

7. SQLによるデータ分析：GROUP BYを用いたグループ化と集約

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

金子邦彦



謝辞：この資料では「いらすとや」のイラストを使用しています

アドバイス

- アクティブな学習を实践しよう

SQLを学ぶ際に、**プログラムを変更した結果を実際に見る**ことも心がけましょう。実際のデータベース操作を通じて学習を深めます。

- 簡単なスタートから

初めはシンプルなものからスタートしましょう。反復練習しましょう。

- ステップ・バイ・ステップで応用に進む

SQLスキルを向上させるために、**少しずつ難易度を上げ**、今まで自分ができなかったことにも**チャレンジ**しましょう。

A blurred background image showing a group of people in a meeting or office setting. One person is seated in the foreground, facing away from the camera, while others are visible in the background, some looking at documents or screens.

アウトライン

1. イントロダクション
2. グループ化と集約
3. GROUP BY の基本
4. 演習

SQLFiddle のサイトにアクセス

Webブラウザを使用

1. ウェブブラウザを開く
2. アドレスバーにSQLFiddleのURLを入力

<http://sqlfiddle.com/>

3. **MySQL** を選ぶ

URLが分からないときは、Googleなどの**検索エンジン**を利用。
「**SQLFiddle**」と**検索**し、表示された結果からSQLFiddleの
ウェブサイトをクリック。

SQLFiddle でのデータベース管理システムの選択

SQL Fiddle

Welcome to SQL Fiddle, an online SQL compiler that lets you write, edit, and execute any SQL query.

Choose which SQL language you would like to practice today:

[SQL Server](#)

[SQLite](#)

[PostgreSQL](#)

[MySQL](#)

[MariaDB](#)

[Oracle](#)


[Oracle PLSQL](#)

データベース管理システムの選択
(この授業では **MySQL** を使用)

SQLFiddle の画面

上のパネル: SQLの入力 (複数可能)

- ・ テーブル定義 CREATE TABLE
- ・ データの追加 INSERT INTO
- ・ SQL問い合わせ。SELECT, FROM, WHERE など



```
1 -- INIT database
2 CREATE TABLE Product (
3   ProductID INT AUTO_INCREMENT KEY,
4   Name VARCHAR(100),
5   Description VARCHAR(255)
6 );
7
8 INSERT INTO Product(Name, Description) VALUES ('Entity Framework Extensions', 'Use
9 INSERT INTO Product(Name, Description) VALUES ('Dapper Plus', 'Use <a href="https
10 INSERT INTO Product(Name, Description) VALUES ('C# Eval Expression', 'Use <a href
11
12 -- QUERY database
13 SELECT * FROM Product;
14 SELECT * FROM Product WHERE ProductID = 1;
15
```

実行ボタン


Execute

< Share

MySQL

結果ウィンドウ

Results



ProductID	Name	Description
1	Entity Framework Extensions	Use Entity Framework Extensions to extend your DbContext with high-performance bulk operations.
2	Dapper Plus	Use Dapper Plus to extend your IDbConnection with high-performance bulk operations.

7-1. 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

購入

購入者	商品番号
X	1
X	3
Y	2

関連

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

SQL 理解のための前提知識

○ テーブル

データを**テーブル**と呼ばれる**表形式**で保存

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

購入者	商品番号
X	1
X	3
Y	2

○ 問い合わせ（クエリ）

- **問い合わせ（クエリ）**は、**データベース**から必要なデータ
を検索、加工するための指令
- SELECT, FROM, WHERE など、**多様**なコマンドが存在。
- **結合、集計、ソート、副問い合わせ**など、高度な操作も可能

SQL によるテーブル定義

- テーブル名 : **成績**
- 属性名 : **科目、受講者、得点**
- 属性のデータ型 : **テキスト、テキスト、数値**
- データの整合性を保つための制約 : **なし**

```
CREATE TABLE 成績 (  
    科目 TEXT,  
    受講者 TEXT,  
    得点 INTEGER) ;
```

データ追加のSQL

成績

科目	受講者	得点
国語	A	85
国語	B	90
算数	A	90
算数	B	96
理科	A	95

```
INSERT INTO 成績 VALUES ('国語', 'A', 85);
```

```
INSERT INTO 成績 VALUES ('国語', 'B', 90);
```

```
INSERT INTO 成績 VALUES ('算数', 'A', 90);
```

```
INSERT INTO 成績 VALUES ('算数', 'B', 96);
```

```
INSERT INTO 成績 VALUES ('理科', 'A', 95);
```



