SQL命令の種類

# データベースを使う２つの立場

今までは、SELECT、INSERT、UPDATE、DELETEなどの命令を使って、既存のテーブルに対して**データを操作する方法**を学習してきました。データベース演習Iではコマンドプロンプトから直接SQL文を発行していますが、一般的な情報システムの内部ではJavaやPHPなどで開発しているプログラムにSQL文を生成させてDBMSにデータの操作を指示することが大半になります。つまり、今まで学習してきた内容は、情報システムなどで「**データの操作を指示する立場**」として役立つ部分になります。

しかし、SELECTやINSERTでデータの検索や登録を行うには、そもそもデータベースにテーブルが存在しなければなりません。そこで必要になるのが、テーブルの作成や各種設定など「**データベース自体の操作を指示する立場**」の存在です。

・データベースを利用する２つの立場

１．データベースにデータの取得や更新を指示する立場

２．必要なテーブルの準備やデータベースの各種設定を指示する立場

学習環境では、すでに表（テーブル）が用意された状態でしたが「データベース自体を操作する」命令を学習することで、テーブルや各種設定などを行うことが出来るようになります。

# 4種類の命令

ここでSQLの命令文の種類について整理したいと思います。

全てのSQL文は以下の４種類に分類することが出来ます。

|  |  |
| --- | --- |
| DML(Data Manipulation Language) | DDL(Data Definition Language) |
| データ操作言語：  データの格納や取出し、更新、削除などの命令（SELECT、INSERT、UPDATE、DELETEなど） | データ定義言語：  テーブルなどの作成や削除、各種設定などの命令（CREATE、ALTER、DROP、TRUNATEなど） |
| TCL(Transaction Control Language) | DCL(Data Control Language) |
| トランザクション制御言語：  トランザクションの開始や終了の命令  （COMMIT、ROLLBACKなど） | データ制御言語：  DMLやDDLの利用に関する許可や禁止を設定する命令（GRANT、REVOKEなど） |

これまでは、「データの操作を行う」のSELECTやUPDATE、COMMITやROLLBACKなどの命令文を学習してきました。これは表の左側にあるデータ操作言語(DML)、トランザクション制御言語(TCL)にあたります。

「データベース自体を操作する」には右側にあるデータ定義言語(DDL)、データ制御言語(DCL)と呼ばれる命令文で、今後は主にこちらの内容を学習していきます。

# DCLとは

データベースは、大事な情報を扱うため誰でも操作出来てはなりません。データベースを操作するためにはユーザを作成してログインを行い、そのユーザがデータベースで管理されているテーブルにアクセスをしています。さらにユーザを作成した時に合わせて、どのデータベースにアクセスしてよいかという許可を出す必要があります。

DCL(データ制御言語)とは、誰に、どのようなデータ操作やテーブル操作を許すかといった権限を設定するためのSQL命令の総称です。基本的にはデータベースの全権を管理する「**データベース管理者**」の立場の人が使う命令文になります。

データベース：studb

一般ユーザ：dbuser

DB管理者：root

MySQL(データベース管理システム)

表：PRODUCT

表：EMPLOYEE

データベースXXX

①

②

①データベース管理者のrootが一般ユーザにstudbにアクセスしてよい権限を与える。

②権限を与えられて初めてdbuserはstudbにアクセスすることが出来る。

・ユーザ作成の基本構文

CREATE USER ユーザ名 IDENTIFIED BY 'パスワード' ;

※WITH MYSQL\_NATIVE\_PASSWORDを追記する場合もあります。

・権限の付与と剥奪の基本構文

付与：GRANT 権限 ON データベース名.テーブル名 TO ユーザ名

剥奪：REVOKE 権限 ON データベース名.テーブル名 FROM ユーザ名

※権限の種類

https://dev.mysql.com/doc/refman/5.6/ja/grant.html#grant-privileges

ハンズオン　ユーザの作成と権限の付与

1. コマンドプロントを開いてrootユーザでログインを行う

mysql –u root –p でenterを押してパスワードrootでログインを行う。

1. testユーザをパスワードabcで作成する。

CREATE USER test IDENTIFIED BY 'abc' ;

1. もう一つコマンドプロンプトを開いてtestユーザでログインを行う

mysql –u test –p studb

1. rootでログインしたmysqlでtestユーザに店舗表のSELECT権限の付与を行う

GRANT SELECT ON studb.STORE TO test ;

1. もう一度、testユーザでログインを行う

mysql –u test –p studb

1. testユーザで店舗表のデータを表示する。

SELECT \* FROM STORE ;

1. testユーザで店舗表のデータを更新する。

UPDATE STORE SET STORE\_NO = '999' ;

