

# DBMS LAB ASSIGNMENT 5

---

NAME: KASHA SINGH

REG. NO.: 19BCS051

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

*/\*ANY OPERATOR\*/*

```
SELECT * FROM T3_CustomerDetails
WHERE customer_id = ANY(SELECT customer_id FROM T3_BookingDetails WHERE booking_id = 111111);
```

```
SELECT * FROM T3_BookingDetails
WHERE booking_id = ANY(SELECT booking_id FROM T3_PackageDetails WHERE cost < 80000);
```

```
SELECT * FROM T3_CustomerDetails
WHERE customer_id = ANY(SELECT customer_id FROM T3_BookingDetails
WHERE booking_id = ANY(SELECT booking_id FROM T3_PackageDetails WHERE booking_id = 111112));
```

*/\*ALL OPERATOR\*/*

```
SELECT * FROM T3_CustomerDetails
WHERE customer_id = ALL(SELECT customer_id FROM T3_BookingDetails
WHERE booking_id = ALL(SELECT booking_id FROM T3_PackageDetails WHERE booking_id = 111112));
```

```
SELECT * FROM T3_CustomerDetails
WHERE customer_id = ALL(SELECT customer_id FROM T3_BookingDetails);
```

```
SELECT * FROM T3_BookingDetails
WHERE booking_id = ALL(SELECT booking_id FROM T3_PackageDetails WHERE cost = 90000);
```

*/\*LIKE OPERATOR\*/*

```
SELECT * FROM T3_CustomerDetails WHERE first_name LIKE 'M%';
SELECT * FROM T3_CustomerDetails WHERE last_name LIKE '%a';
SELECT * FROM T3_PackageDetails WHERE starting_point LIKE '_u%';
```

## DATABASE OUTPUT:

```
/*ANY OPERATOR*/
SELECT * FROM T3_CustomerDetails
WHERE customer_id = ANY(SELECT customer_id FROM T3_BookingDetails WHERE booking_id = 111111);

SELECT * FROM T3_BookingDetails
WHERE booking_id = ANY(SELECT booking_id FROM T3_PackageDetails WHERE cost<80000);

SELECT * FROM T3_CustomerDetails
WHERE customer_id = ANY(SELECT customer_id FROM T3_BookingDetails
WHERE booking_id = ANY(SELECT booking_id FROM T3_PackageDetails WHERE booking_id = 111112));
```

100 %

Results Messages

	customer_id	first_name	last_name	age	gender	phone
1	1000	Manya	Mehta	19	F	7623444591

	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime
1	1000	111111	60000.00	2021-02-27 13:01:20.460	0	NULL	NULL
2	1001	111112	60500.00	2021-02-27 13:03:19.273	0	NULL	NULL
3	1002	111113	70000.00	2021-02-27 13:03:19.273	0	NULL	NULL

	customer_id	first_name	last_name	age	gender	phone
1	1001	Mark	Stone	23	M	7666051848

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 5 rows

```
/*ALL OPERATOR*/
SELECT * FROM T3_CustomerDetails
WHERE customer_id = ALL(SELECT customer_id FROM T3_BookingDetails
WHERE booking_id = ALL(SELECT booking_id FROM T3_PackageDetails WHERE booking_id = 111112));

SELECT * FROM T3_CustomerDetails
WHERE customer_id = ALL(SELECT customer_id FROM T3_BookingDetails);

SELECT * FROM T3_BookingDetails
WHERE booking_id = ALL(SELECT booking_id FROM T3_PackageDetails WHERE cost = 90000);
```

100 %

Results Messages

	customer_id	first_name	last_name	age	gender	phone
1	1001	Mark	Stone	23	M	7666051848

	customer_id	first_name	last_name	age	gender	phone
--	-------------	------------	-----------	-----	--------	-------

	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime
1	1004	111114	90000.00	2021-02-27 13:03:43.393	0	NULL	NULL

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 2 rows

```

/*LIKE OPERATOR*/
SELECT * FROM T3_CustomerDetails WHERE first_name LIKE 'M%';
SELECT * FROM T3_CustomerDetails WHERE last_name LIKE '%a';
SELECT * FROM T3_PackageDetails WHERE starting_point LIKE '_u%';

