

データベース

http://cobayasi.com/koza/kihon/8_database.pdf

1. データベースの基礎 ★★
2. 関係データベース★★
3. データの正規化★★★★
4. SQLの基本★
5. SQLの応用★★
6. データベース管理システム★★★★
7. データベースの応用技術★

★★ 1. データベースの基礎

● データベースとは

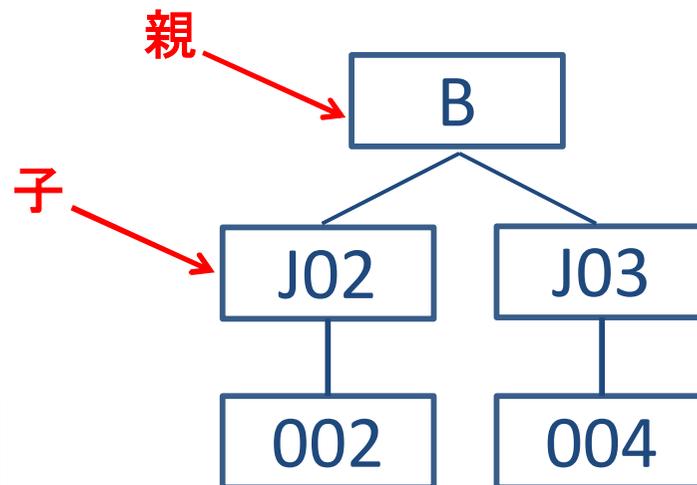
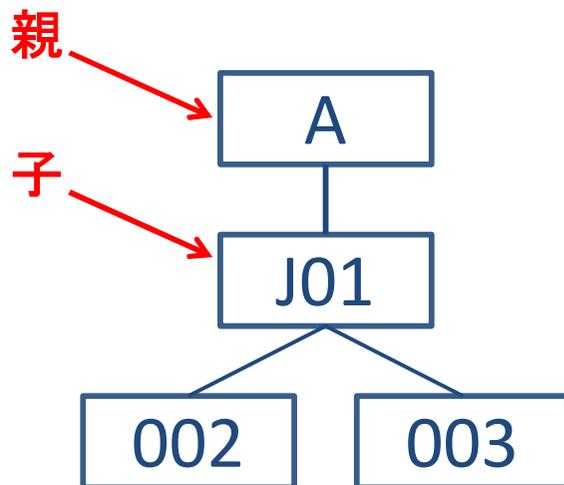
- 大量のデータを一定の規則に従って整理したもの。
- 必要なデータを検索したり抽出したりできる。

● データベースの種類

- 階層型データベース

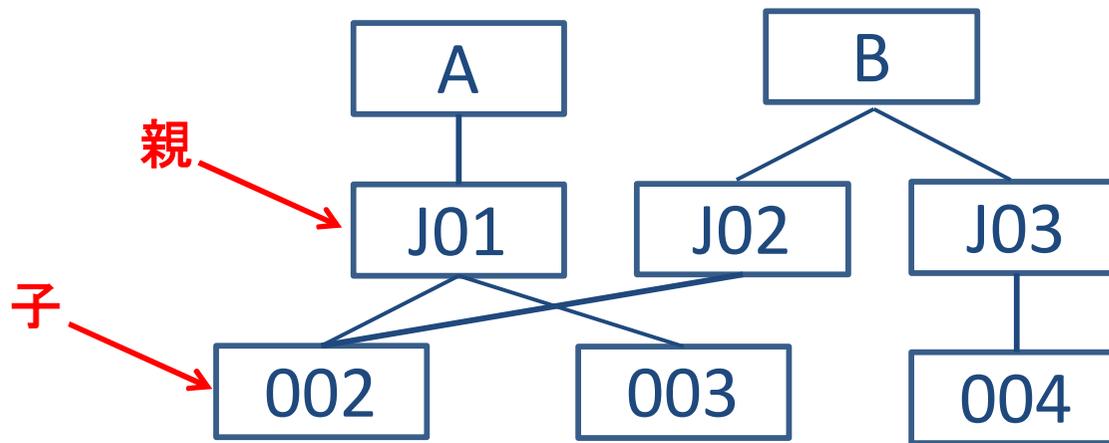
子の親は1つ

データを木のような形の階層構造で表す



- ネットワーク型データベース
データを網の目のような構造で表す

子の親は複数



- 関係データベース (RDB : Relational DataBase)
データを行と列の2次元の表で表し、複数の表と関係づけられている

学籍番号	資格名	取得日
002	J01	2009/6
003	J01	2008/6
004	J01	2009/6

関連付け

資格名	難易度
J01	A
J02	B
J03	C

A red arrow points from the '資格名' column of the first table to the '資格名' column of the second table, indicating a relationship between the two tables.

【過去問題】

DBMSにおいて、スキーマを決める機能はどれか。

ア 機密保護機能

イ 障害回復機能

ウ 定義機能

エ 保全機能

スキーマ(schema)とは、データの内容、データの論理構造、記憶形式や編成などデータベースの構造を記述したもの

★★ 2. 関係データベース

一番広く使われている

● 関係データベースの構成

マス目で、データを管理するデータベース

テーブル

学籍番号	氏名	学年	クラス
001	山田太郎	1	C
002	田中一郎	2	A
003	高橋和子	4	E
004	後藤 正	2	B
005	斉藤美香	1	A

行:レコード
(1件分データ)

列:フィールド
(同じ項目のデータ)

