

课程作业

课程名称 : 数据库系统原理
作业次数 : 作业#4
学号 : 21281280
姓名 : 柯劲帆
班级 : 物联网2101班
指导老师 : 郝爽
修改日期 : 2024年4月14日

1. 题目1

一、视图设计与操作（上机检查）

请针对你在作业#2中所设计的用户界面的数据需求

- 1) 至少设计并创建行列子集视图、带表达式的视图和分组视图各一个，并将代码加入自己的建库脚本中。
- 2) 编写测试代码，通过视图实现数据查询操作
- 3) 对不同类别视图尝试进行数据修改和删除操作，看能否实现，分析其原因
- 4) 在实验室报告中总结视图的主要作用。

1.1 & 1.2 & 1.3

行列子集视图：

```
1  mysql> CREATE VIEW Trainschedule AS
2      -> SELECT
3          ->     Passbys.TrainID,
4          ->     Stations.Name AS StationName,
5          ->     Passbys.arrive_time
6      -> FROM
7          ->     Passbys
8      -> JOIN
9          ->     Stations ON Passbys.StationID = Stations.ID;
10 Query OK, 0 rows affected (0.01 sec)
11
12 mysql> SELECT *
13     -> FROM Trainschedule
14     -> WHERE TrainID = 'G103';
15 +-----+-----+-----+
16 | TrainID | StationName | arrive_time |
17 +-----+-----+-----+
18 | G103    | 沧州西      | 2024-03-28 06:20:00 |
19 | G103    | 济南西      | 2024-03-28 08:01:00 |
20 | G103    | 徐州东      | 2024-03-28 09:07:00 |
21 | G103    | 宿州东      | 2024-03-28 09:28:00 |
22 | G103    | 蚌埠南      | 2024-03-28 09:56:00 |
23 | G103    | 南京南      | 2024-03-28 10:40:00 |
24 | G103    | 无锡南      | 2024-03-28 11:27:00 |
25 | G103    | 上海虹桥    | 2024-03-28 11:58:00 |
26 +-----+-----+-----+
27 8 rows in set (0.00 sec)
28
29 mysql> UPDATE Trainschedule
30     -> SET arrive_time = '2024-03-28 07:00:00'
31     -> WHERE TrainID = 'G103' AND StationName = '北京南';
32 Query OK, 0 rows affected (0.00 sec)
33 Rows matched: 0  Changed: 0  Warnings: 0
```

带表达式的视图：

```

1  mysql> CREATE VIEW TrainStopsWithDuration AS
2      -> SELECT
3      ->     Passbys.TrainID,
4      ->     Stations.Name AS StationName,
5      ->     Passbys.arrive_time,
6      ->     Passbys.leave_time,
7      ->     TIMESTAMPDIFF(MINUTE, Passbys.arrive_time, Passbys.leave_time) AS
StayDuration
8      -> FROM
9      ->     Passbys
10     -> JOIN
11     ->     Stations ON Passbys.StationID = Stations.ID;
12  Query OK, 0 rows affected (0.00 sec)
13
14  mysql> SELECT *
15      -> FROM TrainStopsWithDuration
16      -> WHERE TrainID = 'G103';
17  +-----+-----+-----+-----+-----+
18  | TrainID | StationName | arrive_time          | leave_time          |
StayDuration |
19  +-----+-----+-----+-----+-----+
20  | G103    | 沧州西      | 2024-03-28 06:20:00 | 2024-03-28 06:20:00 |
0 |
21  | G103    | 济南西      | 2024-03-28 08:01:00 | 2024-03-28 08:04:00 |
3 |
22  | G103    | 徐州东      | 2024-03-28 09:07:00 | 2024-03-28 09:09:00 |
2 |
23  | G103    | 宿州东      | 2024-03-28 09:28:00 | 2024-03-28 09:33:00 |
5 |
24  | G103    | 蚌埠南      | 2024-03-28 09:56:00 | 2024-03-28 09:58:00 |
2 |
25  | G103    | 南京南      | 2024-03-28 10:40:00 | 2024-03-28 10:43:00 |
3 |
26  | G103    | 无锡南      | 2024-03-28 11:27:00 | 2024-03-28 11:29:00 |
2 |
27  | G103    | 上海虹桥    | 2024-03-28 11:58:00 | 2024-03-28 11:58:00 |
0 |
28  +-----+-----+-----+-----+-----+
29  8 rows in set (0.01 sec)
30
31  mysql> -- 尝试更新 TrainStopsWithDuration 视图
32  mysql> UPDATE TrainStopsWithDuration
33      -> SET StayDuration = 10
34      -> WHERE TrainID = 'G103' AND StationName = '北京南';
35  ERROR 1348 (HY000): Column 'StayDuration' is not updatable

