

第三次“数据库原理及应用”课程作业

姓名：袁昊男 学号：2018091618008

在一个汽车租赁管理系统中，假定其数据库 CarRentDB 包括客户表（CLIENT）、汽车信息表（CAR）、租赁价目表（RENT_PRICE）、租赁登记表（RENT_REG）、租赁费用表（RENT_FEE）。系统用户角色有客户、业务员、经理、系统管理员。

在 PostgreSQL 数据库中，完成角色管理、权限管理、用户管理，以及数据库备份与恢复管理等操作，具体要求如下：

- 1) 在数据库 CarRentDB 中，创建 R_Client（客户）、R_SalesMan（业务员）、R_Manager（经理）、R_Administrator（系统管理员）角色。
- 2) 在数据库 CarRentDB 中，分别定义各个角色对数据库表对象的访问权限。
- 3) 创建用户 ClientUser 为客户角色用户，用户 SalesManUser 为业务员角色用户，用户 ManagerUser 为经理角色用户，用户 AdministratorUser 为系统管理员角色用户。
- 4) 分别以不同用户登录访问数据库，尝试进行不同类型访问操作。
- 5) 以管理员身份进行 CarRentDB 数据库备份处理。分别创建数据库备份、schema 备份、数据库表备份。
- 6) 当破坏数据库 CarRentDB 后，使用备份文件进行数据库恢复处理。

作业要求：在 PostgreSQL 数据库中创建 CarRentDB 数据库及其数据库表，插入样本数据，然后按照以上数据库系统管理要求进行 SQL 访问编程操作。给出每个问题解决的步骤、SQL 语句操作、执行结果界面，并对结果进行说明。

1、数据表字段结构定义

表 1 客户表（CLIENT）

字段名称	字段编码	数据类型	字段大小	必填字段	备注
客户编号	ClientNo	Char	12	是	主键
姓名	Name	Varchar	10	是	
性别	Gender	Char	2	是	
身份证号	ClientID	Char	18	是	
驾驶证编号	LicenseID	Char	12	否	
联系电话	Tel	Char	11	是	
居住地址	Addr	Varchar	50	是	

表 2 汽车信息表（CAR）

字段名称	字段编码	数据类型	字段大小	必填字段	备注
车辆编号	CarNo	Char	12	是	主键
车主姓名	HostName	Varchar	10	是	
车主身份证号	HostID	Char	18	是	
车牌号	CarID	Char	10	是	
类型	Type	Char	10	是	取值范围：轿车、越野车、SUV、货车、客车
颜色	Color	Char	10	是	
状态	State	Varchar	10	是	取值范围：已出租、未出租

表 3 租赁价目表 (RENT_PRICE)

字段名称	字段编码	数据类型	字段大小	必填字段	备注
车辆编号	CarNo	Char	12	是	主键
小时租赁价格	HourPrice	Money	/	是	

表 4 租赁登记表 (RENT_REG)

字段名称	字段编码	数据类型	字段大小	必填字段	备注
租赁单号	RentNo	Char	12	是	主键
客户编号	ClientNo	Char	12	是	外键
车辆编号	CarNo	Char	12	是	外键
租赁日期	StartDate	Date	/	是	
租赁天数	Period	Numeric	(2, 0)	是	

表 5 租赁费用表 (RENT_FEE)

字段名称	字段编码	数据类型	字段大小	必填字段	备注
租赁单号	RentNo	Char	12	是	主键, 外键
客户编号	ClientNo	Char	12	是	外键
期内费用	FeeWithin	Money	/	是	
超时费用	FeeBeyond	Money	/	是	
车辆损坏费用	DamageFee	Money	/	是	

2、角色与权限

表 6 角色与权限表

表	客户	业务员	经理	系统管理员
客户表	查询、插入 修改、删除	查询	查询	查询、插入 修改、删除
汽车信息表	查询	查询、插入 修改、删除	查询、插入 修改、删除	查询、插入 修改、删除
租赁价目表	查询	查询	查询、插入 修改、删除	查询、插入 修改、删除
租赁登记表	查询、插入	查询、插入、修改	查询、插入、修改	查询、插入 修改、删除
租赁费用表	查询	查询、插入、修改	查询、插入、修改	查询、插入 修改、删除

