

# School of Information Technology and Engineering

## Database Management Systems - Project

**Name- Omkar Kulkarni**

**Reg no- 19BIT0196**

### Using MySQL

---

#### Railway Reservation System

**Train** (train Number, name, source, destination, start\_time, reach\_time, traveltime, distance, class, days, type)

**Ticket** (PNRNo, Transactionid, from\_station, To\_station, date\_of\_journey, class date\_of\_booking, total\_ticket\_fare, train number)

**Passenger** (PNR No, Serial no, Name, Age, Reservation\_status)

**Train\_Route** (Train\_No, route\_no, station\_code, name, arrival\_time, depart\_time, distance, day)

**Train\_Ticket\_fare** (Train\_No, class, base\_fare, reservation\_charge, superfast\_charge, other\_charge, tatkal\_charge, service\_tax)

Create all the tables specified above.

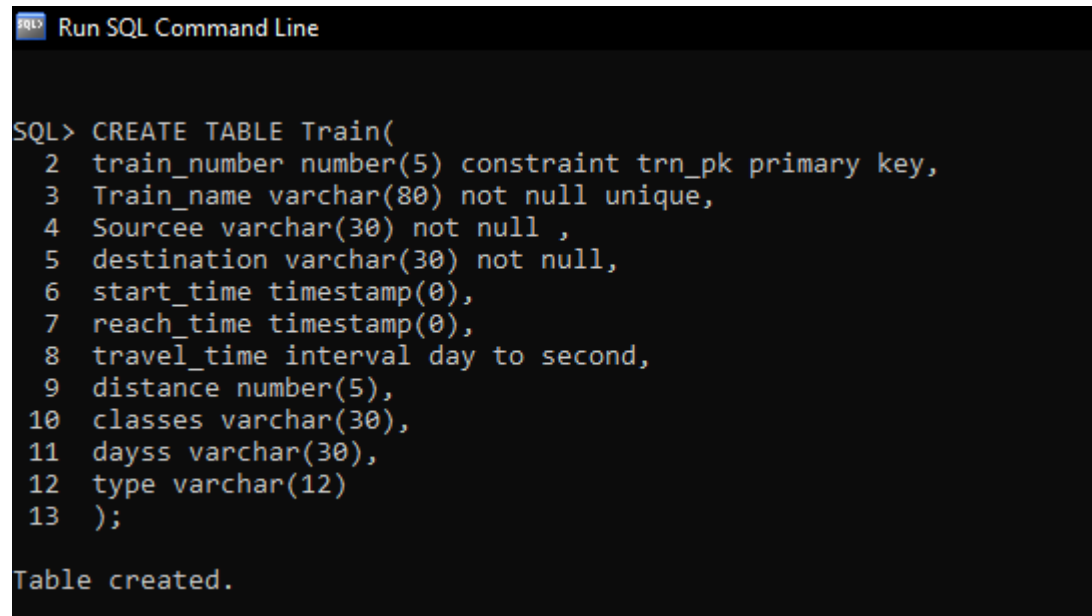
Make underlined columns as primary key. (Use number, number (m, n), varchar(n), date, time, timestamp datatypes appropriately)

Insert at least 5 rows to each table. (Check [www.irctc.co.in](http://www.irctc.co.in) website for actual data)

#### Creating tables:

```
CREATE TABLE Train(  
train_number number(5) constraint trn_pk primary key,  
Train_name varchar(80) not null unique,  
Source varchar(30) not null ,  
destination varchar(30) not null,  
start_time timestamp(0),  
reach_time timestamp(0),  
travel_time interval day to second,
```

```
distance number(5),  
classes varchar(30),  
dayss varchar(30),  
type varchar(12)  
);
```

A screenshot of a SQL Command Line window with a dark background. The title bar reads "Run SQL Command Line". The command prompt shows the creation of a table named 'Train' with 13 columns: train\_number (number(5) primary key), Train\_name (varchar(80) not null unique), Sourcee (varchar(30) not null), destination (varchar(30) not null), start\_time (timestamp(0)), reach\_time (timestamp(0)), travel\_time (interval day to second), distance (number(5)), classes (varchar(30)), dayss (varchar(30)), type (varchar(12)), and a closing semicolon. The response "Table created." is shown at the bottom.

```
SQL> CREATE TABLE Train(  
 2  train_number number(5) constraint trn_pk primary key,  
 3  Train_name varchar(80) not null unique,  
 4  Sourcee varchar(30) not null ,  
 5  destination varchar(30) not null,  
 6  start_time timestamp(0),  
 7  reach_time timestamp(0),  
 8  travel_time interval day to second,  
 9  distance number(5),  
10  classes varchar(30),  
11  dayss varchar(30),  
12  type varchar(12)  
13 );  
  
Table created.
```

```
CREATE TABLE Ticket(  
PNR_no varchar(11) constraint tkt_pk primary key,  
Transactionid number(18) not null constraint tkt_unq  
unique,  
from_station varchar(20),  
To_station varchar(30),  
date_of_journey date,  
classid varchar(8),  
date_of_booking date,  
total_ticket_fare number(5),  
train_number constraint tkt_fk references Train );
```

```

SQL> CREATE TABLE Ticket(
  2  PNR_no varchar(11) constraint tkt_pk primary key,
  3  Transactionid number(18) not null constraint tkt_unq
  4  unique,
  5  from_station varchar(20),
  6  To_station varchar(30),
  7  date_of_journey date,
  8  classid varchar(8),
  9  date_of_booking date,
  10 total_ticket_fare number(5),
  11 train_number constraint tkt_fk references Train );

```

Table created.

```

CREATE TABLE Passenger(
  PNR_no constraint psngr_fk
  references Ticket,
  Serial_no number(2),
  Passenger_Name varchar(30),
  Age number(3),
  Reservation_status varchar(10),
  constraint psngr_pk primary key
  (pnr_no,serial_no)
);

```

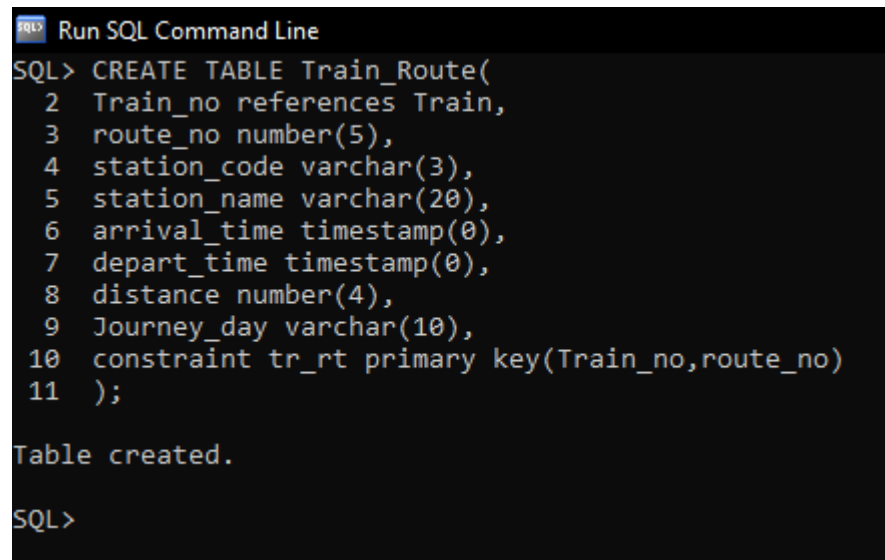
```

SQL> CREATE TABLE Passenger(
  2  PNR_no constraint psngr_fk
  3  references Ticket,
  4  Serial_no number(2),
  5  Passenger_Name varchar(30),
  6  Age number(3),
  7  Reservation_status varchar(10),
  8  constraint psngr_pk primary key
  9  (pnr_no,serial_no)
  10 );

```

Table created.

```
CREATE TABLE Train_Route(  
Train_no references Train,  
route_no number(5),  
station_code varchar(3),  
station_name varchar(20),  
arrival_time timestamp(0),  
depart_time timestamp(0),  
distance number(4),  
Journey_day varchar(10),  
constraint tr_rt primary key(Train_no,route_no)  
);
```



The screenshot shows a dark-themed window titled "Run SQL Command Line". It contains the following SQL code and its output:

```
SQL> CREATE TABLE Train_Route(  
2  Train_no references Train,  
3  route_no number(5),  
4  station_code varchar(3),  
5  station_name varchar(20),  
6  arrival_time timestamp(0),  
7  depart_time timestamp(0),  
8  distance number(4),  
9  Journey_day varchar(10),  
10 constraint tr_rt primary key(Train_no,route_no)  
11 );  
  
Table created.  
  
SQL>
```

```
CREATE TABLE Train_Ticket_fare(  
train_no CONSTRAINT tnrtktfr_fk references  
Train, class varchar(8),  
base_fare number(4),  
reservation_charge number(3),  
superfast_charge number(3),  
other_charge number(3),  
tatkcal_charge number(3),  
service_tax number(3),
```

CONSTRAINT tnrtktfr\_pk primary key(train\_no,class)

);

```
Run SQL Command Line
SQL> CREATE TABLE Train_Ticket_fare(
  2  train_no CONSTRAINT tnrtktfr_fk references
  3  Train, class varchar(8),
  4  base_fare number(4),
  5  reservation_charge number(3),
  6  superfast_charge number(3),
  7  other_charge number(3),
  8  tatkal_charge number(3),
  9  service_tax number(3),
  10 CONSTRAINT tnrtktfr_pk primary key(train_no,class)
  11 );
Table created.
```

### Describing tables:

```
Run SQL Command Line
SQL> desc train
Name                                     Null?   Type
-----
TRAIN_NUMBER                           NOT NULL NUMBER(5)
TRAIN_NAME                             NOT NULL VARCHAR2(80)
SOURCEE                                NOT NULL VARCHAR2(30)
DESTINATION                            NOT NULL VARCHAR2(30)
START_TIME                             TIMESTAMP(0)
REACH_TIME                             TIMESTAMP(0)
TRAVEL_TIME                            INTERVAL DAY(2) TO SECOND(6)
DISTANCE                                NUMBER(5)
CLASSES                                VARCHAR2(30)
DAYSS                                  VARCHAR2(30)
TYPE                                    VARCHAR2(12)

SQL> desc ticket
Name                                     Null?   Type
-----
PNR_NO                                 NOT NULL VARCHAR2(11)
TRANSACTIONID                          NOT NULL NUMBER(18)
FROM_STATION                            VARCHAR2(20)
TO_STATION                              VARCHAR2(30)
DATE_OF_JOURNEY                         DATE
CLASSID                                VARCHAR2(8)
DATE_OF_BOOKING                         DATE
TOTAL_TICKET_FARE                       NUMBER(5)
TRAIN_NUMBER                            NUMBER(5)

SQL> desc passenger
Name                                     Null?   Type
-----
PNR_NO                                 NOT NULL VARCHAR2(11)
SERIAL_NO                               NOT NULL NUMBER(2)
PASSENGER_NAME                          VARCHAR2(30)
AGE                                      NUMBER(3)
RESERVATION_STATUS                       VARCHAR2(10)

SQL> desc train_route
Name                                     Null?   Type
-----
TRAIN_NO                                NOT NULL NUMBER(5)
ROUTE_NO                                NOT NULL NUMBER(5)
STATION_CODE                            VARCHAR2(3)
STATION_NAME                            VARCHAR2(20)
ARRIVAL_TIME                            TIMESTAMP(0)
DEPART_TIME                             TIMESTAMP(0)
DISTANCE                                NUMBER(4)
JOURNEY_DAY                              VARCHAR2(10)

SQL> desc train_ticket_fare
Name                                     Null?   Type
-----
TRAIN_NO                                NOT NULL NUMBER(5)
CLASS                                    NOT NULL VARCHAR2(8)
BASE_FARE                                NUMBER(4)
RESERVATION_CHARGE                       NUMBER(3)
SUPERFAST_CHARGE                         NUMBER(3)
OTHER_CHARGE                             NUMBER(3)
TATKAL_CHARGE                            NUMBER(3)
SERVICE_TAX                             NUMBER(3)

SQL> _
```

## Inserting values in the tables:

### Table- Train

- insert into train values(12164,'Chennai Express','Chennai Egmore','Dadar',to\_timestamp('06:45','hh24:mi'),to\_timestamp('06:00','hh24:mi'),to\_dsinterval('000 23:15:00'),1274,'Sleeper','Monday,Tuesday','Express');
- Insert into train values (12556,'Sapt Kranti Express','New Delhi','Muzaffarpur', to\_timestamp('23:25','hh24:mi'),to\_timestamp('05:55','hh24:mi'),to\_dsinterval('000 07:15:08'),540,'1A,2A,3A,SLEEPER','Monday,Tuesday','Express');
- insert into train values(12434,'masbca sf exp','Chennai Central','Mumbai Central',to\_timestamp('12:30','hh24:mi'),to\_timestamp('10:00','hh24:mi'),to\_dsinterval('000 21:30:00'),1250,'1A,2A,Sleeper','Monday,Wednesday,Friday','Superfast');
- insert into train values(11018,'km ltt weekly express','Karaikal','LokmanyaTilak',to\_timestamp('14:00','hh24:mi'),to\_timestamp('23:45','hh24:mi'),to\_dsinterval('001 01:40:00'),1266,'3A,2A,Sleeper','Monday','Express');
- Insert into train values(12673,'Duronto Express','New Delhi','Calicut',to\_timestamp('12:45','hh24:mi'),to\_timestamp('10:25','hh24:mi'),to\_dsinterval('003 07:11:24'),1580,'1A,2A,3A,SLEEPER','Wednesday','RAJDHANI');

```
SQL> insert into train values(12164,'Chennai Express','Chennai Egmore','Dadar',to_timestamp('06:45','hh24:mi'),to_timestamp('06:00','hh24:mi'),to_dsinterval('000 23:15:00'),1274,'Sleeper','Monday,Tuesday','Express');
```

```
1 row created.
```

```
SQL> Insert into train values (12556,'Sapt Kranti Express','New Delhi','Muzaffarpur', to_timestamp('23:25','hh24:mi'),to_timestamp('05:55','hh24:mi'),to_dsinterval('000 07:15:08'),540,'1A,2A,3A,SLEEPER','Monday,Tuesday','Express');
```

```
1 row created.
```

```
SQL> insert into train values(12434,'masbca sf exp','Chennai Central','Mumbai Central',to_timestamp('12:30','hh24:mi'),to_timestamp('10:00','hh24:mi'),to_dsinterval('000 21:30:00'),1250,'1A,2A,Sleeper','Monday,Wednesday,Friday','Superfast');
```

```
1 row created.
```

```
SQL> insert into train values(11018,'km ltt weekly express','Karaikal','LokmanyaTilak',to_timestamp('14:00','hh24:mi'),to_timestamp('23:45','hh24:mi'),to_dsinterval('001 01:40:00'),1266,'3A,2A,Sleeper','Monday','Express');
```

```
1 row created.
```

```
SQL> Insert into train values(12673,'Duronto Express','New Delhi','Calicut',to_timestamp('12:45','hh24:mi'),to_timestamp('10:25','hh24:mi'),to_dsinterval('003 07:11:24'),1580,'1A,2A,3A,SLEEPER','Wednesday','RAJDHANI');
```

```
1 row created.
```

