


# 12. 中間まとめ

(データベース演習)

URL: <https://www.kkaneko.jp/de/de/index.html>

金子邦彦



- 
- ① SQLスキルの向上
  - ② データベース運用スキル
  - ③ 問題解決能力と論理的思考力

# Access での注意点

- SQLビューでは、SQL文を1つずつ実行  
(複数まとめての一括実行ができない)
- **CREATE TABLE** では、「実行」の後、**画面が変化しない**が実行できている
- **INSERT INTO** では、「実行」の後、**確認表示**が出る。その後、**画面が変化しない**が実行できている

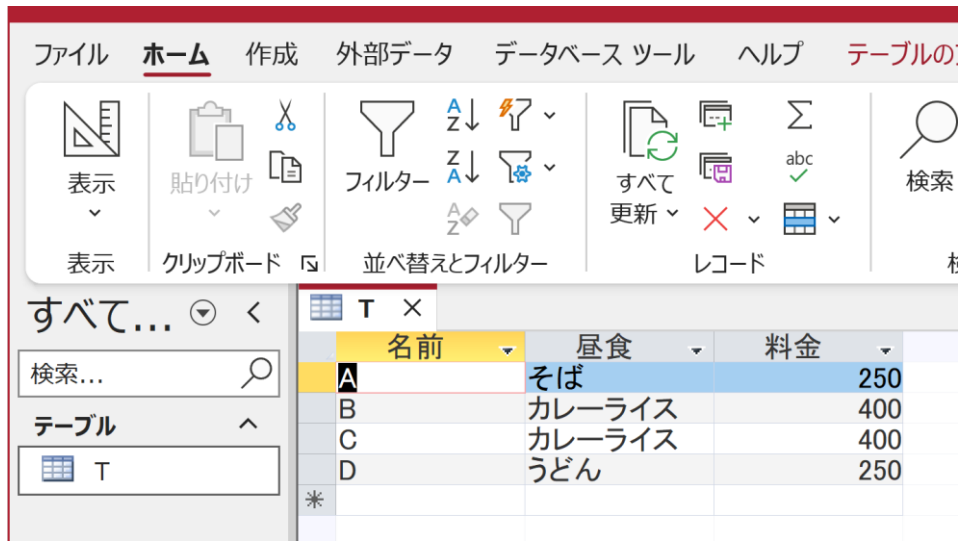
# Access でのテーブルデータの確認

- SQL で確認

```
SELECT * FROM T;
```

	名前	昼食	料金
A		そば	250
B		カレーライス	400
C		カレーライス	400
D		うどん	250
*			

- テーブルビューで、「テーブル名」をダブルクリック



# SQL 問い合わせ（クエリ）で使用する2つのビュー

```
SELECT * from 商品;
```



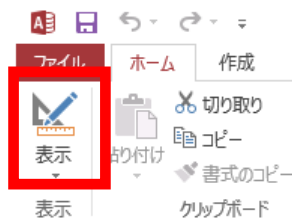
実行



ID	名前	単価
	みかん	50
	2りんご	100
	3りんご	150
*	(新規)	0

## SQL ビュー

SQL 文の 作成、編集



表示 + SQL ビュー



## データシートビュー

問い合わせ（クエリ）の  
結果

マウス操作でビューを切り替え

# Access の SQL ビューを用いた問い合わせ

① Access の **SQLビュー**開く

② **SQL 文**の編集。 **select, from, where** を使用

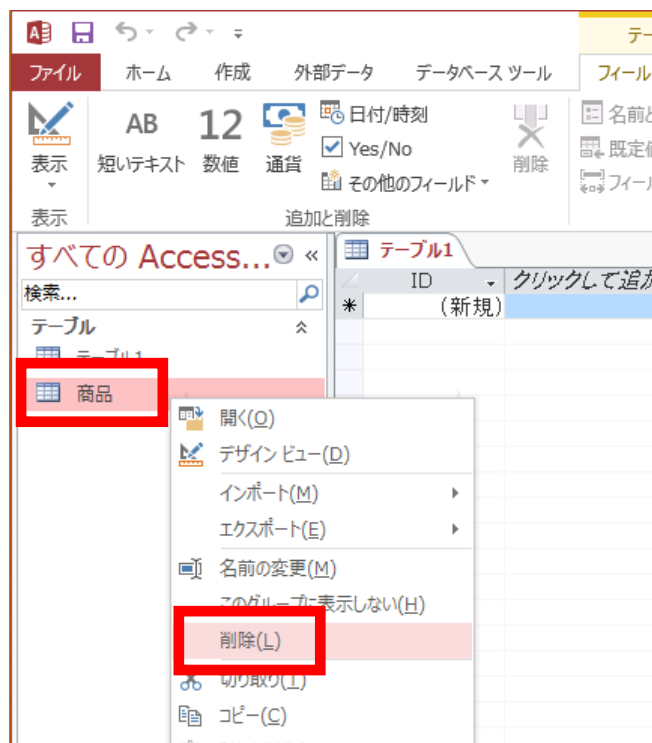
例: `select * from テーブル名 where 列1 = 値1;`

③ **SQL 文**の**実行**

実行の結果、**データシートビュー**に画面が変わり、そこに**問い合わせの結果**が表示される

④ さらにSQL 文の編集、実行を続ける場合には、**画面を SQL ビューに切り替える**

# 間違ってしまったときは、テーブルの削除 を行ってからやり直した方が早い場合がある



テーブルビューで、削除したいテーブルを**右クリック**して、「**削除**」

