2. リレーショナルデータベースとSQL 入門

（データベースシステム）（全15回）

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

ds-2. SQL の基本

SQL の役割、テーブルと属性、テーブル定義、問い合わせ（クエリ）

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

アウトライン

1. データベースシステムの基本概念
2. リレーショナルデータベースの特性
3. テーブルの構造
4. 問い合わせ（クエリ）の役割
5. SQL
6. SQL によるテーブル定義
7. SQL による問い合わせ（クエリ）

2-1. データベースシステムの基本概念

データベースとは

データベースは、**特定のテーマや目的**に従って収集された**大量のデータ**

例：銀行、商店、交通機関、電話会社などさまざま

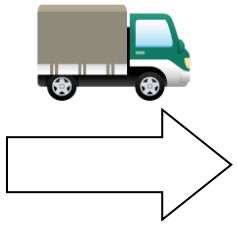
取引

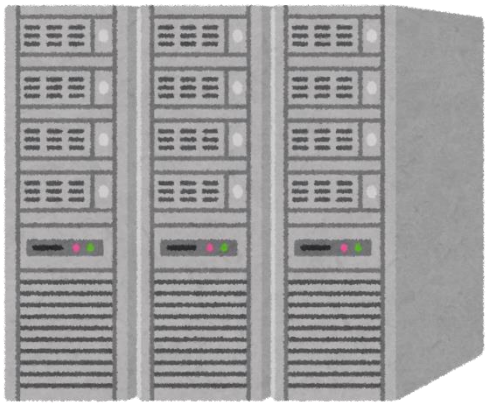
記入

計測　撮影

データ保存

データ収集





データベース
（データの集まり）

データベースシステムとは

データベースシステムは，**データベース**を扱う IT のシステム

データベースシステム
＝　データベース（データの集まり）
＋　データベース管理システム（ソフトウエア）

【主要目的】

大量のデータを**安全かつ効率的に保存、管理、検索、共有すること**で、迅速な業務実行と正確な意思決定が可能になる。

2-2. リレーショナルデータベースの特性

リレーショナルデータベースと他のデータベースの違い

リレーショナルデータベース

- データを**テーブル**と呼ばれる**表形式で保存**
  （エクセルのワークシートに似ている）
- **SQL** 言語を使って、**データの管理と検索**が行える
- **テーブル形式でデータを整理**することで、**データの構造化、整合性、永続性が保証される。**

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

他のデータベース（例：NoSQL）

- **データ形式がより柔軟**
- **データの構造化と整合性に関する能力はリレーショナルデータベースより制限される**

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

リレーショナルデータベースが普及している理由

- 使いやすい
- 信頼性が高い
- データを**テーブル**と呼ばれる**表形式で保存**（エクセルに似ていてなじみやすい）
- **データの管理と検索**が、**SQL** 言語を使って、簡単に行える

2-3. テーブルの構造

テーブルと属性

- **テーブル**：データを**表形式で保存**する構造
- **属性（列）**：データの種類に対応（例：ID，商品名，単価）
- **行**：**属性（列）**に基づいた具体的なデータの集まり

テーブル

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

「**ID**」と「**商品名**」と
「**単価**」の**属性**

テーブルの属性（列）と行

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

ID	購入者	商品ID	数量
1	X	1	10
2	Y	2	5

属性（列）

- データの種類に対応する

例：ID、商品名、単価など

行

- 属性に基づいた具体的なデータの集まり

例：「**1 みかん 50**」など

属性のデータ型

【主なデータ型】

- **整数（INTEGER）**：ID，単価など
- **テキスト（TEXT）**：商品名など
- **日付／時刻（DATETIME）**
- **Yes／No（BIT, BOOL）**：ブール値

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

整数で
オートナンバー　　**テキスト**　　**整数**

※**オートナンバー**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
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**テキスト**

