

数据库第三次上机

本次上机概览

本次的上机任务主要是DQL的练习，实现不同需求的查询。

SQL的基础编写是数据库的常考知识，希望大家认真完成。

PPT仅给出关系模式，大家可以自行建表并插入数据测试查询结果；

(同学们在插入数据前先仔细阅读查询内容，查询对数据有一定要求！)

本次上机任务

均需要提交查询的SQL语句和查询结果截图。为了达到一定的展示效果，可以插入一些符合查询条件的数据。

(同学们在插入数据前先仔细阅读查询内容，查询对数据有一定要求！)

关系模式

职员（职员ID，姓名，性别，出生年月，职级，月薪，部门ID）（部门ID引用部门表的主键）

职员考勤（职员ID，出勤日期时间）（职员ID引用职员表的主键）

部门（部门ID，部门名称，部门经理ID）（部门经理ID引用职员表的主键）

监理（监理员ID，监理姓名）

工程（工程ID，工程工期，工程预算）（工程工期存的是天数，int型）

工程实施（工程ID，部门ID）（工程ID、部门ID分别引用工程、部门表的主键。）

工程监理（工程ID，监理员ID）（工程ID、监理员ID分别引用工程、监理表主键）

完成以下查询

- 1-1. 查找监理过工程ID为1的监理姓名。
- 1-2. 查询监理过部门ID为1的部门干过的工程的监理姓名。
- 1-3. 查询所有职员ID及他们的经理ID（注意有的职员可能没有部门）。
- 1-4. 查询所有张姓员工参与的工程的总预算。
- 1-5. 查询工程预算比所有工程工期大于10天的工程都要多的工程ID。
- 1-6. 查询所有职员最早的考勤记录。（给出查询结果：职员ID，最早考勤时间）
- 1-7. 查询参加过的工程的总预算额在10000以上的部门ID，及其预算额。
- 1-8. 请查询至少监理了三个工程的监理姓名。

本次上机任务

关系模式

学生（学号，姓名，年龄，性别，班级）

课程（课程号，课程名，学分）

选课（学号，课程号，教师号，成绩）

教师（教师号，教师名称）

完成以下查询

2-1. 查找选修了物理课的学生姓名

2-2. 找出所有姓诸的学生姓名（排除姓‘诸葛’的学生）

2-3. 查找教的学生的成绩都大于60分的教师（给出教师号即可）

2-4. 查询每个学生选修的课程数量，（给出查询结果：学号，选修课程数量）

2-5. 查找李力的所有不及格的课程名称和成绩，按成绩降序排列

2-6. 列出每门课的学分，选修的学生人数，及学生成绩的平均分

2-7. 选出所修课程总学分在10分以下的学生（注：不及格的课程没有学分）

均需要提交查询的**SQL语句**和**查询结果截图**。为了达到一定的展示效果，可以插入一些符合查询条件的数据。

（同学们在插入数据前先仔细阅读查询内容，查询对数据有一定要求！）

DQL子句：概览

DQL (Data Query Language) 常用子句

SELECT	-- 列投影
FROM	-- 聚集出原始数据集
JOIN ... ON	
WHERE	-- 行抽取 (判据: 列值; 优先级高)
[NOT] IN [NOT] EXISTS ANY/SOME ALL	
GROUP BY	-- 行分组
HAVING	-- 组抽取 (判据: 聚合函数; 优先级低)
ORDER BY	-- 行排序
TOP/LIMIT	-- 行截断
UNION	-- 行拼接
DISTINCT/ALL	-- 行压缩

用于：关系 **运算** 后 **筛选**、**整理** 而取出数据

[拓展阅读] 完整子句及相关详细文档参考：（两者部分细节有微小区别）

openGauss: <https://opengauss.org/zh/docs/1.0.1/docs/Developerguide/SELECT.html>;

SQL SERVER: <https://docs.microsoft.com/zh-cn/sql/t-sql/queries/select-transact-sql?view=sql-server-2017>

MYSQL: <https://dev.mysql.com/doc/refman/8.0/en/select.html>

DQL子句：概览

SELECT大体上是这样

```
SELECT 字段1, 字段2, ... (或者 *)  
      FROM 表名[, 表名...]  
      [JOIN 表名 ON 相等条件表达式]  
      [WHERE 条件表达式]  
      [GROUP BY 字段]  
      [HAVING 条件表达式]  
      [ORDER BY 字段];
```

(选择哪些数据列)
(从哪些表里选择)
(这些表怎么连接)
(得到的数据怎么筛选)
(需不需要对结果集进行分组)
(筛选分组后通过聚合函数得到的值)
(指定排序方式)

