ASSIGNMENT-5

NAME: MANOJ SAHIT REDDY VANGA
ROLL NUMBER: 19BCS110

QUESTION-1:

Illustrate logical ANY, ALL and LIKE operator- the queries should be relevant to your respective databases 3 queries for each operator. One query explaining the difference between ANY and ALL

QUERY-1 FOR ANY:

```
use [T3 Travel]
select phone_number
from T3_EmployeeDetails
where designation=any(select designation
                     from T3_EmployeeDetails
                     where salary=12500);
```

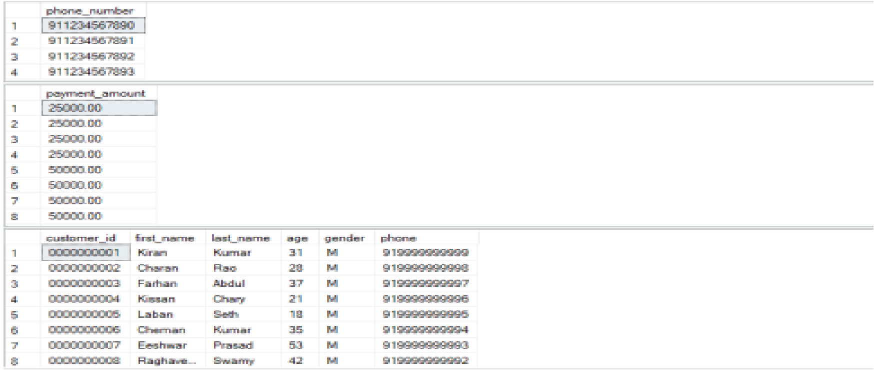
QUERY-2 FOR ANY:

```
use [T3 Travel]
select payment_amount
from T3_BookingDetails
where customer_id=any(select customer_id
                       from T3_CustomerDetails
                       where age<30);
```

QUERY-3 FOR ANY:

```
use [T3 Travel]
select *
from T3_CustomerDetails
where age< any(select age
                from T3_CustomerDetails
                where gender='M');
```

OUTPUT FOR ABOVE THREE QUERIES:



	phone_number
1	911234567890
2	911234567891
3	911234567892
4	911234567893

	payment_amount
1	25000.00
2	25000.00
3	25000.00
4	25000.00
5	50000.00
6	50000.00
7	50000.00
8	50000.00

	customer_id	first_name	last_name	age	gender	phone
1	0000000001	Kiran	Kumar	31	M	919999999999
2	0000000002	Charan	Rao	28	M	919999999998
3	0000000003	Farhan	Abdul	37	M	919999999997
4	0000000004	Kissan	Chary	21	M	919999999996
5	0000000005	Laban	Seth	18	M	919999999995
6	0000000006	Cheman	Kumar	35	M	919999999994
7	0000000007	Eeshwar	Prasad	53	M	919999999993
8	0000000008	Raghave...	Swamy	42	M	919999999992

QUERY-1 FOR ALL:

```
use [T3 Travel]
select phone_number
from T3_EmployeeDetails
where designation=ALL(select designation
                       from T3_EmployeeDetails
                       where salary=12500);
```

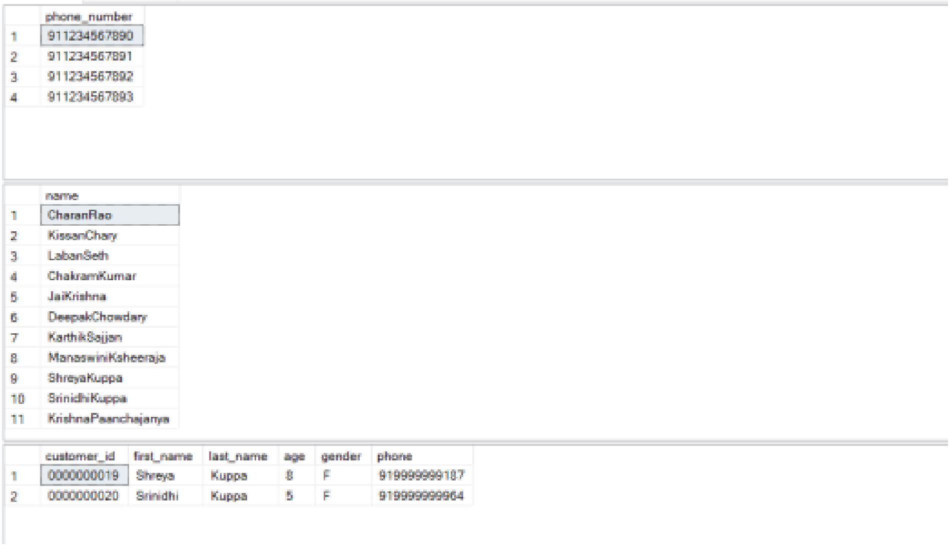
QUERY-2 FOR ALL:

```
use [T3 Travel]
select concat(first_name,last_name) as name
from T3_CustomerDetails
where age<ALL(select age
                            from T3_CustomerDetails
                            where age>30);
```

QUERY-3 FOR ALL:

```
use [T3 Travel]
select *
from T3_CustomerDetails
where age< ALL(select age
               from T3_CustomerDetails
               where gender='M');
```

OUTPUT FOR ABOVE THREE QUERIES:



	phone_number
1	911234567890
2	911234567891
3	911234567892
4	911234567893

	name
1	CharanRao
2	KissanChary
3	LabanSeth
4	ChakramKumar
5	JaiKrishna
6	DeepakChowdary
7	KarthikSajjan
8	ManaswiniKsheeraja
9	ShreyaKuppa
10	SrinidhiKuppa
11	KrishnaPaanchajanya

	customer_id	first_name	last_name	age	gender	phone
1	0000000019	Shreya	Kuppa	8	F	919999999187
2	0000000020	Srinidhi	Kuppa	5	F	919999999964

QUERY-1 FOR LIKE:

```
use [T3 Travel]
select name,designation
from T3_EmployeeDetails
where employee_id LIKE '02%';
```

