Case Study 1: OYO business

Creating Tables:

i . Sales Table:

```
create database oyo;
use oyo;

create table sales
(booking_id int primary key,
customer_id int,
bking_status varchar(10) not null check (bking_status in('Cancelled', 'Stayed', 'No Show')),
check_in date,
check_out date,
no_of_rooms int,
hotel_id int,
amount int,
discount int,
date_of_booking date);

alter table sales add foreign key (hotel_id) references cities(hotel_id);

insert into sales values (2001, 9001, 'Cancelled', '2022-01-10', '2022-01-12', 1, 3, 3000, 500, '2022-01-01');
insert into sales values (2002, 9002, 'Stayed', '2022-02-01', '2022-02-03', 2, 16, 6400, 800, '2022-01-15');
insert into sales values (2003, 9003, 'Stayed', '2022-01-05', '2022-01-07', 1, 25, 3500, 600, '2022-01-01');
insert into sales values (2004, 9004, 'No Show', '2022-01-08', '2022-01-09', 1, 44, 2800, 400, '2022-01-01');
insert into sales values (2005, 9005, 'Cancelled', '2022-02-10', '2022-02-12', 1, 56, 3700, 700, '2022-01-20');
insert into sales values (2006, 9006, 'Stayed', '2022-01-15', '2022-01-18', 2, 68, 7500, 900, '2022-01-10');
insert into sales values (2007, 9007, 'Stayed', '2022-01-20', '2022-01-22', 1, 106, 4000, 500, '2022-01-01');
insert into sales values (2008, 9008, 'Cancelled', '2022-02-01', '2022-02-03', 1, 117, 3200, 600, '2022-01-20');
insert into sales values (2009, 9009, 'Stayed', '2022-01-25', '2022-01-26', 1, 164, 3300, 450, '2022-01-10');
insert into sales values (2010, 9010, 'Stayed', '2022-03-01', '2022-03-03', 2, 186, 8000, 1000, '2022-02-15');
insert into sales values (2011, 9011, 'Cancelled', '2022-01-14', '2022-01-16', 1, 207, 3100, 500, '2022-01-05');
insert into sales values (2012, 9012, 'Stayed', '2022-03-05', '2022-03-06', 1, 219, 2900, 350, '2022-02-20');
insert into sales values (2013, 9013, 'No Show', '2022-03-10', '2022-03-11', 1, 226, 4100, 600, '2022-02-25');
insert into sales values (2014, 9014, 'Stayed', '2022-02-11', '2022-02-13', 2, 232, 6200, 720, '2022-02-01');
insert into sales values (2015, 9015, 'Cancelled', '2022-01-18', '2022-01-19', 1, 346, 3500, 420, '2022-01-05');
insert into sales values (2016, 9016, 'Stayed', '2022-02-20', '2022-02-23', 2, 409, 7800, 980, '2022-02-10');
insert into sales values (2017, 9017, 'No Show', '2022-03-12', '2022-03-13', 1, 449, 2400, 300, '2022-03-01');
insert into sales values (2018, 9018, 'Stayed', '2022-02-14', '2022-02-16', 1, 579, 3500, 480, '2022-02-01');
insert into sales values (2019, 9019, 'Cancelled', '2022-01-12', '2022-01-14', 1, 584, 4000, 560, '2022-01-01');
insert into sales values (2020, 9020, 'Stayed', '2022-02-15', '2022-02-17', 1, 800, 4300, 690, '2022-01-25');

select * from sales;
```

Results Messages

	booking_id	customer_id	bking_status	check_in	check_out	no_of_rooms	hotel_id	amount	discount	date_of_booking
1	2001	9001	Cancelled	2022-01-10	2022-01-12	1	3	3000	500	2022-01-01
2	2002	9002	Stayed	2022-02-01	2022-02-03	2	16	6400	800	2022-01-15
3	2003	9003	Stayed	2022-01-05	2022-01-07	1	25	3500	600	2022-01-01
4	2004	9004	No Show	2022-01-08	2022-01-09	1	44	2800	400	2022-01-01
5	2005	9005	Cancelled	2022-02-10	2022-02-12	1	56	3700	700	2022-01-20
6	2006	9006	Stayed	2022-01-15	2022-01-18	2	68	7500	900	2022-01-10
7	2007	9007	Stayed	2022-01-20	2022-01-22	1	106	4000	500	2022-01-01
8	2008	9008	Cancelled	2022-02-01	2022-02-03	1	117	3200	600	2022-01-20
9	2009	9009	Stayed	2022-01-25	2022-01-26	1	164	3300	450	2022-01-10
10	2010	9010	Stayed	2022-03-01	2022-03-03	2	186	8000	1000	2022-02-15
11	2011	9011	Cancelled	2022-01-14	2022-01-16	1	207	3100	500	2022-01-05
12	2012	9012	Stayed	2022-03-05	2022-03-06	1	219	2900	350	2022-02-20
13	2013	9013	No Show	2022-03-10	2022-03-11	1	226	4100	600	2022-02-25
14	2014	9014	Stayed	2022-02-11	2022-02-13	2	232	6200	720	2022-02-01
15	2015	9015	Cancelled	2022-01-18	2022-01-19	1	346	3500	420	2022-01-05
16	2016	9016	Stayed	2022-02-20	2022-02-23	2	409	7800	980	2022-02-10
17	2017	9017	No Show	2022-03-12	2022-03-13	1	449	2400	300	2022-03-01
18	2018	9018	Stayed	2022-02-14	2022-02-16	1	579	3500	480	2022-02-01
19	2019	9019	Cancelled	2022-01-12	2022-01-14	1	584	4000	560	2022-01-01
20	2020	9020	Stayed	2022-02-15	2022-02-17	1	800	4300	690	2022-01-25

