

DBMS LAB ASSIGNMENT-5

Name: - B.Ieeshasree

Roll No. :-19BCS021.

Database: - Travel Agency

1. Illustrate logical ANY, ALL and LIKE operators

ANY

QUERY 1:-

```
SELECT *  
FROM T10_customerdetailsNEW  
WHERE customer_name < ANY (SELECT customer_name  
                             FROM T10_customerdetailsNEW  
                             WHERE customer_id < '6'  
                             )
```

OUTPUT:

	hotel_id	hotel_name	hotel_type	hotel_rent	hotel_description	hotel_address
1	101	chakri	single	200	NULL	vizag
2	102	Manasa	single	300	NULL	kovur

QUERY 2:-

```
SELECT *  
FROM T10_hoteldetails  
WHERE hotel_id < ANY (SELECT hotel_id  
                       FROM T10_hoteldetails  
                       WHERE hotel_rent=100  
                       );
```

OUTPUT:-



```
SELECT *
FROM T10_hoteldetails
WHERE hotel_id < ANY (SELECT hotel_id
FROM T10_hoteldetails
WHERE hotel_rent=100
);
```

100 %

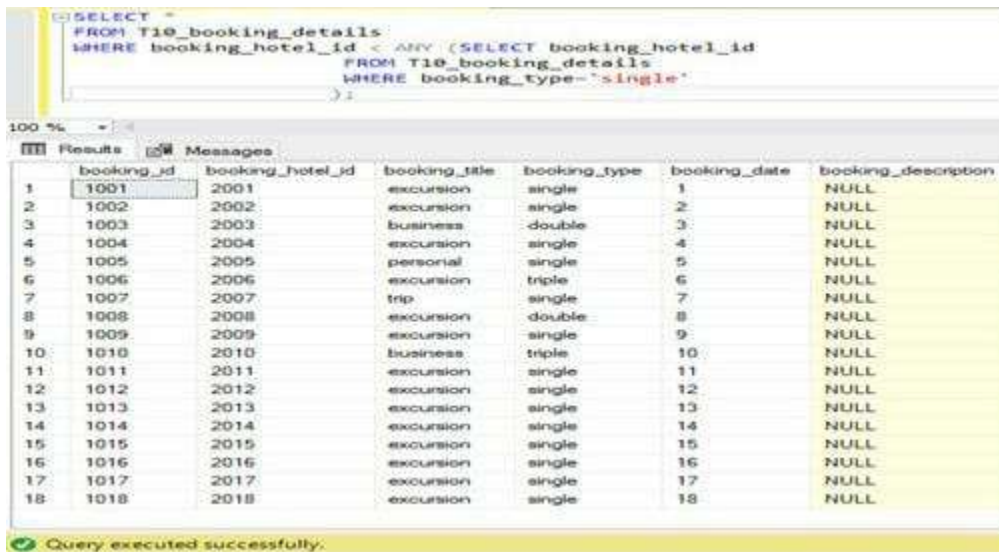
Results Messages

	hotel_id	hotel_name	hotel_type	hotel_rent	hotel_description	hotel_address
1	101	chakri	single	200	NULL	vizag
2	102	Manasa	single	300	NULL	kovur

QUERY 3:-

```
SELECT *
FROM T10_booking_details
WHERE booking_hotel_id < ANY (SELECT booking_hotel_id
FROM T10_booking_details
WHERE booking_type='single'
);
```

OUTPUT:



```
SELECT *
FROM T10_booking_details
WHERE booking_hotel_id < ANY (SELECT booking_hotel_id
FROM T10_booking_details
WHERE booking_type='single'
);
```

100 %

Results Messages

	booking_id	booking_hotel_id	booking_title	booking_type	booking_date	booking_description
1	1001	2001	excursion	single	1	NULL
2	1002	2002	excursion	single	2	NULL
3	1003	2003	business	double	3	NULL
4	1004	2004	excursion	single	4	NULL
5	1005	2005	personal	single	5	NULL
6	1006	2006	excursion	triple	6	NULL
7	1007	2007	trip	single	7	NULL
8	1008	2008	excursion	double	8	NULL
9	1009	2009	excursion	single	9	NULL
10	1010	2010	business	triple	10	NULL
11	1011	2011	excursion	single	11	NULL
12	1012	2012	excursion	single	12	NULL
13	1013	2013	excursion	single	13	NULL
14	1014	2014	excursion	single	14	NULL
15	1015	2015	excursion	single	15	NULL
16	1016	2016	excursion	single	16	NULL
17	1017	2017	excursion	single	17	NULL
18	1018	2018	excursion	single	18	NULL

Query executed successfully.

ALL

QUERY 1:-

```
SELECT *
FROM T10_customerdetailsNEW
WHERE customer_name > ALL (SELECT customer_name
FROM T10_customerdetailsNEW
WHERE customer_id < '6'
);
```

OUTPUT:

```
SELECT *
FROM T10_customerdetailsNEW
WHERE customer_name > ALL (SELECT customer_name
FROM T10_customerdetailsNEW
WHERE customer_id < '6'
);
```

100 %

Results Messages

	customer_id	customer_name	customer_mobile	customer_email	customer_username	customer_password	customer_address	customer_final
1	10	SRISTI	83444	nika	newstat	iknow	hubli	NULL
2	11	SRUTI	85687	seensha	nestat	feeku	dhawwad	NULL
3	15	ROYA	85667	nisha	nesti	dontno	delhi	NULL
4	18	SIRI	86674	rushi	newstar	newsiri	ongol	NULL
5	19	SEEMA	87774	sushma	freshlife	newseema	palem	NULL
6	20	VAISHNAVI	89874	aisha	freshstat	newvaish	banglore	NULL

QUERY 2:-

```
SELECT *
FROM T10_hoteldetails
WHERE hotel_id > ALL (SELECT hotel_id
FROM T10_hoteldetails
WHERE hotel_rent=200
);
```

OUTPUT:

```
SELECT *
FROM T10_hoteldetails
WHERE hotel_id > ALL (SELECT hotel_id
FROM T10_hoteldetails
WHERE hotel_rent=200
);
```

100 %

Results Messages

	hotel_id	hotel_name	hotel_type	hotel_rent	hotel_description	hotel_address
1	105	mehraj	single	250	NULL	kadapa

QUERY 3:-

```
SELECT *
FROM T10_booking_details
WHERE booking_hotel_id > ALL (SELECT booking_hotel_id
FROM T10_booking_details
WHERE booking_type='double'
);
```

OUTPUT:

```
SELECT *
FROM T10_booking_details
WHERE booking_hotel_id > ALL (SELECT booking_hotel_id
FROM T10_booking_details
WHERE booking_type='double'
);
```

	booking_id	booking_hotel_id	booking_title	booking_type	booking_date	booking_description
1	1009	2009	excursion	single	9	NULL
2	1010	2010	business	triple	10	NULL
3	1011	2011	excursion	single	11	NULL
4	1012	2012	excursion	single	12	NULL
5	1013	2013	excursion	single	13	NULL
6	1014	2014	excursion	single	14	NULL
7	1015	2015	excursion	single	15	NULL
8	1016	2016	excursion	single	16	NULL
9	1017	2017	excursion	single	17	NULL
10	1018	2018	excursion	single	18	NULL
11	1019	2019	excursion	single	19	NULL

LIKE

QUERY 1:-

```
SELECT *
FROM T10_customerdetailsNEW
WHERE customer_name LIKE 'C%';
```

OUTPUT:

```
SELECT *
FROM T10_customerdetailsNEW
WHERE customer_name LIKE 'C%';
```

	customer_id	customer_name	customer_mobile	customer_email	customer_username	customer_password	customer_address	customer_final
1	1	CHAITRA	82096	chaitra@gmail.com	msaichaitra	little	kadapa	NULL
2	2	CHAITRA	80825	chaitra	chaitrareddy	msaichaitra	kadapa	NULL

QUERY 2:-

```
SELECT *
FROM T10_hoteldetails
WHERE hotel_rent=200 AND hotel_type LIKE '__n%';
```

OUTPUT:



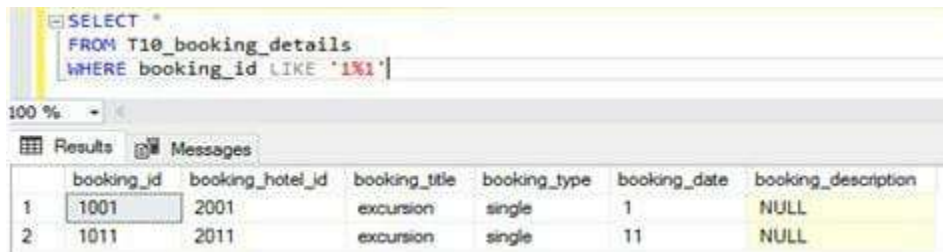
The screenshot shows a SQL query in a text editor: `SELECT * FROM T10_hoteldetails WHERE hotel_rent=200 AND hotel_type LIKE ' _n%';`. Below the query, the 'Results' tab is active, displaying a table with 6 columns: hotel_id, hotel_name, hotel_type, hotel_rent, hotel_description, and hotel_address. Two rows are shown: row 1 with hotel_id 101, hotel_name 'chakri', hotel_type 'single', hotel_rent 200, hotel_description NULL, and hotel_address 'vizag'; row 2 with hotel_id 104, hotel_name 'Maharshi', hotel_type 'single', hotel_rent 200, hotel_description NULL, and hotel_address 'kakinada'.

