



American International University-Bangladesh

RAILWAY TICKETING SYSTEM



Course

Introduction to Database (I)

Date of Submission:

17 APRIL, 2022

Presented By:

Nazmul Hassan (21-45293-2)

Hasnatur Rahman Khan (21-45355-2)

Rafi Mahmud (21-45294-2)

Pritom Chowhan (21-45356-2)

Presented To:

DR. MD ALAMGIR KABIR

Table of contents

Content	Page no.
1. Title and description	3-4
2. ERD	5
3. Normalization	6-8
4. Table creation & Data insertion	9-16
5. Report Query	16-20

Project Description

About the project

This project is about creating the database about Railway Database Management System. The railway database system facilitates the passengers to inquire about the trains available based on Train status, station, and booking of tickets and facilitates many other things.

There are several counters for reserving seats, and one can easily make reservations and get tickets. But if the passengers want to get tickets online, they can simply pay online and get tickets.

Problem

In huge problem in our country is standing in long lines to get tickets. That costs our most valuable time which is very annoying and time-consuming.

But We find a solution!!

The use of a ticket card to get tickets. It is convenient for all passengers. Because in this ticket getting process there have no other 3rd party or someone else. It's fully automatic and a time saver technology.

Each passenger must have a ticket card and per card can afford or access only one ticket at that moment.

Conditions

The railway department is run by some rules. If the passenger doesn't purchase a ticket, they can't enter the station. Then if the passenger doesn't get any seats on the train, they must get a standing ticket.

Prosses descriptions

Passengers can book their tickets for their journey. Passengers are provided with their name, id, gender, and age in the card which is used to purchase a train ticket. On this card, the balance and id are provided. The card id is unique.

They want to buy a ticket. Then they must go to the counter/booth. Then the help of the card they can easily get a ticket which saves their time.

The train name, train no, coach no, journey date, seat no, train time, Station name, booking date, journey date, platform no and the ticket price are provided with the ticket.

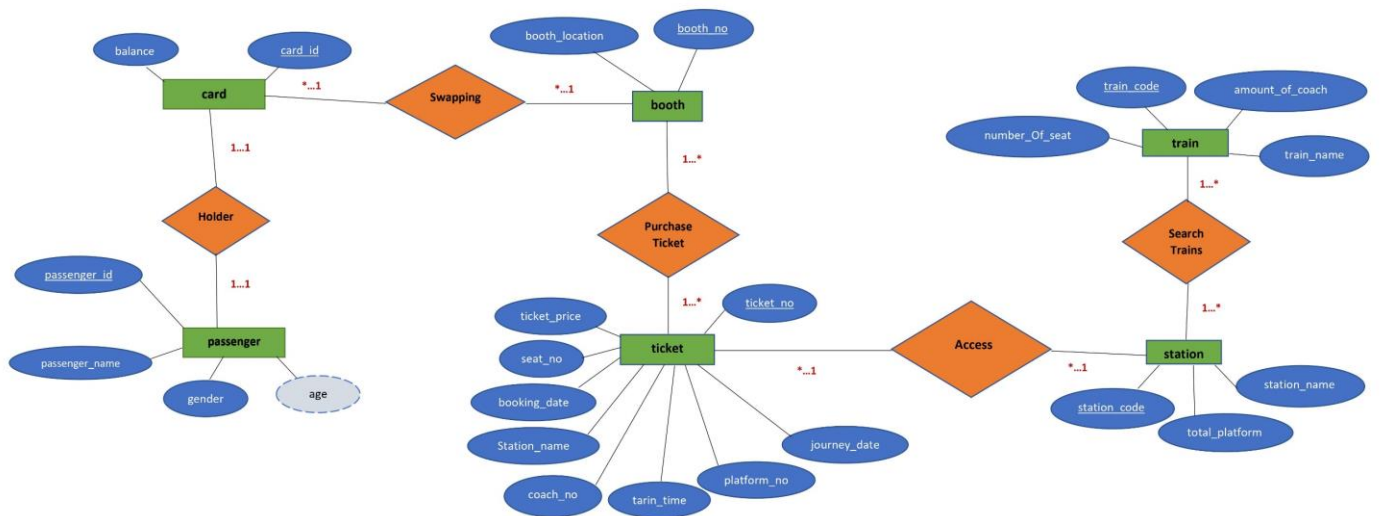
When they get a ticket, they have access to enter the station, and with the help of the ticket, they can find their desired train.

The ticket also has access to get out of the station. Which is more convenient for the user or the passenger.