ii . Cities table:

```
create table cities
(hotel_id int primary key,
city varchar(10) not null check (city in('Gurgaon', 'Delhi', 'Noida', 'Bangalore', 'Mumbai', 'Jaipur', 'Pune', 'Chennai')));

insert into cities(hotel_id, city) values (3,'Gurgaon'),
(16, 'Gurgaon'),
(25, 'Delhi'),
(44, 'Noida'),
(56, 'Bangalore'),
(68, 'Mumbai'),
(106, 'Jaipur'),
(117, 'Pune'),
(164, 'Chennai'),
(186, 'Gurgaon'),
(207, 'Gurgaon'),
(219, 'Delhi'),
(226, 'Mumbai'),
(232, 'Noida'),
(346, 'Delhi'),
(409, 'Bangalore'),
(449, 'Jaipur'),
(579, 'Chennai'),
(584, 'Bangalore'),
(800, 'Mumbai');

select * from cities;
```

Results Messages

	hotel_id	city
1	3	Gurgaon
2	16	Gurgaon
3	25	Delhi
4	44	Noida
5	56	Bangalore
6	68	Mumbai
7	106	Jaipur
8	117	Pune
9	164	Chennai
10	186	Gurgaon
11	207	Gurgaon
12	219	Delhi
13	226	Mumbai
14	232	Noida
15	346	Delhi
16	409	Bangalore
17	449	Jaipur
18	579	Chennai
19	584	Bangalore
20	800	Mumbai

Questions:

1 . Find the average room rates in different cities

```
select c.city, avg(s.amount) as avg_room_rate from cities c join sales s on
c.hotel_id = s.hotel_id group by c.city order by avg_room_rate desc;
```

Results Messages

	city	avg_room_rate
1	Mumbai	5300
2	Bangalore	5166
3	Gurgaon	5125
4	Noida	4500
5	Chennai	3400
6	Delhi	3300
7	Jaipur	3200
8	Pune	3200

2 . Find the no. of bookings of different cities in Jan Feb March months

```
select city, count(*) as no_of_booking from sales s join cities c on s.hotel_id =
c.hotel_id where month(s.date_of_booking) in (1, 2, 3) group by city order by
no_of_booking desc;
```

Results Messages

	city	no_of_booking
1	Gurgaon	4
2	Bangalore	3
3	Mumbai	3
4	Delhi	3
5	Noida	2
6	Chennai	2
7	Jaipur	2
8	Pune	1

3 . Find the frequency of early booking prior to check-in to the hotel

```
select datediff(day, date_of_booking, check_in) as days_prior_checkin, count(*) as
frequecy_of_early_booking from sales group by DATEDIFF(day, date_of_booking,
check_in) order by days_prior_checkin;
```

Results Messages

	days_prior_checkin	frequecy_of_early_booking
1	4	1
2	5	1
3	7	1
4	9	2
5	10	2
6	11	2
7	12	1
8	13	4
9	14	1
10	15	1
11	17	1
12	19	1
13	21	2

4. Find the frequency of booking of no. of rooms in the hotel

```
select no_of_rooms, count(*) as frequency_of_booking from sales group by
no_of_rooms order by no_of_rooms;
```

Results Messages

	no_of_rooms	frequency_of_booking
1	1	15
2	2	5

5. Write a query to find the new customers in the month of January

```
select customer_id, min(date_of_booking) as first_booking_date from sales where
month(date_of_booking) = 1 group by customer_id order by customer_id;
```

Results Messages

	customer_id	first_booking_date
1	9001	2022-01-01
2	9002	2022-01-15
3	9003	2022-01-01
4	9004	2022-01-01
5	9005	2022-01-20
6	9006	2022-01-10
7	9007	2022-01-01
8	9008	2022-01-20
9	9009	2022-01-10
10	9011	2022-01-05
11	9015	2022-01-05
12	9019	2022-01-01
13	9020	2022-01-25

6. Find the net revenue to company due to some cancelled bookings

```
select sum(amount - discount) as net_revenue from sales where bking_status =
'Stayed' or bking_status = 'No Show';
```

Results Messages

	net_revenue
1	57930

7. Find the gross revenue to the company

```
select sum(amount-discount) as gross_revenue from sales;
```

Results Messages

	gross_revenue
1	75150