

# ASSIGNMENT – 2

## Railway Reservation System

Name: Mathew Jerry Meleth

Reg No: 17BIT0050

Slot: L3+L4

### WEEK-4

Use SQL PLUS functions to.

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

SELECT \* FROM TICKET WHERE date\_of\_journey=to\_date(<<select add\_months(sysdate,1) from dual>>);

```
SQL> select *from tticket where date_of_journey=to_date(<<select add_months(sysdate,1) from dual>>);
```

PNRNO	TARNSACTIONID	FROM_STATION	TO_STATION	DATE_OF_J
78964	102	vellore	jsr	20-OCT-17
20-SEP-17	4896	6789		

```
SQL>
```

2. Print the train names in upper case.

SELECT upper<name> from TRAIN;

```
SQL> select upper<name> from tttrain;
```

UPPER<NAME>
TATA ALLEPPEY
GETANJALI

3. Print the passenger names with left padding character.

```
SELECT LPAD<name,25,'-'> FROM PASSENGER;
```

```
SQL> SELECT LPAD<NAME,25,'-'> FROM PASSENGER;

LPAD<NAME,25,'-'>
-----
-----SWETA
-----AKSHAY
-----RISHABH
-----NISHANT
-----SHIVAM
-----ADARSH
-----RAJEEV

7 rows selected.
```

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

```
SELECT STATION_CODE, TRAIN_NO, TRANSLATE<station_code,'K','M'> FROM TRAIN_ROUTE;
```

```
SQL> SELECT STATION_CODE, TRAIN_NO, TRANSLATE<STATION_CODE,'K','M'> FROM TRAIN_ROUTE;

STATION_CO  TRAIN_NO  TRANSLATE<
-----
KPD          1435      MPD
KNR          5677      MNR

SQL>
```

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

```
SELECT class, translate<class,'1A','POT'> FROM TRAIN_TICKET_FARE;
```

```
SQL> select class, translate<class,'1A','POT'> from train_ticket_fare;

CLA  TRA
---  ---
1A   PO
2A   20
3A   30
3A   30
SL   SL
1A   PO

6 rows selected.
```

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

```
SELECT NULLIF(BASE_FARE,0) FROM TRAIN_TICKET_FARE;
```

```
SQL> SELECT NULLIF<BASE_FARE,0> FROM TRAIN_TICKET_FARE;

NULLIF<BASE_FARE,0>
-----
334
345
343
343
545
459

6 rows selected.
```

#### 7. Display the pnrno and transaction id, if transaction id is null, print 'not generated'.

```
SELECT PNRNO, NULLIF<TRANSACTIONID,0> FROM TICKET;
```

```
SQL> SELECT PNRNO, NULLIF<TARNSACTIONID,0> FROM TTICKET;
```

PNRNO	NULLIF<TARNSACTIONID,0>
14350	100
76554	101
78964	102
56978	104
45698	105

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

```
SELECT TO_CHAR<DATE_OF_JOURNEY,'Dth MONTH YYYY'> FROM TICKET;
```

```
SQL> SELECT TO_CHAR<DATE_OF_JOURNEY,'Dth MONTH YYYY'> FROM TTICKET;
```

TO_CHAR<DATE_OF_JO
6th NOVEMBER 2007
1st NOVEMBER 2017
6th OCTOBER 2017
4th OCTOBER 2017
7th DECEMBER 2017

9. Find the maximum fare (total fare).

```
SELECT MAX<TOTAL_TICKET_FARE> FROM TICKET;
```

```
SQL> SELECT MAX<TOTAL_TICKET_FARE> FROM TTICKET;
```

MAX<TOTAL_TICKET_FARE>
4896

```
SQL>
```

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

```
SELECT AVG<AGE> FROM TICKET;
```

```
SQL> SELECT AVG<AGE> FROM PASSENGER;
```

AVG<AGE>
18.7142857

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

```
SELECT MAX<STATION_CODE> FROM TRAIN_ROUTE;
```

```
SQL> SELECT MAX<STATION_CODE> FROM TRAIN_ROUTE;
```

MAX<STATION_CODE>
KPD

```
SQL>
```

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 TTICKET;
ROUND<TOTAL_TICKET_FARE>
-----
          4568
          2365
          4896
          3695
          2698

SQL> _
```

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

ALTER TABLE TRAIN\_ROUTE ADD HALT\_TIME INTERVAL DAY TO SECOND;

```
SQL> ALTER TABLE TRAIN_ROUTE
  2  ADD  HALT_TIME INTERVAL DAY  TO SECOND;

Table altered.

