////////////////////////////.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;
```

//////////////////////////////Event Hall-Average cost of not booked halls

```
SELECT

  City, ROUND(AVG(cost_perday))   ‘Average_Cost’

FROM

  T_hall_details

WHERE

  Hall_id NOT IN (SELECT

       Hall_id

     FROM

       T_hall_booking)

     AND capacity > 100

GROUP BY city;
```

/////////////////////////////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
```

```
////////////////////////////////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;

//////////////////////////////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;

/////////////////////////////////Minimum & Maximum Discount Amount

SELECT MIN(DISCOUNT_AMOUNT) AS MIN_DISCOUNT,

MAX(DISCOUNT_AMOUNT) AS MAX_DISCOUNT

FROM DISCOUNT_MASTER;

/////////////////////////////////Number of Appointments..

SELECT doctor_id, COUNT(app_number) as APPOINTMENT_COUNT

FROM appointment

GROUP BY doctor_id

ORDER BY doctor_id;

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;
```

```
//////////////////////////////Pizza-Delivery Partner Details With Rating..
select partner_id,partner_name,concat(partner_id,substr(partner_name,1,4))
Name,if(rating>=9,'Excellent',if(rating>=7 and rating<9,'Good','Worst')) as FEEDBACK from
delivery_partner order by partner_id;

///////////////////////////////Pizza-Framing Customer password ..

Select concat (cust_name,cust_id)

As USERNAME

Concat (substring(cust_name,1,3),

Substring(cust_phone,-4,4) as PASSWORD

FROM customer

ORDER BY USERNAME;

//////////////////////////////////Pizza-Low cost and High cost pizza..

Select pizza_type, pizza_name from pizza

Where amount IN(select max(amount) from pizza) or

Amount IN(SELECT MIN(amount) from pizza)

LIMIT 2;
//////////////////////////////////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;

/////////////////////////////////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);
```

```
/////////////////////////////////////Event Hall-Customer details with booking done..
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;
//////////////////////////////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;


//////////////////////////////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;


/////////////////////////////Room Details Based On Location..

select
ROOM_DETAILS.ROOM_ID,ROOM_DETAILS.ROOM_TYPE,ROOM_DETAILS.MEMBER_CAPACITY,ROOM_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;


/////////////////////////////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;

////////////////////////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;
```

///////////////////////////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;
```

//////////////////////////////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;
```

Concatenating Details ..

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

ORDER BY MOVIE_DETAILS DESC;
```

/////////////////////////Pizza-Highest Business Customer Details..

```
select pizza.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 Max_Amount
desc limit 1;
```

///////////////////////////Pizza-Total Cost of Pizza Ordered..

```
select cust_id, pizza_name, count(cust_id) as 'Times taken', sum(amount) as 'Total cost'
from pizza
```

```
where amount > 1200 group by pizza_name , cust_id order by cust_id asc;
//////////////////////////// Pizza-Extra Large Pizza ..

///////////////////////////////Event Hall-Customer having Average payment..


select cd.customer_name,round(avg(hd.cost_perday)) as A from t_customer_details cd join
t_hall_booking hb

on cd.customer_id=hb.customer_id

join t_hall_details hd on hd.hall_id=hb.hall_id

group by cd.customer_name

having A>(select max(hdd.cost_perday)from t_hall_details hdd join

t_hall_booking hbb on hbb.hall_id=hdd.hall_id

join t_customer_details cdd on cdd.customer_id=hbb.customer_id

where cdd.customer_name='Suraj Kumar')

order by A desc;
////////////////////////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;

////////////////////////////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');
```

/////////////////////////////Movie - Modify the datatype..

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

////////////////////////////Create Movie_Master table set1..

///////////////////////////////Hospital- Change the datatype/column ..

```
alter table patient modify

contact_number int(10);

alter table patient change p_age patient_age int;
```

///////////////////////////////Hospital- Add a new column set1..

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

///////////////////////////////////Pizza Store- Update PIZZA table discount1.2..

```
UPDATE pizza

set amount  = (amount * 0.95)/100

Where pizza_type = " Extra Large" ;
```

///////////////////////////////////Pizza Store- Alter table-Foreign key 1.1..

```
ALTER TABLE pizza ADD CONSTRAINT FK1 FOREIGN KEY (cust_id) REFERENCES
customer(cust_id);

ALTER TABLE pizza ADD CONSTRAINT FK2 FOREIGN KEY (partner_id) REFERENCES
delivery_partner(partner_id);
```

////////////////////////////////////Pizza Store - Update PIZZA table 1.2..

///////////////////////////Pizza Store- Alter table Pizza1.1..

//////////////////////////////Event Hall- Update the event date1.2..

////////////////////////////Event Hall -Update T_HALL_DETAILS table1.2..

//////////////////////////////////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);

///////////////////////////////////Dream Home- Update t_flat_details table1.2..


///////////////////////////////////Dream Home- Alter table t_flat_booking1.1..

///////////////////////////////////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;
///////////////////////////////////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);
///////////////////////////////////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);


///////////////////////////////////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)
);
//////////////////////////////////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%";

//////////////////////////////////Hospital- Alter T_HOSPITAL table 1.1..
alter table t_hospital
add foreign key (available_doctor) references t_doctor(doctor_id);


//////////////////////////////////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;

//////////////////////////////////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);

//////////////////////////////////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);

//////////////////////////////////Pizza Store-Update PIZZA table 1.2..


//////////////////////////////////Pizza Store- Alter table Pizza 1.1..

//////////////////////////////////Patient Appointment Details based on reason..

////////////////////////////////////2.Pizza-Highest Business Date..

//////////////////////////////////2. 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;

///////////////////////////////////2. 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.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;

///////////////////////////////////2.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

////////////////////////2.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;
```

```
///////////////////////////////////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;

///////////////////////////////////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;

///////////////////////////////////2.Pizza-Delivery partner details..


///////////////////////////////////2.Pizza-Highest Selling Pizza..
SELECT order_date, sum(amount) as Highest_Business
FROM pizza
GROUP BY order_date
ORDER BY Highest_Business DESC
LIMIT 1;
///////////////////////////////////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
;
///////////////////////////////////////2.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;
```