

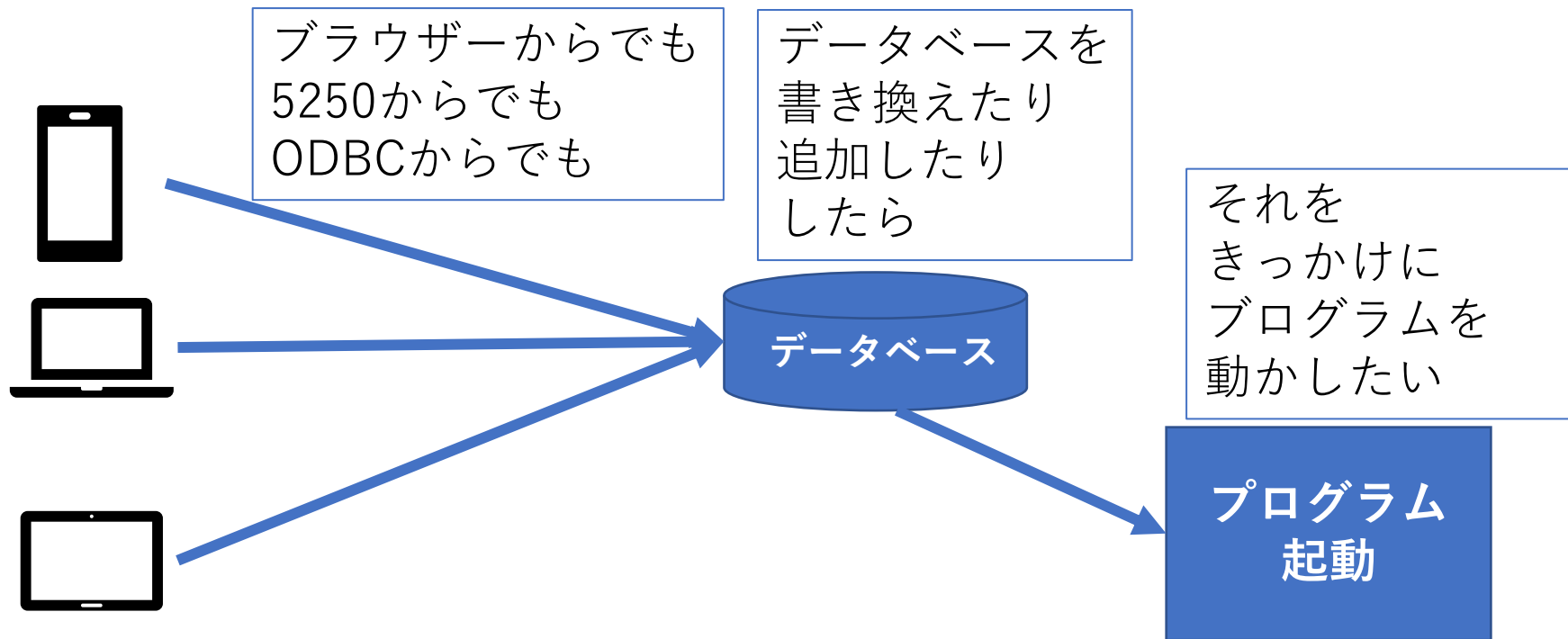
# IBM i World 2022

IBM i コンテンツ (2022年7月版)

## Db2 for i トリガー機能のご紹介

日本アイ・ビー・エム株式会社  
テクノロジー事業本部  
IBM Powerテクニカルセールス  
澤田英寿

# こんな機能が欲しかった!



Db2 for i トリガー機能で、できます!

## 目次

1. Db2 for i トリガー機能のご紹介
2. Db2 for i トリガー機能の設定方法（外部トリガーの設定）
3. Db2 for i トリガー機能の設定方法（SQLトリガーの設定）
4. Db2 for I トリガー機能を使ってみよう（外部トリガーの例）
5. Db2 for I トリガー機能を使ってみよう（SQLトリガーの例）
6. 補足情報

# 1. Db2 for i トリガー機能のご紹介 (1)

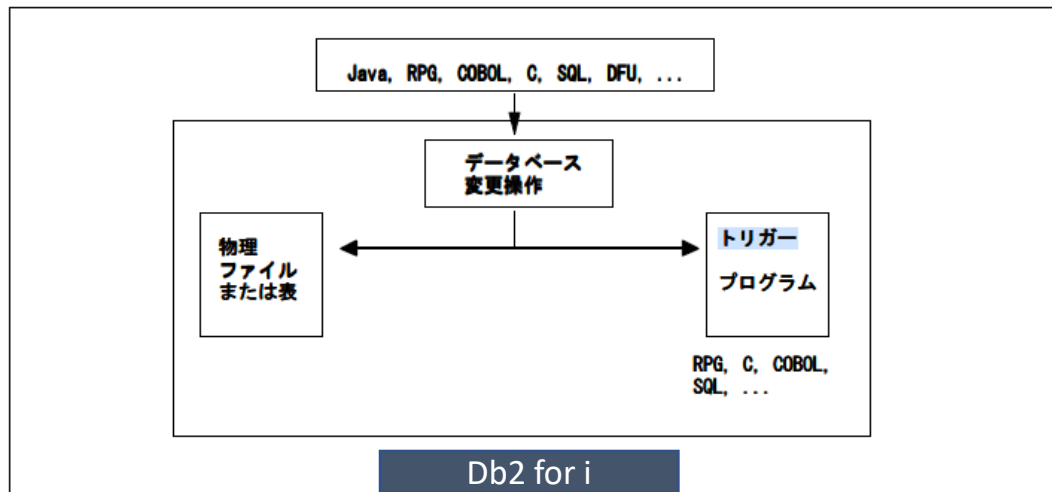
## トリガー機能のご紹介

- トリガーは、データベース・ファイルに関連したユーザー作成のプログラムです。ユーザーは、更新、削除、挿入の操作のためのトリガーを定義できます。データ交換に使用されているインターフェースには関係なく、操作が実行されるたびに、Db2 for i によってトリガー・プログラムが自動的に起動され、そのロジックが実行されます。
- アプリケーション環境からは完全に独立して、データベース・レベルで複雑なルールをインプリメントできます。データベース設計の中で、さまざまな目的にトリガーを利用できます。たとえば、データ妥当性検査や監査証跡が挙げられます。Db2 for i は、下記の2つのタイプのトリガーをサポートしています。
  - ✓ 外部トリガー
  - ✓ SQL トリガー
- 1つの物理ファイルに対して、最大300のトリガーを定義できます。

