Database

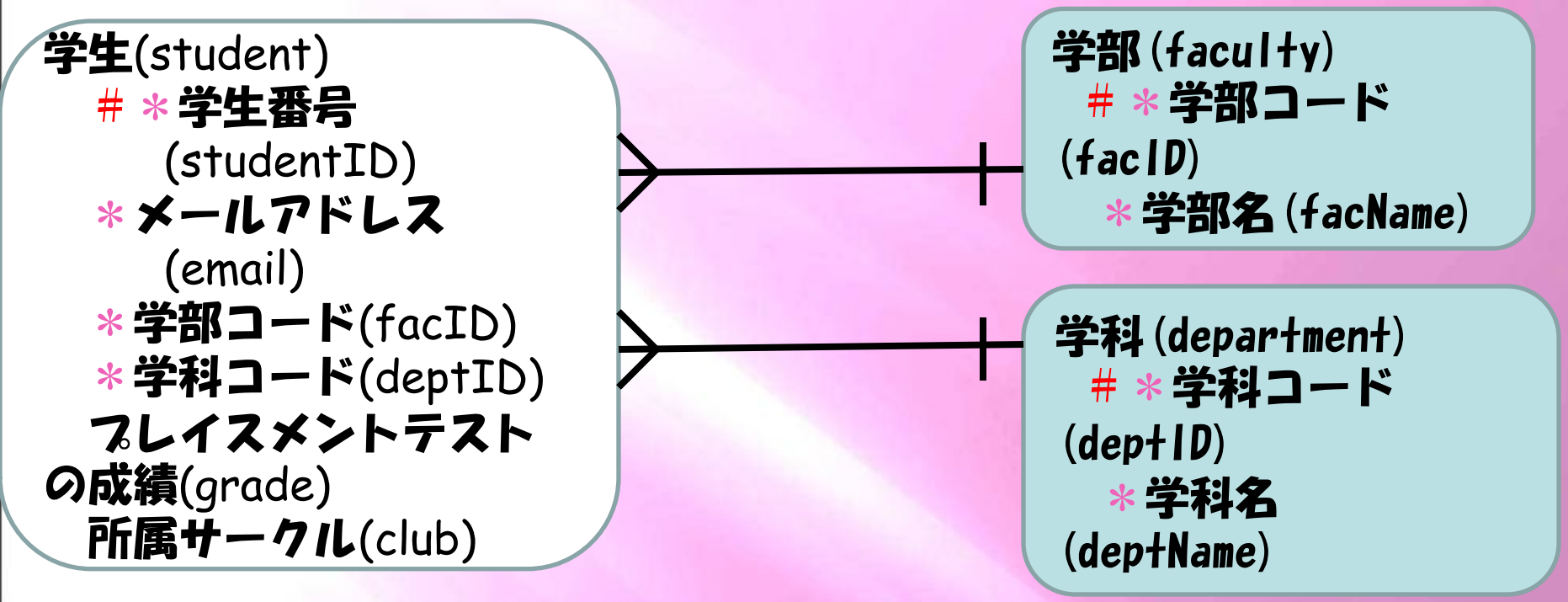
第9回：SQL言語（データベース操作：集合関数・抽出条件・副問い合わせ）

上智大学理工学部情報理工学科
高岡詠子

今日の授業

- データベース操作のおさらい
  - データベースの参照
  - データベースの登録・変更・削除
- 集合関数・抽出条件
- 副問い合わせ

①次の3つのテーブルインスタンスチャートをつくる



②Univという名前のデータベース空間をつくる

③Univという名前のデータベース空間にアクセスする

④次の3つのテーブルをつくる

NOT NULL制約　PRIMARY制約　UNIQUE制約　REFERENCE制約

列名称（属性）	studentID	email	facID	deptID	grade	club
データ型	INT	VARCHAR	INT	CHAR	INT	VARCHAR
最大データ型	12	60	7	7	4	50
キー種						
一意性						
入力必須						

学部(faculty)
＊学部コード(facID)
＊学部名(facName)

学科(department)
＊学科コード(deptID)
＊学科名　(deptName)

