VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

Lab Cycle Sheet, MARCH 2020

B.Tech, Winter-2019-2020

NAME	PRIYAL BHARDWAJ
REG. NO.	18BIT0272
COURSE CODE	ITE1003
COURSE NAME	DATABASE MANAGEMENT SYSTEMS
SLOT	L-33+L-34
FACULTY	Prof. BIMAL KUMAR RAY

3. Create (Alter table to add constraint) the necessary foreign keys by identifying the relationships in the table. (Middle Level)

1. 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
  2  check (train_number between 10001 and 99999);

Table altered.
```

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

```
alter table train_route modify name unique not null;
```

```
SQL> alter table train_route modify name unique not null;

Table altered.
```

4. 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(ticket_class
in('1A','2A','3A','SL','C'));
```

```
SQL> alter table ticket add constraint tkt_chk
  2  check (ticket_class in('1A','2A','3A','SL','C'));

Table altered.

SQL> alter table train_ticket_fare add constraint trn_tkt_fr_chk
  2  check (ticket_class in ('1A','2A','3A','Sl','C'));

Table altered.
```

5. Add a not null constraint for the column distance in train_route.

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

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

Table altered.
```

4. Use SQL PLUS functions to. **(Low Level)**

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

```
select name from passenger natural join ticket where to_char
(date_of_journey,'dd-mm-yy') = to_char
(add_months(sysdate,1),'dd-mm-yy');
```

```
SQL> select name from passenger natural join ticket where
  2  to_char(date_of_journey,'dd-mm-yy')=to_char(add_months(sysdate,1),'dd-mm-yy');

no rows selected
```

2. Print the train names in upper case.

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

```
SQL> select upper(name) "Train Name" from train;

Train Name
------------------------
CHENNAI EXPRESS
AHMEDABAD EXPRESS
MUMBAI EXPRESS
MUMBAI MAIL
PUNE EXPRESS
PUNJAB MAIL

6 rows selected.
```

3. Print the passenger names with left padding character.

```
select lpad (name,20,'*') from passenger;
```

```
SQL> select lpad (name,20,'*') from passenger;

LPAD(NAME,20,'*')
--------------------------------------------------
**************Rohan
**************Nitin
***********Hritisha
************Anushka
**************Nishi
************Salonee
************Priyal
***********Kushagra

8 rows selected.
```

4. Print the station codes replacing K with M.

```
select translate(station_code,’K’,’M’) “S Code” from
train_route;
```

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

S Code
----------------
EM
BNC
MPD
BBC
BBC
BBC
MNC

7 rows selected.
```

5. Translate all the LC in class column (Train_fare) to POT and display.

```
select replace (ticket_class,’LC’,’POT’) “classes” from
train_ticket_fare;
```

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

classes
-----------------
Sl
3A
2A
2A
3A
2A

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)
-------------------
               1800
                280
               1520
               1280
                820
                920

6 rows selected.
```

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

```
select pnr_no, nullif (transactionid,0) from ticket;
```

```
SQL> select pnr_no, nullif (transactionid,0) from ticket;

    PNR_NO NULLIF(TRANSACTIONID,0)
---------- -----------------------
1928091842              5468529634
5365986546              7845632159
1895732123              9517534569
8674920651              7896354865
1864998349              5856696892
3615847290              2553809641
4635718634              5468529634
1234567809

8 rows selected.
```

8. Print the date_of_jounrney in the format '27th November 2010'.