テーブルのセル

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

ID	購入者	商品ID	数量
1	X	1	10
2	Y	2	5

セル

- 属性と行が交差したところに位置する

セルの特徴

- **１つのセル**には、**１つの値**だけが入る
- **セル**に入る値は、**属性のデータ型に応じたもの**でなければならない

テーブルの性質 ①

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は、
１つのセルに１つの値

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

□ 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

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は、**１つのセルに１つの値**

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

☐ マルチカラムには**しない**

まとめ

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

属性（列）

- データの種類に対応する

例：ID、商品名、単価など

- 各属性には特定のデータ型が設定される

行

- 属性に基づいた具体的なデータの集まり

例：「**1 みかん 50**」など

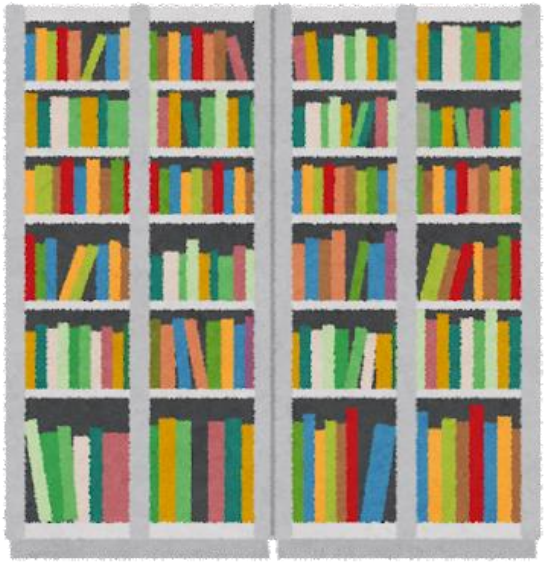
セルの特徴

- **１つのセル**には、**１つの値**だけが入る

2-4. 問い合わせ（クエリ）の役割

問い合わせ（クエリ）とは何か

- **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、加工するためのコマンド（指令）**のこと
- **リレーショナルデータベースシステム**においては、**問い合わせ（クエリ）**は、 **SQL** を使用して実行できる



問い合わせの目的と用途

1. **データの検索**: 何らかの条件に合致するデータを見つけ出す
2. **データの加工: 不要なデータは表示しない**、**集計・集約**や**計算**、**並べ替え（ソート）**、複数テーブルの**結合**など
3. **データの挿入**: データベースに新規データを追加
4. **データの更新**: すでに存在するデータの変更
5. **データの削除**: 不要なデータの除去

問い合わせ（クエリ）の仕組み

1. **発行**：「問い合わせ」をデータベースシステムに送信
2. **解析・取得**：「問い合わせ」が解析され，必要なデータがデータベースから読み込まれる
3. **加工**：計算，集計・集約，並べ替え（ソート），結合などのさまざまな加工が行われる
4. **返却**：結果を，ユーザーやアプリケーションに送る

