《数据库技术》实验 2

课程名称：数据库技术	**学期**：秋季	**实验日期**：
实验名称：数据库安全性控制	**学院**：信息学院 **学时**：4 学时	**指导教师**：周小兵 **实验编号**：2

姓名：胡诚皓 **学号**：20201060330 **专业**：计算机科学与技术

一、实验目的

掌握自主存取控制权限的定义和维护方法，能够定义用户，分配权限给用户，回收权限。

二、实验内容

1．今有以下两个关系模式：

职工(职工号，姓名，年龄，职务，工资，部门号)

部门(部门号，名称，经理名，地址，电话号)

请用 SQL 的 GRANT 和 REVOKE 语句(加上视图机制)完成以下授权定义或存取控制功能：

（1）用户王明对两个表有 SELECT 权限。

（2）用户李勇对两个表有 INSERT 和 DELETE 权限。

（3）每个职工只对自己的记录有 SELECT 权限。

（4）用户刘星对职工表有 SELECT 权限，对工资字段具有更新权限。

（5）用户张新具有修改这两个表的结构的权限。

（6）用户周平具有对两个表的所有权限(读、插、改、删数据)，并具有给其他用户授权的权限。

（7）用户杨兰具有从每个部门职工中 SELECT 最高工资、最低工资、平均工资的权限，她不能查看每个人的工资。

2．针对 1 中实验内容的(1)~(7)的每一种情况，撤销各用户所授予的权限。

三、实验环境

Microsoft SQL Server 2016 或 MySQL

四、实验过程

注：在以下方格里按照题目的序号给出实验的代码和结果的截图

1．请用 SQL 的 GRANT 和 REVOKE 语句(加上视图机制)完成以下授权定义或存取控制功能：

（1）用户王明对两个表有 SELECT 权限。

```
grant select on table 职工 to 王明@localhost;
grant select on table 部门 to 王明@localhost;
```

```
mysql> use exp;
Database changed
mysql> grant select on table 职工 to 王明@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> grant select on table 部门 to 王明@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for 王明@localhost;
+-----------------------------------------------------------+
| Grants for 王明@localhost                                 |
+-----------------------------------------------------------+
| GRANT USAGE ON *.* TO `王明`@`localhost`                  |
| GRANT SELECT ON `exp`.`职工` TO `王明`@`localhost`        |
| GRANT SELECT ON `exp`.`部门` TO `王明`@`localhost`        |
+-----------------------------------------------------------+
3 rows in set (0.00 sec)
```

（2）用户李勇对两个表有 INSERT 和 DELETE 权限。

```
grant insert,delete on table 部门 to 李勇@localhost;
grant insert,delete on table 职工 to 李勇@localhost;
```

```
mysql> grant insert,delete on table 部门 to 李勇@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> grant insert,delete on table 职工 to 李勇@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for 李勇@localhost;
+-------------------------------------------------------------------+
| Grants for 李勇@localhost                                         |
+-------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `李勇`@`localhost`                          |
| GRANT INSERT, DELETE ON `exp`.`职工` TO `李勇`@`localhost`        |
| GRANT INSERT, DELETE ON `exp`.`部门` TO `李勇`@`localhost`        |
+-------------------------------------------------------------------+
3 rows in set (0.00 sec)
```

（3）每个职工只对自己的记录有 SELECT 权限。

用于测试的职工表如下

```
mysql> select * from 职工;
+------------+--------+--------+----------+--------+-----------+
| 职工号     | 姓名   | 年龄   | 职务     | 工资   | 部门号    |
+------------+--------+--------+----------+--------+-----------+
| 1485673838 | 王明   |     35 | 总经理   |   8500 | 3         |
| 2097578002 | 杨兰   |     24 | 设计师   |   4900 | 2         |
| 4726576355 | 周平   |     30 | 管理员   |   6100 | 4         |
| 7366462592 | 李勇   |     29 | 架构师   |   6200 | 2         |
| 8663550057 | 张新   |     32 | 程序员   |   5850 | 1         |
| 8663553524 | 刘星   |     30 | 程序员   |   5450 | 1         |
+------------+--------+--------+----------+--------+-----------+
6 rows in set (0.00 sec)
```

