# **Snowflake Basic Queries -- Solutions**

**============================**

--Create topgear database

create or replace database topgear;

--Create a warehouse topgear\_snowflake

create or replace warehouse topgear\_snowflake;

--Create a schema named topgear\_assignment

create or replace schema topgear\_assignment;

use database topgear;

use warehouse topgear\_snowflake;

show warehouses;

use schema topgear\_assignment;

/\*1. Write a SQL statement to Create below table.

Salesman\_idNameCityCommision

5001James HoogNew York0.15

5002Nail KniteParis0.13

5005Pit AlexLondon0.11

5006Mc LyonParis0.14

5003Lauson HenSydney0.12

5007Paul AdamRome0.13 \*/

**create or replace table salesman**

**(**

**Salesman\_id NUMBER(38),**

**Name VARCHAR(1024),**

**City VARCHAR(1024),**

**Commission NUMBER(38,2)**

**);**

//Insert above records to table Salesman

INSERT INTO topgear.topgear\_assignment.salesman values (5001,'James Hoog','New York',0.15);

INSERT INTO topgear.topgear\_assignment.salesman values (5002,'Nail Knite','Paris',0.13);

INSERT INTO topgear.topgear\_assignment.salesman values (5005,'Pit Alex','London',0.11);

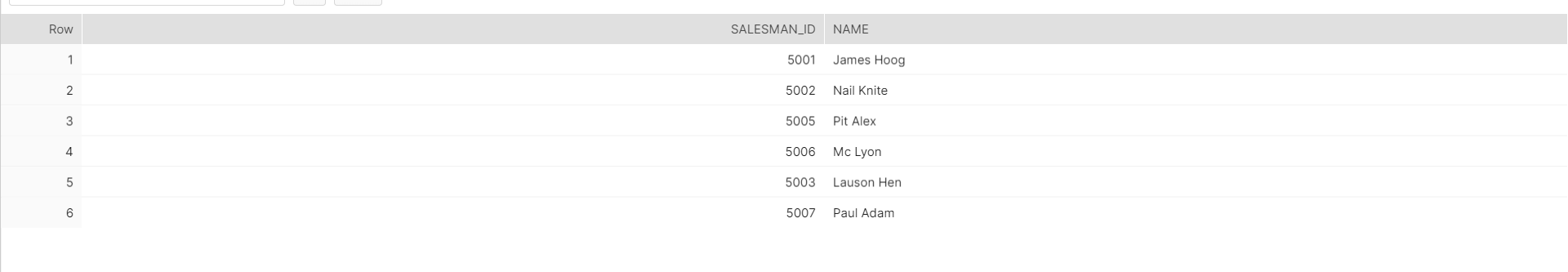
INSERT INTO topgear.topgear\_assignment.salesman values (5006,'Mc Lyon','Paris',0.14);

INSERT INTO topgear.topgear\_assignment.salesman values (5003,'Lauson Hen','Sydney',0.12);

INSERT INTO topgear.topgear\_assignment.salesman values (5007,'Paul Adam','Rome',0.13);

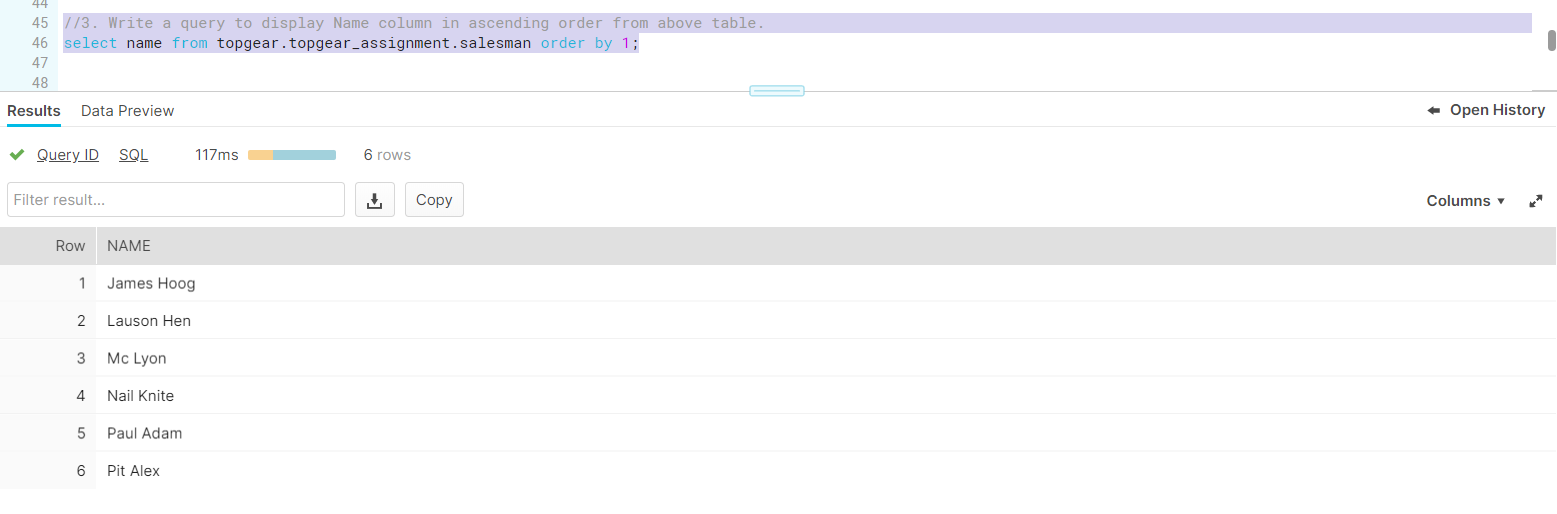
//2. Write a SQL statement to display specific columns like Salesman\_id and Name from above table.