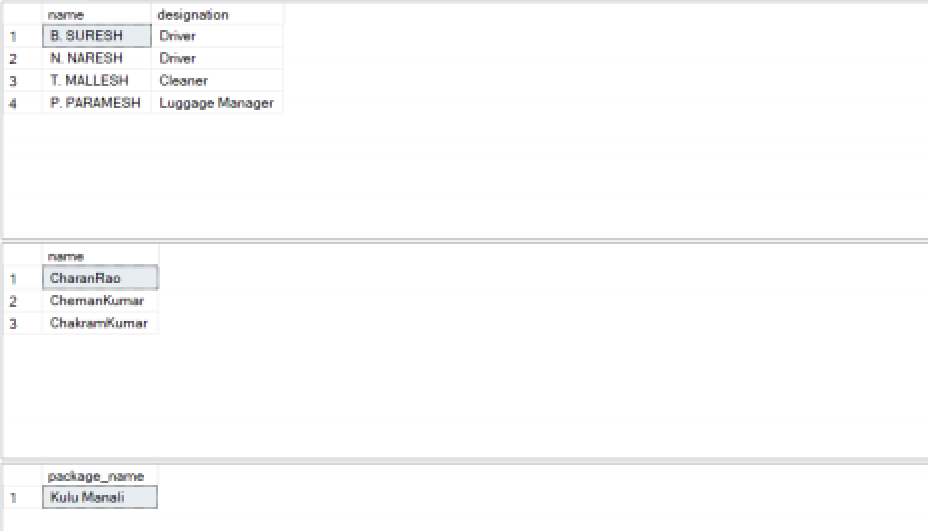
QUERY-2 FOR LIKE:

```
use [T3 Travel]
select concat(first_name,last_name) as name
from T3_CustomerDetails
where first_name LIKE 'C%';
```

QUERY-3 FOR LIKE:

```
use [T3 Travel]
select distinct package_name
from [T3_Package Details]
where booking_id LIKE '01%';
```

OUTPUT FOR ABOVE THREE QUERIES:



	name	designation
1	B. SURESH	Driver
2	N. NARESH	Driver
3	T. MALLESH	Cleaner
4	P. PARAMESH	Luggage Manager

	name
1	CharanRao
2	ChemanKumar
3	ChakramKumar

	package_name
1	Kulu Manali

QUERY TO DIFFERENTIATE BETWEEN ALL AND ANY:

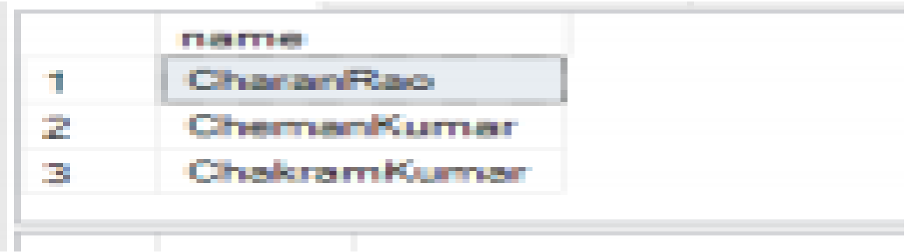
ANY:

```
use [T3 Travel]
select concat(first_name,last_name) as name
from T3_CustomerDetails
where first_name=ANY(select first_name
                                         from T3_CustomerDetails
                                         where first_name like 'C%');
```

ALL:

```
use [T3 Travel]
select concat(first_name,last_name) as name
from T3_CustomerDetails
where first_name=all(select first_name
                                         from T3_CustomerDetails
                                         where first_name like 'C%');
```

OUTPUT:



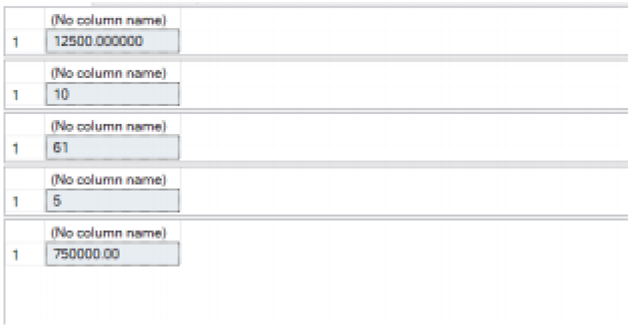
	name
1	CharanRao
2	ChemanKumar
3	ChakramKumar

QUESTION-2:

One query for each Aggregate function.

```
use [T3 Travel]
select avg(salary)
from T3_EmployeeDetails
where designation='Driver';
select count(*)
from [T3_Package Details]
where cost>25000;
select max(age)
from T3_CustomerDetails;
select min(age)
from T3_CustomerDetails;
select sum(payment_amount)
from T3_BookingDetails;
```

OUTPUT:



	(No column name)
1	12500.000000

	(No column name)
1	10

	(No column name)
1	61

	(No column name)
1	5

	(No column name)
1	750000.00

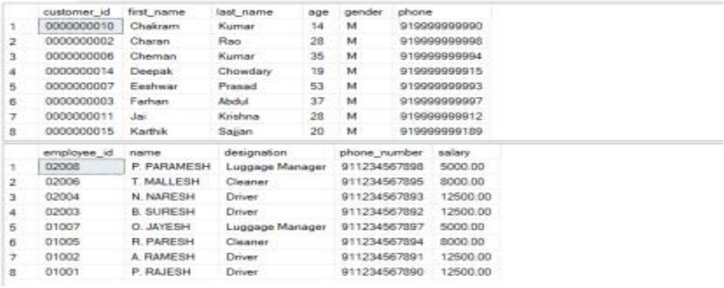
QUESTION-3:

Illustrate the usage of order by, group by and having clause (2 queries for each case)

QUERIES FOR ORDER BY:

```
use [T3 Travel]
select *
from T3_CustomerDetails
order by first_name asc;
select *
from T3_EmployeeDetails
order by employee_id desc;
```

OUTPUT:



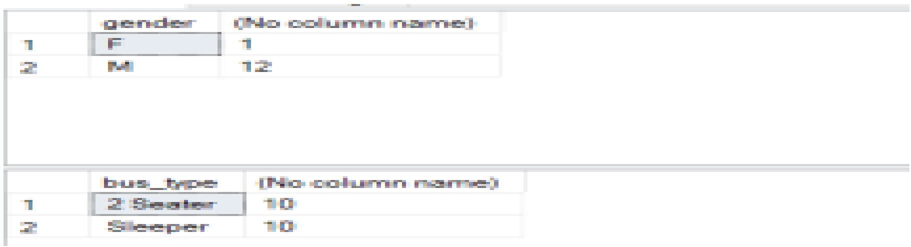
	customer_id	first_name	last_name	age	gender	phone
1	0000000010	Chakram	Kumar	14	M	919999999990
2	0000000002	Charan	Rao	28	M	919999999998
3	0000000006	Cheman	Kumar	35	M	919999999994
4	0000000014	Deepak	Chowdary	19	M	919999999915
5	0000000007	Eeshwar	Prasad	53	M	919999999993
6	0000000003	Farhan	Abdul	37	M	919999999997
7	0000000011	Jai	Krishna	28	M	919999999912
8	0000000015	Karthik	Sajjan	20	M	919999999189

	employee_id	name	designation	phone_number	salary
1	02008	P. PARAMESH	Luggage Manager	911234567898	5000.00
2	02006	T. MALLESH	Cleaner	911234567895	8000.00
3	02004	N. NARESH	Driver	911234567893	12500.00
4	02003	B. SURESH	Driver	911234567892	12500.00
5	01007	O. JAYESH	Luggage Manager	911234567897	5000.00
6	01005	R. PARESH	Cleaner	911234567894	8000.00
7	01002	A. RAMESH	Driver	911234567891	12500.00
8	01001	P. RAJESH	Driver	911234567890	12500.00

QUERIES FOR GROUP BY:

```
use [T3 Travel]
select gender ,count(*)
from T3_CustomerDetails
where age>21
group by gender;
select bus_type,count(*)
from T3Bus
group by bus_type;
```

OUTPUT:



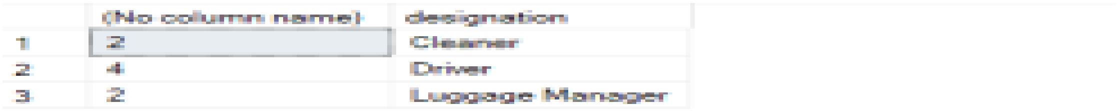
	gender	(No column name)
1	F	1
2	M	12

	bus_type	(No column name)
1	2 Seater	10
2	Sleeper	10

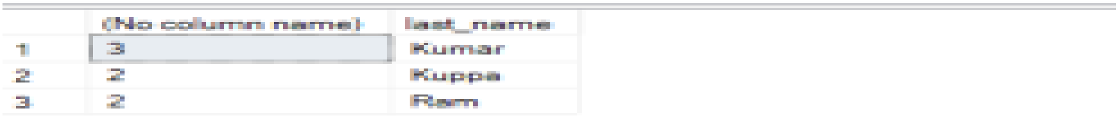
QUERIES FOR HAVING:

```
use [T3 Travel]
select count(employee_id),designation
from T3_EmployeeDetails
group by designation
having count(employee_id)>1;
select count(customer_id),last_name
from T3_CustomerDetails
group by last_name
having count(customer_id)>1;
```

OUTPUT:



	(No column name)	designation
1	2	Cleaner
2	4	Driver
3	2	Luggage Manager



	(No column name)	last_name
1	3	Kumar
2	2	Kuppa
3	2	Ram

QUESTION-4:

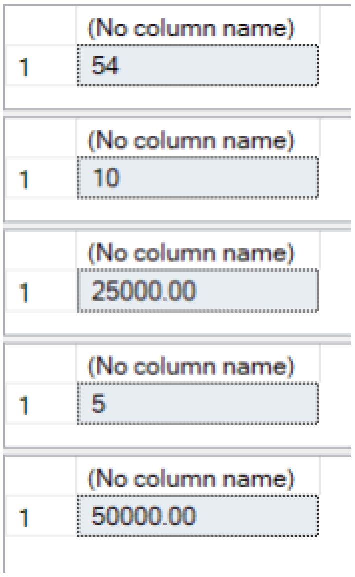
Use Aggregate function with group by and having

QUERIES:

```
use [T3 Travel]
select avg(age)
from T3_CustomerDetails
group by last_name
having last_name='Ram'
select count(booking_id)
from [T3_Package Details]
group by cost
```

```
having cost=50000;
select max(payment_amount)
from T3_BookingDetails
group by payment_dateTime
having payment_dateTime='2021-02-19 09:37:00.000';
select min(age)
from T3_CustomerDetails
group by last_name
having last_name='vanga';
select sum(salary)
from T3_EmployeeDetails
group by designation
having designation='Driver';
```

OUTPUT:



```
     (No column name)
1    54

     (No column name)
1    10

     (No column name)
1    25000.00

     (No column name)
1    5

     (No column name)
1    50000.00
```

QUESTION-5:

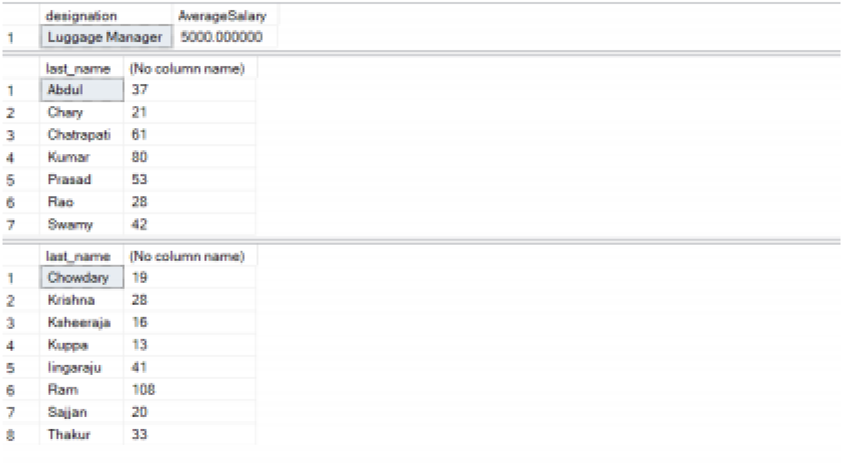
Write at least 3 nested queries using order by, group by and having clause.