通过视图来实现此操作，职工通过此处创建的视图进行查询即可，要把视图授权给各用户

```
create view 自查询 as
select * from 职工 where concat(姓名,'@localhost')=user();
grant select on 自查询 to
王明@localhost,李勇@localhost,刘星@localhost,张新@localhost,
周平@localhost,杨兰@localhost;
```

```
mysql> create view 自查询 as
    -> select * from 职工 where concat(姓名,'@localhost')=user();
Query OK, 0 rows affected (0.00 sec)

mysql> grant select on 自查询 to
    -> 王明@localhost,李勇@localhost,刘星@localhost,张新@localhost,
    -> 周平@localhost,杨兰@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

使用用户杨兰进行测试

```
mysql> use exp;
Database changed
mysql> select user();
+-------------------+
| user()            |
+-------------------+
| 杨兰@localhost    |
+-------------------+
1 row in set (0.00 sec)

mysql> select * from 自查询;
+------------+--------+--------+-----------+--------+-----------+
| 职工号     | 姓名   | 年龄   | 职务      | 工资   | 部门号    |
+------------+--------+--------+-----------+--------+-----------+
| 2097578002 | 杨兰   |     24 | 设计师    |   4900 | 805311    |
+------------+--------+--------+-----------+--------+-----------+
1 row in set (0.00 sec)
```

（4）用户刘星对职工表有 SELECT 权限，对工资字段具有更新权限。

```
grant update (工资) on 职工 to 刘星@localhost;
grant select on 职工 to 刘星@localhost;
```

```
mysql> grant update (工资) on 职工 to 刘星@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> grant select on 职工 to 刘星@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for 刘星@localhost;
+-----------------------------------------------------------------------------+
| Grants for 刘星@localhost                                                   |
+-----------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `刘星`@`localhost`                                    |
| GRANT SELECT, UPDATE (`工资`) ON `exp`.`职工` TO `刘星`@`localhost`         |
+-----------------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

（5）用户张新具有修改这两个表的结构的权限。

```
grant alter on 职工 to 张新@localhost;
grant alter on 部门 to 张新@localhost;
```

```
mysql> grant alter on 职工 to 张新@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> grant alter on 部门 to 张新@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for 张新@localhost;
+-----------------------------------------------------------------+
| Grants for 张新@localhost                                       |
+-----------------------------------------------------------------+
| GRANT USAGE ON *.* TO `张新`@`localhost`                        |
| GRANT ALTER ON `exp`.`职工` TO `张新`@`localhost`               |
| GRANT SELECT ON `exp`.`自查询` TO `张新`@`localhost`            |
| GRANT ALTER ON `exp`.`部门` TO `张新`@`localhost`               |
+-----------------------------------------------------------------+
4 rows in set (0.00 sec)
```

（6）用户周平具有对两个表的所有权限(读、插、改、删数据)，并具有给其他用户授权的权限。

```
grant all privileges on 职工 to 周平@localhost with grant option;
grant all privileges on 部门 to 周平@localhost with grant option;
```

```
mysql> grant all privileges on 职工 to 张新@localhost with grant option;
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on 部门 to 张新@localhost with grant option;
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for 张新@localhost;
+-----------------------------------------------------------------------------------------+
| Grants for 张新@localhost                                                               |
+-----------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `张新`@`localhost`                                                |
| GRANT ALL PRIVILEGES ON `exp`.`职工` TO `张新`@`localhost` WITH GRANT OPTION            |
| GRANT SELECT ON `exp`.`自查询` TO `张新`@`localhost`                                    |
| GRANT ALL PRIVILEGES ON `exp`.`部门` TO `张新`@`localhost` WITH GRANT OPTION            |
+-----------------------------------------------------------------------------------------+
4 rows in set (0.00 sec)
```

（7）用户杨兰具有从每个部门职工中 SELECT 最高工资、最低工资、平均工资的权限，她不能查看每个人的工资。

用于测试的部门表

```
mysql> select * from 部门;
+----------+----------+----------+----------+----------+
| 部门号   | 名称     | 经理名   | 地址     | 电话号   |
+----------+----------+----------+----------+----------+
| 1        | 研发部   | 刘星     | 1栋102   | 111111   |
| 2        | 技术部   | 李勇     | 2栋202   | 222222   |
| 3        | 营销部   | 王明     | 2栋203   | 333333   |
| 4        | 餐饮部   | 周平     | 食堂     | 444444   |
+----------+----------+----------+----------+----------+
4 rows in set (0.00 sec)
```