**select salesman\_id,name from topgear.topgear\_assignment.salesman;**



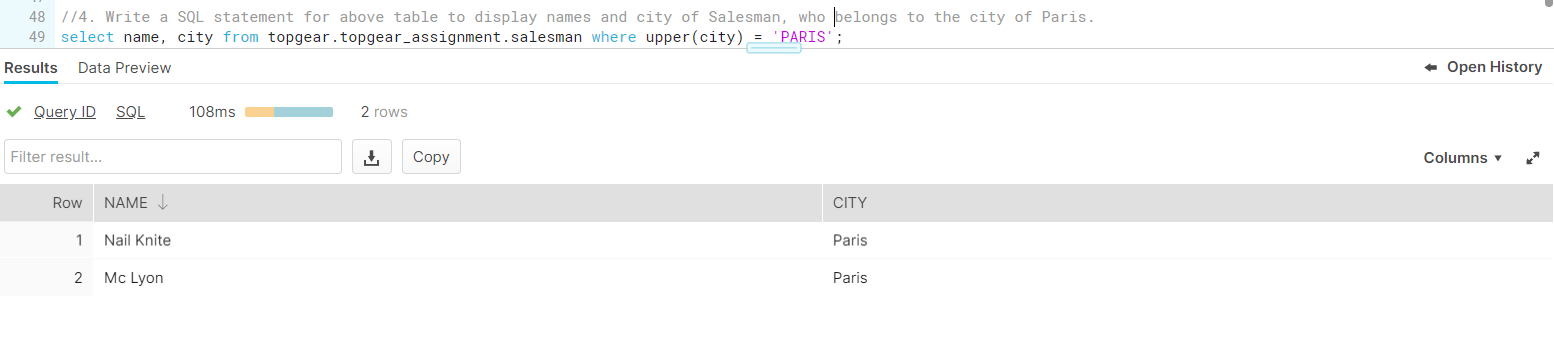
//3. Write a query to display Name column in ascending order from above table.

**select name from topgear.topgear\_assignment.salesman order by 1;**



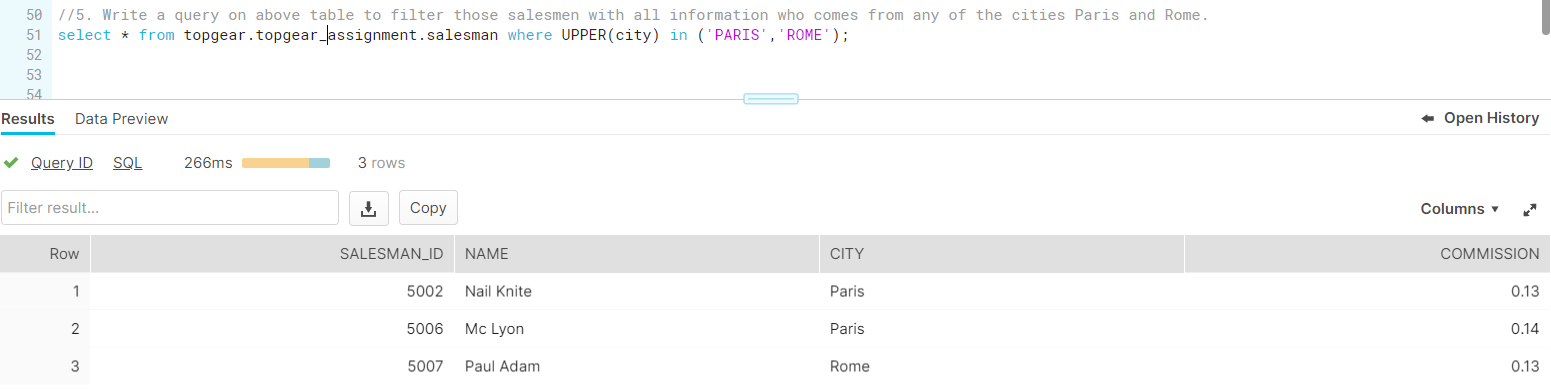
//4. Write a SQL statement for above table to display names and city of Salesman, who belongs to the city of Paris.

**select name, city from topgear.topgear\_assignment.salesman where upper(city) = 'PARIS';**



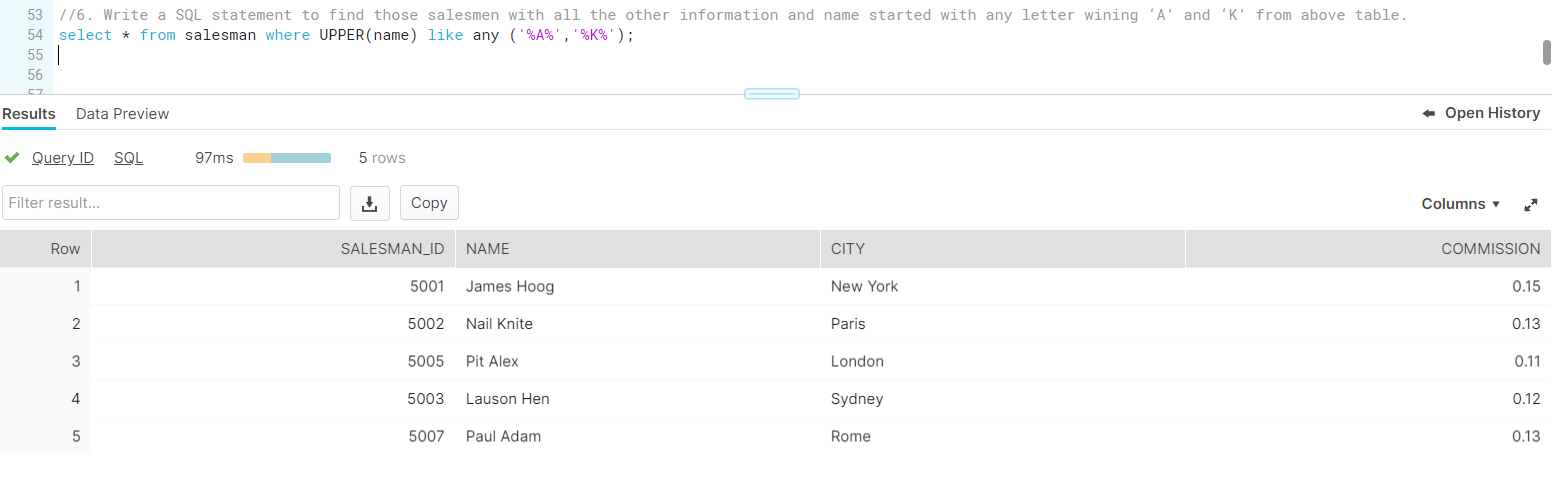
//5. Write a query on above table to filter those salesmen with all information who comes from any of the cities Paris and Rome.