	hotel_id	hotel_name	hotel_type	hotel_rent	hotel_description	hotel_address
1	101	chakri	single	200	NULL	vizag
2	104	Maharshi	single	200	NULL	kakinada

QUERY 3:

```
SELECT *  
FROM T10_booking_details  
WHERE booking_id LIKE '1%1'
```

OUTPUT:



The screenshot shows a SQL query in a text editor: `SELECT * FROM T10_booking_details WHERE booking_id LIKE '1%1';`. Below the query, the 'Results' tab is active, displaying a table with 7 columns: booking_id, booking_hotel_id, booking_title, booking_type, booking_date, and booking_description. Two rows are shown: row 1 with booking_id 1001, booking_hotel_id 2001, booking_title 'excursion', booking_type 'single', booking_date 1, and booking_description NULL; row 2 with booking_id 1011, booking_hotel_id 2011, booking_title 'excursion', booking_type 'single', booking_date 11, and booking_description NULL.

	booking_id	booking_hotel_id	booking_title	booking_type	booking_date	booking_description
1	1001	2001	excursion	single	1	NULL
2	1011	2011	excursion	single	11	NULL

2. One query for each Aggregate function.

AVG() QUERY:-

```
SELECT AVG(hotel_rent)  
FROM T10_hoteldetails
```

OUTPUT:



The screenshot shows a SQL query in a text editor: `SELECT AVG(hotel_rent) FROM T10_hoteldetails;`. Below the query, the 'Results' tab is active, displaying a single row with the value 210. The column name is '(No column name)'.

	(No column name)
1	210

COUNT() QUERY:-

```
SELECT COUNT(*)  
FROM T10_hoteldetails  
WHERE hotel_type='single'
```

OUTPUT:



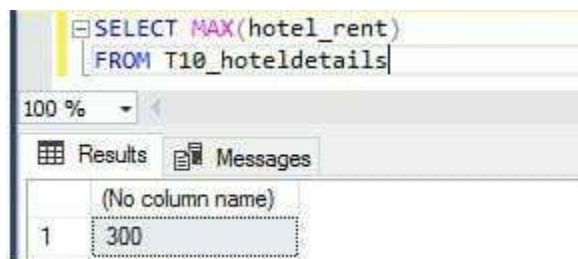
The screenshot shows a SQL query editor with the query: `SELECT COUNT(*) FROM T10_hoteldetails WHERE hotel_type='single'`. Below the editor, the 'Results' tab is active, displaying a single row with the value 4.

	(No column name)
1	4

MAX() QUERY:-

```
SELECT MAX(hotel_rent)  
FROM T10_hoteldetails
```

OUTPUT:



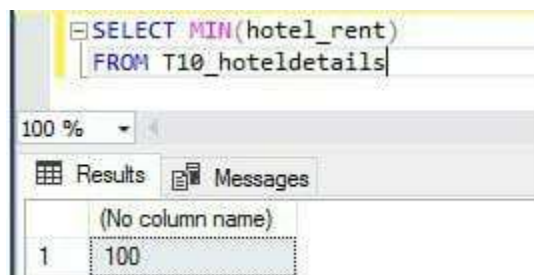
The screenshot shows a SQL query editor with the query: `SELECT MAX(hotel_rent) FROM T10_hoteldetails`. Below the editor, the 'Results' tab is active, displaying a single row with the value 300.

	(No column name)
1	300

MIN() QUERY:-

```
SELECT MIN(hotel_rent)  
FROM T10_hoteldetails
```

OUTPUT:



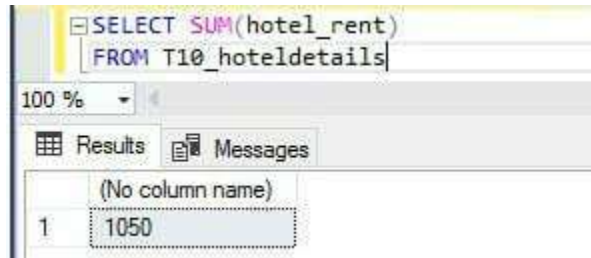
The screenshot shows a SQL query editor with the query: `SELECT MIN(hotel_rent) FROM T10_hoteldetails`. Below the editor, the 'Results' tab is active, displaying a single row with the value 100.

	(No column name)
1	100

SUM() QUERY:-

```
SELECT SUM(hotel_rent)
FROM T10_hoteldetails
```

OUTPUT:-



100 %

Results Messages

(No column name)

1	1050
---	------


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

ORDER BY

QUERY 1:-

```
SELECT * FROM T10_customerdetailsNEW
ORDER BY customer_name ASC;
```

OUTPUT:-



100 %

Results Messages

customer_id	customer_name	customer_mobile	customer_email	customer_username	customer_password	customer_address	customer_final
7	AMRUTHA	8082	amrutha	takenest	meamrutha	vizag	NULL
1	CHAITRA	82096	chaitra@gmail.com	msachaitra	lile	kadapa	NULL
2	CHAITRA	80825	chaitra	chaitrareddy	msachaitra	kadapa	NULL
6	DEEPTI	9083	deepti	life	rightdeepti	belgum	NULL
16	JIYA	84493	shrija	nstat	newparent	bombay	NULL
13	KRITHIKA	83674	grahni	art	livetoful	sangan	NULL
21	LAHARI	95825	lory	reddy	lily	kadapa	NULL
2005	LATHA	85797	luni	etcsart	ufeli	lalubi	NULL
3	MANASA	9127	manasa	hello	hopefulife	kovur	NULL
9	PRACHI	8523	yashh	examalet	prachi	goa	NULL
8	PRAGYA	85654	hemna	raykahani	newpragya	manglore	NULL
12	PREEETA	83444	fazila	neart	makeme	araku	NULL
14	PRIYA	84321	nikitha	newrtt	urgud	buchi	NULL
103	RAGHU	23333	harsh	nshi	raghurise	kokata	NULL
4	RITHIKA	989787	rthika	myself	creativeone	madura	NULL
17	RIYA	85554	nisha	net	lifeone	dargling	NULL
5	ROJA	85674	roja	newstart	newroja	ongole	NULL
15	ROYA	85687	nisha	neatt	dontno	dehi	NULL
19	SEEMA	87774	sushma	freshlife	newseema	palem	NULL
18	SIRI	88674	ruthi	newstar	newari	ongol	NULL
10	SRIISTI	83444	nika	newstat	know	hubli	NULL
11	SRUTI	85687	seensha	newstat	feelu	dhanwad	NULL
2001	SUNITHA	85387	suni	netstat	ufeli	hubli	NULL

Query executed successfully.

CHAITRA (15.0 RTM)

QUERY 2:-

```
SELECT * FROM T10_hoteldetails  
ORDER BY hotel_id DESC;
```

OUTPUT:



The screenshot shows a SQL query editor with the query: `SELECT * FROM T10_hoteldetails ORDER BY hotel_id DESC;`. Below the query, the 'Results' tab is active, displaying a table with 7 columns: `hotel_id`, `hotel_name`, `hotel_type`, `hotel_rent`, `hotel_description`, and `hotel_address`. The results are ordered by `hotel_id` in descending order.

	hotel_id	hotel_name	hotel_type	hotel_rent	hotel_description	hotel_address
1	105	mehraj	single	250	NULL	kadapa
2	104	Maharshi	single	200	NULL	kakinada
3	103	snhari	double	100	NULL	kumool
4	102	Manasa	single	300	NULL	kovur
5	101	chakri	single	200	NULL	vizag

GROUP BY

QUERY 1:

```
SELECT hotel_name, COUNT(*)  
FROM T10_hoteldetails  
WHERE hotel_rent>150  
GROUP BY hotel_name
```

OUTPUT:



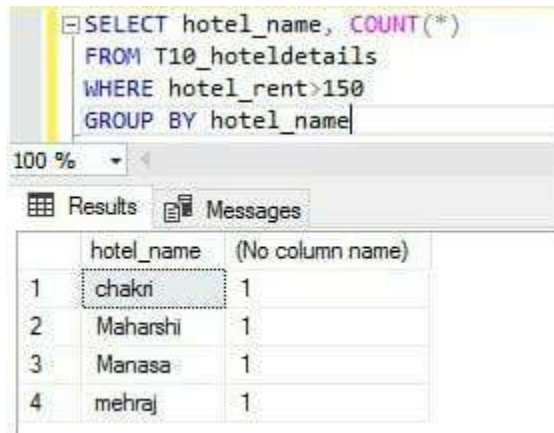
The screenshot shows a SQL query editor with the query: `SELECT hotel_name, COUNT(*) FROM T10_hoteldetails WHERE hotel_rent>150 GROUP BY hotel_name`. Below the query, the 'Results' tab is active, displaying a table with 3 columns: `hotel_name`, `(No column name)`, and `COUNT(*)`. The results are grouped by `hotel_name`.

	hotel_name	(No column name)
1	chakri	1
2	Maharshi	1
3	Manasa	1
4	mehraj	1

QUERY 2:-

```
SELECT customer_name, COUNT(*)  
FROM T10_customerdetailsNEW  
GROUP BY customer_name
```

OUTPUT:



The screenshot shows a SQL query in a text editor and its results in a table. The query is: `SELECT hotel_name, COUNT(*) FROM T10_hoteldetails WHERE hotel_rent > 150 GROUP BY hotel_name`. The results table has two columns: 'hotel_name' and 'COUNT(*)'. It contains four rows of data.

