

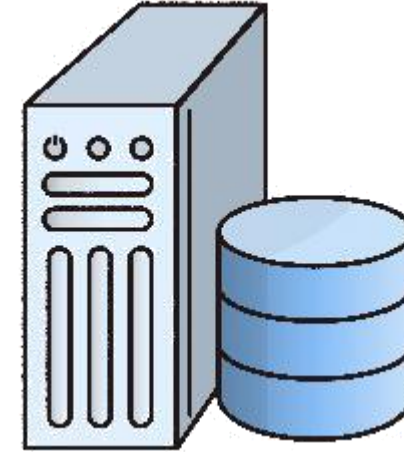
Information Technology Examination

データベース

担当：新田

データベース

決まったデータ構造で整理された
データの集まりのこと



DBMS (Database Management System)

- データベース管理システム
- データの作成、整理、更新、取得、管理などを効率的かつ安全に行うためのソフトウェアシステム

例) MySQL、OracleDB、PostgreSQL、SQLite、MongoDB、Microsoft Access、Microsoft SQL Server など

データベース管理システムの種類

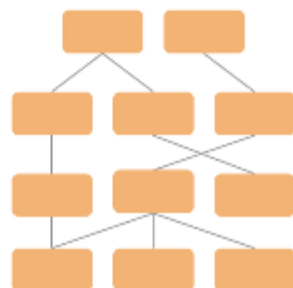
①階層型



ツリー構造で蓄積

- ◎ 検索が速い
- × 重複登録・管理

②ネットワーク型

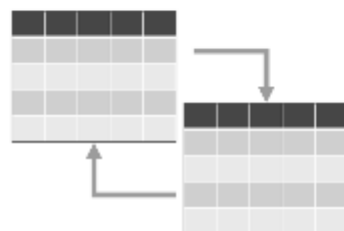


網目状で蓄積

- ◎ 重複登録・管理を避けられる
- × 管理の複雑化
操作の難易度高

③RDB

(リレーショナルデータベース)



表形式で蓄積・管理

- ◎ 処理の一貫性
重複登録の排除
- × プログラムが複雑
処理速度の低下

④NoSQL

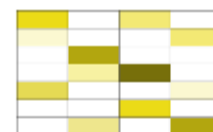
ドキュメント型



キー値型



ワイドカラム型



グラフ型



さまざまな形式でそのまま格納

- ◎ 高速な処理
サーバー数非依存でスケールアウト
- × データ一貫性の担保が困難
データ検索精度の低下

関係データベース (Relational Database)

表の形で管理するデータベース

テーブル

ID	名前	住所	電話番号

カラム

フィールド

レコード

■ テーブル

データを保存しておくための「表」

■ カラム

テーブルの「列」

■ レコード

テーブルの「行」

■ フィールド

レコードの中にあるひとつの「要素」

※テーブルの「列」をフィールドと呼ぶ場合もある

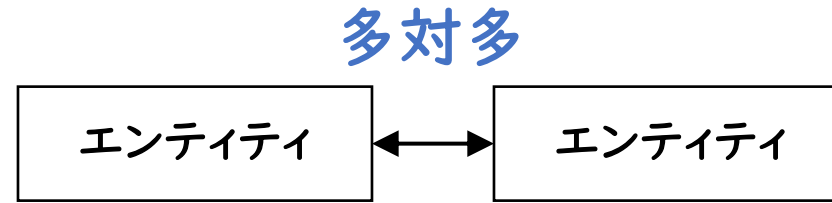
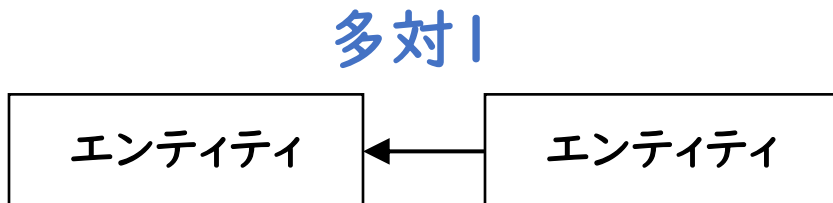
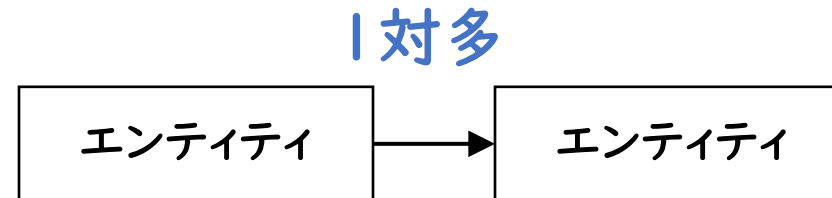
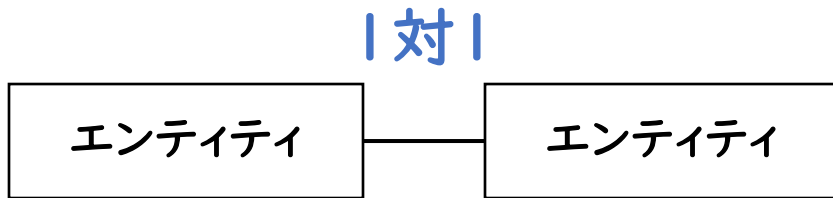
関係データベースを定義したり操作するための言語を**SQL**と言う。