The list of entities and attributes:

Entities	Attributes
Passenger	<u>passenger_id</u> passenger_name gender age
Train	<u>train_code</u> train_name amount_of_coach number_of_seat
Station	station_name <u>station_code</u> total_platform
Ticket	<u>ticket_no</u> coach_no booking_date journey_date seat_no train_time station_name platform_no ticket_price
Booth/counter	booth_locations <u>booth_number</u>
Card	<u>card_id</u> balance

ER Diagram



Normalization

Holder

All attributes are

Holder (passenger_id, passenger_name, gender, age,
card_id, balance)

1nf	No multivalued attributes
2nf	<u>passenger_id</u> , passenger_name, gender, age
	<u>card_id</u> , balance
3nf	<u>passenger_id</u> , passenger_name
	<u>card_id</u> , gender, age
	<u>card_id</u> , balance

So table from Holder are

passenger_id, passenger_name

card_id, gender, age

card_id, gender, age

Swapping

All attributes are

Swapping (card_id, balance,
booth_location, booth_no)

1nf	No multivalued attributes
2nf	<u>card_id</u> , balance
	booth location, <u>booth_no</u>
3nf	No transitive dependencies found

Table from Swapping are

card_id, balance

booth_location, booth_no

Purchase ticket

All attributes are

Purchase ticket (booth location, booth_no,
ticket_price, ticket_no, train_name, train_code(fk), coach_no,
journey_date, seat_no,
train_time, booking_date)

1nf	No multivalued attributes
2nf	<u>booth_no</u> , booth location
	<u>ticket_no</u> , train_name, train_code, coach_no, journey_date, seat_no, train_time, booking_date, ticket_price
3nf	<u>booth_no</u> , booth_location
	<u>ticket_no</u> , booth_no, booth_location
	<u>ticket_no</u> , train_name, train_code, coach_no, journey_date, seat_no, train_time, booking_date, ticket_price

Table from Purchase ticket

booth_no, booth_location

ticket_no, booth_no, booth_location

ticket_no, train_name, train_code, coach_no, journey_date, seat_no, train_time,
booking_date, ticket_price

Access

All attributes are

Access (ticket_no, train_name, train_code, coach_no, journey_date, seat_no,
train_time, booking_date, ticket_price,
station_code, station_name, total_platform)

1nf	No multivalued attributes
2nf	<u>ticket_no</u> , train_name, train_code, coach_no, journey_date, seat_no, train_time, booking_date, ticket_price
	<u>station_code</u> , station_name, total_platform
3nf	<u>ticket_no</u> , train_name, seat_no, coach_no journey_date, train_time, booking_date, ticket_price
	<u>ticket_no</u> , station_name, station_code
	<u>station_code</u> , station_name, total_platform

Tables from Access are

ticket_no, train_name, seat_no, coach_no journey_date, train_time, booking_date, ticket_price

ticket_no, station_name, station_code

station_code, station_name, total_platform

Search Train

All attributes are

search train (station_code, station_name, total_platform,

train_code, train_name, amount_of_coach, number_of_seat)

1nf	No multivalued attributes
2nf	<u>station_code</u> , station_name, total_platform
	<u>train_code</u> , train_name, amount_of_coach, number_of_seat
3nf	<u>station_code</u> , station_name
	<u>Station_code</u> , train_code, train_name, amount_of_coach, number_of_seat
	<u>train_code</u> , train_name, amount_of_coach, number_of_seat

Tables from search train are

station_code, station_name

Station_code, train_code, train_name, amount_of_coach, number_of_seat

train_code, train_name, amount_of_coach, number_of_seat

Table Creations and Data Insertion

1. Train Table Create:

```
create table train(  
train_code number not null,  
train_name varchar2(30),  
amount_of_coach number,  
number_of_seat number,  
constraint pk_train primary key(train_code)  
);
```

```
insert into train values(001, 'Shuborno', 9, 1000);  
insert into train values(002, 'Shemol', 7, 700);  
insert into train values(003, 'Duronto', 7, 700);  
insert into train values(004, 'Ekota', 5, 500);  
insert into train values(005, 'Mohanogor', 6, 600);  
insert into train values(006, 'Egaroshindhur', 8, 800);  
insert into train values(007, 'Kishorganj express', 5, 400);  
insert into train values(008, 'Upokul', 7, 700);  
insert into train values(009, 'Upobon', 8, 800);  
insert into train values(010, 'Kalni', 9, 850);
```