列名称	facID	facName
データ型	int	char
最大データ型	7	30
キー種		
一意性		
入力必須		

列名称	deptID	deptName
データ型	int	char
最大データ型	7	30
キー種		
一意性		
入力必須		

faculty

facID	学部名
1	理工学部
2	文学部
3	神学部
4	外国語学部
5	国際教養学部

CHECK制約：
10の位は学部コードと同じにしたい

department

deptID	学科名
11	機能創造理工
12	物質生命理工
13	情報理工
21	哲学科
22	史学科

```
CREATE   TABLE     student
 (
    studentID   int(12)   PRIMARY  KEY ,
    email          varchar(60)   UNIQUE   NOT NULL,
    facID          INT(7)    NOT NULL,
    deptID         INT(7)    NOT NULL,
    grade          INT(4),
    club           varchar(50),
   CHECK( facID=deptID   div  10)
    ) ;
```

CHECK制約：
10の位は学部コードと同じにしたい

列名称（属性）	student ID	email	facl D	deptID	grade	club
データ型	INT	VARCHAR	INT	CHAR	INT	VARCHAR
最大データ型	12	60	7	7	4	50
キー種	PK		FK1	FK2		
一意性	①	②				
入力必須	NN1	NN2	NN 3	NN4		

```
CREATE    TABLE      faculty
 (
    facID          int(7)   PRIMARY KEY,
    facName      char(30)     NOT NULL
     ) ;
```

列名称	facID	facName
データ型	int	char
最大データ型	7	30
キー種	PK	
一意性	①	
入力必須	NN1	NN2

```
CREATE    TABLE      department
 (
    deptID          int(7)   PRIMARY KEY,
    deptName      char(30)   NOT NULL
     ) ;
```

列名称	deptID	deptName
データ型	int	char
最大データ型	7	30
キー種	PK	
一意性	①	
入力必須	NN1	NN2

データの登録：insert

- 1行追加： insert into テーブル名 values(カラムの内容)；
- insert into faculty values(1,"理工学部");
- insert into faculty values(2,"文学部");
- insert into faculty values(3,"神学部");
- insert into faculty values(4,"外国語学部");
- insert into faculty values(5,"国際教養学部");
- …… 8学部分のデータベースを自分でつくりましょう．順序は問わない

faculty

学部コード	学部名
1	理工学部
2	文学部
3	神学部
4	外国語学部
5	国際教養学部

データの登録：insert

8学部の各学部ごとに
いくつの学科があるか調べて
自分データベースをつくろう

10の位は学部コードと同じ

department

学科コード	学科名
11	機能創造理工
12	物質生命理工
13	情報理工
21	哲学科
22	史学科

- **1行追加：** insert into **テーブル名** values(**カラムの内容**)；
- insert into department values(11,"**機能創造理工**");
- insert into department values(12,"**物質生命理工**");
- insert into department values(13,"**情報理工**");
- insert into department values(21,"**哲学科**");
- insert into department values(22,"**史学科**");

データの登録：insert

student

学生番号	メールアドレス	学部コード	学科コード	プレイスメントテストの成績	所属サークル
12345	aaa@sophia.ac.jp	1	11	89	カト学生
12346	abc@sophia.ac.jp	1	13	NULL	ETL
12347	bbb@sophia.ac.jp	2	21	67	BLT
12348	ccc@sophia.ac.jp	3	31	30	AUX
12349	aaa@sophia.ac.jp	4	42	42	

- **1行追加：** insert into **テーブル名** values(**カラムの内容**)；
- insert into student values(12345,"aaa@sophia.ac.jp",1,11,89,"**カト学**");
- insert into student values(12346,"abc@sophia.ac.jp",1,13,NULL,"ETL");
- insert into student values(12347,"bbb@sophia.ac.jp",2,21,67,"BLT");
- insert into student values(12348,"ccc@sophia.ac.jp",3,31,30,"AUX");
- insert into student values(12349,"ddd@sophia.ac.jp",4,42,90,");

テーブルやデータの削除

- データのみの削除
  - _____ from テーブル名;
  - _____ from テーブル名
    _____ 条件;
  - _____ from subject
    _____ studentID="12352";
- データだけでなくテーブルごと削除する
  - _____ table テーブル名;

データの更新

- _____ 表名
  _____ カラム名＝値, カラム名＝値
  _____ 条件;
- _____ student
  _____ facID=5, deptID=51;
  _____ studentID=12352;

今日の授業

- データベース操作のおさらい
  - データベースの参照
  - データベースの登録・変更・削除
- **集合関数・抽出条件**
- 副問い合わせ

先週の例

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

最も多く使われるSQL問い合わせ

- 基本構文
- select　カラム名1，カラム名2，カラム名3・・・
  from　表名1，表名2，・・・・
  where　抽出条件
  group by　グループ化を行う
  having　グループ化を行ったとの抽出条件
  order by　並べ替えを指定する
- 導出された表：導出表
- データベースに格納されている表：実表