通过视图完成此要求

```
create view 工资概况 as
select 部门.名称 as '部门名称',max(工资) as '最高工资',min(工资) as '最低工资',avg(工资) as '平均工资' from 职工,部门
where 职工.部门号=部门.部门号 group by 职工.部门号;
grant select on 工资概况 to 杨兰@localhost;
```

```
mysql> create view 工资概况 as
    -> select 部门.名称 as '部门名称',max(工资) as '最高工资',min(工资) as '最低工资',avg(工资) as '平均工资' from 职工,部门
    -> where 职工.部门号=部门.部门号 group by 职工.部门号;
Query OK, 0 rows affected (0.00 sec)

mysql> grant select on 工资概况 to 杨兰;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show grants for 杨兰@localhost;
+-----------------------------------------------------------------+
| Grants for 杨兰@localhost                                       |
+-----------------------------------------------------------------+
| GRANT USAGE ON *.* TO `杨兰`@`localhost`                        |
| GRANT SELECT ON `exp`.`工资概况` TO `杨兰`@`localhost`          |
| GRANT SELECT ON `exp`.`自查询` TO `杨兰`@`localhost`            |
+-----------------------------------------------------------------+
3 rows in set (0.00 sec)
```

使用用户杨兰进行测试

```
mysql> select user();
+------------------+
| user()           |
+------------------+
| 杨兰@localhost   |
+------------------+
1 row in set (0.00 sec)

mysql> select * from 工资概况;
+--------------+--------------+--------------+--------------+
| 部门名称     | 最高工资     | 最低工资     | 平均工资     |
+--------------+--------------+--------------+--------------+
| 研发部       |         5850 |         5450 |    5650.0000 |
| 技术部       |         6200 |         4900 |    5550.0000 |
| 营销部       |         8500 |         8500 |    8500.0000 |
| 餐饮部       |         6100 |         6100 |    6100.0000 |
+--------------+--------------+--------------+--------------+
4 rows in set (0.00 sec)
```

2．针对 1 中实验内容的(1)~(7)的每一种情况，撤销各用户所授予的权限。

（1）撤销权限，用户王明对两个表有 SELECT 权限。

```
revoke select on 职工 from 王明@localhost;
revoke select on 部门 from 王明@localhost;
```

```
mysql> revoke select on 职工 from 王明@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> revoke select on 部门 from 王明@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for 王明@localhsot;
ERROR 1141 (42000): There is no such grant defined for user '王明' on host 'localh
sot'
mysql>
```

（2）撤销权限，用户李勇对两个表有 INSERT 和 DELETE 权限。

```
revoke insert,delete on table 部门 from 李勇@localhost;
revoke insert,delete on table 职工 from 李勇@localhost;
```

```
mysql> revoke insert,delete on table 部门 from 李勇@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> revoke insert,delete on table 职工 from 李勇@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for 李勇@localhsot;
ERROR 1141 (42000): There is no such grant defined for user '李勇' on host
'localhsot'
mysql>
```

（3）撤销权限，每个职工只对自己的记录有 SELECT 权限。

收回各用户对自查询视图的权限即可

```
revoke select on 自查询 from
王明@localhost,李勇@localhost,刘星@localhost,张新@localhost,
周平@localhost,杨兰@localhost;
```

下图中以杨兰的权限为例

```
mysql> revoke select on 自查询 from
    -> 王明@localhost,李勇@localhost,刘星@localhost,张新@localhost,
    -> 周平@localhost,杨兰@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for 杨兰@localhost;
+-----------------------------------------------------------------+
| Grants for 杨兰@localhost                                       |
+-----------------------------------------------------------------+
| GRANT USAGE ON *.* TO `杨兰`@`localhost`                        |
| GRANT SELECT ON `exp`.`工资概况` TO `杨兰`@`localhost`          |
+-----------------------------------------------------------------+
2 rows in set (0.00 sec)
```

（4）撤销权限，用户刘星对职工表有 SELECT 权限，对工资字段具有更新权限。

```
revoke update (工资) on 职工 from 刘星@localhost;
revoke select on 职工 from 刘星@localhost;
```

```
mysql> revoke update (工资) on 职工 from 刘星@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> revoke select on 职工 from 刘星@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for 刘星@localhost;
+-----------------------------------------+
| Grants for 刘星@localhost               |
+-----------------------------------------+
| GRANT USAGE ON *.* TO `刘星`@`localhost`  |
+-----------------------------------------+
1 row in set (0.00 sec)
```

