

RPGⅢプログラマー向け

SQLの有効活用

Agenda

1. RPGにおけるファイルアクセス命令
2. SQLの基本構文（条件文節）
3. SQL対比
4. SQL（Db2 For i）利用のメリット
5. STRSQL
6. パフォーマンス
7. SQL使用のRPG基礎知識
8. SQLコーディング上の考慮点等
9. SQL構文（おさらい）
10. SQL記述例

1. RPGにおけるファイルアクセス命令

【RPGにおけるファイルアクセス命令】

- SETLL、SETGT
- READ、READE、READP、READPE
- CHAIN
- WRITE、UPDATE、DELETE

2. SQLの基本構文（条件文節）

【SQLの文法（条件文節）】

任意（必要に応じて指定）

WHERE

条件

GROUP BY

グルーピング項目

ORDER BY

読み込み順

条件 … 検索条件

グルーピング項目 … グルーピングしたい項目名

読み込み順 … 読み込み順の項目名

3. SQL対比

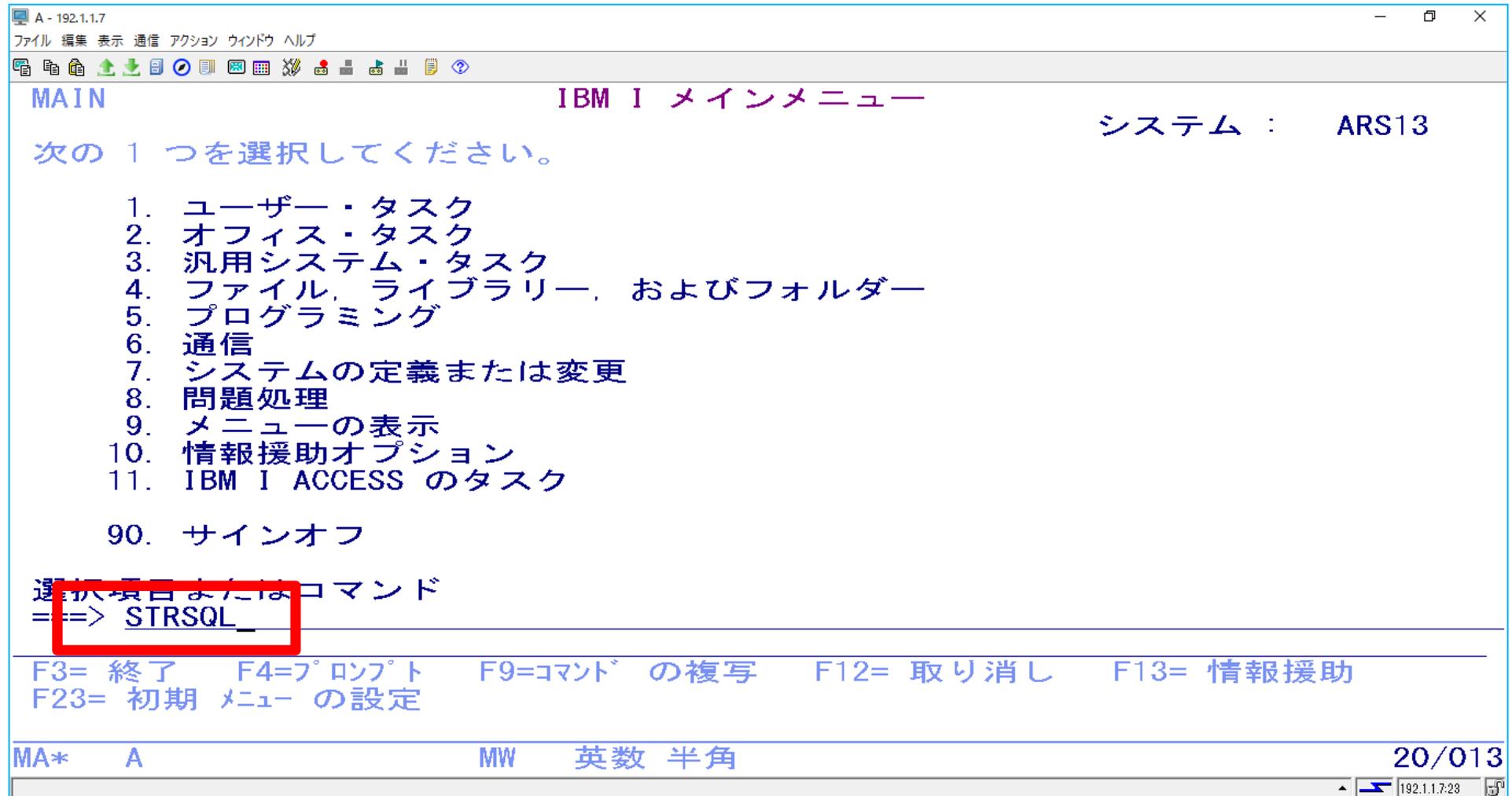
命令	SQLでの対応
SETLL、SETGT	WHERE文節で読みたいデータを限定
READ、READE、READP、READPE	ORDER BY 文節で読み込み順を指定
CHAIN	WHERE文節で読みたいデータを限定
WRITE、UPDATE、DELETE	WHERE文節で条件を指定して一括での処理が可能

