CS3120 Database Management Systems Laboratory

Lab

Mayank Singla
111901030

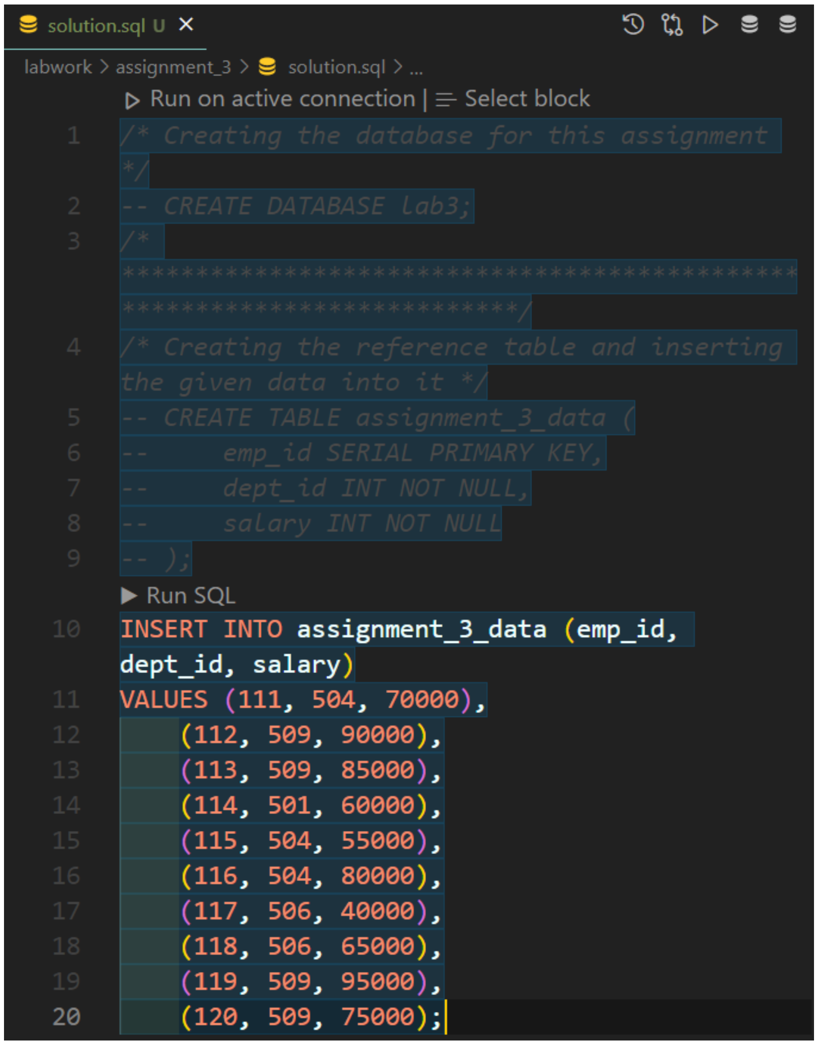
Using VSCode to write the queries in a file and then execute that file from the terminal.
I have pasted the queries as text in code block formatting, its screenshot from VSCode, and the output of executing it in the terminal.

Q0. Creating the database, reference table and inserting the given data into it.

```
CREATE DATABASE lab3;
```

```
CREATE TABLE assignment_3_data (
    emp_id SERIAL PRIMARY KEY,
    dept_id INT NOT NULL,
    salary INT NOT NULL
);
INSERT INTO assignment_3_data (emp_id, dept_id, salary)
VALUES (111, 504, 70000),
    (112, 509, 90000),
    (113, 509, 85000),
    (114, 501, 60000),
    (115, 504, 55000),
    (116, 504, 80000),
    (117, 506, 40000),
    (118, 506, 65000),
    (119, 509, 95000),
    (120, 509, 75000);
```

For creating the database and table I am using `CREATE DATABASE` and `CREATE TABLE` as normal use and then inserting the values into the created table using `INSERT INTO`


solution.sql U ×

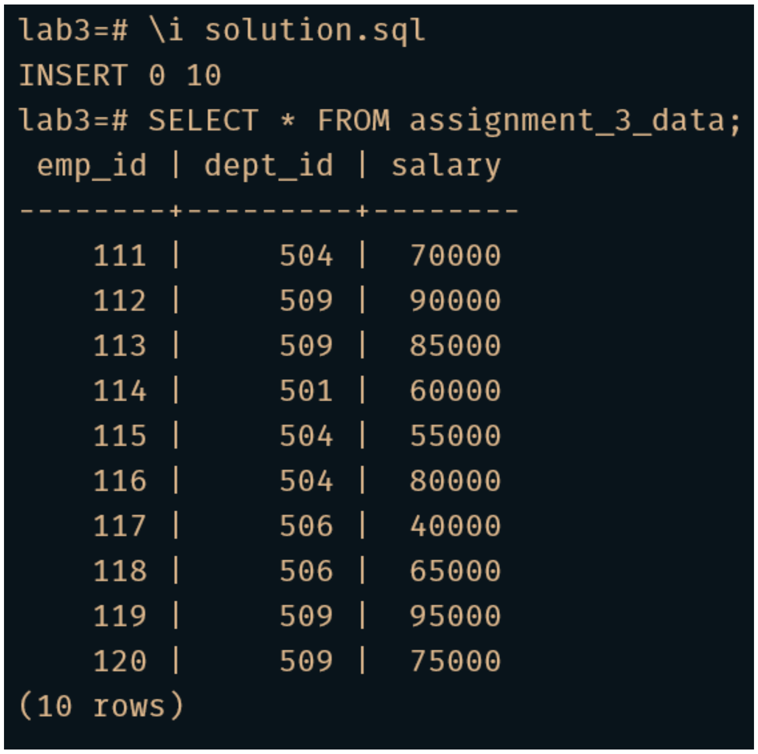
labwork > assignment_3 > solution.sql > ...