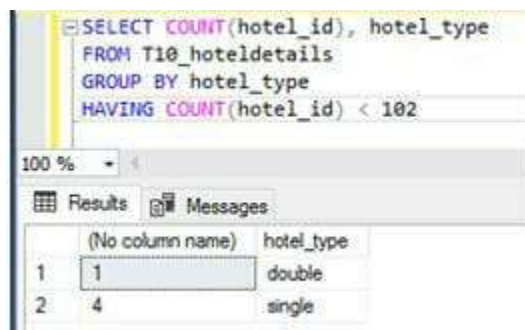
	hotel_name	(No column name)
1	chakri	1
2	Maharshi	1
3	Manasa	1
4	mehraj	1

HAVING CLAUSE

QUERY 1:-

```
SELECT COUNT(hotel_id), hotel_type  
FROM T10_hoteldetails  
GROUP BY hotel_type  
HAVING COUNT(hotel_id) < 102
```

OUTPUT:



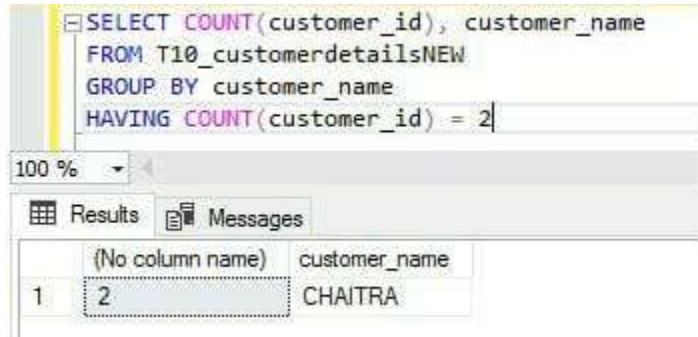
The screenshot shows a SQL query in a text editor and its results in a table. The query is: `SELECT COUNT(hotel_id), hotel_type FROM T10_hoteldetails GROUP BY hotel_type HAVING COUNT(hotel_id) < 102`. The results table has two columns: 'COUNT(hotel_id)' and 'hotel_type'. It contains two rows of data.

	(No column name)	hotel_type
1	1	double
2	4	single

QUERY 2:-

```
SELECT COUNT(customer_id), customer_name  
FROM T10_customerdetailsNEW  
GROUP BY customer_name  
HAVING COUNT(customer_id) = 2
```

OUTPUT:



```
SELECT COUNT(customer_id), customer_name
FROM T10_customerdetailsNEW
GROUP BY customer_name
HAVING COUNT(customer_id) = 2
```

	(No column name)	customer_name
1	2	CHAITRA

4. Use Aggregate function with group by and having AVG() QUERY:-

```
SELECT AVG(hotel_rent)
FROM T10_hoteldetails
GROUP BY hotel_name
HAVING hotel_name = 'manasa'
```

OUTPUT:-



```
SELECT AVG(hotel_rent)
FROM T10_hoteldetails
GROUP BY hotel_name
HAVING hotel_name = 'manasa'
```

	(No column name)
1	300

COUNT() QUERY:-

```
SELECT COUNT(hotel_id)
FROM T10_hoteldetails
GROUP BY hotel_rent
HAVING hotel_rent = 200
```

OUTPUT:



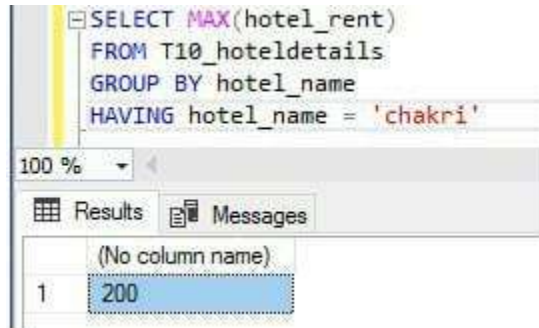
```
SELECT COUNT(hotel_id)
FROM T10_hoteldetails
GROUP BY hotel_rent
HAVING hotel_rent = 200
```

	(No column name)
1	2

MAX() QUERY:-

```
SELECT MAX(hotel_rent)
FROM T10_hoteldetails
GROUP BY hotel_name
HAVING hotel_name = 'chakri'
```

OUTPUT:



The screenshot shows a SQL query editor with the following query:

```
SELECT MAX(hotel_rent)
FROM T10_hoteldetails
GROUP BY hotel_name
HAVING hotel_name = 'chakri'
```

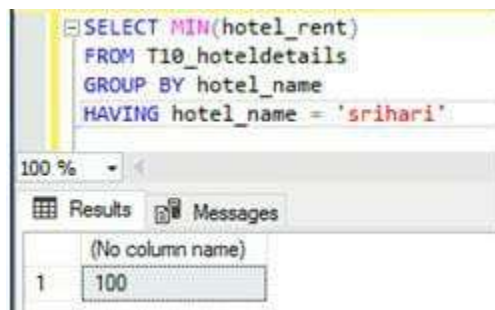
Below the query editor, the 'Results' tab is selected, showing a single row with the value 200.

(No column name)
200

MIN() QUERY:-

```
SELECT MIN(hotel_rent)
FROM T10_hoteldetails
GROUP BY hotel_name
HAVING hotel_name = 'srihari'
```

OUTPUT:



The screenshot shows a SQL query editor with the following query:

```
SELECT MIN(hotel_rent)
FROM T10_hoteldetails
GROUP BY hotel_name
HAVING hotel_name = 'srihari'
```

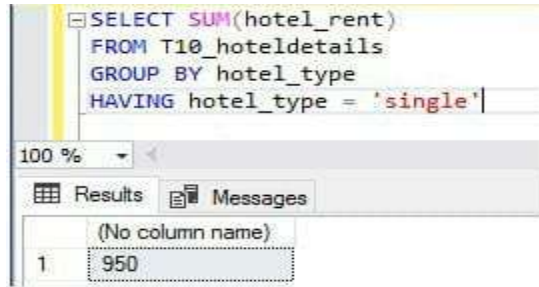
Below the query editor, the 'Results' tab is selected, showing a single row with the value 100.

(No column name)
100

SUM() QUERY:-

```
SELECT SUM(hotel_rent)
FROM T10_hoteldetails
GROUP BY hotel_type
HAVING hotel_type = 'single'
```

OUTPUT:



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

```
SELECT SUM(hotel_rent)
FROM T10_hoteldetails
GROUP BY hotel_type
HAVING hotel_type = 'single'
```

Below the query, there are tabs for "Results" and "Messages". The "Results" tab is active, showing a table with one column labeled "(No column name)" and one row with the value "950".

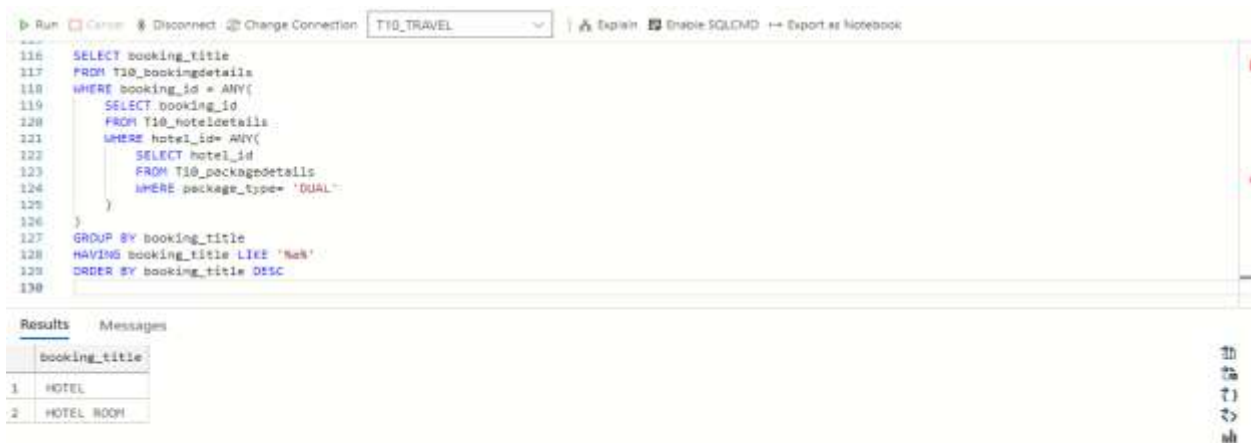
(No column name)
950

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

QUERY:-

```
SELECT booking_title
FROM T10_bookingdetails
WHERE booking_id = ANY(
    SELECT booking_id
    FROM T10_hoteldetails
    WHERE hotel_id= ANY(
        SELECT hotel_id
        FROM T10_packagedetails
        WHERE package_type= 'DUAL'
    )
)
GROUP BY booking_title
HAVING booking_title LIKE '%a%'
ORDER BY booking_title DESC
```

OUTPUT:-



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

```
SELECT booking_title
FROM T10_bookingdetails
WHERE booking_id = ANY(
    SELECT booking_id
    FROM T10_hoteldetails
    WHERE hotel_id= ANY(
        SELECT hotel_id
        FROM T10_packagedetails
        WHERE package_type= 'DUAL'
    )
)
GROUP BY booking_title
HAVING booking_title LIKE 'hak'
ORDER BY booking_title DESC
```

Below the query, there are tabs for "Results" and "Messages". The "Results" tab is active, showing a table with one column labeled "booking_title" and two rows: "HOTEL" and "HOTEL ROOM".