### Table- Ticket

- insert into ticket values(1928091842,7845632159,'Bangalore Cantt','Mumbai Central',to\_date('24-11-21','dd-mm-yyyy'),'2A',to\_date('04-08-21','dd-mm-yy'),1800,12434);
- insert into ticket values(8674920651,7896354865,'Karaikal','Lokmanya Tilak',to\_date('12-11-21','dd-mm-yy'),'2A',to\_date('19-09-21','dd-mm-yy'),3500,11018);
- insert into ticket values(2345127890,7835682186,'Vellore','Akola',to\_date('12-12-21','dd-mm-yy'),'Sleeper',to\_date('18-07-21','dd-mm-yy'),400,12164);
- insert into ticket values(2895127890,7835682142,'Kanpur','Delhi',to\_date('22-12-21','dd-mm-yy'),'3A',to\_date('16-08-21','dd-mm-yy'),1750,12556);
- insert into ticket values(2345168590,7835423186,'Agra','Wayanad',to\_date('14-12-21','dd-mm-yy'),'1A',to\_date('18-10-21','dd-mm-yy'),3000,12673);
- 

```
SQL>
SQL> insert into ticket values(1928091842,7845632159,'Bangalore Cantt','Mumbai Central',to_date('24-11-21','dd-mm-yyyy'),
'2A',to_date('04-08-21','dd-mm-yy'),1800,12434);

1 row created.

SQL> insert into ticket values(8674920651,7896354865,'Karaikal','Lokmanya Tilak',to_date('12-11-21','dd-mm-yy'),'2A',to_
date('19-09-21','dd-mm-yy'),3500,11018);

1 row created.
```

```
Run SQL Command Line

SQL> insert into ticket values(2345127890,7835682186,'Vellore','Akola',to_date('12-12-21','dd-mm-yy'),'Sleeper',to_date(
'18-07-21','dd-mm-yy'),400,12164);

1 row created.

SQL> insert into ticket values(2895127890,7835682142,'Kanpur','Delhi',to_date('22-12-21','dd-mm-yy'),'3A',to_date('16-08
-21','dd-mm-yy'),1750,12556);

1 row created.

SQL> insert into ticket values(2345168590,7835423186,'Agra','Wayanad',to_date('14-12-21','dd-mm-yy'),'1A',to_date('18-10
-21','dd-mm-yy'),3000,12673);

1 row created.

SQL>
```

PNR_NO	TRANSACTIONID		FROM_STATION	TO_STATION	
DATE_OF_J	CLASSID	DATE_OF_B	TOTAL_TICKET_FARE	TRAIN_NUMBER	
1928091842	7845632159	Bangalore Cantt		Mumbai Central	
24-NOV-21	2A	04-AUG-21	1800	12434	
8674920651	7896354865	Karaikal		Lokmanya Tilak	
12-NOV-21	2A	19-SEP-21	3500	11018	
2345127890	7835682186	Vellore		Akola	
12-DEC-21	Sleeper	18-JUL-21	400	12164	
PNR_NO	TRANSACTIONID		FROM_STATION	TO_STATION	
DATE_OF_J	CLASSID	DATE_OF_B	TOTAL_TICKET_FARE	TRAIN_NUMBER	
2895127890	7835682142	Kanpur		Delhi	
22-DEC-21	3A	16-AUG-21	1750	12556	
2345168590	7835423186	Agra		Wayanad	
14-DEC-21	1A	18-OCT-21	3000	12673	

### Table- Passenger

- insert into passenger values(1928091842,01,'Kinap',68,'CNF');
- insert into passenger values(8674920651,02,'Holly',44,'CNF');
- insert into passenger values(2345127890,10,'Kishore',32,'NOT CNF');
- insert into passenger values(2895127890,04,'Jaydev',19,'CNF');
- insert into passenger values(2345168590,12,'Madhav',23,'CNF');

```

Run SQL Command Line

SQL>
SQL> insert into passenger values(1928091842,01,'Kinap',68,'CNF');
1 row created.

SQL> insert into passenger values(8674920651,02,'Holly',44,'CNF');
1 row created.

SQL> insert into passenger values(2345127890,10,'Kishore',32,'NOT CNF');
1 row created.

SQL> insert into passenger values(2895127890,04,'Jaydev',19,'CNF');
1 row created.

SQL> insert into passenger values(2345168590,12,'Madhav',23,'CNF');
1 row created.

SQL> select* from passenger;

PNR_NO      SERIAL_NO  PASSENGER_NAME      AGE  RESERVATIO
-----
1928091842      1  Kinap              68  CNF
8674920651      2  Holly             44  CNF
2345127890     10  Kishore           32  NOT CNF
2895127890      4  Jaydev            19  CNF
2345168590     12  Madhav            23  CNF

SQL> _

```



### Table- Train Route

- insert into train\_route values(12164,01235,'EM','Chennai Egmore',to\_timestamp('06:35','hh24:mi'),to\_timestamp('06:45','hh24:mi'),1335,'Mon');
- insert into train\_route values(12434,03453,'KPD','Chennai Central',to\_timestamp('15:35','hh24:mi'),to\_timestamp('15:40','hh24:mi'),1245,'Wed');
- insert into train\_route values(12556,03435,'NZM','New Delhi',to\_timestamp('23:25','hh24:mi'),to\_timestamp('23:45','hh24:mi'),1084,'Tue');
- insert into train\_route values(11018,02768,'KKL','Karaikal',to\_timestamp('14:35','hh24:mi'),to\_timestamp('14:40','hh24:mi'),1436,'Mon');
- insert into train\_route values(12673,01785,'KMR','Nizamuddin Terminus',to\_timestamp('12:45','hh24:mi'),to\_timestamp('12:50','hh24:mi'),2556,'Wed');

```
Run SQL Command Line
SQL>
SQL> insert into train_route values(12164,01235,'EM','Chennai Egmore',to_timestamp('06:35','hh24:mi'),to_timestamp('06:45','hh24:mi'),1335,'Mon');
1 row created.

SQL> insert into train_route values(12434,03453,'KPD','Chennai Central',to_timestamp('15:35','hh24:mi'),to_timestamp('15:40','hh24:mi'),1245,'Wed');
1 row created.

SQL>
SQL> insert into train_route values(12556,03435,'NZM','New Delhi',to_timestamp('23:25','hh24:mi'),to_timestamp('23:45','hh24:mi'),1084,'Tue');
1 row created.

SQL> insert into train_route values(11018,02768,'KKL','Karaikal',to_timestamp('14:35','hh24:mi'),to_timestamp('14:40','hh24:mi'),1436,'Mon');
1 row created.

SQL>
SQL>
SQL> insert into train_route values(12673,01785,'KMR','Nizamuddin Terminus',to_timestamp('12:45','hh24:mi'),to_timestamp('12:50','hh24:mi'),2556,'Wed');
1 row created.
```

```
Run SQL Command Line
1 row created.

SQL> select * from train_route;

  TRAIN_NO  ROUTE_NO STA STATION_NAME
-----
ARRIVAL_TIME
DEPART_TIME
  DISTANCE  JOURNEY_DA
-----
      12164      1235 EM  Chennai Egmore
01-AUG-21 06.35.00 AM
01-AUG-21 06.45.00 AM
      1335 Mon

  TRAIN_NO  ROUTE_NO STA STATION_NAME
-----
ARRIVAL_TIME
DEPART_TIME
  DISTANCE  JOURNEY_DA
-----
      12434      3453 KPD Chennai Central
01-AUG-21 03.35.00 PM
01-AUG-21 03.40.00 PM
      1245 Wed
```

### Table- Train Ticket Fare

- insert into train\_ticket\_fare  
values(12164, 'Sleeper', 300, 20, null, 10, 10, 10);
- insert into train\_ticket\_fare  
values(12556, '3A', 1500, 220, 10, 5, 10, 10);
- insert into train\_ticket\_fare  
values(12673, '1A', 2000, 150, 250, 250, 250, 100);
- insert into train\_ticket\_fare  
values(12434, '2A', 1500, 100, 25, 75, 50, 50);
- insert into train\_ticket\_fare  
values(11018, '2A', 2500, 150, 250, 250, 250, 100);

```
Run SQL Command Line

SQL> insert into train_ticket_fare
  2 values(12164, 'Sleeper', 300, 20, null, 10, 10, 10);

1 row created.

SQL>
SQL> insert into train_ticket_fare
  2 values(12556, '3A', 1500, 220, 10, 5, 10, 10);

1 row created.

SQL>
SQL> insert into train_ticket_fare
  2 values(12673, '1A', 2000, 150, 250, 250, 250, 100);

1 row created.

SQL>
SQL> insert into train_ticket_fare
  2 values(12434, '2A', 1500, 100, 25, 75, 50, 50);

1 row created.

SQL>
SQL> insert into train_ticket_fare values(11018, '2A', 2500, 150, 250, 250, 250, 100);

1 row created.
```

```
SQL> select*from train_ticket_fare;
```

TRAIN_NO	CLASS	BASE_FARE	RESERVATION_CHARGE	SUPERFAST_CHARGE	OTHER_CHARGE
TATKAL_CHARGE		SERVICE_TAX			

12164	Sleeper	300	20		10
10		10			

12556	3A	1500	220	10	5
10		10			

12673	1A	2000	150	250	250
250		100			

TRAIN_NO	CLASS	BASE_FARE	RESERVATION_CHARGE	SUPERFAST_CHARGE	OTHER_CHARGE
TATKAL_CHARGE		SERVICE_TAX			

12434	2A	1500	100	25	75
50		50			

11018	2A	2500	150	250	250
250		100			

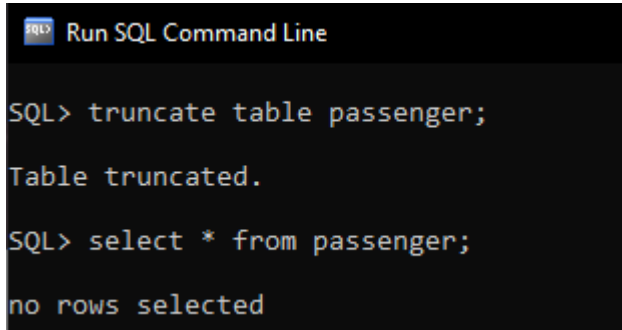
```
SQL> _
```

## simple DDL/DML Queries to

### 1. Remove all the rows from Passenger table permanently.

```
truncate table passenger;
```

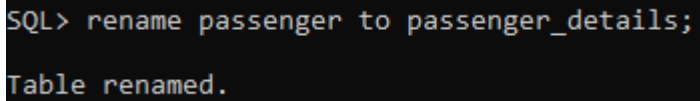
```
select * from passenger;
```

A screenshot of a SQL Command Line window with a dark background. The title bar says "Run SQL Command Line". The command prompt shows the following sequence: "SQL> truncate table passenger;" followed by "Table truncated." on the next line. Then "SQL> select \* from passenger;" followed by "no rows selected" on the next line.

```
SQL> truncate table passenger;
Table truncated.
SQL> select * from passenger;
no rows selected
```

### 2. Change the name of the Passenger table to Passenger\_Details.

```
rename passenger to passenger_details;
```

A screenshot of a SQL Command Line window with a dark background. The command prompt shows the following sequence: "SQL> rename passenger to passenger\_details;" followed by "Table renamed." on the next line.

```
SQL> rename passenger to passenger_details;
Table renamed.
```

### 3. List all train details.

```
select train_number, train_name, source, classes from train;
```

```
Run SQL Command Line

SQL> select train_number,train_name,source,classes from train;

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          CLASSES
-----
12673
Durgam Express
New Delhi      1A,2A,3A,SLEEPER

12164
Chennai Express
Chennai Egmore Sleeper

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          CLASSES
-----

12556
Sapt Kranti Express
New Delhi      1A,2A,3A,SLEEPER

12434
Masbca sf exp

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          CLASSES
-----
Chennai Central 1A,2A,Sleeper

11018
km ltt weekly express
Karaikal      3A,2A,Sleeper
```

#### 4. List all passenger details.

```
select passenger_name,age from passenger_details;
```

```
Run SQL Command Line

SQL> select passenger_name,age from passenger_details;

PASSENGER_NAME      AGE
-----
Kinap               68
Holly               44
Kishore             32
Jaydev              19
Madhav              23
```

**5. Give a list of trains in ascending order of number.**

```
select train_name from train order by train_number;
```

```
Run SQL Command Line

SQL> select train_name from train order by train_number;

TRAIN_NAME
-----
km ltt weekly express
Chennai Express
masbca sf exp
Sapt Kranti Express
Duronto Express

SQL> _
```