問い合わせ（クエリ）

問い合わせ（クエリ）の最終結果

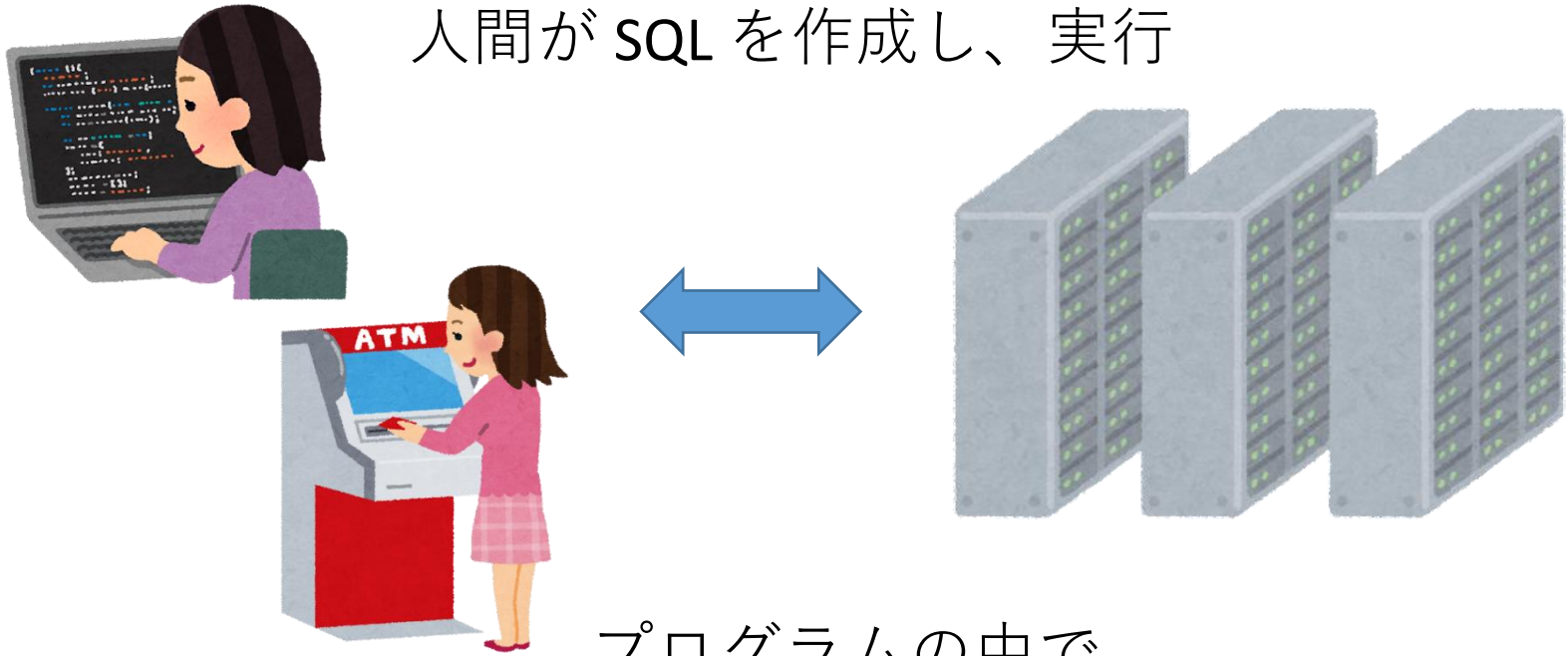
データベースシステム

id	name	price	at
1	orange	50	-05-11 13:30:18
2	apple	100	-05-11 13:30:18
3	melon	500	-05-11 13:30:18
	緑 ZZ	貸出	2021-05-11 13:30:18

2-4. SQL

SQL

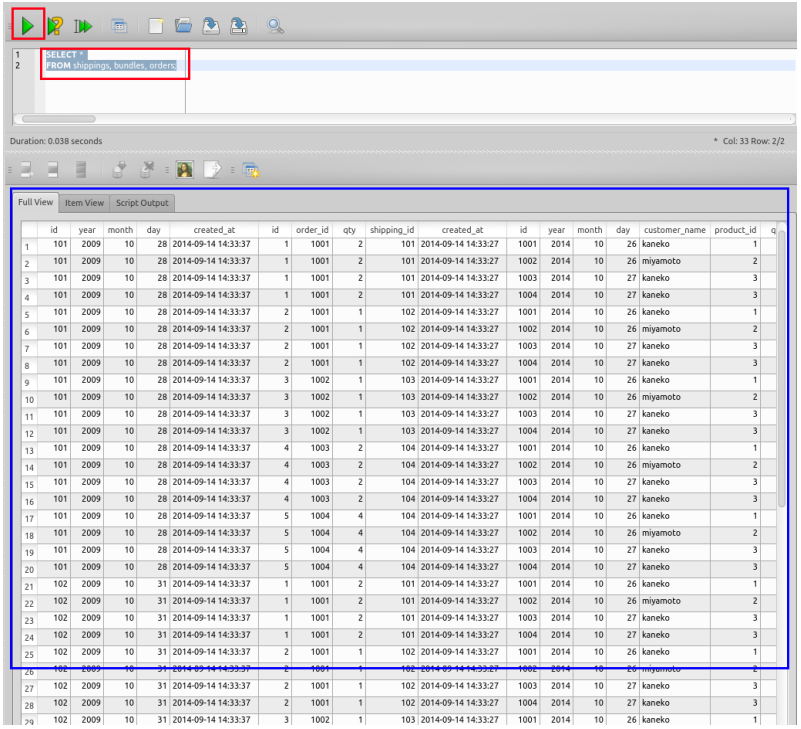
SQL は、**リレーショナルデータベースシステム**で、**問い合わせ（クエリ）**や、その他、さまざまな機能を実行できる



