

データベースシステム

第5回

理工学部情報科学科

松澤 智史

SQL

- 関係データベースの操作, 定義を行う言語
- 1970年代にIBMが開発したSEQUEL(Structured English Query Language)が起源
- 関係データベースへのアクセスとして標準化

SQLの種類

- DDL(Data Definition Language)
 - データベースを定義するための言語
 - テーブル, ビューの定義等
- DCL(Data Control Language)
 - データベースを制御するための言語
 - トランザクションの開始, アクセス制御等
- DML(Data Manipulation Language)
 - データベースを操作するための言語
 - データの参照, 追加, 削除, 更新等

DDL(SQL Data Definition Language)

- CREATE
 - 作成
- DROP
 - 削除
- ALTER
 - 変更

テーブルの作成

```
CREATE TABLE テーブル名(  
    列名 データ型 [列の制約 ]  
    [,...n]  
    [テーブル全体の制約 ] )
```

- []内はオプション
- [...n]は上の文の繰り返し

主なデータ型

- CHAR(n) ..n文字の半角固定長文字列
- NCHAR(n) ..n文字の全角固定長文字列
- VARCHAR(n) ..最大n文字の半角可変長文字列
- NCAHR VARYING(n) ..最大n文字の全角可変長文字列
- SAMLLINT ..2バイト整数
- INTEGER ..4バイト整数
- DATE ..日付型
- BLOB ..バイナリデータ型

テーブルの制約

- 主キー制約
 - PRIMARY KEY
- 外部キー制約(参照制約)
 - REFERENCES テーブル名 (列名) [オプション]
- 非NULL制約
 - NOT NULL
- 一意性制約
 - UNIQUE
- 範囲制約
 - CHECK (範囲の条件)

外部キー制約時のオプション

- NO ACTION (デフォルト)
 - 参照制約に違反した場合はエラーを生成する
- CASCADE
 - 参照する列と参照される列の値を連動させる
 - ※参照される列の値が更新(削除)されると参照する列の値も更新(削除)
- SET NULL
 - 対応行がなくなった参照列の値にNULLを設定する
- SET DEFAULT
 - 対応行がなくなった参照列の値にあらかじめ設定しておいたデフォルト値を設定する

テーブル作成の例

1. 学生(学籍番号, 氏名, 学科コード, EMail)
2. 学科(学科コード, 学科名)

```
CREATE TABALE 学生 (  
    学籍番号 INTEGER PRIMARY KEY,  
    氏名 NCHAR(20) NOT NULL,  
    学科コード INTEGER REFERENCE 学科(学科コード),  
    Email VARCHAR(30) UNIQUE )
```

DCL(SQL Data Control Language)

- GRANT
 - アクセス制御
- トランザクション管理(後の回で詳述)

GRANTの構文

```
GRANT { ALL [PRIVILEGES] } | SELECT
      | INSERT [(列のリスト)] | DELETE
      | UPDATE [(列のリスト)]
      | REFERENCES [(列のリスト)] | USAGE
ON { [TABLE] 表名 | DOMAIN ドメイン名
    | COLLATION 照合順序名
    | CHARACTER SET 文字コード設定名
    | TRANSLATION 置換名 }
TO { ユーザ名 | PUBLIC } [ WITH GRANT OPTION]
```

GRANT 設定権限 ON 設定をするデータ(表) TO ユーザ

GRANTの豆知識

- DBMSによって多少設定できる項目が異なる
- MySQLは, GRANTでユーザやパスワード設定なども可能

DML(SQL Data Manipulation Language)

- SELECT
 - 一つまたは複数の表から行・列や導出した値を取り出す
- INSERT
 - 表やビューにデータを追加する
- UPDATE
 - 表やビューのデータを更新する
- DELETE
 - 表やビューのデータを削除する

SELECTの構文

SELECT [ALL | DISTINCT] 表示する列リスト

FROM 表名1 [, ..表名n]

[JOIN 結合条件]

[WHERE 選択条件]

[GROUP BY 集計のキーとなる列リスト]

[HAVING グループ化後の選択条件]

[ORDER BY 整列のキーとなる列リスト [ASC | DESC]]

