de-10. 設計の基礎：主キー・外部キーを用いたテーブル間の関連付けとSQL実践

（データベース演習）

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

金子邦彦

10-1. イントロダクション

リレーショナルデータベースの仕組み

- 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

商品

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

関連

購入

購入者	商品番号
X	1
X	3
Y	2

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

関連

購入

購入者	商品番号
X	1
X	3
Y	2

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

主キー

- **主キー**は、**テーブルの各行を識別する**ためのキー

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

ID属性は主キーである

主キーの役割

- **商品**テーブルで、**すべての商品は一意の ID を持つ**
- 同じ ID をもつ商品は**2つ以上ない**
- 商品の行を**正確に特定するとき**に便利

例： ID = 2 である行　→　1 つに定まる

ID	商品名	単価
1	**みかん**	**50**
2	**りんご**	100
3	メロン	500
4	**みかん**	30
5	**りんご**	**50**

主キー

主キー制約と PRIMARY KEY

PRIMARY KEY はテーブル定義時に使用し，「**主キー制約**」を示す

```
CREATE TABLE テーブル名 (
    列名1 データ型 PRIMARY KEY,
    列名2 データ型,
    …
);
```

SQL の書き方

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

```
CREATE TABLE 商品 (
    ID INTEGER PRIMAY KEY,
    商品名 TEXT,
    単価 INTEGER);
```

主キー

複数属性を主キーとして指定

但し、**主キー**は1つの属性でも良いし、**複数の属性の組み合わせ**でもよい（書き方は下のようになる）

```
CREATE TABLE テーブル名 (
   列名1 データ型,
   列名2 データ型,
   ...
   PRIMARY KEY (列名1, 列名2)
);
```

SQL の書き方

学生ID	科目ID	得点
1	1001	90
1	1002	100
2	1001	85
2	1002	90
2	1003	95

主キー

```
CREATE TABLE 成績 (
   学生ID INTEGER,
   科目ID INTEGHER,
   得点 INTEGER,
   PRIMARY KEY(学生ID, 科目ID));
```

外部キー

外部キーは、**他のテーブルの主キーを参照する**キー

購入テーブル

外部キー

ID	購入者	商品ID	数量
1	X	1	10
2	Y	2	5

商品テーブル

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

主キー

購入テーブルの**外部キー**「商品ID」は、**購入**テーブルの**主キー**「ID」を**参照**

SQL によるテーブル定義

- **テーブル名**：**購入**
- **属性名**：**ID、購入者、商品ID、数量**
- **属性のデータ型**：**数値、テキスト、数値、数値**
- データの整合性を保つための**制約**：**主キー制約、参照整合性制約**

```
CREATE TABLE 購入 (
  ID INTEGER PRIMARY KEY,
  購入者 TEXT,
  商品ID INTEGER,
  数量 INTEGER,
  FOREIGN KEY (商品ID) REFERENCES 商品(ID));
```

FOREIGN KEY … REFERENCES

PRIMARY KEY … REFERENCES はテーブル定義時に使用し，あるテーブルの**外部キー**が別のテーブルの**主キー**を**参照**する「**参照整合性制約**」を示す

```
CREATE TABLE 購入 (
  ID INTEGER PRIMARY KEY,
  購入者 TEXT,
  商品ID INTEGER,
  数量 INTEGER,
  FOREIGN KEY (商品ID) REFERENCES 商品(ID));
```

