ECS519U – DATABASE SYSTEMS - COURSEWORK 2: DB DESIGN, ER (ENTITY RELATIONSHIP) MODEL AND RM (RELATIONAL MODEL)

Group Information

Group Number: 13

Group members:

- Samuele Joshi (180318461)
- Ali Irzam Kathia (190318158)
- Ismail Khan (190204482)
- Musa Lala (190801410)

Chosen Scenario:

- Eurostar-Scenario

Group contribution:

- This is a confirmation that everyone participated to a good standard, participating in every aspect of the project.

Part 1 – Assumptions/Requirements

Assumptions:

- Trains:
  - All trains are either modern or old
  - All trains are either in service or not
  - Modern trains can only go on modern routes.

- Journeys:
  - All journeys are on a singular date
  - Journeys will have 1 main route
  - Each journey will have one crew and it can't be changed
  - Each journey can have one train
  - Multiple routes can make up one journey

- Routes:
  - A main journey can/will be made up of smaller "routes" (there can be multiple smaller routes in one journey).
  - A route will be timed from when the train leaves to when it reaches its destination as opposed to it being an approximated calculation
  - All routes have a start station and a terminal station
  - All routes will have a start city and terminal city
  - All routes will have a distance that is entered when the train reaches its terminal station, similar to route time, this distance is measured based on the start and terminal stations which vary, thus implying the overall distance can vary if the train loops back on itself/uses an unconventional route that would in turn increase its distance travelled.
  - Different stations in different cities could have the same name or vice versa.

- Passengers:
  - Passengers will fit into one of the allocated types:
    [Child, Teenager, Student, Adult, Senior Citizens]

  - All passengers will be part of one particular journey although they can leave the train after completing one or more routes if required.

- Employees:
  - Management and sales employees are not part of the crew.
  - An employee can only fit into one job, a singular job listing (ID) can only encompass one employee
  - One crew can be responsible for multiple journeys.
  - One employee can be part of different crews.
  - Any of the conductors can be head of the crew.

Analysed Requirements:

- Model train routes from start to finish. (Name of station must be included.)
  - Routes have one start station.
  - Several stations in the middle (referred to as stops)
  - One terminal station.
- Distance and journey time must be stored in Route.
  - Example: London-Paris, Distance: 306-mile route, Time taken: 2hrs 50 mins.
- Trains:
  - Do they operate? (20 in total, average 16 in service, 4 regular service checks and repairs)
  - Trains have different routes.
  - Trains have different types. Modern or old.
- Employees:
  - 400 people in many different roles.
  - There are the train crews operating the trains.
  - Crew: 2 drivers, 2 conductors (1 head of the crew), 5 service team members (responsible for drinks and food), and 2 security guards.
  - Other employees are responsible for management and sales.
- DS MUST:
  - Keep track of each train trip and the employees involved in the trip.
  - The company wants to process queries such as "Show all train journeys where Mary Poppins was part of the train team".
  - Information about number and type of passengers are to be stored for each trip.
  - Then, queries such as "how many students travelled from St Pancras to Paris in Dec 2016" can be processed.

Part 2 – ER model (textual)

Entity (Primary Key, Attributes)

Train (TrainID, TModern, ServiceStatus)

Journey (JourneyID, JourneyDate)

Route (RouteID, StartStation, StartCity, TerminalStation, TerminalCity, Distance, Time, RModern)

Passenger (PassengerID, PFirstName, PSurname, PType)

Crew (CrewID)

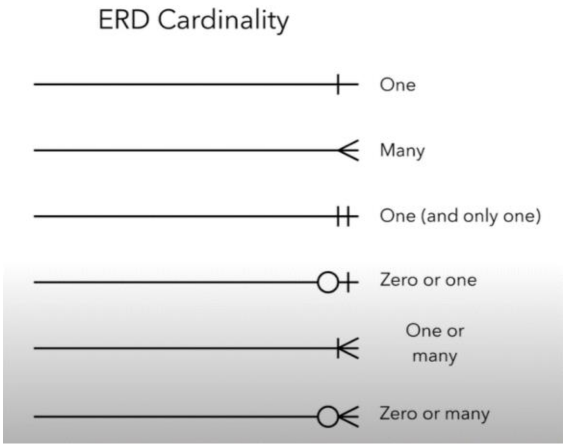
Employees (EmpID, EmpFirstName, EmpSurname)

