さまざまな関数

# 関数とは

関数とは「**あるデータを受取り、定められた処理を実行して、その結果を返す命令**」のことです。データベースは、高度な処理を行いやすくするように関数が準備されています。

例えば、LENGTH関数は、ある文字列を引数に与えることでその文字列の長さ(文字数など)を求めて結果を返す機能を持っています。

関数はDBMS（データベース管理システム）ごとの違いが大きく互換性少ない分野です。

同じ関数名でも動作が違うものや、MySQLでしか使えない関数などもあります。

例：LENGTH関数はOracleでは文字列の文字数を取得するが、MySQLでは文字列の

バイト数を取得する関数となっている。

ハンズオン　関数の使い方

関数を呼び出すには、SQL文の中で次のような構文に従って記述します。

**関数名( 引数 )**

１．商品名の長さを取得する。

SELECT PNAME , **CHAR\_LENGTH( PNAME )**

FROM PRODUCT

WHERE CATEGORY = 'ピザ' ;

|  |  |
| --- | --- |
| PNAME | CHAR\_LENGTH(PNAME) |
| 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  バーベキューチキン | 6  7  7  5  8  9 |

※MySQLで、文字列の長さを取得するにはCHAR\_LENGTH関数を使用します。

# 関数が動作する流れ

関数を使いこなすためには、関数の呼び出しがどのように処理されるのかを理解する必要があります。関数の処理も式の評価と同じく、**各行ごとに繰り返し処理**を行います。そして**関数の処理が完了するとそのまま戻り値**になります。この特性を利用すると、関数の呼び出しを**入れ子(ネスト)**することが可能になります。この場合、**内側の関数から実行**されていきます。

※先ほどのハンズオンで実施したSQLでは、関数が各行ごとに実施されている。

CHAR\_LENGTH(マルゲリータ) → 6

CHAR\_LENGTH(デラックスピザ) → 7

CHAR\_LENGTH(シーフードピザ) → 7

※関数は処理が終われば戻り値になるので入れ子にすることが出来る。

CHAR\_LENGTH( TRIM( ' SQL ' ) ) // 内側のTRIM関数から実行される

→ CHAR\_LENGTH( 'SQL' ) // TRIM関数が戻り値の'SQL'になる

→ 3 // CHAR\_LENGTH関数が戻り値の3になる

関数のルール

・関数の処理は各行で行われる

・関数は処理が完了すると戻り値が設定される

・関数をネストしたら、内側の関数から処理されていく

# DUAL表について

DUAL表とは、端的にいうと中身が無いダミーのテーブルになります。

データベースでは、商品情報や店舗情報など情報を取得（検索）するときには、商品表などのテーブルを指定してSELECT文でデータの取得を行います。今回のように関数や計算式を使う場合、特定の表が不必要な場面が出てきます。DBMSによって仕様に違いがありますがSELECT文にはFROM句を指定する必要があるので、このような場合にダミーのDUAL表をFROM句に指定することで、無駄に特定のテーブルを指定する必要がなくなります。

なお、MySQLはFROM句を省略できるので以下の構文のどちらかで実行可能です。

SELECT 式や関数 FROM DUAL ; // DUAL表を指定する方法

SELECT 式や関数 ; // FROM句を省略する方法

文字列にまつわる関数

データベースで文字列に関する操作を行う関数のことを、**文字列関数**と呼びます。

以下に、MySQLで使用できる文字列関数と簡単な説明を記載します。

全ての文字列関数を調べるときは、公式リファレンスを参照ください。

https://dev.mysql.com/doc/refman/8.0/en/string-functions.html

・文字列関数一覧（一部）

|  |  |
| --- | --- |
| 関数名 | 説明 |
| LENGTH(文字列) | 文字列のバイトサイズを取得する |
| CHAR\_LENGTH(文字列) | 文字列の長さを取得する |
| CONCAT(文字列1 , 文字列2) | 引数に指定された文字列を連結する |
| INSTR(文字列 , 検索文字) | 文字列内で検索文字が最初に出る位置を取得する |
| LOWER(文字列) | 文字列に含まれる英字を小文字に変換する |
| UPPER(文字列) | 文字列に含まれる英字を大文字に変換する |
| TRIM(文字列) | 前後に含まれる半角空白を取り除く |
| REPLACE(文字列, 変更前, 変更後) | 文字列に含まれる変更前の文字を  変更後の文字に置換する |
| SUBSTR(文字列, 開始位置, 文字数) | 指定した開始位置とそこからの文字数分を  文字列から取得する |

ハンズオン　文字列関数を使用する

1. 英字を大文字、小文字に変換する

SELECT LOWER( 'ようこそNew Yorkへ' ) , UPPER( 'Texasバーガー' )

FROM DUAL ;

1. 文字列「 MySQL TEST 」の空白を除いた、文字数を取得する

SELECT CHAR\_LENGTH( TRIM( ' MySQL TEST ' ) ) ;