**6. List the senior citizen passenger's details.**

```
select passenger_name,age from passenger_details where age>=60;
```

```
SQL> select passenger_name,age from passenger_details where age>=60;

PASSENGER_NAME          AGE
-----
Kinap                    68
```

**7. List the station names where code starts with 'M'.**

```
select station_name,station_code from train_route where station_code like 'M%';
```

```
SQL> select station_name,station_code from train_route where station_code like 'M%';

no rows selected
```

## 8. List the train details within a range of numbers.

select \* from train where train\_number between 12321 and 12679 order by train\_number asc;

```
Run SQL Command Line
SQL> select * from train where train_number between 12321 and 12679 order by train_number asc;

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE
-----
12434

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE
-----
masbca sf exp

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE
-----
Chennai Central          Mumbai Central

TRAIN_NUMBER
```

Run SQL Command Line

```
-----  
TRAIN_NAME  
-----  
SOURCEE                                DESTINATION  
-----  
START_TIME  
-----  
REACH_TIME  
-----  
TRAVEL_TIME  
-----  
    DISTANCE CLASSES                                DAYSS  
-----  
TYPE  
-----  
01-AUG-21 12.30.00 PM  
  
TRAIN_NUMBER  
-----  
TRAIN_NAME  
-----  
SOURCEE                                DESTINATION  
-----  
START_TIME  
-----  
REACH_TIME  
-----  
TRAVEL_TIME  
-----  
    DISTANCE CLASSES                                DAYSS  
-----  
TYPE  
-----  
01-AUG-21 10.00.00 AM  
  
TRAIN_NUMBER  
-----  
TRAIN_NAME  
-----  
SOURCEE                                DESTINATION  
-----  
START_TIME  
-----  
REACH_TIME  
-----  
TRAVEL_TIME  
-----  
    DISTANCE CLASSES                                DAYSS  
-----  
TYPE  
-----  
+00 21:30:00.000000  
  
TRAIN_NUMBER  
-----  
TRAIN_NAME  
-----
```





Run SQL Command Line

REACH\_TIME

TRAVEL\_TIME

DISTANCE CLASSES

DAYSS

TYPE

01-AUG-21 05.55.00 AM

TRAIN\_NUMBER

TRAIN\_NAME

SOURCE

DESTINATION

START\_TIME

REACH\_TIME

TRAVEL\_TIME

DISTANCE CLASSES

DAYSS

TYPE

+00 07:15:08.000000

TRAIN\_NUMBER

TRAIN\_NAME

SOURCE

DESTINATION

START\_TIME

REACH\_TIME

TRAVEL\_TIME

DISTANCE CLASSES

DAYSS

TYPE

540 1A,2A,3A,SLEEPER

Monday,Tuesday

TRAIN\_NUMBER

TRAIN\_NAME

SOURCE

DESTINATION

START\_TIME

REACH\_TIME

TRAVEL\_TIME

DISTANCE CLASSES

DAYSS

TYPE

**9. Change the superfast charge value in train fare as zero, if it is null.**

```
update train_ticket_fare set superfast_charge = 0 where train_no=12434;
```

```
select * from train_ticket_fare where superfast_charge=0;
```

```
SQL> Run SQL Command Line

SQL> update train_ticket_fare set superfast_charge = 0 where train_no=12434;
1 row updated.

SQL> select * from train_ticket_fare where superfast_charge=0;

  TRAIN_NO CLASS      BASE_FARE RESERVATION_CHARGE SUPERFAST_CHARGE OTHER_CHARGE
-----
TATKAL_CHARGE SERVICE_TAX
-----
      12434 2A          1500              100              0              75
              50              50
```

**10. List the passenger names whose tickets are not confirmed.**

```
select passenger_name from passenger_details where reservation_status= 'NOT
CNF';
```

```
SQL> select passenger_name from passenger_details where reservation_status= 'NOT CNF';

PASSENGER_NAME
-----
Kishore
```

**CONSTRAINTS**

**11. Create (Alter table to add constraint) the necessary foreign keys by identifying the relationships in the table.**

```
alter table ticket add constraint fk_1 foreign key(train_number) references train;
```

```
alter table passenger_details add constraint fk_2 foreign key (PNR_no) references ticket;
```

```
alter table train_ticket_fare add constraint fk_3 foreign key (train_no) references train;
```

```
alter table train_route add constraint fk_4 foreign key (train_no) references train;
```

already during table creation, I had made constraints for foreign key so when the commands run, the compiler says that a similar constraint already exists:

```
1. CREATE TABLE Ticket(  
    PNR_no varchar(11) constraint tkt_pk primary key,  
    Transactionid number(18) not null constraint tkt_unq  
    unique,  
    from_station varchar(20),  
    To_station varchar(30),  
    date_of_journey date,  
    classid varchar(8),  
    date_of_booking date,  
    total_ticket_fare number(5),  
    train_number constraint tkt_fk references Train );
```

```
2. CREATE TABLE Passenger(  
    PNR_no constraint psngr_fk  
    references Ticket,  
    Serial_no number(2),  
    Passenger_Name varchar(30),  
    Age number(3),  
    Reservation_status varchar(10),  
    constraint psngr_pk primary key  
    (pnr_no,serial_no)  
    );
```

```
3. CREATE TABLE Train_Ticket_fare(  
    train_no CONSTRAINT tnrtktfr_fk references  
    Train, class varchar(8),  
    base_fare number(4),  
    reservation_charge number(3),  
    superfast_charge number(3),  
    other_charge number(3),  
    tatkal_charge number(3),
```

```
service_tax number(3),  
  
CONSTRAINT tnrtktfr_pk primary key(train_no,class)  
  
);
```

4. CREATE TABLE Train\_Route(

```
Train_no references Train,  
  
route_no number(5),  
  
station_code varchar(3),  
  
station_name varchar(20),  
  
arrival_time timestamp(0),  
  
depart_time timestamp(0),  
  
distance number(4),  
  
Journey_day varchar(10),  
  
constraint tr_rt primary key(Train_no,route_no)  
  
);
```

thus, using alter command I get:

```
Run SQL Command Line  
ORA-00907: missing right parenthesis  
  
SQL> alter table ticket add constraint fk_1 foreign key(train_number) references train;  
alter table ticket add constraint fk_1 foreign key(train_number) references train  
*  
ERROR at line 1:  
ORA-02275: such a referential constraint already exists in the table  
  
SQL>
```

```
Run SQL Command Line  
  
SQL> alter table passenger_details add constraint fk_2 foreign key (PNR_no) references ticket  
alter table passenger_details add constraint fk_2 foreign key (PNR_no) references ticket  
*  
ERROR at line 1:  
ORA-02275: such a referential constraint already exists in the table
```

```

Run SQL Command Line
SQL> alter table train_ticket_fare add constraint fk_3 foreign key (train_no) references train;
alter table train_ticket_fare add constraint fk_3 foreign key (train_no) references train
*
ERROR at line 1:
ORA-02275: such a referential constraint already exists in the table

SQL>

```

```

SQL> alter table train_route add constraint fk_4 foreign key (train_no) references train;
alter table train_route add constraint fk_4 foreign key (train_no) references train
*
ERROR at line 1:
ORA-02275: such a referential constraint already exists in the table

```

## 12. Add a suitable constraint to train table to always have train no in the range 10001 to 99999.

alter table train add constraint train\_chk check(train\_number between 10001 and 99999);

```

SQL> alter table train add constraint train_chk check(train_number between 10001 and 99999);
Table altered.

```

## 13. Add a suitable constraint for the column of station name, so that does not take duplicates.

alter table train\_route add constraint trnr\_unq1 station\_name varchar(20) unique;

alter table ticket add constraint trnr\_unq2 from\_station varchar(20) unique;

alter table ticket add constraint trnr\_unq3 to\_station varchar(20) unique;

alter table train add constraint trnr\_unq4 sourcee varchar(20) unique;

```

Run SQL Command Line

SQL> alter table train_route add constraint trnr_unq1 station_name varchar(20) unique;
alter table train_route add constraint trnr_unq1 station_name varchar(20) unique
*
ERROR at line 1:
ORA-01430: column being added already exists in table

```

```
SQL> alter table ticket add constraint trnr_unq2 from_station varchar(20) unique;  
alter table ticket add constraint trnr_unq2 from_station varchar(20) unique  
*
```

```
ERROR at line 1:  
ORA-01430: column being added already exists in table
```

```
SQL> alter table ticket add constraint trnr_unq3 to_station varchar(20) unique;  
alter table ticket add constraint trnr_unq3 to_station varchar(20) unique  
*
```

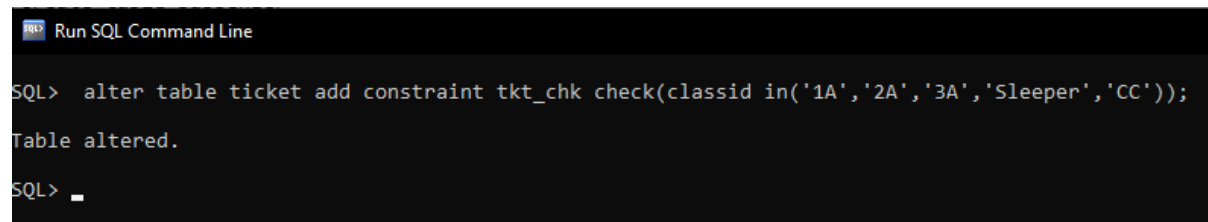
```
ERROR at line 1:  
ORA-01430: column being added already exists in table
```

```
SQL> alter table train add constraint trnr_unq4 sourcee varchar(20) unique;  
alter table train add constraint trnr_unq4 sourcee varchar(20) unique  
*
```

```
ERROR at line 1:  
ORA-01430: column being added already exists in table
```

**14. Add a suitable constraint for the class column that it should take values only as 1A, 2A, 3A, SL, C.**

```
alter table ticket add constraint tkt_chk check(classid in('1A','2A','3A','Sleeper','CC'));
```



```
Run SQL Command Line  
SQL> alter table ticket add constraint tkt_chk check(classid in('1A','2A','3A','Sleeper','CC'));  
Table altered.  
SQL> _
```

**15. Add a not null constraint for the column distance in train\_route.**

```
alter table train_route modify distance not null;
```

```
alter table train_route add constraint trnr_dis distance number(4) not null;
```

SQL> Run SQL Command Line

```
SQL> alter table train_route modify distance not null;
```

Table altered.

SQL>

SQL> Run SQL Command Line

```
SQL> alter table train_route add constraint trnr_dis distance number(4) not null;  
alter table train_route add constraint trnr_dis distance number(4) not null  
*
```

ERROR at line 1:

ORA-01430: column being added already exists in table

SQL>



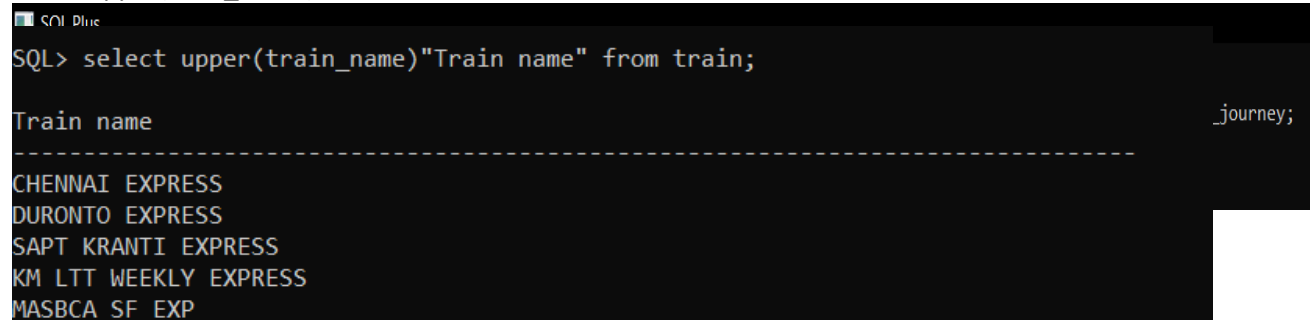
Use SQL PLUS functions for answering the followings:

1. Find the passengers whose date of journey is one month from today.

```
select passenger_name from passenger_details natural join ticket where  
to_char(add_months(sysdate,1),'dd-mon-yy')=date_of_journey;
```

2. Print the train names in upper case.

```
select upper(train_name)"Train name" from train;
```

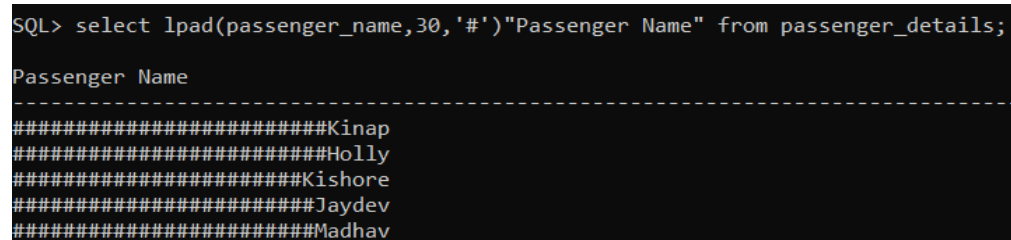


The screenshot shows a terminal window with the title 'SQL Dline'. The command entered is 'SQL> select upper(train\_name)"Train name" from train;'. The output displays a list of train names in uppercase, each preceded by a dashed line separator. The train names are: CHENNAI EXPRESS, DURONTO EXPRESS, SAPT KRANTI EXPRESS, KM LTT WEEKLY EXPRESS, and MASBCA SF EXP.

```
SQL> select upper(train_name)"Train name" from train;  
  
Train name  
-----  
CHENNAI EXPRESS  
DURONTO EXPRESS  
SAPT KRANTI EXPRESS  
KM LTT WEEKLY EXPRESS  
MASBCA SF EXP
```

3. Print the passenger names with left padding character.

```
select lpad(passenger_name,30,'#')"Passenger Name" from passenger_details;
```