データベース設計

E-R図 (Entity Relationship Diagram)

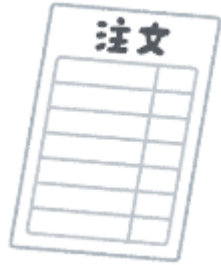
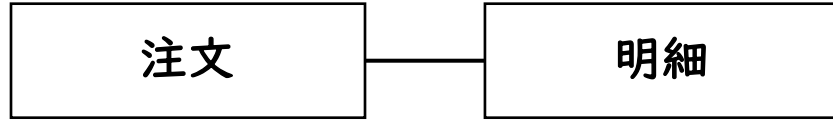
- 実体 (Entity) 同士の関係 (Relation) を示した図
- データベースの設計図としてよく使われる。

エンティティ間の対応する数の関係 ⇒ **多重度** (カーディナリティ)

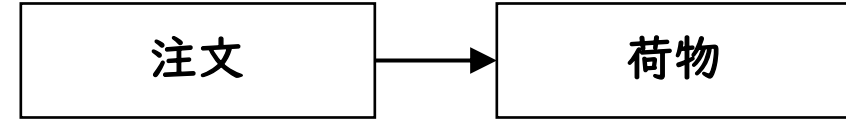


某AMAZON

1対1

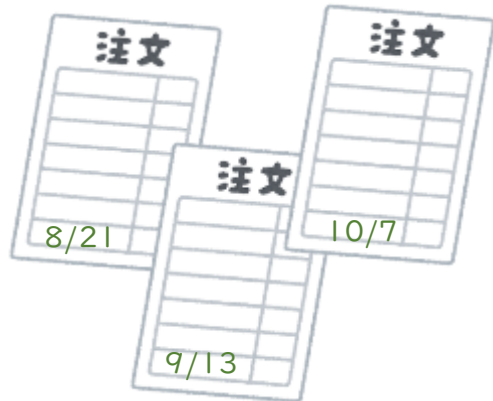
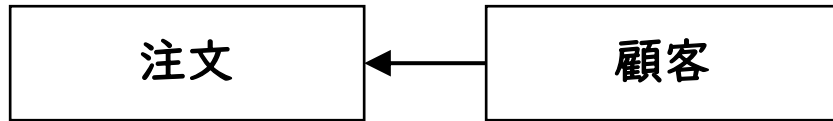


1対多

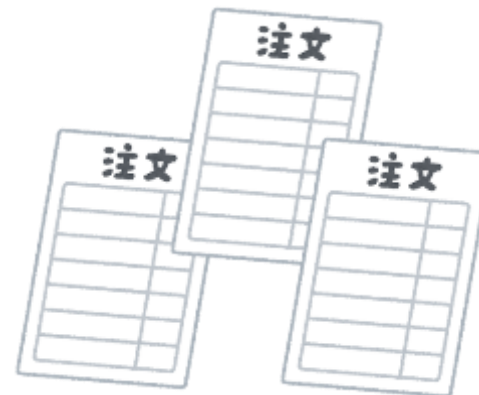
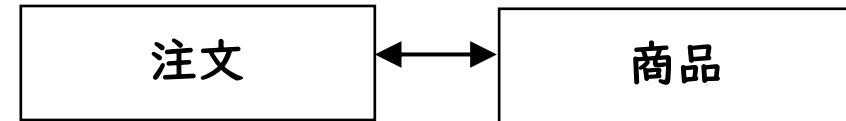


1つに特定できるかどうか？

多対1



多対多



テーブルには制約（決まり事）がある

- 主キー（PRIMARY KEY）制約
- 外部キー（FOREIGN KEY）制約
- NOT NULL制約
- UNIQUE制約
- CHECK制約



主キー (Primary Key)

- テーブルのレコード(行)を一意に識別するための項目
全ての行において、絶対に重複しない
- 通し番号のような項目を設けて、設定することが多い。
- 値を空欄(NULL)にすることができない。

例) 社員番号、商品ID、従業員コード、支店番号、

主キー

社員番号	氏名
10001	田中一郎
10002	高橋二郎
10003	佐藤三郎
10004	鈴木四郎
10005	山田五郎

- 複数の列を組み合わせて一意になる列を主キーとする → 複合主キー

例) クラス + 出席番号 ⇒ 1人の生徒

単独では主キーになれないが、
2つ合わせれば一意になる。

複合主キー

クラス	出席番号	名前
1年A組	1	加藤良子
1年A組	2	伊藤優子
1年B組	1	斉藤京子

主キーがないと...

