**TOURS AND TRAVELS MANAGEMENT SYSTEM**

 **INTRODUCTION: -**

Nilesh Tours and travels is a travel company that provides various types of travel packages and services to the customers. Tours and travels management system is software system where the management of entire company is computerized. The tours and travels management system are designed using secured database. In this project the details are maintained like customers, seller, package, booking details. The company wants to track its customers bookings and generate reports to analyze its business performance.

**AIM OF PROJECT: -**

The aim of this project is to design and implement a Tours and Travels management system using SQL. The system stores and manages data about customers & sellers. The system will also provide various reports and queries to help the salesman and managers perform their tasks efficiently and effectively.

**OBJECTIVE OF PROJECT: -**

* To design a relational database for the tours and travels management system using SQL.
* To generate various reports and queries to support the decision making of the tours salesman and managers
* To document the system design, implementation, and testing process.

**This database contains four tables: -**

* Customers
* Packages
* Booking
* Seller

|  |
| --- |
| CUSTOMER |
| PK cust\_id int |
| Firstname varchar(25) |
| Lastname Varchar(30) |
| City Varchar(20) |
| Pincode int |
| Mob\_NO Bigint |
| Email\_id varchar(30) |
| DOB Date |

**ER DIAGRAM**

One to one

|  |
| --- |
| SELLER |
| PK Seller\_id int |
| S\_firstname varchar(30) |
| S\_lastname varchar(30) |
| S\_city varchar(20) |
| Mob\_no bigint |
| Email\_id varchar(30) |
| DOB date |
| Designation varchar(30) |
| Sal int |

|  |
| --- |
| BOOKING |
| PK Booking\_id int |
| Fk tous\_id int |
| Fk seller\_id int |
| FK cust\_id int |
| No\_of\_people int |
| Booking\_date date |
| Payment\_mode varchar(30) |
| Payment\_status varchar(30) |

One to one

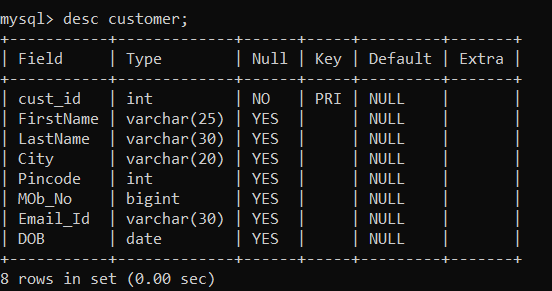
**Tours and Travels**

One to one

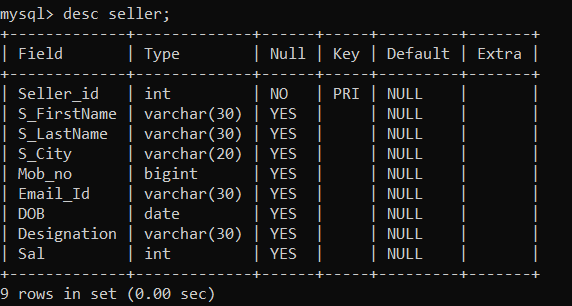
|  |
| --- |
| PACKAGE |
| PK Tour\_id int |
| Tour\_name varchar(200) |
| Duration int |
| Price int |
| Starting\_date Date |
| Ending\_date Date |
| Services varchar(250) |
| Max\_capacity int |