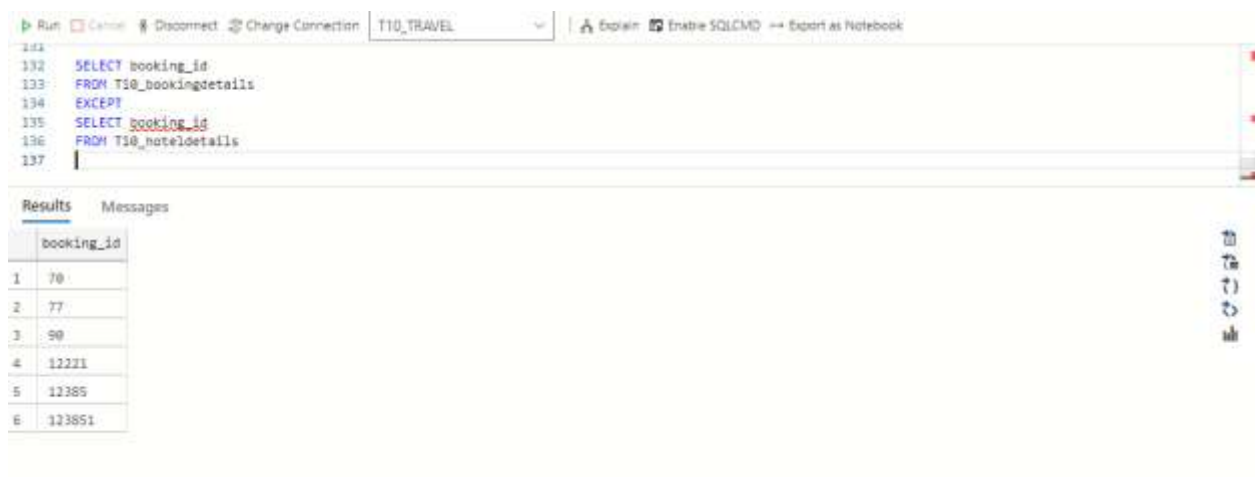
booking_title
1 HOTEL
2 HOTEL ROOM

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

EXCEPT QUERY:-

```
SELECT booking_id
FROM T10_bookingdetails
EXCEPT
SELECT booking_id
FROM T10_hoteldetails
```

OUTPUT:-



The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
132 SELECT booking_id
133 FROM T10_bookingdetails
134 EXCEPT
135 SELECT booking_id
136 FROM T10_hoteldetails
137
```

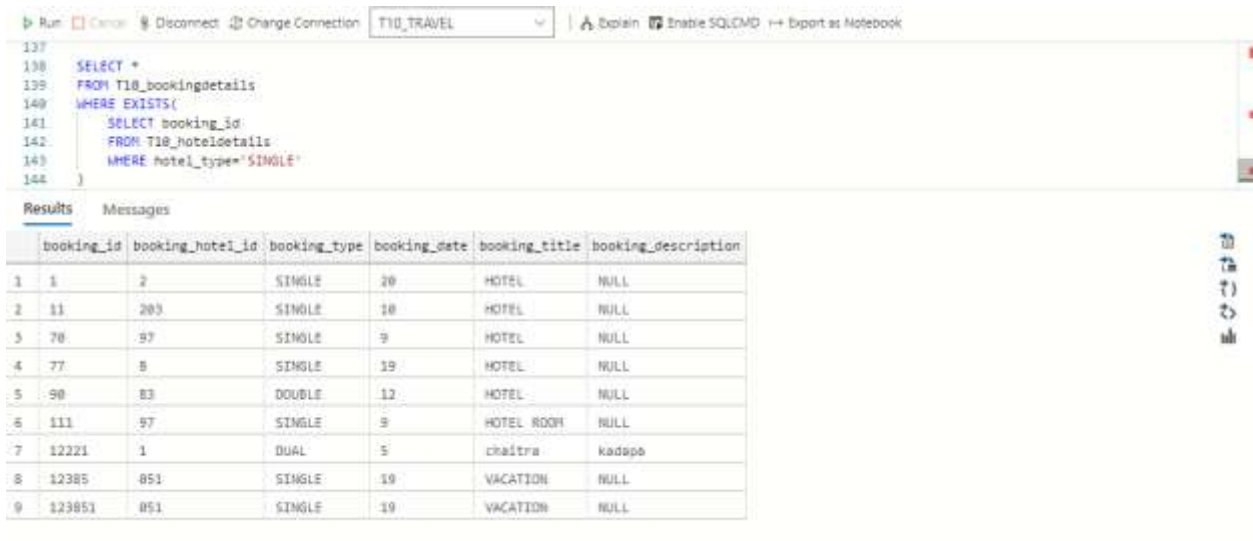
The results pane displays the output of the query, which is a table with one column, 'booking_id', and six rows of data:

booking_id
70
77
90
12221
12385
123851

EXISTS QUERY:-

```
SELECT *
FROM T10_bookingdetails
WHERE EXISTS(
    SELECT booking_id
    FROM T10_hoteldetails
    WHERE hotel_type='SINGLE'
)
```

OUTPUT:-



The screenshot shows a SQL query execution interface. The query is as follows:

```
137 SELECT *
138 FROM T10_bookingdetails
139 WHERE EXISTS(
140     SELECT booking_id
141     FROM T10_hoteldetails
142     WHERE hotel_type='SINGLE'
143 )
144
```

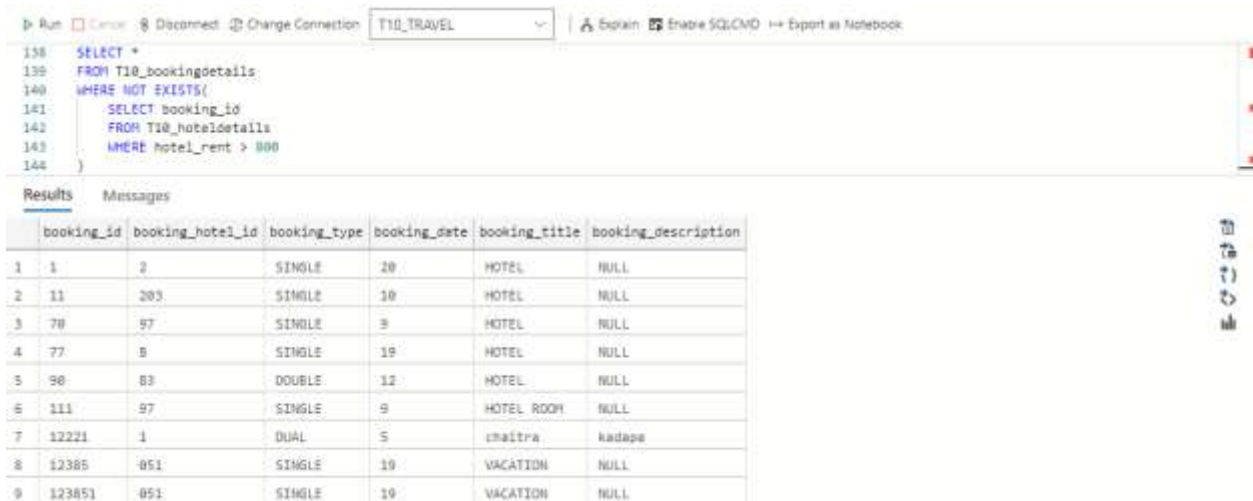
The results are displayed in a table with the following columns: booking_id, booking_hotel_id, booking_type, booking_date, booking_title, and booking_description. The table contains 9 rows of data.

	booking_id	booking_hotel_id	booking_type	booking_date	booking_title	booking_description
1	1	2	SINGLE	20	HOTEL	NULL
2	11	203	SINGLE	10	HOTEL	NULL
3	70	97	SINGLE	9	HOTEL	NULL
4	77	8	SINGLE	19	HOTEL	NULL
5	90	83	DOUBLE	12	HOTEL	NULL
6	111	97	SINGLE	9	HOTEL ROOM	NULL
7	12221	1	DUAL	5	chaitra	kadapa
8	12385	051	SINGLE	19	VACATION	NULL
9	123851	051	SINGLE	19	VACATION	NULL

NOT EXISTS QUERY:-