SELECTの構文

- SELECT句
 - 取り出したい表の列名を指定し, “,”(カンマ)で区切る
 - すべての列を指定する場合は “*” (アスタリスク)で指定する
 - DISTINCTを指定すると, 同じ内容の列があった場合行を一つに纏める
- FROM句
 - 検索対象の表やビューを指定する
 - ASをつけると別名をつけることができる

SELECTの構文

- JOIN句
 - FROM 元の表名 [結合条件] JOIN 結合する表名
ON 元の表名.列名 θ 結合する表名.列名
 - θ は比較演算子(=, >, <, <=, >=, <>)
 - JOINの前に条件指定がない場合は内部結合
 - 自然結合(NATURAL JOIN)
 - 左外部結合([LEFT] [OUTER] JOIN)
 - 右外部結合(RIGHT [OUTER] JOIN)
 - 完全外部結合(FULL [OUTER] JOIN)
 - 直積(CROSS JOIN)
 - 両方の表に同じ名前の列が存在しており, その列を使用して等結合の
ときのみONの代わりにUSINGを用いることができる

SELECTの構文

- WHERE句
 - 取り出す情報の検索条件を指定する
 - 複数の条件を指定する場合は, AND, ORで接続する(ANDが優先)
 - 可能な条件
 - 論理演算子(評点>50等)
 - NULL値の検査(評点 IS NULL , 評点 IS NOT NULL)
 - LIKE検査(氏名 LIKE '鈴木%', 氏名 LIKE '鈴木_')
 - %は0文字以上の任意の文字列 _は任意の1文字
 - BETWEEN検査(評点 BETWEEN 50 AND 60)
 - EXISTS(NOT EXISTS)検査 (EXISTS (SELECT * FROM A))
Aに1行でもデータがあれば真
 - IN範囲検査 (住所 IN '千葉', '東京') 住所が千葉か東京に一致すれば真

SELECTの構文

- GROUP BY句
 - 集約関数(後の回で詳述)を使って問い合わせを行う
- HAVING句
 - GROUP BY句の結果に検索条件を追加する際に使用する
- ORDER BY句
 - 取り出した結果を整列するときに使用する
 - ASC(デフォルト)→昇順
 - DESC→降順

INSERTの構文

```
INSERT INTO テーブル名 (列名1, 列名2, ..., 列名n)  
VALUES (列1の値, 列2の値, ..., 列nの値);
```

```
INSERT INTO テーブル名  
SET 列名1=値, 列名2=値, 列名n=値 ;
```

UPDATEの構文

UPDATE テーブル名

SET 列名1=値, 列名2=値, 列名n=値

[WHERE 条件式]

[ORDERD BY 列名|式 [ASC | DESC]]

[LIMIT レコード数]

