



第7回: SQL言語(データベース定義)

上智大学理工学部情報理工学科
高岡詠子

No reproduction or republication without written *permission*.
許可のない転載、再発行を禁止します



Schedule

	日程	内容
第1回	10月6日	ガイダンス, データベースとは?
第2回	10月13日	三層スキーマ, データモデル, データベース設計のための仕組み
第3回	10月20日	概念設計: 概念モデルとERモデル, 論理設計へ
第4回	10月27日	論理設計と正規化
第5回	11月10日	正規化, 物理設計
第6回	11月17日	物理設計
第7回	11月24日	SQL言語(データベース定義)
第8回	12月1日	SQL言語(データベース操作)
第9回	12月8日	SQL
第10回	12月15日	SQL言語(ビュー定義など)
第11回	12月22日	データベース管理システム: トランザクション処理
第12回	1月5日	データベース管理システム: 同時実行制御, 排他制御
第13回	1月12日	同時実行制御, 排他制御, デッドロック
第14回	1月19日	データベース技術動向, リレーショナル代数, まとめ



今日の授業

- ▶ テーブル設計に基づく必要ディスクスペース量の計算
- ▶ データベース言語SQL
- ▶ データベース定義
 - ▶ データベースやテーブルの作成・削除
 - ▶ テーブルの作成・制約条件を付けて作成
 - ▶ テーブルの名前変更など



受注表

* 受注番号
 * 得意先コード
 * 商品コード
 受注個数
 納品日

各納品日にそれぞれの得意先には
 一種類の商品しか受け付けない



得意先コードと納品日の組合せは
 一意でなければならない

列名称(属性)	受注番号	得意先コード	商品コード	受注個数	納品日
データ型	INT	CHAR	INT	INT	DATE
最大データ長	4	5	3	5	7
キー種	PK	FK1	FK2		
一意性	①	②			②
依存先		得意先表	商品表		
入力必須	NN1	NN2	NN3		
平均データ長	4	5	3	10	7

ツアープラン

* プランNO

* プラン名

* 企画会社コード

* 出発日

特記事項

企画会社

* 企画会社コード

* 社名

担当者

取引開始年月



NOT NULL制約

PRIMARY制約

UNIQUE制約

REFERENCE制約

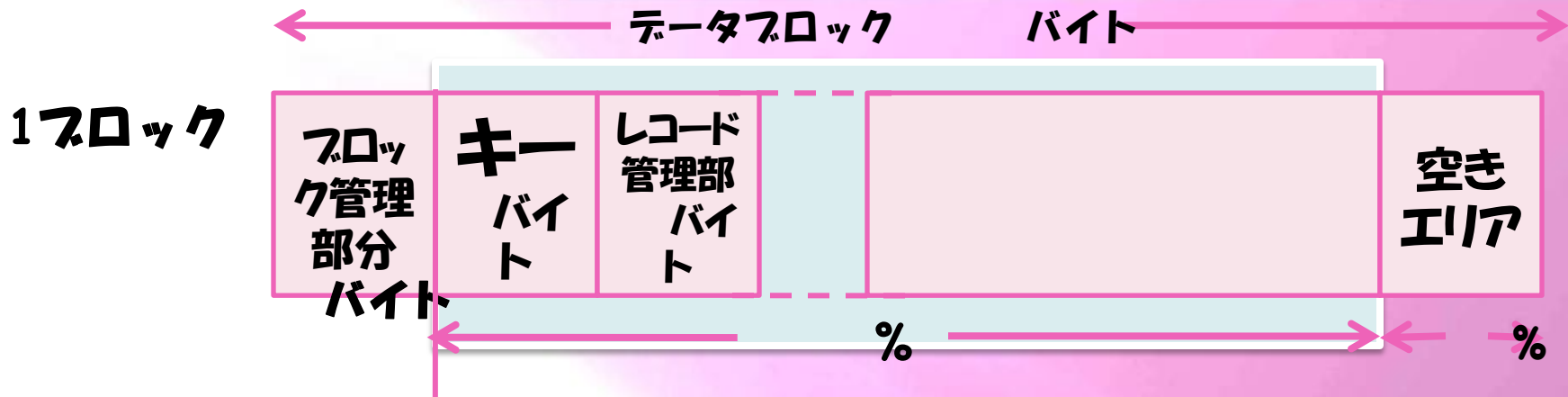
列名称 (属性)	プランNO	プラン名	企画会社コード	出発日	特記事項
データ型	INT	VARCHAR	CHAR	DATE	VARCHAR
最大データ型	12	60	7	7	1000
キー種	PK		FK1		
一意性	①				
依存先			企画会社		
入力必須	NN1	NN2	NN3	NN4	
平均データ長	10	45	7	7	300

