RAILWAY MANAGEMENT SYSTEM

INDEX

PROBLEM STATEMENT

Developing a Railway Management System that makes running trains smoother and will overcome the problems faced by passengers. It will be able to handle multiple actions like –

- Keeping records for train timings
- Records of ticket reservations and ticket bookings
- Details about the passenger
- Knowing routes of trains
- Records of platform number and stations.

INTRODUCTION

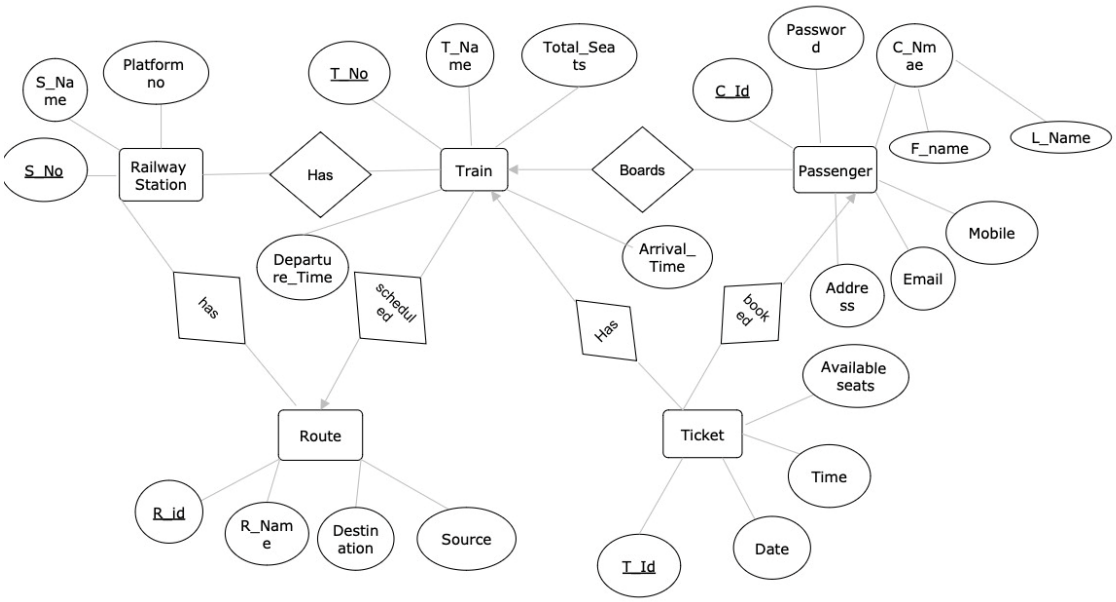
Database management is an essential aspect of managing a railway network. Efficient management of data ensures the smooth functioning of the railway network.

In context of railways, database management system involves organising and storing information related to accurate records of train timings which ensures passengers to plan their journeys accordingly, it also stores details of passengers safely, provides information about routes (source, destination) between which a particular train is running, it provides a passenger to know the prices of different tickets and handles ticket reservations efficiently.

The purpose of the project is to reduce manual work for managing trains, timetable, passengers, stations and routes.

ER DIAGRAM

ER Model stands for Entity Relationship Model and is a high-level conceptual data model diagram. The ER model helps to systematically analyse data requirements to produce a well-designed database. It represents real-world entities and the relationships between them and creating an ER Model in DBMS is considered as a best practice before implementing your database.



The ER diagram made for our project consists of 5 entities with their corresponding attributes describing how these entities are related to each other in the database. The relationship between the entities are represented by lines connecting them.

Entities and Attributes:

-Railway station (S_No,S_Name,Platform_No)

- Train (T_No,T_Name,Arrival Time,Departuret Time,Total seats,Scheduled)

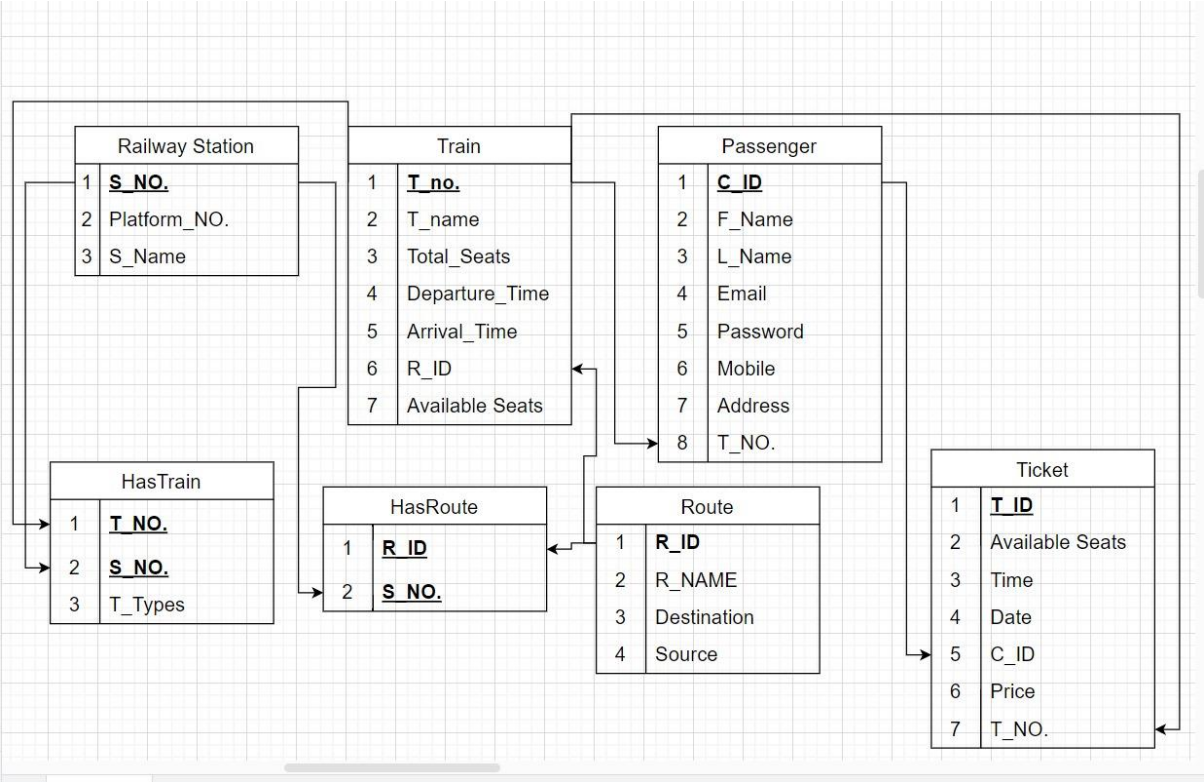
- Route (R_id,R_Name,Destinamtion,Source)

