



管理：ロード管理

SAP Sybase IQ 16.0 SP03

ドキュメント ID：DC02010-01-1603-01

改訂：2013 年 12 月

Copyright © 2013 by SAP AG or an SAP affiliate company. All rights reserved.

このマニュアルの内容を SAP AG による明示的な許可なく複製または転載することは、形態や目的を問わず禁じられています。ここに記載された情報は事前の通知なしに変更されることがあります。

SAP AG およびディストリビュータが販売しているソフトウェア製品には、他のソフトウェアベンダ独自のソフトウェアコンポーネントが含まれているものがあります。国内製品の仕様は変わることがあります。

これらの資料は SAP AG および関連会社 (SAP グループ) が情報のみを目的として提供するものであり、いかなる種類の表明または保証も行わないものではなく、SAP グループはこの資料に関する誤りまたは脱落について責任を負わないものとします。SAP グループの製品およびサービスに関する保証は、かかる製品およびサービスに付属している明確な保証文書がある場合、そこで明記されている保証に限定されます。ここに記載されているいかなる内容も、追加保証を構成するものとして解釈されるものではありません。

ここに記載された SAP および他の SAP 製品とサービス、ならびに対応するロゴは、ドイツおよび他の国における SAP AG の商標または登録商標です。その他の商標に関する情報および通知については、<http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> を参照してください。

目次

| | |
|--------------------------------------|-----------|
| はじめにお読みください | 1 |
| インポートとエクスポートの概要 | 7 |
| インポートとエクスポートの方法選択 | 7 |
| データの入出力フォーマット | 8 |
| Interactive SQL の出力フォーマットの指定 | 8 |
| データ修正のパーミッション | 9 |
| データベース更新のスケジュール | 9 |
| データベースからデータをエクスポートする方法 | 11 |
| 出力リダイレクション | 11 |
| データ抽出機能 | 12 |
| 抽出オプション | 12 |
| データ抽出オプションの有効化 | 18 |
| LOAD TABLE 文を使用したバルクロード | 23 |
| 入力データフォーマットを指定するロード | 25 |
| クライアントからの直接のデータロード | 27 |
| 分割されたテーブルロードに関する考慮事項 | 28 |
| メッセージのロードと挿入 | 29 |
| 整合性制約違反のメッセージ | 29 |
| MESSAGE LOG の内容とフォーマット | 31 |
| ROW LOG の内容とフォーマット | 32 |
| MESSAGE LOG と ROW LOG の例 | 33 |
| BINARY ロード形式 | 35 |
| BINARY ロード形式とロード効率 | 35 |
| オペレーティングシステムのネイティブデータ型 | 36 |
| DATE | 36 |
| TIME | 37 |
| TIMESTAMP | 38 |
| NUMERIC と DECIMAL | 38 |
| NULL 値のロード | 40 |

| | |
|--|-----------|
| INSERT 文の使用 | 45 |
| ロー単位での指定した値の挿入 | 45 |
| データベースからの指定したローの挿入 | 46 |
| 別のデータベースからの挿入 | 47 |
| 対話型データインポート | 51 |
| エンディアンフォーマットが異なるシステム間でのデータの移動 | 53 |
| プライマリキーカラムと外部キーカラムへの挿入 | 57 |
| ラージオブジェクトデータのロードまたは抽出 | 59 |
| 挿入時のデータ変換 | 61 |
| ロード変換オプション | 62 |
| 明示的なデータ変換 | 63 |
| カラム幅の問題 | 67 |
| 日付と時刻のロードの高速化 | 68 |
| ASCII 入力の変換 | 69 |
| NULL またはブランク文字の置換 | 71 |
| DATE オプション | 71 |
| DATE フォーマット | 72 |
| DATETIME 変換オプション | 73 |
| DATETIME 変換フォーマットの指定 | 74 |
| NULL データ変換 | 76 |
| 結果の丸めとトランケート | 79 |
| 対応する Adaptive Server データ型 | 81 |
| サポートされていない Adaptive Server データ型 | 81 |
| Adaptive Server のデータ型に対応するデータ型 | 82 |
| データインポート時の変換エラー | 86 |
| バルクロードのチューニング | 87 |
| データベースを定義するときのロードパフォーマンス | 87 |
| ロード時の環境の調整 | 88 |
| ロード時のスレッド使用 | 89 |
| テーブルローの変更 | 91 |

| | |
|----------------|----|
| データ削除の方法 | 93 |
| 索引 | 95 |

目次

はじめにお読みください

SAP® Sybase® IQ 16 新機能の概要 では、SAP Sybase IQ の機能がすべて説明されていますが、一部の機能では新しいアーキテクチャを利用するためにユーザ側での追加アクションが必要になる可能性があります。

たとえば、以前のリリースからアップグレードしたお客さまは、初期設定の互換性オプションの変更や、別のデータ型を収容するための幅の広いカラムの再構築が必要になることがあります。新しいロードエンジンでは、パフォーマンスは向上しますが、利用可能なすべてのハードウェアリソースを効率的に使用するためにデフォルトのメモリ割り付けを変更する必要があります。

NBit

継続的な *NBit* ディクショナリ圧縮は、1、2、および3バイトのディクショナリ圧縮に代わって、16.0 のデフォルトのカラム保管メカニズムとなりました。LOB (文字とバイナリ) データ型と BIT データ型を除くすべてのデータ型を *NBit* カラムにすることができます。

IQ UNIQUE カラム制約では、カラムが Flat FP と *NBit* FP のいずれとしてロードされるかを決定します。IQ UNIQUE の *n* 値を 0 に設定すると、カラムは Flat FP としてロードされます。0 より大きく、FP_NBIT_AUTOSIZE_LIMIT より小さい *n* 値を設定すると、初期サイズが *n* の *NBit* カラムが作成されます。IQ UNIQUE 制約が設定されていないカラムは、自動サイズ制限値に達するまで暗黙的に *NBit* としてロードされます。

自動サイズ制限値より小さい *n* 値を指定した IQ UNIQUE を使用する必要はありません。ロードエンジンによって、カーディナリティが低いか中程度のカラムはすべて *NBit* としてサイズ決定されます。カラムをフラット Flat FP としてロードする場合や、個別値の数が自動サイズ制限値を超えるとときにカラムを *NBit* としてロードする場合に、IQ UNIQUE を使用します。

ロードとラージメモリ

ラージメモリは、SAP Sybase IQ が一時的に使用するために OS に対して動的に要求できるメモリの最大量を表します。一部のロード操作では、デフォルトの 2 GB よりも多くのメモリが必要になることがあるため、利用可能な総物理メモリ量に基づいてラージメモリとキャッシュメモリの割り付けを制御するように起動オプションを調整してください。

原則として、ラージメモリ所要量は、SAP Sybase IQ に割り付けた利用可能な総物理メモリの 3 分の 1 を表します。IQ のメインストアとテンポラリストアに十分な

はじめにお読みください

メモリを確保するために、起動パラメータ `-iqlm`、`-iqtc`、および `-iqmc` の値をそれぞれ、SAP Sybase IQ に割り付けた利用可能な総物理メモリの 3 分の 1 の量に設定してください。

ほとんどの場合、SAP Sybase IQ プロセスがスワップアウトされないようにするため、総物理メモリの 80% を SAP Sybase IQ に割り付けます。同じシステム上で稼働する他のプロセスを考慮して、実際のメモリ割り付けを調整してください。たとえば、コア数が 32、利用可能な総物理メモリが 128 GB のマシンでは、100 GB (総メモリ 128 GB のおよそ 80%) を SAP Sybase IQ プロセスに割り付けます。上記の原則に従って、`-iqlm`、`-iqtc`、および `-iqmc` パラメータの値をそれぞれ 33 GB に設定します。

『ユーティリティガイド』の「`-iqlm iqsrv16` サーバオプション」と「`-iqmc iqsrv16` サーバオプション」を参照してください。

インデックスの変更

FP および HG インデックスの変更は、新しいカラム圧縮メカニズムを利用したもので、ロードパフォーマンスを向上させます。

| インデックス | 説明 |
|---------------------|--|
| 新しい高速射影 (FP) インデックス | FP (1)、FP (2)、および FP (3) バイトのディクショナリ圧縮の代わりに、新しい継続的 NBit ディクショナリ圧縮を利用する。FP (1)、FP (2)、および FP (3) インデックスは、それぞれ NBit (8)、NBit (16)、および NBit (24) にロールオーバーされる。 FP_NBIT_IQ15_COMPATIBILITY='OFF' の場合、カラムに適用された IQ UNIQUE 制約によってそのカラムが Flat FP と NBit のいずれとしてロードされるかが決定される。 『管理：データベース』の「高速射影 (FP) インデックス」を参照。 |

| インデックス | 説明 |
|-------------------|---|
| 新しい多層 HG インデックス構造 | <p>ロードパフォーマンスを HG インデックスサイズから切り離す。15.x では、HG インデックスのデータ量の増加にともなって、ロードスループットが低下することがあった。インデックスが大きくなるほど、同じ量のデータのロードにかかる時間が増えていた。新しい多層構造によって、HG インデックスサイズからロードパフォーマンスが切り離され、スループットが向上する。</p> <p>CREATE_HG_WITH_EXACT_DISTINCTS オプションは、新しく作成された HG インデックスが多層であるか単層であるかを決定する。このオプションは、すべての新しい 16.0 データベースと 15.x から移行されたすべての 16.0 データベースで ON になる。新しい構造を利用するには、このオプションを OFF に設定する。単層 HG インデックスから多層 HG への変換（およびこの逆の変換）には、sp_iqrebuildindex を使用する。</p> <p>『リファレンス：文とオプション』の「CREATE_HG_WITH_EXACT_DISTINCTS オプション」を参照。</p> |

ストアドプロシージャ

新しいストアドプロシージャは、カラムインデックスとカラム制約に関する情報を返します。

| プロシージャ | 説明 |
|-----------------------------|---|
| sp_iqindexmeta-data | <p>インデックスタイプ (Flat FP、NBit、HG、および多層 HG)、個別カウント、IQ UNIQUE <i>n</i> 値、NBit ディクショナリサイズなど、カラムインデックスに関する詳細を返す。</p> <p>『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「sp_iqindexmetadata プロシージャ」を参照。</p> |
| sp_iqcolumnmeta-data | <p>1 つまたは複数のユーザテーブルまたはデータベース内の全テーブルの FP インデックスメタデータを返す。</p> <p>『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「sp_iqcolumnmetadata プロシージャ」を参照。</p> |

| プロシージャ | 説明 |
|----------------------------------|--|
| sp_iqindexrebuildwidedata | <p>事前に再構築しないと読み込み/書き込みアクティビティに利用できない幅の広いカラムを識別する。出力には、カラムを再構築するために sp_iqrebuildindex とともに使用できる文が含まれる。</p> <p>『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「sp_iqindexrebuildwidedata プロシージャ」を参照。</p> |
| sp_iqrebuildindex | <p>FP インデックス (Flat FP を NBit として、または NBit を Flat FP として) と HG インデックス (単層 HG を多層 HG として、または多層 HG を単層 HG として) を再構築する。新しいデータを挿入または更新する前に、255 バイト幅より大きいカラムをすべて再構築する必要がある。</p> <p>index_clause は、IQ UNIQUE <i>n</i> を 0 (NBit カラムを Flat FP に再キャスト) から FP_NBIT_AUTOSIZE_LIMIT および FP_NBIT_LOOKUP_MB オプションで定義されている制限値までの間の明示的な値にリセットできる。</p> <p>また、sp_iqrebuildindex は、ラージオブジェクト (LOB) データが含まれるカラムに対する読み書きアクセスを有効にする。15.x データベースから移行された LOB カラムは、sp_iqrebuildindex を実行するまでは読み込み専用となる。FP_NBIT_AUTOSIZE_LIMIT 以下の IQ UNIQUE 値が指定された NBit カラムの推定カーディナリティは、FP_NBIT_IQ15_COMPATIBILITY の設定にかかわらず 0 として格納される。これは、sp_iqindexmetadata から返される値に影響する。</p> <p>『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「sp_iqrebuildindex プロシージャ」を参照。</p> |

| プロシージャ | 説明 |
|-----------------------|--|
| sp_iqrebuildindexwide | <p>SAP Sybase IQ は、225 文字より広い CHAR、VARCHAR、BINARY、および VARBINARY カラムを暗黙的に再構築する。また、16.0 以前の非 RLV ベース テーブルが読み書きアクセスのために初めて開かれたときに、SAP Sybase IQ 16.0 SP03 に移行されたデータベース内のすべての LONG VARCHAR カラムと LONG BINARY カラムも再構築する。</p> <p>sp_iqrebuildindexwide プロシージャを使用して幅の広いカラムを明示的に再構築することもできる。sp_iqrebuildindexwide は、table_name、table_owner、および level 別に幅の広いテーブルを再構築できる。引数によっては、このプロシージャは、幅が 255 バイトを超える 16.0 より前のカラムのすべて、トークン化された FP の一部または全部、VARCHAR/VARBINARY カラム、およびデータベース内の指定されたテーブルの固定フラット FP のすべてを再構築できる。</p> <p>『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「sp_iqrebuildindexwide プロシージャ」を参照。</p> |

データベースオプション

一部のデータベースオプションは、16.0 の機能を利用できません。データベースアップグレード後も制限された互換性を維持することで、既存のアプリケーションを移行するためのある程度の柔軟性が提供されます。

| オプション | 説明 |
|----------------------------|--|
| FP_NBIT_IQ15_COMPATIBILITY | <p>15.x で利用可能なものと同様のトークン化された FP をサポートする。このオプションは、15.x からアップグレードされたすべての 16.0 データベースではデフォルトで ON、新しく作成されたすべての 16.0 データベースではデフォルトで OFF になる。</p> <ul style="list-style-type: none"> このオプションが ON の場合、データベースエンジンは MINIMIZE_STORAGE、FP_LOOKUP_SIZE、および FP_LOOKUP_SIZE_PPM オプションを使用して、カラム保管を最適化する。これらのオプションは、16.0 では無視される。 このオプションが OFF の場合、データベースエンジンは 15.x のオプションと SAP Sybase IQ の NBit 保管オプションに従ったカラムを無視する。 <p>NBit カラム圧縮を利用するには、このオプションを OFF に設定する。</p> |

| オプション | 説明 |
|--|---|
| CREATE_HG_WITH_EXACT_DISTINCTS | <p>新しい HG インデックス (CREATE INDEX コマンドを使用して明示的に作成されるものと、PRIMARY KEY または FOREIGN KEY 宣言に基づくテーブルの暗黙的な作成または変更によって作成されるもの) が多層であるか単層であるかを決定する。このオプションは、15.x からアップグレードされたすべてのデータベースと新しく作成されたすべての 16.0 データベースで ON 16.0 になる。このオプションが ON の場合、新しい HG インデックスはすべて単層になる。新しい多層 HG インデックス構造を利用するには、このオプションを OFF に設定する。</p> <p>単層 HG インデックスから多層 HG への変換 (およびこの逆の変換) には、sp_iqrebuildindex を使用する。</p> |
| CREATE_HG_AND_FORCE_PHYSICAL_DELETE | <p>16.0 の多層 HG インデックス削除動作を制御する。このオプションは、SAP Sybase IQ が物理的な削除を直ちに実行するか、それともロードでの後の時点まで遅らせて削除を実行するかを決定する。</p> <p>CREATE_HG_AND_FORCE_PHYSICAL_DELETE はデフォルトで ON になる。ON の場合、SAP Sybase IQ は物理的な削除を実行する。</p> |
| REVERT_TO_V15_OPTIMIZER | <p>REVERT_TO_V15_OPTIMIZER は、SAP Sybase IQ 15.x の動作を模倣するようにクエリオプティマイザに強制する。15.x からアップグレードされたすべての 16.0 データベースでは、デフォルトで REVERT_TO_V15_OPTIMIZER='ON'。新しく作成されたすべての SAP Sybase IQ 16.0 データベースでは、デフォルトで REVERT_TO_V15_OPTIMIZER='OFF'。</p> <p>SAP Sybase IQ ハッシュ分割機能を使用する場合は、15.x から 16.0 にアップグレードされたデータベースでは REVERT_TO_V15_OPTIMIZER='OFF' に設定する。</p> |

インポートとエクスポートの概要

SAP Sybase IQ では、フラットファイルまたは直接データベーステーブルからデータをインポートできます。指定した値をデータベースに直接入力することもできます。Interactive SQL ユーティリティと IQ データ抽出機能から他のフォーマットにデータをエクスポートできます。

SAP Sybase IQ テーブルは論理テーブルで、データは格納されていません。データなど、クエリを解決するために必要なすべての情報は、SAP Sybase IQ のインデックスに格納されています。データを IQ テーブルのカラムに挿入しても、実際はデータはテーブルのカラムではなく、カラムインデックスに追加されます。インデックスを作成するには、テーブル単位でデータを挿入します。

インポートとエクスポートの方法選択

SAP Sybase IQ では、データを追加、変更、または削除する方法を選択できます。

- フラットファイルからテーブルを効率的にバルクロードするには、SQL 文 **LOAD TABLE** を使用します。
- 指定した値をテーブルにロー単位で挿入するには、**VALUES** オプションを指定した SQL 文 **INSERT** を使用します。
- テーブル (他のデータベース内にあるテーブルも含む) から選択したローを挿入するには、**SELECT** 文の句を指定した SQL 文 **INSERT** を使用します。
- テーブルから特定のローを削除するには、**DELETE** 文を使用します。
- テーブルの既存のローを変更するには、**UPDATE** 文を使用します。

IQ データ抽出機能では、データがバイナリまたは ASCII フォーマットでエクスポートされ、それを他のデータベースにロードできます。この機能は、大量のデータを移動する場合や、ロードに使用できる出力ファイルが必要な場合に使用します。

Interactive SQL からは、データをさまざまなフォーマットで別のデータベースにエクスポートしたり、テキストファイルを出力したりできます。コマンドの出力のリダイレクトもできます。

注意： SAP Sybase IQ では、**LOAD TABLE FORMAT BCP** オプションによって BCP をサポートしています。IQ テーブルへの BCP を直接実行できます。また、SAP Sybase IQ では **LOAD TABLE USING CLIENT FILE** オプションを使用したりリモートデータのバルクロードもサポートしています。

参照：

- データの入出力フォーマット (8 ページ)
- データ修正のパーミッション (9 ページ)
- データベース更新のスケジュール (9 ページ)

データの入出力フォーマット

LOAD TABLE 文では、ファイルからロー単位でデータをインポートします。固定長フィールド、または可変長フィールドの ASCII 入力ファイルとバイナリ入力ファイルの両方がサポートされます。可変長フィールドは、フィールドの末尾がデリミタによって指定されます。

INSERT 文は、指定した一連の値から、または直接テーブルから、データを SAP Sybase IQ テーブルに移動します。

OUTPUT Interactive SQL でサポートされている文のファイルフォーマットには、次のようなものがあります。

- **TEXT**
- **FIXED**
- **HTML**
- **SQL**
- **XML**

IQ データ抽出機能では、データはバイナリまたは ASCII フォーマットでエクスポートされます。

参照：

- インポートとエクスポートの方法選択 (7 ページ)
- データ修正のパーミッション (9 ページ)
- データベース更新のスケジュール (9 ページ)

Interactive SQL の出力フォーマットの指定

Interactive SQL のデフォルトの出力フォーマットを指定できます。

1. [SQL 文] ウィンドウで、[オプション]を選択します。
2. [インポート/エクスポート]を選択します。
3. ドロップダウンリストからデフォルトのエクスポートフォーマットを選択します。

4. ドロップダウンリストからデフォルトのインポートフォーマットを選択します。

データ修正のパーミッション

データ修正文は、修正するデータベーステーブルに対する適切なパーミッションがある場合のみ実行できます。

データベースの管理者とデータベースオブジェクトの所有者は GRANT 文と REVOKE 文を使用して、だれがどのデータ修正機能にアクセスするかを決定します。

データを挿入するには、テーブルまたはビューの INSERT パーミッションが必要となります。データを削除するには、テーブルまたはビューの DELETE パーミッションが必要です。データを更新するには、UPDATE パーミッションが必要です。DBA は、テーブルへのデータの挿入や削除を行うことができます。テーブルの所有者には、該当するテーブルの INSERT、DELETE、UPDATE のパーミッションが付与されています。

