データベース

Ver.1.2

データベースとSQL

- リレーショナルデータベース
- SQL

データベース

Goodworks
technology for the enterprise

大量のデータを扱う場合、データベース（DB）を利用してデータの保管や検索を行わせるのが一般的。主流はリレーショナルデータベース（RDB）。

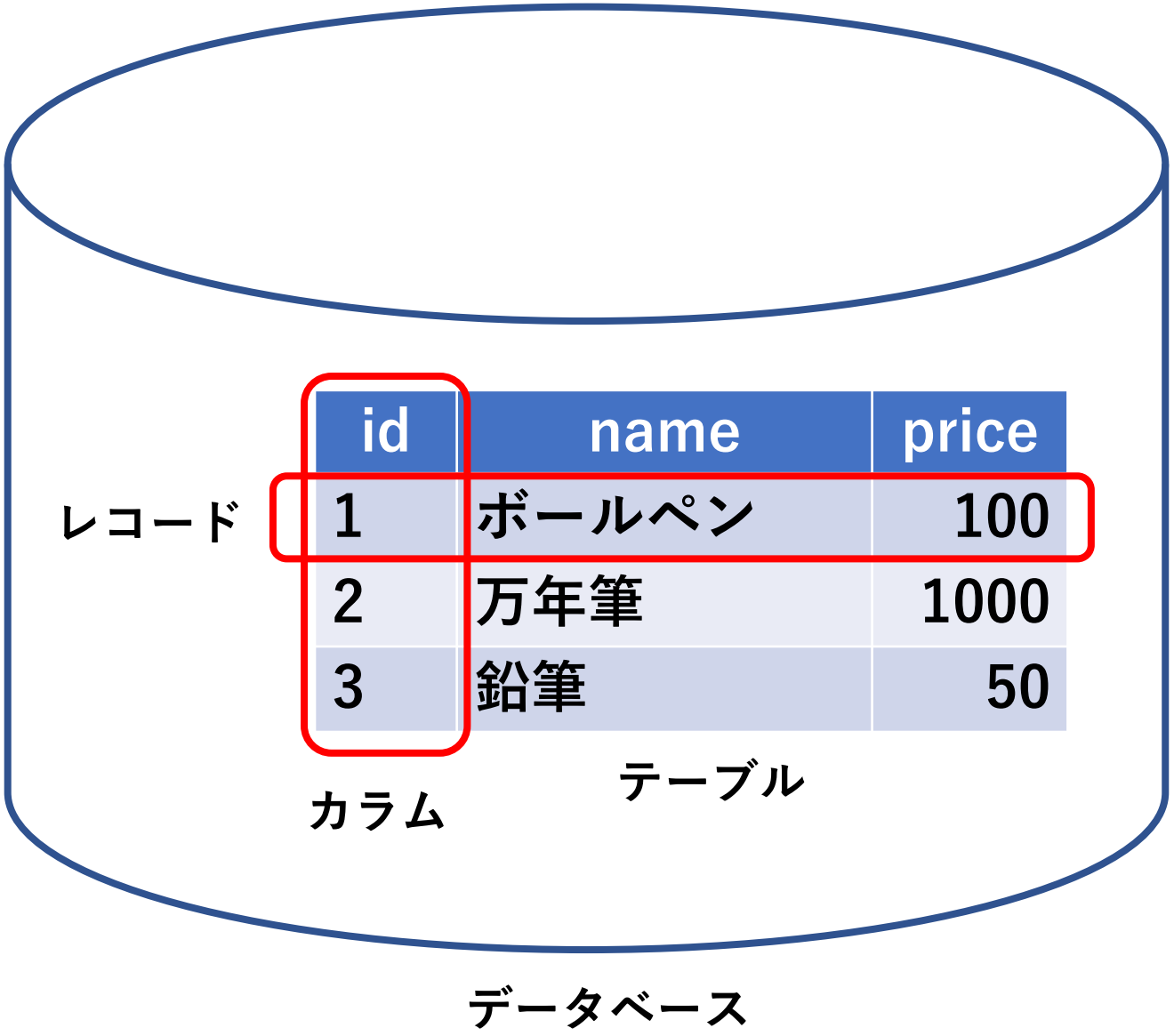
・リレーショナルデータベースは、データを表形式で扱い管理する。

id	name	price
1	ボールペン	100
2	万年筆	1000
3	鉛筆	50

RDB の仕組みと用語

Goodworks
technology for the enterprise

- データベース
  テーブルの集合体
- テーブル（表）
  情報が格納された箱
- レコード（行）
  テーブルの行
- カラム（列）
  テーブルの列



SQL

Goodworks
technology for the enterprise

データベースは SQL（Structured Query Language）を使用して、データの操作や問い合わせを行う。

1	SELECT	テーブルから特定の条件に該当する情報を検索する
2	INSERT	テーブルに新しいデータを挿入（追加）する
3	UPDATE	テーブルに登録されているデータを更新する（書き換える）
4	DELETE	テーブルに登録されているデータを削除する

- 上記は操作言語（DML：Data Manipulation Language）と呼ばれる。
- この他、テーブルを作成・更新・削除する等の処理は定義言語（DDL：Data Definition Language）と呼ばれる。

テーブルの作成

Goodworks
technology for the enterprise

MySQL を利用した、データベースとテーブルの作成手順。

① コマンドプロンプトを起動（windows + R → 「cmd」と入力してEnter）して、以下のコマンドにより MySQL に接続する。

```
> mysql -u root -p
```

② パスワードを聞かれるので、passwordを入力する。

```
Enter password: ********
```

③ 既存のデータベースを表示する。

```
mysql> show databases;
```

④ データベースを作成する。（show databasesで確認しましょう）

```
mysql> create database lesson20xx;
```

⑤ ④ で作成したデータベースに接続する。

テーブルの作成

Goodworks
technology for the enterprise

MySQL を利用した、データベースとテーブルの作成手順。

