

**データベース実習**

# **06. データベースの設計**

株式会社ジードライブ

# 今回学ぶこと

---

- データベース設計の概要
- ER図の表記ルール
- テーブル同士の関連
- テーブルの正規化
- MySQL WorkbenchによるER図の作成

# DB設計の大まかな手順

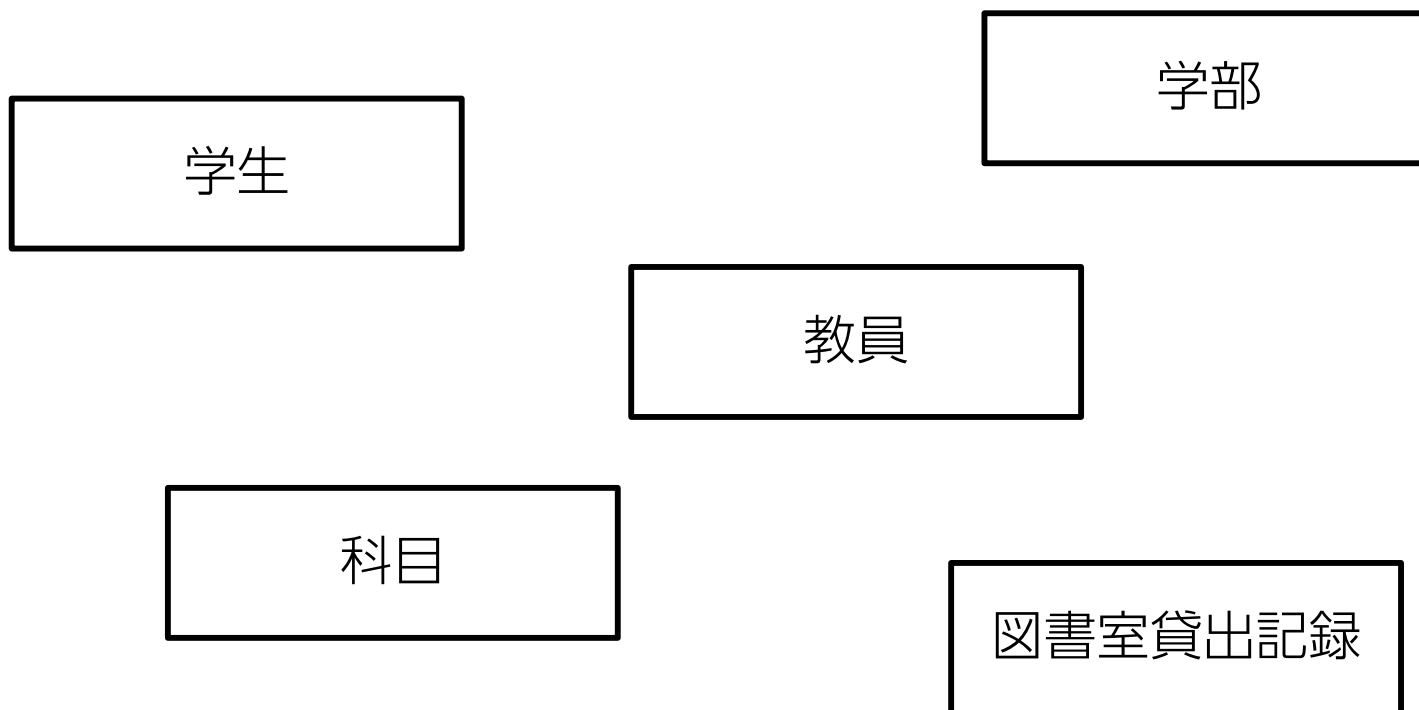
---

- ① エンティティを抽出する
  - 要件から必要になりそうなテーブルを考える
- ② エンティティを定義する(ER図の作成)
  - 各テーブルがどのようなカラムを持つか考える
  - テーブル同士の関係性を考える
- ③ テーブルの正規化
  - テーブルのカラムを整理し、必要に応じてテーブルを分割する
- ④ ER図の完成

# エンティティの抽出

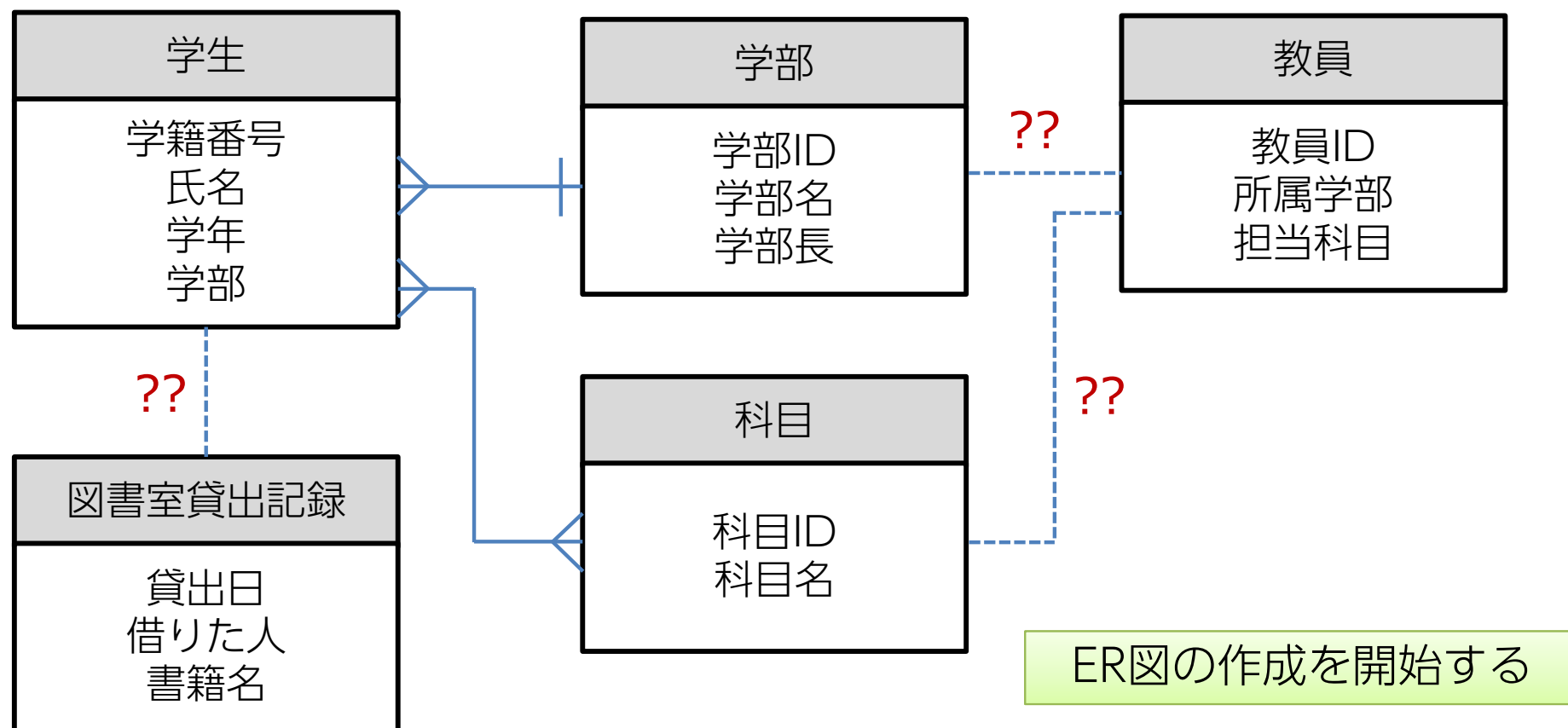
---

- システムの要件から、必要になりそうなテーブルを考える作業



# エンティティの定義

- システムの目的に応じ、各テーブルのカラムやテーブル同士の関連を考えていく



# ER図とは

- Entity Relationship Diagram (ERD : 実体関連図)
  - 実体(Entity)とその関連(Relationship)を表現する図
  - データベースの設計に使われる図法
    - ⇒ エンティティはテーブルの原型となる図
- ER図には用途に応じて様々な表記法がある
  - Peter Chen記法  
アイデフ ワン エックス
  - IDEF1X記法
  - IE記法(Information Engineering)