The screenshot shows the Railway database management tool interface. The top section displays a series of SQL queries for creating a 'train' table and inserting data. The bottom section shows the results of a 'select * from train;' query, which lists 10 train records with their codes, names, coach amounts, and seat numbers.

```

1 create table train(
2   train_code number not null,
3   train_name varchar2(30),
4   amount_of_coach number,
5   number_of_seat number,
6   constraint pk_train primary key(train_code)
7 );
8
9 insert into train values(001, 'Shuborno', 9, 1000);
10 insert into train values(002, 'Shemol', 7, 700);
11 insert into train values(003, 'Durgam', 7, 700);
12 insert into train values(004, 'Ekota', 5, 500);
13 insert into train values(005, 'Mohanogor', 6, 600);
14 insert into train values(006, 'Egaroshindhur', 8, 800);
15 insert into train values(007, 'Kishorganj express', 5, 400);
16 insert into train values(008, 'Upokul', 7, 700);
17 insert into train values(009, 'Upobon', 8, 800);
18 insert into train values(010, 'Kaini', 9, 850);
19
20 select * from train;
21

```

TRAIN_CODE	TRAIN_NAME	AMOUNT_OF_COACH	NUMBER_OF_SEAT
1	1 Shuborno	9	1000
2	2 Shemol	7	700
3	3 Durgam	7	700
4	4 Ekota	5	500
5	5 Mohanogor	6	600
6	6 Egaroshindhur	8	800
7	7 Kishorganj express	5	400
8	8 Upokul	7	700
9	9 Upobon	8	800
10	10 Kaini	9	850

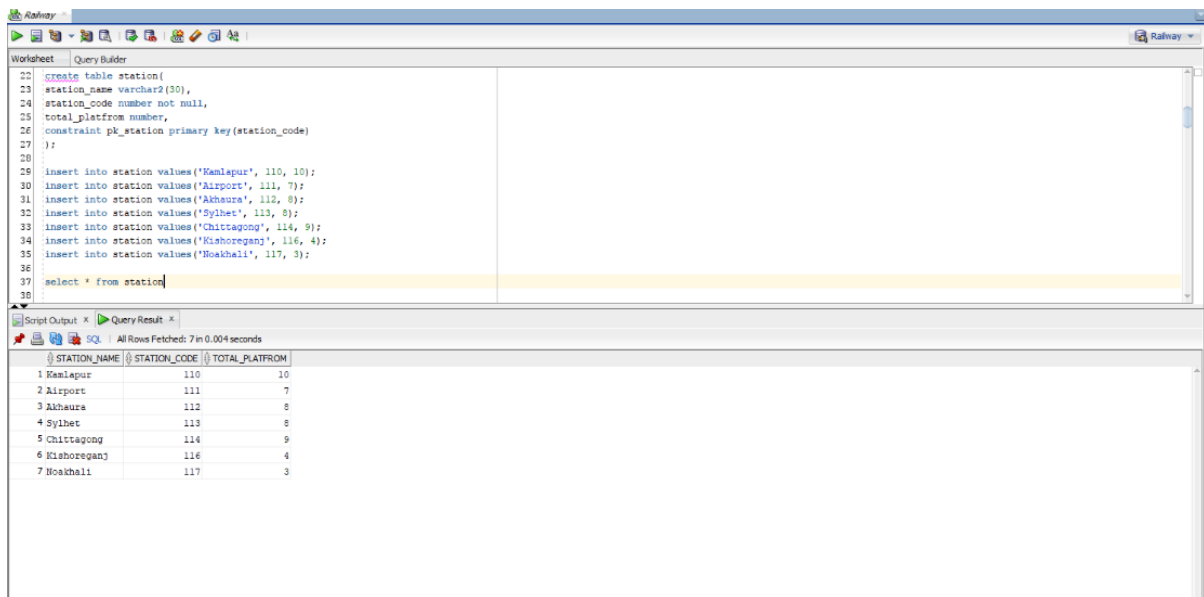
2. Station Table Create:

```

create table station(
station_name varchar2(30),
station_code number not null,
total_platfrom number,
constraint pk_station primary key(station_code)
);

insert into station values('Kamlapur', 110, 10);
insert into station values('Airport', 111, 7);
insert into station values('Akhaura', 112, 8);
insert into station values('Sylhet', 113, 8);
insert into station values('Chittagong', 114, 9);
insert into station values('Kishoreganj', 116, 4);
insert into station values('Noakhali', 117, 3);

```



3. Ticket Table Create:

```

create table ticket(
ticket_no number(25) not null,
train_name varchar2(30),
train_code number,
coach_no number,
booking_date date,
journey_date date,
seat_no number,
train_time varchar2(20),
platform_no number,
ticket_price number,
constraint pk_ticket primary key(ticket_no),
constraint fk_ticket_train_code foreign key (train_code)
references train (train_code));