テーブルを削除するときは、  
間違っても必要な**テーブル**を削除しない  
ように、十分に注意する！  
(元に戻せない)

# 演習の手順

1. パソコンを使用する

前もって Access をインストールしておくこと

2. Access を起動する

3. Access で、「**空のデータベース**」を選び、「**作成**」をクリック。

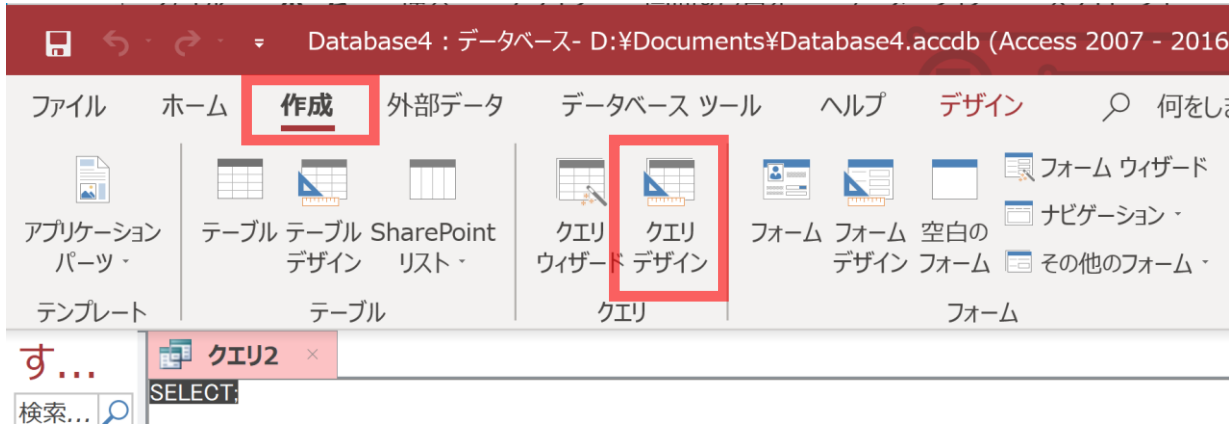




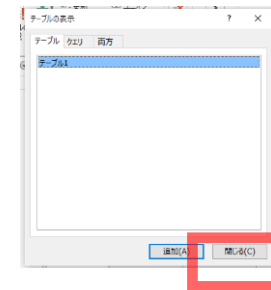
## 4. テーブルツール画面が表示されることを確認

The screenshot displays the Microsoft Access 2016 interface. The title bar shows the file name "Database7 : データベース- D:\¥Documents¥Database7.accdb (Access 2007 - 2016 ファイル形式)..." and the user name "金子 邦彦". The ribbon is set to "フィールド" (Fields) under the "テーブル" (Table) group. The ribbon includes sections for "名前と標題" (Name and Title), "既定値" (Default Value), "フィールド サイズ" (Field Size), "プロパティ" (Properties), "ルックアップの変更" (Change Lookup), "fx 式の変更" (Change Formula), "メモの設定" (Memo Settings), "書式設定" (Formatting), "表示形式" (Display Format), and "フィールドの入力規則" (Field Validation Rules). The main area shows a table named "テーブル1" with a single column "ID" and a row labeled "(新規)". A yellow highlight is on the "ID" column header, and a blue highlight is on the "(新規)" row. The status bar at the bottom indicates "レコード: 1 / 1" and "フィルターなし 検索".