⑤ ④ で作成したデータベースに接続する。

```
mysql> use lesson20xx;
```

⑥ テーブルを作成する。

```
mysql> create table item (
       id int,
       name varchar(20),
       price int
   );
```

テーブル名

カラム名

型名

MySQLのデータ型 ①

Goodworks
technology for the enterprise

型名	値の範囲	UNSIGNEDを付与した場合の範囲
TINYINT	-128～+127	0～+255
SMALLINT	-32768～+32767	0～+65535
MEDIUMINT	-8388608～+8388607	0～+16777215
INT	-2147483648～+2147483647	0～+4294967295
BIGINT	-9223372036854775808～+9223372036854775807	0～+18446744073709551615
FLOAT(M.D)	-3.402823466E+38 ～ -1.175494351E-38, 0, 1.175494351E-38 ～ 3.402823466E+38	負数は使用不可
DOUBLE(M.D)	-1.7976931348623157E+308 ～ -2.2250738585072014E-308, 0, 2.2250738585072014E-308 ～ 1.7976931348623157E+308	負数は使用不可
DECIMAL(M.D)	MとDで変化する	なし（利用不可）

MySQLのデータ型 ②

Goodworks
technology for the enterprise

型名	値の範囲（バイト）	挿入に必要なサイズ
CHAR(M)	255	Mバイト
VARCHAR(M)	255	X+1バイト
TINYBLOB	255	X+1バイト
BLOB	65535	X+2バイト
MEDIUMBLOB	16777215	X+3バイト
LONGBLOB	4294967295	X+4バイト
TINYTEXT	255	X+1バイト
TEXT	65535	X+2バイト
MEDIUMTEXT	16777215	X+3バイト
LONGTEXT	4294967295	X+4バイト
ENUM("選択肢1","選択肢2",・・・)	65535 選択肢	1～2バイト
SET("選択肢1","選択肢2",・・・)	64選択肢	1/2/4/8バイト

【演習】

Goodworks
technology for the enterprise

- 前述の手順を参考に、MySQL に接続して item テーブルを作成しましょう。
- 以下のコマンドでテーブルの一覧を表示しましょう。

```
mysql> show tables;
```

