



InterSystems SQL の基礎： JDBC とインターシステムズの データベース

Version 2023.1

2024-01-02

InterSystems SQL の基礎 : JDBC とインターシステムズのデータベース
InterSystems IRIS Data Platform Version 2023.1 2024-01-02
Copyright © 2024 InterSystems Corporation
All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, および TrakCare は、InterSystems Corporation の登録商標です。HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, および InterSystems TotalView™ For Asset Management は、InterSystems Corporation の商標です。TrakCare は、オーストラリアおよび EU における登録商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼働および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

目次

InterSystems SQL の基礎 : JDBC とインターシステムズのデータベース.....	1
1 JDBC : InterSystems IRIS で使用する方法	1
2 JDBC : InterSystems IRIS の Java 接続オプションの一部	1
2.1 JDBC : 共有メモリ接続のユニークな点	2
3 JDBC : 操作例	2
3.1 開始の前に	2
3.2 サンプル・コードの試行	2
4 JDBC の詳細	3

InterSystems SQL の基礎 : JDBC とインターシステムズのデータベース

ここでは、InterSystems JDBC ドライバを使用して InterSystems IRIS® データ・プラットフォーム・インスタンスに接続し、InterSystems IRIS で Java を使用できるようにする方法を紹介します。

1 JDBC : InterSystems IRIS で使用する方法

インターシステムズは、完全互換 (JDBC 4.2)、pure Java、タイプ 4 の JDBC ドライバを提供しています。これは、依存関係を持たない単一のスタンドアロン JAR ファイルです。JDBC に既に習熟していて、JDK 1.8 がインストールされている場合、必要な作業は JDBC ドライバをローカルの CLASSPATH に追加することのみです ("[JDBC : 操作例](#)" を参照)。JDBC の URL (接続文字列) は、以下のとおりです。

```
jdbc:IRIS://ipAddress:superserverPort/namespace
```

これらの変数は、InterSystems IRIS インスタンスのホストの IP アドレス、インスタンスのスーパーサーバのポート、およびインスタンス上のネームスペースを表します。

ローカル・マシン上のインスタンスに接続している (ホスト名 localhost または IP アドレス 127.0.0.1 を使用している) 場合は、共有メモリ接続と呼ばれる特別な高性能のローカル接続を使用できます。共有メモリ接続についての詳細は、"[JDBC : 共有メモリ接続のユニークな点](#)" を参照してください。

このドキュメントの目的は、詳しいことは抜きにして InterSystems IRIS の基本機能を体験していただくことなので、以下の操作例はシンプルなものにしてあります。InterSystems IRIS をプロダクション・システムで使用する際には、例えばセキュリティに関して (それに限りませんが)、多くの異なる処理が必要になります。したがって、ここに示す InterSystems IRIS の操作例と本番での操作を混同しないようにしてください。このドキュメントの最後に示すソースでは、プロダクション環境の InterSystems IRIS で JDBC を使用するために必要な情報が提供されています。

2 JDBC : InterSystems IRIS の Java 接続オプションの一部

InterSystems IRIS の JDBC ドライバは、InterSystems IRIS のコア Java コンポーネントで、標準のリレーショナル (SQL) アクセスをサポートします。このドライバは、Java 呼び出しの場合の接続メカニズムも提供します。このメカニズムでは、ネイティブに格納されている形式でデータにアクセスする InterSystems IRIS Native SDK for Java が使用されます。オブジェクトに基づいた Java 統合向けに、InterSystems IRIS は別の機能、[InterSystems IRIS XEP コンポーネント](#)も提供しています。

このように、InterSystems IRIS は、同じ物理接続とトランザクションのコンテキストを使用して、ネイティブ、リレーショナル、およびオブジェクト指向の複数のパラダイムに従ってデータを操作するユニークな一連の機能を提供します。より複雑なアプリケーションのために、インターシステムズは Hibernate を完全にサポートします。InterSystems IRIS XEP、Hibernate、さらに InterSystems IRIS Spark コネクタといった、あらゆる形態の接続を可能にするものが、InterSystems IRIS JDBC ドライバです。

2.1 JDBC : 共有メモリ接続のユニークな点

他のデータベース・プラットフォームと同様に、リモートの InterSystems IRIS インスタンスへの JDBC 接続では TCP/IP が使用されます。パフォーマンスを最大限に高めるために、InterSystems IRIS は Java 共有メモリ接続も提供します。共有メモリ接続は、InterSystems IRIS インスタンスと同じマシン上で実行される多くの Java アプリケーションで使用できます。

共有メモリ接続は、同じ物理マシン上で実行される JDBC クライアントと InterSystems IRIS のインスタンスによって共有される一時デバイスであり、仮想メモリに格納されます。また、これらの接続には、高いコストのかかる可能性があるカーネル・ネットワーク・スタックの呼び出しは不要です。JDBC クライアントから InterSystems IRIS へのチャンネルを直接使用することにより、JDBC 操作のために極限までの低遅延と高スループットを実現します。