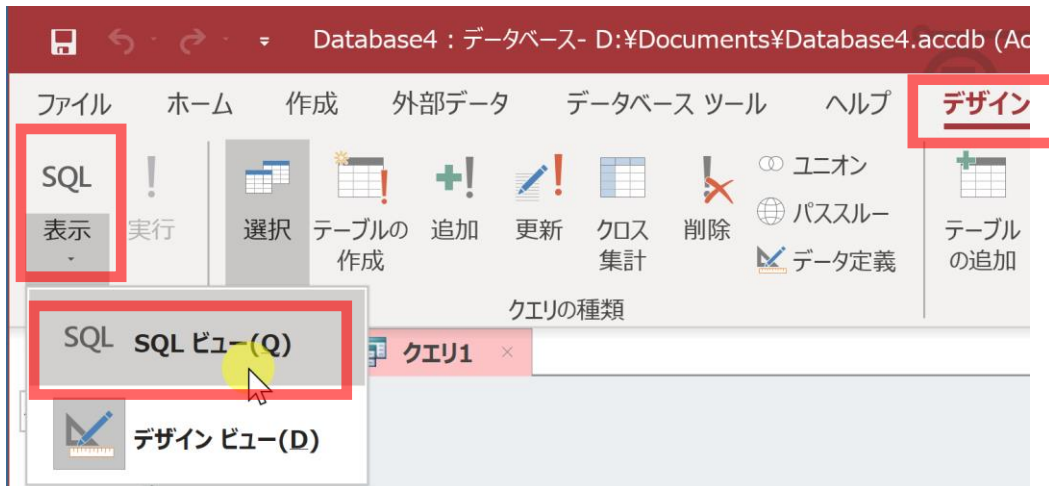
## 5. 次の手順で、SQLビューを開く。



① 「作成」タブで、「クエリデザイン」をクリック



このような表示が出たときは「閉じる」をクリック



② 「デザイン」タブで、「表示」を展開し「SQLビュー」を選ぶ



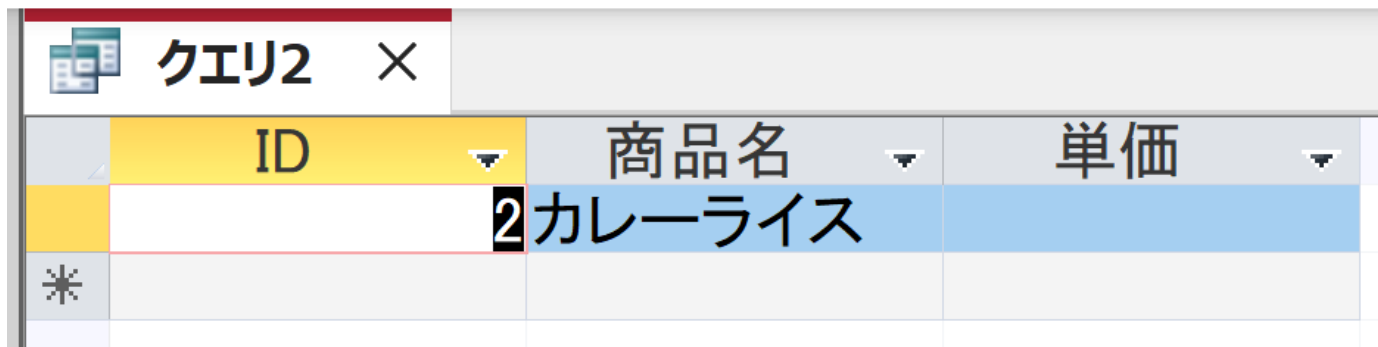
## 演習 1 . NULL

### 【トピックス】

1. NULL
2. IS NULL

## Access での NULL の表示

- Access では「NULL」は空欄で表示される



ID	商品名	単価
2	カレーライス	
*		

「NULL」とは表示されない。これは正常動作である

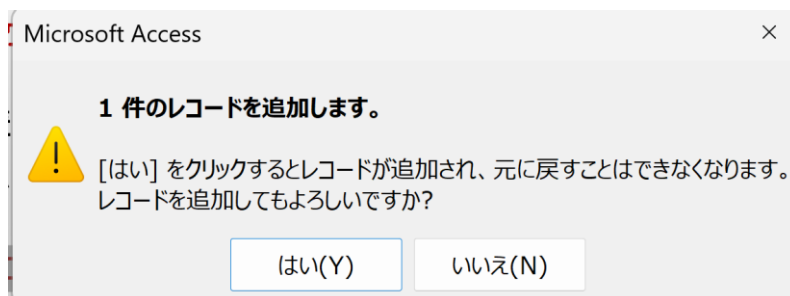
SQL ビューに、次の SQL を1つずつ入れ、「実行」ボタンで、SQL文を実行。結果を確認

```
CREATE TABLE メニュー (  
ID INTEGER PRIMARY KEY,  
商品名 TEXT,  
単価 INTEGER);
```

```
INSERT INTO メニュー VALUES (1, 'かき氷', 400);
```

```
INSERT INTO メニュー VALUES (2, 'カレーライス', NULL);
```

```
INSERT INTO メニュー VALUES (3, 'サイダー', 200);
```