SQL>
```

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

UPDATE TRAIN\_ROUTE SET HALT\_TIME=DEPART\_TIME-ARRIVAL\_TIME;

```
SQL> UPDATE  TRAIN_ROUTE SET HALT_TIME=DEPART_TIME-ARRIVAL_TIME ;

2 rows updated.
```

High Level:

### 15. Update values to arrival time and depart time using conversion functions.

select to\_char(arrival\_time, 'yyyy/mm/dd') FROM Train\_Route; select to\_char(depart\_time, 'yyyy/mm/dd') FROM Train\_Route;

```
SQL> select to_char<arrival_time, 'yyyy/mm/dd'> FROM Train_Route;

TO_CHAR<AR
-----
2017/09/01
2017/09/01
```

```
SQL> select to_char<depart_time, 'yyyy/mm/dd'> FROM Train_Route;

TO_CHAR<DE
-----
2017/09/01
2017/09/01
```

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

select to\_char(arrival\_time,'HH24:MI') from Train\_Route; select to\_char(depart\_time,'HH24:MI') from Train\_Route;

```
SQL> select to_char<arrival_time,'HH24:MI'> from Train_Route;

TO_CH
-----
03:30
03:30
```

```
SQL> select to_char(depart_time,'HH24:MI') from Train_Route;
TO_CH
-----
04:30
04:30
```

## WEEK-5

Write Queries to.

Use SET Operators

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

Select train\_no from train

Minus

Select train\_no from ticket;

```
SQL> Select train_number from tttrain
2 Minus
3 Select train_number from tticket;
no rows selected
```

### 2. Find the train names that donot have a first AC class coach.

Select train\_no from train\_ticket\_fare

Minus

Select train\_no from train\_ticket\_fare where class='1A';

```
SQL> Select train_number from train_ticket_fare
2 Minus
3 Select train_number from train_ticket_fare where class='1A';
TRAIN_NUMBER
-----
5667
6789
7650
9675
```

### 3. Print all the PNR nos available in the database.

Select PNR\_NO from ticket

Union

Select PNR\_NO from Passenger;

```
SQL> Select PNR_NO from tticket
2 Union
3 Select PNR_NO from Passenger;

  PNR_NO
-----
10859
12345
14350
15478
45675
45678
45687
45698
56978
76554
78964

11 rows selected.
```

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

Select pnr\_no,name from passenger where pnr\_no in

(Select PNR\_NO from passenger

Intersect

Select PNR\_NO from ticket where to\_station='pune');

```
SQL> Select pnr_no,name from passenger where pnr_no in
2 <
3 Select PNR_NO from passenger
4 Intersect
5 Select PNR_NO from tticket where to_station='pune'
6 >;

  PNR_NO NAME
-----
14350 SHIVAM
```

### Use Nested Query(in Operators)

#### 1. Find the train names that stop in 'Katpadi'.

select name from train where train\_no in ( select train\_no from train\_route where name='katpadi');

```
SQL> select name from train where train_number in < select train_number from tra
in_route where name='katpadi'>;

NAME
-----
Chennai Express
```

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

Select name from train

Where train\_no in (select train\_no from train\_ticket\_fare where superfast\_charge !=0 AND service\_charge =0) ;

```
SQL> Select name from tttrain
2 Where train_number in <select train_number from train_ticket_fare where sup
erfast_charge !=0 AND service_charge =0> ;

no rows selected
```

#### 3. Find the Passenger name who have booked for the train that starts from 'chennai'.

Select name from passenger where

PNR\_NO in ( select PNR\_NO from ticket where train\_no in (select train\_no from train where source='chennai'));