# 1. Db2 for i トリガー機能のご紹介 (2)

## トリガー機能の概要

- IBM i サーバー上で、任意のサポートされる高水準言語コンパイラーを使用して、トリガー・プログラムを作成できます。SQL ステートメントおよび他の高水準言語を組み込むことができます。CL 言語を使用してトリガーをコーディングすることもできます。
- トリガーを作成した後、それを物理ファイル/表と関連付けることができます。定義の有効範囲はファイル・レベルです。複数のメンバーをもつ物理ファイル(たとえば、各月ごとのメンバーをもつ年間売上ファイル)に対してトリガーを定義すると、メンバーのデータが変更されるたびにトリガーが起動されます。



# 1. Db2 for i トリガー機能のご紹介 (3)

## トリガープログラムの業務用途/メリット

- ✓ ビジネス・ルールを適用する（その複雑さに関係なく）  
たとえば、データベースに注文を入力するたびに、該当するカスタマーのクレジット利用歴が悪くないかどうかを確認できます。受注表に関連付けたトリガーは、このチェックを一貫して実行し、適切な処置を取ることができます。
- ✓ データ妥当性検査と監査証跡  
営業担当者が受注を入力するたびに、その営業担当者が実際にそのカスタマーに割り当てられていることを確認することが必要になる場合があります。違反の試行も追跡したいと思います。この場合も、受注表に対するトリガーを起動して、妥当性検査を実行し、個別の表の違反者を追跡できます。
- ✓ 既存のアプリケーションと最新テクノロジーを統合する  
会社は、受注を受け付けると、確認の EDI をカスタマーに送信します。トリガーは、既存の受注アプリケーションと IBM i サーバーの新規 EDI サービスを統合するのに理想的なソリューションです。カスタマーからの受注を確認する E メールを送信することもできます。
- ✓ 異なるデータベース表の間のデータ整合性を保つ  
この分野では、トリガーは、参照保全やチェック制約サポートを補完することができます。データベース内のデータが変更された場合、トリガーは、はるかに広範囲に、より強力なデータ妥当性検査およびビジネス・アクションを実行できます。

# 1. Db2 for i トリガー機能のご紹介 (4)

## システム上でトリガー機能を使用する利点

- ✓ アプリケーション開発の高速化  
トリガーはデータベースに保管されるため、それぞれのデータベース・アプリケーションごとにトリガーの処置をコード化する必要がありません。
- ✓ 業務規則のグローバルな実行。  
トリガーをいったん定義すると、データベースを使用するどのアプリケーションにも再使用できます。
- ✓ 保守の簡素化  
業務方針が変更になった場合でも、各アプリケーション・プログラムではなく、該当するトリガー・プログラムを変更するだけで済みます。
- ✓ クライアント/サーバー環境でのパフォーマンスの向上  
すべての規則はサーバー内で実行され、結果が戻されます。クライアント・アプリケーションは、サーバー側のロジックを起動するのに特定のコードを必要としません。通信回線のトラフィックが削減されるので、アプリケーションのパフォーマンス上の利益も得られます。

# 1. Db2 for i トリガー機能のご紹介 (5)

## 外部トリガー

- 外部トリガーとは、トリガープログラムを作成し、物理ファイルに登録するタイプのトリガーです。
- 外部トリガーの場合、\*PGM オブジェクトを作成するためにサポートされている任意の高水準言語を使用して、一連のトリガー・アクションを含むプログラムを定義できます。トリガー・プログラムに SQL を組み込むこともできます。
- 外部トリガーを定義するには、物理ファイルにトリガーを追加する ADDPFTRG CL コマンドを使用して、それを表に追加する必要があります。Navigator for iを使用して追加することもできます。
- トリガーを表に追加する際には、次のことが必要です。
  1. 表を識別する。
  2. 操作の種類を識別する。
  3. 必要なアクションを実行するプログラムを識別する。
  4. トリガーに固有の名前を付ける、またはシステムが固有の名前を生成する。



# 1. Db2 for i トリガー機能のご紹介 (6)

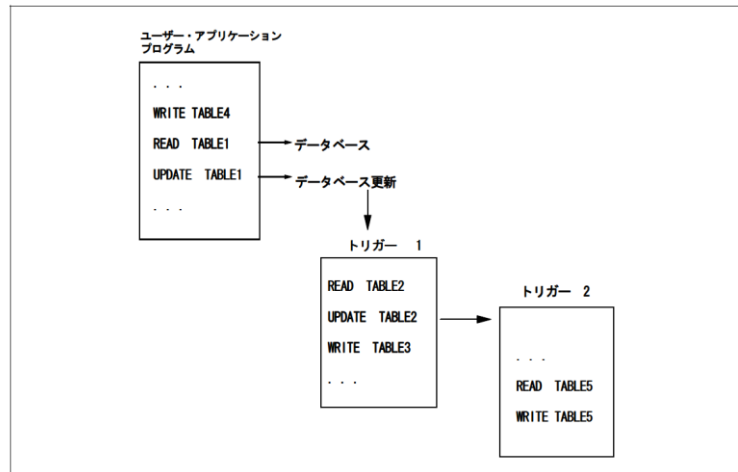
## SQLトリガー

- SQLトリガーは、CREATE TRIGGER SQL ステートメントを指定して作成します。
- テストおよびアクションを実行するプログラムは、SQL ステートメントを使用して書かれます。SQL CREATE TRIGGER ステートメントは、データベース管理システムに対して、挿入、更新、または削除操作が実行されるたびに表グループを制御、モニター、および管理する手段を提供します。挿入、更新、または削除操作が実行されるたびに、SQL トリガーで指定されたステートメントが実行されます。
- トリガーの実行時にストアド・プロシージャまたはユーザー定義関数を呼び出して、追加処理を行うこともできます。
- ストアド・プロシージャとは異なり、SQL トリガーはアプリケーションから直接呼び出すことはできません。代わりに、SQL トリガーは、トリガーの対象となる挿入、更新、または削除操作の実行時に、データベース管理システムによって呼び出されます。SQL トリガーの定義はデータベース管理システムに保管され、トリガーが定義された SQL 表の変更時にデータベース管理システムによって呼び出されます。