**STRUCTURE OF TABLES: -**

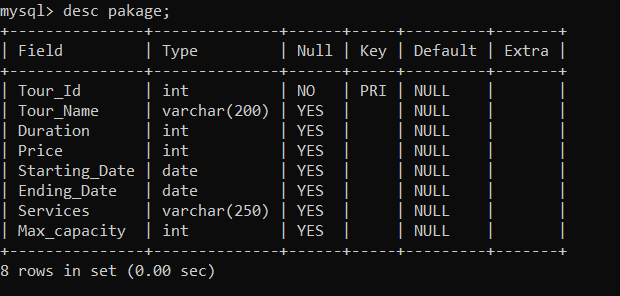
1.Customer: -



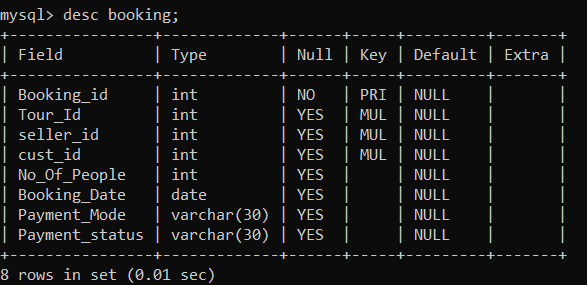
2.Seller: -



3.Pakage: -

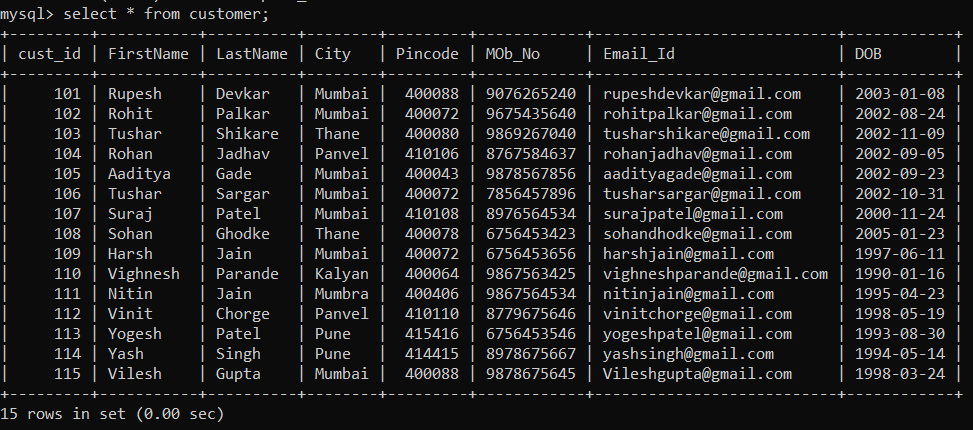


4.Booking: -

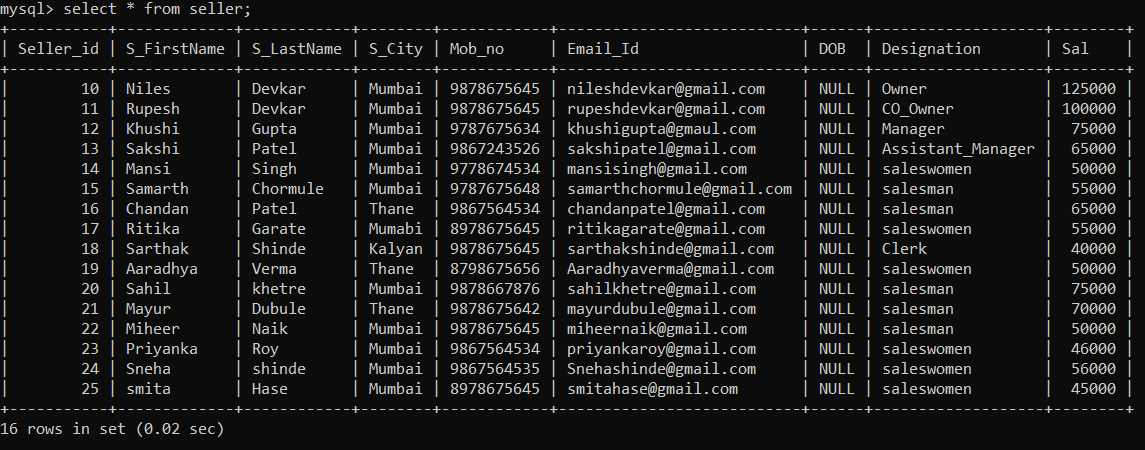


**CONTENT OF TABLES**

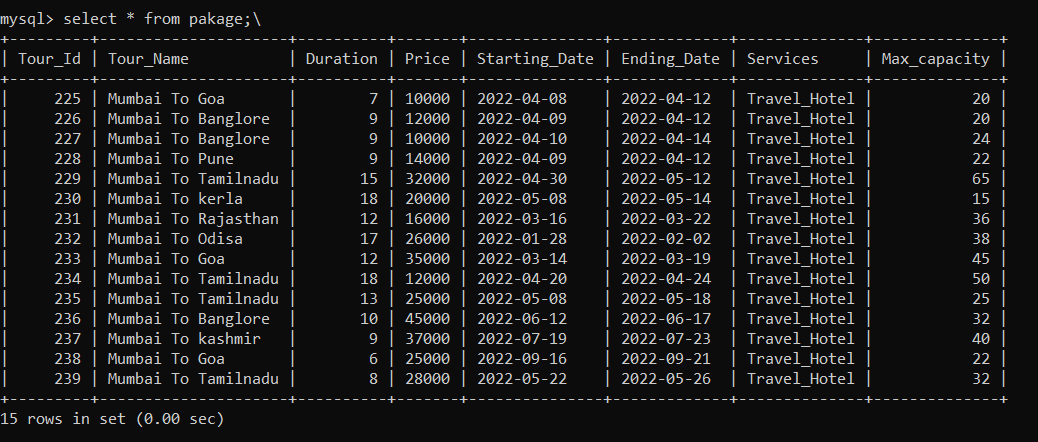
1.customer: -



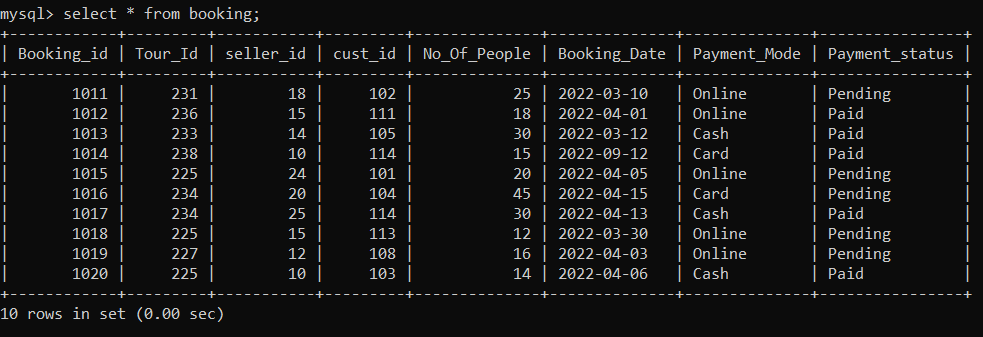
2.Seller: -



3. Package: -



4. Booking: -



**A) Alter: -**

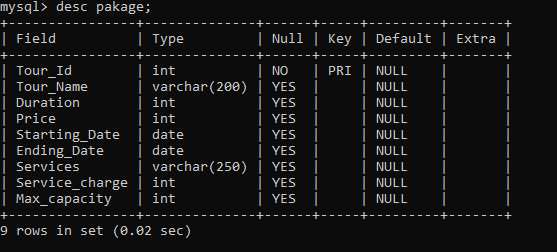
1. Add new column (service charge) after services

MySQL> alter table pakage

-> add column Service charge int after Services;

