



# TRAIN SYSTEM

Railway Reservation System Functionalities

## INTRODUCTION TO DATABASE

### ABSTRACT

This ER (Entity Relationship) diagram represents the model of Railway reservation system entity. The entity relationship diagram of railway reservation system shows all the visual instrument of database tables and the relation between seats availability. Train schedule, ticket, passenger etc.

Nusrat Alam Chaiti

18-37417-1

Tamzid Ahmed

18-37503-1

**COURSE TEACHER:**

**RIFAT TASNIM ANANNYA**

## \\ INDEX //

Topic Name	Page
Scenario	2
ER Diagram	3
Normalization	4
Table Creation	6
Sub Query	11
Joining	15
Constraint	19

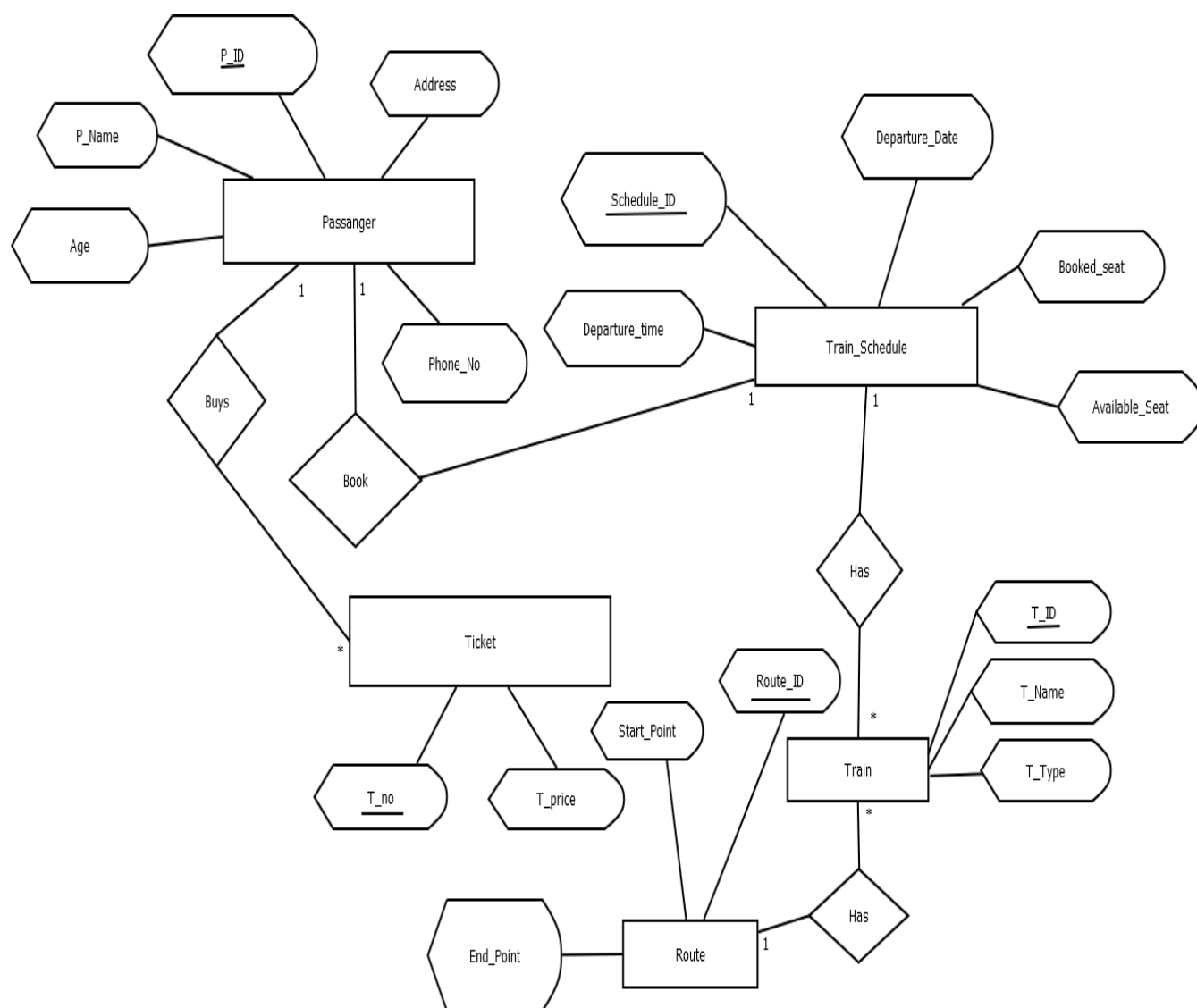
## SCENARIO:

This ER (Entity Relationship) diagram represents the model of Railway reservation system entity. The entity relationship diagram of railway reservation system shows all the visual instrument of database tables and the relation between seats availability. Train schedule, ticket, passenger etc. It used structure data and to define the relationships between structure data groups of railway reservation system functionalities. The main entities of railway reservation system and passenger, train schedule, train, route, ticket.

Railways Reservation, System entities and their attributes:

1. Passenger : P\_ID, P\_name, Address, AGE, Phone\_No
2. Train Schedule: Schedule\_ID, Departure\_Date, Departure\_time,  
Avaiable\_Seat, Booked\_Seat
3. Train : T\_ID, T\_NAME, T\_TYPE
4. Route : Route\_ID, Start\_Point, End\_Point
5. Ticket : T\_NO, T\_PRICE

# ENTITY RELATIONSHIP DIAGRAM



# NORMALIZATION

Buys:

1NF: no multivalued attribute

2NF: table1:p\_name,p\_id,address,age,phone\_no

Table2:ticket\_no,ticket\_price

3NF: table1: p\_name,p\_id,address,age,phone\_no

Table2:ticket\_no,ticket\_price,p\_id

Tables are given below:

Table1: p\_name,p\_id,address,age,phone\_no

Table2: ticket\_no,ticket\_price,p\_id

BOOK

1NF:no multivalued attributes

2NF: table1: p\_name,p\_id,address,age,phone\_no

Table2: schedule\_id,departure\_date,departure\_time,available seat,booked seat

3NF: table 1: p\_name,p\_id,address,age,phone\_no

Table2: schedule\_id,departure\_date,departure\_time,available seat,booked seat,p\_id

Tables are given below:

table 1: p\_name,p\_id,address,age,phone\_no

Table2: schedule\_id,departure\_date,departure\_time,available seat,booked seat,p\_id

HAS

1NF:no multivalued attribute

2NF: table1: schedule\_id,departure\_date,departure\_time,available\_seat,booked\_seat

Table2:t\_id,t\_name,t\_type

3NF: table1: schedule\_id,departure\_date,departure\_time,available\_seat,booked\_seat

Table2:t\_id,t\_name,t\_type,schedule\_id

Tables are given below:

table1: schedule\_id,departure\_date,departure\_time,available seat,booked seat

Table2:t\_id,t\_name,t\_type,schedule\_id

HAS

1NF: no multivalued attribute

2NF: table1 :t\_id,t\_name,t\_type

Table2: route\_id,start\_point,end\_point

3NF : table1 :t\_id,t\_name,t\_type

Table2: route\_id,start\_point,end\_point,t\_id

Tables are given below:

table1 :t\_id,t\_name,t\_type

Table2: route\_id,start\_point,end\_point,t\_id

Final tables are given below:

Table1: p\_name,p\_id,address,age,phone\_no

Table2: ticket\_no,ticket\_price,p\_id

Table3: schedule\_id,departure\_date,departure\_time,available seat,booked seat,p\_id

Table4:t\_id,t\_name,t\_type,schedule\_id

Table5: route\_id,start\_point,end\_point,t\_id

## TABLE CREATION

Passenger:

```
CREATE TABLE PASSANGER
(
P_ID number(10) NOT NULL ,
P_NAME varchar2(20) ,
AGE number(3),
Phone No number(12),
Address varchar2(100),
Primary key(P_ID)
)
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PASSENGER	P_ID	Number	-	10	0	1	-	-	-
	P_NAME	Varchar2	20	-	-	-	✓	-	-
	AGE	Number	-	3	0	-	✓	-	-
	PHONE_NO	Number	-	12	0	-	✓	-	-
	ADDRESS	Varchar2	100	-	-	-	✓	-	-
1 - 5									

After Insertion:

P_ID	P_NAME	AGE	PHONE_NO	ADDRESS
1	Tamzid	20	18999	Malibag
2	Chaiti	18	18956	Mirpur
3	Nusrat	12	18953	Mirpur
4	Fahim	60	189378	Kuril

Train Schedule:

```
CREATE TABLE TRAIN_SCHEDULE
(
STATUS_ID number(10) NOT NULL,

Departure_Date date(5),

Departure_Time number(2) ,
Avaialbe_seat number(3),
Booked_Seat number(3),

P_ID NUMBER(10) NOT NULL,

Primary key(Status_ID),

Foreign Key(P_ID) References Passenger(P_ID)

)
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>TRAIN_SCHEDULE</u>	<u>STATUS_ID</u>	Number	-	10	0	1	-	-	-
	<u>DEPARTURE_DATE</u>	Date	7	-	-	-	✓	-	-
	<u>DEPARTURE_TIME</u>	Varchar2	10	-	-	-	✓	-	-
	<u>AVAILABLE_SEAT</u>	Number	-	3	0	-	✓	-	-
	<u>BOOKED_SEAT</u>	Number	-	3	0	-	✓	-	-
	<u>P_ID</u>	Number	-	10	0	-	-	-	-
1 - 6									

After Insertion:

STATUS_ID	DEPARTURE_DATE	DEPARTURE_TIME	AVAILABLE_SEAT	BOOKED_SEAT	P_ID
1	18-NOV-18	9:00	300	200	1
2	18-NOV-18	11:00	200	300	2
3	19-NOV-18	9:00	400	100	3
4	19-NOV-18	11:00	100	400	4



Train:

```
CREATE TABLE TRAIN
(
  T_ID number(10) NOT NULL,
  T_NAME VARCHAR2(20),
  T_TYPE VARCHAR2(10),
  STATUS_ID NUMBER(10) NOT NULL,
  PRIMARY KEY(T_ID),
  FOREIGN KEY(STATUS_ID) REFERENCES TRAIN_SCHEDULE (STATUS_ID)
)
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRAIN	T_ID	Number	-	10	0	1	-	-	-
	T_NAME	Varchar2	20	-	-	-	✓	-	-
	T_TYPE	Varchar2	10	-	-	-	✓	-	-
	STATUS_ID	Number	-	10	0	-	-	-	-
									1 - 4

After Insertion:

T_ID	T_NAME	T_TYPE	STATUS_ID
1	JAYANTI	1ST CLASS	1
2	SUBORNO	1ST CLASS	2
3	HEMONTA	2ND CLASS	3
4	BORNO	2ND CLASS	4

ROUTE:

```
CREATE TABLE ROUTE
(
ROUTE_ID number(10) NOT NULL,
START_POINT VARCHAR2(20),
END_POINT VARCHAR2(10),
T_ID NUMBER(10) NOT NULL,
PRIMARY KEY(ROUTE_ID),
FOREIGN KEY(T_ID) REFERENCES TRAIN (T_ID)
)
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ROUTE	ROUTE_ID	Number	-	10	0	1	-	-	-
	START_POINT	Varchar2	20	-	-	-	✓	-	-
	END_POINT	Varchar2	10	-	-	-	✓	-	-
	T_ID	Number	-	10	0	-	-	-	-
									1 - 4

After Insertion:

ROUTE_ID	START_POINT	END_POINT	T_ID
1	KAMLAPUR	SYLHET	1
2	SYLHET	KAMLAPUR	2
3	RAJSHAHI	KAMLAPUR	3
4	KAMLAPUR	RAJSHAHI	4

TICKET :

```
CREATE TABLE TICKET
(
TICKET_NO number(10) NOT NULL,
TICKET PRICE NUMBER(5) NOT NULL,
P_ID NUMBER(10) NOT NULL,
PRIMARY KEY(TICKET_NO),
Foreign Key(P_ID) References Passenger(P_ID)
)
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TICKET	TICKET_NO	Number	-	10	0	1	-	-	-
	TICKET_PRICE	Number	-	5	0	-	-	-	-
	P_ID	Number	-	10	0	-	-	-	-
1 - 3									

After Insertion:

TICKET_NO	TICKET_PRICE	P_ID
1	1500	1
2	1500	2
3	800	3
4	800	4

# SUB-QUERY

1. Write a query to display P\_NAME where age is less than the person whose address is kuril.

Ans.

**Home > SQL > SQL Commands**

☒ Autocommit   Display 10000 ▼   **Save**   **Run**