3、创建数据库 CarRentDB

1.	CREATE DATABASE "CarRentDB"
2.	WITH
3.	OWNER = postgres
4.	ENCODING = 'UTF8'
5.	CONNECTION LIMIT = -1;

4、创建数据表

1.	CREATE TABLE public."CLIENT"
2.	(
3.	"ClientNo" character(12) NOT NULL,
4.	"Name" character varying(10) NOT NULL,
5.	"Gender" character(2) NOT NULL,
6.	"ClientID" character(18) NOT NULL,
7.	"LicenseID" character(12) NOT NULL,
8.	"Tel" character(11) NOT NULL,

```

9.      "Addr"      character varying(50) NOT NULL,
10.     PRIMARY KEY ("ClientNo")
11. )
12.
13.
14. CREATE TABLE public."CAR"
15. (
16.     "CarNo"      character(12)          NOT NULL,
17.     "HostName"   character varying(10)  NOT NULL,
18.     "HostID"     character(18)          NOT NULL,
19.     "CarID"      character(10)          NOT NULL,
20.     "Type"       character(10)          NOT NULL CHECK("Type" IN ('轿
        车','越野车','SUV','货车','客车')),
21.     "Color"      character(10).         NOT NULL,
22.     "State"      character varying(10)  NOT NULL CHECK("State" IN ('
        已出租','未出租')),
23.     PRIMARY KEY ("CarNo")
24. )
25.
26.
27. CREATE TABLE public."RENT_PRICE"
28. (
29.     "CarNo"      character(12)          NOT NULL,
30.     "HourPrice"  money                  NOT NULL,
31.     PRIMARY KEY ("CarNo")
32. )
33.
34.
35. CREATE TABLE public."RENT_REG"
36. (
37.     "RentNo"     character(12)          NOT NULL,
38.     "ClientNo"   character(12)          NOT NULL,
39.     "CarNo"      character(12)          NOT NULL,
40.     "StartDate"  date                  NOT NULL,
41.     "Period"     numeric(2, 0)          NOT NULL,
42.     PRIMARY KEY ("RentNo"),
43.     CONSTRAINT "ClientNo_fk" FOREIGN KEY ("ClientNo")
44.         REFERENCES public."CLIENT" ("ClientNo") MATCH SIMPLE
45.         ON UPDATE NO ACTION
46.         ON DELETE NO ACTION,
47.     CONSTRAINT "CarNo_fk" FOREIGN KEY ("CarNo")
48.         REFERENCES public."CAR" ("CarNo") MATCH SIMPLE
49.         ON UPDATE NO ACTION
50.         ON DELETE NO ACTION
51. )
52.
53.
54. CREATE TABLE public."RENT_FEE"
55. (
56.     "RentNo"     character(12)          NOT NULL,
57.     "ClientNo"   character(12)          NOT NULL,
58.     "FeeWithin"  money                  NOT NULL,
59.     "FeeBeyond"  money                  NOT NULL,
60.     "DamageFee"  money                  NOT NULL,
61.     PRIMARY KEY ("RentNo"),
62.     CONSTRAINT "Rent_No_fk" FOREIGN KEY ("ClientNo")
63.         REFERENCES public."CLIENT" ("ClientNo") MATCH SIMPLE
64.         ON UPDATE NO ACTION

```

65.	ON DELETE NO ACTION
66.)	

5、插入数据结果

数据输出								解释	消息	Notifications
	ClientNo [PK] character (12)	Name character varying (10)	Gender character (2)	ClientID character (18)	LicenseID character (12)	Tel character (11)	Addr character varying (50)			
1	202005091001	张三	男	513800196702340915	511342849123	13311001100	四川省成都市			
2	202005091002	李四	男	512418197302230961	479312844310	18911001100	四川省成都市			
3	202005091003	王二麻子	女	526718200301230862	781112235371	19900110011	四川省成都市			

