Name: Rahul Subhash Gouda

StudentID: S4063983

DATABASE CONCEPTS

Assessment 1

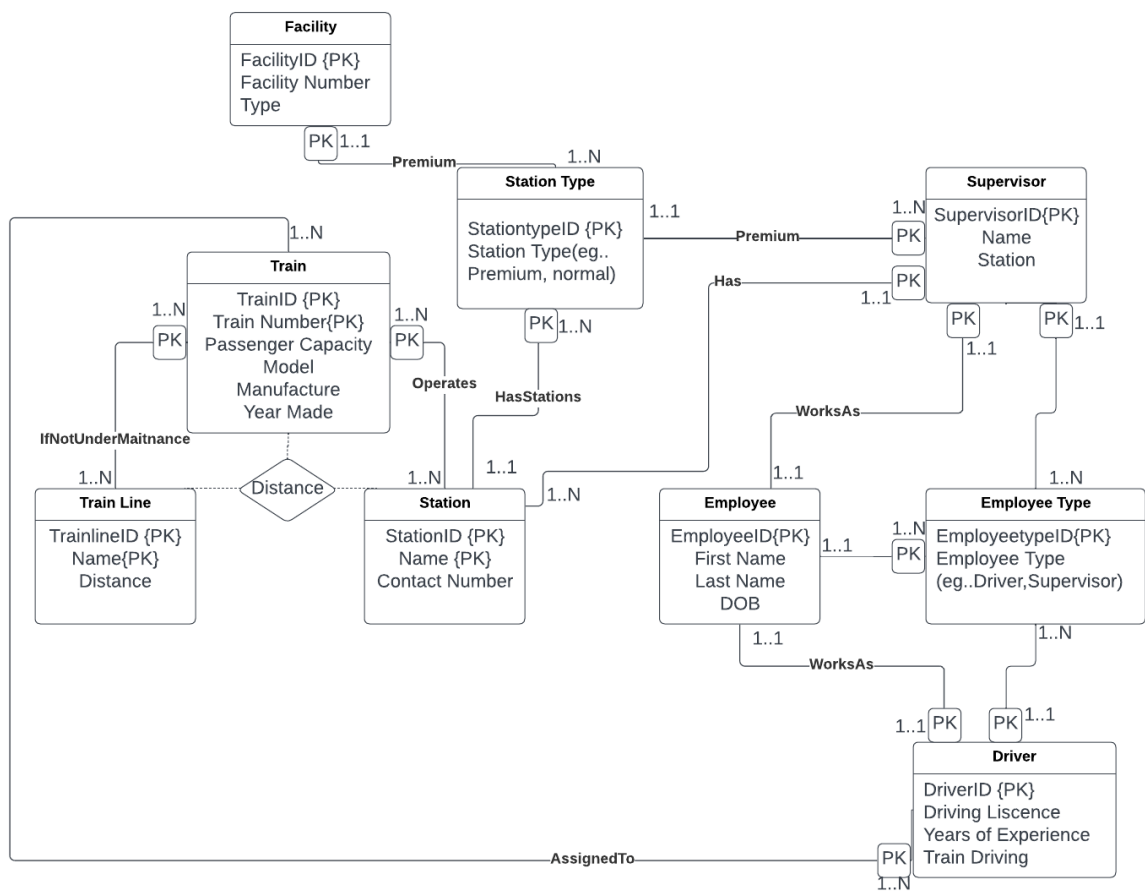## Task 1: - Designing an Entity Relationship Model (ER Model)

Figure:- 1

Assumptions: -

Train to Station Connection:
- Each train is connected to one or more train lines.
- Each train line is connected to one or more distances.
- The train can over on the train line only if the train line is not under maintenance.
- Each station is connected to one or more distances.
- Each distance is associated with one or more train lines and stations.

Station Details:

- Each station has a type, which is connected to a supervisor.
- Station type could have attributes such as 'Premium' and 'Facility', both possibly indicating premium services.
- There could be different types of stations based on their services or facilities offered.