▷ Run on active connection | ≡ Select block

```sql
/* Creating the database for this assignment */
-- CREATE DATABASE lab3;
/* ****************************************************************/
/* Creating the reference table and inserting the given data into it */
-- CREATE TABLE assignment_3_data (
--      emp_id SERIAL PRIMARY KEY,
--      dept_id INT NOT NULL,
--      salary INT NOT NULL
-- );
```

▶ Run SQL

```sql
INSERT INTO assignment_3_data (emp_id, dept_id, salary)
VALUES (111, 504, 70000),
       (112, 509, 90000),
       (113, 509, 85000),
       (114, 501, 60000),
       (115, 504, 55000),
       (116, 504, 80000),
       (117, 506, 40000),
       (118, 506, 65000),
       (119, 509, 95000),
       (120, 509, 75000);
```

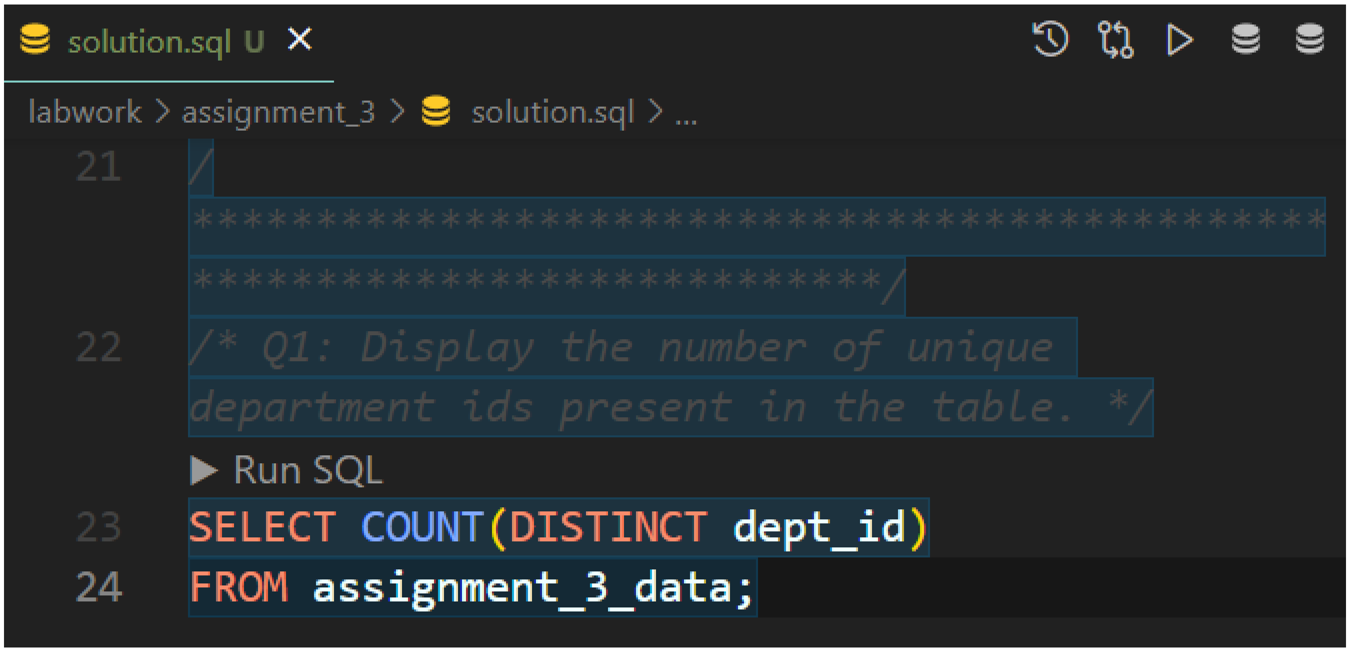


```
lab3=# \i solution.sql
INSERT 0 10
lab3=# SELECT * FROM assignment_3_data;
 emp_id | dept_id | salary
--------+---------+--------
    111 |     504 |  70000
    112 |     509 |  90000
    113 |     509 |  85000
    114 |     501 |  60000
    115 |     504 |  55000
    116 |     504 |  80000
    117 |     506 |  40000
    118 |     506 |  65000
    119 |     509 |  95000
    120 |     509 |  75000
(10 rows)
```