Query OK, 0 rows affected (0.08 sec)

Records: 0 Duplicates: 0 Warnings: 0



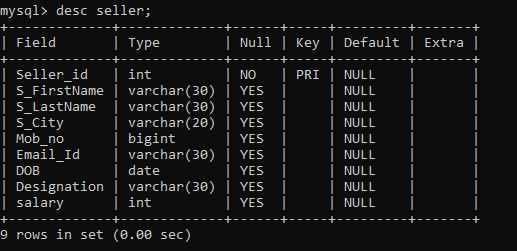
1. Rename column name Sal to salary

MySQL> alter table seller

-> rename column Sal to salary;

Query OK, 0 rows affected (0.03 sec)

Records: 0 Duplicates: 0 Warnings: 0

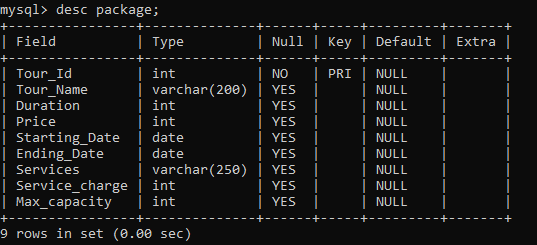


1. Rename table pakage to package

MySQL> alter table pakage

-> rename package;

Query OK, 0 rows affected (0.04 sec)



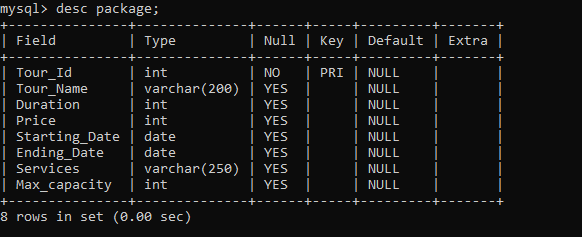
1. Drop column service charge

MySQL> alter table package

-> drop column service charge;