パーミッションを個々のユーザ、ロール、または PUBLIC ロールに付与したり、取り消したりできます。

参照：

- インポートとエクスポートの方法選択 (7 ページ)
- データの入出力フォーマット (8 ページ)
- データベース更新のスケジュール (9 ページ)

データベース更新のスケジュール

複数のユーザがデータベーステーブルに対してクエリを実行でき、データベースを同時に更新できます。

参照：

- インポートとエクスポートの方法選択 (7 ページ)
- データの入出力フォーマット (8 ページ)
- データ修正のパーミッション (9 ページ)

インポートとエクスポートの概要

データベースからデータをエクスポートする方法

データベースからデータをエクスポートするには、出力リダイ렉션やデータ抽出機能の使用など、いくつかの方法があります。

独自の、またはサードパーティ製のフロントエンドツールを使用して、効率よくIQ データベースにクエリを実行して希望どおりにフォーマットしたデータをエクスポートすることもできます。

出力リダイ렉션

出力リダイクションを使用して、クエリ結果をエクスポートできます。

リダイクション記号の **>#** とともに出力先を指定してコマンドを実行することにより、コマンドの出力をファイルまたはデバイスにリダイレクトできます。リダイクション記号の後にはファイル名が続きます。コマンドファイル内では、ファイル名の後に置かれたセミコロンが文のターミネータとして使用されます。ファイルは、Interactive SQL が起動したディレクトリからの相対位置に配置されま

す。

この例では、出力は `empfile` というファイルにリダイレクトされます。

```
SELECT *  
FROM Employees  
># empfile
```

ファイル名を引用符で囲まないようにします。

出力リダイクションは、**SELECT** 文の中で特に有効です。

ファイルの内容を置き換える代わりに、指定したファイルに出力を追加するには、1つのリダイクション記号に **>** を1つではなく2つ使用します (**>>#** など)。指定したファイルの最初から出力を開始する場合、見出しをサポートする出力フォーマットでは、**SELECT** 文の結果に見出しが追加されます。

出力とメッセージのリダイレクト

リダイクション記号の **>&** を使用すると、エラーメッセージと、エラーが発生したコマンドに関する統計を含むすべての出力がリダイレクトされます。次に例を示します。

```
SELECT *  
FROM Employees  
>& empfile
```

データベースからデータをエクスポートする方法

ファイル名を引用符で囲まないようにします。

この例では、ファイル `empfile` に **SELECT** 文を送信し、さらにその文からの出力とコマンドに関係する統計を送信します。

>& リダイレクションメソッドを使用して、**READ** コマンドの動作のログを取得できます。各コマンドの統計とエラーは、リダイレクトされた出力ファイルの中でコマンドに続いて書き出されます。

NULL 値の出力

データを書き出す理由のうち、最も多いのは他のソフトウェアで使用するためですが、他のソフトウェアでは NULL 値を適切に処理できない場合があります。

dbisql オプションの **NULLS** を使用すると、NULL 値を出力する方法を選択できます。または、**IFNULL** 関数を使用して、NULL 値が出てくるたびに指定した値を出力できます。

データ抽出機能

データ抽出機能は、サイズの大きい結果セットを生成するクエリのパフォーマンスを大幅に向上させるデータベースオプションのグループです。

他のデータベースオプションと同様に、データ抽出オプションはテンポラリオプションまたは永久オプションとして設定できます。固有の接続の場合は通常、これらのオプションはテンポラリとして設定されます。

抽出オプションを使用する利点として、次のようなものがあります。

- バイナリフォーマットがサポートされています。したがって、出力データを同一または異なる IQ データベースにロードできます。
- SELECT** 文の出力が大きい場合に、ASCII 出力では最高 4 倍、バイナリ出力では最高 9 倍速く処理できます。

抽出オプション

抽出オプションを使用すると、**SELECT** 文の出力先を標準インタフェースからリダイレクトし、1 つまたは複数のディスクファイルまたは名前付きパイプに直接出力できます。

| オプション名 | 指定可能な値 | デフォルト値 |
|-------------------------------|--------|--------|
| Temp_Extract_Append | ON、OFF | OFF |
| Temp_Extract_Binary | ON、OFF | OFF |
| Temp_Extract_Column_Delimiter | 文字列 | ' ' |

データベースからデータをエクスポートする方法

| オプション名 | 指定可能な値 | デフォルト値 |
|----------------------------|-------------|--------|
| Temp_Extract_Directory | 文字列 | " |
| Temp_Extract_Name1 | 文字列 | " |
| Temp_Extract_Name2 | 文字列 | " |
| Temp_Extract_Name3 | 文字列 | " |
| Temp_Extract_Name4 | 文字列 | " |
| Temp_Extract_Name5 | 文字列 | " |
| Temp_Extract_Name6 | 文字列 | " |
| Temp_Extract_Name7 | 文字列 | " |
| Temp_Extract_Name8 | 文字列 | " |
| Temp_Extract_Null_As_Empty | ON、OFF | OFF |
| Temp_Extract_Null_As_Zero | ON、OFF | OFF |
| Temp_Extract_Quote | 文字列 | " |
| Temp_Extract_Quotes | ON、OFF | OFF |
| Temp_Extract_Quotes_All | ON、OFF | OFF |
| Temp_Extract_Row_Delimiter | 文字列 | " |
| Temp_Extract_Size1 | プラットフォーム固有* | 0 |
| Temp_Extract_Size2 | プラットフォーム固有* | 0 |
| Temp_Extract_Size3 | プラットフォーム固有* | 0 |
| Temp_Extract_Size4 | プラットフォーム固有* | 0 |
| Temp_Extract_Size5 | プラットフォーム固有* | 0 |
| Temp_Extract_Size6 | プラットフォーム固有* | 0 |
| Temp_Extract_Size7 | プラットフォーム固有* | 0 |
| Temp_Extract_Size8 | プラットフォーム固有* | 0 |
| Temp_Extract_Swap | ON、OFF | OFF |

*TEMP_EXTRACT_SIZE_n オプションのデフォルト値はプラットフォーム固有です。これらの値は次のとおりです。

データベースからデータをエクスポートする方法

- AIX と HP-UX : 0 ~ 64GB
- Sun Solaris : 0 ~ 512GB
- Windows : 0 ~ 128GB
- Linux : 0 ~ 512GB

JFS2 など、大規模なファイルシステムでデフォルト値よりも大きなサイズをサポートしている場合は、TEMP_EXTRACT_SIZE_n をそのファイルシステムで許容される最大値に設定します。たとえば、1TB の設定オプションをサポートするには、次のように入力します。

```
SET OPTION TEMP_EXTRACT_SIZE1 = 1073741824 KB
```

注意： 整数値の指定が可能なデータベースオプションの場合、SAP Sybase IQ によって *option-value* の小数設定がすべて整数値にトランケートされます。たとえば、3.8 という値は 3 にトランケートされます。

この中で最も重要なオプションは、TEMP_EXTRACT_NAME1 です。このオプションがデフォルト値 (空の文字列) に設定されている場合、抽出処理は使用できず、リダイレクトされる出力もありません。抽出処理を使用可能にするには、TEMP_EXTRACT_NAME1 にパス名を設定してください。他で使用されていないパスとファイル名を選択してください。存在しないファイルを指定した場合、データ抽出機能がそのファイルを作成します。

指定したファイルが格納されているディレクトリまたはフォルダと指定したファイル自体の両方に対する書き込みパーミッションが IQ を起動したユーザ (たとえば *sybase*) に設定されている必要があります。追加モードの場合、データ抽出機能は、ファイルにすでに書き込まれているデータに上書きするのではなく、抽出されたローをファイルの末尾に追加します。

警告！ 既存のファイルのパス名を指定し、TEMP_EXTRACT_APPEND オプションが OFF (デフォルト) になっている場合は、ファイルの内容が上書きされます。

TEMP_EXTRACT_NAME2 から TEMP_EXTRACT_NAME8 までの各オプションを使用して、複数の出力ファイルの名前を指定します。これらのオプションは順番に使用する必要があります。たとえば、TEMP_EXTRACT_NAME1 と TEMP_EXTRACT_NAME2 を先に設定していない状態で TEMP_EXTRACT_NAME3 を設定した場合、このオプションは無効になります。

TEMP_EXTRACT_SIZE1 から TEMP_EXTRACT_SIZE8 を使用して、それぞれに対応する出力ファイルの最大サイズを指定します。TEMP_EXTRACT_SIZE1 では TEMP_EXTRACT_NAME1 で指定した出力ファイルの最大サイズを指定し、TEMP_EXTRACT_SIZE2 では TEMP_EXTRACT_NAME2 で指定した出力ファイルの最大サイズを指定します (以下、同様)。

データ抽出サイズのオプションのデフォルトの最小値は 0 です。IQ はこのデフォルト値を次の値に変換します。

| デバイスタイプ | サイズ |
|----------|--|
| ディスクファイル | AIX と HP-UX : 0 ~ 64GB Sun Solaris と Linux : 0 ~ 512GB Windows の場合 : 0 ~ 128GB |
| その他 | 無制限 |

TEMP_EXTRACT_APPEND は、TEMP_EXTRACT_SIZE_n オプションとの互換性はありません。抽出追加出力ファイルのサイズを制限しようとすると、SAP Sybase IQ からエラーが返されます。

抽出先のディスクファイルまたは名前付きパイプが 1 つだけの場合は、TEMP_EXTRACT_NAME₂ から TEMP_EXTRACT_NAME₈ と、TEMP_EXTRACT_SIZE₁ から TEMP_EXTRACT_SIZE₈ はデフォルト値のままにします。

注意： SELECT がローを返さず、リダイレクトする出力がない場合は、長さゼロの空のファイルが作成されます。複数の抽出ファイルを指定したが、それらすべてを使用するほどデータがない、という場合でも、すべてのファイルが作成されます。

アクセスの制御

TEMP_EXTRACT_DIRECTORY オプションは、ユーザがデータ抽出機能を使用できるかどうかを制御します。また、一時抽出ファイルが配置されるディレクトリを制御し、TEMP_EXTRACT_NAME_n オプションで指定されたディレクトリパスは上書きされます。

ユーザの TEMP_EXTRACT_DIRECTORY オプションが FORBIDDEN (大文字と小文字を区別しない) という文字列に設定されている場合、そのユーザはデータ抽出を実行できません。実行を試みると、「抽出を実行するためのパーミッションがありません。」というエラーが返されます。

PUBLIC ロールの TEMP_EXTRACT_DIRECTORY が FORBIDDEN に設定されている場合は、どのユーザもデータ抽出を実行できません。

TEMP_EXTRACT_DIRECTORY が有効なディレクトリパスに設定されている場合、一時抽出ファイルは指定されたディレクトリに配置され、TEMP_EXTRACT_NAME_n オプションで指定されたパスは上書きされます。

データベースからデータをエクスポートする方法

TEMP_EXTRACT_DIRECTORY が無効なディレクトリパスに設定されている場合は、次のようなエラーが発生します。「ファイルが存在しません ファイル：
<invalid path>」

TEMP_EXTRACT_DIRECTORY が空白の場合、一時抽出ファイルは、TEMP_EXTRACT_NAME_n で指定されたディレクトリに配置されます。TEMP_EXTRACT_NAME_n でパスが指定されていない場合、抽出ファイルはデフォルトでサーバ起動ディレクトリに配置されます。

TEMP_EXTRACT_DIRECTORY オプションは、大きなデータ抽出ファイルの作成を、ユーザが書き込みアクセスを持つディレクトリに制限することで、より充実したセキュリティを実現し、ディスク管理を制御しやすくします。このオプションを設定するには、SET ANY SYSTEM OPTION システム権限が必要です。このオプションはすぐに有効になります。

抽出の種類

データ抽出には、次のような種類があります。

- バイナリ
- バイナリ／スワップ
- ASCII

バイナリ抽出を実行すると、全体的には「バイナリ」フォーマット、各カラムには「null バイト付きバイナリ」フォーマットが指定されたファイルが生成されます。**LOAD TABLE** 文を使用すると、このファイルをロードできます。

バイナリ／スワップ抽出は、逆のエンディアン別のマシンにロードするように設計されている点以外は、バイナリ抽出と同じ働きをします。

ASCII 抽出を実行すると、テキストファイルが生成されます。

2つのオプション Temp_Extract_Binary と Temp_Extract_Swap によって、3種類の抽出処理のうちどの抽出処理が実行されるかが決まります。

| 種類 | Temp_Extract_Binary | Temp_Extract_Swap |
|-----------|---------------------|-------------------|
| バイナリ | ON | OFF |
| バイナリ／スワップ | ON | ON |
| ASCII | OFF | OFF |

抽出の種類のリフォルトは ASCII です。

TEMP_EXTRACT_BINARY オプションを ON に設定し、抽出機能を使用してデータがアンロードされる場合は、バイナリデータのロード時に **LOAD TABLE** 文の **BINARY WITH NULL BYTE** パラメータを各カラムに使用する必要があります。

カラムとローのデリミタ

ASCII 抽出の場合、デフォルトでは、カラム値はカンマで区切られ、ローは、UNIX プラットフォームでは改行、Windows プラットフォームでは復帰/改行の組み合わせで終了します。文字列には引用符は付きません。これらのデフォルト設定が適切でない場合は、次のオプションを使用してデリミタを変更します。

- Temp_Extract_Column_Delimiter
- Temp_Extract_Row_Delimiter
- Temp_Extract_Quote
- Temp_Extract_Quotes
- Temp_Extract_Quotes_All

デリミタには 1 バイトから 4 バイト必要で、マルチバイト照合順を使用している場合はその照合順でデリミタが有効である必要があります。デリミタとして使用する文字列は、データ出力文字列内で使われないものにしてください。

Temp_Extract_Row_Delimiter オプションのデフォルトは ' ' (空の文字列) です。IQ は、このオプションのデフォルトの空の文字列を、UNIX プラットフォームでは改行文字、Windows プラットフォームでは復帰/改行文字に変換します。

Temp_Extract_Column_Delimiter オプションは、カラム間のデリミタを制御します。ASCII 抽出で、このオプションが空の文字列に設定されている場合は、抽出されたデータは、カラムデリミタなしの固定幅の ASCII で書き込まれます。数値とバイナリデータ型は、 n ブランクのフィールドに右揃えされます。ここで n は、その型の値を表すのに必要な最大バイト数を表します。文字データ型は、 n ブランクのフィールドで左揃えされます。

注意： 固定幅 ASCII 抽出での最小のカラム幅は 4 バイトです。NULL 値に対して文字列 "NULL" を設定できるようにするためです。たとえば、抽出されたカラムが CHAR(2) で、Temp_Extract_Column_Delimiter が空の文字列 " " に設定されている場合、抽出されたデータの後に 2 つのスペースが置かれます。

ASCII 抽出の間、引用符の使用は次のオプションで制御されます。

| オプション | ASCII 抽出での動作 |
|-------------------------|--------------------|
| Temp_Extract_Quotes | 文字列フィールドを引用符で囲む |
| Temp_Extract_Quotes_All | すべてのフィールドを引用符で囲む |
| Temp_Extract_Quote | 引用符として使用する文字列を指定する |

Temp_Extract_Quote オプションで指定する引用符文字列には、デリミタと同じ制限があります。このオプションのデフォルトは空文字列で、IQ はこれを一重引用符に変換します。

null 値の表現

TEMP_EXTRACT_NULL_AS_ZERO と TEMP_EXTRACT_NULL_AS_EMPTY は、ASCII 抽出での null 値の表現を制御します。TEMP_EXTRACT_NULL_AS_ZERO が ON に設定されている場合、null 値は次のように表示されます。

- 算術型には '0'
- CHAR 文字型と VARCHAR 文字型には " (空の文字列)
- 日付には " (空の文字列)
- 時刻には " (空の文字列)
- タイムスタンプには " (空の文字列)

TEMP_EXTRACT_NULL_AS_EMPTY が ON に設定されている場合は、すべてのデータ型に対して、null 値は " (空の文字列) で表現されます。

上記の引用符は、抽出出力ファイルには含まれないので注意してください。TEMP_EXTRACT_NULL_AS_ZERO と TEMP_EXTRACT_NULL_AS_EMPTY が OFF (デフォルト値) に設定されている場合は、すべての状況において、NULL 値の表現には文字列 'NULL' が使用されます。

TEMP_EXTRACT_NULL_AS_ZERO が ON の場合、ASCII 抽出で CHAR または VARCHAR カラムに対してファイルに書き込まれる文字数は、そのカラムの文字数に等しくなります。4 文字未満のカラムの場合でも同じです。

メッセージロギング

Query_Plan オプションが ON の場合、抽出されたカラムのタイムスタンプ付きのリストが IQ メッセージログに出力されます

参照：

- データ抽出オプションの有効化 (18 ページ)

データ抽出オプションの有効化

データ抽出オプションは、慎重に使用してください。

警告！ 抽出オプションを設定して **SELECT** 文を 2 回実行する場合、2 回目の **SELECT** 文を実行するときに抽出ファイル名を変更しないと、1 回目の **SELECT** の実行時に出力したファイルが 2 回目の **SELECT** の実行時に上書きされます。たとえ 1 秒後であっても 1 週間後であっても、**SELECT** 文を再度実行するたびに、抽出は先頭からやり直されます。ただし、Temp_Extract_Append オプションを ON に設定している場合は除きます。

抽出オプションは接続に対して設定されます。抽出オプションを設定した場合に、別のユーザが同じユーザ ID を使用してデータベースに接続すると、そのユーザに

対してもデータ抽出機能が有効化されます。自分が抽出した出力が、同じ接続を使用する他のユーザによって上書きされる可能性があります。

また、別のユーザが同じユーザ ID でログインした場合、オプションを無効にするまでは、このユーザが実行したクエリの出力は、同じ抽出ファイルに書き込まれます。抽出要求は、ユニークなユーザ ID を使用して実行してください。

1. 残す必要のある既存の出力を別の場所に保存します。
2. 以前に使用した抽出ファイルを削除します。
3. 必要な抽出オプションを設定します。Temp_Extract_Name1 に出力先ファイルパスが設定されているか確認してください。
4. **SELECT** 文を発行して必要なデータを抽出します。
5. 抽出処理が終了したら、Temp_Extract_Name1 を空の文字列にリセットするか、一時的な設定の場合は接続を切断します。

参照：

- 抽出オプション (12 ページ)

データ抽出オプションの例

さまざまなデータ抽出シナリオがあります。

例：単一のディスクファイルへの抽出

次の文は、単一のディスクファイル daily_report.txt への抽出を実行します。

```
SET TEMPORARY OPTION Temp_Extract_Name1 = 'daily_report.txt';
```

```
SET TEMPORARY OPTION Temp_Extract_Name2 = '';
```

```
SELECT ....;
```

```
SET TEMPORARY OPTION Temp_Extract_Name1 = '';
```

SELECT 文の実行前に Temp_Extract_Name2 を空の文字列に設定しています。1つのファイルのみに出力されるようにするためです。

SELECT 文の後で Temp_Extract_Name1 を空の文字列に設定しています。抽出を無効にするためです。抽出を無効にしないと、次回の **SELECT** 文の実行時に daily_report.txt ファイルが上書きされます。

例：追加モードでの抽出

ディスク出力ファイル hourly_report.txt はすでに作成されており、ユーザ sybase に対して書き込みパーミッションが設定されています。次の文は、hourly_report.txt への抽出を実行し、各 **SELECT** 文の出力をファイルの末尾に追加していきます。

```
SET TEMPORARY OPTION Temp_Extract_Append = ON;
```

```
SET TEMPORARY OPTION Temp_Extract_Name1 = 'hourly_report.txt';
```

データベースからデータをエクスポートする方法

```
SET TEMPORARY OPTION Temp_Extract_Name2 = '';  
SELECT ....;  
SELECT ....;  
SELECT ....;  
SET TEMPORARY OPTION Temp_Extract_Name1 = '';
```