名前	性別	年齢	出身
:	:	:	:
鈴木	男	44	千葉
鈴木	男	33	東京
鈴木	男	28	千葉
鈴木	女	30	千葉
鈴木	女	28	千葉
鈴木	女	28	神奈川
鈴木	女	24	埼玉
:	:	:	:



鈴木さんの年齢教えて

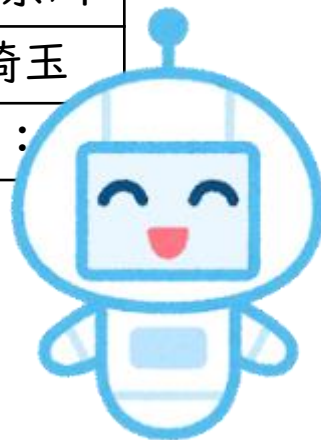
鈴木さん、結婚されたから名前変更して

鈴木さん、退職されたから消して

- どのレコードを指しているのか分からないと検索・更新・削除ができない。
- レコードを特定するために必要な情報が多くなる。
(名前、性別、年齢、出身)
- すべての列の内容が同じレコードがあった場合、識別が不可能。

主キー (ID) があれば...

ID	名前	性別	年齢	出身
:	:	:	:	:
03024	鈴木	男	44	千葉
05211	鈴木	男	33	東京
06238	鈴木	男	28	千葉
03193	鈴木	女	30	千葉
05088	鈴木	女	28	千葉
05090	鈴木	女	28	神奈川
07142	鈴木	女	24	埼玉
:	:	:	:	:



ID03024の年齢教えて

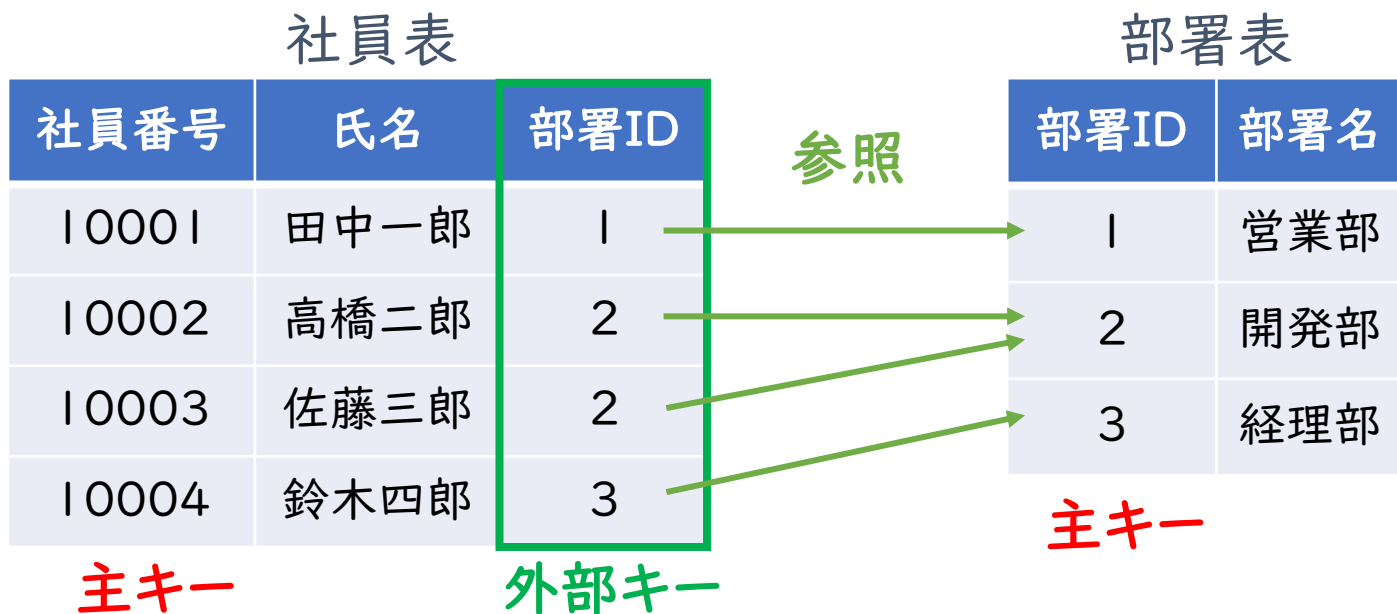
ID05088の名前変更して

ID06238を削除して

- 簡単に1つのレコードを指定することができる。
- すべての列の項目が同じであったとしても、IDは必ず違う値を割り当てるため、識別が可能。

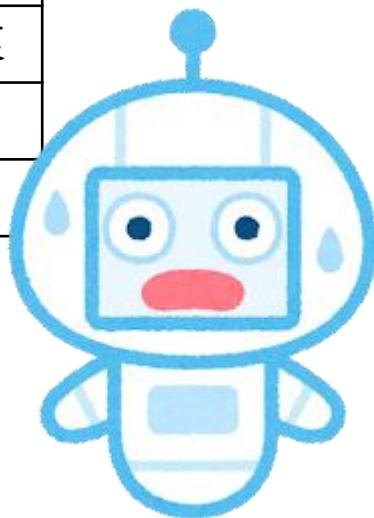
外部キー (Foreign Key)

- 複数のテーブルの関係を結び付ける列。
- データの参照先と参照元で親子関係が成り立つ。
- 親テーブルに存在しないデータを子テーブルに持つことがない。
⇒ データの整合性が保たれる。



外部キーがないと...