**select \* from topgear.topgear\_assignment.salesman where UPPER(city) in ('PARIS','ROME');**



//6. Write a SQL statement to find those salesmen with all the other information and name started with any letter wining ‘A’ and ‘K’ from above table.

**select \* from salesman where UPPER(name) like any ('%A%','%K%');**



/\*7. Write a SQL statement to display all the information for those customers with a Grade of 200.

Customer\_idCust\_nameCityGradeSalesman\_id

3002Nick RimandoNew York1005001

3005Graham ZusiCalifornia2005002

3001Brad GuzanLondon5005

3004Fabian JohnsParis3005006

3007Brad DavisNew York2005001

3009Geoff CameroBerlin1005003

3008Julian GreenLondon3005002

3003Jozy AltidonMoscow2005007 \*/

create or replace table customer

(

Customer\_id NUMBER(38),

Cust\_name VARCHAR(1024),

City VARCHAR(1024),

Grade SMALLINT,

Salesman\_ID NUMBER(38)

);

insert into customer values (3002,'Nick Rimando','New York',100,5001);

insert into customer values (3005,'Graham Zusi','California',200,5002);

insert into customer values (3001,'Brad Guzan','London',NULL,5005);

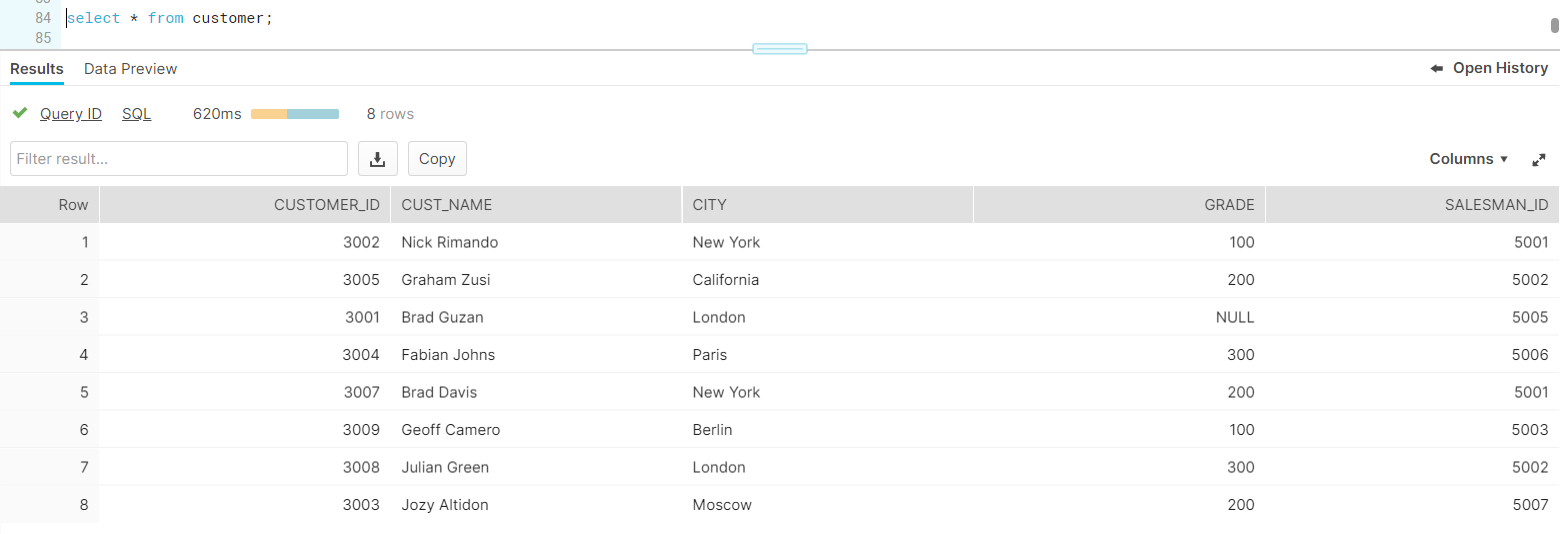
insert into customer values (3004,'Fabian Johns','Paris',300,5006);

insert into customer values (3007,'Brad Davis','New York',200,5001);

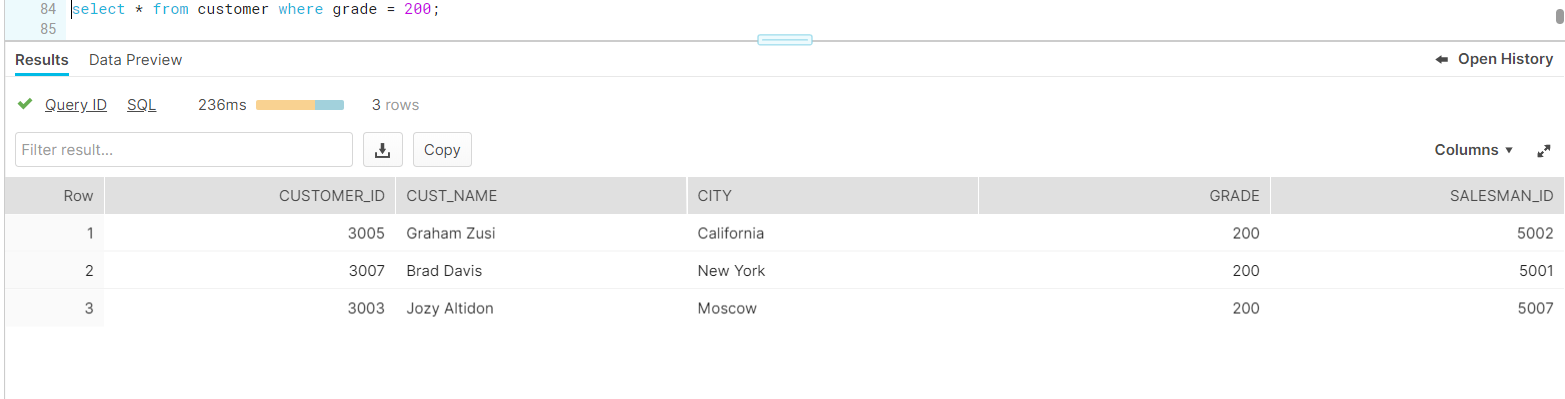
insert into customer values (3009,'Geoff Camero','Berlin',100,5003);

insert into customer values (3008,'Julian Green','London',300,5002);

insert into customer values (3003,'Jozy Altidon','Moscow',200,5007);