3つの **SELECT** 文の出力はいずれも `hourly_report.txt` ファイルに書き込まれます。最後の **SELECT** 文の後で `Temp_Extract_Name1` を空の文字列に設定しています。抽出を無効にするためです。抽出を無効にしないと、次に実行する **SELECT** 文の出力は `hourly_report.txt` の末尾に追加されます。

例：複数のディスクファイルへの抽出

次の文は、`file1.out`、`file2.out`、`file3.out` の各ディスクファイルへの抽出を実行します。

まず、ファイル名のオプションを設定します。

```
SET TEMPORARY OPTION Temp_Extract_Name1 = 'file1.out'; SET TEMPORARY  
OPTION Temp_Extract_Name2 = 'file2.out'; SET TEMPORARY OPTION  
Temp_Extract_Name3 = 'file3.out'; SET TEMPORARY OPTION  
Temp_Extract_Name4 = '';
```

次に、ファイルのサイズを各 **1MB** に制限します。それには、対応する抽出サイズオプションを設定します。

```
SET TEMPORARY OPTION Temp_Extract_Size1 = '1024';  
SET TEMPORARY OPTION Temp_Extract_Size2 = '1024';  
SET TEMPORARY OPTION Temp_Extract_Size3 = '1024';
```

サイズオプションは **KB** (1024 バイト) 単位です。

これらの設定に基づいて、抽出の出力はまず `file1.out` に書き込まれます。そして、次のローを書き込むと `file1.out` のファイルサイズが **1MB** を超えるという時点で、出力は `file2.out` にリダイレクトされます。そして、`file2.out` も満杯(次のローを書き込むと `file2.out` のファイルサイズが **1MB** を超える)という時点で、出力は `file3.out` にリダイレクトされます。IQ がまだ抽出していないローが残っている段階で、`file3.out` のサイズが **1MB** を超えた場合、エラーが通知されます。

参照：

- 抽出処理の制約事項 (20 ページ)

抽出処理の制約事項

次の制約と制限がデータ抽出機能に影響を与えます。

- 抽出は、IQ ストアに格納されたデータに対してのみ実行できます。

- 抽出は、システムテーブルやデータベース間ジョインに対しては実行できません。
- 抽出は、ユーザ定義関数やシステム関数 (システム関数 `suser_id()` と `suser_name()` は除く) を使用したクエリでは実行できません。
- バイナリ **LOAD TABLE** は、常に VARCHAR データからブランクを削除します。VARCHAR データに後続ブランクがある場合、バイナリロードによる挿入処理では、それらのブランクは保持されません。
- VARBINARY データについては、抽出時に後続のゼロが埋め込まれます。たとえば、`varbinary(6)` として宣言されたフィールドに `0x1234` というデータが入っている場合、抽出時にゼロが埋め込まれ、抽出後には `0x123400` としてロードされます。
- 浮動小数点データを正確に再現する場合は、バイナリオプションを使用します。
- テープデバイスは現時点ではサポートされていません。
- q** (クワイエットモード) オプションを指定して **dbisql** (Interactive SQL) を実行するときに、データ抽出コマンドがコマンドファイル内にある場合は、まず **dbisql** の [複数の結果セットを表示] オプションを設定し、これを永続的なものにしておく必要があります。このオプションが設定されていないと、出力ファイルは作成されません。
[複数の結果セットを表示] オプションを設定するには、**dbisql** ウィンドウで [ツール]>[オプション] をクリックし、[複数の結果セットを表示] を選択して [永続化する] をクリックします。

`Temp_Extract_Name1` を設定した場合、**INSERT...SELECT** は実行できません。

イベントは、結果セットを返す文の実行をサポートしません。サーバログは、次のようなエラーを返します。

```
Handler for event 'test_ev' caused SQLSTATE '09W03'  
Result set not permitted in 'test_ev'
```

イベントによってクエリを実行するには、ストアードプロシージャを呼び出すイベントを作成し、そのストアードプロシージャの結果をテンポラリテーブルに挿入します。抽出を使用した場合、テンポラリテーブルは常に空となり、オーバーヘッドはわずかしかな必要ありません。

次に例を示します。

```
CREATE PROCEDURE procl()  
BEGIN  
    SET TEMPORARY OPTION temp_extract_name1 = 'testproc.out';  
    SELECT * FROM iq_table;  
END;  
  
CREATE EVENT "test_ev" ENABLE HANDLER  
BEGIN  
    SELECT * INTO #tmp FROM procl();
```

データベースからデータをエクスポートする方法

```
END;  
TRIGGER EVENT test_ev;
```

参照：

- データ抽出オプションの例 (19 ページ)

LOAD TABLE 文を使用したバルクロード

LOAD TABLE 文を使用すると、テキストファイルまたはバイナリファイルから、既存のデータベーステーブルか、自動的に作成されたかまたはユーザによって定義されたカラムインデックスに効率よくデータをインポートできます。

LOAD TABLE 文を実行するのに必要なパーミッションは、サーバコマンドラインで **-gl** オプションを使用して設定します。それには **-gl all** 設定をおすすめします。この設定は、**start_iq** によって設定されるデフォルト設定です。**-gl all** を設定した場合、**LOAD TABLE** 文を使用するためには、テーブルの所有者であるか、テーブルに対する ALTER または LOAD パーミッションが付与されているか、または ALTER ANY TABLE、LOAD ANY TABLE、ALTER ANY OBJECT のいずれかのシステム権限が付与されている必要があります。また、テーブルに対する書き込みロックも必要です。

大量のデータをロードする場合、通常はコマンドファイルを作成します。

トランザクション処理と LOAD TABLE

IQ テーブルに対して **LOAD TABLE** 文を発行すると、データがロードされる前に自動的にセーブポイントが設定されます。

ロードの完了に成功すると、SAP Sybase IQ はセーブポイントを解放します。ロードに失敗した場合は、セーブポイントまでトランザクションがロールバックします。この方法によって、トランザクションを柔軟にコミットできます。たとえば、2つの **LOAD TABLE** コマンドを発行する場合は、両方のコマンドをコミットしたり、どちらもコミットしないようにしたりできます。

カタログストアテーブルに対して **LOAD TABLE** を発行する場合は、自動セーブポイントは設定されません。ロードに成功すると、自動的にコミットします。ロードに失敗すると、ロールバックします。ロードに成功したカタログストアテーブルはロールバックできません。

フラットファイルからのロード UNIX の例

この例では、明示的にデータを変換する必要はなく、入力カラムの幅が Departments テーブルのカラムの幅と一致することが前提となります。指定された場所にフラットファイル dept.txt が存在する必要があります。

次の文は、ファイル dept.txt から department テーブルのすべてのカラムにデータをロードします。

```
LOAD TABLE Departments  
( DepartmentID, DepartmentName, DepartmentHeadID )  
FROM '/d1/MILL1/dept.txt'
```

ロードのファイル指定要件

FROM 句で、*filename-string* を使用してファイルを指定し、複数の文字列をカンマで区切ります。

ファイルは1つずつ読み込まれ、**FROM** 句で指定された順に左から右へ処理されます。**SKIP** または **LIMIT** の値は、ロードの開始時に適用され、各ファイルには適用されません。

メモリが不足しているなどの理由でロードを完了できない場合は、ロードトランザクション全体がロールバックします。

filename-string は、文字列としてサーバに渡されます。この文字列は、SQL 文字列と同じフォーマット要件に従います。特に、次の点に注意してください。

- 円記号 (¥) が文字 n、x、または ¥ の前に付いている場合は、エスケープ文字とみなされます。このため、Windows システムでディレクトリパスを指定する場合に、円記号の次に n、x、または ¥ が続く場合は、円記号を2つ続けて、1つの円記号を表す必要があります。

ファイル c:¥newinput.dat から employee テーブルにデータをロードするには、次のコマンドを使用します。

```
LOAD TABLE employees
FROM 'c:¥newinput.dat' ...
```

- サーバ側のロード (**LOAD TABLE... FROM** または **LOAD TABLE... USING FILE**) の場合、パス名はクライアントアプリケーションではなく、データベースサーバを起点とします。別のコンピュータのデータベースサーバ上で文を実行している場合、ディレクトリ名はそのサーバマシンのディレクトリを参照し、クライアントマシンのディレクトリは参照しません。ロード用の入力ファイルはサーバマシン上になければなりません。
- クライアント側のデータのロード (**LOAD TABLE ... USING CLIENT FILE**) の場合、パス名はクライアントアプリケーションを起点とする必要があります。ディレクトリ名は、クライアントマシンのディレクトリを参照します。

名前付きパイプを指定するロード

Windows で名前付きパイプを使用してロードする場合は、パイプを記述したプログラムで特殊な方法を使用して、パイプを閉じます。このプログラムは、FlushFileBuffers () を呼び出してから、DisconnectNamedPipe () を呼び出す必要があります。プログラムがこれを行わない場合は、SAP Sybase IQ が hos_io::Read () から例外を報告します。これによって

PIPE_NOT_CONNECTED エラーが発生します。このエラーによって、パイプが停止したのは手順どおりの方法によってであり、制御不可能な接続解除状態になったためではないことが SAP Sybase IQ に通知されます。呼び出しの詳細については、Microsoft のマニュアルを参照してください。

入力データフォーマットを指定するロード

SAP Sybase IQ では、さまざまなロードオプションを指定して、入力ファイルの解釈方法と処理方法、エラーが発生した場合の処理内容を指定できます。

ロードオプションは、どのような順序で指定してもかまいません。

例:引用符を表示するロード

次のように定義されたテーブルを想定します。

```
CREATE TABLE t1 (c1 INT, c2 VARCHAR(20), c3 VARCHAR(20))
```

入力データは次のとおりです。

```
1, apple , fruit1 ,
2, "banana" , "fruit2",
3, " pear " , " fruit3 " ,
```

このクエリを実行すると、このデータのロードの結果が表示されます。

```
SELECT c1, c2, c3, LENGTH(c2), LENGTH(c3) FROM t1
```

次の表に、LOAD TABLE コマンドで使用される **QUOTES** および **STRIP** オプションの値と、各値を指定した場合のこのクエリの結果を示します。各結果は山カッコで囲まれています。

| LOAD TABLE オプション | | SELECT c1, c2, c3, LENGTH(c2), LENGTH(c3) FROM t1 の結果 | | | | |
|------------------|-------|---|-------------|--------------|------------|------------|
| QUOTES | STRIP | c1 | c2 | c3 | length(c2) | length(c3) |
| ON | RTRIM | <1> | <apple> | <fruit1> | <5> | <6> |
| | | <2> | <banana> | <fruit2> | <6> | <6> |
| | | <3> | < pear > | < fruit3 > | <6> | <8> |
| ON | OFF | <1> | <apple> | <fruit1> | <6> | <7> |
| | | <2> | <banana> | <fruit2> | <6> | <6> |
| | | <3> | < pear > | < fruit3 > | <6> | <8> |
| OFF | RTRIM | <1> | <apple> | <fruit1> | <6> | <7> |
| | | <2> | < "banana"> | < "fruit2"> | <9> | <9> |
| | | <3> | < " pear "> | < "fruit3 "> | <9> | <11> |

LOAD TABLE 文を使用したバルクロード

| LOAD TABLE オプション | | SELECT c1, c2, c3, LENGTH(c2), LENGTH(c3) FROM t1 の結果 | | | | |
|------------------|-------|---|------------|-------------|------------|------------|
| QUOTES | STRIP | c1 | c2 | c3 | length(c2) | length(c3) |
| OFF | OFF | <1> | <apple> | <fruit1> | <7> | <8> |
| | | <2> | <"banana"> | <"fruit2"> | <10> | <9> |
| | | <3> | <" pear "> | <"fruit3 "> | <9> | <11> |

結果についての注意

- **QUOTES ON** と **STRIP RTRIM** の場合、c2 のロー 1 の先行スペースと後続スペースの両方が削除されます。
- **QUOTES ON** と **STRIP OFF** の場合、c2 のロー 1 の先行スペースのみが削除されます。
- **QUOTES OFF** と **STRIP RTRIM** の場合、c2 のロー 1 の後続スペースのみが削除されます。
- **QUOTES OFF** と **STRIP OFF** の場合、c2 のロー 1 の先行スペースも後続スペースも削除されません。
- **QUOTES ON** と **STRIP RTRIM** の場合、c2 と c3 のロー 3 の引用符内の先行スペースと後続スペースの両方とも削除されません。

例: 指定したフィールドを省略するロード

Windows の例:

```
LOAD TABLE nn
  (l_orderkey,
   l_quantity ASCII(PREFIX 2),
   FILLER(2),
  FROM 'C:\¥¥iq¥¥archive¥¥mill.txt'
  BYTE ORDER LOW
```

例: 挿入されるローの数を制限するロード

次に示す Windows の例では、ローは省略されずに、1,000,000 個のローが挿入されます。

```
LOAD TABLE lineitem
  (l_shipmode ASCII(15),
   l_quantity ASCII(8),
   FILLER(30))
  FROM 'C:\¥¥iq¥¥archive¥¥mill.txt'
  PREVIEW ON
  LIMIT 1000000
```

例: タブと改行を含めるロード

次の Windows の例では、l_orderkey カラムのカラムデリミタをタブに設定し、ローデリミタを改行 (¥x0a) とそれに続く復帰 (¥x0d) の組に設定します。

```
LOAD TABLE mm
  (l_orderkey '¥x09',
   l_quantity ASCII(4),
   FILLER(6),
   l_shipdate DATE('YYYY/MM/DD'))
FROM 'C:¥¥iq¥¥archive¥¥mill.txt'
ROW DELIMITED BY '¥x0a¥x0d'
```

例: ローを省略するロード

この UNIX の例では、SAP Sybase IQ は入力ファイルから 9,000 個のローを読み込み、最初の 5,000 個を省略して次の 4,000 個のローをロードします。入力ファイルに 8,000 個のローしかない場合は、3,000 個だけロードされます。

```
LOAD TABLE lineitem(
  l_shipmode ASCII(15),
  l_quantity ASCII(8),
  FILLER(30))
FROM '/d1/MILL1/tt.t'
LIMIT 4000
SKIP 5000
PREVIEW ON
```

LOAD TABLE でのローの追加

LOAD TABLE 文は、テーブルの既存のローにファイルの内容を追加します。

既存のテーブルを空にするには、**TRUNCATE TABLE** 文を使用してすべてのローを削除します。

参照:

- クライアントからの直接のデータロード (27 ページ)
- 分割されたテーブルロードに関する考慮事項 (28 ページ)
- メッセージのロードと挿入 (29 ページ)
- 整合性制約違反のメッセージ (29 ページ)

クライアントからの直接のデータロード

SAP Sybase IQ では、**LOAD TABLE USING CLIENT FILE** 文を使用したりリモートデータのバルクロードがサポートされます。**LOAD TABLE USING FILE** は、ローカルサーバ上のデータをロードします。これは、廃止された **iq_bcp** ユーティリティに代わるものです。

注意： クライアントとサーバの両方が、SAP Sybase IQ バージョン 15.0 以降である必要があります。

参照：

- 入力データフォーマットを指定するロード (25 ページ)
- 分割されたテーブルロードに関する考慮事項 (28 ページ)
- メッセージのロードと挿入 (29 ページ)
- 整合性制約違反のメッセージ (29 ページ)

分割されたテーブルロードに関する考慮事項

SAP Sybase IQ は、範囲分割、ハッシュ分割、およびハッシュ範囲分割が行われたテーブルに対する完全な並列バルクロードをサポートしています。

同じ量のデータのロードパフォーマンスは、ロードされるテーブルのタイプによって異なることがあります。分割されていないテーブルは分割されたテーブルよりも高速でロードされます。範囲分割テーブルはハッシュ分割やハッシュ範囲テーブルよりも迅速にロードされます。単一の範囲パーティションへのデータのロードは、分割されていないテーブルへのロードに相当します。ロード速度は、コア数をはじめ、基礎となる I/O システムの帯域幅や物理メモリの量などのさまざまな要因によって決まります。

分割されたテーブルのロードパフォーマンスは、分割キーデータの特性によっても異なります。範囲分割されたテーブルでは、分割キーデータがパーティション順にグループ化されている場合に最大のロードパフォーマンスが得られます。ハッシュ分割およびハッシュ範囲分割されたテーブルでは、分割キーデータの値分布が同一である場合に最大のロードパフォーマンスが得られます。

- 以下は、範囲分割されたテーブルまたはハッシュ範囲分割されたテーブルと、範囲分割キーカラムまたは範囲サブ分割キーカラムへのロードに適用されます。
分割されたテーブルにデータロードする場合は、分割カラムがコマンドのカラムリストの最初のカラムである場合に最高のパフォーマンスが得られます。LOAD 文では、ラージオブジェクト (LOB) カラムなどのその他のカラムより前に分割カラムをロードファイルにリストします。可能であれば、プリロードプロセスを使用して、プライマリファイル内のデータを並べ替えてください。INSERT...LOCATION 文でも同様に、SELECT 文の句にラージオブジェクト (LOB) カラムなどのその他のカラムより前に分割カラムをリストします。
- 以下は、分割されたすべてのテーブルと分割キーカラムまたはサブ分割キーカラムに適用されます。分割カラムの内容を更新しようとすると、次のエラーが発生します。

```
"Updating partition key column on a partitioned table is not allowed."  
(SQLCODE -1009417L, SQLSTATE QCB15, Sybase error code 21055)
```

参照：

- 入力データフォーマットを指定するロード (25 ページ)
- クライアントからの直接のデータロード (27 ページ)
- メッセージのロードと挿入 (29 ページ)
- 整合性制約違反のメッセージ (29 ページ)

メッセージのロードと挿入

データベースオプションとサーバ起動スイッチを使用して、メッセージの挿入とロードを制御できます。

挿入およびロード操作中にメッセージが表示されることがあります。NOTIFY_MODULUS データベースオプションは、ロード中に通知メッセージを表示する頻度のデフォルト値を調節したり、このメッセージを省略したりします。

LOAD コマンドで NOTIFY_MODULUS オプションを指定すると、NOTIFY 設定が上書きされます。

IQMsgMaxSize サーバプロパティと **-iqmsgsz** サーバ起動スイッチによって、メッセージログラッピングとメッセージログファイルのサイズを制御します。

参照：

- 入力データフォーマットを指定するロード (25 ページ)
- クライアントからの直接のデータロード (27 ページ)
- 分割されたテーブルロードに関する考慮事項 (28 ページ)
- 整合性制約違反のメッセージ (29 ページ)

整合性制約違反のメッセージ

LOAD TABLE では、整合性制約違反があった場合のロード動作を制御したり、違反についての情報をログに適宜記録したりすることができます。

- 高速射影 (FP) インデックスでは、継続的な NBit ディクショナリ圧縮が、FP (1)、FP (2)、および FP (3) バイトのディクショナリ圧縮にとって代わります。FP (1)、FP (2)、および FP (3) インデックスは、それぞれ NBit (8)、NBit (16)、および NBit (24) にロールオーバーされます。LOB (文字型とバイナリの両方) データ型と BIT データ型を除くすべてのデータ型は NBit カラムにすることができます。

LOAD TABLE 文を使用したバルクロード

FP_NBIT_IQ15_COMPATIBILITY が OFF の場合、IQ UNIQUE がカラムを Flat FP と NBit のいずれとしてロードするかを決定します。IQ UNIQUE を 0 に設定すると、カラムは Flat FP としてロードされます。IQ UNIQUE 制約が設定されていないカラムは、NBit の自動サイズ制限値に達するまで NBit としてロードされます。

- 新しい多層 HG インデックス構造によって、ロードパフォーマンスは HG インデックスサイズから切り離されます。15.x では、HG インデックスのデータ量の増加にともなって、ロードスループットが低下することがありました。インデックスが大きくなるほど、同じ量のデータのロードにかかる時間が増えました。新しい多層構造によって、HG インデックスサイズとロードパフォーマンスの関係が解消され、スループットが向上します。

CREATE_HG_WITH_EXACT_DISTINCTS オプションは、新しく作成された HG インデックスが多層であるか単層であるかを決定します。このオプションが ON の場合、新しい HG インデックスはすべて単層になります。新しい構造を利用するには、このオプションを OFF に設定します。単層 HG インデックスから多層 HG への変換（およびこの逆の変換）には、`sp_iqrebuildindex` を使用します。