ID	名前	部署名
10001	田中	総務
10002	鈴木	人事
10003	斎藤	経理
:	:	:
:	:	:
70890	山田	第1営業
70891	中村	開発
70892	高橋	営業



部署名が変更になったから、該当する人の部署名を変更して！

部署名欄が自由入力だったから、名称間違えちゃった。てへぺろ。修正しておいて。

- 膨大な量のデータを書き換える必要がある。
- 意図しない(存在しない)データが入力される可能性がある。

外部キーがあれば...

ID	名前	部署ID
10001	田中	101
10002	鈴木	103
10003	斎藤	102
:	:	:
:	:	:
70890	山田	108
70891	中村	105
70892	高橋	108

外部キー

部署ID	部署名
101	総務
102	経理
103	人事
:	:
:	:



- データの変更は1か所で済む。
- 意図しない(存在しない)データが入力されることがない。(整合性が担保される。)

正規化

データベースを設計する際に、データの重複や矛盾を排除する目的でデータ同士の関連を保ったまま、**複数の表に分解**していく。

第1正規化

- 属性値に含まれる繰り返し項目をなくす

第2正規化

- 主キーが複数の項目の組み合わせからなる複合主キーの時、主キーの一部の項目によって一意に決まる項目を別表に移す

第3正規化

- 主キー以外の項目によって一意に決まる項目を別表に移す

正規化前

注文番号	注文日	ユーザID	購入 ユーザ名	商品ID	商品名	単価	数量	商品ID	商品名	単価	数量	商品ID	商品名	単価	数量
T001	2021/1/11	U001	鈴木一郎	S001	A商品	1000	1	S002	B商品	500	2				
T002	2021/1/20	U002	佐藤次郎	S003	C商品	2000	1								
T003	2021/1/23	U003	田中三郎	S004	D商品	1000	2	S005	E商品	5000	1	S006	F商品	1500	1

第1正規化後

【第1正規形】

注文番号 (主キー)	注文日	ユーザID	購入 ユーザ名	商品ID (主キー)	商品名	単価	数量
T001	2021/1/11	U001	鈴木一郎	S001	A商品	1000	1
T001	2021/1/11	U001	鈴木一郎	S002	B商品	500	2
T002	2021/1/20	U002	佐藤次郎	S003	C商品	2000	1
T003	2021/1/23	U003	田中三郎	S004	D商品	1000	2
T003	2021/1/23	U003	田中三郎	S005	E商品	5000	1
T003	2021/1/23	U003	田中三郎	S006	F商品	1500	1

繰り返し項目をなくす
⇒ 凸凹がない、長方形の表にする

第2正規化前

【部分関数従属】

注文番号 (主キー)	注文日	ユーザID	購入 ユーザ名	商品ID (主キー)	商品名	単価	数量
T001	2021/1/11	U001	鈴木一郎	S001	A商品	1000	1
T001	2021/1/11	U001	鈴木一郎	S002	B商品	500	2
T002	2021/1/20	U002	佐藤次郎	S003	C商品	2000	1
T003	2021/1/23	U003	田中三郎	S004	D商品	1000	2
T003	2021/1/23	U003	田中三郎	S005	E商品	5000	1
T003	2021/1/23	U003	田中三郎	S006	F商品	1500	1

第2正規化後

注文テーブル

注文番号 (主キー)	注文日	ユーザID	購入 ユーザ名
T001	2021/1/11	U001	鈴木一郎
T002	2021/1/20	U002	佐藤次郎
T003	2021/1/23	U003	田中三郎

注文詳細テーブル

注文番号 (主キー)	商品ID (主キー)	数量
T001	S001	1
T001	S002	2
T002	S003	1
T003	S004	2
T003	S005	1
T003	S006	1

商品テーブル

商品ID (主キー)	商品名	単価
S001	A商品	1000
S002	B商品	500
S003	C商品	2000
S004	D商品	1000
S005	E商品	5000
S006	F商品	1500

関数従属しているカラム(列)
を切り出す

第3正規化前

【関数従属】

注文番号 (主キー)	注文日	ユーザID	購入 ユーザ名
T001	2021/1/11	U001	鈴木一郎
T002	2021/1/20	U002	佐藤次郎
T003	2021/1/23	U003	田中三郎

主キー以外で、関数従属している
カラム(列)を切り出す

第3正規化後

【第3正規形】

ユーザテーブル

ユーザID (主キー)	購入 ユーザ名
U001	鈴木一郎
U002	佐藤次郎
U003	田中三郎

注文テーブル

注文番号 (主キー)	注文日	ユーザID
T001	2021/1/11	U001
T002	2021/1/20	U002
T003	2021/1/23	U003

注文詳細テーブル

注文番号 (主キー)	商品ID (主キー)	数量
T001	S001	1
T001	S002	2
T002	S003	1
T003	S004	2
T003	S005	1
T003	S006	1

商品テーブル

商品ID (主キー)	商品名	単価
S001	A商品	1000
S002	B商品	500
S003	C商品	2000
S004	D商品	1000
S005	E商品	5000
S006	F商品	1500

関数従属性

「〇〇の値が分かれば、△△の値が導き出せる」というような関係

例) 商品IDが分かれば、商品名、単価が導き出せる。

⇒ 商品テーブルを作って、分ける。

関数従属性がある部分は、別の表に分ける

※注意

分解した表から、必ず元の表に戻すことができること(情報無損失分解)