演習 1. テーブル定義とデータの追加

【トピックス】

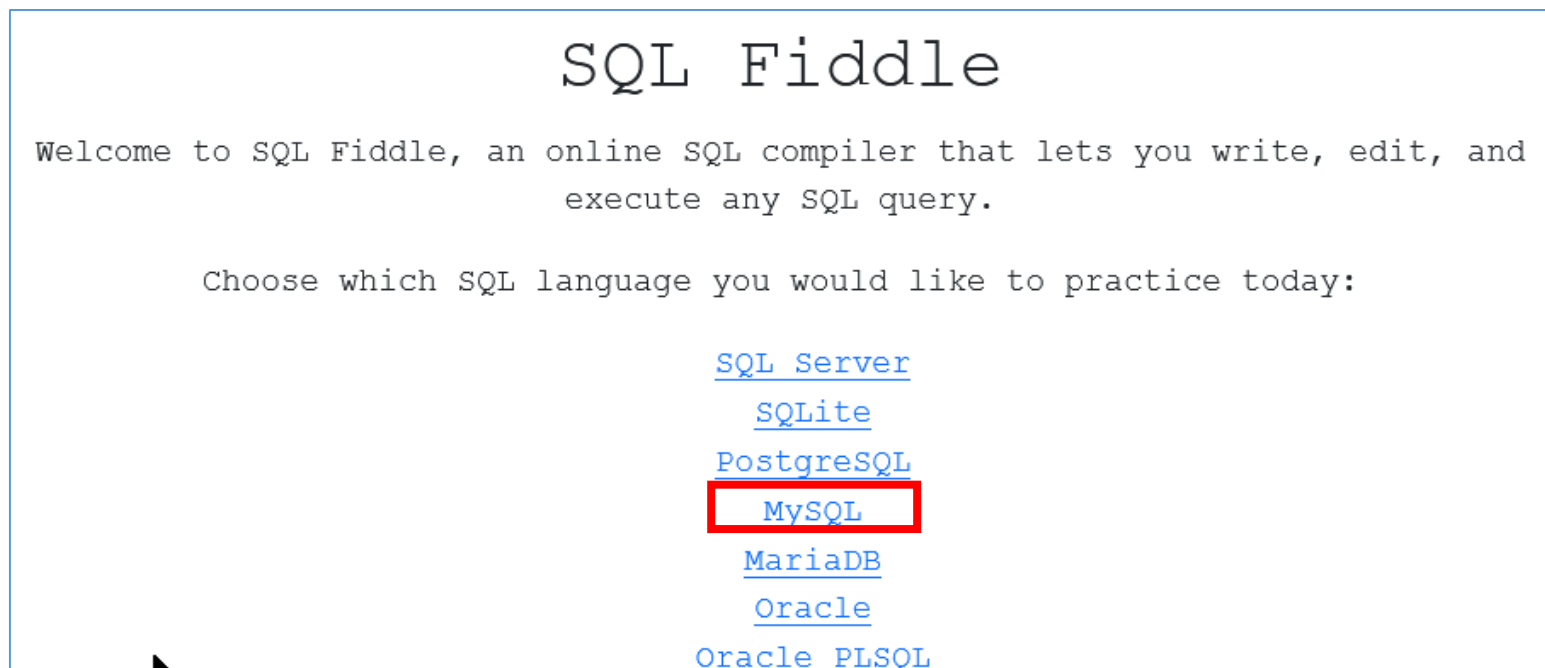
1. SQL によるテーブル定義
2. SQL によるデータの追加
3. 問い合わせ（クエリ）による確認

Webブラウザを使用

① アドレスバーにSQLFiddleのURLを入力

<http://sqlfiddle.com/>

② 「MySQL」を選択



③ 上のパネルに、**テーブル定義とデータの追加と問い合わせ**を行う SQL を入れる。（SQLFiddleで、最初に出てくるSQLは不要なので消す）。

```
CREATE TABLE 成績 (  
    科目 TEXT,  
    受講者 TEXT,  
    得点 INTEGER);  
  
INSERT INTO 成績 VALUES ('国語', 'A', 85);  
INSERT INTO 成績 VALUES ('国語', 'B', 90);  
INSERT INTO 成績 VALUES ('算数', 'A', 90);  
INSERT INTO 成績 VALUES ('算数', 'B', 96);  
INSERT INTO 成績 VALUES ('理科', 'A', 95);  
select * FROM 成績;
```

④ 「**Execute**」をクリック

SQL 文が**実行**され、結果が表示される。

⑤ 下のパネルで、**結果を確認**。

+-----+-----+-----+			
科目	受講者	得点	
+-----+-----+-----+			
国語	A	85	
国語	B	90	
算数	A	90	
算数	B	96	
理科	A	95	
+-----+-----+-----+			

