DATA DEFINOITION LANGUAGE

1. Create a table **COURSE** with the following attributes and constraint as given in the schema.

create table COURSE(COURSEID number(4), CourseName varchar(20), Duration number(2), Fees number(7,2), primary key(COURSEID));

2. Create a table **STUDENT** with the following attributes and constraint as given in the schema.

create table STUDENT(Studid number(4) PRIMARY KEY, FirstName varchar(20), LastName varchar(20), Street varchar(20), City varchar(20), DOB date);

3. Refer the following schema and add a field – **Age** Number(2) to the **student** table.

alter table STUDENT add(Age number(2));

4. Refer the following schema and add a constraint CHK_FEES in course table to ensure that the fee is greater than zero.

alter table COURSE add constraint CHK_FEES check (Fees>0);

5. Refer the following schema and drop Registration table.

drop table Registration;

6. Create a table “Guest” with column specification mentioned below.

create table GUEST(guestID number primary key, name varchar(30), address varchar(30), country varchar(20), email varchar(30), phone varchar(15));

7. Create PointOfInterest and Town tables based on the structure and the constraints given below.

create table PointofInterest(pointID number primary key, describe varchar(30), opentime varchar(10), closetime varchar(10), townID number);

create table Town(townID number primary key, townname varchar(30), state varchar(30), longitude varchar(30), latitude varchar(30), summertemp number, wintertemp number, sealevel number);

8. Refer the below schema, and modify the given query appropriately in order to create the product table.

| COLUMN NAME | DATATYPE | SIZE | CONSTRAINT |
|---|---|---|---|
| prod_id | number | 4 | Primary key |
| prod_name | varchar2 | 25 | |
| Prod_expiry_date | date | | Not null |

```
create table product

( prod_id number(4) primary key,

  prod_name varchar2(25),

  prod_expiry_date date NOT NULL

);
```

9. Modify the given query appropriately in order to create the department table using the schema given below, considering the product table that you have already created.

| COLUMN NAME | DATATYPE | SIZE | CONSTRAINT |
|---|---|---|---|
| dept_id | number | 4 | Primary key |
| prod_id | number | 4 | Foreign key |
| dept_name | varchar2 | 25 | Unique |
| dept_head | varchar2 | 25 | Not null |

```
CREATE TABLE department

( dept_id number primary key,

  prod_id number(4),

  dept_name varchar(25) unique,

  dept_head varchar(25) NOT NULL,

  foreign key (prod_id) references product(prod_id)

);
```

DATA MANIPULATION LANGUAGE

10. Refer the following schema and Insert the following records into the student table.

[Hint : To insert the given date value use dd-Mon-YY format ]

| ID | FirstName | LastName | Street | City | Date of Birth |
|---|---|---|---|---|---|
| 3001 | Dileep | Kumar | JaiNagar | Bangalore | 10th March 1999 |
| 3002 | Anand | Kumar | Indiranagar | Bangalore | 19th January 1998 |
| 3003 | Bala | Krishnan | Annanagar | Chennai | 03rd January 1996 |
| 3004 | Gowri | Shankar | Gandhipuram | Coimbatore | 22nd December 1997 |
| 3005 | Priya | Menon | JPNagar | Cochin | 12th February 1994 |

insert into STUDENT values('3001', 'Dileep', 'Kumar', 'JaiNagar', 'Bangalore','10-Mar-99');

insert into STUDENT values('3002', 'Anand', 'Kumar', 'Indiranagar', 'Bangalore','19-Jan-98');

insert into STUDENT values('3003', 'Bala', 'Krishnan', 'Annanagar', 'Chennai','03-Jan-96');

insert into STUDENT values('3004', 'Gowri', 'Shankar', 'Gandhipuram', 'Coimbatore','22-Dec-97');

insert into STUDENT values('3005', 'Priya', 'Menon', 'JPNagar', 'Cochin','12-Feb-94');