```

分组视图：

创建：

```

1  mysql> CREATE VIEW TrainRoutes AS
2      -> SELECT

```

```

3      ->     Trains.ID AS TrainID,
4      ->     TIMESTAMPDIFF(HOUR, MIN(Passbys.arrive_time),
MAX(Passbys.leave_time)) AS TotalDuration
5      -> FROM
6      ->     Trains
7      -> JOIN
8      ->     Passbys ON Trains.ID = Passbys.TrainID
9      -> GROUP BY
10     ->     Trains.ID;
11 Query OK, 0 rows affected (0.01 sec)
12
13 mysql> SELECT *
14     -> FROM TrainRoutes;
15 +-----+-----+
16 | TrainID | TotalDuration |
17 +-----+-----+
18 | G103    |             5 |
19 +-----+-----+
20 1 row in set (0.01 sec)
21
22 mysql> DELETE FROM TrainRoutes
23     -> WHERE TrainID = 'G103';
24 ERROR 1288 (HY000): The target table TrainRoutes of the DELETE is not
updatable

```

1.4

尝试通过视图进行数据修改或删除受限于视图的结构和 SQL 引擎的限制。

可更新的视图通常需要直接对应到单个底层表的列，且不能包含聚合函数、不同的数据源或其他复杂的 SQL 操作，如 DISTINCT、GROUP BY、ORDER BY 等。

此外，视图中的所有字段都必须直接来源于一个底层数据表，而且这个表中相应的字段必须允许被更新。

2. 题目2

二、用户口令管理设计题

用户注册与认证管理是数据库系统的基本功能，请为你在作业#2中选择的具体案例设计一个用户注册与认证中的口令管理解决方案，要求如下

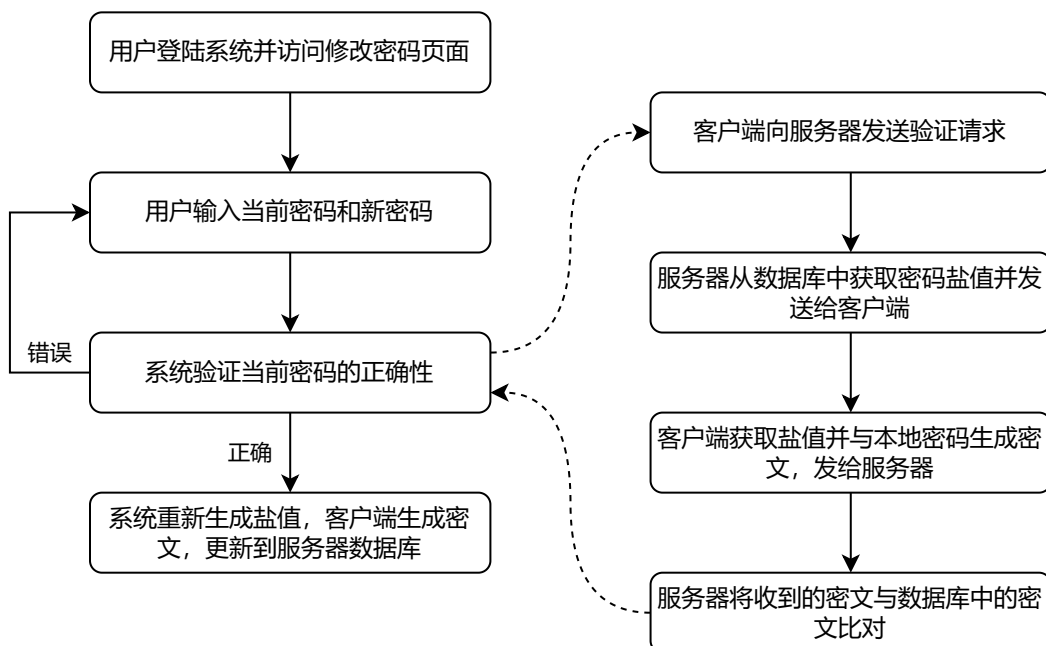
1. 用户口令采用密文存储，采用不可逆加密算法加密，查资料了解现有的不可逆算法。
2. 设计用户修改密码流程，用VISO画出流程图
3. 设计用户密码重置业务流程（可以包含线下流程），用VISO画出业务流程图