图1 “CLIENT 表” 查询结果

数据输出								解释	消息	Notifications
	CarNo [PK] character (12)	HostName character varying (10)	HostID character (18)	CarlD character (10)	Type character (10)	Color character (10)	State character varying (10)			
1	202005123401	李明	512745199506120915	川A22688	轿车	黑色	未出租			
2	202005123402	刘磊	503825199205020617	川A2913Q	SUV	红色	未出租			
3	202005123403	郑阳	512503198712100219	川A1028X	货车	蓝色	已出租			

图2 “CAR 表” 查询结果

	CarNo [PK] character (12)	HourPrice money
1	202005123401	\$88.00
2	202005123402	\$128.00
3	202005123403	\$99.00

图3 “RENT_PRICE 表” 查询结果

数据输出						解释	消息	Notifications		
	<div><div>▲</div><div>RentNo</div><div>[PK] character (12)</div></div>	<div><div></div><div>ClientNo</div><div>character (12)</div></div>	<div><div></div><div>CarNo</div><div>character (12)</div></div>	<div><div></div><div>StartDate</div><div>date</div></div>	<div><div></div><div>Period</div><div>numeric (2)</div></div>					
1	202005041722	202005091001	202005123401	2020-05-04						10
2	202005060912	202005091002	202005123402	2020-05-08						12
3	202005091438	202005091003	202005123403	2020-05-10						5

图4 “RENT_REG 表” 查询结果

数据输出						解释	消息	Notifications		
	RentNo [PK] character (12)	ClientNo character (12)	FeeWithin money	FeeBeyond money	DamageFee money					
1	202005041722	202005091001	\$1,200.00	\$0.00	\$0.00					
2	202005060912	202005091002	\$1,500.00	\$380.00	\$2,000.00					
3	202005091438	202005091003	\$2,600.00	\$520.00	\$0.00					

图5 “RENT_FEE 表” 查询结果

6、创建角色

1.	CREATE USER "R_Client" WITH
2.	LOGIN
3.	NOSUPERUSER
4.	NOCREATEDB
5.	NOCREATEROLE
6.	INHERIT

```

7.     NOREPLICATION
8.     CONNECTION LIMIT -1
9.     PASSWORD '123456';
10.
11. CREATE USER "R_SalesMan" WITH
12.     LOGIN
13.     NOSUPERUSER
14.     NOCREATEDB
15.     CREATEROLE
16.     INHERIT
17.     NOREPLICATION
18.     CONNECTION LIMIT -1
19.     PASSWORD '123456';
20.
21. CREATE USER "R_Manager" WITH
22.     LOGIN
23.     NOSUPERUSER
24.     CREATEDB
25.     CREATEROLE
26.     INHERIT
27.     REPLICATION
28.     CONNECTION LIMIT -1
29.     PASSWORD '123456';
30.
31. CREATE USER "R_Administrator" WITH
32.     LOGIN
33.     SUPERUSER
34.     CREATEDB
35.     CREATEROLE
36.     INHERIT
37.     REPLICATION
38.     CONNECTION LIMIT -1
39.     PASSWORD '123456';

```



图6 角色创建结果

7、授予访问权限

```

1. GRANT SELECT,INSERT,UPDATE,DELETE ON "CLIENT" TO "R_Client";
2. GRANT SELECT ON "CAR" TO "R_Client";
3. GRANT SELECT ON "RENT_PRICE" TO "R_Client";
4. GRANT SELECT,INSERT ON "RENT_REG" TO "R_Client";
5. GRANT SELECT ON "RENT_FEE" TO "R_Client";
6.
7. GRANT SELECT ON "CLIENT" TO "R_SalesMan";
8. GRANT SELECT,INSERT,UPDATE,DELETE ON "CAR" TO "R_SalesMan";
9. GRANT SELECT ON "RENT_PRICE" TO "R_SalesMan";
10. GRANT SELECT,INSERT,UPDATE ON "RENT_REG" TO "R_SalesMan";
11. GRANT SELECT,INSERT,UPDATE ON "RENT_FEE" TO "R_SalesMan";
12.