```

	customer_id	first_name	last_name	age	gender	phone
1	1000	Manya	Mehta	19	F	7623444591
2	1001	Mark	Stone	23	M	7666051848
3	1011	Mona	Azuma	21	F	8899881234

	customer_id	first_name	last_name	age	gender	phone
1	1000	Manya	Mehta	19	F	7623444591
2	1011	Mona	Azuma	21	F	8899881234
3	1018	Tej	Sengupta	46	M	9933456876

	booking_id	package_name	package_description	cost	starting_point
1	111111	KULLU MANALI	KULLU	60000.00	KULLU
2	111114	MUNNAR	KERALA	90000.00	HUBLI
3	111112	OOTY	TAMIL NADU	60500.00	HUBLI

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 9 rows

2. One query for each Aggregate function.

```

/*AGGREGATE FUNCTION*/
SELECT COUNT(first_name) FROM T3_CustomerDetails
WHERE gender = 'M';

SELECT AVG(age) FROM T3_CustomerDetails;

SELECT MAX(cost) FROM T3_PackageDetails;

SELECT MIN(payment_amount) FROM T3_BookingDetails;

```

DATABASE OUTPUT:

```

/*AGGREGATE FUNCTION*/
SELECT COUNT(first_name) FROM T3_CustomerDetails
WHERE gender = 'M';

SELECT AVG(age) FROM T3_CustomerDetails;

SELECT MAX(cost) FROM T3_PackageDetails;

SELECT MIN(payment_amount) FROM T3_BookingDetails;

```

	(No column name)
1	10

	(No column name)
1	29

	(No column name)
1	90000.00

	(No column name)
1	1500.00

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 4 rows

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

/\*ORDER BY, GROUP BY, HAVING CLAUSE\*/

```
SELECT TOP 10 * FROM T3_CustomerDetails  
ORDER BY age DESC;
```

```
SELECT * FROM T3_BookingDetails  
ORDER BY payment_amount;
```

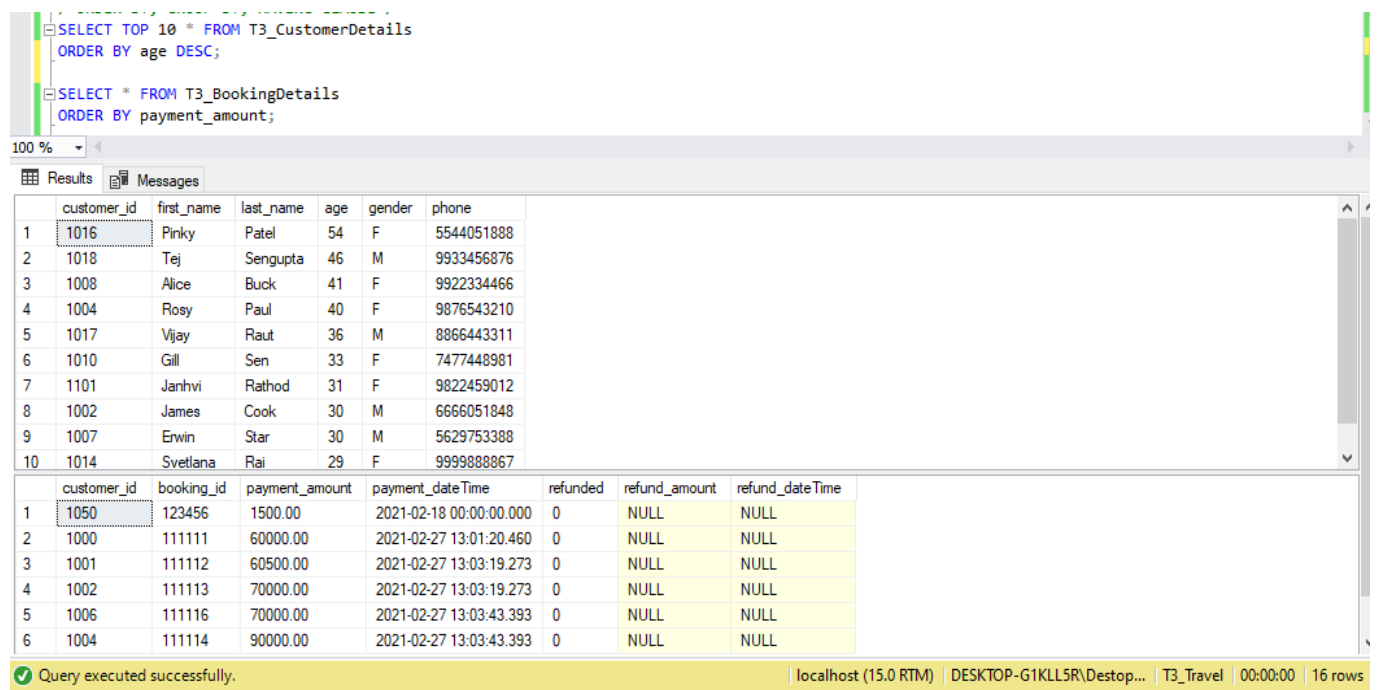
```
SELECT COUNT(*), gender FROM T3_CustomerDetails  
GROUP BY gender;
```

```
SELECT starting_point, AVG(cost) AS AVG_COST FROM T3_PackageDetails  
GROUP BY starting_point ORDER BY AVG_COST;
```

```
SELECT COUNT(*), gender FROM T3_CustomerDetails  
GROUP BY gender  
HAVING COUNT(*)>10;
```

```
SELECT starting_point, AVG(cost) AS AVG_COST FROM T3_PackageDetails  
GROUP BY starting_point  
HAVING starting_point LIKE '_u%'  
ORDER BY AVG_COST;
```

