---SELECT STATEMENTS---

1. Dream Home-Maximum cost of flat

```
SELECT  floor_no AS FLOOR_NO,MAX(cost_inlakh) AS MAX_PRICE FROM
t_flat_details
GROUP BY floor_no
ORDER BY floor_no DESC
```

2. Event Hall-Halls booked more than once

```
select a.hall_name, count(b.hall_id) as no_of_times_booked
from t_hall_details a join t_hall_booking b on a.hall_id = b.hall_id
group by a.hall_name
having length(a.hall_name)>5 and count(b.hall_id)>1
order by a.hall_name desc
;
```

3. Event Hall-Number of booking customer wise

```
select  distinct c.customer_id,c.customer_name,count(h.hall_id) as
NO_OF_BOOKING
from t_customer_details c
right join t_hall_booking h
on c.customer_id = h.customer_id
where h.event_date like '2020%'
group by c.customer_id
having c.customer_name like 'S%'
order by 2;
```

3) Event hall average cost of not booked halls sql

```
select CITY, round(avg(cost_perday),0) as AVERAGE_COST
from t_hall_details
where hall_id not in(select hall_id from t_hall_booking) and capacity>100
group by CITY
order by AVERAGE_COST asc;
```

4.Car pooling-Driver booking details based on name

```
select a.booking_no,b.user_name,c.driver_name,a.pickup_from,a.drop_at,d.distance
from booking a join customer b on a.customer_id=b.id
        join driver c on a.driver_id=c.id
        join city_locations d on ((a.pickup_from=d.city1 and
a.drop_at=d.city2)or(a.pickup_from=d.city2 and a.drop_at=d.city1))
        where upper(c.driver_name)='JOE AMAL'
        order by d.distance
```

5.Cricket-Number of players in each city

```
select player_city as PLAYER_CITY, count(player_id) as NUMBER_OF_PLAYERS from
t_player where player_city not in
(select distinct played_city from t_match_record)
group by player_city
order by NUMBER_OF_PLAYERS,PLAYER_CITY;
```

6.Hospital-Number of doctors based on shift

```
select h.shift_time as SHIFT_TIME,count(h.available_doctor) AS
NUMBER_OF_DOCTORS from t_hospital h
join t_doctor d on d.doctor_id=h.available_doctor
where specialization = 'SURGEON'
group by shift_time
having count(available_doctor)>=1
order by shift_time desc;
```

7.Insurance-List of Policies

```
select distinct p.policy_name, p.policy_type
from t_policy p, t_member m
where p.policy_id = m.policy_id
```

```
 and m.member_id >= '1'
order by  policy_name, policy_type asc;
```

8.Movie details based on Certification and Duration

```
select movie_id,movie_name,director_name,language from movie_master where
certification='U'
    and duration>130
    order by movie_id;
```

9. Patient Appointment Details based on reason

```
select p.patient_id,p_first_name,p_age,app_number,app_date
from appointment a join patient p on a.patient_id=p.patient_id
where app_reason='FEVER' order by 1;
```

10. Student-Room Details

```
select s.student_id,student_name,department,DOJ,r.room_id,
room_type from student_details s join admission_details a
on s.student_id=a.student_id join room_details r
on r.room_id=a.room_id order by 1;
```

----FUNCTIONS AND SUBQUERIES---

11.Car Pooling-Vehicle details

```
select c.vehicle_model,c.vehicle_type,sum(ci.distance)from car c
join booking b on b.vehicle_no=c.vehicle_no
join city_locations ci on (ci.city1=b.pickup_from and
ci.city2=b.drop_at)or(ci.city2=b.pickup_from and ci.city1=b.drop_at)
group by c.vehicle_type,c.vehicle_model
order by sum(ci.distance);
```

11. Cricket-Average runs of players based on name

```
select m.player_id, round(avg(m.player_runs)) as average_runs from
t_match_score_card m
join t_player p
on p.player_id=m.player_id where player_name like 'S%'
group by m.player_id
order by average_runs desc;
```

12. Dream Home-Customer name details based on total cost

```
SELECT c.customer_name,SUM(f.cost_inlakh) FROM t_flat_booking b
JOIN t_flat_details f ON b.flat_no=f.flat_no
JOIN t_customer_details C ON c.customer_id=b.customer_id
WHERE LENGTH(c.customer_name)>'10'
GROUP BY c.customer_name
ORDER BY customer_name
```