MESSAGE LOG ... ROW LOG オプションで **ONLY LOG** 句を指定すると、ロード時に特定の種類の整合性制約違反の情報をロギングできます。メッセージログファイルでは違反ごと、ローログファイルではローごとに記録されます。**ONLY LOG** 句を指定しなかった場合、これらのファイルには、ロードの開始と完了を示すタイムスタンプのみがロギングされます。

整合性制約違反が記録されるメッセージログファイルとローファイルは、IQ メッセージログファイル (.iqmsg) とは別物です。

ロード時に発生する UNIQUE、NULL、DATA VALUE、FOREIGN KEY の制約違反を無視するかどうかや、違反が最大何件に達するまではロールバックを開始せずに無視するかを指定できます。また、特定の種類の整合性制約違反についての情報を、違反単位とロー単位の両方でロギングすることもできます。メッセージログでは違反単位、ローログではロー単位です。

参照：

- 入力データフォーマットを指定するロード (25 ページ)
- クライアントからの直接のデータロード (27 ページ)
- 分割されたテーブルロードに関する考慮事項 (28 ページ)
- メッセージのロードと挿入 (29 ページ)

MESSAGE LOG の内容とフォーマット

MESSAGE LOG ファイルには、それぞれの整合性制約違反のローとカラムの情報がロギングされています。

各ロードにはタイムスタンプが付いたヘッダ、ロー情報、タイムスタンプが付いたトレーラが含まれています。ヘッダは各ロードにつき 1 回記録されます。トレーラは、文が正常に実行された場合に 1 回記録されます。ロー情報は、ロギングされた整合性制約違反それぞれに対して 1 回記録されます。

ヘッダメッセージのフォーマットは次のとおりです。

```
<datetime load started> Load Table <table-name>: Integrity
Constraint Violations
```

次に例を示します。

```
2009-05-24 23:04:31 Load Table Customers: Integrity Constraint
Violations
```

ロー情報メッセージは次のもので構成されています。

- 整合性制約違反がなかったとしたらこのローがロードされていた、テーブル内でのロー番号
- 検出された整合性制約違反の種類
- スキーマで指定されたカラム

次に例を示します。

```
1267 DATA VALUE 4
3216 UNIQUE 1
3216 NULL 3
3216 NULL 6
9677 NULL 1
```

トレーラメッセージのフォーマットは次のとおりです。

```
<datetime load completed> Load Table <table-name> Completed
```

次に例を示します。

```
2009-05-24 23:05:43 LOAD TABLE Customers: Completed
```

注意： **MESSAGE LOG** ファイルのローの数 (レポートされたエラーの数) が、**IGNORE CONSTRAINT** オプションの制限を超えることがあります。並行して動作する複数のスレッドによってロードが実行されるためです。制約違反の数が指定された制限値を超えたということが、複数のスレッドによって報告される場合があります。

参照：

- ROW LOG の内容とフォーマット (32 ページ)

- MESSAGE LOG と ROW LOG の例 (33 ページ)

ROW LOG の内容とフォーマット

ROW LOG ファイルには、発生した整合性制約違反が記録された各ローのロー ID とデータ値がロギングされています。

ローのデータは、各ローにつき 1 回のみ記録されます。そのローで整合性制約違反が何件発生したかは関係ありません。各ロードについて、ロギングされるメッセージは 3 種類あります。タイムスタンプが付いたヘッダ、ローデータ、タイムスタンプが付いたトレーラです。ヘッダは各ロードにつき 1 回記録されます。トレーラは、文が正常に実行された場合に 1 回記録されます。

ヘッダメッセージのフォーマットは次のとおりです。

```
<datetime load started> Load Table <table-name>: Integrity  
Constraint Violations  
<formatting information>
```

ここで、<formatting information> は、ローデータのフォーマットに使用されている、日付、時刻、日付時刻フォーマットを表します。次に例を示します。

```
2009-05-24 23:04:31 Load Table Customers: Integrity Constraint  
Violations  
Date Format: yyyy/mm/dd  
Time Format: hh:mm:ss  
Datetime format: yyyy/mm/dd hh:mm:ss
```

ローデータメッセージは次のもので構成されています。

- 整合性制約違反がなかったとしたらこのローがロードされていた、テーブル内でのロー番号
- カンマ、またはユーザ指定の **LOG DELIMITED BY** セパレータで区切られる、ロー内のデータ値

次に例を示します。

```
3216 #Jones John#NULL#NULL#S#1945/01/12#NULL#
```

ローデータメッセージでのデータ値のフォーマットは、次の規則に応じて決まります。

- データ型が **VARBINARY** または **BINARY** の場合は、データは ASCII の 16 進文字で表されます。
- **DATE** 値は、**DATE_FORMAT** データベースオプションで指定されたフォーマットで表されます。デフォルトフォーマットは **YYYY-MM-DD** です。
- **DATETIME** 値と **TIMESTAMP** 値は、**TIMESTAMP_FORMAT** データベースオプションで指定されたフォーマットで表されます。デフォルトは **YYYY-MM-DD HH:NN:SS.SSS** です。

- TIME 値は、TIME_FORMAT データベースオプションで指定されたフォーマットで表されます。デフォルトは HH:NN:SS.SSS です。
- NULL 値は、トークン NULL で表されます。

注意： ローデータメッセージには、フィルターフィールドは含まれません。

トレーラメッセージのフォーマットは次のとおりです。

```
<datetime load completed> Load Table <table-name>: Completed
```

次に例を示します。

```
2009-05-24 23:05:43 Load Table Customers: Completed
```

注意： MESSAGE LOG ファイルに記録された個々のエラーの数と、ROW LOG ファイルに記録されたローの数は一致しない場合があります。両者のローの数の違いは、複数のスレッドによって実行される、ロードの並行処理によるものです。制約違反の数が指定された制限値を超えたということが、複数のスレッドによって報告される場合があります。

参照：

- MESSAGE LOG の内容とフォーマット (31 ページ)
- MESSAGE LOG と ROW LOG の例 (33 ページ)

MESSAGE LOG と ROW LOG の例

MESSAGE LOG ファイルと **ROW LOG** ファイルの内容とフォーマットを例示します。

この文では、ロードされるテーブルを作成します。

```
CREATE TABLE Customers(name VARCHAR(80) NOT NULL,
age TINYINT NULL,
sex CHAR(1) NOT NULL,
marital_status CHAR(1) NULL,
birthdate DATE NOT NULL,
credit_card VARCHAR(20)NOT NULL)
```

この文では、Customers テーブルにデータをロードします。

```
LOAD TABLE Customers ...
IGNORE CONSTRAINT UNIQUE 200
MESSAGE LOG 'msg.log' ROW LOG 'row.log'
ONLY LOG UNIQUE, NULL, DATA VALUE
LOG DELIMITED BY '#'
```

ディスクファイルから生データがロードされます。

```
Jones John, 19, M, S, 06/19/83, CC
Cleven Bill, 56, M, OSIDJFJ, 02/23/43, CC
Jones John, 339, M, NULL, 01/12/45, NULL
NULL, 55, F, M, 10/02/37, ST
```

LOAD TABLE 文を使用したバルクロード

LOAD TABLE が完了すると、**MESSAGE LOG** ファイル msg.log は次のようになります。

```
2009-05-24 23:04:31 LOAD TABLE Customers: Integrity Constraint
Violations
1267 DATA VALUE 4
3216 UNIQUE 1
3216 NULL 6
9677 NULL 1
2009-05-24 23:05:43 LOAD TABLE Customers Completed
```

ROW LOG ファイル row.log は次のようになります。

```
2009-05-24 23:04:31 LOAD TABLE Customers Integrity Constraint
Violations
Date Format: yyyy/mm/dd
Time Format: hh:mm:ss
Datetime format: yyyy/mm/dd hh:mm:ss

1137 #Jones John#19#M#S#1983/06/19#CC#
1267 #Cleven Bill#56#M#OSIDJFJ#1943/02/23#CC#
3216 #Jones John#NULL#NULL#S#1945/01/12#NULL#
9677 #NULL#55#F#M#1937/10/02#ST#

2009-05-24 23:05:43 LOAD TABLE Customers Completed
```

参照：

- MESSAGE LOG の内容とフォーマット (31 ページ)
- ROW LOG の内容とフォーマット (32 ページ)

BINARY ロード形式

SAP Sybase IQ へのデータロードを高速化するには、バイナリフォーマットでデータファイルを作成し、**LOAD TABLE** の **FORMAT BINARY** および **BINARY** カラム指定句を使用してデータをロードします。

これらのバイナリフォーマットでデータファイルを作成して、対応するデータ型のカラムにロードします。ほとんどの場合、SAP Sybase IQ はプラットフォームに固有のバイナリフォーマットを使用します。例外として、次のデータ型は SAP Sybase IQ に固有のバイナリフォーマットを使用します。

- DATE
- TIME
- DATETIME
- NUMERIC

BINARY ロード形式とロード効率

SAP Sybase IQ の BINARY ロード形式は、固定幅のフォーマットです。

一般的に、固定幅のロードは可変幅のロードよりも高速です。ロードロジックがカラムとローの長さを認識している場合、データの処理効率が向上します。幅が一定でないカラムとローをデリミタによって分割すると、ロード中に入力データをスキャンしてデリミタを探す時間がかかるようになります。

IQ の BINARY ロード形式は、固定幅のフォーマットです。ロードでは、テーブルの定義の情報から各カラムの幅と各ローの長さを確認できます。

注意： BINARY ロード形式は、ビッグエンディアンとリトルエンディアンを区別し、ネイティブバイナリデータ型を使用してデータを表現します。

参照：

- オペレーティングシステムのネイティブデータ型 (36 ページ)
- DATE (36 ページ)
- TIME (37 ページ)
- TIMESTAMP (38 ページ)
- NUMERIC と DECIMAL (38 ページ)
- NULL 値のロード (40 ページ)

オペレーティングシステムのネイティブデータ型

一部のデータ型のデータは、オペレーティングシステムのネイティブバイナリフォーマットで保存され、そのフォーマットで直接データファイルに書き込むことができます。SAP Sybase IQ は、変換することなく各バイト数を関連データ型に直接読み込みます。

- BIT (1 バイト)
- TINYINT (1 バイト)
- SMALLINT (2 バイト)
- INT/UNSIGNED INT (4 バイト)
- BIGINT/UNSIGNED BIGINT (8 バイト)
- FLOAT (4 バイト)
- DOUBLE (8 バイト)
- CHAR/VARCHAR (文字データ)
- BINARY/VARBINARY (バイナリデータ)

デフォルトでは、VARCHAR カラムと VARBINARY カラムが、**LOAD TABLE *column-spec*** により指定された値までのバイト数で読み取られます。

参照：

- BINARY ロード形式とロード効率 (35 ページ)
- DATE (36 ページ)
- TIME (37 ページ)
- TIMESTAMP (38 ページ)
- NUMERIC と DECIMAL (38 ページ)
- NULL 値のロード (40 ページ)

DATE

DATE カラムデータは、0000-01-01 以降の日数を表す 4 バイト (32 ビットの符号なし整数) として SAP Sybase IQ に格納されます。

ある年、月、日に対応する暦日を SAP Sybase IQ のバイナリフォーマットに変換するには、次の式を使用します。

```
year = current_year - 1;  
days_in_year_0000 = 366;  
binaryDateValue = (year * 365)
```

```
+ (year / 4)
- (year / 100)
+ (year / 400)
+ days_in_year_0000
+ day_of_current_year
-1;
```

上記の式の `day_of_current_year` 値には、たとえば 2 月 12 日の場合、第 43 日と指定します。

参照：

- BINARY ロード形式とロード効率 (35 ページ)
- オペレーティングシステムのネイティブデータ型 (36 ページ)
- TIME (37 ページ)
- TIMESTAMP (38 ページ)
- NUMERIC と DECIMAL (38 ページ)
- NULL 値のロード (40 ページ)

TIME

TIME データは、マイクロ秒数 (つまり、 $1.0e-6$ 秒) を表す 64 ビットの符号なし数量として格納されます。

ある時間、分、秒、マイクロ秒 (*usec*) に対応するマイクロ秒数は、次のように計算します。

```
binaryTimeValue = (hour * 3600 + minute * 60 + second + microsecond )
* 1000000
```

参照：

- BINARY ロード形式とロード効率 (35 ページ)
- オペレーティングシステムのネイティブデータ型 (36 ページ)
- DATE (36 ページ)
- TIMESTAMP (38 ページ)
- NUMERIC と DECIMAL (38 ページ)
- NULL 値のロード (40 ページ)

TIMESTAMP

TIMESTAMP データは、マイクロ秒数を表す 64 ビットの符号なし整数として格納されます。

ある年、月、日、時間、分、秒、マイクロ秒に対応するバイナリ TIMESTAMP 値は、次のように計算できます。

この日付の *binaryDateValue* を計算します。

この時刻の *binaryTimeValue* を計算します。

```
binaryDateTimeValue = binaryDateValue *  
86400000000 + binaryTimeValue
```

参照：

- BINARY ロード形式とロード効率 (35 ページ)
- オペレーティングシステムのネイティブデータ型 (36 ページ)
- DATE (36 ページ)
- TIME (37 ページ)
- NUMERIC と DECIMAL (38 ページ)
- NULL 値のロード (40 ページ)

NUMERIC と DECIMAL

NUMERIC データ型と DECIMAL データ型のフォーマットは、精度の関数として変化します。

値の右側には、値のフルスケールまで 0 を埋め込む必要があります。さらに、左側にも 0 を埋め込む必要がありますが、埋め込みはバイナリプログラミングにより自動的に行われます。値に 0 が埋め込まれると、小数点が削除されます。たとえば、値 12.34 は次のようになります。

- NUMERIC(4,2): 1234
- NUMERIC(6,4): 123400
- NUMERIC(8,4): 00123400
- NUMERIC(12,6): 000012340000
- NUMERIC(16,8): 0000001234000000

値に 0 を埋め込み、小数点を削除した後、次の規則が適用されます。

- 精度が 4 以下の場合、バイナリフォーマットは、2 バイトの整数量に対するオペレーティングシステムのネイティブバイナリフォーマットと同じです。
- 精度が 5 から 9 までの場合、バイナリフォーマットは、4 バイトの整数量に対するオペレーティングシステムのネイティブバイナリフォーマットと同じです。
- 精度が 10 から 18 までの場合、バイナリフォーマットは、8 バイトの整数量に対するオペレーティングシステムのネイティブバイナリフォーマットと同じです。
- 精度が 19 以上の場合、次の C++ 構造定義を使用する特別なフォーマットが存在します。

```
struct {
  unsigned char sign; // sign 1 for +, 0 for -
  unsigned char ndig; // # digits
  unsigned char exp; // exponent
  unsigned char erracc; // should be 0
  unsigned short digits[80];
};
```

値がゼロでないかぎり、指数は excess-80 形式です。「ゼロ」値は次のように表現されます。

```
sign = 1
ndig = 0
erracc = 0
exp = 0
```

最大指数値は 159 です。サポートされる最大桁数は 288 です。“digits[0]”には、最下位桁が含まれます。数字は、「符号なし短整数型」(2 バイト) 数量あたり 2 桁のパックされた表現に格納されます。ある数の場合、次のようになります。

- lower order digit = digit[i] & 0x00FF
- high order digit = digit[i] & 0xFF00

たとえば、NUMERIC(20) として表現される値 100 を考えてみます。この値のバイナリレイアウトは次のようになります。

```
0x0101 0x5000 0x0064 0x0000 0x0000 .....
```

```
Sign = 0x01
Number digits = 0x01
Exponent = 0x50
Erracc = 0x00
Digits = 0x0064
```

別の例として、値 32769 を考えてみます。

```
0x0102 0x5000 0x0ad1 0x0003 0x0000 0x0000 ....
```

```
Sign = 0x01
Number digits = 0x02
Exponent = 0x50
Erracc = 0x00
Digits = 0x0ad1 0x0003
```

BINARY ロード形式

これらの数を基数 10 に変換すると、次のようになります。

```
0x0ad1 = 2769  0x0003 = 3
```

参照：

- BINARY ロード形式とロード効率 (35 ページ)
- オペレーティングシステムのネイティブデータ型 (36 ページ)
- DATE (36 ページ)
- TIME (37 ページ)
- TIMESTAMP (38 ページ)
- NULL 値のロード (40 ページ)

NULL 値のロード

NULL 値を挿入する最も適切な方法は、入力ファイルに NULL BYTE を使用し、**LOAD TABLE** 文のカラム指定に **WITH NULL BYTE** を指定することです。

このためには、入力ファイルの各データフィールドの終わりを "x00" か "x01" にします。

入力ファイルの各データフィールドの終わりを "x01" とすると、ロードでカラムに NULL が挿入されます。次に例を示します。

```
create table d1 ( c1 date );
load table d1 ( c1 binary with null byte ) from 'filename' quotes off
escapes off format binary;
```

ロード入力ファイルの内容が 000b32cb00000b32cc00 の場合、テーブルに 2 つのローがロードされます。最初のローは、2009 年 5 月 7 日、2 番目のローは 2009 年 5 月 8 日です。入力ファイルの各バイナリ日付の後に NULL BYTE が追加されています。最初のローに NULL をロードする場合は、入力ファイルの NULL BYTE の値を "x01" に変更します。

```
000b32cb01000b32cc00
```

別の例として、値 32769 を NUMERIC (20) カラムにロードする場合、入力ファイルの内容は次のようになります。

```
0x0102 0x5000 0x0ad1 0x0003 0x0000 0x00
```

これには NULL BYTE が含まれています。

NUMERIC (19,2) と定義されたカラムに 23456789012345678.12 をロードする場合、ロード入力ファイルの内容は次のようになります。

```
0x0106 0x4f00 0x04b0 0x162e 0x04d2 0x1ed2 0x0d80 0x0002 0x0000 0x00
```

数字の後には、NULL BYTE (0x00) が続きます。

この数字を表す構造の桁配列には、7つ (0 ~ 6) の符号なし短整数型が含まれています。"digits[0]" には、最下位桁が含まれます。

```
digits[0] = 0x04b0 (decimal 120)
digits[1] = 0x162e (decimal 5678)
digits[2] = 0x04d2 (decimal 1234)
digits[3] = 0x1ed2 (decimal 7890)
digits[4] = 0x0d80 (decimal 3456)
digits[5] = 0x0002 (decimal 2)
digits[6] = 0x0000
```

カラム指定の NULL の部分は、テーブルのカラムにデータをロードするときに、特定の入力値を NULL として処理する方法を指定します。NULL として処理される文字には、BLANKS、ZEROS、または定義したその他のリテラルのリストなどがあります。NULL 値を指定するか、またはソースファイルから NULL 値を読み込む場合は、ロード先のカラムに NULL を格納できる必要があります。

ZEROS は次のように解釈されます。

- 入力データがすべてバイナリのゼロ (文字のゼロではない) の場合、カラムが NULL に設定されます。
- 入力データが文字のゼロの場合は、次のようになります。
 - NULL(ZEROS) を指定しても、カラムに NULL が設定されることはない。
 - NULL('0') を指定すると、カラムに NULL が設定される。たとえば、次をロードします。

```
CREATE TABLE t1 ( c1 INT, c2 INT );
```

ビッグエンディアンバイト順を使用する入力データファイルを表示します。

```
od -x data.inp
3030 3030 0000 04d2
```

次の行を実行します。

```
LOAD TABLE t1 ( c1 ASCII(4) NULL( '0000' ),
                c2 BINARY )
FROM 'data.inp'
FORMAT BINARY
QUOTES OFF
ESCAPES OFF;
```

次のような結果になります。

```
SELECT * FROM t1;
c1      c2
NULL    1234
```

- 入力データがバイナリのゼロ (全ビットがクリア) の場合は、次のようになります。
 - NULL(ZEROS) を指定すると、カラムに NULL が設定される。

BINARY ロード形式

- NULL('0') を指定しても、カラムに NULL が設定されることはない。たとえば、次をロードします。

```
CREATE TABLE t1 ( c1 INT, C2 INT );
```

