

**NAME : SUJIT MANOHAR GHADI**

**SQL PROJECT**  
**HOTEL RESERVATION SYSTEM**

Begin by providing a brief background of an hotel reservation system, that led to the need for the SQL based solution. This could be an overview of the business environment, the challenges the organization was facing, or the specific needs that prompted the analysis.

**Objectives:**

**Automate the Reservation Process** – Simplify and speed up room booking, check-in, and check-out to replace manual processes.

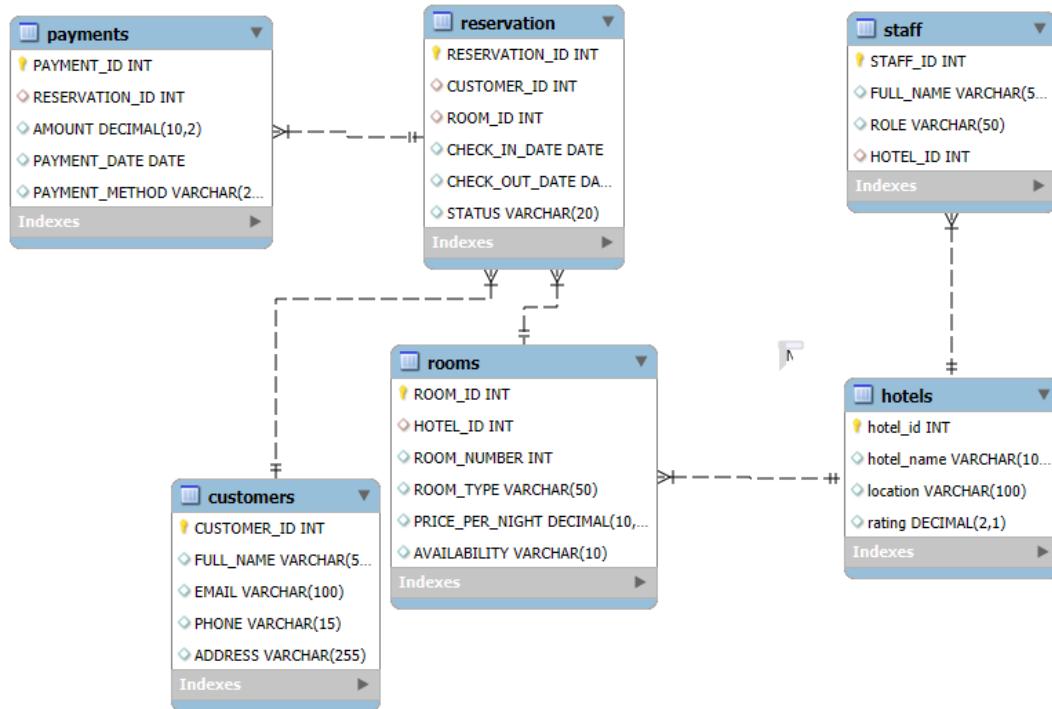
**Track Room Availability in Real-Time** – Ensure accurate and instant updates on available, booked, or under-maintenance rooms.

**Centralized Customer & Hotel Data Management** – Store and manage guest details, reservations, staff, and payment records in one system.

**Improve Customer Satisfaction** – Provide a seamless booking experience, quick service, and transparency in room pricing and availability.

**Generate Reports for Business Insights** – Enable managers to analyze occupancy rates, revenue, and customer trends for better decision-making.

## Entity-Relationship Diagram :



## Database :

```
CREATE DATABASE HOTELS;  
USE HOTELS;
```

## Creating Tables and Inserting values in it.

```
-- CREATE TABLE HOTELS  
  
CREATE TABLE HOTELS (  
    HOTEL_ID INT PRIMARY KEY,  
    HOTEL_NAME VARCHAR(100),  
    LOCATION VARCHAR(100),  
    RATING DECIMAL(2,1));
```

```
INSERT INTO HOTELS VALUES
```

```
(1,'TAJ HOTEL','MUMBAI',4.5),
(2,'VEDANGI','ANDHERI',4.2),
(3,'SAI SHANKAR','MALAD',4.7),
(4,'SAI DARBAR','VADALA',4.6),
(5,'NIKHIL','DADAR',4.1),
(6,'SHOURYWADA','BANDRA',4.8),
(7,'AARADHYA','KURLA',4.5);
```

SELECT \* FROM HOTELS;

-- OUTPUT :

	hotel_id	hotel_name	location	rating
▶	1	TAJ HOTEL	MUMBAI	4.5
	2	VEDANGI	ANDHERI	4.2
	3	SAI SHANKAR	MALAD	4.7
	4	SAI DARBAR	VADALA	4.6
	5	NIKHIL	DADAR	4.1
	6	SHOURYWADA	BANDRA	4.8
*	7	AARADHYA	KURLA	4.5
	NULL	NULL	NULL	NULL