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
  _____ 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
  _____ sum(sales) > 1000000;

抽出条件で指定する演算子

演算子	説明
比較演算子	<|<=|=|>=|>|<>
BETWEEN	～以上～以下
IN	いずれかの値と等しい
IS NULL	NULL判定
LIKE	部分一致検索

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

- **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(_____ shopName) from shopSale;
- **平均値を出す**
  select　_____(grade)　from　student;

抽出条件で指定する演算子

演算子	説明
比較演算子	<|<=|=|>=|>|<>
BETWEEN	～以上～以下
IN	いずれかの値と等しい
IS NULL	NULL判定
LIKE	部分一致検索

BETWEEN演算子

- **期間指定**

  select * from shopSale where date
  _____ 20091101 and 20091109;

- **期間指定とソート**

  select * from shopSale where date
  _____ 20091101 and 20091109
  _____ date;

2009年11月1日から11月9日の間のshopSaleの情報を日付順に表示する

抽出条件で指定する演算子

演算子	説明
比較演算子	<|<=|=|>=|>|<>
BETWEEN	～以上～以下
IN	いずれかの値と等しい
IS NULL	NULL判定
LIKE	部分一致検索

IN演算子

- **普通の問い合わせ**

  select *
  from shopSale
  _____ shopname =“紀尾井町”;

- IN演算

  select *
  from shopSale
  where shopname _____ (“紀尾井町”,“赤坂”);

  紀尾井町または赤坂の情報を含んだ行が出力される

- select *
  from shopSale
  where shopname _____ _____(“紀尾井町”,“赤坂”);

  紀尾井町または赤坂以外の情報を含んだ行が出力される

抽出条件で指定する演算子

比較演算子	<|<=|=|>=|>|<>
BETWEEN	～以上～以下
IN	いずれかの値と等しい
IS NULL	NULL判定
LIKE	部分一致検索

IS NULL演算子

- select * from student
  where club _____ _____;

所属サークルの名前がわからない人の情報を表示したい

NULLと空白の違い

どのサークルにも所属していない場合は空白
所属しているサークル名が不明の場合はNULL

NULLの扱い

値がない場合の入力値として使う.
不明(記録時にわからない), 未定(これから決まる), 無意味など
値がない＝他と比較できない, 平均値を求める演算の対象から
はずさなくてはならないなど

抽出条件で指定する演算子

演算子	説明
比較演算子	<|<=|=|>=|>|<>
BETWEEN	～以上～以下
IN	いずれかの値と等しい
IS NULL	NULL判定
LIKE	部分一致検索

パターンマッチ

select * from shopSale where date like _____;

11月の10日から19日までの売上情報を表示したい

'A_Z'　Aから始まり一文字をいれてZで終わる
ABZ, A2Zなど．ADDZはNG

'ABC%'　ABCから始まる文字列
ABC, ABCD, ABCCCCなど．ABBCはNG

'%AN%'　ANを含む文字列
LOS ANGELS, SAN FRANCISCOなど．

select * from shopSale where date ____ like _____;

11月の10日から19日までを除く期間売上情報を表示したい

今日の授業

- データベース操作のおさらい
  - データベースの参照
  - データベースの登録・変更・削除
- 集合関数・抽出条件
- 副問い合わせ

副問い合わせに指定する演算子

- 副問い合わせ：探索条件の中に指定する問い合わせ（問い合わせの入れ子）

演算子	説明
比較演算子	<|<=|=|>=|>|<>
IN	いずれかの値と等しい
EXISTS	出力があるかどうかの判定

右側の式に副問い合わせを指定

副問い合わせのみで使用

比較演算子

- 副問い合わせの結果は1列かつ1行(一つの値)でなければならない
- 比較するため
- select * from student
  where grade >=
  (select avg(grade) from student)

プレイスメントテストが平均点以上の学生情報

IN演算子

- 副問い合わせの結果は1列の場合は複数行でもよい.
- 比較するため
- select deptName from department where deptID in((select deptID from student where grade>=70)

プレイスメントテストが70点以上の学生が所属する学科名を知る

問題

プレイスメントテストが平均点以上の学生が所属する学科名を知る

```
select deptName from department
where deptID in
 (select deptID from student where
grade>=(select avg(grade) from
student));
```

EXISTS演算子

- select date from netsale where _____ (select * from shopsale where shopsale.date = netsale.date);

- select date from netsale where date ____ (select date from shopsale);

店舗でもネットでも売り上げがあった日を知りたい