# 1. Db2 for i トリガー機能のご紹介 (7)

## トリガーの考慮点

- トリガーを設定した後は、アプリケーション・プログラマーやエンド・ユーザーがそれを回避することができません。トリガーが起動されると、制御はアプリケーション・プログラムからデータベース・マネージャーに移ります。オペレーティング・システムは、トリガー・プログラムにコーディングされた仕様を実行して、ユーザーが設計したアクションを実行します。アプリケーションは、トリガーが終了するのを待った後、制御を再取得します。
- トリガーは、他のトリガーを呼び出すこともできます。図に示した例では、TABLE1 を更新するとトリガーが起動され、データベース表 TABLE2 が更新されます。この操作によって、2 番目のトリガーが起動されます。



## 2. Db2 for i トリガー機能の設定方法 (外部トリガーの設定)

### トリガーの追加 (ADDPFTRG)

- 物理ファイルへのトリガー追加 (ADDPFTRG) コマンドは、トリガー・プログラムと物理ファイルに関連付けます。この関連が確立されると、Db2 for i は、物理ファイル、物理ファイルのメンバー、および物理ファイル上に作成された論理ファイル、またはSQLによって作成されたビューに対して変更操作が実行されるたびに、トリガー・プログラムを呼び出します。
- 1つのデータベースには300までのトリガーを関連づけることができます

物理ファイル・トリガーの追加 (ADDPFTRG)

選択項目を入力して、実行キーを押してください。

物理ファイル	> TOKMSP	名前
ライブラリ	> SAWADALIB	名前, *LIBL, *CURLIB
トリガー時間	> *BEFORE	*BEFORE, *AFTER
トリガー・イベント	> *INSERT	*INSERT, *DELETE, *UPDATE, ...
プログラム	> AAAAPGM	名前
ライブラリ	> SAWADALIB	名前, *LIBL, *CURLIB
トリガーの置き換え	> *NO	*NO, *YES
トリガー	> TESTTRG	
-----		
トリガー・ライブラリー	> *FILE	名前, *FILE, *CURLIB
反復変更許可	> *NO	*NO, *YES

## 2. Db2 for i トリガー機能の設定方法 (外部トリガーの設定)

### トリガーの追加 (ADDPFTRG) (解説)

1. 物理ファイル、ライブラリーの指定：トリガーを設定するファイル/ライブラリーの指定
2. トリガー時間：\*BEFOREか\*AFTERのどちらかを指定する。  
\*BEFOREなら、トリガー・プログラムは、指定された物理ファイルに対する変更操作の前に呼び出されます。  
\*AFTERなら、トリガー・プログラムは、指定された物理ファイルに対する変更操作の後に呼び出されます。
3. トリガー・イベント：  
トリガー・プログラムを呼び出すイベント（物理ファイルに対する変更操作を指定します。出されるコマンドごとに指定できるイベントは1つだけです。\*INSERT、\*DELETE、\*UPDATE、\*READの1つを指定する。
4. トリガー・プログラム、ライブラリー：  
指定されたイベントが物理ファイルで起こった時に呼び出されるプログラムを指定します。プログラムはシステムに存在していなければならない、オブジェクト・タイプ\*PGMでなければなりません。
5. トリガーの置き換え：  
同じトリガー・イベントおよびトリガー時点をもつトリガーがあった時に、既存のトリガーが追加されるトリガーによって置き換えられるかどうかを指定します。\*YESの場合は、既存のトリガーは置き換えられます。トリガー名に対して\*GENが指定されていて、時刻およびイベントがただ1つの項目と一致した場合には、そのトリガーが置き換えられます。トリガー名が指定されていて、それが既存の項目と一致した場合には、そのトリガーが置き換えられます。指定されたトリガー名をもつトリガーが存在していない場合には、新しいトリガーが物理ファイルに追加されます。
6. トリガー：  
追加されるトリガーの名前を指定します。\*GENの場合は、システムがトリガー名を生成します。自動生成よりも、分かり易い名前を付けておいた方が良いでしょう。
7. トリガーライブラリー：  
追加されるトリガーのライブラリーを指定します。「\*FILE」は、物理ファイルが指定されたライブラリーが指定される。
8. 反復変更許可：トリガー・プログラムの実行時に呼び出された1つのトリガーまたは後続のトリガー（複数可）内でのレコードに対する反復変更を可能にするかどうかを指定します。デフォルトは\*NOです。

## 2. Db2 for i トリガー機能の設定方法 (外部トリガーの設定)

### トリガーの有効化・無効化 (CHGPFTRG)

- 対象となるファイルに設定済みのトリガーの変更 (有効化・無効化) するのに、「CHGPFTRG」コマンドを使用します。
- このコマンドは、一時的にトリガー機能を停止したい場合に使われます。バッチ処理などの大量データを扱い、トリガー機能が不要な場合などに一時的に無効化することでパフォーマンスを向上することができます。

```

PF トリガーの変更 (CHGPFTRG)

選択項目を入力して、実行キーを押してください。

ファイル . . . . . > TOKMSP          名前
ライブラリ . . . . . > SAWADALIB  名前 , *LIBL, *CURLIB
トリガー . . . . . > TESTTRG

-----
トリガー・ライブラリ . . . . . *ALL          名前 , *ALL, *CURLIB
トリガー状態 . . . . . *DISABLED      *SAME, *ENABLED, *DISABLED
機密保護のトリガー . . . . . *SAME       *SAME, *YES, *NO
ミラー . . . . . *SAME                *SAME, *YES, *NO
  
```

## 2. Db2 for i トリガー機能の設定方法 (外部トリガーの設定)

### トリガーの除去 (RMVPFTRG)

- 対象となるファイルに設定済みのトリガーを除去するのに、(RMVPFTRG) コマンドを使用します。
- 除去するトリガーは、トリガー時間、トリガー名、トリガーイベントの指定で個別に除去できますが、項目に\*ALLを指定することで一括除去も可能

#### 物理ファイル・トリガーの除去 (RMVPFTRG)

選択項目を入力して、実行キーを押してください。

物理ファイル	> TOKMSP	名前	
ライブラリ	> SAWADALIB	名前	*LIBL, *CURLIB
トリガー時間	*ALL	*ALL, *BEFORE, *AFTER	
トリガー・イベント	*ALL	*ALL, *INSERT, *DELETE...	
トリガー	> TESTTRG		
<hr/>			
トリガー・ライブラリ	*FILE	名前	*FILE, *CURLIB

