

## データベース実習

# 03. MySQLによるテーブル操作

株式会社ジードライブ

# 今回学ぶこと

---

- MySQLのデータ型
- テーブルの作成、削除

# MySQLのデータ型

- データを格納するテーブルの作成においては、あらかじめ項目名とカラム(列)に入るデータの型を決めておく必要がある
  - エクセルのような柔軟性はない
  - 変数と同じく、データを入れるに際して型の指定が必要

項目名 →

id	name	price
1	レターセット	480
2	筆ペン(細字)	180
3	万年筆	5300
4	スタンプセット	1800

整数が入るカラム 文字列が入るカラム 整数が入るカラム

# MySQLのデータ型

## 数値型

- **TINYINT, INT, BIGINT** 等 ... 整数
- **FLOAT, DOUBLE** 等 ... 実数

## 文字列型

- **CHAR(n)** ... サイズ固定文字列
  - n は最大文字数 (0 ~ 255)
- **VARCHAR(n)** ... サイズ可変文字列
  - n は最大文字数 (0 ~ 65535)
    - ⇒ 実際には1行あたりのサイズ制限があるので、65535は設定できない
    - ⇒ utf8mb4の場合、カラムが1つしかない場合であっても、16383が最大
- **TEXT** ... 最大65535文字の文字列
  - utf8mb4の日本語(4byte/1字)は16383文字

