

DATABASE MANAGEMENT SYSTEM

CS23332

EXERCISE 2

MANIPULATING DATA

ISHWARIYA R
241801097

1. Create MY_EMPLOYEE table with the following structure

The screenshot shows a SQL command-line interface with the following details:

- Language:** SQL
- Rows:** 10
- Toolbar:** Includes icons for Undo, Redo, Search, Insert, and Copy.
- SQL Query:**

```
1 CREATE TABLE MY_EMPLOYEE(
2     ID NUMBER(4) NOT NULL,
3     Last_name VARCHAR(25),
4     First_name VARCHAR(25),
5     Userid Varchar(25),
6     Salary Number(9,2)
7 );
```
- Result Tab:** The "Results" tab is selected, displaying the message "Table created."
- Performance:** The execution time is shown as 0.04 seconds.

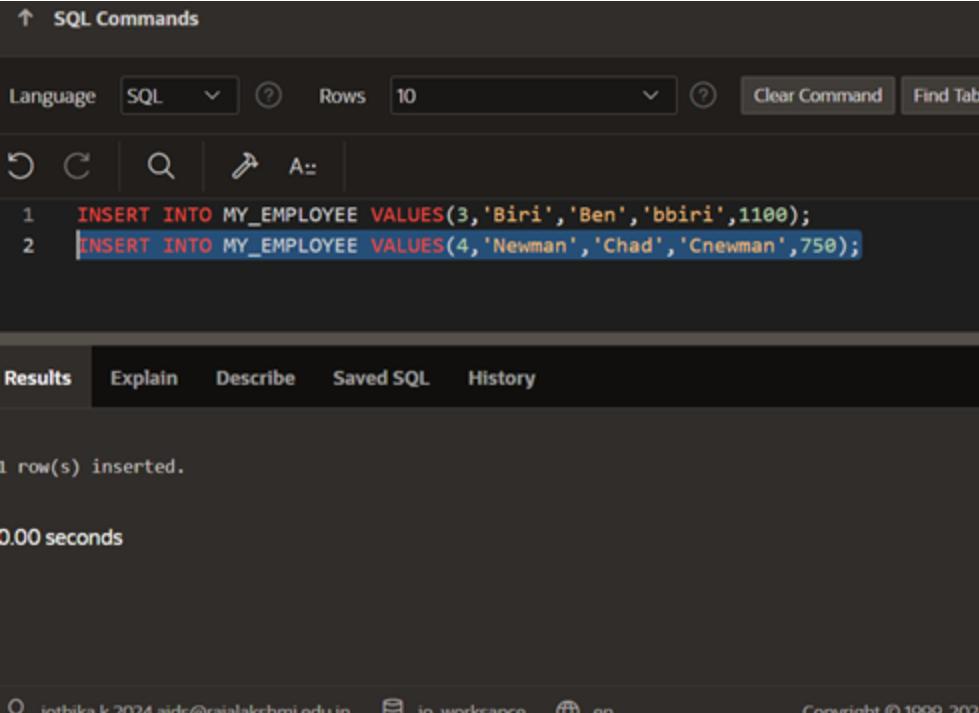
2. Add the first and second rows data to MY_EMPLOYEE table from the following sampledata.
3. Display the table with values.

The screenshot shows a SQL command interface with the following details:

- SQL Commands** tab is selected.
- Language**: SQL
- Rows**: 10
- Schema**: WKSP_JOWORKSAPCE
- Buttons**: Save, Run, Clear Command, Find Tables.
- Toolbar Icons**: Undo, Redo, Search, Filter, Help, Settings.
- SQL Statements** (Copy):
 - 1 `INSERT INTO MY_EMPLOYEE (ID, Last_name, First_name, Userid, Salary) VALUES (1, 'Patel', 'Ralph', 'rpatel', 895);`
 - 2 `INSERT INTO MY_EMPLOYEE (ID, Last_name, First_name, Userid, Salary) VALUES (2, 'Dancs', 'Betty', 'bdancs', 860);`
 - 3 `SELECT * FROM MY_EMPLOYEE;`
- Results** tab is selected.
- Table Headers**: ID, LAST_NAME, FIRST_NAME, USERID, SALARY.
- Table Data**:

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
2	Dancs	Betty	bdancs	860
1	Patel	Ralph	rpatel	895
- Message**: 2 rows returned in 0.02 seconds
- Links**: Download

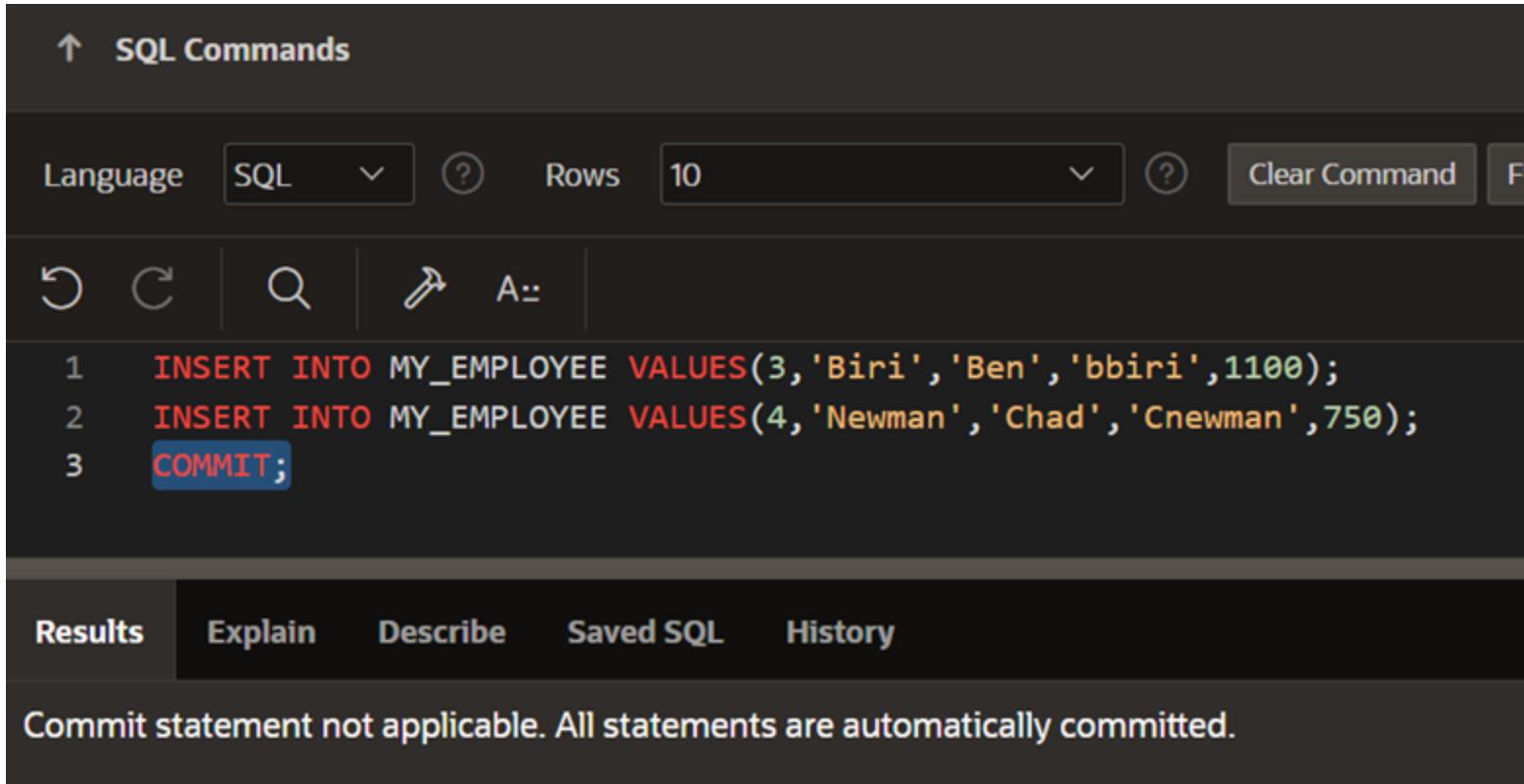
4. Populate the next two rows of data from the sample data. Concatenate the first letter of the `first_name` with the first seven characters of the `last_name` to produce `Userid`.



