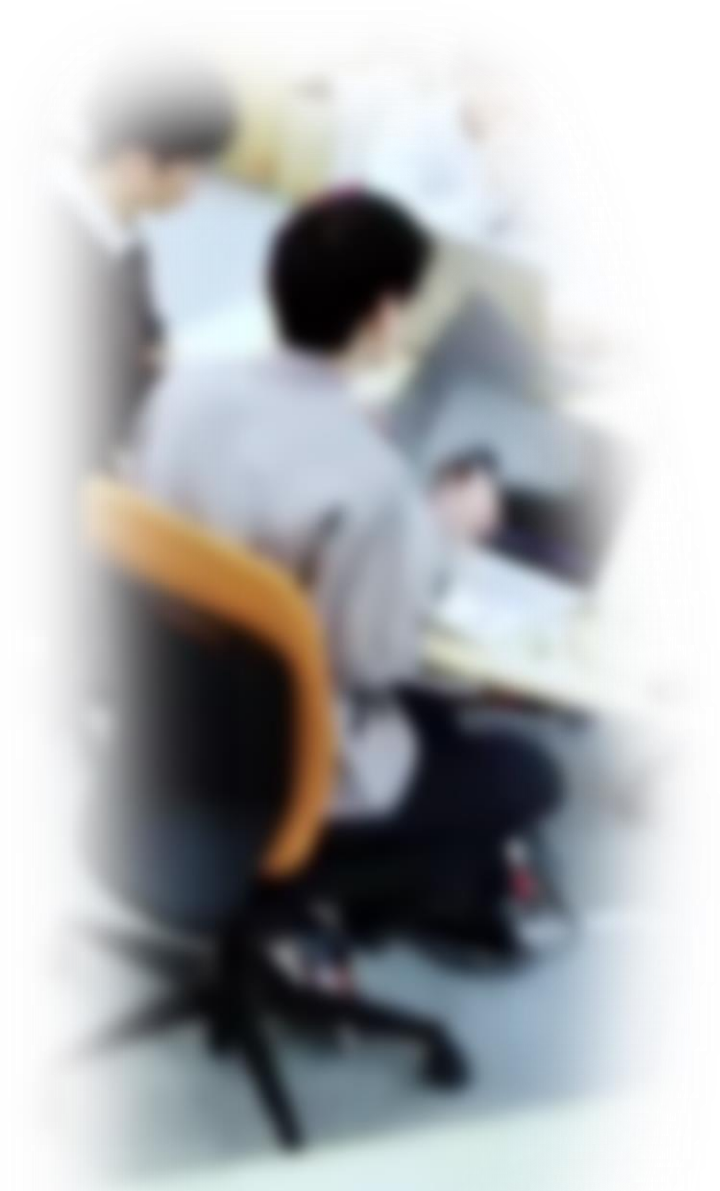
3. データベース設計と正規化

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. リレーショナルデータベースの概要と重要性
2. 冗長なデータの問題点
3. データベースの異状とその影響
4. 正規化の概念と重要性
5. テーブルの分割と結合の方法と理由

3-1. リレーショナルデータベースの概要と重要性

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

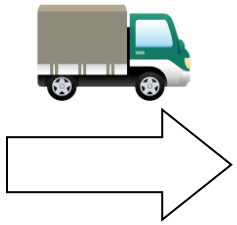
取引

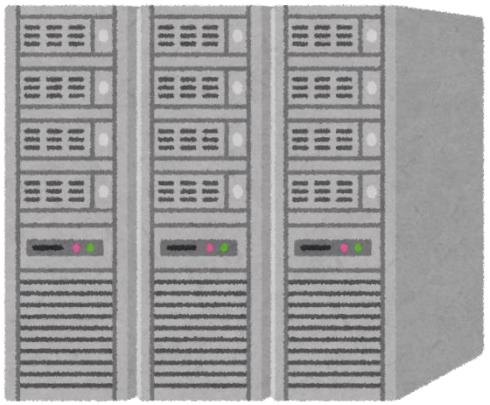
記入

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）

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

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

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

リレーショナルデータベースの重要性

1. **データの整合性**：リレーショナルデータベースは，**データの整合性を保持するための機能**を有する．これにより，誤ったデータや矛盾したデータが保存されるのを防ぐことができる．
2. **柔軟な問い合わせ（クエリ）能力**：リレーショナルデータベースのSQL（Structured Query Language）（データベース操作言語）の使用により，**複雑な検索やデータの抽出**が可能になる．
3. **トランザクション機能**：一連の操作全体を一つの単位として取り扱うことができる機能．これにより，**データの一貫性と信頼性が向上**する．
4. **セキュリティ**：**アクセス権限の設定**などにより，セキュリティを確保する．