● 主キー

主キー制約

- 値が空(0)でなく、絶対に他のデータと内容がダブらない(一意)
- フィールドデータを検索するために必要不可欠

主キー

学籍番号	氏名	学年	クラス
001	山田太郎	1	C
002	田中一郎	2	A
003	高橋和子	4	E
004	後藤 正	2	B
005	斉藤美香	1	A

● 外部キー

表内の列(フィールド)のうち、**関連付けのある他の表の主キー**としても使われる列を“**外部キー**”と呼ぶ

学籍番号	資格名	取得日
002	J01	2009/6
002	J02	2009/12
003	J01	2008/6
004	J01	2009/6

資格名	難易度
J01	ITパスポート
J02	基本情報
J03	応用情報



関係する表同士でレコード間の**参照の整合性を維持**することを、“**参照制約**”と呼ぶ

● 表の操作

表から必要なデータを抽出するには、以下の種類が存在する

- ①必要な行(レコード)を抽出する操作 → “**選択**”
- ②必要な列(フィールド)を抽出する操作 → “**射影**”

〈選択と射影〉

①必要なレコードを抽出する(**選択**)

ID	書籍名	著者
001	ステキ筋肉	犬山太郎
002	イタリア語講座	猫田花子
003	英会話テキスト	牛島 明
004	ダイエット事典	鳥谷茂雄
005	日常イタリア語	猫田花子

(選択の条件: 著者が「猫田花子」の書籍)

002	イタリア語講座	猫田花子
005	日常イタリア語	猫田花子

(射影の条件: 「書籍名」のみ)

ステキ筋肉
イタリア語講座
英会話テキスト
ダイエット事典
日常イタリア語

②必要なフィールドを抽出する(**射影**)

③2つの表から条件に合った列を取り出し組み合わせ、新しい表を作る → “結合”

学籍番号	氏名	クラス
001	山田太郎	1C
002	田中一郎	2A
003	高橋和子	3E
004	後藤 正	2B
005	斎藤美香	2A

学籍番号	資格コード	取得日
002	J01	2009/6
002	J02	2009/12
003	J01	2008/6
004	J01	2009/6

③学籍番号が一致するデータを結合

学籍番号	氏名	クラス	資格コード	取得日
002	田中一郎	2A	J01	2009/6
002	田中一郎	2A	J02	2009/12
003	高橋和子	3E	J01	2008/6
004	後藤 正	2B	J01	2009/6

【過去問題】

関係データベースの操作のうち、射影(projection)の説明として、適切なものはどれか。

- ア ある表の照会結果と、別の表の照会結果を合わせて **和** 一つの表にする。
- イ 表の中から特定の条件に合致した行を取り出す。 **選択**
- ウ** 表の中から特定の列だけを取り出す。
- エ 二つ以上の表の組から条件に合致した組同士を合わせて新しい表を作り出す。 **結合**

射影(しゃえい)は、関係データベースを操作する関係演算の1つで、表から条件に合致する列を取り出す演算。

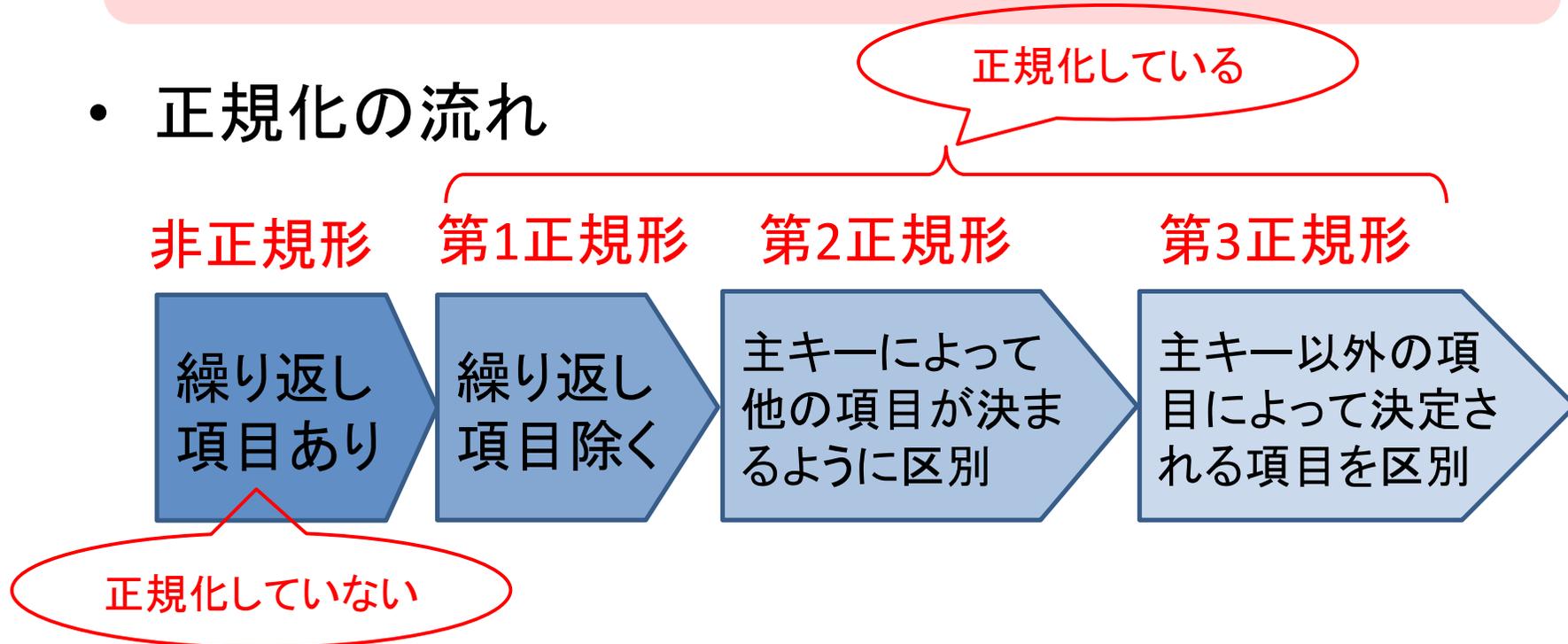
★★★3. データの正規化

●正規化とは

データベースを、効率よく利用するために、**一定のルールに従って整理**する。

データの**重複や矛盾の発生をなくす**ことが目的。

●正規化の流れ



● 非正規形 繰り返し構造がある表

1レコードに複数の
データが存在する

学籍番号	氏名	クラス	担任	資格コード	資格名	取得日
002	田中一郎	2A	松本 勝	J01	ITパスポート	2009/6
				J02	基本情報	2009/12
003	高橋和子	3E	二宮純一	J01	ITパスポート	2008/6
004	後藤 正	2B	相葉一成	J01	ITパスポート	2009/6