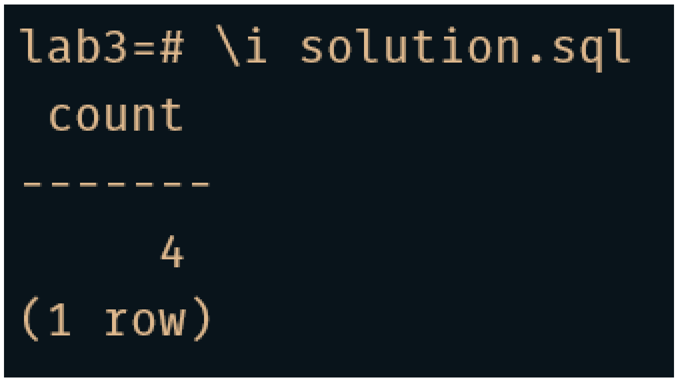
Q1. Display the number of unique department ids present in the table.

```sql
SELECT COUNT(DISTINCT dept_id)
FROM assignment_3_data;
```

I need to display the number of unique department ids, so for the number, I am using `COUNT`, and for displaying the unique ids, I am using `DISTINCT`



```sql
/**********************************************************************/
/* Q1: Display the number of unique department ids present in the table. */
SELECT COUNT(DISTINCT dept_id)
FROM assignment_3_data;
```

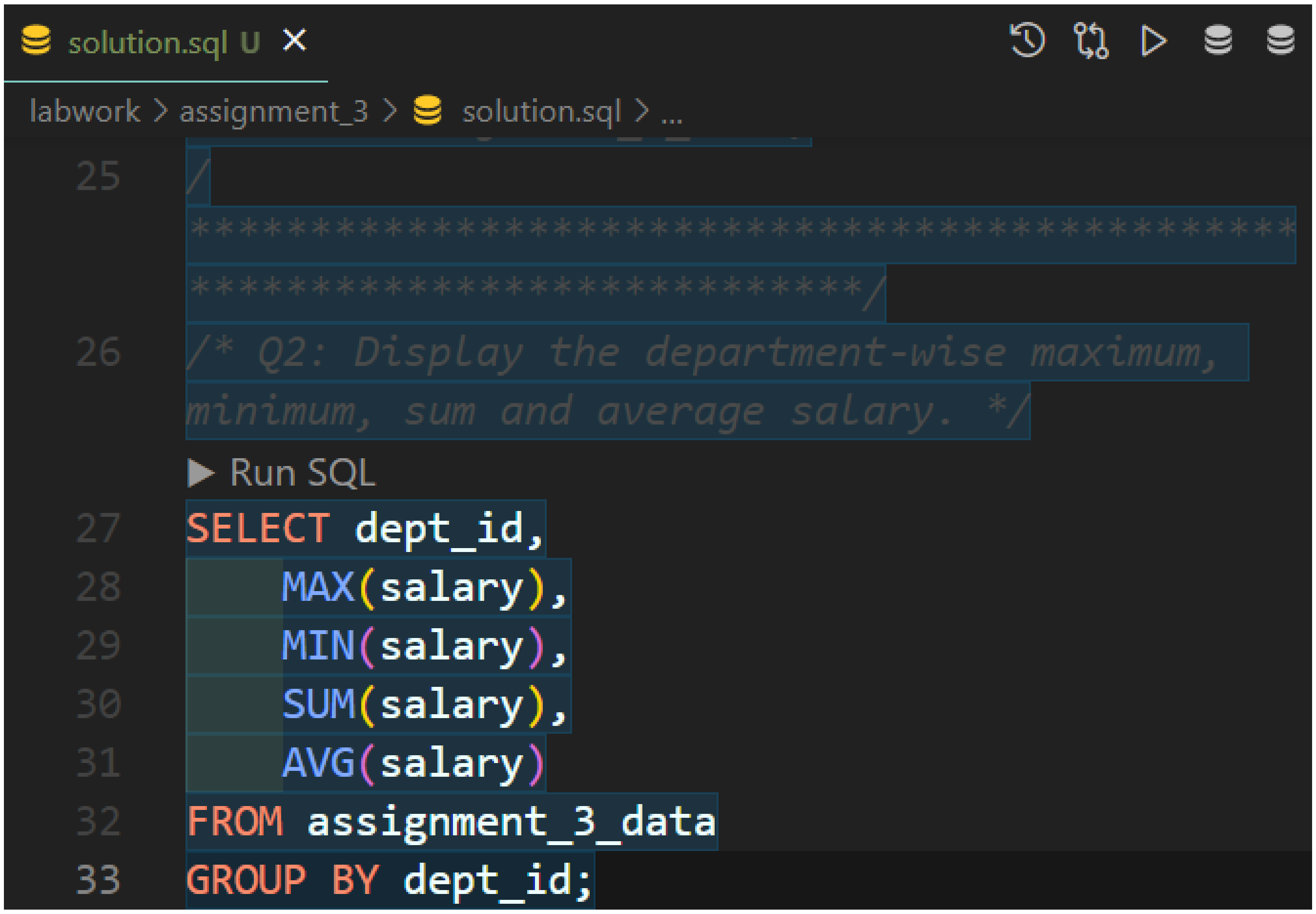


```
lab3=# \i solution.sql
 count
-------
     4
(1 row)
```