**select \* from customer where grade = 200;**



/\*8. Write a SQL query to calculate the average price of all products of the manufacturer which code is 16.

Pro\_IdPro\_NamePro\_PricePro\_com

101Mother Board320015

102Key Board45016

103Zip drive25014

104Speaker55016

105Monitor500011

106DVD drive90012

107CD drive80012

108Printer260013

109Refill cartridge35013 \*/

create or replace table products

(

Pro\_Id SMALLINT,

Pro\_Name VARCHAR(1024),

Pro\_Price NUMBER(38),

Pro\_com NUMBER(38)

);

insert into products values(101,'Mother Board',3200,15);

insert into products values(102,'Key Board',450,16);

insert into products values(103,'Zip drive',250,14);

insert into products values(104,'Speaker',550,16);

insert into products values(105,'Monitor',5000,11);

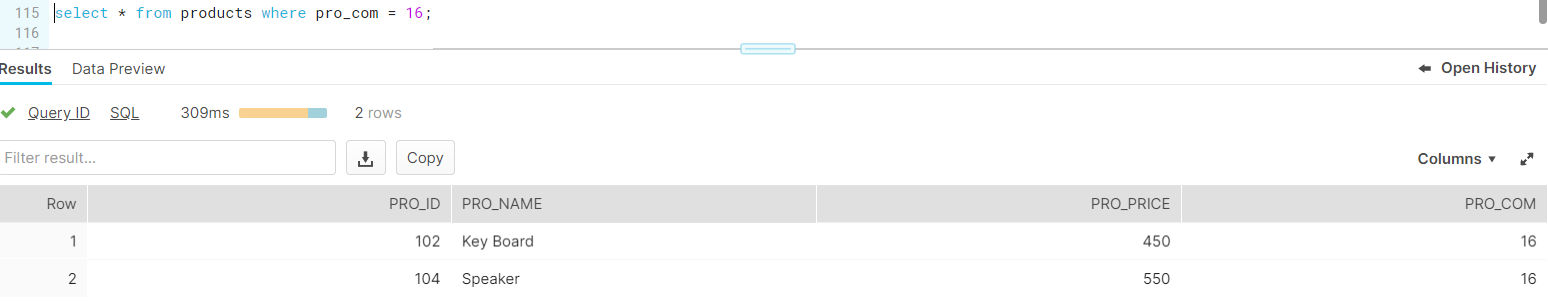
insert into products values(106,'DVD drive',900,12);

insert into products values(107,'CD drive',800,12);

insert into products values(108,'Printer',2600,13);

insert into products values(109,'Refill cartridge',350,13);

**select \* from products where pro\_com = 16;**



//9. Write a SQL query to find the name and price of the cheapest item from above table.

**select \* from products where pro\_price = (select min(pro\_price) from products);**



/\*10. Write a query in SQL to find the last name of all employees, without duplicates.

Emp\_IDNOEMP\_FnameEMP\_LnameEMP\_Dept

127323MichaleRobbin57

526689CarlosSnares63

843795EnricDasio57

328717JhonSnares63

444527JosephDosni47

659831ZaniferEmily47

847674KuleswarSitaraman57

748681HanreyGabriel47

555935AlexManuel57

539569GeorgeMardy27

733843MarioSaule63

631548AlanSnappy27

839139MariaFoster57 \*/

create or replace table employees

(

Emp\_IDNO SMALLINT,

EMP\_Fname VARCHAR(1024),

EMP\_Lname VARCHAR(1024),

EMP\_Dept SMALLINT

);

insert into employees values (127323,'Michale','Robbin',57);

insert into employees values (526689,'Carlos','Snares',63);

insert into employees values (843795,'Enric','Dasio',57);

insert into employees values (328717,'Jhon','Snares',63);

insert into employees values (444527,'Joseph','Dosni',47);

insert into employees values (659831,'Zanifer','Emily',47);

insert into employees values (847674,'Kuleswar','Sitaraman',57);

insert into employees values (748681,'Hanrey','Gabrie',147);

insert into employees values (555935,'Alex','Manuel',57);

