

## **Homework 2**

1. Write a SQL query to display employee\_id, first\_name, last\_name, phone\_number, hire\_date as start date.

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Includes icons for file operations, database management, and search.
- Left Panel:** Shows the "Schemas" tree with the "company" schema selected. Other schemas listed are sakila, school, and sys.
- Query Editor:** Displays the query: `SELECT employee_id, first_name, last_name, phone_number, hire_date as "start date" FROM employees;`. A note on the right says: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."
- Result Grid:** Shows the results of the query in a tabular format. The columns are employee\_id, first\_name, last\_name, phone\_number, and start date. The data includes rows from 100 to 178, with various names and hire dates.
- Right Sidebar:** Contains links to other tools: Result Grid, Form Editor, Field Types, Query Stats, and Execution Plan.
- Bottom Panel:** Shows the "Action Output" section with one row of data: Time (19:35:25), Action (SELECT employee\_id, first\_name, last\_name, phone\_number, hire\_date...), Response (40 row(s) returned), and Duration / Fetch Time (0.00060 sec / 0.000...).

## 2. What is the min salary on the employee's table?

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Includes icons for Home, Local instance 3306, Administration, Schemas, Query 1, SQL File 3\*, and Context Help.
- Schemas Panel:** Shows available schemas: company, sakila, school, and sys. The company schema is selected.
- Query Editor:** Displays the SQL query: `SELECT MIN(salary) FROM employees;`. A tooltip on the right states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."
- Result Grid:** Shows the results of the query. The first row is labeled "MIN..." and contains the value "250...".
- Action Output:** Shows the history of actions taken. One entry is visible: "1 19:38:07 SELECT MIN(salary) FROM employees LIMIT 0,1000".
- Status Bar:** Shows "Query Completed".

3. Display the first name, last name, salaries for employees belonging to Department\_id 6. Sort by salary descending, first name, last name.

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the schema tree with 'company' selected. The main area contains a query editor with the following SQL code:

```
1 •  SELECT first_name, last_name, salary FROM employees WHERE department_id = 6 ORDER BY
2      salary DESC, first_name, last_name;
```

The results grid shows the following data:

first_name	last_name	salary
Alexander	Hunold	9000.00
Bruce	Ernest	8000.00
David	Austin	4800.00
Valli	Pataballa	4800.00
Diana	Lorentz	4200.00

The status bar at the bottom indicates 'Query Completed'.

#### 4. Display all details of the employee whose PHONE\_number is null

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'company' selected. The main area contains a SQL editor with the following query:

```
1 -- 4
2 • SELECT * FROM employees WHERE phone_number IS NULL;
```

The results grid shows the following data:

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	manager_id	department...
145	John	Russell	john.russell@sqltutorial.org	NULL	1996-10-01	15	14000.00	100	8
146	Karen	Partners	karen.partners@sqltutorial.org	NULL	1997-01-05	15	13500.00	100	8
176	Jonathon	Taylor	jonathon.taylor@sqltutorial.org	NULL	1998-03-24	16	8800.00	100	8
177	Jack	Livingston	jack.livingston@sqltutorial.org	NULL	1998-04-23	16	8400.00	100	8
178	Kimberely	Grant	kimberely.grant@sqltutorial.org	NULL	1999-05-24	16	7000.00	100	8
179	Charles	Johnson	charles.johnson@sqltutorial.org	NULL	2000-01-04	16	6200.00	100	8

The results grid has 11 rows. The 'Action Output' pane shows the query and its execution details.

5. Update the phone\_number to 506.789.7890 if it is null. (Use SET SQL\_SAFE\_UPDATES = 0; before executing the 5th query)

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Standard MySQL Workbench toolbar with icons for file operations, schema browser, and query editor.
- Schemas Panel:** Shows available schemas: company, sakila, school, and sys. The company schema is selected.
- Query Editor:** Contains the following SQL script:

```
1 -- 5
2 • SET SQL_SAFE_UPDATES = 0;
3
4 • UPDATE employees SET PHONE_number = '506.789.7890' WHERE PHONE_number IS NULL;
5
6 • SET SQL_SAFE_UPDATES = 1;
```
- Output Panel:** Displays the execution log:

Action	Time	Response	Duration / Fetch Time
SET SQL_SAFE_UPDATES = 0	16:25:20	0 row(s) affected	0.014 sec
UPDATE employees SET PHONE_number = '506.789.7890' WHERE PH...	16:25:20	6 row(s) affected Rows m...	0.016 sec
SET SQL_SAFE_UPDATES = 1	16:25:20	0 row(s) affected	0.00011 sec
- Session Panel:** Shows the current schema is company.
- Status Bar:** Displays "Query Completed".

