de-11. データベース操作：INSERT、UPDATE、DELETE

（データベース演習）

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

金子邦彦

BY NC SA

11-1. 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に。

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

リレーショナルデータベースの重要性

1. **データの整合性**: リレーショナルデータベースは、**データ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。
3. **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。

データベース操作の種類

INSERT（追加）

- テーブルに新しい行を追加する操作

SELECT（問い合わせ）

- 必要なデータを検索・加工する操作

DELETE（削除）

- テーブルから条件に合致する行をすべて削除する操作

UPDATE（更新）

- 条件に合致する部分について，値を変更する操作

普通のプログラミングでのデータ更新とデータベース操作の違い

	データベース操作	普通のプログラミング
①トランザクション管理	**可能**	困難
②データベースの整合性を保つ	**制約機能あり**	機能なし
③複数ユーザからの同時アクセス	**システム側で同時アクセス対応**	原則，ユーザのプログラミングで制御
④データの永続性	**あり**	実行中のみデータ保持

11-2. 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

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

【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

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**
- **SQL文の実行**

演習

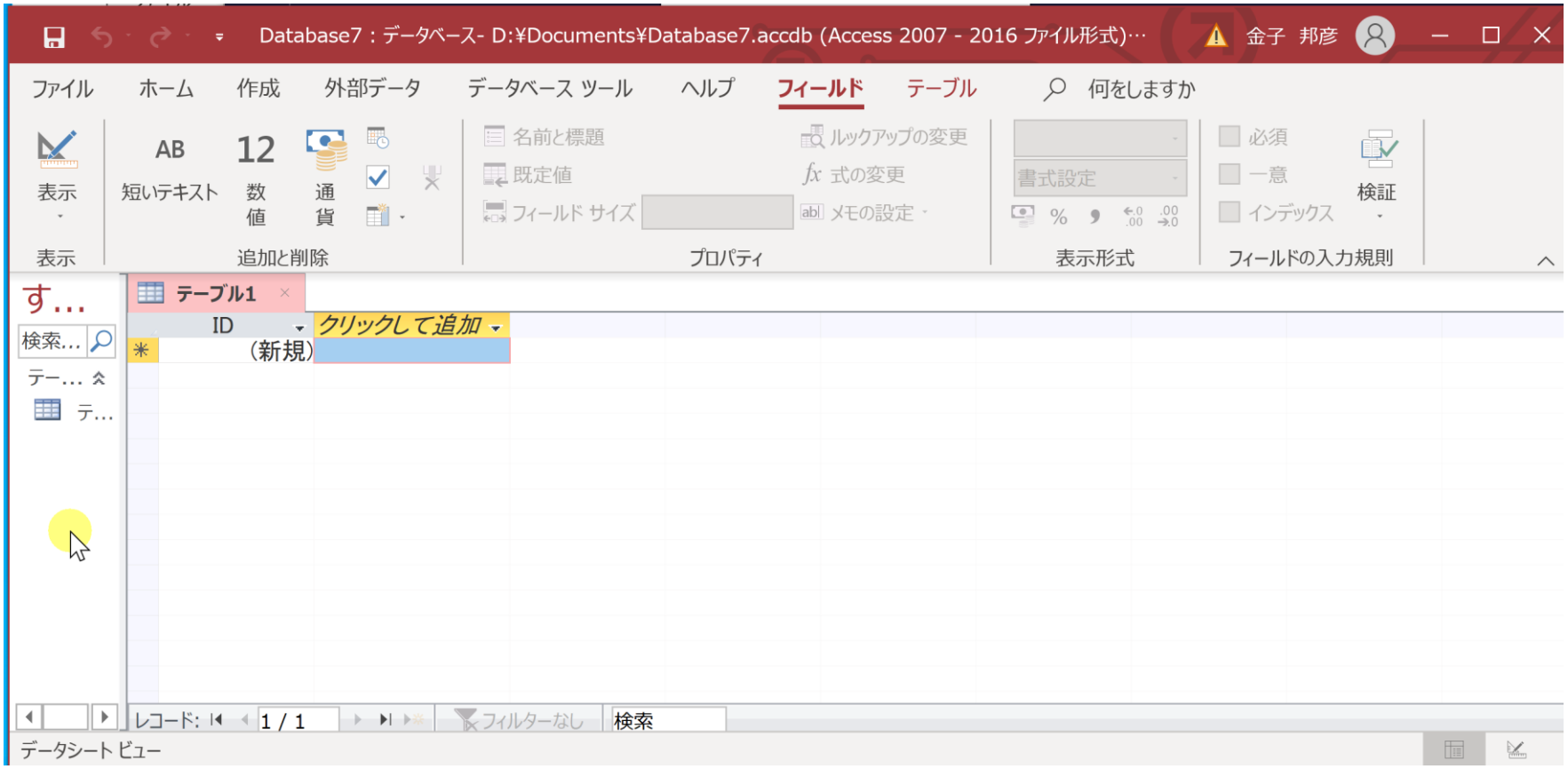
1. パソコンを使用する

   前もって Access をインストールしておくこと

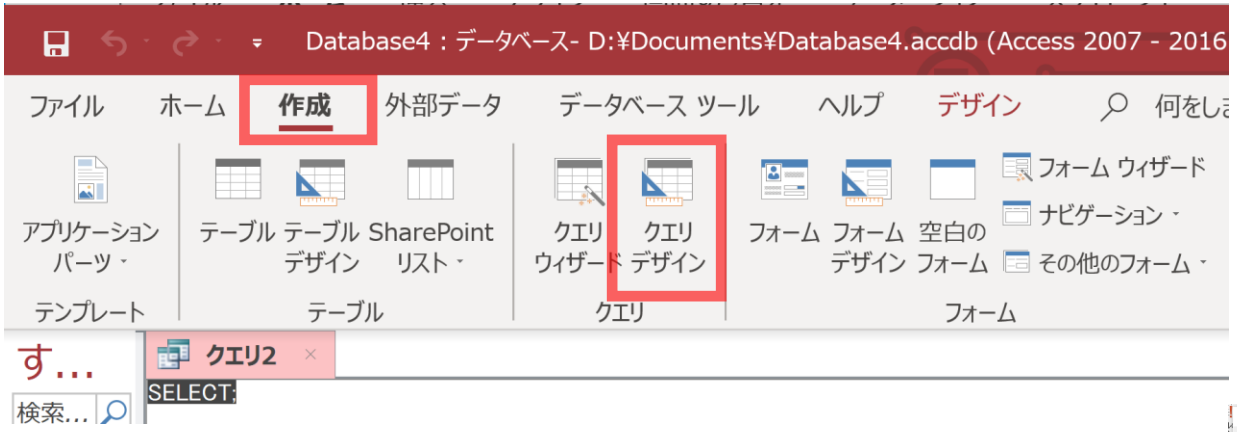
2. Access を起動する

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

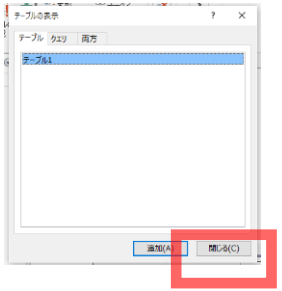
4. **テーブルツール画面**が表示されることを確認



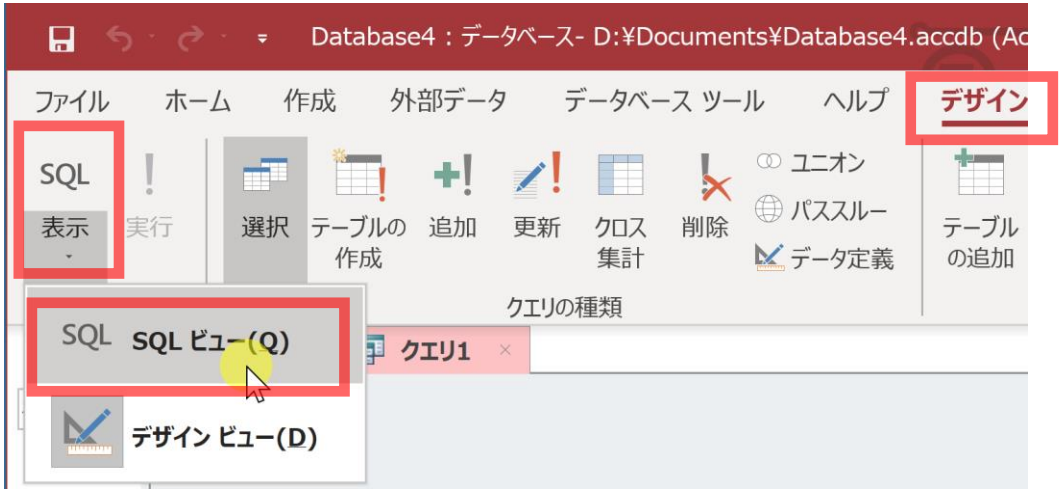
5. 次の手順で、**SQLビュー**を開く.



