////////////////////////////.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;**