繰り返し
(重複)

● 第1正規形 繰り返し構造がない表

レコードごとにデータ
が存在する

学籍番号	氏名	クラス	担任	資格コード	資格名	取得日
002	田中一郎	2A	松本 勝	J01	ITパスポート	2009/6
002	田中一郎	2A	松本 勝	J02	基本情報	2009/12
003	高橋和子	3E	二宮純一	J01	ITパスポート	2008/6
004	後藤 正	2B	相葉一成	J01	ITパスポート	2009/6

● 第2正規形

主キーで項目が決まる

主キーによって項目が決まるように分割した表①～③

第2正規形①

学籍番号	氏名	クラス	担任
002	田中一郎	2A	松本 勝
003	高橋和子	3E	二宮純一
004	後藤 正	2B	相葉一成

主キー

第2正規形②

資格コード	資格名
J01	ITパスポート
J02	基本情報

主キー

関数従属

第2正規形③

学籍番号	資格コード	取得日
002	J01	2009/6
002	J02	2009/12
003	J01	2008/6
004	J01	2009/6

主キー

2つの主キーで項目が決まる

● 第3正規形

レコードの項目すべてが、主キーのみで決まるように分割した
(推移的關係従属がない状態)表①,②

第2正規形①

学籍番号	氏名	クラス	担任
002	田中一郎	2A	松本 勝
003	高橋和子	3E	二宮純一
004	後藤 正	2B	相葉一成

第3正規形①

学籍番号	氏名	クラス
002	田中一郎	2A
003	高橋和子	3E
004	後藤 正	2B

第3正規形②

クラス	担任
2A	松本 勝
3E	二宮純一
2B	相葉一成

• 完全に正規化された表

第2正規形②③は、主キー以外で決定する項目は存在しないので、第3正規形の条件を満たしている

第3正規形①(学生)

学籍番号	氏名	クラス
002	田中一郎	2A
003	高橋和子	3E
004	後藤 正	2B

第3正規形②(担任)

クラス	担任
2A	松本 勝
3E	二宮純一
2B	相葉一成

データの重複
や矛盾を排除
した表

第2正規形②(資格)

資格コード	資格名
J01	ITパスポート
J02	基本情報

第2正規形③(学生資格取得)

学籍番号	資格コード	取得日
002	J01	2009/6
002	J02	2009/12
003	J01	2008/6
004	J01	2009/6

完全に正規化された表

第2正規形②③は、主キー以外で決定する項目は存在しないので、第3正規形の条件を満たしている

第3正規形①(学生)

学籍番号	氏名	クラス
002	田中一郎	2A
003	高橋和子	3E
004	後藤 正	2B

第3正規形②(担任)

クラス	担任
2A	松本 勝
3E	二宮純一
2B	相葉一成

データの重複
や矛盾を排除
した表

第3正規形③(資格)

資格コード	資格名
J01	ITパスポート
J02	基本情報

第3正規形④(学生資格取得)

学籍番号	資格コード	取得日
002	J01	2009/6
002	J02	2009/12
003	J01	2008/6
004	J01	2009/6

【過去問題】

関係を第3正規形まで正規化して設計する目的はどれか。

- ア 値の重複をなくすことによって、格納効率を向上させる。
- イ 関係を細かく分解することによって、整合性制約を排除する。
- ウ 冗長性を排除することによって、更新時異状を回避する。
- エ 属性間の結合度を低下させることによって、更新時のロック待ちを減らす。

正規化の目的は冗長性の排除することで更新時異状を回避し、データベースの一貫性を確保すること。

【過去問題】

関係データベースの主キー制約の条件として、キー値が重複していないことの他に、主キーを構成する列に必要な条件はどれか。

- ア キー値が空でないこと
- イ 構成する列が一つであること
- ウ 表の先頭に定義されている列であること
- エ 別の表の候補キーとキー値が一致していること

主キー(primary key)は、表内のある行を一意に決めることができる特性をもつ属性である。候補キーにはNULL値が許されますが、主キーではNULL値の存在が許されないので、主キー制約は一意性制約にNOT NULL制約を加えたもの。

★ 4. SQLの基本

●SQLとは

データベースから、必要なデータを抽出するために、**選択・射影・結合などの操作**をする言語。

どんなアプリケーションや機種でも、共通して使え、比較的単純な文法で出来ている。

●SELECT文

「どの表から」「どの列を」「どのような条件で」**抽出**するかを指定する

・基本形

SELECT 列名 **FROM** 表名 **WHERE** 条件式

どの列を

どの表から

どのような条件で

- (特定の)列を抽出する(FROM)

“学生”の表から“氏名”の列を抽出する
SELECT 氏名 FROM 学生

〈学生〉

学籍番号	氏名	クラス
001	山田太郎	1C
002	田中一郎	2A
003	高橋和子	3E
004	後藤 正	2B
005	斉藤美香	2A

実行



結果

氏名
山田太郎
田中一郎
高橋和子
後藤 正
斉藤美香

- すべての列を抽出する(*)

“学生”の表からすべての列を抽出する

SELECT * FROM 学生

アスタリスク(*)は、すべて

〈学生〉

学籍番号	氏名	クラス
001	山田太郎	1C
002	田中一郎	2A
003	高橋和子	3E
004	後藤 正	2B
005	齊藤美香	2A

実行



結果

学籍番号	氏名	クラス
001	山田太郎	1C
002	田中一郎	2A
003	高橋和子	3E
004	後藤 正	2B
005	齊藤美香	2A

- 条件で絞り込む(WHERE)

