

The database '**RentalCars**' contains the following 6 tables. Note that the **primary keys** are underlined and **foreign keys** are in italics followed by the symbol '*'.

Table: Clients				
<u>ID</u>	FirstName	LastName	Address	Telephone
1	Mary	Marely	Oxford	343-1232
2	Lisa	Brown	Los Angeles	650-9875
3	Jackie	Jackson	New York	650-9875
4	Charles	Macfee	San Diego	830-6532
5	Mike	Neil	Los Angeles	670-3467

Table: Cars				
<u>Number</u>	Brand*	Model*	Color	PricePerDay
100	1	10	Black	50
101	2	11	Silver	90
102	3	12	Red	70
103	1	13	Silver	100

Table: Renting				
<u>Client*</u>	<u>Car*</u>	<u>RentalDate</u>	ReturnDate	Price
1	101	14/03/2015	20/03/2015	540
4	102	10/02/2015	28/02/2015	1260
1	100	13/04/2015	20/04/2015	350
2	100	25/04/2015	30/04/2015	250
5	103	10/05/2015		

Table: Accident			
Client*	Car*	DateAccident	RepairPrice
4	102	27/02/2015	75
1	100	18/04/2015	115
1	101	17/03/2015	60

Table: Model	
<u>ID</u>	Description
10	C-Class Sedan
11	BMW X6
12	Land Cruiser Prado
13	GLA SUV

Table: Brand	
<u>ID</u>	Description
1	Mercedes
2	BMW
3	Toyota


```

Create database rentalcars
use rentalcars
create table clients (id int primary key not null identity,
firstname varchar(20) not null,
lastname varchar(20) not null,
address_ varchar(20) not null,
telephone varchar(20) not null
)

insert into clients values('Mary','Marely','Oxford','343-1232')
insert into clients values('Lisa','Brown','Los Angeles','343-1232')
insert into clients values('Jackie','Jackson','New York','343-1232')
insert into clients values('Charles','Macfee','San Diago','343-1232')
insert into clients values('Mike','Neil','Los Angeles','343-1232')

create table brand (id int primary key not null identity, descr varchar(20) not null)
insert into brand values('Mercedes')
insert into brand values('BMW')
insert into brand values('Toyota')

create table model (id int primary key not null, descr varchar(20) not null)
insert into model values(10,'C-Class Sedan')
insert into model values(11,'BMW X6')
insert into model values(12,'Land Cruiser Prado')
insert into model values(13,'GLA SUV')

create table cars (number int primary key not null ,
brand int not null foreign key references brand(id),
model int not null foreign key references model(id),
color varchar(20) not null,
priceperday float not null)
insert into cars values(100,1,10,'Black',50)
insert into cars values(101,2,11,'Silver',90)
insert into cars values(102,3,12,'Red',70)
insert into cars values(103,1,13,'Silver',100)
insert into cars values(104,2,11,'While',80)

create table renting (client int not null foreign key references clients (id),
car int not null foreign key references cars (number),
rentaldate date not null,
returndate date,
price float not null)
insert into renting values (1,101,'03-14-2015','03/20/2015',540)
insert into renting values (4,102,'02-10-2015','03/20/2015',1260)
insert into renting values (1,100,'04-13-2015','03/20/2015',350)
insert into renting values (2,100,'04-25-2015','03/20/2015',250)
insert into renting values (5,103,'05-10-2015',NULL,0)

```

Note: you have to submit your answers in one text (.txt) or script (.sql) file.