```
SELECT P_NAME
FROM PASSENGER
WHERE AGE < (
  SELECT AGE
  FROM PASSENGER
  WHERE ADDRESS = 'Kuril'
)
```

**Results**   Explain   Describe   Saved SQL   History

P_NAME
Tamzid
Chaiti
Nusrat

3 rows returned in 0.00 seconds   [CSV Export](#)

2. Write a query to display available seat and booked seat where departure date is after '18-nov-18'.

Ans.

**Home > SQL > SQL Commands**

☒ Autocommit   Display 10000 ▼   **Save**   **Run**

```
SELECT available_seat,booked_seat
FROM train_schedule
WHERE departure_date>ALL
(SELECT departure_date
FROM train_schedule
WHERE departure_date='18-NOV-18'
)
```

**Results**   Explain   Describe   Saved SQL   History

AVAILABLE_SEAT	BOOKED_SEAT
400	100
100	400

2 rows returned in 0.00 seconds   [CSV Export](#)

3. WRITE A QUERY TO DISPLAY PNAME AND PHONE NO WHERE P\_ID IS EQUAL TO THE PERSON WHOSE AGE IS GREATER THAN 20.

**Home > SQL > SQL Commands**

☒ Autocommit   Display 10000 ▼   **Save**   **Run**

```
select p_name,phone_no
from PASSENGER
WHERE P_ID=( SELECT P_ID
              FROM PASSENGER
              WHERE AGE>20)
```

**Results**   Explain   Describe   Saved SQL   History

P_NAME	PHONE_NO
Fahim	189378

1 rows returned in 0.00 seconds   [CSV Export](#)

4. WRITE A QUERY TO DISPLAY ALL INFORMATION OF TRAIN WHERE TRAIN TYPE IS EQUAL TO THE NAME OF THE TRAIN WHICH IS BORNO OR HEMONTO.

Home > SQL > SQL Commands

☒ Autocommit   Display 10000   Save   Run

```
select *
FROM TRAIN
WHERE T_TYPE= ALL( SELECT T_TYPE
                    FROM TRAIN
                    WHERE T_NAME='BORNO'
                        OR T_NAME='HEMONTTO')
```

Results   Explain   Describe   Saved SQL   History

T_ID	T_NAME	T_TYPE	STATUS_ID
3	HEMONTTO	2ND CLASS	3
4	BORNO	2ND CLASS	4

2 rows returned in 0.00 seconds   [CSV Export](#)

# JOINING

## Equi Join:

1.WRITE A QUERY TO DISPLAY THE TRAIN NAME, TRAIN STATUS\_ID, AND DEPARTURE TIME FOR ALL TRAIN.

Ans.

Home > SQL > SQL Commands

☒ Autocommit Display 10000 ▼

Save

Run

```
select T.T_NAME,T.STATUS_ID,S.DEPARTURE_TIME
FROM TRAIN T,TRAIN_SCHEDULE S
WHERE T.STATUS_ID=S.STATUS_ID|
```

**Results** Explain Describe Saved SQL History

T_NAME	STATUS_ID	DEPARTURE_TIME
JAYANTI	1	9:00
SUBORNO	2	11:00
HEMONTA	3	9:00
BORNO	4	11:00

4 rows returned in 0.00 seconds

[CSV Export](#)



2.CREATE A QUERY TO DISPLAY THE START POINT,END POINT ,TRAIN ID,TRAIN NAME FOR ALL ROUTES  
WHERE TRAIN NAME IS 'BORNO'.

Ans.

Home > SQL > SQL Commands

☒ Autocommit Display 10000 ▼ Save Run

```
SELECT E.START_POINT,E.END_POINT,E.T_ID,P.T_NAME
FROM ROUTE E JOIN TRAIN P
ON(E.T_ID=P.T_ID)
WHERE P.T_NAME='BORNO'
```

**Results** Explain Describe Saved SQL History

START_POINT	END_POINT	T_ID	T_NAME
KAMLAPUR	RAJSHAHI	4	BORNO

1 rows returned in 0.00 seconds [CSV Export](#)

## Self Join:

3.CREATE A QUERY TO DISPLAY THE TICKET NO AND TICKET PRICE OF ANY PASSENGER WHOSE TICKET PRICE IS LESS THAN 1500.

Ans.

Home > SQL > **SQL Commands**

☒ Autocommit   Display 10000 ▼   **Save**   **Run**

