| Experiment No.4 |
| --- |
| Integrity Constraints for specified system |
| Date of Performance: 20/02/2024 |
| Date of Submission: 27/02/2024 |

**Experiment No. 4:** Apply Integrity Constraints for the specified system.

**Course Outcome [CSL503.2]:** Create database tables with different DDL and DML statements and apply integrity constraints

**Aim**:- Apply Integrity Constraints for the specified system.

**Theory**:

1. PRIMARY KEY CONSTRAINTS
2. FOREIGN KEY CONSTRAINTS
3. UNIQUE KEY CONSTRAINTS
4. NULL CONSTRAINTS
5. NOT NULL CONSTRAINTS
6. CHECK CONSTRAINTS
7. DEFAULT CONSTRAINTS

1. PRIMARY KEY CONSTRAINTS: Primary key constraints uniquely identify each record in a table, ensuring that no two records have the same key value. They enforce data integrity by preventing duplicate or null values in the primary key column.

2. FOREIGN KEY CONSTRAINTS: Foreign key constraints establish a relationship between tables by linking a column in one table to a primary key column in another table. They enforce referential integrity, ensuring that values in the foreign key column match values in the primary key column.

3. UNIQUE KEY CONSTRAINTS: Unique key constraints ensure that values in a specified column or combination of columns are unique across the table. They prevent duplication of data in the specified columns, similar to primary key constraints, but they allow null values.

4. NULL CONSTRAINTS: Null constraints specify whether a column can contain null values or not. Columns with a null constraint cannot store null values, ensuring that they always have a valid data entry.

5. NOT NULL CONSTRAINTS: Not null constraints specify that a column must have a value, prohibiting the insertion of null values into that column. They enforce data integrity by mandating the presence of data in the specified column.

6. CHECK CONSTRAINTS: Check constraints define conditions that data entered into a column must meet. They ensure that data conforms to predefined rules, allowing only values that satisfy the specified conditions to be inserted or updated.

7. DEFAULT CONSTRAINTS: Default constraints specify a default value for a column when no explicit value is provided during insertion. They automatically insert the default value into the column if no other value is specified, ensuring consistency and reducing the need for manual data entry.

**Code:**

-- Create table with constraints

CREATE TABLE employee(

emp\_id INT(20) PRIMARY KEY,

name CHAR(50),

designation CHAR(50),

CONSTRAINT check\_emp\_id\_gt\_1 CHECK (emp\_id > 1)

);

-- Insert the data again

INSERT INTO employee(emp\_id, name, designation) VALUES

(1, 'Anjali, 'Manager'),

(2, 'Krisha ', 'Assistant Manager'),

(3, 'Pranali ', 'Client');

