



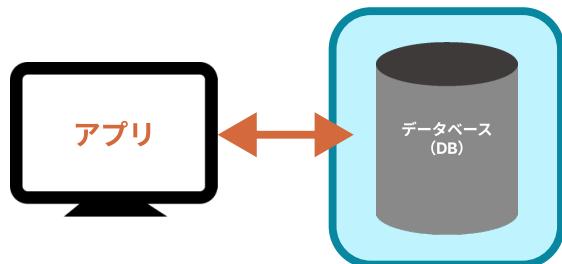
# MySQLはじめの一歩

あなたのデータが“宝の山”に変わる瞬間

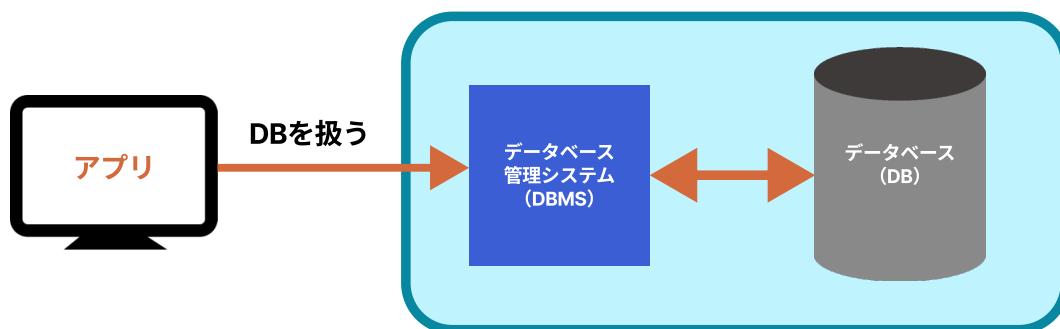


# DBについておさらい

**DB** 必要な時にデータを保存して  
必要な時にデータを参照する



**DBMS** データベースを管理し、  
操作するためのソフトウェア



## DBMSの特徴

- ①構造定義・整合性維持
- ②セキュリティ・アクセス制御
- ③データ操作（検索・追加・削除など）のサポート
- ④同時アクセスの制御
- ⑤バックアップ・リカバリ

## DBについておさらい2

RDB リレーショナル型データベース

データをテーブルと呼ばれる二次元の表で表現し、テーブル間に関連性を持たせて管理

ユーザマスタ

ユーザID	ユーザ名
0001	山田テスト
0002	田中テスト

商品マスタ

商品ID	商品名	単価
A110	ボールペン	100
A120	消しゴム	200

売上テーブル

売上ID	ユーザID	商品ID	数量	売上日
1	0002	A110	3	2025-08-12
2	0001	A120	3	2025-08-13

主キー (Primary Key)