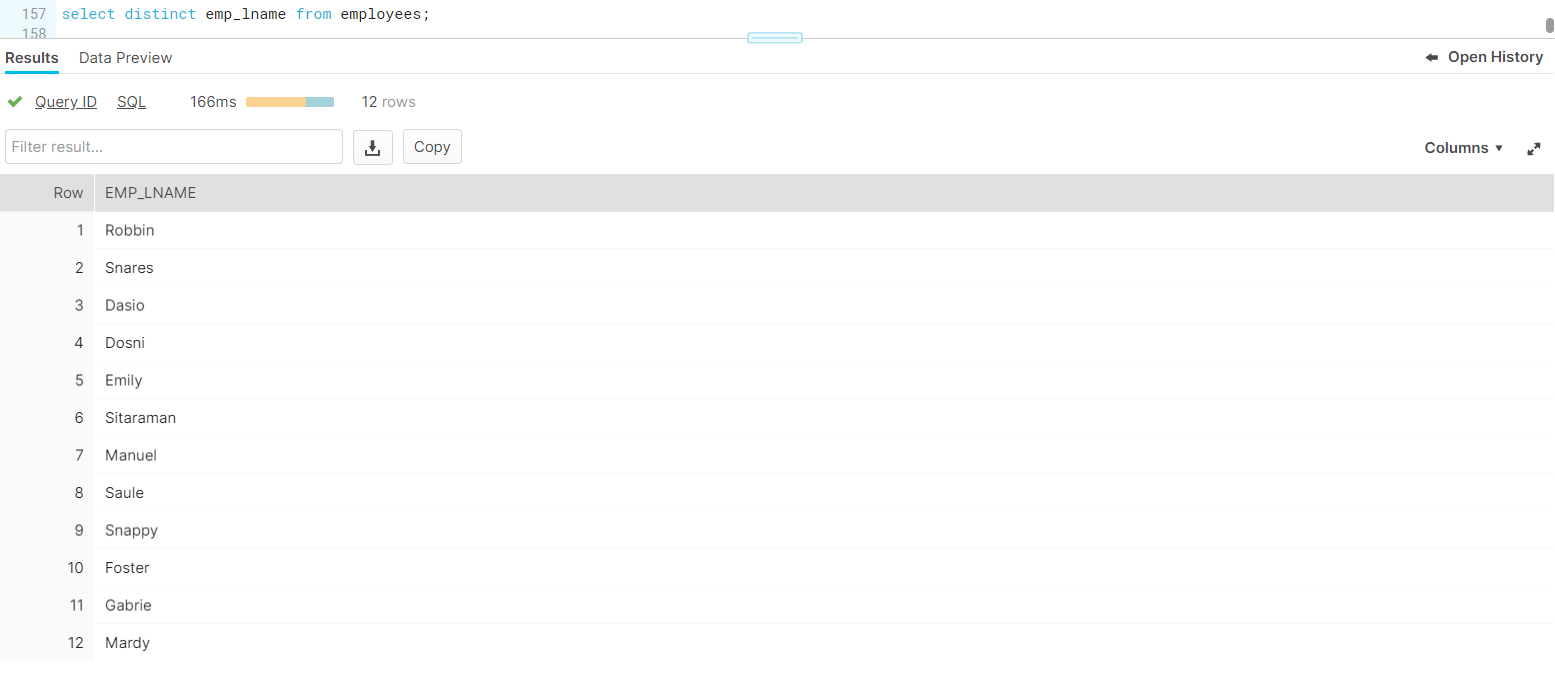
insert into employees values (539569,'George','Mardy',27);

insert into employees values (733843,'Mario','Saule',63);

insert into employees values (631548,'Alan','Snappy',27);

insert into employees values (839139,'Maria','Foster',57);