例)

関数従属



空港コードが分かれば、
空港名、所在地は分かる

関数従属



商品IDが分かれば、
商品名は分かる

空港コード	空港名	所在地	商品ID	商品名	個数
HND	羽田空港	東京都	301	りんご	100
HND	羽田空港	東京都	303	いちご	200
CTS	新千歳空港	北海道	325	メロン	50
NRT	成田空港	千葉県	303	いちご	150
ITM	伊丹空港	兵庫県	305	みかん	130
FUK	福岡空港	福岡県	305	みかん	70
FUK	福岡空港	福岡県	321	すいか	90

別の表に分ける

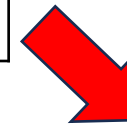
空港コード	商品ID	個数
HND	301	100
HND	303	200
CTS	325	50
NRT	303	150
ITM	305	130
FUK	305	70
FUK	321	90



空港テーブル

分解

空港コード	空港名	所在地
CTS	新千歳空港	北海道
HND	羽田空港	東京都
NRT	成田空港	千葉県
ITM	伊丹空港	兵庫県
FUK	福岡空港	福岡県



商品テーブル

分解

商品ID	商品名
301	りんご
303	いちご
305	みかん
321	すいか
325	メロン

データベース操作

テーブルから目的のデータを取り出す操作

関係演算

- 選択
- 射影
- 結合
- 商

集合演算

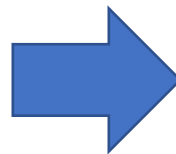
- 和
- 積
- 差
- 直積

選択(Projection)

行を取り出す演算。特定の条件に合致する行だけ抽出する。

部署IDが2の行だけ取り出す

社員番号	氏名	部署ID
10001	田中一郎	1
10002	高橋二郎	2
10003	佐藤三郎	2
10004	鈴木四郎	3
10005	山田五郎	2



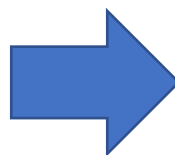
社員番号	氏名	部署ID
10002	高橋二郎	2
10003	佐藤三郎	2
10005	山田五郎	2

射影(Selection)

列を取り出す演算。特定の条件に合致する列だけ抽出する。

社員番号と氏名の列だけ取り出す

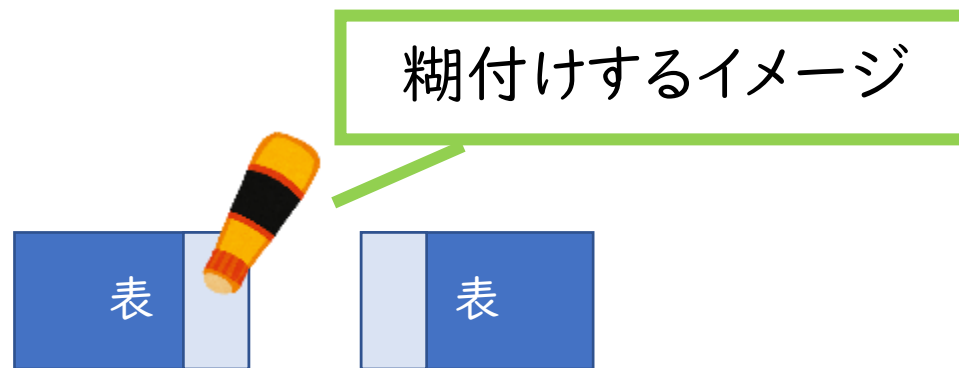
社員番号	氏名	部署ID
10001	田中一郎	1
10002	高橋二郎	2
10003	佐藤三郎	2
10004	鈴木四郎	3
10005	山田五郎	2



社員番号	氏名
10001	田中一郎
10002	高橋二郎
10003	佐藤三郎
10004	鈴木四郎
10005	山田五郎

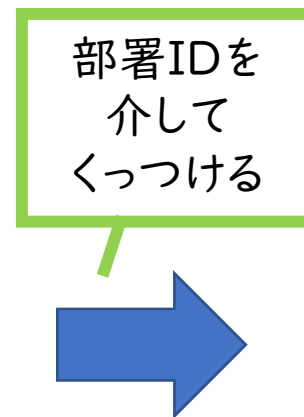
結合(JOIN)

表と表をくっつける演算。表の中にある共通の列を介して2つの表をつなぎあわせる。



社員番号	氏名	部署ID
10001	田中一郎	1
10002	高橋二郎	2
10003	佐藤三郎	2
10004	鈴木四郎	3

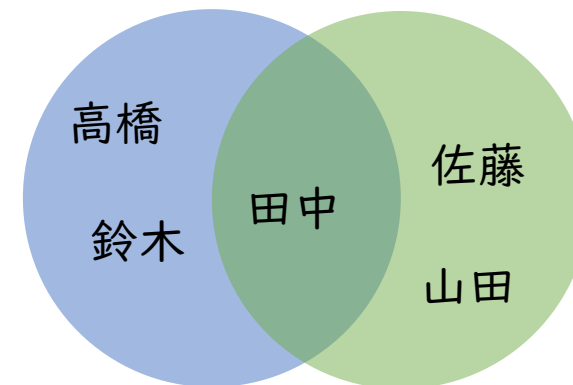
部署ID	部署名
1	営業部
2	開発部
3	経理部



社員番号	氏名	部署ID	部署名
10001	田中一郎	1	営業部
10002	高橋二郎	2	開発部
10003	佐藤三郎	2	開発部
10004	鈴木四郎	3	経理部