① 「**作成**」タブで、「**クエリデザイン**」をクリック



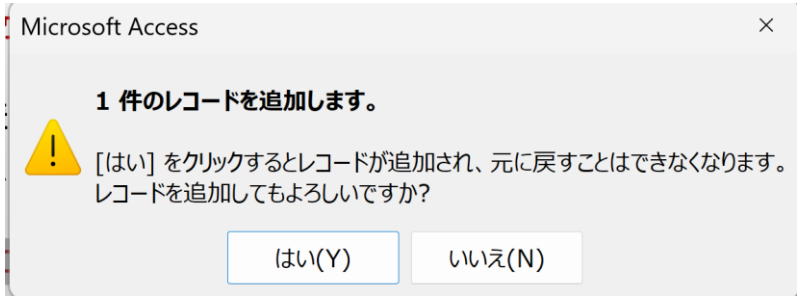
このような表示が出たときは「**閉じる**」をクリック



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

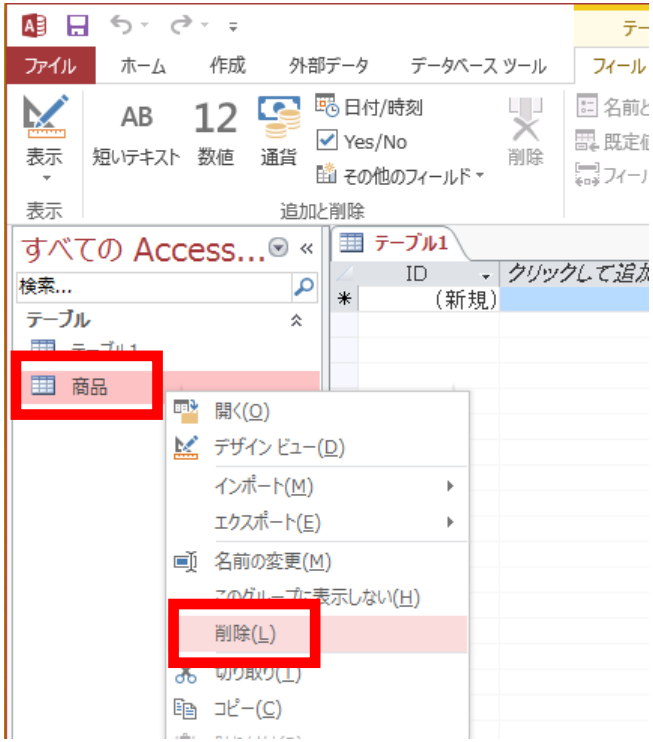
6. **SQL ビュー**に、次の SQL を１つずつ入れ、「**実行**」ボタンで、**SQL文**を実行．結果を確認

```
CREATE TABLE T (
名前 TEXT,
昼食 TEXT,
料金 INTEGER);

INSERT INTO T VALUES('A', 'そば', 250);

INSERT INTO T VALUES('B', 'カレーライス', 400);

INSERT INTO T VALUES('C', 'カレーライス', 400);

INSERT INTO T VALUES('D', 'うどん', 250);
```