```

insert into ticket values(601, 'Shuborno', 001, 3, TO_DATE('10-01-2022', 'DD-MM-YYYY'), TO_DATE('22-01-2022', 'DD-MM-YYYY'), 625, '8:00', 'Airport', 3, 400);

insert into ticket values(512, 'Shuborno', 001, 2, TO_DATE('11-01-2022', 'DD-MM-YYYY'), TO_DATE('20-01-2022', 'DD-MM-YYYY'), 430, '7:30', 'Chittagong', 3, 400);

insert into ticket values(224, 'Shemol', 002, 2, TO_DATE('12-01-2022', 'DD-MM-YYYY'), TO_DATE('18-01-2022', 'DD-MM-YYYY'), 233, '9:00', 'Airport', 1, 350);

insert into ticket values(321, 'Duronto', 003, 5, TO_DATE('10-01-2022', 'DD-MM-YYYY'), TO_DATE('21-01-2022', 'DD-MM-YYYY'), 20, '7:00', 'Kamlapur', 2, 400);

insert into ticket values(412, 'Ekota', 004, 4, TO_DATE('18-01-2022', 'DD-MM-YYYY'), TO_DATE('28-01-2022', 'DD-MM-YYYY'), 100, '12:00', 'Airport', 4, 300);

insert into ticket values(21, 'Mohanogor', 005, 2, TO_DATE('19-01-2022', 'DD-MM-YYYY'), TO_DATE('30-01-2022', 'DD-MM-YYYY'), 132, '10:00', 'Cantonment', 2, 300);

insert into ticket values(142, 'Kalni', 010, 7, TO_DATE('02-02-2022', 'DD-MM-YYYY'), TO_DATE('08-02-2022', 'DD-MM-YYYY'), 452, '16:00', 'Sylhet', 3, 400);

insert into ticket values(80, 'Kishorganj express', 007, 3, TO_DATE('06-02-2022', 'DD-MM-YYYY'), TO_DATE('08-02-2022', 'DD-MM-YYYY'), 333, '14:00', 'Kishoreganj', 3, 350);

insert into ticket values(420, 'Egaroshindhur', 006, 6, TO_DATE('10-02-2022', 'DD-MM-YYYY'), TO_DATE('15-02-2022', 'DD-MM-YYYY'), 215, '8:00', 'Airport', 1, 700);

insert into ticket values(210, 'Shemol', 002, 4, TO_DATE('10-02-2022', 'DD-MM-YYYY'), TO_DATE('15-02-2022', 'DD-MM-YYYY'), 21, '9:00', 'Airport', 1, 450);

insert into ticket values(314, 'Upobon', 009, 1, TO_DATE('11-02-2022', 'DD-MM-YYYY'), TO_DATE('18-02-2022', 'DD-MM-YYYY'), 111, '11:00', 'Sylhet', 4, 500);

insert into ticket values(520, 'Upokul', 008, 5, TO_DATE('13-02-2022', 'DD-MM-YYYY'), TO_DATE('22-02-2022', 'DD-MM-YYYY'), 210, '14:00', 'Noakhali', 2, 420);

insert into ticket values(147, 'Egaroshindhur', 006, 5, TO_DATE('19-02-2022', 'DD-MM-YYYY'), TO_DATE('02-03-2022', 'DD-MM-YYYY'), 175, '8:00', 'Kishoreganj', 3, 700);

