| **SE-COMP Roll number : 9914** |
| --- |
| **Experiment no. : 4 (part3) Date of Implementation : 24/02/24** |
| **Aim : To implement Integrity constraints** |
| **Tool Used : PostgreSQL** |
| **Related Course outcome : At the end of the course, Students will be able to Use**  **SQL : Standard language of relational database** |
| **Rubrics for assessment of Experiment:**   | **Indicator** | **Poor** | **Average** | **Good** | | --- | --- | --- | --- | | **Timeliness**   * **Maintains assignment deadline (3)** | **Assignment not done (0)** | **One or More than One week late (1-2)** | **Maintains deadline (3)** | | **Completeness and neatness**   * **Complete all parts of QUERY assignment(3)** | **N/A** | **< 80% complete (1-2)** | **100% complete (3)** | | **Originality**   * **Extent of plagiarism(2)** | **Copied it from someone else(0)** | **At least few questions have been done without copying(1)** | **Assignment has been solved completely without copying (2)** | | **Knowledge**   * **In depth knowledge of the QUERY assignment(2)** | **Unable to answer 2 questions(0)** | **Unable to answer 1 question (1)** | **Able to answer 2 questions (2)** | |
| **Assessment Marks :**   | **Timeliness(3)** |  | | --- | --- | | **Completeness and neatness(3)** |  | | **Originality (2)** |  | | **Knowledge (2)** |  | | **Total (Out of 10)** |  | |
| **Remark:** |
| **Teacher's Sign :** |

**Experiment No. 4- Integrity Constraints**

AIM:

* To implement database for relational model using DDL statement
* Apply Integrity Constraints for the specified system

**Objective of the Experiment:**

After completing this experiment you will be able to:

* Create database.
* Create table with constraints
* Modify the schema of the table.

**Theory :**

**Pre Lab/ Prior Concepts:**

The Data Definition Language (DDL) is used to create and modify the relational schema. Also it is used to add various constraints to the table like the primary key, foreign key, check constraint, not null constraint and unique constraint.

The DDL statements are:

CREATE

DROP

ALTER

SQL supports the standard int, smallint, real, double precision, char(N), varchar(N), date, time, timestamp, and interval for creating tables.

**Procedure / Algorithm:**

**Create Database and use it:**

$ createdb mydb

$ psql mydb

**Delete a database:**

$ dropdb mydb

**Create table:**

CREATE TABLE my\_first\_table

( first\_column text,

second\_column integer

);

CREATE TABLE products (

product\_no integer,

name text, price numeric);

**Drop Table:**

DROP TABLE my\_first\_table;

DROP TABLE products;

**Default Value:**

CREATE TABLE products (

product\_no integer,

name text,

price numeric **DEFAULT 9.99** );

**Constraints:**

**1. Primary Key**

CREATE TABLE products (

product\_no integer **PRIMARY KEY**,

name text,

price numeric );

Primary keys can also constrain more than one column.

CREATE TABLE example (

1. integer,
2. integer,
3. integer,

**PRIMARY KEY (a, c)**

);

**2. Check Constraint**

CREATE TABLE products (

product\_no integer,

name text,

price numeric **CHECK (price** > **0)** );

**3. Not Null Constraint**

CREATE TABLE products (

product\_no integer **NOT NULL**,

name text **NOT NULL**,

price numeric );

**4. Unique Constraint**

CREATE TABLE products (

product\_no integer **UNIQUE**,

name text,

price numeric );

**5. Foreign Key Constarint**

CREATE TABLE products (

product\_no integer PRIMARY KEY,

name text,

price numeric );

CREATE TABLE orders (

order\_id integer PRIMARY KEY,

product\_no integer **REFERENCES products (product\_no)**,

quantity integer

);

Here a foreign key constraint in the order table references the products table.