き方

外部キー

ID	購入者	商品ID	数量
1	X	1	10
2	Y	2	5

主キー

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

主キー

多対多の関連の例

１人の学生が，複数の科目を受講する．１つの科目には複数の学生が参加する．

学生ID	科目ID	得点
1	1001	90
1	1002	100
2	1001	85
2	1002	90
2	1003	95

外部キー　外部キー

ID	学生名
1	徳川家康
2	豊臣秀吉

主キー

ID	科目名
1001	国語
1002	算数
1003	理科

主キー

データベース設計のプロセス

１．テーブル名の決定

例: 「従業員」、「顧客」、「商品」など、データの種類や目的に応じた名前。

２．属性の設定

例: 従業員テーブルの属性は「従業員ID」、「名前」、「住所」

３．データ型の選択

例: 「従業員ID」は整数型（INTEGER）、「名前」は文字列型(TEXT)。

４．制約の設定

例: 主キー制約、参照整合性制約など

５．索引（インデックス）の作成

例: 頻繁に検索される「従業員ID」や「名前」に索引を設定。

６．テーブル間の関係性

例: 「従業員」テーブルの「部署ID」が「部署」テーブルの「部署ID」を参照（外部キーとして）。

10-2. 演習

いまから演習で行うこと、注意点

- 次のテーブルを作成

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

商品

ID	購入者	商品ID	数量
1	X	1	10
2	Y	2	5

購入

【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

演習１．Access の SQL ビューを用いたテーブル定義とデータの追加

【トピックス】

- **SQLビューを開く**
- **SQL文の編集**
- **create table**
- **insert into**
- **primary key**
- **foreign key … references**
- **SQL文の実行**

