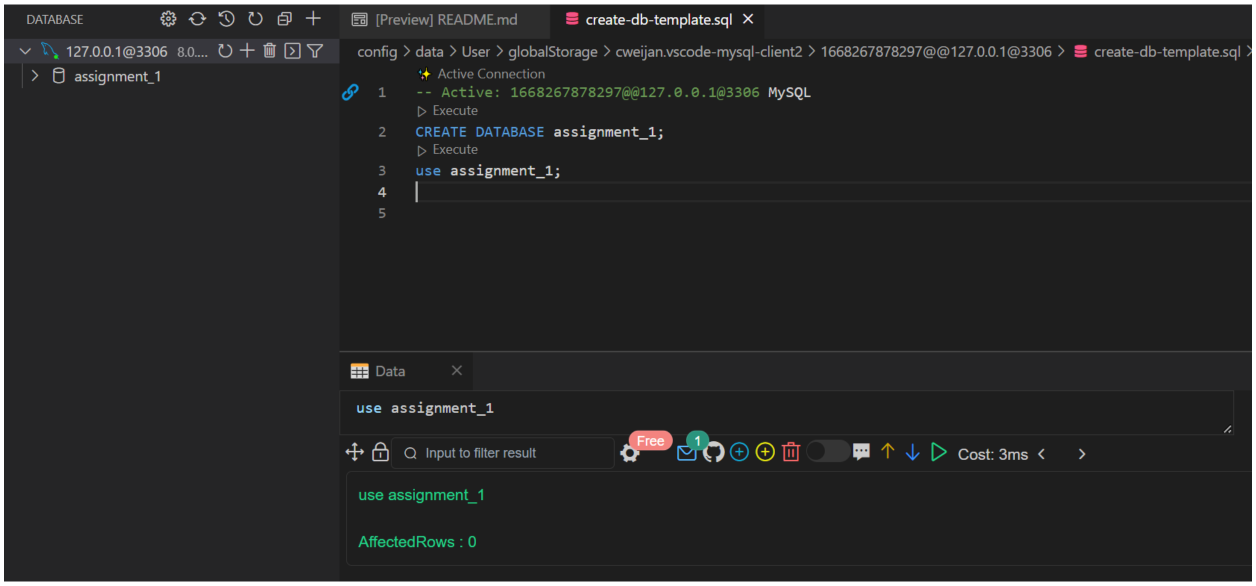
SQL ASSIGNMENT 1:

Create Database => assignment_1

```
CREATE DATABASE assignment_1;
use assignment_1;
```



```
-- Active: 1668267878297@@127.0.0.1@3306 MySQL
CREATE DATABASE assignment_1;
use assignment_1;
```

```
use assignment_1
```

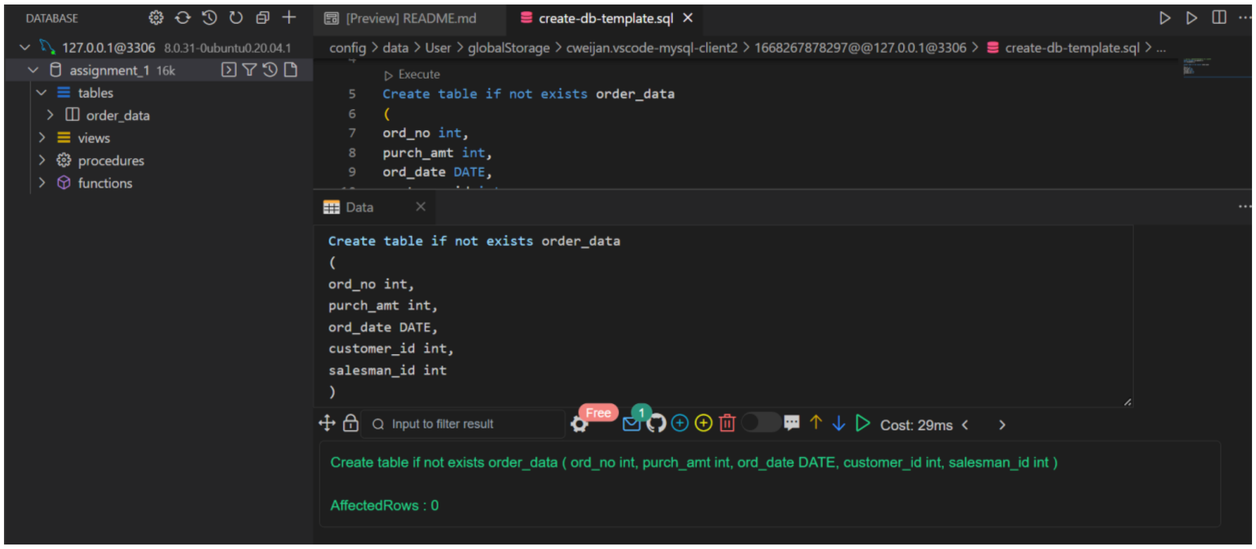
```
use assignment_1

AffectedRows : 0
```

1. Write a query to display the columns in a specific order, such as order date, salesman ID, order number, and purchase amount for all orders.
ord_no purch_amt ord_date customer_id salesman_id

```
---------- ---------- ---------- ----------- ----------- ----------- ----------- ----
70001 150.5 2012-10-05 3005 5002
70009 270.65 2012-09-10 3001 5005
70002 65.26 2012-10-05 3002 5001
70004 110.5 2012-08-17 3009 5003
70007 948.5 2012-09-10 3005 5002
70005 2400.6 2012-07-27 3007 5001
70008 5760 2012-09-10 3002 5001
70010 1983.43 2012-10-10 3004 5006
70003 2480.4 2012-10-10 3009 5003
70012 250.45 2012-06-27 3008 5002
70011 75.29 2012-08-17 3003 5007
70013 3045.6 2012-04-25 3002 5001
```

```
create table order_data(
ord_no int,
purch_amt int,
ord_date DATE,
customer_id int,
salesman_id int
);
```

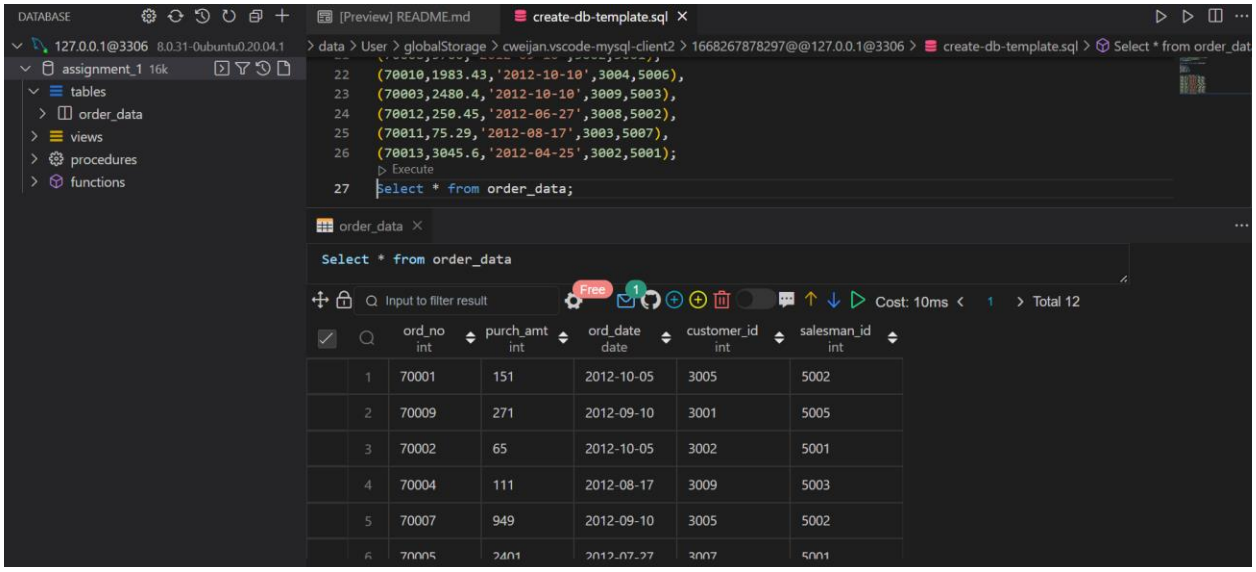
```
insert into order_data values
(70001,150.5,'2012-10-05',3005,5002),
(70009,270.65,'2012-09-10',3001,5005),
(70002,65.26,'2012-10-05',3002,5001),
(70004,110.5,'2012-08-17',3009,5003),
(70007,948.5,'2012-09-10',3005,5002),
(70005,2400.6,'2012-07-27', 3007,5001),
(70008,5760,'2012-09-10',3002,5001),
(70010,1983.43,'2012-10-10',3004,5006),
(70003,2480.4,'2012-10-10',3009,5003),
(70012,250.45,'2012-06-27',3008,5002),
(70011,75.29,'2012-08-17',3003,5007),
(70013,3045.6,'2012-04-25',3002,5001);
```


DATABASE
127.0.0.1@3306 8.0.31-0ubuntu0.20.04.1
assignment_1 16k
tables
order_data
views
procedures
functions
[Preview] README.md
create-db-template.sql
config > data > User > globalStorage > cweijan.vscode-mysql-client2 > 1668267878297@@127.0.0.1@3306 > create-db-template.sql > ...
Execute
Data
Input to filter result
Free
Cost: 29ms

```
Create table if not exists order_data
(
ord_no int,
purch_amt int,
ord_date DATE,
```

```
Create table if not exists order_data
(
ord_no int,
purch_amt int,
ord_date DATE,
customer_id int,
salesman_id int
)
```

```
Create table if not exists order_data ( ord_no int, purch_amt int, ord_date DATE, customer_id int, salesman_id int )

AffectedRows : 0
```

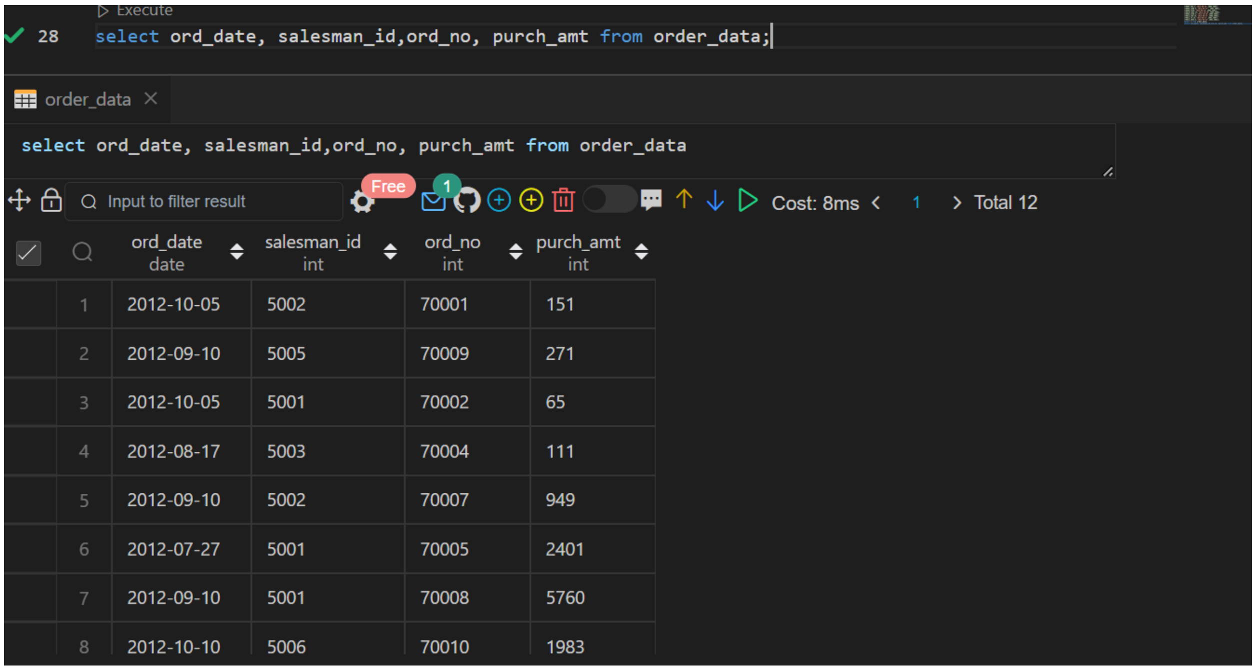

DATABASE
127.0.0.1@3306 8.0.31-0ubuntu0.20.04.1
assignment_1 16k
tables
order_data
views
procedures
functions
[Preview] README.md
create-db-template.sql
> data > User > globalStorage > cweijan.vscode-mysql-client2 > 1668267878297@@127.0.0.1@3306 > create-db-template.sql > Select * from order_dat
Execute
order_data
Input to filter result
Free
Cost: 10ms
1
Total 12

```
(70010,1983.43,'2012-10-10',3004,5006),
(70003,2480.4,'2012-10-10',3009,5003),
(70012,250.45,'2012-06-27',3008,5002),
(70011,75.29,'2012-08-17',3003,5007),
(70013,3045.6,'2012-04-25',3002,5001);
Select * from order_data;
```

```
Select * from order_data
```

	ord_no int	purch_amt int	ord_date date	customer_id int	salesman_id int
1	70001	151	2012-10-05	3005	5002
2	70009	271	2012-09-10	3001	5005
3	70002	65	2012-10-05	3002	5001
4	70004	111	2012-08-17	3009	5003
5	70007	949	2012-09-10	3005	5002
6	70005	2401	2012-07-27	3007	5001

select ord_date, salesman_id,ord_no, purch_amt from order_data;



```
select ord_date, salesman_id,ord_no, purch_amt from order_data;
```

order_data

```
select ord_date, salesman_id,ord_no, purch_amt from order_data
```