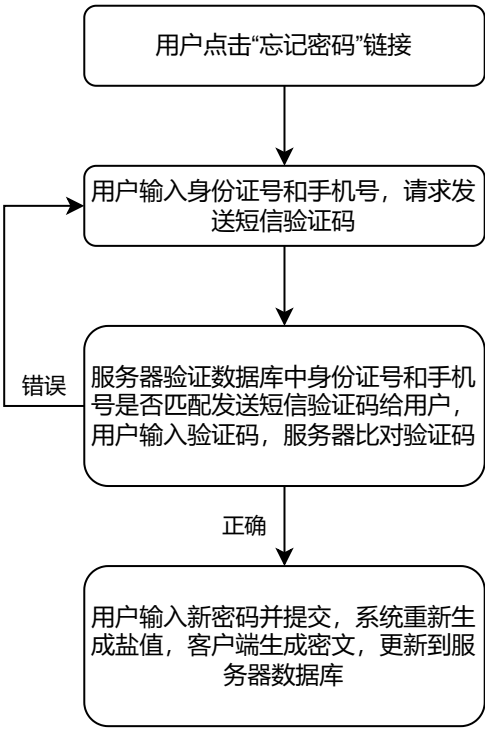
2.1

常见的不可逆加密算法有

- MD5
- HMAC
- SHA1: 默认会产生一个160位的信息摘要，SHA1加密算法的安全性要比MD5更高
- SHA2: (SHA-224、SHA-256、SHA-384、SHA-512，SHA后面的数字表示的是加密后的字符串长度)，SHA2加密算法安全性比SHA1的要高。
- bcrypt: 设计用于密码存储，对抗暴力破解攻击效果更好。
- Argon2: 这是一个较新的算法，获得了密码哈希竞赛的胜利，为抵抗大规模攻击提供了强有力的防御。

2.2





3. 题目3

三、用户授权操作题（上机检查）

为你在作业#2中选择的具体案例增加若干个数据库用户，并对各个用户进行相应的授权操作，至少应完成如下操作：

1. 创建用户
2. 为不同的用户授予和收回表级SELECT, INSERT, UPDATE, DELETE, ALL PRIVILEGES权限
3. 为不同的用户授予和收回属性列级的SELECT, INSERT, UPDATE的权限
4. 创建角色，并把角色授予用户或收回用户的某个角色

将以上语句写成单独的SQL脚本，上机实验调试通过，编写测试代码，测试权限设置或取消是否成功，并将调试结果截屏写入实验报告。

```
1 DROP USER IF EXISTS 'ticketAgent'@'localhost';
2 DROP USER IF EXISTS 'stationManager'@'localhost';
3 DROP USER IF EXISTS 'adminUser'@'localhost';
4 DROP ROLE IF EXISTS 'ticketRole';
5 DROP ROLE IF EXISTS 'stationRole';
6
7 CREATE USER 'ticketAgent'@'localhost' IDENTIFIED BY 'password123';
8 CREATE USER 'stationManager'@'localhost' IDENTIFIED BY 'password123';
9 CREATE USER 'adminUser'@'localhost' IDENTIFIED BY 'password123';
10
11
12 GRANT SELECT, INSERT ON DBLab.Tickets TO 'ticketAgent'@'localhost';
13 GRANT ALL PRIVILEGES ON DBLab.Stations TO 'stationManager'@'localhost';
14 GRANT SELECT, INSERT, UPDATE, DELETE ON DBLab.* TO 'adminUser'@'localhost';
15
16
17 REVOKE INSERT ON DBLab.Tickets FROM 'ticketAgent'@'localhost';
18
19 GRANT SELECT(Price) ON DBLab.Tickets TO 'ticketAgent'@'localhost';
20 GRANT UPDATE(Name, City) ON DBLab.Stations TO 'stationManager'@'localhost';
21
22 CREATE ROLE 'ticketRole';
23 CREATE ROLE 'stationRole';
24
25
26 GRANT SELECT, INSERT ON DBLab.Tickets TO 'ticketRole';
27 GRANT SELECT, UPDATE ON DBLab.Stations TO 'stationRole';
28
29
30 GRANT 'ticketRole' TO 'ticketAgent'@'localhost';
31 GRANT 'stationRole' TO 'stationManager'@'localhost';
32
33 REVOKE 'ticketRole' FROM 'ticketAgent'@'localhost';
34
35 FLUSH PRIVILEGES;
```

