データベースシステム入門

１３．テーブル定義，データベース操作，索引，
一対多でのテーブル定義，多対多でのテーブル定義

https://www.kunihikokaneko.com/free/dbintro/index.html

リレーショナルデータベースシステム

コンピュータ

リレーショナル
データベース
管理システム

記憶
装置

リレーショナル
データベース

あわせて
リレーショナルデータベースシステム

	id	name	teacher_name	student_name	score	created_at	updated_at
1	1	Database	K	KK	85	2014-09-13 00:46:17	{null}
2	2	Database	K	AA	75	2014-09-13 00:48:26	{null}
3	3	Database	K	LL	90	2014-09-13 00:48:26	{null}
4	4	Programming	A	KK	85	2014-09-13 00:48:26	{null}
5	5	Programming	A	LL	75	2014-09-13 00:48:26	{null}

	id	year	month	day	customer_name	product_id	qty	created_at
1	1001	2014	10	26	kaneko	1	10	2014-09-14 14:28:38
2	1002	2014	10	26	miyamoto	2	2	2014-09-14 14:28:38
3	1003	2014	10	27	kaneko	3	8	2014-09-14 14:28:38
4	1004	2014	10	27	kaneko	3	8	2014-09-14 14:28:38

データの種類ごとに分かれた、
たくさんの**テーブル**が格納される

今日学んでもらうこと

◆　「データベースエンジニア」として活躍できるための大事な知識

1．**テーブル定義**
- ・SQLを用いたテーブル定義
- ・「何か」と「何か」の**関係**を記録するのは，どうしたらよいの？

2．データベース処理を**高速化できる技術**

3．データベース操作のコマンド

操作の種類	SQL
新しいレコードの挿入	INSERT INTO
条件に合致するレコードの削除	DELETE FROM WHERE
既存のデータの更新	UPDATE SET WHERE

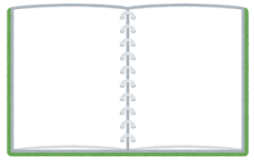
13-1 テーブル定義

データベースの構築

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**



テーブル定義（データ定義ともいう）

※　最初，テーブルは**空**

データベース操作、問い合わせ

テーブル定義とは

リレーショナルデータベースにおいて、

・**テーブル名**

・各フィールドの**フィールド名**（属性名）

・各フィールドの**データ型**

などを定義すること．**主キー**の指定も行う

ノートページ

SQL のデータ型

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

フィールド　フィールド　フィールド

フィールドが３つ

リレーショナルデータベースで、**データ型**というときは

各フィールドの**データの種類**のこと

SQL のデータ型は「英語のキーワード」

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

数値型　短いテキスト　数値型　← Access 2013 のデザインビューを使うとき

integer　char　integer　← **SQL** を使うとき

SQL のデータ型

ノートページ

以前の授業で確認済み

Access 2013 の主なデータ型	SQL のキーワード
短いテキスト	char
長いテキスト	text
数値型	integer, real
日付／時刻型	datetime
YesやNo	bit

※ 「**短いテキスト**」は
半角 255文字分までが目安

※整数は integer,
浮動小数点数は real

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**フィールド**

ノートページ

以前の授業で確認済み

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

主キー

SQL でのテーブル定義の例

テーブル名：**学科**

ID	学科名	学部名
1	情報	工
2	スマート	工

テーブルの例

```
create table 学科 (
    ID integer,
    学科名 char,
    学部名 char,
    primary key(ID)
);
```

primary key ・・・ 主キーの指定

SQL でのテーブル定義例
(Access 2013の画面)

SQL での主キーの指定

- primary key ・・・ **主キー**

```
create table 学科 (
    ID integer,
    学科名 char,
    学部名 char,
    primary key(ID)
);
```

ノートページ

※　他の書き方もあります

主キーは大切

- 主キーは、データ管理のために重要
- **SQL の create table を使ってテーブル定義するとき、定義したいテーブルに、主キーがあるなら、必ず primary key(・・・) を書く**

```
create table 学科 (
    ID integer,
    学科名 char,
    学部名 char,
    primary key(ID)
);
```