ビッグエンディアンバイト順を使用する入力データファイルを表示します。

```
od -x data.inp
0000 0000 0000 04d2
```

次の行を実行します。

```
LOAD TABLE t1 ( c1 ASCII(4) NULL( zeros ),
                c2 BINARY )
FROM 'data.inp'
  FORMAT BINARY
  QUOTES OFF
  ESCAPES OFF;
```

次のような結果になります。

```
SELECT * FROM T1;
c1      c2
NULL    1234
```

別の例として、**LOAD TABLE** 文に col1 date('yymmdd') null(zeros) が記述されていて、ロードするデータが 000000 である場合は、000000 を DATE(4) に変換できないことを示すエラーが表示されます。データが 000000 である場合に、**LOAD TABLE** で NULL 値が col1 に挿入されるようにするには、NULL 句を null('000000') のように記述するか、データを同等のバイナリのゼロに修正して NULL(ZEROS) を使用します。

BINARY ロード形式で NULL をロードする別の方法として、ロード先カラムが NULL 値を受け入れる場合に、**LOAD TABLE** 文のカラムにデータを指定しない方法があります。次に例を示します。

```
CREATE TABLE t1 ( c1 INT, c2 INT );
LOAD TABLE T1 ( c2 BINARY ) FROM 'data.inp'
  FORMAT BINARY
  QUOTES OFF
  ESCAPES OFF;
```

```
SELECT * FROM T1;
c1      c2
NULL    1234
NULL    1234
```

ビッグエンディアンバイト順を使用する入力データファイルを表示します。

```
od -x data.inp
0000 04d2 0000 04d2
```

参照：

- BINARY ロード形式とロード効率 (35 ページ)

- オペレーティングシステムのネイティブデータ型 (36 ページ)
- DATE (36 ページ)
- TIME (37 ページ)
- TIMESTAMP (38 ページ)
- NUMERIC と DECIMAL (38 ページ)

BINARY ロード形式

INSERT 文の使用

INSERT 文を使用すると、フラットファイルに最初にデータを記述しなくても、データを挿入できます。

このコマンドを使用して、次の処理を実行できます。

- 指定した一連の値をロー単位で挿入する。
- データベーステーブルからデータを直接挿入する。

ロー単位での指定した値の挿入

指定した値をテーブルにロー単位で追加するには、INSERT 文の構文 1 を使用します。SAP Sybase IQ は、指定された最初の値を指定された最初の列に挿入し、指定された 2 番目の値を 2 番目の列に挿入し、以下同様に処理します。

列名のリストを指定しない場合は、列の作成順 (SELECT* を使って取り出すときと同じ順序) にテーブル列の中に値を挿入します。SAP Sybase IQ は、テーブルに空きがある場合には必ずローを挿入します。

値には、NULL、任意の正数と負数、またはリテラルを使用できます。

- CHAR、VARCHAR、DATE、TIME、TIMESTAMP または DATETIME 列の値を一重引用符や二重引用符で囲みます。引用符を含む値を指定するには、"Smith's" のように、その引用符の外側を別の引用符の組で囲みます。
- DATE、TIME、TIMESTAMP、または DATETIME 列の場合は、特定のフォーマットを使用します。

注意： TIMESTAMP と DATETIME のデータ型は同一です。

列に NULL が許可されている場合に、ローの列の一部にのみ値を指定すると、値が指定されなかった列には NULL が挿入されます。

NULL 値を指定する場合は、ロード先の列で NULL が許可されている必要があります。許可されていない場合は、INSERT が拒否されて、メッセージログにエラーメッセージが出力されます。SAP Sybase IQ の列では、デフォルトで NULL が許可されていますが、CREATE TABLE 文の列定義で NOT NULL を指定するか、プライマリキーなどの別の方法を使用して、この設定を変更できます。

次の例では、lineitem テーブルの l_shipdate 列に 1995-06-09 が、l_orderkey 列に 123 が、それぞれ追加されます。

INSERT 文の使用

```
INSERT INTO lineitem
  (l_shipdate, l_orderkey)
VALUES ('1995-06-09', 123)
```

データローを多数挿入する場合は、選択したローをデータベースから直接挿入するか、または **LOAD TABLE** 文を使用してフラットファイルからデータをロードする方が、値をロー単位で挿入するよりも効率的です。少数のローに値を挿入するのではなく、サーバへのアクセス回数が1回で済むように、少数の **union** を使用して **SELECT** 文を使用するようにしてください。

INSERT VALUES を使用すると複数のローをサポートできます。

次に例を示します。

```
INSERT INTO lineitem(l_shipdate, l_orderkey)
VALUES ('1995-06-09', 123),
('2001-03-28', 300),
('2010-04-01', 413);
```

参照：

- データベースからの指定したローの挿入 (46 ページ)

データベースからの指定したローの挿入

一般的な **SELECT** 文の結果に基づいてデータのローをいくつでも挿入できます。

現在のデータベースの他のテーブル、または特殊データストアとして定義されたデータベースから SAP Sybase IQ にデータを挿入する場合は、**INSERT** 文を使用します。

効率を最大化するには、1つの **INSERT** 文でできるだけ多くのローを挿入します。最初の挿入後にさらにローを挿入するには、別の **INSERT** 文を使用します。

他の SQL データベースと同様に、SAP Sybase IQ は、ロード先カラムリスト内で指定されたカラムの順番と選択リストを照合して、データを挿入します。つまり、選択リスト内の最初のカラムのデータは最初のロード先カラムに挿入され、以下同じように処理されます。**INSERT SELECT** と **INSERT VALUES** でロード先カラム名を省略した場合は、SAP Sybase IQ によってデータがカラムに挿入される順番は、カラムが作成された順番と同じになります。

挿入先のテーブルは、現在接続しているデータベース内にある必要があります。SAP Sybase IQ はデータをロード先カラムのすべてのインデックスに挿入します。

選択リスト内にあるテーブルのカラムのデータ型は同じか、または互換性がある必要があります。つまり、選択した値はテーブルのカラムのデータ型の値であるか、またはそのデータ型に変換可能でなければなりません。

この形式の **INSERT** 文では、いずれかの **insert-load-options** を指定できます。

例

この例では、テーブル `lineitem` から同じデータベース内の別のテーブル `partsupp` へデータを挿入する場合を示します。

ソースカラム `l_quantity` のデータは、ロード先カラム `ps_availqty` に挿入されます。

```
INSERT INTO partsupp(ps_availqty)
SELECT l_quantity FROM lineitem
```

参照：

- ロー単位での指定した値の挿入 (45 ページ)

別のデータベースからの挿入

以下のアクセス可能なデータベースのテーブルからデータを挿入できます。

- 現在接続しているデータベースの IQ ストアまたはカタログストアのテーブル。
- Adaptive Server® Enterprise データベース内のテーブル。
- リモートサーバ上のデータベースのテーブルに対応する、現在接続しているデータベースのプロキシテーブル。

Adaptive Server Enterprise からの直接のデータ挿入

Adaptive Server (ASE) または SQL Server データベースからデータを挿入するには、**INSERT** 文の **LOCATION** 構文を使用します。

SAP Sybase IQ へデータを直接挿入するには、次のすべての条件が整っている必要があります。

- SAP Sybase のコネクティビティライブラリをシステムにインストールして、プラットフォームのロードライブラリパス環境変数をインストール先ディレクトリに設定する。
- 接続先の Adaptive Server サーバがローカルマシン上の `interfaces` ファイルに登録されている。
- ソースデータベースである ASE または SAP Sybase IQ データベースの読み込みパーミッション、およびターゲットデータベースである SAP Sybase IQ データベースの INSERT パーミッションがある。

1. 同じユーザ ID とパスワードを使用して、Adaptive Server と SAP Sybase IQ データベースの両方に接続します。
2. SAP Sybase IQ データベースで、次の文を発行します。

```
INSERT INTO iq_table
LOCATION 'ase_servername.ase_dbname'
```

INSERT 文の使用

```
{ SELECT col1, col2, col3, ...  
FROM owner.ase_table }
```

3. COMMIT を発行して、挿入を実行します。

SAP Sybase IQ がリモートサーバに接続するときに、**CREATE EXTERNLOGIN** でリモートログインが作成されており、**CREATE SERVER** 文でリモートサーバが定義されている場合、**INSERT...LOCATION** は現在の接続のユーザ ID にリモートログインを使用することもできます。

注意：この方法で、選択したカラムを2つの SAP Sybase IQ データベース間で移動することもできます。

ASE の text データと image データのロード

SAP Sybase IQ は Adaptive Server のデータ型である TEXT をサポートしていませんが、長さが 255 バイトを超える IQ の CHAR または VARCHAR カラム、または LONG VARCHAR カラムから **INSERT...LOCATION** (構文 3) を実行できます。また、ASE データベースのデータ型 TEXT のカラムからも実行可能です。ASE の TEXT および IMAGE カラムは、SAP Sybase IQ が内部変換をサポートしていれば、他の SAP Sybase IQ データ型のカラムに挿入できます。**INSERT...LOCATION** では **SELECT** 文での変数の使用はサポートされていません。デフォルトでは、リモートデータカラムの内容が 2GB を超えている場合、SAP Sybase IQ は暗黙的にカラム値を 2GB にトランケートします。

非構造化データ分析機能を使用するには、正規のライセンスを取得しておく必要があります。

SELECT 文を区切る一重引用符の代わりに中カッコ {} を使用することもできます。なお、中カッコは、ODBC 標準ではエスケープシーケンスの開始と終了を表すため、ODBC のコンテキストではエラーが発生する可能性があります。

例

次のコマンドを実行すると、detroit サーバ上の SAP Sybase IQ データベース iq11db.dba にある lineitem テーブルの l_shipdate カラムと l_orderkey カラムから、現在のデータベースの lineitem テーブルの対応するカラムにデータが挿入されます。

```
INSERT INTO lineitem  
  (l_shipdate, l_orderkey)  
  LOCATION 'detroit.iq11db'  
  { SELECT l_shipdate, l_orderkey  
  FROM lineitem }
```

- ロード先カラムの名前とソースカラムの名前は違ってよい。
- 指定された最初のソースカラムが、指定された最初のターゲットカラムに挿入され、以下同じように処理されるため、カラムを指定する順序は重要である。

- **INSERT** コマンド内で **SELECT** 文の述部を使用して、テーブル内の特定のローからのみデータを挿入できる。

例

この例では、挿入されるカラムは前の例と同じですが、`l_orderkey` が 1 のローのみが挿入されます。また、この例では、TDS パケットサイズを 512 バイトと指定しています。

```
INSERT INTO lineitem
  (l_shipdate, l_orderkey)
LOCATION 'detroit.iqdb'
PACKETSIZE 512
{ SELECT l_shipdate, l_orderkey
FROM lineitem
WHERE l_orderkey = 1 }
```

INSERT 文の使用

対話型データインポート

少量のデータを挿入する場合は、Interactive SQL で **INSERT** 文を使用して対話的に値を入力することもできます。

たとえば、次のコマンドを使用すると、1つのローに対して、リストされた値を一度に挿入できます。

```
INSERT INTO T1  
VALUES ( ... )
```

注意： Interactive SQL の [データ] メニューの [インポート] オプションは使用しないでください。このオプションは、SAP Sybase IQ データベースの使用時にはサポートされません。

エンディアンフォーマットが異なるシステム間でのデータの移動

ビッグエンディアンフォーマットのデータベースからリトルエンディアンフォーマットのデータベースにデータを移動できます。

前提条件

注意：作業を始める前に、データベーススキーマとテーブルスキーマを取得する手段があることを確認してください。

次の例では、`lineitem` という名前のテーブルをロードし、UFS (ファイルシステム) の `lineitem_binary.inp` という名前の抽出ファイルを特定します。

システムの最大ファイルサイズについては、オペレーティングシステムのマニュアルで確認してください。たとえば、Sun Solaris x64 の抽出ファイルの最大サイズは、512GB です。

手順

この手順では、テーブル定義を移動しますが、再作成が必要なストアードプロシージャやイベントなどのデータベースオブジェクトの移行は行いません。

たとえば、Sun64 SPARC システムで作成された SAP Sybase IQ データベースは、ビッグエンディアン (最上位バイトを先頭に配置) フォーマットのバイナリデータを格納します。Sun Solaris x64 はリトルエンディアンシステムであるため、Sun64 SPARC で作成された SAP Sybase IQ データベースを **ALTER DATABASE UPGRADE** を使用してアップグレードし、Sun Solaris x64 システムで実行することはできません。

エンディアン構造が異なるハードウェアプラットフォーム間でデータベースのデータを移動するには、以下を実行する必要があります。

- ソースプラットフォームからデータベーススキーマ (テーブル、インデックスなど) をコピーします。
- ターゲットプラットフォームで新しいデータベースを作成します。
- ソースデータベースからバイナリデータのダンプを実行します。
- 新しいターゲットデータベースにデータをロードします。

1. 抽出オプションを有効にします。

```
SET TEMPORARY OPTION Temp_Extract_Name1 =  
'lineitem_binary.inp'
```

エンディアンフォーマットが異なるシステム間でのデータの移動

```
SET TEMPORARY OPTION Temp_Extract_Name2 = ''
```

2. lineitem テーブルのバイナリ抽出ファイルを設定します。

```
SET TEMPORARY OPTION Temp_Extract_Binary = 'on'
```

```
SET TEMPORARY OPTION Temp_Extract_Swap = 'off'
```

3. lineitem_binary.inp ファイルに出力を配置します。

```
SELECT * FROM lineitem
```

4. 抽出オプションを無効にします。

```
SET TEMPORARY OPTION Temp_Extract_Name1 = ''
```

5. ターゲットシステムにデータベースのコピーを作成します。

6. lineitem テーブルが次のように定義されている場合、以下のように lineitem テーブルをロードします。

```
LOAD TABLE lineitem
( l_orderkey      BINARY WITH NULL BYTE,
  l_partkey       BINARY WITH NULL BYTE,
  l_suppkey       BINARY WITH NULL BYTE,
  l_linenumbers   BINARY WITH NULL BYTE,
  l_quantity      BINARY WITH NULL BYTE,
  l_extendedprice BINARY WITH NULL BYTE,
  l_discount      BINARY WITH NULL BYTE,
  l_tax           BINARY WITH NULL BYTE,
  l_returnflag    BINARY WITH NULL BYTE,
  l_linestatus    BINARY WITH NULL BYTE,
  l_shipdate      BINARY WITH NULL BYTE,
  l_commitdate    BINARY WITH NULL BYTE,
  l_receiptdate   BINARY WITH NULL BYTE,
  l_shipinstruct  BINARY WITH NULL BYTE,
  l_shipmode      BINARY WITH NULL BYTE,
  l_comment       BINARY WITH NULL BYTE )
FROM 'C:\mydata\lineitem_binary.inp'
FORMAT BINARY
STRIP OFF
QUOTES OFF
ESCAPES OFF
PREVIEW ON
BYTE ORDER HIGH;
COMMIT
```

特に次の2つの句に注意してください。

- **BINARY WITH NULL BYTE** はバイナリファイルのロード時に必要になります。
- **BYTE ORDER HIGH** は、元のデータが格納されたシステムからのバイト順序を指定します。この例のソースデータベースは、ビッグエンディアンプラットフォームであるため、このデータにはバイト順序 **HIGH** が必要になります(リトルエンディアンデータベースにはバイト順序 **LOW** が必要になります)。

エンディアンフォーマットが異なるシステム間でのデータの移動

マルチプレックスデータベースをロードする場合、すべてのファイル名に絶対パス(完全修飾パス)を使用します。相対パス名は使用しないでください。

エンディアンフォーマットが異なるシステム間でのデータの移動

プライマリキーカラムと外部キーカラムへの挿入

他のカラムと同様にプライマリキーカラムと外部キーカラムにデータをロードまたは挿入します。

プライマリキーに挿入すると、SAP Sybase IQ は、各値がユニークかどうかをチェックします。各値がユニークではない場合は、エラーが発生します。

プライマリキーカラムと外部キーカラムへの挿入

ラージオブジェクトデータのロードまたは抽出

ラージオブジェクトデータのロードまたは抽出は、非構造化データ分析と関連があります。非構造化データ分析機能を使用するには、正規のライセンスを取得しておく必要があります。

ラージオブジェクトデータのロードまたは抽出

挿入時のデータ変換

SAP Sybase IQ データベースに入力するデータは、さまざまなソースから取り込まれます。

SAP Sybase IQ のデータ型と正確に一致しないデータもあります。このため、データの変換が必要になることがあります。データは、明示的または暗黙的に変換されます。たとえば、INT データを CHAR カラムに挿入する場合は、明示的に変換します。

暗黙的な変換は次の場合に行われます。

- 同じデータベースの別のカラムから選択したデータを挿入する
- 別のデータベースから選択したデータをインポートする
- フラットファイルからデータをロードする

明示的な変換が必要な場合、変換を指定する方法は、フラットファイルからロードするか、選択されたローを挿入するかによって異なります。

- **LOAD TABLE** 文で、*column-spec* にフォーマットを指定することによって、明示的にデータを変換する。
- **INSERT** 文で、**SELECT** 文または **VALUES** でデータ変換関数 **CAST**、**CONVERT**、および **DATEPART** を使用して、明示的にデータを変換する。

SAP Sybase IQ のデータ型の多くは、SAP Sybase SQL Anywhere® および Adaptive Server の同名のデータ型と完全な互換性がありますが、いくつかの相違点もあります。

バージョン間の互換性を確保するために、いくつかのデータ型は他のデータ型の同意語として定義されています。

- DECIMAL は NUMERIC の同意語です。
- INTEGER は INT の同意語です。
- DATETIME は TIMESTAMP の同意語です。
- FLOAT (*precision*) は、REAL または DOUBLE の同意語です (*precision* の値によって決まります)。Adaptive Server では、*precision* が 15 以下の場合 REAL が使用され、*precision* が 15 よりも大きい場合は DOUBLE が使用されます。SAP Sybase IQ と SQL Anywhere の場合、カットオフ値はプラットフォームによって異なりますが、すべてのプラットフォームでカットオフ値は 22 より大きい値になります。
- MONEY は、Adaptive Server の NUMERIC(19,4) と互換性のある同意語で、NULL を許容します。

挿入時のデータ変換

- SMALLMONEY は、Adaptive Server の NUMERIC(10,4) と互換性のある同意語で、NULL を許容します。

標準データ型と同じように同意語を使用できます。同意語が存在する場合、データは内部で標準データ型として格納されます。エラーメッセージには同意語ではなく標準名が表示されます。

注意： デフォルトでは、SAP Sybase IQ は入力データをバイナリ (数値データ) とみなし、バイナリとして挿入します。ただし、入力カラム長 (バイト) は宛先カラム長 (バイト) と一致する必要があります。一致しない場合、挿入は失敗するか、または予期しない結果が生じます。たとえば、4 バイトの整数データの入力カラムを SMALLINT 宛先カラムに挿入する場合、SAP Sybase IQ はその入力カラムの最初の 2 バイトのみをロードします。

ロード変換オプション

LOAD TABLE 文を使用してフラットファイルからロードするための変換オプションは複数あります。

| オプション | SAP Sybase IQ データ型 | アクション |
|-------|---|--|
| ASCII | TINYINT、SMALLINT、INT (または INTEGER)、UNSIGNED INT、BIGINT、UNSIGNED BIGINT、NUMERIC (または DECIMAL)、REAL、DOUBLE、BIT、DATE、TIME、TIMESTAMP (または DATETIME) | デフォルトでは、SAP Sybase IQ は、入力データは当該データ型に適切な幅のバイナリとみなします。ASCII を使用すると、データが文字フォーマットであることを SAP Sybase IQ に伝え、幅を指定できる。このオプションでは、REAL データの E 表記ができるが、パフォーマンスが低下することがある。 これらのデータ型を CHAR または VARCHAR に変換するときに問題が発生すると、SAP Sybase IQ はその問題をエラーまたは警告として .iqmsg ファイルに記録する。CONVERSION_ERROR オプションが ON の場合、SAP Sybase IQ は問題をエラーとして報告する。CONVERSION_ERROR オプションが OFF の場合、問題は警告として報告される。 |
| ASCII | CHAR、VARCHAR | デフォルトでは、SAP Sybase IQ は、ソースカラムと宛先カラムのカラム幅は同じとみなすため、入力ファイルが正しく読み込めない場合がある。このオプションを使用して異なる入力カラム幅を指定できる。 |
| DATE | DATE | 固定フォーマットの ASCII 日付入力をバイナリに変換する。 |

| オプション | SAP Sybase IQ データ型 | アクション |
|----------|-----------------------------------|--|
| DATETIME | TIMESTAMP (または DATETIME) または TIME | 固定フォーマットの ASCII 時刻または日付/時刻入力をバイナリに変換する。入力指定は 12 時間制または 24 時間制に基づく。 |
| TIME | TIME | 固定フォーマットの ASCII 時刻入力をバイナリに変換する。 |
| NULL | すべて | 挿入時に NULL に変換する入力データ値を指定する。 |