6. Write a SQL query to find employees whose salary is greater than 9000 in decreasing order of salary. (Display the combination of first and last name with space in between them as 'full\_name', salary)

The screenshot shows the Oracle SQL Developer interface. The top navigation bar includes tabs for Administration, Schemas, Query 1, and SQL File 3\*. The Schemas panel on the left lists SCHEMAS: company, sakila, school, and sys. The central workspace displays a query editor with the following code:

```
1 -- 6
2 • SELECT CONCAT(first_name, ' ', last_name) AS full_name, salary FROM employees WHERE salary > 9000
3 ORDER BY salary DESC;
```

A context help message on the right states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." Below the query editor is a Result Grid showing the output:

full_name	salary
Steven King	24000.00
Nneka Kochhar	17000.00
Lex De Haan	17000.00
John Russell	14000.00
Karen Partners	13000.00
Maria Hartstein	13000.00
Nancy Greenberg	12000.00
Shelley Higgins	12000.00
Den Raphaely	11000.00
Hermann Baer	10000.00

The bottom section shows the Action Output table with one entry:

Action	Time	Action	Response	Duration / Fetch Time
1	16:38:49	SELECT CONCAT(first_name, ' ', last_name) AS full_name, salary FRO...	10 row(s) returned	0.0094 sec / 0.0005...

7. Write a SQL query to find employees with salaries in the range 7000 and 9000 (Begin and end values are included) in decreasing order of salary. (Display Employee ID, Full name, and Salary)

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the schema tree with the 'company' schema selected. The main area contains a query editor with the following SQL code:

```
1 -- 7
2 • SELECT employee_id, CONCAT(first_name, ' ', last_name) AS full_name, salary
3   FROM employees WHERE salary BETWEEN 7000 AND 9000 ORDER BY salary DESC;
```

To the right of the query editor, a message states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

The results grid shows the output of the query, listing 12 employees with their employee ID, full name, and salary. The results are as follows:

employee_id	full_name	salary
103	Alexander Hunold	9000.00
109	Daniel Faviet	9000.00
176	Jonathon Taylor	8600.00
177	Jack Livingston	8400.00
200	William Gietz	8200.00
111	John Cullum	8200.00
121	Adam Fipp	8200.00
130	Matthew Weiss	8000.00
122	Payam Kaufling	7900.00
112	Jose Manuel Urman	7800.00
111	Ismail Scaria	7700.00
178	Kimberely Grant	7700.00

The session details show the query was executed at 16:41:29 and returned 12 rows in 0.0014 seconds.

8. Display all details of the employee in the employee table containing letters 'C' in their last name

The screenshot shows the MySQL Workbench interface. The top navigation bar includes tabs for Administration, Schemas, Query 1, and SQL File 3\*. The Schemas panel on the left shows the 'company' schema selected, containing tables like employee\_department, Views, Stored Procedures, and Functions. The Sakila, School, and Sys databases are also listed. The main area displays a query result grid for the 'employees' table, showing columns: employee\_id, first\_name, last\_name, email, phone\_number, hire\_date, job\_id, salary, manager\_id, and department\_id. The results show four rows of data. Below the grid, a status bar indicates 'employees 15' and provides 'Apply' and 'Revert' buttons. The bottom status bar shows the schema is 'company', the action is 'Output', and the most recent query was 'SELECT \* FROM employees WHERE last\_name LIKE '%C%' LIMIT 0, 10...'. The duration of the fetch was 0.000076 sec / 0.000000. A message in the top right corner states: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

Local instance 3306

Administration Schemas Query 1 SQL File 3\* Context Help Snippets

Limit to 1000 rows

Filter objects

company

Tables

employee\_department Views Stored Procedures Functions

sakila school sys

1 -- 8

2 • SELECT \* FROM employees WHERE last\_name LIKE '%C%';

Result Grid Filter Rows: Search Edit: Export/Import:

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	manager_id	department_id
101	Neena	Kochhar	neena.kochhar@sqltutorial.org	515.123.4568	1989-09-21	5	17000.00	100	9
110	John	Chen	john.chen@sqltutorial.org	515.124.4269	1997-09-28	6	8200.00	108	10
111	Ismail	Sciara	ismail.sciera@sqltutorial.org	515.124.4369	1997-09-30	6	7700.00	108	10
119	Karen	Colmenares	karen.colmenares@sqltutorial.org	515.127.4568	1999-08-10	13	2500.00	114	3

employees 15

Object Info Session

Action Output

Schema: company

Time	Action	Response	Duration / Fetch Time
16:43:22	SELECT * FROM employees WHERE last_name LIKE '%C%' LIMIT 0, 10...	4 row(s) returned	0.000076 sec / 0.000000

Query Completed

9. Display the employee ID, Full name and Salary of the employees containing 'an' as second and third letters in their first name respectively.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:

```
-- 9
SELECT employee_id, CONCAT(first_name, ' ', last_name) AS full_name, salary FROM employees
WHERE first_name LIKE '_an%';
```
- Result Grid:** Displays the results of the query:

employee_id	full_name	salary
108	Nancy Greenberg	12000.00
109	Daniel Faviet	9000.00
- Session Tab:** Shows the schema is set to **company**. The history table shows one query execution:

Action	Time	Response	Duration / Fetch Time
SELECT employee_id, CONCAT(first_name, ' ', last_name) AS full_name, salary FROM employees WHERE first_name LIKE '_an%'	16:45:49	2 row(s) returned	0.00057 sec / 0.0000...

10. Display the max salary, department\_id for each department\_id and sort the result by max salary?

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL query:

```
-- 10
SELECT department_id, MAX(salary) AS max_salary FROM employees GROUP BY department_id
ORDER BY max_salary DESC;
```
- Result Grid:** Displays the results of the query in a tabular format. The columns are "department\_id" and "max\_salary". The data is as follows:

department_id	max_salary
9	24000.00
8	14000.00
2	13000.00
10	12000.00
11	12000.00
3	11000.00
7	10000.00
6	9000.00
5	8200.00
4	6500.00
1	4400.00
- Session Tab:** Shows the schema is set to "company". The history table contains one entry: "1 16:46:55 SELECT department\_id, MAX(salary) AS max\_salary FROM employees... 11 row(s) returned".
- Help Panel:** A note states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

11. Display the average salary for department\_id's if the avg salary is greater than 6000 and sort by avg salary descending.

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current schema is set to **company**.
- Query Editor:** Contains the following SQL code:

```
-- 11
SELECT department_id, AVG(salary) AS avg_salary FROM employees GROUP BY department_id
HAVING AVG(salary) > 6000 ORDER BY avg_salary DESC;
```
- Result Grid:** Displays the results of the query:

department_id	avg_salary
9	19333.33333
11	10150.00000
7	10000.00000
8	9616.66667
2	9500.00000
10	8600.00000
4	6500.00000
- Session Tab:** Shows the query was run at 16:50:18 and completed in 0.00082 sec / 0.0000... rows returned.
- Help Message:** A tooltip on the right side of the interface states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

## 12. Create an index for employees table on job\_id

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree, with 'company' selected. The main area contains a SQL editor window with the following query:

```
1 -- 12
2 • CREATE INDEX idx_job_id ON employees(job_id);
3 |
4 |
5 |
```

A context help message is visible on the right side of the interface:

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Below the SQL editor, the 'Session' tab is active. It shows the execution of the query:

Action Output	Time	Action	Response	Duration / Fetch Time
1	16:51:14	CREATE INDEX idx_job_id ON employees(job_id)	0 row(s) affected Records: 0 Duplic... 0.029 sec	

At the bottom, a message indicates 'Query Completed'.

### 13. Create a view for question 11

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Standard MySQL Workbench toolbar with icons for file operations, schema navigation, and query execution.
- Schemas:** Left sidebar showing the current schema is "company". Other schemas listed include "employee\_department", "Views", "Stored Procedures", "Functions", "sakila", "school", and "sys".
- Query Editor:** Contains the following SQL code:

```
1 -- 13
2 • CREATE VIEW avg_salary_view AS SELECT department_id, AVG(salary) AS avg_salary FROM employees
3 GROUP BY department_id HAVING AVG(salary) > 6000 ORDER BY avg_salary DESC;
4
5 • SELECT * FROM avg_salary_view;
6
7
8
```
- Result Grid:** Shows the results of the query:

department_id	avg_salary
9	19332.333333
11	10150.000000
7	10000.000000
8	9616.666667
2	9500.000000
10	8600.000000
4	6500.000000
- Session Tab:** Shows the schema is set to "company". The session history table shows two actions:

Action	Output	Time	Action	Response	Duration / Fetch Time
1	CREATE VIEW avg_salary_view AS SELECT department_id, AVG(salary) AS avg_salary FROM employees GROUP BY department_id HAVING AVG(salary) > 6000 ORDER BY avg_salary DESC;	16:52:33		0 row(s) affected	0.0036 sec
2	SELECT * FROM avg_salary_view LIMIT 0, 1000	16:52:33		7 row(s) returned	0.00085 sec / 0.000...
- Status Bar:** Shows "Query Completed".
- Help Message:** A tooltip on the right side of the interface states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

#### 14. Delete records in the employee table if the salary is less than 5000

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Administration, Schemas, Query 1, SQL File 3\*, Context... Snippets.
- Schemas:** Local instance 3306, company (selected), sakila, school, sys.
- Query Editor:** Contains the following SQL code:

```
1 -- 14
2
3 • SET SQL_SAFE_UPDATES = 0;
4
5 • DELETE FROM employees WHERE salary < 5000;
6
7 • SET SQL_SAFE_UPDATES = 1;
```
- Context Help:** Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
- Action Output:** Shows the execution log:

Action	Time	Response	Duration / Fetch Time
1	16:56:24	SET SQL_SAFE_UPDATES = 0	0.00043 sec
2	16:56:24	DELETE FROM employees WHERE salary < 5000	0.0016 sec
3	16:56:24	SET SQL_SAFE_UPDATES = 1	0.00011 sec
- Status:** Query Completed.

15. What is the difference between Delete and Truncate? Can we use truncate to delete a single row in the table?

#### DELETE:

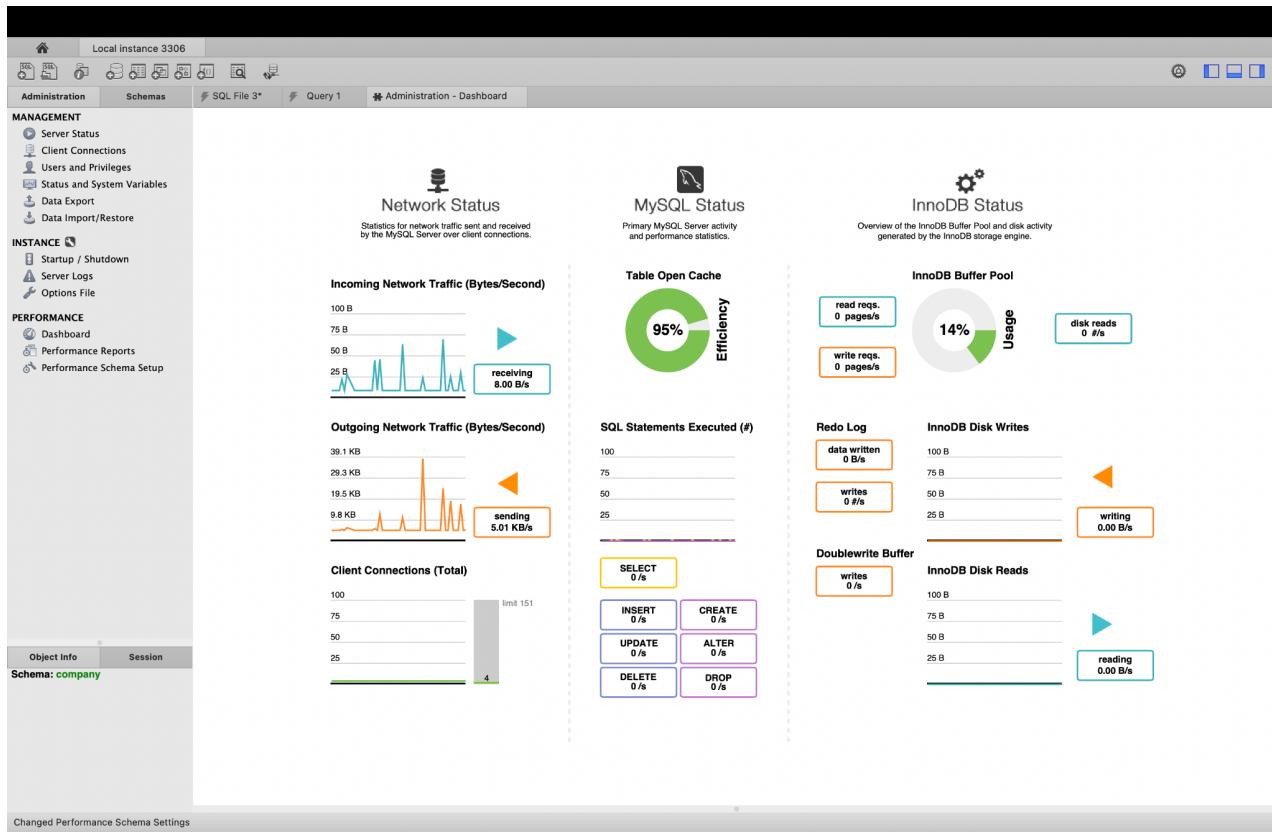
- DELETE is a Data Manipulation Language (DML)
- It is used to delete a specific row or rows of data from a table based on a condition.
- It can be used to delete a single row, multiple rows or a subset of rows from a given condition
- DELETE is slower compared to TRUNCATE as it logs each deleted row
- As the deleted rows are logged, we can roll back the changes if needed

