Andrew Ding

504748356

a) CREATE TABLE Scooter(

id INT PRIMARY KEY, //also a foreign key

flag TINYINT NOT NULL, //can have ints represent the flags

home TINYINT NOT NULL,//have each int represent a home, easy to increase range of homes

CHECK(flag >=0 AND flag <=2 AND home >=0 AND home <=2))

CREATE TABLE User(

id INT PRIMARY KEY, //also a foreign key

credit_card VARCHAR(16),

expiration DATE,

email varchar(100) NOT NULL,

CHECK((credit_card IS NOT NULL AND expiration IS NOT NULL)OR(credit_card IS NULL AND expiration IS NULL)))

b) CREATE TABLE Trip(

id INT PRIMARY KEY, //primary key

uid INT, //foreign key

sid INT, //foreign key

start_location REAL NOT NULL,

end_location REAL,

start_time DATETIME NOT NULL,

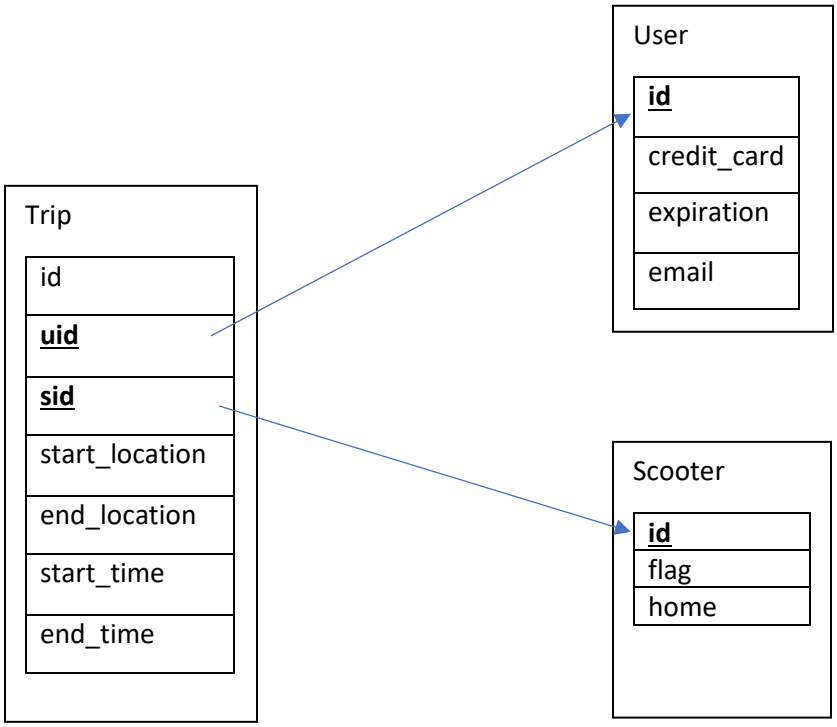
end_time DATETIME,

FOREIGN KEY(uid) REFERENCES User(id),

FOREIGN KEY(sid) REFERENCES Scooter(id))

c) If we cache the ride data, it will help reduce the load for the queries to the database. However, other apps will not have the updated info if they query the database and may get incorrect results. This may happen when a user activates a Bird but another user sees the Bird as offline even though it is online.

I would modify the row on activation and park since accurate data is more important that reducing the load. A better way would to be to use a callback. When we want to know if a Bird is active, we can request the Bird its status. If it is being used, then we would reply with its correct status. Thus, we maintain the accuracy of the Bird's status while also reducing the number of queries to the database.



a) SELECT HOUR(Datetime) as hour, SUM(Throughput) as trips FROM rides2017 GROUP BY hour;
b) SELECT Origin, Destination, SUM(Throughput) FROM rides2017 WHERE(WEEKDAY(Datetime) <= 4) GROUP BY Origin, Desination ORDER BY SUM(Throughput) DESC LIMIT 1;
c) SELECT Destination, AVG(Throughput) FROM rides2017 WHERE(HOUR(DateTime) BETWEEN 7 AND 10 AND WEEKDAY(DateTime) = 0) GROUP BY (Destination) ORDER BY AVG(Throughput) DESC LIMIT 5;
d) SELECT Origin FROM rides2017 GROUP BY Origin Having(MAX(Throughput) > 100*AVG(Throughput));
e) $\Pi_{\text{hour, trips/100}}(\sigma_{(\text{hour} >= 7 \wedge \text{hour} < 10) \vee (\text{hour} >= 17 \wedge \text{hour} < 19)}$ (hourly_rideship))
f) $\Pi_{\text{station, DateTime, Temperature, Condition}}(\sigma_{\text{condition} = \text{“sunny”} \vee \text{condition} = \text{“rainy”}}$ (Occupancy $\bowtie$ Weather))