Cost: 8ms   1   Total 12

	ord_date date	salesman_id int	ord_no int	purch_amt int
1	2012-10-05	5002	70001	151
2	2012-09-10	5005	70009	271
3	2012-10-05	5001	70002	65
4	2012-08-17	5003	70004	111
5	2012-09-10	5002	70007	949
6	2012-07-27	5001	70005	2401
7	2012-09-10	5001	70008	5760
8	2012-10-10	5006	70010	1983

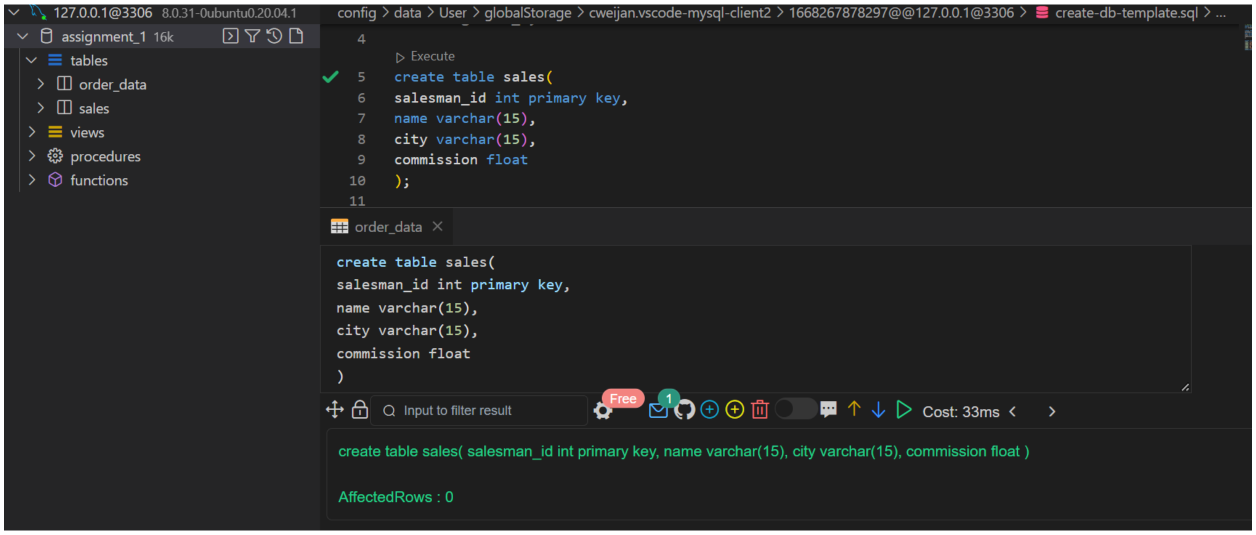
2. From the following table, write a SQL query to locate salespeople who live in the city of 'Paris'. Return salesperson's name, city.

salesman_id | name | city | commission

```
-------------+------------+----------+------------
5001 | James Hoog | New York | 0.15
5002 | Nail Knite | Paris | 0.13
5005 | Pit Alex | London | 0.11
5006 | Mc Lyon | Paris | 0.14
5007 | Paul Adam | Rome | 0.13
5003 | Lauson Hen | San Jose | 0.12
```

```
create table sales(
salesman_id int primary key,
name varchar(15),
city varchar(15),
commission float
);
```

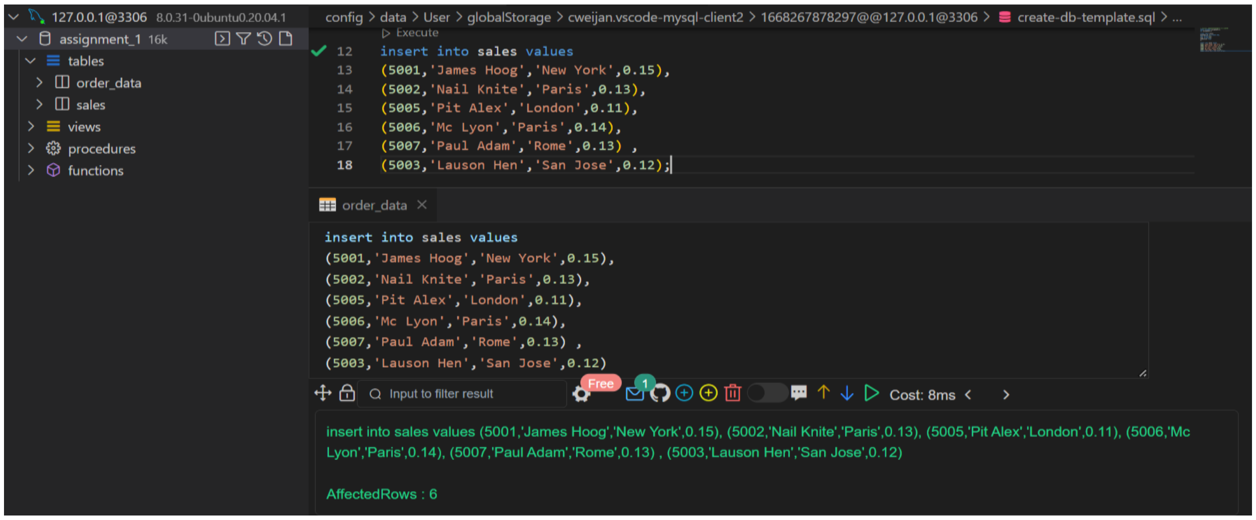
```
insert into sales values
(5001,'James Hoog','New York',0.15),
(5002,'Nail Knite','Paris',0.13),
(5005,'Pit Alex','London',0.11),
(5006,'Mc Lyon','Paris',0.14),
(5007,'Paul Adam','Rome',0.13) ,
(5003,'Lauson Hen','San Jose',0.12);
```


127.0.0.1@3306 8.0.31-0ubuntu0.20.04.1
assignment_1 16k
tables
order_data
sales
views
procedures
functions
config > data > User > globalStorage > cweijan.vscode-mysql-client2 > 1668267878297@@127.0.0.1@3306 > create-db-template.sql > ...
Execute
order_data
Input to filter result
Free
Cost: 33ms

```
create table sales(
salesman_id int primary key,
name varchar(15),
city varchar(15),
commission float
);
```

```
create table sales(
salesman_id int primary key,
name varchar(15),
city varchar(15),
commission float
)
```

```
create table sales( salesman_id int primary key, name varchar(15), city varchar(15), commission float )

AffectedRows : 0
```

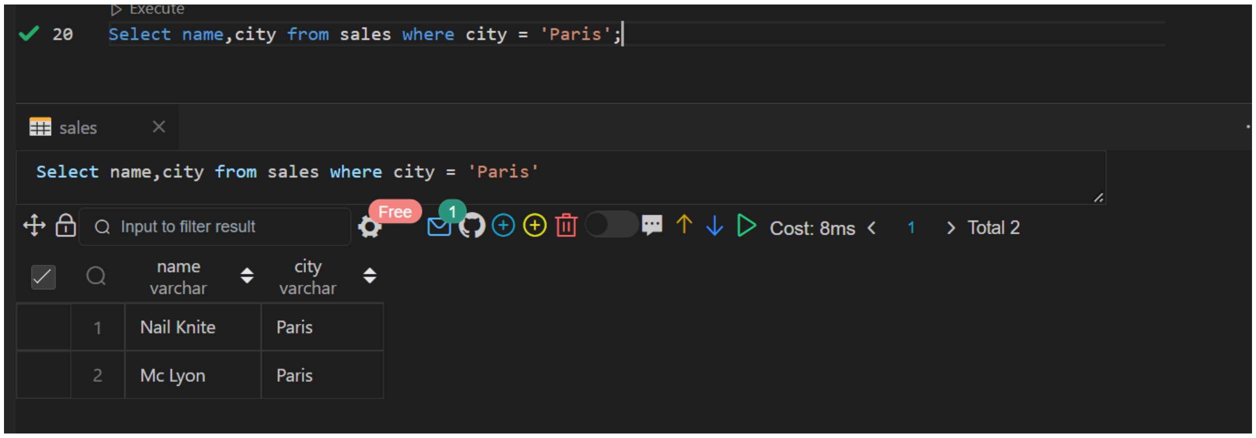

127.0.0.1@3306 8.0.31-0ubuntu0.20.04.1
assignment_1 16k
tables
order_data
sales
views
procedures
functions
config > data > User > globalStorage > cweijan.vscode-mysql-client2 > 1668267878297@@127.0.0.1@3306 > create-db-template.sql > ...
Execute
order_data
Input to filter result
Free
Cost: 8ms

```
insert into sales values
(5001,'James Hoog','New York',0.15),
(5002,'Nail Knite','Paris',0.13),
(5005,'Pit Alex','London',0.11),
(5006,'Mc Lyon','Paris',0.14),
(5007,'Paul Adam','Rome',0.13) ,
(5003,'Lauson Hen','San Jose',0.12);
```

```
insert into sales values
(5001,'James Hoog','New York',0.15),
(5002,'Nail Knite','Paris',0.13),
(5005,'Pit Alex','London',0.11),
(5006,'Mc Lyon','Paris',0.14),
(5007,'Paul Adam','Rome',0.13) ,
(5003,'Lauson Hen','San Jose',0.12)
```

```
insert into sales values (5001,'James Hoog','New York',0.15), (5002,'Nail Knite','Paris',0.13), (5005,'Pit Alex','London',0.11), (5006,'Mc
Lyon','Paris',0.14), (5007,'Paul Adam','Rome',0.13) , (5003,'Lauson Hen','San Jose',0.12)

AffectedRows : 6
```

Select name,city from sales where city = 'Paris';



```
Select name,city from sales where city = 'Paris';
```

```
Select name,city from sales where city = 'Paris'
```

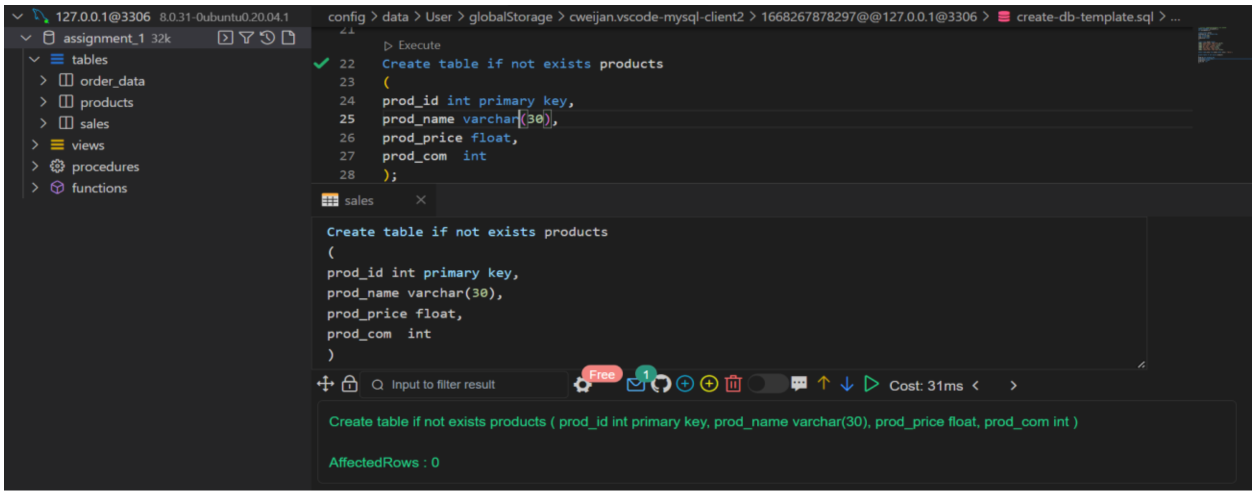
	name varchar	city varchar
1	Nail Knite	Paris
2	Mc Lyon	Paris

3. From the following table, write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro_id, pro_name, pro_price, and pro_com.

```
PRO_ID PRO_NAME PRO_PRICE PRO_COM
------- ------------------------ ------------------- --------------
101 Motherboard 3200.00 15
102 Keyboard 450.00 16
103 ZIP drive 250.00 14
104 Speaker 550.00 16
105 Monitor 5000.00 11
106 DVD drive 900.00 12
107 CD drive 800.00 12
108 Printer 2600.00 13
109 Refill cartridge 350.00 13
110 Mouse 250.00 12
```

```
create table if not exists products
(
prod_id int primary key,
prod_name varchar(30),
prod_price float,
prod_com  int);
```