**select distinct emp\_lname from employees;**



/\* 11. Write a query to display all customers with a grade above 100

Customer\_idCust\_nameCityGradeSalesman\_id

3002Nick RimandoNew York1005001

3005Graham ZusiCalifornia2005002

3001Brad GuzanLondon5005

3004Fabian JohnsParis3005006

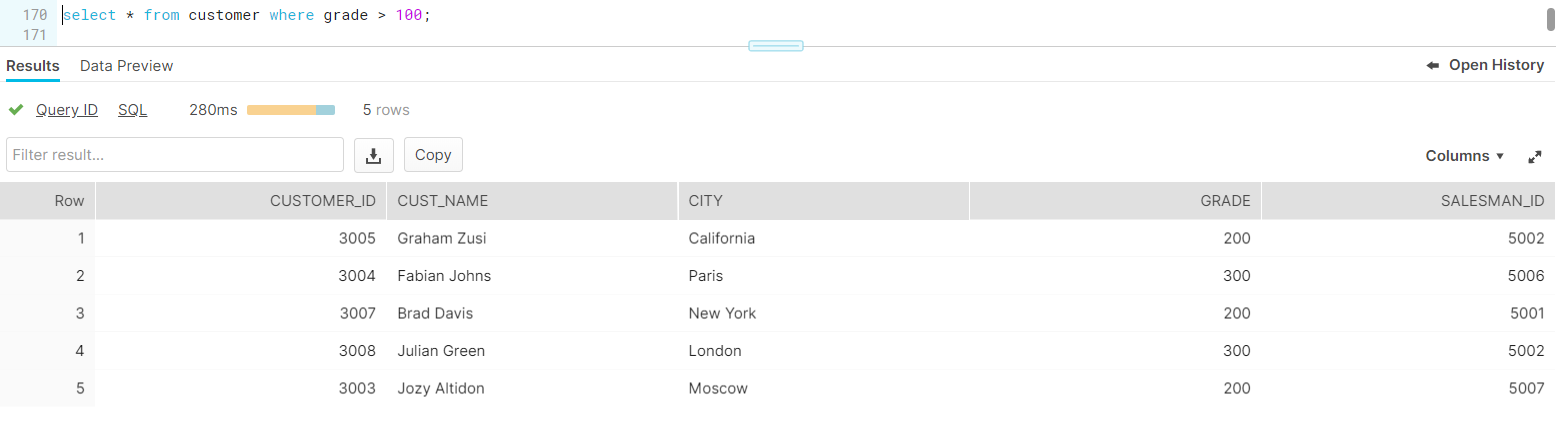
3007Brad DavisNew York2005001

3009Geoff CameroBerlin1005003

3008Julian GreenLondon3005002

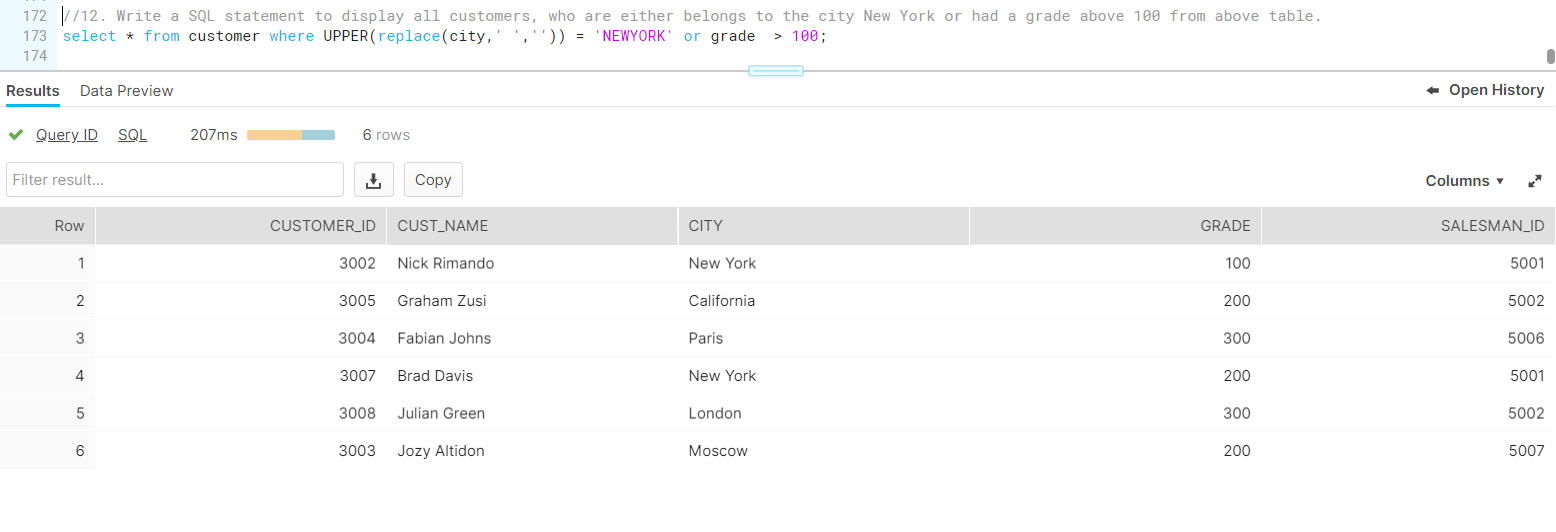
3003Jozy AltidonMoscow2005007 \*/

**select \* from customer where grade > 100;**



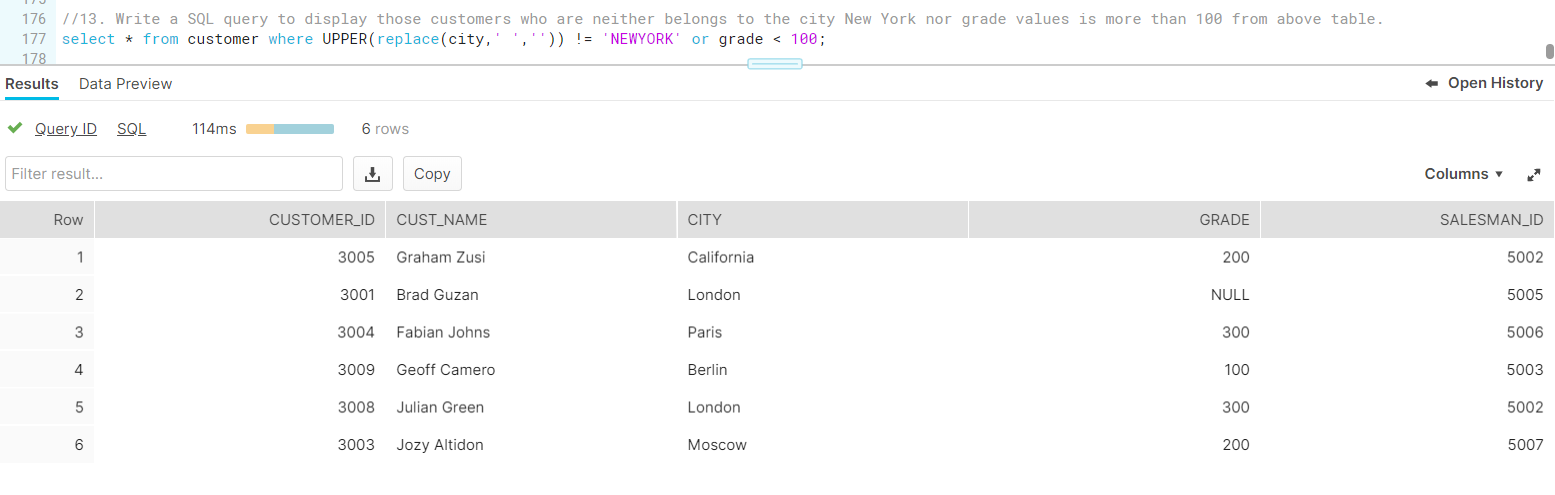
//12. Write a SQL statement to display all customers, who are either belongs to the city New York or had a grade above 100 from above table.

**select \* from customer where UPPER(replace(city,' ','')) = 'NEWYORK' or grade > 100;**



//13. Write a SQL query to display those customers who are neither belongs to the city New York nor grade values is more than 100 from above table.

**select \* from customer where UPPER(replace(city,' ','')) != 'NEWYORK' or grade < 100;**



/\* 14. Write a SQL statement to display either those orders which are not issued on date 2012-09-10 and issued by the salesman whose ID is 505 and below or those orders which purchase amount is 1000.00 and below.

ord\_nopurch\_amtord\_datecustmoer\_idsalesman\_id

70001150.52012-10-0530055002

70009270.652012-09-1030015005

7000265.262012-10-0530025001

70004110.52012-08-1730095003

70007948.52012-09-1030055002

700052400.62012-07-2730075001

7000857602012-09-1030025001

700101983.432012-10-1030045006

700032480.42012-10-1030095003

70012250.452012-06-2730085002

7001175.292012-08-1730035007

700133045.62012-04-2530025001 \*/

create or replace table orders

(

ord\_no SMALLINT,

purch\_amt NUMBER(38,2),

ord\_date DATE,

Customer\_id NUMBER(38),

salesman\_id NUMBER(38)