注意：フラットファイルからロードするときに、バイナリデータまたは文字データを選択できる場合、バイナリデータを使用します。バイナリ入力を使用すると、変換が不要なため、パフォーマンスが向上します。

参照：

- 明示的なデータ変換 (63 ページ)
- カラム幅の問題 (67 ページ)
- 日付と時刻のロードの高速化 (68 ページ)
- ASCII 入力の変換 (69 ページ)
- DATE オプション (71 ページ)
- DATETIME 変換オプション (73 ページ)
- NULL データ変換 (76 ページ)

明示的なデータ変換

INSERT 文を使用してフラットファイルではなくデータベースから直接データを挿入する場合、ロード変換オプションを使用できません。

データを明示的に変換する必要がある場合、挿入するデータを指定する **SELECT** 文または **VALUES** リストで、変換関数 **CAST** または **CONVERT** を使用します。データを暗黙的に変換する場合は、SAP Sybase IQ によって自動的に変換されます。

SELECT 文でデータ型を一致させる必要があるのに一致しない場合、暗黙的または明示的な変換が必要です。あるデータ型から他のデータ型への **INSERT SELECT** を実行したり、データ型が異なる値を比較または計算したりする場合に、**SELECT** 文でデータ型が一致しくなくなります。

これらの変換は、SAP Sybase IQ データベース、SQL Anywhere データベース、または特殊データストアとして接続された他のデータベースのデータに適用されます。

挿入時のデータ変換

最初の表は、**SELECT** 文に **WHERE** 句が指定されていない場合、または **WHERE** 句が比較演算 (=、>、または <) に基づく場合の変換 (暗黙的 (I)、明示的 (E)、未サポート (U)) の変換は次のとおりです。

| 変換先： | |
|-----------------|--|
| 変換元： | ti si in ui bi ub nu rl dl bt dt tm ts ch vc bn vb |
| tinyint | I I I I I I I I I I I E E E E E I I |
| smallint | I I I I I I I I I I I E E E E E I I |
| int | I I I I I I I I I I I E E E E E I I |
| unsigned int | I I I I I I I I I I I E E E E E I I |
| bigint | I I I I I I I I I I I E E E E E I I |
| unsigned bigint | I I I I I I I I I I I E E E E E I I |
| numeric | I I I I I I I I I I I E E E E E U U |
| real | I I I I I I I I I I I E E E E E U U |
| double | I I I I I I I I I I I E E E E E U U |
| bit | I I I I I I I I I I I U U U I I I I |
| date | E E E E E E E E E E U I U I E E U U |
| time | E E E E E E E E E E U U I E E E U U |
| time-stamp | E E E E E E E E E E U E I I E E U U |
| char | E E E E E E E E E E I E E E I I I I |
| varchar | E E E E E E E E E E I E E E I I I I |
| binary | I I I I I I U U U U U U U I I I I |
| varbinary | I I I I I I U U U U U U U I I I I |

表で使用されているコードの意味は次のとおりです。

| コード | データ型 | コード | データ型 | コード | データ型 |
|-----|--------------|-----|---------|-----|-----------|
| ti | tinyint | nu | numeric | ts | timestamp |
| si | smallint | rl | real | ch | char |
| in | int | dl | double | vc | varchar |
| ui | unsigned int | bt | bit | bn | binary |
| bi | bigint | dt | date | vb | varbinary |

| コード | データ型 | コード | データ型 | コード | データ型 |
|-----|-----------------|-----|------|-----|------|
| ub | unsigned bigint | tm | time | | |

SELECT 文の **WHERE** 句が算術演算 (+, -など) に基づく場合の変換は次のとおりです。

表 1 : 算術演算のための SAP Sybase IQ 変換

| 変換元 : | 変換先 : | | | | | | | | | | | | | | | | |
|-----------------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | ti | si | in | ui | bi | ub | nu | rl | dl | bt | dt | tm | ts | ch | vc | bn | vb |
| tinyint | I | I | I | I | I | I | I | I | I | I | U | U | U | E | E | I | I |
| smallint | I | I | I | I | I | I | I | I | I | I | U | U | U | E | E | I | I |
| int | I | I | I | I | I | I | I | I | I | I | U | U | U | E | E | I | I |
| unsigned int | I | I | I | I | I | I | I | I | I | I | U | U | U | E | E | I | I |
| bigint | I | I | I | I | I | I | I | I | I | I | U | U | U | E | E | I | I |
| unsigned bigint | I | I | I | I | I | I | I | I | I | I | U | U | U | E | E | I | I |
| numeric | I | I | I | I | I | I | I | I | I | I | U | U | U | E | E | U | U |
| real | I | I | I | I | I | I | I | I | I | I | U | U | U | E | E | U | U |
| double | I | I | I | I | I | I | I | I | I | I | U | U | U | E | E | U | U |
| bit | I | I | I | I | I | I | I | I | I | I | U | U | U | I | I | I | I |
| date | U | U | U | U | U | U | U | U | U | U | U | I | U | U | U | U | U |
| time | U | U | U | U | U | U | U | U | U | U | I | U | U | U | U | U | U |
| timestamp | U | U | U | U | U | U | U | U | U | U | U | U | U | U | U | U | U |
| char | E | E | E | E | E | E | E | E | E | I | U | U | U | I | I | I | I |
| varchar | E | E | E | E | E | E | E | E | E | I | U | U | U | I | I | I | I |
| binary | I | I | I | I | I | I | U | U | U | U | U | U | U | I | I | I | I |
| varbinary | I | I | I | I | I | I | U | U | U | U | U | U | U | I | I | I | I |

注意 : 算術演算では、bit データは暗黙的に tinyint に変換されます。

INSERT 文と **UPDATE** 文の変換は次のとおりです。

挿入時のデータ変換

| | 変換先： | | | | | | | | | | | | | | | | |
|-----------------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 変換元： | ti | si | in | ui | bi | ub | nu | rl | dl | bt | dt | tm | ts | ch | vc | bn | vb |
| tinyint | I | I | I | I | I | I | I | I | I | I | E | E | E | E | E | I | I |
| smallint | I | I | I | I | I | I | I | I | I | I | E | E | E | E | E | I | I |
| int | I | I | I | I | I | I | I | I | I | I | E | E | E | E | E | I | I |
| unsigned int | I | I | I | I | I | I | I | I | I | I | E | E | E | E | E | I | I |
| bigint | I | I | I | I | I | I | I | I | I | I | E | E | E | E | E | I | I |
| unsigned bigint | I | I | I | I | I | I | I | I | I | I | E | E | E | E | E | I | I |
| numeric | I | I | I | I | I | I | I | I | I | E | E | E | E | E | E | U | U |
| real | I | I | I | I | I | I | I | I | I | I | E | E | E | E | E | U | U |
| double | I | I | I | I | I | I | I | I | I | I | E | E | E | E | E | U | U |
| bit | I | I | I | I | I | I | I | I | I | I | U | U | U | I | I | I | I |
| date | E | E | E | E | E | E | E | E | E | E | I | U | I | E | E | U | U |
| time | E | E | E | E | E | E | E | E | E | E | U | I | E | E | E | U | U |
| time-stamp | E | E | E | E | E | E | E | E | E | E | E | I | I | E | E | U | U |
| char | I | I | I | I | I | I | I | I | I | I | E | E | E | I | I | I | I |
| varchar | I | I | I | I | I | I | I | I | I | I | E | E | E | I | I | I | I |
| binary | I | I | I | I | I | I | U | U | U | I | U | U | U | I | I | I | I |
| varbinary | I | I | I | I | I | I | U | U | U | I | U | U | U | I | I | I | I |

参照：

- ロード変換オプション (62 ページ)
- カラム幅の問題 (67 ページ)
- 日付と時刻のロードの高速化 (68 ページ)
- ASCII 入力の変換 (69 ページ)
- DATE オプション (71 ページ)
- DATETIME 変換オプション (73 ページ)
- NULL データ変換 (76 ページ)

カラム幅の問題

SAP Sybase IQ は、入力データの幅が宛先カラムの幅と同じとみなして、入力ファイルを読み込みます。

幅が同じでない場合、SAP Sybase IQ が入力ファイルの当該カラムから読み込むバイト数が少なすぎたり多すぎたりすることがあります。この結果、カラムのデータを正しく読み込むことができません。また、これに続くカラムの読み込み開始位置がずれてしまうため、以降のカラムについても、データを正しく読み込むことができません。

たとえば、`input_column1` の幅が 15 バイト、`destination_column1` の幅が 10 バイトで、**ASCII** 変換オプションを指定しない場合、SAP Sybase IQ は、入力カラムの幅が 10 バイトであるとみなします。入力データは 10 バイトにトランケートされるため、`destination_column1` については問題ありません。しかし、入力ファイルの次のカラムの正しい開始位置はバイト 16 ですが、SAP Sybase IQ は、最初のカラムの途中のバイト 11 から始まるとみなします。

逆に、`input_column1` の幅が 10 バイト、`destination_column1` の幅が 15 バイトで、**ASCII** 変換オプションを指定しない場合、SAP Sybase IQ は入力カラム幅は 15 バイトとみなします。SAP Sybase IQ は、入力ファイルから `input_column1` のすべてと次のカラムの 5 バイトを読み込み、その値を `destination_column1` に挿入します。このため、`destination_column1` に挿入された値と、それ以降のすべてのカラムの値は、いずれも正しくありません。

このような問題を避けるには、**ASCII** 変換オプションを使用します。このオプションによって、SAP Sybase IQ は、入力カラムの固定または可変幅を指定するさまざまな方法を提供します。入力データには、特定のサイズ(バイト単位)の固定幅入力カラム、カラムデリミタを含む可変幅入力カラム、バイナリプレフィクスバイトで定義される可変幅入力カラムを含めることができます。

参照：

- ロード変換オプション (62 ページ)
- 明示的なデータ変換 (63 ページ)
- 日付と時刻のロードの高速化 (68 ページ)
- ASCII 入力の変換 (69 ページ)
- DATE オプション (71 ページ)
- DATETIME 変換オプション (73 ページ)
- NULL データ変換 (76 ページ)

日付と時刻のロードの高速化

SAP Sybase IQ には、日付、時刻、日付時刻データのロード時に ASCII からバイナリへの変換を行うためのパフォーマンスの最適化が組み込まれています。ロードする生データがこれらのフォーマットのいずれかにぴったり合致する場合は、適切なフォーマットを使用することによってロード時間を大幅に短縮できます。

認識されるフォーマットは次のとおりです。

- "YYYY-MM-DD"
- "YYYY/MM/DD"
- "YYYY.MM.DD"
- "YYYYMMDD"
- "MM-DD-YYYY"
- "MM/DD/YYYY"
- "DD-MM-YYYY"
- "DD/MM/YYYY"
- "DD.MM.YYYY"
- "HH:NN:SS"
- "HHNNSS"
- "HH:NN:SS.S"
- "HH:NN:SS.SS"
- "HH:NN:SS.SSS"
- "HH:NN:SS.SSSS"
- "HH:NN:SS.SSSSS"
- "HH:NN:SS.SSSSSS"
- "YYYY-MM-DD HH:NN:SS"
- "YYYYMMDD HHNNSS"
- "YYYY-MM-DD HH:NN:SS.S"
- "YYYY-MM-DD HH:NN:SS.SS"
- "YYYY-MM-DD HH:NN:SS.SSS"
- "YYYY-MM-DD HH:NN:SS.SSSS"
- "YYYY-MM-DD HH:NN:SS.SSSSS"
- "YYYY-MM-DD HH:NN:SS.SSSSSS"

日付、時刻、または日付時刻カラムが 1 つまたは複数あるテーブルをロードする場合で、入力フォーマットが前述のいずれかのフォーマットのときには、ロード文で適切なフォーマットを明示的に指定すれば、ロード処理は非常に高速になります。そうしないと、ロードは非常に遅くなることがあります。

次のように作成した、日付カラムを持つテーブルがあるとします。

```
CREATE TABLE table1(c1 DATE);
```

このテーブルをロードするには、次のような文を使用します。

```
LOAD TABLE table1 (c1 ASCII(10)) FROM ...
```

生データフォーマットが、最適化されたフォーマット (YYYY-MM-DD など) となっている場合の方が、ロードははるかに速くなります。

参照：

- ロード変換オプション (62 ページ)
- 明示的なデータ変換 (63 ページ)
- カラム幅の問題 (67 ページ)
- ASCII 入力の変換 (69 ページ)
- DATE オプション (71 ページ)
- DATETIME 変換オプション (73 ページ)
- NULL データ変換 (76 ページ)

ASCII 入力の変換

ASCII 入力データをバイナリに変換します。

ASCII 変換オプションによって次の操作を行います。

- ASCII 入力データをバイナリに変換し、カラムのデータが正しく読み取れるように入力カラムの幅を指定します。
- 入力カラムの幅が宛先カラムの幅と異なる場合は、ASCII データを ASCII データ型のカラムに挿入します。このオプションは、各カラムで読み込む入力データのサイズを指定できます。

1、2、または 4 バイトのプレフィックスおよびカラムデリミタによって、すべての SAP Sybase IQ データ型にこのオプションを使用できます。

VARCHAR カラムと *CHAR* カラムのデータのトランケート

SAP Sybase IQ は、入力カラムの幅が宛先カラムの幅より大きい場合に、挿入時にデータをトランケートします。

入力データの幅が宛先カラムの幅より小さい場合、*CHAR* データ型または *VARCHAR* データ型で SAP Sybase IQ は、Sybase IQ は挿入時にデータにスペースを埋め込みます。

VARCHAR カラムへの可変幅挿入では、後続ブランクは削除されませんが、*VARCHAR* カラムへの固定幅挿入では削除されます。たとえば、テーブル `varcolumn` のカラム `varcolumn` に挿入する場合を考えてみます。次の例は固定

挿入時のデータ変換

幅挿入で、2つのブランク (_ で示す) が明示的に指定されているため、値は削除されません。

```
INSERT INTO vartable VALUES ('box_ _')
```

区切られた入力を使用してフラットファイルから同じ値を入力する場合は、可変幅挿入となり、後続ブランクは削除されます。

ASCII 変換オプションは、SAP Sybase IQ データ型で機能します。この例では、フラット ASCII ファイル `shipinfo.t` から SAP Sybase IQ テーブル `lineitem` にデータを挿入し、入力データとテーブルの内容とフォーマットの要約を示します。

| ファイル <code>shipinfo.t</code> | | | テーブル <code>lineitem</code> | | |
|------------------------------|--------|----|----------------------------|---------|----|
| カラム | フォーマット | 幅 | カラム | データ型 | 幅 |
| <code>l_shipmode</code> | CHAR | 15 | <code>l_shipmode</code> | VARCHAR | 30 |
| <code>l_quantity</code> | ASCII | 8 | <code>l_quantity</code> | INT | 4 |

`l_shipmode` カラムでは、ASCII データを ASCII カラム (VARCHAR データ型) に挿入します。2つのカラムの幅が異なることに注意してください。このカラムと次の `l_quantity` カラムから正しく挿入するために、`l_shipmode` カラムの幅を指定して、正しいサイズの入力データを正しい位置で読み込めるようにします。

`l_quantity` カラムでは、ASCII データをバイナリカラム (INT データ型) に挿入します。このカラムから正しく挿入するため、入力データをバイナリに変換し、入力カラムの幅を指定する必要があります。

次に示す UNIX の例は変換を指定するコマンドです。

```
LOAD TABLE lineitem(  
    l_shipmode ASCII(15),  
    l_quantity ASCII(8),  
    FILLER(1))  
FROM '/d1/MILL1/shipinfo.t'  
PREVIEW ON
```

参照：

- ロード変換オプション (62 ページ)
- 明示的なデータ変換 (63 ページ)
- カラム幅の問題 (67 ページ)
- 日付と時刻のロードの高速化 (68 ページ)
- DATE オプション (71 ページ)
- DATETIME 変換オプション (73 ページ)
- NULL データ変換 (76 ページ)

NULL またはブランク文字の置換

SAP Sybase IQ では、長さゼロの VARCHAR データがサポートされています。

VARCHAR セルの長さがゼロで、セルが NULL 以外の場合は、オプション NON_ANSI_NULL_VARCHAR=OFF を設定すると長さゼロのセルが作成されます。

NON_ANSI_NULL_VARCHAR=ON を設定すると、NULL が挿入されます。

その他のデータ型については、セルの長さがゼロの場合に NULL が挿入されます。

DATE オプション

DATE 変換オプションによって、固定フォーマットで格納された ASCII データを DATE カラムに挿入します。

このオプションは、ASCII データ入力をバイナリに変換し、入力データのフォーマットを指定します。DATE フォーマットは入力を解釈するために内部で使用され、データの格納または出力フォーマットに影響はありません。

例

この Windows の例では、l_shipdate カラムのデータは指定フォーマットからバイナリに変換されます。1 バイトの FILLER 値は入力ファイル内の改行を省略します。

```
LOAD TABLE lineitem(
  l_orderkey NULL(ZEROS) ASCII(4),
  l_partkey ASCII(3),
  l_shipdate DATE('MM/DD/YY'),
  l_suppkey ASCII(5),
  FILLER(1))
FROM 'C:¥¥MILL1¥¥shipinfo.t'
PREVIEW ON
```

参照：

- ロード変換オプション (62 ページ)
- 明示的なデータ変換 (63 ページ)
- カラム幅の問題 (67 ページ)
- 日付と時刻のロードの高速化 (68 ページ)
- ASCII 入力の変換 (69 ページ)
- DATETIME 変換オプション (73 ページ)
- NULL データ変換 (76 ページ)

DATE フォーマット

y または Y (年)、m または M (月)、d または D (日)、j または J (ユリウス日付) を使用して入力データのフォーマットを指定します。

フォーマット文字列の長さは入力カラム幅です。

| オプション | 意味 |
|-------------------------------|--|
| yyyy または YYYY yy または YY | 年を表す。デフォルトは 1900。 |
| mm または MM | 月を表す。1桁の月は、月の先頭に 0 を付ける。たとえば、5 月は '05' となる。DATE 値で月を省略すると、日付はユリウス日付として扱われる。たとえば、'03' のように月のみを入力した場合、SAP Sybase IQ はデフォルトの年と日を適用し、'1900-03-01' に変換する。 |
| dd または DD jjj または JJJ | 日付を表す。デフォルトの日付は 01。1桁の日付は、日付の先頭に 0 を付ける。たとえば、最初の日は '01' となる。J または j は年のユリウス日付 (1 ~ 366) を示す。 |

入力時には、フォーマットコードの大文字と小文字は無視されます。

出力時には、フォーマットコードの大文字と小文字は次のように処理されます。

- 大文字と小文字が混在する場合 (たとえば、"Dd")、ゼロを補わない。
- 大文字または小文字のみの場合 (たとえば、"DD"、"dd")、足りない桁の分だけゼロを補う。

たとえば、デフォルトの時刻フォーマットを使用すると、時刻は 17:23:03.774 と出力され、'HH:NN:Ss.SSS' を使用すると、17:23:3.774 と出力されます。