DELETEの構文

DELETE FROM テーブル名

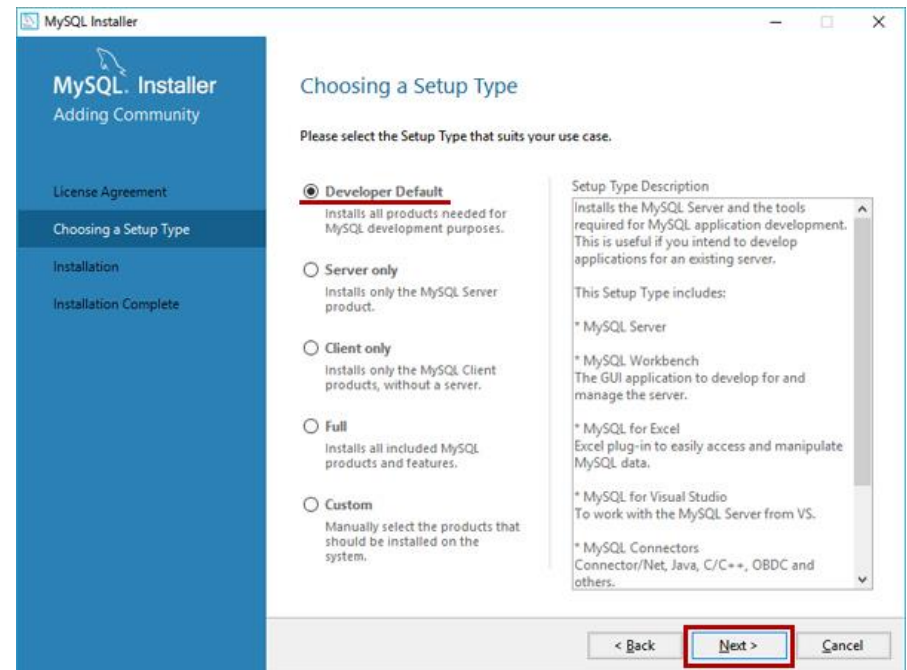
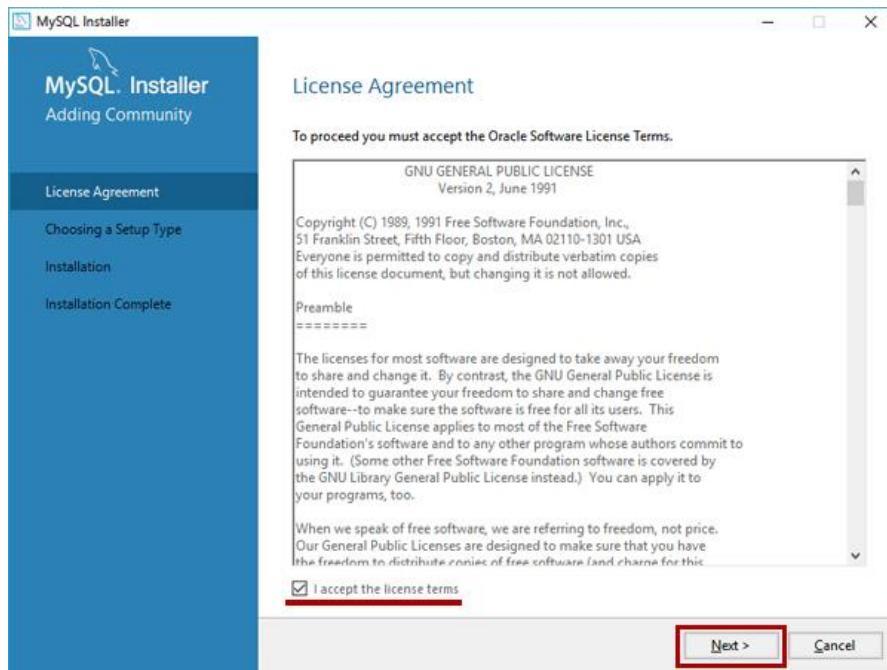
[WHERE 条件式]

[ORDERD BY 列名|式 [ASC | DESC]]

[LIMIT レコード数]

MySQLの使用

- <https://dev.mysql.com/downloads/mysql/>



MySQLの使用

Linux(Ubuntu)の場合

```
$ sudo apt-get install mysql-server mysql-client  
$ sudo mysql_secure_installation
```