-Passenger (C_id,Password,F_Name,L_Name,Moblie,Address,Email,Booked)

-Ticket (T_id,Time,Date,Available seats)

ER To Table

An ER diagram is a graphical representation of entities and relationships in the database and these diagrams can be converted into the tables so that it can be easily implemented by any Relational Database Management System. The above ER diagram contains following relationships which can be mapped into tables as:



NORMALISATION

Railway station **HAS** Train

T_types	S_NO	T_NO.

PASSENGER

EMAIL	PASSWORD	F_NAME	L_NAME	ADDRESS	MOBILE	C_ID	T_NO.

Train

AVAILABLE SEATS	TOTAL SEATS	R_ID	T_NO	ARRIVAL TIME	DEPT.TIME	T_NAME

Route

SOURCE	DESTINATION	R_NAME	R_ID

HasRoute

S_NO.	R_ID

Ticket

T_ID	C_ID	AVAILABLE SEATS

RailwayStation

S_NO.	PLATFORM_NO.

SQL CODE

```
CREATE TABLE Route(
   R_id NUMBER PRIMARY KEY,
   R_name VARCHAR(20),
   Source VARCHAR(20),
   Destination VARCHAR(20)
   );
INSERT INTO Route VALUES(321,'JPJUC','Jaipur','Jalandhar');
INSERT INTO Route VALUES(784,'KOTAUDZ','Kota', 'Udaipur');
INSERT INTO Route VALUES(591,'RKSHCDG','Rishikesh','Chandigarh');
INSERT INTO Route VALUES(843, 'ASRNDLS','Amritsar', 'New Delhi');
INSERT INTO Route VALUES(654,'MMCT','Mumbai','Ambala');
SELECT * FROM Route;

CREATE TABLE Train(
        T_No NUMBER PRIMARY KEY,
   AvailableSeats NUMBER,
   T_Name VARCHAR(200),
   DepartureTime TIMESTAMP,
   ArrivalTime TIMESTAMP,
   TotalSeats NUMBER,
   R_id NUMBER,
   FOREIGN KEY(R_id) REFERENCES Route(R_id));
INSERT INTO Train VALUES(22429,823,'ASR SHTBDI EXP','01-Jan-2024
07:20:00am','01Jan2024 11:30:30pm',1000,843);
INSERT INTO Train VALUES(12926,060,'AII ASR EXP','01-Jan-2024
8:05:05am','01Jan2024 12:35:35pm',500,321);
INSERT INTO Train VALUES(12497,005,'PASCHIM EXP','01-Jan-2024
11:30:00am','01Jan2024 02:30:30pm',500,654);
```

```
INSERT INTO Train VALUES(20807,081,'MEWAR EXP','01-Jan-2024 07:20:00 am','01-Jan2024 10:30:30pm',250,784);

INSERT INTO Train VALUES(12013,450,'HEMKUNT EXP','01-Jan-2024 5:20:20pm','01Jan-2024 9:58:58 pm',1000,591);

Select * from Train;


CREATE TABLE Passenger(
   C_id NUMBER PRIMARY KEY,
   Email VARCHAR(100),
   Pass VARCHAR(100),
   F_Name VARCHAR(200),
   L_Name VARCHAR(200),
   Mobile NUMBER,
   Address VARCHAR(200),
   T_No NUMBER REFERENCES Train(T_No)
);

INSERT INTO Passenger VALUES(11248,'sbansal@gmail.com', ****", 'Sanchit','Bansal',7009548392,'Bathinda', 12926);

INSERT INTO Passenger VALUES(11432,'shanda@yahoo.in','****', 'Shererar', 'Handa',7855623454,'Lucknow', 12013);

INSERT INTO Passenger VALUES(11741,'vgarg@gmail.com', '****', 'Vaibhav', 'Garg',7865432123,'Hyderabad',22429);

INSERT INTO Passenger VALUES(11875,'sloyal@gmail.com','****', 'Sarika', 'Loyal',9876523412,'Mumbai',12497);

INSERT INTO Passenger VALUES(11657,'mkathuria@gmail.com', '****', 'Maulik', 'Kathuria',7895623490,'Jaipur',20807);

INSERT INTO Passenger VALUES(11775,'skumar@gmail.com', '****', 'Suvit', 'Kumar',7007654323,'Patiala',12926);

INSERT INTO Passenger VALUES(11543,'rbansal@gmail.com', '****', 'Rohan', 'Bansal',7895343560,'Amritsar',22429);

INSERT INTO Passenger VALUES(11331,'ngarg@gmail.com', '****', 'Naman', 'Garg',7234568912,'Mumbai',12497);
```

```
INSERT INTO Passenger VALUES(11345,'msingla@gmail.com', '****', 'Mohit',
'Singla',7654632142,'Kota',20807);

INSERT INTO Passenger VALUES(11871,'gsingh@gmail.com', '****', 'Gurman',
'Singh',78956267091,'Rishikesh',12013);

SELECT * FROM Passenger;


CREATE TABLE Ticket(
   T_id NUMBER PRIMARY KEY,
   Timee TIMESTAMP,
   Datee DATE,
   Price NUMBER,
   C_id NUMBER,
   AvailableSeats NUMBER DEFAULT 1000,
   T_No NUMBER REFERENCES Train(T_No),
   FOREIGN KEY(C_id) REFERENCES Passenger(C_id)
   );
INSERT INTO Ticket VALUES(2819542664,'22-Apr-2024
09:15:32am','22Apr2024',186,11248,200,22429);

INSERT INTO Ticket VALUES(2819542612,'14-Apr-2024 10:45:44am','14Apr2024',
125, 11432,300,20807);

INSERT INTO Ticket VALUES(2819542687,'26-Apr-2024 10:10:55pm','26Apr2024',
550, 11741,400,12926);

INSERT INTO Ticket VALUES(2819542655,'10-Apr-2024
03:40:21pm','10Apr2024',240,11875,500,12497);

INSERT INTO Ticket VALUES(2819542694,'07-Apr-2024
01:57:43pm','15Apr2024',605,11657,400,12013);

INSERT INTO Ticket VALUES(2819542692,'09-Apr-2024
01:57:43pm','21Apr2024',605,11775,400,12926);

INSERT INTO Ticket VALUES(2819542685,'21-Apr-2024
01:57:43pm','24Apr2024',605,11543,400,22429);

INSERT INTO Ticket VALUES(2819542613,'23-Apr-2024
01:57:43pm','26Apr2024',605,11331,400,12497);
```

```
INSERT INTO Ticket VALUES(2819542602,'03-Apr-2024
01:57:43pm','29Apr2024',605,11345,400,20807);

INSERT INTO Ticket VALUES(2819542680,'05-Apr-2024
01:57:43pm','30Apr2024',605,11871,400,12013);

SELECT * FROM Ticket;


CREATE TABLE  RailwayStation(
   S_No NUMBER PRIMARY KEY,
   Platform_No NUMBER,
   S_Name VARCHAR(100)  );

INSERT INTO RailwayStation VALUES(54,3,'Rishikesh');

INSERT INTO RailwayStation VALUES(28,6,'Amritsar');

INSERT INTO RailwayStation VALUES(62,2,'Mumbai');

INSERT INTO RailwayStation VALUES(43,5,'Kota');

INSERT INTO RailwayStation VALUES(31,1,'Jaipur');

SELECT * FROM RailwayStation;


CREATE TABLE  HasRoute(
   R_id NUMBER,
   S_No NUMBER,
   foreign key(R_id) references Route(R_id) ,
   foreign key (S_no) references RailwayStation(S_no)
   );

INSERT INTO HasRoute VALUES(654,62);

INSERT INTO HasRoute VALUES(591,54);

INSERT INTO HasRoute VALUES(321,31);

INSERT INTO HasRoute VALUES(843,28);

INSERT INTO HasRoute VALUES(784,43);

SELECT * FROM HasRoute;
```

```
Create table HasTrain(
    T_types varchar(20),
    Tno number,
    Sno number,
    primary key(Tno,Sno),
    foreign key(Tno) references Train(T_no),
    foreign key(Sno) references RailwayStation(S_no)

    );
INSERT INTO HasTrain VALUES('Express',22429,28);
INSERT INTO HasTrain VALUES('Express',20807,43);
INSERT INTO HasTrain VALUES('Express',12926,31);
INSERT INTO HasTrain VALUES('Express',12497,62);
INSERT INTO HasTrain VALUES('Express',12013,54);
SELECT * FROM HasTrain;
```

