Database

第8回：SQL言語（データベース操作）

上智大学理工学部情報理工学科
高岡詠子

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

- データベース定義
- データベース操作
  - データベースの参照
  - データベースの登録・変更・削除

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

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(juchuTable)
＊**受注番号**(orderID)
＊**得意先コード**(shopID)
＊**商品コード**(itemID)
受注個数(orderNum)
納品日(shipDate)

得意先表(shopTable)
＊**得意先コード**(shopID)
＊**得意先名**(shopName)

商品表(itemTable)
＊**商品コード**(itemID)
＊**商品名**(itemName)
＊**商品単価**　(price)

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	2	7

```
CREATE   TABLE     juchuTable
(
   orderID
   shopID
    itemID
    orderNum
    shipDate
    UNIQUE(                          )
```

```
CREATE   TABLE    shopTable
 (
    shopID
    shopName
 );

CREATE   TABLE    itemTable
 (
   itemID
   itemName
   price
);
```

得意先表(shopTable)

列名称(属性)	得意先コード	得意先名
データ型	CHAR	CHAR
最大データ長	5	10
キー種	PK	
入力必須	NN1	NN2

商品表 (itemTable)

商品コード	商品名	商品単価
INT	CHAR	INT
3	20	8
PK		
NN1	NN2	NN3

今日の授業

- データベース定義
- データベース操作
  - データベースの参照
  - データベースの登録・変更・削除

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
  - 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
- 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

テーブルの中身を確認：select

- _____ * from juchuTable;

そのテーブルに登録されているすべての
情報を見ることができる便利なコマンド

- _____ juchuTable;
  - そのテーブルの属性を知るためのコマンド

データの登録：insert

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
	101	a102	35	3	091111

- 1行追加：

_____ _____ juchuTable _____(101,"a11",35,3,091111);

_____ _____ テーブル名 _____(カラムの内容)：

```
mysql> insert   into  jyuchu  values(101,"a11",35,3,091111);
Query OK, 1 row affected (0.00 sec)

mysql> select * from jyuchu;
+---------+--------+--------+----------+------------+
| orderID | shopID | itemID | orderNum | shipDate   |
+---------+--------+--------+----------+------------+
|     101 | a11    |     35 |        3 | 2009-11-11 |
+---------+--------+--------+----------+------------+
1 row in set (0.00 sec)

mysql> insert into jyuchu values(102,"a11",3,5,091111);
ERROR 1062 (23000): Duplicate entry 'a11-2009-11-11' for key 'shopID'
mysql>
```

このエラーはなぜ出るのでしょうか？

テーブルの名前の変更など

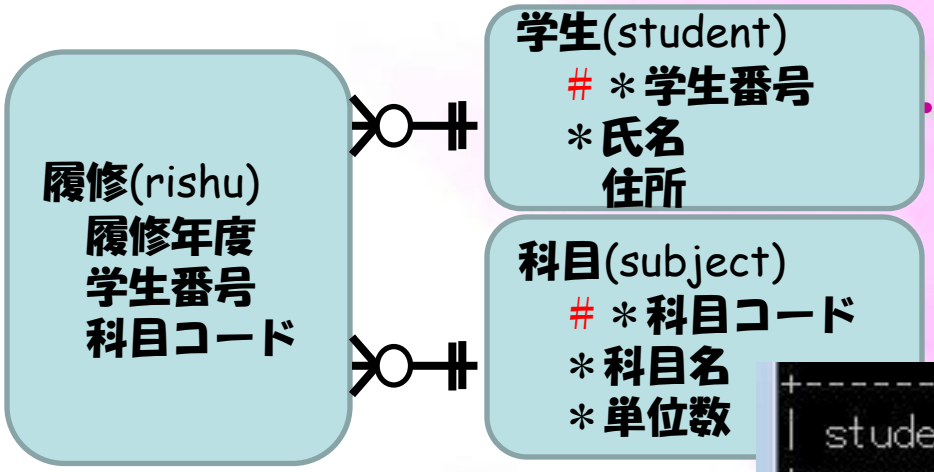
- **テーブルの名前の変更**
- mysql> _____ _____ **テーブル名** _____ **新しいテーブル名**;
- **カラムの型を変える**
- mysql> _____ _____ **テーブル名** _____ **カラム名 型** ～～～
- mysql> ALTER TABLE jyuchu _____ orderID char(3);
- **カラムの名前変更**
- mysql> ALTER TABLE **テーブル名** CHANGE **古いカラム名 新しいカラム名 型**;
- mysql> ALTER TABLE jyuchu _____ orderID oID char(3);
- **カラムを削除する**
- mysql> ALTER TABLE **テーブル名** _____ **削除するカラム名**;

データの登録

データの登録

insert into テーブル名 (カラムの内容);

```
insert into juchuTable values(102,"a10",5,3,091112);
insert into juchuTable values(103,"a10",50,1,091113);
insert into juchuTable values(104,"a11",543,2,091112);
insert into juchuTable values(105,"a11",115,7,091113);
insert into juchuTable values(106,"a12",45,10,091112);
insert into juchuTable values(107,"a13",34,2,091112);
insert into juchuTable values(108,"a13",60,1,091113);
```




```
+-----------+----------+-----------------------+
| studentID | name     | address               |
+-----------+----------+-----------------------+
| a0812343  | 上智太郎 | 千代田区紀尾井町７－１ |
| a0812344  | 桜花子   | 新宿区麹町34          |
+-----------+----------+-----------------------+
2 rows in set (0.00 sec)

mysql> select * from subject;
+-------------+-----------+-----------+
| rishunendo  | studentID | subjectID |
+-------------+-----------+-----------+
| 09          | a0812343  | ict90274  |
| 09          | a0812344  | ict90274  |
| 09          | a0812343  | ict90250  |
| 09          | a0812343  | ict90987  |
+-------------+-----------+-----------+
4 rows in set (0.00 sec)
```