```
select to_char(date_of_journey,'ddth Month yyyy') " Date of
Journey" from ticket;
```

```
SQL> select to_char(date_of_journey,'ddth Month yyyy') "Date of Journey" from ticket;

Date of Journey
-------------------------------------------
22nd October   2020
24th November  2020
21st October   2020
12th November  2020
14th June      2020
30th January   2020
22nd October   2020
22nd January   2020

8 rows selected.
```

9. Find the maximum fare (total fare).

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

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

MAX(TICKET_FARE)
----------------
            2100
```

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

```
select pnr_no, avg(age) from passenger group by pnr_no;
```

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

    PNR_NO   AVG(AGE)
---------- ----------
5365986546       23.5
8674920651       29.5
1928091842         68
1895732123         19
3615847290         83
1864998349         17

6 rows selected.
```

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

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

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

MAX(LENGTH(NAME))
-----------------
               15
```

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

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

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

ROUND(TICKET_FARE)
------------------
              2100
               400
              1800
              1500
              1050
              1050
              2100
              2100

8 rows selected.
```

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 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;
```

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

7 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;
```

```
SQL> select to_char (arrival_time,'yyyy/mm/dd') from train_route;

TO_CHAR(AR
----------
2020/01/01
2020/01/01
2020/01/01
2020/01/01
2020/01/01
2020/01/01
2020/01/01

7 rows selected.
```

```
select to_char (depart_time, ‘yyyy/mm/dd’) from train_route;
```

```
SQL> select to_char (depart_time,'yyyy/mm/dd') from train_route;

TO_CHAR(DE
----------
2020/01/01
2020/01/01
2020/01/01
2020/01/01
2020/01/01
2020/01/01
2020/01/01

7 rows selected.
```

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;
```

```
SQL> select to_char (arrival_time,'HH24:MI') from train_route;

TO_CH
-----
06:35
20:15
15:35
22:40
13:50
05:15
20:15

7 rows selected.
```

```
select to_char (depart_time, ‘HH24:MI’) from train_route;
```

```
SQL> select to_char (depart_time, 'HH24:MI') from train_route;

TO_CH
-----
06:45
20:22
15:40
22:50
14:00
05:30
20:22

7 rows selected.
```

5. Write Queries to. **(Middle Level)**

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 train
natural join ticket;
```

```
SQL> select train_number from train minus select train_number from train natural join ticket;

TRAIN_NUMBER
------------
       20127
```

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

```
select name from train minus select name from train,
table(train.class) where column value like ‘%1A’;
```

```
SQL> select name from train minus select name from train, table(t_class) where column_value like '%1A';

NAME
------------------------
Mumbai Express
Mumbai Mail
```

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

```
select pnr_no from ticket;
```

```
SQL> select pnr_no from ticket;

    PNR_NO
----------
1234567809
1864998349
1895732123
1928091842
3615847290
4635718634
5365986546
8674920651

8 rows selected.
```

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

```
select name from passenger natural join ticket where
to_station = ‘Pune’;
```

```
SQL> select name from passenger natural join ticket where to_station = 'Pune';

no rows selected
```

Use Nested Query (in Operators)

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

```
select distinct name from train where train_number in(select
train_number from train_route where name= ‘Katpadi’);
```

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

NAME
------------------------
Ahmedabad Express
```

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

```
select distinct name from train where type= ‘superfast’ and
train_number in(select train_number from train_ticket_fare
where service_tax=0);
```

```
SQL> select distinct name from train where type='superfast'
  2  and train_number in(select train_number from train_ticket_fare where service_tax=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_number in(select train_number from train
where source = ‘Chennai’));
```

```
SQL> select name from passenger where pnr_no in(
  2  select pnr_no from ticket where train_number in(
  3  select train_number from train where source='Chennai'));

no rows selected
```

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, table(t_class) where column_value in
(‘1A’, ‘2A’, ‘3A’) and column_value not in ‘SL’ and
train_number in(select train_number from train_ticket_fare
where basefare<3000);
```

```
SQL> select name from train, table(t_class) where
  2  column_value in('1A','2A','3A') and column_value not in 'SL'
  3  and train_number in(select train_number from train_ticket_fare where base_fare<3000);

NAME
-------------------------
Chennai Express
Chennai Express
Chennai Express
Mumbai Express
Mumbai Express
Ahmedabad Express
Ahmedabad Express
Ahmedabad Express
Pune Express
Pune Express
Pune Express