SQLの重要性

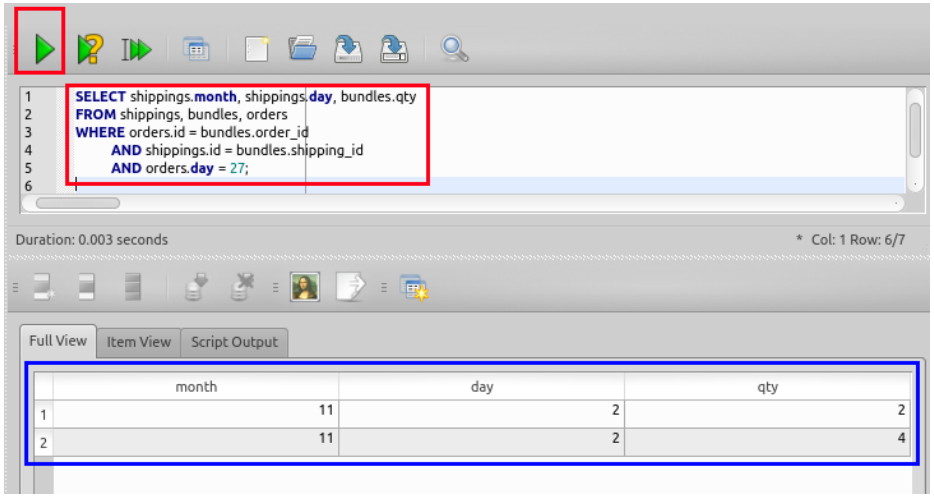
- **問い合わせ**：SQLを使用してデータの検索，加工，挿入，更新，削除が行える
- **テーブル定義**：SQLでデータベースのテーブル構造を定義することができる
- **簡潔さ**：少量のSQLで複雑なデータ操作が可能
- **汎用性**：多くのリレーショナルデータベースで対応し，業界標準として広く採用されている
- **柔軟性**：データの並べ替え（ソート），集計・集約など，多様な操作が可能
- **性能**：小規模から大規模なデータベースに対応し，高いパフォーマンスを発揮