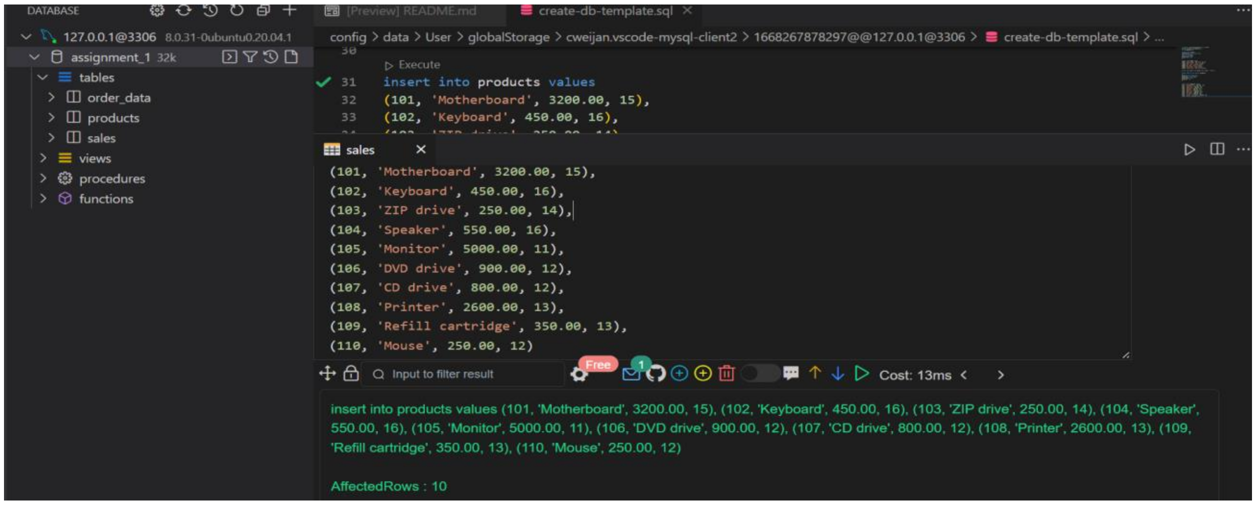
```
insert into products values
(101, 'Motherboard', 3200.00, 15),
(102, 'Keyboard', 450.00, 16),
(103, 'ZIP drive', 250.00, 14),
(104, 'Speaker', 550.00, 16),
(105, 'Monitor', 5000.00, 11),
(106, 'DVD drive', 900.00, 12),
(107, 'CD drive', 800.00, 12),
(108, 'Printer', 2600.00, 13),
(109, 'Refill cartridge', 350.00, 13),
(110, 'Mouse', 250.00, 12);
```


127.0.0.1@3306 8.0.31-0ubuntu0.20.04.1
assignment_1 32k
tables
order_data
products
sales
views
procedures
functions
config > data > User > globalStorage > cweijan.vscode-mysql-client2 > 1668267878297@@127.0.0.1@3306 > create-db-template.sql > ...
Execute
sales
Input to filter result
Free
Cost: 31ms

```
Create table if not exists products
(
prod_id int primary key,
prod_name varchar(30),
prod_price float,
prod_com  int
);
```

```
Create table if not exists products
(
prod_id int primary key,
prod_name varchar(30),
prod_price float,
prod_com  int
)
```

```
Create table if not exists products ( prod_id int primary key, prod_name varchar(30), prod_price float, prod_com int )

AffectedRows : 0
```

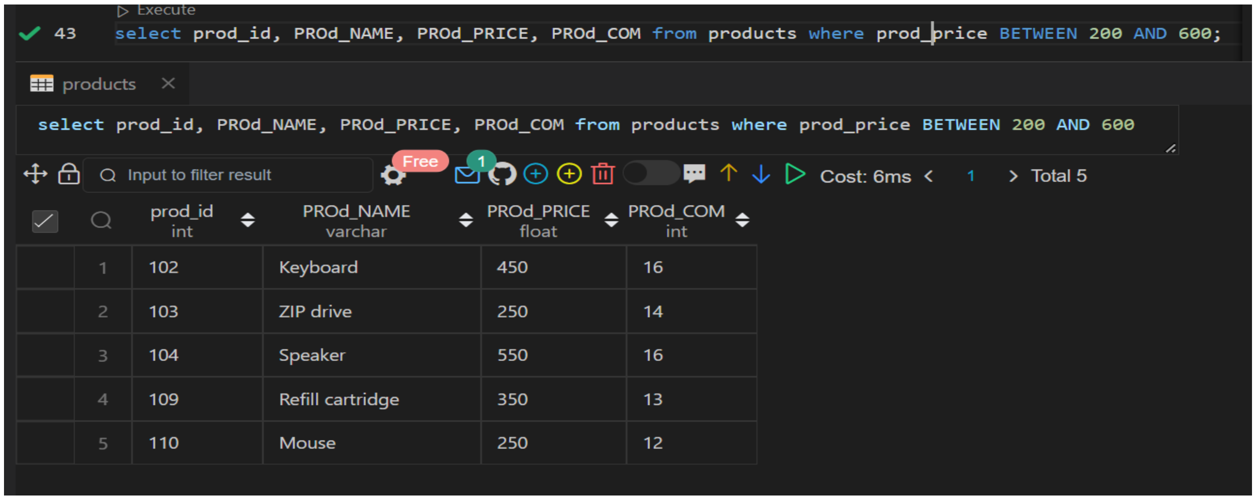


```
insert into products values
(101, 'Motherboard', 3200.00, 15),
(102, 'Keyboard', 450.00, 16),
```

```
(101, 'Motherboard', 3200.00, 15),
(102, 'Keyboard', 450.00, 16),
(103, 'ZIP drive', 250.00, 14),
(104, 'Speaker', 550.00, 16),
(105, 'Monitor', 5000.00, 11),
(106, 'DVD drive', 900.00, 12),
(107, 'CD drive', 800.00, 12),
(108, 'Printer', 2600.00, 13),
(109, 'Refill cartridge', 350.00, 13),
(110, 'Mouse', 250.00, 12)
```

```
insert into products values (101, 'Motherboard', 3200.00, 15), (102, 'Keyboard', 450.00, 16), (103, 'ZIP drive', 250.00, 14), (104, 'Speaker', 550.00, 16), (105, 'Monitor', 5000.00, 11), (106, 'DVD drive', 900.00, 12), (107, 'CD drive', 800.00, 12), (108, 'Printer', 2600.00, 13), (109, 'Refill cartridge', 350.00, 13), (110, 'Mouse', 250.00, 12)

AffectedRows : 10
```

select prod_id, PROd_NAME, PROd_PRICE, PROd_COM from products where prod_price BETWEEN 200 AND 600;



```
select prod_id, PROd_NAME, PROd_PRICE, PROd_COM from products where prod_price BETWEEN 200 AND 600;
```

```
select prod_id, PROd_NAME, PROd_PRICE, PROd_COM from products where prod_price BETWEEN 200 AND 600
```

Cost: 6ms Total 5

	prod_id int	PROd_NAME varchar	PROd_PRICE float	PROd_COM int
1	102	Keyboard	450	16
2	103	ZIP drive	250	14
3	104	Speaker	550	16
4	109	Refill cartridge	350	13
5	110	Mouse	250	12

4. From the following table, write a SQL query to find the items whose prices are higher than or equal to $550. Order the result by product price in descending, then product name in ascending. Return pro_name and pro_price.

PRO_ID PRO_NAME PRO_PRICE PRO_COM

------- ------------------------- -------------- ----------

101 Motherboard 3200.00 15

102 Keyboard 450.00 16

103 ZIP drive 250.00 14

104 Speaker 550.00 16

105 Monitor 5000.00 11

106 DVD drive 900.00 12

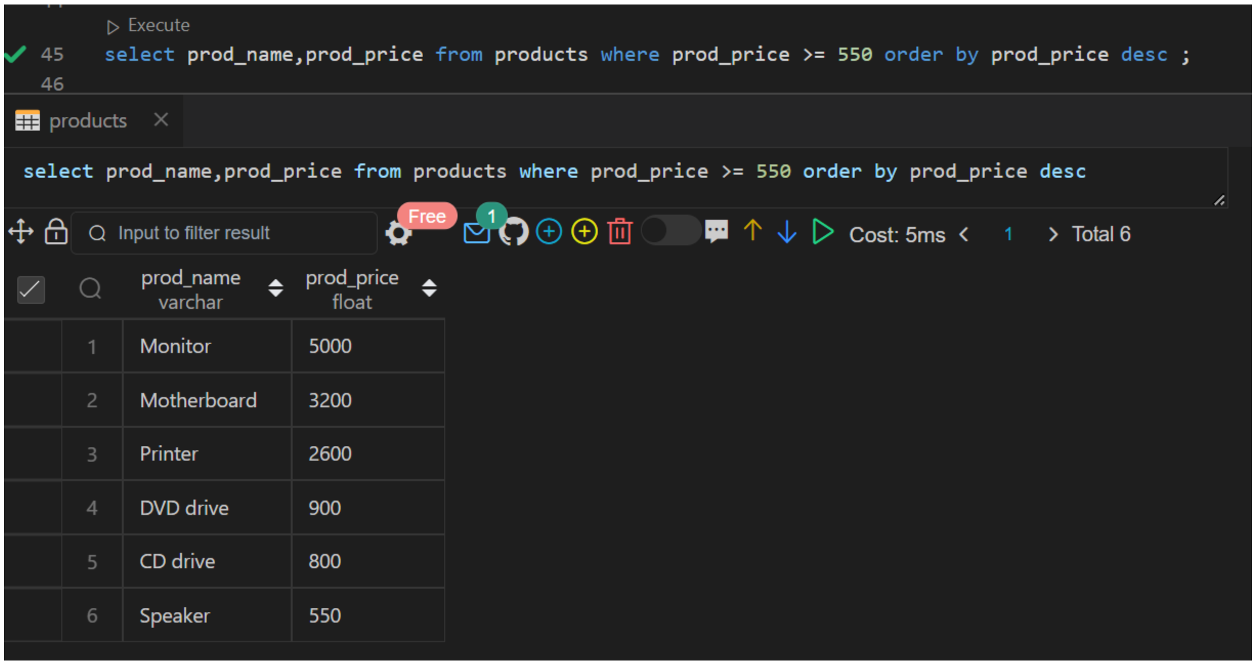
107 CD drive 800.00 12

108 Printer 2600.00 13

109 Refill cartridge 350.00 13

110 Mouse 250.00 12

select prod_name,prod_price from products where prod_price >= 550 order by prod_price desc ;



```
select prod_name,prod_price from products where prod_price >= 550 order by prod_price desc ;
```

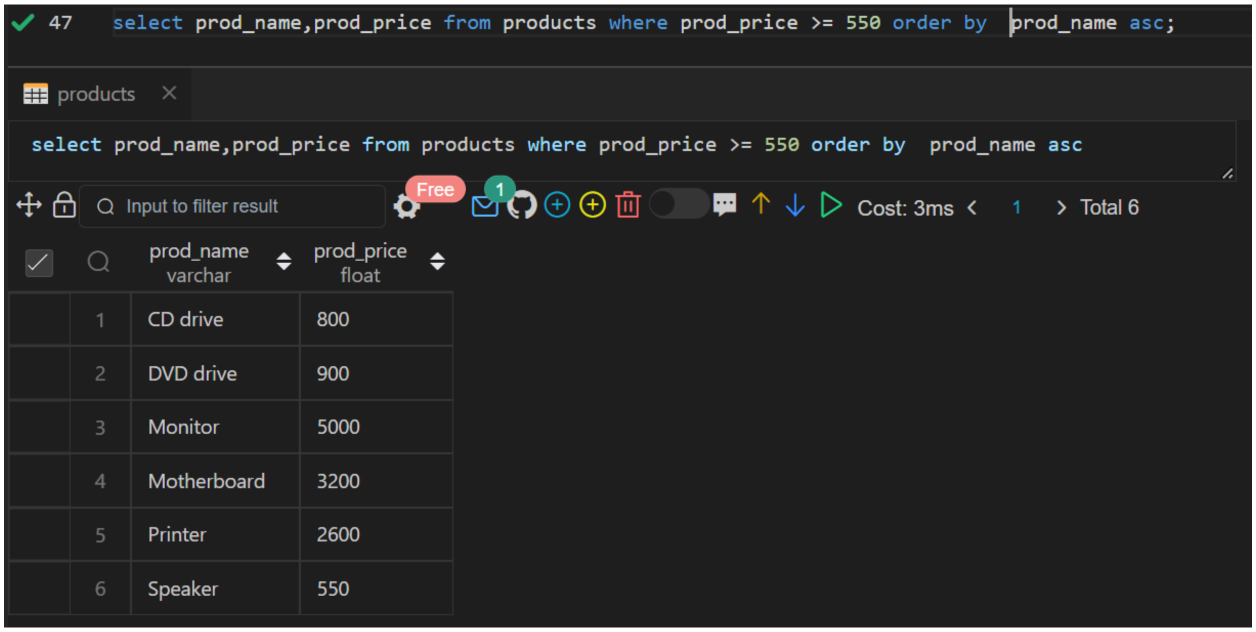
products

```
select prod_name,prod_price from products where prod_price >= 550 order by prod_price desc
```