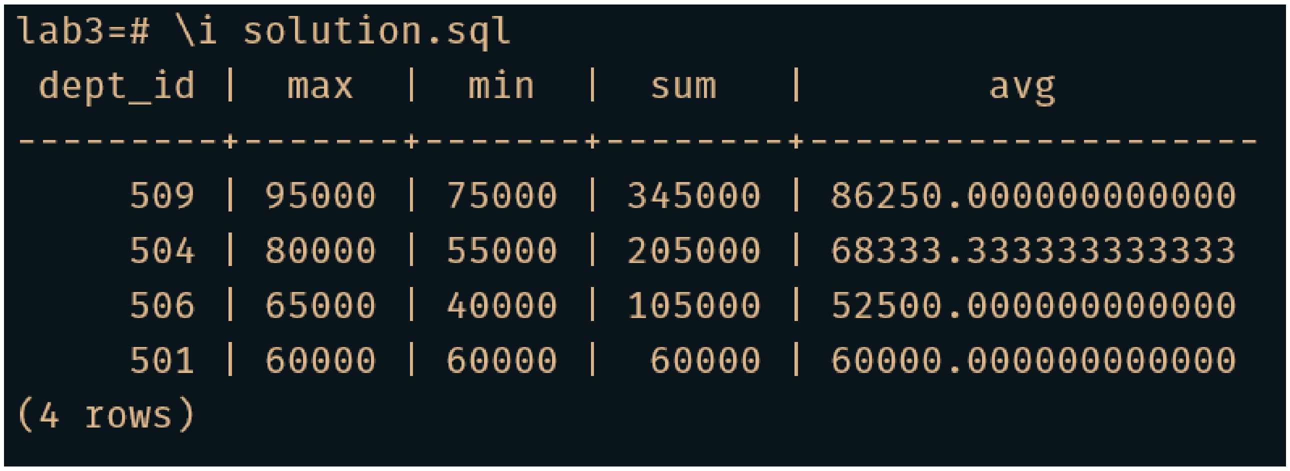
Q2. Display the department-wise maximum, minimum, sum, and average salary.

```sql
SELECT dept_id,
    MAX(salary),
    MIN(salary),
    SUM(salary),
    AVG(salary)
FROM assignment_3_data
GROUP BY dept_id;
```

I need to display those quantities department-wise, that is why I am first grouping the rows by their department ids and then using the aggregating functions to find the required quantities.



```sql
/
*********************************************************************/
/* Q2: Display the department-wise maximum,
minimum, sum and average salary. */
SELECT dept_id,
    MAX(salary),
    MIN(salary),
    SUM(salary),
    AVG(salary)
FROM assignment_3_data
GROUP BY dept_id;
```

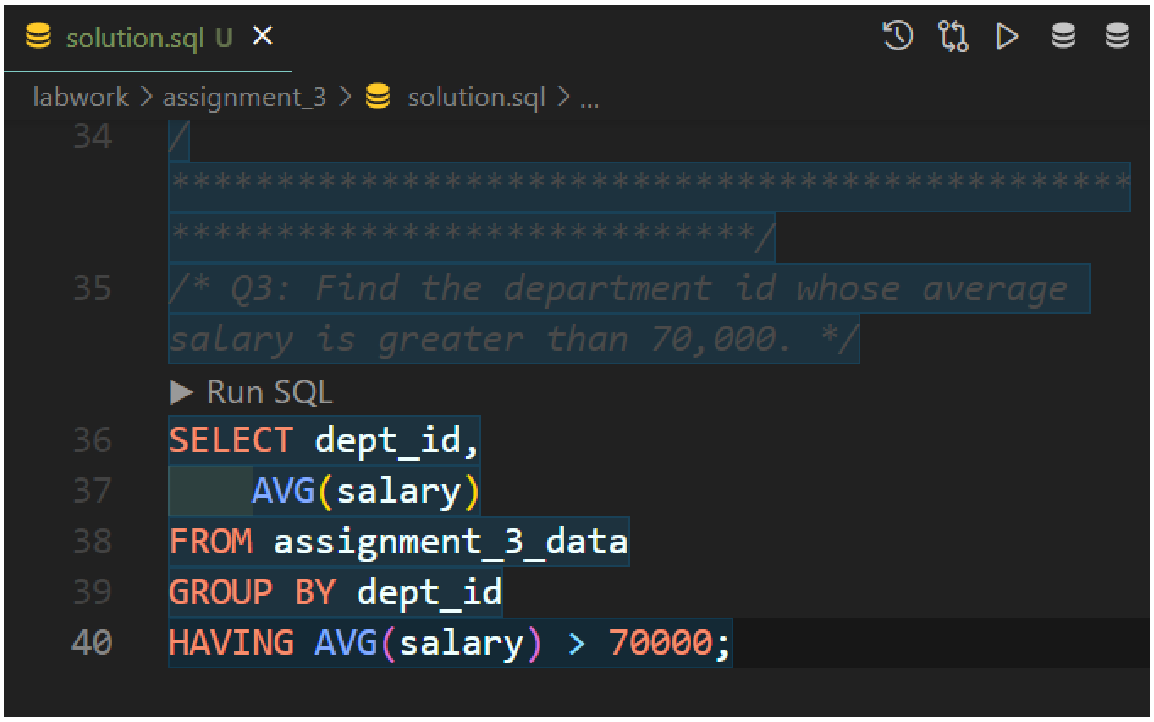


```
lab3=# \i solution.sql
 dept_id |  max  |  min  |  sum   |         avg
---------+-------+-------+--------+--------------------
     509 | 95000 | 75000 | 345000 | 86250.000000000000
     504 | 80000 | 55000 | 205000 | 68333.333333333333
     506 | 65000 | 40000 | 105000 | 52500.000000000000
     501 | 60000 | 60000 |  60000 | 60000.000000000000
(4 rows)
```