問い合わせ（クエリ）の例 ①
データの検索



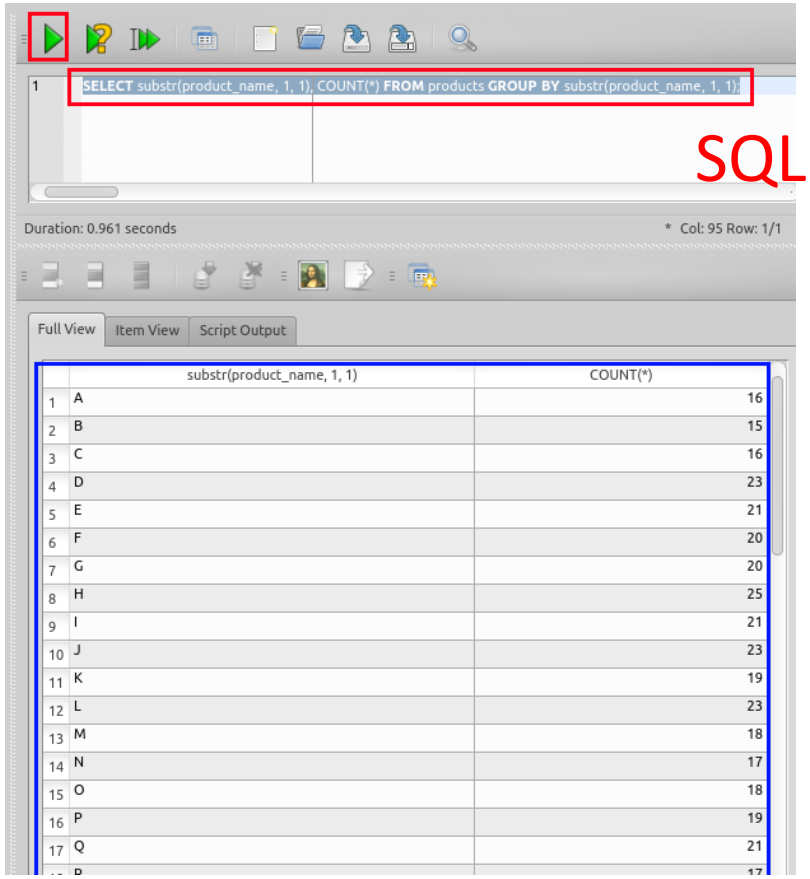
元データ

SQL

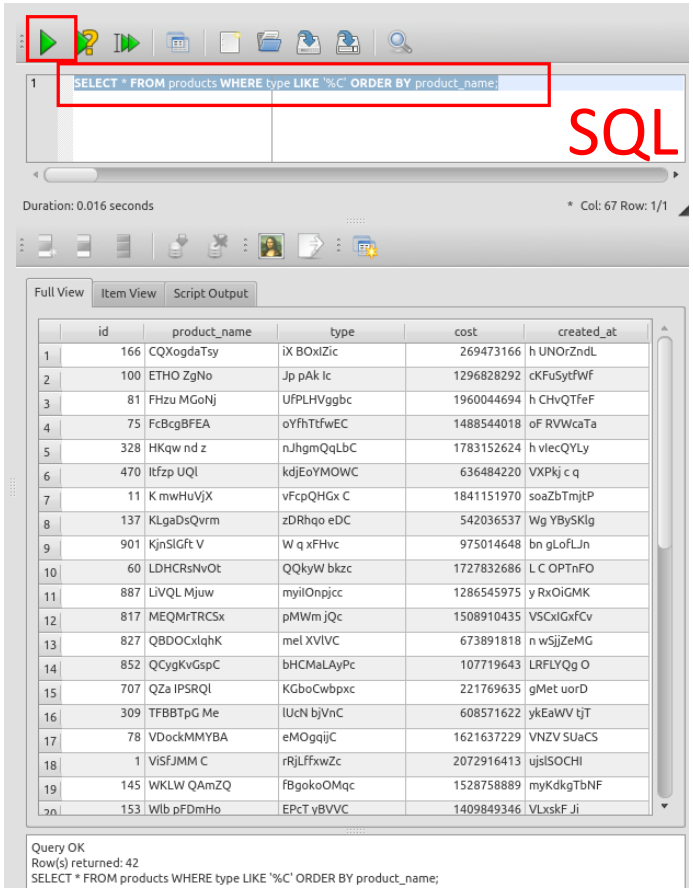


結果

問い合わせ（クエリ）の例 ②
集計・集約，並べ替え（ソート）



	substr(product_name, 1, 1)	COUNT(*)
1	A	16
2	B	15
3	C	16
4	D	23
5	E	21
6	F	20
7	G	20
8	H	25
9	I	21
10	J	23
11	K	19
12	L	23
13	M	18
14	N	17
15	O	18
16	P	19
17	Q	21



	id	product_name	type	cost	created_at
1	166	CQXogdaTsy	iX BOxlZic	269473166	h UNOrZndL
2	100	ETHO ZgNo	Jp pAk Ic	1296828292	cKFuSytfWF
3	81	FHzu MGoNj	UfPLHVggbc	1960044694	h CHvQTfeF
4	75	FcBcgBFEA	oYfhTtfwEC	1488544018	oF RVWcaTa
5	328	HKqw nd z	nJhgmQqLbC	1783152624	h vlecQYLy
6	470	Itfzp UQl	kdjEoYMOWC	636484220	VXPkj c q
7	11	K mwHuVjX	vFcpQHGx C	1841151970	soaZbTmjtP
8	137	KLgaDsQvrm	zDRhqo eDC	542036537	Wg YBySKlg
9	901	KjnSlGft V	W q xFHvc	975014648	bn gLofLJn
10	60	LDHCRsNvOt	QQkyW bkzc	1727832686	L C OPTnFO
11	887	LiVQL Mjuw	myilOnpjcc	1286545975	y RxOiGMK
12	817	MEQMrTRCSx	pMWm jQc	1508910435	VSCxIGxfCv
13	827	QBDOCxlqhK	mel XVlVC	673891818	n wSjjZeMG
14	852	QCygKvGspC	bHCMaLAyPc	107719643	LRFLYQg O
15	707	QZa IPSRQl	KGboCwbpxc	221769635	gMet uorD
16	309	TFBBTpG Me	lUcN bjVnC	608571622	ykEaWV tjT
17	78	VDockMMYBA	eMOgqijC	1621637229	VNZV SUaCS
18	1	ViSfJMM C	rRjLffxwZc	2072916413	ujslSOCHI
19	145	WKLW QAmZQ	fBgokoOMqc	1528758889	myKdkgTbNF
20	153	Wlb pFDmHo	EPcT yBVVC	1409849346	VLxskF Ji

集計・集約

並べ替え（ソート）

問い合わせ（クエリ）の例 ③
２つのテーブルの結合

商品

ID	商品名	単価
1	みかん	50
2	りんご	100
3	りんご	150
4	メロン	500

購入

ID	名前	商品
1	X	3
2	Y	1

SQL

```
select * from 商品, 購入
```

1	みかん	50	1	X	3
1	みかん	50	2	Y	1
2	りんご	100	1	X	3
2	りんご	100	2	Y	1
3	りんご	150	1	X	3
3	りんご	150	2	Y	1
4	りんご	500	1	X	3
4	りんご	500	2	Y	1

