



GEEK JOB

転職エージェント

プログラミングキャンプ

データベース基礎

目次

- データベースとは
- 基本的なSQL文
- XAMMPとMYSQLによるデータベース構築
- 課題

データベースとは(1/2)

Database

-記録しておきたい情報を、コンピュータ上に蓄積したもの

社員情報データベース

社員テーブル

社員ID	氏名	年齢
0001	添田亮司	23
0002	林星河	29
	・	
	・	
	・	

部署テーブル

部署ID	部署名	
	...	

役職テーブル

	...	

データベースとは(2/2)

データベースは、いろいろな種類の構築方法があるが、メジャーなのは RDB (Relational Database)と呼ばれる種類。表(テーブル)として保存された行(ワンセットの情報)と列(情報の種類を細かく分けたもの)でデータを管理する

テーブル名:社員テーブル

→
カラム名がその列の示す値の内容を分類している

社員ID	氏名	年齢
0001	添田亮司	23
0002	林星河	29

↑
林星河さんに関する一件のレコード

RDBの基本構造

- ・複数の表(テーブル)で管理
- ・表にはそれぞれ名前が付く(テーブル名)
- ・テーブルは行と列で構成されている
- ・一つの行(レコード)はまとまった一件のデータに対応している(左図ではIDと氏名、年齢が横軸でひとまとめになっている)
- ・列(カラム)はデータの要素に対応している
- ・対応しているソフトウェアは MySQL、PostgreSQL など

基本的なSQL

/* SQLの基本ルール:コメント

このようにスラッシュ&アスタリスクで囲まれた範囲はコメントとして認識され、実行には影響を与えない。*/

/* 前ページのテーブルから、社員IDが2番の名前と年齢を検索、表示するSQL(SELECT文) */

SELECT 氏名,年齢 FROM 社員テーブル WHERE 社員ID=2;

/* 途中で改行を入れても一つの文として認識される。「;」を入れることで1命令として区切る */

SELECT 氏名,年齢
FROM 社員テーブル
WHERE 社員ID=2;

SQL

-データベースを操作するための専用言語。記述方法はDBの種類によって若干異なるが基本的には同じ

SQLを使用するタイミングは、主に

- ・コマンドライン上で直接入力
- ・別のプログラム(.php)にとある方法で記述
- ・.sqlという拡張子で保存しておいて読み込みなどがある

左図の記述のように、SELECT FROM WHEREといった命令用の単語を用いて記述する(大文字小文字の区別はない)

基本的には、SELECT、DELETE、UPDATE、INSERTという四大命令を軸に、より詳しい指定のFROMやWHEREが修飾語としてくっついている

SELECT-データの検索

/* memberテーブルからbirthdayカラムとtellカラムのレコードをすべて表示 */

```
SELECT birthday, tell FROM member;
```

/* memberテーブルからnameが'林星河'のbirthdayカラムとtellカラムのレコードをすべて表示 */

```
SELECT birthday, tell FROM member  
WHERE name='林星河';
```

/* memberテーブルからレコードをすべて表示 */

```
SELECT * FROM member;
```

SELECT

-最も頻繁に使われる命令。テーブルから目的のデータを指定して取得することができる

基本的な構文

SELECT カラム名 1,カラム名 2....
FROM テーブル名
(+WHEREなどの修飾)

ここを「*」にすると全カラムの値を表示できる

WHEREなどの修飾をうまく使うことで、望んだ形のデータを得ることができる

SELECT-データの検索

ID	name	tell	age	birhday
001	添田亮司	0123-456	32	1983-06-24
002	林星河	3456-321	22	1993-03-01
003	青木仁志	999-000	27	1988-11-11

```
SELECT birthday, tell  
FROM member  
WHERE name='林星河';
```



```
birthday    tell  
1993-03-01 3456-321
```

UPDATE-データの更新

```
/* memberテーブルのtellカラムの内容を指定した値で  
更新(全レコードに対して適応)*/  
UPDATE member  
SET tell=01234567890;
```

```
/* memberテーブルからmemberIDが002のレコード  
の名前と誕生日を更新 */  
UPDATE member  
SET name='山本昌',birthday='1967-8-11'  
WHERE memberID=002;
```

UPDATE

-テーブルに存在するデータを指定して書き換える。検索して指定 → 書き換え

基本的な構文

UPDATE テーブル名
SET カラム名1=値1,カラム名2=値2....
(+WHEREなどの修飾)

WHEREを使用してレコードを特定しないと全件のレコードが更新されてしまうため、UPDATE文ではWHERE句はほぼ必須

UPDATE-データの更新

ID	name	tell	age	birhday
001	添田亮司	0123-456	32	1983-06-24
002	林星河	3456-321	22	1993-03-01
003	青木仁志	999-000	27	1988-11-11

UPDATE member SET
name='山本昌',birthday='1967-
8-11' WHERE ID=002;

ID	name	tell	age	birhday
001	添田亮司	0123-456	32	1983-06-24
002	山本昌	3456-321	22	1967-8-11
003	青木仁志	999-000	27	1988-11-11