OUTPUT SCREENSHOTS OF SQL

Table ROUTE -

```
CREATE TABLE Route(
    R_id NUMBER PRIMARY KEY,
    R_name VARCHAR(20),
    Source VARCHAR(20),
    Destination VARCHAR(20)
    );
INSERT INTO Route VALUES(321,'JPJUC','Jaipur','Jalandhar');
INSERT INTO Route VALUES(784,'KOTAUDZ','Kota', 'Udaipur');
INSERT INTO Route VALUES(591,'RKSHCDG','Rishikesh','Chandigarh');
INSERT INTO Route VALUES(843, 'ASRNDLS','Amritsar', 'New Delhi');
INSERT INTO Route VALUES(654,'MMCT','Mumbai','Ambala');
SELECT * FROM Route;
```

R_ID	R_NAME	SOURCE	DESTINATION
321	JPJUC	Jaipur	Jalandhar
784	KOTAUDZ	Kota	Udaipur
591	RKSHCDG	Rishikesh	Chandigarh
843	ASRNDLS	Amritsar	New Delhi
654	MMCT	Mumbai	Ambala

Download CSV

5 rows selected.

Table TRAIN-

```
CREATE TABLE Train(
    T_No NUMBER PRIMARY KEY,
    AvailableSeats NUMBER,
    T_Name VARCHAR(200),
    DepartureTime TIMESTAMP,
    ArrivalTime TIMESTAMP,
    TotalSeats NUMBER,
    R_id NUMBER,
    FOREIGN KEY(R_id) REFERENCES Route(R_id));
INSERT INTO Train VALUES(22429,823,'ASR SHTBDI EXP','01-Jan-2024 07:20:00am','01Jan2024 11:30:30pm',1000,843);
INSERT INTO Train VALUES(12926,060,'AII ASR EXP','01-Jan-2024 8:05:05am','01Jan2024 12:35:35pm',500,321);
INSERT INTO Train VALUES(12497,005,'PASCHIM EXP','01-Jan-2024 11:30:00am','01Jan2024 02:30:30pm',500,654);
INSERT INTO Train VALUES(20807,081,'MEWAR EXP','01-Jan-2024 07:20:00 am','01-Jan2024 10:30:30pm',250,784);
INSERT INTO Train VALUES(12013,450,'HEMKUNT EXP','01-Jan-2024 5:20:20pm','01Jan-2024 9:58:58 pm',1000,591);
Select * from Train;
```

