップを設定する場合は、次のルールを考慮してください。<br>- 使用されるソースが Amazon S3 の場合にのみ、Amazon S3 へのルックアップを設定できます。<br>- 使用されるソースが Amazon S3 または Amazon Redshift である場合は、Amazon Redshift へのルックアップを設定できます。<br>接続されていないルックアップ関数は、他の式関数とネストできます。<br>ルックアップに対する特定のルールの詳細については、「ルックアップトランスフォーメーションのルールとガイドライン」（ページ 102）を参照してください。 |
| ソータ | X | ソータートランスフォーメーションを選択する際に、適用できるのは次のようなソータの詳細プロパティのみです。<br>- 重複なし。重複する行を削除して、個別のクエリを作成できます。<br>- NULL を下として扱う。NULL 値を持つカラムに基づいて、クエリ内のカラムをソートすることができます。<br>カラムはクエリ内でのみ並べ替えることができます。ソートの順序は、Redshift ターゲットには適用されません。 |
| ルータ | X | ルータートランスフォーメーションを設定する場合は、1 つの出力グループのみをターゲットトランスフォーメーションに接続またはマッピングする必要があります。 |
| Union | X | - |

## 式トランスフォーメーション

完全なプッシュダウンの最適化を設定し、式トランスフォーメーションをプッシュして Amazon Redshift で処理することができます。

マッピング内の各ソースに式トランスフォーメーションを追加してから、マッピング内のダウンストリームに結合することができます。さらに、トランスフォーメーションから分岐してから、マッピングのダウンストリームのトランスフォーメーションに分岐する複数の式トランスフォーメーションを追加できます。

式トランスフォーメーションを設定するときは、次のルールを考慮して、変数を式に含めてください。

- 前の行の処理中に割り当てられた値を現在の行の計算に使用している場合、変数を使用することはできません。この場合、マッピングはプッシュダウンの最適化なしで実行されます。
- 変数はネストできますが、式で定義する前に変数を参照することはできません。
  変数がこの順序で定義されていない場合、マッピングはプッシュダウンの最適化なしで実行されます。

  例:

  ```
   var: AGEPLUS2 = AGEPLUS1 + 1
  var: AGEPLUS1 = AGE + 1
  out: NEXTAGE = AGEPLUS2 + 1
  ```

  ここで、AGE +1 は後で定義されます。最初の変数の AGEPLUS2 は AGEPLUS1 を参照しており、未解決のままです。
  これを解決するには、次の順序で変数を指定します。

  ```
   var: AGEPLUS1 = AGE + 1
  var: AGEPLUS2 = AGEPLUS1 + 1
  out: NEXTAGE = AGEPLUS2 + 1
  ```
- 変数は、循環型の式またはそれ自体を参照する式を持つことはできません。
  例:

  ```
   var: AGEPLUS1 = AGEPLUS2 + 1
  var: AGEPLUS2 = AGEPLUS1 + 1
  out: NEXTAGE= AGEPLUS2
  ```

  ここで、AGEPLUS1 は AGEPLUS2 を参照し、未解決のままです。

## Amazon Redshift V2 マッピングでサポートされる機能

基本的な IAM とロールの引き受け認証を使用して、ソーストランスフォーメーションで Amazon Redshift V2 接続を設定し、マッピングタスクでプッシュダウンの最適化を有効にすることができます。

### 接続プロパティ

プッシュダウンの最適化を設定する際に、Amazon Redshift V2 接続の Amazon Redshift V2 ソースに対して次のような詳細プロパティを使用できます。

- ユーザー名
- パスワード
- アクセスキーとシークレットキー
- JDBC URL
- 領域
- マスタ対称キー
- 顧客マスタキー ID

## ソースのプロパティ

プッシュダウンの最適化を設定する際に、ソーストランスフォーメーションの Amazon Redshift V2 ソースに対して次のようなプロパティを使用できます。

- ソース接続 - パラメータ、実行時にパラメータのオーバーライドを許可する
- ソースタイプ - 単一、クエリ、およびパラメータ
- フィルタ
- S3 バケット名
- 圧縮を有効にする
- Unload オプション
- 暗号化タイプ - SSE-S3、SSE-KMS、SSE-SMK

  **注:** ターゲットプロパティで同じマスタ対称キーを指定する必要があります。
- スキーマ名
- ソーステーブル名
- Pre-SQL
- Post-SQL
- SQL クエリ
- 個別選択

## ターゲットプロパティ

プッシュダウンの最適化を設定する際に、Amazon Redshift V2 ターゲットに対して次のようなプロパティを使用できます。

- ターゲット接続 - パラメータ、実行時にパラメータのオーバーライドを許可する
- ターゲットタイプ - 単一オブジェクト、パラメータ
- 実行時にパラメータのオーバーライドを許可する
- ターゲットオブジェクト - 既存のターゲット、実行時に新規作成
- 操作 - 挿入、更新、更新/挿入、削除、またはデータドリブン
- ターゲットテーブルの分析
- ロールの想定
- Copy コマンド

  - リージョン

  - Truncatecolumn

  - AWS_IAM_Role、Parquet ファイルと ORC ファイル用のみ
- Pre-SQL
- Post-SQL
- Char および Varchar に Null 値が必要
- スキーマ名
- ターゲットテーブル名
- データアップロード前のターゲットテーブルの切り詰め

- テーブルのクリーンアップ
- ターゲットクエリのオーバーライド
- ソース行を（挿入、更新、更新/挿入、削除、およびなし）として扱います。データ依存操作を使用する場合は、［なし］を選択します。
- INSERT 用のアップロードバッチごとの最大エラー数

**注:** 使用できないソースおよびターゲットの詳細プロパティを設定すると、マッピングは Informatica ランタイム環境で実行されます。

### ルックアップのプロパティ

プッシュダウンの最適化を設定する際に、Amazon Redshift V2 ルックアップに対して次のようなプロパティを使用できます。

- ルックアップ接続 - パラメータ、実行時にパラメータのオーバーライドを許可する
- ソースタイプ - 単一オブジェクト、クエリ、パラメータ
- 複数一致 - エラーを報告。接続されていないルックアップに適用できます。
- 複数の一致 - すべての行を返す。接続されたルックアップに適用できます。
- S3 バケット名
- 圧縮を有効にする
- Unload オプション
- 暗号化タイプ
- スキーマ名
- ソーステーブル名
- Pre-SQL
- Post-SQL
- SQL クエリ
- 個別選択

**注:** マッピングタスクでパラメータファイルを指定して、マッピング内の Amazon Redshift V2 のソース、ルックアップ、およびターゲットの接続とオブジェクトをオーバーライドできます。

## Amazon S3 V2 ソースでサポートされる機能

マッピングタスクでプッシュダウンの最適化を有効にした場合は、基本的なロールの引き受け、および IAM 認証を使用して Amazon S3 V2 接続を設定する必要があります。

### ソースのプロパティ

プッシュダウンの最適化を設定する際に、マッピングで Amazon S3 V2 ソースに対して次のようなプロパティを使用できます。

- ソース接続
- ソースタイプ - 単一オブジェクト、パラメータ
- パラメータ
- 暗号化タイプ - クライアントサイド暗号化（フラットファイルに適用可能）、サーバー側の暗号化、KMS を使用したサーバー側の暗号化

- データ圧縮 - Gzip（フラットファイルおよび Parquet ファイルに適用可能）、Deflate（Avro ファイルに適用可能）、Snappy（Avro ファイル、Parquet ファイル、および ORC ファイルに適用可能）、Zlib（ORC ファイルに適用可能）
- ファイル形式
  - 区切り文字
  - 修飾子
  - コードページ
  - ヘッダー行番号
  - 最初のデータ行
  - ソースタイプ
  - フォルダパス
- ファイルソースタイプ
- ファイル名
- 入力形式
  - Avro
  - Parquet
  - ORC
  - JSON
  - 区切りファイル

# 複数のターゲットに対する完全なプッシュダウンの最適化

マッピングに複数の Amazon Redshift V2 ターゲットを追加できます。

複数の Amazon Redshift V2 ターゲットに書き込むようにマッピングを設定した場合、完全なプッシュダウンの最適化を設定すると、書き込み操作をさらに最適化することができます。

最適化を行うためには、複数のターゲットに対して個別に挿入、更新、更新/挿入、削除、またはデータ依存操作を設定するという選択ができます。

複数のターゲットトランスフォーメーションで同じまたは異なる Amazon Redshift V2 ターゲットテーブルを選択し、ターゲットトランスフォーメーションごとに異なる操作を実行して、互いに依存しない状態で実行することができます。

## ルールおよびガイドライン

複数のターゲットに対して完全なプッシュダウンの最適化を行う場合は、次のルールとガイドラインを考慮してください。

- マッピングを実行し、すべてのターゲットで同じターゲットテーブルを使用する場合、行数は、プッシュダウンの最適化を使用せずに実行されるマッピングの行数とは異なります。Amazon S3 ソースを使用して、複数の Redshift V2 ターゲットで更新/挿入操作を実行する場合に適用されます。
- マッピングを実行し、すべてのターゲットで同じターゲットテーブルを使用すると、Secure Agent は、プッシュダウンの最適化を使用せずに実行されるマッピングのデータとは異なるデータセットをターゲットに書き込みます。Amazon S3 ソースを使用して、複数の Redshift V2 ターゲットで挿入操作を実行し、**［データロード前にターゲットテーブルをトランケート］**詳細ターゲットプロパティを有効にした場合に適用されます。

## プッシュダウンの最適化のプレビュー

プッシュダウンの最適化用に設定されたマッピングタスクを実行する前に、マッピングの作成時にプッシュダウン最適化が可能かどうかをプレビューできます。プッシュダウンの最適化は、Mapping Designer の［**プッシュダウンの最適化**］パネルでプレビューできます

プッシュダウンの最適化に関する必要なオプションを選択し、プレビューを実行すると、データ統合により、一時的なプッシュダウンのプレビューマッピングタスクが作成および実行されます。ジョブが完了すると、データ統合によって、実行する SQL クエリと警告が［**プッシュダウンの最適化**］パネルに表示されます。警告メッセージは、設定されたマッピングのどのトランスフォーメーションがプッシュダウンの最適化に該当しないかを判断するのに役立ちます。プッシュダウンの最適化が失敗すると、失敗の時点までに生成されたクエリが一覧表示されます。プッシュダウンの最適化のマッピングを実行する前に、マッピングを編集したり、必要なトランスフォーメーションを修正することが可能です。

［**マイジョブ**］で作成された一時ジョブを表示し、セッションログをダウンロードして、生成されたクエリを確認することもできます。

プッシュダウンの最適化をプレビューする方法については、データ統合のヘルプにある、「*マッピング*」のトピック「プッシュダウンの最適化のプレビュー」を参照してください。

## Amazon Redshift V2 マッピングタスクに対する最適化の設定

次の手順を実行して、Amazon Redshift V2 マッピングタスクに対するプッシュダウンの最適化を設定します。

1. 要件に応じてマッピングを作成します。
   - Amazon S3 から読み取りを行って Amazon Redshift に書き込むには、ソーストランスフォーメーションで Amazon S3 V2 接続を使用し、ターゲットトランスフォーメーションで Amazon Redshift V2 接続を使用します。
   - Amazon Redshift との間で読み取りまたは書き込みを行うには、ソーストランスフォーメーションとターゲットトランスフォーメーションで Amazon Redshift V2 接続を使用します。
2. マッピングタスクを作成します。
   a. 設定されたマッピングを選択します。
   b. ［**スケジュール**］タブの［**プッシュダウンの最適化**］セクションで、プッシュダウンの最適化の値を［**完全**］または［**ソースへ**］に設定します。

      **注:** 最適化コンテキストタイプオプションは、Amazon Redshift V2 マッピングタスクには適用されません。
   c. タスクを保存し、［**終了**］をクリックします。

マッピングタスクの実行時に、トランスフォーメーションロジックが Amazon Redshift データベースにプッシュされます。プッシュダウンの最適化の実行を確認するには、ジョブのセッションログをチェックします。［**マイジョブ**］ページで、開始したジョブを監視することができます。

## プッシュダウンの最適化ジョブのクリーンな停止

プッシュダウンの最適化が有効になっているタスクが実行されている場合は、ジョブにクリーンな停止を使用して、ジョブによって生成されたすべての発行済みの文とプロセスを終了できます。

データ統合の［マイジョブ］ページと［監視］の［すべてのジョブと実行中のジョブ］ページの［**クリーンな停止**］オプションを使用します。

プッシュダウンの最適化タスクにクリーンな停止を使用する場合は、事前に以下の例外を参照してください。

- プッシュダウンの最適化が有効になっているタスクに対してクリーンな停止を実行する場合に、マッピングのターゲットまたはソースのプロパティに SQL ステートメントが含まれていると、進行中のすべての SQL ステートメントを元に戻す操作が試行されます。コミットされた SQL ステートメントはいずれも元に戻すことはできません。
- 実行時に新しいターゲットを作成し、ジョブにすぐにクリーンな停止使用するように設定されたマッピングを実行すると、ジョブが終了した場合でも、データ統合によってターゲットテーブルが作成されます。

## プッシュダウンの最適化に関するルールおよびガイドライン

Amazon Redshift データベースのプッシュダウンの最適化を設定する場合は、次のルールとガイドラインを使用してください。

- Amazon S3 から Amazon Redshift にデータをコピーするには、大きなファイルを分割して複数のデータファイルを使用する必要があります。詳細については、『Amazon documentation』を参照してください。
- JSON ファイルのカラム名に大文字を使用し、プッシュダウン最適化を設定すると、マッピングは次のエラーで失敗します。

  ```
  Error Reason: Invalid JSONPath format: Member is not an object.
  ```
- プッシュダウン最適化を使用してマッピングを設定し、日付/時刻値を使用して ORC ソースを定義すると、Secure Agent がいくつかの値を Redshift ターゲットに正しく書き込まない場合があります。
- 19,0 を超える精度の 10 進値で Parquet ソースを定義し、**COPY** コマンドを使用して Redshift ターゲットに書き込むと、プッシュダウン最適化でマッピングが失敗します。
- 区切りファイル、Avro ファイル、または JSON ファイルのデータの値が精度の値を超える場合は、属性 **TRUNCATECOLUMNS=ON** を指定します。
- ORC ファイルタイプと Parquet ファイルタイプの場合、COPY コマンドの **AWS_IAM_ROLE** を指定して、完全なプッシュダウンの最適化を有効にします。
- ソースが Parquet または ORC ファイル形式で、マッピングタスクに完全なプッシュダウンの最適化を有効にしている場合は、ロールの引き受けを使用できません。
- タスクに複数のトランスフォーメーションのダウンストリームに接続されている単一のトランスフォーメーションを持つマッピングが含まれる場合（逆もまた同様）、マッピングタスクに対して完全なプッシュダウンの最適化を有効にすることはできません。
- Date および Decimal データ型の Avro ソースに対して完全なプッシュダウンの最適化を有効にすることはできません。
- Shift-JIS エンコードを使用して Amazon S3 フラットファイルからデータを読み取り、Amazon Redshift ターゲットに書き込んで完全なプッシュダウン最適化を有効にすると、マッピングが失敗します。
- 大文字のカラム名で Amazon S3 Avro ソースを読み取り、Amazon Redshift V2 接続を使用して Amazon Redshift ターゲットにデータを書き込むマッピングを設定すると、COPY コマンドによりターゲットに空白行が書き込まれます。
- 完全なプッシュダウンの最適化が有効なマッピングの Time カラムを含むデータの更新操作を設定し、詳細ターゲットプロパティの［ターゲットクエリのオーバーライド］フィールドで指定された値を使用してターゲットクエリをオーバーライドした場合、タスクは実行されますがデータは処理されません。
- Amazon S3 バケットの領域と Amazon Redshift の領域が異なる場合、**REGION** 属性を **COPY** コマンドで指定して、完全なプッシュダウンの最適化を有効にします。区切りファイル、Avro ファイル、および JSON ファイルに適用されます。
- 更新、更新/挿入、または削除操作を実行する場合は、ターゲット表のプライマリキーまたは更新カラムを必ず指定してください。

- 更新/挿入操作を実行して JVM オプション`-DuseDeleteForUpsert=true` を設定すると、処理された行のターゲット統計に、JVM オプションを設定しない場合と比較した追加の行が表示されます。ターゲットテーブルの行数は、どちらの場合も同じです。
- Amazon Redshift ターゲットで更新、更新/挿入、または削除操作を実行し、更新カラムを別のタイムゾーンの日付/時刻データ型として指定した場合、データはターゲットに書き込まれません。ただし、タスクは正常に実行されます。この問題は、Amazon S3 Parquet ソースを使用する場合に発生します。
- 更新操作を実行した場合、フィールドマッピングでターゲットテーブルの id カラムをマッピングすることはできません。
- ターゲットオブジェクトで DD_INSERT、DD_UPDATE、または DD_DELETE データ依存操作を使用して条件を設定した場合でも、ログには残りのデータ依存操作のクエリが含まれます。マッピングは正常に実行されます。
- データ依存条件に DD_REJECT 操作のみが含まれている場合、マッピングはクエリを生成せずに実行されます。
- 完全なプッシュダウンの最適化が有効になっているマッピングでトランスフォーメーションをパラメータ化し、入力パラメータをオーバーライドするようにパラメータファイルを設定した場合、Secure Agent はオーバーライドされたパラメータを無視します。
- 共有体トランスフォーメーションにいくつかのソースカラムからの入力があり、残りのカラムをターゲットにマッピングしなかった場合、マッピングされていないカラムにはターゲットに NULL データが表示されます。
- 入力フィールドに decimal フィールドが含まれており、共有体トランスフォーメーションで decimal フィールドをターゲットにマッピングしなかった場合は、完全なプッシュダウンの最適化で実行されている、Amazon Redshift V2 接続を使用するマッピングはエラーで失敗します。
- プッシュダウンの最適化が有効になっているマッピングに共有体トランスフォーメーションおよびアグリゲータトランスフォーメーションが含まれている場合は、集計関数またはグループ化フィールドからの入力フィールドをフィールドマッピングに含めるか、集計関数またはグループ化フィールドからフィールドを完全に削除します。このような操作を行わない場合、マッピングが失敗します。
- Date データ型または Datetime データ型のカラムが `YYYY-DD-MM` 形式ではない場合、マッピングは失敗します。
- プッシュダウンの最適化を正常に実行するには、SQL クエリのすべてのフィールドをターゲットにマッピングする必要があります。
- 重複するカラムがカスタムクエリに含まれている場合、マッピングはプッシュダウンの最適化を使用せずに実行されます。
- Amazon Redshift V2 接続を使用して Amazon Redshift ターゲットの float データ型または integer データ型でデータを更新するプッシュダウンの最適化が有効になっているマッピングを実行すると、マッピングが失敗する場合があります。
- マッピングタスクで式トランスフォーメーションをパラメータ化し、パラメータをオーバーライドするようにパラメータファイルを設定した場合、Secure Agent はオーバーライドされたパラメータを読み取りません。この問題は、Amazon Redshift V2 接続を使用するマッピングに完全なプッシュダウン最適化を設定した場合に発生します。
- 次の形式で詳細ネイティブフィルタを指定すると、実行時にマッピングが失敗します: `schema_name.table_name.column_name`。詳細フィルタを定義する場合は、条件にスキーマ名で修飾されたテーブル名を含めることはできません。

## Amazon Redshift との間で読み取りおよび書き込みを行うマッピングでの完全なプッシュダウンの最適化に対するルール

Amazon Redshift との間で読み取りまたは書き込みを行うマッピングに完全なプッシュダウンを設定する場合は、次のガイドラインを考慮してください。

- Amazon Redshift のソースとターゲットが同じクラスタ内にある場合、Secure Agent は、UNLOAD コマンドと COPY コマンドを使用して Amazon S3 でデータをステージングしてから Amazon Redshift に書き込むという処理は行いません。Secure Agent は、挿入、更新、更新/挿入、削除、またはデータ駆動型の操作を直接実行して、データを Amazon Redshift に書き込むため、タスクのパフォーマンスが向上します。
- 完全なプッシュダウンの最適化が有効なマッピングの Time カラムを含むデータの更新操作を設定し、詳細ターゲットプロパティの［**ターゲットクエリのオーバーライド**］フィールドで指定した値を使用してターゲットクエリをオーバーライドした場合、タスクは実行されますがデータは処理されません。
- IAM ロールを引き受け、ソースまたは完全なプッシュダウンの最適化を有効にした状態でマッピングを実行して Redshift からデータを読み取る場合、ロールの引き受けは適用されません。
  ロールの引き受けを適用し、マッピングを正常に実行するには、AWS_IAM_ROLE プロパティを指定して、その値をソーストランスフォーメーションの［**Unload オプション**］フィールドに設定する必要があります。

  例: `AWS_IAM_ROLE=arn:aws:iam::0093:role/redshift_role`
- 異なる領域にある Amazon S3 バケットをマッピング内のソースとターゲットのオブジェクトが指している場合は、**COPY** コマンドで **REGION** 属性を指定して、ソースの Amazon S3 バケットと同じ領域のターゲットに Amazon S3 バケットを設定します。
- 読み取るデータに区切り文字（|）、引用符（"）、およびエスケープ（\）文字が含まれており、ソーストランスフォーメーションの［**Unload オプション**］フィールドに `ADDQUOTES;DELIMITER = \174;`と設定すると、マッピングは失敗します。
  このエラーを回避するには、ターゲットトランスフォーメーションの［**コピーオプション**］フィールドで次のプロパティを設定します: `ESCAPE;REMOVEQUOTES;CSV=OFF;QUOTE=OFF;DELIMITER = \174;`
- NULL 値を含んだ Bigint、Datetime、および Boolean データ型の単一のカラムテーブルからデータを読み取った場合、NULL 値はターゲットに書き込まれません。
  このエラーを回避するには、ターゲットトランスフォーメーションの［**コピーオプション**］フィールドに `IGNOREBLANKLINES` と設定する必要があります。
- データを読み取るようにマッピングでソーストランスフォーメーションを設定し、`ESCAPE=OFF` または ON と設定して、［**Unload オプション**］フィールドに `ADDQUOTES` と設定してからデータにエスケープ文字と引用符を追加すると、NULL 値がターゲットに書き込まれます。
  これを回避するには、ターゲットトランスフォーメーションの［**コピーオプション**］フィールドに `QUOTE=\042` を追加してから、マッピングを実行する必要があります。
- データの読み取り時にデータにエスケープ（\）文字が含まれている場合は、ソーストランスフォーメーションの［**Unload オプション**］フィールドに ESCAPE=OFF と指定する必要があります。プロパティを設定していない場合、エスケープ文字がターゲットに複製されます。例えば、データ出力 12\12\2012 は 12\\12\\2012 と表示されます。
- float4 または double 値を含んだデータの読み取りと書き込みを行うと、ターゲットの float 値はわずかに変化します。
- トランスフォーメーションでユーザー定義関数を指定すると、関数名がセッションログに正しく表示されません。
- マッピングにシーケンスジェネレータトランスフォーメーションが含まれている場合は、必ず NEXTVAL 出力フィールドをターゲットにマッピングしてください。CURRVAL 出力フィールドをターゲットにマッピングしないでください。CURRVAL 出力フィールドをターゲットにマッピングした場合、マッピングタスクでは、トランスフォーメーションがサポートされている、プッシュダウンの最適化を使用せずに実行されるポイントまでマッピングロジックが部分的にプッシュダウンされることはありません。

## Amazon Redshift から読み取りを行うマッピングでのソースプッシュダウンの最適化に対するルール

Amazon Redshift から読み取りを行うマッピングにソースプッシュダウンを設定する場合は、次のガイドラインを考慮してください。

- 詳細ソースプロパティで SQL クエリまたはカスタムクエリを設定すると、マッピングを Amazon Redshift にプッシュできます。
- カスタムクエリをソースまたは SQL オーバーライドとして使用すると、テーブル名のエイリアスは適切に生成されません。名前は、プッシュダウン最適化クエリで"INF"から始まります。
- ソースカラム名がエイリアス化されている完全なプッシュダウンの最適化で SQL クエリをオーバーライドしてマッピングを実行すると、マッピングが失敗します。エイリアス名とソースカラム名が同じであることを確認してください。
- 複数の出力グループを持つルータートランスフォーメーションを Amazon Redshift ソースにプッシュすることはできません。
- クエリソースタイプを使用して Amazon Redshift から読み取った場合は、フィールドメタデータを保持してマッピングを保存するという選択ができます。クエリを編集してマッピングを実行した場合でも、設計時に指定したフィールドメタデータは保持されます。

## 関数のルールおよびガイドライン

Amazon Redshift データベースに関数をプッシュする際には、以下のルールおよびガイドラインを使用します。

- TO_DATE()と TO_CHAR()を Amazon Redshift にプッシュするには、文字列および形式引数を定義する必要があります。
- ADD_TO_DATE()関数の一部として NS 形式を使用する場合、エージェントはこの関数を Amazon Redshift にプッシュしません。
- 次の形式のいずれかを TO_CHAR()および TO_DATE()関数の一部として使用する場合、エージェントはこれらの関数を Amazon Redshift にプッシュしません。

