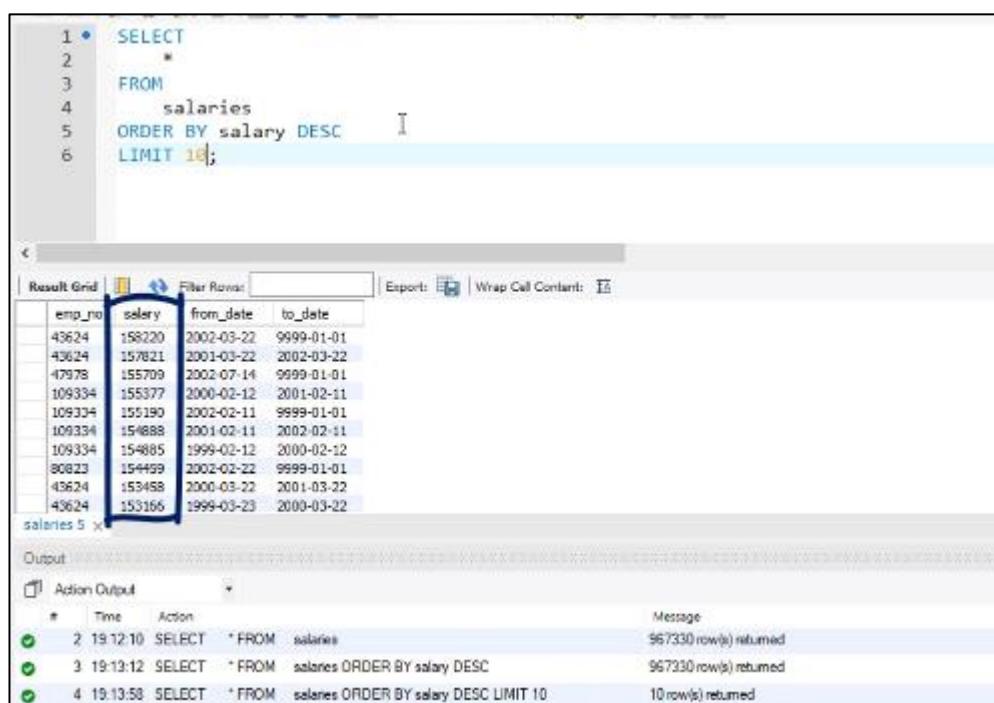


LIMIT

 SQL

```
SELECT column_name(s)
FROM table_name
WHERE conditions
GROUP BY column_name(s)
HAVING conditions
ORDER BY column_name(s)
LIMIT number ;
```



The screenshot shows a MySQL Workbench session. In the query editor, a SELECT statement is written:

```
1 • SELECT
2   *
3   FROM
4     salaries
5   ORDER BY salary DESC
6   LIMIT 10;
```

The results grid displays 10 rows from the salaries table, ordered by salary in descending order. The columns are emp_no, salary, from_date, and to_date.

emp_no	salary	from_date	to_date
43624	158220	2002-03-22	9999-01-01
43624	157821	2001-03-22	2002-03-22
47978	155709	2002-07-14	9999-01-01
109334	155377	2000-02-12	2001-02-11
109334	155190	2002-02-11	9999-01-01
109334	154888	2001-02-11	2002-02-11
109334	154885	1999-02-12	2000-02-12
80823	154459	2002-02-22	9999-01-01
43624	153458	2000-03-22	2001-03-22
43624	153166	1999-03-23	2000-03-22

Output pane shows the execution log:

Action	Time	Action	Message
1	19:12:10	SELECT * FROM salaries	967330 row(s) returned
2	19:13:12	SELECT * FROM salaries ORDER BY salary DESC	967330 row(s) returned
3	19:13:58	SELECT * FROM salaries ORDER BY salary DESC LIMIT 10	10 row(s) returned

The lecturer also explained the other ways one can limit the output of our queries – it's not hard whatsoever, I guess you can always go back to Section 21 video 175 should it ever become something of an enigma, but I highly doubt it.

COUNT(*)

* returns all rows of the table, **NULL values included**

~~SUM(*)~~

* goes well with only the COUNT() function

COUNT() - applicable to both *numeric and non-numeric* data

SUM()

MIN()

MAX()

AVG()

- work *only with numeric* data