必要ディスクスペース量の計算

➡ ツアープログラムのPKであるプログラムNo列をインデックスキー列とした場合の**インデックス**の容量を見積もいなさい。用件は以下の通り

- ➡ テーブル行数: 80000
- ➡ PCTFREE 30%
- ➡ ブロックサイズ 4K(4096バイト)
- ➡ ブロック管理部分 64バイト
- ➡ インデックスレコード長 10バイト
- ➡ インデックスレコード管理部 16バイト



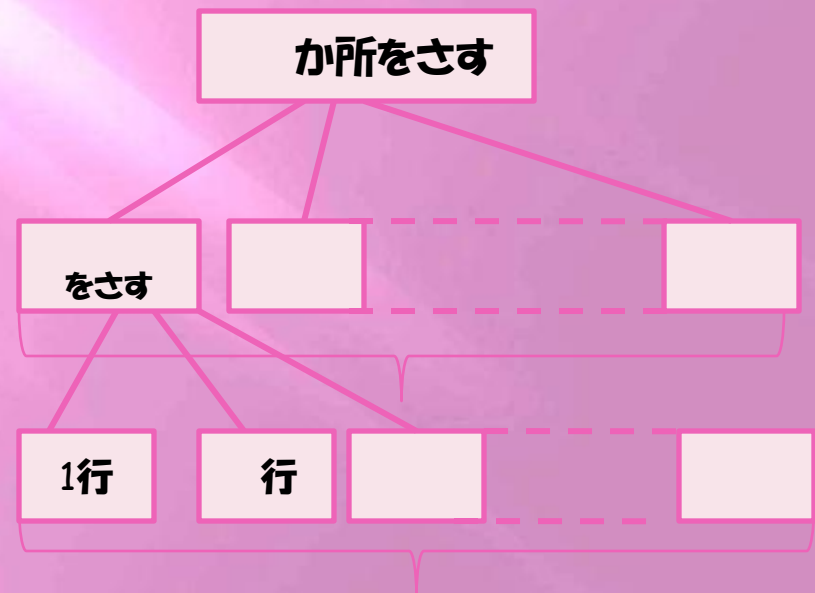


- 1ブロックあたりのデータ用使用可能領域のバイト数
1ブロックあたりの使用可能領域のバイト数 × (1-PCTFREE)

- 1ブロックあたりの格納行数
列長 =

- インデックスブロック数の算出(切り上げ)

- テーブルスペースの必要な容量(Kバイト単位)



今日の授業

- ➡ テーブル設計に基づく必要ディスクスペース量の計算
- ➡ データベース言語SQL
- ➡ データベース定義
 - ➡ データベースやテーブルの作成・削除
 - ➡ テーブルの作成・制約条件を付けて作成
 - ➡ テーブルの名前変更など



リレーショナル型データベースシステム (DBMS)(管理プログラム)

Step1

Relational database management system (RDBMS)

- ➡ **商用**RDBMS Oracle Database, IBM DB2,
- ➡ **オープンソース** MySQL、Postgre,
- ➡ Windows Office **だと** Access