```

```

13. GRANT SELECT ON "CLIENT" TO "R_Manager";
14. GRANT SELECT,INSERT,UPDATE,DELETE ON "CAR" TO "R_Manager";
15. GRANT SELECT,INSERT,UPDATE,DELETE ON "RENT_PRICE" TO "R_Manager";
16. GRANT SELECT,INSERT,UPDATE ON "RENT_REG" TO "R_Manager";
17. GRANT SELECT,INSERT,UPDATE ON "RENT_FEE" TO "R_Manager";
18.
19. GRANT SELECT,INSERT,UPDATE,DELETE ON "CLIENT" TO "R_Administrator";
20. GRANT SELECT,INSERT,UPDATE,DELETE ON "CAR" TO "R_Administrator";
21. GRANT SELECT,INSERT,UPDATE,DELETE ON "RENT_PRICE" TO "R_Administrator"
;
22. GRANT SELECT,INSERT,UPDATE,DELETE ON "RENT_REG" TO "R_Administrator";
23. GRANT SELECT,INSERT,UPDATE,DELETE ON "RENT_FEE" TO "R_Administrator";

```

8、创建用户

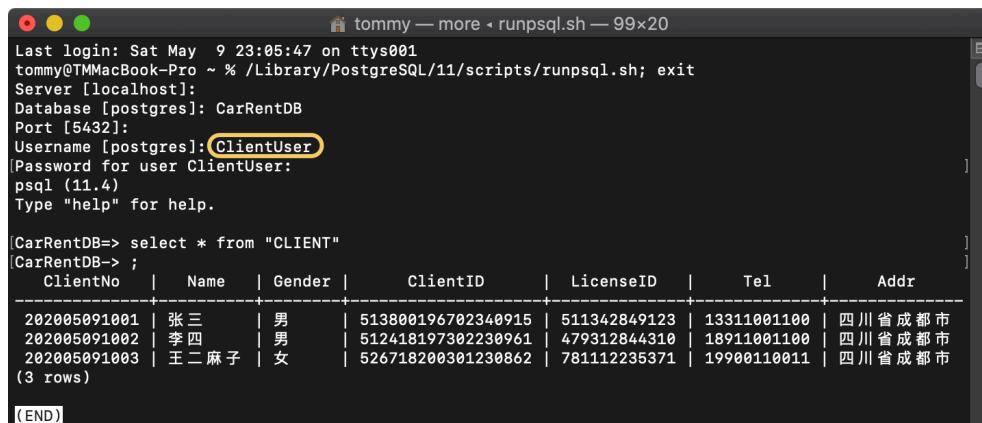
```

1. CREATE USER "ClientUser" WITH
2.     LOGIN
3.     NOSUPERUSER
4.     NOCREATEDB
5.     NOCREATEROLE
6.     INHERIT
7.     NOREPLICATION
8.     CONNECTION LIMIT -1
9.     IN ROLE "R_Client"
10.    PASSWORD '123456';
11.
12. CREATE USER "SalesManUser" WITH
13.     LOGIN
14.     NOSUPERUSER
15.     NOCREATEDB
16.     NOCREATEROLE
17.     INHERIT
18.     NOREPLICATION
19.     CONNECTION LIMIT -1
20.     IN ROLE "R_SalesMan"
21.    PASSWORD '123456';
22.
23. CREATE USER "ManagerUser" WITH
24.     LOGIN
25.     NOSUPERUSER
26.     CREATEDB
27.     CREATEROLE
28.     INHERIT
29.     REPLICATION
30.     CONNECTION LIMIT -1
31.     IN ROLE "R_Manager"
32.    PASSWORD '123456';
33.
34. CREATE USER "AdministratorUser" WITH
35.     LOGIN
36.     SUPERUSER
37.     CREATEDB
38.     CREATEROLE
39.     INHERIT
40.     REPLICATION
41.     CONNECTION LIMIT -1
42.     IN ROLE "R_Administrator"
43.    PASSWORD '123456';