13. Event Hall-Average cost of booked halls

```
select CITY,round(avg(cost_perday),0) as AVERAGE_COST from t_hall_details
where hall_id in(Select hall_id from t_hall_booking) and capacity>150
group by CITY
order by average_cost;
```

14. Hospital-Total fees received based on gender and shift

```
SELECT t_patient.gender, sum(t_doctor.fees) FEES_RECEIVED
FROM t_patient
JOIN t_doctor on t_doctor.doctor_id=t_patient.doctor_id
JOIN t_hospital on t_hospital.available_doctor=t_doctor.doctor_id
WHERE upper(t_hospital.shift_time)="MORNING"
GROUP BY t_patient.gender
ORDER BY t_patient.gender DESC;
```

15. Insurance-List of Agents

```
select a.agent_id,p.policy_name,sum(p.policy_sum)as policy_sum from t_agent a
join t_member m on m.agent_id=a.agent_id
join t_policy p on p.policy_id=m.policy_id
group by a.agent_id,p.policy_name
having count(m.member_id) >=1
order by a.agent_id,p.policy_name,policy_sum;
```

16 .Minimum _ Maximum Discount Amount

```
SELECT MIN(DISCOUNT_AMOUNT) AS MIN_DISCOUNT,
MAX(DISCOUNT_AMOUNT) AS MAX_DISCOUNT
FROM DISCOUNT_MASTER;
```

17 **.Number of Appointments**

```
SELECT doctor_id, COUNT(app_number) as APPOINTMENT_COUNT
FROM appointment
GROUP BY doctor_id
ORDER BY doctor_id;
```

18. Student Details In Capital Case

```
select student_id,upper(student_name) as NAME,department,phone_no
from student_details
where address='BANGALORE'
order by student_id;
```

19. Car Pooling-Maximum time driven driver details

```
select b.driver_id, d.driver_name, count(driver_id) as MaxTimesDriven
from driver d
inner join booking b on d.id=b.driver_id
group by b.driver_id
having count(driver_id)>2
```

```
order by b.driver_id;
```

20. Cricket-Player details

```
SELECT DISTINCT p.PLAYER_ID,p.PLAYER_NAME,p.PLAYER_CITY from t_player p
join t_match_score_card s on p.player_id=s.player_id
join t_match_record r on r.match_id=s.match_id
WHERE s.waysof_dismissal='STUMPED' AND r.played_city='BANGALORE'
order by player_name desc;
```

21. Dream Home -Flat details based on year

```
select a.flat_no FLAT_NO, b.size SIZE, b.area area
from t_flat_booking a
join t_flat_details b
on a.flat_no = b.flat_no
where year(a.registration_date)
in (select year(b.registration_date)
from t_customer_details a
join t_flat_booking b
on a.customer_id=b.customer_id
where upper(a.customer_name='Niraj Kumar'))
order by area asc,a.flat_no desc;
```

22. Hospital-Maximum fees paid patient details

```
select p.patient_name , d.doctor_name , d.fees as 'fees_paid' , h.shift_time as
'checkup_done'
from t_patient p
join  t_doctor d on p.doctor_id = d.doctor_id
join t_hospital h on h.available_doctor = d.doctor_id
where d.fees>( select max(fees) from t_doctor where specialization = 'DERMA')
order by d.doctor_name , p.patient_name;
```

23.Insurance-Agent details

```
select count(b.member_id) as NUMBER_OF_MEMBERS,a.agent_name as
AGENT_NAME
from t_agent a join t_member b
on a.agent_id=b.agent_id
where a.agent_name like 'S%' or a.agent_name like 's%'
group by a.agent_name
order by AGENT_NAME,NUMBER_OF_MEMBERS asc;
```

23. Concatenating Details

```
SELECT CONCAT(MOVIE_NAME," is a ",LANGUAGE," Movie") AS MOVIE_DETAILS
FROM MOVIE_MASTER
ORDER BY MOVIE_DETAILS DESC;
```

24. Patient Appointment details Based On Month

```
SELECT
DISTINCT(PATIENT_ID),P_FIRST_NAME,P_AGE,ADDRESS,CONTACT_NUMBER
FROM PATIENT
WHERE PATIENT_ID IN(SELECT PATIENT_ID FROM APPOINTMENT WHERE
APP_DATE BETWEEN '2019-06-01' AND '2019-06-31')
ORDER BY PATIENT_ID;
```