  - - NS

  - - SSSS

  - - SSSSS

  - - RR
- TRUNC(DATE)、GET_DATE_PART()、および DATE_DIFF()を Amazon Redshift にプッシュするには、次の形式を使用する必要があります。

  - - D

  - - DDD

  - - HH24

  - - MI

  - - MM

  - - MS

  - - SS

  - - US

  - - YYYY
- 単一の文字列値を新しい文字列値に置き換える場合にのみ **REPLACESTR()**を使用できます。

構文

```
REPLACESTR (CaseFlag, InputString, OldString, NewString).
```

- **SUBSTR()**を Amazon Redshift に押し出すには、引数に整数の値を定義する必要があります。
- MD5()を Amazon Redshift にプッシュすると、Secure Agent はすべての MD5 フィールドを小文字で返します。ただし、プッシュダウンの最適化を使用せずにマッピングを実行すると、Secure Agent は大文字の MD5 フィールドを返します。

### 集計関数のルールとガイドライン

Amazon Redshift データベースに集計関数をプッシュする際には、以下のルールおよびガイドラインを使用します。

- 集計式関数で条件句を使用することはできません。
- 集計式では非集計関数を使用できます。
- ネストされた集計関数を直接使用することはできません。例えば、集計式関数カラムに SUM(AVG(col1))を指定することはできません。マッピングに複数の集計トランスフォーメーションを追加して、ネストされた集計を使用します。
- マッピングタスクで GROUP BY フィールドと集計関数をパラメータ化できます。
- 単一の値で構成される式に STDDEV 関数または VARIANCE 関数を使用してマッピングを実行すると、関数の結果は NULL になります。マッピングをプッシュダウンせずに実行すると、関数の結果は 0 になります。
- フィールドマッピング中に、GROUP BY ポートに追加したフィールドのみをマッピングする必要があります。そうしない場合、マッピングは非 PDO モードで実行され、エラーメッセージが表示されます。
- 集計関数の GROUP BY 句で Amazon S3 フラットファイルソースからのポートを指定せず、ポートを Redshift ターゲットにマッピングすると、マッピングタスクは非 PDO モードで正常に実行され、次のメッセージが表示されます。
  ソースに対するプッシュダウンの最適化はトランスフォーメーション[Aggregator]の前で停止します（[f_varchar]がパススルーポートによりグループ化されておらず、許可されないため）。

  Amazon S3 Avro または Parquet ソースを使用してマッピングをプッシュダウンすると、マッピングは失敗します。

## ルータートランスフォーメーションのルールとガイドライン

ルータートランスフォーメーションについては、次のルールとガイドラインを考慮してください。

- ソースフィールドについては、id カラムのメタデータを string から int に変更しないでください。
- ソースにバイナリデータ型のフィールドがある場合はマッピングが失敗し、次のエラーが発生します。
  [Full Pushdown Optimization Failed Due To PARQUET : Not Supported data type binary.]
- ルーターグループに対して条件を作成し、ルーターグループをターゲットテーブルに接続してその条件の 1 つが失敗した場合、マッピングで残りの条件が処理されることはありません。このマッピングはプッシュダウンの最適化を使用せずに実行されます。

## ルックアップトランスフォーメーションのルールとガイドライン

ルックアップトランスフォーメーションを設定する場合は、次のガイドラインに従ってください。

- 動的ルックアップキャッシュと永続キャッシュを設定することはできません。
- 接続されたルックアップと接続されていないルックアップの詳細ルックアッププロパティを設定することはできません。

- ルックアップトランスフォーメーションに詳細ルックアッププロパティを選択した場合でも、Secure Agent は詳細プロパティを無視します。エラーメッセージはログに表示されません。
- Amazon Redshift ルックアップの Unload コマンドに ADDQUOTE オプションを追加した場合は、Amazon Redshift ターゲットの Copy コマンドに QUOTE="という QUOTE オプションも追加する必要があります。
- 異なる領域にある Amazon S3 バケットをマッピング内のソースとターゲットが指している場合は、**COPY** コマンドで **REGION** 属性を指定して、ソースの Amazon S3 バケットと同じ領域のターゲットに Amazon S3 バケットを設定します。
- Amazon Redshift ソースまたはルックアップオブジェクトに CSE-SMK 暗号化タイプを設定した場合は、ターゲットプロパティでマスタ対称キーを指定していることを確認してください。
- =演算子は、完全にパラメータ化されたルックアップ条件でのみ指定できます。複合的な条件の中で<、<=、>>、>=、および! =などの演算子を指定した場合、マッピングは失敗します。
- Amazon S3 ソースのルックアップを設定した場合は、Amazon S3 ソースとルックアップオブジェクトから FileName フィールドを削除します。FileName フィールドは使用できません。

## 接続されたルックアップ

接続されたルックアップトランスフォーメーションについては、次のルールとガイドラインを考慮してください。

- プッシュダウンの最適化が機能するようにするには、接続されたルックアッププロパティで［**複数一致**］プロパティ値に［**すべての行を返す**］を選択する必要があります。
- Amazon S3 の場所に同じプレフィックスを持つソースファイル（abc.txt.1 と abc.txt.2 など）が含まれている場合、COPY コマンドは両方のファイルを Amazon Redshift ターゲットにロードしようとして、マッピングが失敗する場合があります。

## 接続されていないルックアップ

接続されていないルックアップトランスフォーメーションについては、次のルールとガイドラインを考慮してください。

- プッシュダウンの最適化が機能するようにするには、接続されていないルックアップのプロパティで［**複数一致**］プロパティ値に［**エラーを報告**］を選択する必要があります。ただし、ルックアップテーブルで複数の一致が検出された場合、Secure Agent はエラーを報告しません。したがって、ルックアップテーブルでは複数の一致が検出されないようにする必要があります。
- ［**任意の行を返す**］プロパティを選択すると、マッピングタスクはプッシュダウンの最適化を使用せずに実行されます。
- ルックアップ式で接続されていないルックアップに演算子を使用することはできません。演算子を含めるには、追加の式トランスフォーメーションを使用します。
- ソースルックアップテーブルと接続されていないルックアップテーブルで同じカラム名を使用して、完全なプッシュダウンの最適化を有効にすると、マッピングはカラムの重複エラーで失敗します。マッピングを正常に実行するには、プレフィックスまたはポストフィックスを使用して受信フィールドの名前を変更し、ルックアップ式でそのフィールドを使用します。

# Amazon Redshift ODBC 接続を使用したプッシュダウンの最適化

Amazon Redshift ソースからデータを読み取り、Amazon Redshift ターゲットに書き込むとき、および Amazon Redshift 内全体でマッピングロジックを実行する場合、完全なプッシュダウンの最適化またはソースのプッシュダウンの最適化を有効にするには、ODBC 接続を使用します。

プッシュダウンの最適化用に設定されたタスクを実行すると、トランスフォーメーションロジックが SQL 文に変換されます。タスクで SQL 文がデータベースに送信され、データベースで SQL 文が実行されます。

マッピングのパフォーマンスを向上するために Amazon ODBC Redshift ドライバを使用する ODBC 接続タイプに、プッシュダウンの最適化を設定できます。ODBC データソースアドミニストレータでデータソース名を作成する必要があります。

Amazon Redshift ODBC 接続を作成した後で、詳細セッションプロパティで**プッシュダウンの最適化**プロパティの値に［**完全**］または［**ソースへ**］を選択します。セッションログで、プッシュダウンの最適化が実行されていることを確認できます。

**注:** Amazon Redshift は、完全なプッシュダウンの最適化での更新/挿入操作をサポートしていません。

**例**

急成長を遂げているデータサイエンス組織で働いているとします。この組織は、財務分析を行うソフトウェア製品を開発して、個人のプロフィール、企業、職務、広告主、および出版社をつなぐ財務グラフを作成しています。組織では、Amazon Web サービスに基づくインフラストラクチャを使用して、ペタバイト規模のデータウェアハウスである Amazon Redshift にデータを格納しています。組織では、可視化を作成してリアルタイム分析を実現するためにビジネスインテリジェンスサービスを実装する予定です。そのため、Amazon Redshift に格納されている膨大な量のデータをビジネスインテリジェンスサービスに移植する必要があります。
Amazon Redshift V2 コネクタを使用して、Amazon Redshift からデータを読み取ることができます。この大量のデータを読み取るために、ODBC 接続タイプとして、ソースのプッシュダウンを使用できます。プッシュダウンの最適化による ODBC 接続タイプを使用すると、パフォーマンスが向上します。

## Amazon Redshift ODBC 接続の設定

Amazon Redshift は、Windows システムおよび Linux システムで Amazon ODBC Redshift ドライバをサポートしています。システム要件に基づいて Amazon ODBC Redshift 64 ビットドライバをインストールする必要があります。

**注:** Informatica では、プッシュダウンの最適化に使用するドライバとして、Amazon Redshift ODBC ドライババージョン `AmazonRedshiftODBC-64-bit-1.4.8.1000-1.x86_64` を認定しています。

### Windows 上の Amazon Redshift ODBC 接続の設定

Windows 上の Amazon Redshift に接続するように ODBC 接続を確立するには、ODBC 接続を設定する必要があります。

Windows 上で ODBC 接続を設定するには、次の手順を実行します。

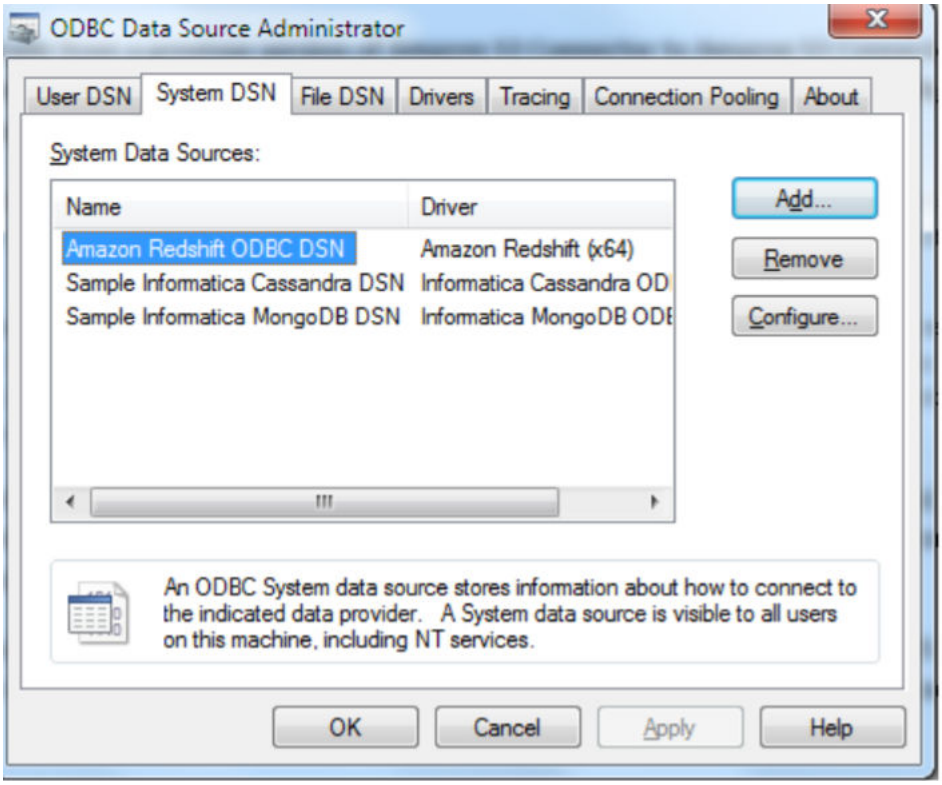
1. AWS サイトから Amazon Redshift ODBC ドライバをダウンロードします。

   Amazon Redshift ODBC 64 ビットドライバをダウンロードする必要があります。
2. Secure Agent がインストールされているマシンに Amazon Redshift ODBC ドライバをインストールします。
3. ODBC データソースファイルのインストール先のフォルダを開きます。
4. `odbcad32.exe` ファイルを実行します。

［**ODBC データソースアドミニストレータ**］ダイアログボックスが表示されます。

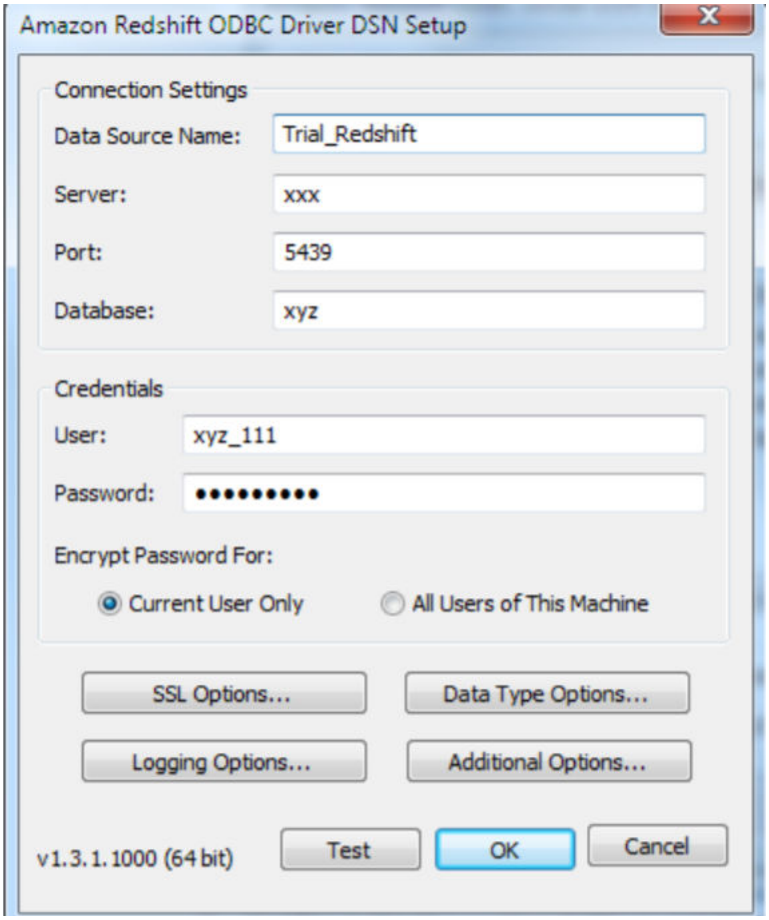
5. ［**システム DSN**］をクリックします。

   ［**システム DSN**］タブが表示されます。次の図は、［**ODBC データソースアドミニストレータ**］ダイアログボックスの［**システム DSN**］タブを示しています。




6. ［**設定**］をクリックします。

**[Amazon Redshift ODBC Driver DSN セットアップ]**ダイアログボックスが表示されます。次の図は、**[Amazon Redshift ODBC Driver DSN セットアップ]**ダイアログボックスを示していて、ここで**[接続設定]**および**[資格情報]**を設定できます。




7. **[接続設定]**セクションで次の接続プロパティを指定します。

| プロパティ | 説明 |
|---|---|
| データソース名 | データソースの名前。 |
| サーバー | Amazon Redshift サーバーの場所。 |
| ポート | Amazon Redshift サーバーのポート番号。 |
| データベース | Amazon Redshift データベースの名前。 |

**注:** JDBC URL から得た**[サーバー]**、**[ポート]**、および**[データベース]**の値を指定する必要があります。

8. **[資格情報]**セクションで次の資格情報を指定します。

| プロパティ | 説明 |
|---|---|
| ユーザー | Amazon Redshift データベースにアクセスするためのユーザー名。 |
| パスワード | Amazon Redshift データベースのパスワード。 |
| パスワードの暗号化 | 次のユーザーのパスワードを暗号化します。<br>- **カレントユーザのみ**<br>- **このマシンのすべてのユーザー**<br>デフォルトは[**カレントユーザのみ**]です。 |

9. **[Amazon Redshift ODBC Driver DSN セットアップ]**で、**[テスト]**をクリックして接続をテストします。
10. **[OK]** をクリックします。

Amazon Redshift ODBC 接続が Windows 上で正常に設定されました。

Amazon Redshift ODBC 接続を設定した後で、Amazon Redshift に接続するための ODBC 接続を作成する必要があります。

Amazon Redshift に接続するための ODBC 接続を作成する方法については、「ODBC 接続の作成」 (ページ 108)を参照してください。

## Linux 上の Amazon Redshift ODBC 接続の設定

Linux 上の Amazon Redshift に接続するように ODBC 接続を確立するには、ODBC 接続を設定する必要があります。

Linux 上で ODBC 接続を設定するには、次の手順を実行します。

1. AWS サイトから Amazon Redshift ODBC ドライバをダウンロードします。

   Amazon Redshift ODBC 64 ビットドライバをダウンロードする必要があります。
2. Secure Agent がインストールされているマシンに Amazon Redshift ODBC ドライバをインストールします。
3. `odbc.ini` ファイルのプロパティを次の形式で設定します。

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file

Host=cluster_endpoint
Port=port_number
Database=database_name
```

4. `odbc.ini` ファイルに次のプロパティを指定します。

| プロパティ | 説明 |
|---|---|
| ODBC Data Sources | データソースの名前。 |
| Driver | Amazon Redshift ODBC ドライバファイルの場所。 |
| Host | Amazon Redshift ホストの場所。 |

| プロパティ | 説明 |
|---|---|
| Port | Amazon Redshift サーバーのポート番号。 |
| Database | Amazon Redshift データベースの名前。 |

**注:** JDBC URL から得た **Host**、**Port**、および **Database** にの値を指定する必要があります。

5. オペレーティングシステムに次の環境変数を設定します。

| 変数 | 説明 |
|---|---|
| LD_LIBRARY_PATH | Amazon Redshift ODBC ドライバがインストールされているディレクトリ。 |
| ODBCINI | odbc.ini ファイルが含まれるディレクトリ。 |
| ODBCHROME | ODBC のインストールディレクトリ。 |

6. 次のコマンドを実行して odbc.ini ファイルをエクスポートします。