11. Abdul Rahman moved to 'Andheri West' Street in Mumbai city, he wants to update his address. Help the admin by providing the required SQL statement to update Abdul's details.

update STUDENT set Street='Andheri West', City='Mumbai' where FirstName='Abdul';

12. Write a query to remove registration details of those joined on 25th May 2018.

DELETE FROM REGISTRATION WHERE DOJ='25-May-18';

13. Insert the following records into "Resort " table based on schema below.

INSERT INTO RESORT values(1001, 'TREE OF LIFE RESORT', 'KACHERAWALA KOOKAS', 'JAIPUR', 3, 3.8, 2);
INSERT INTO RESORT values(1002, 'PALETTE RESORTS', 'LAKE VIEW ROAD NALLVADU POST', 'PONDICHERRY', 2, 4.3, 2);
INSERT INTO RESORT values(1003, 'GRAND GRT', 'SIR THEAGARAYA RD TNAGAR', 'CHENNAI', 1, 4.4, 1);
INSERT INTO RESORT values(1004, 'GREEN COCONUT', 'EAST COAST ROAD, MAHABALIPURAM', 'CHENNAI', 1, 3.7, 1);

14. The rating for “TREE OF LIFE RESORT“ has changed. Write a query to change the value of the rating from 3.8 to 4.2 in the “Resort” table.

Update RESORT set starRating='4.2' where resortName='TREE OF LIFE RESORT';

15. ResortID 1002 is not maintaining good standard and the guest reviews are not up to our standard rating, so the management has decided to remove the resort from our resort management system. Write a SQL statement to remove the resort.

delete from RESORT where resortId='1002';

16. Write a query to display details about all courses and sort the result based on course id.

select * from COURSE order by COURSEID asc ;

17. Write a query to display the course name and the fees which has a duration of 2 to 4 months. Sort the result based on the course name.

select CourseName, Fees from COURSE where Duration between 2 and 4 order by CourseName;

18. Write a query to display city of all students. Display unique values and sort them in ascending order.

select distinct City from STUDENT order by City asc;

19. Write a query to display student's first name whose starting letter is 'S' and ending with 'n'.

Sort the result set by student's first name.

select FirstName from STUDENT where FirstName like 'S%n'order by FirstName;

20. Write a query to display student Id who registered for the course 1001, 1005. If they registered for both the courses then display the student id once. Sort the result by student Id.

select distinct Studid from REGISTRATION where CourseID='1001' or CourseID='1005' order by Studid;

21. Display all the resort name and town name where the resort is located. Sort the result set in the ascending order of resort name.

select resortName, townname from RESORT order by resortName;

22. Display all the resorts names whose star-rating ranges between 4 and 5. Sort the result set in the ascending order of star rating.

select resortName from RESORT where starRating between 4 and 5 order by starRating asc;

23. Resort manager wants to know the type of rooms available in the resorts. Write a query to display the resort id, number of bed rooms, and type of rooms available in resort ids 1001 and 1002. Display the result set in the ascending order of resort id, number of bed rooms and types of rooms available.

```
SELECT RESORT.resortID,bedroomcount,cabintype

from RESORT,CABIN

where(RESORT.resortID=1001 or RESORT.resortID=1002) and RESORT.resortID=CABIN.resortID

order by resortID,bedroomcount,cabintype asc;
```

24. The resort manager wants to know the guest who brought their children and pet to the resort. Write a query to display the guest id, the resort id, the number of days they stayed (give alias name as numberofdays), count of adults, children and pet, their total charge incurred. Display the output in the descending order of pet count.

```
select REVIEW.guestID, REVIEW.resortID, trunc(todate - fromdate) as numberofdays, adultCount, childCount, petcount, totalcharge from REVIEW, BOOKING where (REVIEW.guestID=BOOKING.guestID and REVIEW.resortID=BOOKING.resortID) and (BOOKING.childCount>0 and BOOKING.petcount>0) order by petcount desc;
```

25. Write a query to display to concatenate student's first name and last name, give alias name as STUDENTNAME. Sort result in ascending.