その データベース操作、制御 を行う言語が、

SQL (Structured Query Language): 標準言語

- ➡ 演習ではオープンソースでシェアが最大といわれる
MySqlを使う。



SQLがRDBMSに対して持つ制御機能

➡ データベース定義

- ➡ データを格納すべき表の定義,ビューの定義
- ➡ 複数の表を関連づけるための規約や制約
- ➡ データベースのアクセス権などを定義

➡ データベース操作

- ➡ 表に対するデータの登録・修正・削除
- ➡ 複数の表の結合, ビュー表の作成などの集合操作
- ➡ 表中のデータ検索

➡ トランザクション管理

- ➡ 回復や同時実行のための最小単位として保証される一連の処理の操作



データベース定義

- ➡ (Data Definition Language)
- ➡ データベース定義用のSQL文
 - ➡ :テーブルの作成
 - ➡ :ビューの作成
- ➡ データベースアクセス権限の付与や取り消しのSQL文
 - ➡ 表やビューに対し、データの参照や変更に関する権限を付与・削除



ビューについて(簡単に)

- ▶ 特定の参照(SELECT文)に名前を付けてデータベースに保存しておく機能
 - ▶ 毎回複雑なSELECT文を指定することなく、欲しいデータの取得が可能に
- ▶ データに対するアクセス制限を設ける
 - ▶ 特定のユーザーに必要なデータのみを提供



データベース操作

- ▶ DML (Data Manipulation Language)
- ▶ **データベースアクセス用のSQL文**
- ▶ **参照・更新・追加・削除を行う**
- ▶ select
- ▶ update
- ▶ insert
- ▶ delete



トランザクション管理

- ▶ SQL文の **だけでは、データ**
ベースは更新されない
- ▶ **の後、** **を発行**
した時点で、データベースが更新される
- ▶ **を発行すると、データベース**
は更新されない



MySQLの起動と終了

起動 コマンドプロンプト上で

ドライブ名:¥インストールフォルダ

¥xampp¥mysql¥bin¥mysql -u root -p

Enter password:は無し

↵を入力

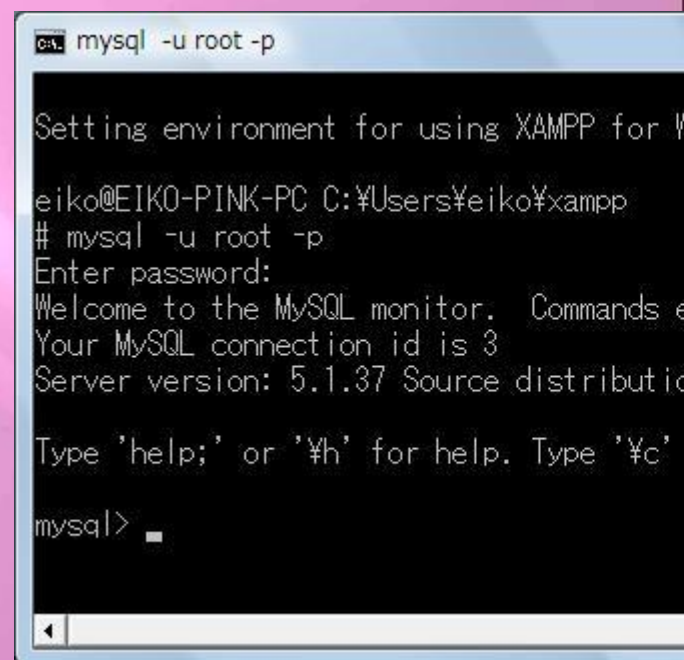
この画面MySQLモニタ

mysql> のプロンプトが出たら成功！

MySQLの終了は

exit ↵

(接続を切る)



```
cmd: mysql -u root -p

Setting environment for using XAMPP for W

eiko@EIKO-PINK-PC C:¥Users¥eiko¥xampp
# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands e
Your MySQL connection id is 3
Server version: 5.1.37 Source distributio
Type 'help;' or '¥h' for help. Type '¥c'
mysql> _
```



存在するデータベースを表示してみよう

mysqlを起動したものとして実際にアクセスしてみます

mysql>

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol |
| db1 |
| mysql |
| phpmyadmin |
| test |
+-----+
6 rows in set (0.00 sec)

mysql>
```

MySQL
RDBMS

データベース空間
information_sche
ma

データベース空間
cocol

データベース空間
db1

データベース空間
mysql

データベース空間
phpmyadmin

データベース空間
test

データベース空間db1にどんな表があるか見てみよう

➡mysql>

➡mysql> show tables;