```
Export ODBCINI=/<odbc.ini file path>/odbc.ini
```

7. Secure Agent を再起動します。

   Linux 上の Amazon Redshift ODBC 接続が正常に設定されました。

Amazon Redshift ODBC 接続を設定した後で、Amazon Redshift に接続するための ODBC 接続を作成する必要があります。
Amazon Redshift に接続するための ODBC 接続を作成する方法については、「ODBC 接続の作成」 (ページ 108)を参照してください。

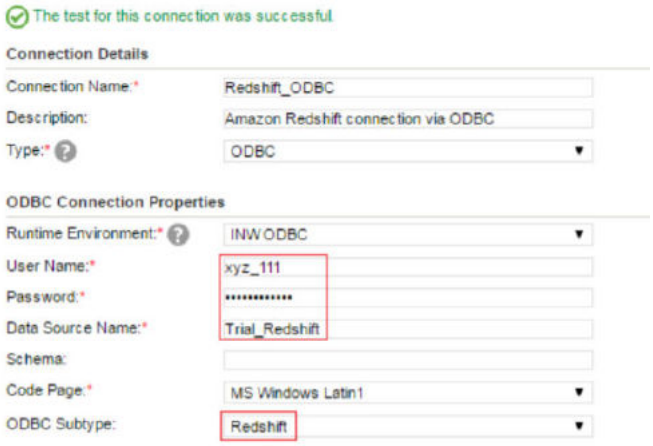
# ODBC 接続の作成

ODBC 接続を設定した後で、Amazon Redshift に接続するための ODBC 接続を作成する必要があります。

次の手順を実行して、**[接続]** ページで Amazon Redshift ODBC 接続を作成します。

1. 管理者で、**[接続]** をクリックします。
2. 右上隅の **[新しい接続]** をクリックします。

   **[新しい接続]** ページが表示されます。次の図は、**[新しい接続]** ページを示しています。



3. **［接続の詳細］**セクションで、次の接続の詳細を設定します。

| プロパティ | 説明 |
|---|---|
| 接続名 | ODBC 接続の名前。 |
| 説明 | 接続の説明。 |
| タイプ | 接続の種類。<br>接続のタイプに［ODBC］を選択します。 |

4. **［接続のプロパティ］**セクションで、次の接続の詳細を設定します。

| プロパティ | 説明 |
|---|---|
| ランタイム環境 | タスクを実行するランタイム環境の名前。 |
| ユーザー名 | Amazon Redshift アカウントのユーザー名。 |
| パスワード | Amazon Redshift アカウントのパスワード。 |
| データソース名 | Amazon Redshift データベース用に作成した ODBC データソースの名前を入力します。 |
| スキーマ | Amazon Redshift スキーマ名。 |
| コードページ | Secure Agent がデータの読み取りまたは書き込みに使用する必要があるコードページを選択します。 |
| ODBC サブタイプ | ［ODBC **サブタイプ**］フィールドの値を［Redshift］と入力します。 |

Amazon Redshift ODBC 接続が正常に作成されました。

## Amazon Redshift ODBC マッピングタスクに対する最適化の設定

次の手順を実行して、Amazon Redshift ODBC マッピングタスクに対するプッシュダウンの最適化を設定します。

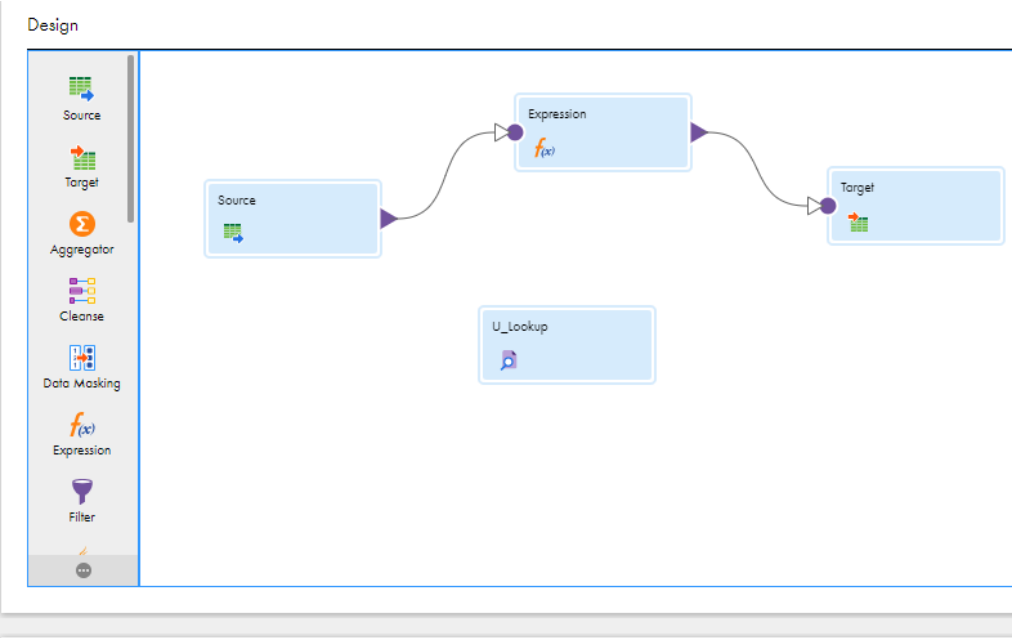
1. Amazon Redshift との間で読み取りまたは書き込みを行うマッピングを作成します。
   a. ソーストランスフォーメーションで Amazon Redshift ODBC 接続を使用します。
   b. ターゲットトランスフォーメーションで Amazon Redshift ODBC 接続を使用します。
2. マッピングタスクを作成します。
   a. 設定されたマッピングを選択します。
   b. **［スケジュール］**タブの**［プッシュダウンの最適化］**セクションで、プッシュダウンの最適化の値を**［完全］**または**［ソースへ］**に設定します。
   c. タスクを保存し、**［終了］**をクリックします。

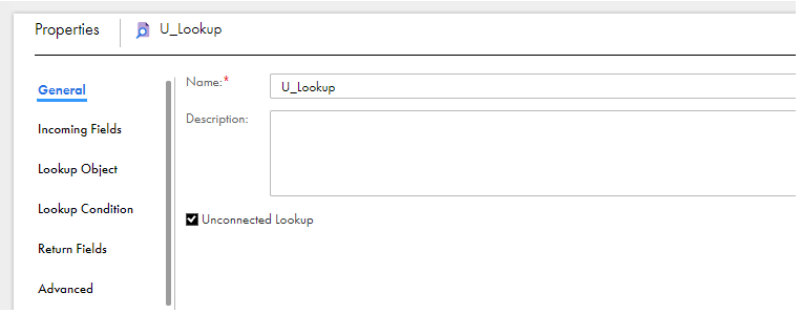
マッピングタスクの実行時に、トランスフォーメーションロジックが Amazon Redshift データベースにプッシュされます。プッシュダウンの最適化の実行を確認するには、ジョブのセッションログをチェックします。モニタにジョブのログを表示します。

## 接続されていないルックアップトランスフォーメーションの設定

接続されていないルックアップトランスフォーメーションを設定するには、**[接続されていないルックアップ]** オプションを選択して受信フィールドを追加し、ルックアップ条件を設定してから戻り値を指定します。次に、別のトランスフォーメーションでのルックアップ式を設定します。

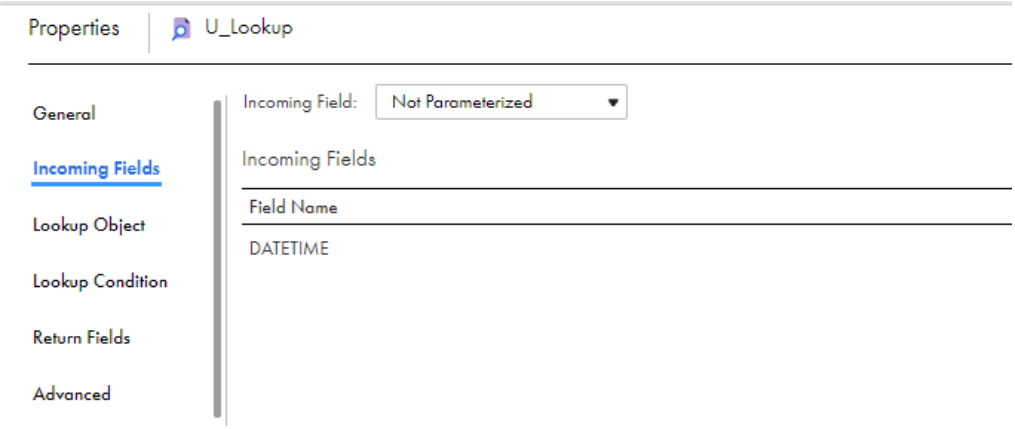
1. マッピングにルックアップトランスフォーメーションを追加します。
2. ルックアップトランスフォーメーションの **[全般]** タブで、**[接続されていないルックアップ]** オプションを有効にします。




3. ルックアップトランスフォーメーションの **[受信フィールド]** タブで、:LKP 式の各引数に対して受信フィールドを作成します。

   作成するルックアップ条件それぞれについて、ルックアップトランスフォーメーションに受信フィールドを追加します。




4. **［ルックアップオブジェクト］**タブでルックアップオブジェクトをインポートします。

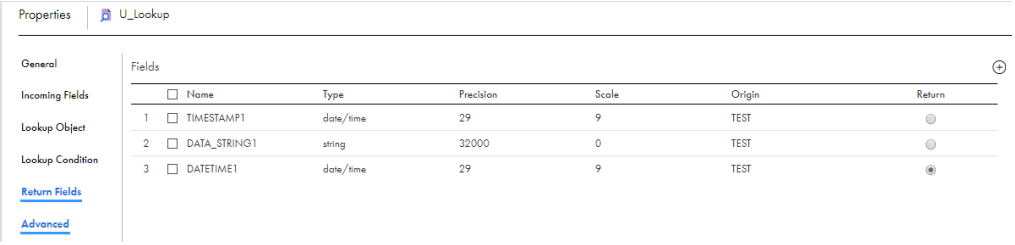
   プッシュダウンの最適化が機能するようにするには、接続されていないルックアップで**［複数一致］**プロパティ値の**［エラーを報告］**を選択する必要があります。

   **注:** ルックアップオブジェクトの**［エラーを報告］**を選択し、ルックアップテーブルで複数の一致が検出された場合、すべての一致がプッシュダウン最適化のターゲットに含まれます。

5. 戻り値を指定します。

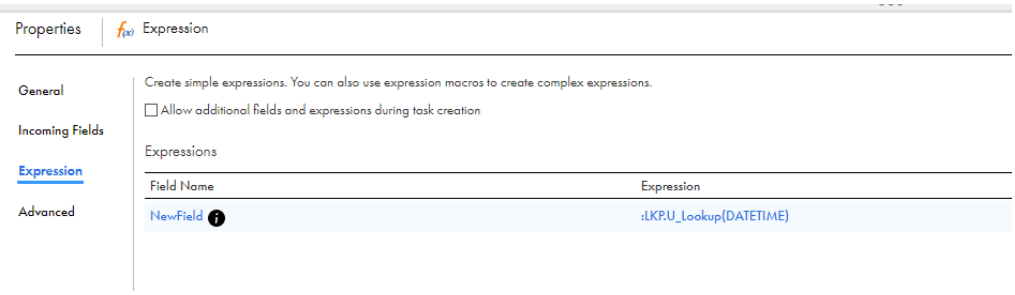
   ルックアップトランスフォーメーションに複数の入力値を渡すと、1 つのカラムのデータが返されます。Secure Agent により、ルックアップクエリから値が返されます。

6. Secure Agent によって返されるようにするフィールドを指定します。




7. 別のトランスフォーメーションのルックアップ式を設定します。

式トランスフォーメーションを使用するトランスフォーメーションの:LKP 式から、未接続のルックアップトランスフォーメーションの入力値を指定します。引数は、ルックアップ条件に使用されているルックアップトランスフォーメーションのローカル入力フィールドと一致するローカル入力ポートです。




8. フィールドをターゲットにマッピングします。

# クロススキーマプッシュダウンの最適化

Amazon Redshift ODBC 接続を使用して同じデータベース内の異なるスキーマの Amazon Redshift オブジェクトへのデータの読み取りまたは書き込みを行うマッピングタスクに対して、クロススキーマプッシュダウンの最適化を設定できます。

クロススキーマプッシュダウンの最適化を使用するには、Amazon Redshift ODBC 接続を作成して、ソース接続およびターゲット接続のスキーマを指定します。ソーススキーマおよびターゲットスキーマは異なる必要がありますが、どちらも同じデータベースに属している必要があります。マッピングタスクにプッシュダウンの最適化を設定して、詳細セッションプロパティでクロススキーマプッシュダウンの最適化を有効にします。デフォルトでは、**[クロススキーマプッシュダウンの最適化の有効化]**チェックボックスが選択されています。

## Amazon Redshift V2 マッピングタスクに対するクロススキーマの最適化の設定

次の手順を実行して、Amazon Redshift V2 マッピングタスクに対するクロススキーマプッシュダウンの最適化を設定します。

1. Amazon Redshift ODBC のソース接続およびターゲット接続を、それぞれ異なるスキーマで定義して作成します。

   以下に例を示します。

   - `rs_odbc1` Amazon Redshift ODBC 接続を作成して、接続プロパティで `CQA_SCHEMA1` スキーマを指定します。
   - `rs_odbc2` Amazon Redshift ODBC 接続を作成して、接続プロパティで `CQA_SCHEMA2` スキーマを指定します。

2. Amazon Redshift V2 マッピングを作成します。

   例えば、`m_rs_pdo_crossSchema` Amazon Redshift V2 マッピングを作成します。

3. ソーストランスフォーメーションを追加します。Amazon Redshift V2 ソースオブジェクト、および接続に指定されたスキーマを使用してデータを読み取るための接続を含めます。

   例えば、ソーストランスフォーメーションを追加します。Amazon Redshift V2 ソースオブジェクト、および `CQA_SCHEMA1` を使用してデータを読み取るための `rs_odbc1` 接続を含めます。

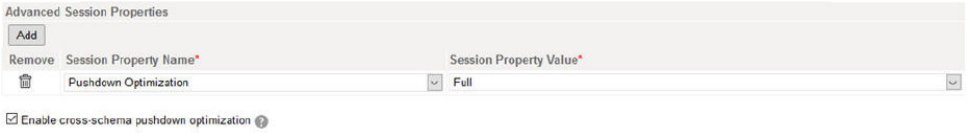
4. ターゲットトランスフォーメーションを追加します。Amazon Redshift V2 ターゲットオブジェクト、および接続に指定されたスキーマを使用してデータを書き込むための接続を含めます。

   例えば、ターゲットトランスフォーメーションを追加します。Amazon Redshift V2 ターゲットオブジェクト、および `CQA_SCHEMA2` を使用してデータを書き込むための `rs_odbc2` 接続を含めます。

5. Amazon Redshift V2 マッピングタスクを作成して、次のタスクを実行します。
   a. 設定された Amazon Redshift V2 マッピングを選択します。

      例えば、`m_rs_pdo_crossSchema` Amazon Redshift V2 マッピングを選択します。
   b. **［スケジュール］**タブの**［詳細オプション］**で、**［プッシュダウンの最適化］**を追加して、値を**［完全］**に設定します。
   c. **［クロススキーマプッシュダウンの最適化の有効化］**を選択します。

      次の図に、設定した**［クロススキーマプッシュダウンの最適化の有効化］**プロパティを示します。




   d. タスクを保存し、**［終了］**をクリックします。

   マッピングタスクを実行すると、Secure Agent は、`CQA_SCHEMA1` スキーマに関連付けられた Amazon Redshift ソースオブジェクトからデータを読み取り、`CQA_SCHEMA2` スキーマに関連付けられた Amazon Redshift V2 ターゲットオブジェクトにデータを書き込みます。

# Amazon Redshift ソースオブジェクトに対するカスタムクエリまたは SQL オーバーライドの設定

カスタムクエリまたは SQL オーバーライドを Amazon Redshift にプッシュダウンできます。

ソースオブジェクトとしてカスタムクエリを含むタスクを実行する前、または SQL オーバーライドを設定する前に、マッピングタスクプロパティの**［一時的なビューの作成］**セッションプロパティを設定する必要があります。

**［一時的なビューの作成］**プロパティを設定していない場合、マッピングはプッシュダウンの最適化を使用せずに実行されます。

プロパティを設定するには次のタスクを実行します。

1. マッピングタスクで、**［スケジュール］**タブの**［詳細セッションプロパティ］**に移動します。
2. **［セッションプロパティ名］**リストから、**［一時的なビューの作成］**を選択し、**［セッションプロパティ値］**を**［はい］**に設定します。
3. **［完了］**をクリックします。

## カスタムクエリ設定のルールおよびガイドライン

カスタムクエリを指定するときは、次のルールを考慮してください。

- SQL クエリにすべての一意のカラムフィールドを含める必要があります。
- ソーストランスフォーメーションでカスタムクエリまたは SQL オーバーライドを使用する場合は、Amazon Redshift のソーステーブルとターゲットテーブルにアクセスして操作を実行するための権限が必要です。
- SQL オーバーライドを設定するときは、接続されているすべてのフィールドを SQL クエリに含めるようにしてください。

# Amazon Redshift マッピングでサポートされる関数および演算子

次の表に、Amazon Redshift データベースでのプッシュダウン関数の可用性を示します。X マークの付いたカラムは、ソース側または完全なプッシュダウンの最適化を使用して、関数を Amazon Redshift データベースにプッシュできることを示しています。S マークの付いたカラムは、ソース側のプッシュダウン最適化を使用し

てのみ、関数を Amazon Redshift データベースにプッシュできることを示しています。ダッシュ（-）記号の付いたカラムは、関数をデータベースにプッシュできないことを示しています。

| 関数 | プッシュダウン | 関数 | プッシュダウン | 関数 | プッシュダウン |
|---|---|---|---|---|---|
| ABORT() | - | INSTR() | X | REG_REPLACE | - |
| ABS() | X | IS_DATE() | - | REPLACECHR() | - |
| ADD_TO_DATE() | X | IS_NUMBER() | - | REPLACESTR() | - |
| AES_DECRYPT() | - | IS_SPACES() | - | REVERSE() | - |
| AES_ENCRYPT() | - | ISNULL() | S | ROUND(DATE) | - |
| ASCII() | - | LAST() | - | ROUND(NUMBER) | X |
| AVG() | S | LAST_DAY() | X | RPAD() | X |
| CEIL() | X | LEAST() | - | RTRIM() | X |
| CHOOSE() | - | LENGTH() | X | SET_DATE_PART() | - |
| CHRCODE() | - | LN() | X | SIGN() | X |
| COMPRESS() | - | LOG() | - | SIN() | X |
| CONCAT() | X | LOOKUP | - | SINH() | - |
| COS() | X | LOWER() | X | SOUNDEX() | - |
| COSH() | - | LPAD() | X | SQRT() | X |
| COUNT() | S | LTRIM() | X | STDDEV() | S |
| CRC32() | - | MAKE_DATE_TIME() | - | SUBSTR() | X |
| CUME() | - | MAX() | S | SUM() | S |
| DATE_COMPARE() | X | MD5() | X | SYSTIMESTAMP() | S |
| DATE_DIFF() | X | MEDIAN() | - | TAN() | S |
| DECODE() | X | METAPHONE() | - | TANH() | - |
| DECODE_BASE64() | - | MIN() | S | TO_BIGINT | X |
| DECOMPRESS() | - | MOD() | S | TO_CHAR(DATE) | S |
| ENCODE_BASE64() | - | MOVINGAVG() | - | TO_CHAR(NUMBER) | X |
| EXP() | X | MOVINGSUM() | - | TO_DATE() | X |
| FIRST() | - | NPER() | - | TO_DECIMAL() | X |

| 関数 | プッシュダウン | 関数 | プッシュダウン | 関数 | プッシュダウン |
|---|---|---|---|---|---|
| FLOOR() | X | PERCENTILE() | - | TO_FLOAT() | X |
| FV() | - | PMT() | - | TO_INTEGER() | X |
| GET_DATE_PART() | X | POWER() | X | TRUNC(DATE) | S |
| GREATEST() | - | PV() | - | TRUNC(NUMBER) | S |
| IIF() | X | RAND() | - | UPPER() | X |
| IN() | S | RATE() | - | VARIANCE() | S |
| INDEXOF() | - | REG_EXTRACT() | - | | |
| INITCAP() | X | REG_MATCH() | - | | |

次の表に、Amazon Redshift データベースで使用できるプッシュダウン演算子を示します。X マークの付いたカラムは、ソース側、ターゲット側、または完全なプッシュダウンの最適化を使用して、演算子を Amazon Redshift データベースにプッシュできることを示しています。

| 演算子 | プッシュダウン |
|---|---|
| + | X |
| - | X |
| * | X |
| / | X |
| % | X |
| \|\| | X |
| > | X |
| = | X |
| >= | X |
| <= | X |
| ! = | X |
| AND | X |
| OR | X |
| NOT | X |
| ^= | X |

## Amazon Redshift ODBC マッピングでサポートされているトランスフォーメーション

プッシュダウンの最適化を設定すると、Secure Agent は、データベースに対する設定済みのトランスフォーメーションのプッシュを試行します。

次の表に、Secure Agent g が Amazon Redshift ソースまたはターゲットにプッシュできるトランスフォーメーションロジックを示します。

| トランスフォーメーション | プッシュダウン |
|---|---|
| Aggregator | ソース、完全 |
| Expression | ソース、完全 |
| Filter | ソース、完全 |
| Joiner | ソース、完全 |
| Sorter | ソース、完全 |
| Union | ソース、完全 |
| Router | 完全 |

## プッシュダウンの最適化に関するルールおよびガイドライン

Amazon Redshift データベースに関数をプッシュする際には、以下のルールおよびガイドラインを使用します。

- TRUNC(DATE)を Amazon Redshift にプッシュするには、日付引数および形式引数を定義する必要があります。定義しないと、エージェントはこの関数を Amazon Redshift にプッシュしません。
- Amazon Redshift の集計関数が受け取る引数は 1 つだけで、それは集計関数のフィールドセットです。フィルタ条件の引数は無視されます。また、ターゲットにマップされているすべてのフィールドが GROUP BY 句に指定されていることを確認します。
- 更新オーバーライド ODBC 詳細ターゲットプロパティは、Amazon Redshift に接続する ODBC 接続を使用するときに適用できません。
- TO_DATE()を Amazon Redshift にプッシュするには、文字列および形式引数を定義する必要があります。
- TO_CHAR()を Amazon Redshift にプッシュするには、日付および形式引数を定義する必要があります。
- SYSTIMESTAMP()に形式を指定して、SYSTIMESTAMP を Amazon Redshift にプッシュしないでください。Amazon Redshift データベースは完全なタイムスタンプを返します。
- INSTR()を Amazon Redshift にプッシュするには、文字列、search_value、および開始引数のみを定義します。Amazon Redshift では、オカレンスおよび comparison_type 引数はサポートされていません。
- TO_BIGINT および TO_INTEGER を Amazon Redshift にプッシュすると、フラグ引数は無視されます。
- IN()を Amazon Redshift にプッシュすると、CaseFlag 引数は無視されます。
- ADD_TO_DATE()関数の一部として NS 形式を使用する場合、エージェントはこの関数を Amazon Redshift にプッシュしません。

- 次の形式のいずれかを TO_CHAR()および TO_DATE()関数の一部として使用する場合、エージェントはこれらの関数を Amazon Redshift にプッシュしません。

  - - NS

  - - SSSS

  - - SSSSS

  - - RR
- TRUNC(DATE)、GET_DATE_PART()、および DATE_DIFF()を Amazon Redshift にプッシュするには、次の形式を使用する必要があります。

  - - D

  - - DDD

  - - HH24

  - - MI

  - - MM

  - - MS

  - - SS

  - - US

  - - YYYY
- Redshift ODBC 接続を使用して DATE_DIFF()関数を Amazon Redshift にプッシュすると、Secure Agent によって誤った差分値が返されます。結果が正の場合、Secure Agent は負の値を返し、結果が負の場合、正の値を返します。
- ソースとして Amazon Redshift ODBC 接続を選択し、ソースオブジェクトとして外部テーブルを選択すると、データプレビューに失敗します。

# 第 12 章

# データ型リファレンス

データ統合では、Amazon Redshift とのマッピングおよびマッピングタスクに次のデータ型が使用されます。

Amazon Redshift **のネイティブデータ型**

Amazon Redshift のデータ型は、フィールドのメタデータを編集するときに、ソースおよびターゲットトランスフォーメーションに表示されます。

**トランスフォーメーションデータ型**

トランスフォーメーションで表示されるデータ型のセットです。ANSI SQL-92 汎用データ型に基づく内部データ型で、プラットフォーム間でデータを移動するとき Secure Agent によって使用されます。トランスフォーメーションデータ型は、マッピング内のすべてのトランスフォーメーションに表示されます。

データ統合は、ソースデータを読み取るときに、ネイティブデータ型を対応するトランスフォーメーションデータ型に変換してから、データのトランスフォームを実行します。データ統合は、ターゲットに書き込むときに、トランスフォーメーションデータ型を対応するネイティブデータ型に変換します。

## Amazon Redshift とトランスフォーメーションのデータ型

以下の表は、データ統合でサポートされる Amazon Redshift データ型と、対応するトランスフォーメーションデータ型を示しています。

| Amazon Redshift **のデータ型** | **トランスフォーメーションデータ型** | **説明と範囲** |
|---|---|---|
| Bigint | Bigint | 符号付き 8 バイト整数<br>-9223372036854775808〜9223372036854775807 |
| Boolean | Small Integer | 論理ブール型（true/false） |
| Char | String | 固定長文字列<br>最大値 4096 バイト |
| Date | Timestamp | カレンダーの日付（年、月、日）<br>紀元前 4713 年から西暦 294276 年 |
| Decimal | Decimal | 選択精度の正確な数値<br>最大精度は 38、位取りは 37 |

| Amazon Redshift **のデータ型** | **トランスフォーメーションデータ型** | **説明と範囲** |
|---|---|---|
| Double Precision | Double | 倍精度浮動小数点数<br>精度の有効数字 15 桁 |
| Integer | Integer | 符号付き 4 バイト整数<br>-2147483647〜+2147483648 |
| Real | Double | 単精度浮動小数点数<br>精度の有効数字 6 桁 |
| Smallint | Small Integer | 符号付き 2 バイト整数<br>-32768〜+32767 |
| Time | タイムスタンプ | 時間（タイムゾーンなし）<br>00:00:00.000000 - 23:59:59.999999<br>**注**: Time データ型は、詳細モードのマッピングには適用されません。 |
| タイムスタンプ | タイムスタンプ | 日時（タイムゾーンなし）。<br>紀元前 4713 年から西暦 294276 年（小数秒で最大 6 桁の精度） |
| Varchar | String | ユーザー定義で制限される可変長文字列<br>最大値 65535 バイト |

# データ型のルールとガイドライン

データ型については、次のルールおよびガイドラインを考慮します。

- 10 進データ型の場合、精度（38、37）の数値は、16 番目の精度の後で四捨五入されます。残りの桁はゼロに置き換えられます。
  例えば、ソース内の値 `2.31213563489578908888986998989899999990` はターゲットで `2.3121356348957890000000000000000000000` に四捨五入されます。

  この問題を解決するには、精度を 28 以下に指定します。

  Time データ型を使用してマッピングを実行すると、タスクは正常に実行されますが、出力の 10 進数値が切り捨てられます。Time データ型の場合、SQL クエリの `CallableStatement .setTime(java.sql.Time)`は、秒までの精度をサポートします。
- Time カラムを含むデータのマッピングを作成し、データ型を Time から Timestamp に編集すると、実行時にマッピングが失敗します。
- Decimal データ型のカラムを含むデータを読み取る場合、Decimal データ型のカラムに対して設定したスケールの変更は、デフォルトで考慮されます。スケールの変更が考慮されないようにするには、Secure Agent プロパティで JVM オプション *-honorDecimalScaleRedshift* の値を *false* に設定します。デフォルトでは、このプロパティは true に設定されます。
- Amazon Redshift との間で読み書きを行うと、マッピングのすべてのトランスフォーメーションのデータプレビューで、Time データ型にダミー値が追加されて表示されます。また、すべてのダウンストリームト

ランスフォーメーションも、追加されたダミーの日付値を受け取ります。ただし、時間の値は、既存の Amazon Redshift ターゲットでのみ正しく表示されます。

- 実行時にターゲットを作成し、入力値に Time カラムが含まれている場合、Time カラムはターゲットの Timestamp データ型に変換されます。Time カラムがソースに追加された場合に、スキーマを動的に更新すると、Timestamp データ型でターゲットにカラムが作成されます。
- Amazon Redshift からストアドプロシージャを呼び出すように SQL トランスフォーメーションを設定し、SQL トランスフォーメーションのパラメータとして小数秒を含む Time データ型を指定すると、精度が切り捨てられます。
- マッピングタスクのクエリで SESSSTARTTIME 変数を指定して Datetime 値を返すようにする場合は、次の形式でクエリを指定します。
  `:select to_timestamp('$$$SESSSTARTTIME','YYYY-MM-DD HH24:MI:SS.MS')::TIMESTAMP as xz;`

  詳細セッションプロパティの **［日時形式文字列］** を *YYYY-MM-DD HH24:MI:SS.MS* に設定する必要があります。

  マッピングタスクで SESSIONSTARTTIME にデフォルトの形式 *MM/DD/YYY* を使用する場合は、クエリで次の形式を使用する必要があります。

  `:select to_timestamp('$$$SESSSTARTTIME','MM-DD-YYYY HH24:MI:SS.MS')::TIMESTAMP as xz;`
- フィルタトランスフォーメーションを使用して Date、Timestamp、または Time カラムにフィルタを適用する場合は、次の形式で to_date()式を使用します。例えば、 `to_date('1970-01-01 23:05:06.123457','YYYY-MM-DD HH24:MI:SS.US'`と指定します。ソース文字列の形式と形式文字列が一致している必要があります。ソース値に基づいて、文字列の小数部分を変更します。TO_DATE 形式の文字列の詳細については、『トランスフォーメーション言語リファレンスガイド』を参照してください。
- ステージングパフォーマンスのために DTM ステージングを有効にして Amazon Redshift V2 ターゲットに書き込み、ソース値に Float データ型のデータがある場合、ターゲットの 10 進数値は切り捨てられ、最も近い整数に四捨五入されます。この問題を回避するには、［システム構成の詳細］セクションの［タイプ］でデータ統合サービスとして［DTM］を選択し、［DisableInfaDoubleHandlingForStaging］の値を［はい］に設定します。
- Time データ型カラムで簡易フィルタを使用して Amazon Redshift ソースから読み取り、マッピングを実行すると、タスクは実行時に次のエラーで失敗します: 操作に失敗しました: フィルタ条件[]の式文字列が無効です。
  回避策として、詳細フィルタを使用して Amazon Redshift ソースから読み取りを行います。

# 第 13 章

# トラブルシューティング概要

Amazon Redshift V2 コネクタのエラーをトラブルシューティングするには、以下のセクションを使用します。

## Amazon Redshift V2 コネクタのトラブルシューティング

### 更新/挿入操作に JVM オプションを使用する

更新/挿入操作の場合、JVM オプション-DuseDeleteForUpsert=true を設定して、ターゲットテーブルのすべてのフィールドを接続できます。JVM オプションを設定するときは、次のルールを使用します。

- JVM オプションを設定すると、Secure Agent は、ソーステーブルとターゲットテーブルの間のプライマリキーの一致に基づいて、ターゲットテーブルのレコードを削除します。ソーステーブルのすべてのレコードがターゲットテーブルに挿入されます。JVM オプションを設定しない場合、Secure Agent は既存の行を更新したり、挿入のマークが付けられているように他の行を挿入したりします。
- 詳細モードのマッピングの場合、マッピングタスクを作成するときに **Spark セッションプロパティ**で JVM オプションを設定します。
- すべてのカラムが複合キーを形成しているターゲットテーブルを使用し、ターゲットで更新カラムを設定しない場合、マッピングが失敗して警告が表示されることがあります。JVM オプションを設定すると、マッピングタスクは正常に実行されます。
- 更新カラムに重複する値があるターゲットテーブルを使用する場合、マッピングを実行した後、ターゲットテーブルの行数は少なくなります。JVM オプションを設定しない場合、ターゲットテーブルの行数は少なくなりません。
- カラムに重複する値があるソーステーブルを使用し、そのカラムをターゲットの更新カラムとして選択すると、追加の行がターゲットに追加されます。JVM オプションを設定しない場合、追加の行はターゲットに追加されません。
- 更新/挿入操作の場合、特殊文字を含むカラム名を持つテーブルを使用し、JVM オプションを設定しないと、マッピングは失敗します。

### Amazon S3 バケットが存在しないか、ユーザーにバケットにアクセスする権限がありません。

Secure Agent をホストするマシンの時刻を変更しないでください。Secure Agent の時刻は、タイムゾーンに従って正確である必要があります。正確でない場合、マッピングは例外で失敗します。

### デフォルト値を持つ NOT NULL カラム

Amazon Redshift ターゲットテーブルにデフォルト値を持つ NOT NULL カラムをマッピングしていない場合でも、挿入、更新、または更新/挿入操作は成功し、NOT NULL カラムのデフォルト値が使用されます。

Secure Agent プロパティで JVM オプション-DRetainUnmappedNotNullColumnValidation 値を true に設定すると、操作は成功せず、NOT NULL カラムのデフォルト値は使用されません。

## Amazon Redshift V2 コネクタを使用して更新/挿入操作を実装するにはどうすればよいですか?

更新/挿入操作の実装については、
Implementing the upsert operation using Amazon Redshift V2 Connector を参照してください。

## Amazon Redshift V2 コネクタの AWS IAM 認証はどのように設定しますか?

AWS IAM 認証の設定の詳細については、
Configuring AWS IAM Authentication for Amazon Redshift and Amazon Redshift V2 Connectors を参照してください。。

## Amazon Redshift V2 コネクタを使用して、Amazon Web Services（AWS）によって提供される Amazon Redshift Serverless に接続するにはどうすればよいですか?

Amazon Redshift Serverless への接続については、Using Amazon Redshift Serverless with Cloud Data Integration を参照してください。

# パート III: Amazon Redshift コネクタを使用したデータ統合

この部には、以下の章があります。


- Amazon Redshift コネクタの概要, 124 ページ
- Amazon Redshift 接続の概要, 129 ページ
- Amazon Redshift ソースとターゲット, 132 ページ
- Amazon Redshift による同期タスク, 143 ページ
- Amazon Redshift によるマッピングとマッピングタスク, 150 ページ
- Amazon Redshift のプッシュダウンの最適化, 162 ページ
- データ型リファレンスの概要, 172 ページ
- トラブルシューティング概要, 174 ページ

# 第 14 章

# Amazon Redshift コネクタの概要

Amazon Redshift コネクタを使用して、Amazon Redshift に対してデータの読み取りおよび書き込みを安全に実施できます。Amazon Redshift のソースとターゲットは、Amazon Redshift のレコードとして表されます。

Amazon Redshift 接続を作成し、同期タスク、マッピング、およびマッピングタスクでこの接続を使用できます。同期タスク、マッピング、およびマッピングタスクで Amazon Redshift オブジェクトを使用する場合は、Amazon Redshift に固有のプロパティを設定する必要があります。

Amazon Redshift 接続を作成して、タスクに追加する Amazon Redshift のソース、ルックアップ、およびターゲットの場所を指定します。マッピングを詳細モードに切り替えて、高度な機能を有効にするためのトランスフォーメーションと関数を含めることができます。Amazon Redshift の同期タスク、マッピング、またはマッピングタスクを実行する場合、エージェントはワークフローと Amazon Redshift の接続設定に基づいてデータを Amazon Redshift に書き込みます。

エージェントは、TCP/IP ネットワークを介して Amazon Simple Storage Service（Amazon S3）に接続し、データを書き込みます。Amazon S3 は、ソースからデータをコピーするのと同時にそのデータを Amazon Redshift クラスタに移動できるストレージサービスです。エージェントは、Amazon S3 から Amazon Redshift のターゲットテーブルにデータをコピーする copy コマンドを発行します。

Virtual Private Cloud（VPC）に配置された Amazon Redshift クラスタともデータの読み取りまたは書き込みを行うことができます。Amazon Redshift に対してデータの読み取りまたは書き込みを行う場合は、Hosted Agent または Secure Agent を指定できます。

任意のデータソースから Amazon Redshift にデータを移動できます。データ統合は Amazon ドライバを使用して Amazon Redshift と通信します。

**注:** Amazon Redshift V2 コネクタに新機能と改良点が提供されているため、Amazon Redshift V2 コネクタを使用することをお勧めします。

## Amazon Redshift コネクタのアセット

データ統合でアセットを作成し、Amazon Redshift コネクタを使用してデータを統合します。

Amazon Redshift コネクタを使用する場合は、次のようなデータ統合アセットを含めることができます。

- マッピング
- 同期タスク

アセットとトランスフォーメーションの設定に関する詳細については、データ統合のドキュメントの「*マッピング*」、「*トランスフォーメーション*」、および「*タスク*」を参照してください。

# Amazon Redshift の概要

Amazon Redshift は、組織がデータの分析と保存に使用できるクラウドベースのペタバイト規模のデータウェアハウスサービスです。

Amazon Redshift は、カラムナデータストレージ、並列処理、およびデータ圧縮を使用し、データを保存して高速なクエリ実行を実現しています。Amazon Redshift は、リーダーノードと計算ノードで構成されたクラスタベースのアーキテクチャを使用します。リーダーノードは計算ノードを管理し、外部クライアントプログラムと通信します。リーダーノードはクライアントアプリケーションと対話し、計算ノードと通信します。計算ノードはデータを保存し、リーダーノードのクエリを実行します。Amazon ドライバを使用するすべてのクライアントは、Amazon Redshift と通信できます。

# Amazon Redshift コネクタの例

顧客 ID、品目コード、品目数量など、注文書の詳細をオンプレミスの MySQL データベースに保存する組織で作業しています。注文書の詳細を分析し、オンプレミスの MySQL データベースから安価なクラウドベースの環境にデータを移動する必要があります。MySQL データベースからすべての購買レコードを読み取り、データ分析用の Amazon Redshift ターゲットに書き込むマッピングを作成します。

# Amazon Redshift コネクタの管理

ユーザーが Amazon Redshift コネクタを使用するには、成功ファイルおよびエラーファイルが格納された Secure Agent ディレクトリに対するアクセス権がユーザーに与えられていることを組織の管理者が確認する必要があります。このディレクトリパスは、各 Secure Agent マシンのランタイム環境で同じである必要があります。組織の管理者は次のタスクを実行する必要もあります。

- Amazon Redshift JDBC URL を取得します。
- 認証を管理します。次の 2 つの方法のいずれかを使用します。
  - アクセスキー ID とシークレットアクセスキーを作成する。
    Amazon Redshift 接続を設定する際に、そのアクセスキー ID とシークレットアクセスキーの値を指定します。アクセスキー ID とシークレットアクセスキーの作成の詳細については、AWS のマニュアルを参照してください。
  - AWS Identity and Access Management（IAM）認証を設定してセキュリティを強化する。
    IAM 認証を使用する場合は、アクセスキー ID とシークレットアクセスキーを Amazon Redshift 接続で明示的に指定しないでください。代わりに、Redshift ロールの Amazon リソース名（ARN）を作成し、最小の Amazon IAM ポリシーを Redshift ロールの ARN に追加して、Redshift ロールの ARN を Redshift クラスタに追加する必要があります。

    タスクを作成する場合、UNLOAD コマンドと COPY コマンドの AWS_IAM_ROLE オプションに Redshift ロールの ARN を指定します。

    接続プロパティのアクセスキー ID とシークレットアクセスキーおよび UNLOAD コマンドと COPY コマンドの AWS_IAM_ROLE の両方を指定した場合は、AWS_IAM_ROLE が優先されます。

    顧客マスタキーを使用した IAM 認証とサーバーサイド暗号化を使用する場合は、IAM EC2 ロールと IAM Redshift ロールを顧客マスタキーに追加する必要があります。

    Hosted Agent は IAM 認証をサポートしていません。Amazon Redshift コネクタの IAM 認証を設定する方法については、「IAM 認証」 (ページ 127)を参照してください。

- SSL 接続をサポートする場合は、Amazon Redshift を SSL 用に設定します。
- クライアントサイド暗号化を有効にする場合は、マスタ対称キーを作成します。
- サーバー側の暗号化を有効にする場合は、AWS Key Management Service（AWS KMS）で管理された顧客マスタキーを作成します。
- Amazon Redshift コネクタ用に最小の Amazon IAM ポリシーを作成します。
- ローカルステージング領域で更新/挿入、更新、または削除の操作のための一時テーブルを作成するときに、次の形式で一時テーブルを作成する必要があります。

  `RecordName + "_" + time-stamp + ProcessID + PartitionId`

  **注:** デフォルトでは、PUBLIC グループのメンバーシップを所有していると一時テーブルを作成する権限があります。権限を拒否するには、PUBLIC グループから TEMP 権限を取り消し、TEMP 権限を特定の個人またはグループの個人に許可します。

## SSL 用の Amazon Redshift の設定

Amazon Redshift への SSL 接続をサポートするように Secure Agent を設定できます。

1. 次の場所から Amazon Redshift 証明書をダウンロードします。
   https://s3.amazonaws.com/redshift-downloads/redshift-ssl-ca-cert.pem
2. 次のコマンドを実行してキーストアに証明書ファイルを追加します。`${JAVA_HOME}/bin/keytool –keystore {JAVA_HOME}/lib/security/cacerts -import -alias <string_value> -file <certificate_filepath>`
3. 管理者で、**［ランタイム環境］** を選択します。
4. 利用可能な Secure Agent のリストからメモリを増加する Secure Agent を選択します。
5. 右上隅の **［編集］** をクリックします。
6. **［システム構成の詳細］** セクションで、**［タイプ］** を **［DTM］** に変更します。
7. **［JVMOption1］** の横にある **［エージェント設定の編集］** アイコンをクリックし、次のコマンドを入力します。`- Djavax.net.ssl.trustStore=<keystore_name>`
8. **［JVMOption2］** の横にある **［エージェント設定の編集］** アイコンをクリックし、次のコマンドを追加します。`- Djavax.net.ssl.trustStorePassword=<password>`
9. Amazon Redshift 接続のプロパティで指定した JDBC URL に次のパラメータを追加します。`ssl=true`。例: `jdbc:redshift://mycluster.xyz789.us-west- 2.redshift.amazonaws.com:5439/dev?ssl=true`
10. **［OK］** をクリックして変更を保存します。

## 最小の Amazon IAM ポリシーの作成

Amazon IAM ポリシーを作成して、Amazon Redshift との間でデータの読み取りと書き込みを行う際に Amazon S3 でデータのステージングを実行するために必要なアクセス権限を定義します。

Amazon S3 でデータのステージングを実行するには、次の必要な最小の権限を使用します。

- PutObject
- GetObject
- DeleteObject
- ListBucket
- GetBucketPolicy

次のサンプル Amazon IAM ポリシーを使用できます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:ListBucket",
                                                                "s3:GetBucketPolicy"
                        ],
            "Resource": [
                "arn:aws:s3:::<bucket_name>/*",
                "arn:aws:s3:::<bucket_name>"
            ]
        }
    ]
}
```