Query OK, 0 rows affected (0.03 sec)

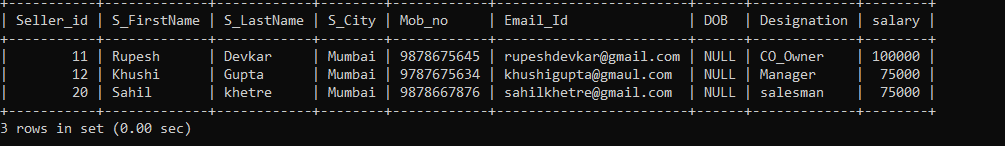
Records: 0 Duplicates: 0 Warnings: 0



**B) Between: -**

5.Display the details of sellers who have salary between 75000 to 100000

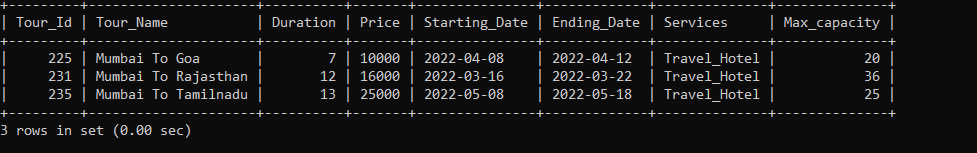
MySQL> select \* from seller where salary between 75000 and 100000;



**C) In: -**

6. Display the details of package where it is 225,231,235.

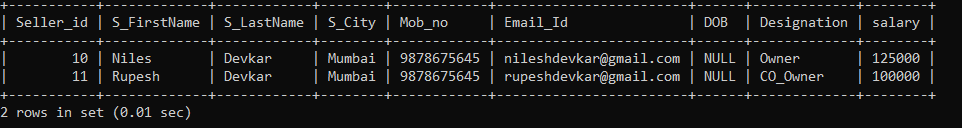
MySQL> select \* from package where tour \_id in (225,231,235);



**D) comparison operators, Where Claus: -**

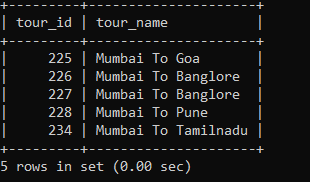
7. display the details of employee who have salary more than 75000.

MySQL> select \* from seller where salary > 75000;



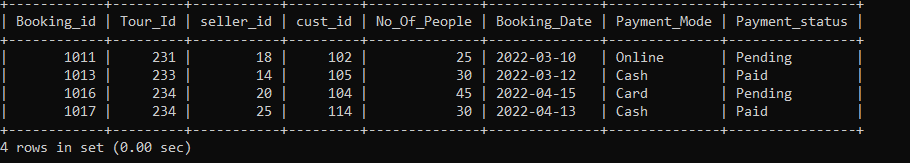
8. Display the details of packages where price is less than 15000.

MySQL> select tour\_id, tour\_name from package where price < 15000;



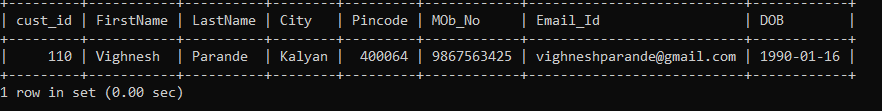
9. Display the details of bookings where 25 or more than 25 people have booked the tour.

select \* from booking where No\_Of\_people >=25;



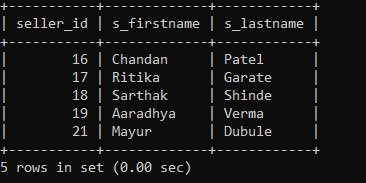
10.display the details of customer name Vighnesh.

select \* from customer where first name= "Vighnesh";



12. Display the details of seller who are not from Mumbai.

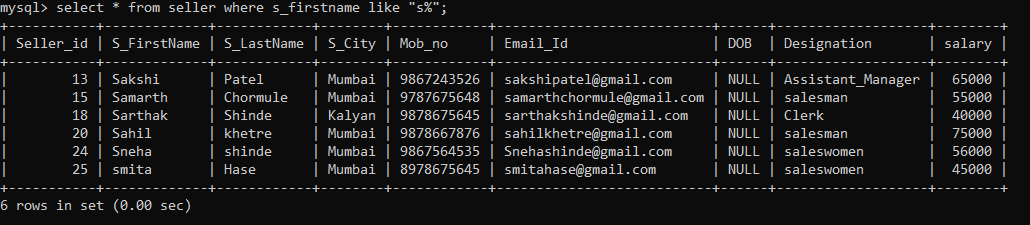
select seller\_id, s\_firstname, s\_lastname from seller where s\_city! ="Mumbai";



