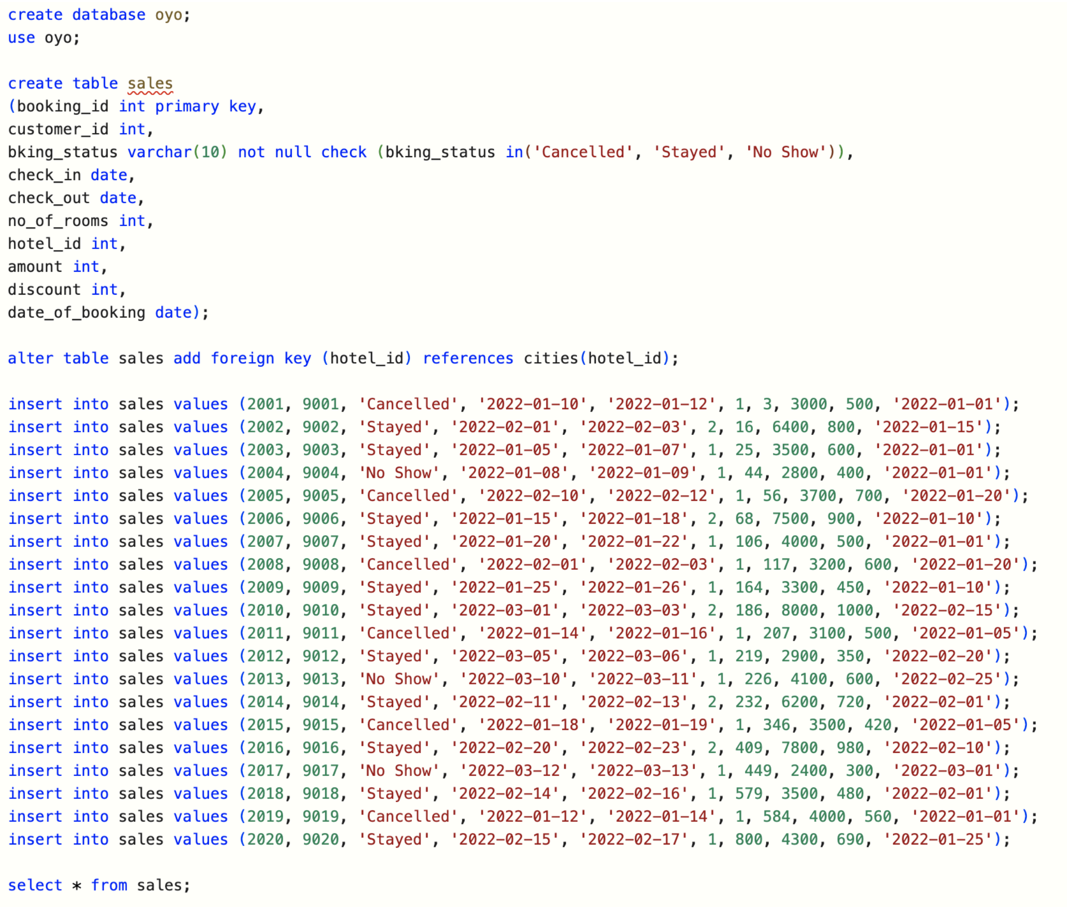
Coding Assessment:

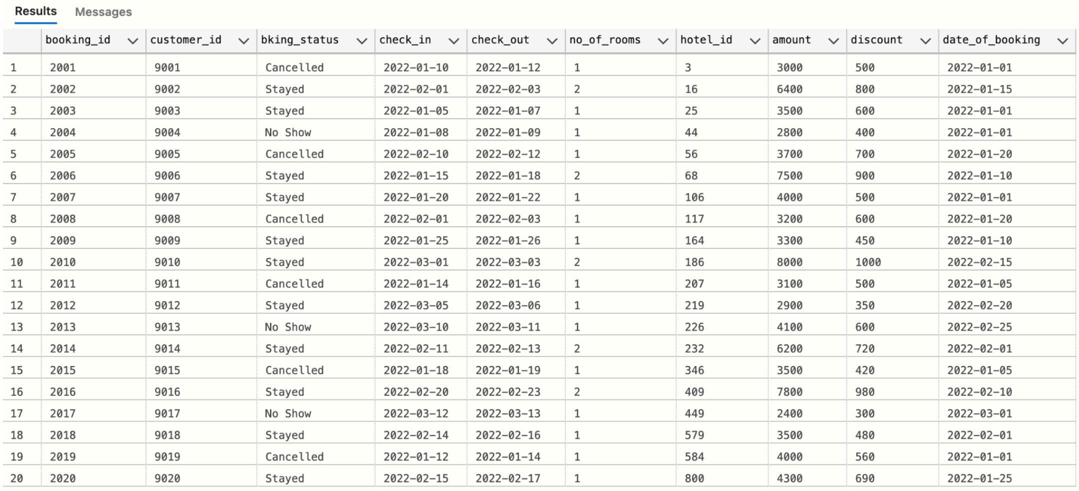
by Esaq

Creating Tables:

i . Sales Table:

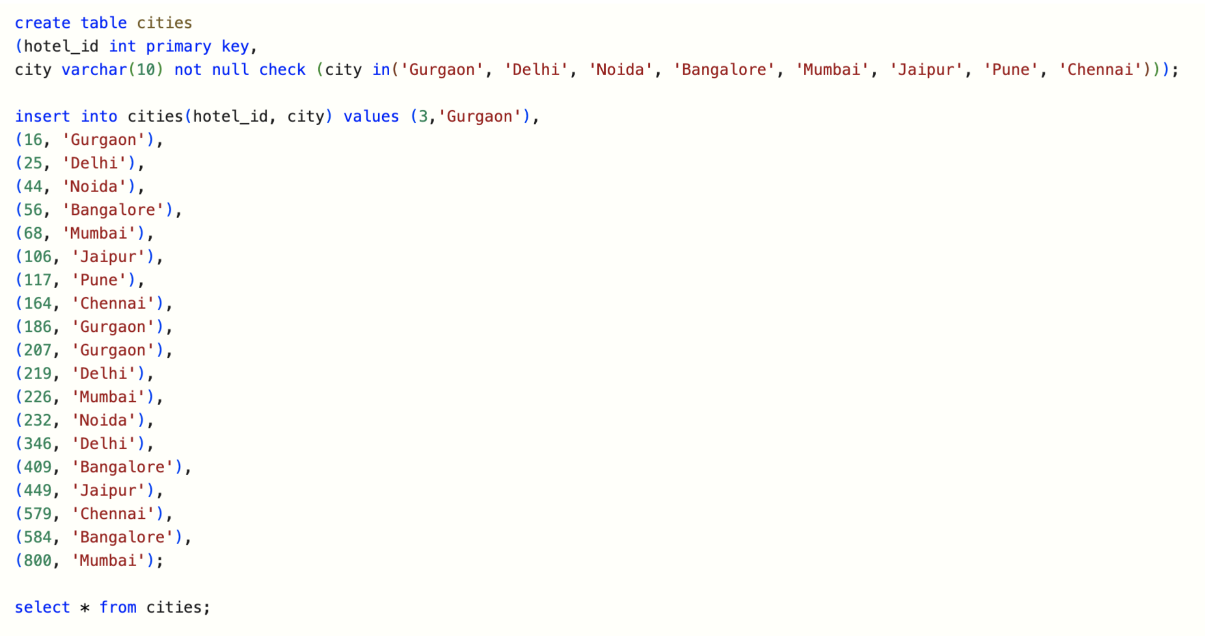


```
create database oyo;
use oyo;

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;
```