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）

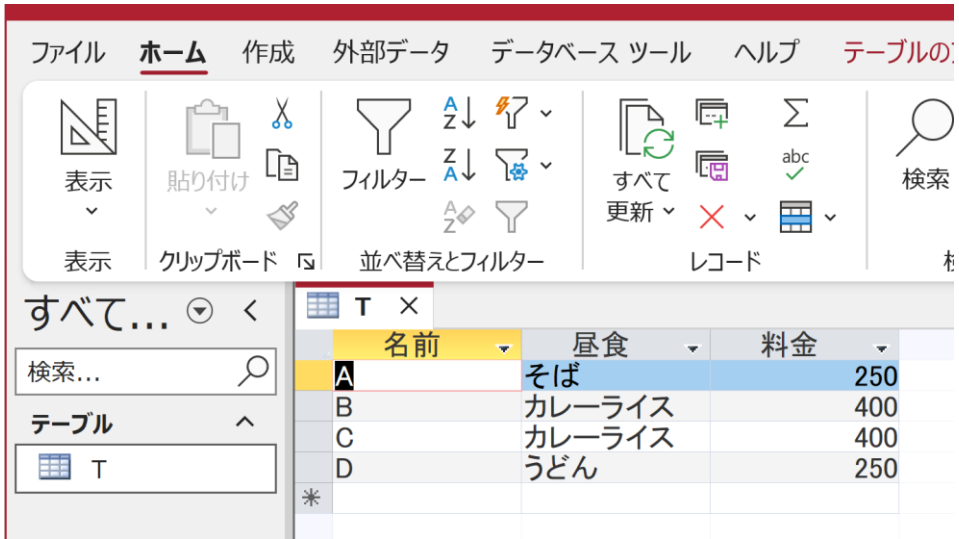
Access でのテーブルデータの確認

- 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**



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. SELECT

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

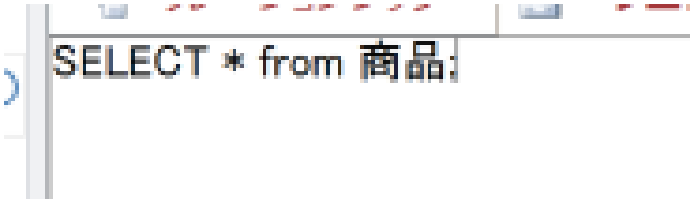
例: select * from テーブル名 where 列1 = 値1;

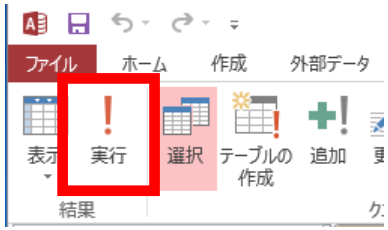
③ **SQL 文**の**実行**

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

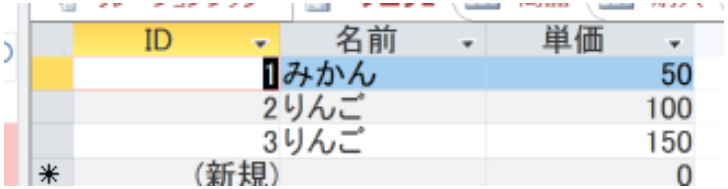
④ さらにSQL 文の編集、実行を続ける場合には、**画面を SQL ビューに切り替える**