13-2 データベース操作

3 種類のデータベース操作

- 新しいレコードの挿入
- 条件に合致するレコードの削除
- 既存のデータの更新

3種類のデータベース操作

操作の種類	SQL
新しいレコードの挿入	**INSERT INTO**
条件に合致するレコードの削除	**DELETE FROM WHERE**
既存のデータの更新	**UPDATE SET WHERE**

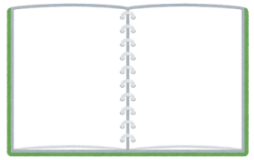
ノートページ

データベースの構築

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**



テーブル定義（データ定義ともいう）

※　最初，テーブルは**空**

データベース操作、問い合わせ

13-3 索引（インデックス）

インデックス（索引）のイメージ

見出し語	場所
アセンブラ	35
データベース	110
マシンラーニング	169

インデックス（索引）

ノートページ

データ本体

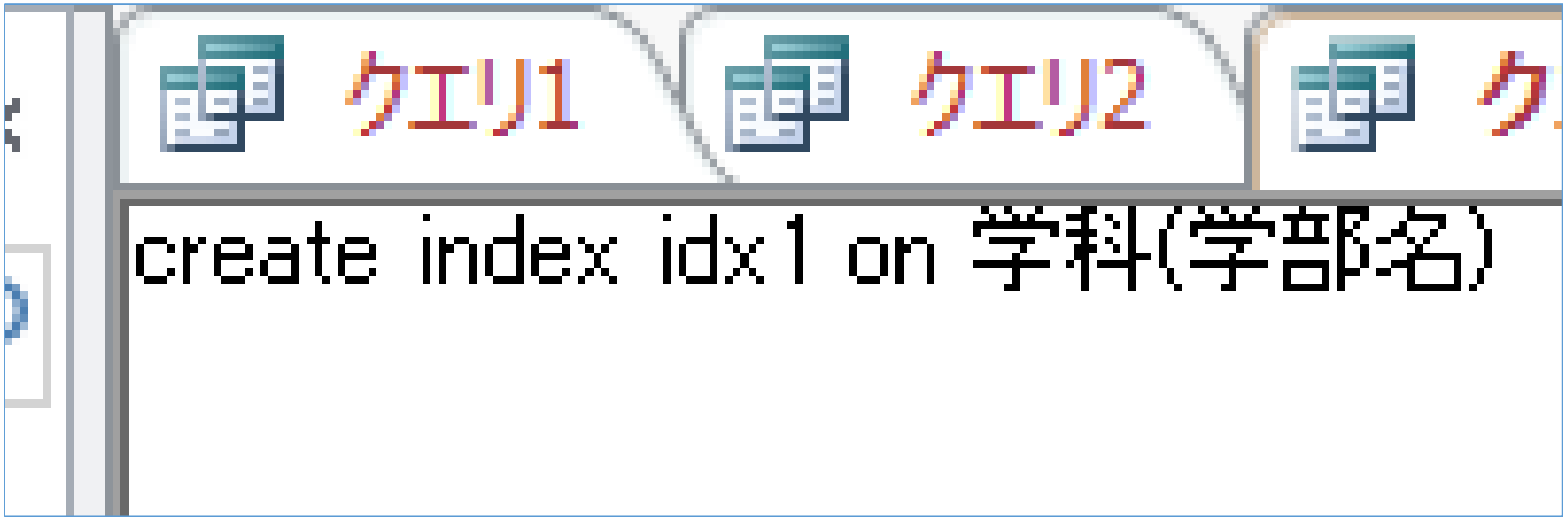
インデックス（索引）とは

インデックス（索引）とは,

テーブルの特定の処理を高速にするための仕掛け

但し、別の処理が遅くなる

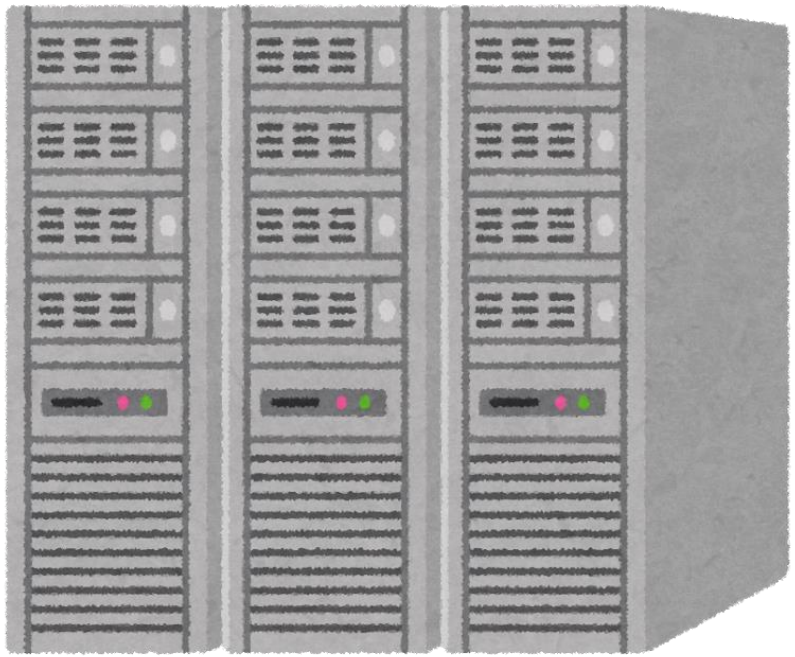
SQL を用いたインデックス（索引）作成例



create index ＜インデックス名＞ on テーブル名(フィールド名)

13-4 データモデル

モデル



データモデル

なぜ？

データを共有などするときは，

データの形をあわせるなど約束が必要

↓

約束のことをデータモデルといいます

データモデル

	約束	特徴
ERモデル、ER図	情報は、**実体**と**関連**の２種類しかない	いろいろなデータベースシステムの種類によらず、同じ
リレーショナルデータベースのデータモデル	１．**テーブル定義**する ２．データベース言語**ＳＱＬ**の演算を使う ３．制約の書き方を統一する	リレーショナルデータタベースだけに使えるやり方