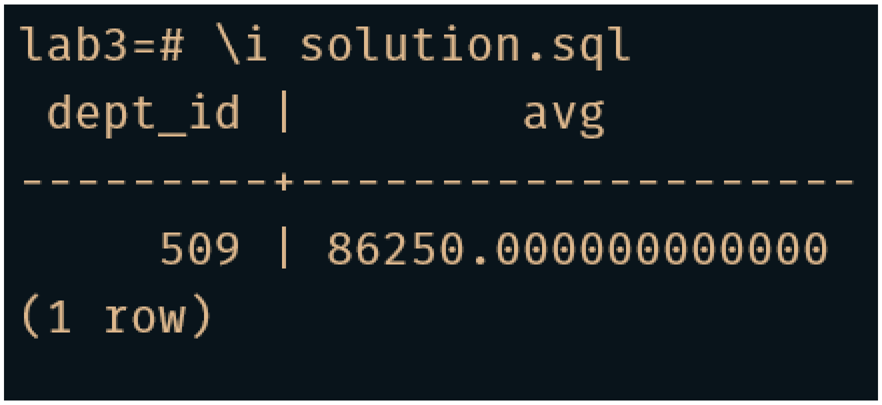
Q3. Find the department id whose average salary is greater than 70,000.

```
SELECT dept_id,
    AVG(salary)
FROM assignment_3_data
GROUP BY dept_id
HAVING AVG(salary) > 70000;
```

As I need to query over the department id and find the average salary for each department id, I am first grouping them by department id and then using the `HAVING` clause I am imposing the conditions on the groups to get only those groups that satisfy that condition.



```
/*********************************************************************/
/* Q3: Find the department id whose average salary is greater than 70,000. */
SELECT dept_id,
    AVG(salary)
FROM assignment_3_data
GROUP BY dept_id
HAVING AVG(salary) > 70000;
```

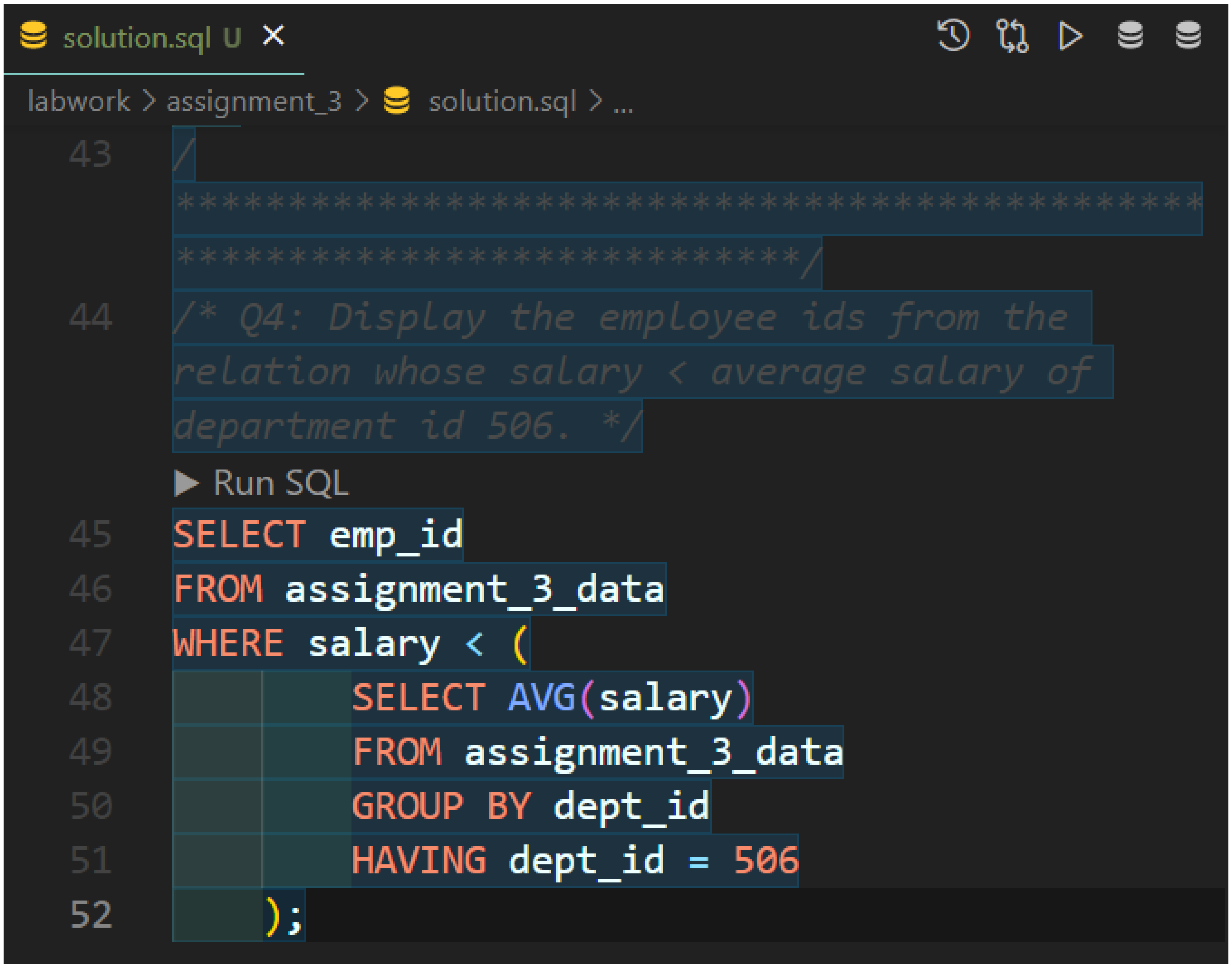


```
lab3=# \i solution.sql
 dept_id |        avg
---------+--------------------
     509 | 86250.000000000000
(1 row)
```