25. Room Details Based On Location

```
select
ROOM_DETAILS.ROOM_ID,ROOM_DETAILS.ROOM_TYPE,ROOM_DETAILS.MEMBER_CAPACITY,R
OOM_DETAILS.ROOM_RENT
from ROOM_DETAILS
   inner join HOSTEL_DETAILS
      on ROOM_DETAILS.HOSTEL_ID=HOSTEL_DETAILS.HOSTEL_ID
where HOSTEL_DETAILS.LOCATION = 'PHASE-A'
    order by ROOM_DETAILS.ROOM_ID;
```

---

26 .Update t_flat_details

```
Set cost_inlakh = cost_inlakh +(cost_inlakh 0.01)
 Where size=”1BHK” and floor_no =3;

Update t_flat_details Set cost_inlakh = cost_inlakh +(cost_inlakh 0.02)
Where size=”2BHK” and floor_no =3;


Or

Update
       t_flat_details
Set
       Cost_in_lakh=case
       When size=”1BHK” THEN cost_in_lakh+(cost_in_lakh*0.01)
       ELSE cost_in_lakh+(cost_in_lakh *0.02)
Where
      Floor_no=3 and size in(“1BHK”,”2BHK”);
```

27 .Flat details based on year

```
Select a.flat_no FLAT_NO,b.size SIZE,b.area AREA
From t_flat_booking a
Join t_flat_details b
On a.flat_no=b.flat_no
Where
Datepart(year,a.registeratio n_date) in

(Select Datepart(year,b.registeratio n_date)
Or
WHERE EXTRACT (YEAR FROM a.registration_date)
 IN (SELECT
EXTRACT(YEAR FROM b.registration date)
From t_customer_details a Join t_flat booking b. On
a.customer id=b.customer
Where
Upper(a.customer_name)=’ NIRAJ KUMAR’)
Order by b.area asc,
a.   flat_no desc
```

28. Total cost

```
Select a.customer_name CUSTOMER_NAME,Sum(c.cost_inlakh)
TOTAL_COST
From
T customer_details a
Join T_flat booking b
a.customer_id=b.customer_id
Join T_flat_details c
c.flat_no=b.flat_no Where
length(a.customer_name)>10
Group by
a.customer_name
```

```
Order by
a.customer_name

Or
Select a.customer_name as CUSTOMER_NAME,sum(c.cost_inlakh)
 as TOTAL_COST from t_customer_detials a
join
T_flat_booking b
 on a.customer_id=b.customer_id
join
 t_flat_details c
 on
 c.flat_no-b.flat_no
  Where
Length(a.customer_name)>10
 group by a.customer_name
order by a.customer_name
```

29 .Max cost of flat

```
Select FLOOR_NO,max(COST_INLAKH) AS MAX_PRICE FROM t_flat_details
Group by FLOOR_NO
Order by FLOOR_NO desc;
```

30 . alter table

```
Alter table[tablename] Add [column name][data type]
                        Drop Column[column_name]
                       Alter column[column_name] data type
```

-----PIZZA SQL-----

31)PIZZA STORE-ALTER TABLE FOREIGN KEY

```
ALTER TABLE PIZZA ADD FOREIGN KEY (cust_id) REFERENCES CUSTOMER(cust_id),
ADD FOREIGN KEY (partner_id) REFERENCES DELIVERY_PARTNER(partner_id);
```

32)UPDATE PIZZA TABLE DISCOUNT

```
UPDATE pizza set amount  = (amount * 95)/100
Where pizza_type = " Extra Large" ;
```

OR

```
UPDATE pizza set amount  = 0.96*amount
Where pizza_type = " Extra Large" ;
```

33) PIZZA STORE-ALTER TABLE PIZZA 1.1

```
alter  table pizza add constraint fk_cid foreign key(cust_id) references customer(cust_id)

alter  table pizza add constraint fk_pid foreign key(partner_id) references
delivery_partner(partner_id)
```

34)Total cost OF PIZZA ORDERED -FUNCTION ANS SUBQUERY

```
select cust_id, pizza_name, count(*) as 'Times taken', sum(amount) as 'Total cost'
from pizza
where amount > 1200 group by pizza_name , cust_id order by 1;
```

35)PIZZA-Delivery partner details-RDBMS SELECT

```
Select partner_id,cust_id,count(cust_id) as times_required from pizza  group by
partner_id, cust_id having count(cust_id)>1
Order by partner_id
```