“学生”の表からクラスが'2A'のすべての列を抽出する
SELECT * FROM 学生 WHERE クラス = '2A'

↑
比較演算子

〈学生〉

学籍番号	氏名	クラス
001	山田太郎	1C
002	田中一郎	2A
003	高橋和子	3E
004	後藤 正	2B
005	斉藤美香	2A

実行



結果

学籍番号	氏名	クラス
002	田中一郎	2A
005	斉藤美香	2A

条件でデータを絞り込むには、WHEREをつけて以下の**演算子**を用いる

	構文	例	意味
比較演算子	A = B	価格=100	価格が100円
	A <> B	価格<>100	価格が100円でない
	A > B	価格>100	価格が100円より高い
	A < B	価格<100	価格が100円より安い(未満)
	A >= B	価格>=100	価格が100円以上
	A <= B	価格<=100	価格が100円以下
	BETWEEN A AND B	価格 BETWEEN 100 AND 200	価格が100円と200円の間
論理演算子	A AND B	価格>=100 AND 価格<=200	価格が100円以上で200円以下
	A OR B	価格=100 OR 価格=200	価格が100円または200円
	NOT A	NOT 価格=100	価格が100円でない

- 表を結合する

“学生”の表と”担任”の表を結合して、'氏名'と'担任'を抽出する

```
SELECT 氏名,担任 FROM 学生,担任  
WHERE 学生,クラス = 担任,クラス
```

↑ ↑ ↑ ↑
表名 列名 表名 列名

<学生>

学籍番号	氏名	クラス
001	山田太郎	1C
002	田中一郎	2A
003	高橋和子	3E
004	後藤 正	2B
005	斉藤美香	2A

<担任>

クラス	担任
1C	櫻井聡史
2A	松本 勝
2B	相葉一成
3E	二宮純一

実行
➡

結果

氏名	担任
山田太郎	櫻井聡史
田中一郎	松本 勝
高橋和子	二宮純一
後藤 正	相葉一成
斉藤美香	松本 勝

【練習問題】

<生徒>の表から、“血液型”が“A型”で、同時に“生年”が“1982”の行だけを取り出すときのSELECT文中の(A)~(C)を埋めなさい。

SELECT (A) FROM 生徒 (B) 血液型= 'A型' (C) 生年= '1982'

<生徒>

ID	名前	生年	誕生日	血液型	出身地
01	高杉晋作	1980	11/26	A型	山口県
02	木戸孝允	1982	01/25	A型	山口県
03	坂本竜馬	1982	12/24	AB型	高知県
04	伊藤博文	1983	06/17	A型	山口県
05	井上香	1983	08/30	A型	山口県

A : *

B : WHERE

C : AND

★★ 5. SQLの応用

● 集計関数で集計する

SQLでは、集計関数を使ってデータを集計(行数、合計、平均など)して表示する

- 合計値, 平均値を求める (SUM関数, AVG関数)

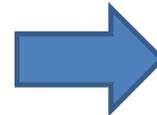
```
SELECT SUM(点数) FROM 成績  
SELECT AVG(点数) FROM 成績
```

<成績>

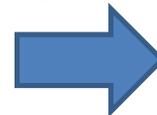
学籍番号	科目	点数
002	数学	90
002	国語	78
002	英語	86
005	数学	100
005	国語	60
005	英語	50

集計関数

合計(SUM)



平均(AVG)



実行結果

SUM(点数)

464

実行結果

AVG(点数)

77.3

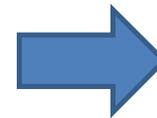
- 行数をカウントする(COUNT(*)関数)

```
SELECT COUNT(*) FROM 成績
```

<成績>

	学籍番号	科目	点数
1	002	数学	90
2	002	国語	78
3	002	英語	86
4	005	数学	100
5	005	国語	60
6	005	英語	50

実行



結果

COUNT(*)
6

- 最小値、最大値を調べる (MIN・MAX)

“成績”の表から、科目’数学’の最小点を求める
`SELECT MIN(点数) FROM 成績 WHERE 科目=‘数学’`

“成績”の表から、科目’国語’の最大点を求める
`SELECT MAX(点数) FROM 成績 WHERE 科目=‘国語’`

学籍番号	科目	点数
002	数学	90
002	国語	78
002	英語	86
005	数学	100
005	国語	60
005	英語	50

実行結果

MIN(点数)
90

実行結果

MAX(点数)
78

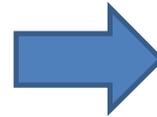
- グループ化する(GROUP BY)
同じ列のデータをまとめる
- グループごとに平均値を求める

“成績”の表で、科目ごとに平均点を求める
`SELECT 科目, AVG(点数) FROM 成績 GROUP BY 科目`

<成績>

学籍番号	科目	点数
002	数学	90
002	国語	78
002	英語	86
005	数学	100
005	国語	60
005	英語	50

実行



結果

科目	AVG(点数)
数学	95
国語	69
英語	68

- グループ化した後の絞り込み(HAVING)
グループ化したデータの結果に、条件を付けて絞り込む

“成績”の表で、科目ごとに求めた平均点が、70点以上の科目を求める

```
SELECT 科目, AVG(点数) FROM 成績 GROUP BY 科目  
HAVING AVG(点数) >= 70
```

↑
条件

<成績>

学籍番号	科目	点数
002	数学	90
002	国語	78
002	英語	86
005	数学	100
005	国語	60
005	英語	50



結果

科目	AVG(点数)
数学	95

● 並べ替える(ORDER BY)

データを大きい順(降順)や小さい順(昇順)などに並べ替える

- 昇順に並べ替える(ASC)

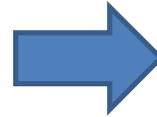
省略可

```
SELECT * FROM 成績 ORDER BY 点数 ASC
```

〈成績〉

学籍番号	科目	点数
002	数学	90
002	国語	78
002	英語	86
005	数学	100
005	国語	60
005	英語	50

実行



結果

学籍番号	科目	点数
005	英語	50
005	国語	60
002	国語	78
002	英語	86
002	数学	90
005	数学	100

- 降順に並べ替える(DESC)

