

Assignment 7

The screenshot shows the MySQL Workbench interface. On the left, the Navigator pane displays the database schema with the following structure:

- SCHEMAS**: bank, employee, handler_lab, sakila, student, sys, world.
- employee**: Tables, Views, Stored Procedures, Functions.
- handler_lab**: Tables (departments, employees), Views, Stored Procedures, Functions.

In the center, the SQL Editor pane contains the following stored procedure code:

```
3    -- BEGIN
4    --     DECLARE done INT DEFAULT 0;
5    --     DECLARE e_name VARCHAR(100);
6    --     DECLARE cur CURSOR FOR
7    --         SELECT emp_name FROM employees WHERE dept_id = did;
8
9    --     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
10
11   --     OPEN cur;
12
13   --     read_loop: LOOP
14   --         FETCH cur INTO e_name;
15   --         IF done THEN
16   --             LEAVE read_loop;
17   --         END IF;
18
19    --     SELECT * FROM departments;
```

At the bottom, the Result Grid pane shows the output of the last query:

Employee
employee_1

On the right, there is a vertical toolbar with a "Result Grid" button highlighted.

The screenshot shows two instances of MySQL Workbench running side-by-side. Both instances have the same database connection, "Local instance MySQL80". The left instance has the "SQL" tab selected, while the right instance has the "Navigator" tab selected. Both panes display the same SQL code for creating a safe add employee procedure and calling it with an invalid department ID.

```
-- DELIMITER $$  
-- CREATE PROCEDURE safe_add_employee(IN ename VARCHAR(100), IN sal DECIMAL(10,2), IN did INT)  
-- BEGIN  
--     DECLARE dept_exists INT;  
--  
--     SELECT COUNT(*) INTO dept_exists FROM departments WHERE dept_id = did;  
--  
--     IF dept_exists = 0 THEN  
--         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid Department ID';  
--     ELSE  
--         INSERT INTO employees (emp_name, salary, dept_id)  
--             VALUES (ename, sal, did);  
--         SELECT CONCAT('Employee ', ename, ' added safely.') AS Message;  
--     END IF;  
-- END$$  
-- DELIMITER ;  
  
CALL safe_add_employee('Grace', 60000, 9);
```

Administration | Schemas

Information

Table: departments

Columns:

dept_id	int AI PK
dept_name	varchar(50)

Action Output

#	Time	Action	Message
1	20 23:41:56	CREATE PROCEDURE delete_employee(IN eid INT) BEGIN	DECLARE rows_affected INT; DELETE F...
2	21 23:42:13	CALL delete_employee(2)	1 row(s) affected
3	22 23:43:17	CREATE PROCEDURE safe_add_employee(IN ename VARCHAR(100), IN sal DECIMAL(10,2), IN did INT) B...	0 row(s) affected
4	23 23:43:28	CALL safe_add_employee('Grace', 60000, 99)	Error Code: 1644. Invalid Department ID
5	24 23:44:12	CALL safe_add_employee('Grace', 60000, 10)	Error Code: 1644. Invalid Department ID
6	25 23:44:25	CALL safe_add_employee('Grace', 60000, 9)	Error Code: 1644. Invalid Department ID

Context Help | Snippets

SQLAdditions

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

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

Assignment 7

The screenshot shows the MySQL Workbench interface. On the left, the Navigator pane displays the database schema with the following structure:

- Schemas:
 - bank
 - employee
 - handler_lab
 - Tables
 - departments
 - employees
 - Views
 - Stored Procedures
 - Functions
 - sakila
 - student
 - sys
 - world

The central pane contains the SQL code for a stored procedure:

```
1  -- DELIMITER $$  
2  -- CREATE PROCEDURE delete_employee(IN eid INT)  
3  -- BEGIN  
4  --     DECLARE rows_affected INT;  
5  
6  --     DELETE FROM employees WHERE emp_id = eid;  
7  --     SET rows_affected = ROW_COUNT();  
8  
9  --     IF rows_affected = 0 THEN  
10 --         SELECT 'No such employee' AS Message;  
11 --     ELSE  
12 --         SELECT CONCAT('Employee ID ', eid, ' deleted successfully') AS Message;  
13 --     END IF;  
14 -- END$$  
15 -- DELIMITER ;
```

The bottom pane shows the result grid with one row:

Message
Employee ID 2 deleted successfully

A blue "Result Grid" button is located on the right side of the result grid panel.