# MySQLのデータ型

---

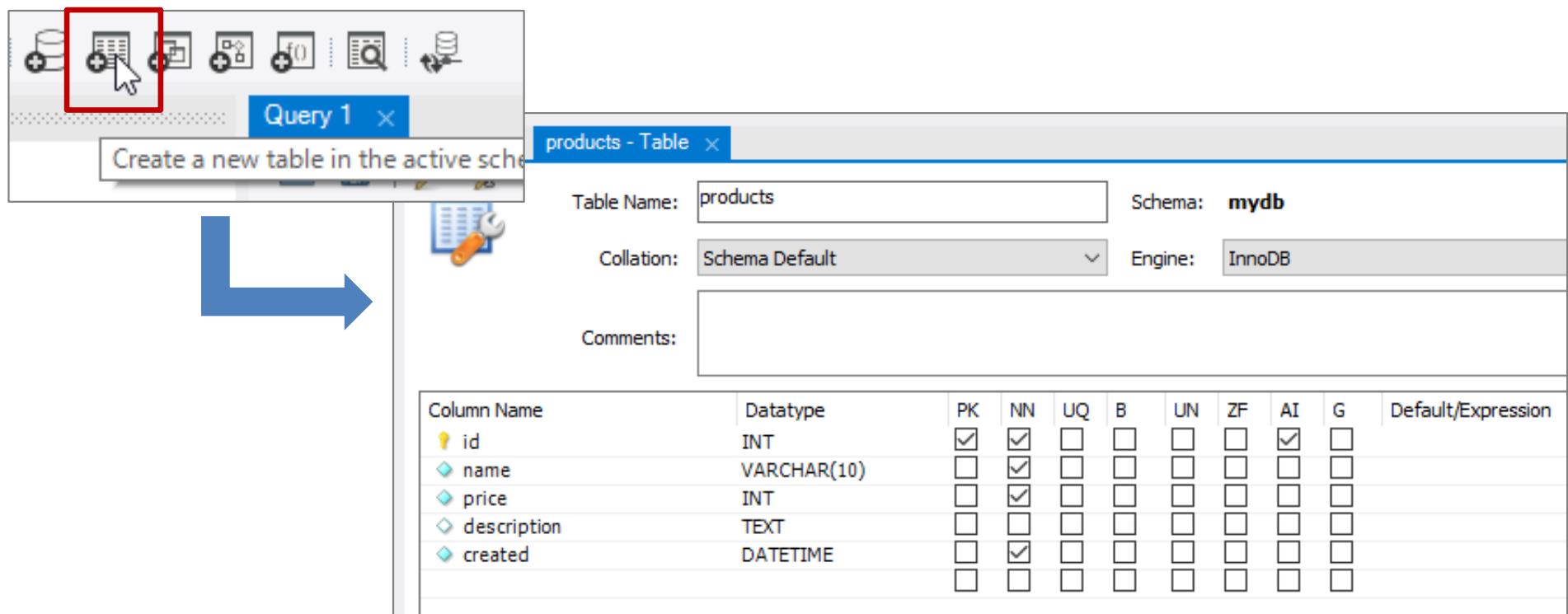
## 日付型と時刻型

- **DATE** ... 日付
  - 例 : '2020-08-05'
- **TIME** ... 時刻
  - 例 : '12:34:56'
- **DATETIME** ... 日付と時刻
  - 例 : '2020-08-05 12:34:56'

# MySQL Workbench によるテーブル操作

# テーブルの作成

- [Create a new table ...]をクリックして表示されたパネルにテーブルの内容を入力する



# テーブル作成時の制約

- PK … Primary Key
  - カラムにプライマリキー制約(後述)を設定する
- NN … Not Null
  - カラムの値が NULL であることを許可しない
- UQ … Unique Key
  - カラムにユニークキー制約(後述)を設定する
- B … Binary Column
  - カラム値の照合順序を大文字小文字の区別ありにする  
(`utf8_general_ci` で作成したDB内でも、`Binary Column`属性を付けると  
`utf8_bin` で比較されるようになる)

# テーブル作成時の制約

- **UN** … Unsigned
  - カラムの値を符号なし整数とする
- **ZF** … Zero Fill
  - 格納する値をゼロ詰めする(例：000001)
- **AI** … Auto Increment
  - カラムに自動的に連番を振る
- **G** … Generated Column
  - 別のカラム値を元に計算された結果を格納する仮想カラム
- **Default**
  - カラムのデフォルト値を指定する

# プライマリキー制約

---

- テーブル内のレコード(一行分のデータ)を一意に識別するためのカラムに設定する
  - 会員番号やユーザIDなど
  - 主キーとも呼ばれる
- プライマリキーの制約
  - プライマリキーは1つのテーブルの中で1つしか設定できない
  - カラム内に、同じ値の重複があってはいけない
  - NULL値を設定できない

# ユニークキー制約

- テーブル内で同じ値のデータの重複を許さないカラムに設定する
  - メールアドレスなど
- プライマリキーとの違い
  - 1つのテーブル内で複数設定できる
  - NULL値を許可することも可能 (NULL値は重複してもよい)