積

2つの表に共通している行を取り出す。



積集合

社員番号	氏名	部署ID
10001	田中一郎	1
10002	高橋二郎	2
10004	鈴木四郎	3

社員番号	氏名	部署ID
10001	田中一郎	1
10003	佐藤三郎	2
10005	山田五郎	2

2つの共通している
行を取り出す



社員番号	氏名	部署ID
10001	田中一郎	1

データベース制御

同時実行制御（排他制御）

- データへの同時アクセスによる矛盾の発生を防止し、データの一貫性を保つ。
- 1人のユーザーが更新処理を行っている間、他のユーザは更新処理の実行を待たせることで同時に更新処理が実行されることを防ぎ、データの不整合が発生しないようにする。
 - 共有ロック
 - 占有ロック

共有ロック

各ユーザは読むことはできるが書くことはできない



ロックして参照中
自分も読むだけ

読むことはできる

占有ロック

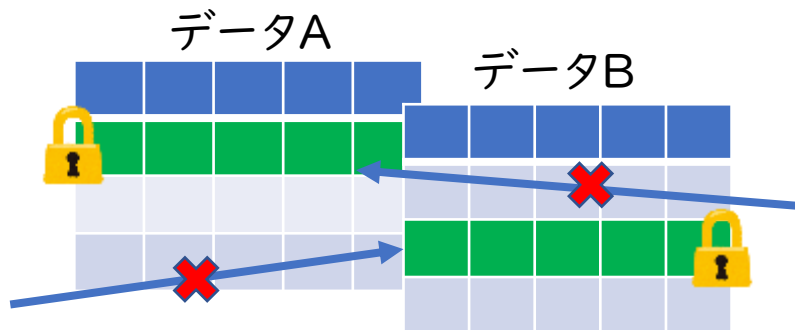
他のユーザは読むことも書くこともできない



ロックして編集中
自分は読み書き可能

読むこともできない

デッドロック



ロック解除待ち



ロック解除待ち

複数のトランザクションがお互いに使いたいデータをロック



お互いが**ロック解除待ち**となり
処理が進行できない。

ACID特性

Atomicity(原子性)

- トランザクションの処理結果は「すべて実行される」か「まったく実行されない」のいずれかで終了すること。

Consistency(一貫性)

- データベースの内容が矛盾のない状態であること。

Isolation(隔離性)

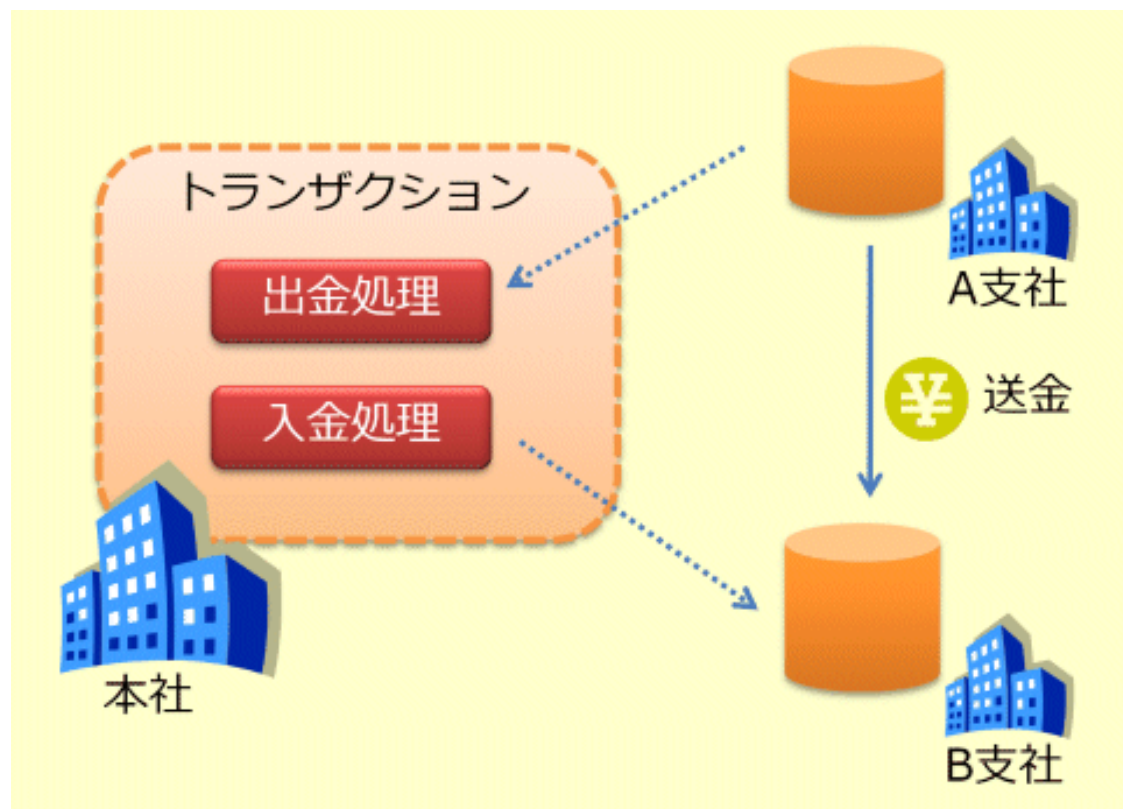
- トランザクション実行中の処理過程がほかの処理などに影響を与えないこと。

Durability(耐久性)