```
mysql> use db1;  
Database changed  
mysql> show tables;
```

```
+-----+  
| Tables_in_db1 |  
+-----+  
| juchu  
| jyuchu  
| jyuchu2  
| web_tb  
+-----+
```

```
4 rows in set (0.00 sec)
```

MySql

RDBMS

データベース空間
information_sche
ma

データベース空間
cocol

データベース空間
db1

データベース空間
mysql

データベース空間
phpmyadmin

データベース空間
test

db1の中のjyuchu2という表に どんな要素があるか見てみよう

mysql>

```
mysql> desc jyuchu2;
```

Field	Type	Null	Key	Default	Extra
orderID	int(11)	NO	PRI	NULL	
shopID	char(5)	YES		NULL	
itemID	char(4)	YES		NULL	
orderNumber	int(11)	YES		NULL	

4 rows in set (0.00 sec)

```
mysql> _
```

MySQL

DBMS

データベース空間
information_sche
ma

データベース空間
cocol

データベース空間
db1

データベース空間
mysql

データベース空間
phpmyadmin

データベース空間
test



データベースへのアクセス制御

➡ ユーザの登録／権限の付与

➡ grant 権限名 on DB名. テーブル名 to ユーザ名 identified by 'パスワード';

user1にhanbaiデータベースのすべての表のすべての権限を与える

➡ これ以降, rootではなくuser1という一般ユーザでの制御が可能となる



データベースへのアクセス制御

▶ 主な権限の種類

- ▶ **all** :すべての権限を与える
- ▶ **deny all** :すべての権限を与えない
- ▶ **update, insert, delete** :テーブル操作の権限