セッションを正常に実行するには、Amazon S3 バケットと Amazon Redshift クラスタが同じ地域内にあることを確認する必要があります。

サポートされている地域は次のとおりです。

- アジアパシフィック（ムンバイ）
- アジアパシフィック（ソウル）
- アジアパシフィック（シンガポール）
- アジアパシフィック(シドニー)
- アジアパシフィック（東京）
- AWS GovCloud
- カナダ（中部）
- 中国（北京）
- 欧州（アイルランド）
- 欧州（フランクフルト）
- 南米（サンパウロ）
- 米国東部(バージニア北部)
- 米国東部(オハイオ)
- 米国西部（北カリフォルニア）
- 米国西部（オレゴン）

## IAM 認証

オプション。Amazon Elastic Compute Cloud（EC2）システムに場合、IAM 認証を設定できます。次を実行するときに、Amazon Redshift リソースへのアクセスを安全に制御するために IAM 認証を使用します。

EC2 システムにを実行する場合に、IAM 認証を使用します。IAM 認証を設定するには、次の手順を実行します。

1. 最小の Amazon IAM ポリシーを作成します。詳細については、「最小の Amazon IAM ポリシーの作成」(ページ 126)を参照してください。
2. Amazon EC2 ロールを作成します。EC2 ロールの作成中に、最小の Amazon IAM ポリシーを関連付けます。Amazon EC2 ロールは、Redshift クラスタで EC2 システムを作成する場合に使用されます。Amazon EC2 ロールの作成の詳細については、AWS のマニュアルを参照してください。

3. EC2 インスタンスを作成します。手順 2 で作成した Amazon EC2 ロールを EC2 インスタンスに割り当てます。
4. Amazon Redshift リソースへの安全なアクセスのために、Amazon Redshift ロールの ARN を作成します。Amazon Redshift ロールの作成中に、最小の Amazon IAM ポリシーを関連付けます。UNLOAD と COPY コマンドで Amazon Redshift ロールの ARN を使用できます。Amazon Redshift ロールの ARN の作成の詳細については、AWS のマニュアルを参照してください。
5. 読み取り操作および書き込み操作を正常に実行するために、Amazon Redshift ロールの ARN を Amazon Redshift クラスタに追加します。Amazon Redshift クラスタへの Amazon Redshift ロールの ARN の追加の詳細については、AWS のマニュアルを参照してください。
6. EC2 システムにをインストールします。

# 第 15 章

# Amazon Redshift 接続の概要

Amazon Redshift 接続を作成して、Amazon Redshift に対してデータの読み取りまたは書き込みを安全に実施します。Amazon Redshift 接続を使用すると、同期タスク、マッピングおよびマッピングタスクでソースまたはターゲットを指定できます。

接続を作成して、同期タスク、マッピング、またはマッピングタスクに関連付けます。Amazon Redshift に対してデータの読み取り/書き込みを行うために、ソースおよびターゲットのプロパティを定義します。

Amazon Redshift 接続は、**[接続]** ページで作成できます。作成した Amazon Redshift 接続は、マッピングの作成時に Mapping Designer で使用したり、タスクの作成時に同期タスクウィザードで使用したりできます。この接続は、組織全体で使用可能になります。

## Amazon Redshift 接続のプロパティ

Amazon Redshift 接続をセットアップする際には、接続プロパティを設定する必要があります。

次の表に、Amazon Redshift 接続プロパティを示します。

| 接続プロパティ | 説明 |
|---|---|
| ランタイム環境 | タスクを実行するランタイム環境の名前。 |
| ユーザー名 | Amazon Redshift アカウントのユーザー名。 |
| パスワード | Amazon Redshift アカウントのパスワード。 |
| スキーマ | Amazon Redshift スキーマ名。<br>デフォルトは public です。 |
| AWS アクセスキー ID | オプション。Amazon S3 バケットアクセスキー ID。<br>EC2 システムにインストールされた Secure Agent でタスクを実行するには、アクセスキー ID を空欄にする必要がある場合があります。<br>EC2 システムにインストールされたのでない Secure Agent でタスクを実行するには、アクセスキー ID を指定する必要があります。 |
| AWS シークレットアクセスキー | オプション。Amazon S3 バケットシークレットアクセスキー ID。<br>EC2 システムにインストールされた Secure Agent でタスクを実行するには、シークレットアクセスキーを空欄にする必要がある場合があります。<br>EC2 システムにインストールされたのでない Secure Agent でタスクを実行するには、シークレットアクセスキーを指定する必要があります。 |

| 接続プロパティ | 説明 |
| --- | --- |
| マスタ対称キー | オプション。Amazon S3 暗号化キー。<br>256 ビット AES 暗号化キーを Base64 形式で指定します。 |
| 顧客マスタキー ID | オプション。AWS Key Management Service（AWS KMS）によって生成された顧客マスタキー ID またはエイリアス名を指定します。Amazon S3 バケットが存在するリージョンの顧客マスタキー ID を生成する必要があります。顧客が生成した顧客マスタキー ID またはデフォルトの顧客マスタキー ID を指定できます。 |
| JDBC URL | Amazon Redshift 接続 URL。 |
| Varchar 用のマルチバイトをサポートするために必要なバイト数 | ［ターゲットの作成］に適用されます。ソーステーブルの Varchar 精度を参照して、ソース精度の 1x/2x/3x/4x 倍のターゲットテーブルを作成し、ターゲットテーブルにマルチバイト文字が正常に書き込めるようにします。<br>**注**: Varchar 精度が、最大である 65535 を超えている場合、ターゲットテーブルは作成できません。 |

**注:** 接続をテストすると、Secure Agent が Redshift 接続を検証します。AWS アクセスキーと AWS 秘密鍵の検証には、Amazon S3 バケット名が高度なソースおよびターゲットのプロパティ内に指定されている必要があります。そのため、Secure Agent は、同期またはマッピングタスクの実行時に、AWS アクセスキーと AWS 秘密鍵を検証します。

# プロキシの設定

組織で送信プロキシサーバーを使用してインターネットに接続する場合、Secure Agent は、そのプロキシサーバー経由で Informatica Intelligent Cloud Services に接続します。

**注:** 認証されていないプロキシサーバーのみを使用して Amazon Redshift に接続できます。

正しいプロキシ設定については、ネットワーク管理者にお問い合わせください。

## Windows でのプロキシ設定

Windows マシンで Secure Agent のプロキシサーバーを設定するには、Secure Agent を使用するか、Secure Agent の JVM オプションを使用します。

### Secure Agent Manager を使用したプロキシ設定

Secure Agent Manager を使用してプロキシサーバーを設定するには、次の手順を実行します。

1. **［開始］** > **［すべてのプログラム］** > **［Informatica Cloud Secure Agent］** > **［Informatica Cloud Secure Agent］**をクリックして、Secure Agent Manager を起動します。

   Secure Agent Manager に Secure Agent のステータスが表示されます。
2. ［Secure Agent Manager］ページで**［プロキシ］**をクリックします。
3. **［プロキシサーバーを使用］**をクリックして、プロキシサーバーの設定を入力します。

4. 次のプロキシサーバーの詳細を設定します。

| フィールド | 説明 |
|---|---|
| プロキシホスト | 必須。Secure Agent が使用する送信プロキシサーバーのホスト名。 |
| プロキシポート | 必須。送信プロキシサーバーのポート番号。 |

5. **［OK］** をクリックします。

   Secure Agent Manager によって Secure Agent が再起動され、設定が適用されます。

### JVM オプションを使用したプロキシの設定

1. Informatica Intelligent Cloud Services にログインします。
2. ［管理］を開いて **［ランタイム環境］** を選択します。
3. プロキシサーバーを設定する Secure Agent を選択します。
4. ページの右上隅にある **［編集］** をクリックします。
5. **［システム構成の詳細］** セクションで、データ統合サービスの **［タイプ］** に **［DTM］** を選択します。
   - **JVMOption** フィールドに次のパラメータを追加し、各パラメータに適した値を指定します。

| パラメータ | 説明 |
|---|---|
| -Dhttps.proxyHost= | 送信 HTTPS プロキシサーバーのホスト名。 |
| -Dhttps.proxyPort= | 送信 HTTPS プロキシサーバーのポート番号。 |

   以下に例を示します。

   ```
   JVMOption1=-Dhttps.proxyHost=<proxy_server_hostname>
   JVMOption2=-Dhttps.proxyPort=8081
   ```

6. **［保存］** をクリックします。

   Secure Agent が再起動して設定が適用されます。

## Linux でのプロキシ設定

Secure Agent のインストーラにより、ブラウザで設定されている設定項目に基づいて Secure Agent のプロキシサーバー設定が設定されます。Secure Agent に定義されているプロキシサーバーの設定は、コマンドラインから更新できます。

Linux マシンで Secure Agent のプロキシサーバーを設定するには、`proxy.ini` ファイルを更新するシェルコマンドを使用します。ネットワーク管理者に問い合わせて、プロキシの設定項目を決めてください。

1. 次のディレクトリに移動します。

   ```
   <Secure Agent のインストールディレクトリ>/apps/agentcore
   ```

2. 次のコマンドを使用して、`proxy.ini` ファイルを更新します。

   ```
   consoleAgentManager.bat configureProxy <proxy host> <proxy port>
   ```

3. Secure Agent を再起動します。

# 第 16 章

# Amazon Redshift ソースとターゲット

Amazon Redshift 接続のソースとターゲットのプロパティを割り当てることができます。

## Amazon Redshift ソース

Amazon Redshift オブジェクトを、同期タスク、マッピング、またはマッピングタスクのソースとして使用できます。同期タスクでソースとして複数の関連する Amazon Redshift 標準オブジェクトを使用することもできます。

Amazon Redshift ソースオブジェクトを使用するときに、標準オブジェクトをプライマリソースとして選択してから、子オブジェクトを追加することができます。

詳細ソースプロパティを設定するとき、Amazon Redshift 固有のプロパティを設定します。データを暗号化し、Amazon S3 にステージングファイルを保持して、Amazon Redshift でファイルにクエリの結果を安全にアンロードできます。

### Amazon Redshift ソースの Amazon Redshift ステージングディレクトリ

エージェントは、ソースプロパティで指定したディレクトリにステージングファイルを作成します。同期タスク、マッピングおよびマッピングタスクは、Amazon Redshift からデータを読み込む前にステージングディレクトリにデータをステージングします。エージェントは、タスクが完了するときに、ステージングディレクトリからステージングファイルを削除します。

Hosted Agent のディレクトリは設定できません。Hosted Agent は、一時的な場所にデータをステージングするディレクトリを作成し、タスクが完了するときにその一時的な場所からそのステージングファイルを削除します。

タスクパフォーマンスを向上させるには、ステージングファイルの圧縮を有効にします。処理するデータのボリュームに最適なディスク領域を使用して、ステージングディレクトリを指定します。各 Secure Agent マシンのランタイム環境で使用可能なディレクトリパスを指定します。

アプリケーションは、タスクが実行している時間を基に、ステージングディレクトリにサブディレクトリを作成します。サブディレクトリは次の命名規則を使用します。

```
<staging directory>/infaRedShiftStaging<MMddHHmmssSSS+xyz>
```

# Amazon Redshift ソースのサーバーサイド暗号化

Amazon Redshift からファイルを取得して Amazon S3 にファイルをステージングするときに Amazon Redshift でデータを暗号化する場合は、サーバーサイド暗号化を有効にする必要があります。

サーバーサイド暗号化用に接続プロパティで AWS Key Management Service（AWS KMS）によって生成された顧客マスタキー ID を設定できます。IAM 認証と、顧客マスタキーを使用したサーバーサイド暗号化を使用する場合は、IAM EC2 ロールと IAM Redshift ロールを顧客マスタキーに追加する必要があります。

詳細ターゲットプロパティでサーバーサイド暗号化を選択する場合は、接続プロパティで顧客マスタキー ID を指定する必要があります。

**注:** Amazon S3 のステージングファイルは、タスクの完了後に削除されます。

## Amazon Redshift ソースのデータ暗号化

サーバーサイド暗号化の AWS Key Management Service（AWS KMS）によって生成された顧客マスタキー ID を使用すると、データを暗号化できます。

**［スケジュール］**ページの Amazon Redshift の詳細ソースプロパティにある**［暗号化タイプ］**フィールドで、暗号化のタイプを選択できます。Unload コマンドは、AWS が管理する暗号化キーと AWS Key Management Service キーを使用して、サーバーサイド暗号化用に Amazon S3 に一時ファイルを作成します。

サーバーサイド暗号化用に Unload コマンドで AWS Key Management Service によって生成された顧客マスタキー ID を使用します。以下のタイプの暗号化を選択できます。

SSE-S3

**［SSE-S3］**暗号化タイプを選択した場合、Unload コマンドは Amazon S3 バケットに一時ファイルを作成し、Amazon S3 はサーバーサイド暗号化用の AWS が管理する暗号化キーを使用してファイルを暗号化します。

SSE-KMS

**［SSE-KMS］**暗号化タイプを選択した場合、Unload コマンドは Amazon S3 バケットに一時ファイルを作成し、Amazon S3 はサーバーサイド暗号化用の AWS KMS が管理する顧客マスタキーを使用してファイルを暗号化します。

接続プロパティで指定した AWS KMS が管理する顧客マスタキーは、Amazon S3 をホストする地域と同じ地域に属する必要があります。例えば、Amazon S3 が**［米国西部（オレゴン）］**地域でホストされている場合、**［SSE-KMS］**暗号化タイプを選択すると、同じ地域で有効にされた AWS KMS が管理する顧客マスタキーを使用する必要があります。