The screenshot shows a SQL command interface with the following details:

- SQL Commands:** The interface title.
- Language:** SQL selected.
- Rows:** Set to 10.
- Commands:** Two `INSERT` statements are present:
 - `1 INSERT INTO MY_EMPLOYEE VALUES(3,'Biri','Ben','bbiri',1100);`
 - `2 INSERT INTO MY_EMPLOYEE VALUES(4,'Newman','Chad','Cnewman',750);`
- Results:** The results section shows:
 - 1 row(s) inserted.
 - 0.00 seconds
- Footer:** User information (jothika.k.2024.aids@rajalakshmi.edu.in), workspace (jo_workspace), and language (en).
- Copyright:** Copyright © 1999, 2024.

5. Make the data additions permanent.



The screenshot shows a SQL command interface with the following details:

- SQL Commands** header with an upward arrow icon.
- Language**: SQL dropdown set to SQL.
- Rows**: Set to 10.
- Clear Command** button.
- Toolbar icons**: Undo (U), Redo (R), Search (Q), Insert (I), and Auto (A).
- SQL Statements**:
 - 1 `INSERT INTO MY_EMPLOYEE VALUES(3, 'Biri', 'Ben', 'bbiri', 1100);`
 - 2 `INSERT INTO MY_EMPLOYEE VALUES(4, 'Newman', 'Chad', 'Cnewman', 750);`
 - 3 `COMMIT;` (highlighted in blue)
- Bottom Navigation**: Results, Explain, Describe, Saved SQL, History tabs. The Results tab is selected.
- Message**: "Commit statement not applicable. All statements are automatically committed."

6. Change the last name of employee 3 to Drexler.

The screenshot shows a SQL command interface with the following details:

- SQL Commands** section at the top.
- Language**: SQL selected.
- Rows**: 10 selected.
- Toolbar icons**: Refresh, Undo, Redo, Search, Insert, and Sort.
- SQL Query**:

```
1 UPDATE MY_EMPLOYEE SET Last_name = 'Drexler' WHERE ID=3;
```
- Results Tab** is active.
- Output**:

```
1 row(s) updated.
```
- Performance**:

```
0.01 seconds
```

7. Change the salary to 1000 for all the employees with a salary less than 900.

The screenshot shows a SQL query editor interface. At the top, there are dropdown menus for 'Language' set to 'SQL' and 'Rows' set to '10'. Below the header are several icons: a refresh circle, a circular arrow, a magnifying glass for search, a wrench for edit, and a double arrow for refresh. The main area contains a single line of SQL code:

```
1 UPDATE MY_EMPLOYEE SET Salary =1000 WHERE Salary < 900;
```

Below the code, there is a horizontal navigation bar with tabs: 'Results' (which is selected), 'Explain', 'Describe', 'Saved SQL', and 'History'. The results section displays the message '3 row(s) updated.' and a performance metric of '0.02 seconds'.

8. Delete Betty dancs from MY_EMPLOYEE table.

Language SQL Rows 10

```
1   DELETE FROM MY_EMPLOYEE WHERE First_name ='betty' and Last_name =' dancs';
```

Results Explain Describe Saved SQL History

0 row(s) deleted.

0.03 seconds

9. Empty the fourth row of the emp table.

Language SQL Rows 10

↻ ↺ | ⚡ A::

```
1 DELETE FROM MY_EMPLOYEE WHERE ID=4;
```

Results Explain Describe Saved SQL History

1 row(s) deleted.

0.00 seconds