- ▶ **create, alter, drop** :テーブルの作成, 変更, 削除
- ▶ **index** :インデックスの作成, 削除

user1に「学生」表のデータ参照と変更の権限を与える



データベースへのアクセス制御

➡ ユーザの権限の削除

➡ revoke 権限名 on DB名. テーブル名 from;

user1にhanbaiデータベースのcreate権限を削除

➡ ユーザ自体の削除

➡ DBを選択後削除すること



今日の授業

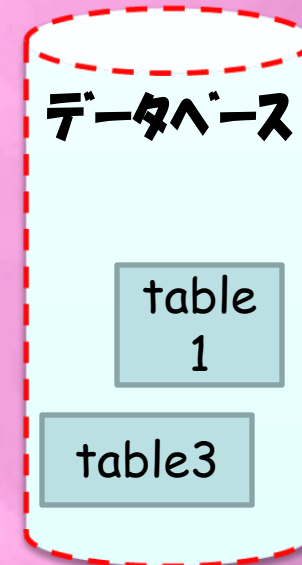
- ▶ テーブル設計に基づく必要ディスクスペース量の計算
- ▶ データベース言語SQL
- ▶ データベース定義
 - ▶ データベースやテーブルの作成・削除
 - ▶ テーブルの作成・制約条件を付けて作成
 - ▶ テーブルの名前変更など



データベースとテーブルの作成の流れ

一度にはできない

1. データベースの作成
2. テーブル(表)の定義
3. テーブルにデータを入力する



databaseの中に
table が入る



データベース作成 と テーブルの作成

①データベースの作成

②データベースの一覧表示

③データベースの指定

④テーブルの一覧表示

⑤テーブルの作成

create table "テーブル名" (フィールド名1 データ型, フィールド名2 データ型 ...);



データベースの削除と テーブルの削除

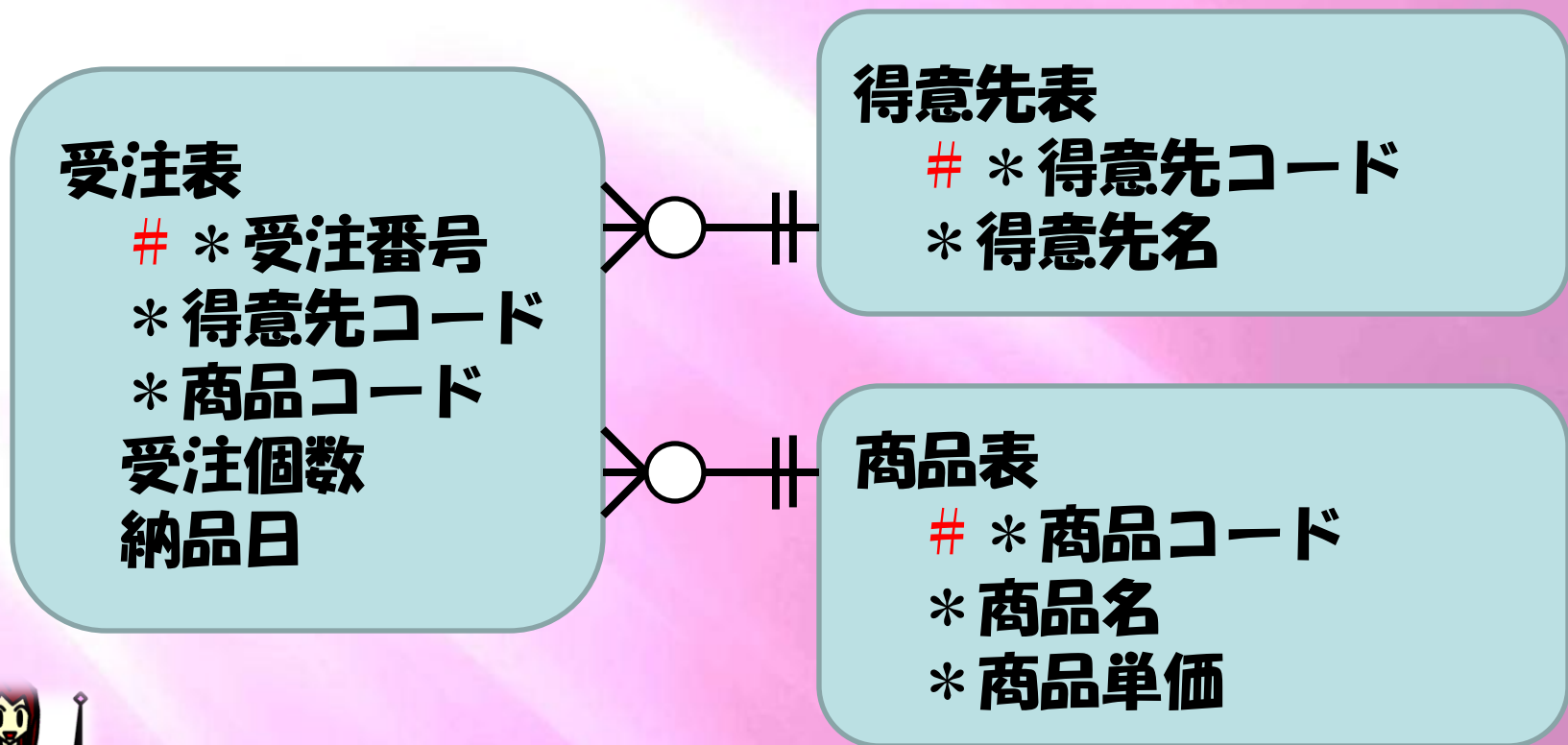
⑥データベースの削除

⑦テーブルの削除

消すのは
簡単!
やり直し
は不可能
!



shohindbというデータベースに 以下のER図で
表されるjuchuテーブルをつくる。



この受注表をデータベースに登録するときは

列名称(属性)	受注番号	得意先コード	商品コード	受注個数	納品日
データ型	INT	CHAR	INT	INT	DATE
最大データ長	4	5	3	5	7



shohindbというデータベースに 以下のER図で表されるjuchuテーブルをつくる。

- mysql> create database shohindb;
- mysql> show databases;
- mysql> use shohindb ;
- mysql>
- mysql>