演習

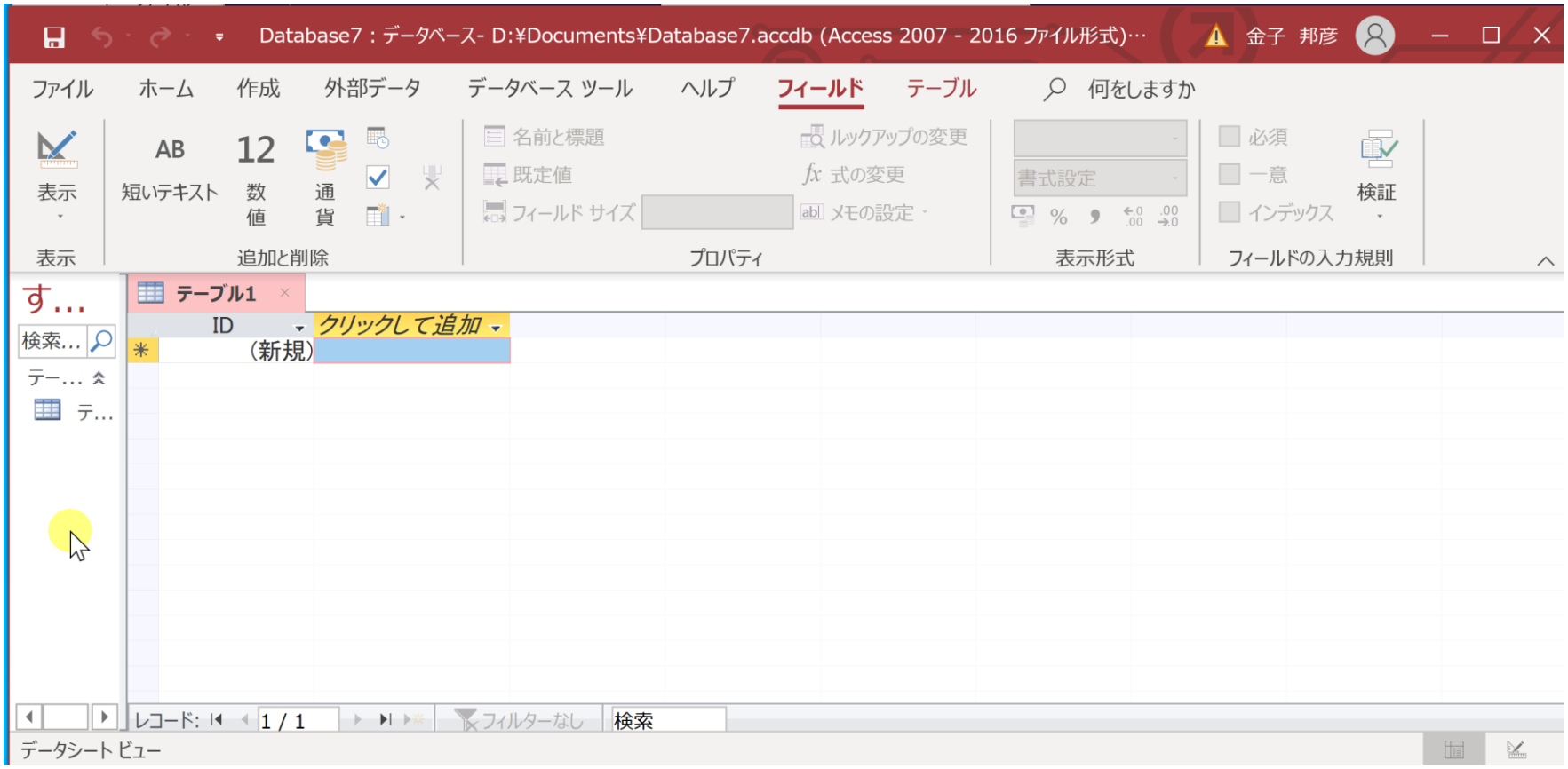
1. パソコンを使用する

   前もって Access をインストールしておくこと

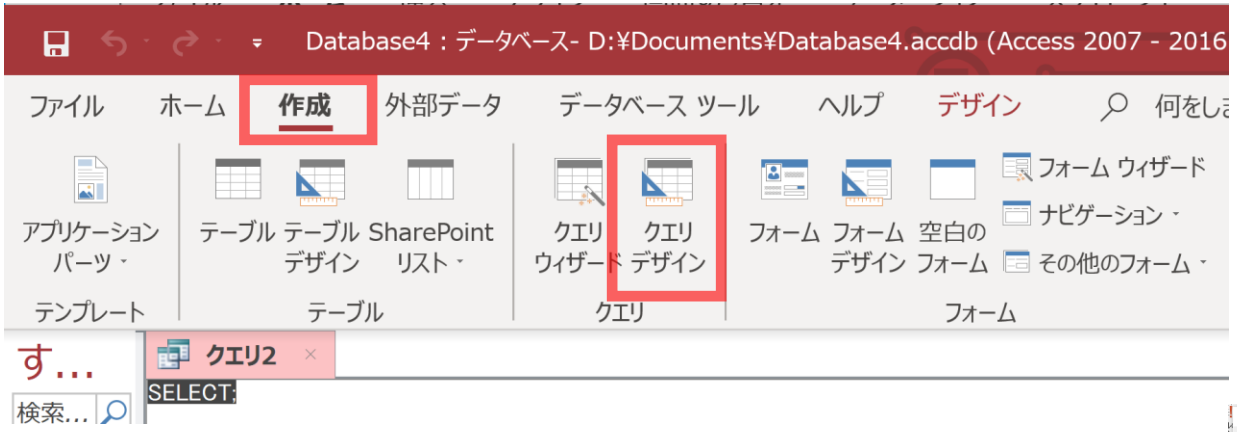
2. Access を起動する

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

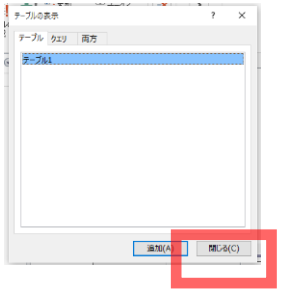
4. テーブルツール画面が表示されることを確認



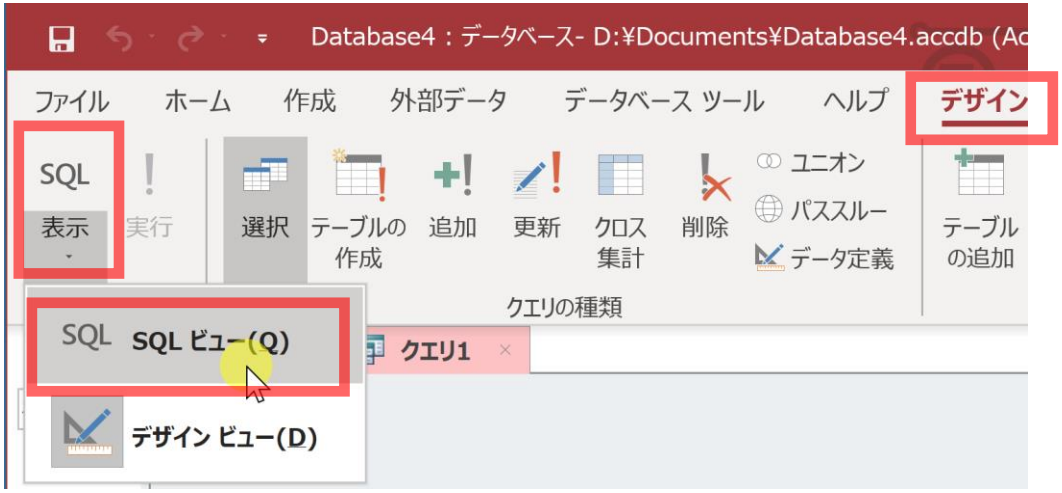
5. 次の手順で、**SQLビュー**を開く.



