# de-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





# 12-1. イントロダクション



- データをテーブルと呼ばれる表形式で保存
- テーブル間は関連で結ばれる。複雑な構造を持ったデータ を効率的に管理することを可能に。



商品テーブルと購入テーブル

#### 商品

ID	商品名	単価
1	みかん	50
2	りんご	100
3	メロン	500



Xさんは、1のみかんと, 3のメロンを買った Yさんは、2のりんごを買った 購入テーブルの情報 商品テーブルの情報

4



- 1. データの整合性: リレーショナルデータベースは、デー
- **夕の整合性を保持するための機能**を有する。これにより、 誤ったデータや矛盾したデータが保存されるのを防ぐこ とができる。
- 2. 柔軟な問い合わせ(クエリ)能力: リレーショナルデー タベースのSQL(Structured Query Language)の使用に より、複雑な検索やデータの抽出が可能になる。
- トランザクションの機能: 一連の操作全体を一つの単位 として取り扱うことができる機能。これにより、データ の一貫性と信頼性が向上する。
- 4. セキュリティ: アクセス権限の設定などにより、セキュ リティを確保。

データの安全な保管、効率的なデータ検索・操作、ビジネス や研究の意思決定をサポート。



# 12-2. 演習

### Access での注意点

- ・SQLビューでは、<u>SQL文を1つずつ</u>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	イス		400
	С			カレーライ	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



8

# SQL 問い合わせ(クエリ)で使用する 2 つのビュー



# 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、<u>画面を SQL</u> 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**を削除するときは、 間違って必要な**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)



### 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、「<mark>作成</mark>」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

<b>.</b> 5	· ? · =	Dat	tabase	7:データベ	ース- D:¥Do	cuments¥l	Database7.	accdb (Acc	ess 2007 -	2016 ファイル	ル形式)…		金子 邦彦	8	— C	
ファイル	ホーム	作成	外语	部データ	データベー	ス ツール	ヘルプ	フィールド	テーブノ	<b>م</b> ا	何をしま	すか				
<b>於</b> 表示	AB 短いテキスト	<b>12</b> 数 値	通貨	■	<ul> <li>三名前</li> <li>二、既定</li> <li>二、二、二、二、二、二、二、二、二、二、二、二、二、二、二、二、二、二、二、</li></ul>	と標題 値 ルド サイズ		武 ルッ fx 式の 回 メモ	クアップの変更 )変更 の設定 -	書式語	设定 9 <u>60</u> -	.00 .00	必須 一意 インデックス	<b>検</b> 証		
表示		追加と	削除				プロパティ	ſ		老	辰示形式	ר <sup> </sup> כ	ィールドの入	力規則		^
す 検索 テー * 団 テ	■ テーフ II *	<b>゙ル1</b> × つ (新邦	▼ <mark>クリ</mark> 見)	リックして違												
•	レコード: 📢	1/	1	▶ <b>▶</b>   <b>▶※</b>	~ フィルター	なし検索										
データシート																×

### 5. 次の手順で、**SQLビュー**を開く.





# 演習1. NULL 【トピックス】 1. NULL 2. IS NULL



• Access では「NULL」は空欄で表示される



## 「NULL」とは表示されない。これは正常動作である

```
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

```
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 では「NULL」は空欄で表示される



## 「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 $\mathcal{O}$ 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

```
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

 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

Microsoft Access × 1 俳のレコードを追加します。 [はい]をクリックするとレコードが追加され、元に戻すことはできなくなります。 しコードを追加してもよろしいですか?

いいえ(N)

×

NSERT 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
23

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;

```
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

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';







発展演習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

発展演習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
```

#### 発展演習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);
```

発展演習2

SELECT 年齡, COUNT(\*) FROM 学生 GROUP BY 年齡;