**S3 クライアントサイドの暗号化の有効化**プロパティを有効にして、**暗号化タイプ**を **SSE-S3** として選択した場合、Amazon S3 はクライアントサイド暗号化にマスタ対称キーを使用してデータを暗号化します。

**S3 クライアントサイドの暗号化の有効化**プロパティを有効にして、**暗号化タイプ**を **SSE-KMS** として選択した場合、Amazon S3 はクライアントサイド暗号化に AWS Key Management Service によって生成された顧客マスタキー ID を使用してデータを暗号化します。

**注:** Amazon Redshift コネクタは、マスタ対称キーを使用するサーバーサイド暗号化と、顧客マスタキーを使用するクライアントサイド暗号化をサポートしていません。

# Amazon Redshift ソースのクライアントサイド暗号化

クライアントサイド暗号化は、データを Amazon Redshift サーバーに送信する前にデータを暗号化する方法です。

Amazon Redshift ソースに対してクライアントサイド暗号化を有効にすると、Amazon Redshift は暗号化されたデータをアンロードしてから、そのデータを Secure Agent にプッシュします。Secure Agent は、タスクまたはマッピングロジックに基づいてデータをターゲットに書き込みます。

クライアントサイド暗号化を有効にするには、接続プロパティでマスタ対称キーを指定し、詳細ターゲットプロパティで［**S3 クライアントサイドの暗号化の有効化**］オプションを選択する必要があります。

Secure Agent は、マスタ対称キーを使用してデータを暗号化します。マスタ対称キーは Base64 形式の 256 ビット AES 暗号化キーです。Amazon Redshift コネクタはマスタ対称キーを使用して Amazon S3 サーバーにデータをアップロードし、［暗号化］オプションを設定した copy コマンドと追加のセキュリティ用のプライベート暗号化キーを使用して、データをロードします。

# Unload コマンド

Unload コマンドを使用すると、Amazon Redshift からデータを抽出し、Amazon S3 でステージングファイルを作成できます。Unload コマンドはセキュアな接続を使用して、Amazon S3 上の 1 つ以上のファイルにデータをロードします。

Unload コマンドオプションを［**UnloadOptions プロパティファイル**］フィールドで直接指定できます。オプションを大文字で入力し、オプションの区切りには、Unload コマンドには次のオプションとデフォルト値があります。

プロパティファイルには Unload コマンドオプションが含まれます。［**UnloadOptions プロパティファイル**］フィールドにプロパティファイルパスを追加します。以下に例を示します。

```
C:\Temp\Redshift\unloadoptions.txt
```

印刷できない文字の 8 進数の表現を DELIMITER および QUOTE として使用することを推奨します。

pre-SQL コマンドまたは post-SQL コマンドとして Unload コマンドを実行する場合は、`ALLOWOVERWRITE` オプションを指定して既存のオブジェクトを上書きします。

## Unload コマンドのオプション

Unload コマンドのオプションは Amazon Redshift からデータを抽出し、Amazon S3 のステージングファイルに特定の形式でデータをロードします。特定の文字を使用してデータを区切ったり、複数のファイルにデータを並行して読み込んだりできます。

Unload コマンドにオプションを追加するには、［**UnloadOptions プロパティファイル**］オプションを使用します。以下のオプションを設定できます。

DELIMITER

入力ファイル内のフィールドを区切るための単一の ASCII 文字。パイプ（|）、チルダ（~）、タブ（\t）などの文字を使用できます。指定した区切り文字をデータに含めることはできません。区切り文字がデータに含まれる場合、ESCAPE を使用すると、区切り文字が通常の文字として読み取られます。デフォルトは \036 で、印刷できない 8 進数のレコード区切り文字です。

ESCAPE

次の文字が出現する前に、区切りアンロードファイルに CHAR カラムと VARCHAR カラムのエスケープ文字を追加できます。

- 改行（\n）
- キャリッジリターン（\r）
- アンロードデータに指定される区切り文字
- エスケープ文字（\）

- 一重引用符または二重引用符

デフォルトは OFF です。

PARALLEL

Unload コマンドは、クラスタ内のスライス数に従ってデータを複数のファイルに並行して書き込みます。デフォルトは ON です。並行オプションをオフにすると、Unload コマンドはデータを順番に書き込みます。データファイルの最大サイズは 6.5 GB です。

AWS_IAM_ROLE

Amazon EC2 システムにインストールされたでを実行する Amazon Redshift ロールリソース名（ARN）を次の形式で指定します。`AWS_IAM_ROLE=arn:aws:iam::<account ID>:role/<role-name>`

例: `arn:aws:iam::123123456789:role/redshift_read`

ADDQUOTES

ADDQUOTES は、デフォルトでは UNLOAD コマンドで実行されます。詳細ソースプロパティで ADDQUOTES オプションを指定しないでください。Unload コマンドは各データフィールドに引用符を追加します。引用符を追加すると、UNLOAD コマンドは区切り文字を含むデータ値を読み取ることができます。二重引用符（"）がデータに含まれる場合、ESCAPE を使用すると、区切り文字が通常の文字として読み取られます。

## パーティション化

Amazon Redshift からデータを読み取る際には、パーティション化を設定して実行時のマッピングのパフォーマンスを最適化できます。パーティションタイプによって、パーティションポイントでパーティションにデータを配布する方法を制御します。

キー範囲パーティションなどのパーティションタイプを定義できます。1 つまたは複数のフィールドの値に基づいて Amazon Redshift データをパーティション化するようにキー範囲パーティション化を設定します。キー範囲パーティション化により、Secure Agent はパーティションキーとして定義したフィールドに基づいてソースデータ行を分散します。Secure Agent は、フィールド値を各パーティションの範囲値と比較して、適切なパーティションに行を送信します。

データ値が均等に分散されているカラムには、キー範囲パーティション化を使用します。そうしないと、パーティションのサイズが等しくなくなる可能性があります。例えば、あるカラムのキー値 1 から 1000 の間には 10 行あり、キー値 1001 から 2000 の間には 999 行があるというようになります。

キー範囲パーティション化では、あるパーティションのクエリが別のパーティションよりも先に行を返す場合があります。または、あるパーティションは行を返し、他のパーティションは行を返さない場合もあります。テーブル内の行がキー範囲と同じ順序である場合にこのようなことが起こります。あるクエリは行を読み取って行を返し、他のクエリは同じ行を読み取ってフィルタすることもあります。

**注:** 推奨されるパーティションの最大数は 32 です。32 を超えるパーティションを設定すると、マッピングタスクがメモリバッファエラーで失敗する場合があります。

# Amazon Redshift ターゲット

Amazon Redshift オブジェクトを、同期タスク、マッピング、またはマッピングタスクでの単一ターゲットとして使用できます。入力ソースに基づいて Amazon Redshift ターゲットを作成することもできます。Amazon

Redshift ターゲットオブジェクトを使用する場合は、標準オブジェクトをプライマリソースとして選択できます。

Amazon Redshift ターゲットからデータを挿入、更新、更新/挿入、および削除することができます。バッチ内にエラーが発生しない場合、更新または挿入タスクによってバッチ全体が Amazon Redshift ターゲットに書き込まれます。バッチ内にエラーが発生した場合は、Secure Agent によってバッチ全体がエラー行ファイルに書き込まれます。

詳細ターゲットプロパティを設定するとき、Amazon Redshift 固有のプロパティを設定します。データを暗号化したり、クエリ効率向上のためにデータベーステーブルの統計メタデータを更新したり、Amazon S3 バケットのフラットファイルから Amazon Redshift にデータをロードしたり、ディスク領域の回復とテーブル内の行のソートのためにテーブルのクリーンアップを使用したりできます。

マッピングにフラットファイルまたは Amazon Redshift ターゲットが含まれる場合、既存のターゲットを使用するか、実行時に新しいターゲットを作成するかを選択できます。Amazon Redshift ターゲットオブジェクト名は小文字で指定する必要があります。

**注:** ターゲットテーブルの配布キーカラムに NULL 値が含まれていて、同じターゲットテーブルの更新/挿入操作を使用してタスクを設定すると、タスクにより重複する行が作成される場合があります。重複する行の作成を防ぐには、次のいずれかのタスクを実行する必要があります。

- データをロードするときに、NULL 値を NULL 以外の値に置き換えます。
- 配布キーカラム内に NULL 値を想定する場合は、カラムを配布キーとして設定しないようにします。
- データをロードするときに、ターゲットテーブルから配布キーカラムを一時的に削除します。Pre-SQL と Post-SQL のプロパティを使用して、ターゲットテーブルの配布キーカラムを削除してから、追加することができます。

## Amazon Redshift ターゲットの Amazon Redshift ステージングディレクトリ

エージェントは、ターゲットプロパティで指定したディレクトリにステージングファイルを作成します。同期タスク、マッピング、および、マッピングタスクは、Amazon Redshift にデータを書き込む前にステージングディレクトリにデータをステージングします。ステージングファイルを保持または削除するタスクを設定できます。

Hosted Agent のディレクトリは設定できません。Hosted Agent は、一時的な場所にデータをステージングするディレクトリを作成し、タスクが完了するときにその一時的な場所からそのステージングファイルを削除します。

タスクパフォーマンスを向上させるには、ステージングファイルの圧縮を有効にします。処理するデータのボリュームに最適なディスク領域を使用して、ステージングディレクトリを指定します。各 Secure Agent マシンのランタイム環境で使用可能なディレクトリパスを指定します。

アプリケーションは、タスクが実行している時間を基に、ステージングディレクトリにサブディレクトリを作成します。サブディレクトリは次の命名規則を使用します。

```
<staging directory>/infaRedShiftStaging<MMddHHmmssSSS+xyz>
```

## ターゲットテーブルの分析

クエリのパフォーマンスを最適化するには、ターゲットテーブルを分析するためのタスクを設定できます。ターゲットテーブル分析は、データベーステーブルの統計メタデータを更新します。

**[ターゲットテーブルの分析]** オプションを使用すると、テーブルからサンプル行を抽出し、サンプルを分析して、列の統計情報を保存できます。Amazon Redshift は、統計メタデータを使用してクエリプランナーを更新します。クエリプランナーは、統計メタデータを使用して、クエリ効率の向上に最適な計画を作成して選択します。

Copy コマンドを使用して既存のテーブルにデータをロードしたら、［**ターゲットテーブルの分析**］オプションを実行できます。新しいテーブルにデータをロードした場合、Copy コマンドはデフォルトで分析を実行します。

# Amazon Redshift ターゲットのデータ暗号化

データを保護するために、サーバーサイド暗号化またはクライアントサイド暗号化を有効にして、Amazon Redshift に挿入するデータを暗号化できます。

Amazon Redshift ターゲットでサーバーサイドとクライアントサイドの両方の暗号化を有効にすると、クライアントサイド暗号化がデータのロードに使用されます。

## Amazon Redshift ターゲットのサーバーサイド暗号化

.csv ファイルを Amazon Redshift にアップロードするときに Amazon Redshift でデータを暗号化する場合は、サーバーサイド暗号化を有効にする必要があります。サーバーサイド暗号化を有効にするには、［**スケジュール**］ページの詳細ターゲットプロパティで、暗号化タイプとしてサーバーサイド暗号化を選択します。

サーバーサイド暗号化用に接続プロパティで AWS Key Management Service（AWS KMS）によって生成された顧客マスタキー ID を設定できます。IAM 認証と、顧客マスタキーを使用したサーバーサイド暗号化を使用する場合は、IAM EC2 ロールと IAM Redshift ロールを顧客マスタキーに追加する必要があります。詳細ターゲットプロパティでサーバーサイド暗号化を選択し、接続プロパティで顧客マスタキー ID を指定しない場合、データの暗号化には Amazon S3 が管理する暗号化キーが使用されます。

## Amazon Redshift ターゲットのクライアントサイド暗号化

クライアントサイド暗号化は、データを Amazon Redshift サーバーに送信する前にデータを暗号化する方法です。

Amazon Redshift ターゲットに対してクライアントサイド暗号化を有効にすると、Secure Agent はソースからデータを取得し、そのデータをステージングディレクトリに書き込み、そのデータを暗号化してから Amazon S3 バケットに書き込みます。次に、Amazon S3 バケットは Amazon Redshift にデータを書き込みます。

クライアントサイド暗号化を有効にするには、接続プロパティでマスタ対称キーを指定し、詳細ターゲットプロパティで［**S3 クライアントサイドの暗号化の有効化**］オプションを選択する必要があります。

Secure Agent は、マスタ対称キーを使用してデータを暗号化します。マスタ対称キーは Base64 形式の 256 ビット AES 暗号化キーです。Amazon Redshift コネクタはマスタ対称キーを使用して Amazon S3 サーバーにデータをアップロードし、［暗号化］オプションを設定した copy コマンドと追加のセキュリティ用のプライベート暗号化キーを使用して、Amazon Redshift にデータをロードします。クライアントサイド暗号化を有効にするには、次のタスクを実行します。

# ステージングファイルの保持

がターゲットにデータを書き込むと、Amazon S3 でステージングファイルを保持できます。ファイルを保持すると、Amazon S3 で組織データのデータレイクを作成できます。保存したファイルはデータのバックアップとしても使用できます。

ターゲット接続を作成する場合、ステージングファイルの保存用にファイルプレフィックスまたはディレクトリプレフィックスを設定できます。プレフィックスを指定すると、は、ターゲット接続で指定した Amazon S3

の場所にあるディレクトリ内でファイルを作成します。［**S3 でステージングファイルを保持するためのプレフィックス**］プロパティに次のいずれかのオプションを設定します。

- ディレクトリプレフィックスおよびファイルプレフィックスを指定します。例えば、backup_dir/backup_file です。は次のディレクトリとファイルを作成します。

  - backup_dir_<year>_<month>_<date>_<timestamp_inLong>

  - backup_file.batch_<batch_number>.csv.<file_number>.<encryption_if_applicable>

- ファイルプレフィックスを指定します。例えば、backup_file です。は次のディレクトリとファイルを作成します。

  - <year>_<month>_<date>_<timestamp_inLong><3 digit of random number>00<ProcessID><PartitionId>

  - backup_file.batch_<batch_number>.csv.<file_number>.<encryption_if_applicable>

- プレフィックスを指定しないでください。はステージングファイルを保存しません。

# Copy コマンド

Copy コマンドを使用すると、テーブルにデータを追加できます。Copy コマンドはセキュアな接続を使用して、ソースから Amazon Redshift にデータをロードします。

Copy コマンドオプションを［**CopyOptions プロパティファイル**］フィールドで直接指定できます。オプションを大文字で入力し、オプションの区切りには、を使用します。Copy コマンドには次のオプションと既定値があります。

プロパティファイルには Copy コマンドオプションが含まれます。［**CopyOptions プロパティファイル**］フィールドにプロパティファイルパスを追加します。以下に例を示します。

```
C:\Temp\Redshift\copyoptions.txt
```

印刷できない文字の 8 進数の表現を DELIMITER および QUOTE として使用することを推奨します。

## Copy コマンドのオプション

Copy コマンドオプションは、Amazon S3 からデータを読み取り、特定の形式で Amazon Redshift にデータを書き込みます。テーブル内のデータに圧縮を適用したり、特定の文字でデータを区切ることができます。

Copy コマンドにオプションを追加するには、［**CopyOptions プロパティファイル**］オプションを使用します。以下のオプションを設定できます。

DELIMITER

入力ファイル内のフィールドを区切るための単一の ASCII 文字。パイプ（|）、チルダ（~）、タブ（\t）などの文字を使用できます。区切り文字をデータに含めることはできません。デフォルトは\036 で、印刷できない 8 進数のレコード区切り文字です。

ACCEPTINVCHARS

データに無効な UTF-8 文字が含まれていても、VARCHAR 列にデータをロードします。ACCEPTINCHARS を指定すると、は、無効な UTF-8 文字を ACCEPTINVCHARS で指定された文字からなる同じ長さの文字に置き換えます。ACCEPTINVCHARS で'|'を指定した場合、は 3 バイトの UTF-8 文字を'|||'に置き換えます。

ACCEPTINVCHARS を指定しない場合、COPY コマンドは、無効な UTF-8 文字が検出されるとエラーを返します。VARCHAR 列で ACCEPTINVCHARS オプションを使用できます。デフォルト疑問符（?）です。

QUOTE

カンマ区切り値で使用する引用符文字を指定します。デフォルトは\037 で、印刷できない 8 進数のユニット区切り文字です。

COMPUPDATE

現在の圧縮エンコーディングをオーバーライドし、空のテーブルに圧縮を適用します。テーブル内の行が 10 万を超える場合は挿入操作で COMPUPDATE オプションを使用します。COMPUPDATE の動作は設定方法に応じて異なります。

- COMPUPDATE を指定しない場合、ターゲットテーブルが空で、テーブル内のすべての列で RAW またはエンコーディングが指定されないと、COPY コマンドは圧縮を適用します。
- COMPUPDATE を［オン］に指定した場合、ターゲットテーブルが空で、テーブル内の列で RAW 以外のエンコーディングが指定されると、COPY コマンドは既存のエンコーディングを置き換えます。
- COMPUPDATE を［オフ］に指定した場合、COPY コマンドは圧縮を適用しません。

デフォルトは OFF です。

AWS_IAM_ROLE

Amazon EC2 システムにインストールされたでを実行する Amazon Redshift ロールリソース名（ARN）を次の形式で指定します。`AWS_IAM_ROLE=arn:aws:iam::<account ID>:role/<role-name>`

例: `arn:aws:iam::123123456789:role/redshift_write`

## フィールドマッピング

フィールドマッピングページにプライマリキーフィールドのキーアイコンが表示されます。フィールドマッピングを設定するときに、すべてのキーフィールドと［非 NULL］フィールドをマッピングして、データを Amazon Redshift ターゲットに正常に挿入または更新/挿入します。Amazon Redshift は、［非 NULL］フィールドを適用しても、キー制約を適用しません。

フィールドマッピングページにプライマリキーフィールドのキーアイコンが表示されます。その他の Amazon Redshift キータイプはマークされません。更新操作にキーのないフィールドをマッピングする必要があります。フィールドマッピングに Amazon Redshift の ID フィールドを使用する場合、使用可能な ID をすべてマッピングするか、何もマッピングしません。ID フィールドには、Amazon Redshift で自動的に生成されるデータが含まれます。

ID カラムがキーに含まれない場合は、フィールドマップに ID カラムをマッピングできません。ID カラムがキーに含まれる場合は、フィールドマップに ID カラムをマッピングする必要があります。ただし、ID カラムの値をソースから設定することはできません。

## テーブルのクリーンアップ

テーブルのクリーンアップを使用してディスク領域を回復し、指定したテーブルまたはデータベース内のすべてのテーブルの行をソートします。

削除やロードなどの一括操作を実行した後、または差分更新を実行した後、データベーステーブルをクリーンアップしてディスク領域を回復し、Amazon Redshift のクエリパフォーマンスを向上する必要があります。行の削除や更新を実行しても、Amazon Redshift は空き領域を再利用しません。

データベースまたはテーブルをクリーンアップすると、多くの場合、クエリパフォーマンスの一貫性を維持できます。データベース全体またはデータベース内の個々のテーブルのディスク領域を回復できます。データベースまたは指定されたデータベース管理スケジュールで最小限の操作を求める場合は、クリーンアップを実行する必要があります。クリーンアップを長く実行すると、データベース操作に影響を与える場合があります。大規模な未ソートの領域では、クリーンアップが長くなることが多いため、クリーンアップを実行します。

詳細ターゲットプロパティを設定する場合、テーブルのクリーンアップオプションを有効にできます。次のリカバリオプションを選択できます。

**なし**

行のソートまたはディスク領域の回復は行わないでください。

**フル**

指定したテーブルまたはデータベース内のすべてのテーブルをソートし、前回の更新操作や削除操作で削除のマークが付けられた行によって使用されたディスク領域を回復します。

**ソートのみ**

削除行から解放された領域を回復せずに、指定したテーブルまたはデータベース内のすべてのテーブルをソートします。

**削除のみ**

前回の更新操作や削除操作で削除のマークが付けられた行によって使用されたディスク領域を回復し、テーブルを圧縮して使用領域を解放します。

**再インデックス**

インターリーブソートキーカラムで値の分布を分析し、**テーブルのクリーンアップ**操作全体を設定してパフォーマンスを改善します。

# 大きいテーブルに関する作業

大きなオブジェクトを、複数の独立したパーツのセットとしてアップロードまたはダウンロードできます。

Amazon Redshift コネクタは AWS TransferManager API を使用して、大きなオブジェクトを複数パーツに分けて Amazon S3 にアップロードします。大きいオブジェクトをダウンロードしている間に、Secure Agent は Amazon S3 から複数パーツにオブジェクトをダウンロードします。

ファイルサイズが 5MB を超える場合、マルチパーツアップロードを設定して、複数パーツに分けたオブジェクトを並行してアップロードできます。Amazon S3 オブジェクトのファイルサイズが 12 MB より大きい場合、並行して複数パーツにオブジェクトをダウンロードできます。

詳細ソースプロパティで、［**S3 ファイルの分割ダウンロードの有効化**］オプションを設定できます。詳細ターゲットプロパティで、［**パートサイズ**］および［**TransferManager スレッドプールサイズ**］オプションを設定できます。

# DELIMITER および QUOTE としての 8 進数値

印刷できる ASCII 文字に加えて、印刷できる ASCII 文字と印刷できない ASCII 文字の 8 進数値を DELIMITER および QUOTE として使用できます。

印刷できる文字を DELIMITER または QUOTE として使用するには、ASCII 文字またはそれぞれの 8 進数値を指定します。ただし、印刷できない文字を DELIMITER または QUOTE として使用するには、それぞれの 8 進数値を指定する必要があります。

印刷できる文字の例:

`DELIMITER=#`または `DELIMITER=\043`

印刷できない文字の例、ファイルセパレータ:

QUOTE=\034

8 進数値 000-037 と 177 は印刷できない文字を表し、040-176 は印刷できる文字を表します。次の表に、Amazon Redshift によりサポートされる、Copy コマンドの QUOTE と DELIMITER、および Unload コマンドの DELIMITER の推奨 8 進数値を示します。

| コマンドオプション | 推奨 8 進数値 |
|---|---|
| COPY QUOTE | 001-010、016-037、041-054、057、073-100、133、135-140、173-177 |
| COPY DELIMITER | 001-011、013、014、016、017、020-046、050-054、057、073-133、135-177 |
| UNLOAD DELIMITER | 001-011、013、014、016、017、020-041、043-045、050-054、056-133、135-177 |

# 成功ファイルおよびエラーファイル

Secure Agent は、セッションの実行後に成功ファイルとエラーファイルを生成します。成功ファイルとエラーファイルは、行レベルの詳細を含む.csv ファイルです。Hosted Agent は、セッションの実行後に成功ファイルとエラーファイルを作成しません。

Secure Agent は、セッションの実行後に成功ファイルを生成します。成功ファイルには、Amazon Redshift に正常に書き込まれた各レコードのエントリが含まれます。各エントリには、レコードのすべてのフィールドに対して書き込まれた値が含まれます。このファイルを使用すると、Secure Agent が Amazon S3 バケットと Amazon Redshift ターゲットに書き込んだデータを把握できます。

エラーファイルには、各データエラーのエントリが含まれます。ファイルの各エントリには、レコードのすべてのフィールドの値とエラーメッセージが含まれます。エラーファイルを使用すると、Secure Agent が Amazon Redshift ターゲットにデータを書き込まない理由を把握できます。

Secure Agent は成功ファイルまたはエラーファイルを上書きしません。エラー行ファイルと成功行ファイルが生成されたディレクトリからこれらのファイルに直接アクセスします。**[すべてのジョブ]** ページからエラー行ファイルにアクセスすることはできません。不要になったファイルは手動で削除できます。

成功ファイルのセッションプロパティを設定する場合は、次のガイドラインを考慮してください。

- デフォルトでは、成功行ファイルはディレクトリ<Secure Agent installation directory>/apps/Data_Integration_Server/data/success に生成されます。**成功ファイルディレクトリ**詳細ターゲットオプションを使用して異なるディレクトリを指定することができます。
- 成功行ファイルは次の命名規則を使用します。infa_rs_<operation>_<schema.table_name>.batch_<batch_number>_file_<file_number>_<timestamp>_success.csv。

エラーファイルのセッションプロパティを設定する場合は、次のガイドラインを考慮してください。

- デフォルトでは、エラー行ファイルはディレクトリ<Secure Agent installation directory>/apps/Data_Integration_Server/data/error に生成されます。**エラーファイルディレクトリ**詳細ターゲットオプションを使用して異なるディレクトリを指定することができます。
- エラーファイルディレクトリを定義する場合は、変数$PMBadFileDir を使用します。変数$PMBadFileDir を使用すると、アプリケーションがファイルを次の Secure Agent ディレクトリに書き込みます。<Secure Agent インストールディレクトリ>/apps/Data_Integration_Server/data/error。

- 挿入タスクの場合、エラー行ファイルは次の命名規則を使用します。
  infa_rs_<operation>_<schema.table>.batch_<batch_number>_file_<file_number>_<timestamp>_error.csv。更新/挿入タスクの場合、次の命名規則を使用します。
  infa_rs_<operation>_<schema.table>_<timestamp_inLong>.batch_<batch_number>_file_<file_number>_<timestamp>_error.csv。