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

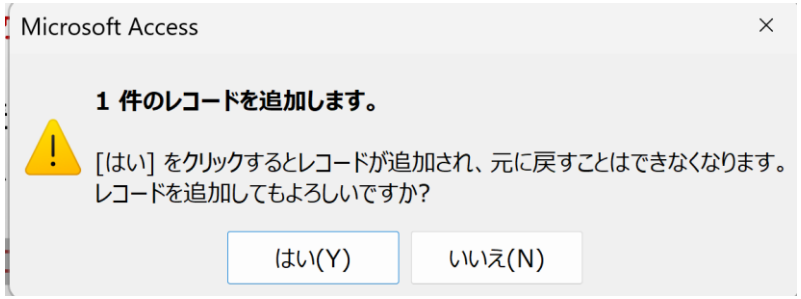


② 「**デザイン**」タブで、「**表示**」を展開し「**SQL ビュー**」を選ぶ

6. **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

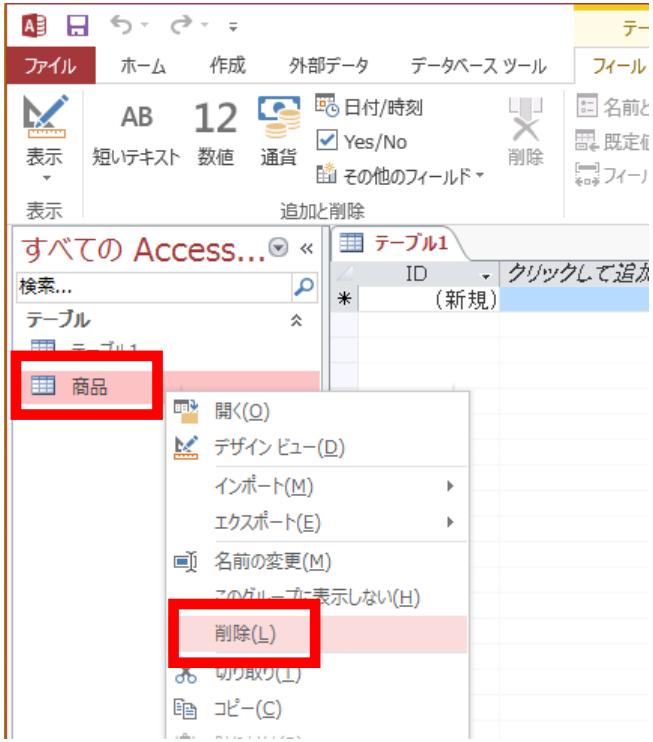
```
CREATE TABLE 商品 (
  ID INTEGER PRIMARY KEY,
  商品名 TEXT,
  単価 INTEGER);

CREATE TABLE 購入 (
 ID INTEGER PRIMARY KEY,
 購入者 TEXT,
 商品ID INTEGER,
 数量 INTEGER,
 FOREIGN KEY (商品ID) REFERENCES 商品(ID));

INSERT INTO 商品 VALUES(1, 'みかん', 50);

INSERT INTO 商品 VALUES(2, 'りんご', 100);

INSERT INTO 商品 VALUES(3, 'メロン', 500);

INSERT INTO 購入 VALUES(1, 'X', 1, 10);

INSERT INTO 購入 VALUES(2, 'Y', 2, 5);
```

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

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. **単純な表示**
2. **結合**

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

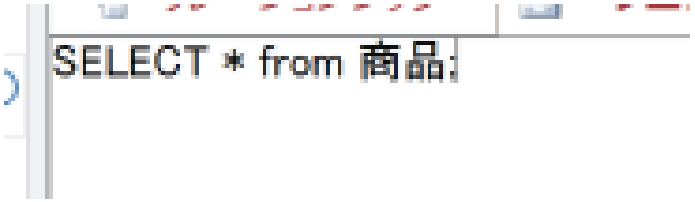
例: select * from テーブル名 where 列1 = 値1;