リレーショナルデータベースでは**両方必要**！

関連

◆ １対１

日本の結婚制度

◆ １対多

「小学生」は、１つの「小学校」にしか所属できない。

「小学校」には、たくさんの「小学生」がいる。

◆多対多

顧客が商品を注文する（１対１でも１対多でもない）

テーブル定義に至るストーリー例（トップダウンの場合）

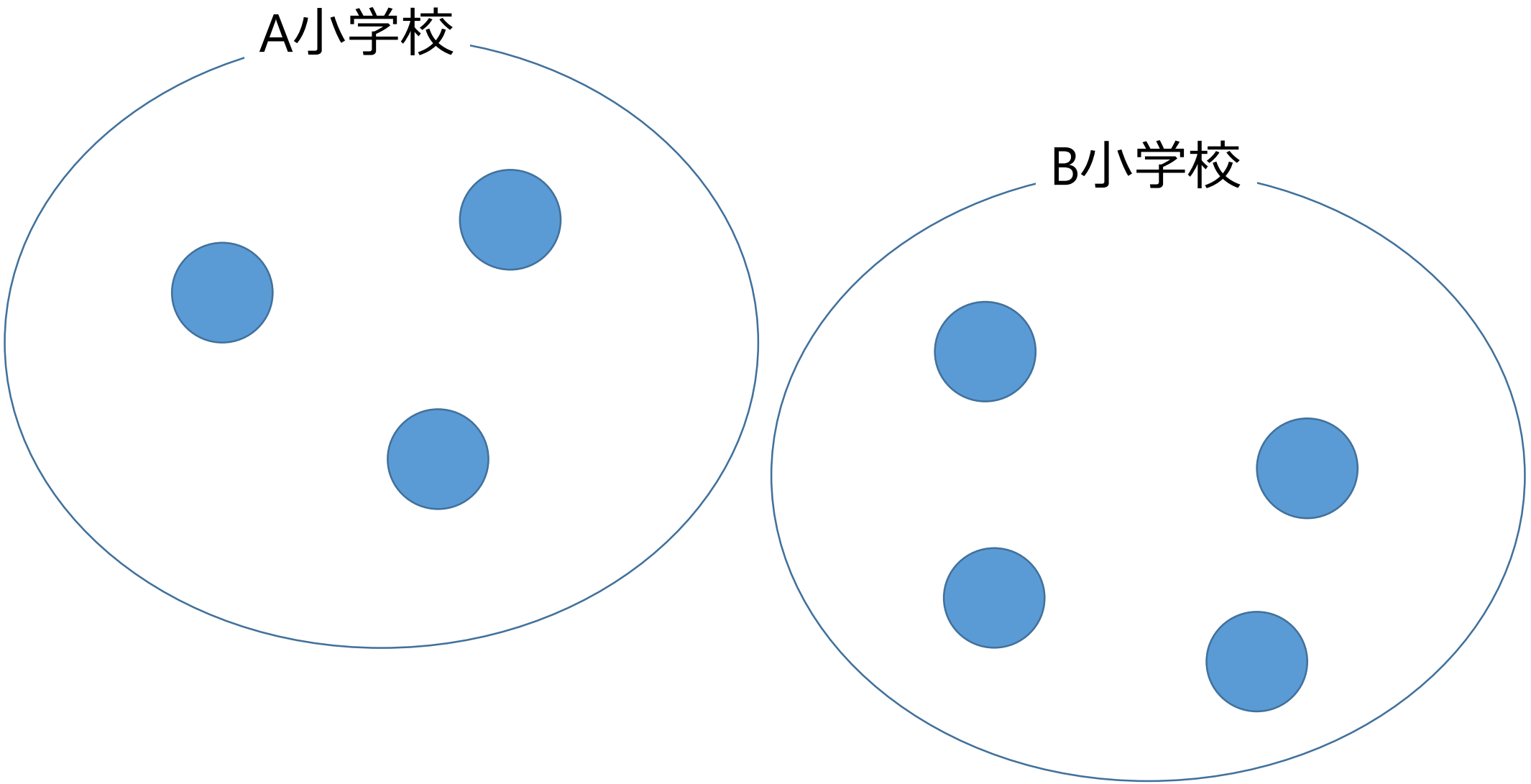
1. 科目の履修についてのデータベースを新しく作りたいと考えた

2.「**科目**」と「**学生**」に関する ER 図を作る

3. テーブル定義を行う

13-4-1 一対多での テーブル定義

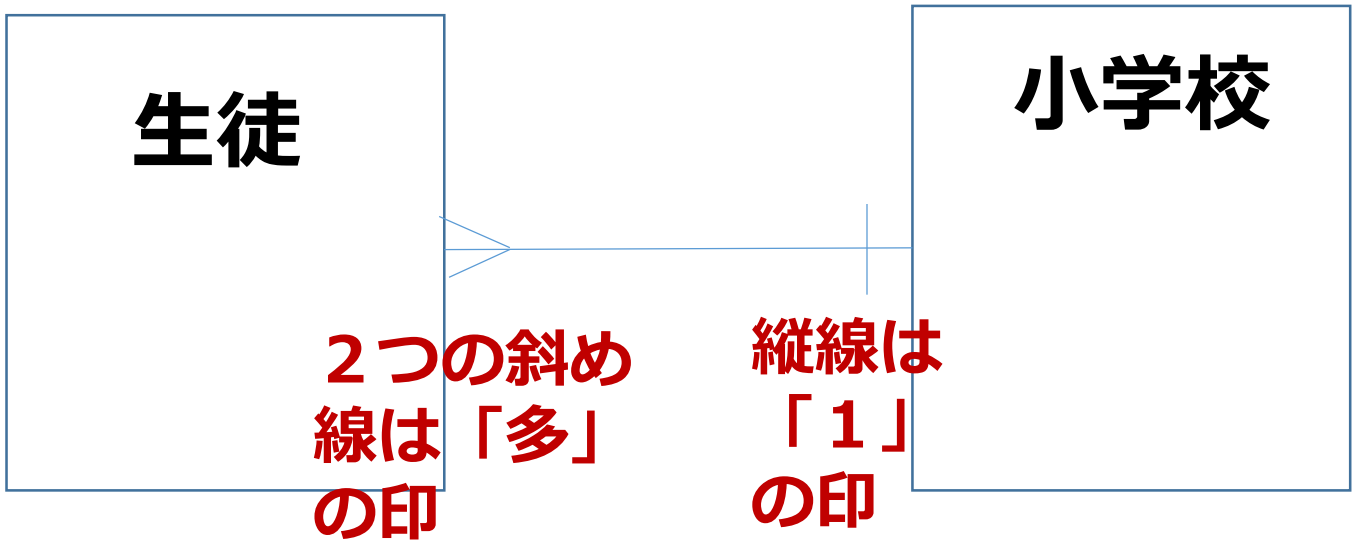
1対多のイメージ



