数据库系统原理

第三章　PG：数据定义与操作

第三章 PG：数据定义与操作

3.1 SQL概述

3.2 数据定义

3.3 单表查询

3.4 联接查询

3.5 嵌套查询

- SQL（Structured Query Language）全称为“结构化查询语言”

◆ PostgreSQL(PG)

✓ 开源数据库管理系统软件

✓ 当前SQL标准的一些重要特征是由PG及其祖先首先开创的

网上考试数据库系统里面有大量各个专业、各门课程、各类考试（包括考试、测验和练习）可用的题目，各个院系的考官每次可以从中选择合适的题目组成试卷称组卷，考生可以根据需要报考某门试卷，答卷后获得一个分数。

考官表 examiner

erid	ername	ersex	erage	ersalary	erdepa
2009040	成志云	女	35	5000	历史学院
1990122	戴小刚	男	53	8000	教育学部
1998039	丁向军	女	42	6500	文学院
2011049	郑博宇	男	32	5000	物理系
2007033	李晓燕	女	38	5000	心理学院
1995057	林永强	男	49	6500	历史学院
2010022	姚翠红	女	36	5000	物理系
2013069	王瑞芬	女	30	5000	心理学院

考生表 examinee

eeid	eename	eesex	eeage	eedepa
218811011013	刘诗诗	男	20	历史学院
218811011014	刘诗诗	男	21	历史学院
218811011219	王琳懿	女	18	文学院
218811011220	王琳懿	女	19	文学院
218811011221	刘慧杰	女	19	文学院
218811011117	刘慧杰	女	19	教育学部
218811011025	张立帆	男	20	心理学院
218811011027	张立帆	男	19	心理学院
218811011028	刘慧杰	男	20	心理学院

试卷表 exampaper

eid	ename	etype	eduration
0205000002	中国近现代史纲要	4	100
0210000001	大学外语	2	180
0201020001	计算机应用基础	4	120
0211000001	大学美育	3	120
0219001014	普通物理学	4	100
0110001001	教育学	1	180
0110001002	心理学	1	180

院系表 department

dname	dloca	dtele
历史学院	主楼B2	58809289
教育学部	英东教育楼	58808855
文学院	主楼B7	58807998
物理系	物理楼	58808135
心理学院	后主楼12	58807832
减灾学院	京师科技大厦	58805461

考生答卷表 eeexam

eeid	**eid**	**achieve**
218811011013	0205000002	92
218811011013	0210000001	85
218811011013	0201020001	88
218811011117	0210000001	90
218811011117	0201020001	80

考官组卷表 erexam

erid	**eid**
2009040	0205000002
1998039	0211000001
2007033	0110001002
2010022	0219001014
1990122	0110001001

基本Select语句的一般形式

```
SELECT [ALL|DISTINCT]
        <目标列表达式> [别名] [ ，<目标列表达式> [别名]] …
[FROM    <表名> [别名] [ <连接表达式>] …
[WHERE <条件表达式>]
[GROUP BY <列名>[<列名>]
[HAVING    <条件表达式>]]
[ORDER BY <列名> [ASC|DESC] [<列名> [ASC|DESC] ]
[LIMIT<行数> [<OFFSET 偏移量>]]];
```

- **整个语句的执行过程：**

（1）如果仅有SELECT子句，按SELECT子句中给出的列名或列表达式求值，否则继续执行（2）；

（2）FROM：从FROM子句获得表；

（3）选取满足WHERE子句所给出条件表达式的行；

（4）按GROUP BY子句中指定列的值分组；

（5）提取满足HAVING子句中组条件表达式的那些组；

（6）按SELECT子句中给出的列名或列表达式求值；

（7）ORDER BY子句对输出的目标表进行排序，ASC：升序，DESC：降序；

（8）按照LIMIT子句的偏移量和行数确定输出元组；

（9）输出。

- 这个执行步骤只是用来帮助我们理解一个查询语句的查询结果是什么样的
- 绝不意味着实际的执行顺序
- 在实际执行时数据库管理系统会为了尽可能快地获得等价查询结果而采取完全不同的执行步骤！

- SQL语言是大小写不敏感的，PG建议SQL的保留字用大写；表名、字段名等所有数据库对象名全部使用小写（除非从定义到使用始终都加双引号，否则所有对象名自动转化为小写）；使用单引号做字符串常量的标识；只有引号里面的字符才区分大小写。