DATABASE OUTPUT:



```
SELECT TOP 10 * FROM T3_CustomerDetails  
ORDER BY age DESC;
```

```
SELECT * FROM T3_BookingDetails  
ORDER BY payment_amount;
```

	customer_id	first_name	last_name	age	gender	phone
1	1016	Pinky	Patel	54	F	5544051888
2	1018	Tej	Sengupta	46	M	9933456876
3	1008	Alice	Buck	41	F	9922334466
4	1004	Rosy	Paul	40	F	9876543210
5	1017	Vijay	Raut	36	M	8866443311
6	1010	Gill	Sen	33	F	7477448981
7	1101	Janhvi	Rathod	31	F	9822459012
8	1002	James	Cook	30	M	6666051848
9	1007	Erwin	Star	30	M	5629753388
10	1014	Svetlana	Rai	29	F	9999888867

	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime
1	1050	123456	1500.00	2021-02-18 00:00:00.000	0	NULL	NULL
2	1000	111111	60000.00	2021-02-27 13:01:20.460	0	NULL	NULL
3	1001	111112	60500.00	2021-02-27 13:03:19.273	0	NULL	NULL
4	1002	111113	70000.00	2021-02-27 13:03:19.273	0	NULL	NULL
5	1006	111116	70000.00	2021-02-27 13:03:43.393	0	NULL	NULL
6	1004	111114	90000.00	2021-02-27 13:03:43.393	0	NULL	NULL

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 16 rows

```
SELECT COUNT(*), gender FROM T3_CustomerDetails
GROUP BY gender;

SELECT starting_point, AVG(cost) AS AVG_COST FROM T3_PackageDetails
GROUP BY starting_point ORDER BY AVG_COST;
```

100 %

Results Messages

	(No column name)	gender
1	13	F
2	10	M

	starting_point	AVG_COST
1	KULLU	60000.000000
2	UP	60500.000000
3	HUBLI	75250.000000

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 5 rows

```
SELECT COUNT(*), gender FROM T3_CustomerDetails
GROUP BY gender
HAVING COUNT(*)>10;

SELECT starting_point, AVG(cost) AS AVG_COST FROM T3_PackageDetails
GROUP BY starting_point
HAVING starting_point LIKE '_u%'
ORDER BY AVG_COST;
```

100 %

Results Messages

	(No column name)	gender
1	13	F

	starting_point	AVG_COST
1	KULLU	60000.000000
2	HUBLI	75250.000000

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 3 rows

#### 4. Use Aggregate function with group by and having

```
/*AGGREGATE FUNCTION WITH GROUP BY AND HAVING*/
SELECT COUNT(*), gender FROM T3_CustomerDetails
GROUP BY gender
HAVING COUNT(*)>10;
```

```
SELECT starting_point, AVG(cost) AS AVG_COST FROM T3_PackageDetails
GROUP BY starting_point
HAVING starting_point LIKE '_u%'
ORDER BY AVG_COST;
```