**E) Like: -**

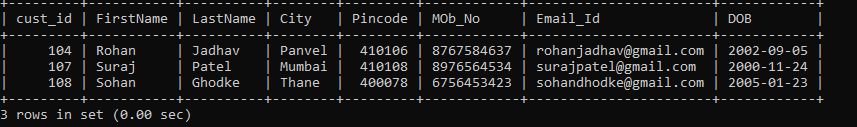
13. display the details of sellers who have first letter s.

select \* from seller where s\_firstname like "s%";



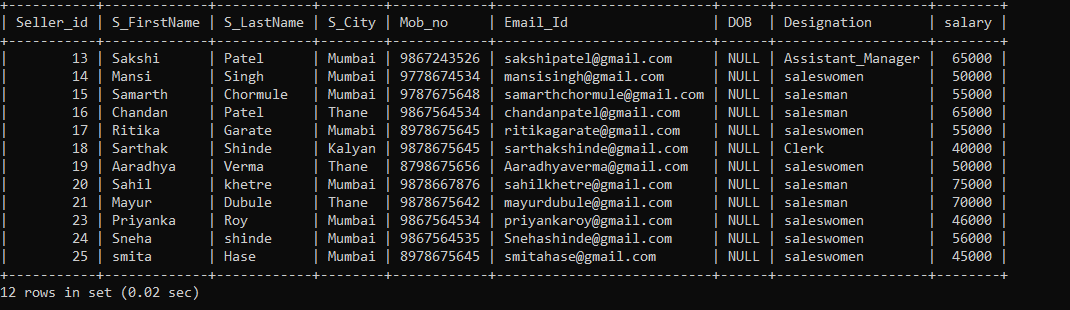
14. Display the details of seller where length of first letter is 5 and fourth letter of name is “a”.

MySQL> select \* from customer where length (first name) =5 and first name like"%\_\_\_a\_%";



15.display the details of seller who have “a, c, d” in their name.

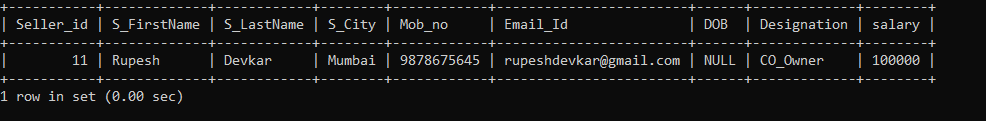
MySQL> select \* from seller where S\_firstname Rilke"[a/c/d]";



**F) Sub query: -**

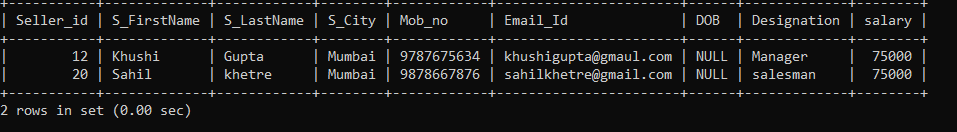
16. Display 2rd highest salary.

MySQL> select \* from seller where salary in (select max(salary) from seller where salary < (select max(salary) from seller));



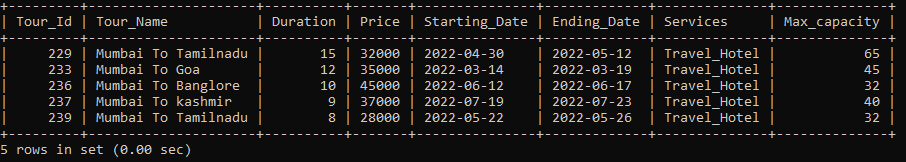
17. Display 3rd highest salary.