36) PIZZA FRAMING Password-SCALAR & AGGREGATE

```
SELECT CONCAT(cust_name,cust_id) AS USERNAME,
CONCAT(SUBSTRING(cust_name, 1, 3), SUBSTR(cust_phone, -4)) AS PASSWORD
FROM customer
ORDER BY USERNAME;
```

37)Extra large pizza

```
Select c.cust_id, c.cust_name, p.pizza_name, count(p.pizza_id) as "# times",
sum(p.amount) as total_Amount
From pizza p, customer c Where p.cust_id = c.cust_id
And  lower(p.pizza_type) like 'extra%'
Group by c.cust_name, c.cust_id, p.pizza_name
having sum(p.amount) > 4*(select min(amount) from pizza)
Order by c.cust_id desc;
```

38)pizza low cost and High cost pizza

```
SELECT distinct pizza_name,pizza_type, amount from pizza
where amount IN(SELECT MAX(amount) from pizza) or
amount IN(SELECT MIN(amount) from pizza)
limit 2;
```

OR

```
select distinct pizza_name, pizza_type, amount from pizza where amount = (select
max(amount) from pizza) and amount = (select min(amount) from pizza)
```

39)Pizza highest selling pizza

```
select  pizza_name, count(amount) as No.sold
from pizza
where pizza.pizza_id= pizza_id
group by pizza.pizza_name
order by count(amount) desc
limit 1;
```

OR

```
select  pizza_name, count(amount) as Highest_selling
from pizza
where pizza.pizza_id= pizza_id
group by pizza.pizza_name
order by count(amount) desc
limit 1;
```

40)pizza highest business date

```
SELECT order_date , SUM(amount) as "Highest Business" FROM pizza
GROUP BY order_date
 ORDER BY SUM(amount) DESC
 limit 1;
```

41) Pizza highest business customer details

```
Select distinct customer.cust_id, customer.cust_name, sum(pizza.amount) as Max_Amount
From customer join pizza
On customer.cust_id=pizza.cust_id
Group by pizza.cust_id
Order by sum(pizza.amount) desc
Limit 1;
```

42)Password generation pizza

```
select concat(cust_name,cust_id) as USERNAME,
concat(left(cust_name,3),right(cust_phone,4)) as PASSWORD
from customer
order by 1;
```

----------QUESTION NOT THERE----------

```
select shift_time,count(available_doctor) as NUMBER_OF_DOCTORS
from t_hospital join t_doctor on available_doctor=doctor_id
where available_doctor in
(select doctor_id from t_doctor
where specialization='SURGEON')
group by shift_time
order by shift_time desc;
```

----

43)EVENT HALL NUMBER OF BOOKING CUSTOMER WISE

```
select b.customer_id as CUSTOMER_ID , c.customer_name as CUSTOMER_NAME,
count(b.hall_id) as NO_OF_BOOKING
FROM t_customer_details c
join t_hall_booking b on b.customer_id  = c.customer_id
where c.customer_name like 'S%' and b.event_date between '2020-01-01' and
'2020-12-31'
group by b.customer_id
order by c.customer_name;
```

-----DDL SQL-----

44)Car Pooling - Update booking table1.2

```
update booking
set fare=(select min(distance)*11 from city_locations ct
join booking b On b.pickup_from=ct.city1 AND b.drop_at=ct.city2);
```

45) Car Pooling- Create BOOKING table 1.1

```
create table booking (
booking_no varchar(50),
pickup_from varchar(50),
drop_at varchar(50),
customer_id varchar(50),
vehicle_no varchar(50),
driver_id varchar(50),
fare decimal(7,2),
primary key (booking_no),
foreign key (customer_id) references customer(id),
foreign key (vehicle_no) references car(vehicle_no),
foreign key (driver_id) references driver(id)
);
```

46)Create Movie_Master table set1

```
create table Movie_Master(
MOVIE_ID varchar(5) primary key,
MOVIE_NAME varchar(4) not null,
DIRECTOR_NAME varchar(4) not null,
CERTIFICATION varchar(4) not null,
DURATION INT(3),
LANGUAGE varchar(10)
);
```

47)Cricket -Alter T_MATCH_SCORE_CARD table(1.1)

```
alter table t_match_score_card add foreign key (match_id) references
t_match_record (match_id);
alter table t_match_score_card add foreign key (player_id) references
t_player(player_id);
```