```
select T.TICKET_NO,T.TICKET_PRICE
FROM TICKET T,TICKET F
WHERE F.TICKET_NO=3
AND T.TICKET_PRICE<1500
```

---

**Results**   Explain   Describe   Saved SQL   History

TICKET_NO	TICKET_PRICE
3	800
4	800

2 rows returned in 0.00 seconds   [CSV Export](#)

## Non Equi Join:

4.Display all information of train schedule table where available seat is larger than booked seat

Ans.

Home > SQL > **SQL Commands**

☒ Autocommit Display 10000 Save Run

```
select *
from train_schedule TS,train_schedule TM
WHERE(TS.AVAILABLE_SEAT>TM.BOOKED_SEAT) AND (TS.P_ID=TM.P_ID)|
```

**Results** Explain Describe Saved SQL History

STATUS_ID	DEPARTURE_DATE	DEPARTURE_TIME	AVAILABLE_SEAT	BOOKED_SEAT	P_ID
1	18-NOV-18	9:00	300	200	1
3	19-NOV-18	9:00	400	100	3

2 rows returned in 0.00 seconds [CSV Export](#)

# CONSTRAINT

FOREIGN KEY ,PRIMARY AND NOT NULL CONSTRAINT:

TRAIN SCHEDULE:

```

Home > SQL > SQL Commands

Autocommit Display 10000 Save Run

CREATE TABLE TRAIN_SCHEDULE
(
  STATUS_ID number(10) NOT NULL,
  Departure_Date date(5),
  Departure_Time number(2) ,
  Available_seat number(3),
  Booked_Seat number(3),
  P_ID NUMBER(10) NOT NULL,
  Primary key(Status_ID),
  Foreign Key(P_ID) References Passenger(P_ID)
)

```

TRAIN:

```

Home > SQL > SQL Commands

Autocommit Display 10000 Save Run

CREATE TABLE TRAIN
(
  T_ID number(10) NOT NULL,
  T_NAME VARCHAR2(20),
  T_TYPE VARCHAR2(10),
  STATUS_ID NUMBER(10) NOT NULL,
  PRIMARY KEY(T_ID),
  FOREIGN KEY(STATUS_ID) REFERENCES TRAIN_SCHEDULE (STATUS_ID)
)

```

TICKET:

```

Home > SQL > SQL Commands

Autocommit Display 10000 Save Run

CREATE TABLE TICKET
(
  TICKET_NO number(10) NOT NULL,
  TICKET PRICE NUMBER(5) NOT NULL,
  P_ID NUMBER(10) NOT NULL,
  PRIMARY KEY(TICKET_NO),
  Foreign Key(P_ID) References Passenger(P_ID)
)

```

**ROUTE:**

Home > SQL > **SQL Commands**

☒ Autocommit   Display 10000 ▼   **Save**   **Run**

```
CREATE TABLE ROUTE
(
  ROUTE_ID number(10) NOT NULL,
  START_POINT VARCHAR2(20),
  END_POINT VARCHAR2(10),
  T_ID NUMBER(10) NOT NULL,
  PRIMARY KEY(ROUTE_ID),
  FOREIGN KEY(T_ID) REFERENCES TRAIN (T_ID)
)
```

## CHECK CONSTRAINT:

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

☒ Autocommit Display 10000 Save Run

```
ALTER TABLE PASSENGER ADD CONSTRAINT VERIFYING_AGE CHECK(AGE>10)
```

**Results** Explain Describe Saved SQL History

Table altered.

0.01 seconds

Language: en-us

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

## DISABLE CONSTRAINT:

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

☒ Autocommit Display 10000 Save Run

```
ALTER TABLE PASSENGER DISABLE CONSTRAINT VERIFYING_AGE
```

**Results** Explain Describe Saved SQL History

Table altered.

0.00 seconds

## ENABLE CONSTRAINT:

Home > SQL > SQL Commands

☒ Autocommit Display 10000 Save Run

```
ALTER TABLE PASSENGER ENABLE CONSTRAINT VERIFYING_AGE
```

**Results** Explain Describe Saved SQL History

Table altered.

0.00 seconds