データを集計する

前回は検索結果のそれぞれの行に対して同じ処理を行う仕組みとして関数を学習しました。

今回は検索結果をひとまとめにして集計する方法を学びます。

# 集計関数（集約関数、グループ関数）とは

集約関数とは、1つの列グループに対して施すことのできる演算機能をいいます。**集約関数は列グループ全体につき1つの値を生成**します。例えば、商品表にある価格(PRICE)列に対して合計を求める集約関数を利用すると、全商品の合計金額を取得できます。

# 集計関数の特徴

集約関数は、前回学習したCHAR\_LENGTHやROUNDなどの普通の関数とは動作や結果の形に違いがあります。通常の関数は検索結果の各行に対して、処理を行うので検索結果の行が増えたり減ったりすることはありません。一方、**集約関数は集計の対象となった複数のデータに対して処理を行うため、結果は１つになります**。

・集約関数の動作イメージ

|  |  |
| --- | --- |
| PNAME | PRICE |
| マルゲリータ  デラックスピザ  シーフードピザ  ギガミート  チキン＆ベーコン  BBQチキン  … | 1200  1700  1500  2500  2200  2200  … |

|  |
| --- |
| SUM(PRICE) |
| 21830 |

・通常の関数の動作イメージ

|  |  |
| --- | --- |
| PNAME | PRICE |
| マルゲリータ  デラックスピザ  シーフードピザ  … | 1200  1700  1500  … |

|  |
| --- |
| CHAR\_LENGTH(PNAME) |
| 6  7  7  … |

# 代表的な集計関数

通常の関数と同様に、DBMSにより使用できる集約関数は異なります。

授業では、ほとんどの製品で利用可能な５つの集約関数を学習します。

|  |  |
| --- | --- |
| 関数名 | 説明 |
| SUM | 各行の値の合計を求める |
| MAX | 各行から最大の値を求める |
| MIN | 各行から最小の値を求める |
| AVG | 各行の値の平均値を求める |
| COUNT | 対象データの行数を求める |

ハンズオン　合計、最大、最小、平均を求める

検索結果の列に対して、合計などの集計結果を求めたい場合はそれぞれ、

SUM、MAX、MIN、AVG関数を利用することで算出することが出来ます。

1. 商品表で管理されている商品の中で価格の合計と最大、最小及び平均値を求める。

SELECT SUM(PRICE), MAX(PRICE), MIN(PRICE), AVG(PRICE)

FROM PRODUCT ;

ハンズオン　データの件数を取得する

検索結果の行数、つまりデータ件数を求めるにはCOUNT関数を使用します。

COUNT関数は引数の指定により、取得される行数に違いがありますので注意が必要です。

1. 材料表を産地の昇順に並び替えて表示する。

SELECT \* FROM MATERIAL ORDER BY ORIGIN ;

1. 材料表のデータ件数を取得する。

SELECT COUNT(\*), COUNT(ORIGIN), COUNT(DISTINCT ORIGIN)

FROM MATERIAL ;

・COUNT関数の違い

|  |  |
| --- | --- |
| 書式 | 説明 |
| COUNT( \* ) | NULL値を含んだ検索結果のデータ件数 |
| COUNT( 列名 ) | 指定した列のNULL以外のデータ件数 |
| COUNT( DISTINCT 列名 ) | 指定した列のNULL以外の重複しないデータ件数 |

# 集計に関する４つの注意点

①SELECT文でしか利用できない

集約関数は「検索結果」に対して集計を行うため、INSERT文やUPDATE、DELETE文

では使用することはできません。

集約関数は、SELECT句の列指定および集計後に並び替えを行うORDER BY句、

HAVING句で使用します。

※なぜ１行になるのにORDER BYで利用するのか？やHAVING句については

この後、学習するグループ化のときに説明します。

②結果表がデコボコになってはならない

集約関数を使用した列と通常の列を同時に表示することは出来ない。

※MySQLでは一応、動作するが不確定な値が表示されるので通常使用しない。

・商品名一覧とその合計値を表示するSELECT文　※不可能なSQL文

SELECT PNAME, SUM( PRICE ) FROM PRODUCT ;

|  |  |
| --- | --- |
| PNAME | SUM(PRICE) |
| マルゲリータ | 21830 |
| デラックスピザ |  |
| シーフードピザ |  |
| ギガミート |  |
| … |  |

このように、複数行取得される可能性のある通常の列と集計後１行になる集約関数の

列は混在して取得することは出来ない。

③引数に許される型が異なる

集約関数はいずれも1つの列を引数として受け取り処理を行います。

引数に指定できるデータ型は関数により異なります。

|  |  |  |  |
| --- | --- | --- | --- |
| 関数名 | 数値型 | 文字列型 | 日付型 |
| SUM | 数値の合計値 | × | × |
| MAX | 数値の最大値 | 文字コードの最大 | 最も新しい日付 |
| MIN | 数値の最小値 | 文字コードの最小 | 最も古い日付 |
| AVG | 数値の平均値 | × | × |
| COUNT | 指定列の行数 | 指定列の行数 | 指定列の行数 |