INSERT INTOでは、「実行」の後、確認表示が出る。その後、画面が変化しないが実行できている

SQL ビューに、次の SQL を 1 つずつ入れ、「実行」ボタンで、SQL文を実行。結果を確認

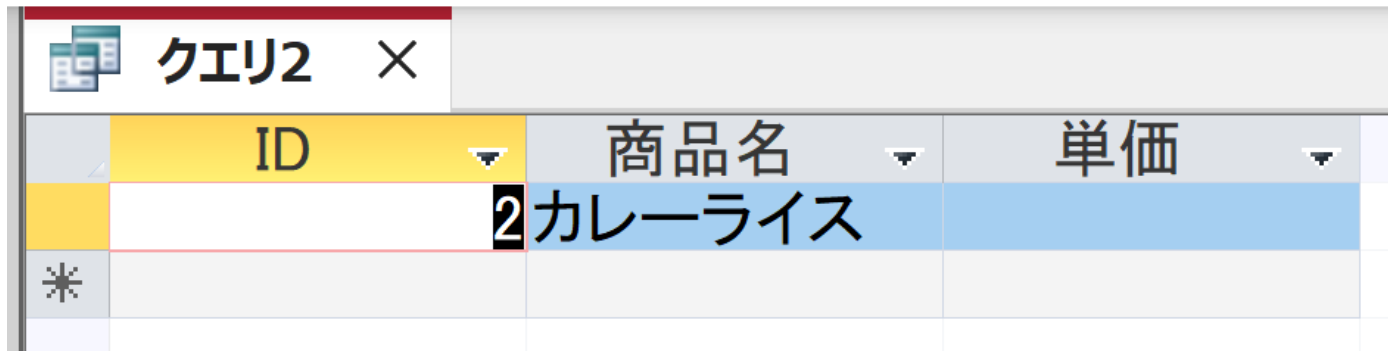
```
SELECT * FROM メニュー ;
```

```
SELECT * FROM メニュー WHERE 単価 IS NULL ;
```

```
SELECT * FROM メニュー WHERE 単価 >= 0 ;
```

## Access での NULL の表示

- Access では「NULL」は空欄で表示される



ID	商品名	単価
2	カレーライス	
*		

「NULL」とは表示されない。これは正常動作である



## 演習 2. 種々のSQL問い合わせ. Access の SQL ビューを使用.

### 【トピックス】

1. AUTOINCREMENT
2. NULL
3. SELECT



# Access の AUTOINCREMENT

Access の AUTOINCREMENT : 自動の通し番号

(Access の固有機能)

MySQL では AUTO\_INCREMENT

- データ追加のたびに 1, 2, 3, . . . の通し番号が自動で設定される

1	商品A	100
2	商品B	200
3	商品C	150

自動の通し番号

SQL ビューに、次の SQL を1つずつ入れ、「実行」ボタンで、SQL文を実行。結果を確認

```
CREATE TABLE 商品 (  
    id AUTOINCREMENT PRIMARY KEY,  
    商品名 TEXT,  
    単価 INTEGER  
);  
  
CREATE TABLE 申し込み (  
    id AUTOINCREMENT PRIMARY KEY,  
    日時 DATETIME,  
    氏名 TEXT,  
    商品番号 INTEGER,  
    個数 INTEGER,  
    FOREIGN KEY (商品番号) REFERENCES 商品(id)  
);
```

「id AUTOINCREMENT PRIMARY KEY」は Access 固有の書き方。  
整数で自動の通し番号

# SQL ビューに、次の SQL を 1 つずつ入れ、「実行」ボタンで、SQL文を実行。結果を確認

```
INSERT INTO 商品(商品名, 単価) VALUES ('商品A', 100);
```

```
INSERT INTO 商品(商品名, 単価) VALUES ('商品B', 200);
```

```
INSERT INTO 商品(商品名, 単価) VALUES ('商品C', 150);
```