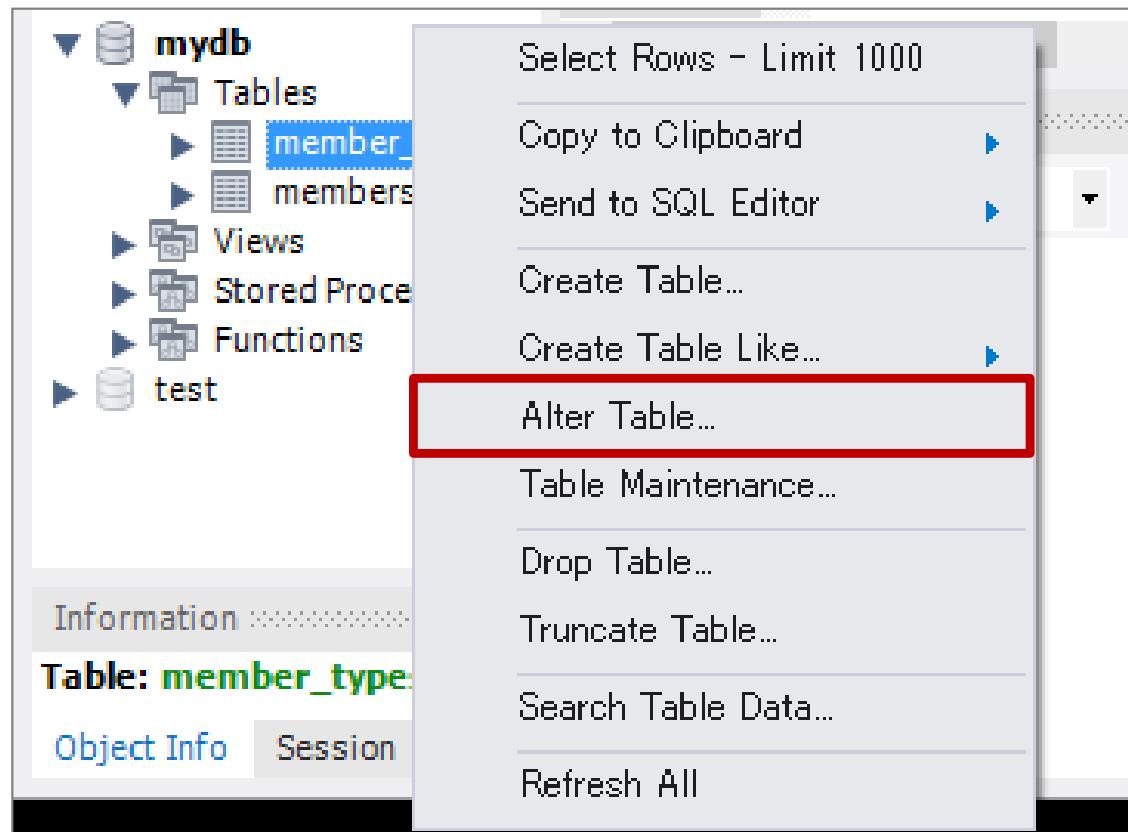
プライマリキー：テーブルに一つだけ

ユニークキー：複数存在可。NULL値も可

<b><u>id</u></b>	<b><u>name</u></b>	<b><u>email</u></b>	<b><u>my_number_card</u></b>
1	Kimura Ichiro	ichi@abc.com	16564546
2	John Kitayama	kita@xyz.com	
3	Kato Jiro		
4	Robert Sagawa	rob@example.com	54983558