4. SQL (Db2 For i) 利用のメリット

- 論理ファイルの数が激減（キー、並び順、結合）
- UPDATE、DELETEが複数レコードを一括で処理可能
- 通常にCRTPF、CTRLFで作成されたファイルでもSQL操作が可能
- 既存RPGⅢシステムでも、SQLを使用可能
- OPNQRYFILの代替
- STRSQLは（慣れてしまえば、）RUNQRY、WRKQRYよりも有用
- プログラム検証に便利
- 他システムとのインターフェースに利用
- パフォーマンス

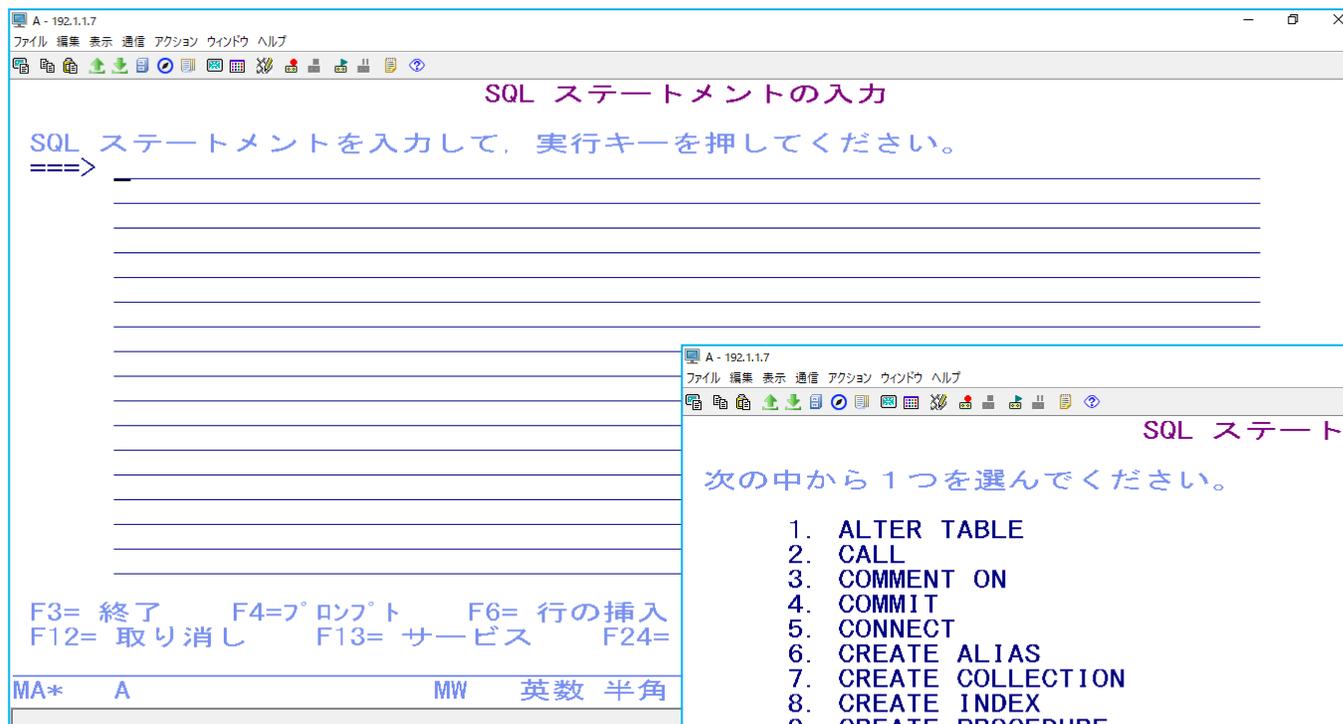
5. STRSQL

【実際の使用例 (STRSQL)】

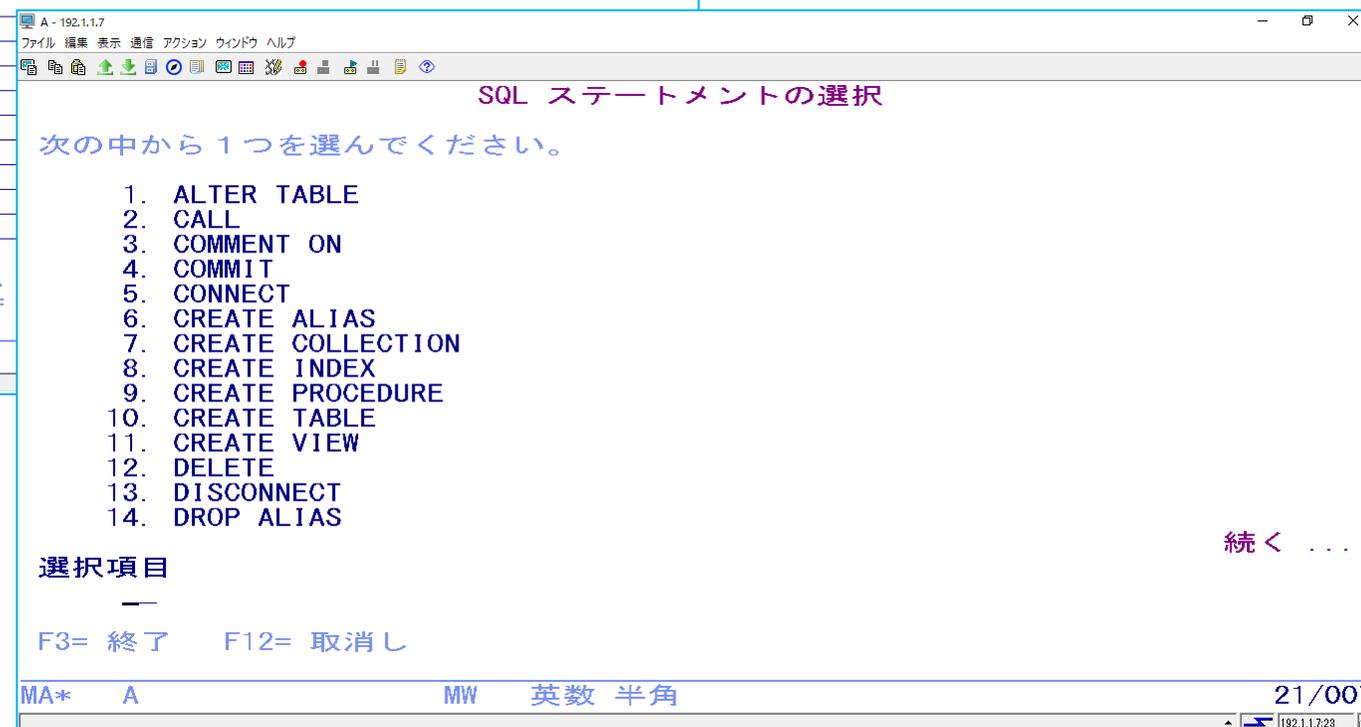


5. STRSQL

【実際の使用例 (STRSQL)】



F4 押下



続く ...