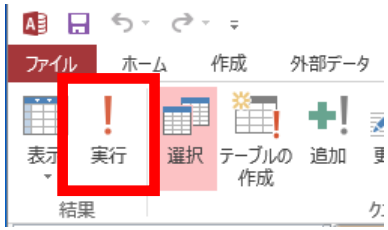
③ **SQL 文**の**実行**

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

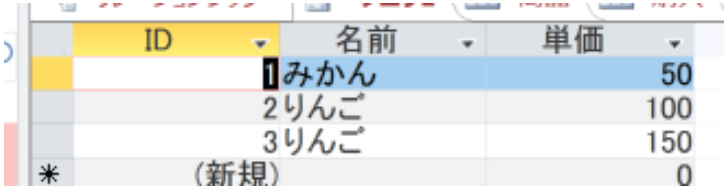
④ さらにSQL 文の編集、実行を続ける場合には、**画面を SQL ビューに切り替える**

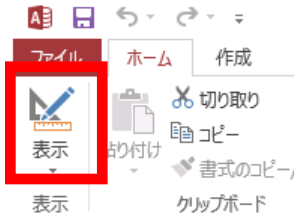
SQL 問い合わせ（クエリ）で使用する２つのビュー





実行





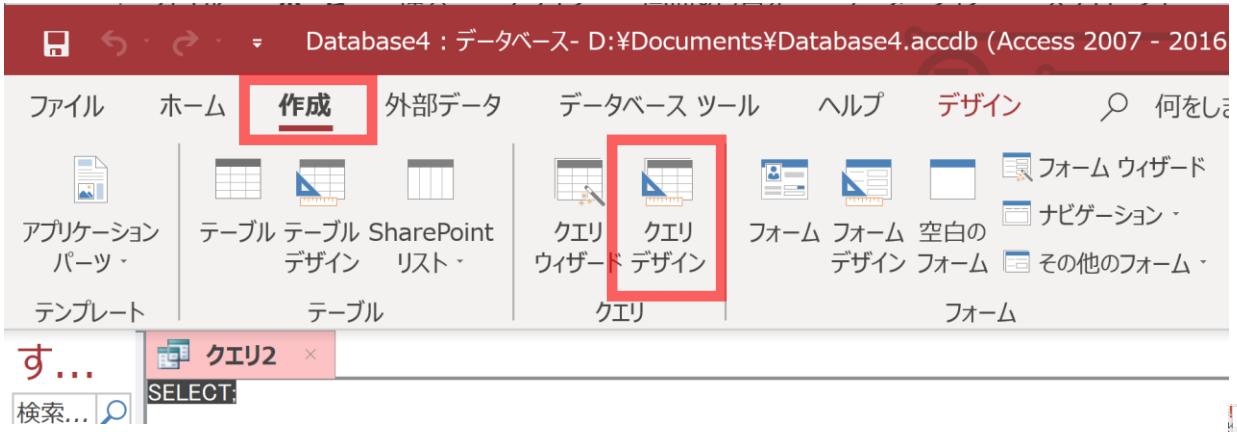
表示 + SQL ビュー

SQL ビュー
SQL 文の**作成**、**編集**

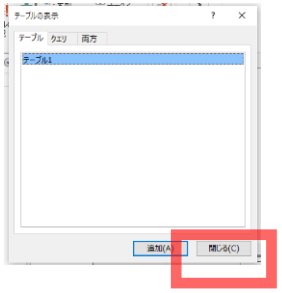
データシートビュー
問い合わせ（クエリ）の結果

マウス操作でビューを切り替え

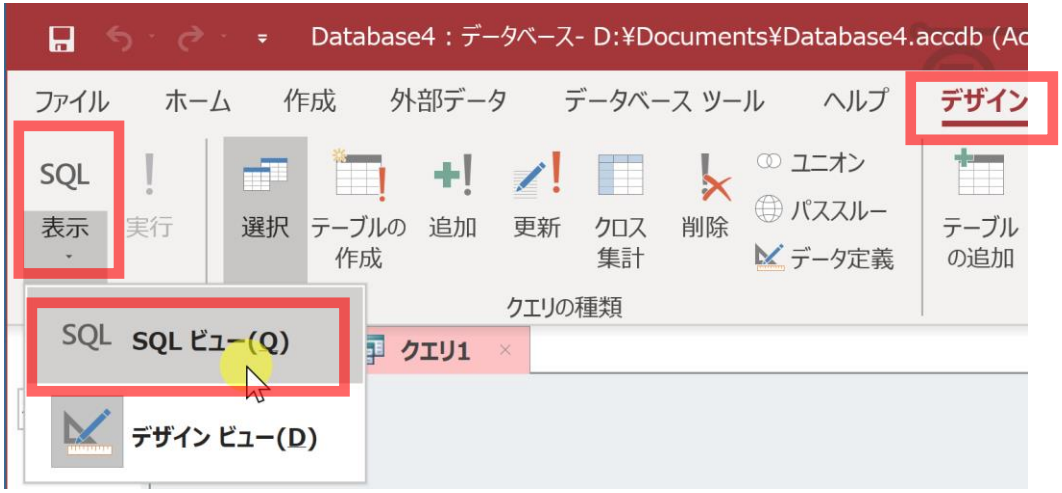
1. 次の手順で、**SQLビュー**を開く.