Assignment 7

Navigator: Schemas

SCHEMAS

- bank
- employee
 - Tables
 - Views
 - Stored Procedures
 - Functions
- handler_lab
 - Tables
 - departments
 - employees
 - Views
 - Stored Procedures
 - Functions
- sakila
- student
- sys
- world

Administration Schemas

Information:

Navigator: Schemas

SCHEMAS

- bank
- employee
 - Tables
 - Views
 - Stored Procedures
 - Functions
- handler_lab
 - Tables
 - departments
 - employees
 - Views
 - Stored Procedures
 - Functions
- sakila
- student
- sys
- world

Administration Schemas

Information:

Navigator: Schemas

SCHEMAS

- bank
- employee
 - Tables
 - Views
 - Stored Procedures
 - Functions
- handler_lab
 - Tables
 - departments
 - employees
 - Views
 - Stored Procedures
 - Functions
- sakila
- student
- sys
- world

Administration Schemas

Information:

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content: |

Message

Employees with department:11

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content: |

annual_salary(5000)

60000.00

Result 5 x

```
1 -- DELIMITER $$  
2 -- CREATE PROCEDURE count_employees()  
3 -- BEGIN  
4 --     DECLARE done INT DEFAULT 0;  
5 --     DECLARE emp_count INT DEFAULT 0;  
6 --     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;  
7  
8 --     SELECT COUNT(*) INTO emp_count FROM employees WHERE dept_id IS NOT NULL;  
9 --     SELECT CONCAT('Employees with department:', emp_count) AS Message;  
10 -- END$$  
11 -- DELIMITER ;  
12  
13 • CALL count_employees();
```

```
1 -- DELIMITER $$  
2 -- CREATE FUNCTION annual_salary(monthly DECIMAL(10,2))  
3 -- RETURNS DECIMAL(10,2)  
4 -- DETERMINISTIC  
5 -- BEGIN  
6 --     RETURN monthly * 12;  
7 -- END$$  
8 -- DELIMITER ;  
9  
10 • SELECT annual_salary(5000);  
11  
12
```

Assignment 7

Navigator Schemas

```
1 -- DELIMITER $$  
2 -- CREATE PROCEDURE check_salary(IN sal DECIMAL(10,2))  
3 -- BEGIN  
4 --     DECLARE EXIT HANDLER FOR SQLEXCEPTION  
5 --     BEGIN  
6 --         SELECT 'Salary check failed: Below minimum allowed' AS Message;  
7 --     END;  
8  
9 --     IF sal < 1000 THEN  
10 --         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Salary too low';  
11 --     ELSE  
12 --         SELECT CONCAT('Salary is valid: ', sal) AS Message;  
13 --     END IF;  
14 -- END$$  
15 -- DELIMITER ;
```

Result Grid

Message
Salary check failed: Below minimum allowed

Administration Schemas

Information Navigator Schemas

```
5 -- DECLARE CONTINUE HANDLER FOR 1062  
6 -- BEGIN  
7 --     SELECT CONCAT('Duplicate found for employee_', i, ', skipped.') AS Info;  
8 -- END;  
9  
10 -- WHILE i <= 5 DO  
11 --     INSERT INTO employees (emp_name, salary, dept_id)  
12 --     VALUES (CONCAT('employee_', i), 40000 + (i * 1000), (i % 5) + 1);  
13 --     SET i = i + 1;  
14 -- END WHILE;  
15  
16 --     SELECT 'Bulk insert completed' AS Status;  
17 -- END$$  
18 -- DELIMITER ;
```

Result Grid

Status
Bulk insert completed

Administration Schemas

Assignment 7

Schemas

```
3 -- BEGIN
4 -- DECLARE EXIT HANDLER FOR 1062
5 -- BEGIN
6 --     SELECT 'Employee already exists' AS Message;
7 -- END;
8
9 -- INSERT INTO employees (emp_name, salary, dept_id)
10 -- VALUES (ename, sal, did);
11
12 -- SELECT CONCAT('Employee ', ename, ' added successfully') AS Message;
13 -- END$$
14 -- DELIMITER ;
15
16 • CALL add_employee('John', 55000, 2);
17
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid

Message
Employee John added successfully

Administration Schemas

Information

Schemas

```
1 -- DELIMITER $$
2 -- CREATE PROCEDURE add_department(IN dname VARCHAR(50))
3 -- BEGIN
4 --     DECLARE EXIT HANDLER FOR 1062
5 --     BEGIN
6 --         SELECT 'Department already exists' AS Message;
7 --     END;
8
9 --     INSERT INTO departments (dept_name) VALUES (dname);
10 --     SELECT CONCAT('Department ', dname, ' added successfully') AS Message;
11 -- END$$
12 -- DELIMITER ;
13
14 • call add_department(12);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid

Message
Department 12 added successfully

Administration Schemas

Information

Result 1