IT19127088

Senarathna A.M.M.D.N

Taxi service

Relational schema

* Account (Account\_ID, F\_name , L\_name, gender , password)
* Account\_phone (Account\_ID , phone)
* **Foreign key (Account\_ID) , references Account (Account\_ID)**
* Account\_Email(Account\_ID , Email)
* **Foreign key (Account\_ID) , references Account (Account\_ID)**
* User (User\_ID , Location , Account\_ID)
* **Foreign key (Account\_ID) , references Account (Account\_ID)**
* Ride(Ride\_ID , Location , Date , Distination , User\_ID)
* **Foreign key (User\_ID) , references User (User\_ID)**
* Taxi (Taxi\_ID , vehicle\_NO , Mode , Ride\_id)
* **Foreign key (Ride\_ID) , references Ride (Ride\_ID)**
* TukTuk(Taxi\_ID , tuk\_number, vehicle\_NO , Mode)
* Cabs (Taxi\_ID , cab\_number, vehicle\_NO , Mode)
* Driver(Driver\_ID , rate , F\_name , L\_name , Taxi\_ID)
* **Foreign key (Taxi\_ID) , references Taxi (Taxi\_ID)**
* Driver\_email(Driver\_ID , Driver\_Email)
* **Foreign key (Driver\_ID) , references Driver (Driver\_ID)**
* Payment(payment\_ID , date , Account , method , User\_ID)
* **Foreign key (User\_ID) , references User (User\_ID)**
* Feedback (feedback\_ID , feedbackMSG)
* **Foreign key (User\_ID) , references user (User\_ID)**

**Assignment 02 / part 01**

Query 01)

a) we can check for pattern of data and check for all null values.I used this query to retrieve the data from the payment to get the payment\_ID , amount and date.

b)select payment\_ID , date , Amount

from payment

where Amount BETWEEN 500 AND 1000

Query 02)

1. Used to check whether attribute values matches any value within a value list. I used this query to retrieve data from payment table but in here there is a special function. That is the user\_ID is present in another table which fulfills WHERE condition (where method = 'cash’) assigned destination for the User\_ID will retrieve.

b)select u.user\_ID , u.Location

from user u

where u.user\_ID IN ( select user\_ID

from payment

where method = 'cash' )

Query 03)

1. his query is used for summarizes the results of an expression over a number of rows, returning a single value and Groups the data in tables and produces a single summary row for each group and also Used to apply conditions on the groupings .

I used the quary to retrieve the amount from the Driver table, I used aggregate function called SUM. From the SUM((rate) it get the sum of rates and it is group by the Driver\_ID and also using having clause. It count one driver rates.

b) select d.Driver\_ID , SUM(d.rate) AS rate

from Driver d , Taxi t

where d.Driver\_ID , d.Taxi\_ID

Group By d.Driver\_ID

Having SUM(d.rate) > 10

Part 02

c)

01) View is a virtual table that is derived from base tables (main tables) , view also have rows and columns as they are in a real table in the database. In a view there will be all the rows in a certain table or sometimes it will have the specific rows based on some circumstances.

In my system I used this view to extract certain data from the base tables and place that data in a view. In here displays the f\_name , phone , location and payment method.

02) create view user\_details(F\_name , phone , location , method) as

select u.F\_name , a.phone , p.method , r.location

from user u , payment p, Account a , Ride r

where a.Account\_ID = u.Account\_ID AND u.user\_ID = r.user\_ID AND u.user\_ID = p.user\_ID

select \* from user\_details

d)

01) This function will get the output of a number of drivers where I pass the num\_rate as he parameter for this function. If we want to calculate the drivers where a particular rating , this function can be used.

02) create function NumOfDrivers(@num\_rate float ) returns int

as

begin

declare @c int

select @c = count(\*)

from Driver

where rate = @num\_rate

return @c

end

declare @MD int

exec @MD = NumOfDrivers 7

print @MD

e)

01) This rigger used to check check payment is success or unsuccess, this trigger ensures that the users should payment after the journey. If user make payment correctly there will be a message will display 'payment success'.

02) create trigger checkpay

on payment

as

begin

declare @paymentid int

declare @pass int

declare @amount int

select @amount = Ammount from payment

if (@paymentid = @pass)

begin

print 'payment success'

rollback transaction

end

end