- 1- Write the necessary SQL Data Definition statements to create the table **Accident** as shown in (page1), where the **RepairPrice** must be ≥ 60 (2 pts).

```

create table accident (
client int not null foreign key references clients (id),
car int not null foreign key references cars(number),
dateaccident date,

```

```

repairprice float check (repairprice >=60)
)
insert into accident values (4,102,'02-27-2015',75)
insert into accident values (1,100,'04-18-2015',115)
insert into accident values (1,101,'03-17-2015',60)

```

- 2- Display all accidents of the car that having a price per day greater than 50. (2 pts)

```

select a.* from dbo.cars a, dbo.accident b where a.number=b.car and a.priceperday>50

```

- 3- Write a function that takes the **Brand name** as a parameter and finds the number of available cars for renting. (2 pts)

```

create function AvailableCars (@brand varchar(20)) returns int as
begin
declare @num int
select @num=count(number)
from cars
where brand = (select id from brand where descr=@brand) and number not in (select
car from renting)
return @num
end
select dbo.AvailableCars ('BMW')

```

- 4- Write a function that takes the client's first and last names as parameters and returns the ID of the client (1 pt)

```

create function findClientID (@fnm varchar(20),@lnm varchar(20)) returns int as
begin
declare @cid int
select @cid=id from clients where firstname=@fnm and lastname=@lnm
return @cid
end

select dbo.findClientID ('Mary','Marely')

```

- 5- Write a procedure that takes as a parameter the client's full name, calls the function written in Q4) and then displays all the rented cars (the RentalDate, ReturnDate and the Price) of that client. (3 pts)

```

create procedure clientrentedcars @fnm varchar(20),@lnm varchar(20) as
begin
declare @cid int
set @cid= dbo.findClientID (@fnm,@lnm)
select * from dbo.renting where client=@cid
end

exec dbo.clientrentedcars 'Mary','Marely'

```

- 6- Given the Renting_History table (**ClientID, CarNumber, RentalDate, ReturnDate, Price, CurrDate**). Write a trigger that adds a record to the table Accident_History after deleting a record from the table Renting. The date attribute of Renting_History table shows the current system date. (3 pts)

Note: write the necessary code to create the table **Renting_history**. (1 pts)

```

create table renting_history (client int not null,
car int not null,
rentaldate date not null,
returndate date,
price float not null,
currdate date default getdate())

CREATE TRIGGER trgrRentDlt ON dbo.renting
After DELETE

```

```

AS
BEGIN
    INSERT INTO dbo.renting_history
    (client,car,rentaldate,returndate,price,currDate)
    SELECT client, car, rentaldate,returndate,price, GETDATE() FROM
    deleted
END

```

- 7- Create a view called MercedesCars from the Cars table. This view shows all Models of cars 'Mercedes'. It contains the attributes Number, Model and Color. (2 pts)

```

create view MercedesCars as
select c.* from cars c, brand b where b.id=c.brand
and b.id = (select id from brand where descr = 'Mercedes')

```

- 8- Write a T-SQL script that determines all the Car models for the brand having 1 as number (Mercedes) (see example 2). If there are no cars to show for this brand, then a message will be displayed (see example 1). (4 pts)

Example of execution	
Example 1 : Brand ID 5	There are no cars to show for this brand.
Example 2 : Brand ID 1	Brand Number 1: Number Model Color PricePerDay Number Model Color PricePerDay -----
	The number of cars to show for this brand is: X
	Sample Output report::
	Brand Number 1: 10 C-Class Sedan Burger Black 50 13 GLA SUV Burger Silver 100 -----
	The number of cars to show for this brand is: 2

```

use rentalcars
declare @nm varchar(20), @mdl varchar(20),@clr varchar(15), @ppd float,@c int
declare c1 CURSOR
for
select c.number,m.descr,c.color,c.priceperday
from cars c,model m
Where c.model=m.id and brand=1
set @c = 0
open c1
fetch next from c1 into @nm,@mdl,@clr,@ppd
if (@@ROWCOUNT = 1)
begin
print 'Brand Number 1:'
end
while (@@FETCH_STATUS = 0)
Begin
print @nm+ ' ' + @mdl+ ' ' + @clr+ ' ' + str(ltrim(@ppd))
fetch next from c1 into @nm, @mdl,@clr,@ppd
set @c = @c+1
end
if (@c=0)
print 'There are no Cars to show for this brand'
else
begin
print '-----'
print 'The number of cars to show for this brand is: ' + ltrim(str(@c))
end
Close c1
deallocate c1

```