T_NO	AVAILABLESEATS	T_NAME	DEPARTURETIME	ARRIVALTIME	TOTALSEATS	R_ID
22429	823	ASR SHTBDI EXP	01-JAN-24 07.20.00.000000 AM	01-JAN-24 11.30.30.000000 PM	1000	843
12926	60	AII ASR EXP	01-JAN-24 08.05.05.000000 AM	01-JAN-24 12.35.35.000000 PM	500	321
12497	5	PASCHIM EXP	01-JAN-24 11.30.00.000000 AM	01-JAN-24 02.30.30.000000 PM	500	654
20807	81	MEWAR EXP	01-JAN-24 07.20.00.000000 AM	01-JAN-24 10.30.30.000000 PM	250	784
12013	450	HEMKUNT EXP	01-JAN-24 05.20.20.000000 PM	01-JAN-24 09.58.58.000000 PM	1000	591

Download CSV

5 rows selected

Table PASSENGER-

```
CREATE TABLE Passenger(
    C_id NUMBER PRIMARY KEY,
    Email VARCHAR(100),
    Pass VARCHAR(100),
    F_Name VARCHAR(200),
    L_Name VARCHAR(200),
    Mobile NUMBER,
    Address VARCHAR(200),
    T_No NUMBER REFERENCES Train(T_No)
);
INSERT INTO Passenger VALUES(11248,'sbansal@gmail.com', '****', 'Sanchit','Bansal',7009548392,'Bathinda', 12926);
INSERT INTO Passenger VALUES(11432,'shanda@yahoo.in','****', 'Shererar', 'Handa',7855623454,'Lucknow', 12013);
INSERT INTO Passenger VALUES(11741,'vgarg@gmail.com', '****', 'Vaibhav', 'Garg',7865432123,'Hyderabad',22429);
INSERT INTO Passenger VALUES(11875,'sloyal@gmail.com','****', 'Sarika', 'Loyal',9876523412,'Mumbai',12497);
INSERT INTO Passenger VALUES(11657,'mkathuria@gmail.com', '****', 'Maulik', 'Kathuria',7895623490,'Jaipur',20807);
INSERT INTO Passenger VALUES(11775,'skumar@gmail.com', '****', 'Suvit', 'Kumar',7007654323,'Patiala',12926);
INSERT INTO Passenger VALUES(11543,'rbansal@gmail.com', '****', 'Rohan', 'Bansal',7895343560,'Amritsar',22429);
INSERT INTO Passenger VALUES(11331,'ngarg@gmail.com', '****', 'Naman', 'Garg',7234568912,'Mumbai',12497);
INSERT INTO Passenger VALUES(11345,'msingla@gmail.com', '****', 'Mohit', 'Singla',7654632142,'Kota',20807);
INSERT INTO Passenger VALUES(11871,'gsingh@gmail.com', '****', 'Gurman', 'Singh',78956267091,'Rishikesh',12013);
SELECT * FROM Passenger;
```

C_ID	EMAIL	PASS	F_NAME	L_NAME	MOBILE	ADDRESS	T_NO
11248	sbansal@gmail.com	****	Sanchit	Bansal	7009548392	Bathinda	12926
11432	shanda@yahoo.in	****	Shererar	Handa	7855623454	Lucknow	12013
11741	vgarg@gmail.com	****	Vaibhav	Garg	7865432123	Hyderabad	22429
11875	sloyal@gmail.com	****	Sarika	Loyal	9876523412	Mumbai	12497
11657	mkathuria@gmail.com	****	Maulik	Kathuria	7895623490	Jaipur	20807
11775	skumar@gmail.com	****	Suvit	Kumar	7007654323	Patiala	12926
11543	rbansal@gmail.com	****	Rohan	Bansal	7895343560	Amritsar	22429
11331	ngarg@gmail.com	****	Naman	Garg	7234568912	Mumbai	12497
11345	msingla@gmail.com	****	Mohit	Singla	7654632142	Kota	20807
11871	gsingh@gmail.com	****	Gurman	Singh	78956267091	Rishikesh	12013

Table TICKET-

```
CREATE TABLE Ticket(
    T_id NUMBER PRIMARY KEY,
    Timee TIMESTAMP,
    Datee DATE,
    Price NUMBER,
    C_id NUMBER,
    AvailableSeats NUMBER DEFAULT 1000,
    T_No NUMBER REFERENCES Train(T_No),
    FOREIGN KEY(C_id) REFERENCES Passenger(C_id)
    );
INSERT INTO Ticket VALUES(2819542664,'22-Apr-2024 09:15:32am','22Apr2024',186,11248,200,22429);
INSERT INTO Ticket VALUES(2819542612,'14-Apr-2024 10:45:44am','14Apr2024', 125, 11432,300,20807);
INSERT INTO Ticket VALUES(2819542687,'26-Apr-2024 10:10:55pm','26Apr2024', 550, 11741,400,12926);
INSERT INTO Ticket VALUES(2819542655,'10-Apr-2024 03:40:21pm','10Apr2024',240,11875,500,12497);
INSERT INTO Ticket VALUES(2819542694,'07-Apr-2024 01:57:43pm','15Apr2024',605,11657,400,12013);
INSERT INTO Ticket VALUES(2819542692,'09-Apr-2024 01:57:43pm','21Apr2024',605,11775,400,12926);
INSERT INTO Ticket VALUES(2819542685,'21-Apr-2024 01:57:43pm','24Apr2024',605,11543,400,22429);
INSERT INTO Ticket VALUES(2819542613,'23-Apr-2024 01:57:43pm','26Apr2024',605,11331,400,12497);
INSERT INTO Ticket VALUES(2819542602,'03-Apr-2024 01:57:43pm','29Apr2024',605,11345,400,20807);
INSERT INTO Ticket VALUES(2819542680,'05-Apr-2024 01:57:43pm','30Apr2024',605,11871,400,12013);
SELECT * FROM Ticket;
```