MySQL> select \* from seller where salary in (select max(salary) from seller where salary< (select max(salary) from seller where salary < (select max (Salary) from seller)));



18. Display details of package where price of package is more than “Mumbai to Odissa” tour.

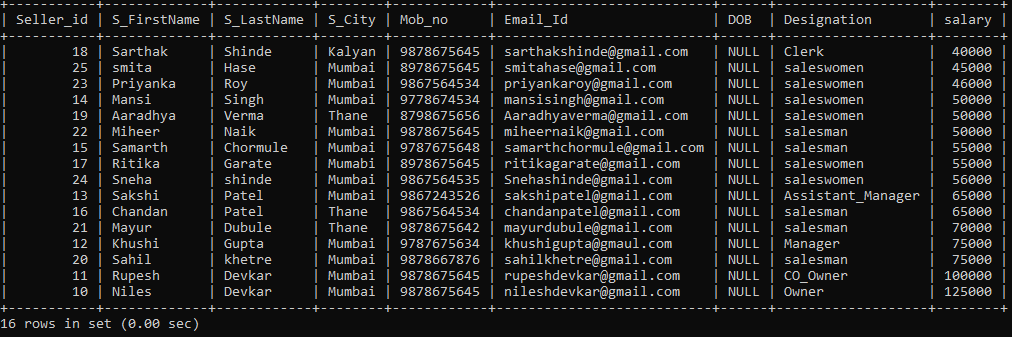
MySQL> select \* from package where price> (select price from package where tour\_name="Mumbai to Odissa");



**G) Order, group, having: -**

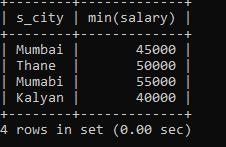
19.Display the salary by ascending order.

MySQL> select \* from seller order by salary;



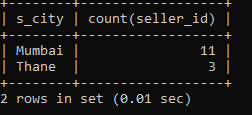
20. Display the details of min salary for each city.

select s\_city, min(salary) from seller group by s\_city;



21. Display the total count of seller working in each city there should be at least 2 sellers.

MySQL> select s\_city, count(seller\_id) from seller group by s\_city having count(s\_city)>2;



**H) Joins: -**

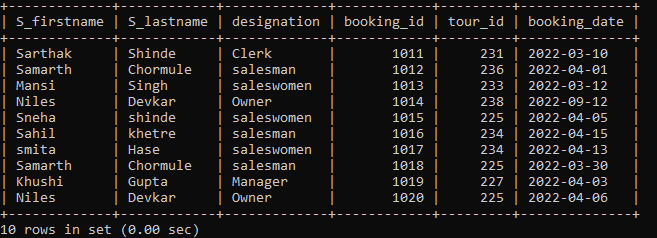
**22. Inner Join**

MySQL> select S\_firstname, S\_lastname, designation, booking\_id, tour\_id, booking date

-> from seller

-> inner join booking

-> on seller. seller\_id=booking. Seller\_id;