集約の方法

AVG, MAX, MIN, SUM : 平均値, 最大値, 最小値, 合計値を算出

COUNT : 行数を計算

記録テーブル

名前	得点	居室
徳川家康	85	1階
源義経	78	2階
西郷隆盛	90	3階
豊臣秀吉	82	1階
織田信長	75	2階

SELECT AVG(得点) FROM 記録;

結果: 82

SELECT MAX(得点) FROM 記録;

結果: 90

SELECT MIN(得点) FROM 記録;

結果: 75

SELECT SUM(得点) FROM 記録;

結果: 410

SELECT COUNT(*) FROM 記録;

結果: 5



演習 2. 集約

【トピックス】

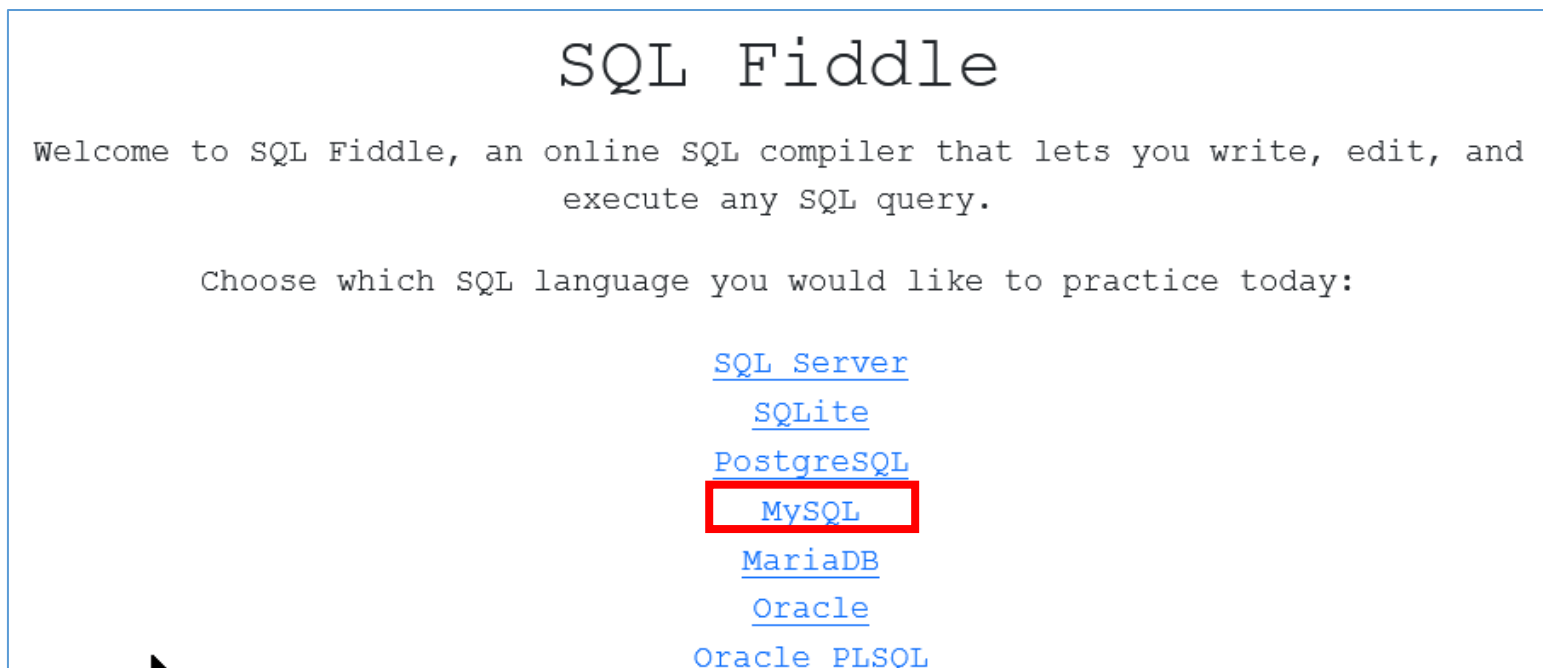
1. 平均 AVG
2. 最大 MAX
3. 最小 MIN
4. 合計 SUM
5. 行数 COUNT

Webブラウザを使用

① アドレスバーにSQLFiddleのURLを入力

<http://sqlfiddle.com/>

② 「MySQL」を選択



③ 上のパネルに、テーブル定義とデータの追加と問い合わせを行う SQL を入れ実行。（以前の SQL は不要なので消す）

```
CREATE TABLE 成績 (  
    科目 TEXT,  
    受講者 TEXT,  
    得点 INTEGER);  
  
INSERT INTO 成績 VALUES ('国語', 'A', 85);  
INSERT INTO 成績 VALUES ('国語', 'B', 90);  
INSERT INTO 成績 VALUES ('算数', 'A', 90);  
INSERT INTO 成績 VALUES ('算数', 'B', 96);  
INSERT INTO 成績 VALUES ('理科', 'A', 95);  
SELECT AVG(得点) FROM 成績 WHERE 科目 = '国語';
```