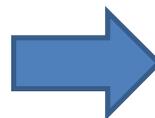
省略不可

```
SELECT * FROM 成績 ORDER BY 点数 DESC
```

<成績>

学籍番号	科目	点数
002	数学	90
002	国語	78
002	英語	86
005	数学	100
005	国語	60
005	英語	50

実行



結果

学籍番号	科目	点数
005	数学	100
002	数学	90
002	英語	86
002	国語	78
005	国語	60
005	英語	50

●副問合せ

SELECT文の中でSELECT文を使って、複数の表の比較ができる

“商品”表と“在庫”表の’商品番号’を比較して、在庫表に存在しない商品番号を求める

```
SELECT 商品番号 FROM 商品 WHERE 商品番号 NOT IN  
(SELECT 商品番号 FROM 在庫)
```

主問合せ

副問合せ

<商品>

商品番号	商品名	単価
A001	りんご	100
A002	みかん	50
A003	もも	300
A004	メロン	500

<在庫>

在庫番号	商品番号	個数
Z0001	A001	50
Z0002	A003	10
Z0003	A004	5

実行



結果

商品番号
A002

WHERE 商品番号 NOT IN (A001,A003,A004)

●ビュー表を作る(CREATE VIEW)

既存の表から、必要なデータを抜き出して仮想的な表を作る

※ビュー表には、実際にはデータは存在せず、参照する表にある

ビュー表を作るための基本的な命令文

- ビュー表を作る(定義する)
 - **CREATE VIEW** ビュー表名(列名,列名,...)
- 参照する表名と列名を指定
 - **AS SELECT** 列名,列名,... **FROM** 表名
- 抽出する条件
 - **WHERE** 条件式

“成績”表から“科目別平均点”のビュー表を作成する
CREATE VIEW 科目別平均点(科目,平均点) AS SELECT
成績,科目,成績.AVG(点数) FROM 成績 GROUP BY 科目

〈成績〉

学籍番号	科目	点数
002	数学	90
002	国語	78
002	英語	86
005	数学	100
005	国語	60
005	英語	50

点数が変更され
ると平均点
も更新される

結果：科目別平均点



科目	平均点
数学	95
国語	69
英語	68

【過去問題】

“社員”表と“部門”表に対し、次のSQL文を実行したときの結果はどれか。

ア 1 イ 2 **ウ 3** エ 4

```
SELECT COUNT(*) FROM 社員,部門
WHERE 社員,所属 = 部門,部門名 AND 部門,フロア = 2
```

<社員>

社員番号	所属
11001	総務
11002	経理
11003	営業
11004	営業
11005	情報システム
11006	営業
11007	企画
12001	営業
12002	情報システム

<部門>

部門名	フロア
企画	1
総務	1
情報システム	2
営業	3
経理	2
法務	2
購買	2

WHERE句にあるように、社員表と部門表を所属列と部門名列を、キー列として結合する。結合処理すると、フロア列が追加された右のような表になる。

社員番号	所属	フロア
11001	総務	1
11002	経理	2
11003	営業	3
11004	営業	3
11005	情報システム	2
11006	営業	3
11007	企画	1
12001	営業	3
12002	情報システム	2

結合された表中で、フロア列の値が“2”である行(部門.フロア=2)の数を、COUNT(*)で集計する。

社員番号	所属	フロア
11001	総務	1
11002	経理	2
11003	営業	3
11004	営業	3
11005	情報システム	2
11006	営業	3
11007	企画	1
12001	営業	3
12002	情報システム	2



COUNT(*)
3

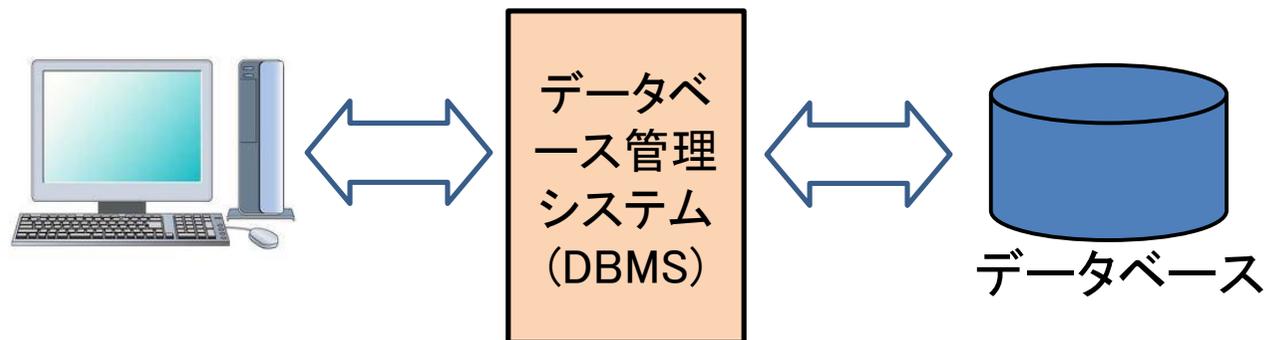


6. データベース管理システム

- データベース管理システム (DBMS : Data Base Management System) とは

複雑化したデータベースを、不都合や不具合が起こらないように、管理するためのソフトウェア。

データベースを効率よく利用するための色々な機能を有し、ユーザは、このDBMSを介してデータベースを操作する。



● DBMSの機能・その1(トランザクション管理)

データベースに対する操作や処理を、操作単位ではなく、トランザクション(複数の操作を一連の処理とする)単位で行うようにする

■ データの一貫性を保証するための4要素

ACID特性

- **A**tomicity(原子性)

トランザクションが、データベースに対して、「更新処理を完全に行う」または「まったく処理しないように取り消す」ことを保証する

- **C**onsistency(一貫性)