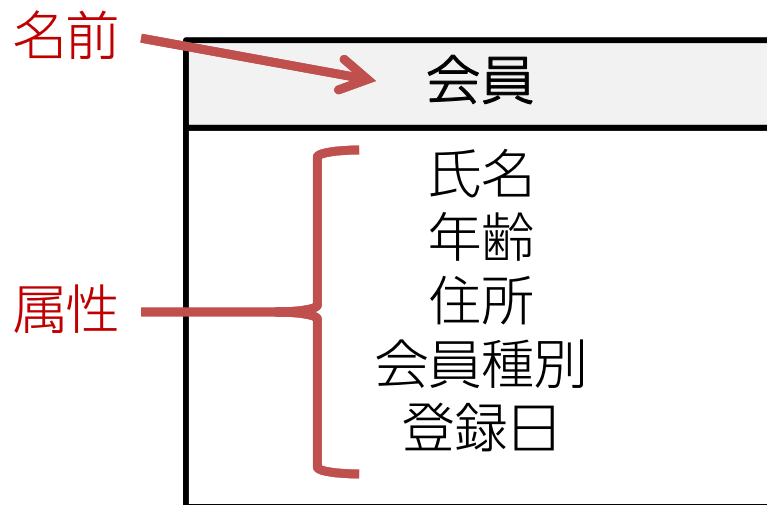


例：Peter Chen記法によるERD  
長方形がエンティティ、菱形が関連を示す

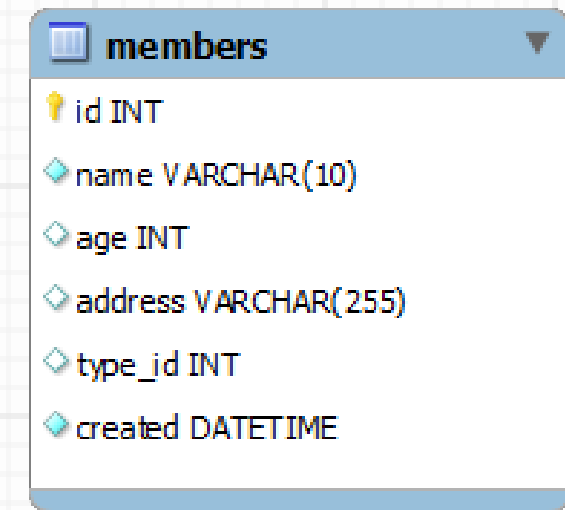
テーブルの設計図／テーブル同士の関連図と考えてよい

# IE記法

- エンティティ(テーブルの原型となる図)は、四角いボックスで表す
  - ボックスの上部にエンティティの名前を記す
  - ボックスの内部にエンティティの持つ属性を記す



エンティティを元に  
実際のテーブル名や  
カラム名を決める

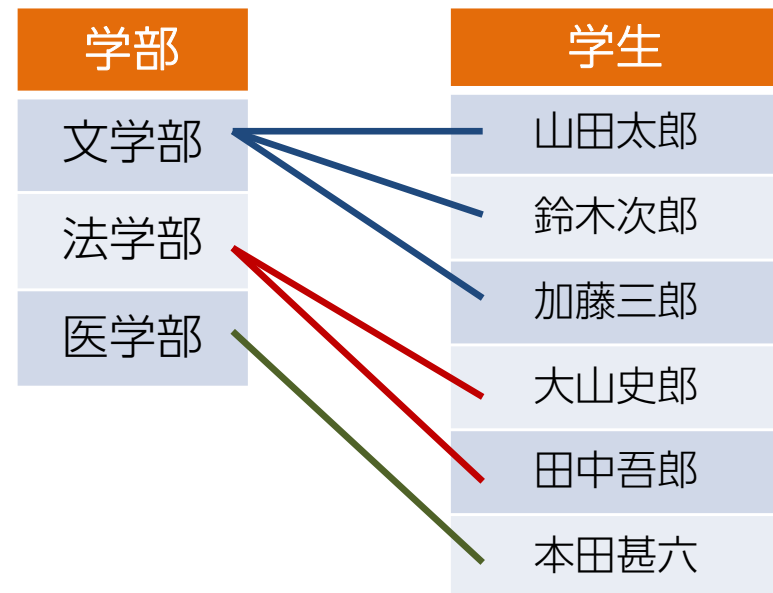


MySQL Workbench の  
モデリングツールによる作図

# 関連の表現

- エンティティ同士 (テーブルの同士)の関係性を関連 (Relationship)と呼ぶ
- 関連には以下のような種類がある
  - 1対1 (One to One)
  - 1対多 (One to Many)
  - 多対多 (Many to Many)

このような対応関係を  
**カーディナリティ(多重度)**という



例：1対多の関係



# 多重度の表現方法

- 多重度の表現に使う記号

記号	意味
○	ゼロ
	1
≧	多

- 実際には  $\geq$  (1以上)のように上限と下限を指定する

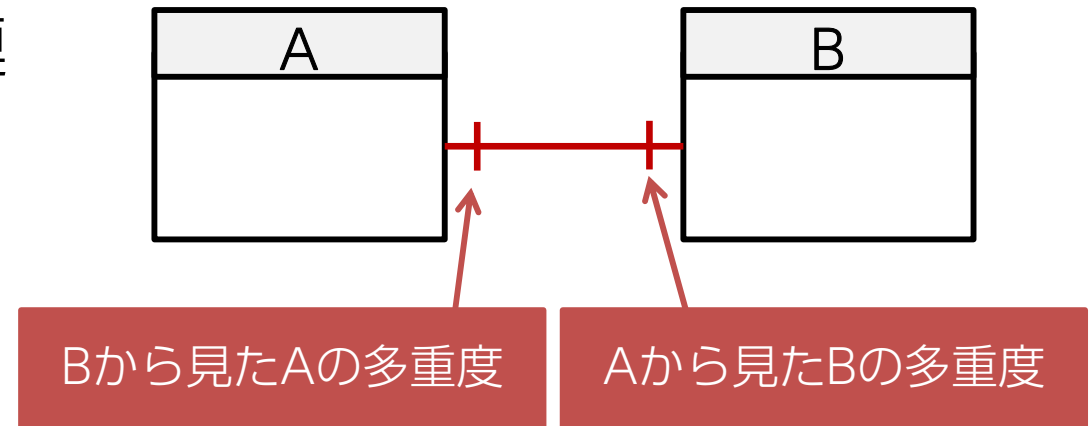
# 定義や関連を考えるにあたって

- できるだけ具体的なデータを想定することで考えやすくなる
  - 足りないカラムや重複するカラムが見えてくる
  - 各カラムに入るデータ型や文字数などが見えてくる

学籍番号	氏名	学年	学部
zd18A2156	金田 新	3	文学部
zd19A2535	アレン・ブレンダー	2	文学部
zd20B5675	伊達 幸代	1	社会学部
zd20C3564	北村 ジョージ	4	経済学部
zd21D5482	澤山 君子	3	社会学部