```
mysql> create table  
-> juchu(  
-> orderID int(4),  
-> shopID char(5),  
-> itemID int(3),  
-> orderNum int(5),  
-> shipDate date  
-> );  
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> desc juchu;
```

Field	Type	Null	Key	Default	Extra
orderID	int(4)	YES		NULL	
shopID	char(5)	YES		NULL	
itemID	int(3)	YES		NULL	
orderNum	int(5)	YES		NULL	
shipDate	date	YES		NULL	

5 rows in set (0.03 sec)



今日の授業

- ➡ テーブル設計に基づく必要ディスクスペース量の計算
- ➡ データベース言語SQL
- ➡ データベース定義
- ➡ データベースやテーブルの作成・削除
- ➡ テーブルの作成・制約条件を付けて作成
- ➡ テーブルの名前変更など



このような表をデータベースに登録するときは

列名称(属性)	受注番号	得意先コード	商品コード	受注個数	納品日
データ型	INT	CHAR	INT	INT	DATE
最大データ長	4	5	3	5	7
キー種	PK	FK1	FK2		
一意性		①			①
依存先		得意先表	商品表		
入力必須	NN1	NN2	NN3		
平均データ長	4	5	3	10	7

ひとつひとつ見ていきましょう



受注表(juchu)

受注番号

得意先コード

商品コード

受注個数

納品日

得意先表(shopTable)

得意先コード

得意先名

商品表(itemTable)

商品コード

商品名

商品単価

受注表(juchu)

* 受注番号

* 得意先コード

* 商品コード

受注個数

納品日

得意先表(shopTable)

* 得意先コード

* 得意先名

商品表(itemTable)

* 商品コード

* 商品名

* 商品単価



列名称(属性)	受注番号	得意先コード	商品コード	受注個数	納品日
データ型	INT	CHAR	INT	INT	DATE
最大データ長	4	5	3	5	7
キー種	PK	FK1	FK2		

受注番号	得意先コード	商品コード	受注個数	納品日
------	--------	-------	------	-----

```
CREATE TABLE jyuchu
(
    orderID          int(4)  PRIMARY KEY ,
    shopID           char(5) ,
    itemID           int(3) ,
    orderNum         int(5),
    shipDate         date
);
```

作ったら
desc データテーブル名
で確認



販売店コード

商品コード

価格

PRIMARY KEY

```
CREATE TABLE priceTable
```

```
(
```

```
    dealerID      char(5),
```

```
    itemID        char(4),
```

```
    price          int(8),
```

```
    PRIMARY KEY(dealerID, itemID)
```

```
);
```

列名称(属性)	販売店コード	商品コード	価格
データ型	CHAR	CHAR	int
最大データ長	5	4	8
キー種	PK	PK	

複合キーのときは
この二つで一つの主キーに
なることを明示するために
このように書く

```
mysql> desc pricetable;
```

Field	Type	Null	Key	Default	Extra
dealerID	char(5)	NO	PRI		
itemID	char(4)	NO	PRI		
price	int(8)	YES		NULL	

```
3 rows in set (0.01 sec)
```

列名称(属性)	商品コード	商品名	商品単価
データ型	CHAR	CHAR	int
最大データ長	4	16	8
キー種	PK		
一意性		①	

UNIQUE KEY

CREATE TABLE itemTable

```
(
    itemID          char(4)  PRIMARY KEY ,
    item            char(16) UNIQUE,
    price           int(6)
);
```

```
mysql> desc itemtable2;
```

Field	Type	Null	Key	Default	Extra
itemID	char(4)	NO	PRI	NULL	
item	char(16)	YES	UNI	NULL	
price	int(6)	YES		NULL	

```
3 rows in set (0.00 sec)
```

```
mysql> desc priceTable;
```

```
3 rows in set (0.00 sec)
```



UNIQUE KEY

```
CREATE TABLE itemTable0
```

```
(
    itemID          char(4)  PRIMARY KEY ,
    item            char(16),
    size            int(4),
    UNIQUE(item, size)
);
```

各商品には同じサイズが
あってはならない



商品名とサイズの組合せは
一意でなければならない

データ型	CHAR	CHAR	int
最大データ長	4	16	4
キー種	PK		
一意性		①	①