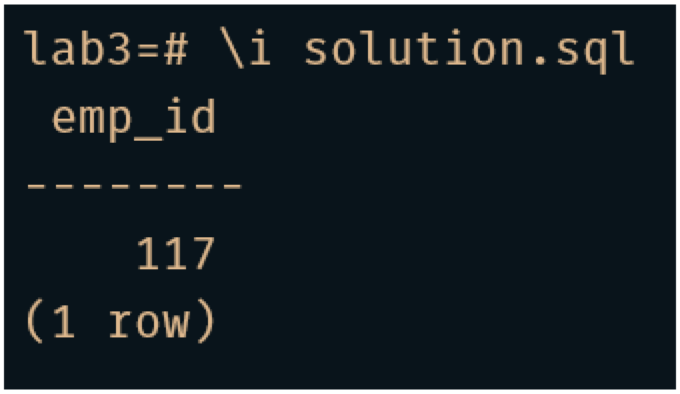
Q4. Display the employee ids from the relation whose salary < average salary of department id 506.

```
SELECT emp_id
FROM assignment_3_data
WHERE salary < (
        SELECT AVG(salary)
        FROM assignment_3_data
        GROUP BY dept_id
        HAVING dept_id = 506
    );
```

To get the average salary of department id 506, I need to perform a separate query for that which I am doing as a nested sub-query by first grouping all the rows by department id and then selecting the department id 506 and taking the average salary of it. Then I am performing a simple select query on the employee ids using the `WHERE` clause and extracting those which satisfy the condition mentioned in the question.



```
/*************************************************************************/
/* Q4: Display the employee ids from the relation whose salary < average salary of department id 506.  */
SELECT emp_id
FROM assignment_3_data
WHERE salary < (
        SELECT AVG(salary)
        FROM assignment_3_data
        GROUP BY dept_id
        HAVING dept_id = 506
    );
```