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

2. **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. 単純な表示

SELECT * FROM 商品;

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

2. 単純な表示

SELECT * FROM 購入;

ID	購入者	商品ID	数量
1	X	1	10
2	Y	2	5

(続き)

3.結合

```
SELECT 購入.購入者, 商品.商品名, 商品.単価 * 購入.数量
FROM 購入
INNER JOIN 商品 ON 購入.商品ID = 商品.ID;
```

購入者	商品名	Expr1002
X	みかん	500
Y	りんご	500

自習１．いまの２つのテーブルを結合して、購入者、商品名、数量、単価を次のように得る SQL を作成しなさい

購入者	商品名	数量	単価
X	みかん	10	50
Y	りんご	5	100

自習１の正解例

```
SELECT 購入.購入者, 商品.商品名, 数量, 単価
FROM 購入
INNER JOIN 商品 ON 購入.商品ID = 商品.ID;
```

データベース設計の実践例

- シナリオ

ある大学の学生、講義、および成績管理システム。

- 目的

学生の個人情報、登録講義、取得成績を効率的に管理。

「成績管理」のデータベース設計の実践例

シナリオ

ある大学の学生、講義、および成績管理システム。

目的

学生の個人情報、登録講義、取得成績を効率的に管理。

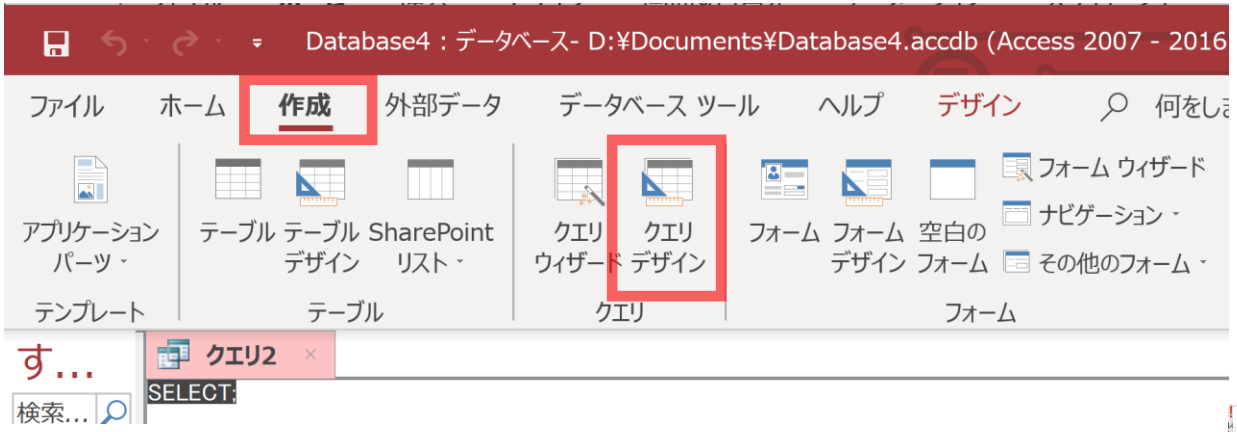
- **学生テーブル**:
  - 属性: 学生ID (主キー), 名前, 専攻
  - 主キー: 学生ID
- **講義テーブル**:
  - 属性: 講義ID (主キー), 講義名, 担当教員
  - 主キー: 講義ID
- **成績テーブル**:
  - 属性: 学生ID (外部キー), 講義ID (外部キー), 成績
  - 外部キー: 学生ID (学生テーブル参照), 講義ID (講義テーブル参照)
  - 主キー: 学生IDと講義ID