The screenshot shows a terminal window with the command 'SQL> select lpad(passenger\_name,30,'#')"Passenger Name" from passenger\_details;'. The output displays passenger names with left padding using the '#' character, each preceded by a dashed line separator. The passenger names are: Kinap, Holly, Kishore, Jaydev, and Madhav.

```
SQL> select lpad(passenger_name,30,'#')"Passenger Name" from passenger_details;  
  
Passenger Name  
-----  
#####Kinap  
#####Holly  
#####Kishore  
#####Jaydev  
#####Madhav
```

#### 4. Print the station codes replacing K with M.

```
select translate(station_code,'K','M')"Station code" from train_route;
```

```
SQL> select translate(station_code,'K','M')"Station code" from train_route;

Station code
-----
EM
MPD
NZM
MML
MMR
```

#### 5. Translate all the LC in class column (Train\_fare) to POT and display.

```
select replace(class,'LC','POT')"class" from train_ticket_fare;
```

```
SQL> select replace(class,'LC','POT')"class" from train_ticket_fare;

class
-----
2A
Sleeper
2A
3A
1A
```

#### 6. Display the fare details of all trains, if any value is ZERO, print as NULL value.

```
select base_fare,reservation_charge,superfast_charge,other_charge,service_tax from
train_ticket_fare;
```

```
select null if (base_fare,0), null if(reservation_charge,0),null if(superfast_charge,0),null
if(other_charge,0), null if(service_tax,0) from train_ticket_fare;
```

BASE_FARE	RESERVATION_CHARGE	SUPERFAST_CHARGE	OTHER_CHARGE	SERVICE_TAX
300	20		10	10
1500	220	10	5	10
2000	150	250	250	100
1500	100	0	75	50
2500	150	250	250	100

```
SQL> select null if (base_fare,0), null if(reservation_charge,0),null if(superfast_charge,0),null if(other_c
harge,0), null if(service_tax,0) from train_ticket_fare;
```

**7. Display the PNR No and transaction id, if transaction id is null, print 'not generated'.**

```
alter table ticket modify transactionid char(14);
```

```
select pnr_no, nvl(transactionid,'not generated') "Transaction id" from ticket;
```

```
SQL> alter table ticket modify transactionid char(14);
alter table ticket modify transactionid char(14)
      *
ERROR at line 1:
ORA-01439: column to be modified must be empty to change datatype
```

```
SQL> select pnr_no, nvl(transactionid,'not generated') "Transaction id" from ticket;
```

**8. Print the date\_of\_journey in the format '27th November 2010'.**

```
select to_char(date_of_journey,'ddth month yyyy') "date_of_journey" from ticket;
```

```
SQL Plus

SQL> select to_char(date_of_journey,'ddth month yyyy') "date_of_journey" from ticket;

date_of_journey
-----
24th november  0021
12th november  2021
12th december  2021
22nd december  2021
14th december  2021
```

**9. Find the maximum fare (total fare).**

```
select max(total_ticket_fare) from ticket;
```

```
SQL> select max(total_ticket_fare) from ticket;

MAX(TOTAL_TICKET_FARE)
-----
3500
```

**10. Find the average age of passengers in one ticket.**

```
select avg(age) from passenger_details group by pnr_no;
```

```
SQL> select avg(age) from passenger_details group by pnr_no;
```

```
AVG(AGE)
-----
        68
        44
        32
        19
        23
```

**11. Find the maximum length of station name available in the database.**

```
select max(length(station_name))from train_route;
```

```
SQL> select max(length(station_name))from train_route;

MAX(LENGTH(STATION_NAME))
-----
                        19
```

**12. Print the fare amount of the passengers as rounded value.**

```
select round(total_ticket_fare)from ticket;
```

```
SQL> select round(total_ticket_fare)from ticket;

ROUND(TOTAL_TICKET_FARE)
-----
                1800
                3500
                 400
                1750
                3000
```

**13. Add the column halt time to train route.**

```
alter table train_route add halt_time interval day to second;
```

```
SQL Plus

SQL> alter table train_route add halt_time interval day to second;

Table altered.
```

#### 14. Update values to it from arrival time and depart time.

```
update train_route set halt_time=depart_time-arrival_time;
```

```
commit;
```

```
SQL> update train_route set halt_time=depart_time-arrival_time;

5 rows updated.

SQL> commit;

Commit complete.
```

#### 15. Display the arrival time, depart time in the format HH:MI (24 hours and minutes).

```
select
to_timestamp(arrival_time,'hh24:mi')"A_time",to_timestamp(depart_time,'hh24:mi')"D_time" from
train_route;
```

```
SQL> select to_timestamp(arrival_time,'hh24:mi')"A_time",to_timestamp(depart_time,'hh24:mi')"D_time" from train_route;
```

```
SQL> select arrival_time,depart_time from train_route;
```

```
ARRIVAL_TIME
```

```
-----
```

```
DEPART_TIME
```

```
-----
```

```
01-OCT-21 06.35.00 AM
```

```
01-OCT-21 06.45.00 AM
```

```
01-OCT-21 03.35.00 PM
```

```
01-OCT-21 03.40.00 PM
```

```
01-OCT-21 11.25.00 PM
```

```
01-OCT-21 11.45.00 PM
```

```
ARRIVAL_TIME
```

```
-----
```

```
DEPART_TIME
```

```
-----
```

```
01-OCT-21 02.35.00 PM
```

```
01-OCT-21 02.40.00 PM
```

```
01-OCT-21 12.45.00 PM
```

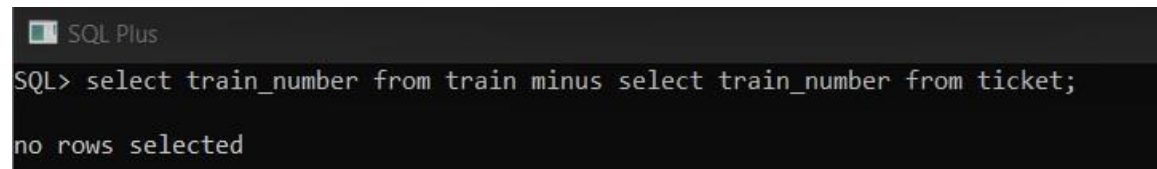
```
01-OCT-21 12.50.00 PM
```

## Use SET Operators

1. Find the train numbers for which reservation have not yet been made.

**Sol:**

select train\_number from train minus select train\_number from ticket;



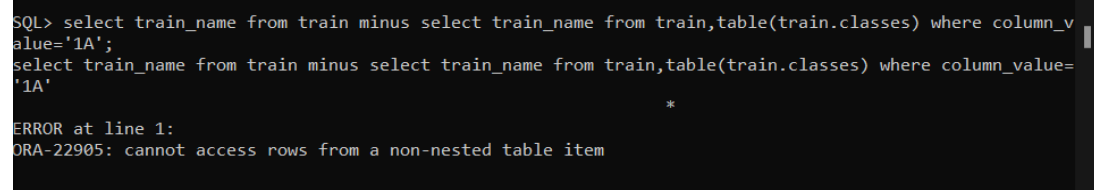
```
SQL Plus
SQL> select train_number from train minus select train_number from ticket;

no rows selected
```

2. Find the train names that do not have a first AC class coach.

**Sol:**

select train\_name from train minus select train\_name from train,table(train.classes) where column\_value='1A';

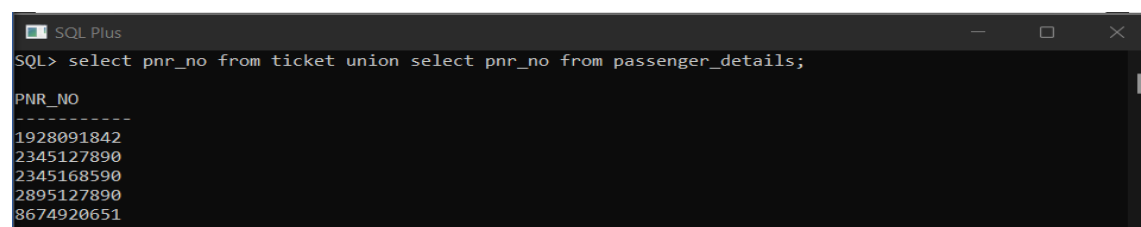


```
SQL> select train_name from train minus select train_name from train,table(train.classes) where column_v
value='1A';
select train_name from train minus select train_name from train,table(train.classes) where column_value=
'1A'
*
ERROR at line 1:
ORA-22905: cannot access rows from a non-nested table item
```

3. Print all the PNR Nos available in the database.

**Sol:**

select pnr\_no from ticket union select pnr\_no from passenger\_details;



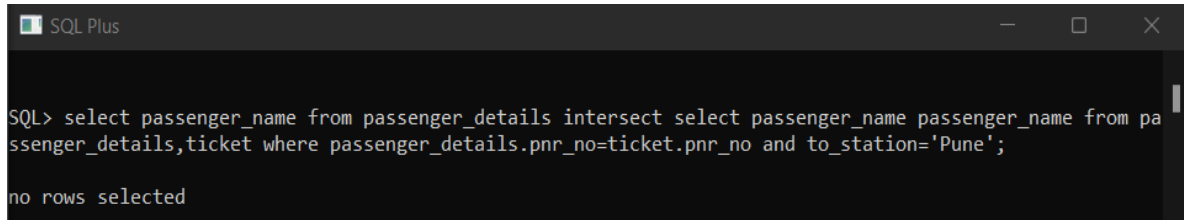
```
SQL Plus
SQL> select pnr_no from ticket union select pnr_no from passenger_details;

PNR_NO
-----
1928091842
2345127890
2345168590
2895127890
8674920651
```

4. Find passenger names who have booked to 'Pune'.

**Sol:**

select passenger\_name from passenger\_details intersect select passenger\_name passenger\_name from passenger\_details,ticket where passenger\_details.pnr\_no=ticket.pnr\_no and to\_station='Pune';

A screenshot of a SQL Plus terminal window. The title bar says "SQL Plus". The terminal shows the following SQL query: `SQL> select passenger_name from passenger_details intersect select passenger_name passenger_name from passenger_details,ticket where passenger_details.pnr_no=ticket.pnr_no and to_station='Pune';` Below the query, it says "no rows selected".

```
SQL> select passenger_name from passenger_details intersect select passenger_name passenger_name from passenger_details,ticket where passenger_details.pnr_no=ticket.pnr_no and to_station='Pune';

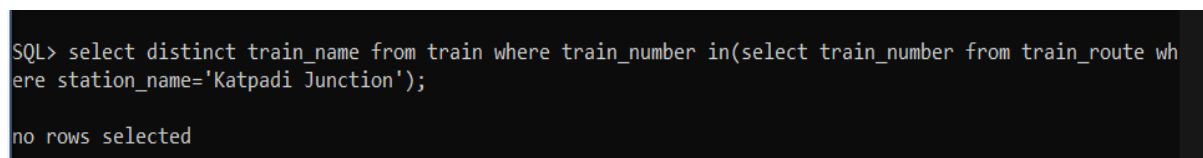
no rows selected
```

### **Use Nested Query (in Operators)**

5. Find the train names that stop in 'Katpadi'.

**Sol:**

select distinct train\_name from train where train\_number in(select train\_number from train\_route where station\_name='Katpadi Junction');

A screenshot of a SQL Plus terminal window. The terminal shows the following SQL query: `SQL> select distinct train_name from train where train_number in(select train_number from train_route where station_name='Katpadi Junction');` Below the query, it says "no rows selected".

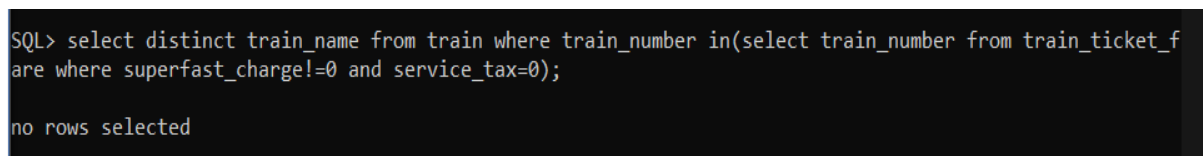
```
SQL> select distinct train_name from train where train_number in(select train_number from train_route where station_name='Katpadi Junction');

no rows selected
```

6. Find the train names that are superfast and the service tax is zero.

**Sol:**

select distinct train\_name from train where train\_number in(select train\_number from train\_ticket\_fare where superfast\_charge!=0 and service\_tax=0);

A screenshot of a SQL Plus terminal window. The terminal shows the following SQL query: `SQL> select distinct train_name from train where train_number in(select train_number from train_ticket_fare where superfast_charge!=0 and service_tax=0);` Below the query, it says "no rows selected".

```
SQL> select distinct train_name from train where train_number in(select train_number from train_ticket_fare where superfast_charge!=0 and service_tax=0);

no rows selected
```

7. Find the Passenger names who have booked for the train that starts from 'Chennai'.

**Sol:**

select distinct passenger\_name from passenger\_details where pnr\_no in(select pnr\_no from ticket where train\_number in(select train\_number from train where source='Chennai'));

```

SQL Plus
SQL> select distinct passenger_name from passenger_details where pnr_no in(select pnr_no from ticket where train_number in(select train_number from train where source='Chennai'));

no rows selected

```

8. Find the trains names that have all the AC coaches and the base fare is less than 3000 for each case.

**Sol:**

select distinct train\_name from train where train\_number in(select train\_no from train\_ticket\_fare where base\_fare<3000 and classes in('1A','2A','3A'));

```

SQL Plus
SQL> select distinct train_name from train where train_number in(select train_no from train_ticket_fare where base_fare<3000 and classes in('1A','2A','3A'));

no rows selected

```

### Use Join Query

9. Find the train names that stop in 'Katpadi'.

**Sol:**

select train.train\_name from train join train\_route on train.train\_number=train\_route.train\_no where train\_route.station\_name='Katpadi Junction';

```

SQL Plus
SQL> select train.train_name from train join train_route on train.train_number=train_route.train_no where train_route.station_name='Katpadi Junction';

no rows selected