DELETE-データの削除

```
/* memberテーブルの全レコードを削除 */  
DELETE  
FROM member;
```

```
/* memberテーブルからnameが'林星河'のデータを削除*/  
DELETE  
FROM member  
WHERE name='林星河';
```

DELETE

-すでにテーブルに存在するデータ行をまるごと指定して削除する。検索して指定 → 削除

基本的な構文

```
DELETE  
FROM テーブル名  
(+WHEREなどの修飾)
```

列を指定して削除することはできない。
WHEREを使用してレコードを特定しないと
全件のレコードが削除されてしまうので注意

DELETE-データの削除

ID	name	tell	age	birhday
001	添田亮司	0123-456	32	1983-06-24
002	林星河	3456-321	22	1993-03-01
0003	青木仁志	999-000	27	1988-11-11
ID	name	tell	age	birhday
001	添田亮司	0123-456	32	1983-06-24
003	青木仁志	999-000	27	1988-11-11

```
DELETE  
FROM member  
WHERE name='林星河';
```

INSERT-データの追加

```
/* memberテーブルに、カラム memberIDとbirthdayの  
値を持ったレコードを追加 (空カラムができてしまう)*/  
INSERT INTO member(memberID,birthday)  
VALUES(002,'1993-03-01');
```

```
/* memberテーブルに、空カラムのないレコードを追加  
*/  
INSERT INTO member  
VALUES(004,'田中将大','0123-456','1988-11-1');
```

```
/* memberテーブルに複数のレコードを追加 */  
INSERT INTO member(name)  
VALUES('林星河'),('添田亮司'),('青木仁志');
```

INSERT

-テーブルに新しいレコード (行)を追加する。
WHERE 句などで追加する行を指定することができず、必ず最終行に追加される

基本的な構文

```
INSERT INTO テーブル名 (列名1,列名2...)  
VALUES (値1,値2....)
```

(列名1,列名2...)に(値1,値2....)が順番に対応して格納されていく。両者の個数が異なるとエラーに。指定していない列にはデータは格納されず、空カラムとなる。

そのテーブルのすべての列に値を追加する場合、(列名1,列名2...)は丸ごと省略できる

INSERT-データの追加

ID	name	tell	age	birhday
001	添田亮司	0123-456	32	1983-06-24
002	林星河	3456-321	22	1993-03-01
003	青木仁志	999-000	27	1988-11-11
ID	name	tell	age	birhday
001	添田亮司	0123-456	32	1983-06-24
002	林星河	3456-321	22	1993-03-01
003	青木仁志	999-000	27	1988-11-11

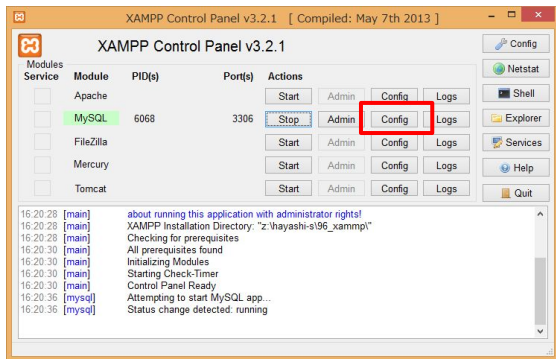
INSERT INTO member(ID,
birthday)
VALUES(004,'1988-11-1');

XAMPPとMySQLによるDB作成の流れ

まずmysqlに対し文字コード設定を行う必要がある

1.XAMPPからMySQLの設定ファイル(my.ini)を開く

2.文字コード設定を追記それぞれの最終行に追加すればいい(最後の[mysql]の項目はwinのみ)



[client]

...
default-character-set=utf8

[mysqld]

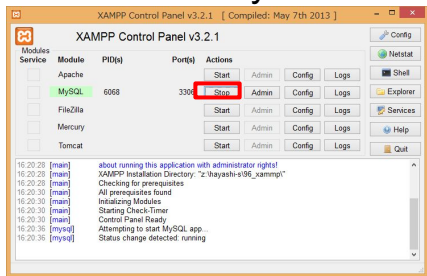
...
character-set-server=utf8

[mysql]

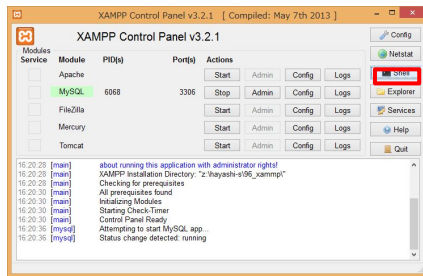
...
default-character-set=cp932

XAMPPとMySQLによるDB作成の流れ

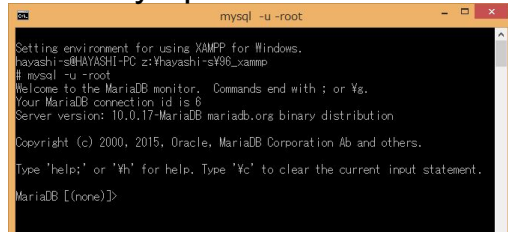
1.XAMPPからMySQLを起動



2.Shellを起動



3.「mysql -h localhost -u root」と入力しmysqlにログイン



4.以下のように入力し、ユーザーを作成(日本語の部分は自分が記述してください)

```
CREATE USER 'ユーザー名を入力' '@localhost'  
IDENTIFIED BY 'パスワードを入力';
```

```
MariaDB [(none)]> CREATE USER 'hayashi'@localhost IDENTIFIED BY 'password';  
Query OK, 0 rows affected (0.00 sec)
```