Yes or Noの問は全部y(yes)で良い

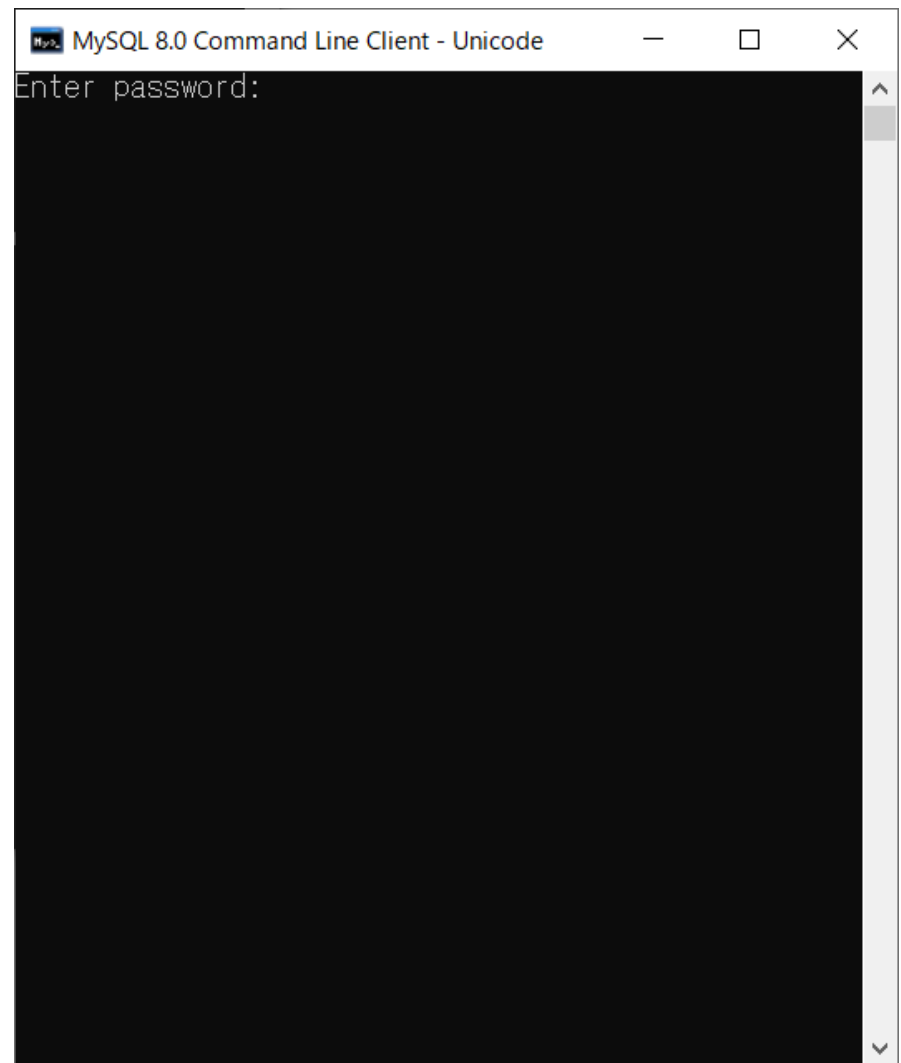
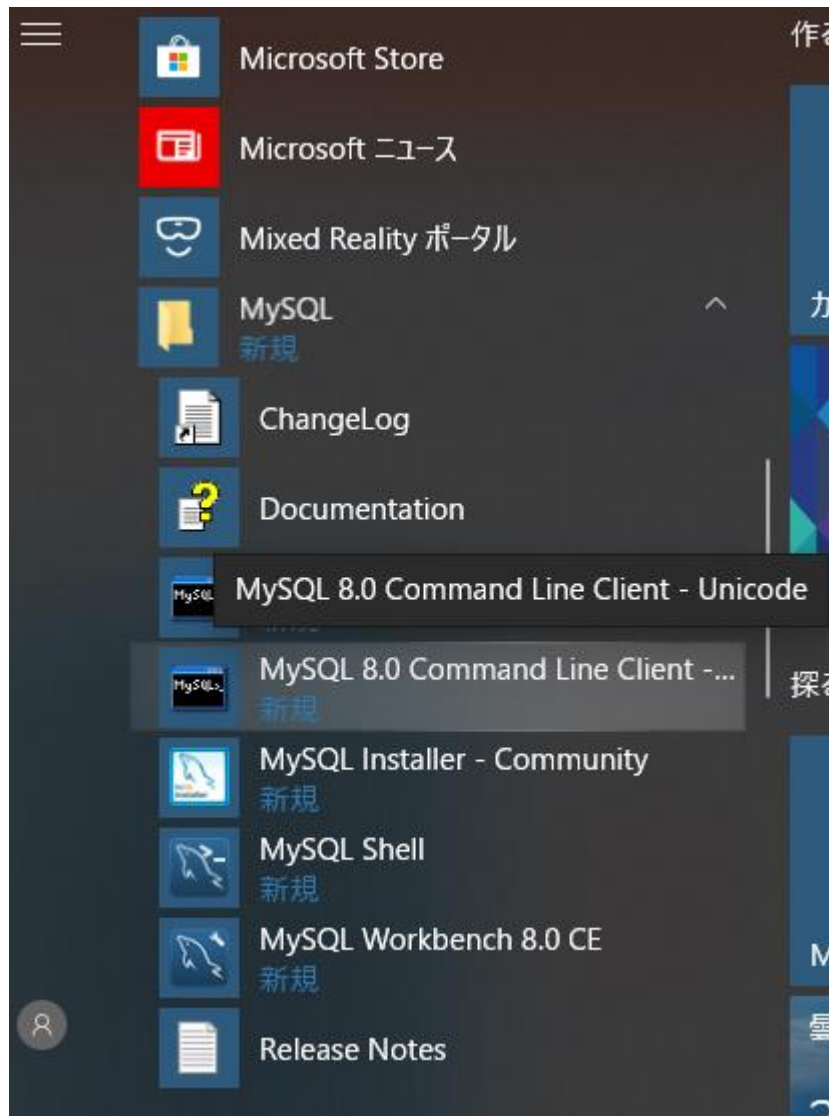
Low Medium Strongの問は Low(0を入力)で良い

