1. a) create table scooter

( scooterIdent smallint not null,

flag tinyint,

homeLoc varchar(50),

PRIMARY KEY (scooterIdent)

);

create table user

( userIdent mediumint not null,

creditCard varchar(20),

expDate DATETIME,

emailId varchar(254),

PRIMARY KEY(UserIdent)

);

b) create table trip

( tripIdent smallint not null,

scooterIdent mediumint not null,

userIdent integer not null,

tripStartLocLat integer,

tripStartLocLon integer,

tripEndLocLat integer,

tripEndLocLon integer,

tripStartTime DATETIME,

tripEndTime DATETIME,

PRIMARY KEY(tripIdent),

FOREIGN KEY(scooterIdent) REFERENCES scooter(scooterIdent),

FOREIGN KEY(userIdent) REFERENCES user(userIdent)

);

Since the number of minutes can be calculated from start time and end time, it does not need to be included. Similarly, since the cost can easily be calculated based on the flat formula, there is no need to include it in the table since it takes up memory.

c) Advantages of caching: Less data to transmit

Disadvantages of caching: If users' phones' die there will be no record of the trip and recovery can potentially be complicated

Advantages of sending start and end information: If phone dies or Bird goes missing, there is a record of the trip starting and the user involved in the database

Disadvantages of above: double transmission of information

As an employee, I would choose to send messages at both start and end, since it is effective and safe.

d)

2. a) select extract(HOUR from DateTime) as hour, sum(throughput) as trips

from rides2017

group by extract(HOUR from DateTime);

b) select Origin, Destination from

(select Origin, Destination, sum(Throughput) as Throughput

from rides2017

where DAYOFWEEK(DateTime) != 1 and DAYOFWEEK(DateTime) != 7

group by Origin, Destination

order by Throughput desc

limit 1) a;

c) select Destination, avg(Throughput) as tp

from rides2017

where WEEKDAY(DateTime) = 2 and (HOUR(DateTime) >= 7 and HOUR(DateTime) <= 10)

group by Destination

order by tp desc

limit 5;

d) select Origin

from rides2017

group by Origin

having max(Throughput) > (10\*avg(Throughput));

e) (πhour, trips/100 (σ((hour>=7 Ʌ hour<10) V (hour>=17 Ʌ hour<19)) (hourly\_ridership))

f) πRiders(σStation=Name(ρName(Station)(σCondition="sunny" V Condition="rainy"(Weather)))(Station))