演習３．成績管理のデータベース

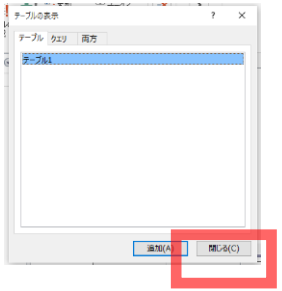
【トピックス】

- SQLビューを開く
- SQL文の編集
- create table
- insert into
- primary key
- foreign key … references
- SQL文の実行

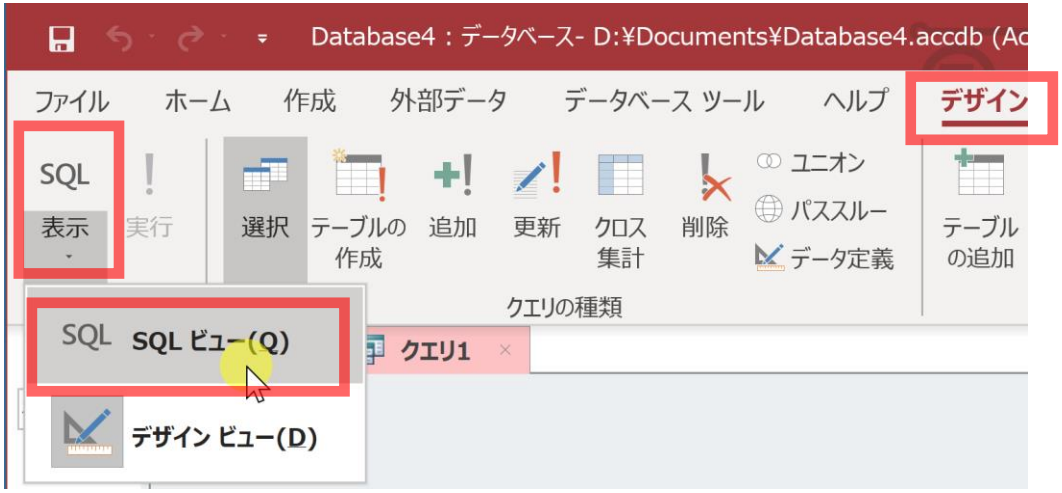
1. 次の手順で、**SQLビュー**を開く.



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

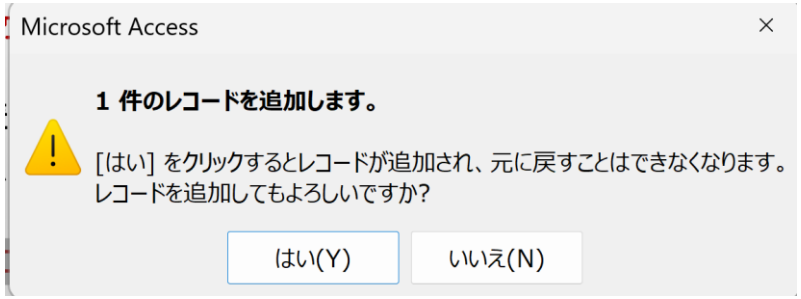
2. **SQL ビュー**に、次の SQL を１つずつ入れ、「**実行**」ボタンで、**SQL文**を実行．結果を確認

```
CREATE TABLE 学生 (
   学生ID INTEGER PRIMARY KEY,
   名前 TEXT,
   専攻 TEXT
);

CREATE TABLE 講義 (
   講義ID INTEGER PRIMARY KEY,
   講義名 TEXT,
   担当教員 TEXT
);

CREATE TABLE 成績 (
   学生ID INTEGER,
   講義ID INTEGER,
   成績 INTEGER,
   PRIMARY KEY (学生ID, 講義ID),
   FOREIGN KEY (学生ID) REFERENCES 学生(学生ID),
   FOREIGN KEY (講義ID) REFERENCES 講義(講義ID)
);
```

CREATE TABLE では、「実行」の後、**画面が変化しない**が実行できている

3. **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 学生 VALUES (1, '山田太郎', '情報科学');

INSERT INTO 学生 VALUES (2, '鈴木花子', '物理学');

INSERT INTO 学生 VALUES (3, '佐藤一郎', '化学');

INSERT INTO 講義 VALUES (101, 'プログラミング基礎', '田中健');

INSERT INTO 講義 VALUES (102, '物理学入門', '伊藤博');

INSERT INTO 講義 VALUES (103, '有機化学', '中村悟');

INSERT INTO 成績 VALUES (1, 101, 85);

INSERT INTO 成績 VALUES (1, 102, 90);