パスワードはお好きなように

```
$ sudo mysql -u root
```

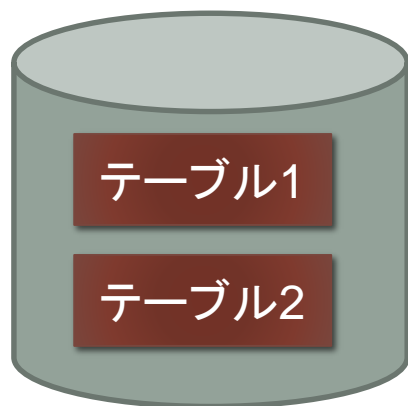
で起動

mysql> のプロンプトが出ればOK

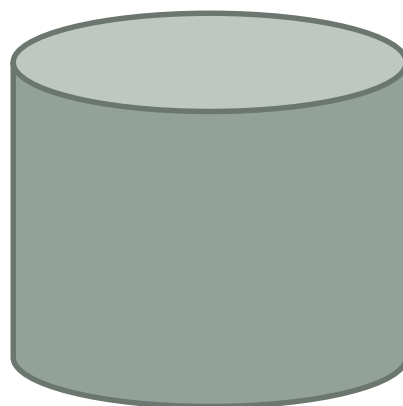


MySQLの構成

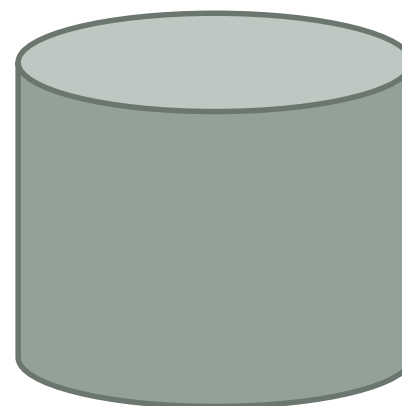
データベースの中へテーブルを複数格納する



データベース1



データベース2



データベース3

MySQL 8.0 Command Line Client - Unicode

Enter password: *********
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.17 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> **show databases;**

