

Project Phase 1 – 41

Introduction to the Mini-World: This mini-world is based on a railway system. It is utilised to track bookings by customers and the schedules of trains.

Purpose of the Database: It is used to store data about passengers, ticket reservations, trains, their schedules, staff etc. This data can be used to analyse booking patterns and other train station trends.

Users of the Database: Potential Passengers are naïve/parametric end users who utilise the database to view trains for their specific requirements. Application Programmers can use this database to build a potential ticket booking website/app. Railway management act as Casual Users to view required data for analysis.

Applications of the Database: It can be used to

- Generate daily schedules for trains
- Generate a list of available seats for a particular train
- Show available trains based on destination and source locations.
- Track the schedule of employees to assign them to a particular train.

DATABASE REQUIREMENTS:

Assumptions: For this database, only passenger trains are being considered.

Only trains with no stopover destinations are considered.

Only passenger coaches are considered not good-transporting coaches.

A train can have a minimum of 12 coaches and a maximum of 24 coaches.

A coach is considered to have a minimum of 35 seats to 45 maximum seats

A passenger can have a maximum of 4 dependent passengers.

Strong entity types:

TRAIN

Train_number – Primary Composite Key (attribute 1), (5-digit constraint)

Distance – Simple Attribute

Train_name – Primary Composite Key (attribute 2)

Departure – Composite Attribute (Date, Time), Date/Time domain

Arrival - Composite Attribute (Date, Time), Date/Time domain

Departure_Station_Code – Foreign Key

Arrival_Station_Code - Foreign Key

Duration – Derived Attribute

STATION

Station_Code – Primary Composite Key (attribute 1) (5-digit constraint)

Station_name – Primary Composite Key (attribute 2)

City_name – Simple Attribute

Zone_name – Simple Attribute

PASSENGER

Passenger_ID – Primary Key (12-digit constraint)

Name – Composite Attribute (First Name, Last Name)

Date_of_Birth – Simple Attribute (Date/Time domain)

Age – Derived Attribute

Gender – Simple Attribute (Single-valued, Domain: M/F/Other)

Phone_number – Multivalued Attribute (10-digit constraint)

Subclasses – Student (add attribute Student_ID – Secondary Key) , Military (add attribute Military_ID – Secondary Key)

TICKET

Ticket_ID – Primary Key (9-digit constraint)

Arrival – Composite Attribute (Date, Time) (Date/Time domain)

Departure – Composite Attribute (Date, Time) (Date/Time domain)

From_location – Simple Attribute

To_location – Simple Attribute

Train_Code – Foreign Key

EMPLOYEE

Employee_ID – Primary Key (12-digit constraint)

Name – Composite Attribute (First_name, Last_name)

Date_of_Birth – Simple Attribute (Date/Time domain)

Age – Derived Attribute

Phone Number – Multivalued Attribute (10-digit constraint)

Working_days – Multivalued Attribute (Domain: Mon, Tue, Wed, Thu, Fri, Sat, Sun)

Working_hours – Simple Attribute

Salary – Simple Attribute

COACH:

Coach_ID: Primary Key (3-digit constraint)

Train_Num: Foreign Key

Seats_per_coach – Simple Attribute

Coach_type – Simple Attribute (Domain: First_class, Second_class, General)

Weak entity types:

DEPENDENT_PASSENGER:

Passenger_ID – Partial Key

Name – Composite Attribute (First Name, Last Name)

Date_of_Birth – Simple Attribute (Date/Time domain)

Age – Derived Attribute

Gender – Simple Attribute (Domain: M/F/Other)

Phone_number – Multivalued Attribute (10-digit constraint)

Subclasses – Children < 18 years of age, Elderly > 65 years of age

SEAT:

Coach_ID – Partial Key

Seat_num – Simple Attribute

Availability – Simple Attribute (Domain: Yes/No)

Seat_type – Simple Attribute (Domain: Upper_berth/Middle_berth/Lower_berth/Seated)

FARE:

Ticket_ID – Partial Key

Payment_ID – Unique Attribute (8-digit constraint)