右のような表を
つくりましょう

実習1

```
CREATE　TABLE　rishu
 (




) ;
```

データ登録

```
select　*　from　rishu;
```

中身確認

```
CREATE　TABLE　student
 (




 ) ;
CREATE　TABLE　subject
 (




 ) ;
```

```
insert　into　student　values("a0812343", "上智太郎", "千代田区紀尾井町7－1 ");
```

subjectID	subjectName	credit
lct90274	データベース	4
lct90250	情報リテラシー	2
lct90009	科学技術英語	4
lct90113	人間学	4
lct90320	社会と情報	4
lct90100	体育	4
lct90110	英語	2
lct90987	コンピュータプログラミング	2

```
insert  into  subject  values("lct90274", "データベース",4);
```

データの更新

- _____ 表名
  _____ カラム名＝値，カラム名＝値
  _____ 条件;
- update subject set credit=credit+2;
- update subject set credit=credit-2 where credit=2;

テーブルやデータの削除

- データのみの削除
  - ＿＿＿＿＿＿　＿＿＿＿＿＿　テーブル名;
  - delete from テーブル名
    where　条件;
  - delete from subject
    where subjectID="lct90987";
- データだけでなくテーブルごと削除する
  - ＿＿＿＿＿　＿＿＿＿＿　テーブル名;

今日の授業

- データベース定義
- データベース操作
  - データベースの参照
  - データベースの登録・変更・削除

最も多く使われる

- select文
- _____ カラム名1, カラム名2, ・・・・
- _____ 抽出条件
- _____ グループ化を行う
- _____ グループ化を行ったときの抽出条件
- _____ 並べ替えを指定する

select文

- すべての列を抽出する(　　)
  - subject表からすべての列を表示

- 特定列の抽出(　　)
  - select　列名(,で区切る) from テーブル名;
  - subject表から2つの列を選択して表示
  - _____ subjectID, subjectName _____ _____;
- 算術表示もできる

select credit * 4 from subject;

select文

- 条件付き参照
  select　カラム名(,で区切る)　from　テーブル名
  　where　条件
  - juchuTable表からshopIDがa10である行を抽出する
  - subject表からcreditが2を超える科目名subjectNameを抽出する
  - juchuTable表からshopIDがa10である行の商品コードと受注個数を抽出する

実習：以下のような出力をする表をつくる

```
mysql> desc shopSale;
+------------+---------+------+-----+---------+-------+
| Field      | Type    | Null | Key | Default | Extra |
+------------+---------+------+-----+---------+-------+
| shopname   | char(20) | NO  |     | NULL    |       |
| sales      | int(10)  | YES |     | NULL    |       |
| date       | date     | NO  |     | NULL    |       |
+------------+---------+------+-----+---------+-------+
mysql> desc netSale;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| sales | int(10) | YES  | NO  | NULL    |       |
| date  | date    | YES  | NO  | NULL    |       |
+-------+---------+------+-----+---------+-------+
```

入力された表

shopSale

shopname	sales	date
紀尾井町	450,000	11/1
高輪	320,000	11/3
赤坂	876,600	11/5
品川	438,000	11/3
紀尾井町	200,000	11/10
赤坂	120,000	11/13
赤坂	40,000	11/20
高輪	450,000	11/3
品川	220,000	11/5
高輪	110,000	11/18
品川	220,000	11/15

netSale

sales	date
120,000	11/1
150,000	11/7
250,550	11/13
320,000	11/20

```
insert  into  shopSale values("紀尾井町", 450000, 20091101);
```

- create table shopSale(

  );
- create table netSale(

  );

select文

- 重複行を除外する
- shopSale表からshopNameに関して重複行を除外して表示させる

select文

- テーブルをソートして表示
  - shopSale表を日付順に並べ替えする

select * from テーブル名 _____ カラム名(,で区切る);

- カラム名を変えて表示

select shopName as “店舗名” from shopSale;

関係演算子と論理演算子

- select * from rishu where rishunendo>=2009 and subjectID='lct90274';

関係演算子

=	左辺が右辺と等しい
<	左辺が右辺より小さい
<=	左辺が右辺以下
>	左辺が右辺より大きい
>=	左辺が右辺以上
<>	左辺と右辺が等しくない

論理演算子

AND	かつ
OR	または
NOT	否定

集合関数一覧

- SUM( )
  指定条件によって得られた列の値の合計を求める関数
- AVG( )
  指定条件によって得られた列の値の平均値を求める関数
- MAX( )
  指定条件によって得られた列の値の中で最大値を返す関数
- MIN( )
  指定条件によって得られた列の値の中で最小値を返す関数
- COUNT( )
  指定条件によって得られた表の基数、すなわち行数を求める関数

集合関数を使ったselect文

- グループ化を行う
- shopSale表においてshopnameごとの売り上げを表示させたいとき

```
select  shopname, sum(sales) from shopSale
group by shopname;
```

- 関数の値に条件をつける(whereは使えないから)

```
 select  shopname, sum(sales) from shopSale
group by shopname
having sum(sales) > 1000000;
```

集合関数を使ったselect文

- 重複行を除外する
- shopSale表からshopNameに関して重複行を除外して表示させる
- 数える

```
select  count(shopName) from
shopSale;
```

- 異なる列の値を数える

```
 select  count(distinct shopName) from
shopSale;
```