```

9、使用不同角色登录数据库并尝试数据操作

(1) 以 ClientUser 登录并查询 CLIENT 表所有信息



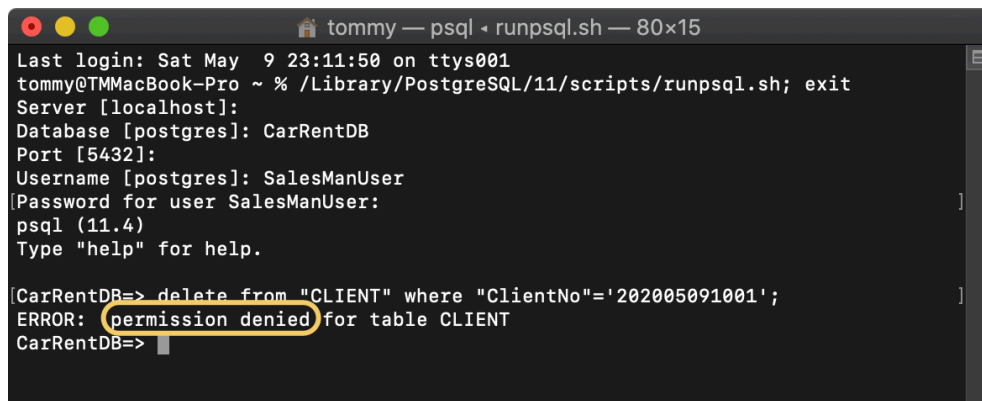
```
tommy — more - runpsql.sh — 99x20
Last login: Sat May 9 23:05:47 on ttys001
tommy@TMMacBook-Pro ~ % /Library/PostgreSQL/11/scripts/runpsql.sh; exit
Server [localhost]:
Database [postgres]: CarRentDB
Port [5432]:
Username [postgres]: ClientUser
[Password for user ClientUser:
psql (11.4)
Type "help" for help.

[CarRentDB=> select * from "CLIENT"
[CarRentDB=> ;
 ClientNo | Name | Gender | ClientID | LicenseID | Tel | Addr
-----+-----+-----+-----+-----+-----+-----
 202005091001 | 张三 | 男 | 513800196702340915 | 511342849123 | 13311001100 | 四川省成都市
 202005091002 | 李四 | 男 | 512418197302230961 | 479312844310 | 18911001100 | 四川省成都市
 202005091003 | 王二麻子 | 女 | 526718200301230862 | 781112235371 | 19900110011 | 四川省成都市
(3 rows)

(END)
```

图7 ClientUser 查询表结果：成功

(2) 以 SalesManUser 登录并删除 CLIENT 表信息

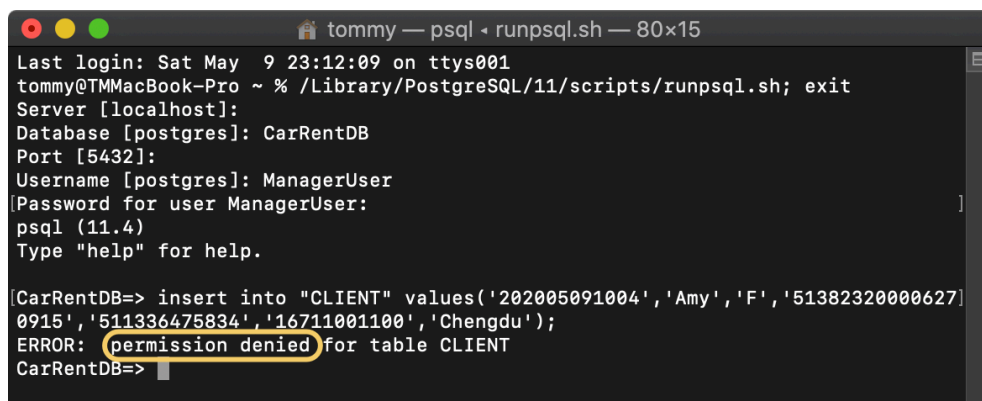


```
tommy — psql - runpsql.sh — 80x15
Last login: Sat May 9 23:11:50 on ttys001
tommy@TMMacBook-Pro ~ % /Library/PostgreSQL/11/scripts/runpsql.sh; exit
Server [localhost]:
Database [postgres]: CarRentDB
Port [5432]:
Username [postgres]: SalesManUser
[Password for user SalesManUser:
psql (11.4)
Type "help" for help.

[CarRentDB=> delete from "CLIENT" where "ClientNo"='202005091001';
ERROR: permission denied for table CLIENT
CarRentDB=>
```