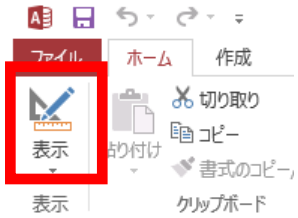
SQL 問い合わせ（クエリ）で使用する２つのビュー





実行





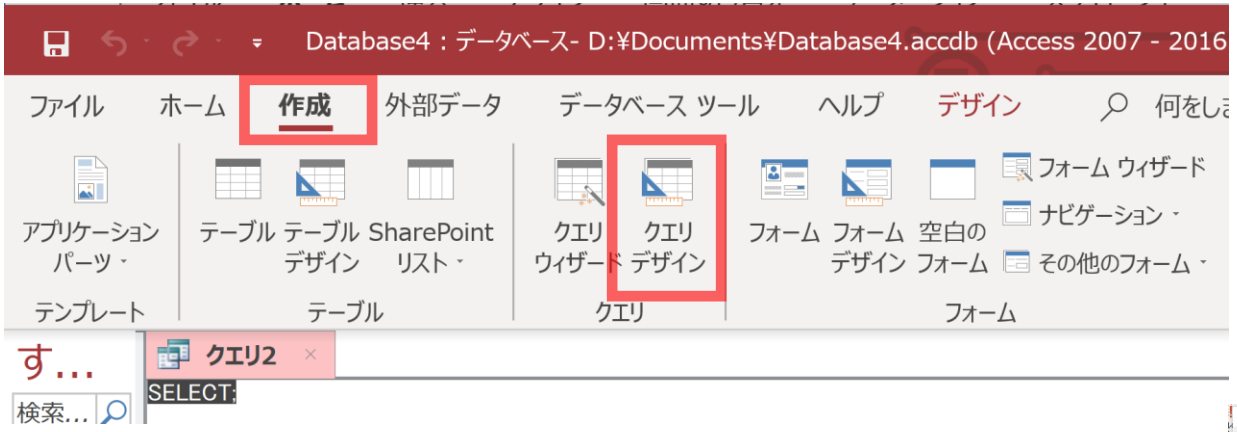
表示 + SQL ビュー

SQL ビュー
SQL 文の**作成**、**編集**

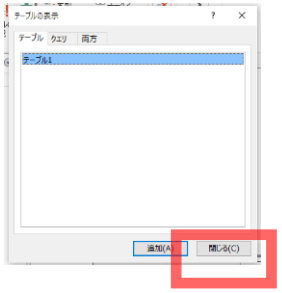
データシートビュー
問い合わせ（クエリ）の結果

マウス操作でビューを切り替え

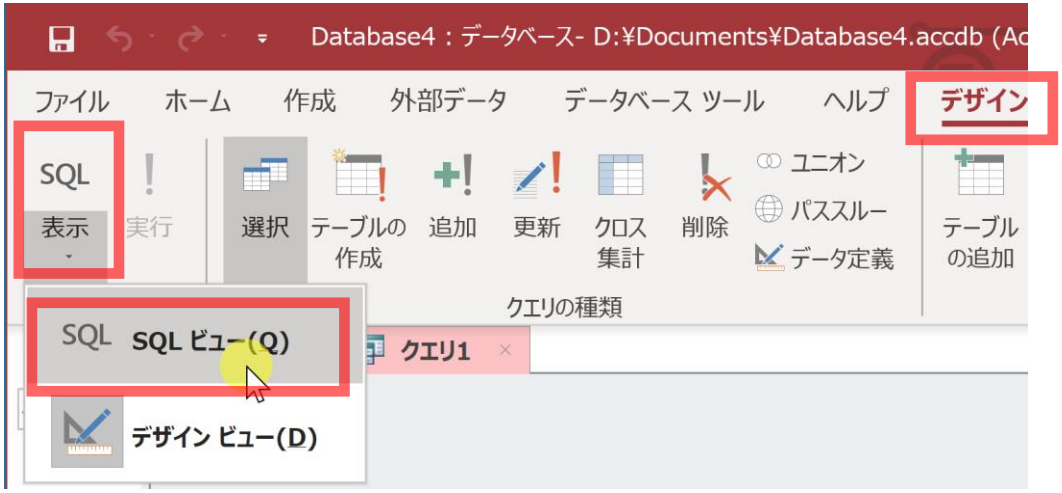
1. 次の手順で、**SQLビュー**を開く.



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

単純な表示

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
＊		

演習３． INSERT, DELETE, UPDATE

【トピックス】

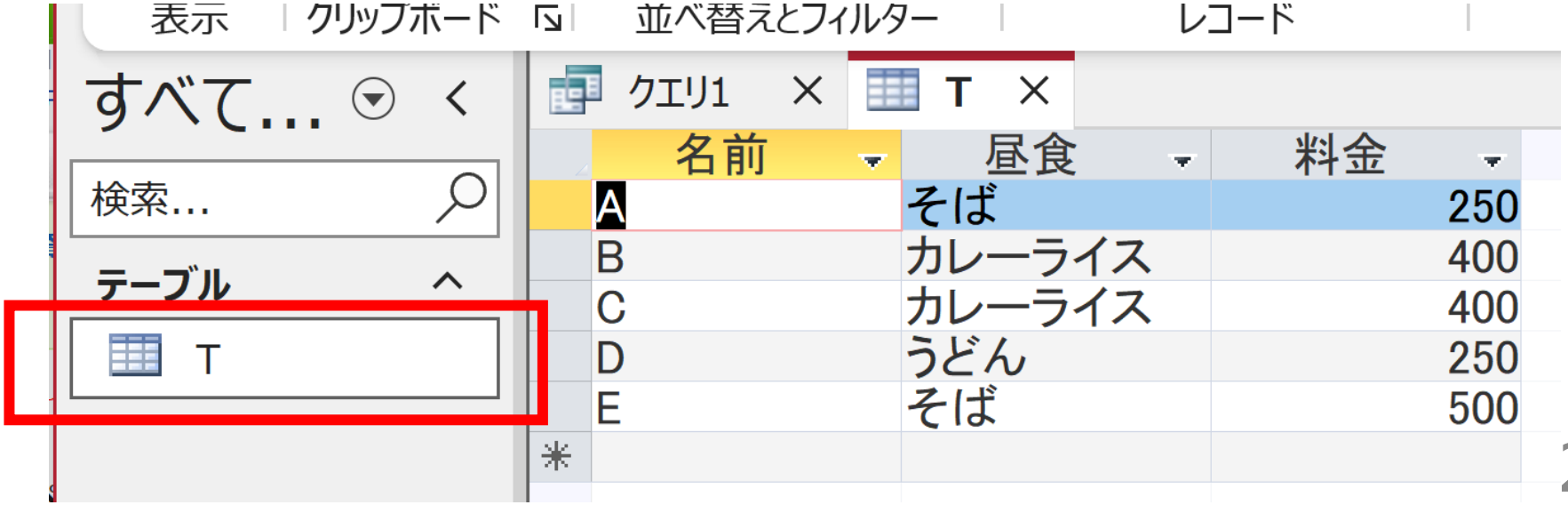
1. 追加: INSERT
2. 削除: DELETE
3. 更新: UPDATE