The screenshot shows a SQL query in a 'Query Builder' window. The query creates a table named 'ticket' with various attributes and constraints, inserts 11 rows of data, and then selects all rows from the table. Below the query, the 'Query Result' window displays the fetched data in a table format.

```

39 create table ticket(
40 ticket_no number(25) not null,
41 train_name varchar2(30),
42 train_code number,
43 coach_no number,
44 booking_date date,
45 journey_date date,
46 seat_no number,
47 train_time varchar2(20),
48 station_name varchar2(30),
49 platform_no number,
50 ticket_price number,
51 constraint pk_ticket primary key(ticket_no),
52 constraint fk_ticket_train_code foreign key (train_code)
53 references train (train_code)
54 );
55
56 insert into ticket values(601, 'Shuborno', 001, 3, TO_DATE('10-01-2022', 'DD-MM-YYYY'), TO_DATE('22-01-2022', 'DD-MM-YYYY'), 625, '8:00', 'Airport', 3, 400);
57 insert into ticket values(512, 'Shuborno', 001, 2, TO_DATE('11-01-2022', 'DD-MM-YYYY'), TO_DATE('20-01-2022', 'DD-MM-YYYY'), 430, '7:30', 'Chittagong', 3, 400);
58 insert into ticket values(224, 'Shemol', 002, 2, TO_DATE('13-01-2022', 'DD-MM-YYYY'), TO_DATE('18-01-2022', 'DD-MM-YYYY'), 333, '9:00', 'Airport', 1, 350);
59 insert into ticket values(321, 'Duronco', 003, 5, TO_DATE('10-01-2022', 'DD-MM-YYYY'), TO_DATE('21-01-2022', 'DD-MM-YYYY'), 20, '7:00', 'Kamlapur', 2, 400);
60 insert into ticket values(412, 'Ekota', 004, 4, TO_DATE('18-01-2022', 'DD-MM-YYYY'), TO_DATE('28-01-2022', 'DD-MM-YYYY'), 100, '12:00', 'Airport', 4, 300);
61 insert into ticket values(21, 'Mohanopor', 005, 2, TO_DATE('19-01-2022', 'DD-MM-YYYY'), TO_DATE('30-01-2022', 'DD-MM-YYYY'), 132, '10:00', 'Cantonment', 2, 300);
62 insert into ticket values(142, 'Kalni', 010, 7, TO_DATE('02-02-2022', 'DD-MM-YYYY'), TO_DATE('08-02-2022', 'DD-MM-YYYY'), 452, '16:00', 'Sylhet', 3, 400);
63 insert into ticket values(80, 'Kishoreganj express', 007, 3, TO_DATE('06-02-2022', 'DD-MM-YYYY'), TO_DATE('08-02-2022', 'DD-MM-YYYY'), 333, '14:00', 'Kishoreganj', 3, 350);
64 insert into ticket values(420, 'Egaroshindhur', 006, 6, TO_DATE('10-02-2022', 'DD-MM-YYYY'), TO_DATE('15-02-2022', 'DD-MM-YYYY'), 215, '8:00', 'Airport', 1, 700);
65 insert into ticket values(210, 'Shemol', 002, 4, TO_DATE('10-02-2022', 'DD-MM-YYYY'), TO_DATE('15-02-2022', 'DD-MM-YYYY'), 21, '9:00', 'Airport', 1, 450);
66 insert into ticket values(314, 'Upobon', 009, 1, TO_DATE('11-02-2022', 'DD-MM-YYYY'), TO_DATE('18-02-2022', 'DD-MM-YYYY'), 111, '11:00', 'Sylhet', 4, 500);
67 insert into ticket values(520, 'Upokul', 008, 5, TO_DATE('13-02-2022', 'DD-MM-YYYY'), TO_DATE('22-02-2022', 'DD-MM-YYYY'), 210, '14:00', 'Hoakhal', 2, 420);
68 insert into ticket values(147, 'Egaroshindhur', 006, 5, TO_DATE('19-02-2022', 'DD-MM-YYYY'), TO_DATE('02-03-2022', 'DD-MM-YYYY'), 175, '8:00', 'Kishoreganj', 3, 700);
69
70 select * from ticket

```

TICKET_NO	TRAIN_NAME	TRAIN_CODE	COACH_NO	BOOKING_DATE	JOURNEY_DATE	SEAT_NO	TRAIN_TIME	STATION_NAME	PLATFORM_NO	TICKET_PRICE
1	601 Shuborno	1	3	310-JAN-22	22-JAN-22	625	8:00	Airport	3	400
2	512 Shuborno	1	2	211-JAN-22	20-JAN-22	430	7:30	Chittagong	3	400
3	224 Shemol	2	2	212-JAN-22	18-JAN-22	233	9:00	Airport	1	350
4	321 Duronco	3	5	510-JAN-22	21-JAN-22	20	7:00	Kamlapur	2	400
5	412 Ekota	4	4	418-JAN-22	28-JAN-22	100	12:00	Airport	4	300
6	21 Mohanopor	5	2	219-JAN-22	30-JAN-22	132	10:00	Cantonment	2	300
7	142 Kalni	10	7	702-FEB-22	08-FEB-22	452	16:00	Sylhet	3	400
8	80 Kishoreganj express	7	3	306-FEB-22	08-FEB-22	333	14:00	Kishoreganj	3	350
9	420 Egaroshindhur	6	6	610-FEB-22	15-FEB-22	215	8:00	Airport	1	700
10	210 Shemol	2	4	410-FEB-22	15-FEB-22	21	9:00	Airport	1	450
11	314 Upobon	9	1	111-FEB-22	18-FEB-22	111	11:00	Sylhet	4	500