データの安全な保管，効率的なデータ検索・操作，ビジネスや研究の意思決定をサポート．

3-2. 冗長なデータの問題点

冗長なデータ

このバスは運賃１０００円です

このバスは運賃１０００円です

冗長なデータ　＝　データベースとしてはNG

「更新の際、すべての個所を更新しないといけない」
などの問題がある

冗長なデータ

冗長なデータは、データベース内で**不必要に重複して保存されるデータ**を指す。

名前	昼食	料金
A	そば	250
B	**カレーライス**	**400**
C	**カレーライス**	**400**
D	うどん	250

冗長なデータ

- **そば**は、２５０円
- **カレーライス**は、４００円
- **うどん**は、２５０円
- **A**さんは**そば**を食べた
- **B**さんは**カレーライス**を食べた
- **C**さんは**カレーライス**を食べた
- **D**さんは**うどん**を食べた

冗長なデータの問題点

- **データの更新**の際，**全ての重複箇所を更新しなければならず**，それが**漏れる**と**データの不整合**が生じる．（**更新による不整合**）
- データベースのサイズが不必要に増大．
- データの検索やクエリの実行速度が遅くなる可能性．

正規化

目的

- **データベースの構造を最適化**
- **効率的なデータベース管理**を実現

方針

テーブルの数を減らすことよりも、**データの冗長性（重複）を減らす**ことを行う

冗長なデータの更新

更新

カレーライスが
400円から 350円に値下げ

名前	昼食	料金
A	そば	250
B	カレーライス	~~400~~ **350**
C	カレーライス	~~400~~ **350**
D	うどん	250

「400」をすべて「350」に変更

正規化前の問題

更新

カレーライスが
400円から 350円に値下げ

名前	昼食	料金
A	そば	250
B	カレーライス	~~400~~ **350**
C	カレーライス	400
D	うどん	250

「400」をすべて「350」に変更する必要があるが、変更を間違ったとする

昼食の値段が 1 つのはずなのに、350, 400 の違った値段の記録があり、不整合がある

冗長なデータの例①

生徒の名前、生徒が所属する**クラス、教科、成績**を記録するテーブル

生徒名	クラス	教科	成績
田中	3年A組	数学	85
田中	3年A組	英語	90
佐藤	3年B組	数学	88
佐藤	3年B組	英語	92

「生徒名」と「クラス」の情報
が冗長に繰り返されている

冗長なデータの例②

ID	氏名	部署	部屋
1234	AA	会計	A
2000	XX	窓口	B
2122	YY	総務	C
300	BB	開発	B
3400	EE	会計	A
3780	FF	窓口	B

会計ーA，窓口ーB　が複数個所にあるのが冗長

ここまでのまとめ

- **冗長なデータの定義**

不必要に**データベース内で重複して保存されるデータ**

- **冗長なデータ**により生じる主な**問題**：

更新時の不整合：**全ての重複箇所を更新しないとデータの不整合が生じる.**

例：書き換え漏れにより，**朝食の値段が350円と400円の2つの異なる価格**で記録．**不整合**が生じている.