T_ID	TIMEE	DATEE	PRICE	C_ID	AVAILABLESEATS	T_NO
2819542664	22-APR-24 09.15.32.000000 AM	22-APR-24	186	11248	200	22429
2819542612	14-APR-24 10.45.44.000000 AM	14-APR-24	125	11432	300	20807
2819542687	26-APR-24 10.10.55.000000 PM	26-APR-24	550	11741	400	12926
2819542655	10-APR-24 03.40.21.000000 PM	10-APR-24	240	11875	500	12497
2819542694	07-APR-24 01.57.43.000000 PM	15-APR-24	605	11657	400	12013
2819542692	09-APR-24 01.57.43.000000 PM	21-APR-24	605	11775	400	12926
2819542685	21-APR-24 01.57.43.000000 PM	24-APR-24	605	11543	400	22429
2819542613	23-APR-24 01.57.43.000000 PM	26-APR-24	605	11331	400	12497
2819542602	03-APR-24 01.57.43.000000 PM	29-APR-24	605	11345	400	20807
2819542680	05-APR-24 01.57.43.000000 PM	30-APR-24	605	11871	400	12013

Table RAILWAY STATION

```
CREATE TABLE  RailwayStation(
    S_No NUMBER PRIMARY KEY,
    Platform_No NUMBER,
    S_Name VARCHAR(100)
    );
INSERT INTO RailwayStation VALUES(54,3,'Rishikesh');
INSERT INTO RailwayStation VALUES(28,6,'Amritsar');
INSERT INTO RailwayStation VALUES(62,2,'Mumbai');
INSERT INTO RailwayStation VALUES(43,5,'Kota');
INSERT INTO RailwayStation VALUES(31,1,'Jaipur');
SELECT * FROM RailwayStation;
```

S_NO	PLATFORM_NO	S_NAME
54	3	Rishikesh
28	6	Amritsar
62	2	Mumbai
43	5	Kota
31	1	Jaipur

Table HasRoute-

```
CREATE TABLE  HasRoute(
    R_id NUMBER,
    S_No NUMBER,
    foreign key(R_id) references Route(R_id) ,
    foreign key (S_no) references RailwayStation(S_no)
    );
INSERT INTO HasRoute VALUES(654,62);
INSERT INTO HasRoute VALUES(591,54);
INSERT INTO HasRoute VALUES(321,31);
INSERT INTO HasRoute VALUES(843,28);
INSERT INTO HasRoute VALUES(784,43);
SELECT * FROM HasRoute;
```

R_ID	S_NO
654	62
591	54
321	31
843	28
784	43

Table HasTrain-

```
Create table HasTrain(
    T_types varchar(20),
    Tno number,
    Sno number,
    primary key(Tno,Sno),
    foreign key(Tno) references Train(T_no),
    foreign key(Sno) references RailwayStation(S_no)

    );
INSERT INTO HasTrain VALUES('Express',22429,28);
INSERT INTO HasTrain VALUES('Express',20807,43);
INSERT INTO HasTrain VALUES('Express',12926,31);
INSERT INTO HasTrain VALUES('Express',12497,62);
INSERT INTO HasTrain VALUES('Express',12013,54);
SELECT * FROM HasTrain;
```

T_TYPES	TNO	SNO
Express	22429	28
Express	20807	43
Express	12926	31
Express	12497	62
Express	12013	54

PLSQL CODE

1. Trigger to update train departure date to current date :-

```
CREATE OR REPLACE TRIGGER UpdateTrainDepartureTrigger

BEFORE INSERT OR UPDATE ON Train

FOR EACH ROW

BEGIN

   :NEW.DepartureTime := TRUNC(SYSDATE) + ( :NEW.DepartureTime -
TRUNC(:NEW.DepartureTime) );

   :NEW.ArrivalTime := TRUNC(SYSDATE) + (:NEW.ArrivalTime - TRUNC
(:NEW.ArrivalTime));

   :NEW.TotalSeats := 1000;

END;

SELECT * FROM Train;
```

2. Create a cursor which displays the details of passengers:-

```
DECLARE

   P_FName

   Passenger.F_Name%TYPE;

P_SName

   Passenger.L_Name%TYPE;

Mob Passenger.Mobile%TYPE;

Address

   Passenger.Address%TYPE;

CURSOR c IS SELECT

   F_Name,L_Name,Mobile,Address

   FROM Passenger;

BEGIN OPEN c;

LOOP
```

```
   FETCH c INTO
   P_FName,P_SName,Mob,Address;
EXIT WHEN c%NOTFOUND;
dbms_output.put_line(P_FName||' '||P_SName||'  '||Mob||'  '||Address);
END LOOP;
CLOSE c;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
   DBMS_OUTPUT.PUT_LINE('NO SUCH PERSON EXISTS');
WHEN OTHERS THEN
   DBMS_OUTPUT.PUT_LINE('ERROR');
END;
/
```

3. Trigger to check no. of seats available is less than 0:-

```
CREATE OR REPLACE TRIGGER MIN_SEATS
BEFORE INSERT OR UPDATE OF AvailableSeats ON Train
FOR EACH ROW
BEGIN
   IF (:new.AvailableSeats < 0) THEN
      raise_application_error(-20000, 'Number of seats cannot be less than 0');
   END IF;
END;
/
```

4. Create a function which tells whether a specific train runs on that route or not :-

```
CREATE OR REPLACE FUNCTION CheckTrainRoute(
   p_T_No IN Train.T_No%TYPE,
   p_R_id IN Route.R_id%TYPE
) RETURN BOOLEAN
IS
   v_Count NUMBER;
BEGIN
   SELECT COUNT(*)
   INTO v_Count
   FROM Train
   WHERE T_No = p_T_No AND R_id = p_R_id;
   IF v_Count > 0 THEN
      RETURN TRUE;
   ELSE
      RETURN FALSE;
   END IF;
END CheckTrainRoute;
/
DECLARE
   v_Result BOOLEAN;
BEGIN
   v_Result := CheckTrainRoute(22429, 843);
   IF v_Result THEN
      DBMS_OUTPUT.PUT_LINE('Train 22429 runs on route 843.');
   ELSE
      DBMS_OUTPUT.PUT_LINE('Train 22429 does not run on route 843.');
   END IF;
```

```
    v_Result := CheckTrainRoute(12926, 321);
    IF v_Result THEN
        DBMS_OUTPUT.PUT_LINE('Train 12926 runs on route 321.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Train 12926 does not run on route 321.');
    END IF;
END;
/
```