Relationships:

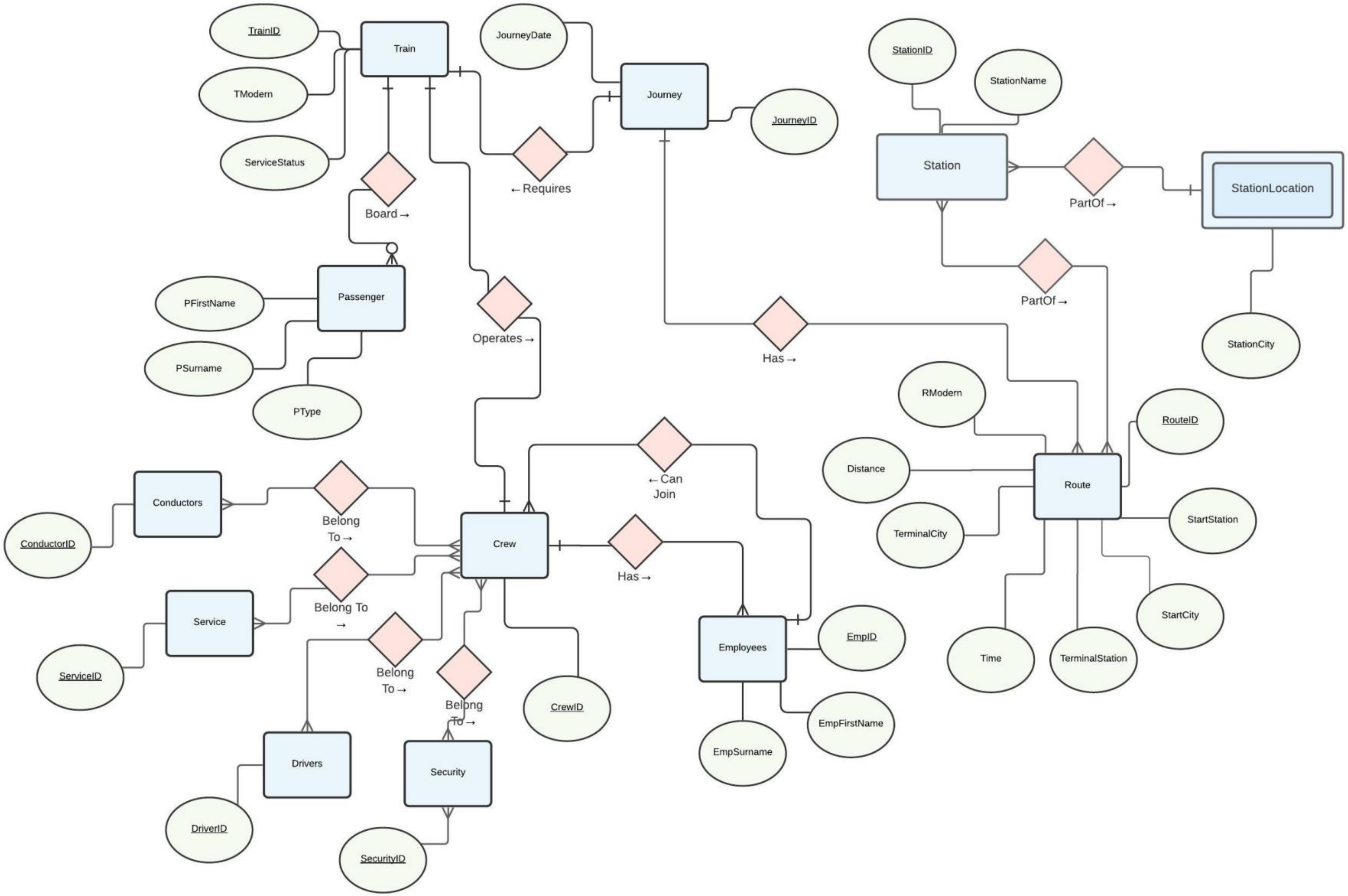
- Many Passengers **Board** 1 Train – a variation in the number of passengers per train is possible but for one train multiple passengers can board.
- 1 Crew **Operates** 1 Train - for a journey 1 set of crew members (that being the eleven members in total under a crew) are required.
- 1 Crew **Has** Many Employees - a crew is made up of a set number of employees (11).
- 1 Employee **Can Join** Many Crews – an employee from the Eurostar company has the ability to join as many crews as required.
- 1 Journey **Requires** 1 Train – One train must be available for a journey to commence. A journey only consists of one train from the starting station to the
- 1 Journey **Has** Many Routes – A singular journey can encompass many routes essentially building it out of an array of sub journeys (called routes in our case) that in themselves are encompassed by the longest route on that journey.
- Many conductors **belong** to many crews - 2 Conductors are required for every crew however the conductors themselves are not locked to 1 crew and can thus be part of multiple crews.
- Many Services **belong** to many crews - See above
- Many Drivers **belong** to many crews - See above
- Many Securities **belong** to many crews. - See above
- Many Stations are **PartOf** many routes - A station can be included on multiple routes as a route encompasses the traversal from a start station to a terminal station and any station can be listed as a start or terminal station for any number of routes.
- Many Stations are **PartOf** 1 StationLocation - a station location (i.e., city) can encompass multiple stations.

