ECS519U – Database Systems

Coursework 2 – Eurostar Scenario

06-11-2020

Group members (Group 70):
Yasir M Dahir: 190009658, Michael Elia: 180336591 and Omar Hassan 180374892

All group members participated actively and engaged in numerous work sessions between now and when the coursework was assigned. This report is the product of every group member's work.

1.) A list of reasonable assumptions you have made for your design (3 marks)

Assumptions:

- Each train (specifically train journey) is operated by one crew at a time
- We assume that each route can have any number of intermediate stops. (0 or more)
- We assume that trains carry a finite amount of passengers
- For every journey, there is only 1 train crew
- Each train crew consists of 2 drivers, 2 conductors, 5 service members and 2 security guards, as well as management and sales employees
- A conductor who is the supervisor on one crew plays the role of supervisor for every crew that he/she is a member of (In other words, a conductor who is a supervisor is always a supervisor, no matter what crew they are a member of at any time)
- 16 trains are operational and 4 are in service at any given point in time
- PassengerIDs are randomly generated each time a passenger gets a ticket for a train, so the same person can have several different passengerIDs for each different journey
- Employees can be shuffled between crews, I.e., the driver on one crew can also be the driver on a different crew

2.) The ER model described in textual form (4 marks)

Entities & Attributes:

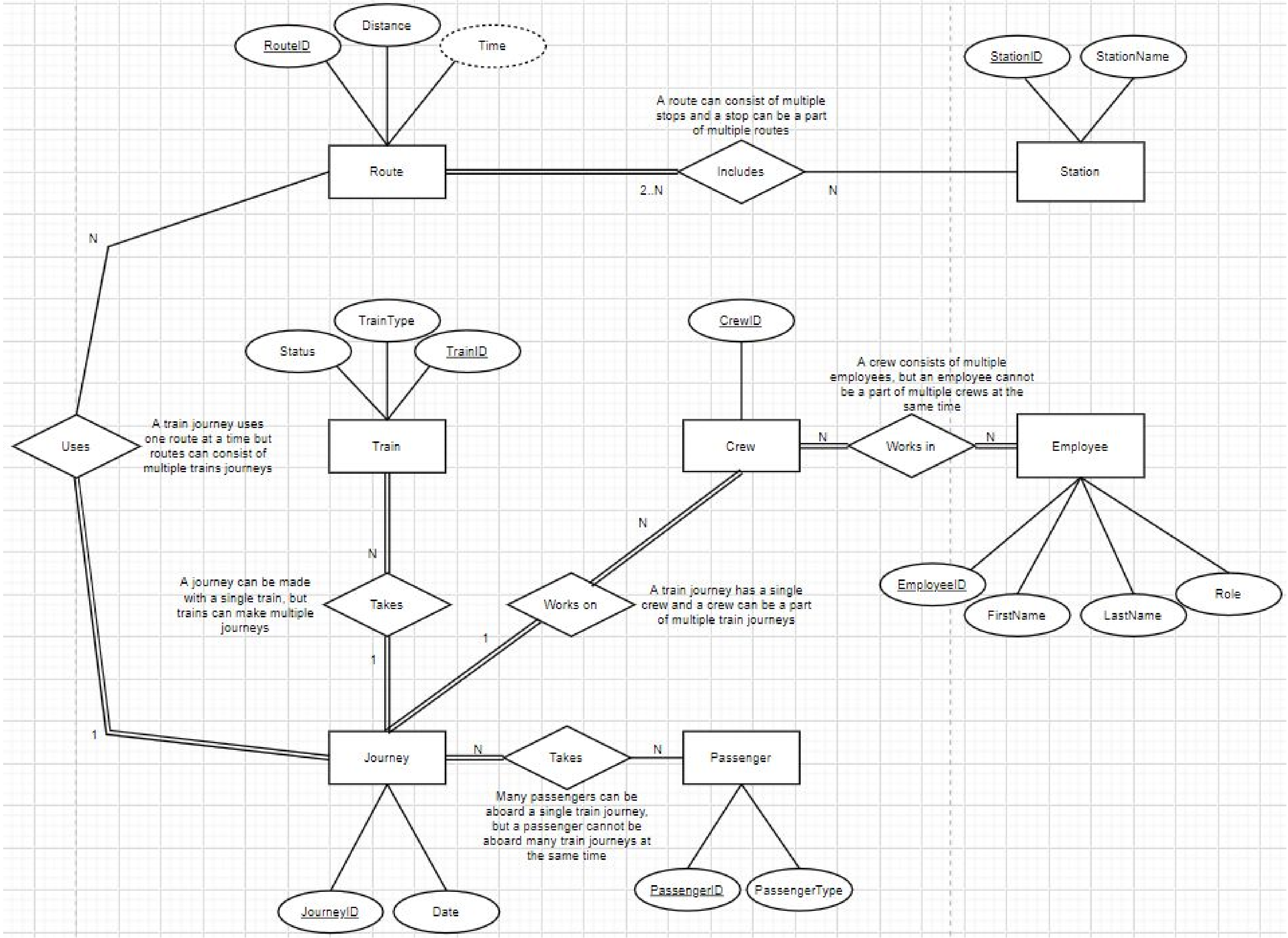
- Route
    - routeID(Primary Key)
    - startStation(StationName [Foreign key referencing Station)
    - terminalStation(StationName [Foreign key referencing Station)
    - distance
    - time
- Train
    - trainID (Primary Key)
    - trainType (Whether train is modern or not)
    - status (Whether train is operational or in service)
- Station
    - stationID (Primary Key)
    - stationName
- Passenger
    - passengerID (Primary Key)
    - passengerType (Occupation of passenger (students, accountant etc.)
    - journeyID (Foreign Key referencing Journey)