## 2. Db2 for i トリガー機能の設定方法 (外部トリガーの設定)

### トリガー定義の確認 (DSPFD)

- 物理ファイルに対するトリガーの状況や設定の確認は (DSPFD) コマンドで可能です。  
(例) DSPFD FILE(SAWADALIB/TOKMSP) TYPE(\*TRG)

```

DSPFD コマンドの入力
ファイル . . . . . : FILE      TOKMSP
ライブラリー . . . . . :          SAWADALIB
情報のタイプ . . . . . : TYPE      *TRG
ファイル属性 . . . . . : FILEATR   *ALL
システム . . . . . : SYSTEM    *LCL
ファイル記述の見出し
ファイル . . . . . : FILE      TOKMSP
ライブラリー . . . . . :          SAWADALIB
ファイルのタイプ . . . . . :          物理
ファイル・タイプ . . . . . : FILETYPE  *DATA
補助記憶域プール ID. . . . . :          00001
トリガー記述
トリガー名 . . . . . : TRG      TESTTRG
トリガー・ライブラリー . . . . . :          SAWADALIB
トリガーの状態 . . . . . : STATE    *ENABLED
トリガー状況 . . . . . :          *OPERATIVE
トリガー・イベント . . . . . : TRGEVENT *INSERT
トリガー時刻 . . . . . : TRGTIME  *BEFORE
反復変更の許可 . . . . . : ALWREPC  *NO
プログラム名 . . . . . : PGM      AAAAPGM
ライブラリー . . . . . :          SAWADALIB
プログラムはスレッド・セーフ . . . . . : THDSAFE  *UNKNOWN
マルチスレッド・ジョブの処置 . . . . . : MLTTHDAG *SYSVAL
トリガー・タイプ . . . . . :          *SYS
トリガーの方向 . . . . . :          *ROW
トリガー作成日/時刻 . . . . . :          22/07/01 16:29
トリガー更新桁の数 . . . . . :          0
  
```

## 2. Db2 for i トリガー機能の設定方法 (外部トリガーの設定)

### トリガー定義の印刷 ( PRTRGPGM )

- トリガー・プログラムの印刷 (PRTRGPGM) コマンドを使用して、特定のライブラリーや全ライブラリーの中のすべてのトリガー・プログラムのリストを印刷できます。以下は、報告書の例です。

(例) PRTRGPGM LIB(SAWADALIB)

スプールを確認すると、下記のように表示されます

```

                                     トリガー・プログラム (全報告書)
5770SS1 V7R4M0 190621                                DEM000  22/07/01 16:54:05
指定したライブラリー . . . . . :   SAWADALIB

----- トリガー -----
ライブラリー   ファイル   ASP   装置   名前           タイプ ライブラリー   プログラム   時刻   事象   条件           反復
SAWADALIB     TOKMSP   *SYSBAS  TESTTRG       *SYS SAWADALIB   AAAAPGM     前       挿入           NO
***  切り捨てられた行の全テキスト  ***
( リストするオブジェクトがない )

```



## 3. Db2 for i トリガー機能の設定方法 (SQLトリガーの設定)

### SQLトリガーの設定1

- SQL トリガーは、CREATE TRIGGER SQL ステートメントを指定するか、オペレーション・ナビゲーターを使用するかのどちらかの方法で作成できます。SQL ルーチン本体は、データベース・マネージャーによってプログラムに変換される、トリガーの実行可能な部分です。
- SQL ルーチンまたはトリガーを作成する際に、SQL は、組み込み SQL ステートメントを含む C ソース・コードを入れる一時ソース・ファイル (QTEMP/QSQLSRC) を作成します。
- SQL トリガーは、SQL CI 作成 (CRTSQLCI) およびプログラム作成 (CRTPGM) コマンドを使用して、プログラム (\*PGM) オブジェクトとして作成されます。プログラムは、トリガー名の暗黙または明示修飾子であるライブラリー内に作成されます。
- SQL ルーチン本体は、単一の SQL ステートメントで、SQL 制御ステートメントもこれに含まれます。プログラムが作成されると、制御ステートメント以外の SQL ステートメントは、プログラム内の組み込み SQL ステートメントになります。

## 3. Db2 for i トリガー機能の設定方法 (SQLトリガーの設定)

### SQLトリガーの設定2

CREATE TRIGGER name-of-trigger	・ ・ 1
ON TABLE_NAME	・ ・ 2
FOR EACH ROW / FOR EACH STATEMENT	・ ・ 3
MODE DB2ROW / DB2SQL	・ ・ 4
WHEN (Condition)	・ ・ 5
BEGIN	・ ・ 6
Routine body of the trigger	・ ・ 7
END	・ ・ 8

#### (解説)

1. トリガーは、CREATE TRIGGER とその名前で開始します。名前は、既存のトリガーの名前に一致してはなりません。名前は、“SQL”で始めることはできません。
2. TABLE\_NAME は、このトリガーを定義する表の名前を指定します。
3. FOR EACH は、データベース・マネージャーがトリガー・アクションを各行操作ごとに実行するのか、各ステートメントにつき 1 回 (アプリケーションによって表の行が更新されなくても) 実行するのかを指定します。
4. MODE DB2ROW は、それぞれの行操作の後でトリガーを実行することを指定します。MODE DB2SQL は、すべての行操作の後でトリガーを実行します。
5. 真または偽として評価される条件を指定します。これはオプションです。
6. トリガーの本体は、文節 BEGIN で開始します。
7. トリガーのルーチン本体では、トリガーが実行するアクションを定義します。
8. トリガーの本体は、END 文節で終了します。

## 3. Db2 for i トリガー機能の設定方法 (SQLトリガーの設定)

### SQLトリガーの設定3

- トリガー・プログラムのルーチン本体は、単一の SQL ステートメント (SELECT、UPDATE、INSERT、DELETE) または SQL 制御ステートメントで構成できます。
- SQL 制御ステートメントには、割り当てステートメント、ローカル変数の宣言、対話式ステートメントを組み込むことができます。SQL 制御ステートメントは、ほとんどのプロシージャー型言語に見られる基本的なプログラミング構造です。
- トリガー本体は、1つ以上の SQL ステートメントで構成されます。一般に、本体には任意の SQL ステートメントを使用できます。1つの制限は、BEFORE トリガーでは UPDATE や CREATE などのデータ変更ステートメントを実行できないことです。この制限以外は、SQLプロシージャーで使用されるのと同じステートメントのほとんどのものとユーザー定義関数を、SQL トリガーでも使用可能です。