2. **SQL ビュー**に、次の SQL を入れ、「**実行**」ボタンで、**SQL文**を実行.

新しい行をTテーブルに追加し、名前は「E」、昼食は「天ぷら」、料金は500円

INSERT INTO T VALUES ('E', 'そば', 500);

そして、テーブル **T** をダブルクリックして、結果を確認

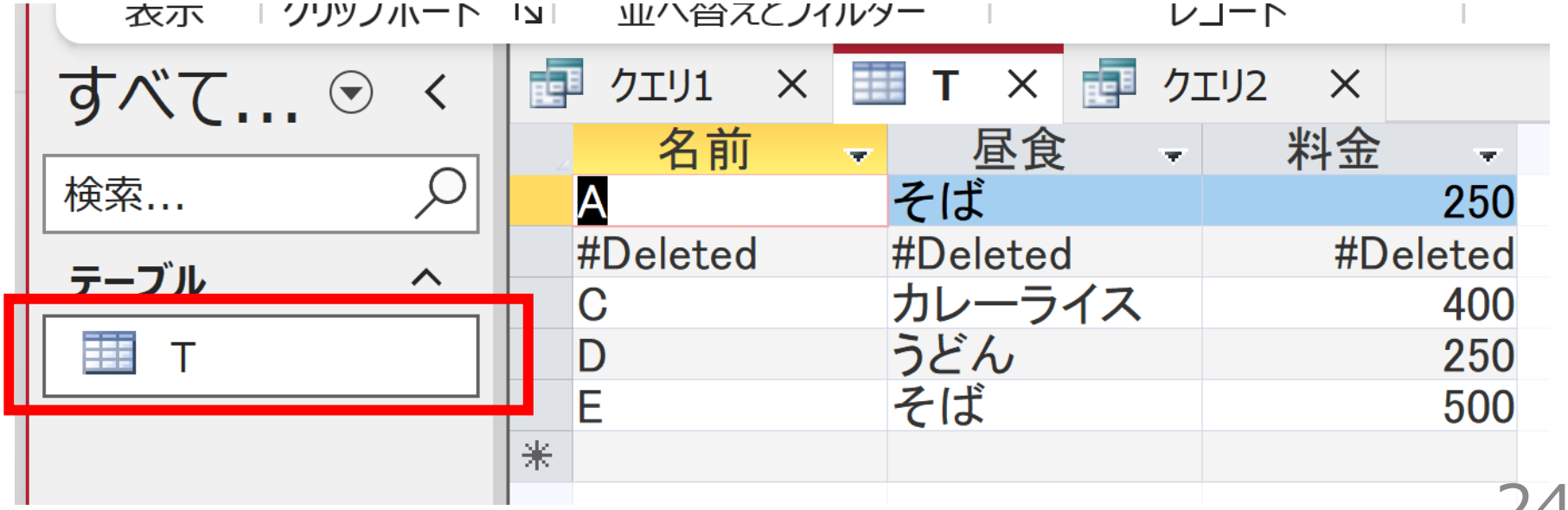


3. **SQL ビュー**に、次の SQL を入れ、「**実行**」ボタンで、**SQL文**を実行.