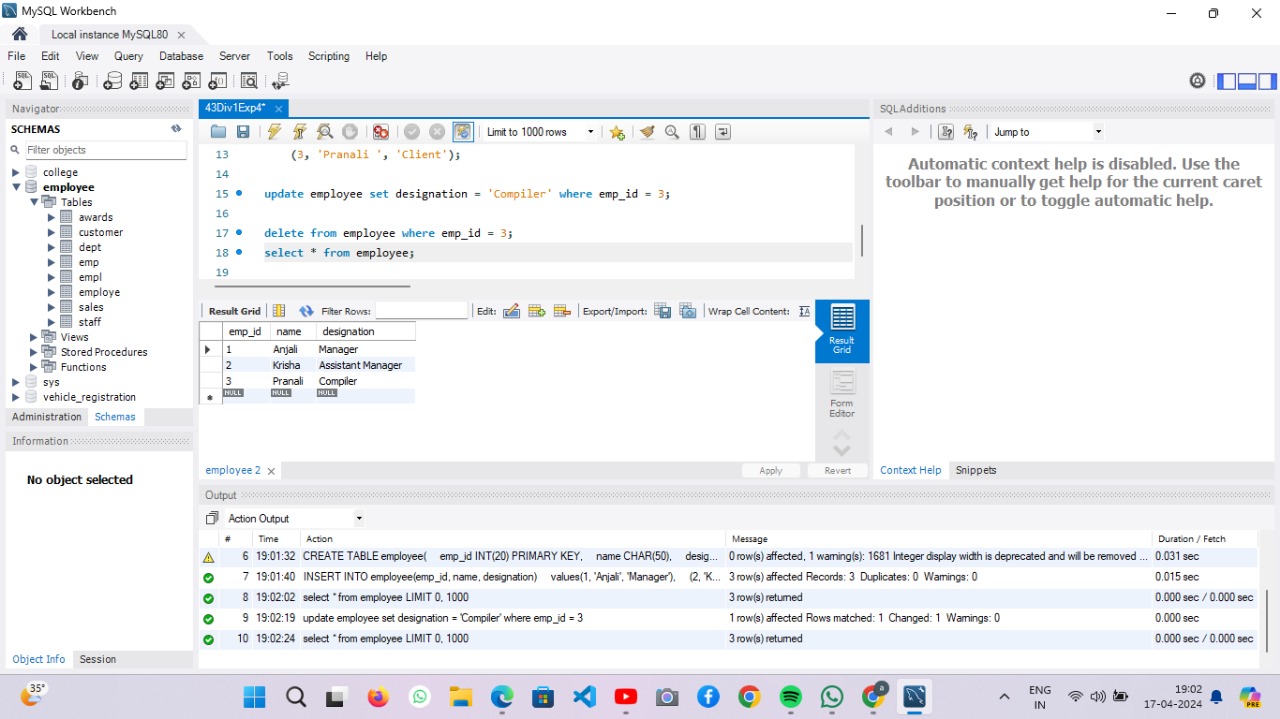
update employee set designation = 'Compiler' where emp\_id = 3;

delete from employee where emp\_id = 3;

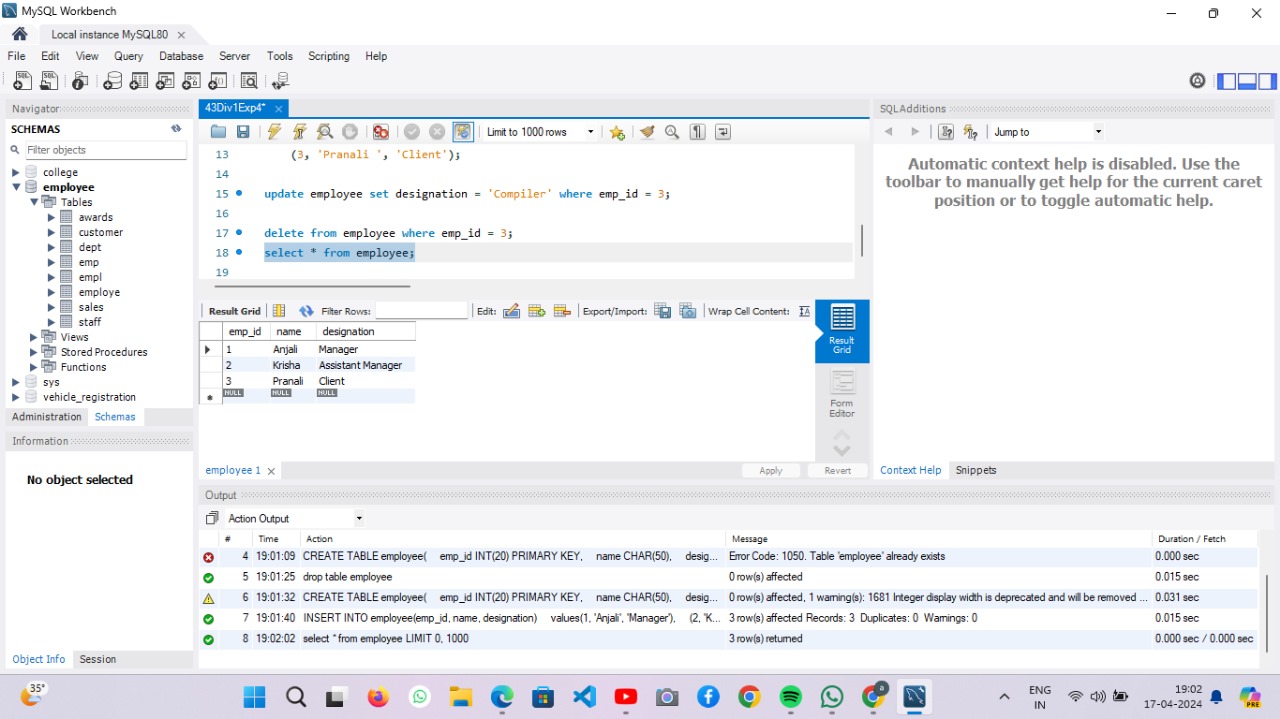
select \* from employee;