# 1対1の関連 (1)

- 右図で示されるような関連を1対1の関連という
  - Aから見てBは1個存在する
  - Bから見てAは1個存在する



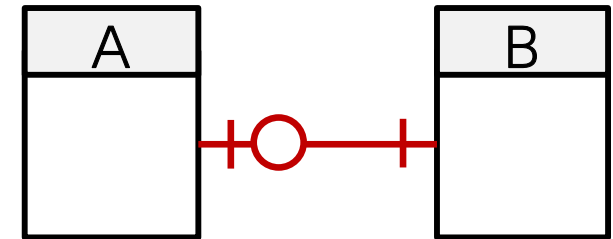
例：学籍番号テーブルと学生テーブル

学籍番号			学生	
1	S041807001	—	1	山田太郎
2	S041807002	—	2	鈴木次郎
3	S041807003	—	3	加藤三郎

実際には分割不要なテーブルが分かれている状態と考えられる

# 1対1の関連（2）

- 1対1の関連のバリエーションとして、右図のような場面も考えられる
  - Aから見てBは1個存在する
  - Bから見てAは0個または1個存在する

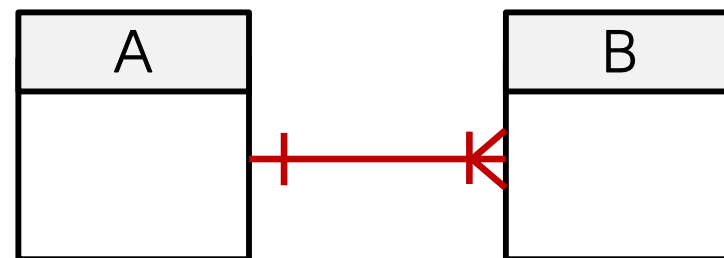


例：学生とロッカー（定員分のロッカーが存在し、定員に満たないとロッカーが余る）

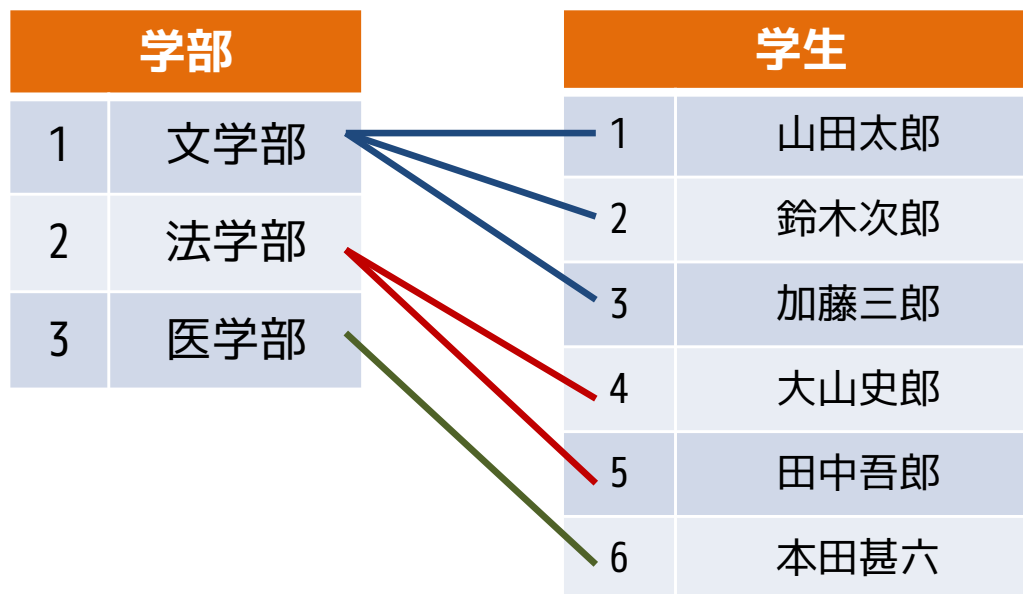
学生		ロッカー
1	山田太郎	L001
2	鈴木次郎	L002
3	加藤三郎	L003
	使用者なし	L004
	使用者なし	L005

# 1対多の関連 (1)

- 右図は 1 対多の関連を示している
  - Aから見てBは1個以上存在する
  - Bから見てAは1個存在する

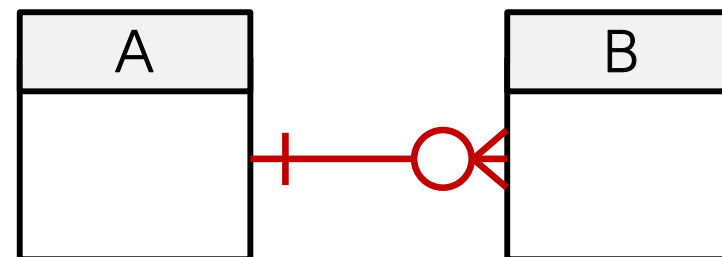


例：学部と学生 (必ずどの学部にも学生がいる場合)