5. Write a PL/SQL block to display ticket details for a specified train number.

```
DECLARE
    v_Train_No Train.T_No%TYPE := 22429; -- Example train number
    CURSOR c_tickets IS
        SELECT T_id, Timee, Datee, Price, C_id, AvailableSeats
        FROM Ticket
        WHERE T_No = v_Train_No;
BEGIN
    FOR ticket_rec IN c_tickets
    LOOP
        DBMS_OUTPUT.PUT_LINE('Ticket ID: ' || ticket_rec.T_id || ', Time: ' ||
TO_CHAR(ticket_rec.Timee, 'DD-MON-YY HH:MI:SS AM') ||
                             ', Date: ' || TO_CHAR(ticket_rec.Datee, 'DD-MON-YY') || ', Price: '
|| ticket_rec.Price ||
                             ', Passenger ID: ' || ticket_rec.C_id || ', Available Seats: ' ||
ticket_rec.AvailableSeats);
    END LOOP;
END;
/
```

6.Create a function which is used to insert details of new passenger.

```
CREATE SEQUENCE passenger_id_seq
   START WITH 13000
   INCREMENT BY 1
   NOCACHE
   NOCYCLE;


CREATE OR REPLACE FUNCTION InsertPassenger(
   p_Email IN VARCHAR2,
   p_Pass IN VARCHAR2,
   p_F_Name IN VARCHAR2,
   p_L_Name IN VARCHAR2,
   p_Mobile IN NUMBER,
   p_Address IN VARCHAR2,
   p_T_No IN Train.T_No%TYPE
) RETURN NUMBER
IS
   v_Passenger_ID Passenger.C_id%TYPE;
BEGIN
   SELECT passenger_id_seq.NEXTVAL INTO v_Passenger_ID FROM dual;
   INSERT INTO Passenger (C_id, Email, Pass, F_Name, L_Name, Mobile, Address,
T_No)
   VALUES (v_Passenger_ID, p_Email, p_Pass, p_F_Name, p_L_Name, p_Mobile,
p_Address, p_T_No);
   RETURN v_Passenger_ID;
EXCEPTION
   WHEN OTHERS THEN
      RETURN -1;
```

```
END;
/
DECLARE
   v_New_Passenger_ID Passenger.C_id%TYPE;
BEGIN
   v_New_Passenger_ID := InsertPassenger('RANDOM@GMAIL.COM', 'XYZZ',
'VAIBHAV', 'GARG', 1234567890, 'BATHINDA', 22429);
   IF v_New_Passenger_ID = -1 THEN
      DBMS_OUTPUT.PUT_LINE('Error: Failed to insert new passenger.');
   ELSE
      DBMS_OUTPUT.PUT_LINE('New Passenger ID: ' || v_New_Passenger_ID);
   END IF;
END;
/
```

7. Create a function to purchase a ticket for passenger by generating unique ticket no. and deducting the seats available.

```
CREATE SEQUENCE ticket_id_seq
   START WITH 2820000000
   INCREMENT BY 1
   NOCACHE
   NOCYCLE;
CREATE OR REPLACE FUNCTION buyTicket(
   p_Passenger_ID IN Passenger.C_id%TYPE,
   p_Train_ID IN Ticket.T_No%TYPE
) RETURN Ticket.T_id%TYPE
IS
   v_Ticket_ID Ticket.T_id%TYPE;
   v_TotalSeats Train.TotalSeats%TYPE;
```

```
BEGIN
    SELECT ticket_id_seq.NEXTVAL INTO v_Ticket_ID FROM dual;
    SELECT TotalSeats INTO v_TotalSeats
    FROM Train
    WHERE T_No = p_Train_ID;
    INSERT INTO Ticket (T_id, Timee, Datee, Price, C_id, AvailableSeats, T_No)
    VALUES (v_Ticket_ID, SYSTIMESTAMP, TRUNC(SYSDATE), 0,
p_Passenger_ID, v_TotalSeats, p_Train_ID);
    UPDATE Ticket
    SET AvailableSeats = AvailableSeats - 1
    WHERE T_No = p_Train_ID;
    RETURN v_Ticket_ID;
END buyTicket;
/
DECLARE
    v_Passenger_ID Passenger.C_id%TYPE := 13000;
    v_Train_ID Ticket.T_No%TYPE := 22429;
    v_New_Ticket_ID Ticket.T_id%TYPE;
BEGIN
    -- Call the buyTicket function with predefined passenger ID and train ID
    v_New_Ticket_ID := buyTicket(v_Passenger_ID, v_Train_ID);
    DBMS_OUTPUT.PUT_LINE('New ticket purchased successfully! Ticket ID: ' ||
v_New_Ticket_ID);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
SELECT * FROM Passenger;
SELECT * FROM Ticket;
```

OUTPUT SCREENSHOTS OF PLSQL CODE

1. Trigger to update train departure date to current date :-

```
CREATE OR REPLACE TRIGGER UpdateTrainDepartureTrigger
BEFORE INSERT OR UPDATE ON Train
FOR EACH ROW
BEGIN
    :NEW.DepartureTime := TRUNC(SYSDATE) + ( :NEW.DepartureTime - TRUNC(:NEW.DepartureTime) );
    :NEW.ArrivalTime := TRUNC(SYSDATE) + (:NEW.ArrivalTime - TRUNC (:NEW.ArrivalTime));
    :NEW.TotalSeats := 1000;
END;
/

UPDATE Train
SET DepartureTime = TRUNC(SYSDATE) + (DepartureTime - TRUNC(DepartureTime));

SELECT * FROM Train;
```

```
Trigger created.

5 row(s) updated.
```