-- CREATE TABLE ROOMS

```
CREATE TABLE ROOMS(
    ROOM_ID INT PRIMARY KEY,
    HOTEL_ID INT,
    ROOM_NUMBER INT,
    ROOM_TYPE VARCHAR(50),
    PRICE_PER_NIGHT DECIMAL(10,2),
    AVAILABILITY VARCHAR(10),
    FOREIGN KEY (HOTEL_ID) REFERENCES HOTELS(HOTEL_ID));
```

```
INSERT INTO ROOMS VALUES
```

```
(101,1,1001,'SINGLE',1000.00,'YES'),  
(102,2,1002,'SUITE',3000.00,'NO'),  
(103,3,1003,'DOUBLE',1500.00,'YES'),  
(104,4,1004,'SINGLE',1000.00,'YES'),  
(105,5,1005,'SUITE',3000.00,'NO'),  
(106,6,1006,'DOUBLE',1500.00,'YES'),  
(107,7,1007,'SINGLE',1000.00,'NO');
```

```
SELECT * FROM ROOMS;
```

-- OUTPUT :

	ROOM_ID	HOTEL_ID	ROOM_NUMBER	ROOM_TYPE	PRICE_PER_NIGHT	AVAILABILITY
▶	101	1	1001	SINGLE	1000.00	YES
	102	2	1002	SUITE	3000.00	NO
	103	3	1003	DOUBLE	1500.00	YES
	104	4	1004	SINGLE	1000.00	YES
	105	5	1005	SUITE	3000.00	NO
	106	6	1006	DOUBLE	1500.00	YES
	107	7	1007	SINGLE	1000.00	NO
*	NULL	NULL	NULL	NULL	NULL	NULL

-- CREATE TABLE CUSTOMERS

```
CREATE TABLE CUSTOMERS(  
CUSTOMER_ID INT PRIMARY KEY,  
FULL_NAME VARCHAR(50),  
EMAIL VARCHAR(100) UNIQUE,  
PHONE VARCHAR(15),  
ADDRESS VARCHAR(255));
```

```
INSERT INTO CUSTOMERS VALUES
```

```
(201,'SAKSHI DONGARE','SAKSHI@GMAIL.COM','9402829493','SAI MANDIR ANDHERI EAST'),  
(202,'OMKAR PARAB','OMKAR@GMAIL.COM','9502842492','LINK ROAD GOREGAON WEST'),  
(203,'TUSHAR ANBHAVANE','TUSHAR@GMAIL.COM','9002129488','SHIVAJI CHOUK DADAR WEST'),
```

(204,'SOHAM TANAVADE','SOHAM@GMAIL.COM','9402832125','RAM MANDIR ANDHERI EAST'),  
 (205,'OM MAPHSEKAR','OM@GMAIL.COM','9826836171','GANESH CHOUK BORIVALI WEST'),  
 (206,'SHAMIKA SINGH','SHAMIKA@GMAIL.COM','9402827575','SAI MANDIR BANDRA WEST');

INSERT INTO CUSTOMERS VALUES (207,'SAHIL JADHAV','SAHILJADHAV@GMAIL.COM','8543542342','VIRVANI GOREGAON EAST');

SELECT \* FROM CUSTOMERS;

-- OUTPUT :

	CUSTOMER_ID	FULL_NAME	EMAIL	PHONE	ADDRESS
▶	201	SAKSHI DONGARE	SAKSHI@GMAIL.COM	9402829493	SAI MANDIR ANDHERI EAST
	202	OMKAR PARAB	OMKAR@GMAIL.COM	9502842492	LINK ROAD GOREGAON WEST
	203	TUSHAR ANBHAVANE	TUSHAR@GMAIL.COM	9002129488	SHIVAJI CHOUK DADAR WEST
	204	SOHAM TANAVADE	SOHAM@GMAIL.COM	9402832125	RAM MANDIR ANDHERI EAST
	205	OM MAPHSEKAR	OM@GMAIL.COM	9826836171	GANESH CHOUK BORIVALI WEST
	206	SHAMIKA SINGH	SHAMIKA@GMAIL.COM	9402827575	SAI MANDIR BANDRA WEST
*	207	SAHIL JADHAV	SAHILJADHAV@GMAIL.COM	8543542342	VIRVANI GOREGAON EAST
	NULL	NULL	NULL	NULL	NULL

-- CREATE TABLE RESERVATION

CREATE TABLE RESERVATION(  
 RESERVATION\_ID INT PRIMARY KEY,  
 CUSTOMER\_ID INT,  
 ROOM\_ID INT,  
 CHECK\_IN\_DATE DATE,  
 CHECK\_OUT\_DATE DATE,  
 STATUS VARCHAR(20),  
 FOREIGN KEY (CUSTOMER\_ID) REFERENCES CUSTOMERS(CUSTOMER\_ID),  
 FOREIGN KEY (ROOM\_ID) REFERENCES ROOMS(ROOM\_ID));