```
mysql> desc itemTable3;
```

Field	Type	Null	Key	Default	Extra
itemID	char(4)	NO	PRI	NULL	
item	char(16)	YES	MUL	NULL	
size	int(4)	YES		NULL	

```
3 rows in set (0.01 sec)
```

```
mysql>
```

NOT NULL制約

```
CREATE TABLE itemTable
```

```
(
```

```
    itemID
```

```
        char(4) PRIMARY KEY ,
```

```
    item
```

```
        char(16) UNIQUE NOT NULL,
```

```
    price
```

```
        int(8)
```

```
);
```

```
mysql> desc itemtable;
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| itemID | char(4) | NO   | PRI | NULL    |       |
| item   | char(16) | NO   | UNI | NULL    |       |
| price  | int(8)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```



列名称(属性)	受注番号	得意先コード	商品コード	受注個数	納品日
データ型	INT	CHAR	INT	INT	DATE
最大データ長	4	5	3	5	7
キー種	PK	FK1	FK2		

受注番号	得意先コード	商品コード	受注個数	納品日
------	--------	-------	------	-----

```

CREATE TABLE jyuchu0
(
    orderID      int(4)  PRIMARY KEY ,
    shopID       char(5) REFERENCES shopTable(shopID),
    itemID       int(3) ,
    orderNum     int(5),
    shipDate     date
);

```

参照制約(Reference)



CHECK制約

受注番号

得意先コード

商品コード

受注個数

納品日

```
CREATE TABLE jyuchu1
(
    orderID          int(4)  PRIMARY KEY ,
    shopID           char(5) ,
    itemID           int(3) ,
    orderNum         int(5)  CHECK(orderNumber >=10),
    shipDate         date
);
```

**受注個数は 10個以上で
しか受け付けない**

**入力したデータが正しい範囲を逸脱して
いないか、正しいフォーマットになって
いるかなど、テーブルに入力するデータ
を必要に応じて制約を適用したい場合**



今日の授業

- ▶ テーブル設計に基づく必要ディスクスペース量の計算
- ▶ データベース言語SQL
- ▶ **データベース定義**
 - ▶ データベースやテーブルの作成・削除
 - ▶ テーブルの作成・制約条件を付けて作成
 - ▶ **テーブルの名前変更など**



テーブルの名前の変更など

➡ テーブルの名前の変更

➡ mysql> ALTER TABLE テーブル名 RENAME 新しいテーブル名;

➡ カラムの型を変える

➡ mysql> ALTER TABLE テーブル名 MODIFY カラム名 型 ~~~

➡ カラムの名前変更

➡ mysql> ALTER TABLE テーブル名 CHANGE 古いカラム名 新しいカラム名;