图8 SalesManUser 删除数据结果：被拒绝

(3) 以 ManagerUser 登录并插入 CLIENT 表信息



```
tommy — psql - runpsql.sh — 80x15
Last login: Sat May 9 23:12:09 on ttys001
tommy@TMMacBook-Pro ~ % /Library/PostgreSQL/11/scripts/runpsql.sh; exit
Server [localhost]:
Database [postgres]: CarRentDB
Port [5432]:
Username [postgres]: ManagerUser
[Password for user ManagerUser:
psql (11.4)
Type "help" for help.

[CarRentDB=> insert into "CLIENT" values('202005091004','Amy','F','51382320000627'
0915','511336475834','16711001100','Chengdu');
ERROR: permission denied for table CLIENT
CarRentDB=>
```

图9 ManagerUser 插入数据结果：被拒绝

(4) 以 AdministratorUser 登录并更新 CLIENT 表信息

```
tommy — more - runpsql.sh — 107x22
Last login: Sat May 9 23:13:28 on ttys001
tommy@TMMacBook-Pro ~ % /Library/PostgreSQL/11/scripts/runpsql.sh; exit
Server [localhost]:
Database [postgres]: CarRentDB
Port [5432]:
Username [postgres]: AdministratorUser
Password for user AdministratorUser:
psql (11.4)
Type "help" for help.

CarRentDB=# update "CLIENT" set "Tel"='11011011011' where "ClientNo"='202005091001';
UPDATE 1
CarRentDB=# ;
CarRentDB=# select * from "CLIENT";
 ClientNo | Name | Gender | ClientID | LicenseID | Tel | Addr
-----+-----+-----+-----+-----+-----+-----
 202005091002 | 李四 | 男 | 512418197302230961 | 479312844310 | 18911001100 | 四川省成都市
 202005091003 | 王二麻子 | 女 | 526718200301230862 | 781112235371 | 19900110011 | 四川省成都市
 202005091001 | 张三 | 男 | 513800196702340915 | 511342849123 | 11011011011 | 四川省成都市
(3 rows)
```

图 10 AdministratorUser 插入数据结果：成功

10、备份数据库

Backup (数据库: CarRentDB)

通常 转储选项

文件名

/Users/tommy/Desktop/Backup/CarRentDB.bak

格式

Custom

压缩率

编码

UTF8

任务数目

角色名称

R_Administrator

取消

备份

图 11 使用 pgAdmin 的 GUI 界面备份数据库

Backup (数据库: CarRentDB)

通常 转储选项

Pre-Data

否

数据

否

Post-Data

否

对象类型

只有数据

否

只有schema

是

Blobs

是

取消

备份

图 12 可选择“只有数据”或“只有 schema”

名称	修改日期	大小	种类
CarRentDB-data.bak	今天 23:33	3 KB	文稿
CarRentDB-schema.bak	今天 23:32	7 KB	文稿
CarRentDB.bak	今天 23:31	9 KB	文稿

图 13 数据库、schema、数据库表三个备份文件

11、从备份文件恢复数据库

Restore (数据库: CarRentDB)

通常
还原选项

格式

Custom or tar

文件名

/Users/tommy/Desktop/Backup/CarRentDB.bak

任务数目

角色名称

R_Administrator

取消

还原

图 14 使用 pgAdmin 的 GUI 界面还原数据库

Restoring backup on the server

Restoring backup on the server 'PostgreSQL 11 (localhost:5432)'
Sat May 09 2020 23:37:57 GMT+0800 (中国标准时间)

0.73 秒

More details...

Stop Process

成功完成

图 15 数据库还原成功