注意上面这些操作总体是一句话。

DQL：FROM子句

语法

FROM (子查询) AS 子结果集名

例如

……(如SELECT *)

```
FROM (SELECT Fid, Name FROM food WHERE Price<10)
      AS cheap_food
```

此时子查询应返回一个表

作用：表裁剪，一般用在表连接之前以减小开销

DQL：WHERE子句

语法

WHERE <子查询条件表达式>

<子查询条件表达式>可以形如

EXISTS (<SELECT语句>)

columnName [NOT] IN (<SELECT语句>)

columnName >= [ANY/SOME|ALL] (<SELECT语句>)

作用：进一步对结果集作行提取

DQL： EXISTS子查询

语法

WHERE EXISTS (子查询)

例如

.....

WHERE EXISTS

(SELECT * FROM Food WHERE City='北京')

EXISTS用于检查子查询是否至少会返回一行数据，该子查询实际上并不返回任何数据，而是返回值True或False。

(因此，EXISTS后面接的子查询SELECT后一般直接用*)

作用：指定一个子查询，检测‘行’的存在（EXISTS）与否（NOT EXISTS）

DQL： EXISTS子查询

例子：

有两张表： Customers（包含客户信息）和 Orders（包含订单信息）

```
SELECT *  
FROM Customers  
WHERE EXISTS (  
    SELECT *  
    FROM Orders  
    WHERE Customers.CustomerID = Orders.CustomerID  
    AND Orders.OrderDate = '2024-01-01'  
);
```

这个SQL查询将返回满足以下条件的所有顾客的记录：

在Orders表中存在订单的顾客，且该订单的订单日期为'2024-01-01'。

换句话说，它将返回在'2024-01-01'这一天有订单的所有顾客的信息。

DQL：IN子查询

语法

WHERE columnName [NOT] IN (子查询)

例如

.....

WHERE Fid NOT IN
(SELECT Fid FROM Food WHERE City='北京')

IN之后的子查询返回一个结果集，然后判断给定的值是否与其匹配

作用：确定给定的值是否与子查询或列表中的值相匹配（就是判断' \in '的关系）

IN 与 EXISTS 的原理区别：

- IN语句是把外表和内表作HASH JOIN（所以适用于内表小的情况）
- EXISTS语句是对外表作LOOP循环，每次LOOP循环再对内表进行查询。（所以适用于外表小的情况）

DQL：算符子查询

语法

WHERE columnName 运算符 [ANY/SOME|ALL] (子查询)

例如

.....

WHERE Price = (SELECT MAX(Price) FROM Food)

.....

WHERE Price >=

ANY(SELECT Price FROM Food WHERE City='北京')

All：对所有数据都满足条件，整个条件才成立。

Any：只要有一条数据满足条件，整个条件成立。

Some的作用和Any一样。

作用：表项交叉比较

DQL：子查询实例

```
SELECT 学号,姓名  
FROM 学生表  
WHERE 年龄 =  
      (SELECT MIN(年龄)  
       FROM 学生表)
```

```
SELECT SUM(成绩) AS 总分  
FROM 选课表  
WHERE 学号 =  
      (SELECT 学号  
       FROM 学生表  
       WHERE 姓名='Reimu')
```

```
SELECT 学号,姓名 FROM 学生表 S1  
WHERE NOT EXISTS  
      (SELECT * FROM 选课表 C1 JOIN 学生表 S2  
       ON S2.学号= C1.学号  
       WHERE 姓名='Marisa' AND NOT EXISTS  
            (SELECT * FROM 选课表 C2  
             WHERE C2.学号 = S1.学号  
                   AND C2.课程号 = C1.课程号))  
AND 学号 NOT IN  
      (SELECT 学号 FROM 选课表  
       WHERE 课程号 NOT IN  
            (SELECT 课程号 FROM 选课表  
             WHERE 学号 =  
                   (SELECT 学号  
                    FROM 学生表  
                    WHERE 姓名='Marisa' )))
```

DQL：TOP修饰符/LIMIT子句

SQL Server语法

SELECT TOP M * FROM ... //截取前M条记录
嵌入SELECT子句中作为列名前的修饰符，并非独立子句

MySQL语法

LIMIT [起始行N,] 行数M //截取从第N行开始的M条记录
是一个独立子句，一般放在SELECT结构的最末尾

OpenGauss语法

LIMIT { count | ALL }
OFFSET start
count声明返回的最大行数，而start声明开始返回行之前忽略的行数。如果两个都指定了，会在开始计算count个返回行之前先跳过start行。

作用：对数据集进行行截取

DQL：DISTINCT修饰符

语法

SELECT [DISTINCT|ALL] * FROM ...