※指定したデータベースに対して、全ての権限を付与する場合は以下のように記述します。

GRANT ALL ON データベース名.\* TO ユーザ名 ;

テーブルの作成

# テーブル作成の基本

テーブル（表）を作成するには、**CREATE TABLE**文を使います。

作成したいテーブルの名前とテーブルを構成する列、データ型を定義します。

・テーブル作成の基本構文

CREATE TABEL 表名 (

列名1 データ型 ,

列名２ データ型

) ;

# 列とデータ型

テーブルは列と行から成り立っています。それぞれの列には名前がついていますが、列に格納するデータの種類を**データ型**といい、テーブルを作成する時に定義する必要があります。

・MySQLの代表的なデータ型

|  |  |  |
| --- | --- | --- |
| データ型 | 種別 | 説明 |
| CHAR(n) | 文字列型 | n文字の固定長文字列 |
| VARCHAR(n) | 最大n文字の可変長文字列 |
| INT(m) | 数値型 | 4バイトの整数 ※mは表示桁数を表す |
| DOUBLE(m,d) | 有効桁数mで小数点以下d桁の倍精度浮動小数点 |
| DATE | 日付型 | 日付データ  サポート範囲は'1000-01-01' から '9999-12-31'まで |
| DATETIME | 日付時刻データ  サポート範囲は'1000-01-01 00:00:00.000000'から  '9999-12-31 23:59:59.999999'まで |
| TIME | 時刻データ |

・データ型について

１．テーブルの各列には、データ型を定義する必要がある。

２．列には、データ型で指定されている種類のデータしか格納できない。

３．利用可能なデータ型は、DBMS製品によって異なる。

# デフォルト値の指定

テーブルを定義する際に、列に対してデフォルト値を設定が可能です。デフォルト値を設定しておくと、INSERT文で具体的な値を指定なかった場合に、NULLではなく特定のデフォルト値(初期値)を格納することが出来ます。

・テーブル作成時のデフォルト値の設定

CREATE TABEL 表名 (

列名1 データ型 **DEFAULT 初期値** ,

列名２ データ型

) ;

ハンズオン　テーブルの作成　**※必須**

※dbuserでログインをして作業をして下さい。

１．職種番号と職種名を持つ、職種表の作成を行います。

CREATE TABLE JOB (

JOB\_NO CHAR(1),

JNAME VARCHAR(4) DEFAULT '未設定'

) ;

２．職種情報の登録を行います。

INSERT INTO JOB ( JOB\_NO, JNAME )

VALUES ( '1', '店長' ), ( '2' , '配達' ),( '3' , '受付' ),( '4' , '調理' );

３．名称がまだ決まっていない職種の登録を行います。

INSERT INTO JOB ( JOB\_NO ) VALUES ( '5' ) ;

４．職種表のデータを表示してください。

SELECT \* FROM JOB ;

５．トランザクションの確定をしてください。

COMMIT;

# ALTER TABLE文

テーブル定義の変更を行うには、**ALTER TABLE**文を使用します。変更にあたる命令なので具体的にテーブルの「何を」「どう」変えるのかを指定する必要があります。様々な変更をすることが出来ます。

・列定義の変更の一例

列の追加：ALTER TABLE テーブル名 ADD 列名 データ型 制約

列の削除：ALTER TABLE テーブル名 DROP 列名

列の変更：ALTER TABLE テーブル名 CHANGE 旧列名 新列名 データ型など

ALTER TABLE テーブル名 MODIFY 列名 データ型など

※列名を変えない場合はMODIFY、列名を変える場合はCHANGE

・テーブル名の変更

ALTER TABLE テーブル名 RENAME TO 新テーブル名

ハンズオン　テーブルの列定義を変更する **※必須**

１．職種表に備考列の追加を行う。

ALTER TABLE JOB ADD NOTE CHAR(50) ;

SELECT \* FROM JOB ;

２．職種表の職種番号を固定文字列から整数型に変更する。

ALTER TABLE JOB MODIFY JOB\_NO INT(1) ;

SELECT \* FROM JOB ;

SHOW COLUMNS FROM JOB ;

３．職種表の備考列の削除を行う。

ALTER TABLE JOB DROP NOTE ;

SELECT \* FROM JOB ;

# DROP TABLE文

過去に学習したDELETE文はデータの削除は行いますが、データの入れ物である表は削除されません。テーブル自体を削除する時には、**DROP TABLE**文を使用します。

・テーブル削除の構文

DROP TABLE テーブル名 ;

# DDL(データ定義言語)について

今回学習した、CREATE、ALTER、DROPなどはデータ定義言語のDDLに属する命令文です。DDL文は基本的に暗黙コミットを行うデータベースがほとんどですので、ロールバック(取消し)が出来ません。DDL文を扱う際には注意が必要です。