結果

SQL 活用のビジョン

リアルタイムのデータ管理

例: **オンラインショッピングサイトでの在庫確認**

データの自動分析

例: **毎月の売上レポートの自動生成**

データの安全性と一貫性（トランザクションの機能を活用）

例：リレーショナルデータベースシステムの例: **銀行での送金処理**

データの共有

例: **在庫データをサプライヤーと共有。**SQLでデータアクセス

ここまでのまとめ

SQLの特徴

- **簡潔さ**: 多機能であり、短い SQL でもさまざまな機能を使える
- **汎用性**: 複数のリレーショナルデータベースシステムで使用可能。

活用のビジョン

- リアルタイムのデータ管理，自動分析など

2-6. SQL によるテーブル定義

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	名前	単価

テーブル定義

「こういうテーブルを使いたい」と設定するだけなので、テーブルは**空**

ID	購入者	商品ID	数量
1	X	1	10
2	Y	2	5

ID	名前	単価
1	みかん	50
2	りんご	100
3	りんご	150

データ追加

テーブル定義とデータの追加

① **テーブル定義**

- **テーブル名**：**商品**
- **属性名**：**ID、商品名、単価**
- **属性のデータ型**：**数値・オートナンバー、テキスト、数値**
- データの整合性を保つための**制約**：**なし**

② 続いて、**テーブルに実際のデータを追加**

ID	商品名	単価
1	みかん	50
2	りんご	100

1, 2, 3 のような
通し番号が
自動設定される

数値・
オートナンバー

テキスト

半角の**数値**

SQL によるテーブル定義

- **create table**を使用してテーブル定義
- 構文
  create table テーブル名 (列1 型1, 列2 型2);
- 例

```
create table 商品 (
   ID autoincrement,
   商品名 text,
   単価 integer
);
```

- 注意：**Accessでは** integerをautoincrement 属性に設定する場合，「integer autoincrement」と書かずに「**autoincrement**」とだけ記述する

SQL によるテーブル定義

テーブル名：**商品**

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

数値・オートナンバー　　**テキスト**　　半角の**数値**

```
create table 商品 (
   ID autoincrement,
   商品名 text,
   単価 integer
);
```

区切りの半角カンマ

SQL文

※ **Access では**
「integer autoincrement」と書かずに「autoincrement」

2-7. SQL による問い合わせ (クエリ)

SQL のキーワード select, from, where

select

問い合わせ（クエリ）のための基本的な命令。

取得したいデータの指定

from

データ取得の対象となるテーブルを指定

例：select * from テーブル名;

where

特定の条件を満たす行の選択

例：select * from テーブル名 where 列1 = 値1;

問い合わせ（クエリ）のバリエーション

① 全データの取得

select * from 商品;

② 特定の属性（列）のみ取得

select 商品名, 単価 **from** 商品;

③ 条件付き検索

select 商品名, 単価 **from** 商品 **where** 単価 > 80;

問い合わせ（クエリ）の仕組み

1. **発行**：「問い合わせ」をデータベースシステムに送信
2. **解析・取得**：「問い合わせ」が解析され，必要なデータがデータベースから読み込まれる
3. **加工**：計算，集計・集約，並べ替え（ソート），結合などのさまざまな加工が行われる
4. **返却**：結果を，ユーザーやアプリケーションに送る

問い合わせ（クエリ）

問い合わせ（クエリ）の最終結果

データベースシステム

id	name	price	at
1	orange	50	-05-11 13:30:18
2	apple	100	-05-11 13:30:18
3	melon	500	-05-11 13:30:18
	緑 ZZ	貸出	2021-05-11 13:30:18