T_NO	AVAILABLESEATS	T_NAME	DEPARTURETIME	ARRIVALTIME	TOTALSEATS	R_ID
22429	823	ASR SHTBDI EXP	06-MAY-24 07.20.00.000000 AM	06-MAY-24 11.30.30.000000 PM	1000	843
12926	60	AII ASR EXP	06-MAY-24 08.05.05.000000 AM	06-MAY-24 12.35.35.000000 PM	1000	321
12497	5	PASCHIM EXP	06-MAY-24 11.30.00.000000 AM	06-MAY-24 02.30.30.000000 PM	1000	654
20807	81	MEWAR EXP	06-MAY-24 07.20.00.000000 AM	06-MAY-24 10.30.30.000000 PM	1000	784
12013	450	HEMKUNT EXP	06-MAY-24 05.20.20.000000 PM	06-MAY-24 09.58.58.000000 PM	1000	591

2. Create a cursor which displays the details of passengers:-

```
DECLARE
    P_FName
    Passenger.F_Name%TYPE;
P_SName
    Passenger.L_Name%TYPE;
Mob Passenger.Mobile%TYPE;
Address
    Passenger.Address%TYPE;
CURSOR c IS SELECT
    F_Name,L_Name,Mobile,Address
    FROM Passenger;
BEGIN OPEN c;
LOOP
    FETCH c INTO
    P_FName,P_SName,Mob,Address;
EXIT WHEN c%NOTFOUND;
dbms_output.put_line(P_FName||' '||P_SName||'  '||Mob||'  '||Address);
END LOOP;
CLOSE c;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('NO SUCH PERSON EXISTS');
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('ERROR');
END;
/
```

```
Statement processed.
Sanchit Bansal  7009548392  Bathinda
Shererar Handa  7855623454  Lucknow
Vaibhav Garg  7865432123  Hyderabad
Sarika Loyal  9876523412  Mumbai
Maulik Kathuria  7895623490  Jaipur
Suvit Kumar  7007654323  Patiala
Rohan Bansal  7895343560  Amritsar
Naman Garg  7234568912  Mumbai
Mohit Singla  7654632142  Kota
Gurman Singh  78956267091  Rishikesh
```

3. . Trigger to check no. of seats available is less than 0:-

```
CREATE OR REPLACE TRIGGER MIN_SEATS
BEFORE INSERT OR UPDATE OF AvailableSeats ON Train
FOR EACH ROW
BEGIN
    IF (:new.AvailableSeats < 0) THEN
        raise_application_error(-20000, 'Number of seats cannot be less than 0');
    END IF;
END;
/
```

```
Trigger created.
```

4. Create a function which tells whether a specific train runs on that route or not :-

```
CREATE OR REPLACE FUNCTION CheckTrainRoute(
    p_T_No IN Train.T_No%TYPE,
    p_R_id IN Route.R_id%TYPE
) RETURN BOOLEAN
IS
    v_Count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_Count
    FROM Train
    WHERE T_No = p_T_No AND R_id = p_R_id;

    IF v_Count > 0 THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
END CheckTrainRoute;
/
```

```
DECLARE
    v_Result BOOLEAN;
BEGIN
    v_Result := CheckTrainRoute(22429, 843);
    IF v_Result THEN
        DBMS_OUTPUT.PUT_LINE('Train 22429 runs on route 843.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Train 22429 does not run on route 843.');
    END IF;
    v_Result := CheckTrainRoute(12926, 321);
    IF v_Result THEN
        DBMS_OUTPUT.PUT_LINE('Train 12926 runs on route 321.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Train 12926 does not run on route 321.');
    END IF;
END;
/
```

```
Function created.

Statement processed.
Train 22429 runs on route 843.
Train 12926 runs on route 321.
```

5. Write a PL/SQL block to display ticket details for a specified train number.

```
DECLARE
    v_Train_No Train.T_No%TYPE := 22429; -- Example train number
    CURSOR c_tickets IS
        SELECT T_id, Timee, Datee, Price, C_id, AvailableSeats
        FROM Ticket
        WHERE T_No = v_Train_No;
BEGIN
    FOR ticket_rec IN c_tickets
    LOOP
        DBMS_OUTPUT.PUT_LINE('Ticket ID: ' || ticket_rec.T_id || ', Time: ' || TO_CHAR(ticket_rec.Timee, 'DD-MON-YY HH:MI:SS AM') ||
                             ', Date: ' || TO_CHAR(ticket_rec.Datee, 'DD-MON-YY') || ', Price: ' || ticket_rec.Price ||
                             ', Passenger ID: ' || ticket_rec.C_id || ', Available Seats: ' || ticket_rec.AvailableSeats);
    END LOOP;
END;
/
```

```
Statement processed.
Ticket ID: 2819542664, Time: 22-APR-24 09:15:32 AM, Date: 22-APR-24, Price: 186, Passenger ID: 11248, Available Seats: 200
Ticket ID: 2819542685, Time: 21-APR-24 01:57:43 PM, Date: 24-APR-24, Price: 605, Passenger ID: 11543, Available Seats: 400
```