Supervisor Details:

- A supervisor oversees one or more stations.
- Each supervisor is associated with a station type.
- A supervisor could be part of the employees.
- Each station can have one or more supervisors

Employee Details:

- Each employee has a type, which could indicate their role or position.
- Employees might have different roles or responsibilities.
- An employee might also be a driver, implying they are involved in operating a train.

Driver and Train Relationship:

- A driver operates a train.
- Each driver is associated with a train they operate.
- A driver is an employee, so they would have all attributes and relationships associated with an employee.

Employee Type:

- There are different types of employees, each with specific roles or responsibilities.
- An employee type could determine the kind of work an employee does.
- An employee type is associated with supervisors and drivers.
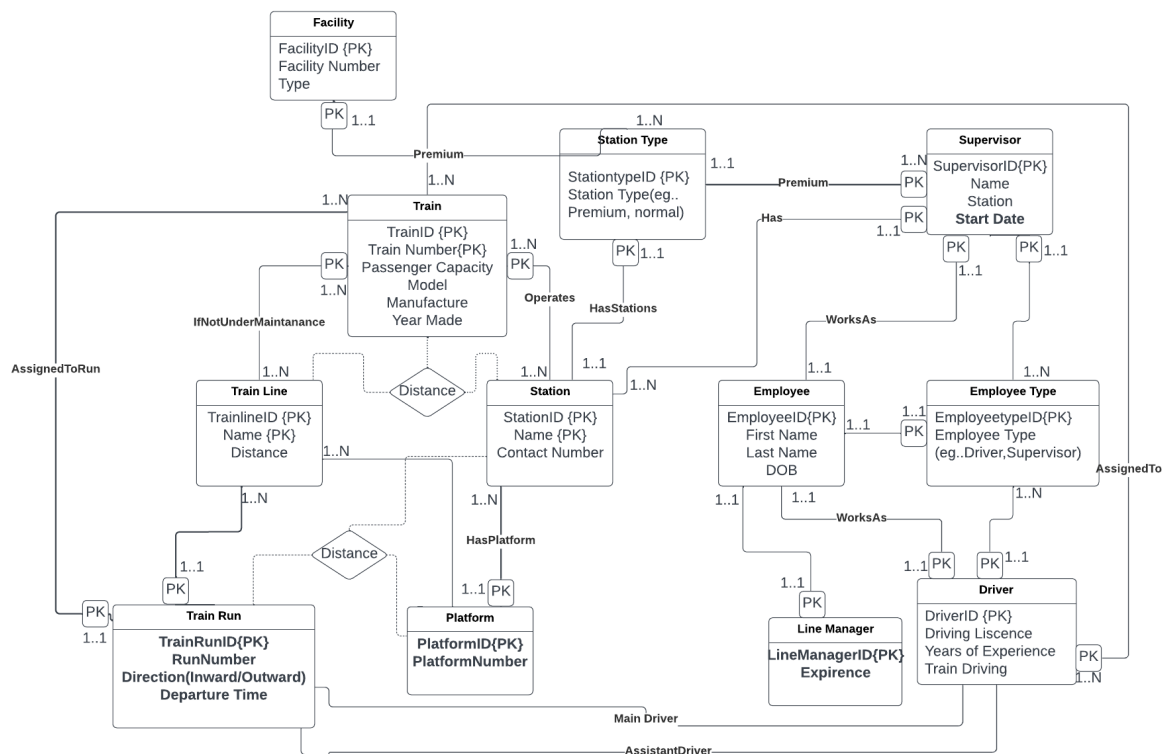
Task 2: - Make Additional adjustments on Task 1.

Figure:-2

Note: - In figure 2 all the additional changes are made in bold letters

Assumptions: -

Train Run Details:

- Each train run is associated with a train line.
- A train run is connected to a train it operates.
- There are two connections to drivers for each train run: one for the main driver and one for the assistant driver.
- There is a classification associated with each train run denoting the distance between each train run.
- The classification of the train run could indicate factors such as the length of the journey or the type of service provided.