```

10. Find the train names that are superfast and the service tax is zero.

**Sol:**

select train\_name from train, train\_ticket\_fare where train.train\_number=train\_ticket\_fare.train\_no and type='Superfast' and service\_tax!=0;

```

SQL Plus
SQL> select train_name from train, train_ticket_fare where train.train_number=train_ticket_fare.train_no and type='Superfast' and service_tax!=0;

TRAIN_NAME
-----
masbca sf exp

```



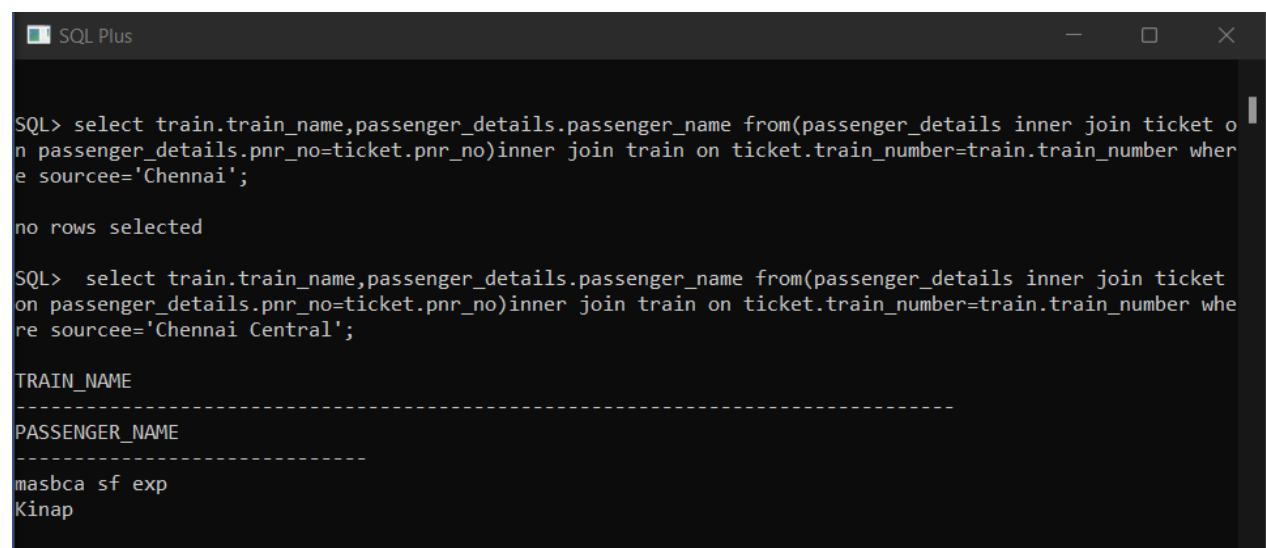
11. Find the Passenger name (and train name) who have booked for the train that starts from 'Chennai'.

**Sol:**

```
select train.train_name,passenger_details.passenger_name from(passenger_details inner join ticket
on passenger_details.pnr_no=ticket.pnr_no)inner join train on
ticket.train_number=train.train_number where sourcee='Chennai';
```

```
select train.train_name,passenger_details.passenger_name from(passenger_details inner join ticket
on passenger_details.pnr_no=ticket.pnr_no)inner join train on
ticket.train_number=train.train_number where sourcee='Chennai Central';
```

```
select train.train_name,passenger_details.passenger_name from(passenger_details inner join ticket
on passenger_details.pnr_no=ticket.pnr_no)inner join train on
ticket.train_number=train.train_number where sourcee='Chennai Egmore';
```



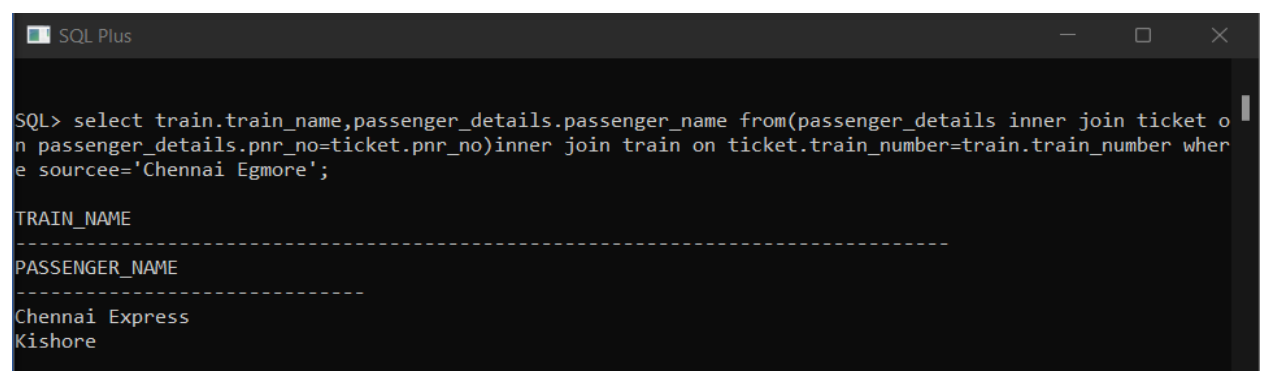
```
SQL Plus

SQL> select train.train_name,passenger_details.passenger_name from(passenger_details inner join ticket o
n passenger_details.pnr_no=ticket.pnr_no)inner join train on ticket.train_number=train.train_number whe
e sourcee='Chennai';

no rows selected

SQL> select train.train_name,passenger_details.passenger_name from(passenger_details inner join ticket
on passenger_details.pnr_no=ticket.pnr_no)inner join train on ticket.train_number=train.train_number whe
re sourcee='Chennai Central';

TRAIN_NAME
-----
PASSENGER_NAME
-----
masbca sf exp
Kinap
```



```
SQL Plus

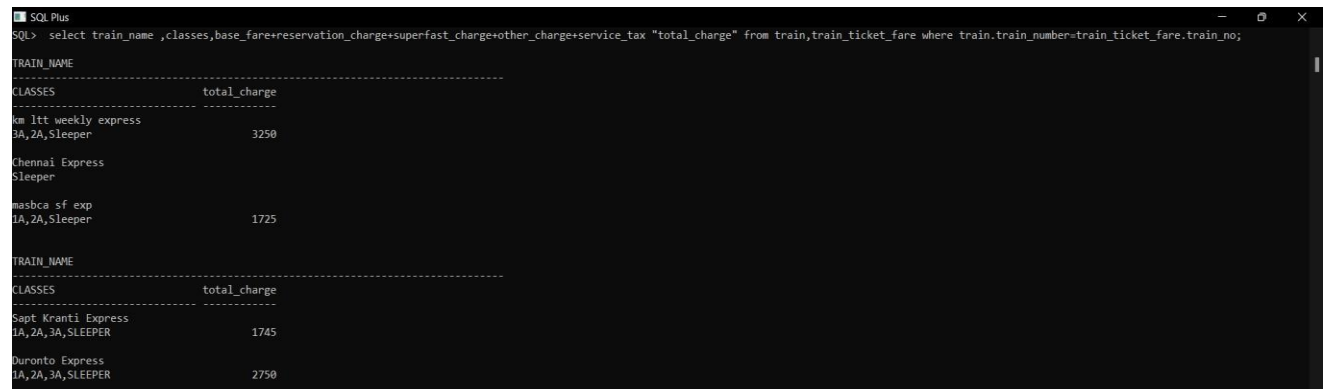
SQL> select train.train_name,passenger_details.passenger_name from(passenger_details inner join ticket o
n passenger_details.pnr_no=ticket.pnr_no)inner join train on ticket.train_number=train.train_number whe
e sourcee='Chennai Egmore';

TRAIN_NAME
-----
PASSENGER_NAME
-----
Chennai Express
Kishore
```

12. Display the trains' names, each type of class and the total fare for each type of class.

**Sol:**

```
select train_name
,classes,base_fare+reservation_charge+superfast_charge+other_charge+service_tax "total_charge"
from train,train_ticket_fare where train.train_number=train_ticket_fare.train_no;
```

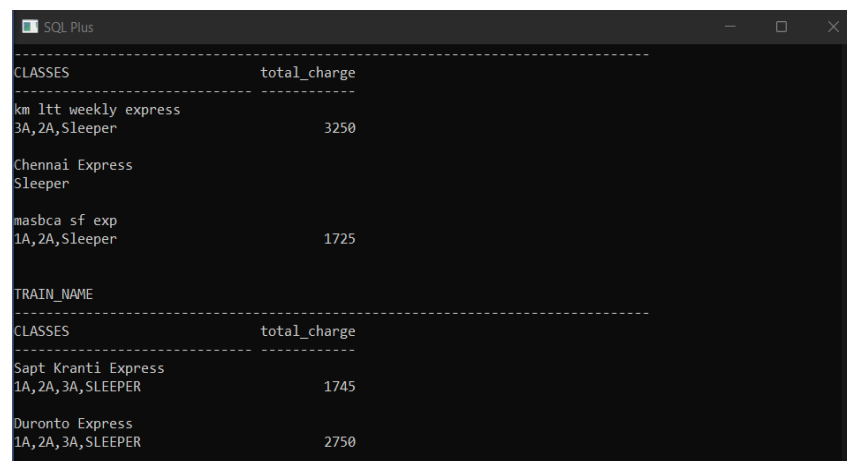


The screenshot shows the SQL Plus terminal with the following output:

TRAIN_NAME	CLASSES	total_charge
km ltt weekly express	3A,2A,Sleeper	3250
Chennai Express	Sleeper	
masbca sf exp	1A,2A,Sleeper	1725

TRAIN_NAME	CLASSES	total_charge
Sapt Kranti Express	1A,2A,3A,SLEEPER	1745
Duronto Express	1A,2A,3A,SLEEPER	2750



This screenshot shows the same SQL Plus terminal output as above:

TRAIN_NAME	CLASSES	total_charge
km ltt weekly express	3A,2A,Sleeper	3250
Chennai Express	Sleeper	
masbca sf exp	1A,2A,Sleeper	1725

TRAIN_NAME	CLASSES	total_charge
Sapt Kranti Express	1A,2A,3A,SLEEPER	1745
Duronto Express	1A,2A,3A,SLEEPER	2750

13. Display all the train details and the ticket details (if booked any).

**Sol:**

```
select * from train, ticket where train.train_number=ticket.train_number;
```

```
SQL Plus
SQL> select * from train, ticket where train.train_number=ticket.train_number;

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCEEE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE      PNR_NO      TRANSACTIONID FROM_STATION
-----
TO_STATION          DATE_OF_J CLASSID  DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
11018

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCEEE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE      PNR_NO      TRANSACTIONID FROM_STATION
-----
TO_STATION          DATE_OF_J CLASSID  DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
km ltt weekly express

TRAIN_NUMBER
-----
TRAIN_NAME
-----
```

```
SQL Plus

START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE      PNR_NO      TRANSACTIONID FROM_STATION
-----
TO_STATION          DATE_OF_J CLASSID  DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
Karaikal          LokmanyaTilak

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCEEE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE      PNR_NO      TRANSACTIONID FROM_STATION
-----
TO_STATION          DATE_OF_J CLASSID  DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
01-OCT-21 02.00.00 PM

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCEEE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
```

```

SQL Plus
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE PNR_NO TRANSACTIONID FROM_STATION
-----
TO_STATION DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
01-OCT-21 11.45.00 PM
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE PNR_NO TRANSACTIONID FROM_STATION
-----
TO_STATION DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
+01 01:40:00.000000
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----

```

```

SQL Plus
-----
TYPE PNR_NO TRANSACTIONID FROM_STATION
-----
TO_STATION DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
1266 3A,2A,Sleeper Monday
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE PNR_NO TRANSACTIONID FROM_STATION
-----
TO_STATION DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
Express 8674920651 7896354865 Karaikal
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE PNR_NO TRANSACTIONID FROM_STATION
-----
TO_STATION DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----

```

```

SQL Plus
-----
TRAIN_NUMBER
-----
Lokmanya Tilak          12-NOV-21 2A          19-SEP-21          3500
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE          PNR_NO          TRANSACTIONID FROM_STATION
-----
TO_STATION          DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
11018
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE          PNR_NO          TRANSACTIONID FROM_STATION
-----
TO_STATION          DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----

```

```

SQL Plus
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE          PNR_NO          TRANSACTIONID FROM_STATION
-----
TO_STATION          DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
12164
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE          PNR_NO          TRANSACTIONID FROM_STATION
-----
TO_STATION          DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
Chennai Express
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----

```

```

SQL Plus
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE PNR_NO TRANSACTIONID FROM_STATION
-----
TO_STATION DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
Chennai Egmore Dadar
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCEEE DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE PNR_NO TRANSACTIONID FROM_STATION
-----
TO_STATION DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
01-OCT-21 06.45.00 AM
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCEEE DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----

```

```

SQL Plus
-----
TYPE PNR_NO TRANSACTIONID FROM_STATION
-----
TO_STATION DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
01-OCT-21 06.00.00 AM
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCEEE DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE PNR_NO TRANSACTIONID FROM_STATION
-----
TO_STATION DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
+00 23:15:00.000000
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCEEE DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE PNR_NO TRANSACTIONID FROM_STATION
-----
TO_STATION DATE_OF_J CLASSID DATE_OF_B TOTAL_TICKET_FARE
-----

```

```

SQL Plus
-----
TRAIN_NUMBER
-----
      1274 Sleeper                      Monday, Tuesday
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCEE                      DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
      DISTANCE CLASSES                      DAYSS
-----
TYPE          PNR_NO          TRANSACTIONID FROM_STATION
-----
TO_STATION                      DATE_OF_J CLASSID  DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
Express      2345127890      7835682186 Vellore
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCEE                      DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
      DISTANCE CLASSES                      DAYSS
-----
TYPE          PNR_NO          TRANSACTIONID FROM_STATION
-----
TO_STATION                      DATE_OF_J CLASSID  DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
Ako1a                      12-DEC-21 Sleeper  18-JUL-21      400
-----