④ 「**Execute**」をクリック

SQL 文が**実行**され、結果が表示される。

⑤ 下のパネルで、**結果を確認**。

```
+-----+
|  AVG (得点)  |
+-----+
|  87.5000    |
+-----+
```

発展学習 1．算数の平均点の計算

目的：成績テーブルから**算数の平均得点**を算出する。

科目が算数の行について、得点の平均値を求めるSQL文を書いてください。

ヒント：AVGを使い、WHERE 句で科目を算数に絞り込みます。

発展学習 2. 科目は問わず全体の最高点の計算

目的：成績テーブルから**最高得点**を算出する。

得点の最大値を求めるSQL文を書いてください。

ヒント：MAXを使います。

発展学習 1 の解答例

```
SELECT AVG(得点) FROM 成績 WHERE 科目 = '算数';
```

AVG(得点)
93

発展学習 2 の解答例

```
SELECT MAX(得点) FROM 成績;
```

MAX(得点)
96

これらの発展学習は、基本的な集約関数の使い方と、WHERE 及び GROUP BY の使用方法を理解し、練習するのに役立つ

7-2. グループ化と集約

集約の方法

AVG, MAX, MIN, SUM : 平均値, 最大値, 最小値, 合計値を算出

COUNT : 行数を計算

記録テーブル

名前	得点	居室
徳川家康	85	1階
源義経	78	2階
西郷隆盛	90	3階
豊臣秀吉	82	1階
織田信長	75	2階

SELECT AVG(得点) FROM 記録;

結果: 82

SELECT MAX(得点) FROM 記録;

結果: 90

SELECT MIN(得点) FROM 記録;

結果: 75

SELECT SUM(得点) FROM 記録;

結果: 410

SELECT COUNT(*) FROM 記録;

結果: 5

グループ化

グループ化は、同じ属性値を共有するデータを集めるプロセス。

例：科目の「国語」、「算数」、「理科」でグループ化

科目	受講者	得点
国語	A	85
国語	B	90
算数	A	90
算数	B	96
理科	A	95



科目	受講者	得点
国語	A	85
国語	B	90

科目	受講者	得点
理科	A	95

科目	受講者	得点
算数	A	90
算数	B	96

例：受講者の「A」、「B」でグループ化

科目	受講者	得点
国語	A	85
国語	B	90
算数	A	90
算数	B	96
理科	A	95



科目	受講者	得点
国語	A	85
算数	A	90
理科	A	95

科目	受講者	得点
国語	B	90
算数	B	96

それぞれの値ごとにグループに分けることで、データの分析が容易になる

成績テーブルの「科目」によるグループ化

① 成績テーブルには科目、受講者、得点が記載されている

科目	受講者	得点
国語	A	85
国語	B	90
算数	A	90
算数	B	96
理科	A	95

②科目の「国語」、「算数」、「理科」のグループを形成

科目	受講者	得点
国語	A	85
国語	B	90

科目	受講者	得点
算数	A	90
算数	B	96

科目	受講者	得点
理科	A	95

データのグループ化を用いた集約

③ グループで、集約（行数、平均、合計など）を実施

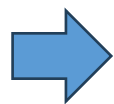
科目	受講者	得点
国語	A	85
国語	B	90
算数	A	90
算数	B	96
理科	A	95



科目	受講者	得点
国語	A	85
国語	B	90

科目	受講者	得点
理科	A	95

科目	受講者	得点
算数	A	90
算数	B	96



	国語	算数	理科
行数 :	2	2	1
平均 :	87.5	93	95
合計 :	175	186	95

グループ化と集約

- **グループ化**は、**同じ属性値を共有するデータを集めるプロセス**。

例：「科目」や「受講者」ごとにデータを分類できる

- こうして形成された**グループを集約（行数、平均値、合計値）**し、元のデータの概要を把握することが可能になる。

科目	受講者	得点
国語	A	85
国語	B	90
算数	A	90
算数	B	96
理科	A	95



科目	受講者	得点
国語	A	85
国語	B	90
科目	受講者	得点
算数	A	90
算数	B	96
科目	受講者	得点
理科	A	95



概要

得点の平均

国語	87.5
算数	93
理科	95

グループ化

集約

SQL における GROUP BY の基礎

- **GROUP BY** は、特定の属性（例：科目，受講者）を基準として、**グループ化**を行う
- **集約をGROUP BYと組み合わせることで，グループごとの集約結果を得ることができる**
- これにより，データの傾向を効率的に分析することが可能である．