テーブル内で行を一意に識別するために使われる列  
値がユニークである必要  
Nullは許可されない（必ずデータがある）

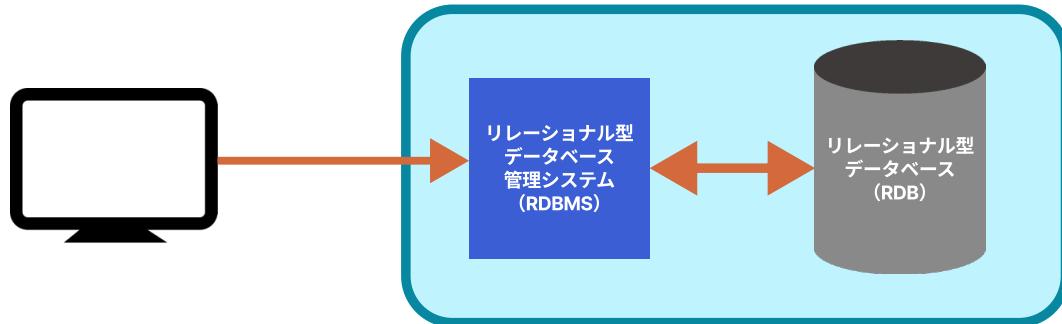
レコード

フィールド テーブル

商品ID	商品名	単価
A110	ボールペン	100
A120	消しゴム	200

# DBについておさらい3

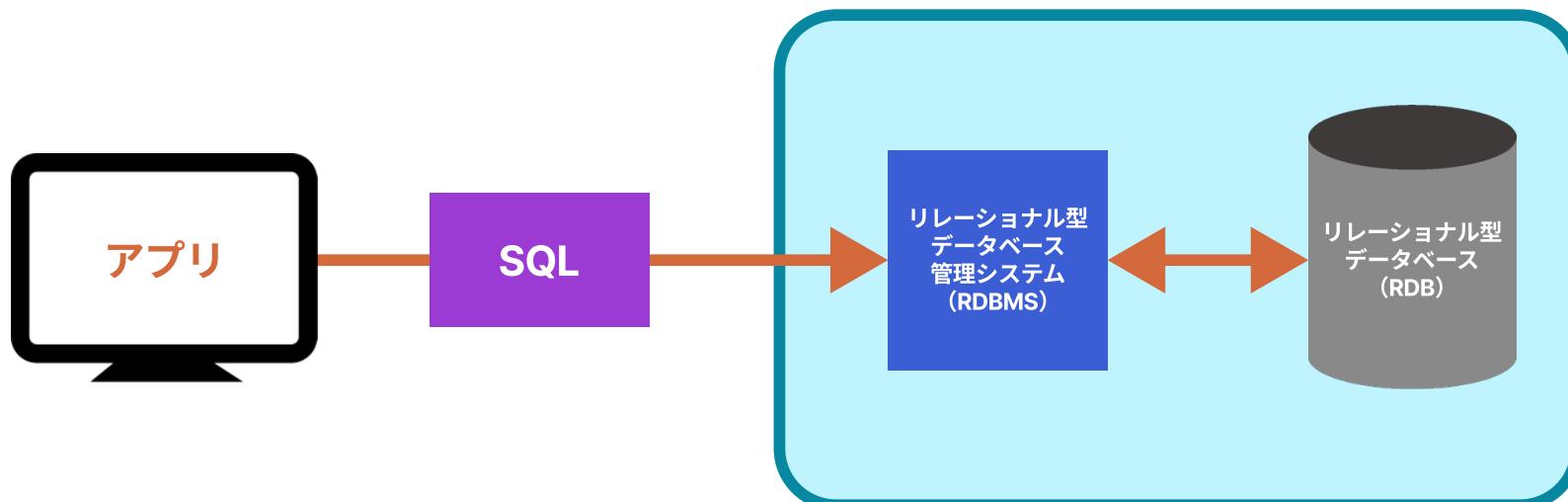
## RDBMS



- Oracle
- SQL Server
- MySQL
- PostgreSQL
- MariaDB
- Db2

# SQL

RDBの各種定義・操作・制御を行う 構造化問い合わせ言語



大文字と小文字は区別しない

(慣習的に キーワードは大文字、テーブル名やカラム名は小文字で書く)

# SQLの分類

## DDL(Data Definition Language):データ定義言語

CREATE(データベースやテーブルを作成)      ALTER(データベースやテーブルを変更)

DROP(データベースやテーブルを削除) など

★使う頻度が高いのはこれ

## DML(Data Manipulation Language):データ操作言語

SELECT(データを取得)      INSERT(データを挿入)

UPDATE(データを更新)      DELETE(データを削除) など

## DCL(Data Control Language):データ制御言語

GRANT(権限割当)

COMMIT(トランザクションの確定)

REVOKE(権限取消)

ROLLBACK(トランザクションの破棄)など

### ・ログインする時

パスワードの情報をSELECT文で取得して同じだったらログインさせる

### ・アカウントを取得した時

INSERT文でアカウント情報を追加

### ・パスワードを変える時

UPDATE文でデータを書き換える

### ・アカウントを退会するとき

DELETE文でデータを削除

### トランザクションとは

SQLを書き間違えて大事なデータを消してしまったり、

変更してしまったりするのを防ぐための機能

SQL実行後の結果を確認してから結果の確定・破棄を選択できる便利機能

# 基本的なSELECT文1

SELECT [取得したい要素] FROM [使用テーブル];