# 第 17 章

# Amazon Redshift による同期タスク

このセクションでは、同期タスクのソース、ターゲット、およびルックアッププロパティを設定する方法について解説します。

## 同期タスクの Amazon Redshift ソース

Amazon Redshift ソースプロパティは、同期タスクウィザードの［**ソース**］ページで設定します。

パフォーマンスを最適化するには、［**データフィルタ**］ページでフィルタを設定します。簡易フィルタまたは詳細フィルタを設定し、ソースの行を削除します。データフローの初期段階でフィルタリングを行うことによって効率を上げることができます。簡易フィルタには、フィールド名、演算子、値が含まれます。 詳細フィルタを使用するとより複合フィルタ条件を定義できます。例えば、AND 論理演算子または OR 論理演算子を使用して、複数の条件を組み込むことができます。

次の表に、Amazon Redshift のソースプロパティを示します。

| プロパティ | 説明 |
|---|---|
| 接続 | ソース接続の名前。 |
| ソースタイプ | ソースオブジェクトのタイプ単一または複数を選択します。 |
| ソースオブジェクト | ソースオブジェクトの名前。単一のソースまたは複数の関連するソースのソースオブジェクトを選択します。 |

Amazon Redshift ソースを使用するように同期タスクを設定するときに、詳細ソースプロパティを設定できます。詳細ソースプロパティは、同期タスクウィザードの［**スケジュール**］ページに表示されます。

次の表に、Amazon Redshift の詳細なソースプロパティを示します。

| 詳細プロパティ | 説明 |
|---|---|
| S3 バケット名 | Amazon Redshift ソースデータの Amazon S3 バケット名。<br>Amazon Redshift クラスタと同じリージョンで S3 バケットを使用します。 |
| 圧縮を有効にする | ファイルを Amazon Redshift に書き込む前に、ステージングファイルを圧縮します。<br>Secure Agent がステージングファイルを圧縮すると、タスクのパフォーマンスが向上します。<br>デフォルトではオンに設定されています。 |
| ステージングディレクトリの場所 | Amazon Redshift のステージングディレクトリ。<br>Secure Agent ランタイム環境でタスクを実行する場合は、ランタイム環境内の各 Secure Agent マシンで利用できるディレクトリパスを指定します。<br>Hosted Agent のランタイム環境でタスクを実行する場合は、ステージングディレクトリの場所を空のままにしてください。Hosted Agent は、一時的な場所にディレクトリを作成します。 |
| UnloadOptions プロパティファイル | Unload コマンドのオプション。<br>Amazon Redshift オブジェクトから S3 バケットにデータを書き込むためには、Unload コマンドにオプションを追加します。次のオプションを追加できます。<br>- DELIMITER<br>- PARALLEL<br>- ESCAPE<br>- AWS_IAM_ROLE<br>Secure Agent のランタイム環境でタスクを実行するときは、アンロードオプションを含むプロパティファイルのパスを指定するか、[UnloadOptions **プロパティファイル**] フィールドで直接アンロードオプションを指定します。<br>Hosted Agent ランタイム環境でタスクを実行するときは、[UnloadOptions **プロパティファイル**] フィールドで直接オプションを指定します。 |
| S3 クライアントサイドの暗号化の有効化 | Secure Agent がプライベート暗号化キーを使用してデータを暗号化することを示します。 |
| 暗号化タイプ | ソースの暗号化タイプを選択します。以下の暗号化タイプから選択できます。<br>- SSE-S3<br>- SSE-KMS<br>デフォルトは SSE-S3 です。詳細については、「Amazon Redshift ソースのデータ暗号化」 (ページ 133)を参照してください。 |
| S3 ファイルの分割ダウンロードの有効化 | 大きな Amazon S3 オブジェクトを複数パーツに分けてダウンロードします。<br>Amazon S3 オブジェクトのファイルサイズが 5MB を超える場合は、オブジェクトを複数パーツに分けて同時にダウンロードできます。 |
| パーツサイズ | オブジェクトのパーツサイズを指定します。デフォルトは 5MB です。 |
| Infa 詳細フィルタ | Amazon Redshift コネクタには適用されません。 |
| Pre-SQL | Amazon Redshift からデータを読み取る前にクエリを実行するための Pre-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| Post-SQL | Amazon Redshift にデータを書き込んだ後にクエリを実行するための Post-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |

| 詳細プロパティ | 説明 |
| --- | --- |
| SQL クエリ | デフォルトクエリーを上書きします。カラム名を二重引用符で囲みます。SQL クエリでは大文字と小文字が区別されます。Amazon Redshift データベースでサポートされている SQL 文を指定します。 |
| Number of Sorted Ports | クエリが行われたソースの行をソートする場合に使用されるカラム数。エージェントでは、ソース行を読み込む際にデフォルトのクエリに ORDER BY 句が追加されます。ORDER BY 句には、指定された数のポートが、トランスフォーメーションの最初のポートから順に取り込まれます。ソートするポート数を指定する場合、データベースのソート順は、セッションのソート順に一致しなければなりません。<br>デフォルトは 0 です。 |
| 個別選択 | 一意の値を選択します。このオプションを選択すると、エージェントに SELECT DISTINCT 文が含まれます。Amazon Redshift は末尾のスペースを無視します。したがって、予期されるより少ない行がエージェントで抽出される可能性があります。 |
| ソーステーブル名 | デフォルトのソーステーブル名を上書きできます。 |

# 同期タスクの Amazon Redshift ターゲット

同期タスクのターゲットとして Amazon Redshift オブジェクトを使用できます。

Amazon Redshift ターゲットオブジェクトを使用する場合は、標準オブジェクトをプライマリソースとして選択できます。

Amazon Redshift ターゲットプロパティは、同期タスクウィザードの［**ターゲット**］ページで設定します。

次の表に、Amazon Redshift ターゲットプロパティを示します。

| プロパティ | 説明 |
| --- | --- |
| 接続 | ターゲット接続の名前。 |
| ターゲットオブジェクト | ターゲットオブジェクトの名前。プライマリターゲットオブジェクトを選択します。 |
| ターゲットの作成 | ターゲットを作成します。<br>ターゲットオブジェクトの名前を入力して、使用するソースフィールドを選択します。デフォルト名はソースオブジェクトの名前です。また、デフォルトではすべてのソースフィールドが使用されます。必要に応じて、ターゲットオブジェクトのファイル拡張子を入力します。 |

Amazon Redshift ターゲットを使用するように同期タスクを設定するときに、詳細ターゲットプロパティを設定できます。

次の表に、Amazon Redshift の詳細ターゲットプロパティを示します。

| プロパティ | 説明 |
| --- | --- |
| S3 バケット名 | Amazon Redshift ターゲットデータの Amazon S3 バケット名。<br>Amazon Redshift クラスタと同じリージョンで S3 バケットを使用します。 |
| 圧縮を有効にする | ファイルを Amazon Redshift に書き込む前に、ステージングファイルを圧縮します。<br>Secure Agent でステージングファイルを圧縮するときに同期タスクのパフォーマンスが向上します。<br>デフォルトではオンに設定されています。 |
| ステージングディレクトリの場所 | Amazon Redshift のステージングディレクトリ。<br>Secure Agent ランタイム環境でタスクを実行する場合は、ランタイム環境内の各 Secure Agent マシンで利用できるディレクトリパスを指定します。<br>Hosted Agent のランタイム環境でタスクを実行する場合は、ステージングディレクトリを空のままにしてください。Hosted Agent は、一時的な場所にディレクトリを作成します。 |
| バッチサイズ | バッチ内の最小行数。0 より大きい値を入力します。<br>デフォルトは 2000000 です。 |
| INSERT 用のアップロードバッチごとの最大 Redshift エラー数 | バッチの失敗の原因となるバッチ内のエラーの数。正の整数値を入力します。<br>エラー数がプロパティ値以上になると、Secure Agent はバッチ全体をエラー行ファイルに書き込みます。<br>デフォルトは 1 です。 |
| データロード前にターゲットテーブルをトランケート | データをターゲットに書き込む前に、Amazon Redshift ターゲットをトランケートします。 |
| CHAR データ型および VARCHAR データ型の NULL 値 | Amazon Redshift ターゲットの CHAR および VARCHAR フィールドの NULL 値を表すために使用される文字列値（NULL、スペース文字など）。<br>デフォルトは空の文字列です。 |
| S3 でのファイル整合性のための待ち時間（秒） | Secure Agent でステージングファイルを使用できるようになるまでの待ち時間（秒）。<br>デフォルトは 5 です。 |
| CopyOptions プロパティファイル | Copy コマンドのオプション。<br>Amazon S3 バケットから Amazon Redshift ターゲットにデータを書き込むためには、Copy コマンドにオプションを追加します。次のオプションを追加できます。<br>- DELIMITER<br>- ACCEPTINVCHARS<br>- QUOTE<br>- COMPUPDATE<br>- AWS_IAM_ROLE<br>Secure Agent ランタイム環境でタスクを実行するときは、コピーオプションを含むプロパティのパスを指定するか、[CopyOptions **プロパティファイル**] フィールドで直接コピーオプションを指定します。<br>Hosted Agent ランタイム環境でタスクを実行するときは、[CopyOptions **プロパティファイル**] フィールドで直接オプションを指定する必要があります。 |

| プロパティ | 説明 |
| --- | --- |
| S3 サーバーサイドの暗号化の有効化 | Amazon S3 がアップロードおよびアクセス時にデータを暗号化および復号化することを示します。 |
| S3 クライアントサイドの暗号化の有効化 | Secure Agent がプライベート暗号化キーを使用してデータを暗号化することを示します。<br>サーバーサイドとクライアントサイドの両方で暗号化を有効にすると、ランタイム環境はサーバーサイドの暗号化を無視します。 |
| ターゲットテーブルのクリーンアップ | ディスク領域を回復し、指定したテーブルまたはデータベース内のすべてのテーブルの行をソートします。<br>次のリカバリオプションを選択できます。<br>- なし<br>- 全体<br>- ソートのみ<br>- 削除のみ<br>- 再インデックス<br>デフォルトは［なし］です。 |
| ターゲットテーブルの分析 | 読み取りおよび書き込み操作の効率が向上します。<br>Amazon Redshift のクエリプランナーは、統計メタデータを更新して、クエリ効率の向上に最適な計画を作成して選択します。 |
| S3 でステージングファイルを保持するためのプレフィックス | Amazon S3 でステージングファイルを保持します。<br>Amazon S3 でステージングファイルを保持するには、ディレクトリプレフィックスとファイルプレフィックスの両方をスラッシュ（/）で区切って指定するか、ファイルプレフィックスのみを指定します。例えば、`backup_dir/backup_file` や `backup_file` です。 |
| Pre-SQL | Amazon Redshift からデータを読み取る前にクエリを実行するための Pre-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| Post-SQL | Amazon Redshift にデータを書き込んだ後にクエリを実行するための Post-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| ターゲットテーブル名 | デフォルトのターゲットテーブル名を上書きできます。 |
| パーツサイズ | オブジェクトのパーツサイズを指定します。<br>デフォルトは 5MB です。 |
| TransferManager スレッドプールサイズ | データを並行して書き込むスレッドの数を指定します。<br>デフォルトは 10 です。 |
| バッチごとのファイル数 | バッチごとにターゲットステージングファイル数を計算するためのファイル数を指定します。<br>値を指定しない場合、ターゲットステージングファイル数は内部的に計算されます。<br>**注**: クラスタタイプと AmazonRedshift クラスタ内のノード数に基づいて最小値を指定します。エラーを回避するには、1500 未満の値を指定します。 |

| プロパティ | 説明 |
| --- | --- |
| 成功ファイルディレクトリ | Amazon Redshift の成功行ファイルのディレクトリ。各 Secure Agent マシンのランタイム環境で使用可能なディレクトリパスを指定します。デフォルトでは、データ統合は成功ファイルをディレクトリ<Secure Agent installation directory>/apps/Data_Integration_Server/data/success に書き込みます。<br>Hosted Agent は成功行ファイルを作成しません。Hosted Agent ランタイム環境でタスクを実行するときは［**成功ファイルディレクトリ**］フィールドを空白のままにします。 |
| エラーファイルのディレクトリ | エラー行ファイルのディレクトリ。各 Secure Agent マシンのランタイム環境で使用可能なディレクトリパスを指定します。デフォルトでは、データ統合はエラー行ファイルをディレクトリ<Secure Agent installation directory>/apps/Data_Integration_Server/data/error に書き込みます。<br>デフォルトのエラーファイルディレクトリを指定するとき、［スケジュール］タブからエラーファイルをダウンロードすることができます。<br>エラーファイルディレクトリがデフォルトのエラーディレクトリ以外の場合、［スケジュール］タブからエラーファイルをダウンロードすることはできません。エラーファイルを見付けるには、指定されたディレクトリに移動する必要があります。<br>Hosted Agent はエラー行ファイルを作成しません。Hosted Agent ランタイム環境でタスクを実行するときは［**エラーファイルディレクトリ**］フィールドを空白のままにします。 |

# 同期タスクの Amazon Redshift ルックアップ

同期タスクでフィールドマッピングを設定するときに、Amazon Redshift オブジェクトに対してキャッシュを使用しないルックアップを作成できます。接続プロパティで指定した JDBC URL を使用して、キャッシュを使用しないルックアップを作成します。

**注:** Amazon Redshift コネクタは接続されていないルックアップトランスフォーメーションをサポートしていません。

# 同期タスクのルールおよびガイドライン

Amazon Redshift のソースとターゲットに同期タスクを使用する場合は、次のルールとガイドラインを考慮してください。

- 同期タスクで複数の Amazon Redshift ソースオブジェクトを使用する場合、［**フィールドマッピング**］タブの［**フィールドの更新**］オプションにソースオブジェクトのメタデータが表示されません。タスクを保存して実行することはできますが、タスクは実行時に失敗します。そのため、単一のソースオブジェクトを使用して新しい同期タスクを作成するか、マッピングタスクを使用する必要があります。
- フィルタまたはクエリの WHERE 句で単一引用符を使用しないでください。

# 同期タスクの例

MySQL データベースで受注詳細を保存する電子商取引の組織に勤めているとします。組織は、MySQL データベースから Amazon Redshift ターゲットにデータを移動する必要があります。

Amazon Redshift に書き込む同期タスクを設定します。

次の同期タスクを実行します。

**同期タスクを定義します。**

挿入操作に使用する同期タスクを設定します。

MySQL **ソースオブジェクトを使用します。**

マッピングのソースは、受注詳細を含む MySQL 接続です。MySQL オブジェクトは同期タスク内の単一のソースです。顧客 ID、Item_codes、Item_quantity、および PRICE カラムを含めることができます。*sales_order_details* をソースオブジェクトのリソースとして指定します。

Amazon Redshift **ターゲットオブジェクトを作成します。**

ターゲットオブジェクトに挿入するソースオブジェクトからフィールド *[Customer_ID]*、*[Item_codes]*、*[Item_quantity]*、および *[Price]* を選択します。ターゲットオブジェクトの名前として *sales_order_details* を入力し、接続タイプとして MySQL を指定します。同期タスクは Amazon Redshift にデータを書き込みます。既存のターゲットオブジェクトも使用できます。

**フィールドマッピングを設定します。**

*sales_order_details* ソースデータにあるフィールドすべてをターゲット *sales_order_details* のフィールドすべてにマッピングします。同期アプリケーションは、マッピングされたソースデータを Amazon Redshift に書き込みます。

**詳細ターゲットプロパティを設定します。**

詳細ターゲットプロパティで、Amazon Redshift に固有のプロパティを選択します。Amazon Redshift ターゲットデータの Amazon S3 バケット名を指定します。Amazon Redshift クラスタと同じリージョンで S3 バケットを使用します。Copy コマンドにオプションを指定し、サーバー側の暗号化とクライアント側の暗号化を有効にすることもできます。**[保存]** および **[完了]** タスクをクリックします。

Amazon Redshift を開いてエクスポートされたデータを視覚化します。

**タスクをスケジュールします。**

要件ごとにタスクをスケジュールし、保存することができます。**[参照]** ページから同期タスクを選択して、タスクを実行します。**[監視]** で、タスクの実行後にログのステータスを監視できます。

# 第 18 章

# Amazon Redshift によるマッピングとマッピングタスク

マッピングを作成するときに、ソースまたはターゲットのトランスフォーメーションを Amazon Redshift V2 オブジェクトを表すように設定できます。

マッピングタスクを作成して、マッピングまたは統合テンプレートで定義されているデータフローロジックに基づきデータを処理します。Oracle CDC ソースから変更データをキャプチャして Amazon Redshift のターゲットテーブルに書き込むマッピングタスクを作成することもできます。マッピングを詳細モードに切り替えて、高度な機能を有効にするためのトランスフォーメーションと関数を含めることができます。

**注意:** ソース、ターゲット、およびルックアップトランスフォーメーションで設定された pre-SQL および post-SQL コマンドは、実行時には検証されません。安全でない可能性があるコンテンツがコマンド構文に含まれていないこと、および接続オブジェクトで設定したデータベースユーザーに適切な権限があることを確認してください。安全なネットワーク通信のために、TLS 対応のデータベース接続を使用することをお勧めします。

## マッピングの Amazon Redshift ソース

マッピングでは、ソーストランスフォーメーションを設定して単一の Amazon Redshift ソースまたは複数の Amazon Redshift ソースを表すことができます。

Amazon Redshift 標準オブジェクトをソースとして使用できます。標準オブジェクトをプライマリソースとして選択してから、1 つ以上の子オブジェクトを追加することができます。

以下の表に、ソーストランスフォーメーションで設定できる Amazon Redshift ソースプロパティを示します。

| プロパティ | 説明 |
|---|---|
| 接続 | ソース接続の名前。 |
| ソースタイプ | ソースオブジェクトのタイプ［単一オブジェクト］、［複数のオブジェクト］、［クエリ］、または［パラメータ］を選択します。 |
| オブジェクト | ソースオブジェクトの名前。ソースごとにソースオブジェクトを選択します。 |

以下の表に、ソーストランスフォーメーションで設定できる Amazon Redshift クエリオプションを示します。

| プロパティ | 説明 |
|---|---|
| Filter | 読み取り操作で値をフィルタ処理します。［**設定**］をクリックして、条件を追加してレコードをフィルタ処理し、Secure Agent がソースから読み取る行数を縮小します。<br>指定できるフィルタ条件を次に示します。<br>- **パラメータを使用しません。**基本フィルタを使用して、オブジェクト、フィールド、演算子、および値を指定して特定のレコードを選択します。<br>- **すべてパラメータを使用します。**パラメータを使用してフィールドマッピングを表現します。<br>- **詳細。**詳細フィルタを使用して、Amazon Redshift クエリ形式を使用するさらに複合的なフィルタ条件を定義します。 |
| ソート | 該当なし。 |

以下の表に、ソーストランスフォーメーションで設定できる Amazon Redshift ソースの詳細プロパティを示します。

| 詳細プロパティ | 説明 |
|---|---|
| S3 バケット名 | Amazon Redshift ターゲットデータの Amazon S3 バケット名。<br>Amazon Redshift クラスタと同じリージョンで S3 バケットを使用します。 |
| 圧縮を有効にする | ファイルを Amazon Redshift に書き込む前に、ステージングファイルを圧縮します。<br>ランタイム環境でステージングファイルを圧縮するときにタスクパフォーマンスが向上します。<br>デフォルトではオンに設定されています。 |
| ステージングディレクトリの場所 | Amazon Redshift のステージングディレクトリ。<br>Secure Agent ランタイム環境でタスクを実行する場合は、ランタイム環境内の各 Secure Agent マシンで利用できるディレクトリパスを指定します。<br>Hosted Agent のランタイム環境でタスクを実行する場合は、ステージングディレクトリを空のままにしてください。Hosted Agent は、一時的な場所にディレクトリを作成します。 |
| UnloadOptions プロパティファイル | Unload コマンドのオプション。<br>Amazon Redshift オブジェクトから S3 バケットにデータを書き込むためには、Unload コマンドにオプションを追加します。次のオプションを追加できます。<br>- DELIMITER<br>- PARALLEL<br>- ESCAPE<br>- AWS_IAM_ROLE<br>Secure Agent のランタイム環境でタスクを実行するときは、アンロードオプションを含むプロパティファイルのパスを指定するか、［UnloadOptions **プロパティファイル**］フィールドで直接アンロードオプションを指定します。<br>Hosted Agent ランタイム環境でタスクを実行するときは、［UnloadOptions **プロパティファイル**］フィールドで直接オプションを指定します。 |
| S3 クライアントサイドの暗号化の有効化 | Secure Agent がプライベート暗号化キーを使用してデータを暗号化することを示します。 |

| 詳細プロパティ | 説明 |
| --- | --- |
| 暗号化タイプ | ソースの暗号化タイプを選択します。以下の暗号化タイプから選択できます。<br>- SSE-S3<br>- SSE-KMS<br>デフォルトは SSE-S3 です。 |
| S3 ファイルの分割ダウンロードの有効化 | 大きな Amazon S3 オブジェクトを複数パーツに分けてダウンロードします。<br>Amazon S3 オブジェクトのファイルサイズが 5MB を超える場合は、オブジェクトを複数パーツに分けて同時にダウンロードできます。 |
| パーツサイズ | オブジェクトのパーツサイズを指定します。デフォルトは 5MB です。 |
| Infa 詳細フィルタ | Amazon Redshift コネクタには適用されません。 |
| Pre-SQL | Amazon Redshift からデータを読み取る前にクエリを実行するための Pre-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| Post-SQL | Amazon Redshift にデータを書き込んだ後にクエリを実行するための Post-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| SQL クエリ | デフォルトクエリーを上書きします。カラム名を二重引用符で囲みます。SQL クエリでは大文字と小文字が区別されます。Amazon Redshift データベースでサポートされている SQL 文を指定します。 |
| Number of Sorted Ports | クエリが行われたソースの行をソートする場合に使用されるカラム数。エージェントでは、ソース行を読み込む際にデフォルトのクエリに ORDER BY 句が追加されます。ORDER BY 句には、指定された数のポートが、トランスフォーメーションの最初のポートから順に取り込まれます。ソートするポート数を指定する場合、データベースのソート順は、セッションのソート順に一致しなければなりません。<br>デフォルトは 0 です。 |
| 個別選択 | 一意の値を選択します。このオプションを選択すると、エージェントに SELECT DISTINCT 文が含まれます。Amazon Redshift は末尾のスペースを無視します。したがって、予期されるより少ない行がエージェントで抽出される可能性があります。 |
| ソーステーブル名 | デフォルトのソーステーブル名を上書きできます。 |
| トレースレベル | ログファイルに表示される詳細情報の量を設定します。<br>[簡易]、[通常]、[詳細 - 初期化]、[詳細 - データ] から選択できます。デフォルトは [通常] です。 |

**注:** フィルタまたはクエリの WHERE 句で単一引用符を使用しないでください。

## キー範囲パーティション化の設定

フィールド値に基づいて Amazon Redshift データをパーティション化するようにキー範囲パーティションを設定します。

1. **[ソースのプロパティ]** で、**[パーティション]** タブをクリックします。
2. リストから必要な **[パーティションキー]** を選択します。

3. **［新しいキー範囲の追加］**をクリックしてパーティションを追加します。
4. **［開始範囲］**および**［終了範囲］**を指定します。

   **注:** パーティション化に指定するキー範囲には、単一引用符で囲まれた文字列値を含めることはできません。

# マッピングの Amazon Redshift ターゲット

マッピングでは、ターゲットトランスフォーメーションを設定して単一の Amazon Redshift ターゲットを表すことができます。入力フィールドに基づいて、実行時に Amazon Redshift ターゲットを作成することもできます。

Amazon Redshift ターゲットオブジェクトを使用するときに、標準オブジェクトをプライマリターゲットとして選択してから、子オブジェクトを追加します。カスタムオブジェクトを単一ターゲットとして使用できます。

以下の表に、ターゲットトランスフォーメーションで設定できる Amazon Redshift ターゲットプロパティを示します。

| プロパティ | 説明 |
|---|---|
| 接続 | ターゲット接続の名前。 |
| ターゲットタイプ | ターゲットオブジェクトのタイプ［単一オブジェクト］または［パラメータ］を選択します。 |
| オブジェクト | ターゲットオブジェクトの名前。単一ターゲットのターゲットオブジェクト。 |
| 操作 | ターゲット操作。［挿入］、［更新］、［更新/挿入］、または［削除］を選択します。 |
| ターゲットの作成 | ターゲットを作成します。<br>ターゲットオブジェクトの名前を入力して、使用するソースフィールドを選択します。デフォルト名はソースオブジェクトの名前です。また、デフォルトではすべてのソースフィールドが使用されます。必要に応じて、ターゲットオブジェクトのファイル拡張子を入力します。 |

