Database Systems

Spring 2023

Problem Set 1

Goals:

- Nested Queries or Sub Queries

Reading Content:

- Chapter 4 (T-SQL Fundamentals)

Note: All questions should be done using subquery, From Clause should have only one relational instance

Activities:

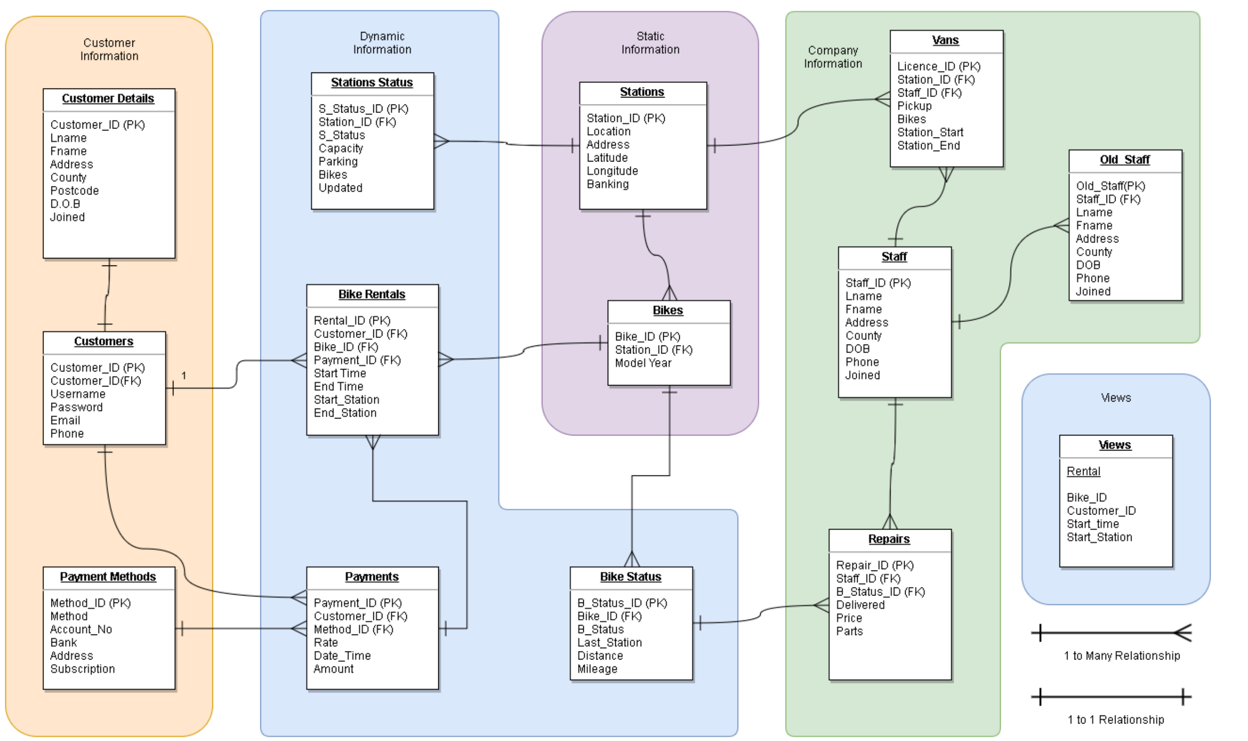
Database BikeShare:

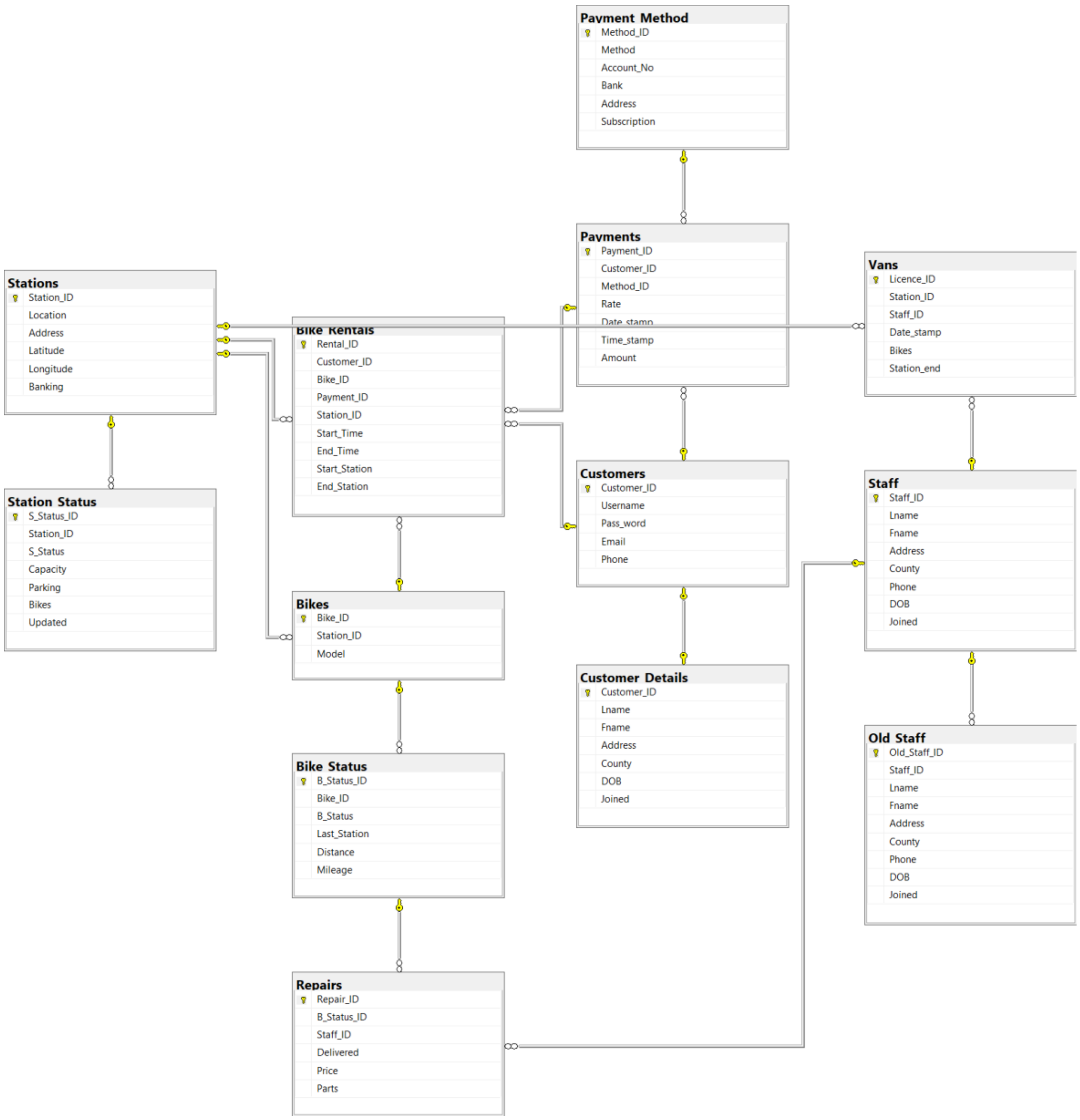
(Reference: https://github.com/codefromjames/DataBase-DataMart)