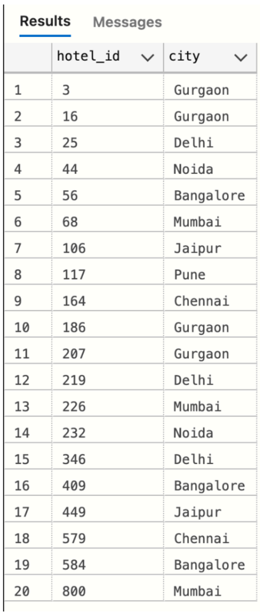


	booking_id	customer_id	bking_status	check_in	check_out	no_of_rooms	hotel_id	amount	discount	date_of_booking
1	2001	9001	Cancelled	2022-01-10	2022-01-12	1	3	3000	500	2022-01-01
2	2002	9002	Stayed	2022-02-01	2022-02-03	2	16	6400	800	2022-01-15
3	2003	9003	Stayed	2022-01-05	2022-01-07	1	25	3500	600	2022-01-01
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;
```

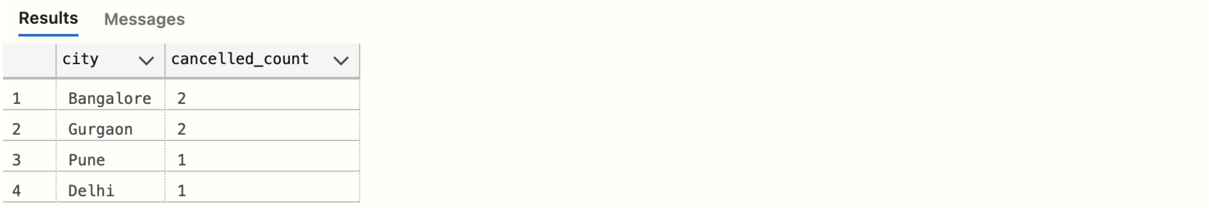


	hotel_id	city
1	3	Gurgaon
2	16	Gurgaon
3	25	Delhi
4	44	Noida
5	56	Bangalore
6	68	Mumbai
7	106	Jaipur
8	117	Pune
9	164	Chennai
10	186	Gurgaon
11	207	Gurgaon
12	219	Delhi
13	226	Mumbai
14	232	Noida
15	346	Delhi
16	409	Bangalore
17	449	Jaipur
18	579	Chennai
19	584	Bangalore
20	800	Mumbai

Questions:

1 . Find the frequency of cancellation in each city

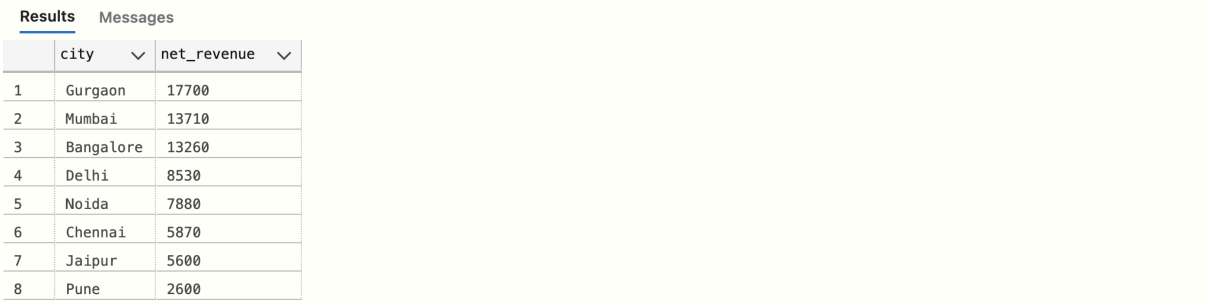
```
select c.city, count(*) as cancelled_count from cities c join sales s on c.hotel_id
= s.hotel_id where s.bking_status = 'Cancelled' group by c.city order by
cancelled_count desc;
```



	city	cancelled_count
1	Bangalore	2
2	Gurgaon	2
3	Pune	1
4	Delhi	1

2 . Find the net revenue from each city to the company.

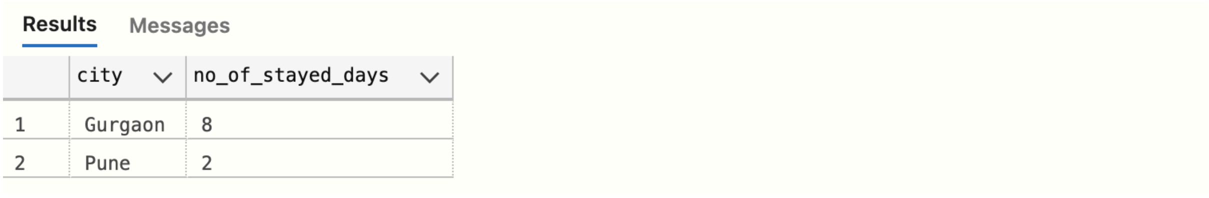
```
select c.city, sum(s.amount-s.discount) as net_revenue from cities c join sales s
on c.hotel_id = s.hotel_id group by c.city order by net_revenue desc;
```



	city	net_revenue
1	Gurgaon	17700
2	Mumbai	13710
3	Bangalore	13260
4	Delhi	8530
5	Noida	7880
6	Chennai	5870
7	Jaipur	5600
8	Pune	2600