INSERT INTO RESERVATION VALUES

(301,201,101,'2025-06-02','2025-06-03','CHACKED OUT'),  
 (302,202,102,'2025-08-08','2025-08-10','BOOKED'),

```
(303,203,103,'2025-08-18','2025-08-19','CANCELLED'),
(304,204,104,'2025-08-24','2025-08-26','CHECKED IN'),
(305,205,105,'2025-08-25','2025-09-03','BOOKED'),
(306,206,106,'2025-09-05','2025-09-08','CANCELLED');
```

SELECT \* FROM RESERVATION;

-- OUTPUT :

	RESERVATION_ID	CUSTOMER_ID	ROOM_ID	CHECK_IN_DATE	CHECK_OUT_DATE	STATUS
▶	301	201	101	2025-06-02	2025-06-03	CHACKED OUT
	302	202	102	2025-08-08	2025-08-10	BOOKED
	303	203	103	2025-08-18	2025-08-19	CANCELLED
	304	204	104	2025-08-24	2025-08-26	CHECKED IN
	305	205	105	2025-08-25	2025-09-03	BOOKED
*	306	206	106	2025-09-05	2025-09-08	CANCELLED
	NULL	NULL	NULL	NULL	NULL	NULL

-- CREATE TABLE PAYMENTS

```
CREATE TABLE PAYMENTS(
PAYMENT_ID INT PRIMARY KEY,
RESERVATION_ID INT,
AMOUNT DECIMAL(10,2),
PAYMENT_DATE DATE,
PAYMENT_METHOD VARCHAR(20),
FOREIGN KEY (RESERVATION_ID) REFERENCES RESERVATION(RESERVATION_ID));
```

INSERT INTO PAYMENTS VALUES

```
(401,301,2000.00,'2025-06-02','CARD'),
(402,302,3000.00,'2025-08-08','ONLINE'),
(403,303,2500.00,'2025-08-18','CASH'),
(404,304,2000.00,'2025-08-24','CARD'),
(405,305,2000.00,'2025-08-25','ONLINE'),
(406,306,3000.00,'2025-09-05','ONLINE');
```

SELECT \* FROM PAYMENTS;

-- OUTPUT :

	PAYMENT_ID	RESERVATION_ID	AMOUNT	PAYMENT_DATE	PAYMENT_METHOD
▶	401	301	2000.00	2025-06-02	CARD
	402	302	3000.00	2025-08-08	ONLINE
	403	303	2500.00	2025-08-18	CASH
	404	304	2000.00	2025-08-24	CARD
	405	305	2000.00	2025-08-25	ONLINE
	406	306	3000.00	2025-09-05	ONLINE
*	NULL	NULL	NULL	NULL	NULL

-- CREATE TABLE STAFF

```
CREATE TABLE STAFF(  
    STAFF_ID INT PRIMARY KEY,  
    FULL_NAME VARCHAR(50),  
    ROLE VARCHAR(50),  
    HOTEL_ID INT,  
    FOREIGN KEY (HOTEL_ID) REFERENCES HOTELS(HOTEL_ID));
```

INSERT INTO STAFF VALUES

```
(501,'HARSHIT RANA', 'MANAGER',1),  
(502,'MITALI SINGH', 'RECEPTIONIST',2),  
(503,'NAMRTA YADAV', 'HOUSEKEEPING',3),  
(504,'SHIVA RAMA', 'HOUSEKEEPING',4),  
(505,'SURAJ RAI', 'WORKER',5),  
(506,'RAMRAJU SINHA', 'CLEANER',6);
```

SELECT \* FROM STAFF;

-- OUTPUT :

	STAFF_ID	FULL_NAME	ROLE	HOTEL_ID
▶	501	HARSHIT RANA	MANAGER	1
	502	MITALI SINGH	RECEPTIONIST	2
	503	NAMRTA YADAV	HOUSEKEEPING	3
	504	SHIVA RAMA	HOUSEKEEPING	4
	505	SURAJ RAI	WORKER	5
	506	RAMRAJU SINHA	CLEANER	6
*	NULL	NULL	NULL	NULL

## Queries :

-- Q.1 GET HOTEL NAME AND ROOM NUMBER OF ALL AVAILABLE ROOMS.

```
SELECT H.HOTEL_NAME, R.ROOM_NUMBER, ROOM_TYPE, R.AVAILABILITY FROM ROOMS R
LEFT JOIN HOTELS H
ON R.HOTEL_ID = H.HOTEL_ID
WHERE R.AVAILABILITY = 'YES';
```