１対多

- それぞれの生徒は、**１つ**の小学校に所属する
- 小学校には、**複数**の生徒が所属する

１対多の対応



テーブル定義

```
create table 小学校 (
    ID integer,
    小学校名 char,
    primary key(ID)
);
```

◆ テーブル名：**小学校**

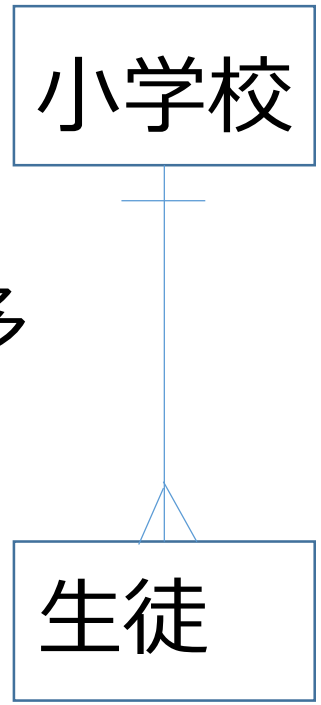
ID	小学校名
1	AA小学校
2	CC小学校
3	BB小学校
4	DD小学校

```
create table 生徒 (
    ID integer,
    氏名 char,
    小学校 integer,
    primary key(ID)
);
```