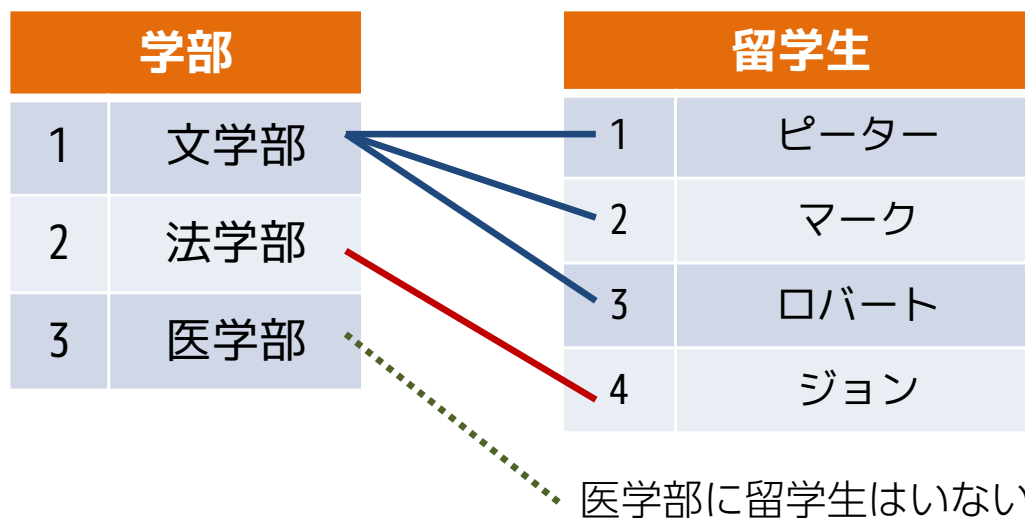


# 1対多の関連（2）

- 1 対多の関連のバリエーションとして、右図のような場面も考えられる
  - Aから見てBは0個以上存在する
  - Bから見てAは1個存在する

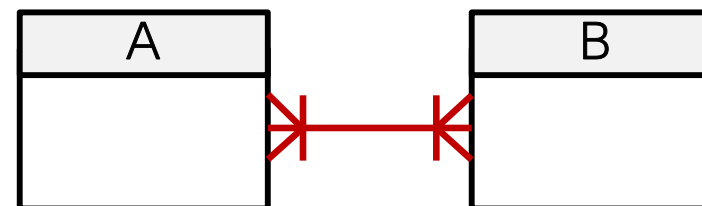


例：学部と留学生（留学生がいない学部もある場合）

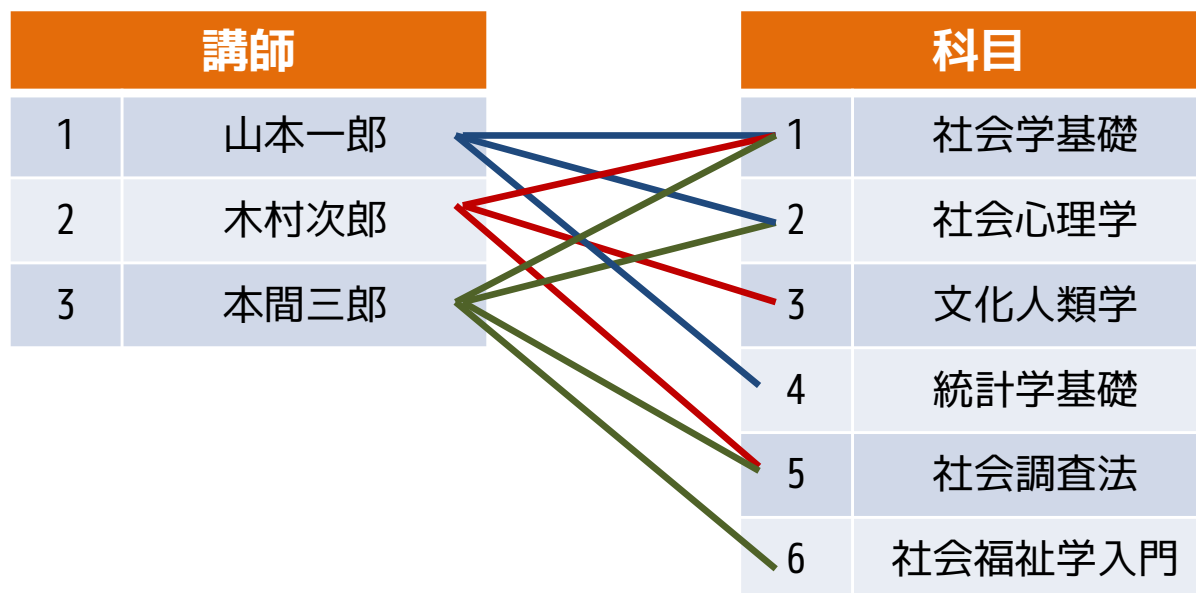


# 多対多の関連

- 多対多の関連は右図のように表すことができる
  - Aから見てBは1個以上存在する
  - Bから見てAは1個以上存在する

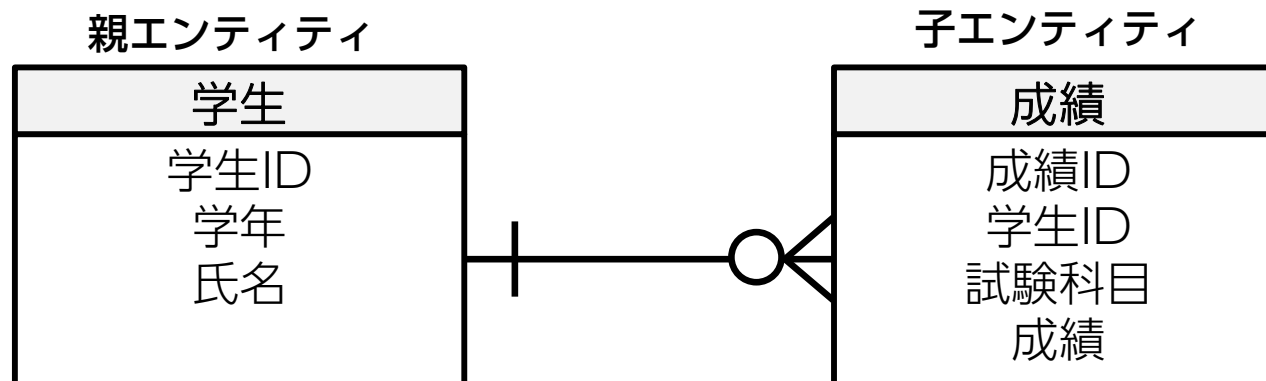


例：講師と科目



# 親エンティティと子エンティティ

- 1対1や1対多の関連を持つエンティティは、片方が**親エンティティ**でもう片方が**子エンティティ**となる
  - 1対多の場合は1の側が親エンティティ
  - 1対1の場合はエンティティの意味から、どちらが親か考える

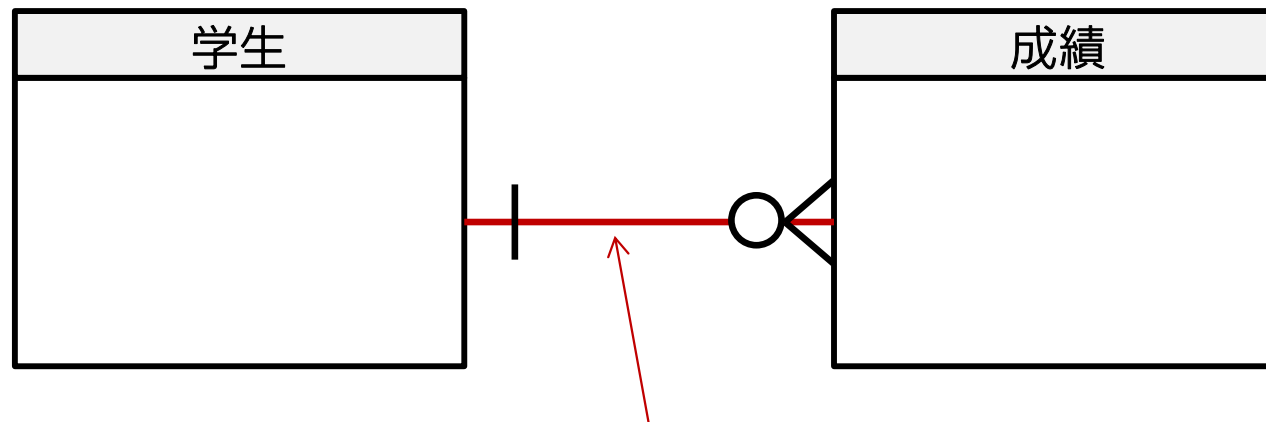


学生ID	学年	氏名	成績ID	学生ID	試験科目	成績
S01	4	山田太郎	1	S01	社会学基礎	C
S02	3	鈴木次郎	1	S02	社会学基礎	A
S03	1	加藤三郎	2	S01	文化人類学	B



# 依存関係と非依存関係

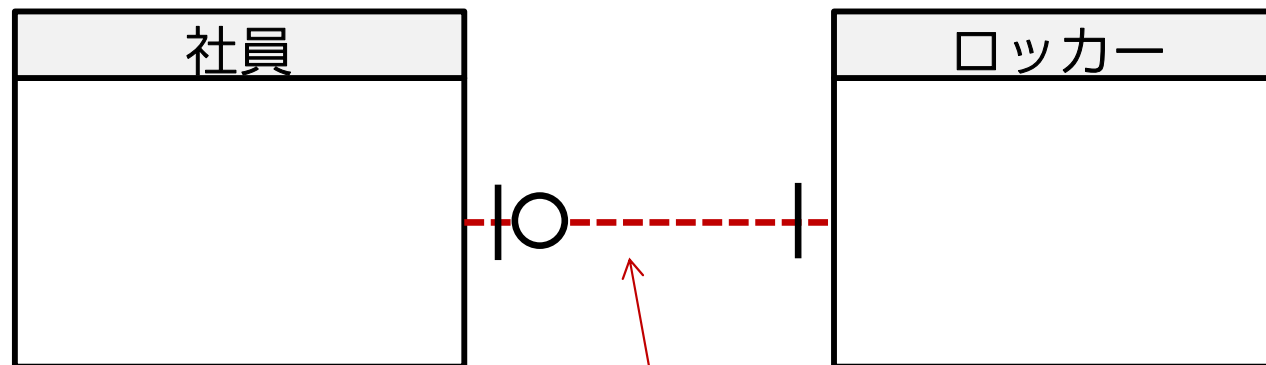
- 依存関係 (Identifying Relationship)
  - 親エンティティが存在しないと子エンティティも存在し得ない関係(学生と成績など)
  - 子エンティティの主キーが親エンティティの全ての主キーを含む(複合主キー)関係