4. Passenger

create table passenger(

passenger_id number not null,

passenger_name varchar2(30),

gender varchar2(20),

age number,

card_id number,

constraint pk_passenger primary key(passenger_id),

constraint fk_passenger foreign key (card_id)

references card (card_id)

);

insert into passenger values(201, 'Nazmul', 'Male', 20, 'Yes');

insert into passenger values(202, 'Hasnatur', 'Male', 19, 'Yes');

insert into passenger values(203, 'Pritom', 'Male', 20, 'Yes');

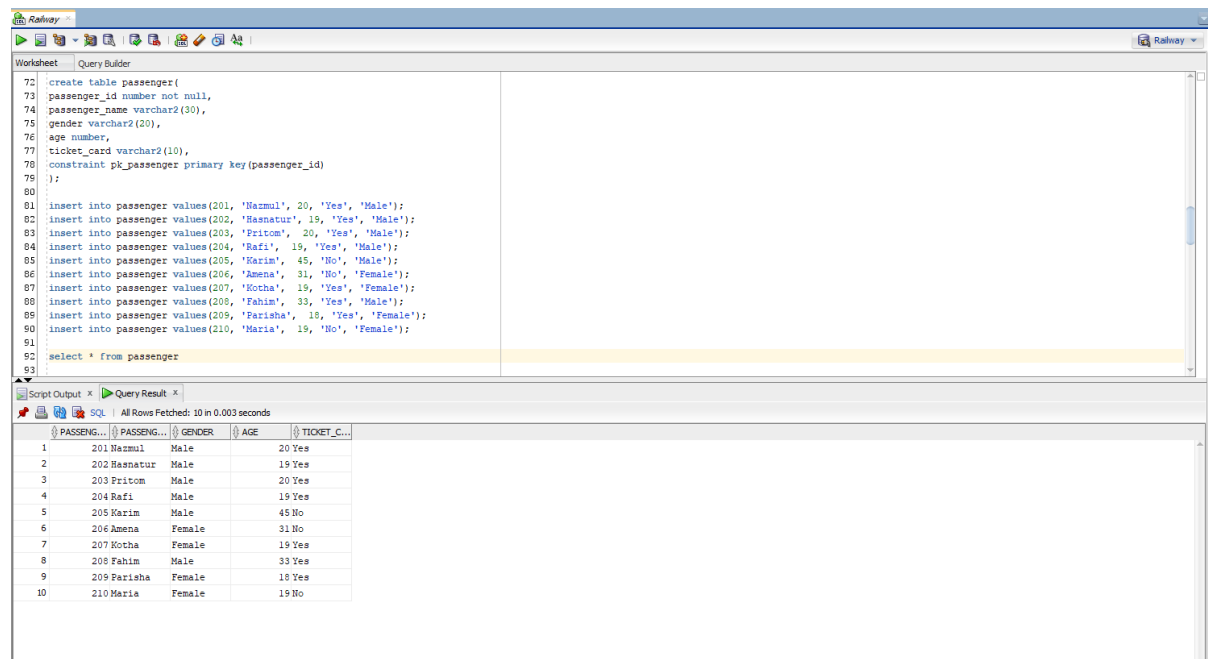
insert into passenger values(204, 'Rafi', 'Male', 19, 'Yes');

insert into passenger values(205, 'Karim', 'Male', 45, 'No');

```

insert into passenger values(206, 'Amena', 'Female', 31, 'No');
insert into passenger values(207, 'Kotha', 'Female', 19, 'Yes');
insert into passenger values(208, 'Fahim', 'Male', 33, 'Yes');
insert into passenger values(209, 'Parisha', 'Female', 18, 'Yes');
insert into passenger values(210, 'Maria', 'Female', 19, 'No');