**Modifying table:**

**Adding column**

ALTER TABLE products ADD COLUMN description text;

**Removing column**

ALTER TABLE products DROP COLUMN description;

**Adding Constraint**

ALTER TABLE products ADD CONSTRAINT some\_name UNIQUE (product\_no); ALTER TABLE products ADD FOREIGN KEY (product\_group\_id) REFERENCES product\_groups;

**Removing Constraint**

ALTER TABLE products DROP CONSTRAINT some\_name;

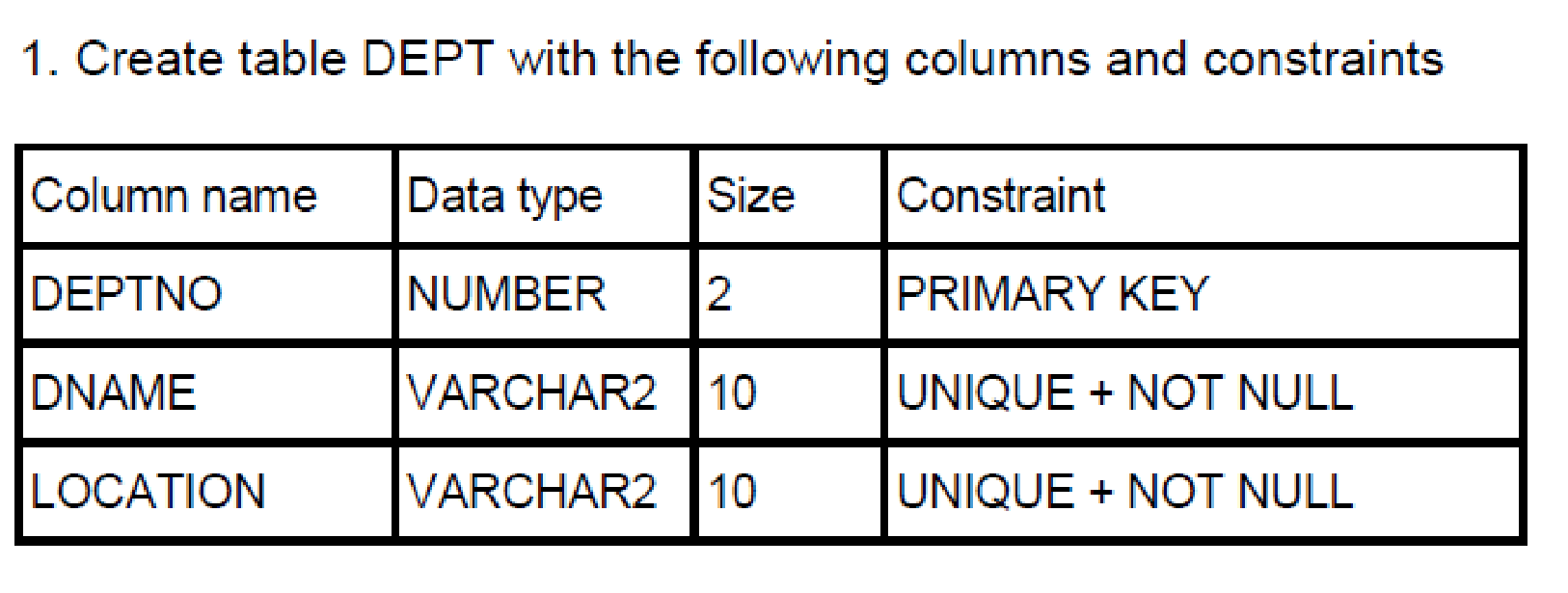
**Adding Not Null Constraint**

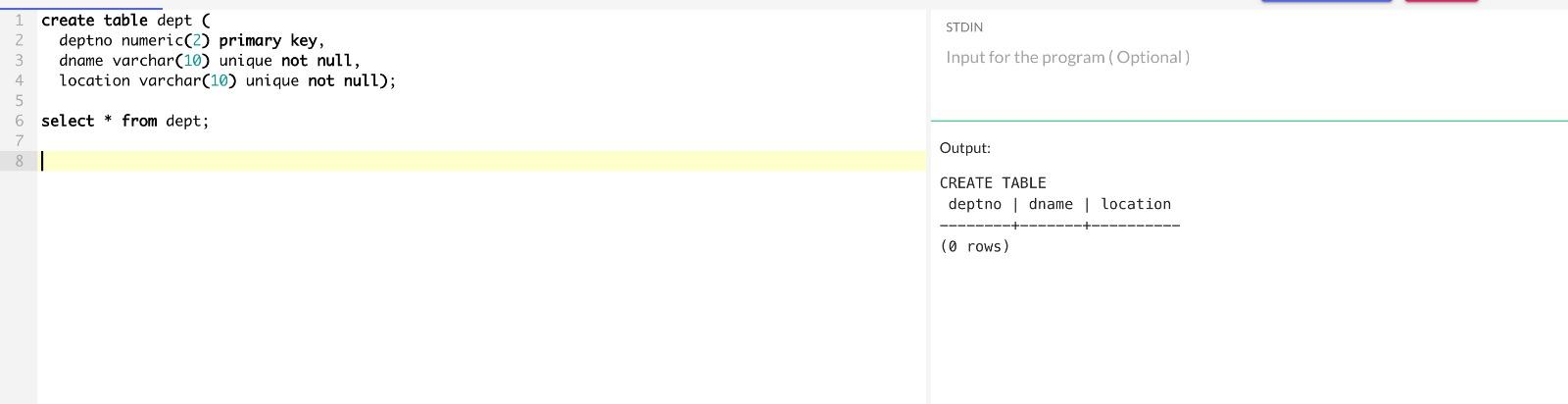
ALTER TABLE products ALTER COLUMN product\_no SET NOT NULL;

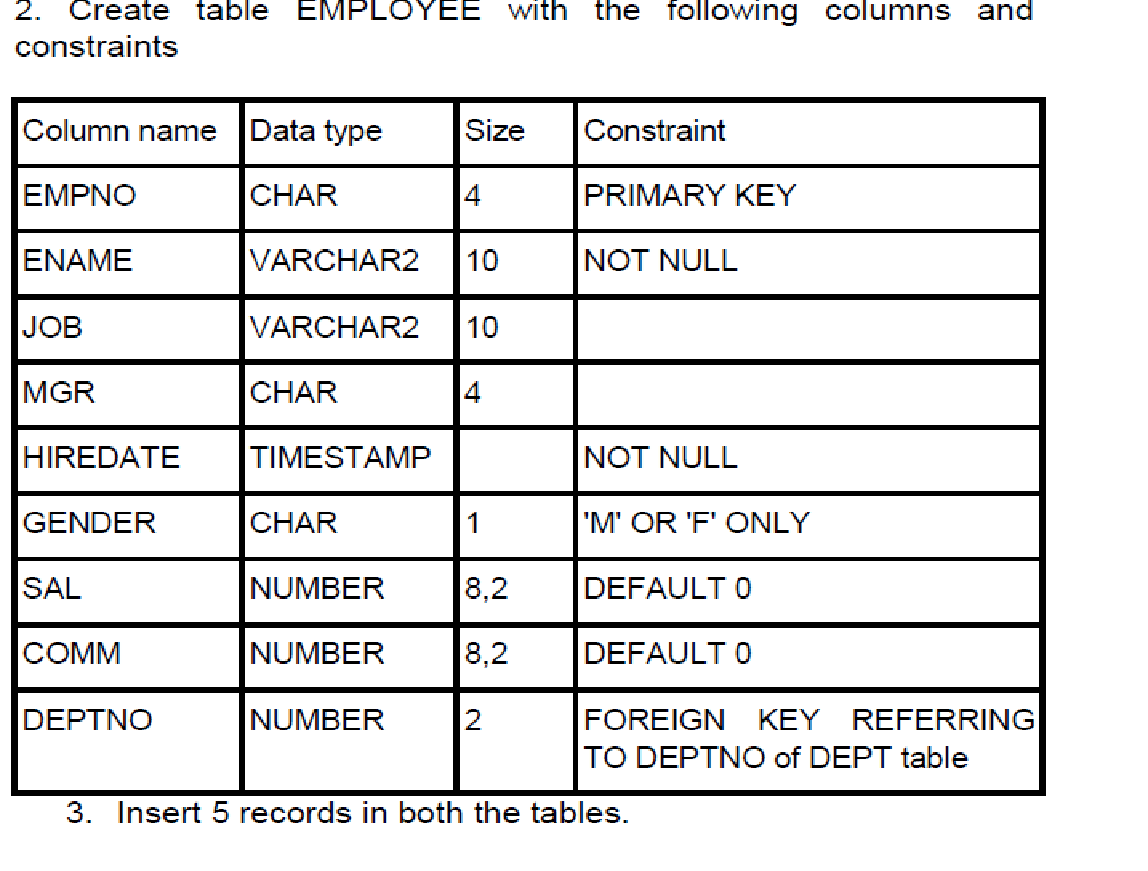
**Removing Not Null Constraint**

ALTER TABLE products ALTER COLUMN product\_no DROP NOT NULL;