QUERIES:

```
use [T3 Travel]
select designation,avg(salary) as AverageSalary
from T3_EmployeeDetails
where designation='Luggage Manager'
group by designation
having avg(salary)<(select avg(salary)
                                         from T3_EmployeeDetails
                                         where designation='Cleaner');
select last_name,sum(age)
from T3_CustomerDetails
where customer_id=ANY(select customer_id
                                         from T3_BookingDetails
                                         where payment_amount=25000)
group by last_name
having last_name like '%a%';
```

```
select last_name,sum(age)
from T3_CustomerDetails
where customer_id=ANY(select customer_id
                                          from T3_BookingDetails
                                          where payment_amount=50000)
group by last_name
having last_name like '%a%';
```

OUTPUT:



	designation	AverageSalary
1	Luggage Manager	5000.000000

	last_name	(No column name)
1	Abdul	37
2	Chary	21
3	Chatrapati	61
4	Kumar	80
5	Prasad	53
6	Rao	28
7	Swamy	42

	last_name	(No column name)
1	Chowdary	19
2	Krishna	28
3	Kaheeraja	16
4	Kuppa	13
5	lingaraju	41
6	Ram	108
7	Sajjan	20
8	Thakur	33

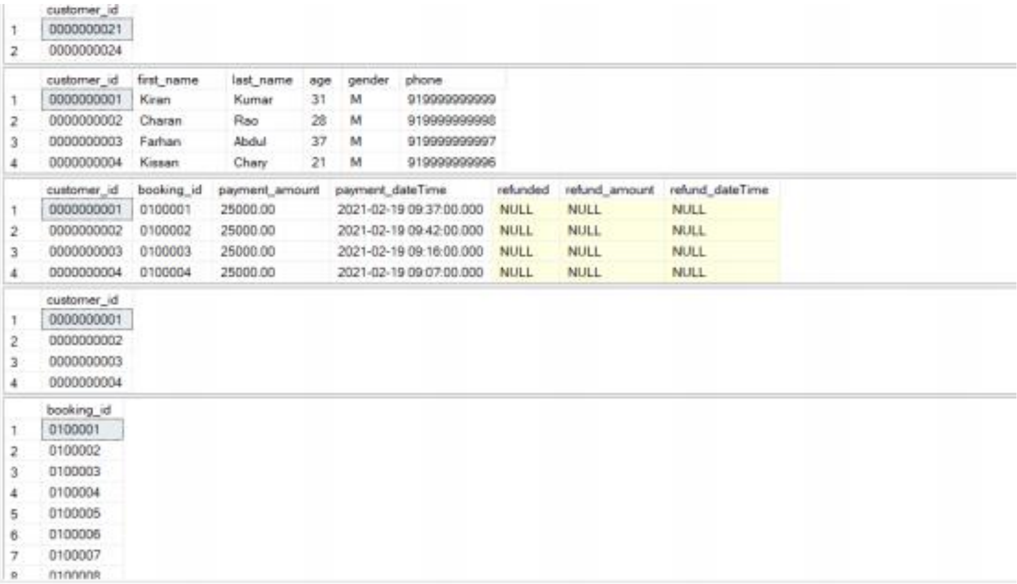
QUESTION-6:

Illustrate the Usage of Except, Exists, Not Exists, Union, Intersection

QUERIES:

```
use [T3 Travel]
select customer_id
from T3_CustomerDetails
except select customer_id
              from T3_BookingDetails;
select *
from T3_CustomerDetails
where exists(select customer_id
                            from T3_BookingDetails
                            where payment_amount=25000);
select *
from T3_BookingDetails
where not exists(select customer_id
                                   from T3_CustomerDetails
                                   where age>18);
select customer_id
from T3_BookingDetails union select customer_id
                                                        from T3_CustomerDetails;
select booking_id
from T3_BookingDetails intersect select booking_id
                                                               from T3_DestinationDetails;
```

OUTPUT:



	customer_id
1	0000000021
2	0000000024

	customer_id	first_name	last_name	age	gender	phone
1	0000000001	Kiran	Kumar	31	M	919999999999
2	0000000002	Charan	Rao	28	M	919999999998
3	0000000003	Farhan	Abdul	37	M	919999999997
4	0000000004	Kissan	Chary	21	M	919999999996

	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime
1	0000000001	0100001	25000.00	2021-02-19 09:37:00.000	NULL	NULL	NULL
2	0000000002	0100002	25000.00	2021-02-19 09:42:00.000	NULL	NULL	NULL
3	0000000003	0100003	25000.00	2021-02-19 09:16:00.000	NULL	NULL	NULL
4	0000000004	0100004	25000.00	2021-02-19 09:07:00.000	NULL	NULL	NULL

	customer_id
1	0000000001
2	0000000002
3	0000000003
4	0000000004

	booking_id
1	0100001
2	0100002
3	0100003
4	0100004
5	0100005
6	0100006
7	0100007
8	0100008

