# モバイル プログラミング 実習2

M.Katsube

2016/11/21

日本工学院八王子専門学校

# 午後

- SQL演習
  - データの選択
  - データの挿入
  - データの更新
  - データの削除

# 来週の予告

# ペアプロ

- 来週から「ペアプログラミング」を実施
  - ペアプロとは2人1組でコードを書くこと。1人がコードを打ち、もう1人はサポート役。
  - この授業では15分程度で交代します
- パートナーは毎週変更
  - 隣の席の人とペアを組みます
  - 席を毎回ランダムに変更します。**入室時に自分の席を確認**してください。

# SQL演習

# ※注意※

SQLの説明は驚くほど眠くなりますが、聞かないと確実にわからなくなる苦行です。

# ※注意※

先生も新人の頃、会社に10万円ほど出してもらって研修に行きましたが、地獄でした(眠くて倒れそうだが寝れないため)。

# 結論

# 耐える

# SQL is not Programming

- SQLはデータベースに「問い合わせ」を行うための物
  - データをください
  - データを保存してください
  - データを更新してください
  - データを削除してください

# CRUD(クラッド)

- Create　生成
- Read　読み込み
- Update 更新
- Delete　削除

# CRUD(クラッド)

↓SQL

- Create = INSERT
- Read = SELECT
- Update = UPDATE
- Delete = DELETE

# 文法と呼称

# 文法と呼称

```
SELECT id, name
FROM Monster;
```

# 文法と呼称

SELECT文

```
SELECT id, name
FROM Monster;
```

※○○文と呼びます

# 文法と呼称

大文字でも小文字どちらでも。
全角はダメ。

途中で改行してもOK

SELECT句 SELECT id, name

FROM句 FROM Monster;

○○句という言い方をします。

最後はセミコロンで終わる

# SELECT

# SELECT 1 - 射影

Monsterテーブルから
すべてのレコードの
IDと名前を取得しなさい

※稀に「射影」といったりもします。

# SELECT 2 - 射影

Monsterテーブルから
すべてのレコードの
すべてのカラムを取得しなさい

※稀に「射影」といったりもします。

# SELECT句

- 取得したいカラムを列挙する
  - –カラム名をカンマで区切る
  - –すべてのカラムを対象とする場合はアスタリスク(*)で代用可能

# SELECT 3 - WHERE

Monsterテーブルから
HPが100を超えるレコードのすべてのカラムを取得しなさい

# SELECT 4 - WHERE

Monsterテーブルから

IDが1のレコードのすべての
カラムを取得しなさい

# SELECT 5 - WHERE

Monsterテーブルから

IDが1又は2のレコードの

すべてのカラムを取得しなさい

# SELECT 6 - WHERE

Monsterテーブルから

IDが1,3,6,8,20のレコードのすべてのカラムを取得しなさい

# SELECT 7 - WHERE

Monsterテーブルから

HPが100を超え、

なおかつ

MPが50以下のレコード

のすべてのカラムを取得しなさい

# SELECT 8 - WHERE

Monsterテーブルから

nameが「スライム」から始まるレコードのすべてのカラムを取得しなさい

# SELECT 9 - WHERE

Monsterテーブルから

nameが「スライム」を**含む**レコードのすべてのカラムを取得しなさい

# SELECT 10 - WHERE

Monsterテーブルから

nameが「スライム」を**含まない**レコードのすべてのカラムを取得しなさい

# SELECT 11 - WHERE

Monsterテーブルから
HPが10～100のレコードのすべてのカラムを取得しなさい

# WHERE句

- 演算子を用い、レコードの絞込を行う句。
- **演算子**
  - =, >, <, >=, <=, !=, <>
  - AND, OR
  - NOT
  - LIKE
  - IN
  - BETWEEN a AND b

# SELECT 10 – ORDER BY

Monsterテーブルからすべてのレコード、すべてのカラムを取得しなさい。

その際にHPが低い物から順番(昇順)に取得しなさい。

# SELECT 11 – ORDER BY

Monsterテーブルからすべてのレコード、すべてのカラムを取得しなさい。

その際にHPが高い物から順番(降順)に取得しなさい。

# SELECT 12 – ORDER BY

Monsterテーブルからすべてのレコード、すべてのカラムを取得しなさい。

その際にHPが高い物, 次にMPの高い物から順番に取得しなさい。

# ORDER BY句

- レコードの並び替え(ソート)を行う
- カラム名を列挙する
  - カンマ区切り
  - 優先順位の高い物ほど先に記述
- 昇順と降順を指定できる
  - カラム名 ASC =昇順
  - カラム名 DESC = 降順
  - カラム名 = 降順

# SELECT 13 – DISTINCT

HPのバリエーションがどれくらいあるのか知りたい。

例えば次のデータの場合は

1, 10, 10, 30, 30

以下の結果がほしい

1, 10, 30

# SELECT 14 – GROUP BY

HPのバリエーションとレコード数が知りたい。

例えば次のデータの場合は

1, 10, 10, 30, 30

以下の結果がほしい

1=1件, 10=2件, 30=2件

# SELECT 15 – 関数

1. 最も大きいHP値が知りたい。
2. HPの平均値が知りたい
3. HPの合計が知りたい
4. レコード数が知りたい

# GROUP BY句 / 関数

- レコードを "まとめる" ことができる
  - DISTINCTは重複をなくすだけだが、GROUP BY句は関数などと相性がよくパワフルな集計が可能になる。
- SQLには集計などを行う関数が用意されている
  - COUNT(), MAX(), MIN(), AVG() ...

# まだまだある SELECT文

- HAVING句
- CASE
- LIMIT, OFFSET
- サブクエリー (副問合せ)
- JOIN (複数のテーブルから取出す)
  - これは後日授業でやります

# INSERT

# INSERT 1

```
INSERT INTO Monster (id, name)
VALUES(1, 'スライム');
```

```
INSERT INTO Monster
VALUES(1, 'スライム');
```

※id, nameしかないテーブルの場合、カラム名は省略できる

# INSERT 2

```
INSERT INTO Monster (id, name)
VALUES
  (1, 'スライム'),(2, 'スライムベス');
```

※一つのINSERT文で複数のレコードを追加できる

# INSERT 3

```
INSERT INTO Monster (id, name)
  SELECT id, name
  FROM Monster2;
```

※SELECTの結果を、直接INSERTすることもできる

# UPDATE

# UPDATE 1

```
UPDATE Monster
SET    HP=10
```

※全レコードが対象となる

# UPDATE 2

```
UPDATE Monster
SET      HP=10
WHERE  ID=1;
```

※WHERE句を忘れないように!

# UPDATE 3

```
UPDATE Monster
SET      HP=HP+10
WHERE  ID=1;
```

※計算式を書くこともできる

# UPDATE 4

```
UPDATE Monster
SET      HP=HP+10,
         MP=MP+5
WHERE  ID=1;
```

※複数のカラムを一度に変更することもできる

# DELETE

# DELETE 1

```
DELETE FROM Monster;
```

※すべてのレコードが対象となる

# DELETE 2

```
DELETE FROM Monster
WHERE id=1;
```

※WHERE句を忘れないように!

# DELETE 3

```
TRUNCATE TABLE Monster;
```

- これでもテーブルを空にできます。
- TRNCATEはテーブルが裏側で保持している値も初期化します。真っ白な状態で作り直したいときはこちらを利用します。