Part 3 – ER model (diagram)

Please refer to the key below:

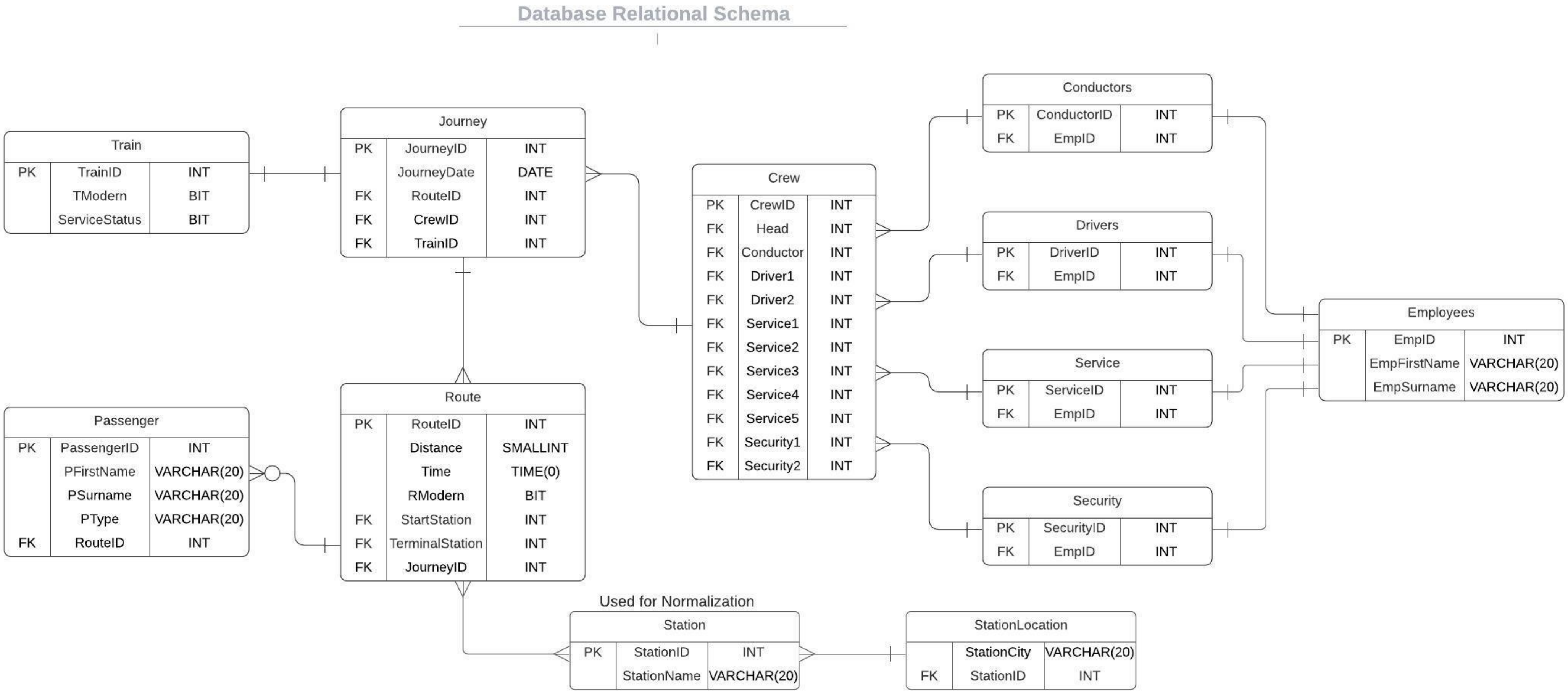


ER MODEL


TrainID
Train
TModern
ServiceStatus
JourneyDate
Journey
JourneyID
←Requires
Board→
Passenger
PFirstName
PSurname
PType
Operates→
StationID
StationName
Station
PartOf→
StationLocation
PartOf→
StationCity
Has→
RModern
RouteID
Distance
Route
TerminalCity
StartStation
StartCity
Time
TerminalStation
←Can Join
Conductors
ConductorID
Belong To→
Crew
Has→
Belong To →
Service
ServiceID
Belong To→
Belong To→
CrewID
Employees
EmpID
EmpFirstName
EmpSurname
Drivers
DriverID
Security
SecurityID

Part 4 – Relational Database Schema

Schema:



Database Relational Schema

Train

PK	TrainID	INT
	TModern	BIT
	ServiceStatus	BIT

Journey

PK	JourneyID	INT
	JourneyDate	DATE
FK	RouteID	INT
FK	CrewID	INT
FK	TrainID	INT

Crew

PK	CrewID	INT
FK	Head	INT
FK	Conductor	INT
FK	Driver1	INT
FK	Driver2	INT
FK	Service1	INT
FK	Service2	INT
FK	Service3	INT
FK	Service4	INT
FK	Service5	INT
FK	Security1	INT
FK	Security2	INT

Conductors

PK	ConductorID	INT
FK	EmpID	INT

Drivers

PK	DriverID	INT
FK	EmpID	INT

Service

PK	ServiceID	INT
FK	EmpID	INT

Security

PK	SecurityID	INT
FK	EmpID	INT

Employees

PK	EmpID	INT
	EmpFirstName	VARCHAR(20)
	EmpSurname	VARCHAR(20)

Passenger

PK	PassengerID	INT
	PFirstName	VARCHAR(20)
	PSurname	VARCHAR(20)
	PType	VARCHAR(20)
FK	RouteID	INT

Route

PK	RouteID	INT
	Distance	SMALLINT
	Time	TIME(0)
	RModern	BIT
FK	StartStation	INT
FK	TerminalStation	INT
FK	JourneyID	INT

Used for Normalization

Station

PK	StationID	INT
	StationName	VARCHAR(20)

StationLocation

	StationCity	VARCHAR(20)
FK	StationID	INT

Normalisation:

Firstly, as there are multiple drivers, service and security personnel within a singular crew, the duplicities of these fields have been split up to encompass as many of the singular, duplicated roles we need within a crew (i.e., The Driver field has been split into driver1 & driver2 since we need two drivers) we then have those as foreign keys from their own respective tables, thus implying the fields are all atomic. Another place in which we did this is since a singular journey has multiple stops, instead of storing multiple stops inside of one field we created a route table that creates sub routes between all of the stops (including the start and terminal stops) so that all of these sub journeys are accounted for.

2NF

(All non-key attributes must be fully dependent on primary keys) - Route has been split so that its station data is now compiled into a separate table, thus implying the data within route is now dependent upon its primary key whereas the in-depth station data is available (whilst still accessible for use in the same manner) elsewhere.

3NF

Start stations and Terminal stations (encompassing stops) have been moved into a separate table to prevent a station and its respective city being updated (i.e., one without the other) to prevent a non-key attribute from determining the attribute of another entity which would in turn cause the schema to stop adhering to 3rd normal form.