- 以下のコマンドでテーブルのカラム一覧を表示しましょう。

```
mysql> show columns from item;
```

- 以下のコマンドで作成した item テーブルを削除しましょう。

```
mysql> drop table item;
```

プライマリキー

Goodworks
technology for the enterprise

テーブルからレコードを一意に特定するためのキーとなる列。
「プライマリキー」は「主キー」、「PKEY」とも呼ばれる。
プライマリキーに指定された列は値を重複して登録できない。

（例）id をプライマリキーとしてテーブルを作成

```
mysql> create table item (
       id int primary key,
       name varchar(20),
       price int
   );
```

not null

必ず情報を入力しなければいけない列。
プライマリキーの列は自動的に not null となる。

（例）name を not null としてテーブルを作成

```
mysql> create table item (
       id int,
       name varchar(20) not null,
       price int
   );
```

【演習】

Goodworks
technology for the enterprise

- 前頁の手順を参考に、id をプライマリキー、name を not null として item テーブルを作成しましょう。
- 以下のコマンドでテーブルのカラム一覧を表示しましょう。

```
mysql> show columns from item;
```

テーブルの更新

Goodworks
technology for the enterprise

テーブルの更新(列の追加)方法

① コマンドプロンプトを起動して、以下のコマンドにより MySQL に接続する。

```
> mysql -u root -p
```

② 作成済のデータベースに接続する。

```
mysql> use lesson20xx;
```

③ テーブルを更新(列を追加)する。

```
mysql> alter table item add cat_id int;
```

④テーブルを更新(列を削除)する。

```
mysql> alter table item drop cat_id;
```

INSERT 文

Goodworks
technology for the enterprise

テーブルにレコードを追加するには INSERT 文を使用する。

・基本文法

insert into テーブル名 values (値,値, ...);

（例）insert into item values (1, "ボールペン", 100);

～複数のレコードを一度に追加する場合～
（例）insert into item values (2,”万年筆”,1000),(3,”鉛筆”,50);

～カラムを指定してレコードの追加をする場合～
（例）insert into item (id,name) values (4, “消しゴム");

SELECT 文

Goodworks
technology for the enterprise

テーブルに必要なデータを問い合わせるには SELECT 文を使用する。

・基本文法

select 列名 from テーブル名 where 条件文;

（例）select price from item where price >= 100;

（例）select * from item;

「*」は、すべての列を表す指定
「where 条件文」を省略すると指定テーブル内の全データが対象となる

条件文

Goodworks
technology for the enterprise

条件文には以下の演算子が利用できる。

演算子	説明
=	等しい
>	大きい
<	小さい
>=	大きい、もしくは等しい
<=	小さい、もしくは等しい
!= <>	等しくない
AND	2つの条件を結合し、両方の条件が真
OR	2つの条件のうち、どちらか一方が真
NOT	式の結果を反転
BETWEEN a AND b	対象のフィールドが a と b の範囲内
IN	対象のフィールドが式の一覧の1つに一致
LIKE	対象のフィールドがパターンに一致

ORDER BY 句

データをソートするには ORDER BY 句を使用する。

・SELECT 文への ORDER BY 句の適用

select 列名 from テーブル名 where 条件文 order by 列名 [asc|desc];

（例）select * from item order by price asc;

最大値、最小値

最大値、最小値を抽出するには max, min を使用する。

・SELECT 文への max の適用

select max(列名) from テーブル名 where 条件文;

（例）select max(price) from item;

【演習】

Goodworks
technology for the enterprise

- INSERT 文を使用し、以下の情報を item テーブルに追加しましょう。

id	name	price
1	ボールペン	100
2	万年筆	1000
3	鉛筆	50

- SELECT 文を使用し、item テーブルの全データを表示しましょう。
- SELECT 文を使用し、「price が 100 以下」かつ「id が 2 以上」のデータを表示しましょう。

UPDATE 文

Goodworks
technology for the enterprise

テーブル内のレコードを更新するには UPDATE 文を使用する。

・基本文法

update テーブル名 set カラム名=値 where 条件文;

（例）update item set price = 110 where id = 1;