#### DATABASE OUTPUT:

The screenshot shows a database query editor with two queries. The first query filters customers by gender and count. The second query filters packages by starting point and orders by average cost.

```
/*AGGREGATE FUNCTION WITH GROUP BY AND HAVING*/
SELECT COUNT(*), gender FROM T3_CustomerDetails
GROUP BY gender
HAVING COUNT(*)>10;

SELECT starting_point, AVG(cost) AS AVG_COST FROM T3_PackageDetails
GROUP BY starting_point
HAVING starting_point LIKE '_u%'
ORDER BY AVG_COST;
```

Results:

(No column name)	gender
13	F

starting_point	AVG_COST
KULLU	60000.000000
HUBLI	75250.000000

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 3 rows

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

```
SELECT * FROM T3_CustomerDetails
WHERE customer_id IN (SELECT customer_id FROM T3_BookingDetails)
ORDER BY age;
```

```
SELECT AVG(payment_amount) AS avg_cost, payment_dateTime FROM T3_BookingDetails
WHERE booking_id IN (SELECT booking_id FROM T3_PackageDetails)
GROUP BY payment_dateTime;
```

```
SELECT age, COUNT(*) AS count FROM T3_CustomerDetails
WHERE customer_id IN (SELECT customer_id FROM T3_BookingDetails)
GROUP BY age HAVING age<30
ORDER BY age DESC;
```

#### DATABASE OUTPUT:

The screenshot shows three nested SQL queries. The first query filters customers by booking details and orders by age. The second query calculates average cost and payment date by booking ID. The third query filters customers by age and count.

```
SELECT * FROM T3_CustomerDetails
WHERE customer_id IN (SELECT customer_id FROM T3_BookingDetails)
ORDER BY age;

SELECT AVG(payment_amount) AS avg_cost, payment_dateTime FROM T3_BookingDetails
WHERE booking_id IN (SELECT booking_id FROM T3_PackageDetails)
GROUP BY payment_dateTime;

SELECT age, COUNT(*) AS count FROM T3_CustomerDetails
WHERE customer_id IN (SELECT customer_id FROM T3_BookingDetails)
GROUP BY age HAVING age<30
ORDER BY age DESC;
```

Results:

customer_id	first_name	last_name	age	gender	phone
1000	Manya	Mehta	19	F	7623444591
1006	Susan	Noble	21	F	8887051648
1001	Mark	Stone	23	M	7666051848
1050	Rubina	Iyer	23	F	7666054409
1002	James	Cook	30	M	6666051848
1004	Rosy	Paul	40	F	9876543210

avg_cost	payment_dateTime
60000.000000	2021-02-27 13:01:20.460
65250.000000	2021-02-27 13:03:19.273
90000.000000	2021-02-27 13:03:43.393

age	count
23	2
21	1
19	1

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 12 rows

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

```
SELECT customer_id FROM T3_CustomerDetails  
EXCEPT  
SELECT customer_id FROM T3_BookingDetails;
```

```
SELECT customer_id FROM T3_CustomerDetails  
INTERSECT  
SELECT customer_id FROM T3_BookingDetails;
```

```
SELECT booking_id FROM T3_PackageDetails  
UNION  
SELECT booking_id FROM T3_BookingDetails;
```

```
SELECT first_name, last_name FROM T3_CustomerDetails  
WHERE EXISTS(SELECT customer_id FROM T3_BookingDetails  
WHERE T3_CustomerDetails.customer_id = T3_BookingDetails.customer_id);
```

```
SELECT * FROM T3_BookingDetails  
WHERE NOT EXISTS(SELECT booking_id FROM T3_PackageDetails  
WHERE T3_BookingDetails.booking_id = T3_PackageDetails.booking_id);
```

DATABASE OUTPUT:

The screenshot shows a SQL query window with the following code:

```
SELECT customer_id FROM T3_CustomerDetails  
EXCEPT  
SELECT customer_id FROM T3_BookingDetails;
```

The results pane displays a table with one column, 'customer\_id', containing 17 rows of data. The status bar at the bottom indicates 'Query executed successfully.' and '17 rows'.

customer_id
1007
1008
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1040
1101
1105
1109
1111