```



5. Booth

```

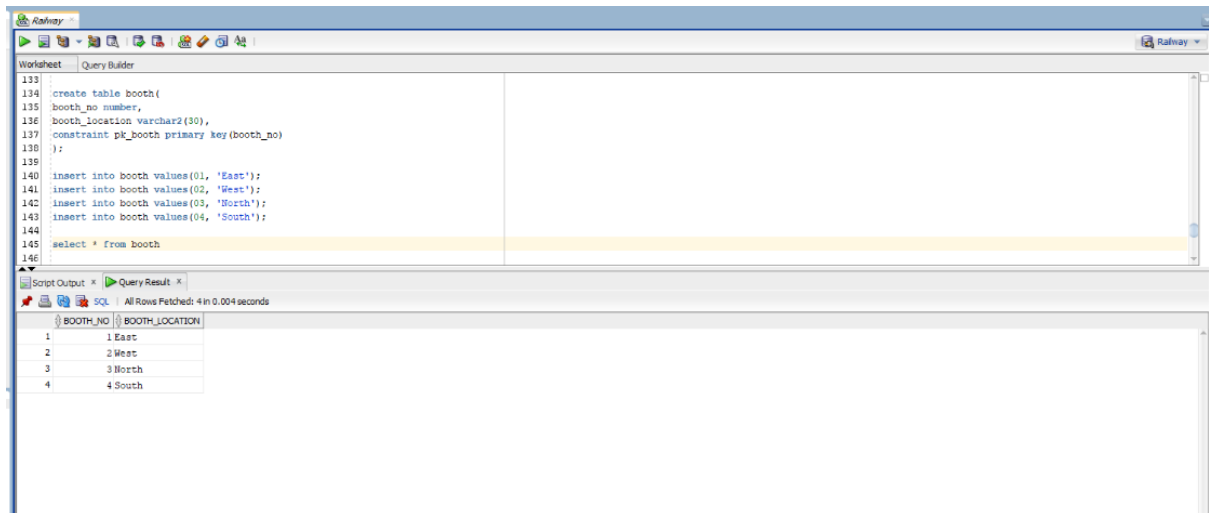
create table booth(
booth_no number,
booth_location varchar2(30),
constraint pk_booth primary key(booth_no)
);

```

```

insert into booth values(01, 'East');
insert into booth values(02, 'West');
insert into booth values(03, 'North');
insert into booth values(04, 'South');

```



6. Card

```
create table card(  
  card_id number not null,  
  balance number,  
  constraint pk_card primary key(card_id)  
)  
  
insert into card values(921, 1000);  
insert into card values(922, 1200);  
insert into card values(923, 1500);  
insert into card values(924, 200);  
insert into card values(925, 1100);  
insert into card values(926, 600);  
insert into card values(927, 1200);
```

The screenshot shows a SQL IDE with two tabs: 'Worksheet' and 'Query Builder'. The 'Query Builder' tab contains the following SQL script:

```

147 create table card(
148   card_id number not null,
149   balance number,
150   constraint pk_card primary key(card_id)
151 );
152
153 insert into card values(921, 1000);
154 insert into card values(922, 1200);
155 insert into card values(923, 1500);
156 insert into card values(924, 200);
157 insert into card values(925, 1100);
158 insert into card values(926, 600);
159 insert into card values(927, 1200);
160
161 select * from card
162

```

The 'Query Result' tab shows the output of the 'select * from card' query, displaying 7 rows of data:

CARD_ID	BALANCE
921	1000
922	1200
923	1500
924	200
925	1100
926	600
927	1200

Report Query

EQUI-JOIN

select passenger.passenger_id, passenger.card_id, passenger.passenger_name,
card.balance

from passenger , card

where passenger.card_id = card.card_id;

The screenshot shows a SQL IDE with two tabs: 'Script Output' and 'Query Result'. The 'Query Result' tab displays the output of an equi-join query between the 'passenger' and 'card' tables, showing 7 rows of data:

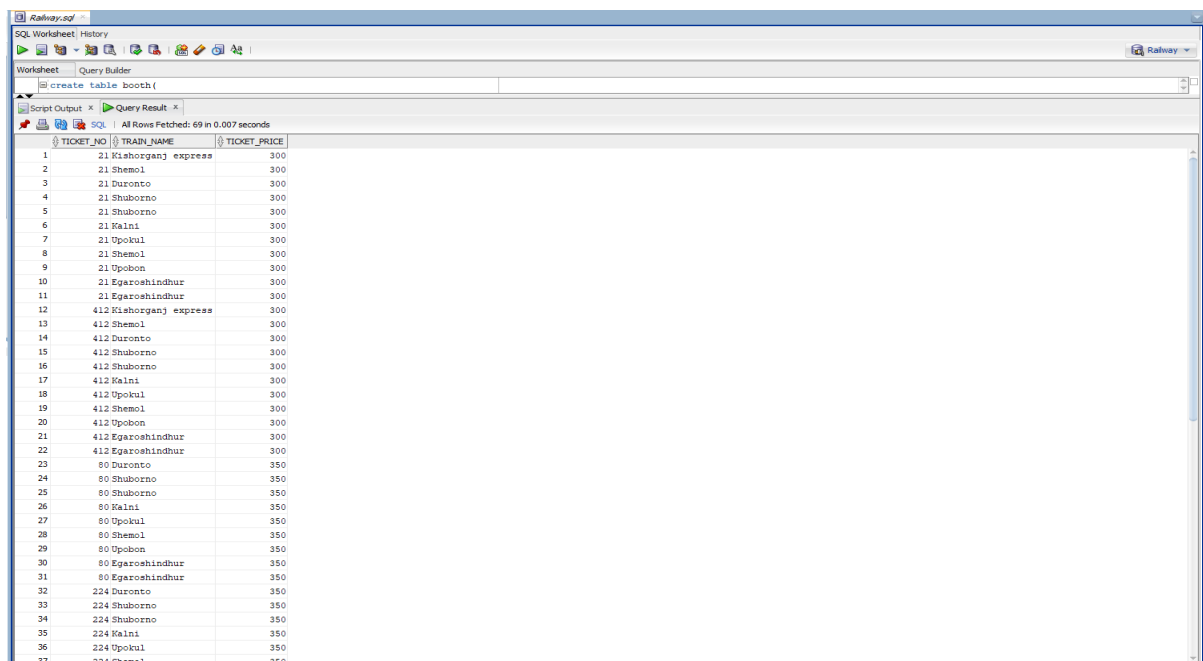
PASSENGER_ID	CARD_ID	PASSENGER_NAME	BALANCE
201	921	Nazmul	1000
202	922	Hasnatur	1200
203	923	Pritom	1500
204	924	Rafi	200
205	925	Karim	1100
206	926	Amena	600
207	927	Kotha	1200