- Journey
  - journeyID (Primary Key)
  - date
  - routeID (Foreign Key referencing Route)
  - trainID (Foreign Key referencing Train)
  - crewID (Foreign Key referencing Crew)
- Crew
  - crewID (Primary Key)
  - supervisor(EmployeeID[Foreign Key referencing employee])
- Employee
  - employeeID (Primary Key)
  - firstName
  - lastName
  - role
  - crewID (Foreign Key referencing crew)[Updated every time an employee works in a new crew]

Relationships:

- **Includes** 2-N:N (between Route and Station)
  - Each route must have a minimum of 2 stations to as many as the route needs and each station can be an intermediary stop on many routes
- **Takes** N:N (between Passenger and Journey)
  - Any given passenger can go on many journeys and each journey can be taken by many passengers
- **Works_in** N:N (between Employee and Crew)
  - An employee can work in many different crews and each crew can have many employees
- **Works_On** 1:N (between Crew and Journey)
  - A given crew can operate many journeys, but each journey can only be run by one crew
- **Takes** N:1 (between Train and Journey)
  - A train can take many different journeys, but each journey can only be taken by 1 train
- **Uses** 1:N (between Journey and Route)
  - A journey can only have 1 route, but each route can be used by many journeys

3.) Entity-Relationship (ER) Diagram



A route can consist of multiple stops and a stop can be a part of multiple routes

A train journey uses one route at a time but routes can consist of multiple trains journeys