```
SELECT *
FROM T10_bookingdetails
WHERE NOT EXISTS(
    SELECT booking_id
    FROM T10_hoteldetails
    WHERE hotel_rent > 800
)
```

OUTPUT:



The screenshot shows a SQL query execution interface. The query is as follows:

```
138 SELECT *
139 FROM T10_bookingdetails
140 WHERE NOT EXISTS(
141     SELECT booking_id
142     FROM T10_hoteldetails
143     WHERE hotel_rent > 800
144 )

```

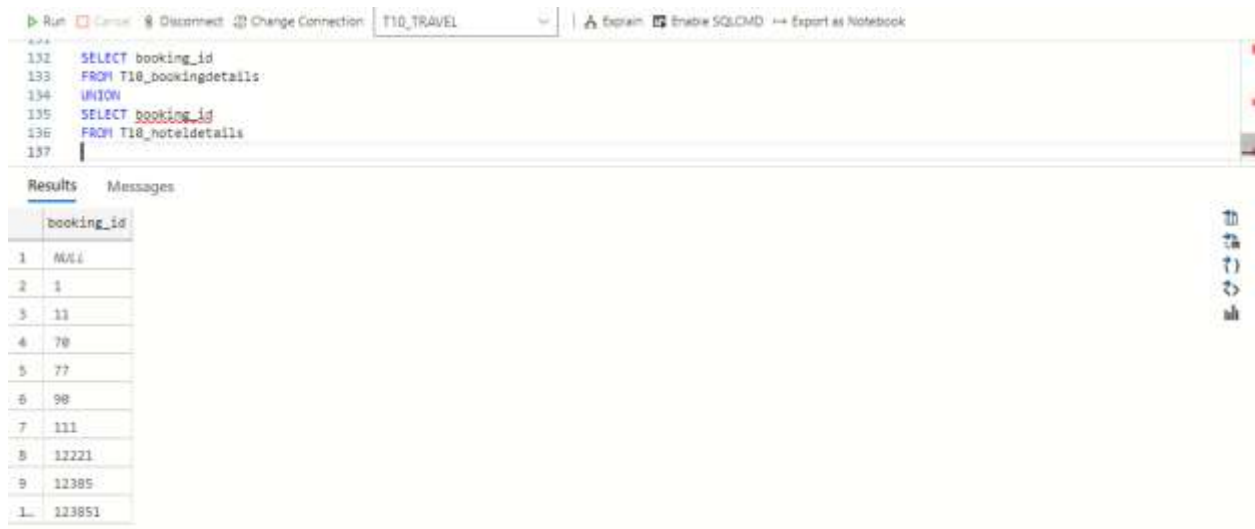
The results are displayed in a table with the following columns: booking_id, booking_hotel_id, booking_type, booking_date, booking_title, and booking_description. The table contains 9 rows of data.

	booking_id	booking_hotel_id	booking_type	booking_date	booking_title	booking_description
1	1	2	SINGLE	20	HOTEL	NULL
2	11	203	SINGLE	10	HOTEL	NULL
3	70	97	SINGLE	9	HOTEL	NULL
4	77	8	SINGLE	19	HOTEL	NULL
5	90	83	DOUBLE	12	HOTEL	NULL
6	111	97	SINGLE	9	HOTEL ROOM	NULL
7	12221	1	DUAL	5	chaitra	kadapa
8	12385	051	SINGLE	19	VACATION	NULL
9	123851	051	SINGLE	19	VACATION	NULL

UNION QUERY:-

```
SELECT booking_id  
FROM T10_bookingdetails  
UNION  
SELECT booking_id  
FROM T10_hoteldetails
```

OUTPUT:



The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
132 SELECT booking_id  
133 FROM T10_bookingdetails  
134 UNION  
135 SELECT booking_id  
136 FROM T10_hoteldetails  
137
```

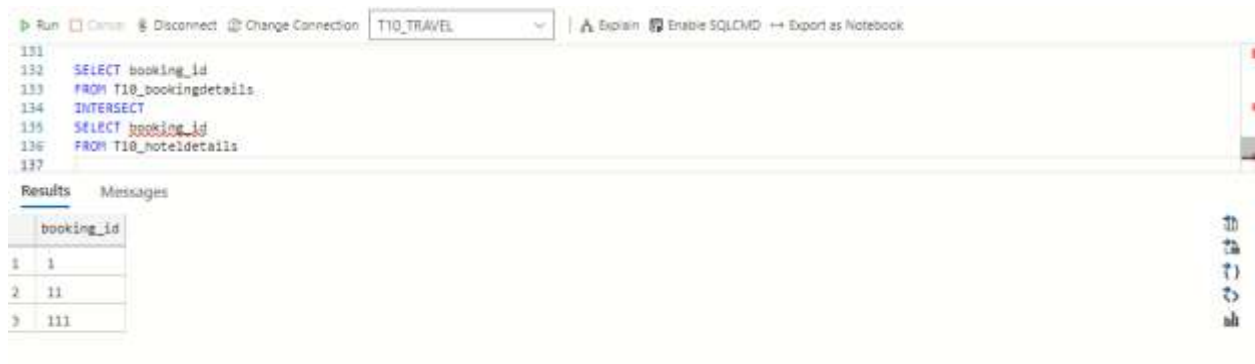
The results pane displays the output of the query, which is a table with one column, 'booking_id'. The table contains 10 rows of data:

booking_id
1
1
11
70
77
90
111
12221
12385
123851

INTERSECTION QUERY:-

```
SELECT booking_id  
FROM T10_bookingdetails  
INTERSECT  
SELECT booking_id  
FROM T10_hoteldetails
```

OUTPUT:



The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
131  
132 SELECT booking_id  
133 FROM T10_bookingdetails  
134 INTERSECT  
135 SELECT booking_id  
136 FROM T10_hoteldetails  
137
```

The results pane displays the output of the query, which is a table with one column, 'booking_id'. The table contains 3 rows of data:

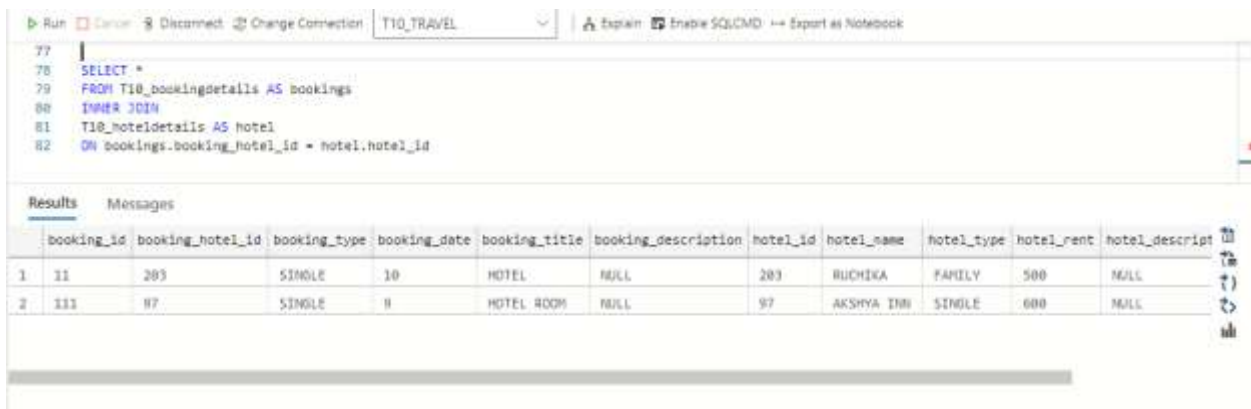
booking_id
1
11
111

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

INNER JOIN QUERY 1:-

```
SELECT *
FROM T10_bookingdetails AS bookings
INNER JOIN
T10_hoteldetails AS hotel
ON bookings.booking_hotel_id = hotel.hotel_id
```

OUTPUT:-



The screenshot shows a SQL query execution interface with the following query:

```
77
78 SELECT *
79 FROM T10_bookingdetails AS bookings
80 INNER JOIN
81 T10_hoteldetails AS hotel
82 ON bookings.booking_hotel_id = hotel.hotel_id
```

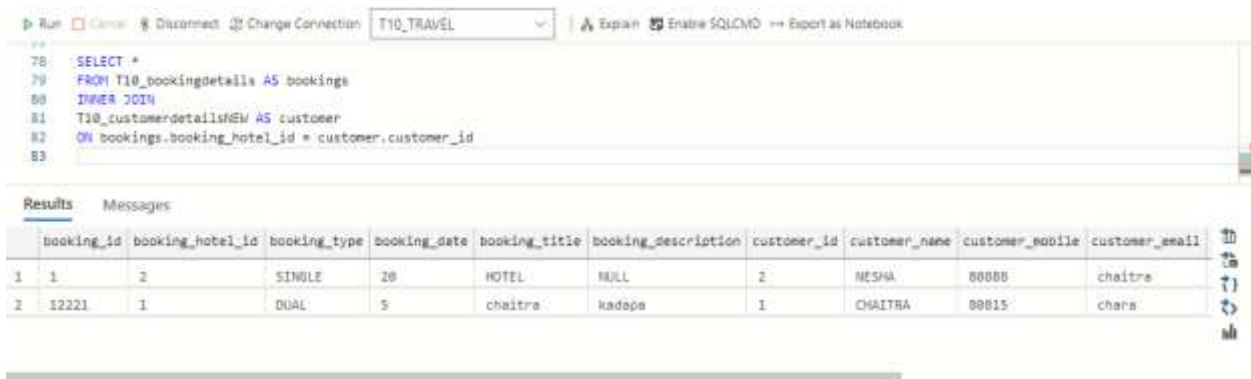
The results table is displayed below the query:

	booking_id	booking_hotel_id	booking_type	booking_date	booking_title	booking_description	hotel_id	hotel_name	hotel_type	hotel_rent	hotel_description
1	11	203	SINGLE	10	HOTEL	NULL	203	BUCHIKA	FAMILY	500	NULL
2	111	97	SINGLE	11	HOTEL ROOM	NULL	97	AKSHYA INN	SINGLE	600	NULL

QUERY 2:-

```
SELECT *
FROM T10_bookingdetails AS bookings
INNER JOIN
T10_customerdetailsNEW AS customer
ON bookings.booking_hotel_id = customer.customer_id
```

OUTPUT:-



The screenshot shows a SQL query execution interface with the following query:

```
78 SELECT *
79 FROM T10_bookingdetails AS bookings
80 INNER JOIN
81 T10_customerdetailsNEW AS customer
82 ON bookings.booking_hotel_id = customer.customer_id
83
```

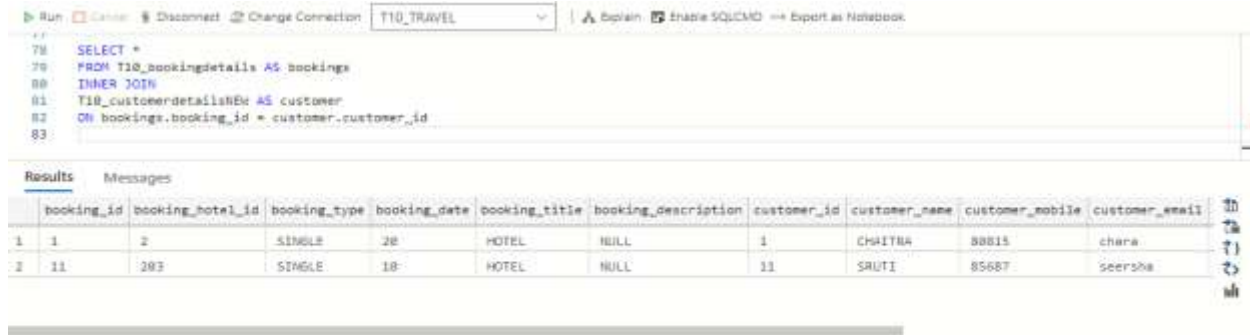
The results table is displayed below the query:

	booking_id	booking_hotel_id	booking_type	booking_date	booking_title	booking_description	customer_id	customer_name	customer_mobile	customer_email
1	1	2	SINGLE	20	HOTEL	NULL	2	NESHA	88888	chaitra
2	12221	1	DUAL	5	chaitra	kadapa	1	CHAITRA	88815	chara

QUERY 3:-

```
SELECT *
FROM T10_bookingdetails AS bookings
INNER JOIN
T10_customerdetailsNEW AS customer
ON bookings.booking_id = customer.customer_id
```

OUTPUT:-



The screenshot shows a SQL query execution interface with the following query:

```
--
78 SELECT *
79 FROM T10_bookingdetails AS bookings
80 INNER JOIN
81 T10_customerdetailsNEW AS customer
82 ON bookings.booking_id = customer.customer_id
83
```

The results table contains the following data:

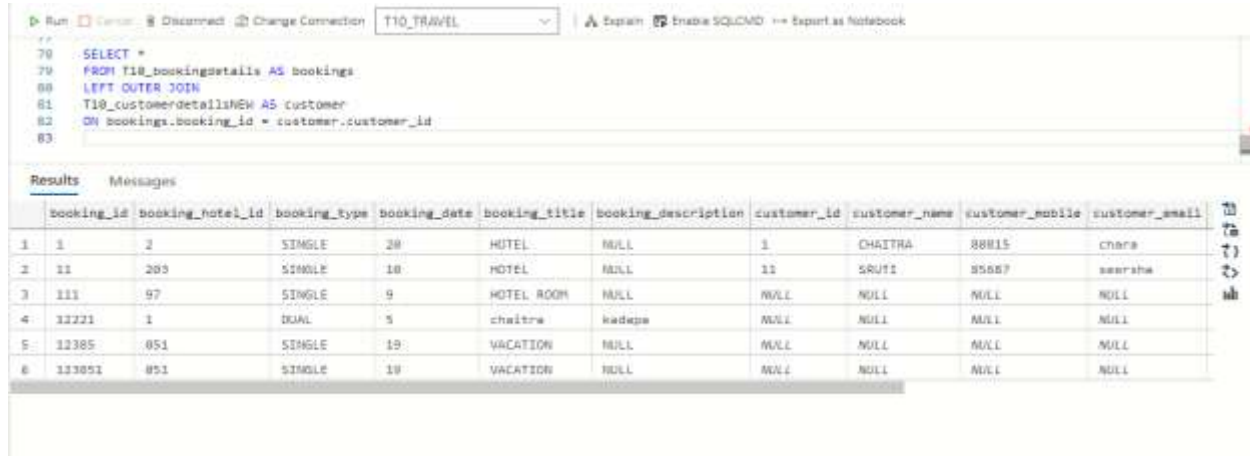
	booking_id	booking_hotel_id	booking_type	booking_date	booking_title	booking_description	customer_id	customer_name	customer_mobile	customer_email
1	1	2	SINGLE	20	HOTEL	NULL	1	CHAITRA	88815	chaitra
2	11	203	SINGLE	18	HOTEL	NULL	11	SRUTI	85687	seersha

LEFT OUTER JOIN

QUERY 1:-

```
SELECT *
FROM T10_bookingdetails AS bookings
LEFT OUTER JOIN
T10_customerdetailsNEW AS customer
ON bookings.booking_id = customer.customer_id
```

OUTPUT:-



The screenshot shows a SQL query execution interface with the following query:

```
--
78 SELECT *
79 FROM T10_bookingdetails AS bookings
80 LEFT OUTER JOIN
81 T10_customerdetailsNEW AS customer
82 ON bookings.booking_id = customer.customer_id
83
```

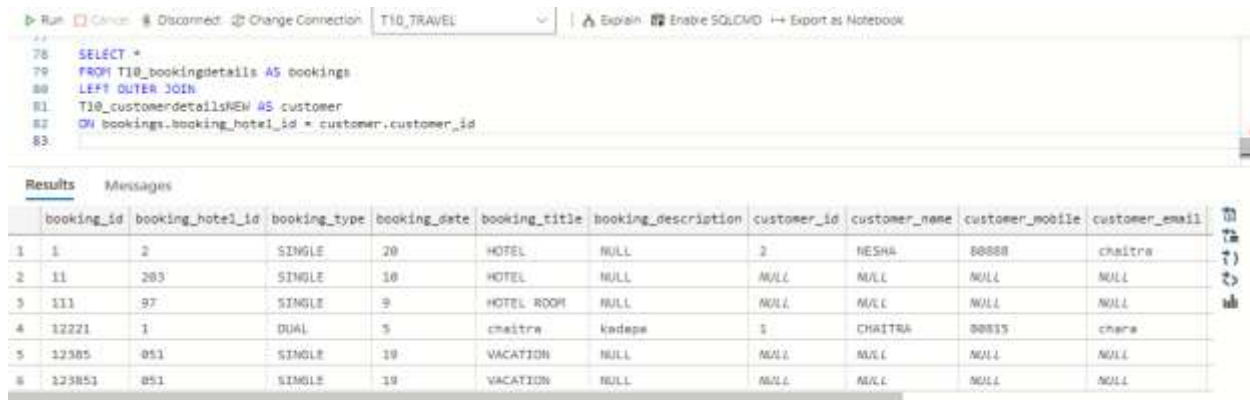
The results table contains the following data:

	booking_id	booking_hotel_id	booking_type	booking_date	booking_title	booking_description	customer_id	customer_name	customer_mobile	customer_email
1	1	2	SINGLE	20	HOTEL	NULL	1	CHAITRA	88815	chaitra
2	11	203	SINGLE	18	HOTEL	NULL	11	SRUTI	85687	seersha
3	111	97	SINGLE	9	HOTEL ROOM	NULL	NULL	NULL	NULL	NULL
4	11221	1	DUAL	5	chaitra	kadapa	NULL	NULL	NULL	NULL
5	11385	051	SINGLE	19	VACATION	NULL	NULL	NULL	NULL	NULL
6	113851	051	SINGLE	19	VACATION	NULL	NULL	NULL	NULL	NULL

QUERY 2:-