-- OUTPUT :

	HOTEL_NAME	ROOM_NUMBER	ROOM_TYPE	AVAILABILITY
▶	TAJ HOTEL	1001	SINGLE	YES
	SAI SHANKAR	1003	DOUBLE	YES
	SAI DARBAR	1004	SINGLE	YES
	SHOURYWADA	1006	DOUBLE	YES

-- Q.2 LIST ALL ROOMS IN SAI DARBAR HOTEL.

```
SELECT R.ROOM_ID, H.HOTEL_NAME, R.ROOM_NUMBER, R.ROOM_TYPE, R.PRICE_PER_NIGHT
FROM ROOMS R
RIGHT JOIN HOTELS H
ON R.HOTEL_ID = H.HOTEL_ID
WHERE HOTEL_NAME = 'SAI DARBAR';
```

-- OUTPUT :

	ROOM_ID	HOTEL_NAME	ROOM_NUMBER	ROOM_TYPE	PRICE_PER_NIGHT
▶	104	SAI DARBAR	1004	SINGLE	1000.00

-- Q.3 FIND AVAILABLE ROOMS IN HOTELS

```
SELECT * FROM ROOMS
WHERE AVAILABILITY = 'YES';
```

-- OUTPUT :

	ROOM_ID	HOTEL_ID	ROOM_NUMBER	ROOM_TYPE	PRICE_PER_NIGHT	AVAILABILITY
▶	101	1	1001	SINGLE	1000.00	YES
	103	3	1003	DOUBLE	1500.00	YES
	104	4	1004	SINGLE	1000.00	YES
	106	6	1006	DOUBLE	1500.00	YES
+	NUL	NUL	NUL	NUL	NUL	NUL

-- Q.4 Show customers who never made a reservation.

```
SELECT C.CUSTOMER_ID, C.FULL_NAME FROM CUSTOMERS C  
LEFT JOIN RESERVATION R  
ON C.CUSTOMER_ID = R.CUSTOMER_ID  
WHERE R.RESERVATION_ID IS NULL;
```

-- OUTPUT :

	CUSTOMER_ID	FULL_NAME
▶	207	SAHIL JADHAV

-- Q.5 FIND THE PHONE NUMBER OF THE CUSTOMER OMKAR PARAB.

```
SELECT FULL_NAME, PHONE FROM CUSTOMERS  
WHERE CUSTOMER_ID = 202;
```

-- OUTPUT :

	FULL_NAME	PHONE
▶	OMKAR PARAB	9502842492

-- Q.6 LIST STAFF WORKING IN EACH HOTEL.

```
SELECT H.HOTEL_NAME, S.FULL_NAME, S.ROLE  
FROM STAFF S  
JOIN HOTELS H  
ON S.HOTEL_ID = H.HOTEL_ID;
```

-- OUTPUT :

	HOTEL_NAME	FULL_NAME	ROLE
▶	TAJ HOTEL	HARSHIT RANA	MANAGER
	VEDANGI	MITALI SINGH	RECEPTIONIST
	SAI SHANKAR	NAMRTA YADAV	HOUSEKEEPING
	SAI DARBAR	SHIVA RAMA	HOUSEKEEPING
	NIKHIL	SURAJ RAI	WORKER
	SHOURYWADA	RAMRAJU SINHA	CLEANER

-- Q.7 LIST ALL ROOMS IN HOTELS SHOURYWADA.

```
SELECT R.ROOM_ID,R.ROOM_NUMBER, R.ROOM_TYPE,R.PRICE_PER_NIGHT,R.AVAILABILITY  
FROM ROOMS R  
  
JOIN HOTELS H  
  
ON R.HOTEL_ID = H.HOTEL_ID  
  
WHERE H.HOTEL_NAME = 'SHOURYWADA';
```

-- OUTPUT :

	ROOM_ID	ROOM_NUMBER	ROOM_TYPE	PRICE_PER_NIGHT	AVAILABILITY
▶	106	1006	DOUBLE	1500.00	YES

-- Q.8 COUNT HOW MENY ROOMS A HOTEL NIKHIL.

```
SELECT H.HOTEL_NAME ,COUNT(R.ROOM_ID) AS TOTAL_ROOM FROM HOTELS H  
LEFT JOIN ROOMS R  
ON H.HOTEL_ID = R.ROOM_ID  
WHERE H.HOTEL_ID = '5';
```

-- OUTPUT :

	HOTEL_NAME	TOTAL_ROOM
▶	NIKHIL	0

-- Q.9 SHOW ALL CUSTOMERS WHO 'BOOKED'A SUITE.

```
SELECT C.FULL_NAME, RO.ROOM_TYPE, R.STATUS FROM ROOMS RO  
JOIN RESERVATION R  
ON R.ROOM_ID = RO.ROOM_ID  
JOIN CUSTOMERS C  
ON R.CUSTOMER_ID = C.CUSTOMER_ID  
WHERE R.STATUS = 'BOOKED';
```