名前が'B'の行を削除

DELETE FROM T WHERE 名前 = 'B';

そして、テーブル **T** をダブルクリックして、結果を確認

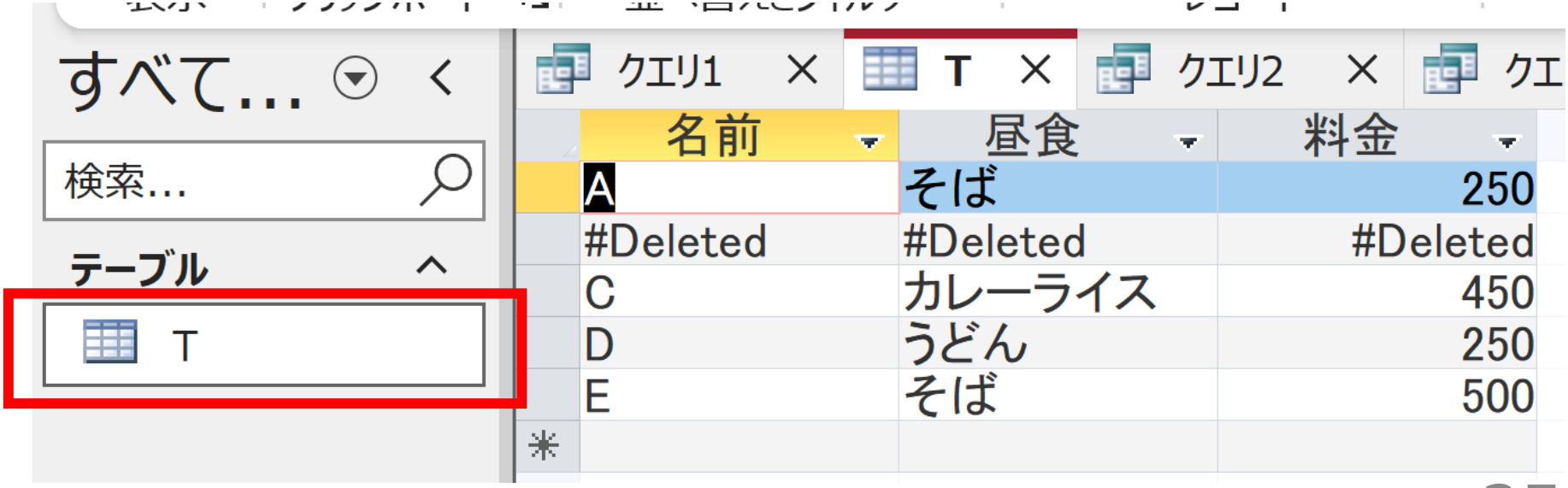


4. **SQL ビュー**に、次の SQL を入れ、「**実行**」ボタンで、**SQL 文**を実行.

「カレーライス」の料金を450円に変更

UPDATE T SET 料金 = 450 WHERE 昼食 = 'カレーライス';