DATE フォーマットのオプションの例には、日付入力データと、**DATE** 変換オプションによるフォーマットの指定方法が示されています。

| 入力データ | フォーマット指定 |
|----------|-------------------|
| 12/31/09 | DATE ('MM/DD/YY') |
| 12-31-09 | DATE ('MM/DD/YY') |
| 20091231 | DATE ('YYYYMMDD') |
| 12/09 | DATE ('MM/YY') |
| 2009/123 | DATE ('YYYY/JJJ') |

日付を指定する場合の一般的な規則は次のとおりです。

- DATE 指定は、カッコと一重引用符または二重引用符で囲みます。
- SAP Sybase IQ は、年、月、日の値のみを格納し、入力データの他の文字は格納しません。ただし、入力データに、スラッシュ (/)、ダッシュ (-)、ブランクなどの月、日、年の区切り文字が含まれる場合は、DATE フォーマットにその文字の位置を示し、無視できるようにします。
- Y、M、J、D 以外の文字を使用して SAP Sybase IQ が省略する区切り文字を指定します。ブランクも使用できます。
- DATE フォーマットに年と、その年の日数のみを指定すると、SAP Sybase IQ は日付をユリウス日付とみなします。たとえば、2009-33 は 2009 年の 33 日目、つまり、2009 年 2 月 2 日です。
- "5/27/32" などのように年を 2 桁のみで指定する場合、SAP Sybase IQ は年および **NEAREST_CENTURY** オプションの設定に応じて 19yy または 20yy に変換します。

| NEAREST_CENTURY 設定 | 指定された年 | 想定される年 |
|--------------------|---------|-------------|
| デフォルト (50) | 00 - 49 | 2000 - 2049 |
| | 50 - 99 | 1950 - 1999 |
| 0 | 任意 | 1900 年代 |
| 100 | 任意 | 2000 年代 |

DATETIME 変換オプション

DATETIME 変換オプションによって、固定フォーマットで格納された ASCII データを TIME、TIMESTAMP、または DATETIME カラムに挿入します。

このオプションは、ASCII データ入力をバイナリに変換し、入力データのフォーマットを指定します。**DATETIME** フォーマットは入力を解釈するために内部で使用され、データの格納または出力フォーマットに影響はありません。

注意： 以前のリリースとの互換性を保つため、カラムに DATETIME データが含まれるよう指定できます。ただし、DATETIME データは内部では同等のフォーマットの TIMESTAMP として格納されます。

構文は次のとおりです。

```
DATETIME ('input-datetime-format')
```

この UNIX の例では、スラッシュは入力データの日付部分の区切り文字、コロンは時刻部分の区切り文字です。

```
LOAD TABLE lineitem(
  l_quantity ASCII(4),
  l_shipdate DATETIME('MM/DD/YY hh:mm:ss'),
  FILLER(1))
```

挿入時のデータ変換

```
FROM '/d1/MILL1/tt.t'  
PREVIEW ON
```

この UNIX の例では、FILLER(1) 句があるため、SAP Sybase IQ は DATETIME カラムの次のカラム (VWAP) に NULL を挿入しません。

```
LOAD TABLE snapquote_stats_base  
SYMBOL '\%x09',  
snaptime DATETIME('MM/DD/YY hh:mm:ss'),  
FILLER(1)  
VWAP '\%x09',  
RS_DAY '\%x09',  
FROM '/d1/MILL1/tt.t'  
PREVIEW ON
```

この UNIX の例では、宛先カラムに TIME データが含まれていますが、入力データは DATETIME です。**DATETIME** 変換オプションと **FILLER** 句を使用して日付部分をスキップします。

```
LOAD TABLE Customers(  
    open_time DATETIME('hh:mm:aa'),  
    close_time DATETIME('hh:mm:aa'),  
FILLER(9))  
FROM '/d1/MILL1/tt.t'  
PREVIEW ON
```

参照：

- ロード変換オプション (62 ページ)
- 明示的なデータ変換 (63 ページ)
- カラム幅の問題 (67 ページ)
- 日付と時刻のロードの高速化 (68 ページ)
- ASCII 入力の変換 (69 ページ)
- DATE オプション (71 ページ)
- NULL データ変換 (76 ページ)

DATETIME 変換フォーマットの指定

DATETIME 入力のフォーマットを指定します。

使用するフォーマットを次のように指定します。

- 年は Y または y
- 月は M または m
- 日は D または d
- 時刻は H または h
- 分は N または n (コロンを区切り文字として使用する場合、mm も使用できる)

- ・ 秒および 1 秒未満の秒数は S または s

フォーマット文字列の長さは入力カラム幅です。

| オプション | 意味 |
|--------------------|---|
| hh HH | 24 時間制で時間を示す。1 桁の時間は時間の前に 0 を付ける。たとえば、午前 1 時は '01' となる。午前 12 時 (深夜) を表す場合、'00' も有効な値である。 |
| nn | 分を示す。1 桁の分は、分の前に 0 を付ける。たとえば、8 分は '08' となる。 |
| ss[.ssssss] | 秒数と 1 秒未満の秒数を示す。 |
| aa | 午前または午後を示す。 |
| pp | 必要な場合にのみ、午後を示す(このオプションはバージョン 12.0 より前の SAP Sybase IQ とは互換性がない。以前は「pp」と「aa」は同義であった)。 |
| hh | SAP Sybase IQ は、分と秒をゼロとみなす。たとえば、入力した DATETIME 値が「03」の場合、SAP Sybase IQ では、「03:00:00.0000」に変換される。 |
| hh:nn または hh:mm | SAP Sybase IQ は秒をゼロとみなす。たとえば、入力した時刻値が「03:25」の場合、SAP Sybase IQ では「03:25:00.0000」に変換される。 |

DATETIME フォーマットのオプションの例には、時刻入力データと、DATETIME 変換オプションのフォーマットの指定方法が示されています。

| 入力データ | フォーマット指定 |
|------------------------|------------------------------------|
| 12/31/00 14:01:50 | DATETIME ('MM/DD/YY hh:nn:ss') |
| 123100140150 | DATETIME ('MMDDYYhhnnss') |
| 14:01:50 12-31-00 | DATETIME ('hh:mm:ss MM-DD-YY') |
| 12/31/00 14:01:12.456 | DATETIME ('MM/DD/YY hh:nn:ssssss') |
| 12/31/00 14:01:.123456 | DATETIME ('MM/DD/YY hh:mm:ssssss') |
| 12/31/00 02:01:50AM | DATETIME ('MM/DD/YY hh:mm:ssaa') |
| 12/31/00 02:01:50pm | DATETIME ('MM/DD/YY hh:mm:sspp') |

日付を指定する場合の一般的な規則は次のとおりです。

- ・ 時刻要素の指定文字は、カッコと一重引用符または二重引用符で囲みます。
- ・ 浮動小数点を含め、9 桁までの秒数を入力できるため、1 秒未満の秒数を指定できます。入力とクエリでは、小数点の位置が変わるため、6 つまでの小数点

位置を指定できます。ただし、SAP Sybase IQ では、常に 2 桁の秒数と 6 桁の小数点以下の数値のみが格納されます (ss.ssssss)。それ以上の小数点以下の数値は指定できません。

- 区切り文字は、時間要素を区切るために使用します。ブランクを含むすべての文字は、区切り文字として使用できます。例ではコロン(:)を使用しています。
- SAP Sybase IQ は、時、分、秒の数値のみを格納し、入力データの他の文字は格納しません。ただし、時、分、秒を区切るためにコロンやブランクなどの他の文字がデータに含まれる場合は、フォーマット指定の時刻部分にこれらの文字の位置を指定し、SAP Sybase IQ がスキップできるようにします。
- 特定の値が午前または午後であることを示す場合は、入力データの特定の位置に大文字または小文字の 'a' または 'p' を指定してください。SAP Sybase IQ が午前または午後の指定を検索する場所を示すため、フォーマット指定の適切な場所に小文字のみで 'aa' または 'pp' を指定します。'aa' は常に午前/午後を示すことを指定し、'pp' は必要な場合にのみ午後であることを示すことを指定します。
- フォーマット指定には、入力のすべての文字と一致する文字を使用する必要があります。'm' は分を表すために使用されているため、入力の 'm' と一致させる目的では使用できません。
- 時刻セクションで時、分、または秒を指定しないと、SAP Sybase IQ はそれぞれの値を 0 と見なします。

NULL データ変換

SAP Sybase IQ カラムインデックスに挿入するときに入力データの特定の値を NULL に変換するには、**NULL** 変換オプションを使用します。

このオプションは、NULL 値が許可されているカラムで使用できます。すべての SAP Sybase IQ のデータ型に、この変換オプションを指定できます。

構文は次のとおりです。

```
NULL ({BLANKS | ZEROS | literal' ['literal']...})
```

各パラメータの意味は次のとおりです。

- **BLANKS** は、ブランクを NULL に変換することを示します。
- **ZEROS** は、バイナリゼロを NULL に変換することを示します。
- **literal** は、指定したリテラルをすべて NULL に変換することを示します。指定したリテラルが入力ファイルの値と一致していると SAP Sybase IQ に認識させるためには、先行ブランクおよび後続ブランクも含めて、両者が正確に一致している必要があります。20 までのリテラル値をリストできます。

同じカラムに追加の変換オプションを指定することが必要な場合があります。たとえば、ASCII データを INT カラムに挿入し、バイナリフォーマットで格納し、

入力データのブランクを挿入時に NULL に変換する場合、**ASCII** 変換オプションを使用して入力をバイナリに変換し、さらに **NULL** 変換オプションを使用してブランクを NULL に変換します。

Windows での例を示します。

```
LOAD TABLE lineitem(  
    l_orderkey NULL(ZEROS) ASCII(4),  
    l_partkey ASCII(3),  
    l_shipdate date('MM/DD/YY'),  
    l_suppkey ascii(5),  
    FILLER(1))  
FROM 'C:¥¥MILL1¥¥tt.t'  
PREVIEW ON
```

参照：

- ロード変換オプション (62 ページ)
- 明示的なデータ変換 (63 ページ)
- カラム幅の問題 (67 ページ)
- 日付と時刻のロードの高速化 (68 ページ)
- ASCII 入力の変換 (69 ページ)
- DATE オプション (71 ページ)
- DATETIME 変換オプション (73 ページ)

結果の丸めとトランケート

SAP Sybase IQ では、クエリまたは挿入時にデータ型を明示的または暗黙的に変換する必要がある場合、結果は常にトランケートされます。

- データを明示的に低い位取りに変換する場合、SAP Sybase IQ は結果の値をトランケートします。たとえば、位取り 4 に格納されているクエリのカラム値を位取り 2 に **CAST** する場合は、2.4561 の値は 2.45 となります。
- SAP Sybase IQ は、挿入時に低い位取りに暗黙的に変換する場合は、データをテーブルに挿入する前に値をトランケートします。たとえば、データ型 NUMERIC (7, 3) のテーブルからデータ型 DECIMAL (12, 2) の別のテーブルへ挿入する場合、2.456 の値は 2.45 となります。
- 算術演算結果の位取りが事前に定義された位取りより高くなった場合、SAP Sybase IQ は事前定義の位取りに合うように結果をトランケートします。

結果の値をトランケートではなく、丸める必要がある場合は、コマンドに **ROUND** 関数を使用します。ただし、挿入の場合、**ROUND** 関数をクエリ式の一部にのみ適用できます。

数値データの最大精度は 126 です。

結果の丸めとトランケート

対応する Adaptive Server データ型

SAP Sybase IQ データ型は、Adaptive Server データ型と互換性があります。

一般的な規則は次のとおりです。

- SAP Sybase IQ 文字列型は、いかなる Adaptive Server 文字列型でも受け入れられます。
- SAP Sybase IQ 真数型は、いかなる Adaptive Server 数値型でも受け入れられます。ただし、SAP Sybase IQ データ型が、Adaptive Server データ型よりも少ないデータしか格納できない場合は、値は NULL に変換されます (たとえば、基礎となるデータベースからデータをテーブルに挿入する場合)。
- SAP Sybase IQ 日付/時刻型は、いかなる Adaptive Server 日付/時刻型でも受け入れられます。

サポートされていない Adaptive Server データ型

サポートされていない Adaptive Server データ型もあります。

SAP Sybase IQ で現在サポートされていない Adaptive Server データ型は、次のとおりです。

- date
- text
- nchar, nvarchar
- unichar, univarchar, unitext
- text
- image
- unsigned smallint
- native Java data types
- XML data type

次の点に注意してください。

- SAP Sybase IQ は、バイナリラージオブジェクト (BLOB) データ型とキャラクターラージオブジェクト (CLOB) データ型を通じて Adaptive Server の text データ型と image データ型をサポートします。
- SAP Sybase IQ では、Adaptive Server データ型の DATE、TEXT、UNSIGNED SMALLINT、NCHAR、NVARCHAR、UNICHAR、UNIVARCHAR、および UNITEXT をサポートしていませんが、**INSERT** 文の **LOCATION** 構文を使用すると、データ型が DATE、TEXT、UNSIGNED SMALLINT、NCHAR、NVARCHAR、

対応する Adaptive Server データ型

UNICHAR、UNIVARCHAR、または UNITEXT の Adaptive Server データベースからデータを挿入できます。

参照：

- Adaptive Server のデータ型に対応するデータ型 (82 ページ)
- データインポート時の変換エラー (86 ページ)

Adaptive Server のデータ型に対応するデータ型

Adaptive Server 真数値型には、対応する SAP Sybase IQ データ型があります。

| Adaptive Server データ型 | SAP Sybase IQ データ型 | 説明 |
|----------------------|---|--|
| int | INT、BIGINT、UNSIGNED INT、UNSIGNED BIGINT、または NUMERIC | SAP Sybase IQ では、INT(7,3) などの位取りされた整数は使用できません。INT(<i>precision, scale</i>) 形式のデータは NUMERIC (<i>precision, scale</i>) に変換されます。12.0 より前のバージョンの SAP Sybase IQ と Adaptive Server では、int データ型は -2,147,483,648 ～ 2,147,483,647 の値となり、これと異なる。 大きな整数値を扱う場合は、BIGINT データ型、符号なし整数 (UNSIGNED INT) データ型、または UNSIGNED BIGINT データ型を使用できる。UNSIGNED INT の場合、最後のビットは値の一部として使用される。正または負の指定はない。すべての数値は正とみなされるため、値は最大で 4,294,967,295 とできる。 |
| numeric | 適切な精度の DECIMAL または NUMERIC | 定義する SAP Sybase IQ のデータ型の精度が低すぎるために Adaptive Server の値を格納できない場合、その値は NULL に変換される。 |
| decimal | 適切な精度の DECIMAL または NUMERIC | 上記を参照。 |
| smallint | SMALLINT または NUMERIC | SAP Sybase IQ の SMALLINT では、精度と位取りは使用できない。Adaptive Server の smallint (<i>precision, scale</i>) は NUMERIC (<i>precision, scale</i>) に変換される。上記の INT を参照。 |

| Adaptive Server データ型 | SAP Sybase IQ データ型 | 説明 |
|-------------------------|-----------------------|--|
| tinyint | TINYINT | SAP Sybase IQ の TINYINT カラムでは、精度と位取りは使用できない。Adaptive Server の tinyint (precision, scale) は NUMERIC (precision, scale) に変換される。上記の INT を参照。 |
| unsigned smallint | サポートされていない | SAP Sybase IQ は、Adaptive Server の unsigned smallint データ型をサポートしていないが、 INSERT...LOCATION を使用して、Adaptive Server データベースの unsigned smallint データ型のカラムからデータを挿入できる。 |

Adaptive Server 概数値データ型と、それに対応する SAP Sybase IQ のデータ型は次のとおりです。

| Adaptive Server データ型 | SAP Sybase IQ データ型 | 説明 |
|-------------------------|-----------------------|--|
| float (precision) | FLOAT (precision) | IQ は FLOAT のより高い精度をサポートしている。 HNG インデックスでは、FLOAT、REAL、または DOUBLE のデータは使用できない。 |

Adaptive Server の文字データ型と、それに対応する SAP Sybase IQ のデータ型は次のとおりです。

| Adaptive Server データ型 | SAP Sybase IQ データ型 | 説明 |
|-------------------------|-----------------------|---|
| char | CHAR | SAP Sybase IQ と Adaptive Server の文字データ型 (char または CHAR) は、ほぼ同じである。唯一の違いは、SAP Sybase IQ では NULL を扱える点である。SAP Sybase IQ の CHAR カラムを Adaptive Server の char カラムと完全に一致させるには、SAP Sybase IQ カラムに NOT NULL と指定する。SAP Sybase IQ のデフォルト設定では NULL を使用できる。Adaptive Server の、NULL を許容する char カラムは、内部で varchar に変換される。 |
| varchar | VARCHAR | 上記の char の説明を参照。 |

対応する Adaptive Server データ型

| Adaptive Server データ型 | SAP Sybase IQ データ型 | 説明 |
|----------------------|--------------------|--|
| nchar | サポートされていない | SAP Sybase IQ は、Adaptive Server の nchar データ型をサポートしていないが、 INSERT...LOCATION を使用して、ASE データベースの nchar データ型のカラムからデータを挿入できる。 |
| nvarchar | サポートされていない | SAP Sybase IQ は、Adaptive Server の nvarchar データ型をサポートしていないが、 INSERT...LOCATION を使用して、Adaptive Server データベースの nvarchar データ型のカラムからデータを挿入できる。 |
| text | サポートされていない | SAP Sybase IQ は、Adaptive Server の text データ型をサポートしていないが、 INSERT...LOCATION を使用して、Adaptive Server データベースの text データ型のカラムからデータを挿入できる。 |
| unichar | サポートされていない | SAP Sybase IQ は、Adaptive Server の unichar データ型をサポートしていないが、 INSERT...LOCATION を使用して、Adaptive Server データベースの unichar データ型のカラムからデータを挿入できる。 |
| univarchar | サポートされていない | SAP Sybase IQ は、Adaptive Server の univarchar データ型をサポートしていないが、 INSERT...LOCATION を使用して、Adaptive Server データベースの univarchar データ型のカラムからデータを挿入できる。 |
| unitext | サポートされていない | SAP Sybase IQ は、Adaptive Server の unitext データ型をサポートしていないが、 INSERT...LOCATION を使用して、Adaptive Server データベースの unitext データ型のカラムからデータを挿入できる。 |

Adaptive Server の money データ型と、それに対応する SAP Sybase IQ のデータ型は次のとおりです。

| Adaptive Server データ型 | SAP Sybase IQ データ型 | 説明 |
|----------------------|--------------------|---|
| money | NUMERIC(19,4) | money データは、暗黙的に NUMERIC (19, 4) に変換される。 |
| smallmoney | NUMERIC(10,4) | |

Adaptive Server の DATE/TIME データ型と、それに対応する SAP Sybase IQ のデータ型は次のとおりです。

| Adaptive Server データ型 | SAP Sybase IQ データ型 | 説明 |
|----------------------|------------------------------|--|
| datetime | TIMESTAMP、DATE、または TIME | <p>Adaptive Server の datetime カラムでは、仮想的な日付である 0/0/0000 を基準日として、その前後の日数を 4 バイトの数値、時刻を 8 バイトの数値として保持し、1/1,000,000 秒の精度で日付を指定できる。SAP Sybase IQ の TIMESTAMP (または DATETIME) カラムは、24 時間制に基づき 1/10,000 秒の精度で、1/1/0 以降の日数と時刻の 2 つをそれぞれ 4 バイト整数値として保持する。SAP Sybase IQ はこの変換を自動的に行う。</p> <p>SAP Sybase IQ には、DATE データ型も別にある。これは単一の 4 バイト整数である。SQL Server または Adaptive Server の datetime カラムから日付のみを抽出する場合は、SAP Sybase IQ の DATE データ型を使用して行う。それには、SAP Sybase IQ の DATE カラムを、Adaptive Server の datetime カラムと同じ名前で作成する。SAP Sybase IQ は datetime 値の該当部分を自動的に選択する。</p> |
| smalldatetime | TIMESTAMP、DATETIME、DATE、TIME | <p>Adaptive Server の smalldatetime カラムを、SAP Sybase IQ では TIMESTAMP (または DATETIME) データ型として定義する。SAP Sybase IQ はこの変換を正しく行う。通常の datetime と同様に、Adaptive Server の smalldatetime カラムから日付のみを抽出する場合は、SAP Sybase IQ の DATE データ型を使用する。</p> |
| date | date | <p>INSERT...LOCATION を使用して、Adaptive Server データベースの date データ型のカラムからデータを挿入できる。</p> |
| time | time | <p>SAP Sybase IQ のデータ型は、時、分、秒、秒以下で構成される時間である。秒以下は 6 桁まで格納される。time 値には 8 バイトの記憶領域が必要。</p> <p>Adaptive Server の time データ型は 00:00:00.000 から 23:59:59.999 の間である。24 時間制、または正午を 12AM、真夜中を 12PM とする 12 時間制のどちらでも使用できる。time 値には、コロンか AM/PM 指示子を含める。AM または PM は、大文字でも小文字でもよい。time 値には 4 バイトの記憶領域が必要。</p> <p>INSERT...LOCATION を使用して、Adaptive Server データベースの time データ型のカラムからデータを挿入できる。</p> |