ER図では実線で表す

# 依存関係と非依存関係

- 非依存関係 (Non-identifying Relationship)
  - 親エンティティが存在しなくても子エンティティが存在できる関係(学生とロッカーなど)
  - 依存関係のケースに該当しない関係



ER図では破線で表す

# テーブルの正規化

- テーブルから重複している項目やデータを整理し、扱いやすい形に設計すること
  - 必要に応じてテーブルを分割する
- 正規化のメリット
  - データ量の低減
  - テーブルのメンテナンス性の向上

非正規形のテーブルの例：図書館の貸出記録（書籍の項目や利用者が重複している）

貸出ID	日付	利用者ID	利用者名	書籍ID1	書籍名1	書籍ID2	書籍名2
1	2015-03-17	1	山田太郎	B001	坊ちゃん	B002	こころ
2	2015-03-18	2	鈴木次郎	B004	蜘蛛の糸		
3	2015-03-18	1	山田太郎	B003	羅生門		
...	...	...	...	...	...	...	...

# テーブルの正規化

- 正規化後の例

貸出テーブル

貸出ID	日付	利用者ID
1	2015-03-17	1
2	2015-03-18	2
3	2015-03-18	1
4	...	...

利用者テーブル

利用者ID	利用者名
1	山田太郎
2	鈴木次郎
3	...
4	...

関連  
(リレーション)

貸出-書籍テーブル

貸出ID	書籍ID
1	B001
1	B002
2	B004
3	B003
4	...

書籍テーブル

書籍ID	書籍名
B001	坊ちゃん
B002	こころ
B003	羅生門
B004	蜘蛛の糸
B005	...

# 正規化のチェックポイント

---

- テーブルが以下のような状態になっている場合、十分に正規化が行われていないと考えられる  
⇒ テーブル設計を見直す必要がある
- ① 主キーが存在しない
  - ② 同じようなカラムが複数存在する
  - ③ 一つのカラム内に複数のデータが入っている
  - ④ 主要なカラムに直接的には関係のない情報が含まれる
  - ⑤ 他のカラムから導き出せる項目が含まれている

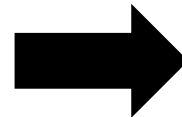
# ①主キーが存在しない

- 行を一意に識別するための主キーが存在しないと、データの更新や削除が難しくなってしまう



主キーが存在しない

名前	住所
山田太郎	東京都新宿区
佐藤花子	神奈川県横浜市
鈴木次郎	東京都豊島区
山田太郎	千葉県市原市



主キーが存在する

ID	名前	住所
1	山田太郎	東京都新宿区
2	佐藤花子	神奈川県横浜市
3	鈴木次郎	東京都豊島区
4	山田太郎	千葉県市原市

## ②カラムの重複 ③カラム内に複数データ

✖ 同じようなカラムが重複している

ID	名前	部活1	部活2	部活3
1	山田太郎	卓球部		
2	佐藤花子	剣道部	茶道部	華道部
3	鈴木次郎	卓球部	茶道部	

✖ 1つのカラムに複数のデータが存在する

ID	名前	部活
1	山田太郎	卓球部
2	佐藤花子	剣道部、茶道部、華道部
3	鈴木次郎	卓球部、茶道部

## ②カラムの重複 ③カラム内に複数データ

○ 適切にテーブルを分けて管理する

生徒テーブル

生徒ID	名前
s1	山田太郎
s2	佐藤花子
s3	鈴木次郎

所属テーブル

生徒ID	部活ID
s1	c1
s2	c2
s2	c3
s2	c4
s3	c1
s3	c3

部活テーブル

部活ID	部活名
c1	卓球部
c2	剣道部
c3	茶道部
c4	華道部



## ④ 主要カラムに関係のない情報

✖ 部活に直接的な関係のない情報が含まれている

ID	部活名	顧問教員	担当科目	出身
1	卓球部	本田三郎	生物	埼玉県
2	剣道部	加藤良子	国語	東京都
3	茶道部	木村五郎	数学	神奈川県

○ 適切にテーブルを分けて管理する

部活テーブル

ID	部活名	顧問ID
1	卓球部	t1
2	剣道部	t2
3	茶道部	t3

教員テーブル

ID	教員氏名	担当科目	出身
t1	本田三郎	生物	埼玉県
t2	加藤良子	国語	東京都
t3	木村五郎	数学	神奈川県

## ⑤他のカラムから導き出せる項目



合計は1～3年生の人数から導き出せるのでカラムとしては不要

ID	部活名	1年生	2年生	3年生	合計
1	卓球部	8人	5人	6人	19人
2	剣道部	3人	4人	4人	11人
3	茶道部	2人	1人	3人	6人



必要に応じて、SELECT文で対応する

```
SELECT ID, 部活名, 1年生, 2年生, 3年生,  
       (1年生 + 2年生 + 3年生) AS 合計  
FROM 部活テーブル;
```

---

# 正規化の手順

# テーブルの正規化

---

- 正規化(正規形)は、以下のような段階に分かれている
  - 第1正規化
  - 第2正規化
  - 第3正規化
  - ボイス・コッド正規化
  - 第4正規化
  - 第5正規化
- 一般的には、第3正規化まで行われる

# 正規化前の状態

- テーブル内に重複している項目やデータが見られる

非正規形のテーブルの例：図書館の貸出記録（書籍の項目や利用者が重複している）

貸出ID	日付	利用者ID	利用者名	書籍ID1	書籍名1	書籍ID2	書籍名2
1	2015-03-17	1	山田太郎	B001	坊ちゃん	B002	こころ
2	2015-03-18	2	鈴木次郎	B004	蜘蛛の糸		
3	2015-03-18	1	山田太郎	B003	羅生門		
...	...	...	...	...	...	...	...