## 4. Db2 for i トリガー機能を使ってみよう (外部トリガーの例)

まず外部トリガーを使ってみよう。仕様は下記になります。

- 社員マスター (SHAINP) を更新 (UPDATE) すると、社員マスターログファイル (SHAINPR) に出力。
- 社員マスターに対して、トリガー設定 (TRG\_TEST) し、トリガープログラムは、TRG\_PGMとして作成

社員マスター (SHAINP) は、下記レイアウトで作成

```

      A          R SHAINPR                UNIQUE
      A          SYSYCD          5S      COLHDG('社員' '番号')
      A          SYSNAM          200     COLHDG('社員名' '漢字')
      A          SYSSNAK         22A     COLHDG('社員名' 'カナ')
      A          SYSEXC          01A     COLHDG('性別')
      A          SYSXPS          02A     COLHDG('役職コード')
      A*                                     )
      A          K SYSYCD
  
```

社員マスターログファイル (SHAINP) は、下記レイアウトで作成

```

      A          R SHAINPR                UNIQUE
      A          SYSYCD          5S      COLHDG('社員' '番号')
      A          SYSNAM          200     COLHDG('社員名' '漢字')
      A          SYSSNAK         22A     COLHDG('社員名' 'カナ')
      A          SYSEXC          01A     COLHDG('性別')
      A          SYSXPS          02A     COLHDG('役職コード')
      A*                                     )
      A          SYLCDT          8S      COLHDG('出力日')
      A          SYLGTM          6S      COLHDG('出力時間')
      A          SYWSID          10A     COLHDG('端末ID')
      A          SYUSER          10A     COLHDG('ユーザー')
      A          SYEVEN          1A      COLHDG('トリガー事象')
      A          K SYSYCD
  
```

# 4. Db2 for i トリガー機能を使ってみよう (外部トリガーの例)

## トリガープログラム (TRG PGM) は下記 (ソース1/2)

```

H*****
H*TRG_PGM トリガープログラム サンプル
H*****
H DATEDIT(*YMD)
H*****
F*社員マスターログファイル
FSHAINPR 0 E K DISK
F*
D* I/F FILE (OLD)
DOLDREC E DS EXTNAME(SHAINP) PREFIX(0)
D BASED(OLDRP)
D* I/F FILE (NEW)
DNEWREC E DS EXTNAME(SHAINP)
D BASED(NEWRP)
D*****
D** 構造体
D*****
D*プログラム状況データ構造
D SDS
DD@PGID *PROC
DWKWSID 244 253
DWKUSRP 254 263
DWKNOJB 264 269
D*トリガー用受け取りパラメータ 1
DPARM1 DS
D*物理ファイル名
DFNAME 10
D*物理ファイルライブラリー名
DLNAME 10
D*物理ファイルメンバー名
DMNAME 10
D*トリガー事象

```

```

DTEVEN 1
D*トリガー時刻
DTTIME 1
D*コミットロックレベル
DCMTLCK 1
D*予約済み 1
DFILL1 3
D*ccsid
DCCSID 10I 0
D*相対レコードNO
DRECNO 10I 0
D*予約済み2
DFILL2 10I 0
D*元のレコードのオフセット
DOLDOFF 10I 0
D*元のレコードの長さ
DOLDLEN 10I 0
D*元のレコードの空バイトマップのオフセット
DONOFF 10I 0
D*空バイトマップの長さ
DONLEN 10I 0
D*新しいレコードのオフセット
DNEWOFF 10I 0
D*新しいレコードの長さ
DNEWLEN 10I 0
D*新しいレコードの空バイトマップのオフセット
DNNOFF 10I 0
D*トリガー用受け取りパラメータ 2
DPARM2 S 10I 0
DOLDMAP DS BASED(OLDNP)
DONFLD 1 DIM(4)
DNEWMAP DS BASED(NEWNP)
DNNFLD 1 DIM(4)

```

# 4. Db2 for i トリガー機能を使ってみよう (外部トリガーの例)

(続き：ソース 2/2)

```

C*****
C*INITIAL SET ROUTINE (パラメータリスト)
C*****
C   *ENTRY      PLIST
C   PARM1      PARM          PARM1
C   PARM2      PARM          PARM2
C*****
C** MAIN ROUTINE
C*****
C           EVAL      NEWRP = %ADDR(PARM1) + NEWOFF
C           EVAL      OLDRP = %ADDR(PARM1) + OLDOFF
C*
C           Z-ADD     *DATE          SYLCDT
C           TIME      SYLGTM
C           MOVE(L(P) WKWSID        SYWSID
C           MOVE(L(P) WKUSRP        SYUSER
C           MOVE(L(P) TEVEN         SYEVEN
C
C*書き出し
C*
C           WRITE     SHAINRR
C*終了処理
C           EVAL      *INLR = *ON
C           RETURN

```

## 4. Db2 for i トリガー機能を使ってみよう (外部トリガーの例)

### トリガープログラム (TRG PGM) の解説1/2

```
FSHAINPR O E K DISK
```

・・・F仕様書：社員マスターログファイル (SHAINPR)を出力 (O) で定義する

```
D* I/F FILE (OLD)
```

```
DOLDREC E DS EXTNAME(SHAINP) PREFIX(O)
```

```
D BASED(OLDRP)
```

```
D* I/F FILE (NEW)
```

```
DNEWREC E DS EXTNAME(SHAINP)
```

```
D BASED(NEWRP)
```

・・・構造体「OLDREC」ではEXTNAME (SHAINP)を指定して、OLDRECの構造体レイアウトを社員マスターの内容同期している。  
NEWRECも同様。BASEDで基底ポインター名をOLDRPを指定。

```
C EVAL NEWRP = %ADDR(PARM1) + NEWOFF
```

```
C EVAL OLDRP = %ADDR(PARM1) + OLDOFF
```