INSERT INTO 成績 VALUES (2, 101, 75);

INSERT INTO 成績 VALUES (2, 103, 80);

INSERT INTO 成績 VALUES (3, 103, 95);
```



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

演習４．種々のSQL問い合わせ. Access の SQL ビューを使用.

【トピックス】

1. **単純な表示**
2. **結合**

1. **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. 単純な表示

select * from 学生;

学生ID	名前	専攻
1	山田太郎	情報科学
2	鈴木花子	物理学
3	佐藤一郎	化学

2. 単純な表示

select * from 講義;

講義ID	講義名	担当教員
101	プログラミング	田中健
102	物理学入門	伊藤博
103	有機化学	中村悟

2. **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

3. 単純な表示

select * from 成績;

学生ID	講義ID	成績
1	101	85
1	102	90
2	101	75
2	103	80
3	103	95

4. 結合

select 学生.名前, 学生.専攻, 成績.講義ID, 成績.成績
from 学生 inner join 成績 on 学生.学生ID = 成績.学生ID;

名前	専攻	講義ID	成績
山田太郎	情報科学	101	85
山田太郎	情報科学	102	90
鈴木花子	物理学	101	75
鈴木花子	物理学	103	80
佐藤一郎	化学	103	95

自習２．3つのテーブルの結合

INNER JOIN … ON を２回書くことにより、３つのテーブルを結合できることを知る

次の SQL を Access で実行してください

SELECT 学生.名前, 学生.専攻, 講義.講義名, 講義.担当教員, 成績.成績
FROM (講義 INNER JOIN 成績 ON 講義.講義ID = 成績.講義ID)
INNER JOIN 学生 ON 成績.学生ID = 学生.学生ID;

名前	専攻	講義名	担当教員	成績
山田太郎	情報科学	プログラミング基	田中健	85
山田太郎	情報科学	物理学入門	伊藤博	90
鈴木花子	物理学	プログラミング基	田中健	75
鈴木花子	物理学	有機化学	中村悟	80
佐藤一郎	化学	有機化学	中村悟	95

※ Access は機能限定されています。SQL の世界標準ではもっと簡単な書き方があります。
SELECT 学生.名前, 学生.専攻, 講義.講義名, 講義.担当教員, 成績.成績
FROM 講義
INNER JOIN 成績 ON 講義.講義ID = 成績.講義ID
INNER JOIN 学生 ON 成績.学生ID = 学生.学生ID;

自習３．Access のリレーションシップウインドウの機能

次の手順で、リレーションシップウインドウを開き、テーブル間の関連が視覚的に表示されることを確認

１．**メニューバー**で「**データベースツール**」を選択

２．「**リレーションシップ**」をクリック

テーブル名と属性
　四角の中に表示

主キー：
　鍵マークで表示

外部キーが参照する別のテーブルの主キー
　テーブル間の線

全体まとめ

- **リレーショナルデータベースの仕組み**

データはテーブルと呼ばれる表形式で保存され、テーブル間は関連で結ばれる。

- **主キー (PRIMARY KEY)**

テーブル内の各行を一意に識別するためのキー。例えば、商品テーブルでは各商品に一意のIDが割り当てられる。

- **外部キー (FOREIGN KEY)**

他のテーブルの主キーを参照する。例えば、購入テーブルの「商品ID」は商品テーブルの主キー「ID」を参照する。

- **データベース設計のプロセス**

テーブル名の決定、属性の設定、データ型の選択、制約の設定、索引の作成、テーブル間の関係性の設定などが含まれる。

① SQLスキルの向上

データベース設計の学習は、SQLスキルの向上に貢献します。主キーと外部キーの設定、テーブルの作成、データの追加、問い合わせ（クエリ）などを反復練習することで、SQLの基本的な文法と構造を深く理解することができます。このスキルは、データベース管理やデータ分析の分野で重要です。

② データベース運用スキル

データベース設計の学習は、データベース運用スキルにつながります。テーブル間の関係性の理解、データの整合性の維持などを理解することにより、データベースの設計から実装、メンテナンスに至るまでの全プロセスに関わる運用スキルを身につけることができます。

③ 問題解決能力と論理的思考力

データベースの設計プロセスでは、複数の要素を考慮し、最適な構造を決定する必要があります。このプロセスは、問題に対する解決策を考え、論理的に推論する能力に深くかかわります。