5. STRSQL

【実際の使用例 (STRSQL)】

SQL ステートメントの入力

SQL ステートメントを入力して、実行キーを押してください。
 ==> SELECT * FROM CDDMRY/CDMSO_

データの表示

データの幅 : 384
 桁移動

行の位置指定 : 1 2 3 4 5 6 7 8 9 10 11 12 13

本部コード	本支店コード	部コード	課コード	状態フラグ	組織名称 (カナ)	組織名称 (漢字)	組織総称 (略)
A				0			
A	0			0			
A	1			0			
A	0	10		0	カノホンソウカム	長野本店総務課	長野本店総務
A	0	19		0	カノケンカクゾウブ	長野建築本店 工務部	長野建築本店
A	0	20		0	カノホンエイヨウ	長野本店営業部	長野本店営業
A	0	30		0	カノホンコクム	長野本店工務部	長野本店工務
A	0	72		0	カノエイヨウショ	上田営業所	上田営業所
A	0	74		0	サノエイヨウショ	佐久営業所	佐久営業所
A	1	00		0	コソゾブシャジザイカ	工務部・社寺・在宅部	工務部・社寺
D	0			0			
D	1			0			
D	0	10		0	コクムカク	工務統括部	工務統括部
D	0	19		0	ドホクエイカクゾウブ	土木事業本部 工務部	土木事業本部
D	0	20		0	ドホクエイヨウ	土木事業本部 営業部	土木事業本部
D	0	21		0		不明	不明
D	0	30		0	コクム	工務部	工務部
D	1	00		0	コソゾブ	工務部	工務部