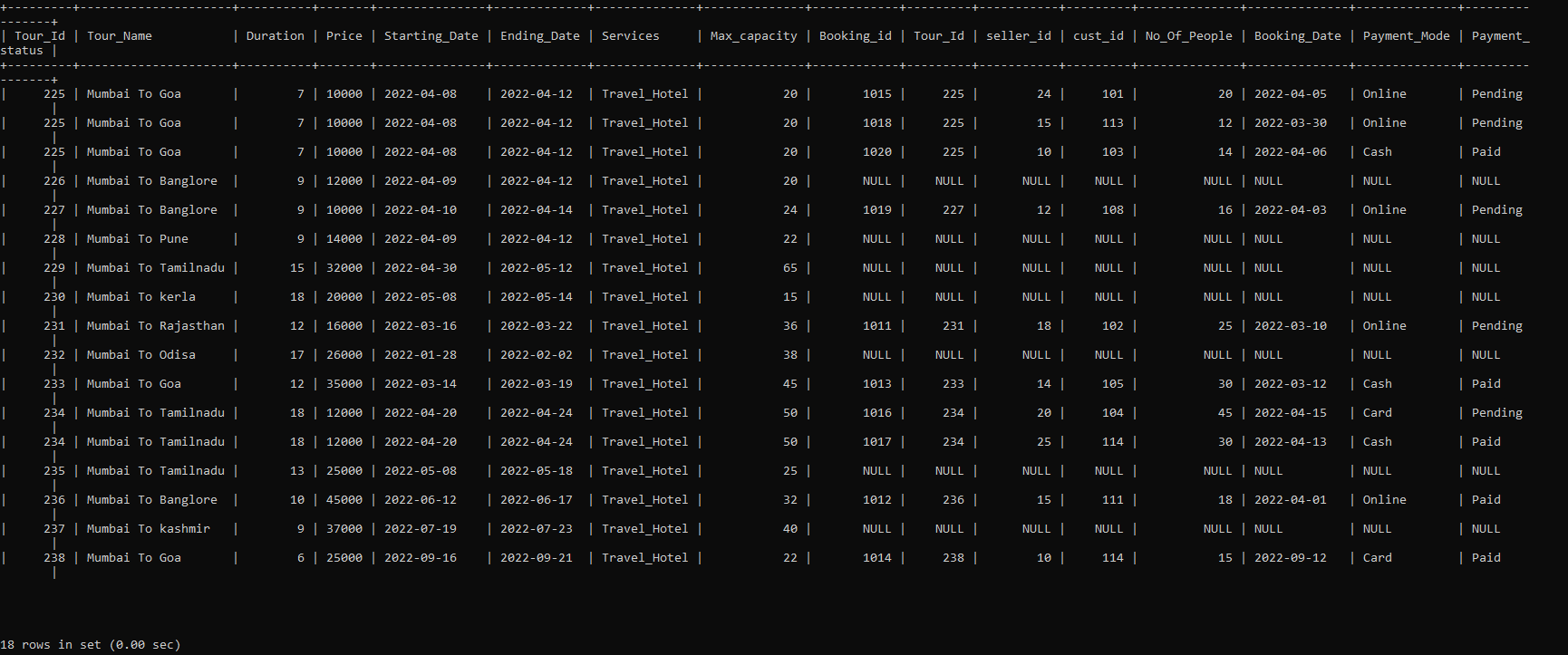


**23. Left Join**

MySQL> select \* from package

-> left join booking

-> on package. Tour\_id=booking. Tour\_id;

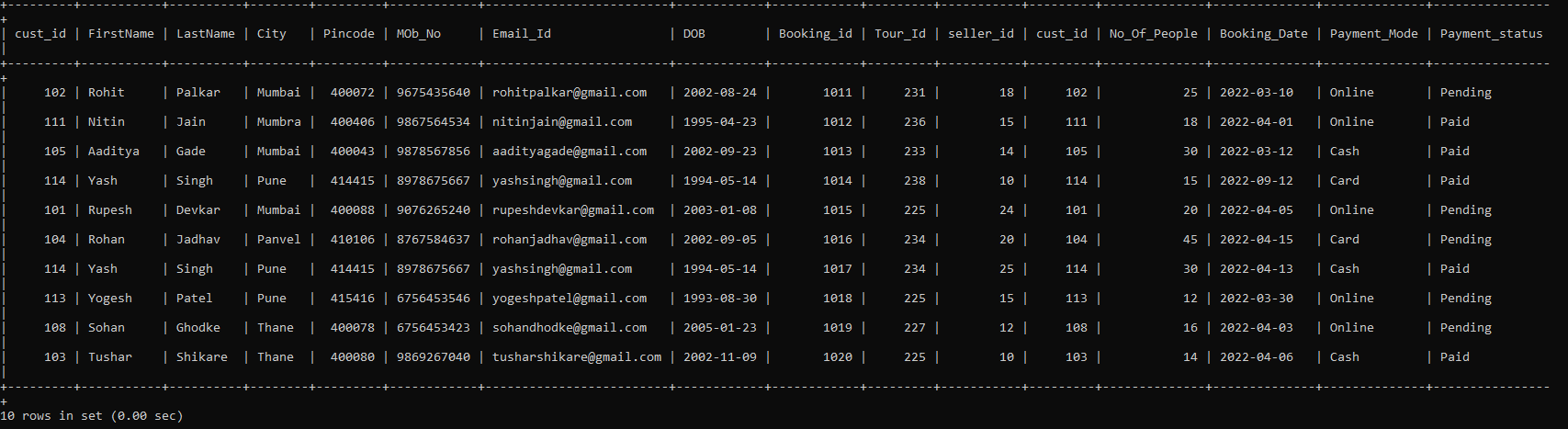


**24. Right Join**

MySQL> select \* from customer

-> right join booking

-> on customer. cystoid=booking. cust\_id;



25. Alias

mysql> select salary as sal,s\_firstname,S\_lastname from seller;