第三章 PG：数据定义与操作

3.1 SQL概述

3.2 数据定义

3.3 单表查询

3.4 联接查询

3.5 嵌套查询

表的定义和数据修改

表模式的创建、修改与删除

添加、修改和删除数据

表的创建

```
CREATE TABLE <表名>
        （<列名> <数据类型> ，
          <列名> <数据类型> ，
         …
    ）；
```

名字	存储空间	描述
SMALLINT	2 字节	小整数
INT（INTEGER）	4 字节	普通整数
BIGINT	8 字节	大整数
NUMERIC(p,d)	变长	用户声明精度，精确
REAL	4 字节	变精度，不精确,6 位十进制数字精度
DOUBLE PRECISION	8 字节	变精度，不精确, 15 位十进制数字精度
VARCHAR(n)	变长	有长度限制
CHAR(n)	定长	不足补空白
DATE	4 字节	日期 YYYY-MM-DD
TIME	8 字节	日内时间 HH:MM:SS
BOOLEAN	1 字节	两个值 TRUE 和 FALSE

```
CREATE TABLE examiner
    (  erid    CHAR(20)，
       ername  CHAR(20)，
       ersex     CHAR(2)，
       erage    SMALLINT，
       ersalary INT，
       erdepa  CHAR(20)
      );
```

```
ALTER TABLE examiner RENAME TO erexamine;

DROP TABLE examiner;
```

```
ALTER TABLE <表名>
        [ ADD COLUMN  <新列名> <数据类型> ]
        [ RENAME COLUMN<旧列名> TO <新列名> ]
        [ ALTER COLUMN<列名> TYPE <数据类型> ]
        [DROP COLUMN <列名> ];
```

```
ALTER TABLE examinee ADD COLUMN er_entrance DATE；

ALTER TABLE examinee ALTER COLUMN erage TYPE  INT；

ALTER TABLE examiner RENAME COLUMN ersalary TO erwage;
```

- 插入元组
    - INSERT INTO <表名> [(<属性1>[，<属性2 >…)]
    - VALUES (<常量1> [，<常量2>]　…　)

- 插入查询结果
    - INSERT　INTO<表名>[(<列名序列>)]
    - <SELECT查询语句>

- 插入表
    - INSERT　INTO<表名1>[(<列名序列>)]
    - TABLE<表名2>

INSERT INTO examinee VALUES（‘201615126’,‘张强’,’男’,20,‘历史学院’）

INSERT INTO examinee (eename,eeid,eedepa) VALUES (‘张强’,‘201211015126’,‘历史学院’）

DELETE FROM <表名> [WHERE<条件表达式>]

```
DELETE FROM examiner
        WHERE erage >
            (SELECT AVG(erage)
             FROM examiner)

DELETE FROM examiner
```

UPDATE <表名>

SET <列名>=<表达式>[，<列名>=<表达式>]…

[WHERE <条件>];

```
UPDATE examiner SET erage=erage+1

WHERE erage > (SELECT AVG(erage) FROM examiner);
```

第三章 PG：数据定义与操作

3.1 SQL概述

3.2 数据定义

3.3 单表查询

3.4 联接查询

3.5 嵌套查询

SELECT [ALL|DISTINCT] <目标列表达式> [,<目标列表达式>] …

FROM <表名>

```
SELECT eid，ename

FROM exampaper;
```

```
SELECT  *
FROM exampaper;
```

```
SELECT eeeid,100-achieve
FROM eeexam;
```

```
SELECT *

FROM examiner

ORDER BY erdepa ASC，erage DESC；
```

SELECT [ALL|DISTINCT] <目标列表达式> [,<目标列表达式>] …

FROM <表名>

DISTINCT：去重

默认或ALL：不去重

```
SELECT *

FROM examinee

WHERE eeage>20;
```

```
SELECT  *

FROM examiner

WHERE  erdepa  IN (‘历史学院’，‘心理学院' ) AND erage>58 AND ersalary BETWEEN 6000
AND 9000 AND ername LIKE '罗%';
```

```
SELECT COUNT(*)

FROM  examiner;


SELECT AVG(achieve)

FROM eeexam

WHERE eeid= '218811011013';
```

```
SELECT erdepa, AVG(erage) AS avg_age

FROM examiner

GROUP BY erdepa

ORDER BY avg_age;
```

```
SELECT eeid, COUNT(*)

FROM  eeexam

GROUP BY eeid

HAVING  COUNT(*) >3

ORDER BY COUNT(*) DESC;
```