Database
information_schema
mysql
performance_schema
sakila
sys
world

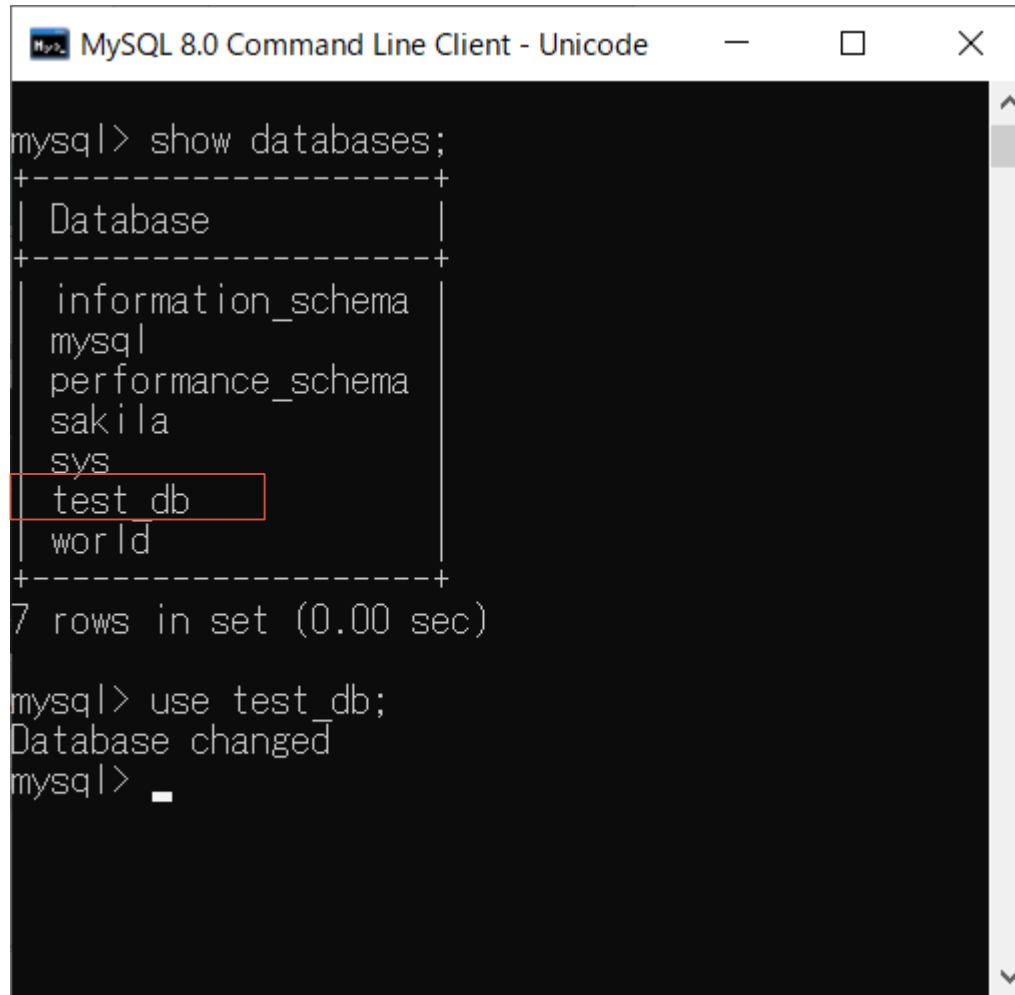
6 rows in set (0.01 sec)

mysql> **create database test_db;**

Query OK, 1 row affected (0.01 sec)

mysql> **_**

まずは新データベース test_db を作成



```
MySQL 8.0 Command Line Client - Unicode
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| test_db |
| world |
+-----+
7 rows in set (0.00 sec)

mysql> use test_db;
Database changed
mysql> _
```

```
mysql> show databases;
```

でtest_db が作成されていることを確認する

```
mysql> use test_db;
```

で、以後の処理はtest_db内で行うことを宣言 ※test_dbの指定を省略可能

テーブル作成

MySQL 8.0 Command Line Client - Unicode

```
mysql> create table dep(  
-> dep_code integer primary key,  
-> dep_name char(2) not null  
-> );
```

Query OK, 0 rows affected (0.05 sec)

```
mysql> create table student(  
-> id integer primary key,  
-> name nchar(20) not null,  
-> dep_code integer references dep(dep_code),  
-> email varchar(30) unique  
-> );
```

Query OK, 0 rows affected, 1 warning (0.04 sec)