演習．冗長なデータ、データの不整合

演習① 冗長なデータの発見

- 「商品注文データベース」の「注文」テーブルです。顧客名、住所、商品名、価格、購入日などの属性があります。
- **同じ顧客からの異なる注文**で**顧客名や住所が繰り返し登録されている**ことを確認してください。それはどこですか？

注文ID	顧客名	住所	商品名	価格	購入日
1	田中太郎	東京都中央区1-1-1	テレビ	50000	2024-10-01
2	田中太郎	東京都中央区1-1-1	冷蔵庫	100000	2024-10-02
3	山田花子	大阪府北区2-2-2	洗濯機	30000	2024-10-03
4	田中太郎	東京都中央区1-1-1	掃除機	15000	2024-10-04

ヒント

- まず、**顧客名の列**を順に見て、**同じ顧客名**が**複数回出現**するかどうかを確認します。
- 次に、**同じ顧客名の行**を見て、**住所が同じ**であることを確認します。
- 顧客名や住所が複数回登録されている場合、その**データは冗長である**と言えます。

注文ID	顧客名	住所	商品名	価格	購入日
1	田中太郎	東京都中央区1-1-1	テレビ	50000	2024-10-01
2	田中太郎	東京都中央区1-1-1	冷蔵庫	100000	2024-10-02
3	山田花子	大阪府北区2-2-2	洗濯機	30000	2024-10-03
4	田中太郎	東京都中央区1-1-1	掃除機	15000	2024-10-04

解答例

注文ID 1, 2, 4において、顧客「田中太郎」の住所「東京都中央区1-1-1」が繰り返し登録されています。これは冗長なデータです。

注文ID	顧客名	住所	商品名	価格	購入日
1	**田中太郎**	**東京都中央区1-1-1**	テレビ	50000	2024-10-01
2	**田中太郎**	**東京都中央区1-1-1**	冷蔵庫	100000	2024-10-02
3	山田花子	大阪府北区2-2-2	洗濯機	30000	2024-10-03
4	田中太郎	東京都中央区1-1-1	掃除機	15000	2024-10-04

演習② データの不整合の確認

- 在庫管理データベースです。**商品の価格情報**が、**商品テーブル**と**注文履歴テーブル**の2つの場所に保存されています。
- 価格の不一致はどこで起きていますか？

商品テーブル

商品ID	商品名	価格
1	テレビ	50000
2	冷蔵庫	100000
3	洗濯機	30000

注文履歴テーブル

注文ID	商品ID	商品名	価格	注文日
A	1	テレビ	50000	2024-10-01
B	2	冷蔵庫	95000	2024-10-02
C	3	洗濯機	31000	2024-10-03

ヒント

商品テーブルと注文履歴テーブルの同じ**商品ID**に基づいて、価格の一致を確認します。

商品テーブル

商品ID	商品名	価格
1	テレビ	50000
2	冷蔵庫	100000
3	洗濯機	30000

注文履歴テーブル

注文ID	商品ID	商品名	価格	注文日
A	1	テレビ	50000	2024-10-01
B	2	冷蔵庫	95000	2024-10-02
C	3	洗濯機	31000	2024-10-03

解答例

- 注文ID「B」の「冷蔵庫」の価格が、商品テーブルと異なっています（100,000 vs. 95,000）。
- 注文ID「C」の「洗濯機」の価格も、商品テーブルと異なっています（30,000 vs. 31,000）。

商品テーブル

商品ID	商品名	価格
1	テレビ	50000
2	**冷蔵庫**	**100000**
3	**洗濯機**	**30000**

注文履歴テーブル

注文ID	商品ID	商品名	価格	注文日
A	1	テレビ	50000	2024-10-01
B	2	**冷蔵庫**	**95000**	2024-10-02
C	3	**洗濯機**	**31000**	2024-10-03

3-3. データベースの異状とその影響

異状とは

- **異状**とは，**データベースの設計が不適切**な場合に，**データの操作**（追加，更新，削除）時に起こる**予期しない問題や振る舞い**のことを指す．

- **不適切に設計されたデータベース**では，**データの冗長性**が生じることが多く，これが**異状の原因**となる．（「異常」ではありません）

異状の具体例

ペット名	オーナー	住所
ミミ	徳川家康	東京都中央区1-1-1
タロウ	徳川家康	東京都中央区1-1-1

- これは、動物病院のデータベース。**徳川家康**さんの**２匹のペット**の記録がある。
- 徳川家康さんが**引っ越す**と、**２か所の変更**が必要
  → 不便、書き換え漏れによる**不整合の危険**
  → データベースの設計が不適切

異状の問題点

- データの**不整合のリスク**が高まる.
- 不整合が生じると，**情報の信頼性が失われ，誤った意思決定や業務処理のミス**を引き起こす可能性がある.
- 大きな問題となる.