GROUP BY の役割と書き方

- **SQL 問い合わせ「SELECT ...」の中で、GROUP BY を使用してデータをグループ化する**
- **1つ以上の属性を GROUP BY に指定してグループ化の基準とする。**

すべての科目ごとに、受講者の数を計算

SELECT 科目, COUNT(*) FROM 成績 GROUP BY 科目;

科目	COUNT(*)
国語	2
理科	1
算数	2

データ分析とビジネスインテリジェンス

主要な分析手法

- **グループ化（データをまとめること）と集約（まとめたデータの計算）**

行数, 平均, 合計などの生成

- **データ分析**

カテゴリ別, 時系列別（時間の経過に沿った変化）のデータ分析

- **ビジネスインテリジェンス（BI）**

売上のトレンド分析（傾向や変化を調べること）

顧客セグメント分析（顧客を年齢や購買傾向などで分類して分析すること）

集約の種類の変換

科目	受講者	得点
国語	A	85
国語	B	90
算数	A	90
算数	B	96
理科	A	95

元データ



A	270
B	186

得点の**合計**

A	3
B	2

行数

A	90
B	93

得点の**平均**

A	95
B	96

得点の**最大**

A	85
B	90

得点の**最小**

グループ化における基準のバリエーション

科目	受講者	得点
国語	A	85
国語	B	90
算数	A	90
算数	B	96
理科	A	95

元データ



A	90
B	93

得点の平均

「受講者」でグループ化

国語	2
算数	2
理科	1

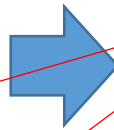
行数

「科目」でグループ化

「受講者」でグループ化

元データ

科目	受講者	得点
国語	A	85
国語	B	90
算数	A	90
算数	B	96
理科	A	95



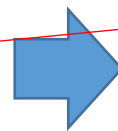
A	90
B	93

得点の平均

「科目」でグループ化

元データ

科目	受講者	得点
国語	A	85
国語	B	90
算数	A	90
算数	B	96
理科	A	95



国語	87.5
算数	93
理科	95

得点の平均

グループ化と集約のまとめ

グループ化

- ・ **グループ化**は、**同じ属性値を共有するデータを集める**プロセス。

例： 科目ごとにグループを作成（国語、算数、理科）
受講者ごとにグループを作成（A、B）。

グループ化と集約による分析

- ・ 各グループの行数、平均点、合計点などを算出してデータの概要を把握。

集約の種類

- ・ 平均 AVG、最大 MAX、最小 MIN、合計 SUM、行数 COUNT

データ分析への応用例

- ・ カテゴリ別、時系列別などでデータ分析を行う。
- ・ ビジネスインテリジェンスとして、売上のトレンド分析や顧客セグメント分析に利用。

7-3. GROUP BY の基本

GROUP BY の役割と書き方

- **SQL 問い合わせ「SELECT ...」の中で、GROUP BY を使用してデータをグループ化**
- **1つ以上の属性を GROUP BY に指定してグループ化の基準とする。**

例

SELECT で「科目」と「COUNT(*)」を指定し、GROUP BY で「科目」をグループ化の基準として設定

すべての科目ごとに、受講者の数を計算

SELECT 科目, COUNT(*) FROM 成績 GROUP BY 科目;

科目	COUNT(*)
国語	2
理科	1
算数	2



演習 3 . GROUP BY による グループ化と集約

【トピックス】

1. SELECT での集約の指定
2. GROUP BY での基準の指定

Webブラウザを使用

① アドレスバーにSQLFiddleのURLを入力

<http://sqlfiddle.com/>

② 「MySQL」を選択



The screenshot shows the SQL Fiddle website. At the top, it says "SQL Fiddle". Below that, a welcome message reads: "Welcome to SQL Fiddle, an online SQL compiler that lets you write, edit, and execute any SQL query." Underneath, it asks the user to "Choose which SQL language you would like to practice today:". A list of database engines is provided as links: "SQL Server", "SQLite", "PostgreSQL", "MySQL", "MariaDB", "Oracle", and "Oracle PLSQL". The "MySQL" link is highlighted with a red rectangular box.

```
SQL Fiddle
```

Welcome to SQL Fiddle, an online SQL compiler that lets you write, edit, and execute any SQL query.