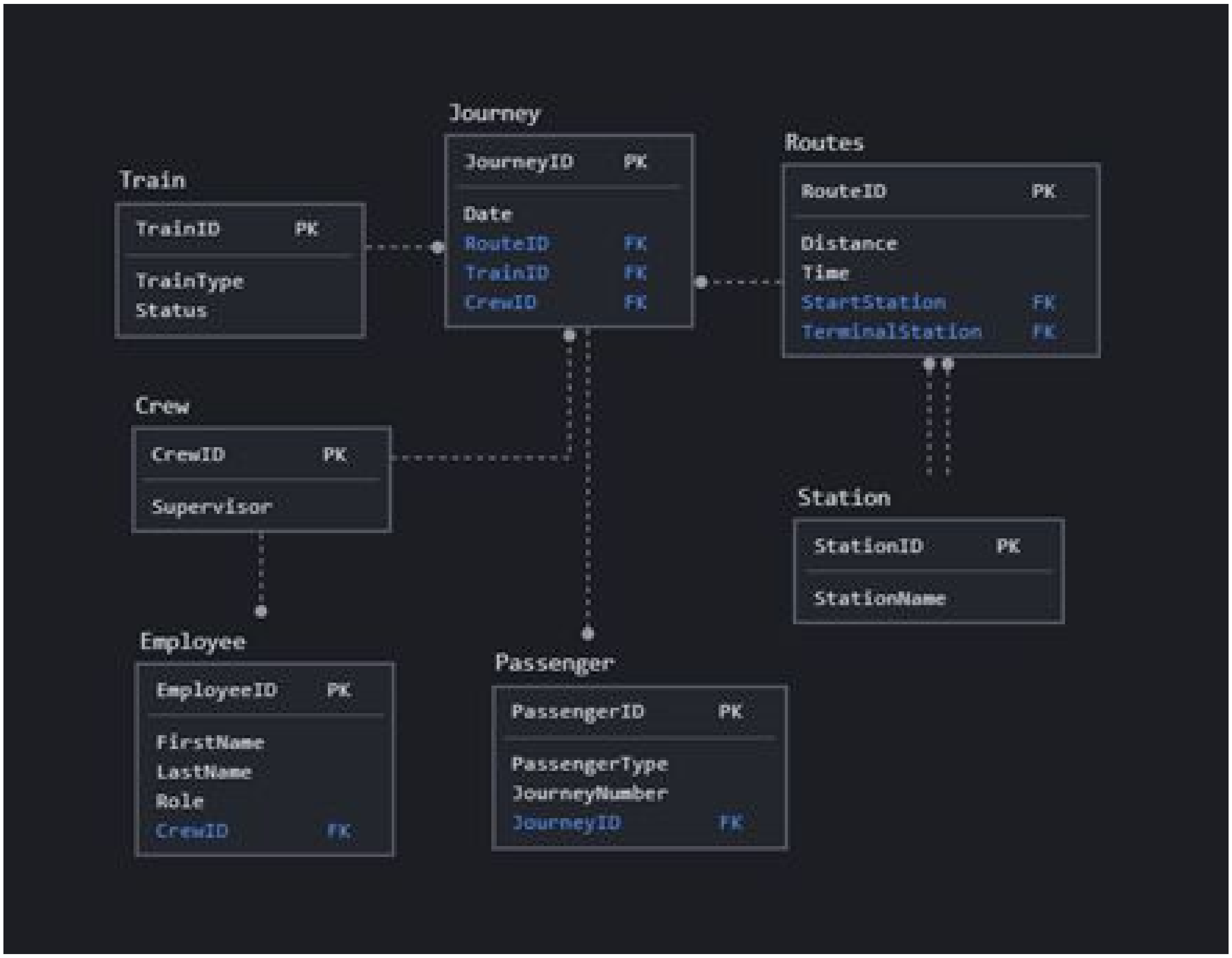
A crew consists of multiple employees, but an employee cannot be a part of multiple crews at the same time

A journey can be made with a single train, but trains can make multiple journeys

A train journey has a single crew and a crew can be a part of multiple train journeys

Many passengers can be aboard a single train journey, but a passenger cannot be aboard many train journeys at the same time

4.1) Relational Database Schema



The relationships between the entities outlined in the ER model in the previous section have been translated into the relational model via foreign keys. Any two entities that have either a “works_on, takes, or uses” relationship, will either have a foreign key referencing the entity’s counterpart or one of its attributes is referenced by a foreign key in another entity. For example, in order to translate the Takes relationship between the Train and Journey entities: TrainID is used as a foreign key in Journey to reference the train entity. This is outlined by the lines connecting the blue foreign key to the primary key in the appropriate entity.

4.2) You should clearly indicate the primary and foreign keys. Also, explain where normalization is required, and transform the model where required such that your design is in 3rd Normal Form.

1st normalisation form –

1st normal form describes a relation in which all attributes are atomic/single valued. The model above does not contain multivalued attributes. Thus, by default, the conditions of the first normal form have been met.

2nd normalisation form -

2nd normal form refers to a relation in which all non-primary key attributes must be functionally dependent on the primary key. An attribute X is functionally dependent on another if the same value of the other attribute is always associated with the same value of X. Evidently, each relation in the model above has one primary key and all non-primary key attributes are functionally dependent on the primary key.

After much thorough and critical discussion between group members, a consensus was reached that all attributes that are neither foreign nor primary keys are fully functionally dependent on the primary key in their respective entities. The question lied in whether the foreign keys were functionally dependent on the primary keys of their respective entities. It is clear however, that when each relation is analysed separately, each foreign key in every entity is functionally dependent on the primary key of that entity. For example, the passengerID (automatically generated each time a ticket is purchased) will always be associated with the same value for journeyID. This is because every time a passenger purchases a ticket for a specific journey with a certain journeyID, they will only ever have the passengerID associated with the ticket purchased for that journey. With that said, the conditions of the second normal form have been met by all relations.

3rd normalisation form –

3rd normal form describes a relation in which there are zero transitive dependencies. A transitive dependency occurs when one attribute indirectly depends on the primary key through an intermediary attribute. No relations contain any transitive dependencies as all attributes in all relations are directly dependent on the primary key. Therefore, the conditions for the 3rd normal form have also been met by default.