《数据库原理(1)》实验报告

实验三

姓名：严昕宇 学号：20121802 实验日期：2022.12.15

实验环境：

实验设备：Lenovo Thinkbook16+ 2022

操作系统：Windows 11 22H2

数据库管理系统：MySQL 8.0.31

数据库管理软件：JetBrains DataGrip 2022.3

实验内容：

对 school 数据库做如下查询：

1. 查询每门课程中分数最高的学生学号和学生姓名。
2. 查询年龄小于本学院平均年龄，所有课程总评成绩都高于所选课程平均总评成绩的学生学号、姓名和平均总评成绩，按年龄排序。
3. 求年龄大于所有女同学年龄的男学生姓名和年龄。
4. 检索每学期每门课的学生排名情况，输出学期，课程号，学号，成绩，排名；按学期降序，相同按课程号升序，课程相同按排名从高到低。
5. 查询计算机学院男生选修本学院教授开设的课不及格的且还未重修的课，输出学生的学期、学号、课号，按学期升序，学期相同按学号升序排列。

**实验过程1：**

查询每门课程中分数最高的学生学号和学生姓名

|  |
| --- |
| select distinct x.xh as '学号', s.xm as '姓名' from e as x  inner join s on x.xh = s.xh where (x.kh, x.zpcj) in (SELECT y.kh, *max*(y.zpcj) as MaxZpcj  from e as y  group by y.kh); |

实验结果1：

表格

描述已自动生成

**实验过程2：**

查询年龄小于本学院平均年龄，所有课程总评成绩都高于所选课程平均总评成绩的学生学号、姓名和平均总评成绩，按年龄排序

|  |
| --- |
| select s.xh as '学号', s.xm as '学期', e.zpcj as '成绩' from s  , e  , (select kh, *avg*(zpcj) as AvgZpcj from e group by kh) e1 where s.xh = e.xh  and e1.kh = e.kh  and (*UNIX\_TIMESTAMP*(s.csrq) > (select *avg*(*UNIX\_TIMESTAMP*(s.csrq)) from s))  and e.zpcj > e1.AvgZpcj order by *UNIX\_TIMESTAMP*(s.csrq) asc; |

这里的主要问题是如何处理年龄，由于年龄是date变量，无法进行avg()操作和排序操作。因此，使用了UNIX\_TIMESTAMP函数转化为时间戳，才可进行计算。

主要思路如下：将查询到的课号和平均总评成绩放在一个逻辑上的临时表e1,可以节省很多的步骤。把s表和e表通过学号进行连接，e表和e1表通过课号连接。这样就能比较出s表中大于平均出生日期的同学，也可以比较出e表中大于e1表中的平均成绩的同学。

实验结果2：

图形用户界面, 应用程序, 表格

描述已自动生成

**实验过程3：**

求年龄大于所有女同学年龄的男学生姓名和年龄

|  |
| --- |
| select xm as '姓名', *year*(*sysdate*()) - *year*(csrq) as '年龄' from s where xb = '男'  and csrq < all (select csrq  from s  where xb = '女'); |

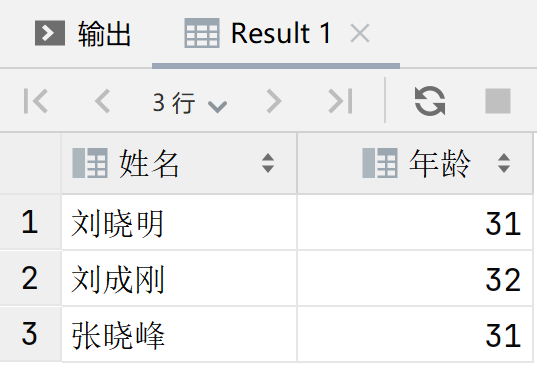
**由于s表中的csrq（出生日期）数据类型设定为datetime。因此，此处使用了MySQL的时间函数year ()和sysdate()，并以此判断年龄。**

在MySQL中，year()函数接受date参数，并返回日期的年份。

而获得当前日期+时间(date + time)的函数有now()、sysdate()。

其中sysdate() 日期时间函数跟 now() 类似，但不同之处在于：now() 在执行开始时值就得到了， sysdate() 在函数执行时动态得到值。

实验结果3：



**实验过程4：**

检索每学期每门课的学生排名情况，输出学期，课程号，学号，成绩，排名；按学期降序，相同按课程号升序，课程相同按排名从高到低

|  |
| --- |
| select xq as '学期',  kh as '课程号',  xh as '学号',  zpcj as '成绩',  *rank*() over (  partition by xq,kh  order by zpcj desc  ) as '排名' from e order by xq desc, kh, 排名; |

实验结果4：

表格

描述已自动生成

**实验过程5：**

查询计算机学院男生选修本学院教授开设的课不及格的且还未重修的课，输出学生的学期、学号、课号，按学期升序，学期相同按学号升序排列

|  |
| --- |
| select xq as '学期', xh as '学号', kh as '课号' from e where e.xh in (select xh  from s  where xb = '男')  and kh in (select kh  from o  where gh in (select gh  from t  where xl = '教授'))  and (select *count*(*\**)  from e x  where x.xh = e.xh  and x.kh = e.kh  and x.zpcj) = 1  and zpcj < 60 order by xq, xh; |

实验结果5：

图形用户界面, 应用程序

描述已自动生成

无目标结果

实验体会：

本次实验是我第二次使用结构化查询语言SQL。通过对于查询操作的进一步练习，以及聚类函数的练习，我初步了解了分组与聚类函数。与此同时，我也发现相比阅读书本上的SQL语句，在实验中亲自动手使用SQL语句，才是更好的学习方法，能加深我的记忆与理解。