```
SQL> Select name from passenger where
2 PNR_NO in ( select PNR_NO from tticket where train_number in (select train_
number from tttrain where source='chennai')));

NAME
-----
SHIVAM
```

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

Select name from train where train\_no in  
 (select train\_no from train\_ticket\_fare where base\_fare < 3000  
 Intersect  
 Select train\_no from train\_ticket\_fare where  
 Class ='1A' AND Class = '2A' AND CLASS ='3A')

```
SQL>
2 Select name from tttrain where train_number in
3 (select train_number from train_ticket_fare where base_fare < 3000
4 Intersect
5 Select train_number from train_ticket_fare where
6 Class ='1A' AND Class = '2A' AND CLASS ='3A')
;

no rows selected
```

## Use Join Query

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

SELECT name FROM train INNER JOIN ticket ON train.train\_no=ticket.train\_no WHERE  
 (train.source='KATAPADI' OR train.destination='KATAPADI' OR ticket.to\_station='KATAPADI' OR  
 ticket.from\_station='KATAPADI');

```
SQL> SELECT name FROM tttrain INNER JOIN tticket ON tttrain.train_number=tticke
t.train_number WHERE (tttrain.source='KATAPADI' OR tttrain.destination='KATAPADI
' OR tticket.to_station='KATAPADI' OR tticket.from_station='KATAPADI');

no rows selected
```

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

SELECT train.name FROM train INNER JOIN train\_ticket\_fare ON  
 train.train\_number=train\_ticket\_fare.train\_no WHERE (train.type='SUPERFAST' AND train\_ticket\_fare.service\_charge=0);

```
SQL> SELECT tttrain.name FROM tttrain INNER JOIN train_ticket_fare ON tttrain.tr
ain_number=train_ticket_fare.train_number WHERE (tttrain.type='SUPERFAST' AND tr
ain_ticket_fare.service_charge=0);

no rows selected
```

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

Select name from passenger where PNR\_NO in ( select PNR\_NO from ticket where train\_no in (select train\_no from train where  
 source='chennai')));

```
SQL> Select name from passenger where PNR_NO in ( select PNR_NO from tticke
t where train_number in (select train_number from tttrain where source='chennai'
));

no rows selected
```

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

SELECT train.name,train.class,ticket.total\_ticket\_fare FROM train INNER JOIN ticket ON train.train\_no=ticket.train\_no;

```
SQL> SELECT tttrain.name,tttrain.class,ttticket.total_ticket_fare FROM tttrain
INNER JOIN tticket ON tttrain.train_number=tticket.train_number;
```

NAME	CLASS	TOTAL_TICKET_FARE
TATA ALLEPPEY	1A	4568
GETANJALI	3A	2365

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

```
SELECT train.name,train.class,ticket.total_ticket_fare FROM train INNER JOIN ticket ON train.train_no=ticket.train_no;
```

```
SQL> Select name,source,destination from tttrain inner join tticket on tttrain.t
rain_number=tticket.train_number;
```

NAME	SOURCE	DESTINATION
TATA ALLEPPEY	katpadi	ALLEPPEY
GETANJALI	DELHI	katpadi

## 6. Create a sequence to provide values for the PNR no.

```
CREATE Sequence pnr_no start with 1 increment by 1 maxvalue 9 nocycle;
```

For inserting into pnr\_no:

```
INSERT into ticket value(pnr_no.nextval,...);
```

For viewing the sequence created:

```
SQL> CREATE Sequence pnr_no
2 start with 1
3 increment by 1
4 maxvalue 9
5 nocycle ;
```

Sequence created.

```
SQL> select * from user_sequences;
```

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	C	O	CACHE_SIZE
LAST_NUMBER						
PNR_NO	1	9	1	N	N	20

select \* from user\_sequences;

## 7. Write a query for full outer join using any of the tables above.

```
SELECT pnr_no FROM ticket FULL OUTER JOIN train ON ticket.train_no=train.train_no;
```

```
SQL> SELECT pnr_no FROM tticket
2 FULL OUTER JOIN tttrain ON tticket.train_number=tttrain.train_number;
```

PNR\_NO

14350  
76554  
78964  
56978  
45698