（5）撤销权限，用户张新具有修改这两个表的结构的权限。

```
revoke alter on 职工 from 张新@localhost;
revoke alter on 部门 from 张新@localhost;
```

```
mysql> revoke alter on 职工 from 张新@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> revoke alter on 部门 from 张新@localhost;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show grants for 张新@localhost;
+-----------------------------------------+
| Grants for 张新@localhost               |
+-----------------------------------------+
| GRANT USAGE ON *.* TO `张新`@`localhost`  |
+-----------------------------------------+
1 row in set (0.00 sec)
```

（6）撤销权限，用户周平具有对两个表的所有权限(读、插、改、删数据)，并具有给其他用户授权的权限。

```
revoke all privileges on 职工 from 周平@localhost;
revoke all privileges on 部门 from 周平@localhost;
revoke grant option on 职工 from 周平@localhost;
revoke grant option on 部门 from 周平@localhost;
```

```
mysql> revoke all privileges on 职工 from 周平@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> revoke all privileges on 部门 from 周平@localhost;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> revoke grant option on 职工 from 周平@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> revoke grant option on 部门 from 周平@localhost;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show grants for 周平@localhost;
+-----------------------------------------+
| Grants for 周平@localhost               |
+-----------------------------------------+
| GRANT USAGE ON *.* TO `周平`@`localhost`  |
+-----------------------------------------+
1 row in set (0.00 sec)
```

（7）撤销权限，用户杨兰具有从每个部门职工中 SELECT 最高工资、最低工资、平均工资的权限，她不能查看每个人的工资。

收回杨兰对工资概况视图的 select 权限即可

```
revoke select on 工资概况 from 杨兰@localhost;
```

```
mysql> revoke select on 工资概况 from 杨兰@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for 杨兰@localhost;
+-----------------------------------------+
| Grants for 杨兰@localhost               |
+-----------------------------------------+
| GRANT USAGE ON *.* TO `杨兰`@`localhost`  |
+-----------------------------------------+
1 row in set (0.00 sec)
```

五、实验总结

1．遇到的问题及解决过程

一开始创建用户并授予权限后，切换到该用户登录数据库，发现在该用户的状态下看不到要使用的数据库，在网上搜索也没有找到有效的解决方法。

在经过和同学讨论交流后，发现在表示用户时，只使用了用户名，并没有在后面添加@localhost。MySQL 默认会在用户后添加@%，意味着允许用户从任何 TCP/IP 连接登录数据库。然而通过命令行登录并不算通过 TCP/IP 登录，而是通过 localhost，因此用户无法看到数据库。将用户通过“用户名@localhost”表示后解决了上述问题。

2．产生的错误及原因分析

书上的一些命令格式是在 SQL Server 上使用的，在 MySQL 上并不兼容，需要用 MySQL 的方言语法进行转换，甚至要转换思路，通过其他灵活的方式实现相同的功能。这次实验中用户权限授予的语法就有较大的区别。在第 3 小题中“给予用户查看自身职工表项”权限，就使用视图和 MySQL 的函数灵活实现了功能。在编写 MySQL 的 SQL 时要注意查看 MySQL 的相关文档，进一步熟悉 MySQL 的方言。

3．体会和收获

本次实验真正应用视图实现了相关功能，相较于单纯的用户授权，使用视图可以更好地实现信息权限的管理。同时，视图可帮助用户屏蔽真实表结构变化带来的影响。其次，在使用视图实现数据访问的控制时，发现视图明显可以简化用户权限的管理。只需授予用户使用视图的权限，而不必指定用户只能使用表的特定列，也增加了安全性。另外，视图可以为用户集中数据，简化用户的数据查询和处理。有时用户所需要的数据分散在多个表中或是是需要一定基本计算才能得到的数据，定义视图可将它们集中在一起，从而方便用户的数据查询和处理。

六、参考文献

1．数据库系统概论（第 5 版）[M]，王珊、萨师煊主编，高等教育出版社，2014 年。

2．数据库系统概论（第 5 版）实验指导与习题解析[M]，王珊、萨师煊主编，高等教育出版社，2014 年。

3．数据库系统实验设计[M],王浩鸣、李秀娟主编，中国铁道出版社，2021

七、教师评语