第三章 PG：数据定义与操作

3.1 SQL概述

3.2 数据定义

3.3 单表查询

3.4 联接查询

3.5 嵌套查询

- 联接
  - 条件
  - 类型

```
SELECT *
FROM  examinee, eeexam          /* examinee CROSS JOIN eeexam */
WHERE  examinee.eeid = eeexam.eeid;
```

```
SELECT *

FROM  examinee NATURAL JOIN eeexam;
```

```
SELECT *

FROM examinee JOIN eeexam using(eeid);
```

```
SELECT *

FROM examinee JOIN eeexam ON examinee.eeid = eeexam.eeid;
```

```
SELECT *

FROM erexam NATURAL LEFT OUTER JOIN exampaper;
```

或：

```
SELECT *

FROM erexam NATURAL LEFT JOIN exampaper;
```

第三章 PG：数据定义与操作

3.1 SQL概述

3.2 数据定义

3.3 单表查询

3.4 联接查询

3.5 嵌套查询

一个SELECT-FROM-WHERE语句称为一个查询块，将一个查询块嵌套在另一个查询块的SELECT、FROM、WHERE、GROUP BY、HAVING、ORDER BY、LIMIT、OFFSET、WITH子句中的查询称为嵌套查询。

查询块可以出现在表名出现的地方。

```
WITH avgach (eeid,avgachieve) AS
                    (  SELECT eeid,AVG(achieve)
                            FROM eeexam
                            GROUP BY eeid
                            )
SELECT COUNT(*)
FROM avgach
WHERE avgachieve>=80;
```

```
SELECT COUNT(*)
        FROM (SELECT eeid,avg(achieve)
                    FROM eeexam
                    GROUP BY eeid
                    )AS avgach (eeid,avgachieve)
      WHERE avgachieve>=80
```

查询块可以出现在集合出现的地方。

```
SELECT eid,ename FROM exampaper
    WHERE eid in
            (SELECT eid FROM eeexam
            WHERE eeid=‘218811011013')
```

```
SELECT  *
FROM examinee
WHERE EXISTS
        (SELECT *
         FROM eeexam
        WHERE eeid=examinee.eeid AND eid= '0205000002')；
```

```
SELECT eedepa, AVG(eeage)

FROM examinee

GROUP BY eedepa

HAVING eedepa IN (SELECT eedepa FROM examinee WHERE eename='刘诗诗')
```

查询块只返回单个值，可以出现在单个属性名、单个表达式、单个常量出现的地方。

```
SELECT  dname,
        (SELECT COUNT(*)
         FROM examiner
         WHERE  department.dname = examiner.erdepa)
FROM department;
```

```
SELECT eeid，eename，eedepa
    FROM   examinee
    WHERE eedepa =  (SELECT eedepa
                FROM    examinee
                WHERE eeid='218811011028')；
```

```
SELECT (SELECT eedepa FROM examinee WHERE eeexam.eeid=examinee.eeid), avg(achieve)

FROM eeexam

GROUP BY (SELECT eedepa FROM examinee WHERE eeexam.eeid=examinee.eeid);
```

```
SELECT (SELECT eedepa FROM examinee WHERE eeexam.eeid=examinee.eeid),
        (SELECT eename FROM examinee WHERE eeexam.eeid=examinee.eeid),achieve
FROM eeexam
ORDER BY (SELECT eedepa FROM examinee WHERE eexam.eeid=examinee.eeid ),
          achieve ASC
```

```
SELECT *

FROM examinee

ORDER BY eeid

LIMIT ((SELECT count(*) FROM examinee)/4) OFFSET ((SELECT count(*) FROM examinee)/4);
```

- 查询块可以出现在任何表名出现的地方
- 查询块可以出现在行集合出现的地方
- 如果能确定查询块只返回单行单列单个值，查询块可以出现在单个属性名、单个表达式、单个常量出现的地方