Platform Details:

- Each platform is associated with a station.
- Platforms are also associated with train runs, indicating which train runs stop at each platform.
- The platform is connected to the station, possibly indicating its physical location within the station premises.
- There is a connection between the platform and the train line, which suggests that platforms are linked to specific train lines.

Line Manager Details:
- A line manager is associated with employees.
- Line managers likely oversee the work of the employees within the organization.
- The relationship between line managers and employees indicates a hierarchical structure where line managers supervise and manage the employees

Task 3: - ER Model to a Relational Database Schema
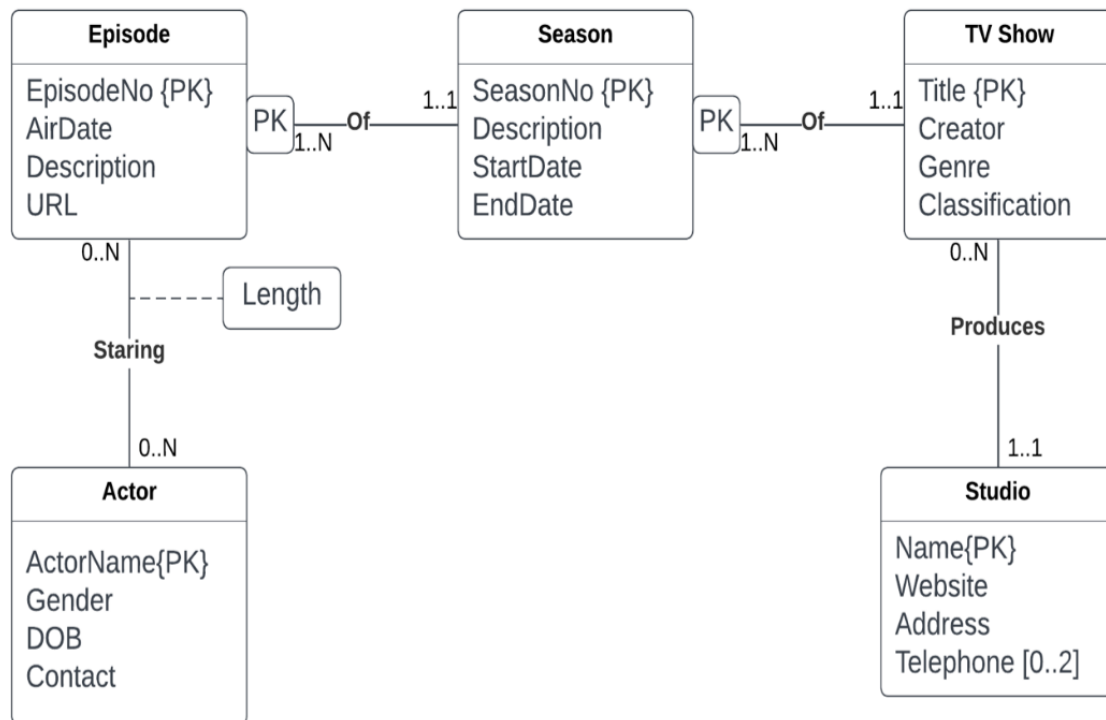
Step1: Map Strong Entity
1) TV Show (<u>Title,</u> creator, genre, Classification)
2) Studio (<u>Name,</u> Website, Address, Telephone)
3) Actor (<u>ActorName,</u> Gender, DOB, Contact)

Step2: Map Weak Entities
1) Episode (<u>EpisodeNo,</u> AirDate, Description, URL, EpisodeNo*)
2) Season (<u>SeasonNo,</u> Description, StartDate, EndDate, Title*)

Step3: Map 1:1 Relationships
1) There are no one is to one relationship.