Cost: 5ms 1 Total 6

	prod_name varchar	prod_price float
1	Monitor	5000
2	Motherboard	3200
3	Printer	2600
4	DVD drive	900
5	CD drive	800
6	Speaker	550

select prod_name,prod_price from products where prod_price >= 550 order by prod_name asc;



```
select prod_name,prod_price from products where prod_price >= 550 order by prod_name asc;
```

products

```
select prod_name,prod_price from products where prod_price >= 550 order by  prod_name asc
```

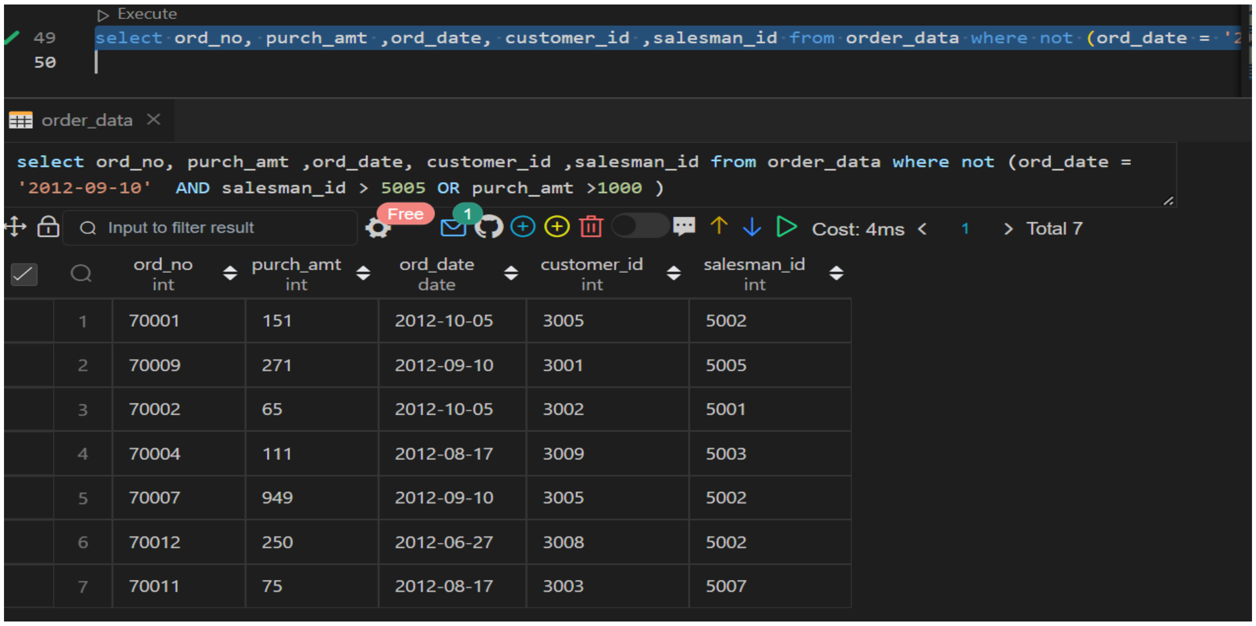
Cost: 3ms 1 Total 6

	prod_name varchar	prod_price float
1	CD drive	800
2	DVD drive	900
3	Monitor	5000
4	Motherboard	3200
5	Printer	2600
6	Speaker	550

5. From the following table, write a SQL query to find details of all orders excluding those with ord_date equal to '2012-09-10' and salesman_id higher than 5005 or purch_amt greater than 1000.Return ord_no, purch_amt, ord_date, customer_id and salesman_id.
ord_no purch_amt ord_date customer_id salesman_id
---------- ---------- ---------- ----------- ----------- —--------------
70001 150.5 2012-10-05 3005 5002
70009 270.65 2012-09-10 3001 5005

70002 65.26 2012-10-05 3002 5001
70004 110.5 2012-08-17 3009 5003
70007 948.5 2012-09-10 3005 5002
70005 2400.6 2012-07-27 3007 5001
70008 5760 2012-09-10 3002 5001
70010 1983.43 2012-10-10 3004 5006
70003 2480.4 2012-10-10 3009 5003
70012 250.45 2012-06-27 3008 5002
70011 75.29 2012-08-17 3003 5007
70013 3045.6 2012-04-25 3002 5001

select ord_no, purch_amt ,ord_date, customer_id ,salesman_id from order_data where not (ord_date = '2012-09-10' AND salesman_id > 5005 OR purch_amt >1000);



```
select ord_no, purch_amt ,ord_date, customer_id ,salesman_id from order_data where not (ord_date = '2
```

order_data

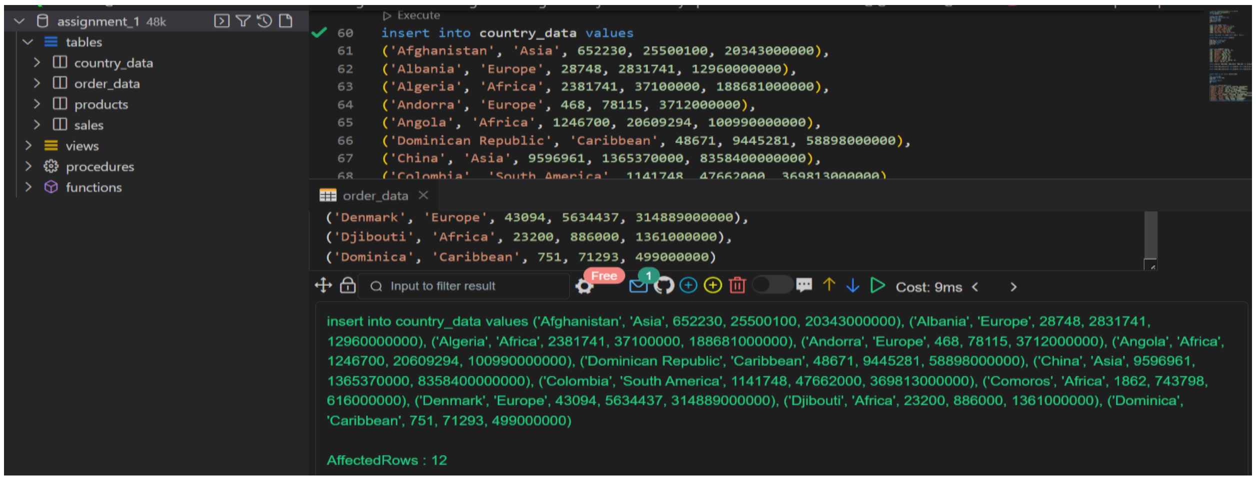
```
select ord_no, purch_amt ,ord_date, customer_id ,salesman_id from order_data where not (ord_date =
'2012-09-10'  AND salesman_id > 5005 OR purch_amt >1000 )
```

Cost: 4ms 1 Total 7

	ord_no int	purch_amt int	ord_date date	customer_id int	salesman_id int
1	70001	151	2012-10-05	3005	5002
2	70009	271	2012-09-10	3001	5005
3	70002	65	2012-10-05	3002	5001
4	70004	111	2012-08-17	3009	5003
5	70007	949	2012-09-10	3005	5002
6	70012	250	2012-06-27	3008	5002
7	70011	75	2012-08-17	3003	5007

6. Create the table world with your schema and find the below queries !
name continent area population gdp
Afghanistan Asia 652230 25500100 20343000000
Albania Europe 28748 2831741 12960000000
Algeria Africa 2381741 37100000 188681000000
Andorra Europe 468 78115 3712000000
Angola Africa 1246700 20609294 100990000000
Dominican Republic Caribbean 48671 9445281 58898000000
China Asia 9596961 1365370000 8358400000000
Colombia South America 1141748 47662000 369813000000
Comoros Africa 1862 743798 616000000
Denmark Europe 43094 5634437 314889000000
Djibouti Africa 23200 886000 1361000000
Dominica Caribbean 751 71293 499000000

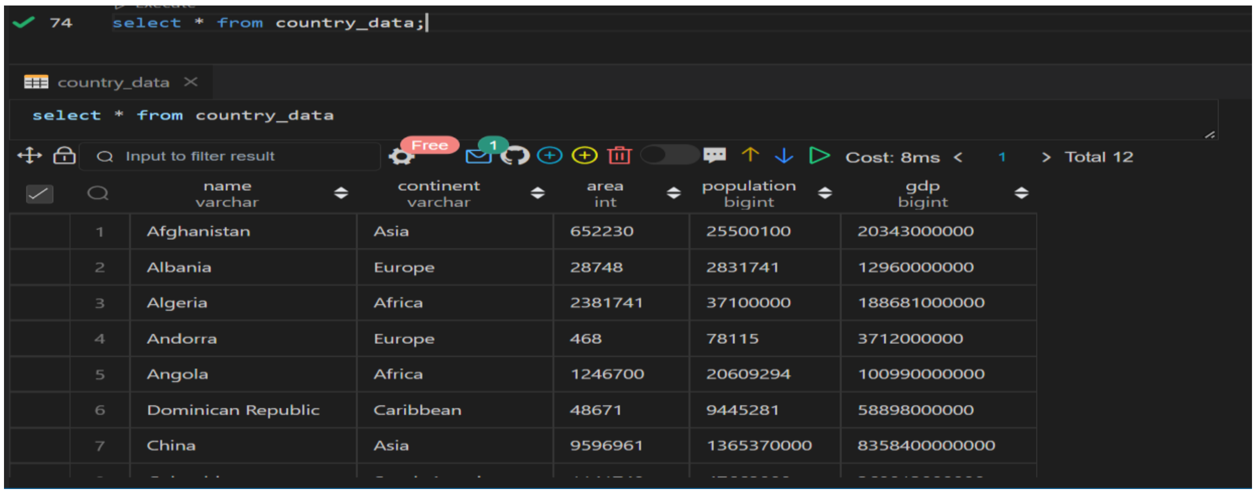
```
create table if not exists country_data (
name varchar(30),
continent varchar(30),
area int ,
population bigint,
gdp bigint);

insert into country_data values
('Afghanistan', 'Asia', 652230, 25500100, 20343000000),
('Albania', 'Europe', 28748, 2831741, 12960000000),
('Algeria', 'Africa', 2381741, 37100000, 188681000000),
('Andorra', 'Europe', 468, 78115, 3712000000),
('Angola', 'Africa', 1246700, 20609294, 100990000000),
('Dominican Republic', 'Caribbean', 48671, 9445281, 58898000000),
('China', 'Asia', 9596961, 1365370000, 8358400000000),
('Colombia', 'South America', 1141748, 47662000, 369813000000),
('Comoros', 'Africa', 1862, 743798, 616000000),
('Denmark', 'Europe', 43094, 5634437, 314889000000),
('Djibouti', 'Africa', 23200, 886000, 1361000000),
('Dominica', 'Caribbean', 751, 71293, 499000000);
```



```
insert into country_data values
('Afghanistan', 'Asia', 652230, 25500100, 20343000000),
('Albania', 'Europe', 28748, 2831741, 12960000000),
('Algeria', 'Africa', 2381741, 37100000, 188681000000),
('Andorra', 'Europe', 468, 78115, 3712000000),
('Angola', 'Africa', 1246700, 20609294, 100990000000),
('Dominican Republic', 'Caribbean', 48671, 9445281, 58898000000),
('China', 'Asia', 9596961, 1365370000, 8358400000000),
('Colombia', 'South America', 1141748, 47662000, 369813000000)
```

```
('Denmark', 'Europe', 43094, 5634437, 314889000000),
('Djibouti', 'Africa', 23200, 886000, 1361000000),
('Dominica', 'Caribbean', 751, 71293, 499000000)
```

```
insert into country_data values ('Afghanistan', 'Asia', 652230, 25500100, 20343000000), ('Albania', 'Europe', 28748, 2831741, 12960000000), ('Algeria', 'Africa', 2381741, 37100000, 188681000000), ('Andorra', 'Europe', 468, 78115, 3712000000), ('Angola', 'Africa', 1246700, 20609294, 100990000000), ('Dominican Republic', 'Caribbean', 48671, 9445281, 58898000000), ('China', 'Asia', 9596961, 1365370000, 8358400000000), ('Colombia', 'South America', 1141748, 47662000, 369813000000), ('Comoros', 'Africa', 1862, 743798, 616000000), ('Denmark', 'Europe', 43094, 5634437, 314889000000), ('Djibouti', 'Africa', 23200, 886000, 1361000000), ('Dominica', 'Caribbean', 751, 71293, 499000000)

AffectedRows : 12
```