);

insert into orders values (70001,150.5,TO\_DATE('2012-10-05','YYYY-MM-DD'),3005,5002);

insert into orders values (70009,270.65,TO\_DATE('2012-09-10','YYYY-MM-DD'),3001,5005);

insert into orders values (70002,65.26,TO\_DATE('2012-10-05','YYYY-MM-DD'),3002,5001);

insert into orders values (70004,110.5,TO\_DATE('2012-08-17','YYYY-MM-DD'),3009,5003);

insert into orders values (70007,948.5,TO\_DATE('2012-09-10','YYYY-MM-DD'),3005,5002);

insert into orders values (70005,2400.6,TO\_DATE('2012-07-27','YYYY-MM-DD'),3007,5001);

insert into orders values (70008,5760,TO\_DATE('2012-09-10','YYYY-MM-DD'),3002,5001);

insert into orders values (70010,1983.43,TO\_DATE('2012-10-10','YYYY-MM-DD'),3004,5006);

insert into orders values (70003,2480.4,TO\_DATE('2012-10-10','YYYY-MM-DD'),3009,5003);

insert into orders values (70012,250.45,TO\_DATE('2012-06-27','YYYY-MM-DD'),3008,5002);

insert into orders values (70011,75.29,TO\_DATE('2012-08-17','YYYY-MM-DD'),3003,5007);

insert into orders values (70013,3045.6,TO\_DATE('2012-04-25','YYYY-MM-DD'),3002,5001);

**select \* from orders where (ord\_date != TO\_DATE('2012-09-10','YYYY-MM-DD') AND salesman\_id <= 5005) OR purch\_amt <= 1000.00;**

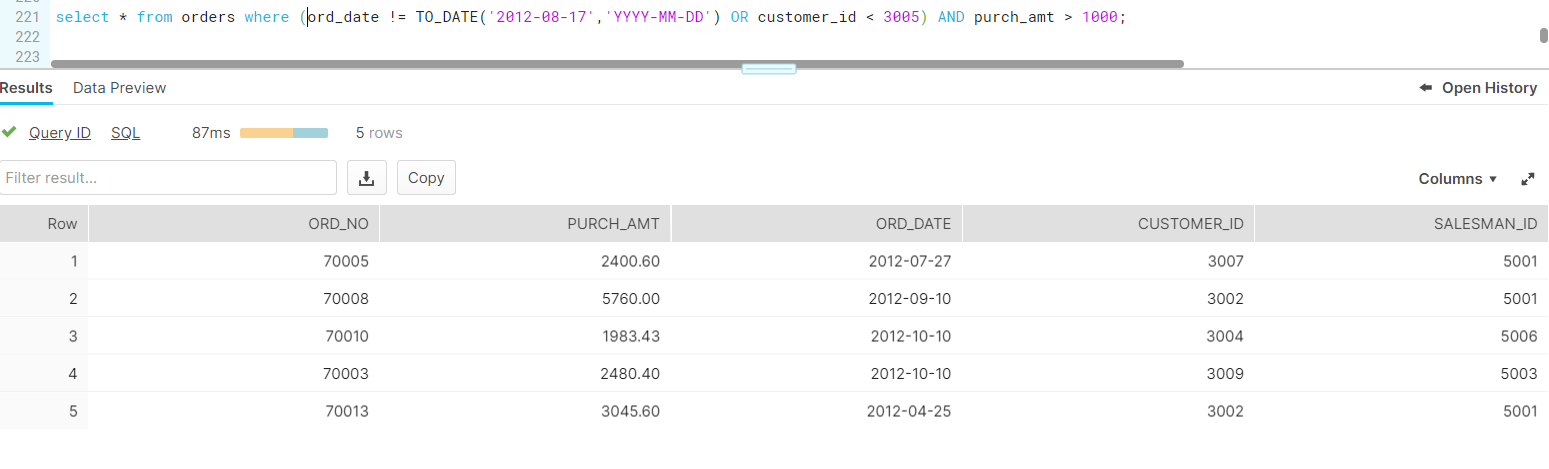


/\* 15. Write a SQL statement for above table where

i) order dates are anything but not 2012-08-17 or customer\_id is not greater than 3005.

Ii) and purchase amount is not below 1000. \*/

**select \* from orders where (ord\_date != TO\_DATE('2012-08-17','YYYY-MM-DD') OR customer\_id < 3005) AND purch\_amt > 1000;**