48)Cricket-Update T_PLAYER table(1.2)

```
update t_player
set total_wickets=case
when(player_city='BANGALORE' and player_name like 'A%')
THEN total_wickets+5
when(player_city='DELHI' and player_name like 'A%')
THEN total_wickets+7
ELSE total_wickets
END;
```

49)Dream Home- Alter table t_flat_booking1.1

```
alter table t_flat_booking modify payment_completed varchar(5) not null;
```

50)Event Hall- Alter table Hall Booking 1.1

```
alter table t_hall_booking
modify hall_id varchar(10) not null;
alter table t_hall_booking
add foreign key(hall_id) references t_hall_details(hall_id);
```

51)Event Hall-Alter T_HALL_BOOKING table1.1

```
alter table t_hall_booking
modify hall_id varchar(10) not null;
alter table t_hall_booking
add foreign key(hall_id) references t_hall_details(hall_id);
alter table t_hall_booking
modify customer_id varchar(10) not null;
alter table t_hall_booking
add foreign key(customer_id) references t_customer_details(customer_id);
```

52)Event Hall Customer details with booking

```
select  distinct c.customer_id,c.customer_name,count(h.hall_id) as
NO_OF_BOOKING
from t_customer_details c
right join t_hall_booking h
on c.customer_id = h.customer_id
where h.event_date like '2020%'
group by c.customer_id
having c.customer_name like 'S%'
order by 2;
```

OR

```
SELECT customer_id
 ,customer_name
 ,mobile_no
FROM t_customer_details
WHERE length(customer_name) > 10
 AND customer_id IN (
  SELECT customer_id
  FROM (
   SELECT customer_id
    ,count(hall_id)
   FROM t_hall_booking
   GROUP BY customer_id
   HAVING count(hall_id) > (
     SELECT count(h.hall_id)
     FROM t_hall_booking h
     INNER JOIN t_customer_details c ON c.customer_id = h.customer_id
     WHERE c.customer_name = 'Suman Singh'
     GROUP BY h.customer_id )) AS T1)
ORDER BY customer_name;
```

53)**Hospital- Add a new column set1**

```
alter table doctor add column dr_contact_number int(10);
```

54)**Hospital- Alter T_HOSPITAL table 1.1**

```
alter table t_hospital
add foreign key (available_doctor) references t_doctor(doctor_id);
```

55)Hospital- Change the datatype_column

```
alter table patient modify
contact_number int(10);
alter table patient change p_age patient_age int;
```

56)Hospital-Update T_DOCTOR table 1.2

```
update t_doctor set fees=350
where specialization="ENT" and doctor_name like "J%";
update t_doctor set fees=600
where specialization="DERMA" and doctor_name like "J%";
update t_doctor set fees=null
where specialization="SURGEON" and doctor_name like "J%";
update t_doctor set fees=null
where specialization="ORTHO" and doctor_name like "J%";
```

57)Hostel-Insert Student Records

```
insert into Student_details values
('S1001','Varsha','ECE','1999-06-12','CHENNAI',9845712345,'varsha123@gmail.com'),
('S1002','William','ECE','1999-02-04','CALCUTTA',6845712345,'william123@gmail.com'),
('S1003','Basha','EEE','1999-06-14','DELHI',9945712345,'basha222@gmail.com'),
('S1004','Catherine','CSE','1998-08-16','DELHI',6785712345,'cathu123@gmail.com'),
('S1005','Kate','ECE','1999-06-30','BANGALORE',7685712345,'katedd@gmail.com'),
('S1006','Michel','ECE','1998-06-04','COIMBATORE',6645712345,'michel000@gmail.com');
```

58) Hostel-Update Student Record

```
UPDATE STUDENT_DETAILS
SET EMAIL_ID='mic.hudson@gmail.com'
WHERE STUDENT_ID='S1006';
```

59)Insurance- Alter table-add constraint(1.1)

```
alter table T_MEMBER
ADD foreign key(AGENT_ID) references T_AGENT(agent_id),
ADD foreign key(POLICY_ID) references T_POLICY(policy_id);
```

60)Insurance-Update Agent details(1.2)

```
Update t_agent
set target_policy_sum=case
when upper(agent_city)='PUNE' and upper(agent_id) like 'M%'
then 400000
when upper(agent_city)='CHENNAI' and upper(agent_id) like 'M%'
then 250000
else target_policy_sum
end;
```

61)Movie - Modify the datatype

```
ALTER TABLE CUSTOMER_MASTER MODIFY COLUMN PHONE_NO INT(10);
```