不整合が起きている例

顧客名	**お気に入りのドリンク**	**価格**
田中	**カフェラテ**	**400円**
田中	カプチーノ	450円
佐藤	**カフェラテ**	**420円**

データベース内に異なる価格が混在し、どちらが正しいのかが不明確になっている

不整合が起きている例①

田中さんのクラスが変更になった。書き換え漏れにより、**所属クラス情報の矛盾が発生**

生徒名	クラス	教科	成績
田中	3年A組	数学	85
田中	3年D組	英語	90
佐藤	3年B組	数学	88
佐藤	3年B組	英語	92

不整合が起きている例②

- **新しい生徒**が転入した。成績が無いので、行全体を完成できない。

生徒名	クラス	教科	成績
田中	3年A組	数学	85
田中	3年A組	英語	90
佐藤	3年B組	数学	88
佐藤	3年B組	英語	92
鈴木	**3年B組**		

不整合が起きている例③

- 佐藤さんの英語と数学の成績が**取り消し**になった。成績が無いので、行全体を完成できない。

生徒名	クラス	教科	成績
田中	3年A組	数学	85
田中	3年A組	英語	90
佐藤	**3年B組**		
佐藤	**3年B組**		

異状のまとめ

異状とは

- **データベース操作（追加、更新、削除）時の予期しない問題**
- 主に**データベースの不適切な設計**から生じる
- **冗長性の存在**が、異状の主な原因

異状の具体例

- 動物病院のデータベースにおいて、徳川家康さんの住所と、２匹のペットの記録
- 住所変更時、**２つのレコードを別々に更新する必要**がある。
  ＝　予期しない問題。**書き換え漏れのリスク**

異状の問題点

- データの不整合のリスク増加

3-4. 正規化の概念と重要性

リレーショナルデータベースの設計における考慮事項

次のことを考慮して設計を行う

- **冗長なデータの排除**を行う**正規化**
- **データの整合性を保証**する**制約**
  - 例：関連するテーブル間の参照に関する制約
  - 例：同じ値が２度現れないという制約
- **データの検索や操作の効率化**：**索引、データベースの構造の簡素化**

正規化とその重要性

- **正規化**とは，リレーショナルデータベースの**テーブルを適切な形に再構成**することで，**データの冗長性を排除**し，**データの整合性を向上**させるプロセス.
- **正規化**を適切に実施することで**データの不整合を防ぐ**ことができる.
- **正規化**は，**データベース設計において欠かせないステップ**です.

正規化の例

正規化前

名前	昼食	料金
A	そば	250
B	**カレーライス**	**400**
C	**カレーライス**	**400**
D	うどん	250

冗長なデータがある

正規化後

名前	昼食
A	そば
B	カレーライス
C	カレーライス
D	うどん

昼食	料金
そば	250
カレーライス	400
うどん	250

冗長なデータがない

正規化により、元のテーブルにあった**冗長性を排除.**

正規化前の問題

更新

カレーライスが
400円から 350円に値下げ

名前	昼食	料金
A	そば	250
B	カレーライス	~~400~~ **350**
C	カレーライス	400
D	うどん	250

「400」をすべて「350」に変更する必要があるが、変更を間違ったとする

昼食の値段が１つのはずなのに、350, 400 の違った値段の記録があり、不整合がある

正規化によるデータの不整合の防止

名前	朝食
A	カレーライス
B	うどん
C	カレーライス

朝食	値段
カレーライス	~~400~~ 350
うどん	250

テーブル

カレーライスが
400円から 350円に値下げ

データの不整合はない

正規化による問題解消

更新

カレーライスが
400円から 350円に値下げ

名前	昼食
A	カレーライス
B	うどん
C	カレーライス

昼食	値段
カレーライス	~~400~~
うどん	250

350

「400」をすべて「350」に変更するのは1か所で済む。
間違いが起きにくい。

正規化による冗長なデータの排除の例①

生徒名	クラス	教科	成績
田中	3年A組	数学	85
田中	3年A組	英語	90
佐藤	3年B組	数学	88
佐藤	3年B組	英語	92

正規化

生徒ID	生徒名	クラス
1	田中	3年A組
2	佐藤	3年B組

生徒ID	教科	成績
1	数学	85
1	英語	90
2	数学	88
2	英語	92

正規化により、元のテーブルにあった**冗長性を排除.**

正規化による冗長なデータの排除の例②

ID	氏名	部署	部屋
1234	AA	会計	A
2000	XX	窓口	B
2122	YY	総務	C
300	BB	開発	B
3400	EE	会計	A
3780	FF	窓口	B