-- OUTPUT :

	FULL_NAME	ROOM_TYPE	STATUS
▶	OMKAR PARAB	SUITE	BOOKED
	OM MAPHSEKAR	SUITE	BOOKED

-- Q.10 FIND THE STAFF ROLE 'MANAGER' IN EACH HOTEL.

```
SELECT S.FULL_NAME, S.ROLE, H.HOTEL_NAME FROM STAFF S  
LEFT JOIN HOTELS H  
ON S.HOTEL_ID = H.HOTEL_ID  
WHERE S.ROLE = 'MANAGER';
```

-- OUTPUT :

	FULL_NAME	ROLE	HOTEL_NAME
▶	HARSHIT RANA	MANAGER	TAJ HOTEL

-- Q.11 SHOW ALL RESERVATIONS FOR 'TUSHAR ANBHAVANE' .

```
SELECT R.RESERVATION_ID, C.FULL_NAME, RO.ROOM_NUMBER, RO.ROOM_TYPE, R.STATUS  
FROM RESERVATION R  
JOIN CUSTOMERS C  
ON R.CUSTOMER_ID = C.CUSTOMER_ID  
JOIN ROOMS RO  
ON R.ROOM_ID = RO.ROOM_ID  
WHERE C.CUSTOMER_ID = 203;
```

-- OUTPUT :

	RESERVATION_ID	FULL_NAME	ROOM_NUMBER	ROOM_TYPE	STATUS
▶	303	TUSHAR ANBHAVANE	1003	DOUBLE	CANCELLED

-- Q.12 GET ALL RESERVATIONS WHERE CHECK\_OUT\_DATE AFTER 2025-08-08.

```
SELECT R.RESERVATION_ID, R.CUSTOMER_ID, C.FULL_NAME, R.CHECK_OUT_DATE FROM  
RESERVATION R  
RIGHT JOIN CUSTOMERS C  
ON R.CUSTOMER_ID = C.CUSTOMER_ID  
WHERE R.CHECK_OUT_DATE > '2025-08-10';
```

-- OUTPUT :

	RESERVATION_ID	CUSTOMER_ID	FULL_NAME	CHECK_OUT_DATE
▶	302	202	OMKAR PARAB	2025-08-10
	303	203	TUSHAR ANBHAVANE	2025-08-19
	304	204	SOHAM TANAVADE	2025-08-26
	305	205	OM MAPHSEKAR	2025-09-03
	306	206	SHAMIKA SINGH	2025-09-08

-- Q.13 FIND ROOMS THAT COST MORE THAN 1500.00 PER NIGHT.

```
SELECT ROOM_ID, ROOM_NUMBER, ROOM_TYPE, PRICE_PER_NIGHT FROM ROOMS  
WHERE PRICE_PER_NIGHT > 1500.00;
```

-- OUTPUT :

	ROOM_ID	ROOM_NUMBER	ROOM_TYPE	PRICE_PER_NIGHT
▶	102	1002	SUITE	3000.00
	105	1005	SUITE	3000.00
*	NULL	NULL	NULL	NULL

-- Q.14 SHOW LIST USING ONLY ONLINE PAYMENT METHOD.

```
SELECT P.PAYMENT_ID, P.AMOUNT, P.PAYMENT_DATE, P.PAYMENT_METHOD, C.FULL_NAME  
FROM PAYMENTS P  
JOIN RESERVATION R  
ON P.RESERVATION_ID = R.RESERVATION_ID  
JOIN CUSTOMERS C  
ON R.CUSTOMER_ID = C.CUSTOMER_ID  
WHERE P.PAYMENT_METHOD = 'ONLINE';
```

-- OUTPUT :

	PAYMENT_ID	AMOUNT	PAYMENT_DATE	PAYMENT_METHOD	FULL_NAME
▶	402	3000.00	2025-08-08	ONLINE	OMKAR PARAB
	405	2000.00	2025-08-25	ONLINE	OM MAPHSEKAR
	406	3000.00	2025-09-05	ONLINE	SHAMIKA SINGH

-- Q.15 GET HOTEL NAME AND ROOM NUMBERS OF ALL AVAILABLE ROOMS.

```
SELECT H.HOTEL_NAME, R.ROOM_NUMBER, R.AVAILABILITY FROM HOTELS H  
LEFT JOIN ROOMS R  
ON H.HOTEL_ID = R.HOTEL_ID  
WHERE AVAILABILITY = 'YES';
```

-- OUTPUT :