# 第1正規化

- 重複する項目を別テーブルに分離して整理する

重複

貸出ID	日付	利用者ID	利用者名	書籍ID1	書籍名1	書籍ID2	書籍名2
1	2015-03-17	1	山田太郎	B001	坊ちゃん	B002	こころ
2	2015-03-18	2	鈴木次郎	B004	蜘蛛の糸		

第1正規化

複合主キー:  
複数の項目の組を主キーとすることで一意に識別できる

貸出ID	日付	利用者ID	利用者名
1	2015-03-17	1	山田太郎
2	2015-03-18	2	鈴木次郎

貸出ID	書籍ID	書籍名
1	B001	坊ちゃん
1	B002	こころ
2	B004	蜘蛛の糸

## 第2正規化

- 複合主キーの一部だけから決まる(部分関数従属する)項目を別テーブルに分離する



# 第3正規化

- 主キー以外の項目によって決まる(関数従属する)項目を別テーブルに分離する

貸出ID	日付	利用者ID	利用者名
1	2015-03-17	1	山田太郎
2	2015-03-18	2	鈴木次郎

利用者名は、主キーではない  
利用者IDに從属する

第3正規化

貸出ID	日付	利用者ID
1	2015-03-17	1
2	2015-03-18	2

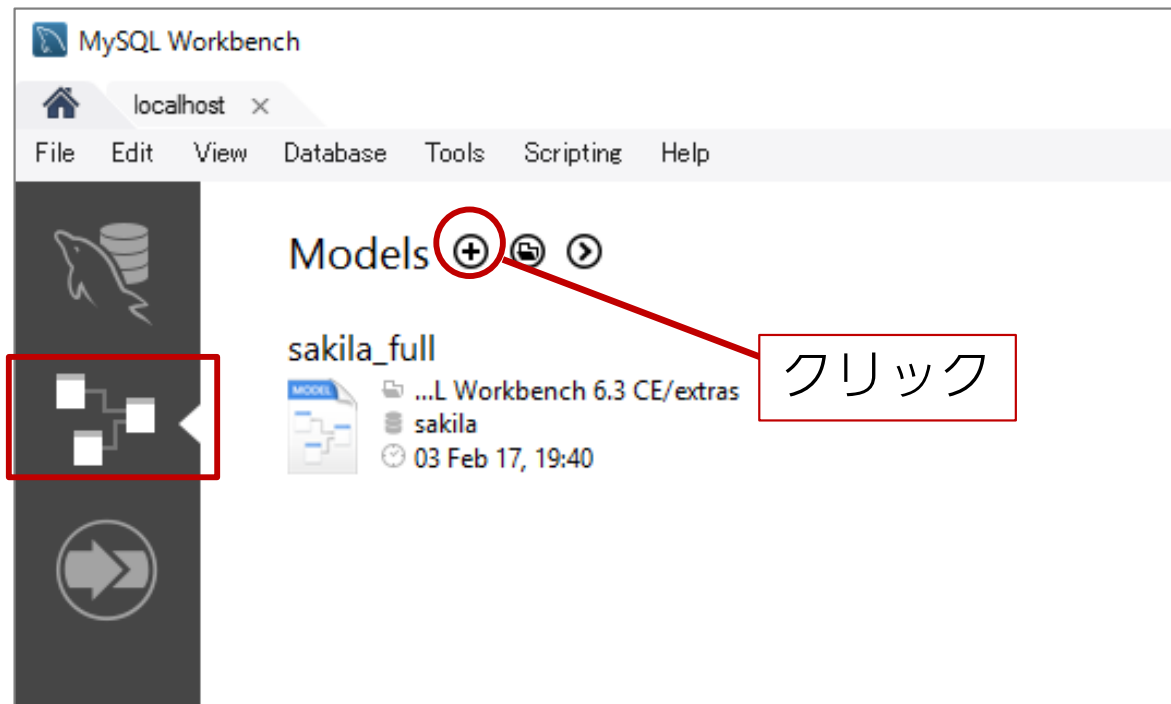
利用者ID	利用者名
1	山田太郎
2	鈴木次郎



# MySQLWorkbenchによる ER図の作成

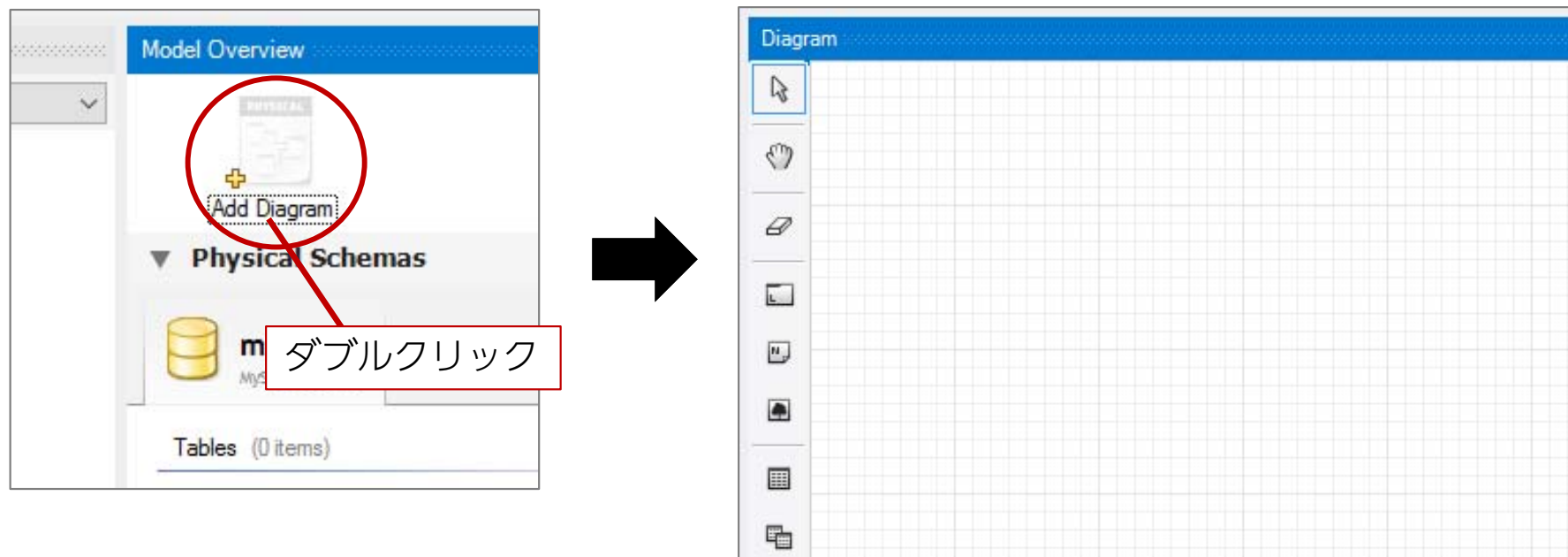
# 新規ER図の作成

- Models の ⊕ をクリック




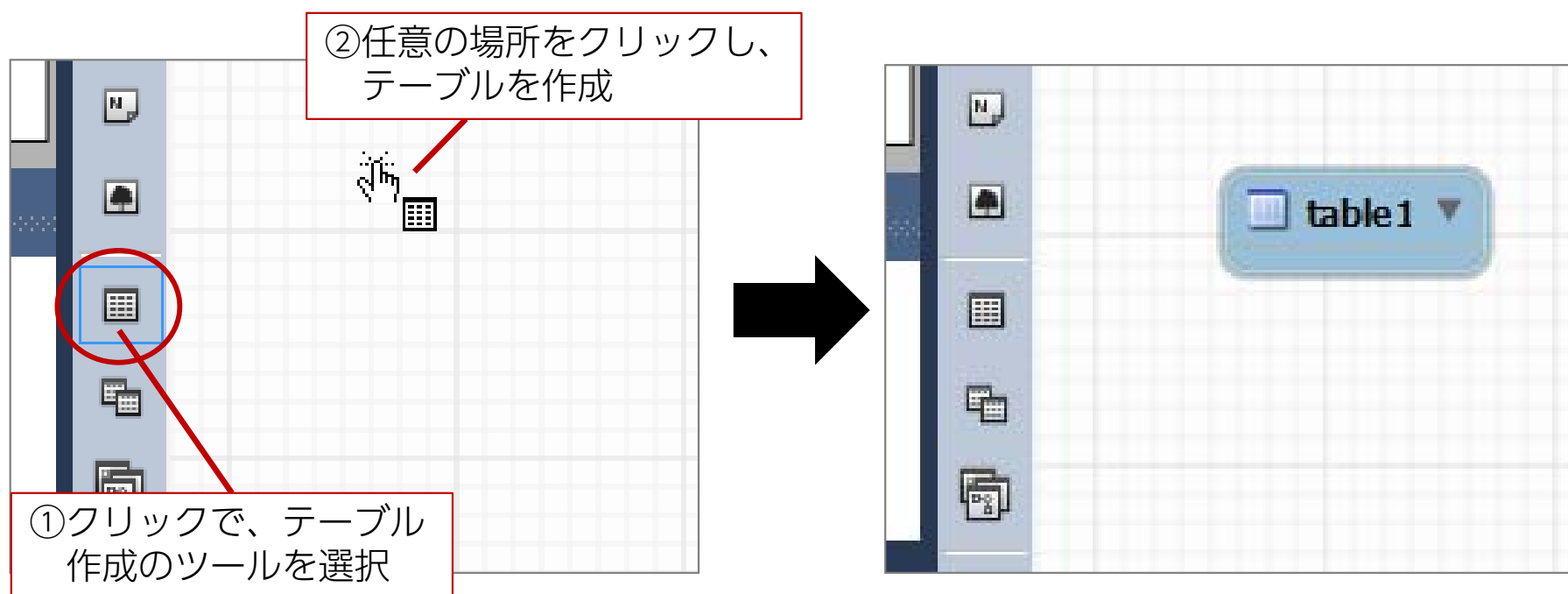
# 新規ER図の作成

- [Add Diagram]をダブルクリックすると新しいシートが作成される



# テーブルの作成

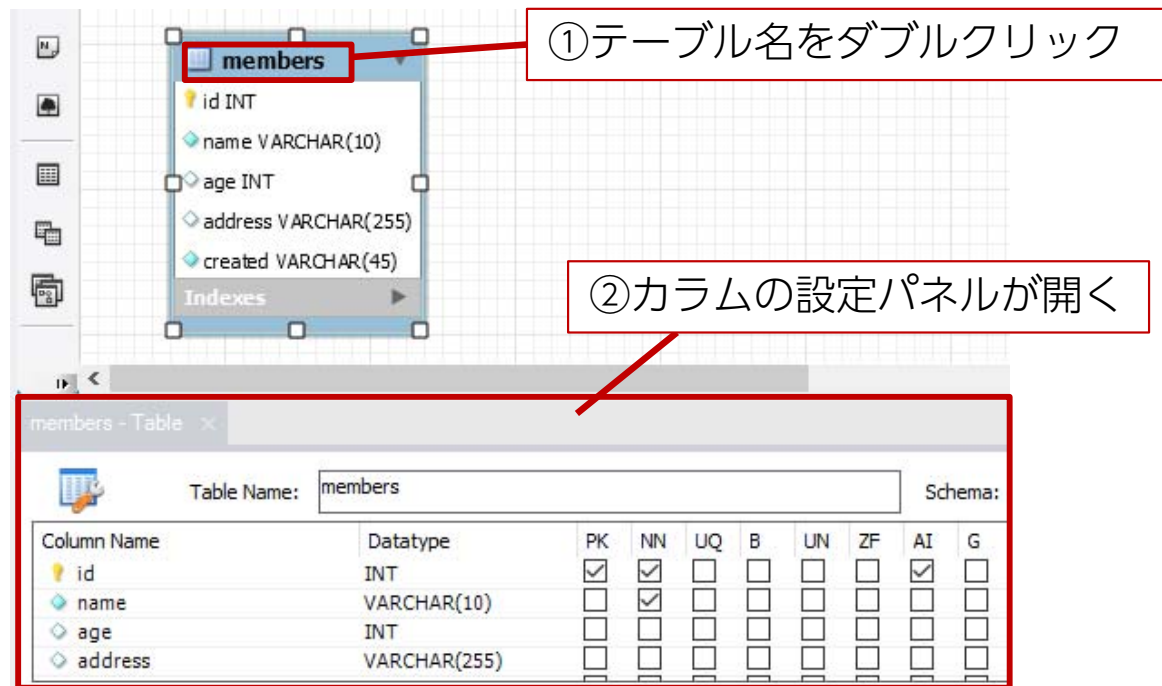
- テーブルの作成
  - シート左側のパレットから  を選び、テーブルを配置したい場所をクリックする