**Output:**

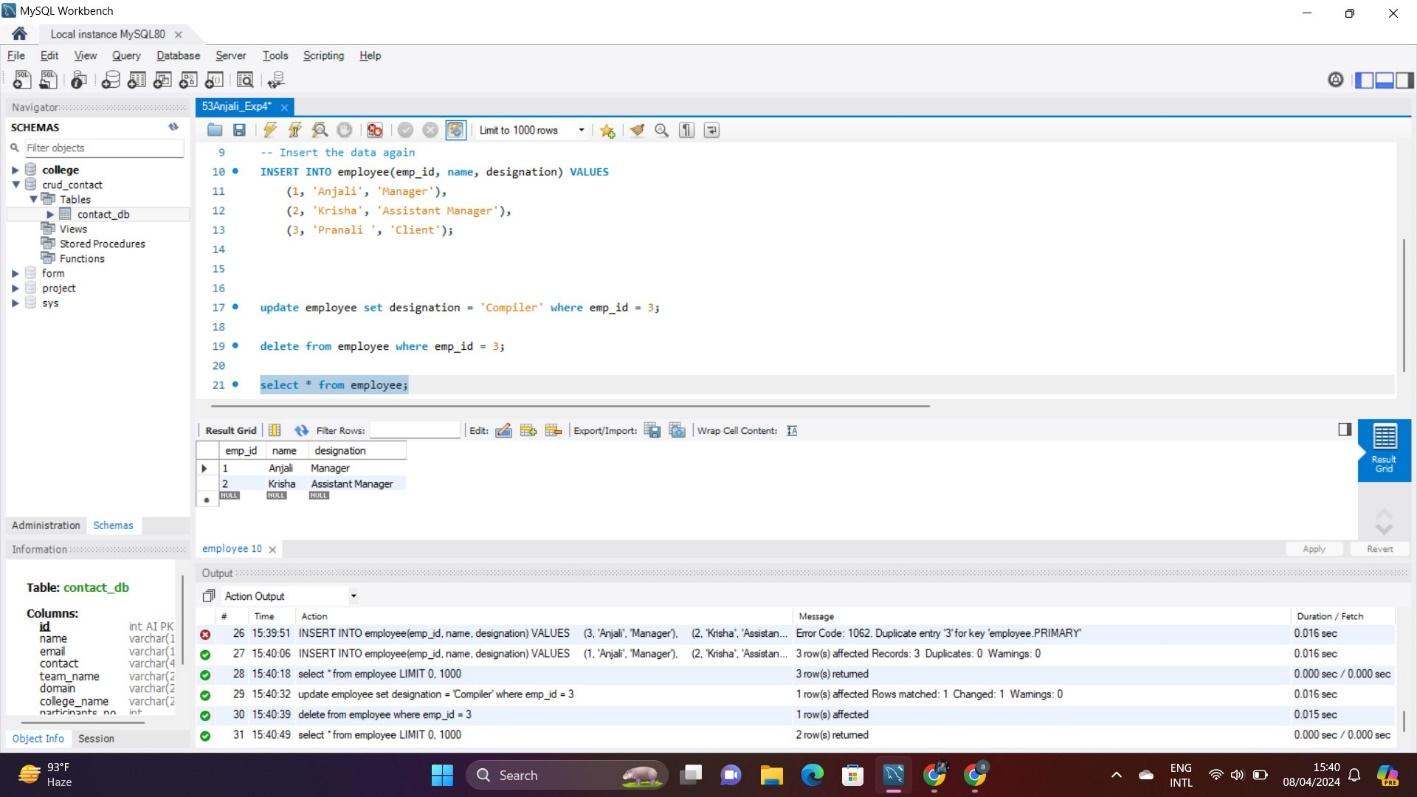
Creating the table:

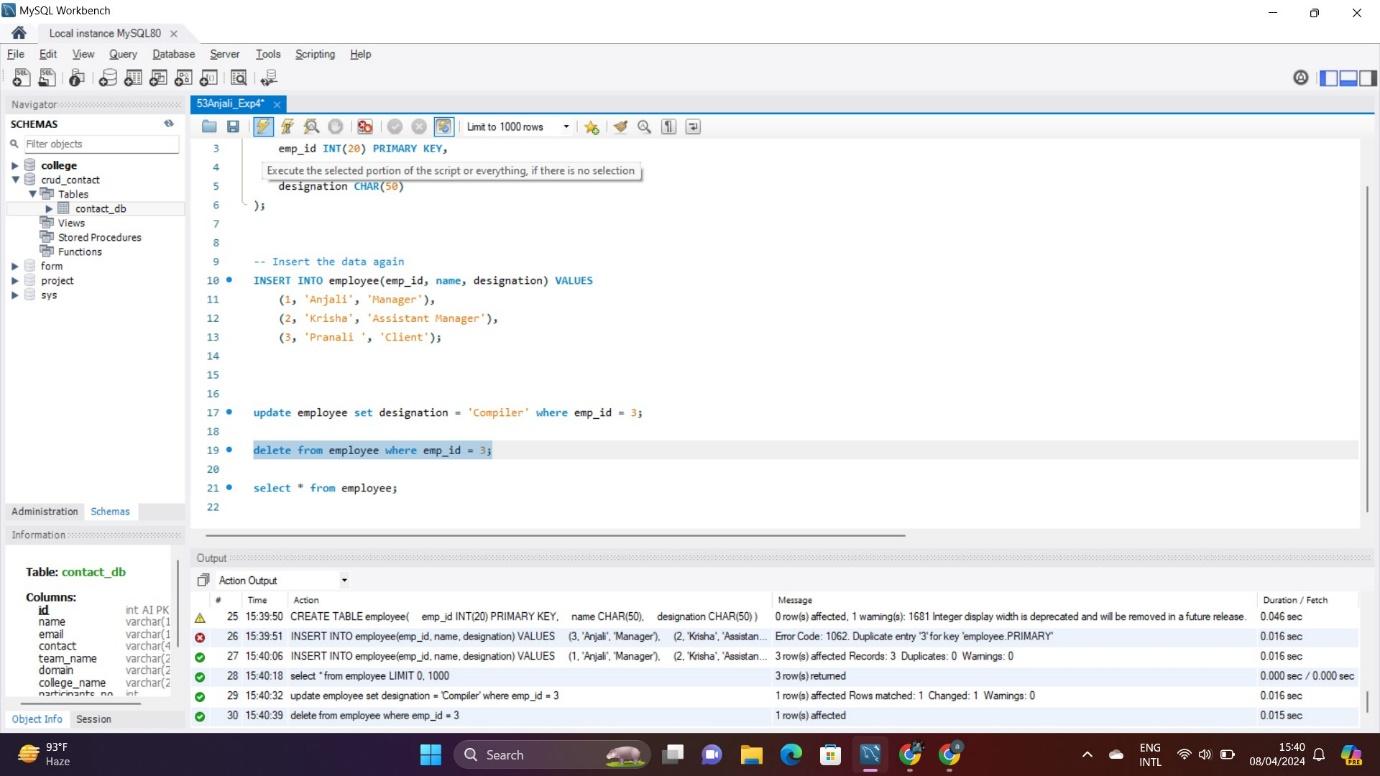


Inserting the values in Table:



Update and Delete :





**CONCLUSION:**

In this practical experiment, various integrity constraints, including primary key, foreign key, unique key, null, not null, check, and default constraints, were applied to a specified system. These constraints ensure data accuracy, consistency, and reliability within the database by enforcing rules on the data stored in tables. By practicing the implementation of these constraints, students gain a deeper understanding of database design principles and how to maintain data integrity. Overall, this experiment helps students develop essential skills in creating robust database schemas that adhere to organizational requirements and industry standards, thus preparing them for real-world database management scenarios.