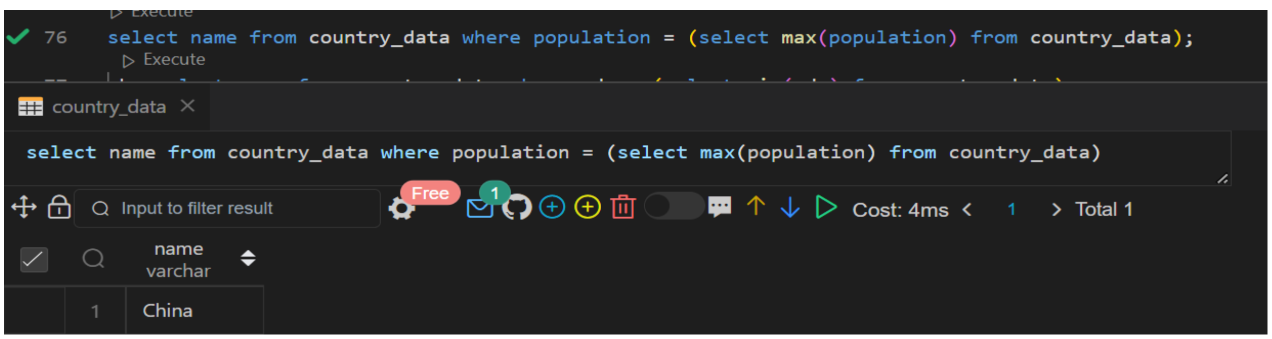
```
select * from country_data;
```

```
select * from country_data
```

	name varchar	continent varchar	area int	population bigint	gdp bigint
1	Afghanistan	Asia	652230	25500100	20343000000
2	Albania	Europe	28748	2831741	12960000000
3	Algeria	Africa	2381741	37100000	188681000000
4	Andorra	Europe	468	78115	3712000000
5	Angola	Africa	1246700	20609294	100990000000
6	Dominican Republic	Caribbean	48671	9445281	58898000000
7	China	Asia	9596961	1365370000	8358400000000

1. Write a query to fetch which country has the highest population?

select name from country_data where population = (select max(population) from country_data);



```
select name from country_data where population = (select max(population) from country_data);
```

```
select name from country_data where population = (select max(population) from country_data)
```

	name varchar
1	China

2.write a query to fetch the name of the country which has the least gdp?

select name from country_data where gdp = (select min(gdp) from country_data);

3.Write a query to fetch the name of the country which ends with letter C?

Select name from country_data where name like '%C';

4.write a query to fetch the name of the country which starts with letter D?

Select name from country_data where name like 'D%';

5.write query to fetch which continent has highest gdp?

select continent from country_data where gdp = (select max(gdp) from country_data);

6.Give the total GDP of Africa?

select sum (gdp) gdp from country_data where continent = 'Africa';

7.write a query to fetch the total population for each continent?

select continent,sum (population)population from country_data group by continent;

8. For each relevant continent show the number of countries that has a population of at least 200000000?

select continent, count(name)no_of_countries from country_data where population >= 200000000 group by continent;

Hint: Can be solved using aggregate function

7. Problem statement: Suppose we have two table students and course

create table students(student_id int,
student_name varchar(60) not null,
city varchar(60) not null,
primary key(student_id));

create table course(student_id int,
course_name varchar(60) not null,
Marks int not null,
primary key(student_id),
foreign key(student_id) references students(student_id));

insert into students values(200,'John Doe','Delhi'),
(210,'John Doe','Delhi'),(220,'Moon ethan','Rajasthan'),

(230,'Jessie','Bangalore'),(240,'Benbrook','Bihar'),(250,'Ethan','Bihar'),
(260,'Johnnie','Bangalore'),(270,'Goh','Delhi'),(380,'John Doe','Delhi'),
(280,'Pavi','Delhi'),(290,'Sanvi','Rajasthan'),(300,'Navyaa','Bangalore'),
(310,'Ankul','Bihar'),(311,'Hitanshi','Bihar'),(312,'Aayush','Bangalore'),
(313,'Rian','Delhi');

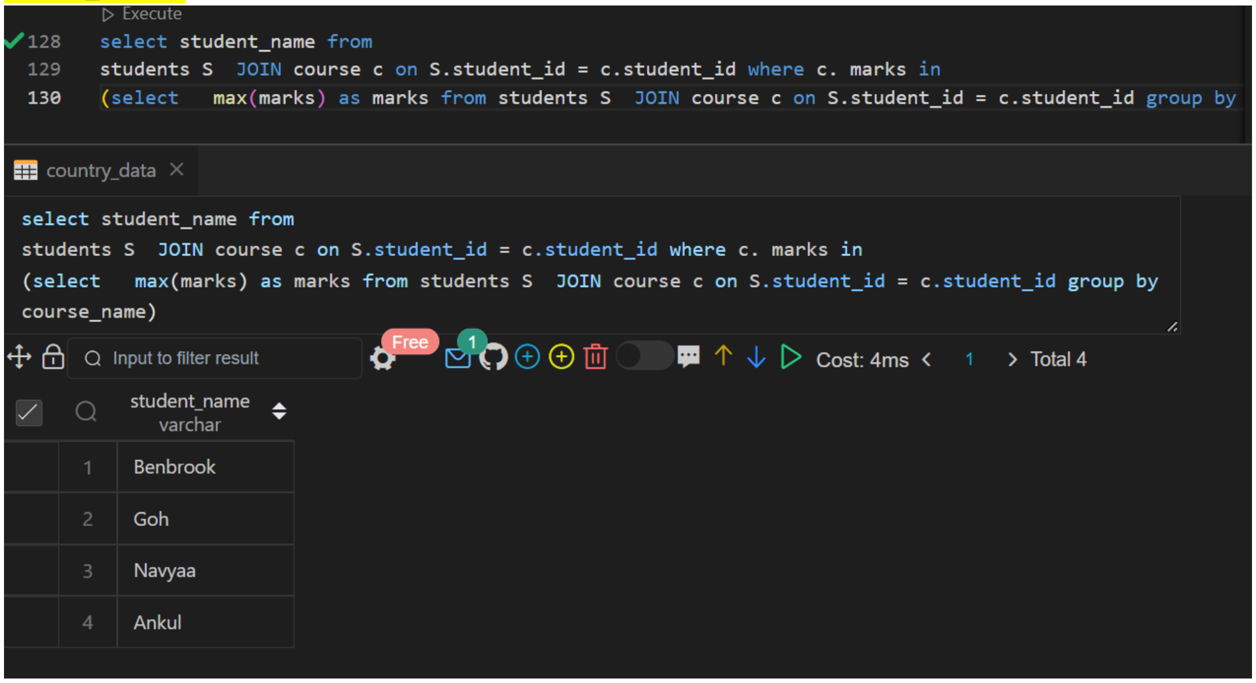
insert into course values(200,'Datascience',75),
(210,'Datascience',75),(220,'Dataanalyst',80),(230,'Dataanalyst',80),
(240,'Dataanalyst',84),(250,'Dataanalyst',50),(260,'Datascience',80),
(270,'Datascience',99),(380,'Datascience',45),(280,'Datascience',78),
(290,'Dataanalyst',78),(300,'Computer vision',90),(310,'Computer vision',90),
(311,'Computer vision',75),(312,'Computer vision',39);

Questions :

q1. write a query to fetch the names of the students having maximum marks in each course?

**** <u>Using JOINS</u>**

select student_name from
students S JOIN course c on S.student_id = c.student_id where c. marks in
(select   max(marks) as marks from students S JOIN course c on S.student_id = c.student_id group by
course_name);



```
▷ Execute
select student_name from
students S  JOIN course c on S.student_id = c.student_id where c. marks in
(select   max(marks) as marks from students S  JOIN course c on S.student_id = c.student_id group by
```

country_data

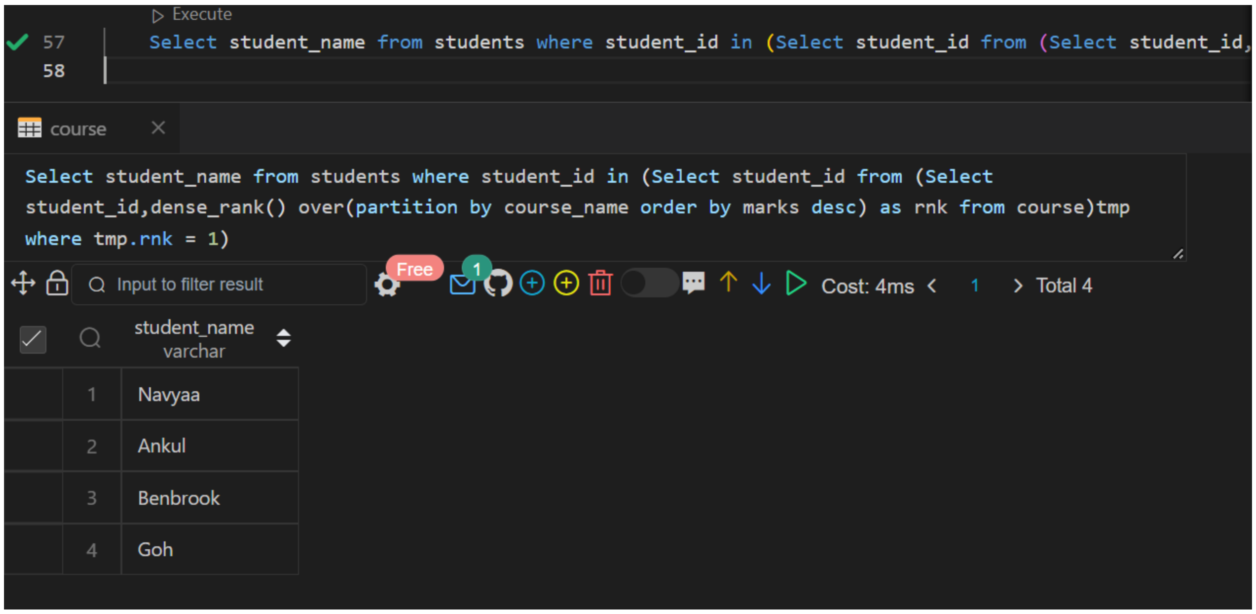
```
select student_name from
students S  JOIN course c on S.student_id = c.student_id where c. marks in
(select   max(marks) as marks from students S  JOIN course c on S.student_id = c.student_id group by
course_name)
```

Input to filter result   Cost: 4ms < 1 > Total 4

	student_name varchar
1	Benbrook
2	Goh
3	Navyaa
4	Ankul

**** <u>USING Window Function</u>**

Select student_name from students where student_id in (Select student_id from (Select student_id,dense_rank() over(partition by course_name order by marks desc) as rnk from course)tmp where tmp.rnk = 1);



```
Select student_name from students where student_id in (Select student_id from (Select student_id,
```

```
Select student_name from students where student_id in (Select student_id from (Select
student_id,dense_rank() over(partition by course_name order by marks desc) as rnk from course)tmp
where tmp.rnk = 1)
```

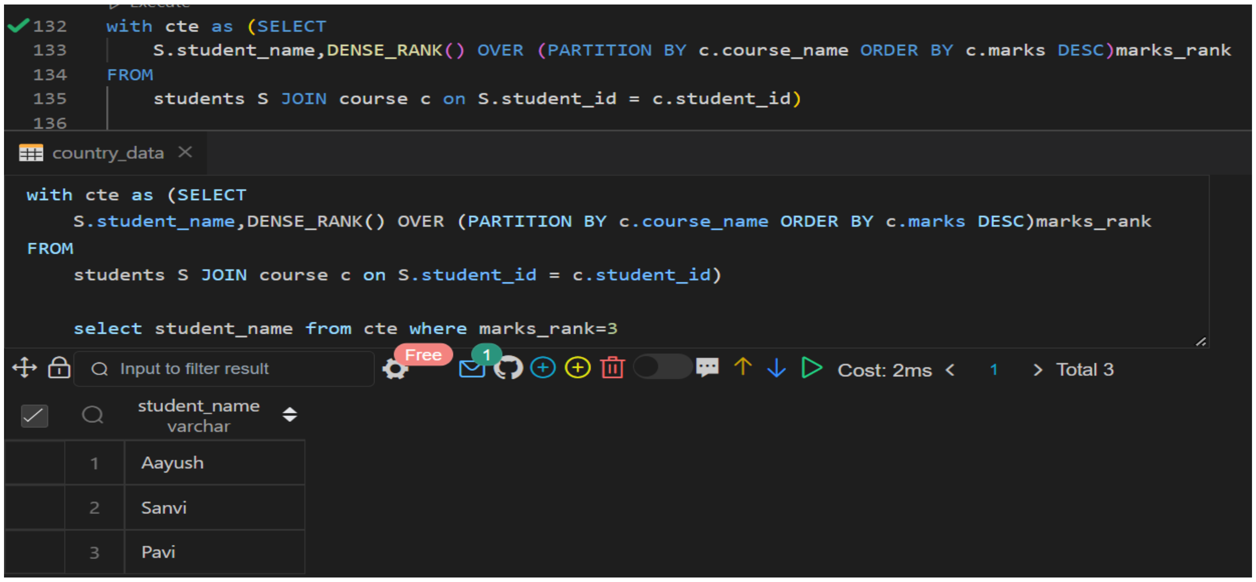
Cost: 4ms   Total 4

student_name varchar
Navyaa
Ankul
Benbrook
Goh

q2. write a query to fetch the names of the students having 3th highest marks from each course?