- トランザクションの更新結果は障害が生じてでも失われないこと。復旧手段が保証されていること。

トランザクション

一連の操作を全体として、1つの処理として管理する



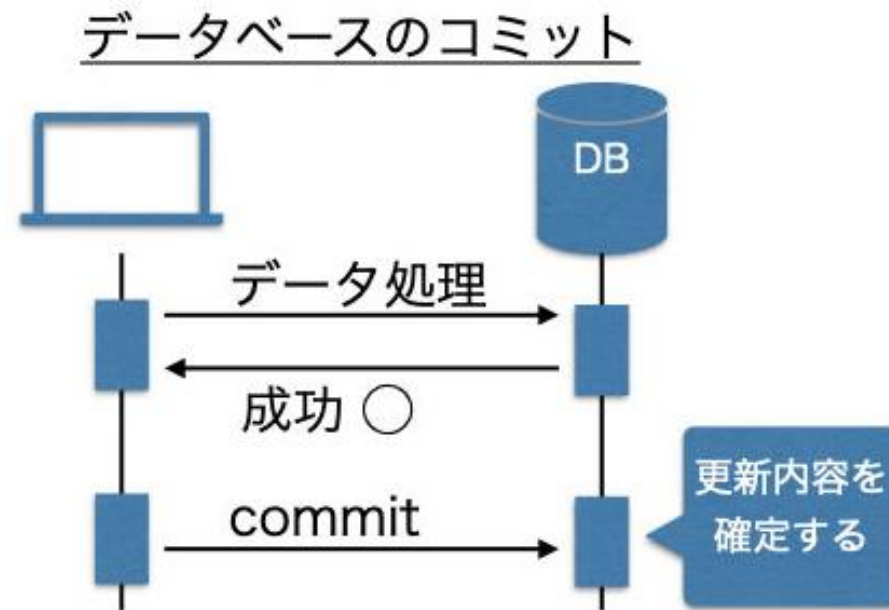
出金処理と入金処理の両方が
必ずセットで(1つの処理として)
行われること!

入金処理に失敗した場合、出金
処理を取り消し、金銭の整合性
を保つようにする

コミット

トランザクション内のすべての操作が
正常に完了したことを確認し、その結
果を**データベースに確定**する操作

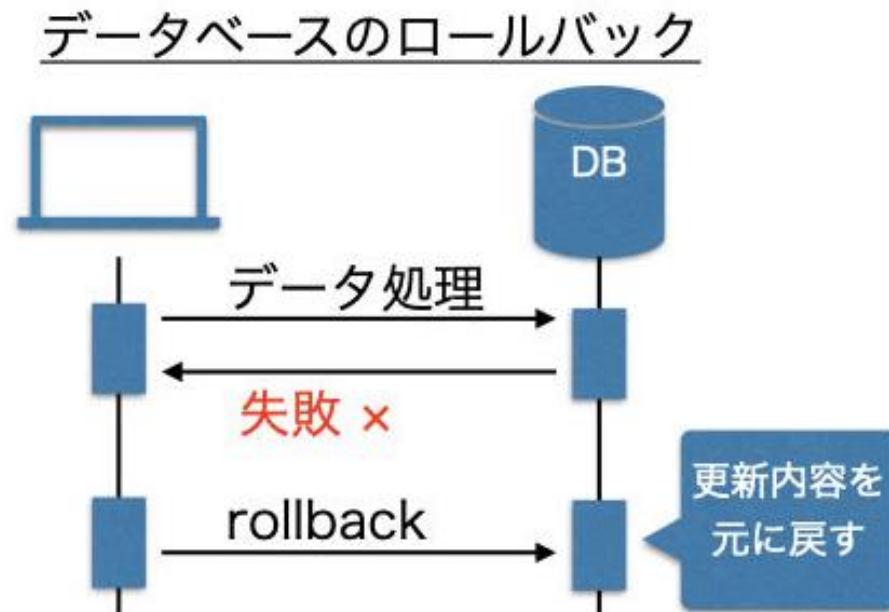
→ データの永続化・一貫性の確保



ロールバック

トランザクション内でエラーや問題が発生した場合に、そのトランザクション内で行われた**すべての操作を取り消し**、データベースを**トランザクション開始前の状態に戻す**操作