```

```

SQL Plus
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCEE                      DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
      DISTANCE CLASSES                      DAYSS
-----
TYPE          PNR_NO          TRANSACTIONID FROM_STATION
-----
TO_STATION                      DATE_OF_J CLASSID  DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
      12164
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCEE                      DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
      DISTANCE CLASSES                      DAYSS
-----
TYPE          PNR_NO          TRANSACTIONID FROM_STATION
-----
TO_STATION                      DATE_OF_J CLASSID  DATE_OF_B TOTAL_TICKET_FARE
-----
TRAIN_NUMBER
-----
-----
TRAIN_NUMBER
-----
TRAIN_NAME
-----

```

### Complex queries (use group by/group by having/join/nested)

14. Take the start station code and end station code and display the train details.

**Sol:**

select \* from train where source=&source and destination=&destination;

```
SQL Plus
SQL> select * from train where source=&source and destination=&destination;
Enter value for source: 'Chennai Egmore'
Enter value for destination: 'Dadar'
old 1: select * from train where source=&source and destination=&destination
new 1: select * from train where source='Chennai Egmore' and destination='Dadar'

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE
-----
12164

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE
-----
Chennai Express

TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
```



```

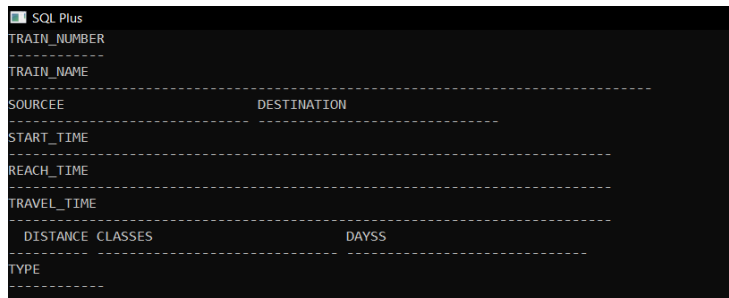
SQL Plus
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE
-----
Chennai Egmore          Dadar
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE
-----
01-OCT-21 06.45.00 AM
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE
-----
01-OCT-21 06.00.00 AM
TRAIN_NUMBER
-----

```

```

SQL Plus
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE
-----
+00 23:15:00.000000
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE
-----
1274 Sleeper          Monday,Tuesday
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES DAYSS
-----
TYPE
-----

```



```

SQL Plus
TRAIN_NUMBER
-----
TRAIN_NAME
-----
SOURCE          DESTINATION
-----
START_TIME
-----
REACH_TIME
-----
TRAVEL_TIME
-----
DISTANCE CLASSES          DAYSS
-----
TYPE
-----

```

15. List the train names and the number of sub stations it has.

**Sol:**

select train.train\_name from train,train\_route where train.train\_number=train\_route.train\_no group by train.train\_name having count(station\_code)>=5;

```

SQL> select train.train_name from train,train_route where train.train_number=train_route.train_no group by train.train_name having count(station_code)>=5;

no rows selected

```

16. List the stations where all types of trains stop.

**Sol:**

select station\_name from train\_route where not exists(select type from train minus select type from train where train.train\_number=train\_route.train\_no);

```

SQL> select station_name from train_route where not exists(select type from train minus select type from train where train.train_number=train_route.train_no);

no rows selected

```

17. List the trains names that has at least four bookings

**Sol:**

select train\_name from train,ticket,passenger\_details where train.train\_number=ticket.train\_number and ticket.pnr\_no=passenger\_details.pnr\_no group by train\_name having count(distinct pnr\_no)>=5;

```

SQL> select train_name from train,ticket,passenger_details where train.train_number=ticket.train_number and ticket.pnr_no=passenger_details.pnr_no group by train_name having count(distinct pnr_no)>=5;
select train_name from train,ticket,passenger_details where train.train_number=ticket.train_number and ticket.pnr_no=passenger_details.pnr_no group by train_name having count(distinct pnr_no)>=5
*
ERROR at line 1:
ORA-00918: column ambiguously defined

SQL>

```

## **PL/SQL block to**

### **1. Print the Fibonacci series.**

```
set serveroutput on;
```

```
declare
```

```
a number := 0;
```

```
b number := 1;
```

```
c number := 0;
```

```
i number := 0;
```

```
begin
```

```
dbms_output.put_line(a);
```

```
dbms_output.put_line(b);
```

```
while i<20 loop
```

```
c:=a+b;
```

```
dbms_output.put_line(c);
```

```
b:=a;
```

```
a:=c;
```

```
i:=i+1;
```

```
end loop;
```

```
end;
```

```
/
```

SQL Plus

```
SQL>
SQL> set serveroutput on;
SQL> declare
  2  a number := 0;
  3  b number := 1;
  4  c number := 0;
  5  i number := 0;
  6  begin
  7  dbms_output.put_line(a);
  8  dbms_output.put_line(b);
  9  while i<20 loop
10  c:=a+b;
11  dbms_output.put_line(c);
12  b:=a;
13  a:=c;
14  i:=i+1;
15  end loop;
16  end;
17  /
0
1
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765

PL/SQL procedure successfully completed.

SQL> _
```

## 2. Print the factorial of a given number.

```
create function fact(x number)
```

```
return number
```

```
is
```

```
f number;
```

```
begin
```

```
if x=0 then
```

```
f:=1;
```

```
else
```

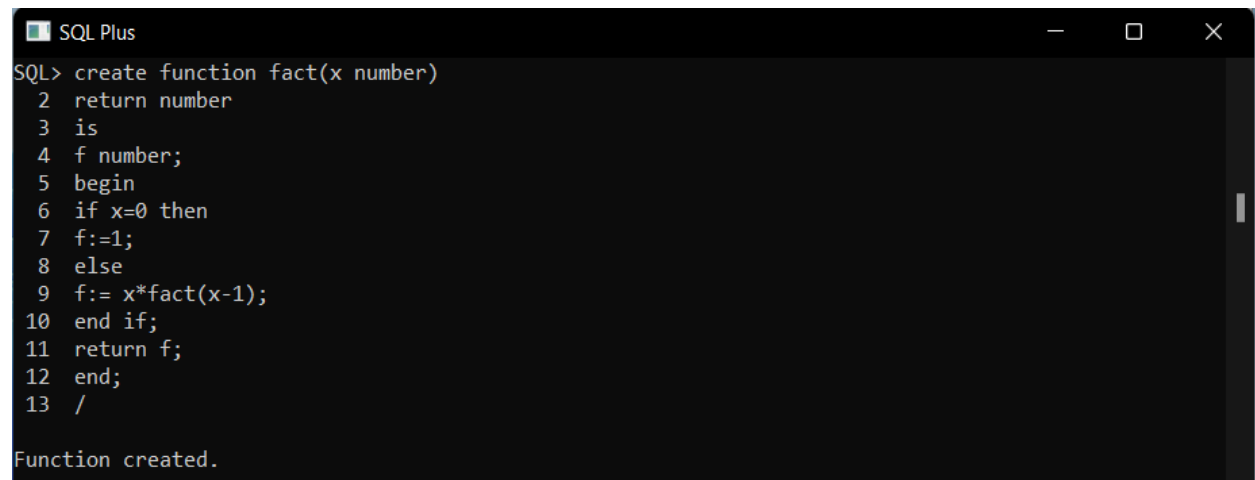
```
f:= x*fact(x-1);
```

```
end if;
```

```
return f;
```

```
end;
```

```
/
```

A screenshot of a SQL Plus window titled "SQL Plus". The window has a dark background with light-colored text. The text shows the creation of a function named "fact" which takes a number "x" as input. The function body includes a "return number" statement, followed by "is", then "f number;", "begin", an "if x=0 then" block where "f:=1;", an "else" block where "f:= x\*fact(x-1);", "end if;", "return f;", "end;", and finally a "/" symbol to end the function definition. Below the code, the message "Function created." is displayed. The window has standard OS controls (minimize, maximize, close) in the top right corner.

```
SQL> create function fact(x number)
  2  return number
  3  is
  4  f number;
  5  begin
  6  if x=0 then
  7  f:=1;
  8  else
  9  f:= x*fact(x-1);
 10  end if;
 11  return f;
 12  end;
 13  /

Function created.
```

```
declare
```

```
num number;
```

```
factorial number;
```

```
begin
```

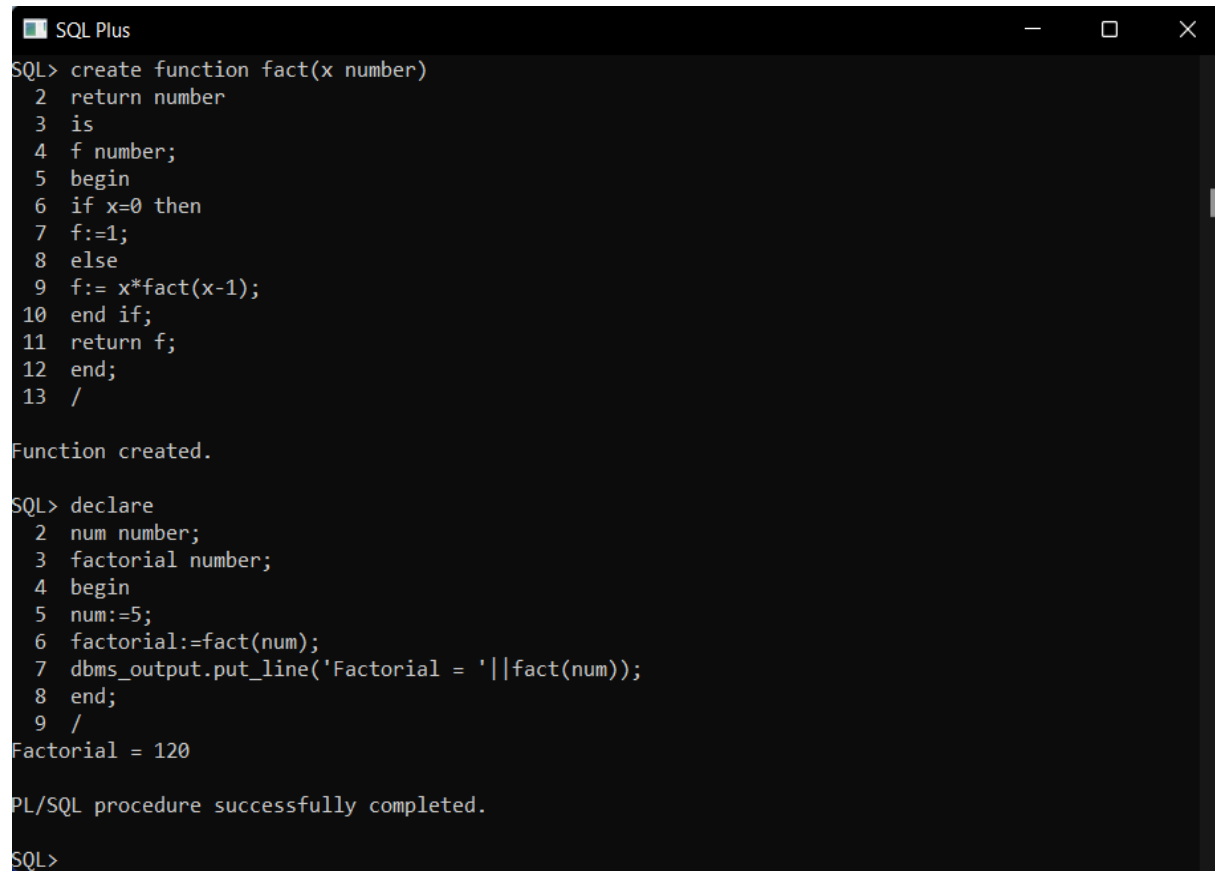
```
num:=5;
```

```
factorial:=fact(num);
```

```
dbms_output.put_line('Factorial = ' || fact(num));
```

end;

/

A screenshot of a SQL Plus window titled "SQL Plus". The window has a dark background with light-colored text. The text shows the creation of a function named 'fact' and its execution. The function 'fact' takes a number 'x' as input and returns its factorial. The execution part declares a variable 'num' as 5 and a variable 'factorial' to hold the result. It then calls the 'fact' function with 'num' and prints the result using 'dbms\_output.put\_line'. The output shows 'Factorial = 120' and a message 'PL/SQL procedure successfully completed.'.

```
SQL> create function fact(x number)
  2  return number
  3  is
  4  f number;
  5  begin
  6  if x=0 then
  7  f:=1;
  8  else
  9  f:= x*fact(x-1);
 10  end if;
 11  return f;
 12  end;
 13  /

Function created.

SQL> declare
  2  num number;
  3  factorial number;
  4  begin
  5  num:=5;
  6  factorial:=fact(num);
  7  dbms_output.put_line('Factorial = '||fact(num));
  8  end;
  9  /

Factorial = 120

PL/SQL procedure successfully completed.

SQL>
```

### 3. Print 'NOT confirmed' based on the reservation status of a particular passenger.

set serveroutput on;

declare

cursor pass\_cur is

select passenger\_name,reservation\_status from passenger\_details where  
reservation\_status!='CNF';

pass\_rec pass\_cur%rowtype;

begin

open pass\_cur;

loop

fetch pass\_cur into pass\_rec;