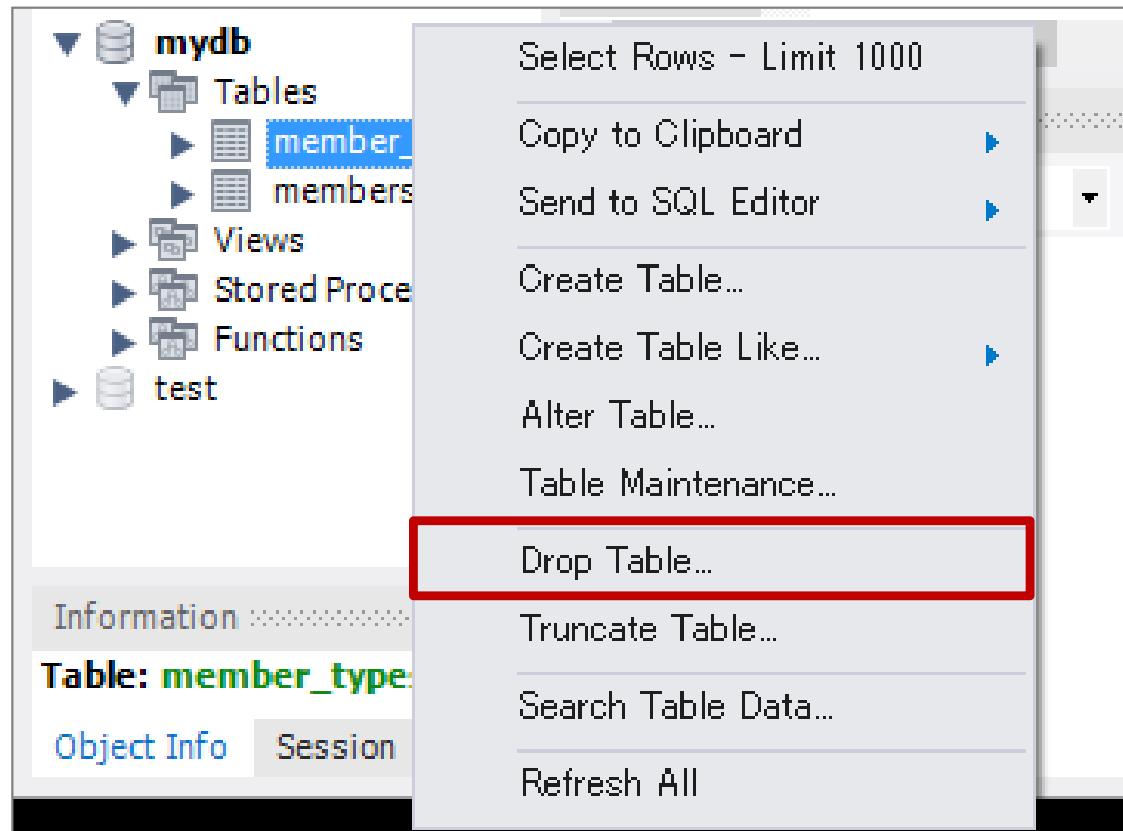
# テーブルの修正

- テーブル名を右クリック  
⇒ [Alter Table...] を選択する



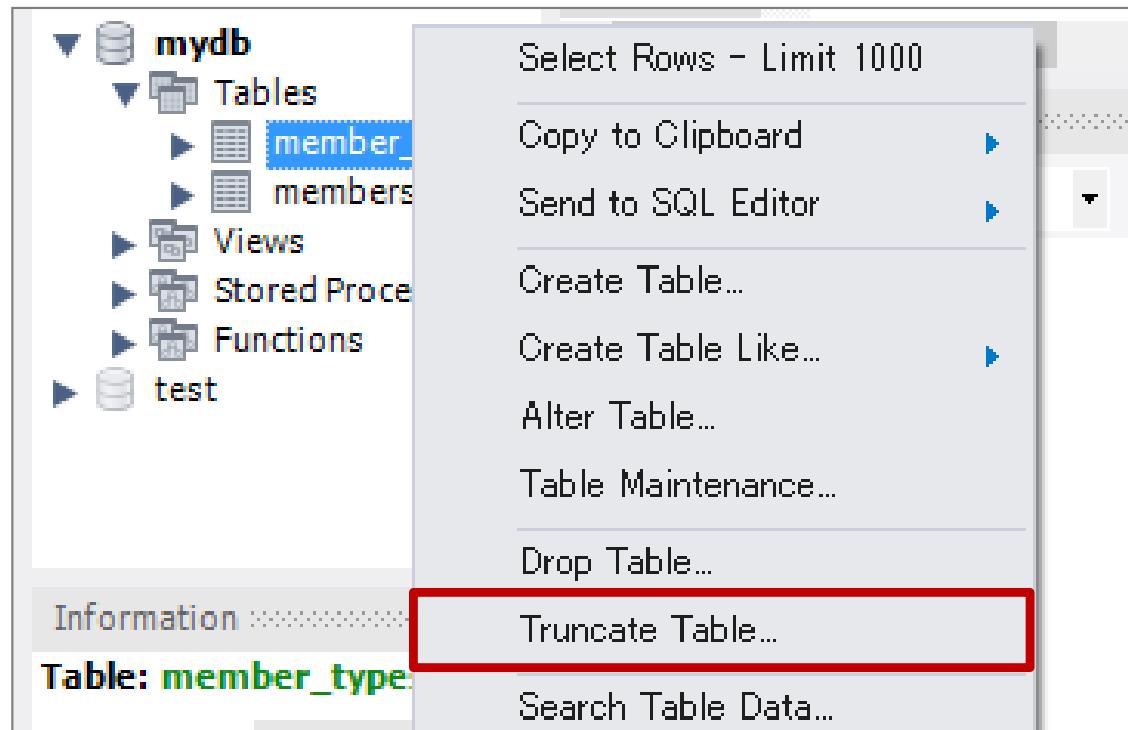
# テーブルの削除

- テーブル名を右クリック  
⇒ [Drop Table...] を選択する



# テーブルを空にする

- ・ テーブル名を右クリック  
⇒ [Truncate Table...] を選択する
  - テーブル自体は削除されない（中身が空になるだけ）



# SQLによるテーブル操作

# テーブル一覧の表示

- 書式

```
SHOW TABLES;
```

- 選択されたデータベース内にあるテーブルの一覧を表示する

# テーブルの作成

- 書式

```
CREATE TABLE テーブル名(  
    カラム名1 型 オプション,  
    カラム名2 型 オプション,  
    ...);
```

- オプションには以下のようなものが入る
  - NULL / NOT NULL
  - AUTO\_INCREMENT
  - PRIMARY KEY
  - UNIQUE
  - DEFAULT値

# テーブルの作成

例：id, name, age, address, created のカラムをもつテーブルの作成