The screenshot shows a SQL query window with the following code:

```
SELECT customer_id FROM T3_CustomerDetails  
INTERSECT  
SELECT customer_id FROM T3_BookingDetails;
```

The results pane displays a table with one column, 'customer\_id', containing 6 rows of data. The status bar at the bottom indicates 'Query executed successfully.' and '6 rows'.

customer_id
1000
1001
1002
1004
1006
1050

```

SELECT booking_id FROM T3_PackageDetails
UNION
SELECT booking_id FROM T3_BookingDetails;

```

100 %

	booking_id
1	111111
2	111112
3	111113
4	111114
5	111116
6	123456

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 6 rows

```

SELECT first_name, last_name FROM T3_CustomerDetails
WHERE EXISTS(SELECT customer_id FROM T3_BookingDetails
WHERE T3_CustomerDetails.customer_id = T3_BookingDetails.customer_id);

SELECT * FROM T3_BookingDetails
WHERE NOT EXISTS(SELECT booking_id FROM T3_PackageDetails
WHERE T3_BookingDetails.booking_id = T3_PackageDetails.booking_id);

```

100 %

	first_name	last_name
1	Manya	Mehta
2	Mark	Stone
3	James	Cook
4	Rosy	Paul
5	Susan	Noble
6	Rubina	Iyer

	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime
1	1006	111116	70000.00	2021-02-27 13:03:43.393	0	NULL	NULL
2	1050	123456	1500.00	2021-02-18 00:00:00.000	0	NULL	NULL

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 8 rows

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

/\*INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN\*/

```

SELECT * FROM T3_CustomerDetails C
INNER JOIN T3_BookingDetails B
ON C.customer_id = B.customer_id;

```

```

SELECT customer_id, package_name, package_description FROM T3_PackageDetails P
INNER JOIN T3_BookingDetails B
ON P.booking_id = B.booking_id;

```

```

SELECT first_name, last_name, payment_amount, package_name FROM T3_CustomerDetails C
INNER JOIN T3_BookingDetails B ON C.customer_id = B.customer_id

```



```
INNER JOIN T3_PackageDetails P ON B.booking_id = P.booking_id;
```

```
SELECT TOP 10 * FROM T3_CustomerDetails C
LEFT OUTER JOIN T3_BookingDetails B
ON C.customer_id = B.customer_id;
```

```
SELECT customer_id, package_name, package_description FROM T3_PackageDetails P
LEFT OUTER JOIN T3_BookingDetails B
ON P.booking_id = B.booking_id;
```

```
SELECT TOP 10 first_name, last_name, payment_amount, package_name FROM T3_CustomerDetails C
LEFT OUTER JOIN T3_BookingDetails B ON C.customer_id = B.customer_id
LEFT OUTER JOIN T3_PackageDetails P ON B.booking_id = P.booking_id;
```

```
SELECT * FROM T3_CustomerDetails C
RIGHT OUTER JOIN T3_BookingDetails B
ON C.customer_id = B.customer_id;
```

```
SELECT customer_id, package_name, package_description FROM T3_PackageDetails P
RIGHT OUTER JOIN T3_BookingDetails B
ON P.booking_id = B.booking_id;
```

```
SELECT first_name, last_name, payment_amount, package_name FROM T3_CustomerDetails C
RIGHT OUTER JOIN T3_BookingDetails B ON C.customer_id = B.customer_id
RIGHT OUTER JOIN T3_PackageDetails P ON B.booking_id = P.booking_id;
```

#### DATABASE OUTPUT:

100 %

Results Messages

```

SELECT * FROM T3_CustomerDetails C
INNER JOIN T3_BookingDetails B
ON C.customer_id = B.customer_id;

SELECT customer_id, package_name, package_description FROM T3_PackageDetails P
INNER JOIN T3_BookingDetails B
ON P.booking_id = B.booking_id;

SELECT first_name, last_name, payment_amount, package_name FROM T3_CustomerDetails C
INNER JOIN T3_BookingDetails B ON C.customer_id = B.customer_id
INNER JOIN T3_PackageDetails P ON B.booking_id = P.booking_id;