For example, student's first name is 'Rahul' and last name is 'Dravid' then output should be 'Rahul Dravid'

```
select FirstName || ' ' || LastName as STUDENTNAME from STUDENT order by STUDENTNAME asc;
```

26. Write a query to display unique student ID who joined in the month of June. Sort the result in ascending order.

```
select distinct(StudId) from REGISTRATION where extract(month from DOJ)=6 order by StudId asc;
```

27. Write a query to display student ID, First name and age. Sort the result based on student ID.

[Note : Age can be calculated using DOB and give alias name as AGE]

```
select StudId, FirstName, trunc((SYSDATE - DOB)/366,0) as AGE from STUDENT order by StudId;
```

28. Write a query to display student ID and number of course registered by students. Display student ID only if student has registered for a course. Give alias name for the count as NOOFCOURSES. Sort the result based on count in descending and student ID ascending.

select StudId, count(courseID) as NOOFCOURSES from REGISTRATION group by StudID order by count(courseID) desc, StudID;

29. Write a query to display the guest name in proper case. For instance the guest name "geeta govind" should be displayed as "Geeta Govind". Give alias name as GUESTNAME. Sort the result based on Guest Name.

select initcap(Name) AS GUESTNAME from GUEST order BY GUESTNAME;

30. The manager has to calculate the maximum duration of stay in any resort. Write a query to find maximum number of days stayed by the guest. Give alias name to the result as "MAXIMUMDAYS".

[Hint : number of days can be calculated using fromdate and todate of booking details]

select max(trunc(todate - fromdate)) as MAXIMUMDAYS from Booking;

31. Select the highest summer & lowest winter temperature and display the output as "HOT SUMMER" and "CHILL WINTER"

select max(summertemp) as "HOT SUMMER", min(wintertemp) as "CHILL WINTER" from Town;

32. Write a query to calculate and display the resort id , total count of bed rooms each resort . The result set should be in the ascending order of resort id . Give alias name to count of bed rooms as TOTALCOUNT.

select resortID, sum(bedroomcount) as TOTALCOUNT from CABIN group by resortID order by resortID asc;

33. Each manager is assigned with a few resorts. Management wants to know how many resorts (rated 4 and above) assigned to each manager. Write a query to display the manager id and number of resort assigned to each of them. Give alias name to number of resort as "NUMBEROFRESORT". Sort the result set in ascending order of manager id.

select managerID, count(resortID) as NUMBEROFRESORT from RESORT where starRating=4 or starRating>4 group by managerID order by managerID asc;

34. Guest wants to know the resort id , bed room count and sleeping capacity if the resort has total of 60 or more bed rooms and 100 or more sleeping capacity. Display the resort id total bed room count as TotalRoom and its total sleep capacity as Capacity. The output should be in the ascending order of resort id.

select resortID, sum(bedroomcount) as TotalRoom, sum(sleepcapacity) as Capacity from CABIN group by resortID having (sum(bedroomcount)>=60) and (sum(sleepcapacity)>=100) order by resortID asc;

35. Write a query to display Course ID and course name of the course which has second maximum fee. If there are more than two courses then sort it by using Course ID.

```
select courseid, CourseName from COURSE where Fees = (

   select max(Fees) from COURSE where Fees <

      (select max(Fees) from COURSE)

)

order by courseid;
```

36. Write a query to display the student's first name alone with the course name that they have registered. Sort the result based on student's first name and course name.

```
select FirstName,CourseName from ((STUDENT s

inner join REGISTRATION r on s.StudID=r.StudId)

inner join course c on r.CourseID=c.CourseID)

order by FirstName asc,CourseName asc;
```