执行结果：

```
1 $ mysql -u root -p
2 Enter password:
3 welcome to the MySQL monitor.  Commands end with ; or \g.
4 Your MySQL connection id is 42
5 Server version: 8.0.33 MySQL Community Server - GPL
6
7 Copyright (c) 2000, 2023, Oracle and/or its affiliates.
8
9 Oracle is a registered trademark of Oracle Corporation and/or its
10 affiliates. Other names may be trademarks of their respective
11 owners.
12
13 Type 'help;' or '\h' for help. Type '\c' to clear the current input
    statement.
14
15 mysql> source ./privilege.sql
16 Query OK, 0 rows affected (0.01 sec)
17
18 Query OK, 0 rows affected (0.01 sec)
19
20 Query OK, 0 rows affected (0.00 sec)
21
22 Query OK, 0 rows affected (0.01 sec)
23
24 Query OK, 0 rows affected (0.00 sec)
25
26 Query OK, 0 rows affected (0.01 sec)
27
28 Query OK, 0 rows affected (0.01 sec)
29
30 Query OK, 0 rows affected (0.02 sec)
31
32 Query OK, 0 rows affected (0.01 sec)
33
34 Query OK, 0 rows affected (0.00 sec)
35
36 Query OK, 0 rows affected (0.00 sec)
37
38 Query OK, 0 rows affected (0.01 sec)
39
40 Query OK, 0 rows affected (0.00 sec)
41
42 Query OK, 0 rows affected (0.01 sec)
43
44 Query OK, 0 rows affected (0.00 sec)
45
46 Query OK, 0 rows affected (0.01 sec)
47
48 Query OK, 0 rows affected (0.00 sec)
49
50 Query OK, 0 rows affected (0.01 sec)
51
52 Query OK, 0 rows affected (0.00 sec)
53
54 Query OK, 0 rows affected (0.01 sec)
55
```

```
56 Query OK, 0 rows affected (0.00 sec)
57
58 Query OK, 0 rows affected (0.01 sec)
59
60 mysql> exit
61 Bye
62 $ mysql -u ticketAgent -ppassword123
63 mysql: [warning] Using a password on the command line interface can be
insecure.
64 welcome to the MySQL monitor.  Commands end with ; or \g.
65 Your MySQL connection id is 45
66 Server version: 8.0.33 MySQL Community Server - GPL
67
68 Copyright (c) 2000, 2023, Oracle and/or its affiliates.
69
70 Oracle is a registered trademark of Oracle Corporation and/or its
71 affiliates. Other names may be trademarks of their respective
72 owners.
73
74 Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.
75
76 mysql> use DBLab;
77 Reading table information for completion of table and column names
78 You can turn off this feature to get a quicker startup with -A
79
80 Database changed
81 mysql> select * from Stations;
82 ERROR 1142 (42000): SELECT command denied to user 'ticketAgent'@'localhost'
for table 'Stations'
83 mysql>
```

4. 题目4

四、数据库完整性实验题

完善作业#3中你的数据模型完整性设计，模型中至少应体现如下完整性要求：

- 1) 实体完整性
- 2) 参照完整性
- 3) 用户定义的完整性
- 4) 对部分完整性约束进行命名
- 5) 创建域以实现某些属性内容完整性

将以上实现代码加入到作业#3 的脚本中并调试通过，用违犯相应约束的语句尝试，以验证完整性约束实施效果。注意观察不同的RDBMS平台对完整性的支持程度。

原本的建表语句中已满足1、2、3。做如下修改：

```
1 CREATE TABLE Passengers (  
2     ID BIGINT PRIMARY KEY,  
3     `Name` VARCHAR ( 255 ) NOT NULL,  
4     Phone_number BIGINT UNIQUE NOT NULL,  
5     CHECK ( REGEXP_LIKE ( ID, '^\\d{18}$' ) ),  
6     CHECK ( REGEXP_LIKE ( Phone_number, '^\\d{11}$' ) )  
7 );
```

对完整性约束进行命名：

```
1 CREATE TABLE Passengers (  
2     ID BIGINT PRIMARY KEY,  
3     `Name` VARCHAR(255) NOT NULL,  
4     Phone_number BIGINT UNIQUE NOT NULL,  
5     CONSTRAINT CHK_ID_FORMAT CHECK (REGEXP_LIKE(ID, '^\\d{18}$')),  
6     CONSTRAINT CHK_PHONE_FORMAT CHECK (REGEXP_LIKE(Phone_number, '^\\d{11}$'))  
7 );
```

尝试违反：

```
1 mysql> INSERT INTO Passengers (ID, `Name`, Phone_number) VALUES  
2     (123456789012345678, 'John Doe', 1234567890);  
2 ERROR 3819 (HY000): Check constraint 'CHK_PHONE_FORMAT' is violated.
```

MySQL不直接支持域，但是可以通过 CHECK 约束来实现类似的功能。