```

	customer_id	first_name	last_name	age	gender	phone	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime
1	1000	Manya	Mehta	19	F	7623444591	1000	111111	60000.00	2021-02-27 13:01:20.460	0	NULL	NULL
2	1001	Mark	Stone	23	M	7666051848	1001	111112	60500.00	2021-02-27 13:03:19.273	0	NULL	NULL
3	1002	James	Cook	30	M	6666051848	1002	111113	70000.00	2021-02-27 13:03:19.273	0	NULL	NULL
4	1004	Rosy	Paul	40	F	9876543210	1004	111114	90000.00	2021-02-27 13:03:43.393	0	NULL	NULL
5	1006	Susan	Noble	21	F	8887051648	1006	111116	70000.00	2021-02-27 13:03:43.393	0	NULL	NULL
6	1050	Rubina	Iyer	23	F	7666054409	1050	123456	1500.00	2021-02-18 00:00:00.000	0	NULL	NULL

	customer_id	package_name	package_description
1	1002	HARIDWAR	SACRED PLACE
2	1000	KULLU MANALI	KULLU
3	1004	MUNNAR	KERALA
4	1001	OOTY	TAMIL NADU

	first_name	last_name	payment_amount	package_name
1	James	Cook	70000.00	HARIDWAR
2	Manya	Mehta	60000.00	KULLU MANALI

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Despot... | T3\_Travel | 00:00:00 | 14 rows

```

SELECT TOP 10 * FROM T3_CustomerDetails C
LEFT OUTER JOIN T3_BookingDetails B
ON C.customer_id = B.customer_id;

SELECT customer_id, package_name, package_description FROM T3_PackageDetails P
LEFT OUTER JOIN T3_BookingDetails B
ON P.booking_id = B.booking_id;

SELECT TOP 10 first_name, last_name, payment_amount, package_name FROM T3_CustomerDetails C
LEFT OUTER JOIN T3_BookingDetails B ON C.customer_id = B.customer_id
LEFT OUTER JOIN T3_PackageDetails P ON B.booking_id = P.booking_id;

```

	customer_id	first_name	last_name	age	gender	phone	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime
1	1000	Manya	Mehta	19	F	7623444591	1000	111111	60000.00	2021-02-27 13:01:20.460	0	NULL	NULL
2	1001	Mark	Stone	23	M	7666051848	1001	111112	60500.00	2021-02-27 13:03:19.273	0	NULL	NULL
3	1002	James	Cook	30	M	6666051848	1002	111113	70000.00	2021-02-27 13:03:19.273	0	NULL	NULL
4	1004	Rosy	Paul	40	F	9876543210	1004	111114	90000.00	2021-02-27 13:03:43.393	0	NULL	NULL
5	1006	Susan	Noble	21	F	8887051648	1006	111116	70000.00	2021-02-27 13:03:43.393	0	NULL	NULL
6	1007	Erwin	Star	30	M	5629753388	NULL	NULL	NULL	NULL	NULL	NULL	NULL
7	1008	Alice	Buck	41	F	9922334466	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	1010	Gill	Sen	33	F	7477448981	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9	1011	Mona	Azuma	21	F	8899881234	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10	1012	Hanna	Khan	24	F	6666115848	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	customer_id	package_name	package_description
1	1002	HARIDWAR	SACRED PLACE
2	1000	KULLU MANALI	KULLU
3	1004	MUNNAR	KERALA
4	1001	OOTY	TAMIL NADU

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 24 rows

	first_name	last_name	payment_amount	package_name
1	Manya	Mehta	60000.00	KULLU MANALI
2	Mark	Stone	60500.00	OOTY
3	James	Cook	70000.00	HARIDWAR
4	Rosy	Paul	90000.00	MUNNAR
5	Susan	Noble	70000.00	NULL
6	Erwin	Star	NULL	NULL
7	Alice	Buck	NULL	NULL
8	Gill	Sen	NULL	NULL
9	Mona	Azuma	NULL	NULL
10	Hanna	Khan	NULL	NULL

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 24 rows

```

SELECT * FROM T3_CustomerDetails C
RIGHT OUTER JOIN T3_BookingDetails B
ON C.customer_id = B.customer_id;

SELECT customer_id, package_name, package_description FROM T3_PackageDetails P
RIGHT OUTER JOIN T3_BookingDetails B
ON P.booking_id = B.booking_id;

SELECT first_name, last_name, payment_amount, package_name FROM T3_CustomerDetails C
RIGHT OUTER JOIN T3_BookingDetails B ON C.customer_id = B.customer_id
RIGHT OUTER JOIN T3_PackageDetails P ON B.booking_id = P.booking_id;