次の表に、ターゲットトランスフォーメーションで設定できる Amazon Redshift ターゲット詳細プロパティを示します。

| プロパティ | 説明 |
|---|---|
| S3 バケット名 | Amazon Redshift ターゲットデータの Amazon S3 バケット名。<br>Amazon Redshift クラスタと同じリージョンで S3 バケットを使用します。 |
| 圧縮を有効にする | ファイルを Amazon Redshift に書き込む前に、ステージングファイルを圧縮します。<br>ランタイム環境でステージングファイルを圧縮するときにタスクパフォーマンスが向上します。<br>デフォルトではオンに設定されています。 |

| プロパティ | 説明 |
| --- | --- |
| ステージングディレクトリの場所 | Amazon Redshift のステージングディレクトリ。<br>Secure Agent ランタイム環境の場合、ランタイム環境内の各 Secure Agent マシンで利用できるディレクトリパスを指定します。<br>Hosted Agent ランタイム環境の場合、ステージングディレクトリは空白のままにします。Hosted Agent は、一時的な場所にディレクトリを作成します。 |
| バッチサイズ | バッチ内の最小行数。0 より大きい値を入力します。<br>デフォルトは 2000000 です。 |
| INSERT 用のアップロードバッチごとの最大 Redshift エラー数 | バッチの失敗の原因となるバッチ内のエラーの数。正の整数値を入力します。<br>エラー数がプロパティ値以上になると、ランタイム環境はバッチ全体をエラー行ファイルに書き込みます。<br>デフォルトは 1 です。 |
| データロード前にターゲットテーブルをトランケート | データをターゲットに書き込む前に、Amazon Redshift ターゲットをトランケートします。 |
| CHAR データ型および VARCHAR データ型の NULL 値 | Amazon Redshift ターゲットの CHAR および VARCHAR フィールドの NULL 値を表すために使用される文字列値（NULL、スペース文字など）。<br>デフォルトは空の文字列です。 |
| S3 でのファイル整合性のための待ち時間（秒） | ランタイム環境でステージングファイルを使用できるようになるまでの待ち時間（秒）。<br>デフォルトは 5 です。 |
| CopyOptions プロパティファイル | Copy コマンドのオプション。<br>Amazon S3 バケットから Amazon Redshift ターゲットにデータを書き込むためには、Copy コマンドにオプションを追加します。次のオプションを追加できます。<br>- DELIMITER<br>- ACCEPTINVCHARS<br>- QUOTE<br>- COMPUPDATE<br>- AWS_IAM_ROLE<br>Secure Agent ランタイム環境でタスクを実行するときは、コピーオプションを含むプロパティのパスを指定するか、[CopyOptions **プロパティファイル**] フィールドで直接コピーオプションを指定します。<br>Hosted Agent ランタイム環境でタスクを実行するときは、[CopyOptions **プロパティファイル**] フィールドで直接オプションを指定する必要があります。 |
| S3 サーバーサイドの暗号化の有効化 | Amazon S3 がアップロードおよびアクセス時にデータを暗号化および復号化することを示します。 |
| S3 クライアントサイドの暗号化の有効化 | ランタイム環境がプライベート暗号化キーを使用してデータを暗号化することを示します。<br>サーバーサイドとクライアントサイドの両方で暗号化を有効にすると、ランタイム環境はサーバーサイドの暗号化を無視します。 |

| プロパティ | 説明 |
| --- | --- |
| ターゲットテーブルのクリーンアップ | ディスク領域を回復し、指定したテーブルまたはデータベース内のすべてのテーブルの行をソートします。<br>次のリカバリオプションを選択できます。<br>- なし<br>- 全体<br>- ソートのみ<br>- 削除のみ<br>- 再インデックス<br>デフォルトは［なし］です。 |
| S3 でステージングファイルを保持するためのプレフィックス | Amazon S3 でステージングファイルを保持します。<br>Amazon S3 でステージングファイルを保持するには、ディレクトリプレフィックスとファイルプレフィックスの両方をスラッシュ（/）で区切って指定するか、ファイルプレフィックスのみを指定します。例えば、`backup_dir/backup_file` や `backup_file` です。 |
| ターゲットテーブルの分析 | 読み取りおよび書き込み操作の効率が向上します。<br>Amazon Redshift のクエリプランナーは、統計メタデータを更新して、クエリ効率の向上に最適な計画を作成して選択します。 |
| Pre-SQL | Amazon Redshift からデータを読み取る前にクエリを実行するための Pre-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| Post-SQL | Amazon Redshift にデータを書き込んだ後にクエリを実行するための Post-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| ターゲットテーブル名 | デフォルトのターゲットテーブル名を上書きできます。 |
| パーツサイズ | オブジェクトのパーツサイズを指定します。<br>デフォルトは 5MB です。 |
| TransferManager スレッドプールサイズ | データを並行して書き込むスレッドの数を指定します。<br>デフォルトは 10 です。 |
| バッチごとのファイル数 | バッチごとにターゲットステージングファイル数を計算するためのファイル数を指定します。<br>値を指定しない場合、ターゲットステージングファイル数は内部的に計算されます。<br>**注**: クラスタタイプと Amazon Redshift クラスタ内のノード数に基づいて最小値を指定します。エラーを回避するには、1500 未満の値を指定します。 |
| 成功ファイルディレクトリ | Amazon Redshift の成功行ファイルのディレクトリ。各 Secure Agent マシンのランタイム環境で使用可能なディレクトリパスを指定します。デフォルトでは、データ統合は成功ファイルをディレクトリ`<Secure Agent installation directory>/apps/Data_Integration_Server/data/success` に書き込みます。<br>Hosted Agent は成功行ファイルを作成しません。Hosted Agent ランタイム環境でタスクを実行するときは［**成功ファイルディレクトリ**］フィールドを空白のままにします。 |

| プロパティ | 説明 |
| --- | --- |
| エラーファイルのディレクトリ | Amazon Redshift のエラー行ファイルのディレクトリ。<br>エラー行ファイルのディレクトリ。各 Secure Agent マシンのランタイム環境で使用可能なディレクトリパスを指定します。デフォルトでは、データ統合はエラー行ファイルをディレクトリ`<Secure Agent installation directory>/apps/Data_Integration_Server/data/error` に書き込みます。<br>デフォルトのエラーファイルディレクトリを指定するとき、[スケジュール] タブからエラーファイルをダウンロードすることができます。<br>エラーファイルディレクトリがデフォルトのエラーディレクトリ以外の場合、[スケジュール] タブからエラーファイルをダウンロードすることはできません。エラーファイルを見付けるには、指定されたディレクトリに移動する必要があります。<br>Hosted Agent はエラー行ファイルを作成しません。Hosted Agent ランタイム環境でタスクを実行するときは **[エラーファイルディレクトリ]** フィールドを空白のままにします。 |
| 拒否された行を転送 | トランスフォーメーションが、拒否された行を次のトランスフォーメーションに渡すか、拒否された行を削除するかを決定します。デフォルトでは、マッピングアプリケーションは拒否された行を次のトランスフォーメーションに転送します。 |

ターゲットタスクを編集するときに異なる Amazon Redshift 接続を選択すると、詳細ターゲットプロパティがクリアされます。選択された Amazon Redshift 接続に適用可能な S3 バケット名とその他の詳細プロパティを入力します。

# マッピングの Amazon Redshift ルックアップ

マッピングでは、ルックアップトランスフォーメーションを Amazon Redshift オブジェクトを表すように設定できます。

Amazon Redshift オブジェクトをルックアップとして使用するときに、Amazon Redshift プロパティに Amazon S3 バケット名を設定する必要があります。

Amazon Redshift 接続でキャッシュを使用するルックアップを使用するときに、ルックアップ条件に NULL 値を含まれる場合は、ルックアップ条件が無視されます。

**注:** Amazon Redshift コネクタは接続されていないルックアップトランスフォーメーションをサポートしていません。

# テンプレートベースのマッピングタスクの Amazon Redshift オブジェクト

統合テンプレートに基づいてマッピングタスクを設定する場合は、Amazon Redshift のソースとターゲットに詳細プロパティを設定できます。

## マッピングタスクの Amazon Redshift ソース

テンプレートベースのマッピングタスクで使用する Amazon Redshift ソース接続の場合、マッピングタスクウィザードの［**ソース**］ページで詳細プロパティを設定できます。

以下の詳細プロパティを設定できます。

| 詳細プロパティ | 説明 |
|---|---|
| S3 バケット名 | Amazon Redshift ターゲットデータの Amazon S3 バケット名。<br>Amazon Redshift クラスタと同じリージョンで S3 バケットを使用します。 |
| 圧縮を有効にする | ファイルを Amazon Redshift に書き込む前に、ステージングファイルを圧縮します。<br>ランタイム環境でステージングファイルを圧縮するときにタスクパフォーマンスが向上します。<br>デフォルトではオンに設定されています。 |
| ステージングディレクトリの場所 | Amazon Redshift のステージングディレクトリ。<br>Secure Agent ランタイム環境でタスクを実行する場合は、ランタイム環境内の各 Secure Agent マシンで利用できるディレクトリパスを指定します。<br>Hosted Agent のランタイム環境でタスクを実行する場合は、ステージングディレクトリを空のままにしてください。Hosted Agent は、一時的な場所にディレクトリを作成します。 |
| UnloadOptions プロパティファイル | Unload コマンドのオプション。<br>Amazon Redshift オブジェクトから S3 バケットにデータを書き込むためには、Unload コマンドにオプションを追加します。次のオプションを追加できます。<br>- DELIMITER<br>- PARALLEL<br>- ESCAPE<br>- AWS_IAM_ROLE<br>Secure Agent のランタイム環境でタスクを実行するときは、アンロードオプションを含むプロパティファイルのパスを指定するか、［UnloadOptions **プロパティファイル**］フィールドで直接アンロードオプションを指定します。<br>Hosted Agent ランタイム環境でタスクを実行するときは、［UnloadOptions **プロパティファイル**］フィールドで直接オプションを指定します。 |
| S3 クライアントサイドの暗号化の有効化 | Secure Agent がプライベート暗号化キーを使用してデータを暗号化することを示します。 |
| 暗号化タイプ | ソースの暗号化タイプを選択します。以下の暗号化タイプから選択できます。<br>- SSE-S3<br>- SSE-KMS<br>デフォルトは、SSE-S3 です。詳細については、「Amazon Redshift ソースのデータ暗号化」（ページ 133)を参照してください。 |

| 詳細プロパティ | 説明 |
| --- | --- |
| S3 ファイルの分割ダウンロードの有効化 | 大きな Amazon S3 オブジェクトを複数パーツに分けてダウンロードします。<br>Amazon S3 オブジェクトのファイルサイズが 5MB を超える場合は、オブジェクトを複数パーツに分けて同時にダウンロードできます。 |
| パーツサイズ | オブジェクトのパーツサイズを指定します。デフォルトは 5MB です。 |
| Infa 詳細フィルタ | Amazon Redshift コネクタには適用されません。 |
| Pre-SQL | Amazon Redshift からデータを読み取る前にクエリを実行するための Pre-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| Post-SQL | Amazon Redshift にデータを書き込んだ後にクエリを実行するための Post-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| SQL クエリ | デフォルトクエリーを上書きします。カラム名を二重引用符で囲みます。SQL クエリでは大文字と小文字が区別されます。Amazon Redshift データベースでサポートされている SQL 文を指定します。 |
| Number of Sorted Ports | クエリが行われたソースの行をソートする場合に使用されるカラム数。エージェントでは、ソース行を読み込む際にデフォルトのクエリに ORDER BY 句が追加されます。ORDER BY 句には、指定された数のポートが、トランスフォーメーションの最初のポートから順に取り込まれます。ソートするポート数を指定する場合、データベースのソート順は、セッションのソート順に一致しなければなりません。<br>デフォルトは 0 です。 |
| 個別選択 | 一意の値を選択します。このオプションを選択すると、エージェントに SELECT DISTINCT 文が含まれます。Amazon Redshift は末尾のスペースを無視します。したがって、予期されるより少ない行がエージェントで抽出される可能性があります。 |
| ソーステーブル名 | デフォルトのソーステーブル名を上書きできます。 |
| トレースレベル | ログファイルに表示される詳細情報の量を設定します。<br>［簡易］、［通常］、［詳細 - 初期化］、［詳細 - データ］から選択できます。デフォルトは［通常］です。 |

## マッピングタスクの Amazon Redshift ターゲット

テンプレートベースのマッピングタスクで使用する Amazon Redshift ターゲット接続の場合、マッピングタスクウィザードの **［ターゲット］** ページで詳細プロパティを設定できます。

以下の詳細プロパティを設定できます。

| プロパティ | 説明 |
| --- | --- |
| S3 バケット名 | Amazon Redshift ターゲットデータの Amazon S3 バケット名。<br>Amazon Redshift クラスタと同じリージョンで S3 バケットを使用します。 |
| 圧縮を有効にする | ファイルを Amazon Redshift に書き込む前に、ステージングファイルを圧縮します。<br>ランタイム環境でステージングファイルを圧縮するときにタスクパフォーマンスが向上します。<br>デフォルトではオンに設定されています。 |
| ステージングディレクトリの場所 | Amazon Redshift のステージングディレクトリ。<br>ランタイム環境をホストするマシン上のディレクトリを指定します。 |
| バッチサイズ | バッチ内の最小行数。0 より大きい値を入力します。<br>デフォルトは 2000000 です。 |
| INSERT 用のアップロードバッチごとの最大 Redshift エラー数 | バッチの失敗の原因となるバッチ内のエラーの数。正の整数値を入力します。<br>エラー数がプロパティ値以上になると、ランタイム環境はバッチ全体をエラー行ファイルに書き込みます。<br>デフォルトは 1 です。 |
| データロード前にターゲットテーブルをトランケート | データをターゲットに書き込む前に、Amazon Redshift ターゲットをトランケートします。 |
| CHAR データ型および VARCHAR データ型の NULL 値 | Amazon Redshift ターゲットの CHAR および VARCHAR フィールドの NULL 値を表すために使用される文字列値（NULL、スペース文字など）。<br>デフォルトは空の文字列です。 |
| S3 でのファイル整合性のための待ち時間（秒） | ランタイム環境でステージングファイルを使用できるようになるまでの待ち時間（秒）。<br>デフォルトは 5 です。 |
| CopyOptions プロパティファイル | プロパティファイルのパス。<br>Amazon S3 から Amazon Redshift ターゲットにデータを書き込むために Copy コマンドにオプションを追加できるようになります。次のオプションを追加できます。<br>- DELIMITER<br>- ACCEPTINVCHARS<br>- QUOTE<br>- COMPUPDATE<br>Secure Agent ランタイム環境でタスクを実行するときは、コピーオプションを含むプロパティのパスを指定するか、[CopyOptions **プロパティファイル**] フィールドで直接コピーオプションを指定します。<br>Hosted Agent ランタイム環境でタスクを実行するときは、[CopyOptions **プロパティファイル**] フィールドで直接オプションを指定する必要があります。 |
| S3 サーバーサイドの暗号化の有効化 | Amazon S3 がアップロードおよびアクセス時にデータを暗号化および復号化することを示します。 |

| プロパティ | 説明 |
| --- | --- |
| S3 クライアントサイドの暗号化の有効化 | ランタイム環境がプライベート暗号化キーを使用してデータを暗号化することを示します。<br>サーバーサイドとクライアントサイドの両方で暗号化を有効にすると、ランタイム環境はサーバーサイドの暗号化を無視します。 |
| ターゲットテーブルのクリーンアップ | ディスク領域を回復し、指定したテーブルまたはデータベース内のすべてのテーブルの行をソートします。<br>次のリカバリオプションを選択できます。<br>- なし<br>- 全体<br>- ソートのみ<br>- 削除のみ<br>- 再インデックス<br>デフォルトは［なし］です。 |
| ターゲットテーブルの分析 | 読み取りおよび書き込み操作の効率が向上します。<br>Amazon Redshift のクエリプランナーは、統計メタデータを更新して、クエリ効率の向上に最適な計画を作成して選択します。 |
| S3 でステージングファイルを保持するためのプレフィックス | Amazon S3 でステージングファイルを保持します。<br>Amazon S3 でステージングファイルを保持するには、ディレクトリプレフィックスとファイルプレフィックスの両方をスラッシュ（/）で区切って指定するか、ファイルプレフィックスのみを指定します。例えば、`backup_dir/backup_file` や `backup_file` です。 |
| Pre-SQL | Amazon Redshift からデータを読み取る前にクエリを実行するための Pre-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| Post-SQL | Amazon Redshift にデータを書き込んだ後にクエリを実行するための Post-SQL コマンド。UNLOAD コマンドまたは COPY コマンドを使用することもできます。ここで指定するコマンドは、プレーンテキストとして処理されます。 |
| ターゲットテーブル名 | デフォルトのターゲットテーブル名を上書きできます。 |
| パーツサイズ | オブジェクトのパーツサイズを指定します。<br>デフォルトは 5MB です。 |
| TransferManager スレッドプールサイズ | データを並行して書き込むスレッドの数を指定します。<br>デフォルトは 10 です。 |
| バッチごとのファイル数 | バッチごとにターゲットステージングファイル数を計算するためのファイル数を指定します。<br>値を指定しない場合、ターゲットステージングファイル数は内部的に計算されます。<br>**注**: クラスタタイプと AmazonRedshift クラスタ内のノード数に基づいて最小値を指定します。大きい値を指定しないでください。そうしない場合、タスクは `OutOfMemoryError` エラーメッセージを表示して失敗します。 |

| プロパティ | 説明 |
| --- | --- |
| 成功ファイルディレクトリ | Amazon Redshift の成功行ファイルのディレクトリ。各 Secure Agent マシンのランタイム環境で使用可能なディレクトリパスを指定します。デフォルトでは、データ統合は成功ファイルをディレクトリ<Secure Agent installation directory>/apps/Data_Integration_Server/data/success に書き込みます。<br>Hosted Agent は成功行ファイルを作成しません。Hosted Agent ランタイム環境でタスクを実行するときは［**成功ファイルディレクトリ**］フィールドを空白のままにします。 |
| エラーファイルのディレクトリ | Amazon Redshift のエラー行ファイルのディレクトリ。<br>エラー行ファイルのディレクトリ。各 Secure Agent マシンのランタイム環境で使用可能なディレクトリパスを指定します。デフォルトでは、データ統合はエラー行ファイルをディレクトリ<Secure Agent installation directory>/apps/Data_Integration_Server/data/error に書き込みます。<br>デフォルトのエラーファイルディレクトリを指定するとき、［スケジュール］タブからエラーファイルをダウンロードすることができます。<br>エラーファイルディレクトリがデフォルトのエラーディレクトリ以外の場合、［スケジュール］タブからエラーファイルをダウンロードすることはできません。エラーファイルを見付けるには、指定されたディレクトリに移動する必要があります。<br>Hosted Agent はエラー行ファイルを作成しません。Hosted Agent ランタイム環境でタスクを実行するときは［**エラーファイルディレクトリ**］フィールドを空白のままにします。 |

# 第 19 章

# Amazon Redshift のプッシュダウンの最適化

プッシュダウンの最適化を使用し、ソースデータベースまたはターゲットデータベースにトランスフォーメーションロジックをプッシュできます。プッシュダウンの最適化は、データベースのリソースを使用することによってタスクのパフォーマンスを向上させることが可能な場合に使用します。

プッシュダウンの最適化用に設定されたタスクを実行すると、トランスフォーメーションロジックが SQL クエリに変換されます。タスクでクエリがデータベースに送信され、データベースでクエリが実行されます。

Amazon Redshift コネクタは、マッピングに Amazon ODBC Redshift ドライバを使用する ODBC 接続タイプに対して、**完全**および**ソース**のプッシュダウンの最適化をサポートしています。

**注:** 完全なプッシュダウンの最適化が有効になっているマッピングに更新/挿入操作を設定することはできません。詳細モードのマッピングには、プッシュダウン最適化は設定できません。

**例**

急成長を遂げているデータサイエンス組織で働いているとします。この組織は、財務分析を行うソフトウェア製品を開発して、個人のプロフィール、企業、職務、広告主、および出版社をつなぐ財務グラフを作成しています。組織では、Amazon Web サービスに基づくインフラストラクチャを使用して、ペタバイト規模のデータウェアハウスである Amazon Redshift にデータを格納しています。組織では、可視化を作成してリアルタイム分析を実現するためにビジネスインテリジェンスサービスを実装する予定です。そのため、Amazon Redshift に格納されている膨大な量のデータをビジネスインテリジェンスサービスに移植する必要があります。Amazon Redshift コネクタを使用して、Amazon Redshift からデータを読み取ることができます。この大量のデータを読み取るために、ODBC 接続タイプとして、ソースのプッシュダウンを使用できます。プッシュダウンの最適化による ODBC 接続タイプを使用すると、パフォーマンスが向上します。

## Amazon Redshift マッピングでサポートされる関数および演算子

次の表に、Amazon Redshift データベースでのプッシュダウン関数の可用性を示します。X マークの付いたカラムは、ソース側または完全なプッシュダウンの最適化を使用して、関数を Amazon Redshift データベースにプッシュできることを示しています。S マークの付いたカラムは、ソース側のプッシュダウン最適化を使用し

てのみ、関数を Amazon Redshift データベースにプッシュできることを示しています。ダッシュ（-）記号の付いたカラムは、関数をデータベースにプッシュできないことを示しています。

| 関数 | プッシュダウン | 関数 | プッシュダウン | 関数 | プッシュダウン |
|---|---|---|---|---|---|
| ABORT() | - | INSTR() | X | REG_REPLACE | - |
| ABS() | X | IS_DATE() | - | REPLACECHR() | - |
| ADD_TO_DATE() | X | IS_NUMBER() | - | REPLACESTR() | - |
| AES_DECRYPT() | - | IS_SPACES() | - | REVERSE() | - |
| AES_ENCRYPT() | - | ISNULL() | S | ROUND(DATE) | - |
| ASCII() | - | LAST() | - | ROUND(NUMBER) | X |
| AVG() | S | LAST_DAY() | X | RPAD() | X |
| CEIL() | X | LEAST() | - | RTRIM() | X |
| CHOOSE() | - | LENGTH() | X | SET_DATE_PART() | - |
| CHRCODE() | - | LN() | X | SIGN() | X |
| COMPRESS() | - | LOG() | - | SIN() | X |
| CONCAT() | X | LOOKUP | - | SINH() | - |
| COS() | X | LOWER() | X | SOUNDEX() | - |
| COSH() | - | LPAD() | X | SQRT() | X |
| COUNT() | S | LTRIM() | X | STDDEV() | S |
| CRC32() | - | MAKE_DATE_TIME() | - | SUBSTR() | X |
| CUME() | - | MAX() | S | SUM() | S |
| DATE_COMPARE() | X | MD5() | X | SYSTIMESTAMP() | S |
| DATE_DIFF() | X | MEDIAN() | - | TAN() | S |
| DECODE() | X | METAPHONE() | - | TANH() | - |
| DECODE_BASE64() | - | MIN() | S | TO_BIGINT | X |
| DECOMPRESS() | - | MOD() | S | TO_CHAR(DATE) | S |
| ENCODE_BASE64() | - | MOVINGAVG() | - | TO_CHAR(NUMBER) | X |
| EXP() | X | MOVINGSUM() | - | TO_DATE() | X |
| FIRST() | - | NPER() | - | TO_DECIMAL() | X |

| 関数 | プッシュダウン | 関数 | プッシュダウン | 関数 | プッシュダウン |
|---|---|---|---|---|---|
| FLOOR() | X | PERCENTILE() | - | TO_FLOAT() | X |
| FV() | - | PMT() | - | TO_INTEGER() | X |
| GET_DATE_PART() | X | POWER() | X | TRUNC(DATE) | S |
| GREATEST() | - | PV() | - | TRUNC(NUMBER) | S |
| IIF() | X | RAND() | - | UPPER() | X |
| IN() | S | RATE() | - | VARIANCE() | S |
| INDEXOF() | - | REG_EXTRACT() | - | | |
| INITCAP() | X | REG_MATCH() | - | | |

次の表に、Amazon Redshift データベースで使用できるプッシュダウン演算子を示します。X マークの付いたカラムは、ソース側、ターゲット側、または完全なプッシュダウンの最適化を使用して、演算子を Amazon Redshift データベースにプッシュできることを示しています。

| 演算子 | プッシュダウン |
|---|---|
| + | X |
| - | X |
| * | X |
| / | X |
| % | X |
| \|\| | X |
| > | X |
| = | X |
| >= | X |
| <= | X |
| ! = | X |
| AND | X |
| OR | X |
| NOT | X |
| ^= | X |

# Amazon Redshift ODBC 接続の設定

Amazon Redshift は、Windows システムおよび Linux システムで Amazon ODBC Redshift ドライバをサポートしています。システム要件に基づいて Amazon ODBC Redshift 64 ビットドライバをインストールする必要があります。

**注:** Informatica では、プッシュダウンの最適化に使用するドライバとして、Amazon Redshift ODBC ドライババージョン `AmazonRedshiftODBC-64-bit-1.4.8.1000-1.x86_64` を認定しています。