正規化

ID	氏名	部署
1234	AA	会計
2000	XX	窓口
2122	YY	総務
300	BB	開発
3400	EE	会計
3780	FF	窓口

部署	部屋
会計	A
窓口	B
総務	C
開発	B

データの冗長性が排除され、更新における不整合のリスクが低減

正規化のメリットとデメリット

正規化のメリット

- データの**冗長性を排除**
- データの**整合性が向上**
- 更新、削除、挿入時の**異状を減少**

正規化のデメリット

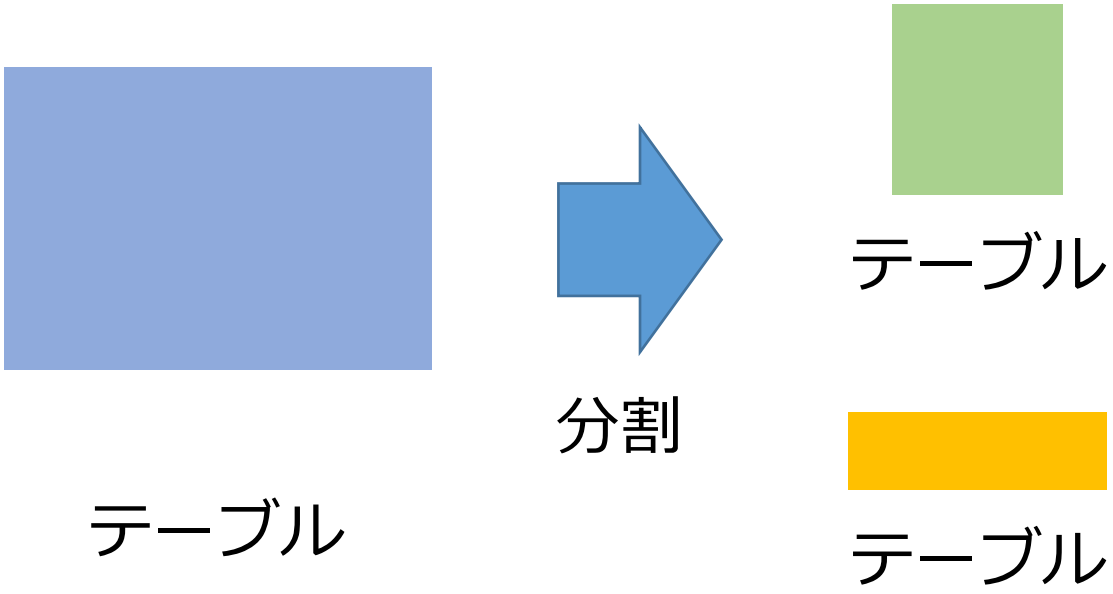
- **過度**に正規化されたデータベースでは、テーブルの数が多くなり、利用が複雑になる場合がある。性能上の問題が発生する可能性もある。

正規化のまとめ

- **正規化**：リレーショナルデータベースのテーブル再構成
- **目的**：データの冗長性排除、整合性向上
- **結果**：データの不整合防止
- **位置付け**：データベース設計の必須ステップ