・・・トリガーバッファ内 (PARM1) のレコード位置のポインターをNEWRPにセットして、構造体NEWRECに変更後の情報がセットされる。変更前情報のポインターはOLDRPにセットされる。変更前のデータも取得することができます。

```
C Z-ADD *DATE SYLCDT
```

```
C TIME SYLGTM
```

```
C MOVEL(P) WKWSID SYWSID
```

```
C MOVEL(P) WKUSRP SYUSER
```

```
C MOVEL(P) TEVEN SYEVEN
```

・・・プログラム状況データ構造から、端末IDや、ユーザー名、トリガー事象を取得している。

# 4. Db2 for i トリガー機能を使ってみよう (外部トリガーの例)

## トリガープログラム (TRG PGM) の解説 2/2

### D仕様書の構造体について

・・・その他のD仕様書のトリガーバッファからの受け取りパラメータについては、下記を参照してください。

詳細は、下記のIBM i 7.4データベースプログラミングのp309 「トリガーバッファセクション」を参照してください。

[https://www.ibm.com/docs/ja/ssw\\_ibm\\_i\\_74/dbp/rbafopdf.pdf](https://www.ibm.com/docs/ja/ssw_ibm_i_74/dbp/rbafopdf.pdf)

次の表には、トリガー・バッファ内のフィールドの要約が示されています。

オフセット	10 進数	オフセット	16 進数	タイプ	フィールド
0		0		CHAR(10)	物理ファイル名
10		A		CHAR(10)	物理ファイルのライブラリー名
20		14		CHAR(10)	物理ファイル・メンバー名
30		1E		CHAR(1)	トリガー事象
31		1F		CHAR(1)	トリガー時間
32		20		CHAR(1)	コミット・ロック・レベル
33		21		CHAR(3)	予約済み
36		24		BINARY(4)	データの CCSID
40		28		BIN(4)	相対レコード番号
44		2C		CHAR(4)	予約済み
48		30		BINARY(4)	元のレコードのオフセット
52		34		BINARY(4)	元のレコードの長さ
56		38		BINARY(4)	元のレコードの NULL バイト・マップのオフセット
60		3C		BINARY(4)	元のレコードの NULL バイト・マップの長さ
64		40		BINARY(4)	新しいレコードのオフセット
68		44		BINARY(4)	新しいレコードの長さ
72		48		BINARY(4)	新しいレコードの NULL バイト・マップのオフセット
76		4C		BINARY(4)	新しいレコードの NULL バイト・マップの長さ
80		50		CHAR(*)	予約済み
*		*		CHAR(*)	元のレコード
*		*		CHAR(*)	元のレコードの NULL バイト・マップ



## 4. Db2 for i トリガー機能を使ってみよう (外部トリガーの例)

### 物理ファイル (SHAINP) へのトリガープログラムの登録

- 社員マスター (SHAINP)を更新 (UPDATE)時に、トリガープログラム(TRG\_PGM)を起動
- このトリガープログラム (TRG\_PGM)は、社員マスターログに更新時の情報を出力します。

```

物理ファイル・トリガーの追加 (ADDPFTRG)

選択項目を入力して、実行キーを押してください。

物理ファイル . . . . . SHAINP          名前
ライブラリー . . . . . SAWADALIB       名前 , *LIBL, *CURLIB
トリガー時間 . . . . . *AFTER         *BEFORE, *AFTER
トリガー・イベント . . . . . *UPDATE    *INSERT, *DELETE, *UPDATE...
プログラム . . . . . TRG_PGM          名前
ライブラリー . . . . . SAWADALIB       名前 , *LIBL, *CURLIB
トリガーの置き換え . . . . . *NO        *NO, *YES
トリガー . . . . . TRG_TEST_

-----
トリガー・ライブラリー . . . . . *FILE      名前 , *FILE, *CURLIB
反復変更許可 . . . . . *NO          *NO, *YES
  
```

## 4. Db2 for i トリガー機能を使ってみよう (外部トリガーの例)

### トリガー設定とプログラムの確認

DFUで社員マスター (SHAINP)のデータを1件更新します。  
(役職コードを「03」から「04」へ変更。)

```
社員番号 :      1
社員 名漢字 : 澤田英寿
社員 名カナ : サワダヒデシ
性別       :      M
役職コード :    03
```



```
社員番号 :      1
社員 名漢字 : 澤田英寿
社員 名カナ : サワダヒデシ
性別       :      M
役職コード :    04
```



社員マスターログファイル (SHAINPR) に自動でログファイルが1件追加されました。更新時間、端末IDなどの情報が出力されています。

```
社員番号 :      1
社員 名漢字 : 澤田英寿
社員 名カナ : サワダヒデシ
性別       :      M
役職コード :    04
出力日     :    20220703
出力時間   :    184250
端末 ID    :    QPADEV0001
ユーザー   :    SAWADA
トリガー事象 :    3
```

トリガー事象：トリガー・プログラムが呼び出される原因となった事象。  
値は以下のとおりです。

- '1' 挿入操作
- '2' 削除操作
- '3' 更新操作
- '4' 読み取り操作

## 5. Db2 for i トリガー機能を使ってみよう (SQLトリガーの例)

次にSQL トリガーの単純な例を作成してみましょう。

- この例のトリガーは NEW\_HIRE という名前で、その名前が示すように、新しい人を雇用するたびに従業員数を増分します。
- EMPLOYEE表に新しい行が挿入されるたびに、表 COMPANY\_STATS の列 NBEMP の値を 1 つ増やします。

### SQLトリガーのサンプル

```
CREATE TRIGGER NEW_HIRE          . . . 1
AFTER INSERT ON EMPLOYEE       . . . 2
FOR EACH ROW MODE DB2SQL      . . . 3
BEGIN                          . . . 4
UPDATE COMPANY_STATS SET NBEMP = NBEMP + 1; . . . 5
END                             . . . 6
```

(解説)

- 1.トリガーの名前を定義します。この例の場合、名前は NEW\_HIRE です。
- 2.トリガー・イベントを定義します。この例の場合、イベントは INSERT で、起動時は AFTER です。EMPLOYEE表が対象
- 3.データベース・マネージャーがトリガー・アクションを実行する方法を定義します。  
MODE DB2 SQLは、全ての行操作の後で起動。AFTERトリガーのみ有効。
- 4.すべてのSQLトリガーは、トリガー本体の先頭に BEGIN 文節がなければなりません。
- 5.これは、実行される SQL ステートメントです。COMPANY\_STATS表の従業員数 (NBEMP)を1つ増やします。
- 6.トリガー本体の末尾に END 文節を置く必要があります。

