

記事

[Mihoko Iijima](#) · 2021年3月19日 3m read

## 複数のSQL 文を実行する方法

これは [InterSystems FAQ サイト](#) の記事です。

複数の SQL 文を実行する GUI はありませんが、複数の SQL 文を含むファイルを作成しファイルをインポートしながら SQL を実行する方法で対応できます。

ご利用バージョンによって使用するユーティリティメソッドが異なります。

**2020.3以降をご利用の場合は** [%SYSTEM.SQL.Schema](#) クラスの [ImportDDL\(\)](#) メソッドを使用します。

**2020.2以前をご利用の場合は** [%SYSTEM.SQL](#) クラスの [DDLImport\(\)](#) メソッドを使用します。

### 1) バージョン 2020.3 以降での方法：[%SYSTEM.SQL.Schema](#) クラスの [ImportDDL\(\)](#) メソッドの利用

インポート用ファイルに記述するSQL文が複数行ある場合は、記述する SQL の後ろに GO 文を記述する必要があります。

インポートファイル例は以下の通りです。

```
INSERT INTO Test2.Person (Name) values('test1')
go
INSERT INTO Test2.Person (Name) values('test2')
go
INSERT INTO Test2.Person (Name) values('test3')
go
```

[%SYSTEM.SQL.Schema](#) クラスの [ImportDDL\(\)](#) メソッドの利用例は以下の通りです。

第1引数：インポート対象ファイルのフルパス

第2引数：インポート時のログ出力用ファイルのフルパス

第3引数：DDLモードを指定（指定文字詳細はクラスリファレンスをご参照ください）

```
USER>do $system.SQL.Schema.ImportDDL(file,"/irisdev/src/ddlimport.log","IRIS")
Importing SQL Statements from file: /irisdev/src/inserts.txt
Recording any errors to principal device and log file: /irisdev/src/ddlimport.log
SQL statement to process (number 1):
  INSERT INTO Test2.Person (Name) values('test1')
  Preparing SQL statement...
  Executing SQL statement...
DONE
```

## 複数のSQL文を実行する方法

Published on InterSystems Developer Community (<https://community.intersystems.com>)

---

```
SQL statement to process (number 2):
  INSERT INTO Test2.Person (Name) values('test2')
  Preparing SQL statement...
  Executing SQL statement...
DONE
SQL statement to process (number 3):
  INSERT INTO Test2.Person (Name) values('test3')
  Preparing SQL statement...
  Executing SQL statement...
DONE
Elapsed time: .010847 seconds
USER>
```

詳細はドキュメントもご参照ください。

[InterSystems SQL のインポート](#)

---

## 2) バージョン 2020.2 以下での方法 : [%SYSTEM.SQL](#) クラスの [DDLImport\(\)](#) メソッドの利用

インポート用ファイルに記述するSQL文が複数行ある場合は、記述する SQL の後ろに GO 文を記述する必要があります。

GO 文の他に、SQLの文末に任意の区切り文字を指定することもできます。

以下実行例は、[DDLImport\(\)](#) の7番目の引数に指定できる文末の区切り文字にセミコロン (;) を指定した方法の実行例です。

以下のような query.txt を用意しているとします。  
文末区切り文字は ";" を指定しています。

```
C:\temp\query.txt
insert into Sample.Person(Name,SSN) values('test1','999-99-9901');
insert into Sample.Person(Name,SSN) values('test2','999-99-9902');
insert into Sample.Person(Name,SSN) values('test3','999-99-9903');
insert into Sample.Person(Name,SSN) values('test4','999-99-9904');
```

[DDLImport\(\)](#) の7番目の引数に文末区切り文字を指定しながら以下のように実行します。

```
DO $SYSTEM.SQL.DDLImport("CACHE",$Username,"c:\temp\query.txt",,,,";")
```

第2引数は必須で、SQLを実行するユーザ名を指定します。例では、現在のユーザを \$Username で取得し指定しています (テーブルにアクセスできるユーザを指定してください)。

詳細は、以下のドキュメントをご覧ください。

---

[SQLのインポートについて【IRIS】](#)

[SQLのインポートについて](#)

[#SQL #ヒントとコツ #Caché #Ensemble #HealthShare #InterSystems IRIS #InterSystems IRIS for Health](#)

---

ソースURL:

<https://jp.community.intersystems.com/post/%E8%A4%87%E6%95%B0%E3%81%AEsql-%E6%96%87%E3%82%92%E5%AE%9F%E8%A1%8C%E3%81%99%E3%82%8B%E6%96%B9%E6%B3%95>