NAME
-------------------------
Mumbai Mail
Mumbai Mail
Punjab Mail
Punjab Mail
Punjab Mail

16 rows selected.
```

Use Join Query

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

```
select name from train inner join ticket on
train.train_number=ticket.train_number where(train.source=
‘Katpadi’ or train.destination= ‘Katpadi’ or ticket.to_station
= ‘Katpadi’ or ticket.from_station= ‘Katpadi’);
```

```
SQL> select name from train inner join ticket on train.train_number=ticket.train_number
  2  where(train.source='Katpadi' or train.destination='Katpadi' or
  3  ticket.to_station='Katpadi' or ticket.from_station='Katpadi');

NAME
------------------------
Mumbai Express
Mumbai Mail
```

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_tax=0);
```

```
SQL> select train.name from train inner join train_ticket_fare
  2  on train.train_number=train_ticket_fare.train_number where(
  3  train.type='superfast' and train_ticket_fare.service_tax=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_number in (select train_number from train
where source=’Chennai’));
```

```
SQL> select name from passenger where pnr_no in(
  2  select pnr_no from ticket where train_number in(
  3  select train_number from train where source='Chennai'));

no rows selected
```

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

```
select train.name , t_class , ticket.ticket_fare from train
inner join ticket on train.train_number = ticket.train_number;
```

```
SQL> select train.name,t_class,ticket.ticket_fare from train
  2  inner join ticket on train.train_number=ticket.train_number;

NAME            T_CLASS                                      TICKET_FARE
--------------- -------------------------------------------- -----------
Chennai Express  CLASS('1A', '2A', '3A', 'SL', NULL, NULL)          2100
Mumbai Express   CLASS('2A', '3A', 'SL', 'Gen', NULL, NULL)          400
Mumbai Express   CLASS('2A', '3A', 'SL', 'Gen', NULL, NULL)         1800
Pune Express     CLASS('1A', '2A', '3A', 'Sl', 'gen', NULL)         1500
Mumbai Mail      CLASS('2A', '3A', 'SL', NULL, NULL, NULL)         1050
Punjab Mail      CLASS('1A', '2A', '3A', 'SL')                     1050
Chennai Express  CLASS('1A', '2A', '3A', 'SL', NULL, NULL)         2100
Chennai Express  CLASS('1A', '2A', '3A', 'SL', NULL, NULL)         2100
```

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

```
select name,source,destination from train inner join ticket on
train.train_number=ticket.train_number;
```

```
SQL> select name,source,destination from train inner join
  2  ticket on train.train_number=ticket.train_number;

NAME            SOURCE              DESTINATION
--------------- ------------------- -------------------
Chennai Express  Chennai Egmore      Dadar
Mumbai Express   Chennai Central    Mumbai Cst
Mumbai Express   Chennai Central    Mumbai Cst
Pune Express     Delhi               Pune
Mumbai Mail      Chennai Central    Mumbai Cst
Punjab Mail      Mumbai              New Delhi
Chennai Express  Chennai Egmore      Dadar
Chennai Express  Chennai Egmore      Dadar

8 rows selected.
```