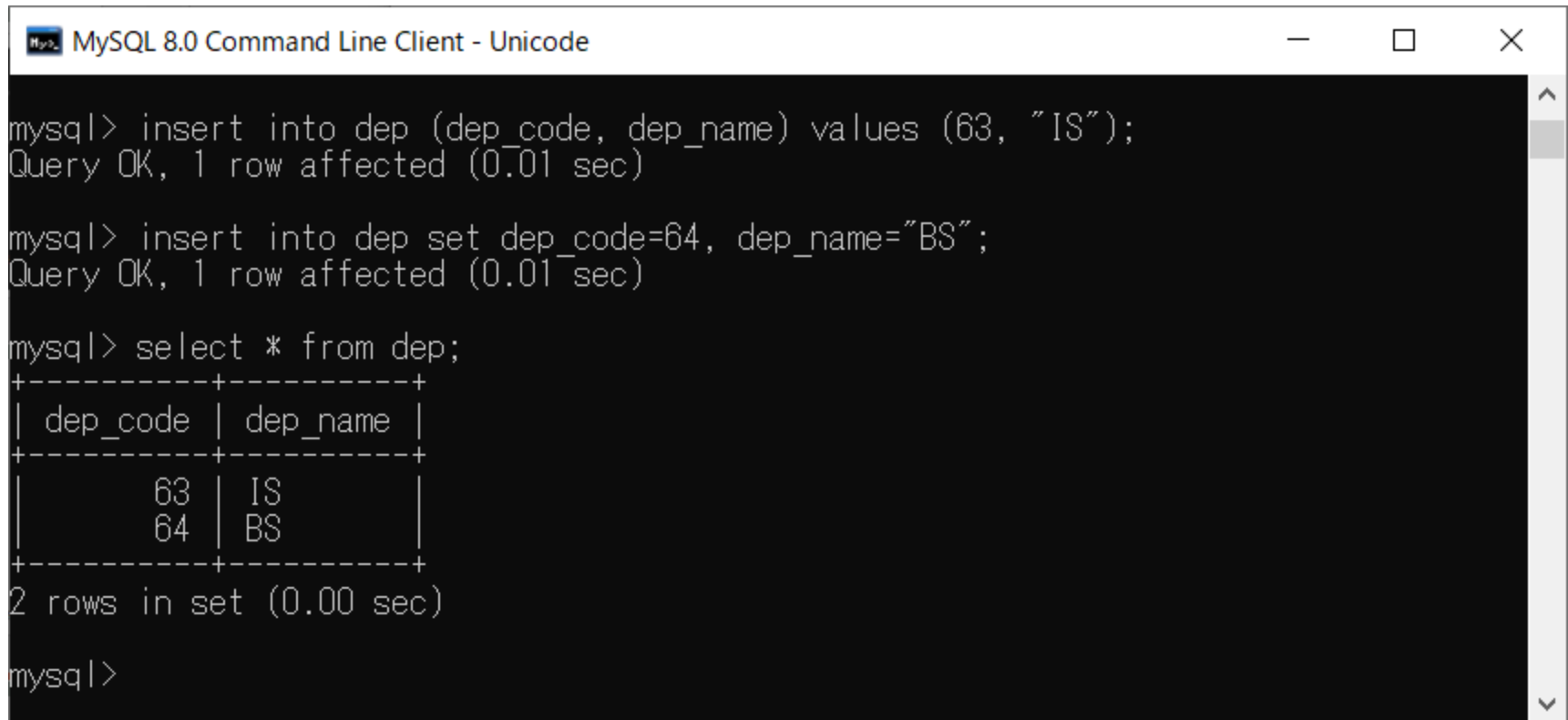
```
mysql> show tables;
```

```
+-----+  
| Tables_in_test_db |  
+-----+  
| dep                |  
| student            |  
+-----+
```

2 rows in set (0.01 sec)

```
mysql> _
```

データ格納(dep)



```
MySQL 8.0 Command Line Client - Unicode

mysql> insert into dep (dep_code, dep_name) values (63, "IS");
Query OK, 1 row affected (0.01 sec)

mysql> insert into dep set dep_code=64, dep_name="BS";
Query OK, 1 row affected (0.01 sec)

mysql> select * from dep;
+-----+-----+
| dep_code | dep_name |
+-----+-----+
|        63 | IS      |
|        64 | BS      |
+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

データ格納(student)

```
MySQL 8.0 Command Line Client - Unicode
mysql> insert into student(id,name,dep_code,email)
      -> values(1, "Alice", 63, "alice@is.jp");
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> insert into student(id,name,dep_code,email)
      -> values(2, "Bob", 64, "bob@bs.jp");
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> insert into student(id,name,dep_code,email)
      -> values(3, "Charlie", 63, "char@is.jp");
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> insert into student(id,name,dep_code,email)
      -> values(4, "Dave", 64, "dave@bs.jp");
Query OK, 1 row affected (0.00 sec)

mysql> select * from student;
+----+-----+-----+-----+
| id | name  | dep_code | email |
+----+-----+-----+-----+
| 1  | Alice | 63      | alice@is.jp |
| 2  | Bob   | 64      | bob@bs.jp   |
| 3  | Charlie | 63     | char@is.jp  |
| 4  | Dave  | 64      | dave@bs.jp  |
+----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> _
```

SELECTを試す