そして、テーブル **T** をダブルクリックして、結果を確認

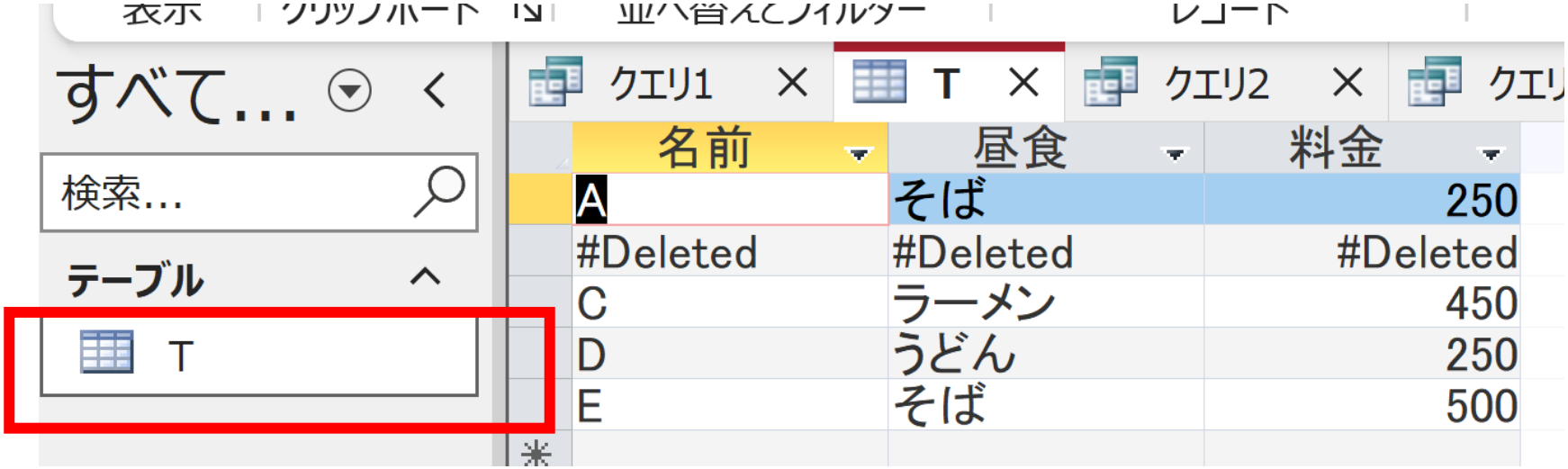


5. **SQL ビュー**に、次の SQL を入れ、「**実行**」ボタンで、**SQL文**を実行.

名前が'C'の人の昼食を「ラーメン」に変更

UPDATE T SET 昼食 = 'ラーメン' WHERE 名前 = 'C';

そして、テーブル **T** をダブルクリックして、結果を確認



11-2. 発展演習

発展演習１．**基本的なデータベース操作の練習**

- 目的：SQLを使用してテーブル定義、追加を行い、データベース操作を反復練習する

```
CREATE TABLE products (
    ID INTEGER PRIMARY KEY,
    name TEXT,
    color TEXT,
    price INTEGER
);
insert into products values(1, 'apple', 'red', 100);
insert into products values(2, 'apple', 'yellow', 200);
insert into products values(3, 'orange', 'orange', 300);
select * from products;
```

ID	name	color	price
1	apple	red	100
2	apple	yellow	200
3	orange	orange	300

発展演習２．**データの更新**

- 目的: データを更新を反復練習する

SQL を用いて products テーブル内の赤いリンゴの価格を 150 に更新してください。

- ヒント: UPDATE を使用し、name が 'apple' と color が 'red' の行を対象にする

ID	name	color	price	クリックして追加
1	apple	red	150	
2	apple	yellow	200	
3	orange	orange	300	

発展演習３．データの削除

- 目的: 条件に基づいてテーブルからデータを削除する練習

SQL を用いて products テーブルから価格が 300 以上の全てのレコードを削除してください。

- ヒント: DELETE FROM を使用し、条件で price >= 300 を指定

ID	name	color	price	クリックして追加
1	apple	red	150	
2	apple	yellow	200	
#Deleted	#Deleted	#Deleted	#Deleted	

解答例

発展演習２.

UPDATE products SET price = 150 WHERE name = 'apple' AND color = 'red';

発展演習３.

DELETE FROM products WHERE price >= 300;

全体まとめ

データベース操作

- 追加：INSERT INTO テーブル名 VALUES (値1, 値2, ...);
- 更新：UPDATE テーブル名 SET 列名 = 新しい値 WHERE 条件;
- 削除：DELETE FROM テーブル名 WHERE 条件;

実践的な操作例

- 新しいデータを挿入

INSERT INTO products VALUES(1, 'apple', 'red', 100);

- 特定データの更新

UPDATE T SET 昼食 = 'ラーメン' WHERE 名前 = 'C';

- 条件に基づくデータの削除

DELETE FROM products WHERE price >= 300;

① SQLスキルの向上

実践的なSQLコマンドの使用を通じて、データベース操作の基本を習得します。データ分析、データベースの管理といった幅広い分野で活用できます。

② データベース運用スキル

データベースの操作方法を学ぶことで、学生は実際の業務シナリオで適用可能な運用スキルを習得します。テーブルの作成からデータの管理までを理解し、その運用を通じてビジネスに貢献する能力が身につきます。これは、将来的にデータベース管理者やシステム開発者としてのキャリアに直接つながる実践的なスキルです。

③ 問題解決能力と論理的思考力

SQLコマンドの適用を学ぶことは、問題解決能力と論理的思考力を同時に養うことにつながります。論理的な思考を強化し、与えられた問題に対して最適な解を導き出す問題解決力を鍛え上げます。これは、日常生活のあらゆる決定にも役立つ普遍的なスキルです。