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

```
create sequence pnr_no start with 1 increment by 1 maxvalue 9
nocycle ;

select * from user_sequences;
```

```
SQL> create sequence pnr_no start with 1 increment by 1
  2  maxvalue 9 nocycle;

Sequence created.

SQL> select * from user_sequences;

SEQUENCE_NAME                  MIN_VALUE  MAX_VALUE INCREMENT_BY C O CACHE_SIZE LAST_NUMBER
------------------------------ ---------- ---------- ------------ - - ---------- -----------
LOGMNR_EVOLVE_SEQ$                      1 1.0000E+28            1 N Y          0           1
LOGMNR_SEQ$                             1 1.0000E+28            1 N Y          0           1
LOGMNR_UIDS$                            1 1.0000E+28            1 N Y          0         100
MVIEW$_ADVSEQ_GENERIC                   1 4294967295            1 N N         50           1
MVIEW$_ADVSEQ_ID                        1 4294967295            1 N N         20           1
PNR_NO                                  1          9            1 N N         20           1
REPCAT$_EXCEPTIONS_S                    1 1.0000E+28            1 N N         20           1
REPCAT$_FLAVORS_S              -2.147E+09 2147483647            1 N N          0           1
REPCAT$_FLAVOR_NAME_S                   1 1.0000E+28            1 N N          0           1
REPCAT$_REFRESH_TEMPLATES_S             1 1.0000E+28            1 N N         20           1
REPCAT$_REPPROP_KEY                     1 1.0000E+28            1 N N         20           1
REPCAT$_RUNTIME_PARMS_S                 1 1.0000E+28            1 N N         20           1
REPCAT$_TEMPLATE_OBJECTS_S              1 1.0000E+28            1 N N         20           1
REPCAT$_TEMPLATE_PARMS_S                1 1.0000E+28            1 N N         20           1
REPCAT$_TEMPLATE_REFGROUPS_S            1 1.0000E+28            1 N N         20           1
REPCAT$_TEMPLATE_SITES_S                1 1.0000E+28            1 N N         20           1
REPCAT$_TEMP_OUTPUT_S                   1 1.0000E+28            1 N N         20           1
REPCAT$_USER_AUTHORIZATIONS_S           1 1.0000E+28            1 N N         20           1
REPCAT$_USER_PARM_VALUES_S              1 1.0000E+28            1 N N         20           1
REPCAT_LOG_SEQUENCE                     1 1.0000E+28            1 N N         20           1
TEMPLATE$_TARGETS_S                     1 1.0000E+28            1 N N         20           1

21 rows selected.
```

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_number = train.train_number;
```

```
SQL> select pnr_no from ticket full outer join train
  2  on ticket.train_number=train.train_number;

    PNR_NO
----------
1928091842
5365986546
1895732123
8674920651
1864998349
3615847290
4635718634
1234567809


9 rows selected.
```

6. Write Queries to. **(Middle Level)**

Use Correlated (and nested) Query

1. Find the train names for which ten tickets have been reserved.

```
select name from train where train_number in (select
train_number from train intersect select train_number from
ticket where pnr_no in (select pnr_no from ticket group by
pnr_no having count(*)>10));
```

```
SQL> select name from train where train_number in(
  2  select train_number from train intersect
  3  select train_number from ticket where pnr_no in(
  4  select pnr_no from ticket group by pnr_no having count(*)>10));

no rows selected
```

2. Find the trains that have more than ten substations.

```
select train_number from train_route where station_code in
(select station_code from train_route group by station_code
having count(*)>10);
```

```
SQL> select train_number from train_route where station_code in(
  2  select station_code from train_route group by station_code having count(*)>10);

no rows selected
```

3. Find the passengers who do not pass through 'Mettupalam'.

```
select name from passenger minus select name from passenger
where pnr_no in (select pnr_no from ticket where train_number
in (select train_number from train_route where name=
‘mettupalam’));
```

```
SQL> select name from passenger minus select name from passenger where pnr_no in(
  2  select pnr_no from ticket where train_number in( select train_number from
  3  train_route where name='Mettupalam'));

NAME
---------------
Anushka
Hritisha
Kushagra
Nishi
Nitin
Priyal
Rohan
Salonee

8 rows selected.
```

4. Find passengers who have booked for superfast trains.