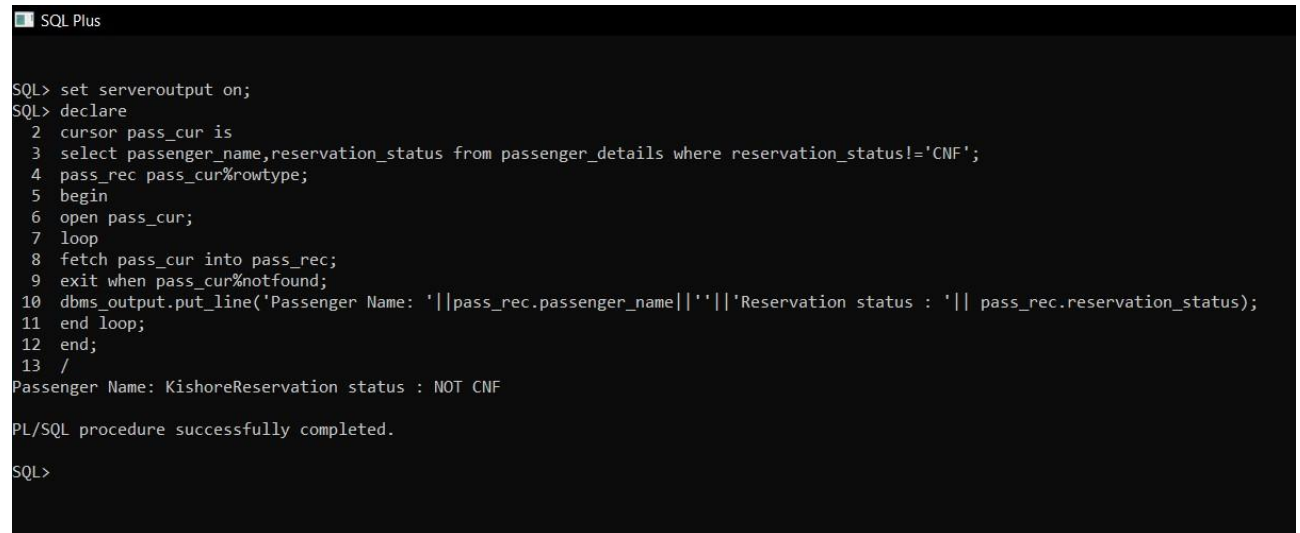
exit when pass\_cur%notfound;

```
dbms_output.put_line('Passenger Name: ' || pass_rec.passenger_name || ' || 'Reservation  
status : ' || pass_rec.reservation_status);
```

```
end loop;
```

```
end;
```

```
/
```

A screenshot of a SQL Plus terminal window. The window title is "SQL Plus". The terminal shows a PL/SQL procedure being executed. The code starts with "SQL> set serveroutput on;" followed by "SQL> declare". Then, a cursor "pass\_cur" is declared to select "passenger\_name" and "reservation\_status" from "passenger\_details" where "reservation\_status" is not 'CNF'. The procedure then opens the cursor, enters a loop to fetch rows into "pass\_rec", and prints the passenger name and reservation status. The output shows "Passenger Name: Kishore" and "Reservation status : NOT CNF". The procedure ends with "end;" and a slash. The terminal then shows "PL/SQL procedure successfully completed." and "SQL>".

```
SQL> set serveroutput on;  
SQL> declare  
  2  cursor pass_cur is  
  3  select passenger_name,reservation_status from passenger_details where reservation_status!='CNF';  
  4  pass_rec pass_cur%rowtype;  
  5  begin  
  6  open pass_cur;  
  7  loop  
  8  fetch pass_cur into pass_rec;  
  9  exit when pass_cur%notfound;  
 10  dbms_output.put_line('Passenger Name: ' || pass_rec.passenger_name || ' || 'Reservation status : ' || pass_rec.reservation_status);  
 11  end loop;  
 12  end;  
 13  /  
Passenger Name: KishoreReservation status : NOT CNF  
  
PL/SQL procedure successfully completed.  
  
SQL>
```

**4. Print the total seats available for a particular train and for a particular class. Write a cursor for the following**

```
alter table train add seat number(5);
```

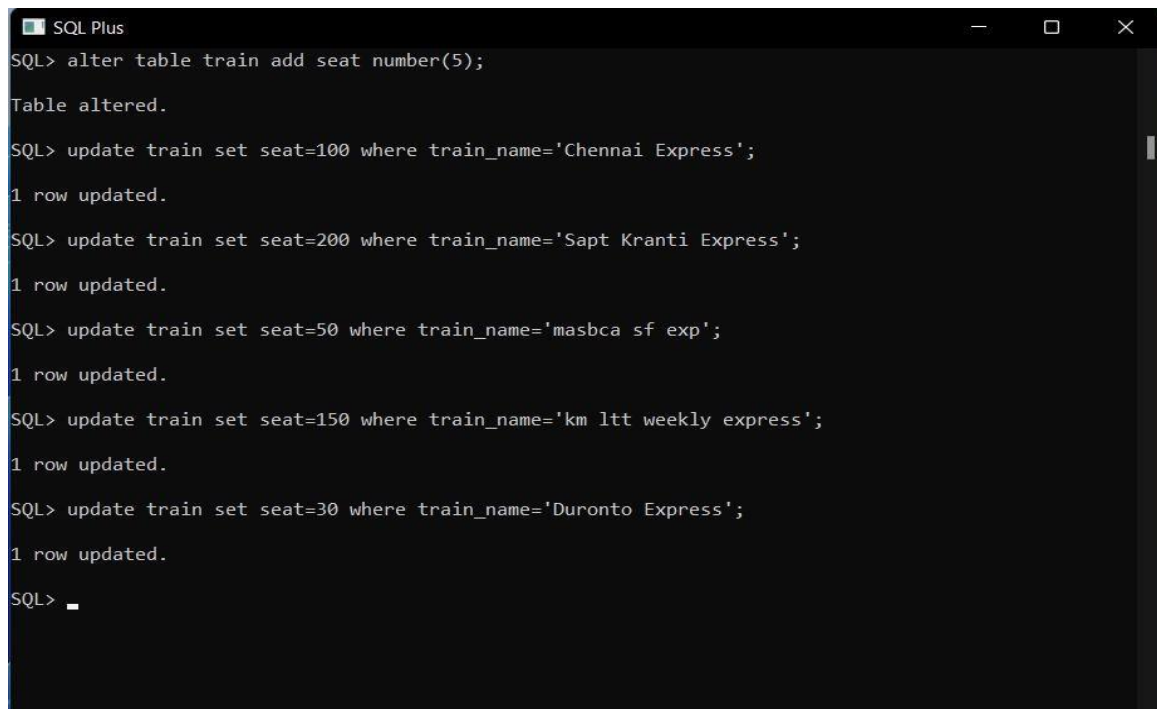
```
update train set seat=100 where train_name='Chennai Express';
```

```
update train set seat=200 where train_name='Sapt Kranti Express';
```

```
update train set seat=50 where train_name='masbca sf exp';
```

```
update train set seat=150 where train_name='km ltt weekly express';
```

```
update train set seat=30 where train_name='Duronto Express';
```

A screenshot of a SQL Plus terminal window. The window has a title bar with the text "SQL Plus" and standard window controls (minimize, maximize, close). The terminal displays a series of SQL commands and their outputs. The commands are: 1. "alter table train add seat number(5);" followed by "Table altered." 2. "update train set seat=100 where train\_name='Chennai Express';" followed by "1 row updated." 3. "update train set seat=200 where train\_name='Sapt Kranti Express';" followed by "1 row updated." 4. "update train set seat=50 where train\_name='masbca sf exp';" followed by "1 row updated." 5. "update train set seat=150 where train\_name='km ltt weekly express';" followed by "1 row updated." 6. "update train set seat=30 where train\_name='Duronto Express';" followed by "1 row updated." The prompt "SQL>" is visible at the end of the last command.

```
SQL Plus
SQL> alter table train add seat number(5);
Table altered.
SQL> update train set seat=100 where train_name='Chennai Express';
1 row updated.
SQL> update train set seat=200 where train_name='Sapt Kranti Express';
1 row updated.
SQL> update train set seat=50 where train_name='masbca sf exp';
1 row updated.
SQL> update train set seat=150 where train_name='km ltt weekly express';
1 row updated.
SQL> update train set seat=30 where train_name='Duronto Express';
1 row updated.
SQL> _
```

set serveroutput on;

declare

cursor seat is

select train\_name,classes,seat from train where classes='1A';

seat\_rec seat%rowtype;

begin

open seat;

loop

fetch seat into seat\_rec;

exit when seat%notfound;

dbms\_output.put\_line('Train Name:' ||

seat\_rec.train\_name || " || 'Class:' || seat\_rec.classes || 'total seats available:' || seat\_rec.seat);

end loop;

end;

/



```
SQL Plus

SQL> set serveroutput on;
SQL> declare
  2 cursor seat is
  3 select train_name,classes,seat from train where classes='1A';
  4 seat_rec seat%rowtype;
  5 begin
  6 open seat;
  7 loop
  8 fetch seat into seat_rec;
  9 exit when seat%notfound;
 10 dbms_output.put_line('Train Name:' || seat_rec.train_name||''||'Class:'||seat_rec.classes||'total seats available:'||seat_rec.seat);
 11 end loop;
 12 end;
 13 /

PL/SQL procedure successfully completed.
```

```
SQL Plus

SQL> update train set seat=30 where train_name='Duronto Express';

1 row updated.

SQL> set serveroutput on;
SQL> declare
  2 cursor seat is
  3 select train_name,classes,seat from train where classes='1A';
  4 seat_rec seat%rowtype;
  5 begin
  6 open seat;
  7 loop
  8 fetch seat into seat_rec;
  9 exit when seat%notfound;
 10 dbms_output.put_line('Train Name:' || seat_rec.train_name||''||'Class:'||seat_rec.classes||'to
tal seats available:'||seat_rec.seat);
 11 end loop;
 12 end;
 13 /

PL/SQL procedure successfully completed.
```

## 5. Retrieve the passenger details for “x” train number and given journey date.

set serveroutput on;

declare

train number(5);

journey\_date date;

cursor pas\_cur is

select passenger\_name,age from passenger\_details where pnr\_no in(select pnr\_no from ticket where train\_number=&train and date\_of\_journey='&journey\_date');

pas\_rec pas\_cur%rowtype;

begin

open pas\_cur;

loop

fetch pas\_cur into pas\_rec;

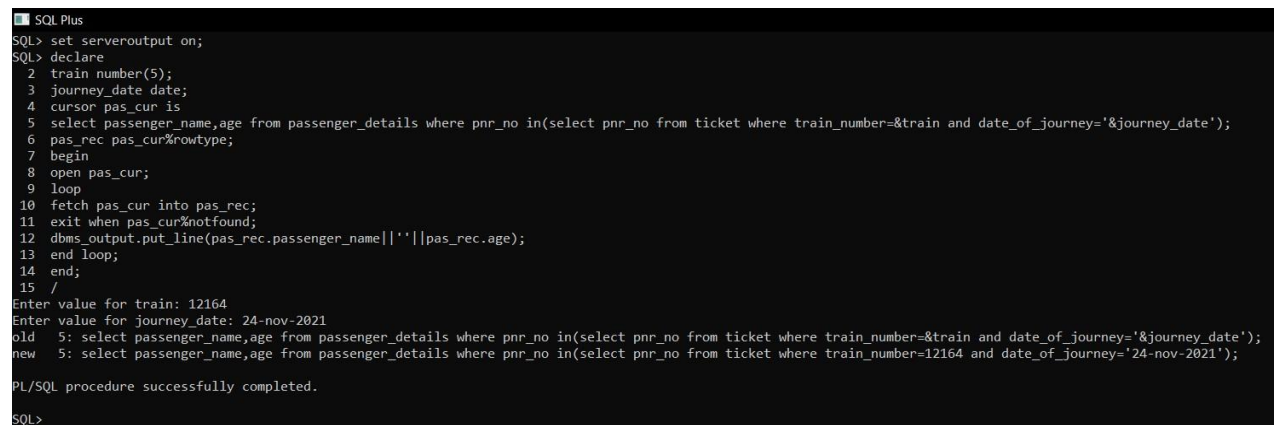
exit when pas\_cur%notfound;

dbms\_output.put\_line(pas\_rec.passenger\_name||' '||pas\_rec.age);

end loop;

end;

/



```
SQL Plus
SQL> set serveroutput on;
SQL> declare
  2 train number(5);
  3 journey_date date;
  4 cursor pas_cur is
  5 select passenger_name,age from passenger_details where pnr_no in(select pnr_no from ticket where train_number=&train and date_of_journey='&journey_date');
  6 pas_rec pas_cur%rowtype;
  7 begin
  8 open pas_cur;
  9 loop
 10 fetch pas_cur into pas_rec;
 11 exit when pas_cur%notfound;
 12 dbms_output.put_line(pas_rec.passenger_name||' '||pas_rec.age);
 13 end loop;
 14 end;
 15 /
Enter value for train: 12164
Enter value for journey_date: 24-nov-2021
old 5: select passenger_name,age from passenger_details where pnr_no in(select pnr_no from ticket where train_number=&train and date_of_journey='&journey_date');
new 5: select passenger_name,age from passenger_details where pnr_no in(select pnr_no from ticket where train_number=12164 and date_of_journey='24-nov-2021');

PL/SQL procedure successfully completed.
SQL>
```

```
SQL Plus
SQL> set serveroutput on;
SQL> declare
  2 train number(5);
  3 journey_date date;
  4 cursor pas_cur is
  5 select passenger_name,age from passenger_details where pnr_no in(select pnr_no from ticket whe
re train_number=&train and date_of_journey='&journey_date');
  6 pas_rec pas_cur%rowtype;
  7 begin
  8 open pas_cur;
  9 loop
 10 fetch pas_cur into pas_rec;
 11 exit when pas_cur%notfound;
 12 dbms_output.put_line(pas_rec.passenger_name||' '||pas_rec.age);
 13 end loop;
 14 end;
 15 /
Enter value for train: 12164
Enter value for journey_date: 24-nov-2021
old 5: select passenger_name,age from passenger_details where pnr_no in(select pnr_no from ticket
where train_number=&train and date_of_journey='&journey_date');
new 5: select passenger_name,age from passenger_details where pnr_no in(select pnr_no from ticket
where train_number=12164 and date_of_journey='24-nov-2021');

PL/SQL procedure successfully completed.

SQL>
```