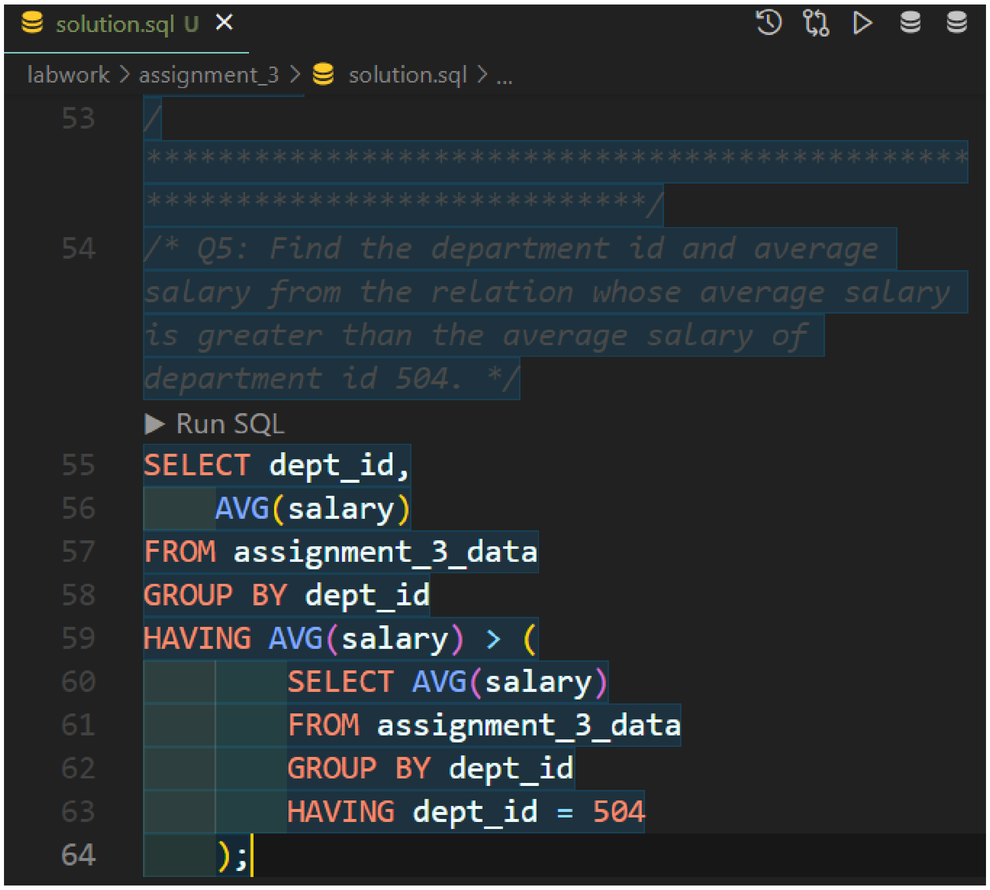


```
lab3=# \i solution.sql
 emp_id
--------
    117
(1 row)
```

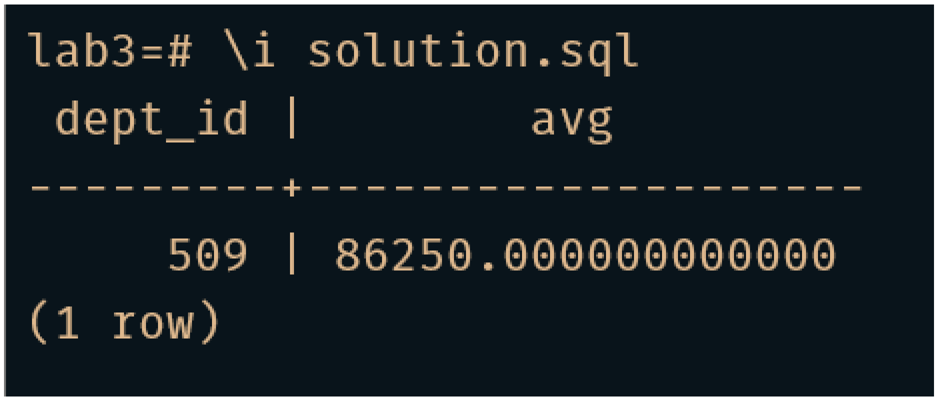
Q5. Find the department id and average salary from the relation whose average salary is greater than the average salary of department id 504.

```sql
SELECT dept_id,
    AVG(salary)
FROM assignment_3_data
GROUP BY dept_id
HAVING AVG(salary) > (
        SELECT AVG(salary)
        FROM assignment_3_data
        GROUP BY dept_id
        HAVING dept_id = 504
    );
```

First I need to find the average salary of department id 504, for that, I am doing a nested sub-query as done in the previous question to get the average salary for that department. Then, I need to get the department id and average salary whose average salary is greater than the found average salary. For that, I am first grouping the rows by department id and then taking the average of their salaries and comparing from the found average salary in nested sub-query.