対応する Adaptive Server データ型

Adaptive Server の binary データ型と、それに対応する SAP Sybase IQ のデータ型は次のとおりです。

| Adaptive Server データ型 | SAP Sybase IQ データ型 | 説明 |
|----------------------|--------------------|--|
| binary | BINARY | SAP Sybase IQ は、すべての BINARY カラムに対し、後続のゼロを埋め込む。 BINARY カラムの長さは、必ず偶数の文字数とする。 HNG インデックスでは、BINARY データは使用できない。 |
| varbinary | VARBINARY | SAP Sybase IQ は、VARBINARY カラムについては、後続のゼロの埋め込みやトランケートは行わない。 VARBINARY カラムの長さは、必ず偶数の文字数とする。 HNG インデックスでは、VARBINARY データは使用できない。 |

サポートされていない Adaptive Server データ型のカラムを省略します。

- nchar, nvarchar
- univar, univarchar
- unsigned smallint
- native Java data types

また、カスタムの Adaptive Server データ型も省略します。

参照：

- サポートされていない Adaptive Server データ型 (81 ページ)
- データインポート時の変換エラー (86 ページ)

データインポート時の変換エラー

外部ソースからロードされたデータは、エラーを含む場合があります。

たとえば、無効な日付や数字が含まれている可能性があります。

CONVERSION_ERROR データベースオプションを使用すると、変換エラーが NULL 値に変換されるため、それらのエラーを無視できます。

参照：

- サポートされていない Adaptive Server データ型 (81 ページ)
- Adaptive Server のデータ型に対応するデータ型 (82 ページ)

バルクロードのチューニング

データベースに大量のデータをロードする場合、時間がかかり、多くのディスク領域が使用されます。チューニングを行うとパフォーマンスが向上します。

データベースを定義するときのロードパフォーマンス

データベース定義、テーブル定義、およびインデックス定義はロードパフォーマンスに影響を与えます。

個別値

IQ UNIQUE は、カラムの予期されるカーディナリティを定義し、カラムをフラット FP または NBit FP のどちらとしてロードするかを決定します。IQ UNIQUE(n) の値を明示的に 0 に設定すると、カラムはフラット FP としてロードされます。IQ UNIQUE 制約のないカラムは、FP_NBIT_AUTOSIZE_LIMIT オプションおよび FP_NBIT_LOOKUP_MB オプションで定義された上限まで暗黙的に NBit としてロードされます。

- FP_NBIT_AUTOSIZE_LIMIT は、NBit としてロードする、重複しない値の数を制限します。
- FP_NBIT_LOOKUP_MB は、NBit ディクショナリの合計サイズのスレッシュホールドを設定します。
- FP_NBIT_ROLLOVER_MAX_MB は、NBit からフラット FP への暗黙的な NBit ロールオーバーで使用するディクショナリサイズを設定します。
- FP_NBIT_ENFORCE_LIMITS は、NBit ディクショナリのサイズ制限を強制します。このオプションはデフォルトで OFF になっています。

IQ UNIQUE を FP_NBIT_AUTOSIZE_LIMIT 未満の n 値とともに使用する必要はありません。自動サイズ機能によって、カーディナリティが低いか中程度のカラムはすべて NBit としてサイズ決定されます。カラムをフラット FP としてロードする場合や、重複しない値の数が FP_NBIT_AUTOSIZE_LIMIT を超えるときにカラムを NBit としてロードする場合は、IQ UNIQUE を使用します。

インデックス

データをロードする前に必要なすべてのインデックスを作成します。データをロードした後で新しいインデックスを追加できますが、すべてのインデックスを一度にロードした方がロード操作がすばやく完了します。

参照：

- ロード時の環境の調整 (88 ページ)

- ロード時のスレッド使用 (89 ページ)

ロード時の環境の調整

データをロードするときいくつかの要素を調整すると、ロードパフォーマンスが向上します。

- ASCII またはバイナリフォーマットで生データにアクセスし、特に数百以上のローをロードするような場合、**LOAD TABLE** コマンドを使用します。**LOAD TABLE** コマンドを使用すると、最も速く挿入できます。
- フラットファイルからロードするときに、バイナリデータと文字データのいずれかを選択できる場合、バイナリデータを使用します。バイナリデータを使用すると、変換処理が行われず、I/O が少なくなるためパフォーマンスが向上します。
- **LOAD TABLE** コマンドオプションを適切に設定します。無視された整合性制約違反をログGINGする場合は、**LOAD TABLE IGNORE CONSTRAINT** オプションの **limit** を 0 以外の値に設定します。ログGINGする違反の数が多すぎると、ロードのパフォーマンスに影響します。
- データファイルとデータベースファイルを別々の物理ディスクドライブに置くと、ロード中のディスクヘッドの動きを削減できます。
- 起動パラメータを変更してラージメモリとキャッシュのサイズを増やします。ロードのために十分なメモリを確保することは、パフォーマンスを向上させるための重要な要素です。シンプレックスサーバでは、ラージメモリ所要量は利用可能な総メモリの 3 分の 1 です。IQ のメインストアとテンポラリストアに十分なメモリを確保するために、起動パラメータ **-iqlm**、**-iqtc**、および **-iqmc** の値をそれぞれ、利用可能な総メモリの 3 分の 1 の量に設定してください。マルチプレックスサーバの場合、ラージメモリ所要量は、ロード操作を処理するノードによって決定されます。コーディネータまたはライターノードで、ラージメモリオプションの値をロードに適したレベルまで増やしてください。リーダーロードの場合、クエリ操作に必要なメモリは大幅に削減されます。
- ロック競合を避けるために、データベースまたはサーバのバッファパーティションの程度を調整します。CPU 数に基づくバッファパーティションはデフォルトで有効になっており、**-iqpartition** サーバコマンドラインオプションか **Cache_Partitions** データベースオプションを設定することで調整できます。
- 混雑していない時間帯に、主要な更新をスケジュールしてください。更新中に数多くのユーザがテーブルを照会できますが、このためには CPU サイクル、ディスク領域、メモリが必要となります。これらのリソースによって挿入の処理速度が上がります。
- **INSERT** 文を使用している場合は、可能であれば、サーバと同じマシン上で Interactive SQL またはクライアントアプリケーションを実行します。ネットワークを介してデータをロードすると、通信のために余分な負荷がかかりま

す。新しいデータをロードする場合は、混雑している時間帯を避けてください。

INSERT...LOCATION を使用して、リモートの Adaptive Server Enterprise データベースから大量のテキストまたはバルクデータをネットワーク経由でロードしている場合は、**LOCATION** 句の **PACKETSIZE** パラメータを使用して TDS パケットサイズを大きくします。この変更により、ロードパフォーマンスが大きく向上することがあります。

参照：

- データベースを定義するときのロードパフォーマンス (87 ページ)
- ロード時のスレッド使用 (89 ページ)

ロード時のスレッド使用

SAP Sybase IQ は、可能なときにはマルチスレッドを使用してロードパフォーマンスを向上させます。

ロード/挿入は、コアの要求に応じて、すべてのコア (各コアに割り当てる作業の可用性と十分なリソースに依存) を使用し、割り当てられた作業を完了しようとします。ロード/挿入中の任意の時点で使用されるコアの数は、マシンのワークロード、利用可能なリソース、コアに割り当てることができる作業の可用性によって変化します。

次の場合、ロードは部分的に並列実行されます。

- 完全な並列処理を実行するためにロードに割り付けるサーバスレッドが不足している場合
- 接続ごとやチームごとに、完全に並列実行するために必要なスレッドが不足している場合
- ロードが部分幅ロードである場合、つまりテーブルには x カラムあるのに、ロードでは x カラムより少ない値を指定している場合

上記の条件のいずれかを満たす場合に、次のタイプのロードが並列実行されます。

- BINARY ロード (FORMAT BINARY)
- ASCII 固定幅ロード
- FORMAT BCP ロード
- ROW DELIMITED BY および DELIMITED BY と FORMAT ASCII、FORMAT BINARY、または FORMAT BCP を含むロード

参照：

- データベースを定義するときのロードパフォーマンス (87 ページ)

バルクロードのチューニング

- ロード時の環境の調整 (88 ページ)

テーブルローの変更

1つまたは複数のローを更新するには、**UPDATE** 文を使用します。新しいデータは、ユーザが指定する定数や式、または他のテーブルから取得するデータです。すべてのデータ修正文と同様に、一度に1つのテーブルまたはビュー内のデータしか更新できません。

UPDATE 文が整合性制約に違反すると、更新は行われずにエラーメッセージが表示されます。たとえば、追加された値の1つが誤ったデータ型であったり、カラムやデータ型のいずれかに定義された制約に違反した場合には、更新は実行されません。

UPDATE 構文の簡略バージョンは次のとおりです。

```
UPDATE table-name
SET column_name = expression
WHERE search-condition
```

例

Newton Ent. 社 (デモデータベースの Customers テーブル内の会社) が Einstein, Inc. に吸収される場合は、次のような文を使用して会社名を更新できます。

```
UPDATE Customers
SET company_name = 'Einstein, Inc.'
WHERE company_name = 'Newton Ent.'
```

WHERE 句で条件を使用できます。入力された会社名が不明の場合、次のような文を使用して会社名に Newton と付くデータに更新を試みます。

```
UPDATE Customers
SET company_name = 'Einstein, Inc.'
WHERE company_name LIKE 'Newton%'
```

探索条件は更新されるカラムを参照する必要はありません。Newton Ent. の会社 ID は 109 です。ID 値はテーブルのプライマリキーなので、次の文を使用して正しいローを確実に更新できます。

```
UPDATE Customers
SET company_name = 'Einstein, Inc.'
WHERE id = 109
```

SET 句

SET 句は、更新されるカラムとその新しい値を指定します。**WHERE** 句で更新されるローを指定します。**WHERE** 句がない場合、指定されたすべてのローのカラムが **SET** 句の値によって更新されます。

テーブルローの変更

SET 句では、データ型が正しければどんな式でも使用できます。

WHERE 句

WHERE 句で更新されるローを指定します。たとえば、次の文は "One Size Fits All Tee Shirt" を "Extra Large Tee Shirt" に書き換えます。

```
UPDATE Products
SET size = 'Extra Large'
WHERE name = 'Tee Shirt'
      AND size = 'One Size Fits All'
```

FROM 句

FROM 句を使用して、1 つ以上のテーブルから更新するテーブルにデータを引き出せます。また、**FROM** 句は、更新するローを制御する選択条件を他方のテーブルに対して適用するために使用することもできます。

データ削除の方法

データを削除するには、**DELETE** 文、**DROP TABLE** 文、**TRUNCATE TABLE** 文を使用します。

データベースからデータを削除するには、次のようにします。

- 指定した基準を満たすすべてのローをテーブルから削除するには、**DELETE** 文を使用する。
- すべてのデータのローを含むテーブル全体を削除するには、**DROP TABLE** 文を使用する。
- テーブル定義は削除しないでテーブルからすべてのローを削除するには、**TRUNCATE TABLE** 文を使用する。

削除用の領域

DELETE 文または **TRUNCATE TABLE** 文を使用する場合、SAP Sybase IQ がデータページのバージョンを格納する方法によっては、データベースに領域を追加する必要が生じることがあります。

DROP TABLE を使用する場合は、追加のバージョンページは必要ないため、領域を追加する必要はありません。

索引

数字

2000 年

変換オプション 72

A

Adaptive Server

text データと image データの挿入 47

unichar データ型 82

unitext データ型 82

univarchar データ型 82

データの挿入 47

ASCII

データ抽出 12

データフォーマット 8

変換オプション 62

挿入時の変換 69

変換のパフォーマンス 68

B

BINARY ロード形式

LOAD TABLE 35

データファイル 35

BIT データ

変換 63

BLOB データ 47

C

CHAR データ

長さ 0 のセルの置換 71

CLOB データ 47

CONVERSION_ERROR データベース オプション 86

CS_TEXT_TYPE 47

D

DATE データ型

ロードの最適化 68

変換フォーマットの指定 72

DATETIME

変換オプション 62

ロード変換オプション 73

DATETIME データ型 73

ロードの最適化 68

変換フォーマット 74

DATE オプション 62, 71

DATE フォーマット

2 桁の日付の変換 72

dbisql

出力フォーマットの指定 8

対話的なデータの挿入 51

DELIMITED BY オプション 25

F

FROM 句

UPDATE 文 91

I

INSERT LOCATION 文 47

INSERT 文

VALUES オプション 45

インクリメンタル 46

パフォーマンス 46

説明 45

分割されたテーブル 28

Interactive SQL

出力フォーマット 8

iq_bcp

LOAD TABLE USING FILE に置換 27

廃止 27

L

LOAD TABLE

BINARY 35

LOAD TABLE 文

QUOTES オプションの例 25

QUOTES キーワード 25

索引

STRIP キーワード 25
USING CLIENT FILE 27
USING FILE 句 27
バイナリフォーマット 35
バイナリフォーマットデータファイル 35
パフォーマンス 68
整合性制約 30
説明 23
分割されたテーブル 28
LOAD TABLE 内 73
LOB 47

M

MESSAGE LOG
内容 31
例 33

N

NEAREST_CENTURY オプション 72
NULL 73
NULL への変換 76
挿入 45
変換オプション 62, 76
NULLS オプション
dbisql 11
NULL 値
出力 11

P

PIPE_NOT_CONNECTED エラー 23

Q

QUOTES
LOAD TABLE キーワード 25

R

REAL データ型
対応する Adaptive Server と SAP Sybase IQ
データ 82
ROW LOG
内容 32

例 33

S

SAP Sybase IQ
対応するデータ型 Adaptive Server 81
SELECT 文
デリミタ 47
SET 句
UPDATE 文 91
START ROW ID オプション
INSERT 文 46
STRIP
LOAD TABLE キーワード 25

T

TEMP_EXTRACT_NULL_AS_EMPTY 12
TEMP_EXTRACT_NULL_AS_ZERO 12
TIME データ型
ロードの最適化 68

U

unichar Adaptive Server データ型 82
unitext Adaptive Server データ型 82
univarchar Adaptive Server データ型 82
UPDATE 文
ジョイン操作を使用 91
使用 91

V

VALUES オプション
INSERT 文 45
VARCHAR データ
長さ 0 のセルの置換 71

W

WHERE 句
UPDATE 文 91

あ

値

丸め 79

え

エラー

データ変換 86

エラーメッセージ

PIPE_NOT_CONNECTED 23

ファイルへのリダイレクト 11

お

オプション値

トランケーション 12

か

カラムデリミタ

ロード形式オプション 25

カラム、ロード時に設定される 73

カラム幅

挿入の問題 67

く

クライアント

直接のデータロード 27

こ

後続ブランク

削除 20

し

時刻データ型

対応する Adaptive Server と SAP Sybase IQ
データ 82

ジョイン

使用した更新 91

す

スレッド

ロード時の使用 89

せ

整数データ型

対応する Adaptive Server と SAP Sybase IQ
82

ゼロ

NULL への変換 76

そ

挿入変換オプション 62

つ

通貨データ型 82

て

データ

インポート 7

エクスポート 7, 11

クライアント 27

データの入出力フォーマット 8

ロード 7

削除 93

抽出 12

データのインポート

Adaptive Server Enterprise から 47

LOAD TABLE 文 23

バージョン 12 より前の IQ データベース
47

変換エラー 86

データのエクスポート

概要 7

説明 11

データのバルクロード

クライアントデータ 27

データのロード

Adaptive Server Enterprise データ 47

ASCII 変換オプション 69

クライアントデータ 27

名前付きパイプ 23

ファイル指定 23

フォーマットオプション 25

ラージオブジェクト 47

索引

- 概要 7
- 制約違反のロギング 29
- 整合性制約違反 29
- 必要な権限 9
- 変換エラー 86
- 変換オプション 62
- データの修正
 - パーミッション 9
- データの挿入
 - 「挿入」参照 62
- データの抽出
 - オプション 12
 - オプションリスト 12
- 説明 12
- データ型
 - FLOAT 82
 - integer 82
 - REAL 82
 - Adaptive Server unichar 82
 - Adaptive Server unitext 82
 - Adaptive Server univarchar 82
 - 通貨 82
 - 対応する SAP Sybase IQ および Adaptive Server 81
 - 変換 61, 79
 - 変換 between Adaptive Server と SAP Sybase IQ 81
 - 文字 82
 - ロード時の変換 62
- データ抽出
 - ASCII 12
 - アクセスの制御 12
 - オプション 12
 - オプションリスト 12
 - バイナリ 12
 - バイナリ/スワップ 12
 - 説明 12
- テーブル
 - ロード 23
- デリミタ
 - SELECT 文 47
- と
- トランケートされた結果 79
- な
- 名前付きパイプ 23

は

- バイナリ
 - データ抽出 12
- パフォーマンス
 - 挿入 46
 - フラットファイルからのロード 62
- バルクロード, チューニング
 - INSERT 文 28
 - LOAD TABLE 28
 - インデックス 87
 - 環境 88
 - 個別値 87
 - スレッド 89
 - パフォーマンス 87
 - 分割されたテーブル 28
 - マルチスレッド 89
- バルクロードのチューニング
 - INSERT 文 28
 - LOAD TABLE 28
 - インデックス 87
 - 環境 88
 - 個別値 87
 - スレッド 89
 - チューニング 87
 - パフォーマンス 87
 - 分割されたテーブル 28
 - マルチスレッド 89

ひ

- 日付データ型
 - 対応する Adaptive Server と SAP Sybase IQ データ 82

ふ

- ファイル
 - 出力のリダイレクト 11, 12
- プライマリキー
 - データの挿入 57
- フラットファイル
 - ロード元 23
 - ロード変換オプション 62

ブランク

- NULL への変換 76
- 後続ブランクの削除 20

へ

変換オプション

- DATE 71
- DATE フォーマットの指定 72
- DATETIME 73, 74
- パフォーマンス 68
- フラットファイルロード 62
- 長さ 0 のセルの置換 71

ま

- マルチスレッド
- ロード時 89

め

- メッセージ
- ファイルへのリダイレクト 11

メモリ

- キャッシュメモリ 88
- シンプレックスサーバ 88
- マルチプレックスノード 88
- ラージメモリ 88

も

文字データ型

- Adaptive Server unichar 82
- Adaptive Server unitext 82
- Adaptive Server univarchar 82
- 対応する Adaptive Server と SAP Sybase IQ
データ 82

ゆ

- ユーティリティプログラム
- 廃止された iq_bcp 27

ら

- ラージオブジェクトデータ 47
- ラージメモリ
- キャッシュサイズ 88
- 所要量 88
- シンプレックスサーバ 88
- マルチプレックスノード 88

り

- リダイレクト
- ファイルへの出力 11, 12
- リモートデータ
- バルクロード 27
- ロード 27
- リモートデータアクセス
- プロキシテーブル 47

ろ

- ロード
- スレッド使用 89
- パフォーマンス 87
- バルクロード 87
- ロードオプション 25
- ロードの最適化 68
- ロードメモリ
- キャッシュメモリ 88
- シンプレックスサーバ 88
- マルチプレックスノード 88
- ラージメモリ 88
- ロード変換
- 次を参照：変換オプション