```
select name from passenger where pnr_no in (select pnr_no from
passenger intersect select pnr_no from ticket where
train_number in (select train_number from ticket intersect
select train_number from train_ticket_fare where
superfast_charge is not null));
```

```
SQL> select name from passenger where pnr_no in(
  2  select pnr_no from passenger intersect select pnr_no from
  3  ticket where train_number in(
  4  select train_number from ticket intersect select train_number
  5  from train_ticket_fare where superfast_charge is not null));

NAME
---------------
Rohan
Nitin
Hritisha
Anushka
Nishi
Salonee
Priyal
Kushagra

8 rows selected.
```

Complex queries(use groupby/groupby having/join/nested)

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

```
select name,source,destination from train,ticket where
train.train_number = ticket.train_number and
ticket.from_station= ‘TATA’ and ticket.to_station='KPD';
```

```
SQL> select name,source,destination from train,ticket where
  2  train.train_number=ticket.train_number and
  3  ticket.from_station='TATA' and ticket.to_station='KPD';

no rows selected
```

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

```
select train_number, count(station_code) from train_route
group by train_number;
```

```
SQL> select train_number,count(station_code) from train_route group by train_number;

TRAIN_NUMBER COUNT(STATION_CODE)
------------ --------------------
       11028                    1
       11042                    2
       12138                    1
       12164                    1
       20127                    1
       22554                    1

6 rows selected.
```

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

```
select station_code from train_route where train_number in
(select train_number from train);
```

```
SQL> select station_code from train_route where train_number in (select train_number from train);

STAT
----
EM
BNC
KPD
BBC
BBC
BBC
MNC

7 rows selected.
```

4. List the train names that have at least four bookings.

```
select name from train where train_number in (select
train_number from train intersect select train_number from
ticket where pnr_no in (select pnr_no from ticket group by
pnr_no having count(*)>4));
```

```
SQL> select name from train where train_number in(
  2  select train_number from train intersect select train_number
  3  from ticket where pnr_no in( select pnr_no from ticket
  4  group by pnr_no having count(*)>4));

no rows selected
```

5. Create a table cancellation history (Insert values from ticket and passenger table).

```
create table cancellation_history(
cancel_id varchar(20) primary key,
cancel_date date,
pnr_no number(10),
constraint fk_tp foreign key(pnr_no)references ticket(pnr_no),
Train_number number(5),
```

```
constraint fk_tn foreign key(train_number) references
train(train_number));
```

```
SQL> create table cancellation_history(
  2  cancel_id varchar(20) primary key,
  3  cancel_date date,
  4  pnr_no number(10),
  5  constraint fk_tp foreign key(pnr_no) references ticket(pnr_no),
  6  train_number number(5),
  7  constraint fk_tn foreign key(train_number) references train(train_number));

Table created.
```

```
create Sequence cancel_id start with 1 increment by 1 maxvalue
9 nocycle ;
```

```
SQL> create sequence cancel_id start with 1 increment by 1 maxvalue 9 nocycle;

Sequence created.
```

```
insert into cancellation_history values(‘cancel_id.nextval’,
’10-Nov-2017’,’1234567809’,11028);
```

```
SQL> insert into cancellation_history values('cancel_id.nextval','10-Nov-2017','1234567809',11028);

1 row created.

SQL> select * from cancellation_history;

CANCEL_ID            CANCEL_DA     PNR_NO TRAIN_NUMBER
-------------------- --------- ---------- ------------
cancel_id.nextval    10-NOV-17 1234567809        11028
```

6. Create a table for all the train numbers and class available in train_ticket_fare with total seats.

```
Create table seat as select train_number ,t_class from
train_ticket_fare;
```

```
SQL> create table seat as select train_number,ticket_class from train_ticket_fare;

Table created.
```

7. Find the station name that has highest number of trains stopping at.

```
select to_station, count(*) as nor from ticket group by
to_station having count(*)=(select max(nor) from (select
to_station,count(*) as nor from ticket group by to_station));
```

```
SQL> select to_station,count(*) as nor from ticket group by
  2  to_station having count(*)=(select max(nor) from(
  3  select to_station,count(*) as nor from ticket group by to_station));

TO_STATION                  NOR
-------------------- ----------
Mumbai Central                3
```

******