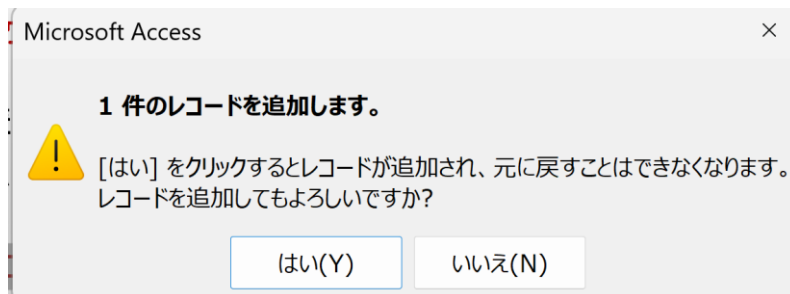
```
INSERT INTO 申し込み(日時, 氏名, 商品番号, 個数) VALUES (Now(), 'X', 1, 1);
```

```
INSERT INTO 申し込み(日時, 氏名, 商品番号, 個数) VALUES (Now(), 'X', 2, 10);
```

```
INSERT INTO 申し込み(日時, 氏名, 商品番号, 個数) VALUES (Now(), 'Y', 2, 5);
```

```
INSERT INTO 申し込み(日時, 氏名, 商品番号, 個数) VALUES (Now(), 'X', 1, 1);
```

Access では AUTOINCREMENT に設定した id に対して NULL を使って INSERT を実行できないので、別の書き方にしている



INSERT INTOでは、「実行」の後、確認表示が出る。その後、画面が変化しないが実行できている

SQL ビューに、次の SQL を 1 つずつ入れ、「実行」ボタンで、SQL文を実行。結果を確認

```
SELECT * FROM 商品;
```

```
SELECT * FROM 申し込み;
```

```
SELECT * FROM 申し込み  
INNER JOIN 商品 ON 申し込み.商品番号 = 商品.id;
```

```
SELECT 申し込み.日時, 申し込み.氏名, 申し込み.個数 * 商品.単価  
FROM 申し込み  
INNER JOIN 商品 ON 申し込み.商品番号 = 商品.id;
```

```
SELECT 氏名, SUM(個数 * 商品.単価)  
FROM 申し込み  
INNER JOIN 商品 ON 申し込み.商品番号 = 商品.id  
GROUP BY 氏名;
```

SQL ビューに、次の SQL を 1 つずつ入れ、「実行」ボタンで、SQL文を実行。結果を確認

```
SELECT *  
FROM 商品  
WHERE 単価 > (SELECT AVG(単価) FROM 商品);
```

```
SELECT MAX(単価)  
FROM 商品;
```


```
SELECT *  
FROM 申し込み  
WHERE 氏名 = 'X';
```

SQL ビューに、次の SQL を 1 つずつ入れ、「実行」ボタンで、SQL文を実行。結果を確認

```
SELECT DISTINCT (申し込み.氏名)
FROM 申し込み
INNER JOIN 商品 ON 申し込み.商品番号 = 商品.id
WHERE 商品.商品名 = '商品A';
```

```
SELECT 氏名, COUNT(*) AS 申し込み数
FROM 申し込み
GROUP BY 氏名;
```

```
SELECT SUM(申し込み.個数)
FROM 申し込み
INNER JOIN 商品 ON 申し込み.商品番号 = 商品.id
WHERE 商品.商品名 = '商品B';
```



## 演習 3 . データの更新、データの削除

### 【トピックス】

1. UPDATE ... SET
2. DELETE FROM

**SQLビュー**に、次のSQLを1つずつ入れ、「**実行**」ボタンで、**SQL文**を実行。結果を確認

1つずつ実行するたびに、商品テーブルの変化を確認してください。

```
UPDATE 商品  
SET 単価 = 120  
WHERE 商品名 = '商品A';
```

```
DELETE FROM 商品  
WHERE 商品名 = '商品C';
```

```
UPDATE 商品  
SET 単価 = 1000  
WHERE 商品名 = '商品B';
```



# 自習

## 自習 1. テーブルの作成とデータの挿入

SQLを使用して新しいテーブルを定義し、データを追加

学生テーブルを作成してください。

このテーブルには

ID (整数、主キー)

名前 (文字列)

年齢 (整数) の属性があります。

テーブルを定義し、学生のデータを追加してください。

ヒント: CREATE TABLE、INSERT INTO

## 自習 2. テーブルの集約とグループ化

学生 テーブルを使用して、年齢ごとに学生の数をカウントしてください。

作成したテーブルに学生のデータを追加してください。

ヒント:COUNT, GROUP BY

## 解答例

### 自習 1

```
CREATE TABLE 学生 (  
    ID INTEGER PRIMARY KEY,  
    名前 TEXT,  
    年齢 INTEGER  
);  
INSERT INTO 学生 VALUES(1, '山田太郎', 20);  
INSERT INTO 学生 VALUES(2, '鈴木花子', 19);
```

### 自習 2

```
SELECT 年齢, COUNT(*) FROM 学生 GROUP BY 年齢;
```