ゲームキャラクター テーブル (game\_character)

<b>id</b>	<b>name</b>	<b>job</b>	<b>level</b>
1	ヤマダ	勇者	30
2	スズキ	戦士	42
3	タナカ	戦士	25
4	サトウ	魔法使い	60
5	ワタナベ	僧侶	20

特定要素だけ表示する

**SELECT name FROM game\_character;**

```
+-----+
| name   |
+-----+
| ヤマダ |
| スズキ |
| タナカ |
| サトウ |
| ワタナベ|
+-----+
```

「game\_characterテーブル」から「name」だけを全て表示

## 基本的なSELECT文2

ゲームキャラクター テーブル (game\_character)

<b>id</b>	<b>name</b>	<b>job</b>	<b>level</b>
1	ヤマダ	勇者	30
2	スズキ	戦士	42
3	タナカ	戦士	25
4	サトウ	魔法使い	60
5	ワタナベ	僧侶	20

「,」で区切ることで複数の要素を取得

**SELECT name,level FROM game\_character;**

name	level
ヤマダ	30
スズキ	42
タナカ	25
サトウ	60
ワタナベ	20

## 基本的なSELECT文3

SELECT [取得したい要素] FROM [使用テーブル] WHERE [条件文];

ゲームキャラクター テーブル (game\_character)

id	name	job	level
1	ヤマダ	勇者	30
2	スズキ	戦士	42
3	タナカ	戦士	25
4	サトウ	魔法使い	60
5	ワタナベ	僧侶	20

「WHERE」で条件を絞る

SELECT \* FROM game\_character WHERE 30 <= level;

+-----+   id   name   job   level   +-----+
1   ヤマダ   勇者   30
2   スズキ   戦士   42
4   サトウ   魔法使い   60
+-----+

該当するレベル30以上のキャラクターのみ取得

# 基本的なSELECT文4

SELECT [取得したい要素] FROM [使用テーブル] ORDER BY [ソートしたい要素] [昇順・降順の指定(省略時昇順)];

ゲームキャラクター テーブル (game\_character)

id	name	job	level
1	ヤマダ	勇者	30
2	スズキ	戦士	42
3	タナカ	戦士	25
4	サトウ	魔法使い	60
5	ワタナベ	僧侶	20

ORDER BYでソート

**SELECT \* FROM game\_character ORDER BY level desc;**

[昇順・降順の指定]部分には、asc(昇順)、desc(降順)のどちらかを指定

+-----+   id   name   job   level   +-----+
4   サトウ   魔法使い   60
2   スズキ   戦士   42
1   ヤマダ   勇者   30
3   タナカ   戦士   25
5   ワタナベ   僧侶   20

# 基本的なSELECT文5

SELECT [取得したい要素] FROM [使用テーブル] GROUP BY [グループ化したい要素];

ゲームキャラクター テーブル (game\_character)

<b>id</b>	<b>name</b>	<b>job</b>	<b>level</b>
1	ヤマダ	勇者	30
2	スズキ	戦士	42
3	タナカ	戦士	25
4	サトウ	魔法使い	60
5	ワタナベ	僧侶	20

「GROUP BY」でグループわけ

職業の種類・所属人数を知りたい時など

**SELECT job, count( \* ) FROM game\_character GROUP BY job;**

job	count(*)
僧侶	1
勇者	1
戦士	2
魔法使い	1



LINE公式アカウントにて  
最新情報配信中！



無料で自由に使える  
学習＆カフェスペース開放中！

## 変化を楽しみ、自分らしく未来へ。

デザインエンジニアリングは、  
挑戦するエンジニアの一歩を応援する会社です。

“好き”や“ワクワク”をそのままキャリアに変え、  
自分の可能性を信じて前へ進む人には、無限のチャンスが広がっています。

失敗も学びに変え、仲間と共に笑い、共に成長しながら、  
毎日が少しずつ楽しくなる未来へ。  
未経験でも大丈夫。あなたの最初の一歩を、心からお待ちしています！



イベント・セミナー開催中！

カジュアル面談・エントリーは  
こちらから！



LINE ID:@749gaovb

検索

URL : <https://design-engineering.jp/>