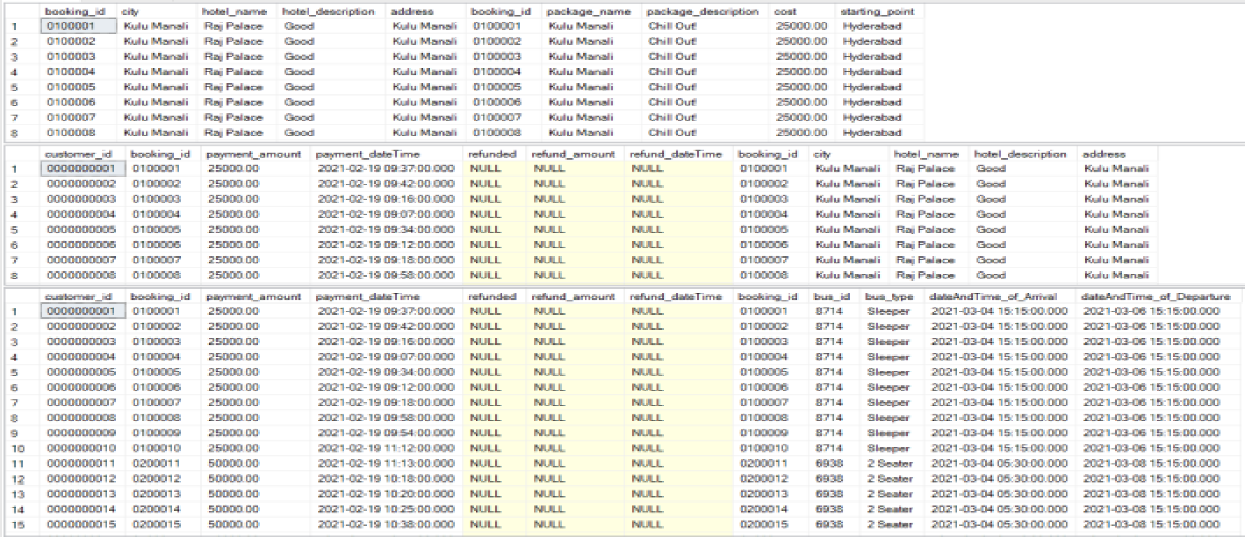
QUESTION-7:

INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN- 3 queries for each instance

QUERIES FOR INNER JOIN:

```
use [T3 Travel]
select *
from T3_DestinationDetails as dest inner join [T3_Package Details] as pack on
dest.booking_id=pack.booking_id;
select *
from T3_BookingDetails as booking inner join T3_DestinationDetails as dest on
booking.booking_id=dest.booking_id;
select *
from T3_BookingDetails as booking inner join T3Bus as bus on
booking.booking_id=bus.booking_id;
```

OUTPUT:



	booking_id	city	hotel_name	hotel_description	address	booking_id	package_name	package_description	cost	starting_point
1	0100001	Kulu Manali	Raj Palace	Good	Kulu Manali	0100001	Kulu Manali	Chill Out!	25000.00	Hyderabad
2	0100002	Kulu Manali	Raj Palace	Good	Kulu Manali	0100002	Kulu Manali	Chill Out!	25000.00	Hyderabad
3	0100003	Kulu Manali	Raj Palace	Good	Kulu Manali	0100003	Kulu Manali	Chill Out!	25000.00	Hyderabad
4	0100004	Kulu Manali	Raj Palace	Good	Kulu Manali	0100004	Kulu Manali	Chill Out!	25000.00	Hyderabad
5	0100005	Kulu Manali	Raj Palace	Good	Kulu Manali	0100005	Kulu Manali	Chill Out!	25000.00	Hyderabad
6	0100006	Kulu Manali	Raj Palace	Good	Kulu Manali	0100006	Kulu Manali	Chill Out!	25000.00	Hyderabad
7	0100007	Kulu Manali	Raj Palace	Good	Kulu Manali	0100007	Kulu Manali	Chill Out!	25000.00	Hyderabad
8	0100008	Kulu Manali	Raj Palace	Good	Kulu Manali	0100008	Kulu Manali	Chill Out!	25000.00	Hyderabad

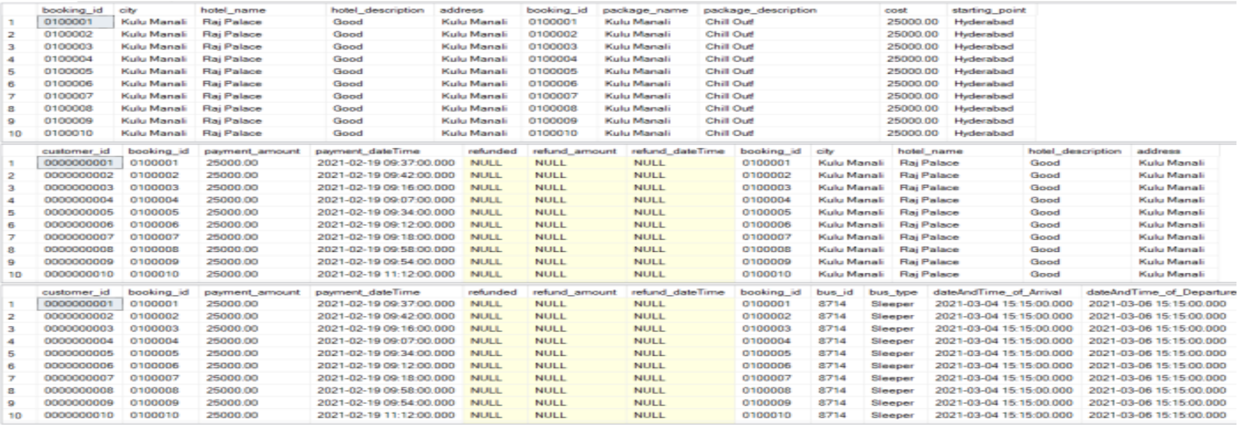
	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime	booking_id	city	hotel_name	hotel_description	address
1	0000000001	0100001	25000.00	2021-02-19 09:37:00.000	NULL	NULL	NULL	0100001	Kulu Manali	Raj Palace	Good	Kulu Manali
2	0000000002	0100002	25000.00	2021-02-19 09:42:00.000	NULL	NULL	NULL	0100002	Kulu Manali	Raj Palace	Good	Kulu Manali
3	0000000003	0100003	25000.00	2021-02-19 09:16:00.000	NULL	NULL	NULL	0100003	Kulu Manali	Raj Palace	Good	Kulu Manali
4	0000000004	0100004	25000.00	2021-02-19 09:07:00.000	NULL	NULL	NULL	0100004	Kulu Manali	Raj Palace	Good	Kulu Manali
5	0000000005	0100005	25000.00	2021-02-19 09:34:00.000	NULL	NULL	NULL	0100005	Kulu Manali	Raj Palace	Good	Kulu Manali
6	0000000006	0100006	25000.00	2021-02-19 09:12:00.000	NULL	NULL	NULL	0100006	Kulu Manali	Raj Palace	Good	Kulu Manali
7	0000000007	0100007	25000.00	2021-02-19 09:18:00.000	NULL	NULL	NULL	0100007	Kulu Manali	Raj Palace	Good	Kulu Manali
8	0000000008	0100008	25000.00	2021-02-19 09:58:00.000	NULL	NULL	NULL	0100008	Kulu Manali	Raj Palace	Good	Kulu Manali