6. Create a function which is used to insert details of new passenger.

```
CREATE SEQUENCE passenger_id_seq
    START WITH 13000
    INCREMENT BY 1
    NOCACHE
    NOCYCLE;
```

```
Sequence created.
```

```
CREATE OR REPLACE FUNCTION InsertPassenger(
    p_Email IN VARCHAR2,
    p_Pass IN VARCHAR2,
    p_F_Name IN VARCHAR2,
    p_L_Name IN VARCHAR2,
    p_Mobile IN NUMBER,
    p_Address IN VARCHAR2,
    p_T_No IN Train.T_No%TYPE
) RETURN NUMBER
IS
    v_Passenger_ID Passenger.C_id%TYPE;
BEGIN
    SELECT passenger_id_seq.NEXTVAL INTO v_Passenger_ID FROM dual;
    INSERT INTO Passenger (C_id, Email, Pass, F_Name, L_Name, Mobile, Address, T_No)
    VALUES (v_Passenger_ID, p_Email, p_Pass, p_F_Name, p_L_Name, p_Mobile, p_Address, p_T_No);
    RETURN v_Passenger_ID;
EXCEPTION
    WHEN OTHERS THEN
        RETURN -1;
END;
/

DECLARE
    v_New_Passenger_ID Passenger.C_id%TYPE;
BEGIN
    v_New_Passenger_ID := InsertPassenger('RANDOM@GMAIL.COM', 'XYZZ', 'VAIBHAV', 'GARG', 1234567890, 'BATHINDA', 22429);
    IF v_New_Passenger_ID = -1 THEN
        DBMS_OUTPUT.PUT_LINE('Error: Failed to insert new passenger.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('New Passenger ID: ' || v_New_Passenger_ID);
    END IF;
END;
/
```

```
Function created.

Statement processed.
New Passenger ID: 13000
```

7. Create a function to purchase a ticket for passenger by generating unique ticket no. and deducting the seats available.

```
CREATE SEQUENCE ticket_id_seq
    START WITH 2820000000
    INCREMENT BY 1
    NOCACHE
    NOCYCLE;

CREATE OR REPLACE FUNCTION buyTicket(
    p_Passenger_ID IN Passenger.C_id%TYPE,
    p_Train_ID IN Ticket.T_No%TYPE
) RETURN Ticket.T_id%TYPE
IS
    v_Ticket_ID Ticket.T_id%TYPE;
    v_TotalSeats Train.TotalSeats%TYPE;
BEGIN
    SELECT ticket_id_seq.NEXTVAL INTO v_Ticket_ID FROM dual;
    SELECT TotalSeats INTO v_TotalSeats
    FROM Train
    WHERE T_No = p_Train_ID;
    INSERT INTO Ticket (T_id, Timee, Datee, Price, C_id, AvailableSeats, T_No)
    VALUES (v_Ticket_ID, SYSTIMESTAMP, TRUNC(SYSDATE), 0, p_Passenger_ID, v_TotalSeats, p_Train_ID);
    UPDATE Ticket
    SET AvailableSeats = AvailableSeats - 1
    WHERE T_No = p_Train_ID;
    RETURN v_Ticket_ID;
END buyTicket;
```

```
Sequence created.

Function created.
```

```
DECLARE
    v_Passenger_ID Passenger.C_id%TYPE := 13000;
    v_Train_ID Ticket.T_No%TYPE := 22429;
    v_New_Ticket_ID Ticket.T_id%TYPE;
BEGIN
    -- Call the buyTicket function with predefined passenger ID and train ID
    v_New_Ticket_ID := buyTicket(v_Passenger_ID, v_Train_ID);
    DBMS_OUTPUT.PUT_LINE('New ticket purchased successfully! Ticket ID: ' || v_New_Ticket_ID);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/
SELECT * FROM Passenger;
SELECT * FROM Ticket;
```

C_ID	EMAIL	PASS	F_NAME	L_NAME	MOBILE	ADDRESS	T_NO
11248	sbansal@gmail.com	-	Sanchit	Bansal	7009548392	Bathinda	12926
11432	shanda@yahoo.in	-	Shererar	Handa	7855623454	Lucknow	12013
11741	vgarg@gmail.com	-	Vaibhav	Garg	7865432123	Hyderabad	22429
11875	sloyal@gmail.com	-	Sarika	Loyal	9876523412	Mumbai	12497
11657	mkathuria@gmail.com	-	Maulik	Kathuria	7895623490	Jaipur	20807
11775	skumar@gmail.com	-	Suvit	Kumar	7007654323	Patiala	12926
11543	rbansal@gmail.com	-	Rohan	Bansal	7895343560	Amritsar	22429
11331	ngarg@gmail.com	-	Naman	Garg	7234568912	Mumbai	12497
11345	msingla@gmail.com	-	Mohit	Singla	7654632142	Kota	20807
11871	gsingh@gmail.com	-	Gurman	Singh	78956267091	Rishikesh	12013
13000	RANDOM@GMAIL.COM	XYZZ	VAIBHAV	GARG	1234567890	BATHINDA	22429

T_ID	TIMEE	DATEE	PRICE	C_ID	AVAILABLESEATS	T_NO
2819542664	22-APR-24 09.15.32.000000 AM	22-APR-24	186	11248	199	22429
2819542612	14-APR-24 10.45.44.000000 AM	14-APR-24	125	11432	300	20807
2819542687	26-APR-24 10.10.55.000000 PM	26-APR-24	550	11741	400	12926
2819542655	10-APR-24 03.40.21.000000 PM	10-APR-24	240	11875	500	12497
2819542694	07-APR-24 01.57.43.000000 PM	15-APR-24	605	11657	400	12013
2819542692	09-APR-24 01.57.43.000000 PM	21-APR-24	605	11775	400	12926
2819542685	21-APR-24 01.57.43.000000 PM	24-APR-24	605	11543	399	22429
2819542613	23-APR-24 01.57.43.000000 PM	26-APR-24	605	11331	400	12497
2819542602	03-APR-24 01.57.43.000000 PM	29-APR-24	605	11345	400	20807
2819542680	05-APR-24 01.57.43.000000 PM	30-APR-24	605	11871	400	12013
2820000000	05-MAY-24 08.26.01.962131 AM	05-MAY-24	0	13000	999	22429

CONCLUSION

The system developed is able to meet all the basic requirements. The management of the records (trains, passengers and tickets) will be also benefited from the proposed system, as it will automate the whole procedure, which will reduce the workload.

The system is flexible enough for future modifications. Though most part of the system is supposed to act in the background, efforts have been made to make the foreground interaction with user(owner) as smooth as possible.