```
CREATE TABLE members (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(10) NOT NULL,
    age INT,
    address VARCHAR(255),
    created DATETIME NOT NULL);
```

カラム名	データ型	主キー	自動連番	NULL
id	整数	◎	○	×
name	可変長文字列(最大10字)			×
age	整数			○
address	可変長文字列(最大255字)			○
created	日時			×

# テーブルの削除

- 書式

```
DROP TABLE テーブル名;
```

例

```
DROP TABLE members;
```

- テーブルを削除せず、中のデータだけを空にする場合は、TRUNCATE文を使う

```
TRUNCATE テーブル名;
```

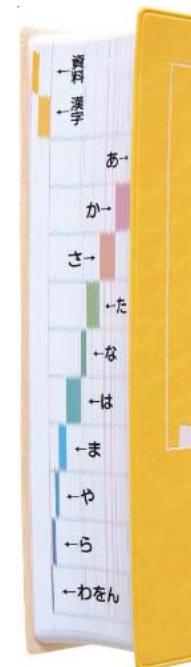
# 練習

---

- 練習03-1

# インデックス

- 特定のカラムのデータを高速に検索するための機能
  - カラムに対してインデックスを作成する
  - プライマリーキー やユニークキーに設定されたカラムは自動的にインデックスが作成される
  - 検索に使用するカラムにインデックスを設定すると効果がある