トランザクション処理の前後で、データが矛盾なく、常にデータベースの整合性が保たれる

- **I**solation(独立性)

複数のトランザクションを同時に実行しても、順番に実行しても、結果は同じである

- **D**urability(永続性)

障害が発生しても、正常に終了したトランザクションの結果は失われずに復旧できる

● DBMSの機能・その2(排他制御)

データベースを**複数の人が同時にアクセスできないように操作を制限(ロック)**して、データの整合性を保持する



- 占有ロック(専有ロック)

データベースを更新するときのロック。他のユーザは、**読み取りや更新できない**。最初に操作したユーザのみが、データベースに操作できる。



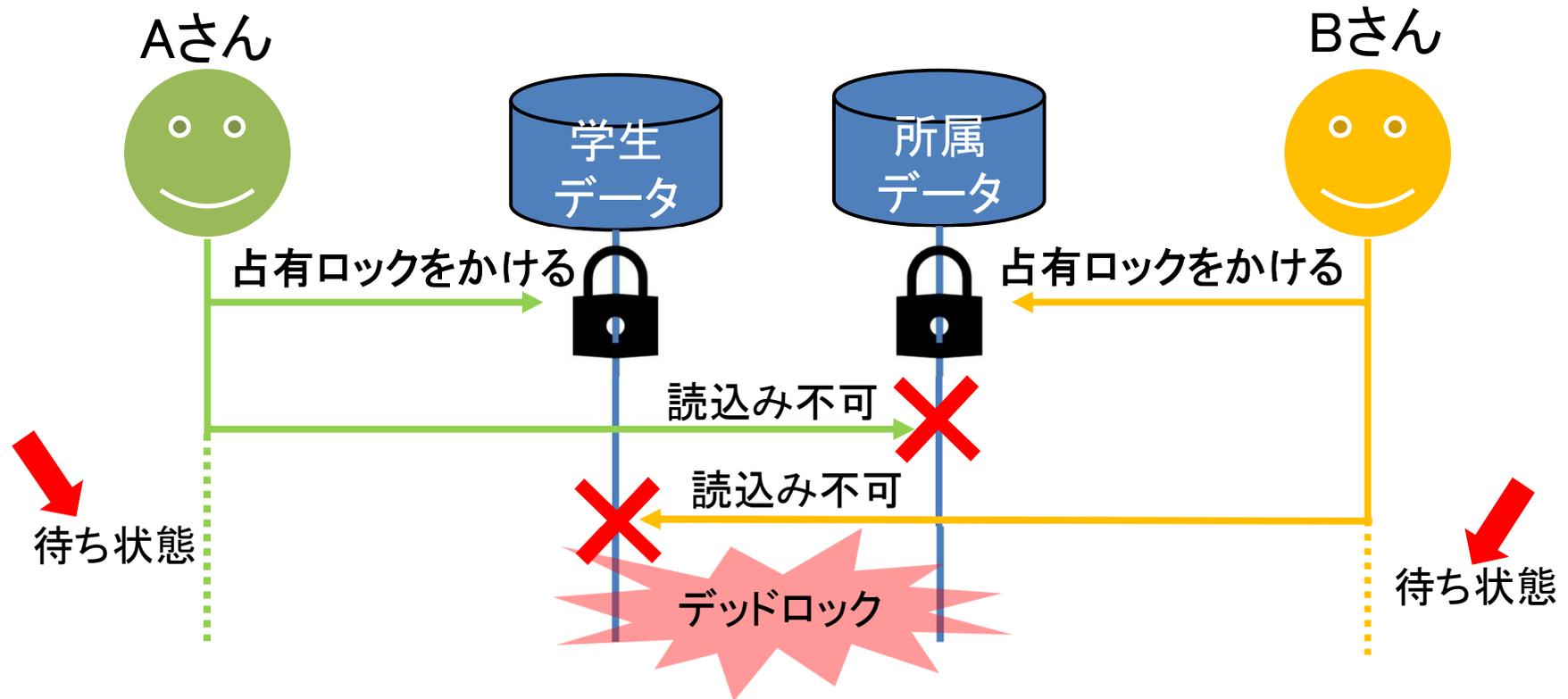
- 共有ロック

データベースを参照するときのロック。他のユーザは、読み取りはできるが、更新はできない。



- デッドロック

複数のユーザが、お互いにロックをかけて、アクセスの待ち状態が続き処理が進まない。



- ロック粒度

データベースで、**ロックをかけるデータの単位**(全体、表、行など)