3 . Comparing the no. of days of stay in Pune and Gurgaon

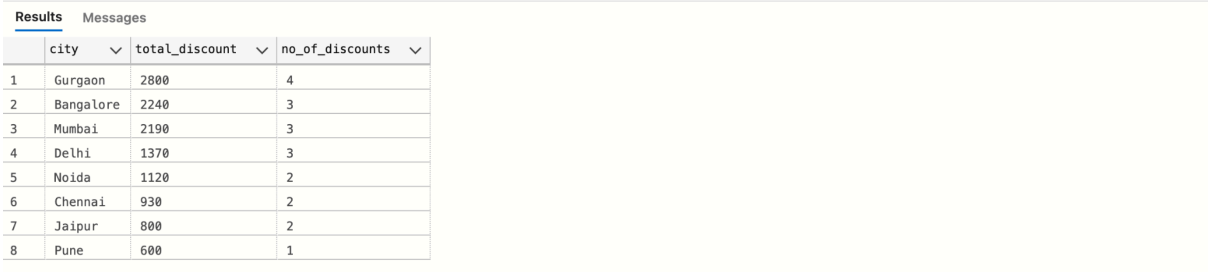
```
select c.city, sum(DATEDIFF(day, s.check_in, s.check_out)) as no_of_stayed_days
from cities c join sales s on s.hotel_id = c.hotel_id where c.city in ('Pune',
'Gurgaon') group by c.city order by no_of_stayed_days desc;
```



	city	no_of_stayed_days
1	Gurgaon	8
2	Pune	2

4. Find the overall discount given by each city and its count.

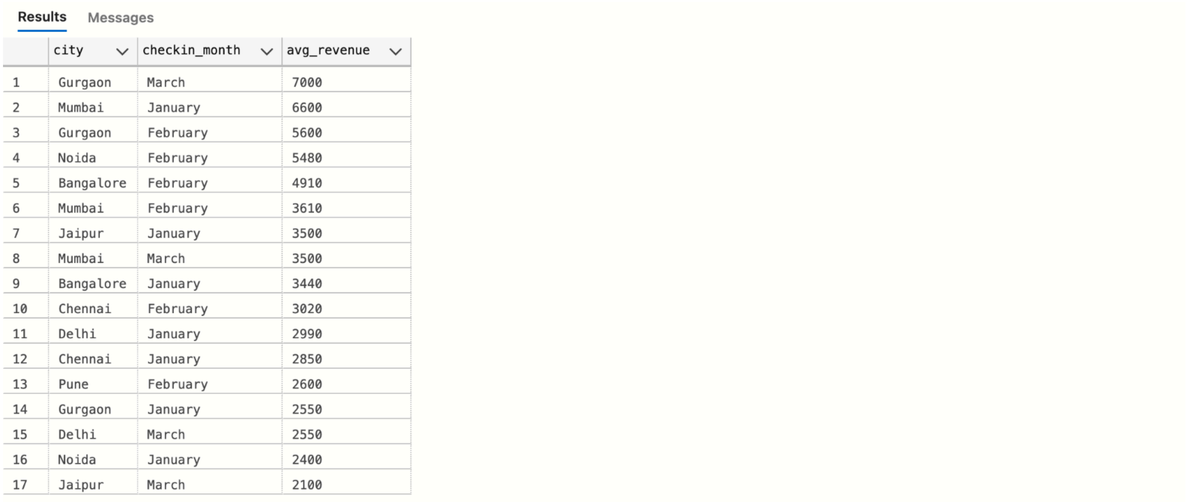
```
select c.city, sum(s.discount) as total_discount, COUNT(*) as no_of_discounts from
cities c join sales s on c.hotel_id = s.hotel_id group by c.city order by
total_discount desc;
```



	city	total_discount	no_of_discounts
1	Gurgaon	2800	4
2	Bangalore	2240	3
3	Mumbai	2190	3
4	Delhi	1370	3
5	Noida	1120	2
6	Chennai	930	2
7	Jaipur	800	2
8	Pune	600	1

5 . Find the month-wise revenue from each city.

```
select c.city, datename(month, s.check_in) as checkin_month,
avg(s.amount-s.discount) as avg_revenue from sales s join cities c on s.hotel_id =
c.hotel_id group by c.city, DATENAME(month, s.check_in) order by avg_revenue desc;
```



	city	checkin_month	avg_revenue
1	Gurgaon	March	7000
2	Mumbai	January	6600
3	Gurgaon	February	5600
4	Noida	February	5480
5	Bangalore	February	4910
6	Mumbai	February	3610
7	Jaipur	January	3500
8	Mumbai	March	3500
9	Bangalore	January	3440
10	Chennai	February	3020
11	Delhi	January	2990
12	Chennai	January	2850
13	Pune	February	2600
14	Gurgaon	January	2550
15	Delhi	March	2550
16	Noida	January	2400
17	Jaipur	March	2100