__Using CTE,Joins and Window Function__**

with cte as (SELECT
S.student_name,DENSE_RANK() OVER (PARTITION BY c.course_name ORDER BY c.marks DESC)marks_rank
FROM
students S JOIN course c on S.student_id = c.student_id)
select student_name from cte where marks_rank=3;



```
with cte as (SELECT
    S.student_name,DENSE_RANK() OVER (PARTITION BY c.course_name ORDER BY c.marks DESC)marks_rank
FROM
    students S JOIN course c on S.student_id = c.student_id)
```

```
with cte as (SELECT
    S.student_name,DENSE_RANK() OVER (PARTITION BY c.course_name ORDER BY c.marks DESC)marks_rank
FROM
    students S JOIN course c on S.student_id = c.student_id)

    select student_name from cte where marks_rank=3
```

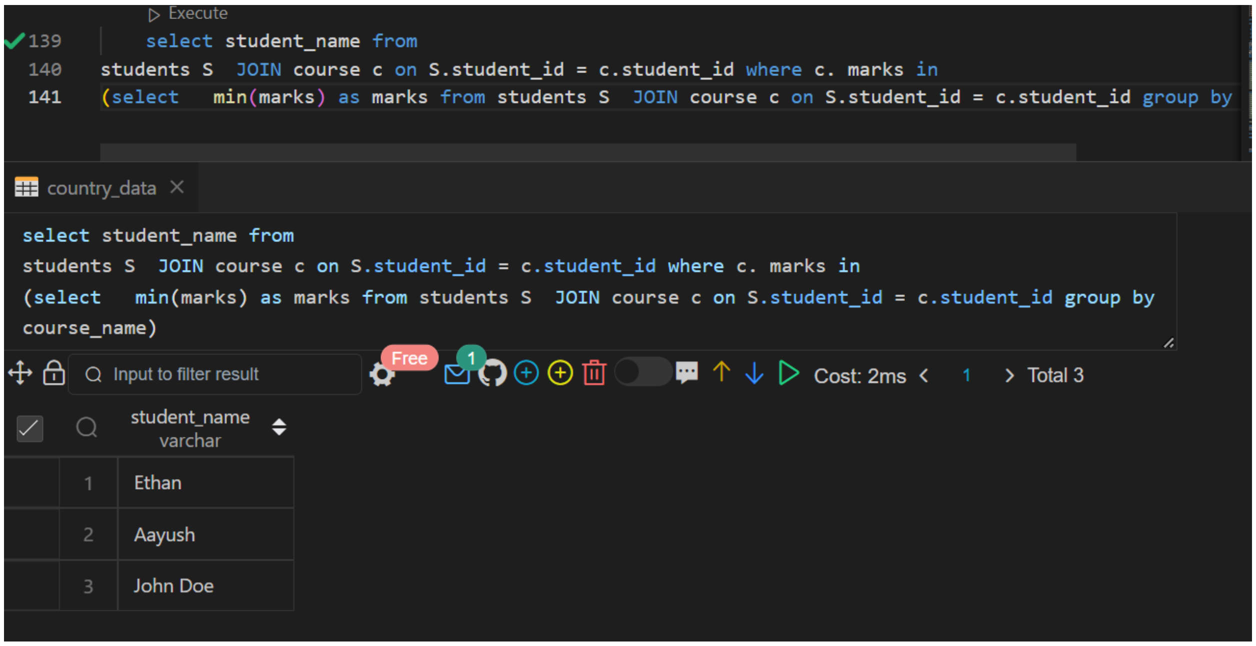
Cost: 2ms   Total 3

student_name varchar
Aayush
Sanvi
Pavi

q3. write a query to fetch the names of the students having minimum marks in each course?

<u>Using JOINS</u>**

select student_name from
students S JOIN course c on S.student_id = c.student_id where c. marks in
(select   min(marks) as marks from students S JOIN course c on S.student_id = c.student_id group by course_name);



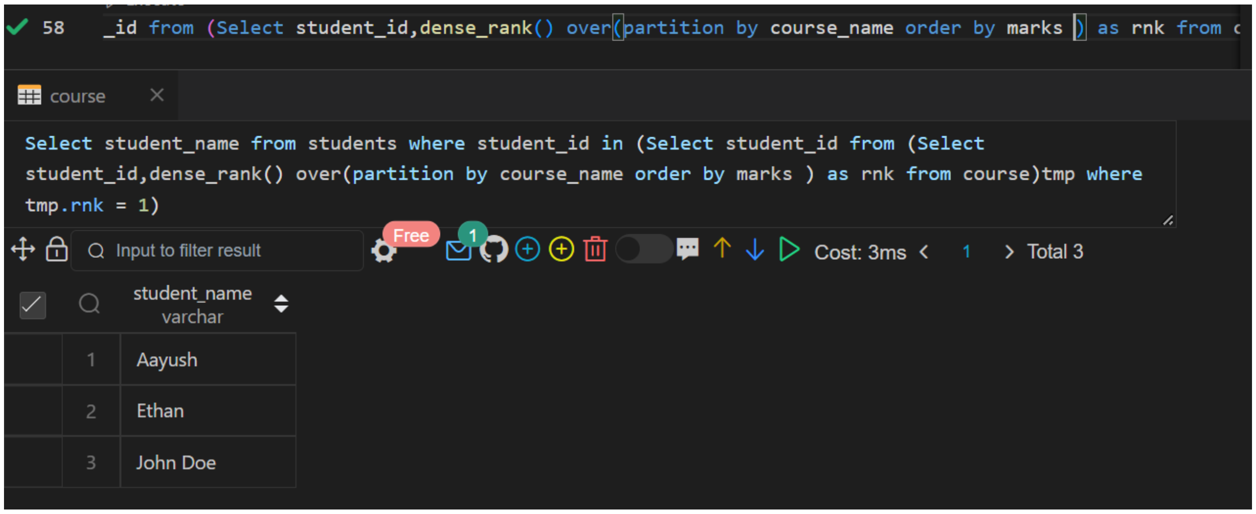
```
▷ Execute
select student_name from
students S  JOIN course c on S.student_id = c.student_id where c. marks in
(select   min(marks) as marks from students S  JOIN course c on S.student_id = c.student_id group by
```

```
select student_name from
students S  JOIN course c on S.student_id = c.student_id where c. marks in
(select   min(marks) as marks from students S  JOIN course c on S.student_id = c.student_id group by
course_name)
```

	student_name varchar
1	Ethan
2	Aayush
3	John Doe

<u>Using Window Function</u>**

Select student_name from students where student_id in (Select student_id from (Select student_id,dense_rank() over(partition by course_name order by marks) as rnk from course)tmp where tmp.rnk = 1);



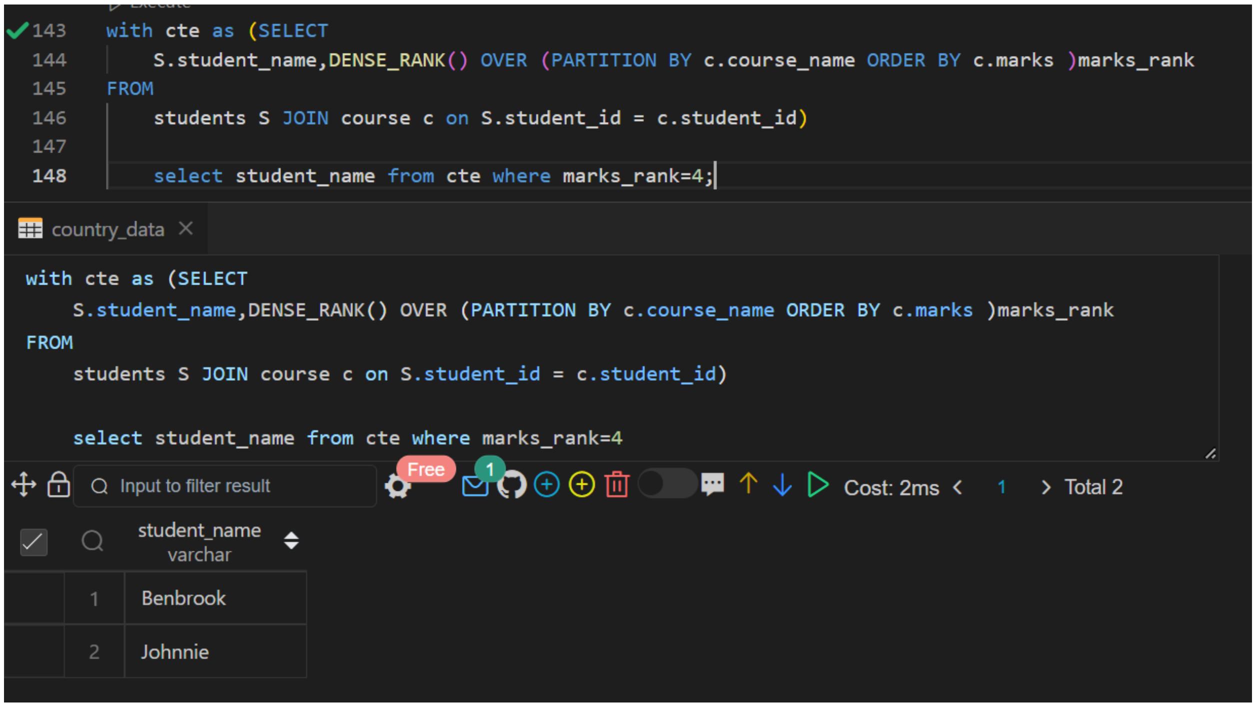
```
_id from (Select student_id,dense_rank() over(partition by course_name order by marks ) as rnk from c
```

```
Select student_name from students where student_id in (Select student_id from (Select
student_id,dense_rank() over(partition by course_name order by marks ) as rnk from course)tmp where
tmp.rnk = 1)
```

	student_name varchar
1	Aayush
2	Ethan
3	John Doe

q4. write a query to fetch the names of the students having 4th least marks from each course?

with cte as (SELECT
S.student_name,DENSE_RANK() OVER (PARTITION BY c.course_name ORDER BY c.marks)marks_rank
FROM
students S JOIN course c on S.student_id = c.student_id)
select student_name from cte where marks_rank=4;



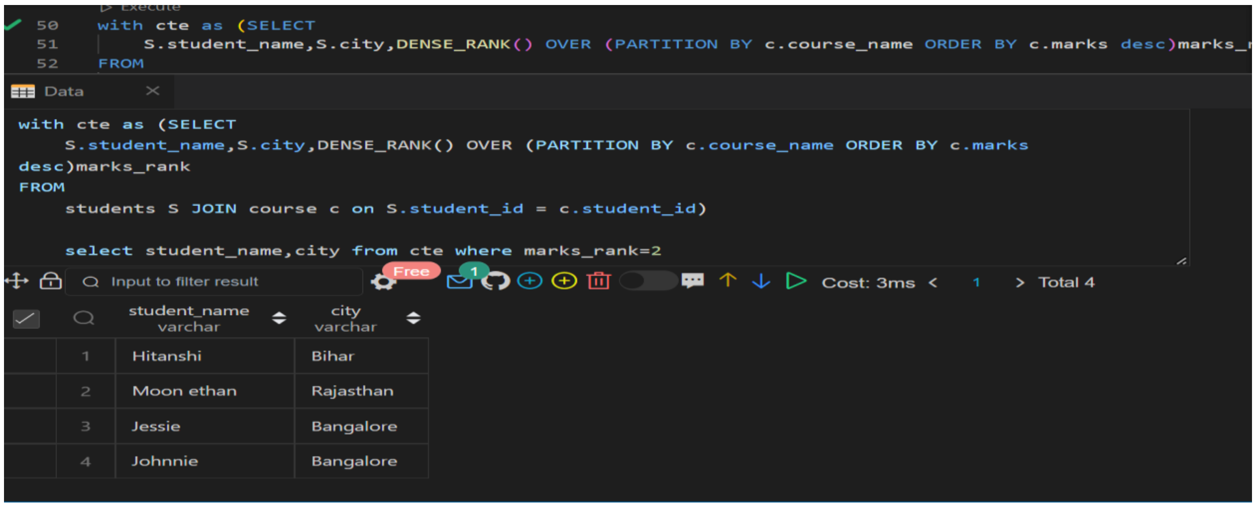
```
with cte as (SELECT
    S.student_name,DENSE_RANK() OVER (PARTITION BY c.course_name ORDER BY c.marks )marks_rank
FROM
    students S JOIN course c on S.student_id = c.student_id)

    select student_name from cte where marks_rank=4;
```

```
with cte as (SELECT
    S.student_name,DENSE_RANK() OVER (PARTITION BY c.course_name ORDER BY c.marks )marks_rank
FROM
    students S JOIN course c on S.student_id = c.student_id)

    select student_name from cte where marks_rank=4
```

	student_name varchar
1	Benbrook
2	Johnnie

q5. write a query to fetch the city name of the students who have 2nd highest marks?