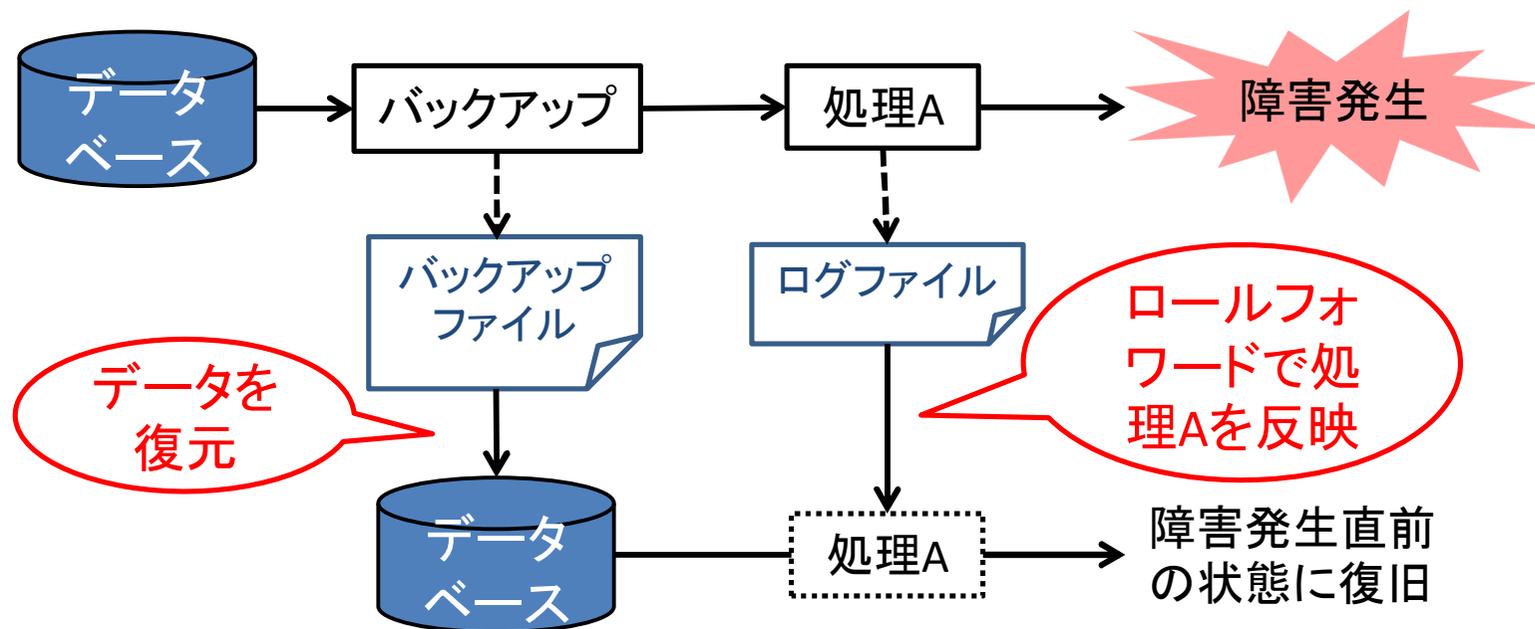
- ✓ ロックの粒度が細かいと、ロック解除までの待ち時間が短く、
ロックの競合が起こりにくい
- ✓ ロックした数が多いと、メモリの使用容量が増える

● 障害回復（リカバリ）機能

トランザクションが正常に行われなかった場合に、データの更新履歴（ログファイル）を使って、**データを正常な状態に戻す作業**

• ロールフォワード

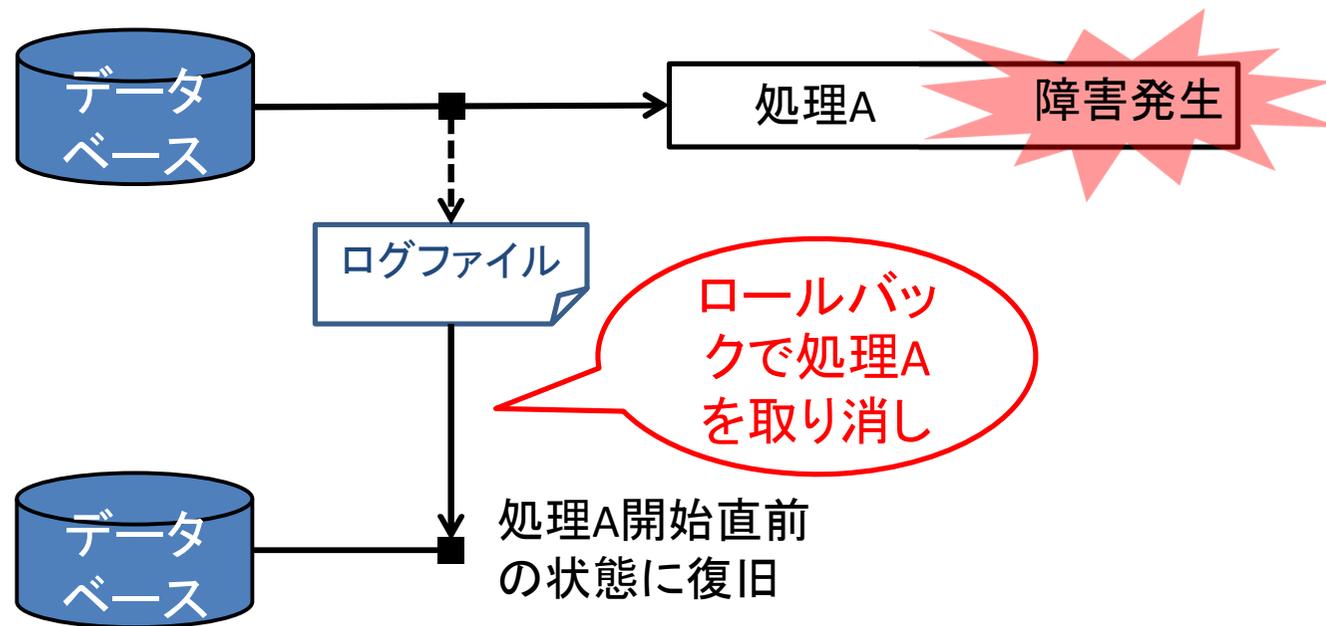
物理的な障害（データベースの故障など）で行うリカバリ。
バックアップ時点の状態に復元後、**ログファイル更新後の情報を使って、障害発生前の状態に復旧する。**



- ロールバック

論理的な障害(トランザクションの異常終了など)で行うリカバリ。

ログファイル更新前の情報を使って、トランザクションの開始直前の状態に復旧する。



● 再編成機能

データベースの更新を繰り返していると、データの格納位置が不規則になるなどの理由により、データベースのアクセス効率が低下する。この効率を向上させるための機能を**再編成機能**と呼ぶ。

● インデックス機能

データベース内の大量のデータから、目的のデータを高速で検索するための機能。どのデータがどこにあるかを示した一覧表。

【過去問題】

DBMSにおいて、複数のトランザクション処理プログラムが同一データベースを同時に更新する場合、論理的な矛盾を生じさせないために用いる技法はどれか。

- ア 再編成
- イ 正規化
- ウ 整合性制約
- エ 排他制御**

排他制御は、トランザクションの整合性を保つために、1つのタスクがデータ更新のトランザクションを実行している間は、その処理が終わるまで、同じデータに対する他のトランザクション要求を待機させる仕組みである。