Step4: Map 1: Many Relationship
1) TV Show (<u>Title</u>, Creator, Genre, Classification, Name*)
2) Season (<u>SeasonNo</u>, Description, StartDate, EndDate, Title*)
3) Episode (<u>EpisodeNo</u>, AirDate, Description, URL, ActorName*)

Step5: Map Many: Many Relationships
1) Length (ActorName*, EpisodeNo*, Length)

Step6: Map All the Multi-valued Attributes
1) StudioPhone (Name*, <u>Telephone)</u>

Step7: Map Higher Degree Attributes
1) There are no Higher Degree Attributes or Ternary Relationship

Final Data Base Schema

1) TV Show (<u>Title</u>, Creator, Genre, Classification, Name*)
2) Studio (<u>Name,</u> Website, Address, Telephone)
3) Season (<u>SeasonNo</u>, Description, StartDate, EndDate, Title*)
4) Episode (<u>EpisodeNo</u>, AirDate, Description, URL, ActorName*)
5) Length (ActorName*, EpisodeNo*, Length)
6) Actor (<u>ActorName,</u> Gender, DOB, Contact)
7) StudioPhone (Name*, <u>Telephone)</u>

Task 4: - Relational Database Model

Question 4.1: -
Ans-
The Foreign keys listed in the database schema are as follows:-

Employees.department_id ——> Departments.department_id
Departments.location_id ——> Locations.location_id
Locations.country_id ——> Countries.country_id
JobHistory.employee_id——> employees.employee_id
JobHistory.job_id ——> Jobs.job_id
JobHistory.department_id ——> Departments.department_id

Question 4.2: -
Ans - In the database schema, both the start date and end date serve as primary keys, necessitating unique entries. Consequently, if an employee intends to work in two different departments, this scenario is not feasible due to the recording of start and end dates.

Question 4.3: -
Ans: - In the database schema, the table "job history" includes the "department_id" as a foreign key, while the start and end dates must be unique. This design allows for employees to work in different departments at distinct time periods, ensuring flexibility in their employment across various roles.

Question 4.4: -
Ans: - The current schema does not facilitate different locations for the same department since the "department_id" serves as a primary key and must remain unique. Introducing multiple location IDs for the same department ID would violate this uniqueness constraint. To address this, a new table, "departmentLocation," can be created. This table would include foreign keys for both location ID and department

ID, allowing for multiple entries and enabling the representation of different locations for the same department.

Question 4.5: -
Ans: - The provided SQL query may encounter issues during execution due to its reliance on the "salary" column for identification, which lacks uniqueness and can potentially affect multiple records. Additionally, the date format specified in the query ('YYYY-MM-DD') does not match the format of the provided date ('DD-MM-YYYY'), increasing the risk of incorrect input errors.

Solution: - UPDATE Employees SET empjob_id = 33, hire_date = '2012-02-10' WHERE employee_id =50;

Question 4.6: -
Ans: - The query will likely fail and throw an error due to foreign key constraints. Since the "Locations" table is referenced by other tables, such as the "Departments" table where "location_id" is used as a foreign key, attempting to delete the location with ID 10 will cause issues with the referencing relationships.

Question 4.7: -
Ans: -
CREATE TABLE JobHistory
(job_history_id INTEGER PRIMARY KEY AUTOINCREMENT,
 employee_id INTEGER NOT NULL
start_date DATE NOT NULL,
 end_date DATE NOT NULL,
 job_id INTEGER NOT NULL,
 department_id INTEGER NOT NULL,
 FOREIGN KEY (employee_id) REFERENCES Employees(employee_id),
FOREIGN KEY (job_id) REFERENCES Jobs(job_id),
FOREIGN KEY (department_id) REFERENCES Departments(department_id));

Question 4.8: -

Ans: - The provided SQL statements won't function correctly due to a syntax error in the date format; it should be "DD-MM-YYYY" instead of "YYYY-MM-DD". Adjusting the query to use the correct date format resolves this issue.

INSERT INTO Employees VALUES (88, 'Scott', 'Wallace', '1111', '01/01/2020', 10,140000, 4);
However, it's essential to execute the query for creating the "Art" department entry in the Departments table before inserting Scott Wallace's record into the Employees table. This ensures that there is a valid department entry to reference when assigning Scott Wallace as the manager of the "Art" department.
INSERT INTO Departments VALUES (4, 'ART', 88, 20);