	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime	booking_id	bus_id	bus_type	dateAndTime_of_Arrival	dateAndTime_of_Departure
1	0000000001	0100001	25000.00	2021-02-19 09:37:00.000	NULL	NULL	NULL	0100001	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
2	0000000002	0100002	25000.00	2021-02-19 09:42:00.000	NULL	NULL	NULL	0100002	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
3	0000000003	0100003	25000.00	2021-02-19 09:16:00.000	NULL	NULL	NULL	0100003	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
4	0000000004	0100004	25000.00	2021-02-19 09:07:00.000	NULL	NULL	NULL	0100004	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
5	0000000005	0100005	25000.00	2021-02-19 09:34:00.000	NULL	NULL	NULL	0100005	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
6	0000000006	0100006	25000.00	2021-02-19 09:12:00.000	NULL	NULL	NULL	0100006	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
7	0000000007	0100007	25000.00	2021-02-19 09:18:00.000	NULL	NULL	NULL	0100007	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
8	0000000008	0100008	25000.00	2021-02-19 09:58:00.000	NULL	NULL	NULL	0100008	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
9	0000000009	0100009	25000.00	2021-02-19 09:54:00.000	NULL	NULL	NULL	0100009	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
10	0000000010	0100010	25000.00	2021-02-19 11:12:00.000	NULL	NULL	NULL	0100010	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
11	0000000011	0200011	50000.00	2021-02-19 11:13:00.000	NULL	NULL	NULL	0200011	6938	2 Seater	2021-03-04 05:30:00.000	2021-03-08 15:15:00.000
12	0000000012	0200012	50000.00	2021-02-19 10:18:00.000	NULL	NULL	NULL	0200012	6938	2 Seater	2021-03-04 05:30:00.000	2021-03-08 15:15:00.000
13	0000000013	0200013	50000.00	2021-02-19 10:20:00.000	NULL	NULL	NULL	0200013	6938	2 Seater	2021-03-04 05:30:00.000	2021-03-08 15:15:00.000
14	0000000014	0200014	50000.00	2021-02-19 10:25:00.000	NULL	NULL	NULL	0200014	6938	2 Seater	2021-03-04 05:30:00.000	2021-03-08 15:15:00.000
15	0000000015	0200015	50000.00	2021-02-19 10:38:00.000	NULL	NULL	NULL	0200015	6938	2 Seater	2021-03-04 05:30:00.000	2021-03-08 15:15:00.000

QUERIES FOR LEFT OUTER JOIN:

```
use [T3 Travel]
select *
from T3_DestinationDetails as dest left outer join [T3_Package Details] as pack on
dest.booking_id=pack.booking_id;
select *
from T3_BookingDetails as booking left outer join T3_DestinationDetails as dest on
booking.booking_id=dest.booking_id;
select *
from T3_BookingDetails as booking left outer join T3Bus as bus on
booking.booking_id=bus.booking_id;
```

OUTPUT:



	booking_id	city	hotel_name	hotel_description	address	booking_id	package_name	package_description	cost	starting_point
1	0100001	Kulu Manali	Raj Palace	Good	Kulu Manali	0100001	Kulu Manali	Chill Out	25000.00	Hyderabad
2	0100002	Kulu Manali	Raj Palace	Good	Kulu Manali	0100002	Kulu Manali	Chill Out	25000.00	Hyderabad
3	0100003	Kulu Manali	Raj Palace	Good	Kulu Manali	0100003	Kulu Manali	Chill Out	25000.00	Hyderabad
4	0100004	Kulu Manali	Raj Palace	Good	Kulu Manali	0100004	Kulu Manali	Chill Out	25000.00	Hyderabad
5	0100005	Kulu Manali	Raj Palace	Good	Kulu Manali	0100005	Kulu Manali	Chill Out	25000.00	Hyderabad
6	0100006	Kulu Manali	Raj Palace	Good	Kulu Manali	0100006	Kulu Manali	Chill Out	25000.00	Hyderabad
7	0100007	Kulu Manali	Raj Palace	Good	Kulu Manali	0100007	Kulu Manali	Chill Out	25000.00	Hyderabad
8	0100008	Kulu Manali	Raj Palace	Good	Kulu Manali	0100008	Kulu Manali	Chill Out	25000.00	Hyderabad
9	0100009	Kulu Manali	Raj Palace	Good	Kulu Manali	0100009	Kulu Manali	Chill Out	25000.00	Hyderabad
10	0100010	Kulu Manali	Raj Palace	Good	Kulu Manali	0100010	Kulu Manali	Chill Out	25000.00	Hyderabad