## 6. Display the train name and the substation names.

```
set serveroutput on;

declare

t train%rowtype;

tr train_route%rowtype;

cursor train1 is select * from train;

n number(10);

cursor t_r is select * from train_route where train_no=n;

begin

for t in train1

loop

n :=t.train_number;

dbms_output.put_line(t.train_name);

for tr in t_r

loop

dbms_output.put_line(tr.station_name);

end loop;
```

end loop;

end;

/

```
SQL Plus
SQL>
SQL> set serveroutput on;
SQL> declare
  2  t train%rowtype;
  3  tr train_route%rowtype;
  4  cursor train1 is select * from train;
  5  n number(10);
  6  cursor t_r is select * from train_route where train_no=n;
  7  begin
  8  for t in train1
  9  loop
 10  n :=t.train_number;
 11  dbms_output.put_line(t.train_name);
 12  for tr in t_r
 13  loop
 14  dbms_output.put_line(tr.station_name);
 15  end loop;
 16  end loop;
 17  end;
 18  /
Chennai Express
Chennai Egmore
Sapt Kranti Express
New Delhi
masbca sf exp
Chennai Central
km ltt weekly express
Karaikal
Duronto Express
Nizamuddin Terminus

PL/SQL procedure successfully completed.

SQL>
SQL>
SQL> _
```

## 7. Display the fare details of a particular train (use basic exceptions)

```
set serveroutput on;

declare

train number(5) := &train;

cursor pas_cur is

select base_fare from train_ticket_fare where train_no=train;

pas_rec pas_cur%rowtype;

begin

open pas_cur;

loop

fetch pas_cur into pas_rec;

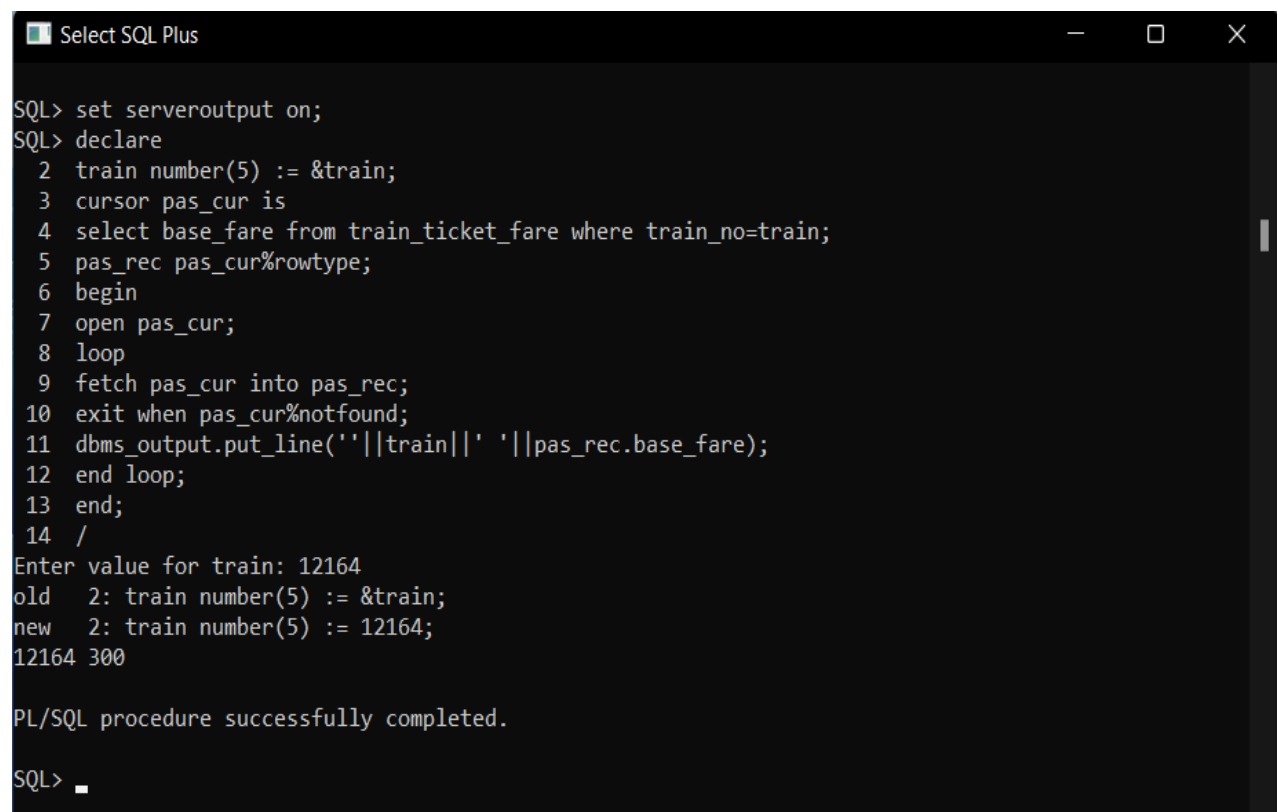
exit when pas_cur%notfound;

dbms_output.put_line(''||train||' '||pas_rec.base_fare);

end loop;

end;

/
```

A screenshot of a SQL Plus window titled "Select SQL Plus". The window has a dark background with light-colored text. The SQL prompt "SQL>" is followed by the PL/SQL code from the previous block. The code is executed line by line, with line numbers 2 through 14 visible. The prompt "Enter value for train:" is followed by the input "12164". The output shows the variable "train" being assigned the value "12164". The final output is "12164 300", which is the result of the query. The message "PL/SQL procedure successfully completed." is displayed at the bottom. The prompt "SQL>" is followed by a cursor character "\_".

```
Select SQL Plus

SQL> set serveroutput on;
SQL> declare
  2 train number(5) := &train;
  3 cursor pas_cur is
  4 select base_fare from train_ticket_fare where train_no=train;
  5 pas_rec pas_cur%rowtype;
  6 begin
  7 open pas_cur;
  8 loop
  9 fetch pas_cur into pas_rec;
 10 exit when pas_cur%notfound;
 11 dbms_output.put_line(''||train||' '||pas_rec.base_fare);
 12 end loop;
 13 end;
 14 /
Enter value for train: 12164
old  2: train number(5) := &train;
new  2: train number(5) := 12164;
12164 300

PL/SQL procedure successfully completed.

SQL> _
```

**Write a PL/SQL procedure to**

**8. Find PNR No. of a passengers for a given source and a destination.**

```
set serveroutput on;
```

```
declare
```

```
source varchar(20);
```

```
dest varchar(20);
```

```
cursor pas_cur is
```

```
select pnr_no from ticket where from_station = '&source' and to_station='&dest';
```

```
pas_rec pas_cur%rowtype;
```

```
begin
```

```
open pas_cur;
```

```
loop
```

```
fetch pas_cur into pas_rec;
```

```
exit when pas_cur%notfound;
```

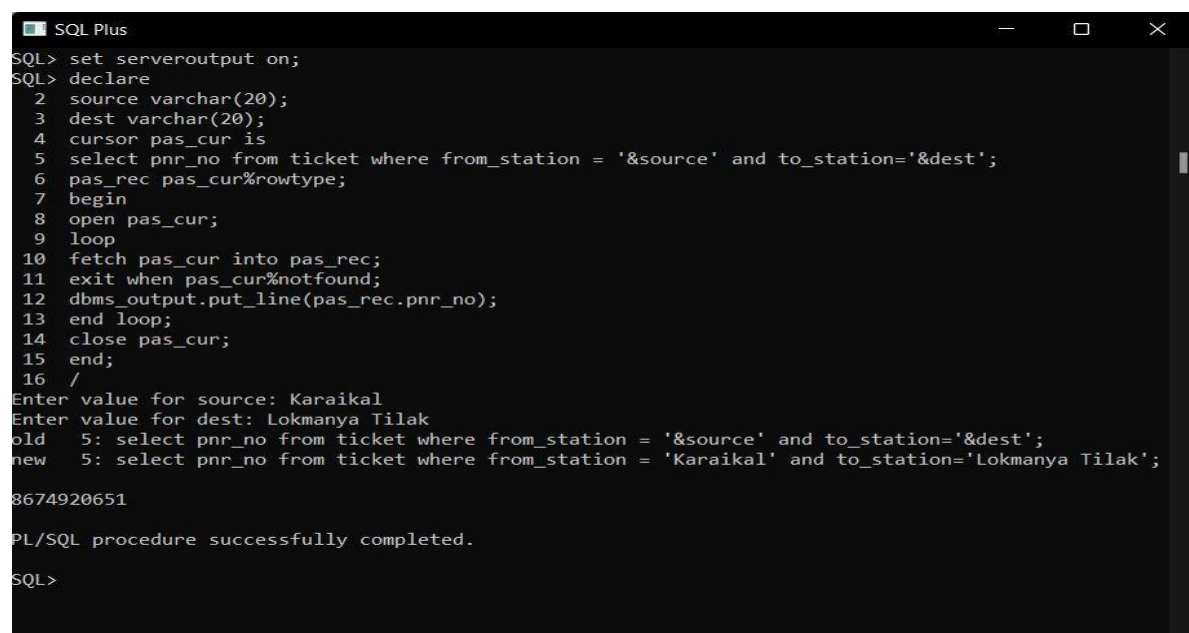
```
dbms_output.put_line(pas_rec.pnr_no);
```

```
end loop;
```

```
close pas_cur;
```

```
end;
```

```
/
```

A screenshot of a SQL Plus window titled "SQL Plus". The window has a dark background with white text. It shows the execution of a PL/SQL procedure. The code is entered line by line, and the output is displayed. The procedure is designed to find the PNR number for a given source and destination. The source is 'Karaikal' and the destination is 'Lokmanya Tilak'. The output shows the PNR number '8674920651'. The procedure is successfully completed.

```
SQL> set serveroutput on;
SQL> declare
  2 source varchar(20);
  3 dest varchar(20);
  4 cursor pas_cur is
  5 select pnr_no from ticket where from_station = '&source' and to_station='&dest';
  6 pas_rec pas_cur%rowtype;
  7 begin
  8 open pas_cur;
  9 loop
 10 fetch pas_cur into pas_rec;
 11 exit when pas_cur%notfound;
 12 dbms_output.put_line(pas_rec.pnr_no);
 13 end loop;
 14 close pas_cur;
 15 end;
 16 /
Enter value for source: Karaikal
Enter value for dest: Lokmanya Tilak
old 5: select pnr_no from ticket where from_station = '&source' and to_station='&dest';
new 5: select pnr_no from ticket where from_station = 'Karaikal' and to_station='Lokmanya Tilak';

8674920651

PL/SQL procedure successfully completed.

SQL>
```

**Write a PL/SQL function to**

**9. Get the PNR No and return the total ticket fare**

```
set serveroutput on;

declare

n number(10);

pd ticket%rowtype;

begin

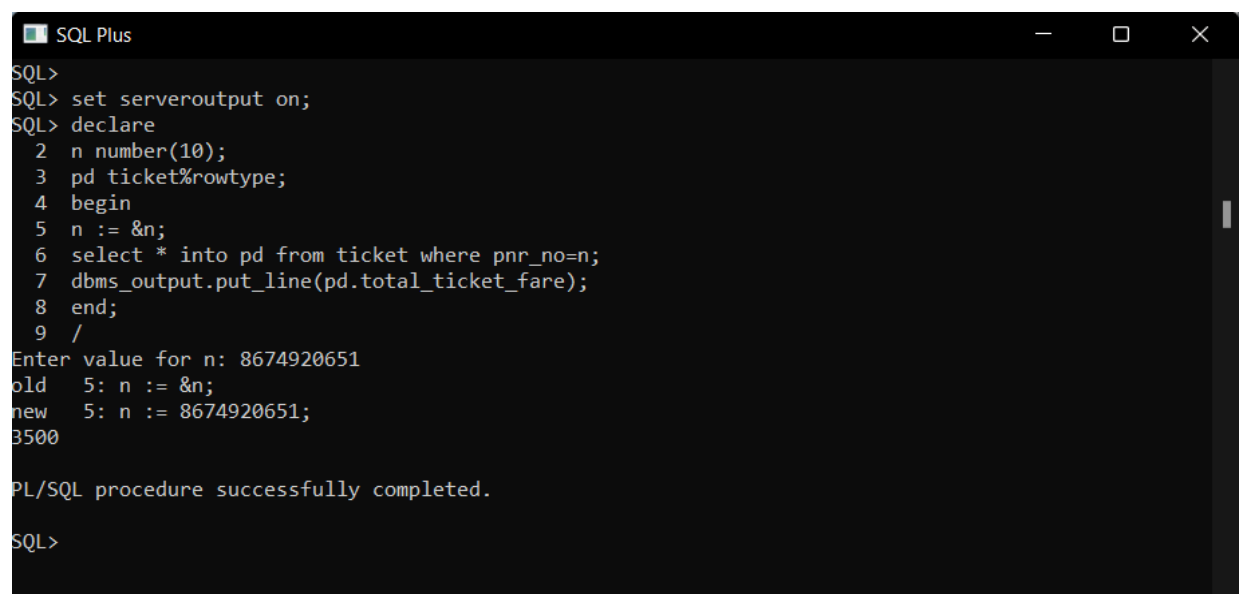
n := &n;

select * into pd from ticket where pnr_no=n;

dbms_output.put_line(pd.total_ticket_fare);

end;

/
```

A screenshot of a SQL Plus terminal window. The window title is "SQL Plus". The terminal shows the following commands and output:

```
SQL>
SQL> set serveroutput on;
SQL> declare
  2 n number(10);
  3 pd ticket%rowtype;
  4 begin
  5 n := &n;
  6 select * into pd from ticket where pnr_no=n;
  7 dbms_output.put_line(pd.total_ticket_fare);
  8 end;
  9 /
Enter value for n: 8674920651
old   5: n := &n;
new   5: n := 8674920651;
3500

PL/SQL procedure successfully completed.

SQL>
```

*Thank  
you*