```

	customer_id	first_name	last_name	age	gender	phone	customer_id	booking_id	payment_amount	payment_dateTime	refunded	refund_amount	refund_dateTime
1	1000	Manya	Mehta	19	F	7623444591	1000	111111	60000.00	2021-02-27 13:01:20.460	0	NULL	NULL
2	1001	Mark	Stone	23	M	7666051848	1001	111112	60500.00	2021-02-27 13:03:19.273	0	NULL	NULL
3	1002	James	Cook	30	M	6666051848	1002	111113	70000.00	2021-02-27 13:03:19.273	0	NULL	NULL
4	1004	Rosy	Paul	40	F	9876543210	1004	111114	90000.00	2021-02-27 13:03:43.393	0	NULL	NULL
5	1006	Susan	Noble	21	F	8887051648	1006	111116	70000.00	2021-02-27 13:03:43.393	0	NULL	NULL
6	1050	Rubina	Iyer	23	F	7666054409	1050	123456	1500.00	2021-02-18 00:00:00.000	0	NULL	NULL

	customer_id	package_name	package_description
1	1000	KULLU MANALI	KULLU
2	1001	OOTY	TAMIL NADU
3	1002	HARIDWAR	SACRED PLACE
4	1004	MUNNAR	KERALA
5	1006	NULL	NULL
6	1050	NULL	NULL

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 16 rows

	first_name	last_name	payment_amount	package_name
1	James	Cook	70000.00	HARIDWAR
2	Manya	Mehta	60000.00	KULLU MANALI
3	Rosy	Paul	90000.00	MUNNAR
4	Mark	Stone	60500.00	OOTY

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 16 rows

8. Use all the above condition in JOIN as well.

```
SELECT customer_id, package_name, package_description FROM T3_PackageDetails P
INNER JOIN T3_BookingDetails B
ON P.booking_id = B.booking_id AND P.cost>60000
ORDER BY 1;
```

```
SELECT COUNT(*), age FROM T3_CustomerDetails C
LEFT OUTER JOIN T3_BookingDetails B
ON C.customer_id = B.customer_id
GROUP BY age HAVING age<30;
```

```
SELECT first_name, last_name, payment_amount, package_name FROM T3_CustomerDetails C
RIGHT OUTER JOIN T3_BookingDetails B ON C.customer_id = B.customer_id
RIGHT OUTER JOIN T3_PackageDetails P ON B.booking_id = P.booking_id
ORDER BY payment_amount;
```

#### DATABASE OUTPUT:

```
SELECT customer_id, package_name, package_description FROM T3_PackageDetails P
INNER JOIN T3_BookingDetails B
ON P.booking_id = B.booking_id AND P.cost>60000
ORDER BY 1;
```

```
SELECT COUNT(*), age FROM T3_CustomerDetails C
LEFT OUTER JOIN T3_BookingDetails B
ON C.customer_id = B.customer_id
GROUP BY age HAVING age<30;
```

```
SELECT first_name, last_name, payment_amount, package_name FROM T3_CustomerDetails C
RIGHT OUTER JOIN T3_BookingDetails B ON C.customer_id = B.customer_id
RIGHT OUTER JOIN T3_PackageDetails P ON B.booking_id = P.booking_id
ORDER BY payment_amount;
```

100 %

Results Messages

	customer_id	package_name	package_description
1	1001	OOTY	TAMIL NADU
2	1002	HARIDWAR	SACRED PLACE
3	1004	MUNNAR	KERALA

	(No column name)	age
1	1	19
2	2	21
3	1	22
4	4	23
5	3	24
6	1	26
7	1	28
8	1	29

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 15 rows

	first_name	last_name	payment_amount	package_name
1	Manya	Mehta	60000.00	KULLU MANALI
2	Mark	Stone	60500.00	OOTY
3	James	Cook	70000.00	HARIDWAR
4	Rosy	Paul	90000.00	MUNNAR

Query executed successfully. | localhost (15.0 RTM) | DESKTOP-G1KLL5R\Destop... | T3\_Travel | 00:00:00 | 15 rows