F3= 終了 F4=プロンプト F6= 行
 F12= 取り消し F13= サービス

F3= 終了 F12= 取り消し F19= 左 F20= 右 F21= 分割 F22= 幅 80

MA* A MW 英数 半角 03/032

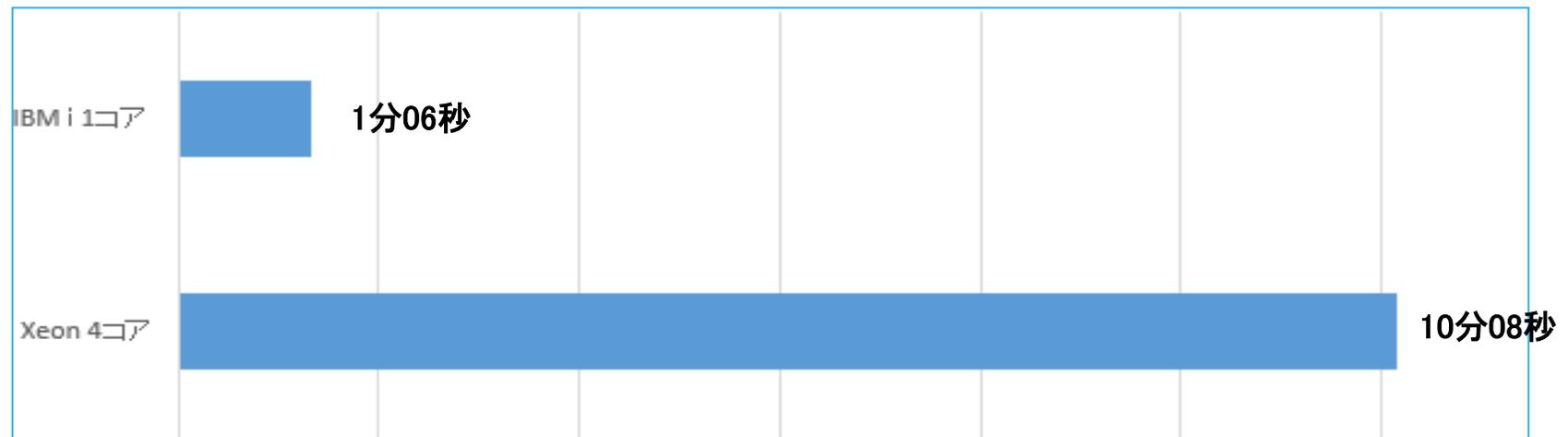
6. パフォーマンス

【パフォーマンス比較】

IBMi vs PC系メジャーDB

- ・ IBM i は、1コア
- ・ SQL Serverは、Windows2008で、インテル Xeon 4コア

お客様のアプリケーション（Visual Basic）とSQL（ODBC）接続して、約10万件の受注バッチの処理を実行



7. SQL使用のRPG基礎知識

【ソース・メンバー属性】

	従来(RPGⅢ)	ILE-RPG
• SQLを含まないRPG	RPG	RPGLE
• SQL組込のRPG	SQLRPG	SQLRPGLE
• SQLを含まないCOBOL	CBL	CBLLE
• SQL組込のCOBOL	SQLCBL	SQLCBLLE

【コンパイル】

- SQLを含むものも含まないものも同じコマンド（PDM：14番）でコンパイル可能です。
（モジュール作成の場合は、PDM：15番も同様）

8. SQLコーディング上の考慮点等

- SELECTやFETCHで使用するフィールドを個々に定義して指定する事もできますが、一括指定する事もできます（定義仕様でのEXTFILEなどのキーワード）
- SQLのみで使用するファイルは、F仕様書に記載する必要はありません。
- 同一プログラム内で、SQLとREAD、CHAINなどを混在させる事も可能です
- UPDATEやDELETEなどファイル更新する場合は、SET OPTIONでCOMMITの指定が必要
- SET OPTIONの記述位置に注意

9. SQL構文

【SQLの文法 (READ)】

SELECT **項目** FROM **ファイル**

以下、オプション

WHERE

条件

GROUP BY

グルーピング項目

ORDER BY

読み込み順

項目 … 読みたい項目名
(*指定は全項目)

ファイル … 読みたいファイル名
(ライブラリ名/ファイル名)

条件 … 検索条件

グルーピング項目 … グルーピングしたい項目名

読み込み順 … 読み込み順の項目名

9. SQL構文

【SQLの文法 (READ)】

【項目】

- ・ 全項目を読みたい場合は、*****を指定
- ・ 項目を指定した場合は、**項目名, 項目名, 項目名, …**と指定
- ・ その他、以下の関数（抜粋）が利用可能

最大値を取得 **MAX (項目名)** … WRKQRYの計算機能（最大）

最小値を取得 **MIN (項目名)** … WRKQRYの計算機能（最小）

値を集計 **SUM (項目名)** … WRKQRYの計算機能（合計）

件数を取得 **COUNT (*)** … WRKQRYの計算機能（カウント）

9. SQL構文

【SQLの文法 (READ)】

【条件】

- レコードを読む条件を演算子で指定 例. **WHERE A = 'ABC'**
- 利用可能な演算子は右表の通り

例.

A LIKE '%ABC%'

B BETWEEN 100 AND 200

演算子	説明
=	等しい
>	大きい
<	小さい
>=	大きい、もしくは等しい
<=	小さい、もしくは等しい
!=	等しくない
<>	等しくない
AND	2つの条件を結合し、両方の条件が真
OR	2つの条件のうち、どちらか一方が真
NOT	式の結果を反転
BETWEEN a AND b	対象のフィールドが a と b の範囲内
IN	対象のフィールドが式の一覧の1つに一致
LIKE	対象のフィールドがパターンに一致