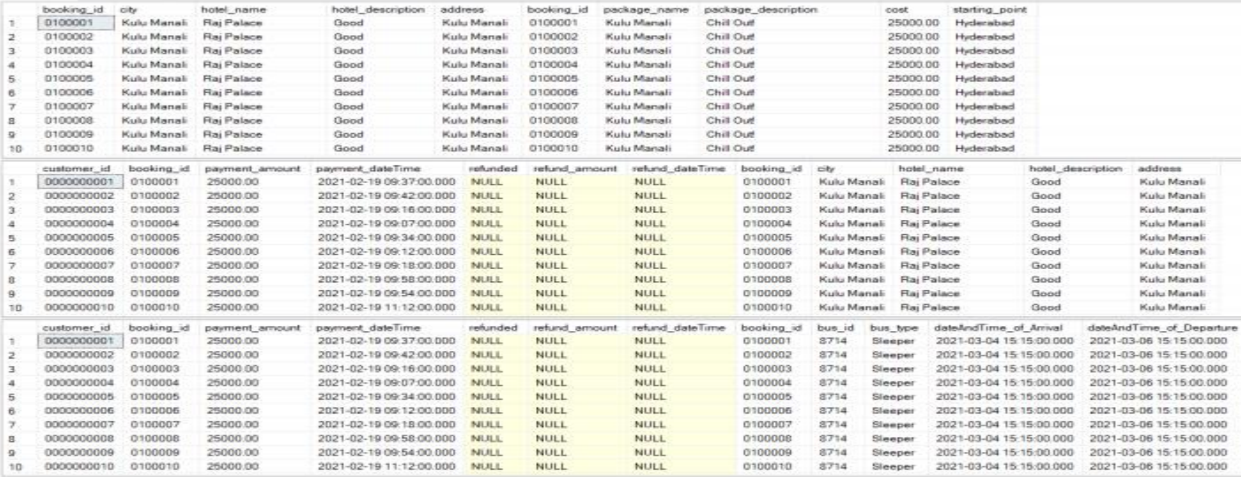
	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime	booking_id	city	hotel_name	hotel_description	address
1	0000000001	0100001	25000.00	2021-02-19 09:37:00.000	NULL	NULL	NULL	0100001	Kulu Manali	Raj Palace	Good	Kulu Manali
2	0000000002	0100002	25000.00	2021-02-19 09:42:00.000	NULL	NULL	NULL	0100002	Kulu Manali	Raj Palace	Good	Kulu Manali
3	0000000003	0100003	25000.00	2021-02-19 09:16:00.000	NULL	NULL	NULL	0100003	Kulu Manali	Raj Palace	Good	Kulu Manali
4	0000000004	0100004	25000.00	2021-02-19 09:07:00.000	NULL	NULL	NULL	0100004	Kulu Manali	Raj Palace	Good	Kulu Manali
5	0000000005	0100005	25000.00	2021-02-19 09:34:00.000	NULL	NULL	NULL	0100005	Kulu Manali	Raj Palace	Good	Kulu Manali
6	0000000006	0100006	25000.00	2021-02-19 09:12:00.000	NULL	NULL	NULL	0100006	Kulu Manali	Raj Palace	Good	Kulu Manali
7	0000000007	0100007	25000.00	2021-02-19 09:18:00.000	NULL	NULL	NULL	0100007	Kulu Manali	Raj Palace	Good	Kulu Manali
8	0000000008	0100008	25000.00	2021-02-19 09:58:00.000	NULL	NULL	NULL	0100008	Kulu Manali	Raj Palace	Good	Kulu Manali
9	0000000009	0100009	25000.00	2021-02-19 09:54:00.000	NULL	NULL	NULL	0100009	Kulu Manali	Raj Palace	Good	Kulu Manali
10	0000000010	0100010	25000.00	2021-02-19 11:12:00.000	NULL	NULL	NULL	0100010	Kulu Manali	Raj Palace	Good	Kulu Manali

	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime	booking_id	bus_id	bus_type	dateAndTime_of_Arrival	dateAndTime_of_Departure
1	0000000001	0100001	25000.00	2021-02-19 09:37:00.000	NULL	NULL	NULL	0100001	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
2	0000000002	0100002	25000.00	2021-02-19 09:42:00.000	NULL	NULL	NULL	0100002	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
3	0000000003	0100003	25000.00	2021-02-19 09:16:00.000	NULL	NULL	NULL	0100003	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
4	0000000004	0100004	25000.00	2021-02-19 09:07:00.000	NULL	NULL	NULL	0100004	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
5	0000000005	0100005	25000.00	2021-02-19 09:34:00.000	NULL	NULL	NULL	0100005	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
6	0000000006	0100006	25000.00	2021-02-19 09:12:00.000	NULL	NULL	NULL	0100006	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
7	0000000007	0100007	25000.00	2021-02-19 09:18:00.000	NULL	NULL	NULL	0100007	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
8	0000000008	0100008	25000.00	2021-02-19 09:58:00.000	NULL	NULL	NULL	0100008	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
9	0000000009	0100009	25000.00	2021-02-19 09:54:00.000	NULL	NULL	NULL	0100009	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
10	0000000010	0100010	25000.00	2021-02-19 11:12:00.000	NULL	NULL	NULL	0100010	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000

QUERIES FOR RIGHT OUTER JOIN:

```
use [T3 Travel]
select *
from T3_DestinationDetails as dest right outer join [T3_Package Details] as pack on
dest.booking_id=pack.booking_id;
select *
from T3_BookingDetails as booking right outer join T3_DestinationDetails as dest on
booking.booking_id=dest.booking_id;
select *
from T3_BookingDetails as booking right outer join T3Bus as bus on
booking.booking_id=bus.booking_id;
```

OUTPUT:



	booking_id	city	hotel_name	hotel_description	address	booking_id	package_name	package_description	cost	starting_point
1	0100001	Kulu Manali	Raj Palace	Good	Kulu Manali	0100001	Kulu Manali	Chill Out	25000.00	Hyderabad
2	0100002	Kulu Manali	Raj Palace	Good	Kulu Manali	0100002	Kulu Manali	Chill Out	25000.00	Hyderabad
3	0100003	Kulu Manali	Raj Palace	Good	Kulu Manali	0100003	Kulu Manali	Chill Out	25000.00	Hyderabad
4	0100004	Kulu Manali	Raj Palace	Good	Kulu Manali	0100004	Kulu Manali	Chill Out	25000.00	Hyderabad
5	0100005	Kulu Manali	Raj Palace	Good	Kulu Manali	0100005	Kulu Manali	Chill Out	25000.00	Hyderabad
6	0100006	Kulu Manali	Raj Palace	Good	Kulu Manali	0100006	Kulu Manali	Chill Out	25000.00	Hyderabad
7	0100007	Kulu Manali	Raj Palace	Good	Kulu Manali	0100007	Kulu Manali	Chill Out	25000.00	Hyderabad
8	0100008	Kulu Manali	Raj Palace	Good	Kulu Manali	0100008	Kulu Manali	Chill Out	25000.00	Hyderabad
9	0100009	Kulu Manali	Raj Palace	Good	Kulu Manali	0100009	Kulu Manali	Chill Out	25000.00	Hyderabad
10	0100010	Kulu Manali	Raj Palace	Good	Kulu Manali	0100010	Kulu Manali	Chill Out	25000.00	Hyderabad