## 5. Db2 for i トリガー機能を使ってみよう (SQLトリガーの例)

SQLトリガーの確認をしましょう。

- 前頁のコマンドを実行すると、下記が作成されます。

OPT	オブジェクト	タイプ	属性	テキスト
__	NEW_HIRE	*PGM	GLE	SQL_TRIGGER NEW_HIRE

DFUで1件データをEMPLOYEE表に追加します。



COMPANY\_STATS表の列NBEMPに+1されます。

```

様式 . . . . : EMPLOYEE
社員番号 : 022222
名前 : BENJAMIN
ミドルネーム : H
ラストネーム : HIDETOSHI
部門番号 : A00
電話番号 : 8941
入社日付 : 1988-04-01
JOB: PRES
レベル : 18
性別 : F
誕生日 : 1965-08-28
給与 : 5000000
ボーナス : 10000000
COMM: 1000000
  
```

```

*RECNR: 1
NBEMP: 43
  
```

## 5. Db2 for i トリガー機能を使ってみよう (SQLトリガーの例2)

もう少し複雑なSQL トリガーの単純な例を作成してみましょう。

- ここでは、WHEN 条件を使用した少し複雑な例を示します。この例のトリガーは SALARY\_TRACK という名前で、その名前が示すように、社員の給与をトラックします。トリガーは、従業員の給与が減額されるたびに、SALARY\_CTL 表にレコードを挿入します。SQL トリガーは、列 SALARY が更新される場合にのみ起動されます。外部トリガーには、特定の列が更新される場合にのみ起動するという機能はありません。(SQLトリガーのみの機能になります)

### SQLトリガーのサンプル

```
CREATE TRIGGER SALARY_TRACK . . . 1
AFTER UPDATE OF SALARY ON EMPLOYEE . . . 2
REFERENCING NEW ROW AS NROW . . . 3
        OLD ROW AS OROW
FOR EACH ROW MODE DB2SQL . . . 4
WHEN (NROW.SALARY < OROW.SALARY) . . . 5
BEGIN . . . 6
INSERT INTO SALARY_CTL (EMPNO, NEW_SALARY, OLD_SALARY, UPDATE_TIMESTAMP)
VALUES (NROW.EMPNO, NROW.SALARY, OROW.SALARY, CURRENT_TIMESTAMP);
        . . . 7
END . . . 8
```

## 5. Db2 for i トリガー機能を使ってみよう (SQLトリガーの例2)

前頁のSQLトリガーのサンプルの解説

- 1.トリガーの名前は SALARY\_TRACK です。
- 2.これは列 SALARY の更新後トリガーです。
- 3.REFERENCING NEW ROW AS NROW と REFERENCING OLD ROW AS OROW は、トリガー行の新しい変更値とオリジナル値を参照するのに使用される相関変数です。
- 4.モードを DB2SQL として定義し、トリガーが各変更行ごとに起動されます。
- 5.新しい給与が古い給与より少ない場合、トリガーが実行されます。
- 6.トリガー・プログラムの本体を開始します。
- 7.SALARY\_CTL 表にレコードを挿入します。
- 8.トリガー・プログラムの本体を終了します。

SQLトリガーは、比較的簡単です。いろんな場面で利用してみましよう。

### 3. 補足情報

- (1) Db2 トリガー マニュアル  
<https://www.ibm.com/docs/ja/i/7.5?topic=files-triggering-automatic-events-in-your-database>
- (2) IBMiトリガー機能を活かしたセキュリティログ対応  
[https://www.migaro.co.jp/contents/support/technical\\_report\\_search/no10/tech/10\\_01\\_04.pdf](https://www.migaro.co.jp/contents/support/technical_report_search/no10/tech/10_01_04.pdf)
- (3) SQL トリガーの作成  
<http://www63.tok2.com/home2/jd4/AddSQLTrigger.html>
- (4) SQL トリガーの起動タイミング (Each Row/Each Statement)  
<http://jd400.web.fc2.com/TriggerTiming.html>
- (5) SQL トリガーのモード (DB2SQL/DB2ROW)  
<http://jd400.web.fc2.com/TriggerMode.html>
- (6) READ トリガー/RPG でトリガー処理  
<http://www63.tok2.com/home2/jd4/ReadTrigger.html>
- (7) 【第25回】 RPGをストアドプロシージャとして再利用する  
[https://www.i-cafe.info/column/serials/new\\_decade\\_25](https://www.i-cafe.info/column/serials/new_decade_25)

# IBM Power10 Advantage



AIX

IBM i

Linux

2022年7月20日(水)

午前10:00～11:30開催 (Webセミナー形式)

常に変化し持続可能性が求められる時代において、デジタル変革を加速し、新たなビジネス価値を生み出す取り組みはもはや必要不可欠です。そうした中、その実現を支えるIT基盤が果たす役割も再定義され、可能性も広がり、選択肢も急速に増えています。当イベントでは、変化を前提としたIT基盤に求められる要件とテクノロジーで革新的な変革を目指す取り組みをお客様よりご紹介いただきます。ぜひお申込みいただけますようお願い申し上げます。

## 開催概要

【開催日時】 2022年7月20日 (水曜日) 10:00 – 11:30

【対象】 AIX・IBM i・Linux・IBM Powerをご利用中の企業、ミッション・クリティカルな業務を稼働中のお客様

【参加費用】 無料/事前登録制

## 【プログラム】

- ご挨拶
- Power10 製品概要
- AIX ユーザー様ご講演 (住信SBIネット銀行株式会社様)
- IBM i ユーザー様ご講演 (カシオ計算機株式会社様)
- Red Hat on Power ユーザー様ご講演 (キヤノン株式会社様)
- 摩擦レスなハイブリッド・クラウド (デモ)
- RISE with SAP、IBM Cloud における価値
- Oracle SE2 に最適な IBM Power の価値

【お申込み】 [https://ibm.biz/power10advantage\\_bpjbcc](https://ibm.biz/power10advantage_bpjbcc)