3-5.テーブルの分割と結合の方法と理由

テーブル分割



テーブル分割により、１つのテーブルが２つ以上のテーブルに分割される。

テーブル分割を行う理由

① **冗長なデータを排除**する**正規化**を行う

② **より小さなテーブル**に分割することで、**問い合わせ（クエリ）の性能を向上**させる

テーブル分割を行う SQL

```
SELECT DISTINCT ID, 商品名, 単価
INTO A FROM 購入記録;
```

ID	商品名	単価	購入者
1	みかん	50	aa
1	みかん	50	bb
2	りんご	100	cc
3	メロン	500	dd

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

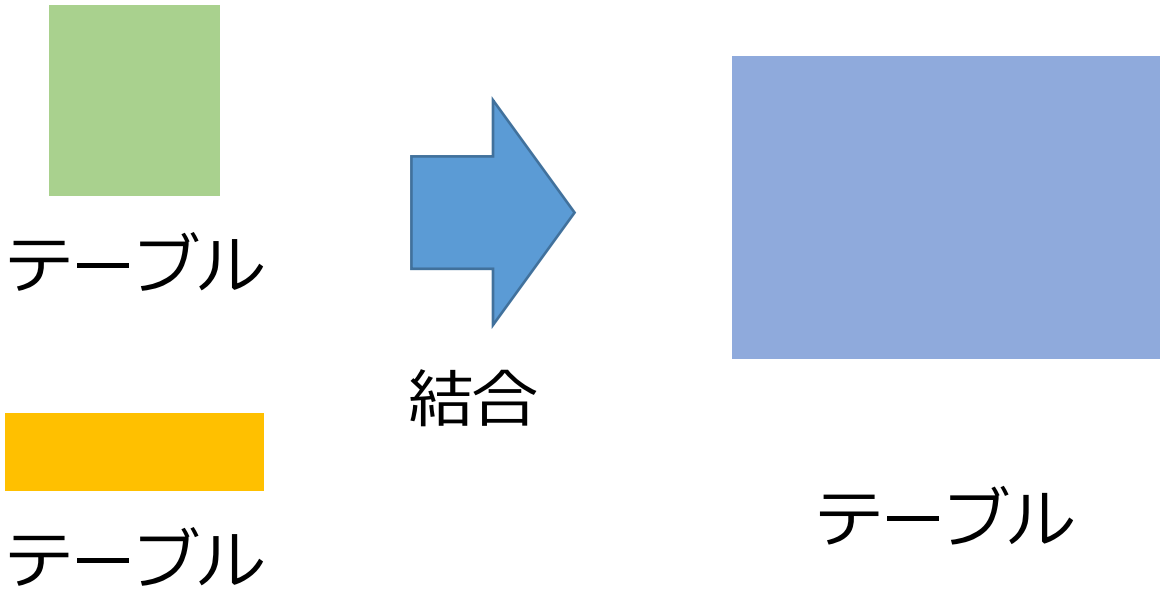
購入者	ID
aa	1
bb	1
cc	2
dd	3

```
SELECT DISTINCT 購入者, ID
INTO B FROM 購入記録;
```

2 つの SQL の実行により、テーブル分割を行っている

テーブル結合

結合は，テーブル間の関連に基づいて**複数のテーブル**を**1つにまとめる操作**
例：**従業員テーブル**と**部署テーブル**を結合，従業員の名前と所属部署の名前を**１つのテーブル**に集める．



テーブル結合を行う理由

- 関連のある複数のテーブルを組み合わせて１つにするため
- 特に、正規化のために分割されたテーブルをもとに戻すため

テーブル結合を行う SQL

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

購入者	ID
aa	1
bb	1
cc	2
dd	3

```
SELECT A.ID, 商品名, 単価, 購入者
FROM A, B
WHERE A.ID = B.ID;
```

ID	商品名	単価	購入者
1	みかん	50	aa
1	みかん	50	bb
2	りんご	100	cc
3	メロン	500	dd

テーブルの分割と結合に役立つ SQL コマンド

- DISTINCT ・・・ 重複行除去
- INTO ・・・　Access だけの機能．SQL の結果をテーブルに保存

テーブルの分割と結合

```
SELECT DISTINCT ID, 商品名, 単価
INTO A FROM 購入記録;
```

ID	商品名	単価	購入者
1	みかん	50	aa
1	みかん	50	bb
2	りんご	100	cc
3	メロン	500	dd

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

購入者	ID
aa	1
bb	1
cc	2
dd	3

```
SELECT DISTINCT 購入者, ID
INTO B FROM 購入記録;
```

```
SELECT A.ID, 商品名, 単価, 購入者
FROM A, B
WHERE A.ID = B.ID;
```