with cte as
(SELECT S.student_name,S.city,DENSE_RANK() OVER (PARTITION BY c.course_name ORDER BY c.marks desc)marks_rank FROM students S JOIN course c on S.student_id = c.student_id)
select student_name,city from cte where marks_rank=2;



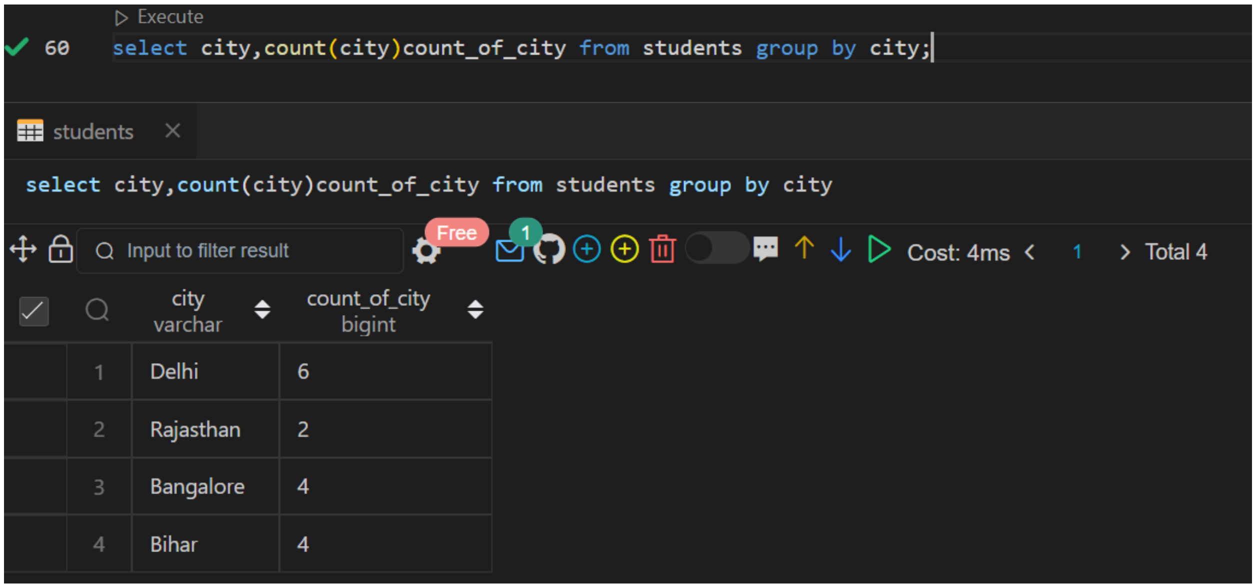
```
with cte as (SELECT
    S.student_name,S.city,DENSE_RANK() OVER (PARTITION BY c.course_name ORDER BY c.marks desc)marks_
FROM
```

```
with cte as (SELECT
    S.student_name,S.city,DENSE_RANK() OVER (PARTITION BY c.course_name ORDER BY c.marks
desc)marks_rank
FROM
    students S JOIN course c on S.student_id = c.student_id)

    select student_name,city from cte where marks_rank=2
```

	student_name varchar	city varchar
1	Hitanshi	Bihar
2	Moon ethan	Rajasthan
3	Jessie	Bangalore
4	Johnnie	Bangalore

q6. write a query to fetch the count of each city?

select city,count(city)count_of_city from students group by city;



```
select city,count(city)count_of_city from students group by city;
```

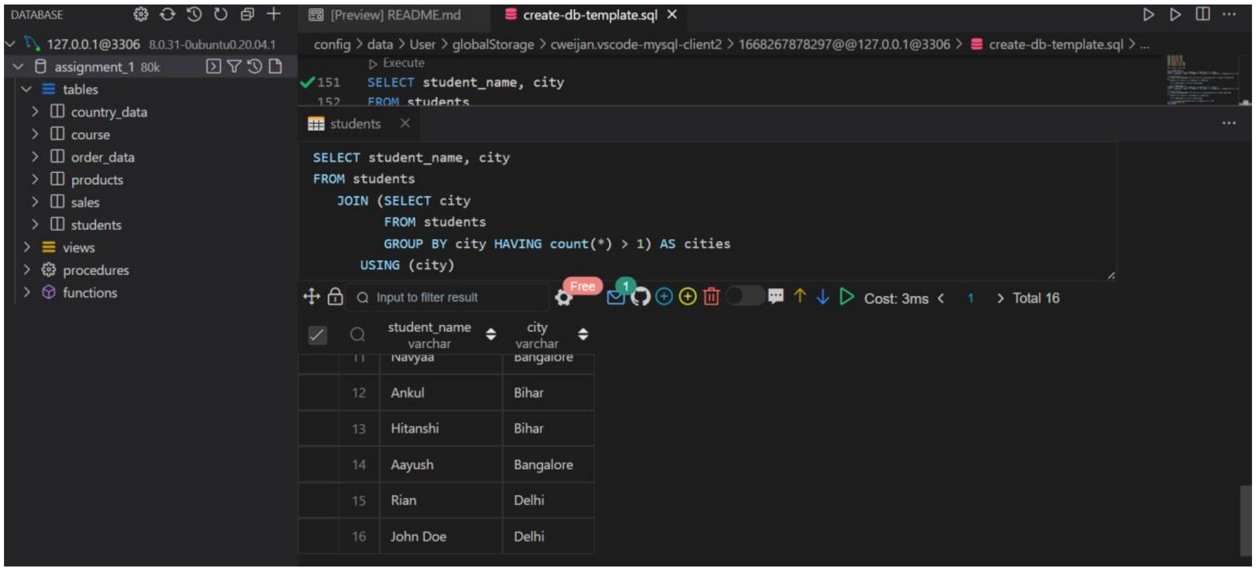
```
select city,count(city)count_of_city from students group by city
```

Cost: 4ms Total 4

	city varchar	count_of_city bigint
1	Delhi	6
2	Rajasthan	2
3	Bangalore	4
4	Bihar	4

q7. write a query to fetch the names of the students who are from the same city?

SELECT student_name, city FROM students JOIN (SELECT city FROM students GROUP BY city HAVING count(*) > 1) AS cities USING (city);



config > data > User > globalStorage > cweijan.vscode-mysql-client2 > 1668267878297@@127.0.0.1@3306 > create-db-template.sql > ...

```
SELECT student_name, city
FROM students
   JOIN (SELECT city
         FROM students
         GROUP BY city HAVING count(*) > 1) AS cities
      USING (city)
```

Cost: 3ms Total 16

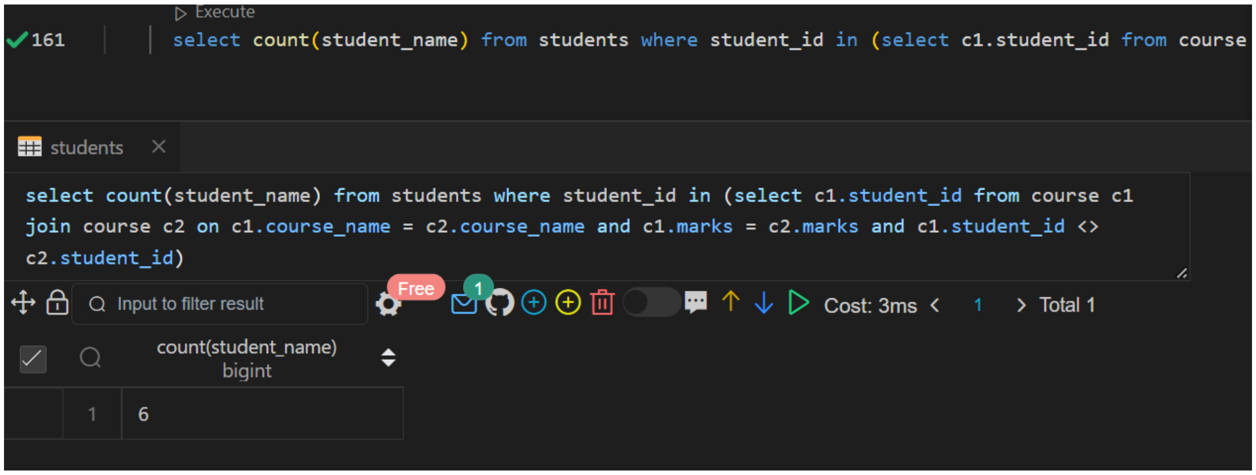
	student_name varchar	city varchar
11	Navyaa	Bangalore
12	Ankul	Bihar
13	Hitanshi	Bihar
14	Aayush	Bangalore
15	Rian	Delhi
16	John Doe	Delhi

q8.write a query to fetch the names of students starting with 'A'?

select student_name from students where student_name like 'A%';

q9.write a query to fetch the count of students' names having the same marks in each course?

select count(student_name) from students where student_id in (select c1.student_id from course c1 join course c2 on c1.course_name = c2.course_name and c1.marks = c2.marks and c1.student_id <> c2.student_id);



```
select count(student_name) from students where student_id in (select c1.student_id from course
```

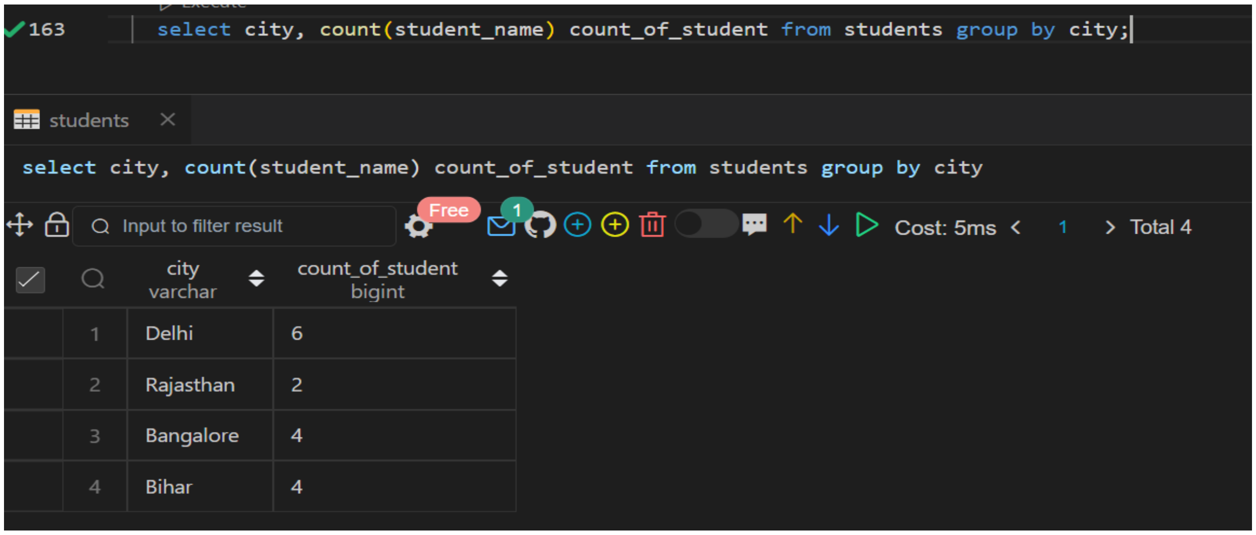
```
select count(student_name) from students where student_id in (select c1.student_id from course c1
join course c2 on c1.course_name = c2.course_name and c1.marks = c2.marks and c1.student_id <>
c2.student_id)
```

Cost: 3ms Total 1

	count(student_name) bigint
1	6

q10.write a query to fetch the count of students from each city?

select city, count(student_name) count_of_student from students group by city;



```
select city, count(student_name) count_of_student from students group by city;
```

```
select city, count(student_name) count_of_student from students group by city
```

Cost: 5ms Total 4

	city varchar	count_of_student bigint
1	Delhi	6
2	Rajasthan	2
3	Bangalore	4
4	Bihar	4

Hint : You must use Joins, Windows functions and CTE

8. Create a table below.

| Column Name | Type |
+--------------+---------+
player_id	int
device_id	int
event_date	date
games_played	int

(player_id, event_date) is the primary key of this table. This table shows the activity of players of some games.Each row is a record of a player who logged in and played a number of games (possibly 0)before logging out on someday using some device.