```
SELECT *
FROM T10_bookingdetails AS bookings
LEFT OUTER JOIN
T10_customerdetailsNEW AS customer
ON bookings.booking_hotel_id = customer.customer_id
```

OUTPUT:-



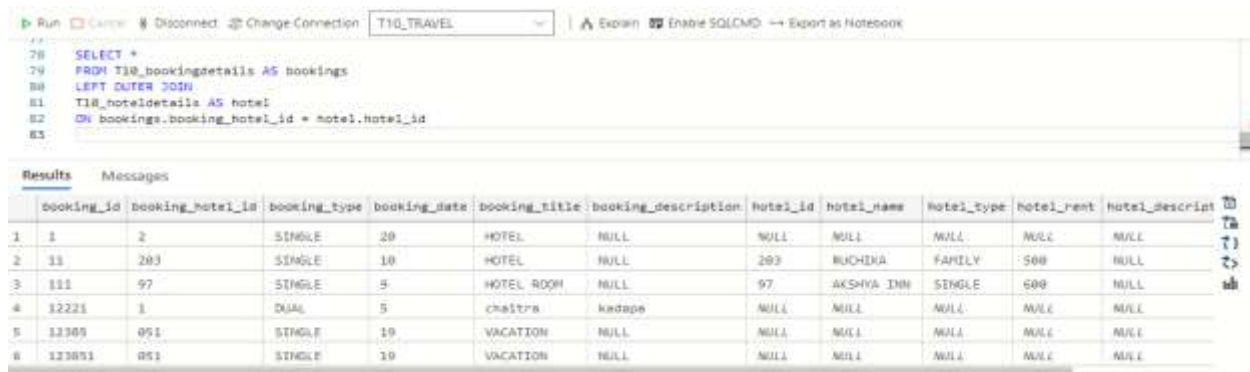
The screenshot shows a SQL query execution interface for 'T10_TRAVEL'. The query is a LEFT OUTER JOIN between 'T10_bookingdetails' (aliased as 'bookings') and 'T10_customerdetailsNEW' (aliased as 'customer') on the condition 'bookings.booking_hotel_id = customer.customer_id'. The results table has 11 columns: booking_id, booking_hotel_id, booking_type, booking_date, booking_title, booking_description, customer_id, customer_name, customer_mobile, and customer_email. The results show 6 rows, with the first row having customer details and the others being NULL.

	booking_id	booking_hotel_id	booking_type	booking_date	booking_title	booking_description	customer_id	customer_name	customer_mobile	customer_email
1	1	2	SINGLE	20	HOTEL	NULL	2	NESHA	88888	chaitra
2	11	283	SINGLE	10	HOTEL	NULL	NULL	NULL	NULL	NULL
3	111	97	SINGLE	9	HOTEL ROOM	NULL	NULL	NULL	NULL	NULL
4	12221	1	DUAL	5	chaitra	kadape	1	CHAITRA	88815	chaitra
5	12385	851	SINGLE	19	VACATION	NULL	NULL	NULL	NULL	NULL
6	123851	851	SINGLE	19	VACATION	NULL	NULL	NULL	NULL	NULL

QUERY 3:-

```
SELECT *
FROM T10_bookingdetails AS bookings
LEFT OUTER JOIN
T10_hotelldetails AS hotel
ON bookings.booking_hotel_id = hotel.hotel_id
```

OUTPUT:-



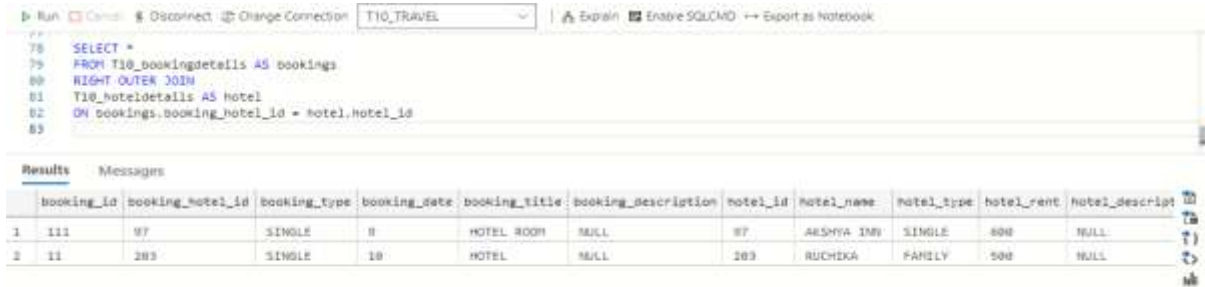
The screenshot shows a SQL query execution interface for 'T10_TRAVEL'. The query is a LEFT OUTER JOIN between 'T10_bookingdetails' (aliased as 'bookings') and 'T10_hotelldetails' (aliased as 'hotel') on the condition 'bookings.booking_hotel_id = hotel.hotel_id'. The results table has 11 columns: booking_id, booking_hotel_id, booking_type, booking_date, booking_title, booking_description, hotel_id, hotel_name, hotel_type, hotel_rent, and hotel_description. The results show 6 rows, with the first three rows having hotel details and the last three being NULL.

	booking_id	booking_hotel_id	booking_type	booking_date	booking_title	booking_description	hotel_id	hotel_name	hotel_type	hotel_rent	hotel_description
1	1	2	SINGLE	20	HOTEL	NULL	NULL	NULL	NULL	NULL	NULL
2	11	283	SINGLE	10	HOTEL	NULL	283	BUCHIKA	FAMILY	500	NULL
3	111	97	SINGLE	9	HOTEL ROOM	NULL	97	AKSHYA INN	SINGLE	600	NULL
4	12221	1	DUAL	5	chaitra	kadape	NULL	NULL	NULL	NULL	NULL
5	12385	851	SINGLE	19	VACATION	NULL	NULL	NULL	NULL	NULL	NULL
6	123851	851	SINGLE	19	VACATION	NULL	NULL	NULL	NULL	NULL	NULL

RIGHT OUTER JOIN QUERY 1:-

```
SELECT *
FROM T10_bookingdetails AS bookings
RIGHT OUTER JOIN
T10_hoteldetails AS hotel
ON bookings.booking_hotel_id = hotel.hotel_id
```

OUTPUT:-



The screenshot shows a SQL query execution interface with the following query:

```
78 SELECT *
79 FROM T10_bookingdetails AS bookings
80 RIGHT OUTER JOIN
81 T10_hoteldetails AS hotel
82 ON bookings.booking_hotel_id = hotel.hotel_id
83
```

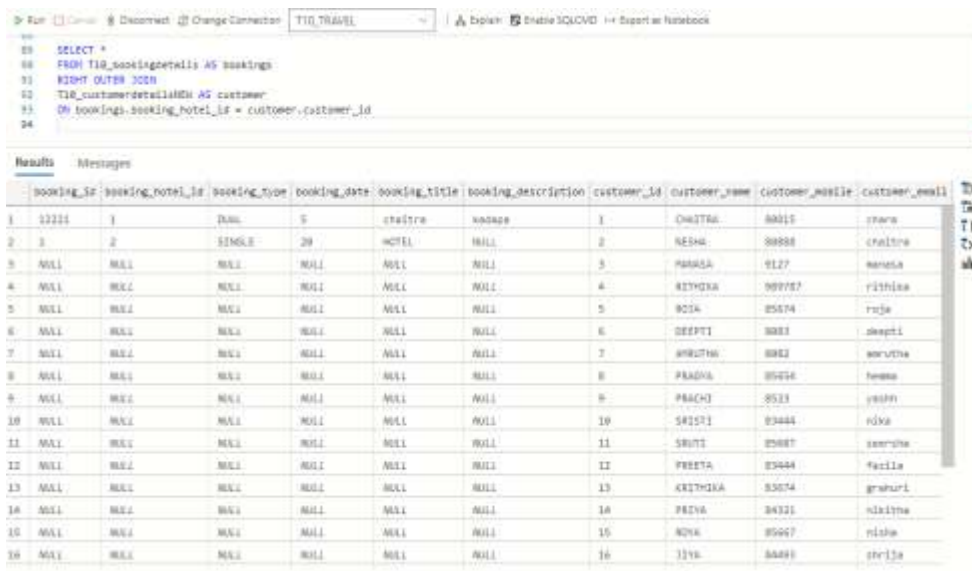
The results table displays the following data:

	booking_id	booking_hotel_id	booking_type	booking_date	booking_title	booking_description	hotel_id	hotel_name	hotel_type	hotel_rent	hotel_description
1	111	87	SINGLE	8	HOTEL ROOM	NULL	87	ARSHYA INN	SINGLE	800	NULL
2	11	283	SINGLE	10	HOTEL	NULL	283	RUCHIKA	FAMILY	500	NULL

QUERY 2:-