ID	商品名	単価	購入者
1	みかん	50	aa
1	みかん	50	bb
2	りんご	100	cc
3	メロン	500	dd

テーブルの分割と結合により、もとに戻る場合がある

全体まとめ ①

リレーショナルデータベースの重要性

- **データの整合性**：誤ったデータや矛盾したデータの保存を防ぐ。
- **柔軟な問い合わせ能力**：SQLを使用し、複雑な検索やデータ抽出が可能。
- **トランザクション**：一連の操作を一つの単位として扱う機能。
- **セキュリティ**：アクセス権限の設定などで確保。

冗長なデータの問題点

- 更新の際、全ての個所を更新する必要性。

異状とは

- データベース操作時の予期しない問題。
- **主にデータベースの不適切な設計**により、**冗長なデータが発生**してしまうことが**主な原因**

全体まとめ ②

正規化とその重要性

- 正規化は、テーブルを適切な形に再構成するプロセス。
- 冗長性の排除とデータ整合性の向上が目的。
- データベース設計の必須ステップ。

テーブル分割の理由

- 正規化のための冗長なデータの排除。
- 問い合わせの性能向上。

テーブル結合の理由

- 関連する複数テーブルの組み合わせ。
- 正規化で分割されたテーブルの再結合。

① データの整合性の保持

- テーブルを正規化することで、**データの冗長性を排除**し、**整合性を高める**ことが可能となります。
- この学習を通じて、データ管理における**新しい視角や考え方**を獲得できます。

② データベース設計スキル

- データベース設計の基本を学ぶことは、多様なデータベースシステムの設計や運用能力に直結します。
- このとき、データの冗長性、異状、正規化について理解しておくことは有用です。

③ 問題解決能力

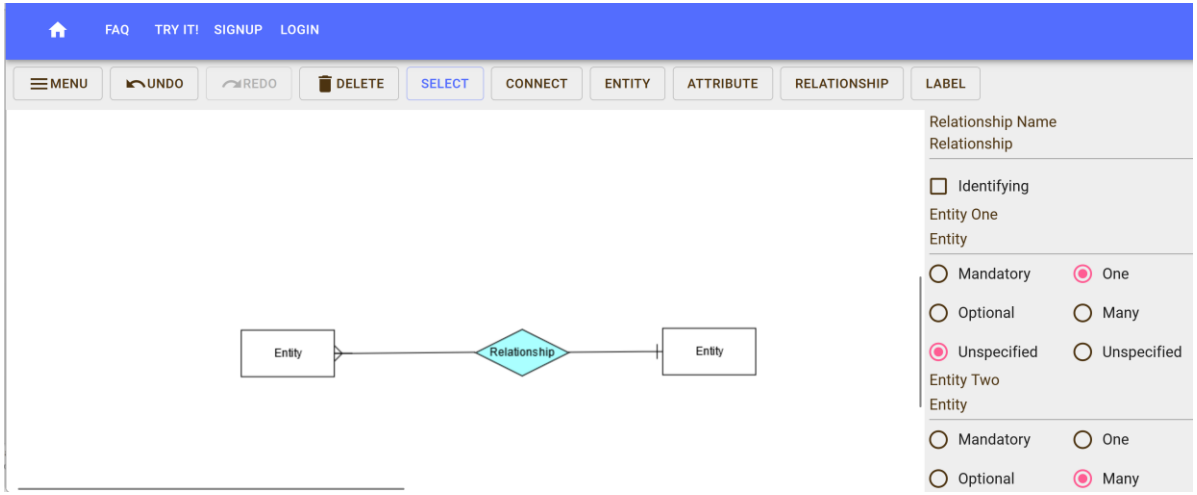
データの整合性やデータベース設計スキルを深化させることで、**現実の問題解決能力**が向上します。

　学びの過程は、新しい発見や視野の広がり、さらにはそれらがもたらす実益と直結しています。

自習 ER図の作図サイト

ERDPlus は、ER図などを作成するためのオンラインツールです

https://erdplus.com/standalone



ITパスポート試験対策などで、すでに自分でER図のことを学んだ経験がある人に向いた自習です。
ER図については、授業の別の回で説明するので待っていてください。
（提出する必要はありません。）