◆ テーブル名：**生徒**

ID	氏名	小学校
1	徳川家康	1
2	豊臣秀吉	3
3	織田信長	1

ER図とテーブルの関係



一対多

ＥＲ図の例

ＥＲ図の1つの実体に対して
１つのテーブル

◆ テーブル名：**小学校**

ID	小学校名
1	AA小学校
2	CC小学校
3	BB小学校
	小学校

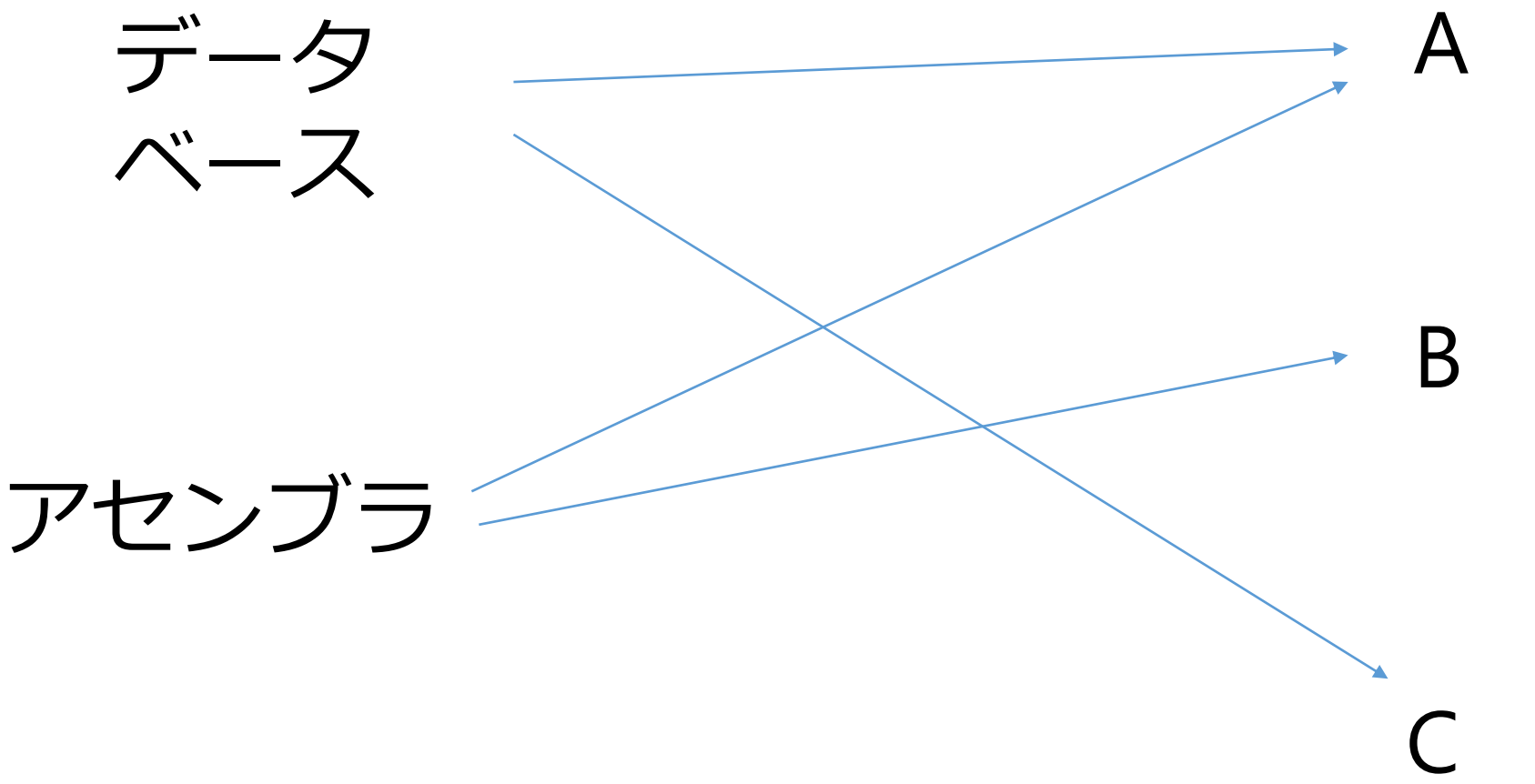
ＥＲ図の実体と関連をあわせて
１つのテーブル

◆ テーブル名：**ビデオ**

ID	氏名	小学校
1	徳川家康	1
2	豊臣秀吉	3
3	織田信長	1

13-4-2 多対多でのテーブル定義

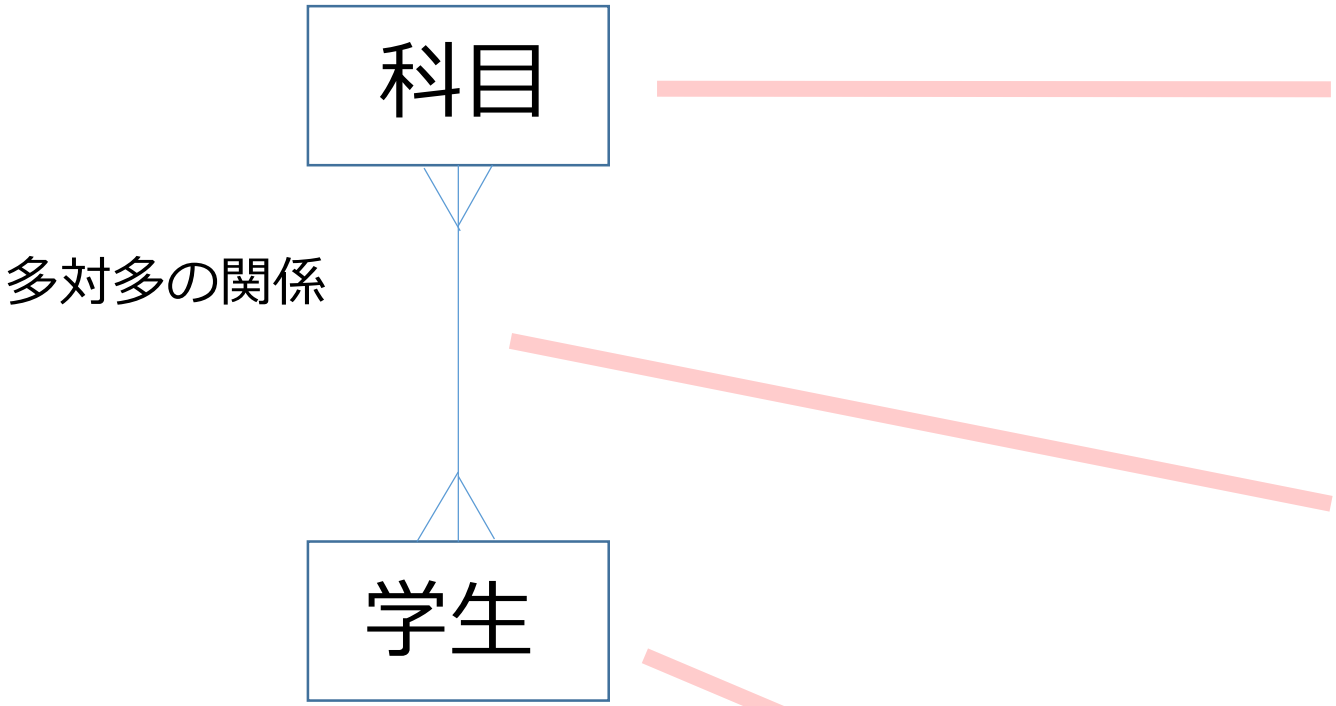
多対多の対応のイメージ



科目

学生

ER図とテーブルの関係



ＥＲ図の例

◆ テーブル名：**科目**

科目番号	科目名
1	AA
2	CC
3	BB

◆ テーブル名：**履修**

学生番号	科目番号
1	1
1	3
2	1
2	2

◆ テーブル名：学生

学生番号	氏名
1	ＡＡ
2	ＸＸ

テーブル定義

```
create table 科目 (
    科目番号 integer,
    氏名 char,
    primary key(科目番号)
);
```

```
create table 履修 (
    学生番号 integer,
    科目番号 integer
);
```

```
create table 学生 (
    学生番号 integer,
    氏名 char,
    primary key(学生番号)
);
```

◆ テーブル名：**科目**

科目番号	科目名
1	AA
2	CC
3	BB

◆ テーブル名：**履修**

学生番号	科目番号
1	1
1	3
2	1
2	2

◆ テーブル名：学生

学生番号	氏名
1	A A
2	X X

SQL は簡潔（シンプル）

- **制御構造がない**
  if, else, for, while がない
- **データの種類は「テーブル」のみ**
  配列、構造体、ポインタなどはない
- **短い**
  ほとんどの場合、数行程度で済む
- **簡単**
  プログラム製作　＝
  問題の解き方を考えて、 正しく動くように手順をプログラムで書く
  SQLだと、これが簡単に済む
  （SQLにはリレーショナルデータベースに**特化**した機能しかないので）