```
SELECT *
FROM T10_bookingdetails AS bookings
RIGHT OUTER JOIN
T10_customerdetailsNEW AS customer
ON bookings.booking_hotel_id = customer.customer_id
```

OUTPUT:



The screenshot shows a SQL query execution interface with the following query:

```
89 SELECT *
90 FROM T10_bookingdetails AS bookings
91 RIGHT OUTER JOIN
92 T10_customerdetailsNEW AS customer
93 ON bookings.booking_hotel_id = customer.customer_id
94
```

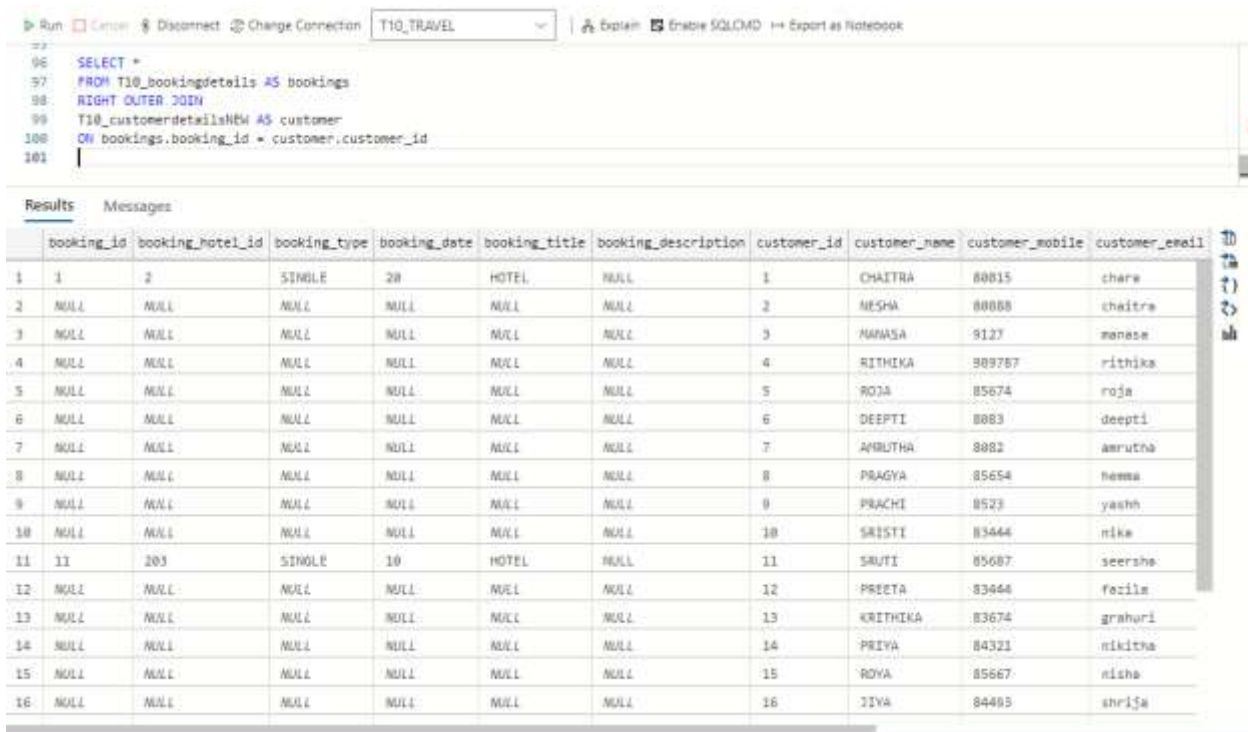
The results table displays the following data:

	booking_id	booking_hotel_id	booking_type	booking_date	booking_title	booking_description	customer_id	customer_name	customer_mobile	customer_email
1	12221	1	NULL	5	chaire	wedage	1	CHAITRA	8811	chaitra
2	3	2	SINGLE	20	HOTEL	NULL	2	NEHA	88888	chaitra
3	NULL	NULL	NULL	NULL	NULL	NULL	3	PARAG	9127	namita
4	NULL	NULL	NULL	NULL	NULL	NULL	4	RITHIKA	987787	rithika
5	NULL	NULL	NULL	NULL	NULL	NULL	5	ROJA	85574	ruje
6	NULL	NULL	NULL	NULL	NULL	NULL	6	DEEPTI	8881	deepthi
7	NULL	NULL	NULL	NULL	NULL	NULL	7	ANJITHA	8882	anurtha
8	NULL	NULL	NULL	NULL	NULL	NULL	8	PRADNA	85516	hemu
9	NULL	NULL	NULL	NULL	NULL	NULL	9	PRACHI	8513	yashu
10	NULL	NULL	NULL	NULL	NULL	NULL	10	SRISTI	83444	roja
11	NULL	NULL	NULL	NULL	NULL	NULL	11	SRUTI	85887	saurdha
12	NULL	NULL	NULL	NULL	NULL	NULL	12	PREETA	83444	parita
13	NULL	NULL	NULL	NULL	NULL	NULL	13	KRITHIKA	85574	grahuri
14	NULL	NULL	NULL	NULL	NULL	NULL	14	PRINA	84321	nikitha
15	NULL	NULL	NULL	NULL	NULL	NULL	15	ADVA	85567	nisha
16	NULL	NULL	NULL	NULL	NULL	NULL	16	JIYA	84489	shrifa

QUERY 3:-

```
SELECT *  
FROM T10_bookingdetails AS bookings  
RIGHT OUTER JOIN  
T10_customerdetailsNEW AS customer  
ON bookings.booking_id = customer.customer_id
```

OUTPUT:



The screenshot shows a SQL query execution interface. The query is as follows:

```
SELECT *  
FROM T10_bookingdetails AS bookings  
RIGHT OUTER JOIN  
T10_customerdetailsNEW AS customer  
ON bookings.booking_id = customer.customer_id
```

The results are displayed in a table with the following columns: booking_id, booking_hotel_id, booking_type, booking_date, booking_title, booking_description, customer_id, customer_name, customer_mobile, and customer_email. The results are as follows:

booking_id	booking_hotel_id	booking_type	booking_date	booking_title	booking_description	customer_id	customer_name	customer_mobile	customer_email
1	1	SINGLE	28	HOTEL	NULL	1	CHAITRA	88815	chaitra
2	NULL	NULL	NULL	NULL	NULL	2	MESHA	88888	chaitra
3	NULL	NULL	NULL	NULL	NULL	3	MANASA	9127	manasa
4	NULL	NULL	NULL	NULL	NULL	4	RITHIKA	989787	rithika
5	NULL	NULL	NULL	NULL	NULL	5	ROJA	85674	roja
6	NULL	NULL	NULL	NULL	NULL	6	DEEPTI	8883	deepti
7	NULL	NULL	NULL	NULL	NULL	7	AMRUTHA	8882	amrutha
8	NULL	NULL	NULL	NULL	NULL	8	PRAGYA	85654	nema
9	NULL	NULL	NULL	NULL	NULL	9	PRACHI	8523	yashh
10	NULL	NULL	NULL	NULL	NULL	10	SAISTI	83444	nika
11	11	SINGLE	10	HOTEL	NULL	11	SRUTI	85687	seersha
12	NULL	NULL	NULL	NULL	NULL	12	PREETA	83444	fezilm
13	NULL	NULL	NULL	NULL	NULL	13	KRITHIKA	83674	grahuri
14	NULL	NULL	NULL	NULL	NULL	14	PRIYA	84321	nikitha
15	NULL	NULL	NULL	NULL	NULL	15	ROYA	85667	nisha
16	NULL	NULL	NULL	NULL	NULL	16	JIVA	84485	shrifa

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

QUERY:-

```
SELECT customer_name, MIN(customer_id) AS customer_id, AVG(customer_mobile) AS customer_mobile  
FROM T10_customerdetailsNEW AS customers  
JOIN  
T10_bookingdetails AS bookings  
ON customers.customer_id = bookings.booking_hotel_id  
GROUP BY customer_name  
HAVING customer_name LIKE '%e%'  
ORDER BY customer_name DESC
```

OUTPUT:

SQLQuery_1.sql

Run Cancel Disconnect Change Connection T10_TRAVEL Explain Enable SQLCMD Export as Notebook

```
68 |
69 | SELECT customer_name, MIN(customer_id) AS customer_id, AVG(customer_mobile) AS customer_mobile
70 | FROM T10_customerdetailsNEW AS customers
71 | JOIN
72 | T10_bookingdetails AS bookings
73 | ON customers.customer_id = bookings.booking_hotel_id
74 | GROUP BY customer_name
75 | HAVING customer_name LIKE 'N%'
76 | ORDER BY customer_name DESC
```

Results Messages

	customer_name	customer_id	customer_mobile
1	NESHA	2	88888