# テーブルの作成

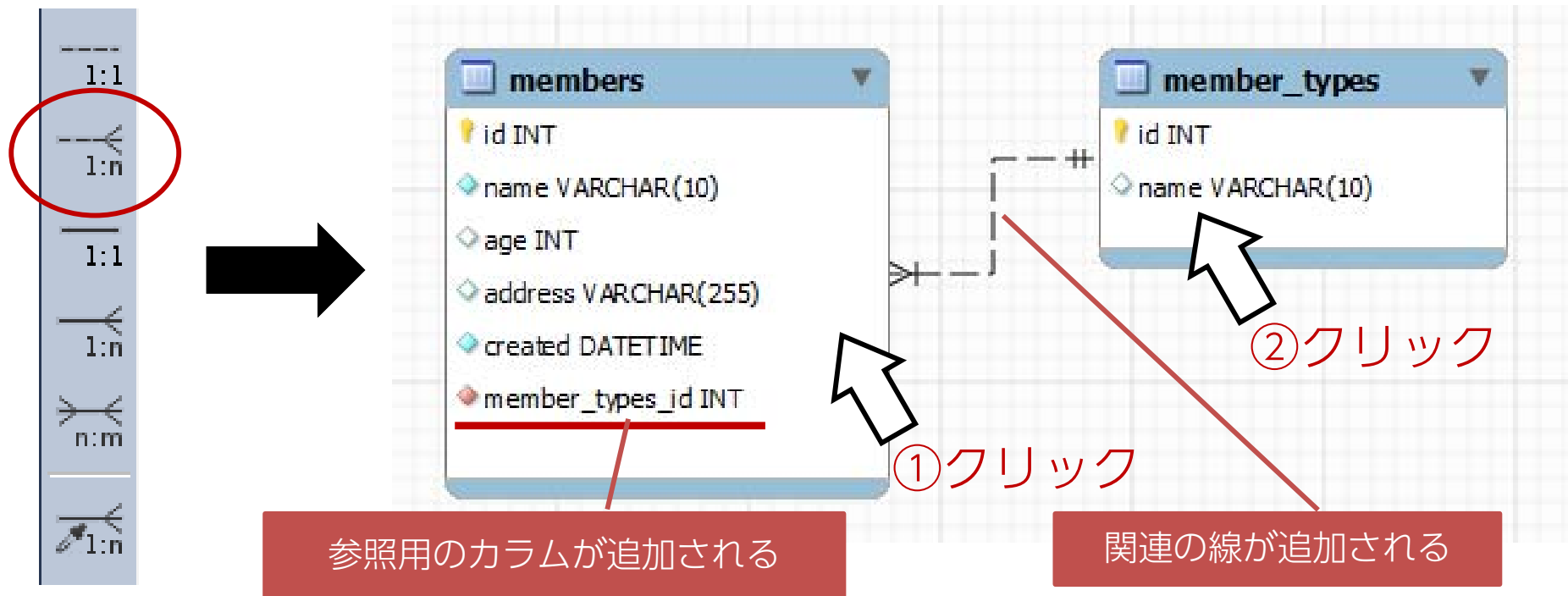
- カラムの設定

- 図中のテーブル名をダブルクリックするとウィンドウ下部に表示されるパネル内でカラムの設定を行う



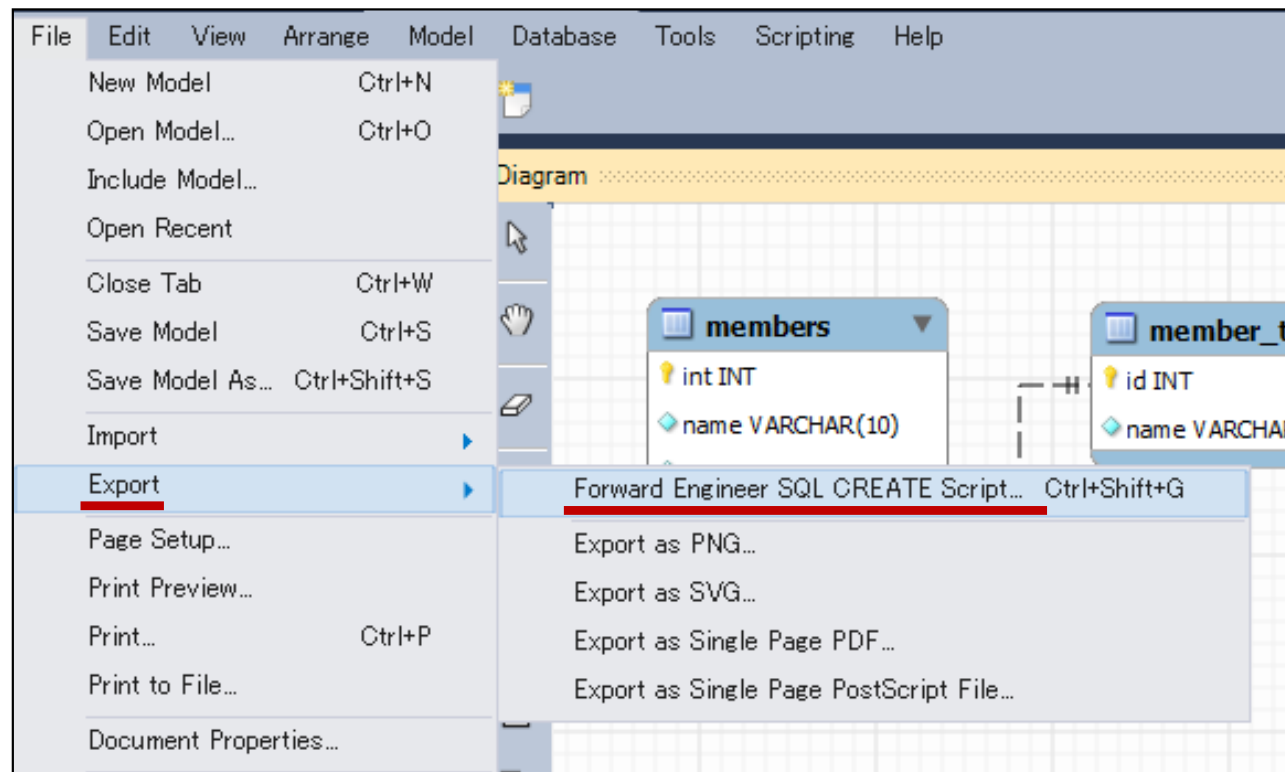
# テーブル間の関連の作成

- シート左にあるパレットから関連の種類を選び、①参照する側、②参照される側、の順番でテーブルをクリックする



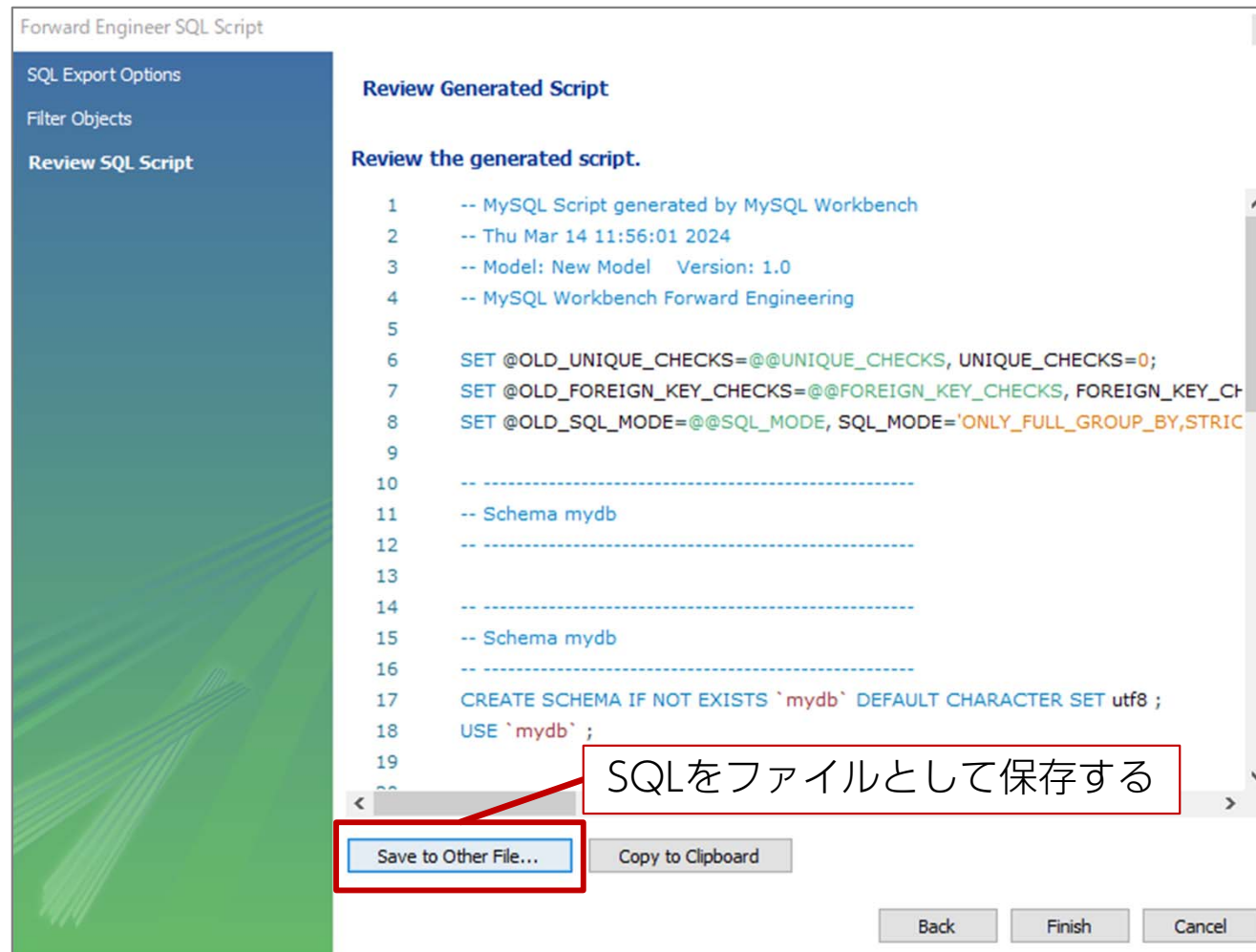
# ER図からのSQLの生成

- [File] ⇒ [Export] ⇒  
[Forward Engineer SQL CREATE Script...] を選択する



# ER図からのSQLの生成

- [Next]ボタンで進んでいくと、SQLが表示される





# 練習

---

- 練習06-1
- 練習06-2