37. Write a query to display the student's ID and the total fees paid by each student. Give alias name to total fees as TOTALFEES. Sort the result based on student ID.

```
select r.studid, sum(c.fees) as TOTALFEES

from course c join

    registration r

    on c.courseid = r.courseid

group by r.studid

order by r.studid;
```

38. Write a query to display the point id, description, town name and state where the pointofinterest is located. Sort the result in the ascending order of point id.

```
select p.pointID, p.describe,

                (select townname from town t where p.townid=t.townid ) as townname,

               (select state from town t where p.townid=t.townid) as state

from PointofInterest p

order by 1;
```

39. Managements wants to know the manager of each resort.

Display resortid , resort name, manager name and his phone. Sort the output in ascending order of resort id. Give alias to manager name as MANAGERNAME and phone as PHONENO.

If manager is not available then display manager name 'NA'. If phone number is not available then display as 'NA'.

```
select resort.resortID, resort.resortName,

nvl2(manager.name, manager.name, 'NA') as MANAGERNAME,

nvl2(manager.phone, manager.phone, 'NA') as PHONENO from resort

left join manager on resort.managerid=manager.managerid

order by resort.resortID;
```

40. Write a query to fetch the various pointofinterest locations in the town. Display townname,state ,description as "Tourist Spots", in the ascending order of townid.

```
select t.townname, t.state, p.describe as "Tourist Spots"

from town t join pointofinterest p

        on t.townid=p.townid

order by t.townid;
```

41. Managements wants to display city and manager details if more than one manager from same city. Write a query to display city , manager name and their phone number. Sort the result based on city and manager name.

```
select distinct m1.city, m1.name, m1.phone

from manager m1 join manager m2

    on m2.city=m1.city

    where m1.name!=m2.name

order by 1,2;
```

42. Write a query to display the revenue generated by each resort. Display the resortid, name of the resort ,booking count and the total amount collected in each resort. Sort the result by resort id. Give alias name as TOTALBOOKING to booking count and TOTALAMOUNT to total amount collected.

```
select b.resortid, (select resortname from resort rs where rs.resortid=b.resortid ) as
"RESORTNAME",

count (b.bookingid) as TOTALBOOKING, sum (b.totalcharge) as

TOTALAMOUNT

from booking b

group by b.resortid

order by 1;
```

43. Display guest details who paid total charges RS.50000 and above. Write a query to fetch Guest id, Guest name and Sum of total charges. Give alias name to total charges as  TOTALPAID. Sort the result by guest id.

```
select guest.guestid, guest.name, sum(booking.totalcharge) as

TOTALPAID from guest

join booking on booking.guestid=guest.guestid

group by guest.guestid,guest.name

having sum(booking.totalcharge)>=50000

order by guest.guestid;
```

44. Write to query to display the name and town name of the resort that has least bed room count. Sort the output on the resort name.

```
select rs.resortname, rs.townname

from resort rs left join cabin c

    on rs.resortid = c.resortid

    where c.bedroomcount = (select min(bedroomcount) from cabin)

    order by 1;
```

45. Write a query to display the pointId, description, open time and close time for all the pointofinterest in “CHENNAI” location. Output should be sorted on pointid.

```
select p.pointid, p.describe, p.opentime, p.closetime

from pointofinterest p join town t

                on t.townid=p.townid

                where t.townname='CHENNAI'

                order by 1;
```

46. Write a query to display resort id and review comments of all resort which has star rating above 4.2. sort the output in ascending order of resort id.

```
select r.resortid, rv.comments from Resort r join review rv

on r.resortid=rv.resortid where r.starRating > 4.2

order by 1;
```

47. Write a query to display the resort name , guest name , review comments for all resort. sort the output in ascending order of resort ID.

```
select r.resortname, g.name, rv.comments from resort r join review rv

on  r.resortid=rv.resortid join guest g

on g.guestid = rv.guestid

order by r.resortid;
```