5.以下のように入力し、データベースを作成

```
CREATE DATABASE データベース名
```

```
MariaDB [(none)]> create database challenge_db;  
Query OK, 1 row affected (0.00 sec)
```

6.ユーザーにデータベース操作の権限を付与

```
GRANT ALL ON データベース名.* TO 'ユーザー名' '@localhost';
```

XAMPPとMySQLによるDB作成の流れ

7.「exit」と入力していったんMySQLから抜け、「mysql -u ユーザー名 -p」と入力しエンターで更にパスワードを入力

```
MariaDB [(none)]> exit
Bye
hayashi-s@HAYASHI-PC z:¥hayashi-s¥96_xampp
# mysql -u hayashi -p
Enter password: *****
```



8. 以下のように入力して操作するデータベースを選択

```
USE データベース名 ;
MariaDB [(none)]> use challenge_db;
Database changed
MariaDB [challenge_db]> _
```



9. 以下のように入力してテーブルを作成

```
CREATE TABLE テーブル名 (カラム名1 データ型
1,カラム名2 データ型2,カラム名3 データ型3...);
MariaDB [challenge_db]> create table test(testID int, name varchar(10), age int)
;
Query OK, 0 rows affected (0.17 sec)
```



10. あとはデータをINSERTしていけばOK!

- ・SQLのデータ型について基本的には
INT型(数字型)
VARCHAR型(文字列型)
DATE型(日付型)
を覚えておけば問題ない

RDBの大きな特徴

主キー(primary key)

-必ず値が存在し、同じ値が二度と現れない固有の値。この値さえわかれば、
唯一のレコードを特定できる。こうしたカラムのことを 主キーと呼ぶ

社員テーブル

社員ID	氏名	年齢
0001	添田亮司	26
0002	林星河	23
0003	青木仁志	26

社員テーブルの主キー

主キーの特性

- 1.必ず値がある(NOT NULL)
- 2.重複しない(UNIQUE)

RDBの大きな特徴

外部キー(foreign key)

-他のテーブルに情報を分けたとき、参照したいテーブルの主キーの値を入力していくことになる。このようなカラムを **外部キー**と呼ぶ

外部キーの特性

- ・参照先のテーブルにちゃんと対応したレコードがある(所属カラムに003などは存在できない)
- ・外部の主キーとなるテーブルを先に用意しないといけない

社員テーブル

社員ID	氏名	年齢	所属
0001	添田亮司	26	0001
0002	林星河	23	0001
0003	青木仁志	26	0002

外部の主キー
を参照

部署テーブル

部署ID	部署名
0001	開発
0002	マーケ

部署テーブル
の主キー

Q.そもそもテーブルを分ける意味は？

A.データを管理しやすくするため。例えば部署名が社員テーブルにそのまま書かれている場合、部署の名称が変わってしまった際に社員テーブルの該当するレコードすべてを書き直す必要がある。データを分けることで、該当するテーブルのレコード一つを書き直すだけでよくなる

課題

導入:まずは「XAMMPとMySQLによるDB作成の流れ」を参考に「Challenge_db」というデータベースを作成しておきましょう

提出方法:課題の成否を確認するため、各SQL文の入力をテキスト化したものと、最終的な結果が表示されたコンソール画面をスクリーンショットしたものを、PDF形式でまとめて提出してください。

課題1:[このリンク](#)にあるようなテーブルをCREATEし、記述されているレコードをINSERTしてください。このINSERT文は別途テキスト化してパワポに追記をしておいてください。全件INSERT後、SELECT *により全要素を表示してください。

課題

課題1で作成したテーブルに対し、以下の処理を行う 期限:2日

課題2:profilesIDが3のレコードのみを検索して表示

課題3:nameが「高橋 清」のレコードのみを検索して表示

課題4:ageが「24」のレコードのみを検索して表示

課題5:ageが「25以上」のレコードのみを検索して表示(WHERE句の記述を変える)

課題6:nameに「実」が含まれているレコードを表示(部分一致の修飾語句を追記する)

課題

課題7:profilesIDが2のnameを「吉田 茂」に更新する。この1件をSELECTして表示する

課題8:birthdayが2000-12-24のレコードを削除する。SELECT*で表示する

課題

以下の課題を行いなさい 期限:5日

課題9:[このリンク](#)にあるようなテーブル群(複数シートあり)をCREATEし、記述されているレコードをINSERTしてください。CREATE時には同時に主キーと外部キーの指定もしてください(Primary Key と Foreign Keyを宣言)。全件INSERT後、SELECT *を実行することにより全要素を表示してください