```
MySQL 8.0 Command Line Client - Unicode

mysql> select * from student cross join dep; 直積
+-----+-----+-----+-----+-----+-----+
| id | name | dep_code | email | dep_code | dep_name |
+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 63 | alice@is.jp | 63 | IS |
| 1 | Alice | 64 | alice@is.jp | 64 | BS |
| 2 | Bob | 63 | bob@bs.jp | 63 | IS |
| 2 | Bob | 64 | bob@bs.jp | 64 | BS |
| 3 | Charlie | 63 | char@is.jp | 63 | IS |
| 3 | Charlie | 64 | char@is.jp | 64 | BS |
| 4 | Dave | 63 | dave@bs.jp | 63 | IS |
| 4 | Dave | 64 | dave@bs.jp | 64 | BS |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> select * from student natural join dep; 自然結合
+-----+-----+-----+-----+-----+
| dep_code | id | name | email | dep_name |
+-----+-----+-----+-----+-----+
| 63 | 1 | Alice | alice@is.jp | IS |
| 64 | 2 | Bob | bob@bs.jp | BS |
| 63 | 3 | Charlie | char@is.jp | IS |
| 64 | 4 | Dave | dave@bs.jp | BS |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

外部結合を試す

```
MySQL 8.0 Command Line Client - Unicode

mysql> insert into dep set dep_code=61, dep_name="MA";
Query OK, 1 row affected (0.01 sec)

mysql> select * from dep;
+-----+-----+
| dep_code | dep_name |
+-----+-----+
|        61 | MA      |
|        63 | IS      |
|        64 | BS      |
+-----+-----+
3 rows in set (0.00 sec)
```

右外部結合(studentにMAの学生がない状態)

```
mysql> select * from student right join dep on student.dep_code=dep.dep_code;
+-----+-----+-----+-----+-----+-----+
| id | name | dep_code | email | dep_code | dep_name |
+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 63 | alice@is.jp | 63 | IS |
| 2 | Bob | 64 | bob@bs.jp | 64 | BS |
| 3 | Charlie | 63 | char@is.jp | 63 | IS |
| 4 | Dave | 64 | dave@bs.jp | 64 | BS |
| NULL | NULL | NULL | NULL | 61 | MA |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
```


検索条件(WHERE)を試す

```
MySQL 8.0 Command Line Client - Unicode

mysql> select * from student where name like 'A%';
+-----+-----+-----+-----+
| id | name | dep_code | email |
+-----+-----+-----+-----+
| 1 | Alice | 63 | alice@is.jp |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from student where dep_code between 62 and 64;
+-----+-----+-----+-----+
| id | name | dep_code | email |
+-----+-----+-----+-----+
| 1 | Alice | 63 | alice@is.jp |
| 2 | Bob | 64 | bob@bs.jp |
| 3 | Charlie | 63 | char@is.jp |
| 4 | Dave | 64 | dave@bs.jp |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from student where dep_code in (61,63);
+-----+-----+-----+-----+
| id | name | dep_code | email |
+-----+-----+-----+-----+
| 1 | Alice | 63 | alice@is.jp |
| 3 | Charlie | 63 | char@is.jp |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

ORDER BYを試す

```
MySQL 8.0 Command Line Client - Unicode
mysql> select * from student order by dep_code asc;
+----+-----+-----+-----+
| id | name  | dep_code | email      |
+----+-----+-----+-----+
| 1  | Alice | 63       | alice@is.jp |
| 3  | Charlie | 63       | char@is.jp  |
| 2  | Bob   | 64       | bob@bs.jp   |
| 4  | Dave  | 64       | dave@bs.jp  |
+----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from student order by id desc;
+----+-----+-----+-----+
| id | name  | dep_code | email      |
+----+-----+-----+-----+
| 4  | Dave  | 64       | dave@bs.jp |
| 3  | Charlie | 63       | char@is.jp  |
| 2  | Bob   | 64       | bob@bs.jp   |
| 1  | Alice | 63       | alice@is.jp |
+----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from student order by email desc;
+----+-----+-----+-----+
| id | name  | dep_code | email      |
+----+-----+-----+-----+
| 4  | Dave  | 64       | dave@bs.jp |
| 3  | Charlie | 63       | char@is.jp  |
| 2  | Bob   | 64       | bob@bs.jp   |
| 1  | Alice | 63       | alice@is.jp |
+----+-----+-----+-----+
4 rows in set (0.00 sec)
```

質問あればどうぞ