OUTER-JOIN

```
select passenger.passenger_id, passenger.card_id, passenger.passenger_name,  
card.balance  
  
from passenger , card  
  
where passenger.card_id(+) = card.card_id  
  
order by passenger.card_id;
```

SELF-JOIN

```
select a.ticket_no, b.train_name, a.ticket_price  
  
from ticket a, ticket b  
  
where a.ticket_price < b.ticket_price;
```



The screenshot shows a SQL Worksheet interface with a query result displayed. The query is a self-join on a table named 'ticket'. The result shows 37 rows, each with three columns: 'TICKET_NO', 'TRAIN_NAME', and 'TICKET_PRICE'. The rows are numbered 1 through 37. The data is as follows:

TICKET_NO	TRAIN_NAME	TICKET_PRICE
21	Wishorganj express	300
21	Shemol	300
21	Duronto	300
21	Shuborno	300
21	Shuborno	300
21	Kalini	300
21	Upokul	300
21	Shemol	300
21	Upobon	300
21	Egaroshindhur	300
21	Egaroshindhur	300
412	Wishorganj express	300
412	Shemol	300
412	Duronto	300
412	Shuborno	300
412	Shuborno	300
412	Kalini	300
412	Upokul	300
412	Shemol	300
412	Upobon	300
412	Egaroshindhur	300
412	Egaroshindhur	300
80	Duronto	350
80	Shuborno	350
80	Shuborno	350
80	Kalini	350
80	Upokul	350
80	Shemol	350
80	Upobon	350
80	Egaroshindhur	350
80	Egaroshindhur	350
224	Duronto	350
224	Shuborno	350
224	Shuborno	350
224	Kalini	350
224	Upokul	350
224	Shemol	350

Display ticket_no, train_name, ticket_price which is greater than 314 no ticket price(subquery)

```
select ticket_no, train_name, ticket_price
from ticket
where ticket_price >
(select ticket_price
from ticket
where ticket_no = 314);
```



The screenshot shows a database query result window with the following data:

TICKET_NO	TRAIN_NAME	TICKET_PRICE
1	420 Egaroshindur	700
2	147 Egaroshindur	700

Display ticket_no, train_name, ticket_price where train_code 002(subquery)

```
select ticket_no, train_name, ticket_price
from ticket
where train_code in
( select train_code
from train
where train_code = 002);
```



The screenshot shows a database query result window with the following data:

TICKET_NO	TRAIN_NAME	TICKET_PRICE
1	224 Shesol	350
2	210 Shesol	450

Subquery using group by and having clause

```
select train_code, min(ticket_price)
from ticket
group by train_code
having min(ticket_price) > (select min(ticket_price)
from ticket
where train_code = 001);
```



Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.002 seconds

	PASSENGER_ID	PASSENGER_NAME
1	201 Nazmul	
2	202 Hannatur	
3	203 Pritom	
4	204 Rafi	
5	205 Karim	

Creating a view

```
create view passenger_view as
select passenger_id, passenger_name
from passenger
where gender = 'Male';

select*from passenger_view;
```



Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.003 seconds

	TRAIN_CODE	MIN(TICKET_PRICE)
1	6	700
2	8	420
3	9	500

Constraint

alter table train

add check(number_of_seat >= 300);

Sequence

create sequence train_train_code

increment by 1

start with 003

maxvalue 10

nocache

nocycle;