

## 2：データの操作(DML)入門3

製作：清水健二

### データを集計して表示する

#### データ集計関数(SUM、AVG、MAX、MIN、COUNT)

データを集計して結果を表示する際には、SELECT文に加えてGROUP BY句で指定した列をキーにすることによって、集計を行います。GROUP BY句は、集計関数とセットで利用します。

#### 集計関数の例

集計関数名	SUM	AVG	MAX	MIN	COUNT
内容	合計	平均	最大	最小	データの個数

ここではworldデータベースを使用します。次のSQL文を実行してください。

```
USE world;
```

ex. 世界の総人口の合計を表示する

```
SELECT SUM(Population) FROM Country;
```

ex. 世界の大陸ごとの平均GNPを表示する

```
SELECT Continent, AVG(GNP) FROM Country GROUP BY Continent;
```

ex. 世界の地域ごとの最新(最大)の設立年を表示する

```
SELECT Region, MAX(IndepYear) FROM Country GROUP BY Region;
```

ex. 世界の地域ごとの最古(最小)の設立年を表示する

```
SELECT Region, MIN(IndepYear) FROM Country GROUP BY Region;
```

ただし、この方法では、最新や最古の設立年の数値はわかっても、その国の名前はわかりません。国名を知るには後述するサブクエリを使う必要があります。

ex. 世界の大陸ごとの国の数を表示する

```
SELECT Continent, COUNT(Name) FROM Country GROUP BY Continent;
```

### 表示するフィールド名を別名に変更する

#### 表示フィールド名の変更(AS)

集計関数を使ったフィールドを表示する場合、関数名がそのまま表示されます。そこでSELECT文で表示するフィールド名にAS 別名をつけることで、表示名を別名に変更できます。

```
SELECT 列名 AS 別名 [,列名2 AS 別名] FROM [データベース名].[テーブル名];
```

ex.世界の大陸ごとの平均人口を日本語名で表示する

```
SELECT Continent as 大陸,AVG(Population) as 平均人口 FROM Country GROUP BY 大陸;
```

## 集計した結果に条件を設定する

### 集計結果に条件を設定(HAVING句)

条件指定には通常WHERE句を使用しますが、GROUP BYを使った集計列を条件に指定して検索を行う時には、HAVING句で条件指定します。

ex.世界の大陸ごとの平均人口の中から2億人以上の大陸のみ表示する

```
1 SELECT Continent as 大陸,AVG(Population) as 平均人口 FROM Country GROUP BY 大陸
2 HAVING 平均人口>=20000000;
```

## テーブル名を別名に変更する

### フィールド名にテーブル名をつける

今まで省略していましたが、フィールド名に対してテーブル名での修飾が可能です。  
テーブル名と列名の間を.で繋げることで列名の区別をつけることができます。

```
SELECT テーブル名.列名 FROM テーブル名 WHERE テーブル名.列名=xx;
```

テーブル名.フィールド名という記述法は次項のテーブルの結合(JOIN)の際に多用されます。SQLでは、修飾を容易にする、テーブル名への別名（エイリアス）付けと呼ばれる方法が準備されています。  
'FROM テーブル名'の後に半角スペースをつけてテーブル別名を指定できます。

```
SELECT テーブル別名.列名 FROM テーブル名 テーブル別名 WHERE テーブル別名.フィールド名=xx;
```

ex.国データを別名(kuni)に変更して、国コードが「JPN」の国名と人口を表示する

```
SELECT kuni.Name,kuni.Population FROM Country kuni WHERE kuni.code='JPN';
```

## データベース game\_rankを確認する

ここでは、データベースファイルgame\_rankを使用します。  
PHPMyAdminや下記のSOURCE文を使ってgame\_rank.sqlをインポートします。

### SQLファイルをインポートする(SOURCE)

```
SOURCE path/to/file.sql
```

ただし、この方法だと日本語が文字化けすることが多いため、PHPMyAdmin等からのインポートを推奨します。

データベースgame\_rankのインポートを確認します。

```
SHOW DATABASES;
```

データベース一覧にgame\_rankが表示されていれば成功です。

次にテーブルの内容を確認します

```
SHOW TABLES;
```

テーブルの内容をSELECT文で一通り確認してください。

## 複数のテーブルを結合する

### 複数のテーブルの内部結合(INNER JOIN)

内部結合(INNER JOIN)結合する両方の表に存在するデータのみ取得します。テーブルを結合する際はテーブルに別名をつけるとどのテーブルのフィールドか区別しやすいです。

#### ex.ゲーム売上一覧を表示する

ここでは、game\_rankのmt\_softwareテーブル(A)とtt\_software\_salesテーブル(B)を結合してゲーム名と売上を表示します。

```
1 SELECT A.name,B.sales_num
2 FROM mt_software A
3 INNER JOIN tt_software_sales B
4 ON A.id= B.software_id;
```

#### ex.ゲームメーカーが持つメーカーブランドを一覧で表示する

ここでは、game\_rankのmt\_makerテーブル(A)とmt\_maker\_brandテーブル(B)を結合してメーカー名とその傘下のブランドを表示します。

```
1 SELECT A.name,B.name
2 FROM mt_maker A
3 INNER JOIN mt_maker_brand B
4 ON A.id = B.maker_id;
```

### 外部結合 (LEFT JOIN、RIGHT JOIN)

一方の表にあってもう一方の表にないデータも取得します。

#### ex.LEFT JOINで、ゲーム機を販売しているメーカーブランド名とゲーム機名を表示する

```
1 SELECT A.name,B.name
2 FROM mt_models A
3 LEFT JOIN mt_maker_brand B
4 ON A.maker_brand_id = B.id;
```

#### ex.RIGHT JOINで、ゲーム機を販売しているメーカーブランド名とゲーム機名を表示する

```
1 SELECT A.name,B.name
2 FROM mt_models A
3 RIGHT JOIN mt_maker_brand B
4 ON A.maker_brand_id = B.id;
```

LEFT JOINとRIGHT JOINで結果が違うことを確認してください。それぞれ、**右のテーブルのレコードをすべて表示するか、左のテーブルのレコードをすべて表示するか**という違いがあります。

対してINNER JOINは、左右それぞれ共通したレコードのみ表示します。

通常はINNER JOINで事足りますが、あえて、NULLを含んでいても全レコード表示したい場合にLEFT JOIN、RIGHT JOINを活用してください。

※テーブルの結合には他にOUTER JOINも存在します。左右どちらのテーブルでも一致しないレコードのみ表示します。