The database was based on a model of Bike Share. The model assumes that bike rotation, repairs and payments are all connected on a singular system. The isolated tables, payment methods and customer details were created first without the requirement for foreign keys. A one-to-one relationship was created between customer details and customers. The only way I found in MySql was using the primary key as the foreign key to create this relationship. The three tables represented all the customer's information which was required for subscribing to the service. The next table created was payments which used the two new primary keys as foreign keys. This was because customers will have many payments and many different methods will be used to pay. The next two tables created were stations and staff as they were independent from surrounding entities not requiring a foreign key in creation from a non-existing table. Once station's primary key was available it was used as the foreign key in station status and bikes tables as stations will have many states and bike rotation respectively. Bikes' primary key was used as a foreign key in bike status. A trigger was also created in bikes to upgrade bike status information automatically once a bike was activated. This was the most difficult element to incorporate. The stations and bikes information was grouped as static information as they will be relatively fixed attributes. The company information was expanded upon next using the staff and stations foreign keys in vans table as staff members will use many vans and stations will be rotated by several vans. The primary key for vans was based on the license plate present. Repairs used the foreign keys of staff and bike status as staff members will engage in several repairs due to several different individual bike states overtime. Any updates to staff will be recorded as a trigger to the old staff table. The relationship was one to many as one staff member could have several amendments such as changes of address. A procedure (GetOld_Staff) was created to verify that an update to staff would create an entry in old staff. This completed all the customer information table. The final section was dynamic information completed by the introduction of the bike rentals table. The table has three one to many connects. The customer will have many bike rentals requiring many payments and using many different bikes overtime. The cyclical structure between customers, payments and bike rentals requires data recorded to be unilaterally cross-referenceable (customer's events=payments=bike rentals) due to shared foreign keys and the unique event recorded by repeatable customers. This was verified using a view (rental) which looked at the results from the bike rentals table. The customer (10) made two successive rentals using two different payments for two different bikes using two different payment transactions. The database could be improved upon using more triggers to updates status tables and calculating geometry between journey from stations. Once a bike reaches a certain mileage, it would active a bike status which requires a mechanic to repair elements from the bike. The database can currently be used to assess which staff member has worked the most or which customer uses the service the longest. The database was complicated to construct but the organisation of the entity-relationship schema aided in providing context to each tables connectivity and functionality.

<u>Tasks:</u>

Answer these queries in the context of **Bike** schema. You can understand the database using following diagrams.





<u>Part 1:</u>

1. Give the list of all stations (Station Name) from where the rides were taken in year 2004.
2. Give the total repairing cost of each bike. Schema should be like this. (Bike Id, Repairing Cost)
3. Bikes of which station needed most repairing. Give the station name.
4. How many bikes are owned by each station? Schema should be (StationName, TotalBikes)
5. Given the name customers who never rented a bike. Schema is as follow. (CustomerFullName)
6. Give the bike ids of those bikes who were renter after year 2016.
7. Identify the customers who always pay using mastercard. Give the full name of customers.
8. For which station (Station Name) the most bikes are moved using vans in year 2015.
9. Give the average cost of repairing that was spent on each bike. Schema includes (BikeId, AverageCost)

10. Give the BikeIds which were repaired in at least 3 year.

Part 2

Answer these queries in the context of **Northwind schema** and solve them using subqueries only.

1. Give the names of customers whose orders were delayed. Your answer should have the following schema.
   Customers(CustomerId, CustomerName)
2. Give the products details with its supplier company.
   Products(ProductName, SupplierName)
3. Give the name of top products which have highest sale in the year 1998.
4. Give the name of employees with its manager name. Schema should have the following schema. (EmployeeName, ManagerName)
5. Give the full names of managers who have less than two employees.
6. List all the products whose price is more than average price.
7. Find second highest priced product without using TOP statement
8. Are there any employees who are elder than their managers? List that names of those employees. Schema should look like this
   Employees(EmployeeName,ManagerName,EmployeeAge,ManagerAge)
9. List the names of products which were ordered on 8th August 1997.
10. List the names of suppliers whose supplied products were ordered in 1997.
11. How many employees are assigned to Eastern region. Give count.
12. Give the name of products which were not ordered in 1996.