

# dd-2. テーブルと テーブル定義

リレーショナルデータベースの  
基本（短縮版）（全7回）

基本を把握したい人へ

<https://www.kkaneko.jp/data/dd/index.html>

金子邦彦



謝辞：この資料では「かわいいフリー素材集 いらすとや」のイラストを使用しています

# 第2回のアウトライン



- テーブル
- テーブル定義
- データ型
- 主キー
- NULL
- 一貫性制約

# テーブルの例



テーブル名: products

id	name	price
1	orange	50
2	apple	100
3	melon	500

# テーブルの性質 ①



## リレーショナルデータベースならではの決まり事

テーブル名：福山駅行き

時	分
8	0
8	20
8	45
12	30
17	20
17	40

リレーショナルデータベースで  
扱えないテーブルの例

時	分
8	0, 20, 45
12	30
17	20, 40

☑ リレーショナルデータベース  
では、1つのセルに1つの値

☐ 1つのセルに複数の値を  
入れることは**ない**

# テーブルの性質 ②



## リレーショナルデータベースならでの決まり事

テーブル名：福山駅行き

リレーショナルデータベースで  
扱えないテーブルの例

時	分
8	0
8	20
8	45
12	30
17	20
17	40

時	分
8	0
	20
	45
12	30
17	20
	40

リレーショナルデータベース  
では、1つのセルに1つの値

マルチカラムにはしない

# テーブル定義



テーブル名: products

テーブル

id	name	price
1	orange	50
2	apple	100
3	melon	500

「id」と「name」と  
「price」の属性

# リレーショナルデータベースの構築手順



データベース  
設計



データベース  
生成

※ 最初、デー  
タベースは空



id	購入者	商品 ID	数量

  

id	name	price

テーブル定義

※ 最初、テーブルは空



id	購入者	商品 ID	数量
1	X	1	10
2	Y	2	5

id	name	price
1	orange	50
2	apple	100
3	melon	500

テーブル生成

# テーブル定義



テーブル名: products

テーブル定義では,

- ・ **テーブル名**
- ・ **属性の属性名**
- ・ **属性のデータ型**

などを設定して, **テーブル**を**定義**する

id	name	price
1	orange	50
2	apple	100
3	melon	500

```
CREATE TABLE products (  
  id INTEGER PRIMARY KEY NOT NULL,  
  name TEXT NOT NULL,  
  price REAL);
```



# 属性のデータ型



id	name	price
1	orange	50
2	apple	100
3	melon	500

属性名

テーブル  
の本体

自動インクリメント 短いテキスト

INTEGER  
AUTOINCREMENT

※ マイクロソフト Access  
では, AUTOINCREMENT

TEXT

整数

INTEGER

← Access での  
日本語表示

← SQL のキーワード

それぞれの属性のデータ型

※自動インクリメントは, 自動で 1,2,3  
のように通し番号が付くもの

# 属性のデータ型



Access の主なデータ型	SQL のキーワード	
	NULL	空値
短いテキスト	CHAR	文字列
長いテキスト	TEXT	文字列
数値	INTEGER, REAL	整数や浮動小数 点数
日付／時刻	DATETIME	日付や時刻など
Yes／No	BIT, BOOL	ブール値

※ **整数**は **INTEGER**, **浮動小数点数** (小数付きの数) は **REAL**

※ **短いテキスト**は半角 **255文字分**までが目安  
それ以上になる可能性があるときは**長いテキスト**

# 主キー



通し番号，学生番号のように，1つのテーブルの中で同じ値が2回以上出ないと前もって分かっている**属性**

id	name	price
1	orange	50
2	apple	100
3	melon	500

主キー

# リレーショナルデータベースの NULL



- **NULL** は「ヌル」あるいは「ナル」と読む
- **リレーショナルデータベース**で **NULL** は、次の場合に使う
  1. **未定, 未知, 不明** (分からない場合)
  2. **非存在** (もともと存在しない場合)

# テーブル定義と一貫性制約



## 【SQL プログラム】

```
CREATE TABLE products (  
  id INTEGER PRIMARY KEY NOT NULL,  
  name TEXT NOT NULL,  
  price REAL);
```

id:

**主キー** (PRIMARY KEY),  
**NULL になることはない** (NOT NULL)

name:

**NULL になることはない** (NOT NULL)

テーブルの制約について記述.

データベースの一貫性を維持するのに役立つ.