# インデックスの作成

- 書式

```
CREATE INDEX インデックス名 ON テーブル名(カラム名);
```

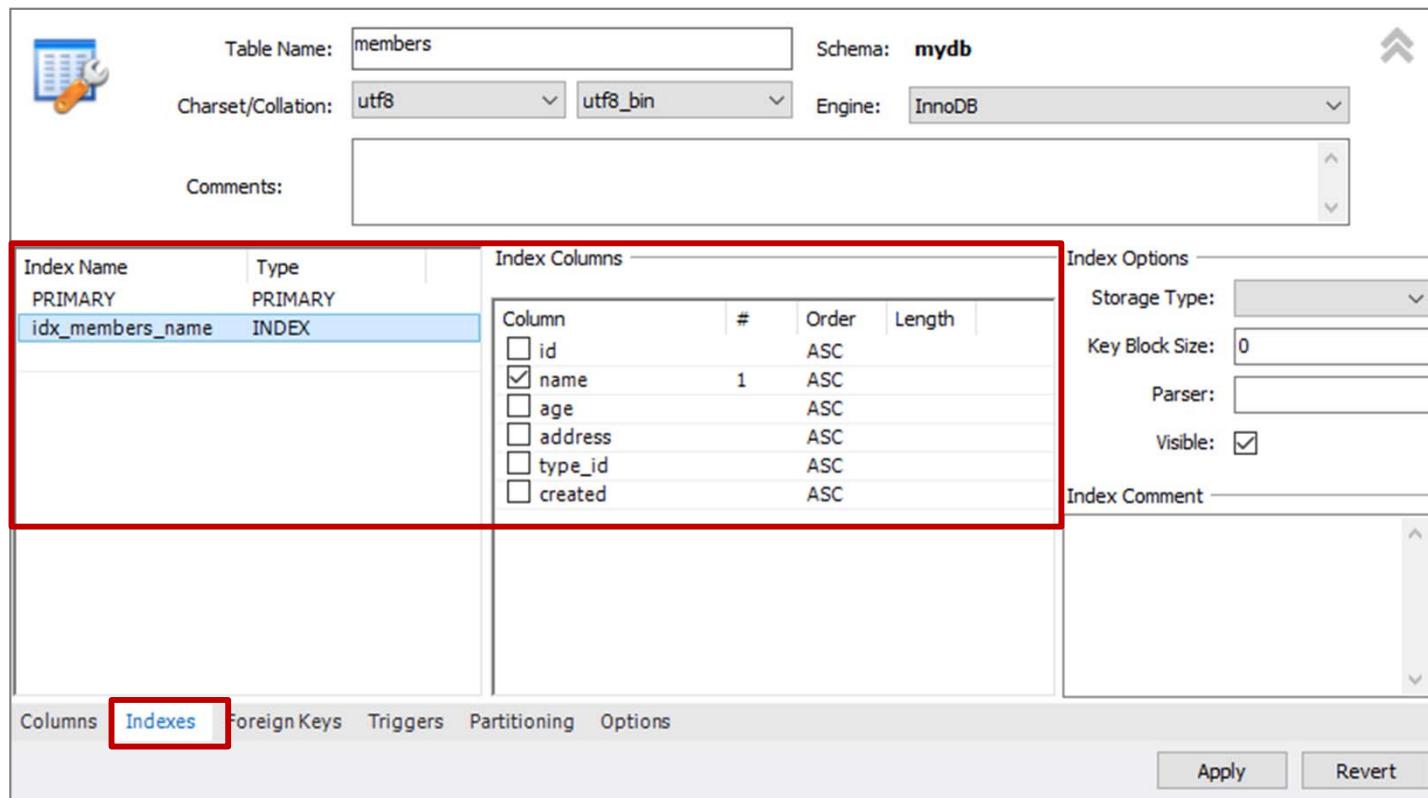
例

```
CREATE INDEX idx_members_name ON members(name);
```

membersテーブルのnameカラムにidx\_members\_nameというインデックスを追加する

# インデックスの確認

- テーブルに設定されたインデックスは、テーブル情報パネルの[ Indexes ] タブで確認できる



# 複合主キー

- テーブルに設定できる主キーは1つだが、必ずしも1つのカラムで主キーの役割を担う必要はない  
⇒ 複数のカラムで、1つの主キーとしての役割（データを一意に識別する役割）を果たすこともできる

2つのカラムで主キーの役割を果たしている

<b>school_id</b>	<b>student_id</b>	<b>name</b>	<b>age</b>
ABC	0036587	山田太郎	25
ABC	0054215	山田太郎	25
XYZ	0054215	山田太郎	25
XYZ	0087443	山田太郎	25
STU	0036587	山田太郎	25

# 複合主キー

- 書式

```
PRIMARY KEY(カラム名1, カラム名2, ...)
```

例

```
CREATE TABLE student_list (
    school_id CHAR(3),
    student_id INT,
    name VARCHAR(20),
    age INT,
    PRIMARY KEY(school_id, student_id)
);
```