Choose which SQL language you would like to practice today:

- [SQL Server](#)
- [SQLite](#)
- [PostgreSQL](#)
- [MySQL](#)
- [MariaDB](#)
- [Oracle](#)
- [Oracle PLSQL](#)

③ 上のパネルに、テーブル定義とデータの追加と問い合わせを行う SQL を入れ実行。（以前の SQL は不要なので消す）

```
CREATE TABLE 成績 (  
    科目 TEXT,  
    受講者 TEXT,  
    得点 INTEGER);  
  
INSERT INTO 成績 VALUES ('国語', 'A', 85);  
INSERT INTO 成績 VALUES ('国語', 'B', 90);  
INSERT INTO 成績 VALUES ('算数', 'A', 90);  
INSERT INTO 成績 VALUES ('算数', 'B', 96);  
INSERT INTO 成績 VALUES ('理科', 'A', 95);  
SELECT 科目, COUNT(*) FROM 成績 GROUP BY 科目;
```

科目ごとの受講者数

④ 「**Execute**」をクリック

SQL 文が**実行**され、結果が表示される。

⑤ 下のパネルで、**結果を確認**。

+-----+-----+	
科目	COUNT (*)
+-----+-----+	
国語	2
算数	2
理科	1
+-----+-----+	

⑥ 上のパネルに、テーブル定義とデータの追加と問い合わせを行う SQL を入れ実行。（以前の SQL は不要なので消す）

```
CREATE TABLE 成績 (  
    科目 TEXT,  
    受講者 TEXT,  
    得点 INTEGER) ;  
INSERT INTO 成績 VALUES ('国語', 'A', 85) ;  
INSERT INTO 成績 VALUES ('国語', 'B', 90) ;  
INSERT INTO 成績 VALUES ('算数', 'A', 90) ;  
INSERT INTO 成績 VALUES ('算数', 'B', 96) ;  
INSERT INTO 成績 VALUES ('理科', 'A', 95) ;  
SELECT 科目, COUNT(*) FROM 成績 WHERE 得点 >= 90 GROUP BY 科目;
```