➡ カラムの名前と変更

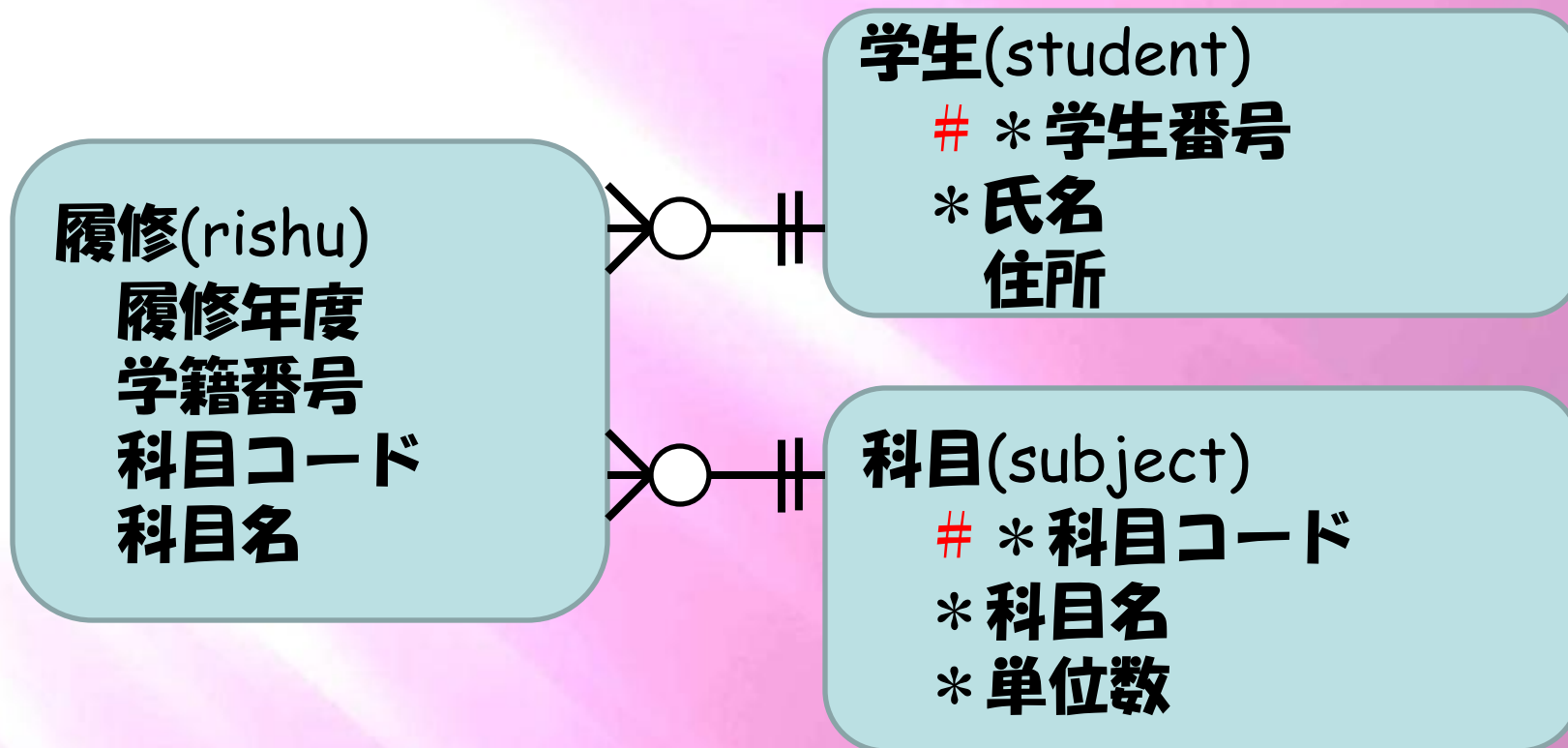
➡ mysql> ALTER TABLE テーブル名 CHANGE 古いカラム名 新しいカラム名 型;

➡ カラムを削除する

➡ mysql> ALTER TABLE テーブル名 DROP 削除するカラム名;



実習1:このようなER図を データベースに登録するときは



実習1



履修(rishu)

列名称(属性)	履修年度	学生番号	科目コード
データ型	INT	CHAR	CHAR
最大データ長	4	10	10
キー種		FK1	FK2
依存先		学生	科目

実習1

履修(rishu)

列名称(属性)	履修年度	学生番号	科目コード
データ型	char	CHAR	CHAR
最大データ長	4	10	10
キー種		FK1	FK2
依存先		学生	科目



実習1

履修(rishu)
履修年度
学生番号
科目コード

学生(student)
* 学生番号
* 氏名
* 住所

科目(subject)
* 科目コード
* 科目名
* 単位数

学生(student)

	学生番号	氏名	住所
データ型	CHAR	CHAR	CHAR
最大データ長	10	20	40
キー種	PK		
依存先			
入力必須	NN1	NN2	

科目(subject)

科目コード	科目名	単位数
CHAR	CHAR	INT
10	10	3
PK		
NN1	NN2	NN3

学生(student)

	学生番号	氏名	住所
データ型	CHAR	CHAR	CHAR
最大データ長	10	20	40
キー種	PK		
依存先			
入力必須	NN1	NN2	

科目(subject)

科目コード	科目名	単位数
CHAR	CHAR	INT
10	10	3
PK		
NN1	NN2	NN3