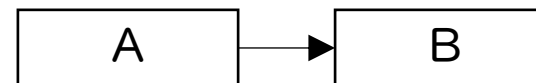
→ データの整合性を保持



過去問

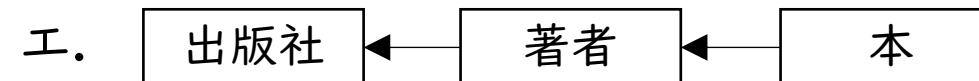
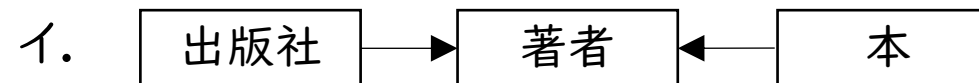
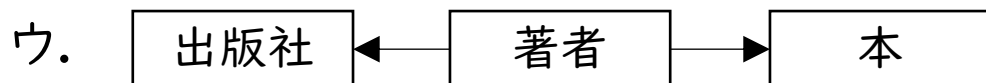
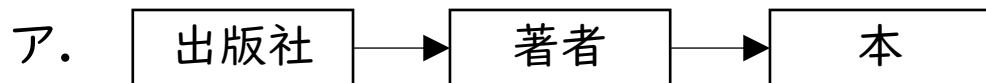
問1

条件①～④をすべて満たすとき、出版社と著者と本の関係を表すE-R図はどれか
なお、ER図の表記法は次のとおりとする。



AとBが1対多の関係

- ① 出版社は複数の著者と契約している
- ② 著者は1つの出版社とだけ契約している
- ③ 著者は複数の本を書いている
- ④ 1つの本は1人の著者が書いている



問2

条件①～⑤によって、関係データベースで管理する従業員表と部門表を作成した。従業員表の主キーとして最も適切なものはどれか？

- ① 各従業員には重複のない従業員番号を1つもつ
- ② 同姓同名の従業員がいてもよい
- ③ 各部門は重複のない部門コードを1つもつ
- ④ 1つの部門には複数名の従業員が所属する
- ⑤ 1人の従業員が所属する部門は1つだけである

従業員表

従業員番号	従業員名	部門コード	生年月日	住所
-------	------	-------	------	----

部門表

部門コード	部門名	所在地
-------	-----	-----

ア. 従業員番号

イ. 従業員名

ウ. 従業員番号と部門コード

エ. 従業員名と部門コード

問3 1つの表で管理されていた受注データを受注に関する情報と商品に関する情報に分割して、正規化を行った上で関係データベースの表で管理する。
正規化を行った結果の表の組合せとして最も適切なものはどれか？
ここで、同一商品で単価が異なるときは商品番号も異なり、発注者に同姓同名がないものとする。

受注データ

受注番号	発注者名	商品番号	商品名	個数	単価
T0001	田中一郎	M001	商品A	5	3000
T0002	鈴木次郎	M002	商品B	3	5000
T0003	佐藤花子	M001	商品A	2	3000

- ア.

受注番号	発注者名
------	------

 +

商品番号	商品名	個数	単価
------	-----	----	----
- イ.

受注番号	発注者名	商品番号
------	------	------

 +

商品番号	商品名	個数	単価
------	-----	----	----
- ウ.

受注番号	発注者名	商品番号	個数	単価
------	------	------	----	----

 +

商品番号	商品名
------	-----
- エ.

受注番号	発注者名	商品番号	個数
------	------	------	----

 +

商品番号	商品名	単価
------	-----	----

問4 関係データベースにおいて、表Aと表Bの積集合演算を実行した結果はどれか。

表A

品名	価格
ガム	100
せんべい	250
チョコレート	150

表B

品名	価格
せんべい	250
チョコレート	150
どらやき	120

ア.

品名	価格
せんべい	250
チョコレート	150

ウ.

品名	価格
ガム	100
せんべい	250
チョコレート	150
どらやき	120

エ.

品名	価格
ガム	100
せんべい	250
チョコレート	300
せんべい	250
チョコレート	300
どらやき	120

イ.

品名	価格
せんべい	500
チョコレート	300

問5

トランザクション処理におけるコミットの説明として適切なものはどれ。

ア. あるトランザクションが共有データを更新しようとしたとき、そのデータに対するほかのトランザクションからの更新を禁止すること

イ. トランザクションが正常に処理されたとき、データベースへの更新を確定させること

ウ. 何らかの理由でトランザクションが正常に処理されなかったときに、データベースをトランザクション開始前の状態にすること

エ. 複数の表を互いに関連付ける列をキーとして、1つの表にすること

問1

ア.

それぞれ、1つに特定できるかを考える。

例) 出版社が決まれば著作者が決まる

×

複数の著者と契約している

著作者が決まれば出版社が決まる

○

著者は1つの出版社とだけ契約している

著作者が決まれば、本が決まる

×

著者は複数の本を書いている

本が決まれば、著作者が決まる

○

1つの本は1人の著者が書いている

問2

ア.

同姓同名の人を区別できないため、従業員名は不適切。

従業員番号は重複がないため、主キーとして適切。

部門コードは部門表の主キーであり、従業員表において外部キーとして参照している。

問3

エ.

関数従属性を考える。

商品テーブルにおいて、商品コードで分かるのは商品名と単価のみ。

個数は受注に関するデータであるため、商品テーブルには含めない。

問4

ア.

積集合はどちらのテーブルにも属しているものを抽出したもの。

問5

イ.

ア.は排他制御(ロック)、ウ.はロールバック、エ.は結合 の説明。