科目ごとの、90点以上の成績の受講者数

⑦ 「**Execute**」をクリック

SQL 文が**実行**され、結果が表示される。

⑧ 下のパネルで、**結果を確認**。

+-----+-----+	
科目 COUNT (*)	
+-----+-----+	
国語 1	
算数 2	
理科 1	
+-----+-----+	

発展学習 3. 受講者ごとの科目数

目的：SELECT, FROM の中での COUNT(*), GROUP BY の使用に上達する

次の SQL を実行し結果を確認

```
SELECT 受講者, COUNT(*) FROM 成績 GROUP BY 受講者;
```

発展学習 4. 科目ごとの平均点

目的：SELECT, FROM の中での AVG, GROUP BY の使用に上達する

次の SQL を実行し結果を確認

```
SELECT 科目, AVG(得点) FROM 成績 GROUP BY 科目;
```

AVG(得点) のように書いていることに注意

発展学習 5. 受講者ごとの平均点

目的： GROUP BY の使用に上達する

受講者ごとの平均点を求める SQL 文はどうなるか？

結果として、次の結果を得たい。

受講者	AVG(得点)
A	90
B	93

ヒント： AVG, GROUP BY を使用する。発展学習 4 を参考に。

発展学習 3

受講者	COUNT(*)
A	3
B	2

発展学習 4

科目	AVG(得点)
国語	87.5
理科	95
算数	93

発展学習 5

```
SELECT 受講者, AVG(得点) FROM 成績 GROUP BY 受講者;
```

GROUP BY のまとめ

グループ化の概念

- GROUP BYは、指定した属性（例：「科目」、「受講者」）に基づいてデータをグループ化する。

集約の利用

- 集約（AVG, MAX, MIN, SUM, COUNT）を使って、大量のデータから意味のある情報を抽出する。

グループ化と集約の組み合わせ

- GROUP BYを集約関数と組み合わせることで、グループ毎の統計を計算する

7-4. 演習

今からの演習で行うこと

① 売上を記録するためのテーブルを作成

- 架空の売上データを追加
- 日付、商品、個数、単価の4つの属性を持つ

② 売り上げトレンドの分析

- このテーブルに対して**GROUP BY**を使って、**日付や商品を基準としてグループ化**
- **SUMで売上合計を計算**



演習 4 . GROUP BY による グループ化と集約

【トピックス】

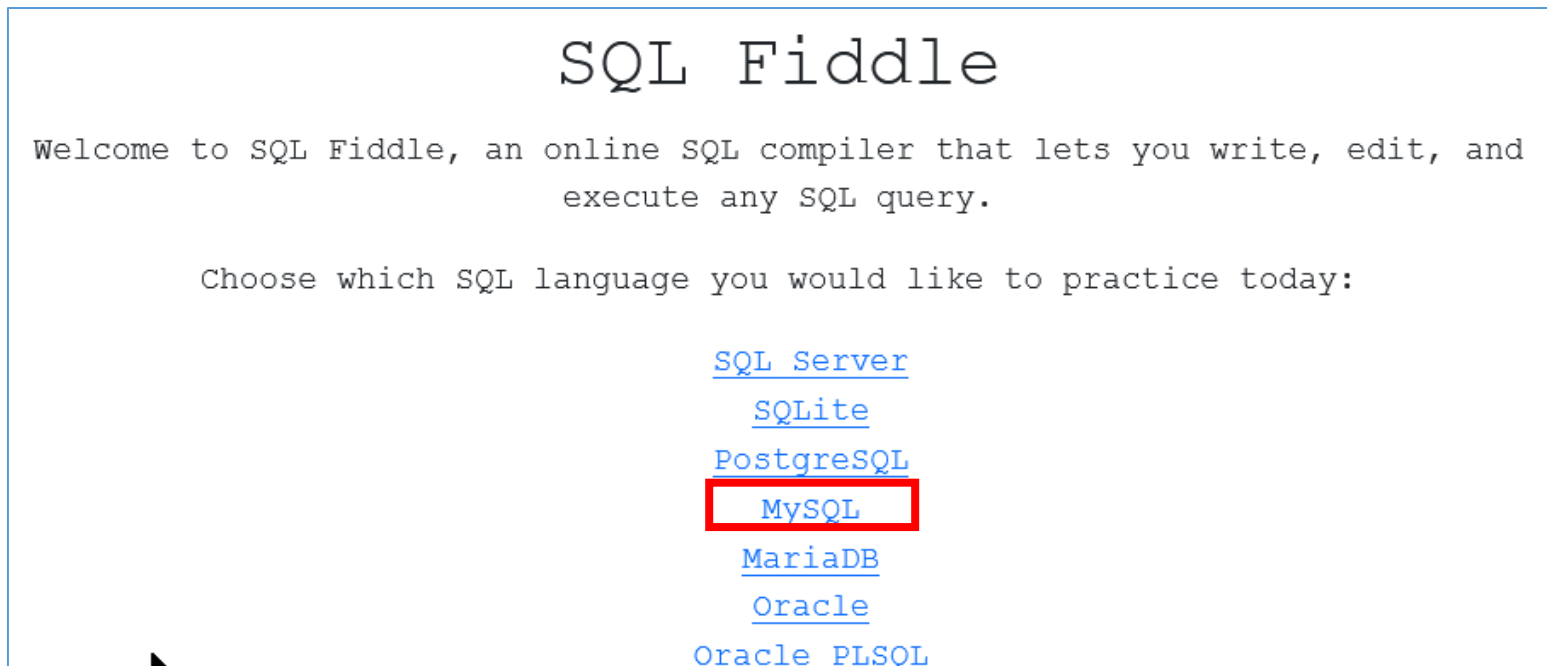
1. SELECT での集約の指定
2. GROUP BY での基準の指定
3. 各日付ごとに売り上げご受け
を計算

Webブラウザを使用

① アドレスバーにSQLFiddleのURLを入力

<http://sqlfiddle.com/>

② 「MySQL」を選択



③ 上のパネルに、テーブル定義とデータの追加と問い合わせを行う SQL を入れ実行。（以前の SQL は不要なので消す）

```
CREATE TABLE 売上 (
  日付 DATETIME,
  商品 TEXT,
  個数 INTEGER,
  単価 INTEGER
);
INSERT INTO 売上 VALUES ('2024-01-01', '商品A', 1, 500);
INSERT INTO 売上 VALUES ('2024-01-01', '商品A', 2, 500);
INSERT INTO 売上 VALUES ('2024-01-01', '商品A', 3, 500);
INSERT INTO 売上 VALUES ('2024-01-01', '商品A', 4, 500);
INSERT INTO 売上 VALUES ('2024-01-01', '商品B', 1, 1500);
INSERT INTO 売上 VALUES ('2024-01-01', '商品B', 2, 1500);
INSERT INTO 売上 VALUES ('2024-01-01', '商品B', 3, 1500);
INSERT INTO 売上 VALUES ('2024-01-01', '商品B', 4, 1500);
INSERT INTO 売上 VALUES ('2024-01-02', '商品A', 1, 500);
INSERT INTO 売上 VALUES ('2024-01-02', '商品A', 2, 500);
INSERT INTO 売上 VALUES ('2024-01-02', '商品A', 3, 500);
INSERT INTO 売上 VALUES ('2024-01-02', '商品A', 4, 500);
INSERT INTO 売上 VALUES ('2024-01-02', '商品B', 1, 1500);
INSERT INTO 売上 VALUES ('2024-01-02', '商品B', 2, 1500);
INSERT INTO 売上 VALUES ('2024-01-02', '商品B', 3, 1500);
INSERT INTO 売上 VALUES ('2024-01-02', '商品B', 4, 1500);
INSERT INTO 売上 VALUES ('2024-01-03', '商品A', 1, 500);
INSERT INTO 売上 VALUES ('2024-01-03', '商品A', 2, 500);
INSERT INTO 売上 VALUES ('2024-01-03', '商品A', 3, 500);
INSERT INTO 売上 VALUES ('2024-01-03', '商品A', 4, 500);
INSERT INTO 売上 VALUES ('2024-01-03', '商品B', 1, 1500);
INSERT INTO 売上 VALUES ('2024-01-03', '商品B', 2, 1500);
INSERT INTO 売上 VALUES ('2024-01-03', '商品B', 3, 1500);
INSERT INTO 売上 VALUES ('2024-01-03', '商品B', 4, 1500);
```