このように、合計や平均値を求める集約関数は数値が格納されている列以外を指定する

ことはできません。

④NULLの取り扱いについて

通常の関数および計算や比較などでは、NULLを含んでいると基本的には結果が不明の

ためNULLとなることがほとんどです。しかし、集約関数の場合はNULLを無視して

集計を行います。

|  |  |  |
| --- | --- | --- |
| 関数名 | NULLの扱い | 全行NULLの時 |
| SUM | 無視して集計 | NULL |
| MAX | 無視して集計 | NULL |
| MIN | 無視して集計 | NULL |
| AVG | 無視して集計 | NULL |
| COUNT(列名) | NULLを含まない行数 | 0 |
| COUNT( \* ) | NULLを含んだ行数 | NULLを含んだ行数 |

平均値などはNULL値を除いた集計結果が出力されます。NULLを対象に含んだ集計を

取りたい場合はCOALESCE関数を使用してNULLを0に置き換えて集計させることが出来ます。

SELECT AVG( COALESCE( PRICE, 0 ) ) FROM PRODUCT ;

※演習の環境ではPRICE列にNULLのデータがないため、結果に変化はありません。

|  |  |
| --- | --- |
| 関数 | 説明 |
| COALESCE( 値1, 値2 ) | リストの中でNULLでない値を出力する。引数に列名、初期値を設定することでNULL値の排除が出来る |

データをグループに分ける

# グループ別の集計

集約関数を用いると、検索結果をひとまとまりにして集計を行い１つの結果を得ることができます。つまり、検索条件のすべての行に対して処理が行われます。集約関数のみだと、商品情報の中からカテゴリー別の合計金額を求めることが出来ません。

|  |  |
| --- | --- |
| CATEGORY | SUM(PRICE) |
| ピザ  サイド  ドリンク  その他 | ????  ????  ????  ???? |

・商品表の価格の合計値

|  |
| --- |
| SUM(PRICE) |
| 21830 |

SELECT文は「結果表がデコボコになってはならない」ので、以下のようなSQL文では

正しい結果は出力できません。

SELECT CATEGORY , SUM( PRICE )

FROM PRODUCT ;

# グループ化(P184)

SQLには、指定した基準で検索結果を分ける**グループ化**と呼ばれる機能が備わっています。

今回の場合、**集計単位をカテゴリーに指定してグループ化**を行うことでカテゴリーごとの集計を行うことが出来ます。書式は以下の通りです。

SELECT グループ化の基準列名, 集約関数( 列名 )

FROM 表名

WHERE 検索条件

GROUP BY グループ化の基準列名 ;

※**集約関数の鉄則**

集約関数を使用する場合、SELECTで指定できる列はGROUP BYの指定列のみとなる。

集計を行い結果が１つまたは、グループごとにひとまとまりになるので、集計前の列は使用するとこができなくなる。

ハンズオン　グループごとに集計を行う

1. 商品表から商品情報をカテゴリーの昇順で取得する。

SELECT \* FROM PRODUCT

ORDER BY CATEGORY ;

1. カテゴリーごとの価格の最大値を取得する。

SELECT CATEGORY , MAX( PRICE ) FROM PRODUCT

GROUP BY CATEGORY

ORDER BY CATEGORY ;

# グループ集計後の絞り込み

グループ集計後に対して条件を付ける場合には、WHERE句では無くHAVING句を使用します。**WHERE句は集計を行う前に処理されるので、集計後の結果が分かりません**。

その為、集計後に条件を付けれるようにHAVING句が用意されています。

SELECT グループ化の基準列名, 集約関数( 列名 )

FROM 表名

WHERE 検索条件

GROUP BY グループ化の基準列名

HAVING 集計結果に対する検索条件

ハンズオン　集計後の結果に対して検索条件を入れる

１．商品表からカテゴリーごとの価格の最大値が500を上回るデータを取得する。

SELECT CATEGORY , MAX( PRICE )

FROM PRODUCT

WHERE CATEGORY != 'その他'

GROUP BY CATEGORY

HAVING MAX( PRICE ) > 500

ORDER BY MAX( PRICE) ;

# グループ集計の流れ

先程のハンズオンで実行したSQLは以下のように処理されていきます。

・商品表の全データ（一部表示）

|  |  |  |  |
| --- | --- | --- | --- |
| PRODUCT\_NO | PNAME | CATEGORY | PRICE |
| 0001  0002  …  3002  3003 | マルゲリータ  デラックスピザ  …  保温バック  エコバック | ピザ  ピザ  …  その他  その他 | 1200  1700  …  500  400 |

・WHERE句による絞り込み【WHERE CATEGORY != 'その他'】

|  |  |  |  |
| --- | --- | --- | --- |
| PRODUCT\_NO | PNAME | CATEGORY | PRICE |
| 0001  0002  …  2007  2008 | マルゲリータ  デラックスピザ  …  オレンジジュース  アップルジュース | ピザ  ピザ  …  ドリンク  ドリンク | 1200  1700  …  160  160 |

・GROUP BY句によるグループ分け及び集約関数実行【GROUP BY CATEGORY】

|  |  |
| --- | --- |
| CATEGORY | MAX( PRICE ) |
| ピザ  サイド  ドリンク | 3000  550  160 |

・HAVING句による集計結果に対して絞りこみ【HAVING MAX( PRICE ) > 500】

|  |  |
| --- | --- |
| CATEGORY | MAX( PRICE ) |
| ピザ  サイド | 3000  550 |

・ORDER BY句による並び替え【ORDER BY MAX( PRICE)】

|  |  |
| --- | --- |
| CATEGORY | MAX( PRICE ) |
| サイド  ピザ | 550  3000 |