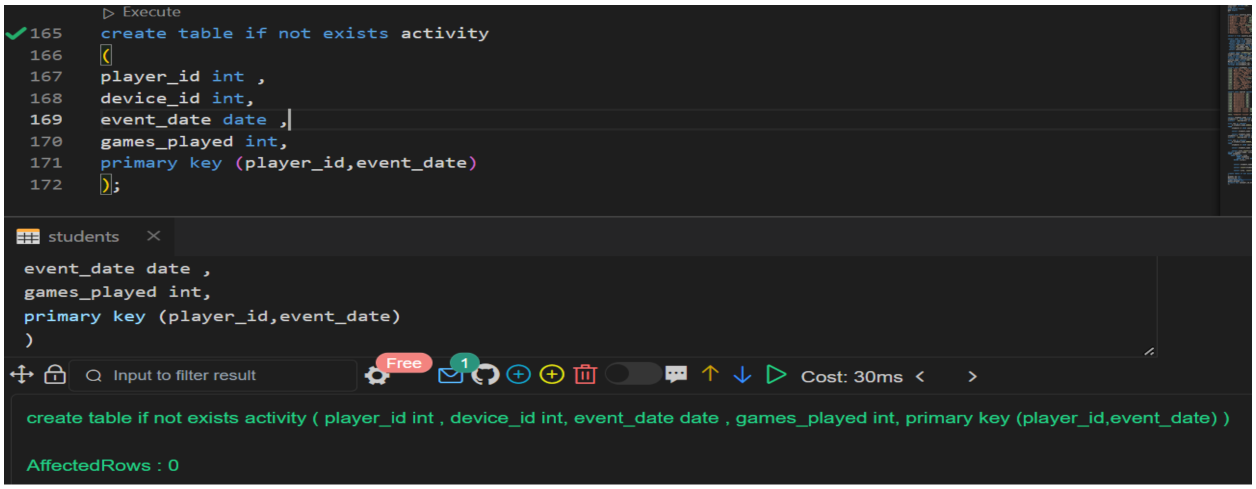
Write an SQL query to report the first login date for each player. Return the result table in any order.
The query result format is in the following example.
Input:
 Activity table:

```
+-----------+-----------+------------+--------------+
| player_id | device_id | event_date | games_played |
+-----------+-----------+------------+--------------+
| 1 | 2 | 2016-03-01 | 5 |
| 1 | 2 | 2016-05-02 | 6 |
| 2 | 3 | 2017-06-25 | 1 |
| 3 | 1 | 2016-03-02 | 0 |
| 3 | 4 | 2018-07-03 | 5 |
+-----------+-----------+------------+--------------+
```

Output:

```
+-----------+-------------+
| player_id | first_login |
+-----------+-------------+
| 1 | 2016-03-01 |
| 2 | 2017-06-25 |
| 3 | 2016-03-02 |
+-----------+-------------+
```

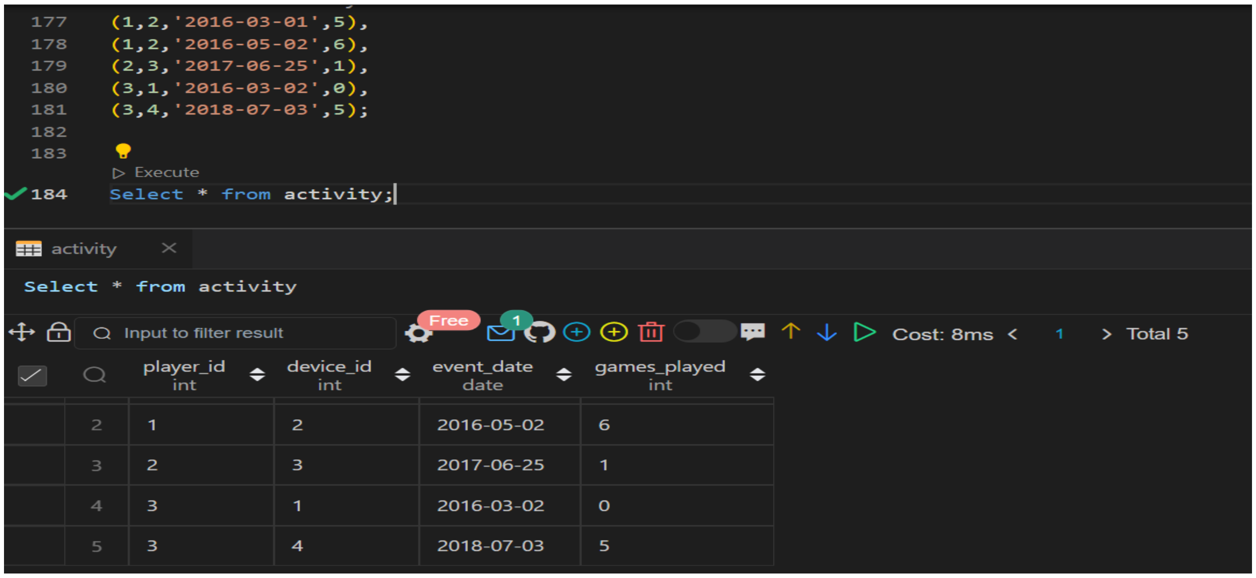
```
create table if not exists activity
(
player_id int ,
device_id int,
event_date date ,
games_played int
primary key (player_id,event_date)
);

insert into activity values
(1,2,'2016-03-01',5),
(1,2,'2016-05-02',6),
(2,3,'2017-06-25',1),
(3,1,'2016-03-02',0),
(3,4,'2018-07-03',5);
```



```
create table if not exists activity
(
player_id int ,
device_id int,
event_date date ,
games_played int,
primary key (player_id,event_date)
);
```

```
event_date date ,
games_played int,
primary key (player_id,event_date)
)
```

```
create table if not exists activity ( player_id int , device_id int, event_date date , games_played int, primary key (player_id,event_date) )

AffectedRows : 0
```

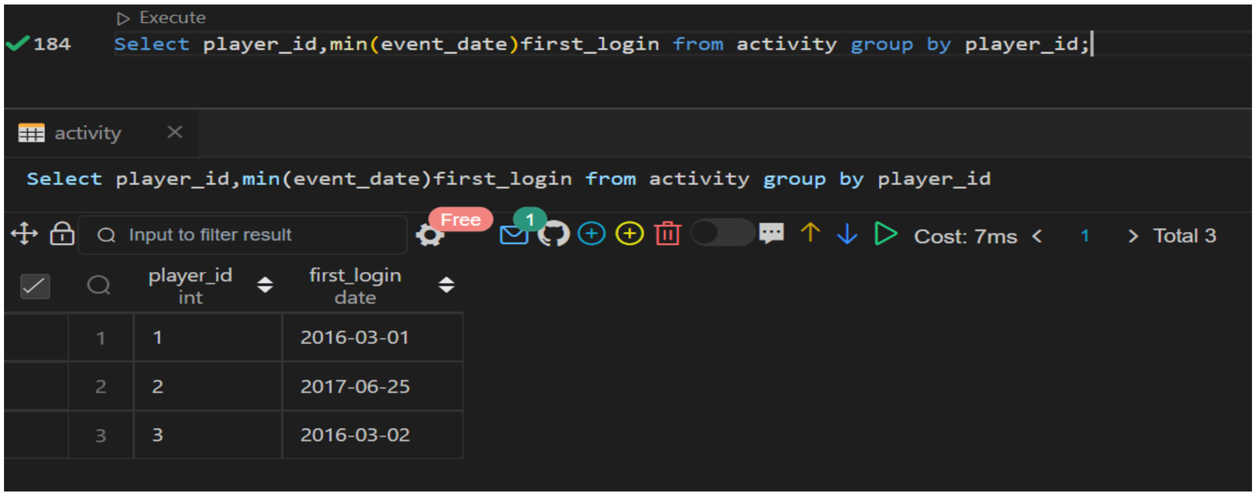


```
(1,2,'2016-03-01',5),
(1,2,'2016-05-02',6),
(2,3,'2017-06-25',1),
(3,1,'2016-03-02',0),
(3,4,'2018-07-03',5);


Select * from activity;
```

```
Select * from activity
```

	player_id int	device_id int	event_date date	games_played int
2	1	2	2016-05-02	6
3	2	3	2017-06-25	1
4	3	1	2016-03-02	0
5	3	4	2018-07-03	5

Select player_id,min(event_date)first_login from activity group by player_id;



```
Select player_id,min(event_date)first_login from activity group by player_id;
```

```
Select player_id,min(event_date)first_login from activity group by player_id
```

	player_id int	first_login date
1	1	2016-03-01
2	2	2017-06-25
3	3	2016-03-02

9. Create a table below.

```
+-------------+---------+
| Column Name | Type |
+-------------+---------+
| product_id | int |
| low_fats | enum |
| recyclable | enum |
+-------------+---------+
```

product_id is the primary key for this table. low_fats is an ENUM of type ('Y', 'N') where 'Y' means this product is low fat and 'N' means it is not. recyclable is an ENUM of types ('Y', 'N') where 'Y' means this product is recyclable and 'N' means it is not.

Write an SQL query to find the ids of products that are both low fat and recyclable. Return the result table in any order. The query result format is in the following example.

Input:

Products table:

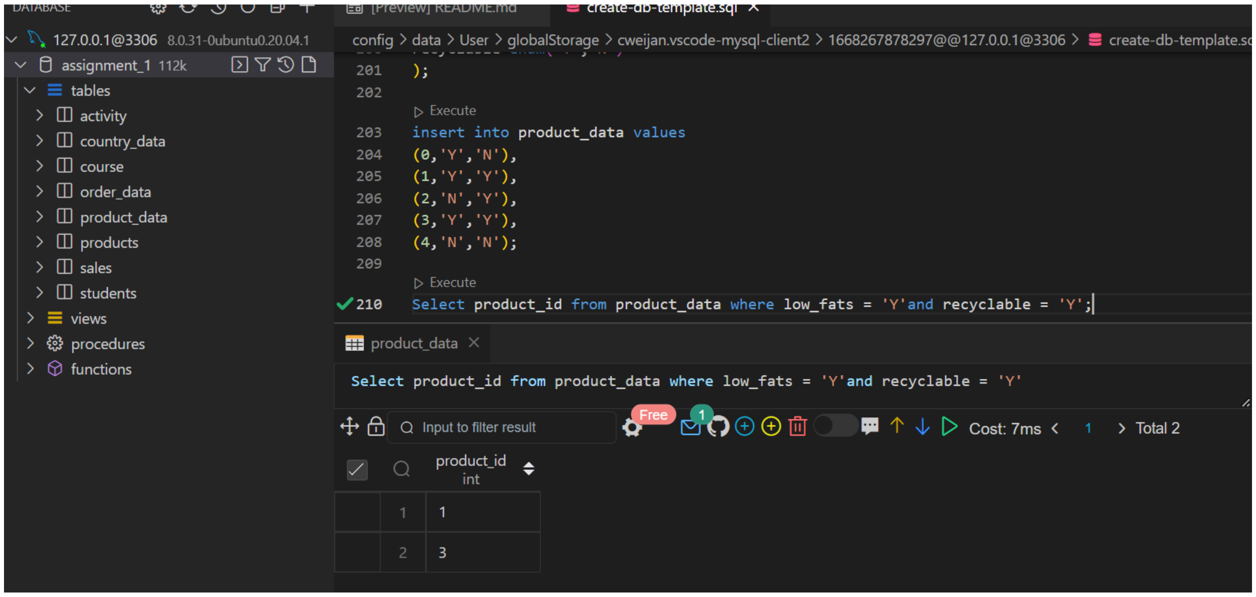
```
+-------------+----------+------------+
| product_id | low_fats | recyclable |
+-------------+----------+------------+
| 0 | Y | N |
| 1 | Y | Y |
| 2 | N | Y |
| 3 | Y | Y |
| 4 | N | N |
+-------------+----------+------------+
```

Output:

```
+-------------+
| product_id |
+-------------+
| 1 |
| 3 |
+-------------+
```

```
create table product_data
(
product_id int primary key,
low_fats enum('Y','N'),
recyclable enum('Y','N')
);

insert into product_data values
(0,'Y','N'),
(1,'Y','Y'),
(2,'N','Y'),
(3,'Y','Y'),
(4,'N','N');
```

Select product_id from product_data where low_fats = 'Y'and recyclable = 'Y';



```
insert into product_data values
(0,'Y','N'),
(1,'Y','Y'),
(2,'N','Y'),
(3,'Y','Y'),
(4,'N','N');

Select product_id from product_data where low_fats = 'Y'and recyclable = 'Y';
```

```
Select product_id from product_data where low_fats = 'Y'and recyclable = 'Y'
```

```
product_id
int
1  1
2  3
```

Cost: 7ms Total 2

10. Create a table below.

name region area population gdp
Afghanistan South Asia 652225 26000000
Albania Europe 28728 3200000 6656000000
Algeria MiddleEast 2400000 32900000 75012000000
Andorra Europe 468 64000
...

1.Select the statement that shows the sum of population of all countries .
Select name,sum(population) from countries group by (name);
2. Select the statement that shows the number of countries with population smaller than 150000
SELECT COUNT(name) FROM countries WHERE population < 150000;
3. Select the list of core SQL aggregate functions
AVG(), COUNT(), MAX(), MIN(), SUM()
4. Select the result that would be obtained from the following code:
5. Select the statement that shows the average population of 'Poland', 'Germany' and 'Denmark'
SELECT AVG(population) FROM countries WHERE name IN ('Poland', 'Germany', 'Denmark');
6. Select the statement that shows the medium population density of each region
SELECT region, SUM(population)/SUM(area) AS density FROM countries GROUP BY region;
7. Select the statement that shows the name and population density of the country with the largest population
SELECT name, population/area AS density FROM countries WHERE population = (SELECT MAX(population) FROM countries);