SELECT 日付, **SUM**(個数 * 単価)

FROM 売上

GROUP BY 日付;

各日付ごとに売上合計（個数 × 単価）を計算

④ 「**Execute**」をクリック

SQL 文が**実行**され、結果が表示される。

⑤ 下のパネルで、**結果を確認**。

```
+-----+-----+
| 日付 | SUM(個数 * 単価) |
+-----+-----+
| 2024-01-01 00:00:00 | 20000 |
| 2024-01-02 00:00:00 | 20000 |
| 2024-01-03 00:00:00 | 20000 |
+-----+-----+
```

- 発展学習 6 . 商品Aの総売上の計算

次のように、売上テーブルから商品Aの総売上を計算したい。
そのために、

商品Aに関するすべての行の個数と単価を掛け合わせて総売上を求めるSQL文を書いてください。

SUM(個数 * 単価)
15000

ヒント：SUMを使い、WHEREで商品Aのみを選択

- 発展学習 7. 日付別の商品別総売上の計算

各日付における商品別の総売上を求めたい。

**日付と商品ごとにグループ化し、それぞれの総売上を求める
次の SQL文を実行し結果を確認してください**

SELECT 日付, 商品, SUM(個数 * 単価) FROM 売上 GROUP BY 日付, 商品;

ヒント：GROUP BY句を使用して日付と商品によってグループ化し、SUMで総売上を計算

解答例

発展学習 6 .

SELECT SUM(個数 * 単価) FROM 売上 WHERE 商品 = '商品A';

発展学習 7

日付	商品	SUM(個数 * 単価)
2023-01-01T00:00:00Z	商品A	5000
2023-01-01T00:00:00Z	商品B	15000
2023-01-02T00:00:00Z	商品A	5000
2023-01-02T00:00:00Z	商品B	15000
2023-01-03T00:00:00Z	商品A	5000
2023-01-03T00:00:00Z	商品B	15000

演習まとめ

集約

- 集約の種類：SUM, AVG, COUNT, MAX, MIN

SELECT AVG(得点) FROM 成績 WHERE 科目 = '国語';

(国語の平均点を計算)

SELECT SUM(個数 * 単価) FROM 売上 WHERE 商品 = '商品A';

(商品Aの総売上を計算)

データ分析：

- グループ化を通じてデータのパターンやトレンドを明らかにする。
- GROUP BYを用いて関連するデータをグループ分け

SELECT 科目, AVG(得点) FROM 成績 GROUP BY 科目;

(科目別の平均点を計算)

SELECT 日付, SUM(個数 * 単価) FROM 売上 GROUP BY 日付;

(日付ごとの総売上を計算)

全体まとめ①

グループ化

- ・ **グループ化**は、**同じ属性値を共有するデータを集める**プロセス。

例： 科目ごとにグループを作成（国語、算数、理科）

受講者ごとにグループを作成（A、B）

集約の種類

- ・ 平均 AVG、最大 MAX、最小 MIN、合計 SUM、行数 COUNT

GROUP BY の利用

- ・ **SQL 問い合わせ「SELECT ...」の中で、GROUP BY を使用してデータをグループ化**
- ・ **1つ以上の属性を GROUP BY に指定してグループ化の基準とする。**

科目ごとに受講者の数を数える

SELECT 科目, COUNT(*) FROM 成績 GROUP BY 科目;

全体まとめ②

データ分析の応用

- グループ化によりカテゴリ別、時系列別の分析が可能に。
- ビジネスインテリジェンスにおける売上トレンドや顧客セグメントの分析に利用。



① データの把握

複雑なデータセットを簡単なSQLコマンドで分析し、集約することで、データをより深く把握できるようになる。

② データ分析と洞察のためのスキル獲得

GROUP BY を活用してデータをグループ化し、特定の属性に基づいたパターンやトレンドを明らかにするスキルが身につく。

③ 意思決定に役立つスキル獲得

効率的なデータ整理と分析を通じて、ビジネスの意思決定に必要な洞察を提供する分析を行うことができ、データを視覚化することで他者に情報を伝えるスキルが強化される。