Payment_type – Simple Attribute (Domain: Cash/Credit/Debit/UPI/NetBanking)

Amount – Simple Attribute

Payment_date_time – Simple Attribute (Date/Time domain)

DISCOUNT:

Passenger_ID: Partial Key

Refund_amt: Simple Attribute

Relationship Types:

1) ELIGIBLE_FOR -

Degree: 2

Participating Entity Types: Passenger, Discount (Passenger ELIGIBLE_FOR Discount)

Cardinality Ratio: 1:1

Constraints: Passenger (1,1) – Discount (1,1)

2) HAS -

Degree: 2

Participating Entity Types: Ticket, Fare (Ticket HAS Fare)

Cardinality Ratio: 1:1

Constraints: Ticket (1,1) – Fare (1,1)

3) CONSISTS_OF:

Degree: 2

Participating Entity Types: Train, Coach (Train CONSISTS_OF Coach)

Cardinality Ratio: 1:N

Constraints: Trains(1,1) – Coaches (12, 24)

4) CONTAIN -

Degree: 2

Participating Entity Types: Coach, Seat (Coach CONTAIN seat)

Cardinality Ratio: 1:N

Constraints: Coach (1,1) – Seat (30,45)

5) DEPEND_ON -

Degree: 2

Participating Entity Types: Dependent_Passenger, Passenger
(Dependent_Passenger DEPEND_ON Passenger)

Cardinality Ratio: N:1

Constraints: Dependent_Passengers (1,4) – Passengers (1,1)

6) SUPERVISES –

Degree: 1 (Recursive)

Participating Entity Types: Employee (Employee as Supervisor
SUPERVISES Employee as Supervisee)

Cardinality Ratio: 1:N

Constraints: Supervisor (1,1) – Supervisee (1,N)

Degree > 2 Relationship Types:

1) WORK_FOR–

Degree: 3

Participating Entity Types: Employee, Train, Station
(Employee WORKS_FOR Train in Station)

Cardinality Ratio: N:M:1

Constraints: Employee (1, N) – Train (1,M) – Station(1,1)

2) TRAVEL

Degree: 4

Participating Entity Types: Passenger, Train, Stations, Tickets
(Passenger TRAVEL on trains between stations with tickets)

Cardinality Ratio: N:1:2:1

Constraints: Passenger (1,N) – Train (1,1) - Station (2,2) - Ticket(1,1)

Functional Requirements:

Retrieval:

SELECTION – Select all trains travelling on a certain date, retrieve all trains travelling between two particular locations or a particular station, the names and working hours of all employees working on a particular day.

PROJECTION QUERY – Trains that have more than 100 passengers, trains that have more than 50 available seats, names of passengers who have paid via

credit card, Names of trains and coaches that have military passengers, names of passengers that have booked a first-class seat.

AGGREGATE – Average number of passengers travelling in a particular train, maximum number of working hours for a particular employee, the maximum discount amount issued, the sum of all fares paid, and the maximum and minimum ticket price for a particular train.

SEARCH – Search for families of passengers based on last name, search for a particular train based on substring of train_name, search for an employee based on substring of name.

Modifications:

INSERT – Insert data of new passenger checking integrity constraints of id and other attributes, insert data of employee on hiring, generate a ticket based on passenger specifications and seat availability.

DELETE – Delete ticket on cancellation, delete employee record on resignation/firing, delete ticket after expiration of departure date and time without violation of referential constraints.

UPDATE – Update passenger and dependant details, update employee work timings, and update train timings in case of delay/early arrival.

ANALYSIS –

Check the station details where bookings for first-class coach tickets for trains are maximum or minimum.

Type of seats most preferred by a certain demographic of passengers (children, elderly people).

Names of trains that carry more/less than the average number of passengers.

Find the payment method which is most/least preferred by passengers of a particular station.

Salaries earned by employees on days where there are maximum number of ingoing and outgoing trains.

Summary:

This is an initial database requirements document for a railway management database that keeps track of customers, employees and trains for stations so as to ensure the smooth operation of the railway system. End users can use the data abstraction provided by this mini-world to meet their respective needs.