問い合わせ（クエリ）のバリエーション

select * from 商品

問い合わせ（クエリ）のコマンド

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

元のテーブルのまま表示

商品テーブル

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

問い合わせ（クエリ）のバリエーション

select 商品名, 単価 from 商品

問い合わせ（クエリ）
のコマンド

商品名	単価
みかん	50
りんご	100
メロン	500

必要な属性を選ぶ（**射影**）

商品テーブル

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

問い合わせ（クエリ）のバリエーション

```
select 商品名, 単価 from 商品
where 単価 > 80;
```

問い合わせ（クエリ）
のコマンド

	商品名	単価
	りんご	100
	メロン	500
*		

必要な属性を選び（**射影**），
行を絞り込む（**選択**）

商品テーブル

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

SELECT * FROM <テーブル名>

元の単一テーブルを**そのまま**出力

ID	名前	単価
1	みかん	50
2	りんご	100
3	りんご	150
4	メロン	500

```
SELECT * FROM 商品;
```

ID	名前	単価
1	みかん	50
2	りんご	100
3	りんご	150
4	メロン	500

SELECT ＜属性名リスト＞ FROM ＜テーブル名＞

属性を限定する

ID	名前	単価
1	みかん	50
2	りんご	100
3	りんご	150
4	メロン	500

```
SELECT 名前, 単価 FROM 商品;
```

名前	単価
みかん	50
りんご	100
りんご	150
メロン	500

```
SELECT 名前 FROM 商品;
```

名前
みかん
りんご
りんご
メロン

WHERE 付き

行の**選択**

ID	名前	単価
1	みかん	50
2	りんご	100
3	りんご	150
4	メロン	500

```
SELECT 名前, 単価 FROM 商品
WHERE 単価 > 80;
```

名前	単価
りんご	100
りんご	150
メロン	500

SQLによる問い合わせ（クエリ）

- **select**：取得したいデータの指定
- **from**：データ取得の対象となるテーブルを指定
- **where**：特定の条件を満たす行の選択
- 基本的な構文：

例 **select * from テーブル名;**

- 条件付き検索：

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

- **group by**：データを条件でグループ化（集計・集約のため）
- **order by**：データを昇順・降順で並べる

SQLの文法

- **大文字小文字**：

SQLは大文字・小文字を区別しない（SELECTとselectは同じ）

- **改行**：

SQL文は途中で改行しても問題ない（改行はコードの可読性を高める）

- **セミコロン (;)**：
  - 一つのSQL文のみの場合，末尾の;は省略可能
  - ２つ以上のSQL文を連ねる場合は，;で文を区切る必要がある

SQL問い合わせの実用例

① SQL コマンドによるデータベースの利用

データ検索、給与が50,000以上の従業員の情報を全て取得

SQL: SELECT * FROM employees WHERE salary > 50000;

データ挿入、名前が'John'、年齢が30、給与が55,000の新しい従業員を追加

SQL: INSERT INTO employees (name, age, salary) VALUES ('John', 30, 55000);

データ削除、年齢が60より上の従業員のデータを削除

SQL: DELETE FROM employees WHERE age > 60;

② プログラム中の処理の一部を SQL で実行

注文処理、ユーザーIDが1で商品IDが101の商品を2個注文

SQL: INSERT INTO orders (user_id, product_id, quantity) VALUES (1, 101, 2);

③ データ分析での SQL の役割

2024年1月1日から2024年1月31日までの売上合計を計算

SQL: SELECT SUM(sales_amount) FROM sales WHERE date BETWEEN ‘2024-01-01' AND ‘2024-01-31';

全体まとめ

- **データベースシステム**は**大量のデータを効率的に管理**するためのシステム
- **リレーショナルデータベース**は**テーブル形式**でデータを保存し，SQLで操作する
- **テーブル**は**属性（列）**と**行**から構成され，各セルには**データ型に応じた値**が入る
- **SQL**を使用して，**テーブル定義**や様々な**問い合わせ（クエリ）**を行うことができる
- SQLの基本的な構文（select, from, where）を理解することで，データの検索や加工が可能になる

① キャリア面での成長

データベース管理およびSQLスキルは、多くの産業や職種で非常に重要です。データアナリスト、データベース管理者、ソフトウェアエンジニアなどの職種において、SQLスキルを持つことはキャリア面での成長につながります。

② データベースを用いた意思決定のスキル向上

データベースとSQLの理解は、データベースを利用した意思決定に不可欠です。データを収集、整理、分析し、意思決定に活かす能力は、リーダーシップ力を強化します。データ分析を通して、洞察を得ることことは、戦略的な意思決定のスキル向上につながります。

③ 自己成長と問題解決能力の向上

SQLを学ぶことは、問題解決能力の向上につながります。SQLを考えるプロセスは、論理的思考を養います。これらスキルの向上で、自己成長が促進されます。また、新しい技術やツールを学びたいという意欲向上につながります。

現代のデジタル化されたビジネス環境において価値があり、個人および組織の成功に寄与