默认为ALL，不忽略重复行

指定DISTINCT后即可忽略重复行

嵌入SELECT子句中作为列名前的修饰符，并非独立子句

作用：对得到的数据 进行 行压缩

例如：若从学生表中筛选出班级，应当使用 DISTINCT 来删去重复行。

DQL：UNION子句

语法

```
<SELECT语句>  
    UNION [ALL]      //默认无ALL，这样会压缩重复的行  
<SELECT语句>;
```

要求：两个SELECT语句的结果集拥有相同的列结构

作用：对数据集进行行拼接

DQL：子查询VS联表查询

简单的子查询与联表查询一般可以相互转化

子查询一般更加强大但效率稍低

联表查询：

```
select 学号
from 选课表 join 课程表
on 选课表.课程号= 课程表.课程号
where 课程名='数据库'
```

子查询：

```
select 学号
from 选课表
where 课程号 = (select 课程号
                 from 课程表
                 where 课程名='数据库')
```

DQL：其它常用语句/修饰符

- AS (或Alias) :用于给表起别名

例如：

```
SELECT po.OrderID, p.LastName, p.FirstName  
FROM Persons AS p, Product_Orders AS po  
WHERE p.LastName='BRANDO' AND p.FirstName='DIO'
```

- JOIN：表连接（分为INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN, 默认INNER JOIN）

例如：

```
SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo  
FROM Persons  
JOIN Orders  
ON Persons.Id_P = Orders.Id_P
```

效果等同于用逗号连接后再用WHERE筛选：

```
SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo  
FROM Persons, Orders  
WHERE Persons.Id_P = Orders.Id_P
```

DQL：其它常用语句/修饰符

- GROUP BY: 用于结合聚合函数，根据一个或多个列对结果集进行分组。
- 聚合函数: AVG(), COUNT(), MAX(), MIN(), FIRST(), LAST(), SUM() 等等。
- HAVING: WHERE 关键字无法与聚合函数一起使用（因为WHERE是对行筛选），因此用HAVING进行分组后聚合函数的筛选。
- ORDER BY: 默认按照升序（ASC）对记录进行排序，降序（DESC）需手动指定，可用于多关键字排序。

例如:

```
SELECT Customer, Age, SUM(OrderPrice)
FROM Orders, Customers
WHERE (Customer='Fulao' OR Customer='Zerone') AND (Orders.CID = Customers.CID)
GROUP BY Customer
HAVING SUM(OrderPrice)>114514
ORDER BY Age DESC, Customer ASC ;
```

相关参考

一般SQL语法:

<http://www.w3school.com.cn/sql/index.asp>

查询语句详细文档:

SQL SERVER: <https://docs.microsoft.com/zh-cn/sql/t-sql/queries/select-transact-sql?view=sql-server-2017>

MYSQL: <https://dev.mysql.com/doc/refman/8.0/en/select.html>

openGauss: <https://opengauss.org/zh/docs/1.0.1/docs/Developerguide/SELECT.html>

关系代数:

<https://www.2cto.com/database/201405/300939.html>

<https://blog.csdn.net/lin1094201572/article/details/79057694>

以及数据库课程PPT

多用搜索引擎:

<https://cn.bing.com/>

<https://www.google.com/>

<https://www.baidu.com/>

关于作业提交

TASK1

总结近几次上机来遇到的问题和解决方案（至少一个问题+解决方案），发布一个issue，title 设为“**学号_姓名_问题总结**”，链接为

<https://bhpan.buaa.edu.cn/link/AA49A06E0690F44AA6AD00C63CD9C353A9>

有效期限：2024-04-14 23:59

TASK2

T1-1 ~ T1-8

T2-1 ~ T2-7

请在**PDF/WORD**等任何方便助教阅读查看的文档中按照各个作业要求提交相关内容，记得**标清题号**。

打包成.zip .rar .7z等常见压缩格式，命名为“**学号_姓名_第三次实验**”。

提交网址：软件学院云平台**第三次上机**[北航软件学院-云平台 \(scs.buaa.edu.cn\)](https://scs.buaa.edu.cn)

(按要求提交)

TASK1 统计截止时间为本周日**24:00之前**

TASK2 作业截止时间为本周日**24:00之前，提交方式为提交到云平台。**