共有メモリの詳細は、“InterSystems JDBC ドライバでの Java の使用法”の“共有メモリ接続”を参照してください。

3 JDBC : 操作例

JDBC と InterSystems IRIS の操作方法と、その操作がどれだけ簡単なのかを示すデモが用意されています。

このコードは、共有メモリ接続によって効率的に処理できる大量のデータを扱うわけではないため、[InterSystems Java 共有メモリ接続](#)のパフォーマンス能力の向上を示すものではないことに注意してください。

3.1 開始の前に

この手順を使用するには、バージョン 1.8 の JDK および選択した Java IDE がインストールされた、操作するためのシステム、および接続先として稼働している InterSystems IRIS インスタンスが必要です。InterSystems IRIS の選択肢としては、いくつかのタイプのライセンス付与されたインスタンスおよび無料の評価版インスタンスがあります。操作しているシステムでインスタンスをホストする必要はありません（ただし、相互のネットワーク・アクセスが必要です）。操作を実行するインスタンスをまだ用意できていない場合にインスタンスのタイプ別の導入方法の詳細を確認するには、“InterSystems IRIS の基礎 : IDE の接続”の“[InterSystems IRIS の導入](#)”を参照してください。同じドキュメントの“[InterSystems IRIS 接続情報](#)”および“[Java IDE](#)”の情報を使用して、IDE を InterSystems IRIS インスタンスに接続します。このデモでは、以下のコードで示しているように **USER** ネームスペースに接続することも、インストールしたインスタンスで作成した別のネームスペースを指定することもできます。

InterSystems IRIS JDBC ドライバ `intersystems-jdbc-version.jar` をローカルの CLASSPATH に追加することも必要です。このファイルは、“[InterSystems IRIS Driver Packages](#)” ページからダウンロードできます。InterSystems IRIS をローカル・マシンまたはアクセス可能な他のマシンにインストールしている場合、このファイルは、`install-dir¥dev¥java¥lib¥JDK18` にあります。install-dir は、InterSystems IRIS のインストール・ディレクトリです。

3.2 サンプル・コードの試行

サンプル・コードを切り取って IDE に貼り付け、“InterSystems IRIS の基礎 : IDE の接続”で説明している該当の**インスタンス用の接続設定**で、url 変数、connection 変数、ユーザ名、およびパスワードを更新します。

Java

```
import java.sql.*;

public class JDBCExample {
    public static void main(String[] str) throws Exception {
        String url = "jdbc:IRIS://127.0.0.1:1972/USER";

        Class.forName("com.intersystems.jdbc.IRISDriver");
        Connection connection = DriverManager.getConnection(url, "_SYSTEM", "SYS");
        // Replace _SYSTEM and SYS with a username and password on your system
    }
}
```

```

String createTable = "CREATE TABLE People(ID int, FirstName varchar(255), LastName varchar(255))";

String insert1 = "INSERT INTO People VALUES (1, 'John', 'Smith')";
String insert2 = "INSERT INTO People VALUES (2, 'Jane', 'Doe')";
String query = "SELECT * FROM People";

Statement statement = connection.createStatement();
statement.executeUpdate(createTable);
statement.executeUpdate(insert1);
statement.executeUpdate(insert2);
ResultSet resultSet = statement.executeQuery(query);
System.out.println("Printing out contents of SELECT query: ");
while (resultSet.next()) {
    System.out.println(resultSet.getString(1) + ", " + resultSet.getString(2) + ", " +
resultSet.getString(3));
}
resultSet.close();
statement.close();
connection.close();
}
}

```

接続とクエリが正常に完了すると、コンソール・ウィンドウに SELECT クエリの結果が表示されます。

4 JDBC の詳細

JDBC、InterSystems IRIS のその他の Java 相互運用性テクノロジー、およびその他の関連トピックの詳細は、以下を参照してください。

- ・ [“InterSystems IRIS での Java JDBC の使用法” の “InterSystems Java 接続オプション”](#) – JDBC ドライバによって有効になるすべての InterSystems IRIS Java テクノロジーの概要
InterSystems JDBC ドライバでの Java の使用法 – インターシステムズのドキュメント：手順を追った JDBC の使用法の説明
- ・ [Java Overview](#) – インターシステムズのオンライン学習資料 (概要ビデオ)
- ・ [InterSystems IRIS デモ：XEP による Java オブジェクト永続性](#) – インターシステムズのドキュメント：Java XEP デモ
- ・ [InterSystems XEP による Java オブジェクトの永続化](#) – インターシステムズのドキュメント：手順を追った XEP の使用法の説明
- ・ [InterSystems Native SDK for Java の使用法](#) – インターシステムズのドキュメント：InterSystems Native SDK
- ・ [Java サード・パーティ API 用 InterSystems 実装リファレンス](#) – インターシステムズのドキュメント：JDBC、Hibernate、および Spark を使用した InterSystems IRIS への接続
- ・ [InterSystems Spark コネクタの使用法](#) – インターシステムズのドキュメント：Apache データ・ソースとしての InterSystems IRIS の使用
- ・ [Hibernate と JDBC の比較](#) – Stack Overflow の記事