【イベントに関するお問い合わせ】 <https://ibm.biz/askpower10japan>



# IBM Power Salon ご案内



**IBM Powerユーザーのための自由な語り場。**  
**オンラインサロンで、お客様同士、IBMスペシャリストと繋がりませんか**  
**是非、お気軽にご参加ください。**

内容：お客様によるDX事例、クラウド活用事例  
 IBM i/AIX/Linuxの技術情報、サポート情報  
 IBMスペシャリストによるQ&A 他

主催：日本アイ・ビー・エム（株）IBM Power 事業部

参加方法：オンライン開催  
 どなたでも参加可能、無料  
 事前申し込み不要

ご参加URL：[ibm.biz/Powersalon-webex](https://ibm.biz/Powersalon-webex)

お問い合わせ：[NO1POWER@jp.ibm.com](mailto:NO1POWER@jp.ibm.com)

第7回	5月11日(水) 9:00-10:00	IBM i エム・シーシー食品株式会社 石川様
第8回	6月8日(水) 9:00-10:00	IBM i World 2022 Q&A
第9回	7月13日(水) 9:00-10:00	IBM i 日本サニパック様 ご講演
-	8月 休会	
第10回	9月14日(水) 9:00-10:00	お客様講演予定
第11回	10月8日(水) 9:00-10:00	
第12回	11月9日(水) 9:00-10:00	
第13回	12月14日(水) 9:00-10:00	



## IBM Community Japan

## IBM i Club ご案内

## ■ IBM i Clubとは

- 自社システムとしてIBM iをご利用いただいている皆様同士で、各社の工夫や事例を紹介し合ったり、ディスカッションをしていただく場です。
- IBMの技術者も参加し、IBM iの最新情報のご提供や、ディスカッションに入らせていただくこともあります。
- 年5回(予定)、それぞれテーマを変えて開催します。

## ■ 2021年のテーマ例 (ご参加の皆様からいただいた課題をもとにテーマを選出しました)

- IBM i ユーザーハイブリッドクラウドへの道
- IBM i 人材育成の勘所
- 新技術の活用
- IBM i ユーザーのDX
- IBM i の優位性と今後

## ■ 開催期間

- 2022年2月24日から2022年12月末(予定)

## ■ 開催内容

- オンライン(Webex)セッションの開催: 2022年2月24日(木)を第1回とし年間5回(予定)
- 情報共有・ディスカッションの場: コミュニケーションツール(Slack)をご利用いただき、セッション以外の時も情報交換など可能です。

## ■ ご参加にあたってのお願い・ご注意点

- 守秘義務をお守りください。
- 営業活動を目的としたご参加はお断りいたします。
- IBM i Clubお申し込みには、事前にIBM Community Japanのメンバー登録が必要です。
- 開催期間の途中からのご参加も可能です。



## ■ 2022年開催予定

\*日時・内容が変更になる可能性があります

	日時	実施内容	
1	2/24 (終了)	最新情報	「数字で見るIBM i小辞典」IBM 佐々木
		事例紹介	「利用部門からの要望」にどう対応しているか? トクラス(株)様
		ディスカッション	テーマ: これからのIT部門の役割
2	5月	最新情報	「IBM i 新リリース発表」
		事例紹介	(調整中)
		ディスカッション	テーマ: 人材確保・人材育成
3	7月	最新情報	
		事例紹介	
		ディスカッション	テーマ: 新技術の活用(仮)
4	9月	最新情報	
		事例紹介	
		ディスカッション	テーマ: DX(仮)
5	11月	最新情報	
		事例紹介	
		ディスカッション	

## ■ コース詳細・お申込み

<https://www.ibm.com/ibm/jp/ja/ibmcommunityjapan-product-community.html>

## ■ ご参考 昨年 (2021年) 開催内容

[https://higherlogicdownload.s3.amazonaws.com/IMWUC/2fde9da6-6e7d-43b4-bae3-7f25168bbbd0/UploadedImages/japan/2022/2021\\_IBM\\_i\\_Club.pdf](https://higherlogicdownload.s3.amazonaws.com/IMWUC/2fde9da6-6e7d-43b4-bae3-7f25168bbbd0/UploadedImages/japan/2022/2021_IBM_i_Club.pdf)

## IBM i 関連情報

IBM i ポータル・サイト

<https://ibm.biz/ibmijapan>

月イチIBM Power情報セミナー「IBM Power Salon」

<https://ibm.biz/power-salon>

IBM i 関連セミナー・イベント

<https://ibm.biz/powerevents-j>

IBM i Club (日本のIBM i ユーザー様のコミュニティー)

<https://ibm.biz/ibmiclubjapan>

i Magazine (IBM i 専門誌。春夏秋冬の年4回発刊)

<https://www.imagazine.co.jp/IBMi/>

IBM i 研修サービス (i-ラーニング社提供)

<https://www.i-learning.jp/service/it/iseriess.html>

IBM Power Systems Virtual Server 情報

<https://ibm.biz/pvsjapan>

IBM i 7.5 技術資料

<https://www.ibm.com/docs/ja/i/7.5>

IBM Power ソフトウェアのダウンロードサイト (ESS)

<https://ibm.biz/powerdownload>

Fix Central (HW・SWのFix情報提供)

<https://www.ibm.com/support/fixcentral/>

IBM My Notifications (IBM IDの登録 [無償] が必要)  
「IBM i」「9009-41G」などPTF情報の必要な製品を  
選択して登録できます。

<https://www.ibm.com/support/mynotifications>

IBM i 各バージョンのライフサイクル

<https://www.ibm.com/support/pages/release-life-cycle>

IBM i 以外のSWのライフサイクル (個別検索)

<https://www.ibm.com/support/pages/lifecycle/>



ワークショップ、セッション、および資料は、IBMによって準備され、IBM独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる読者に対しても法律的またはその他の指導や助言を意図したのではなく、またそのような結果を生むものでもありません。本資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引き出すことを意図したもので、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでなく、またそのような結果を生むものでもありません。

本資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本資料に含まれている内容は、読者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したもので、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、Db2、Rational、Power、POWER8、POWER9、AIXは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。

他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。

現時点での IBM の商標リストについては、[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) をご覧ください。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Centrino、Intel Centrino ロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、およびPentium は Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは Microsoft Corporationの米国およびその他の国における商標です。

ITILはAXELOS Limitedの登録商標です。

UNIXはThe Open Groupの米国およびその他の国における登録商標です。

JavaおよびすべてのJava関連の商標およびロゴは Oracleやその関連会社の米国およびその他の国における商標または登録商標です。