#### TRUNCATE:

- TRUNCATE is a Data Definition Language (DDL)
- It removes all the rows from a table without considering any condition.
- It cannot be used to delete a specific row.
- It deallocates the data pages used by the table and resets the table to an empty state.
- TRUNCATE is faster compared to DELETE as it does not generate the same level of logs as DELETE.
- It is generally used to quickly remove all data from a table and reset it to the initial state.

No, We cannot use TRUNCATE to delete a single row in a table. It is designed to work on the entire table.

## 16. Measure the performance from the SQL workbench on different DDL and DML queries performed in this homework so far



Local instance 3306

Administration Schemas SQL File 3\* Query 1 Administration - Performance Reports

**MANAGEMENT**

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

**INSTANCE**

- Startup / Shutdown
- Server Logs
- Options File

**PERFORMANCE**

- Dashboard
- Performance Reports
- Performance Schema Setup

Local instance 3306 Performance Reports

**Report**

**Statement Statistics**  
Shows statement execution statistics for each user

User	Statement	Total Events...	Total Time (...)	Max Time (us)	Lock Time (...)	Rows Sent (#)	Rows Examined	Rows Affected	Full Scans (#)
root	show_status	395	799981.00	10518.00	1514.00	187465	187465	0	394
root	select	293	421411.00	50604.00	969.00	4796	11267	0	171
root	show_fields	403	225544.00	6824.00	536.00	3640	15923	0	0
root	show_variables	60	172768.00	19123.00	56.00	3418	3418	0	60
root	show_keys	49	92751.00	11135.00	420.00	80	986	0	0
root	create_table	5	66527.00	24954.00	103.00	0	0	0	0
root	show_tables	14	40786.00	6985.00	92.00	603	1354	0	0
root	create_db	2	38833.00	30413.00	56.00	0	0	2	0
root	create_index	5	33773.00	28516.00	49.00	0	0	0	0
root	set_option	134	33258.00	13130.00	0.00	0	0	0	0
root	...create_table	46	30903.00	7925.00	91.00	0	0	0	0
root	showCharsets	32	30772.00	7545.00	190.00	192	520	0	4
root	Ping	416	26984.00	528.00	0.00	0	0	0	0
root	...cedure_Status	13	22767.00	5450.00	64.00	93	584	0	0
root	insert	59	21989.00	2927.00	62.00	0	0	66	0
root	...ow_databases	9	18391.00	5023.00	45.00	58	241	0	9
root	update	3	18136.00	15918.00	60.00	0	1304	528	0
root	show_collations	4	14518.00	5302.00	7.00	1144	2452	0	4
root	drop_table	2	12612.00	11145.00	64.00	0	0	0	0
root	...nction_Status	13	12136.00	2710.00	28.00	81	674	0	0
root	...orage_engines	4	5497.00	2212.00	2.00	44	44	0	4
root	create_view	1	3481.00	3481.00	19.00	0	0	0	0
root	delete	4	3389.00	1520.00	24.00	0	40	12	0
root	change_db	23	3140.00	384.00	5.00	0	0	0	0
root	freturn	1691	2883.00	23.00	0.00	0	0	0	0
root	stmt	1	2832.00	2832.00	0.00	0	1	0	0
root	set	2	2721.00	2715.00	0.00	0	1	0	0

Statement Statistics Export... Copy Selected Copy Query Refresh

Changed Performance Schema Settings