/\* 16.Use Unpivot function on below data

student

And produce below result \*/

create or replace table students

(

sname VARCHAR(1024),

maths smallint,

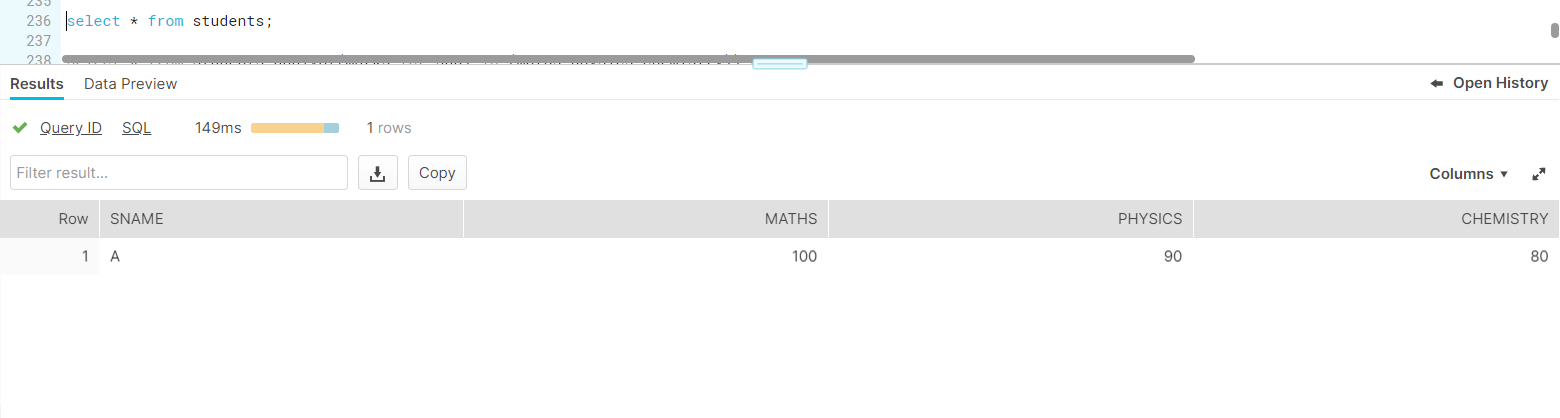
physics smallint,

chemistry smallint

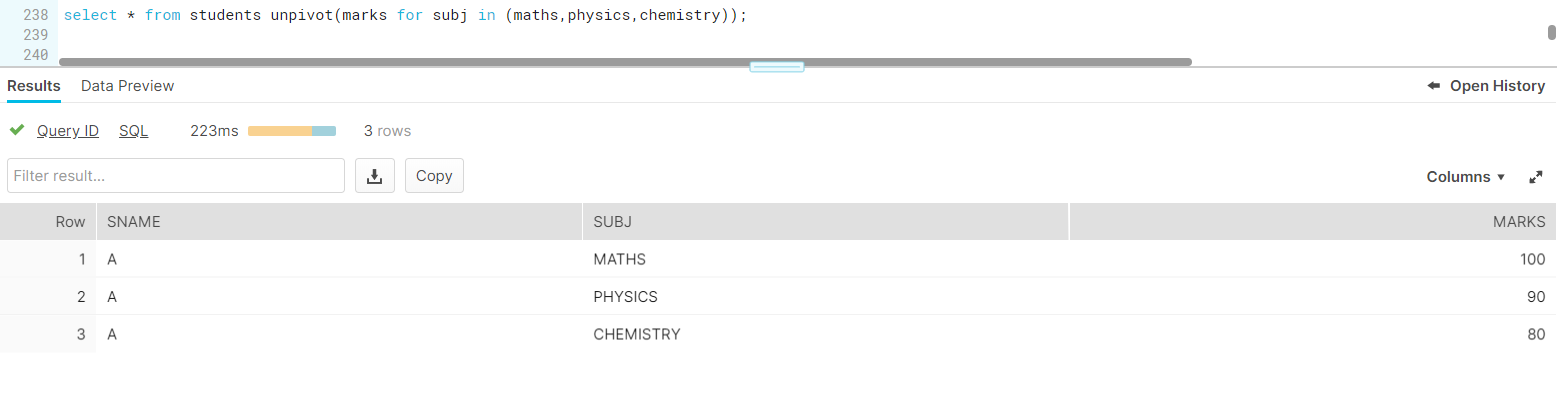
);

insert into students values ('A',100,90,80);

**select \* from students;**



**select \* from students unpivot(marks for subj in (maths,physics,chemistry));**



//17. Use Pivot function on below data

create or replace table students\_pivot

(

sname VARCHAR(1024),

subj VARCHAR(128),

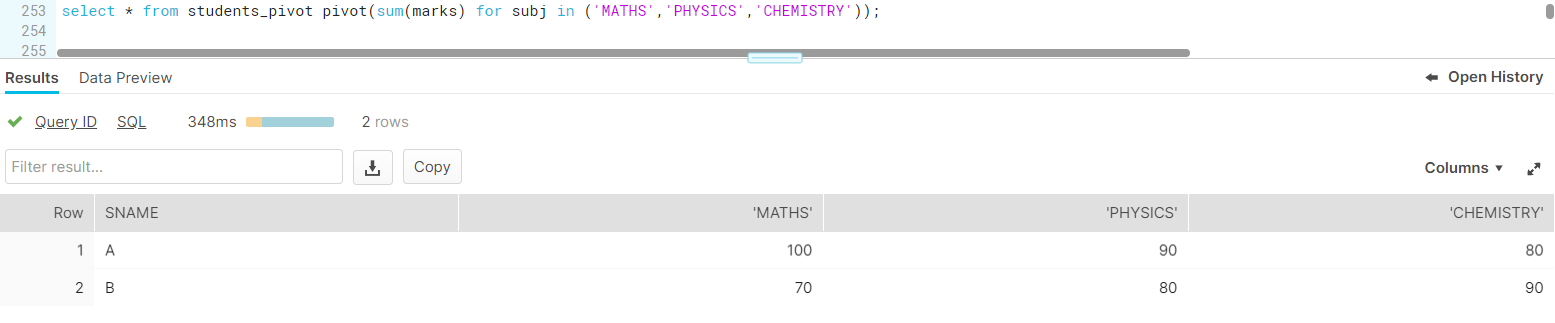
marks number(38)

);

insert into students\_pivot select \* from students unpivot(marks for subj in (maths,physics,chemistry));

**select \* from students\_pivot;**

**select \* from students\_pivot pivot(sum(marks) for subj in ('MATHS','PHYSICS','CHEMISTRY'));**



/\*18.Create employee table with Deptno,Employeenumber,Employeename and Salry fields

Create department table with deptno,deptname and location fields write queries to perform

Inner join ,Left outer join and Right Outer Join, Full outer join between employee and department tables using common column deptno \*/

create or replace table emp

(

Emp\_IDNO SMALLINT,

EMP\_name VARCHAR(1024),

sal NUMBER(38),

Deptno SMALLINT

);

create or replace table dept

(

deptno SMALLINT,

deptname VARCHAR(1024)

);

insert into emp values (127323,'Michale',10000,10);

insert into emp values (526689,'Carlos',25000,20);

insert into emp values (843795,'Enric',15000,20);

insert into emp values (328717,'Jhon',50000,30);

insert into emp values (444527,'Joseph',12500,30);

insert into emp values (659831,'Zanifer',60000,10);

insert into emp values (123456,'Zara',60000,50);

insert into dept values (10,'Sales');

insert into dept values (20,'Tech');

insert into dept values (30,'HR');

insert into dept values (40,'Mkt');

--Inner join

**select emp.\*,dept.\* from emp emp join dept dept on emp.deptno = dept.deptno;**

--Left Outer join

**select emp.\*,dept.\* from emp emp left join dept dept on emp.deptno = dept.deptno;**

--Right Outer join

**select emp.\*,dept.\* from emp emp right join dept dept on emp.deptno = dept.deptno;**

--Full Outer join

**select emp.\*,dept.\* from emp emp full join dept dept on emp.deptno = dept.deptno;**