9. SQL構文

【SQLの文法 (READ)】

【グルーピング】

- ・ 集計したい項目を指定 **GROUP BY** 項目名、項目名 …

例. 部、課ごとの基本給の集計値

```
SELECT SUM(基本給) FROM TABLE GROUP BY 部コード, 課コード
```

9. SQL構文

【SQLの文法（READ）】

【読み込み順】・・・ WRKQRYの分類

- ・読み込み順を指定 **ORDER BY 項目名、項目名・・・ (DESC)**
(**DESC** は降順に読み込みたい場合に指定)

例. 部、課ごとの基本給の集計値を部課コード昇順に読み込む

```
SELECT SUM(基本給) FROM TABLE
```

```
GROUP BY 部コード, 課コード
```

```
ORDER BY 部コード, 課コード
```

9. SQL構文

【SQLの文法 (WRITE)】

INSERT INTO ファイル
VALUES (値1、値2、値3…)

ファイル … 出力したいファイル名
(ライブラリ名/ファイル名)

値1 … 出力項目に該当する値を記述
変数でも可

9. SQL構文

【SQLの文法 (UPDAT)】

UPDATE ファイル

SET 項目1 = 値1、

項目2 = 値2...

以下、オプション

WHERE 条件

```
0057.00 C/EXEC SQL
0058.00 C+ UPDATE PC3PROCESS SET AV_PE00003 = NULL WHERE ACTIV00003 <> '4'
0059.00 C+ AND TRXNO = :P@SNO
0060.00 C/END-EXEC
0061.00 C*
0062.00 C*SQLエラーモニター
0063.00 C/EXEC SQL
0064.00 C+ WHENEVER SQLERROR GOTO LBLERR
0065.00 C/END-EXEC
```

9. SQL構文

【SQLの文法 (DELETE)】

DELETE FROM ファイル

以下、オプション

WHERE 条件

10. SQL記述例

【SQLの文法（INSERT、UPDATE、DELETEの場合）】

RPG組み込みで使用する場合は、初期処理で下記の記述が必要

C/EXEC SQL

C+ SET OPTION

C+ commit=*none,

C+ datfmt=*iso

C/END-EXEC

10. SQL記述例

【実際の使用例 (ILERPG)】

SQL文の指定

1109.00	C
1110.00	C
1111.00	C
1112.00	C
1113.00	C
1114.00	C
1115.00	C
1116.00	C
1117.00	C
1118.00	C
1119.00	C
1120.00 *002	C
1121.00	C
1122.00	C
1123.00	C
1124.00	C

```

EVAL    SQL = ' +
        SELECT +
          U2DNO +
        , U2TRD +
        , U2AVCU +
        , U2AVGO +
        , U2TKCO +
        , U2TKCD +
        , U2TKSC +
        , "" AS SRCD +
        , "" AS SRSC +
        , U2ASYC +
        FROM SDTU2L01 WHERE U2PRCD = ' +
        "" + D1PRCD + "" +
        ' AND U2DNO >= ' +
        "" + D1DNO + ""
    
```

10. SQL記述例

【実際の使用例 (ILERPG)】

SQLの前処理
(最適化)

1167.00	C/EXEC SQL
1168.00	C+ PREPARE QRY FROM :SQL
1169.00	C/END-EXEC
1170.00	C*
1171.00	C*SQLエラーモニター
1172.00	C/EXEC SQL
1173.00	C+ WHENEVER SQLERROR GOTO LBLERR
1174.00	C/END-EXEC
1175.00	C*
1176.00	C*ファイルアクセス用カーソル定義
1177.00	C/EXEC SQL
1178.00	C+ DECLARE QRY1 SCROLL CURSOR FOR QRY
1179.00	C/END-EXEC
1180.00	C*
1181.00	C*カーソルを開く
1182.00	C/EXEC SQL
1183.00	C+ OPEN QRY1
1184.00	C/END-EXEC
1185.00	C*
1186.00	C*=> S01処理開始