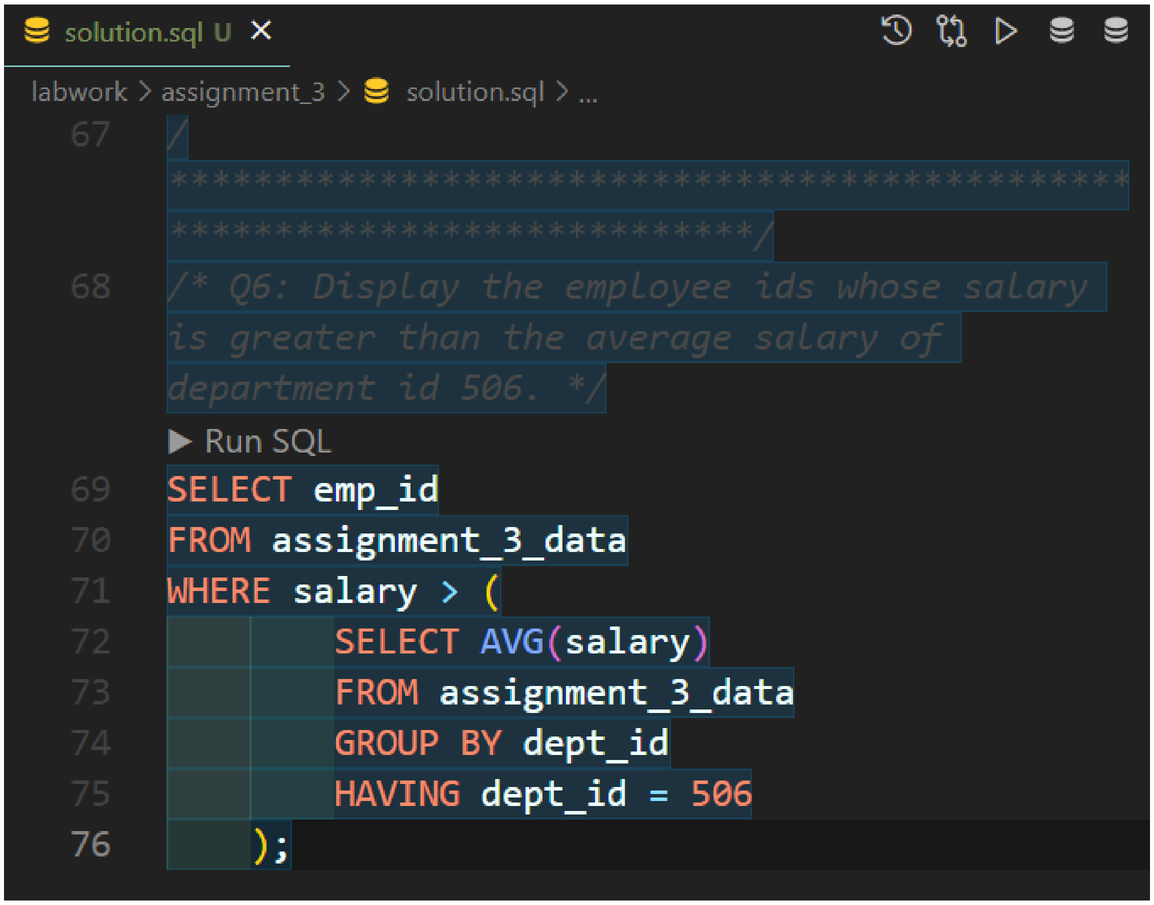
```sql
/*******************************************************************/
/* Q5: Find the department id and average salary from the relation whose average salary is greater than the average salary of department id 504. */
SELECT dept_id,
    AVG(salary)
FROM assignment_3_data
GROUP BY dept_id
HAVING AVG(salary) > (
        SELECT AVG(salary)
        FROM assignment_3_data
        GROUP BY dept_id
        HAVING dept_id = 504
    );
```


```
lab3=# \i solution.sql
 dept_id |         avg
---------+---------------------
     509 | 86250.000000000000
(1 row)
```

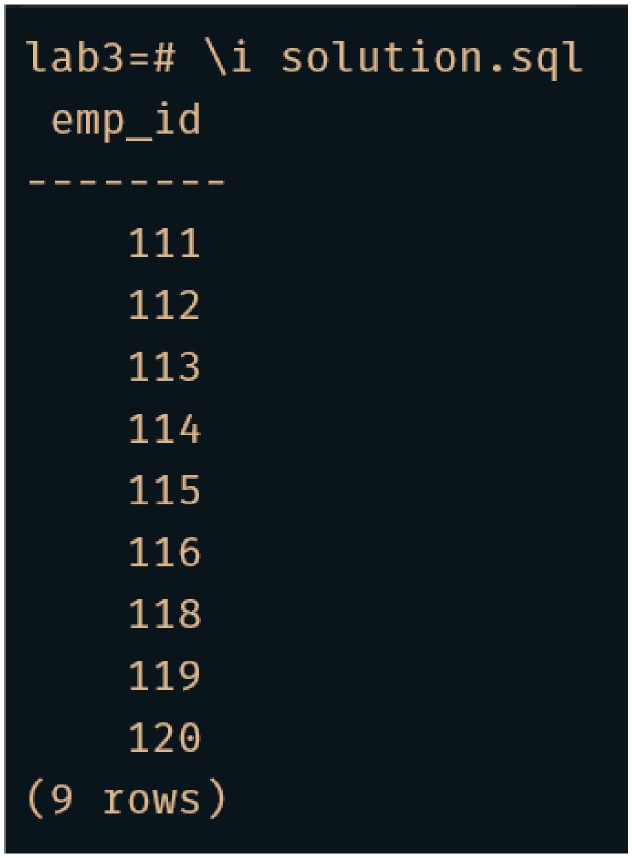
Q6. Display the employee ids whose salary is greater than the average salary of department id 506.

```
SELECT emp_id
FROM assignment_3_data
WHERE salary > (
        SELECT AVG(salary)
        FROM assignment_3_data
        GROUP BY dept_id
        HAVING dept_id = 506
    );
```

Similar explanation as in Q4, just reversing the inequality symbol.



```sql
/****************************************************************************************/
/* Q6: Display the employee ids whose salary is greater than the average salary of department id 506.  */
SELECT emp_id
FROM assignment_3_data
WHERE salary > (
        SELECT AVG(salary)
        FROM assignment_3_data
        GROUP BY dept_id
        HAVING dept_id = 506
    );
```



```
lab3=# \i solution.sql
 emp_id
--------
    111
    112
    113
    114
    115
    116
    118
    119
    120
(9 rows)
```