	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime	booking_id	city	hotel_name	hotel_description	address
1	0000000001	0100001	25000.00	2021-02-19 09:37:00.000	NULL	NULL	NULL	0100001	Kulu Manali	Raj Palace	Good	Kulu Manali
2	0000000002	0100002	25000.00	2021-02-19 09:42:00.000	NULL	NULL	NULL	0100002	Kulu Manali	Raj Palace	Good	Kulu Manali
3	0000000003	0100003	25000.00	2021-02-19 09:16:00.000	NULL	NULL	NULL	0100003	Kulu Manali	Raj Palace	Good	Kulu Manali
4	0000000004	0100004	25000.00	2021-02-19 09:07:00.000	NULL	NULL	NULL	0100004	Kulu Manali	Raj Palace	Good	Kulu Manali
5	0000000005	0100005	25000.00	2021-02-19 09:34:00.000	NULL	NULL	NULL	0100005	Kulu Manali	Raj Palace	Good	Kulu Manali
6	0000000006	0100006	25000.00	2021-02-19 09:12:00.000	NULL	NULL	NULL	0100006	Kulu Manali	Raj Palace	Good	Kulu Manali
7	0000000007	0100007	25000.00	2021-02-19 09:18:00.000	NULL	NULL	NULL	0100007	Kulu Manali	Raj Palace	Good	Kulu Manali
8	0000000008	0100008	25000.00	2021-02-19 09:58:00.000	NULL	NULL	NULL	0100008	Kulu Manali	Raj Palace	Good	Kulu Manali
9	0000000009	0100009	25000.00	2021-02-19 09:54:00.000	NULL	NULL	NULL	0100009	Kulu Manali	Raj Palace	Good	Kulu Manali
10	0000000010	0100010	25000.00	2021-02-19 11:12:00.000	NULL	NULL	NULL	0100010	Kulu Manali	Raj Palace	Good	Kulu Manali

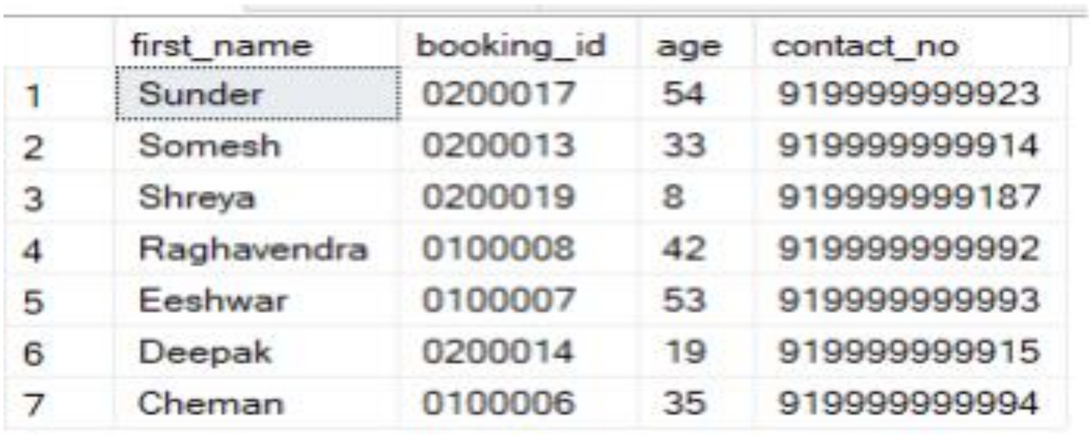
	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime	booking_id	bus_id	bus_type	dateAndTime_of_Arrival	dateAndTime_of_Departure
1	0000000001	0100001	25000.00	2021-02-19 09:37:00.000	NULL	NULL	NULL	0100001	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
2	0000000002	0100002	25000.00	2021-02-19 09:42:00.000	NULL	NULL	NULL	0100002	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
3	0000000003	0100003	25000.00	2021-02-19 09:16:00.000	NULL	NULL	NULL	0100003	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
4	0000000004	0100004	25000.00	2021-02-19 09:07:00.000	NULL	NULL	NULL	0100004	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
5	0000000005	0100005	25000.00	2021-02-19 09:34:00.000	NULL	NULL	NULL	0100005	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
6	0000000006	0100006	25000.00	2021-02-19 09:12:00.000	NULL	NULL	NULL	0100006	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
7	0000000007	0100007	25000.00	2021-02-19 09:18:00.000	NULL	NULL	NULL	0100007	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
8	0000000008	0100008	25000.00	2021-02-19 09:58:00.000	NULL	NULL	NULL	0100008	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
9	0000000009	0100009	25000.00	2021-02-19 09:54:00.000	NULL	NULL	NULL	0100009	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000
10	0000000010	0100010	25000.00	2021-02-19 11:12:00.000	NULL	NULL	NULL	0100010	8714	Sleeper	2021-03-04 15:15:00.000	2021-03-06 15:15:00.000

QUESTION-8:

Use all the above condition in JOIN as well.

```
select first_name,min(booking_id) as booking_id,avg(age) as age,max(phone) as
phone_number
from T3_CustomerDetails as customer
join
T3_BookingDetails as booking on customer.customer_id=booking.customer_id
group by first_name having first_name like '%e%' order by first_name desc;
```

OUTPUT:



	first_name	booking_id	age	contact_no
1	Sunder	0200017	54	919999999923
2	Somesh	0200013	33	919999999914
3	Shreya	0200019	8	919999999187
4	Raghavendra	0100008	42	919999999992
5	Eeshwar	0100007	53	919999999993
6	Deepak	0200014	19	919999999915
7	Cheman	0100006	35	919999999994