	HOTEL_NAME	ROOM_NUMBER	AVAILABILITY
▶	TAJ HOTEL	1001	YES
	SAI SHANKAR	1003	YES
	SAI DARBAR	1004	YES
	SHOURYWADA	1006	YES

-- Q.16 FIND CUSTOMERS WITH THEIR RESERVATION 'TAJ HOTEL'.

```
SELECT H.HOTEL_NAME,R.CUSTOMER_ID, C.FULL_NAME, R.ROOM_ID, R.STATUS FROM  
RESERVATION R  
JOIN CUSTOMERS C  
ON R.CUSTOMER_ID = C.CUSTOMER_ID  
JOIN ROOMS RO  
ON RO.ROOM_ID = R.ROOM_ID  
JOIN HOTELS H  
ON RO.HOTEL_ID = H.HOTEL_ID  
WHERE H.HOTEL_ID = 1;
```

-- OUTPUT :

	HOTEL_NAME	CUSTOMER_ID	FULL_NAME	ROOM_ID	STATUS
▶	TAJ HOTEL	201	SAKSHI DONGARE	101	CHACKED OUT

-- Q.17 CALCULATE TOTAL REVENUE PER HOTEL.

```
SELECT H.HOTEL_NAME, sum(P.AMOUNT) AS TOTAL_REVENUE  
FROM PAYMENTS P  
JOIN RESERVATION R  
ON P.RESERVATION_ID = R.RESERVATION_ID
```

```

JOIN ROOMS RO
ON RO.ROOM_ID = R.ROOM_ID
JOIN HOTELS H
ON RO.HOTEL_ID = H.HOTEL_ID
GROUP BY H.HOTEL_NAME
ORDER BY TOTAL_REVENUE;

```

**-- OUTPUT :**

	HOTEL_NAME	TOTAL_REVENUE
▶	TAJ HOTEL	2000.00
	VEDANGI	3000.00
	SAI SHANKAR	2500.00
	SAI DARBAR	2000.00
	NIKHIL	2000.00
	SHOURYWADA	3000.00

**-- Q.18 FIND CUSTOMER FROM MULTIPLE RESERVATIONS.**

```

SELECT C.FULL_NAME, COUNT(R.RESERVATION_ID)AS TOTAL_RESERVATIONS
FROM CUSTOMERS C
JOIN RESERVATION R
ON C.CUSTOMER_ID = R.CUSTOMER_ID
GROUP BY C.CUSTOMER_ID
HAVING COUNT(R.RESERVATION_ID);

```

**-- OUTPUT :**

	FULL_NAME	TOTAL_RESERVATIONS
▶	SAKSHI DONGARE	1 1
	OMKAR PARAB	1
	TUSHAR ANBHAVANE	1
	SOHAM TANAVADE	1
	OM MAPHSEKAR	1
	SHAMIKA SINGH	1

-- Q.19 LIST RESARVATION WITH CUSTOMER AND HOTEL DETAILED.

```
SELECT H.HOTEL_ID, H.HOTEL_NAME, H.LOCATION, H.RATING, C.CUSTOMER_ID,  
C.FULL_NAME, R.STATUS FROM RESERVATION R  
  
JOIN CUSTOMERS C  
  
ON R.CUSTOMER_ID = C.CUSTOMER_ID  
  
JOIN ROOMS RO  
  
ON RO.ROOM_ID = R.ROOM_ID  
  
JOIN HOTELS H  
  
ON RO.HOTEL_ID = H.HOTEL_ID;
```

-- OUTPUT :

	HOTEL_ID	HOTEL_NAME	LOCATION	RATING	CUSTOMER_ID	FULL_NAME	STATUS
▶	1	TAJ HOTEL	MUMBAI	4.5	201	SAKSHI DONGARE	CHECKED OUT
	2	VEDANGI	ANDHERI	4.2	202	OMKAR PARAB	BOOKED
	3	SAI SHANKAR	MALAD	4.7	203	TUSHAR ANBHAVANE	CANCELLED
	4	SAI DARBAR	VADALA	4.6	204	SOHAM TANAVADE	CHECKED IN
	5	NIKHIL	DADAR	4.1	205	OM MAPHSEKAR	BOOKED
	6	SHOURYWADA	BANDRA	4.8	206	SHAMIKA SINGH	CANCELLED

-- Q.20 SHOW PAYMENT WITH CUSTOMER 'SHAMIKA SINGH' OR HOTELS NAME 'NIKHIL'.

```
SELECT C.FULL_NAME, H.HOTEL_NAME, P.PAYMENT_METHOD FROM HOTELS H
```

```
JOIN ROOMS RO
```

```
ON H.HOTEL_ID = RO.HOTEL_ID
```

```
JOIN RESERVATION R
```

```
ON R.ROOM_ID = RO.ROOM_ID
```

```
JOIN PAYMENTS P
```

```
ON P.RESERVATION_ID = R.RESERVATION_ID
```

```
JOIN CUSTOMERS C
```

```
ON C.CUSTOMER_ID = R.CUSTOMER_ID
```

```
WHERE C.FULL_NAME = 'SHAMIKA SINGH' OR H.HOTEL_NAME = 'NIKHIL';
```