1. 商品表から商品名の一部や商品名の文字の入れ替えを行う

SELECT PNAME , SUBSTR(PNAME, 5, 3) , REPLACE(PNAME, 'ポテト', 'パンチ')

FROM PRODUCT WHERE PRODUCT\_NO = '1002' ;

数値にまつわる関数

データベースで数学の計算処理を行う関数のことを、**数値関数**と呼びます。

以下に、MySQLで使用できる数値関数と簡単な説明を記載します。

全ての数値関数を調べるときは、公式リファレンスを参照ください。

https://dev.mysql.com/doc/refman/8.0/en/mathematical-functions.html

・数値関数一覧(一部)

|  |  |
| --- | --- |
| 関数名 | 説明 |
| FORMAT(数値 , 小数部) | 数値を#,###.##形式の文字列に置換する  小数点以下は少数部で指定した桁数までになる |
| HEX(数値) | 引数の値を16進数の文字列で取得する |
| OCT(数値) | 引数の値を8進数の文字列で取得する |
| MOD(数値, 除数) | 除数で割った余りを求める |
| POW(数値, 指数) | 指数で累乗した結果を求める |
| RAND(シード値) | 0~1.0の範囲内でランダムな浮動小数点値を取得する  シード値で疑似乱数を指定できる |
| ROUND(数値, 小数部) | 数値の値を指定した小数部に四捨五入する |
| TRUNCATE(数値, 小数部) | 数値の値を指定した小数部に切捨て処理する |

ハンズオン　数値関数を使用する

1. 商品表の価格をフォーマットして表示させる

SELECT PNAME , FORMAT( PRICE , 0 )

FROM PRODUCT ;

1. 数値の四捨五入、切捨て、切上げ処理を行う

SELECT ROUND(1234.56, 0), ROUND(1234.56, 1), ROUND(1234.56, -1),

TRUNCATE(1234.56, 0), TRUNCATE(1234.56, 1), TRUNCATE(1234.56, -1)

FROM DUAL;

1. 数値の11を１６進数と８進数で表示する

SELECT HEX(11), OCT(11) FROM DUAL ;

日付にまつわる関数

データベースで時間の値を処理に使用できる関数のことを、**日付関数**と呼びます。

以下に、MySQLで使用できる数値関数と簡単な説明を記載します。

全ての日付関数を調べるときは、公式リファレンスを参照ください。

https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html

・日付関数一覧(一部)

|  |  |
| --- | --- |
| 関数名 | 説明 |
| NOW() | 現在の日付と時間を取得する |
| DATE(日付) | 引数の日付部分を抽出する |
| DATE\_ADD(日付, INTERVAL 数値 単位) | 日付値に指定した時間を加算する |
| DATE\_FORMAT(日付, フォーマット) | 日付をフォーマットに従って文字列にする |
| STR\_TO\_DATE(文字列, フォーマット) | 文字列をフォーマットに従って日付にする |
| DAYOFWEEK(日付) | 日付の曜日番号を求める  1:日曜日 ~ 7:土曜日 |
| DAYOFMONTH(日付) | ひと月(1~31)に対応する日にちを求める |
| DAYOFYEAR(日付) | 一年(1~366)に対応する日にちを求める |
| DATEDIFF(日付1, 日付2) | 2つの日数差を求める(日付1 – 日付２) |

・日付フォーマット一覧

|  |  |
| --- | --- |
| 指定子 | 説明 |
| %Y | 年４桁 |
| %y | 年２桁 |
| %m | 月(00～12) |
| %d | 日(00～31) |
| %c | 月(0～12) |
| %e | 日(0～31) |
| %H | 時(0～23) |
| %h | 時(1～12) |
| %i | 分(00～59) |
| %s | 秒(00～59) |

日付の算術について

MySQLでは、DATE\_ADD関数を使用しなくても、INTERVAL句を指定することで

日付の算術を行うことが出来ます。

DATE + INTERVAL 数値 単位

・単位一覧

|  |  |  |  |
| --- | --- | --- | --- |
| 単位 | | 単位 | |
| 年 | YEAR | 時 | HOUR |
| 月 | MONTH | 分 | MINUTE |
| 日 | DAY | 秒 | SECOND |
| 週 | WEEK |  |  |

ハンズオン　日付関数を使用する

1. 現在の時刻と日付を表示する

SELECT NOW(), DATE( NOW() )

FROM DUAL ;

1. 本日から10日後の日付を表示する

SELECT DATE\_ADD( NOW(), INTERVAL 10 DAY )

FROM DUAL ;

1. 従業員の誕生日をフォーマットして文字列で表示する

SELECT ENAME , DATE\_FORMAT( BIRTHDAY, '%Y年%m月%d日')

FROM EMPLOYEE ;

1. 文字列の内容を日付にフォーマットして表示する

SELECT STR\_TO\_DATE( '1999年07月31日' , '%Y年%m月%d日' )

FROM DUAL ;

1. 生まれた日から今日までの日数を表示する（誕生日と現在日付の日数差）

SELECT ENAME , DATEDIFF( NOW(), BIRTHDAY )

FROM EMPLOYEE ;