複数のトランザクションが、同じデータを更新しようとしたときに、データを「ロック」することで、同時更新による更新消失を防ぐことができる。

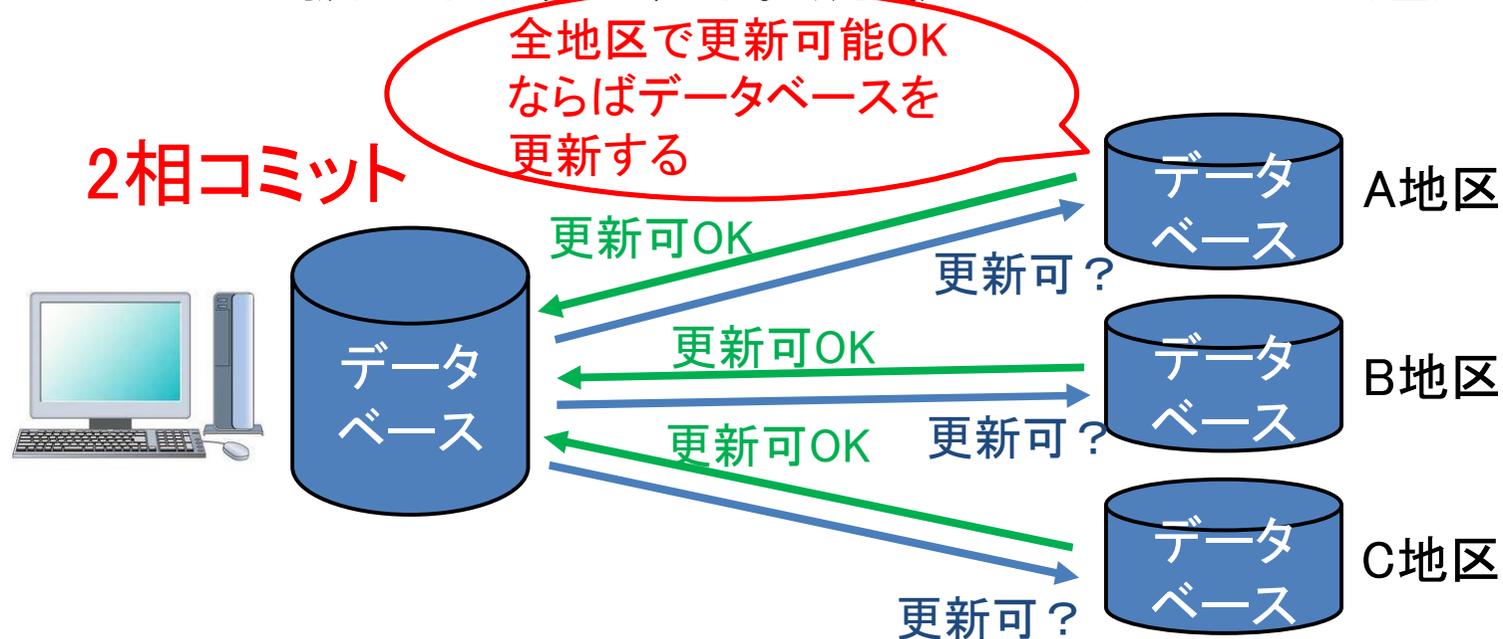
★7. データベースの応用技術

- 分散データベース

データベース内の重要で大量のデータを、**複数個所で分散して管理**する

- 透過性

複数個所に分散しているデータベースを、**見かけ上1つのデータベース**として扱えるようにする。(分散データベースの透過性)



【過去問題】

分散データベースシステムにおいて、一連のトランザクション処理を行う複数サイトに更新処理が確定可能かどうかを問い合わせ、すべてのサイトが確定可能である場合、更新処理を確定する方式はどれか。

- ア 2相コミット
- イ 排他制御
- ウ ロールバック
- エ ロールフォワード

2相コミット(Two Phase Commit)は、トランザクションのコミットを次の2つの段階に分けて行うことで、分散データベース環境でのトランザクションの原子性・一貫性を保証する仕組みである。

第1フェーズ: 他のサイトに更新可能かどうかを確認する

第2フェーズ: 全サイトからの合意が得られた場合に更新を確定する

令和元年度 秋期 基本情報処理技術者試験問題・解答(データベース)

【問27】

関係モデルにおいて、関係から特定の属性だけを取り出す演算はどれか。

ア 結合(join)

イ 射影(projection)

ウ 選択(selection)

エ 和(union)

射影は、表から指定された列(属性)を抽出する操作

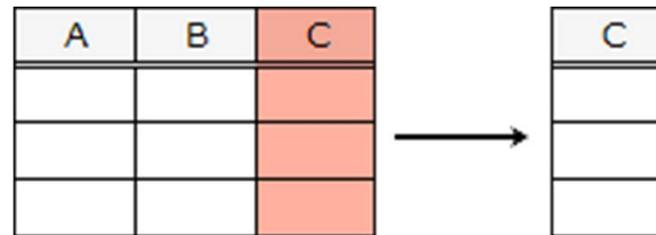


図 射影

平成31年度 春期

基本情報処理技術者試験問題・解答(データベース)

【問27】

“中間テスト”表からクラスごと、教科ごとの平均点を求め、クラス名、教科名を昇順に表示するSQL文中のaに入れる字句はどれか。

中間テスト(クラス名,教科名,学生番号,名前,点数)

[SQL文]

```
SELECT クラス名, 教科名, AVG(点数) AS 平均点  
FROM 中間テスト
```

a

- ア GROUP BY クラス名, 教科名 ORDER BY クラス名, AVG(点数)
- イ GROUP BY クラス名, 教科名 ORDER BY クラス名, 教科名
- ウ GROUP BY クラス名, 教科名, 学生番号 ORDER BY クラス名, 教科名, 平均点
- エ GROUP BY クラス名, 平均点 ORDER BY クラス名, 教科名