※where 条件文を指定しない場合はレコード全件が更新されてしまうので注意。

DELETE 文

Goodworks
technology for the enterprise

テーブル内のレコードを削除するには DELETE 文を使用する。

・基本文法

delete from テーブル名 where 条件文;

（例）delete from item where id = 1;

※where 条件文を指定しない場合はレコード全件が削除されてしまうので注意。

【演習】

Goodworks
technology for the enterprise

- UPDATE 文を使用し、ボールペンの価格を 110 に更新しましょう。
- DELETE 文を使用し、ボールペンのレコードを削除しましょう。

JOIN によるテーブル結合

Goodworks
technology for the enterprise

複数のテーブルから情報を取得する必要がある場合、JOINによるテーブル結合を行う場合があります。

■ item テーブル

id	name	price	cat_id
1	ボールペン	100	1
2	万年筆	1000	1
3	鉛筆	50	1
4	口紅	2500	2
5	マウス	3200	3

■ category テーブル

id	name
1	文房具
2	化粧品
9	その他

JOIN によるテーブル結合

Goodworks
technology for the enterprise

・内部結合（INNER JOIN）の基本構文

SELECT (列名) FROM テーブル名1 INNER JOIN テーブル名2 ON (結合条件);

（例）select item.name, category.name

from item inner join category on item.cat_id = category.id;

実行結果

name	name
ボールペン	文房具
万年筆	文房具
鉛筆	文房具
口紅	化粧品

※ 結合できなかった「マウス」や「その他」は表示されない。

JOIN によるテーブル結合

Goodworks
technology for the enterprise

・外部結合（OUTER JOIN）の基本構文

SELECT (列名) FROM テーブル名1 [LEFT|RIGHT] OUTER JOIN テーブル名2

ON (結合条件);

（例）select item.name, category.name

from item left outer join category on item.cat_id = category.id;

実行結果

name	name
ボールペン	文房具
万年筆	文房具
鉛筆	文房具
口紅	化粧品
マウス	NULL

※ left outer join の場合、
左の表（item）の結合できない
「マウス」も出力の対象となる。

※ right outer join の場合、
右の表（category）の結合できない
「その他」も出力の対象となる。

副問い合わせ

SELECT 文の中に SELECT 文を記述できる。

SELECT 文の実行結果を WHERE 句の条件に組み込む。

（例）select name from item where cat_id =

(select id from category where name=“文房具”);

name
ボールペン
万年筆
鉛筆

【演習】

Goodworks
technology for the enterprise

SQLを用いて以下の操作を実行しましょう。

⓪ 在庫テーブルの作成

```
create table zaiko (
        id int primary key,
        name varchar(20) not null,
        price int,
        quantity int
);
```

【演習】

Goodworks
technology for the enterprise

① 値の挿入

1001、ノート、100、400
1002、マウス、3000、500
1003、口紅、130、80
1004、万年筆、1000、50
1005、模造紙、100、1000
1006、鉛筆、50、100

② 「在庫テーブル」から「id」「name」「price」を一覧で出力。

③ id 1003 の価格を「2500」に更新、id 1005 の name を「上質紙」に更新。

④ id 1006の行を削除する。

【演習】

Goodworks
technology for the enterprise

⑤「在庫テーブル」から「商品コード」が 1001 から 1003 までの商品を抽出。

⑥ 価格を昇順にして表示させる。

⑦ 数量を降順にして表示させる。

⑧ id が 1007 から 1010 までレコードを作成（値は任意でよい）。

⑨ 価格の最大値、最小値をそれぞれ表示させる。

【演習】

Goodworks
technology for the enterprise

⑩ カテゴリテーブルを作成

```
create table category (
        id int primary key,
        name varchar(20) not null
);
```

⑪ カテゴリテーブルに値の挿入

1、文房具

2、化粧品

3、パソコン用品

9、その他

【演習】

Goodworks
technology for the enterprise

⑫ 在庫テーブルに cat_id 列（int）を追加

⑬ cat_id 列に値を格納

⑭ JOIN を利用し、商品名とカテゴリ名を一覧で表示

⑮ 副問い合わせを利用し、カテゴリが「化粧品」以外の全ての商品を表示する