10. SQL記述例

【実際の使用例 (ILERPG)】

読み込み用

カーソルの定義

```

1167.00 C/EXEC SQL
1168.00 C+ PREPARE QRY FROM :SQL
1169.00 C/END-EXEC
1170.00 C*
1171.00 C*SQLエラーモニター
1172.00 C/EXEC SQL
1173.00 C+ WHENEVER SQLERROR GOTO LBLERR
1174.00 C/END-EXEC
1175.00 C*
1176.00 C*ファイルアクセス用カーソル定義
1177.00 C/EXEC SQL
1178.00 C+ DECLARE QRY1 SCROLL CURSOR FOR QRY
1179.00 C/END-EXEC
1180.00 C*
1181.00 C*カーソルを開く
1182.00 C/EXEC SQL
1183.00 C+ OPEN QRY1
1184.00 C/END-EXEC
1185.00 C*
1186.00 C*=> S01処理開始

```

10. SQL記述例

【実際の使用例 (ILERPG)】

カーソルのOpen

```
1180.00 C*  
1181.00 C*カーソルを開く  
1182.00 C/EXEC SQL  
1183.00 C+ OPEN QRY1  
1184.00 C/END-EXEC  
1185.00 C*  
1186.00 C*=> S01処理開始  
1187.00 C 1 DO 500  
1188.00 C*  
1189.00 C*行を取得  
1190.00 C/EXEC SQL  
1191.00 C+ FETCH NEXT FROM QRY1 INTO :D2RECX  
1192.00 C/END-EXEC
```

10. SQL記述例

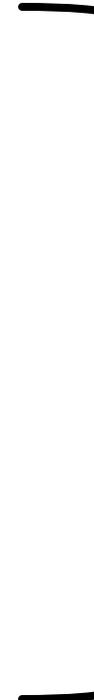
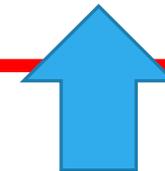
【実際の使用例 (ILERPG)】

レコードの
読み込み

```

1187.00 C 1 DO 500
1188.00 C*
1189.00 C*行を取得
1190.00 C/EXEC SQL
1191.00 C+ FETCH NEXT FROM QRY1 INTO :D2RECX
1192.00 C/END-EXEC
1193.00 C*
1194.00 C SQLCOD IFNE 0
1195.00 C LEAVE
1196.00 C ENDIF
1197.00 C*
1198.00 C ADD 1 WKCNT
1199.00 C EXSR SBSET
1200.00 C*
1201.00 C ENDDO

```



10. SQL記述例

【実際の使用例 (ILERPG)】

レコードの

読み込み

フィールドの

定義

0136.00	D D2RECX	DS	
0137.00	D D2DNO		8A
0138.00	D D2TRD		8S 0
0139.00	D D2AVCU		20A
0140.00	D D2AVGO		2S 0
0141.00	D D2TKCO		20A
0142.00	D D2TKCD		7A
0143.00	D D2TKSC		3A
0144.00	D D2SRCD		7A
0145.00	D D2SRSC		3A
0146.00 : 002	D D2SYCD		6A

10. SQL記述例

【実際の使用例 (ILERPG)】

カーソルのClose

1204.00	C*カーソルを閉じる
1205.00	C/EXEC SQL
1206.00	C+ CLOSE QRY1
1207.00	C/END-EXEC
1208.00	C*

10. SQL記述例

【実際の使用例 (ILERPG)】

```
dcl-proc updateUcctlp;
```

```
// 処理=1:データ作成
```

```
if (p0prc = 1);
```

```
  exec sql
```

```
    INSERT INTO ucctlp
```

```
    VALUES (:p0tcod, :p0sncd, :p0wsid, current_timestamp);
```

```
// 処理=2:データ削除
```

```
elseif (p0prc = 2);
```

```
  exec sql
```

```
    DELETE FROM ucctlp a
```

```
    WHERE
```

```
      a.uttcod = :p0tcod AND a.utsncd = :p0sncd;
```

```
endif;
```

```
end-proc;
```

10. SQL記述例

【実際の使用例 (ILERPG)】

処理結果	SQLCODE	...	0	処理正常
			100	対象レコードなし

質疑応答