-- OUTPUT :

	FULL_NAME	HOTEL_NAME	PAYMENT_METHOD
▶	OM MAPHSEKAR	NIKHIL	ONLINE
	SHAMIKA SINGH	SHOURYWADA	ONLINE

-- Q.21 FIND RESERVATION WITH THEIR ASSIGNED STAFF (RECEPTIONIST).

```
SELECT S.FULL_NAME, H.HOTEL_NAME, S.ROLE, R.STATUS FROM STAFF S
JOIN HOTELS H
ON S.HOTEL_ID = H.HOTEL_ID
JOIN ROOMS RO
ON H.HOTEL_ID = RO.HOTEL_ID
JOIN RESERVATION R
ON RO.ROOM_ID = R.ROOM_ID
WHERE S.ROLE = 'RECEPTIONIST';
```

-- OUTPUT :

	FULL_NAME	HOTEL_NAME	ROLE	STATUS
▶	MITALI SINGH	VEDANGI	RECEPTIONIST	BOOKED

-- Q.22 GET THE AVERAGE PAYMENT PER CUSTOMER.

```
SELECT C.FULL_NAME, AVG(P.AMOUNT) AS AVG_PAYMENT FROM PAYMENTS P
JOIN RESERVATION R
ON P.RESERVATION_ID = R.RESERVATION_ID
JOIN CUSTOMERS C
ON R.CUSTOMER_ID = C.CUSTOMER_ID
GROUP BY C.CUSTOMER_ID ,C.FULL_NAME;
```

-- OUTPUT :

	FULL_NAME	AVG_PAYMENT
▶	SAKSHI DONGARE	2000.000000
	OMKAR PARAB	3000.000000
	TUSHAR ANBHAVANE	2500.000000
	SOHAM TANAVADE	2000.000000
	OM MAPHSEKAR	2000.000000
	SHAMIKA SINGH	3000.000000

-- Q.23 FIND HOTEL WITH MAXIMUM TOTAL REVENUE.

```
SELECT H.HOTEL_ID,H.HOTEL_NAME,SUM(P.AMOUNT) AS MAXIMUM_TOTAL_REVENUE FROM PAYMENTS P
JOIN RESERVATION R
ON P.RESERVATION_ID = R.RESERVATION_ID
```

```
JOIN ROOMS RO
ON R.ROOM_ID = RO.ROOM_ID
JOIN HOTELS H
ON RO.HOTEL_ID = H.HOTEL_ID
GROUP BY H.HOTEL_NAME,H.HOTEL_ID
ORDER BY MAXIMUM_TOTAL_REVENUE DESC
LIMIT 1;
```

-- OUTPUT :

	HOTEL_ID	HOTEL_NAME	MAXIMUM_TOTAL_REVENUE
▶	6	SHOURYWADA	3000.00

-- Q.24 COUNT HOW MANY CUSTOMERS BOOKED EACH ROOM TYPE.

```
SELECT R.ROOM_TYPE,COUNT(DISTINCT C.CUSTOMER_ID) AS TOTAL_CUSTOMERS
FROM ROOMS R
JOIN RESERVATION RO
ON R.ROOM_ID = RO.ROOM_ID
JOIN CUSTOMERS C
ON RO.CUSTOMER_ID = C.CUSTOMER_ID
GROUP BY R.ROOM_TYPE
ORDER BY TOTAL_CUSTOMERS DESC;
```

-- OUTPUT :

	ROOM_TYPE	TOTAL_CUSTOMERS
▶	DOUBLE	2
	SINGLE	2
	SUITE	2

-- Q.25 LIST CUSTOMERS WHO PAID MORE THAN THE AVERAGE PAYMENT.

```
SELECT C.CUSTOMER_ID,C.FULL_NAME,P.AMOUNT FROM PAYMENTS P
JOIN RESERVATION R
ON P.RESERVATION_ID = R.RESERVATION_ID
JOIN CUSTOMERS C
ON C.CUSTOMER_ID = R.CUSTOMER_ID
WHERE P.AMOUNT > (SELECT AVG(P.AMOUNT) FROM PAYMENTS P);
```

-- OUTPUT :