## Windows 上の Amazon Redshift ODBC 接続の設定

Windows 上の Amazon Redshift に接続するように ODBC 接続を確立するには、ODBC 接続を設定する必要があります。

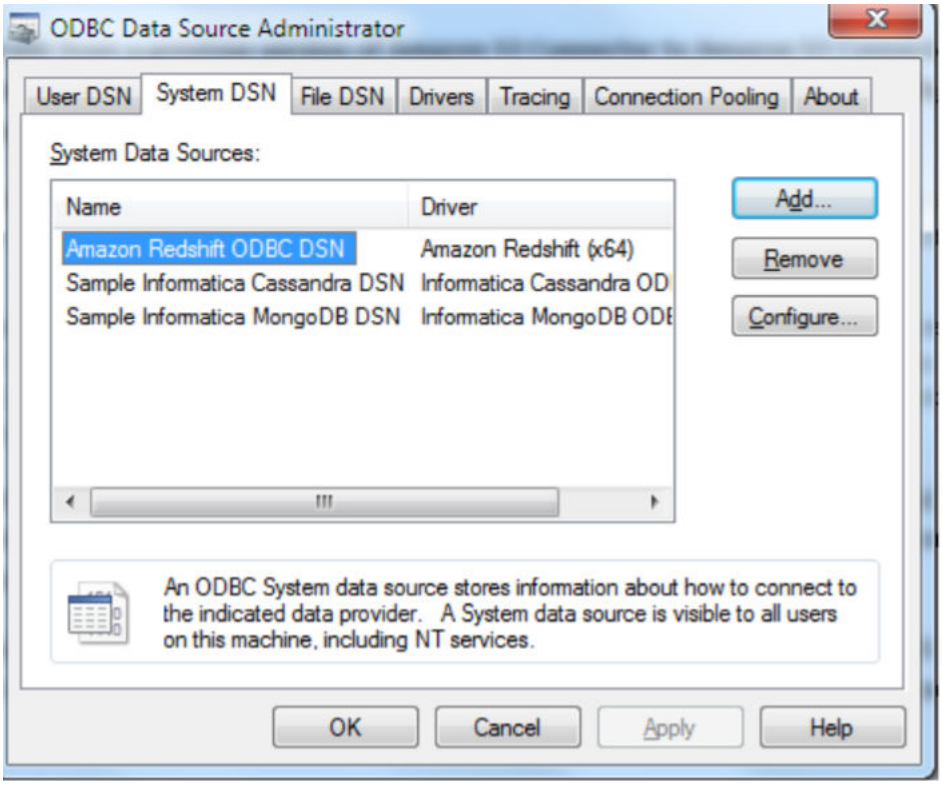
Windows 上で ODBC 接続を設定するには、次の手順を実行します。

1. AWS サイトから Amazon Redshift ODBC ドライバをダウンロードします。

   Amazon Redshift ODBC 64 ビットドライバをダウンロードする必要があります。
2. Secure Agent がインストールされているマシンに Amazon Redshift ODBC ドライバをインストールします。
3. ODBC データソースファイルのインストール先のフォルダを開きます。
4. `odbcad32.exe` ファイルを実行します。

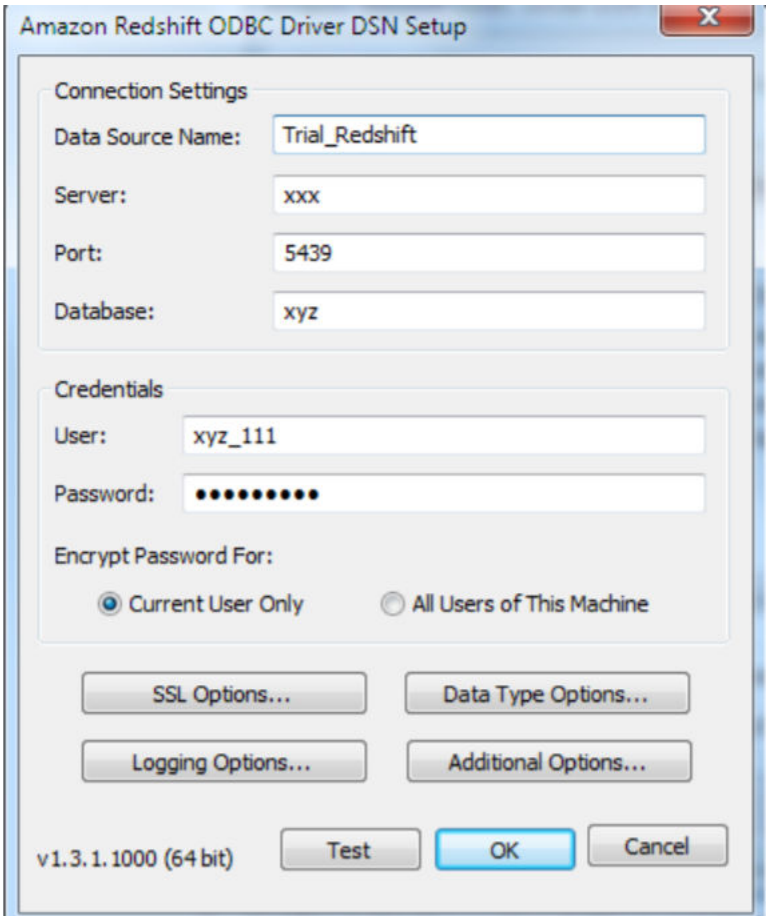
   **[ODBC データソースアドミニストレータ]** ダイアログボックスが表示されます。
5. **[システム DSN]** をクリックします。

   **[システム DSN]** タブが表示されます。次の図は、**[ODBC データソースアドミニストレータ]** ダイアログボックスの **[システム DSN]** タブを示しています。



6. **[設定]** をクリックします。

   **[Amazon Redshift ODBC Driver DSN セットアップ]**ダイアログボックスが表示されます。次の図は、**[Amazon Redshift ODBC Driver DSN セットアップ]**ダイアログボックスを示していて、ここで**[接続設定]**および**[資格情報]**を設定できます。




7. **[接続設定]**セクションで次の接続プロパティを指定します。

| プロパティ | 説明 |
|---|---|
| データソース名 | データソースの名前。 |
| サーバー | Amazon Redshift サーバーの場所。 |
| ポート | Amazon Redshift サーバーのポート番号。 |
| データベース | Amazon Redshift データベースの名前。 |

**注:** JDBC URL から得た**[サーバー]**、**[ポート]**、および**[データベース]**の値を指定する必要があります。

8. **[資格情報]**セクションで次の資格情報を指定します。

| プロパティ | 説明 |
|---|---|
| ユーザー | Amazon Redshift データベースにアクセスするためのユーザー名。 |
| パスワード | Amazon Redshift データベースのパスワード。 |
| パスワードの暗号化 | 次のユーザーのパスワードを暗号化します。<br>- **カレントユーザのみ**<br>- **このマシンのすべてのユーザー**<br>デフォルトは[**カレントユーザのみ**]です。 |

9. **[Amazon Redshift ODBC Driver DSN セットアップ]**で、**[テスト]**をクリックして接続をテストします。
10. **[OK]** をクリックします。

Amazon Redshift ODBC 接続が Windows 上で正常に設定されました。

Amazon Redshift ODBC 接続を設定した後で、Amazon Redshift に接続するための ODBC 接続を作成する必要があります。

Amazon Redshift に接続するための ODBC 接続を作成する方法については、「ODBC 接続の作成」 (ページ 108)を参照してください。

# Linux 上の Amazon Redshift ODBC 接続の設定

Linux 上の Amazon Redshift に接続するように ODBC 接続を確立するには、ODBC 接続を設定する必要があります。

Linux 上で ODBC 接続を設定するには、次の手順を実行します。

1. AWS サイトから Amazon Redshift ODBC ドライバをダウンロードします。

   Amazon Redshift ODBC 64 ビットドライバをダウンロードする必要があります。
2. Secure Agent がインストールされているマシンに Amazon Redshift ODBC ドライバをインストールします。
3. `odbc.ini` ファイルのプロパティを次の形式で設定します。

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file

Host=cluster_endpoint
Port=port_number
Database=database_name
```

4. `odbc.ini` ファイルに次のプロパティを指定します。

| プロパティ | 説明 |
|---|---|
| ODBC Data Sources | データソースの名前。 |
| Driver | Amazon Redshift ODBC ドライバファイルの場所。 |

| プロパティ | 説明 |
| --- | --- |
| Host | Amazon Redshift ホストの場所。 |
| Port | Amazon Redshift サーバーのポート番号。 |
| Database | Amazon Redshift データベースの名前。 |

**注:** JDBC URL から得た **Host**、**Port**、および **Database** にの値を指定する必要があります。

5. オペレーティングシステムに次の環境変数を設定します。

| 変数 | 説明 |
| --- | --- |
| LD_LIBRARY_PATH | Amazon Redshift ODBC ドライバがインストールされているディレクトリ。 |
| ODBCINI | odbc.ini ファイルが含まれるディレクトリ。 |
| ODBCHROME | ODBC のインストールディレクトリ。 |

6. 次のコマンドを実行して odbc.ini ファイルをエクスポートします。

```
Export ODBCINI=/<odbc.ini file path>/odbc.ini
```

7. Secure Agent を再起動します。

   Linux 上の Amazon Redshift ODBC 接続が正常に設定されました。

Amazon Redshift ODBC 接続を設定した後で、Amazon Redshift に接続するための ODBC 接続を作成する必要があります。
Amazon Redshift に接続するための ODBC 接続を作成する方法については、「ODBC 接続の作成」 (ページ 108)を参照してください。

# ODBC 接続の作成

ODBC 接続を設定した後で、Amazon Redshift に接続するための ODBC 接続を作成する必要があります。

次の手順を実行して、**[接続]** ページで Amazon Redshift ODBC 接続を作成します。

1. 管理者で、**[接続]** をクリックします。
2. 右上隅の **[新しい接続]** をクリックします。

［**新しい接続**］ページが表示されます。次の図は、［**新しい接続**］ページを示しています。

3. ［**接続の詳細**］セクションで、次の接続の詳細を設定します。

| プロパティ | 説明 |
|---|---|
| 接続名 | ODBC 接続の名前。 |
| 説明 | 接続の説明。 |
| タイプ | 接続の種類。<br>接続のタイプに［ODBC］を選択します。 |

4. ［**接続のプロパティ**］セクションで、次の接続の詳細を設定します。

| プロパティ | 説明 |
|---|---|
| ランタイム環境 | タスクを実行するランタイム環境の名前。 |
| ユーザー名 | Amazon Redshift アカウントのユーザー名。 |
| パスワード | Amazon Redshift アカウントのパスワード。 |
| データソース名 | Amazon Redshift データベース用に作成した ODBC データソースの名前を入力します。 |
| スキーマ | Amazon Redshift スキーマ名。 |
| コードページ | Secure Agent がデータの読み取りまたは書き込みに使用する必要があるコードページを選択します。 |
| ODBC サブタイプ | ［ODBC **サブタイプ**］フィールドの値を［Redshift］と入力します。 |

Amazon Redshift ODBC 接続が正常に作成されました。

# クロススキーマプッシュダウンの最適化

Amazon Redshift ODBC 接続を使用して同じデータベース内の異なるスキーマの Amazon Redshift オブジェクトへのデータの読み取りまたは書き込みを行うマッピングタスクに対して、クロススキーマプッシュダウンの最適化を設定できます。

クロススキーマプッシュダウンの最適化を使用するには、Amazon Redshift ODBC 接続を作成して、ソース接続およびターゲット接続のスキーマを指定します。ソーススキーマおよびターゲットスキーマは異なる必要がありますが、どちらも同じデータベースに属している必要があります。マッピングタスクにプッシュダウンの最適化を設定して、詳細セッションプロパティでクロススキーマプッシュダウンの最適化を有効にします。デフォルトでは、**[クロススキーマプッシュダウンの最適化の有効化]**チェックボックスが選択されています。

## Amazon Redshift マッピングタスクに対するクロススキーマの最適化の設定

次の手順を実行して、Amazon Redshift マッピングタスクに対するクロススキーマプッシュダウンの最適化を設定します。

1. Amazon Redshift ODBC のソース接続およびターゲット接続を、それぞれ異なるスキーマで定義して作成します。

   以下に例を示します。

   - `rs_odbc1` Amazon Redshift ODBC 接続を作成して、接続プロパティで `CQA_SCHEMA1` スキーマを指定します。
   - `rs_odbc2` Amazon Redshift ODBC 接続を作成して、接続プロパティで `CQA_SCHEMA2` スキーマを指定します。

2. Amazon Redshift マッピングを作成します。

   例えば、`m_rs_pdo_crossSchema` Amazon Redshift マッピングを作成します。

3. ソーストランスフォーメーションを追加します。Amazon Redshift ソースオブジェクト、および接続に指定されたスキーマを使用してデータを読み取るための接続を含めます。

   例えば、ソーストランスフォーメーションを追加します。Amazon Redshift ソースオブジェクト、および `CQA_SCHEMA1` を使用してデータを読み取るための `rs_odbc1` 接続を含めます。

4. ターゲットトランスフォーメーションを追加します。Amazon Redshift ターゲットオブジェクト、および接続に指定されたスキーマを使用してデータを書き込むための接続を含めます。

   例えば、ターゲットトランスフォーメーションを追加します。Amazon Redshift ターゲットオブジェクト、および `CQA_SCHEMA2` を使用してデータを書き込むための `rs_odbc2` 接続を含めます。

5. Amazon Redshift マッピングタスクを作成して、次のタスクを実行します。

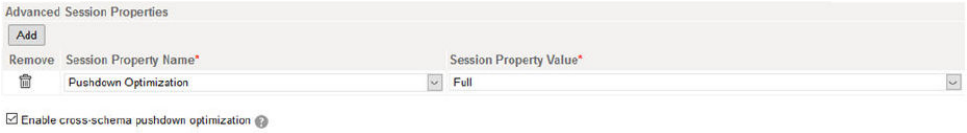
   a. 設定された Amazon Redshift マッピングを選択します。

      例えば、`m_rs_pdo_crossSchema` Amazon Redshift マッピングを選択します。

   b. **[スケジュール]** タブの **[詳細オプション]** で、**[プッシュダウンの最適化]**を追加して、値を **[完全]** に設定します。

   c. **[クロススキーマプッシュダウンの最適化の有効化]** を選択します。

      次の図に、設定した **[クロススキーマプッシュダウンの最適化の有効化]** プロパティを示します。



d. タスクを保存し、**[終了]** をクリックします。

マッピングタスクを実行すると、Secure Agent は、CQA_SCHEMA1 スキーマに関連付けられた Amazon Redshift ソースオブジェクトからデータを読み取り、CQA_SCHEMA2 スキーマに関連付けられた Amazon Redshift ターゲットオブジェクトにデータを書き込みます。

# プッシュダウンの最適化の関数に関するルールおよびガイドライン

Amazon Redshift データベースに関数をプッシュする際には、以下のルールおよびガイドラインを使用します。

- TRUNC(DATE)を Amazon Redshift にプッシュするには、日付および形式引数を定義する必要があります。定義しないと、エージェントはこの関数を Amazon Redshift にプッシュしません。
- Amazon Redshift の集計関数が受け取る引数は 1 つだけで、それは集計関数のフィールドセットです。フィルタ条件引数は認められません。また、ターゲットにマッピングされているすべてのフィールドが、GROUP BY 句に指定されていることを確認します。
- SYSTIMESTAMP()に形式を指定して、SYSTIMESTAMP を Amazon Redshift にプッシュしないでください。Amazon Redshift データベースは完全なタイムスタンプを返します。
- INSTR()を Amazon Redshift にプッシュするには、文字列、search_value、および開始引数のみを定義します。Amazon Redshift では、オカレンスおよび comparison_type 引数はサポートされていません。
- TO_BIGINT および TO_INTEGER を Amazon Redshift にプッシュすると、フラグ引数は無視されます。
- IN()を Amazon Redshift にプッシュすると、CaseFlag 引数は無視されます。
- ADD_TO_DATE()関数の一部として NS 形式を使用する場合、エージェントではこの関数が Amazon Redshift にプッシュされません。
- 次の形式のいずれかを TO_CHAR()および TO_DATE()関数の一部として使用する場合、エージェントではこれらの関数が Amazon Redshift にプッシュされません。

  - NS

  - SSSS

  - SSSSS

  - RR
- TRUNC(DATE)、GET_DATE_PART()、および DATE_DIFF()を Amazon Redshift にプッシュするには、次の形式を使用する必要があります。

  - D

  - DDD

  - HH24

  - MI

  - MM

  - MS

  - SS

  - US

  - YYYY

# 第 20 章

# データ型リファレンスの概要

データ統合では、Amazon Redshift との同期タスク、マッピング、およびマッピングタスクに次のデータ型が使用されます。

Amazon Redshift **のネイティブデータ型**

Amazon Redshift のデータ型は、フィールドのメタデータを編集するときに、ソースおよびターゲットトランスフォーメーションに表示されます。

**トランスフォーメーションデータ型**

トランスフォーメーションで表示されるデータ型のセットです。ANSI SQL-92 汎用データ型に基づく内部データ型で、プラットフォーム間でデータを移動するときランタイム環境によって使用されます。同期タスク、マッピング、およびマッピングタスクのトランスフォーメーションすべてにトランスフォーメーションデータ型が表示されます。

データ統合は、ソースデータを読み取るときに、ネイティブデータ型を対応するトランスフォーメーションデータ型に変換してから、データのトランスフォームを実行します。データ統合は、ターゲットに書き込むときに、トランスフォーメーションデータ型を対応するネイティブデータ型に変換します。

## Amazon Redshift とトランスフォーメーションのデータ型

次の表に、ランタイム環境がサポートしている Amazon Redshift データ型と、対応するトランスフォーメーションデータ型の一覧を示します。

| Amazon Redshift **のデータ型** | **トランスフォーメーションデータ型** | **説明** |
|---|---|---|
| Bigint | Bigint | 符号付き 8 バイト整数。 |
| Boolean | Small Integer | 論理ブール型（true/false）。 |
| Char | String | 固定長文字列。 |
| 日付 | タイムスタンプ | カレンダーの日付（年、月、日）。 |
| Decimal | Decimal | 選択精度の正確な数値。 |
| Double Precision | ダブル | 倍精度浮動小数点数。 |

| Amazon Redshift のデータ型 | トランスフォーメーションデータ型 | 説明 |
| --- | --- | --- |
| Integer | Integer | 符号付き 4 バイト整数。 |
| Real | Double | 単精度浮動小数点数。 |
| Smallint | Small Integer | 符号付き 2 バイト整数。 |
| タイムスタンプ | タイムスタンプ | 日時（タイムゾーンなし）。 |
| Varchar | String | ユーザー定義で制限される可変長文字列。 |

# 第 21 章

# トラブルシューティング概要

Amazon Redshift コネクタのエラーをトラブルシューティングするには、以下のセクションを使用します。

## Amazon Redshift コネクタのトラブルシューティング

**ODBC 接続を使用して Amazon Redshift に接続し、UTF 文字を読み取り、Secure Agent が Linux にインストールされるときにタスクが失敗する問題を解決するにはどうすればよいですか?**

この問題の詳細については、https://kb.informatica.com/solution/23/Pages/62/516325.aspx?myk=516325 を参照してください。

**読み取り操作または書き込み操作を実行する Amazon Redshift 接続の圧縮オプションを有効にするときに、ローカルステージング領域の最大サイズの値はどのくらいですか?**

この問題の詳細については、https://kb.informatica.com/faq/7/Pages/16/497540.aspx?myk=497540 を参照してください。

**Amazon Redshift コネクタの AWS IAM 認証はどのように設定しますか?**

AWS IAM 認証の設定の詳細については、Configuring AWS IAM Authentication for Amazon Redshift and Amazon Redshift V2 Connectors を参照してください。。

**Amazon Redshift タスクの結果は、ID とプライマリキーを含むターゲットフィールドのマッピングを同じカラムまたは異なるカラムに行うかによって異なりますか?**

この問題の詳細については、https://kb.informatica.com/faq/7/Pages/21/535693.aspx?myk=535693 を参照してください。

## Amazon Redshift 接続のトラブルシューティング

**タスクを実行して Amazon Redshift にデータを書き込むときに次のエラーが原因でタスクが失敗します。**

```
Amazon_RedshiftWriter_30007 [ERROR] Copy command on record 'public.basic_data_types' failed due to
[ERROR: S3ServiceException:The bucket you are attempting to access must be addressed using the specified
endpoint. Please send all future requests to this endpoint.,Status 301,Error PermanentRedirect,Rid
A8BA401CC765AC53,ExtRid NAbd1uxKirJVjDas1zo3WONdQ/+6p674RYkO
```

この問題は、Amazon Redshift ユーザーと接続プロパティのクラスタがタスクの S3 バケットと異なる領域にあるために発生します。

タスクを設定して同じ領域の S3 バケットを接続のユーザーとクラスタとして使用する必要があります。異なる接続を使用して S3 バケットに書き込むこともできます。

# 索引

## 数字


8 進数値
  DELIMITER 47, 140
  QUOTE 47, 140


## A


Amazon Redshift
  ODBC 接続を経由したプッシュダウン 104
  Redshift V2 接続を経由したプッシュダウン 89
  spectrum 21
  SSL 設定 23, 126
  プッシュダウンの最適化の概要 162
  マッピングタスク 157
  概要 21
  接続のトラブルシューティング 174
  接続プロパティ 129
Amazon Redshift ODBC 接続
  設定 104, 165
Amazon Redshift ODBC 接続の設定
  Linux 107, 167
  Windows 104, 165
Amazon Redshift Spectrum
  前提条件タスク 30
Amazon Redshift V2
  サポート対象のタスクの種類 20
  ソース 38
  ターゲット 41
  接続プロパティ 33
Amazon Redshift V2 オブジェクト
  マッピング 49, 150
Amazon Redshift V2 コネクタ
  概要 20
  管理 24, 25
Amazon Redshift V2 ソース
  mapping 51
Amazon Redshift V2 ターゲット
  マッピング 57
Amazon Redshift V2 接続
  概要 33
  管理 22
  設定 88, 97, 109
Amazon Redshift コネクタ
  データフロー 12
Amazon Redshift ソース
  サーバーサイド暗号化 133
  ステージングディレクトリ 132
  マッピング 150
Amazon Redshift ターゲット
  ステージングディレクトリ 136
  マッピング 153
Amazon Redshift とトランスフォーメーション
  データ型 118
Amazon Redshift のデータ型
  データ型 172
Amazon Redshift ルックアップ
  マッピングタスク 64, 67
Amazon Redshift 接続
  概要 129
  管理 125


## C


CDC ソース
  Amazon Redshift マッピングタスク 71
Cloud Application Integration コミュニティ
  URL 8
Cloud 開発者コミュニティ
  URL 8
Copy コマンド
  オプション 43
  概要 43


## I


IAM 認証
  管理 27, 127
Informatica Intelligent Cloud Services
  Web サイト 8
Informatica グローバルカスタマサポート
  連絡先情報 9


## L


Linux
  プロキシの設定 37
Lookup ルックアップトランスフォーメーション
  ルックアップキャッシング 70


## O


ODBC の作成
  接続 108, 168


## P


pushdown optimization：プッシュダウンの最適化
  トランスフォーメーション 89, 91, 116
  関数 89, 91

## S

SQL トランスフォーメーション
　ストアドプロシージャの選択 74
　設定 73

## U

unload コマンド
　オプション 39
　概要 39

## W

Web サイト 8
Windows
　プロキシの設定 36

## あ

アップグレード通知 9

## き

キー範囲パーティション
　設定 77
キー範囲パーティション化 41
キャッシュ
　ルックアップキャッシュの有効化 70

## く

クロススキーマ
　プッシュダウン最適化 112, 170

## し

システムステータス 9

## す

スケールマッピング 18
ステータス
　Informatica Intelligent Cloud Services 9

## そ

ソースのパーティション化 41

## て

データのステージング
　フラットファイル 31
データ型リファレンス
　概要 118, 172

## と

トラブルシューティング
　Amazon Redshift V2 コネクタ 121
トランスフォーメーション
　pushdown optimization：プッシュダウンの最適化 116
　プッシュダウンの最適化 92

## ふ

フィールドマッピング
　Amazon Redshift ルックアップ 148
プッシュダウンの最適化
　トランスフォーメーション 92
　プレビュー 97
　関数のルールおよびガイドライン 98
プッシュダウンの最適化プレビュー 97
プッシュダウン最適化
　トランスフォーメーション 113, 162
　関数 113, 162
　関数のルールおよびガイドライン 116
フラットファイル
　データのステージング 31
プロキシ設定
　JVMOption 37
　Linux での設定 37
　Windows での設定 36

## ま

マッピング
　Amazon Redshift ルックアップ 156
　Oracle CDC ソース 17
マッピングタスク
　ターゲット 158
　ターゲットプロパティ 158
　ソース 157
　ソースのプロパティ 157
マッピングとマッピングタスク 17

## め

メンテナンスの停止 9

## り

リカバリおよびリスタート処理 45

## る

ルックアップキャッシュ
　動的 70
　保持 70