	CUSTOMER_ID	FULL_NAME	AMOUNT
▶	202	OMKAR PARAB	3000.00
	203	TUSHAR ANBHAVANE	2500.00
	206	SHAMIKA SINGH	3000.00

-- Q.26 FIND CUSTOMERS WITH PAYMENTS GREATER THAN ALL OTHERS.

```
SELECT C.CUSTOMER_ID,C.FULL_NAME,P.AMOUNT FROM PAYMENTS P
JOIN RESERVATION R
ON P.RESERVATION_ID = R.RESERVATION_ID
JOIN CUSTOMERS C
ON C.CUSTOMER_ID = R.CUSTOMER_ID
WHERE P.AMOUNT > ALL (SELECT P2.AMOUNT FROM PAYMENTS P2 WHERE R.CUSTOMER_ID <>
C.CUSTOMER_ID);
```

-- OUTPUT :

	CUSTOMER_ID	FULL_NAME	AMOUNT
▶	201	SAKSHI DONGARE	2000.00
	202	OMKAR PARAB	3000.00
	203	TUSHAR ANBHAVANE	2500.00
	204	SOHAM TANAVADE	2000.00
	205	OM MAPHSEKAR	2000.00
	206	SHAMIKA SINGH	3000.00

-- Q.27 Find the second highest total payment by customer.

```
SELECT CUSTOMER_ID, FULL_NAME, TOTAL_PAYMENT FROM  
(SELECT C.CUSTOMER_ID,C.FULL_NAME,SUM(P.AMOUNT) AS TOTAL_PAYMENT  
FROM CUSTOMERS C  
JOIN RESERVATION R ON C.CUSTOMER_ID = R.CUSTOMER_ID  
JOIN PAYMENTS P ON R.RESERVATION_ID = P.RESERVATION_ID  
GROUP BY C.CUSTOMER_ID, C.FULL_NAME  
ORDER BY TOTAL_PAYMENT DESC  
LIMIT 2)AS T  
ORDER BY TOTAL_PAYMENT ASC  
LIMIT 1;
```

-- OUTPUT :

	CUSTOMER_ID	FULL_NAME	TOTAL_PAYMENT
▶	206	SHAMIKA SINGH	3000.00

-- Q.28 FIND STAFF IN HOTELS WHERE ALL ROOMS ARE AVAILABLE.

```
SELECT S.STAFF_ID, S.FULL_NAME, H.HOTEL_NAME FROM STAFF S  
JOIN HOTELS H ON S.HOTEL_ID = H.HOTEL_ID  
WHERE NOT EXISTS (SELECT 1 FROM ROOMS R WHERE R.HOTEL_ID = H.HOTEL_ID AND  
R.AVAILABILITY = 'YES');
```

-- OUTPUT :

	STAFF_ID	FULL_NAME	HOTEL_NAME
▶	502	MITALI SINGH	VEDANGI
	505	SURAJ RAI	NIKHIL

-- Q.29 FIND RESERVATION WITH PAYMENT HIGHER THAN HOTEL'S AVERAGE ROOM PRICE.

```
SELECT R.RESERVATION_ID,C.CUSTOMER_ID,C.FULL_NAME,H.HOTEL_NAME,P.AMOUNT AS  
PAYMENT,RO.PRICE_PER_NIGHT AS ROOM_PRICE  
FROM RESERVATION R  
JOIN CUSTOMERS C ON R.CUSTOMER_ID = C.CUSTOMER_ID
```

```

JOIN ROOMS RO ON R.ROOM_ID = RO.ROOM_ID
JOIN HOTELS H ON RO.HOTEL_ID = H.HOTEL_ID
JOIN PAYMENTS P ON R.RESERVATION_ID = P.RESERVATION_ID
WHERE P.AMOUNT > (SELECT AVG(RO2.PRICE_PER_NIGHT) FROM ROOMS RO2 WHERE
RO2.HOTEL_ID = H.HOTEL_ID);

```

-- OUTPUT :

	RESERVATION_ID	CUSTOMER_ID	FULL_NAME	HOTEL_NAME	PAYMENT	ROOM_PRICE
▶	301	201	SAKSHI DONGARE	TAJ HOTEL	2000.00	1000.00
	303	203	TUSHAR ANBHAVANE	SAI SHANKAR	2500.00	1500.00
	304	204	SOHAM TANAVADE	SAI DARBAR	2000.00	1000.00
	306	206	SHAMIKA SINGH	SHOURYWADA	3000.00	1500.00

-- Q.30 GET STAFF WHO WORK IN HOTELS WITH HIGHEST RATING.

```

SELECT S.STAFF_ID,S.FULL_NAME,H.HOTEL_NAME,H.RATING
FROM STAFF S
JOIN HOTELS H ON S.HOTEL_ID = H.HOTEL_ID
WHERE H.RATING = (SELECT MAX(H.RATING) FROM HOTELS H);

```

-- OUTPUT :

	STAFF_ID	FULL_NAME	HOTEL_NAME	RATING
▶	506	RAMRAJU SINHA	SHOURYWADA	4.8