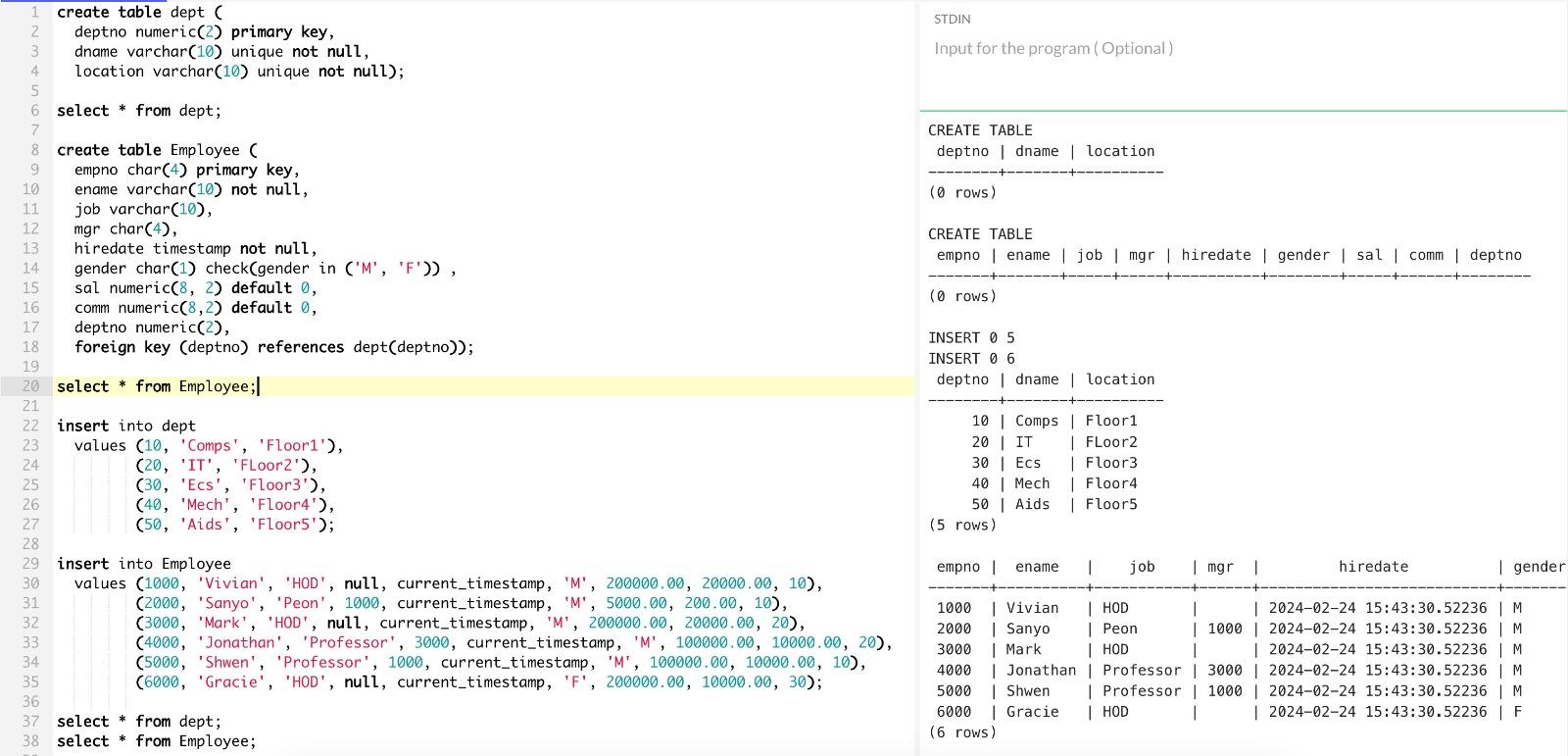
Task1: Exercise –

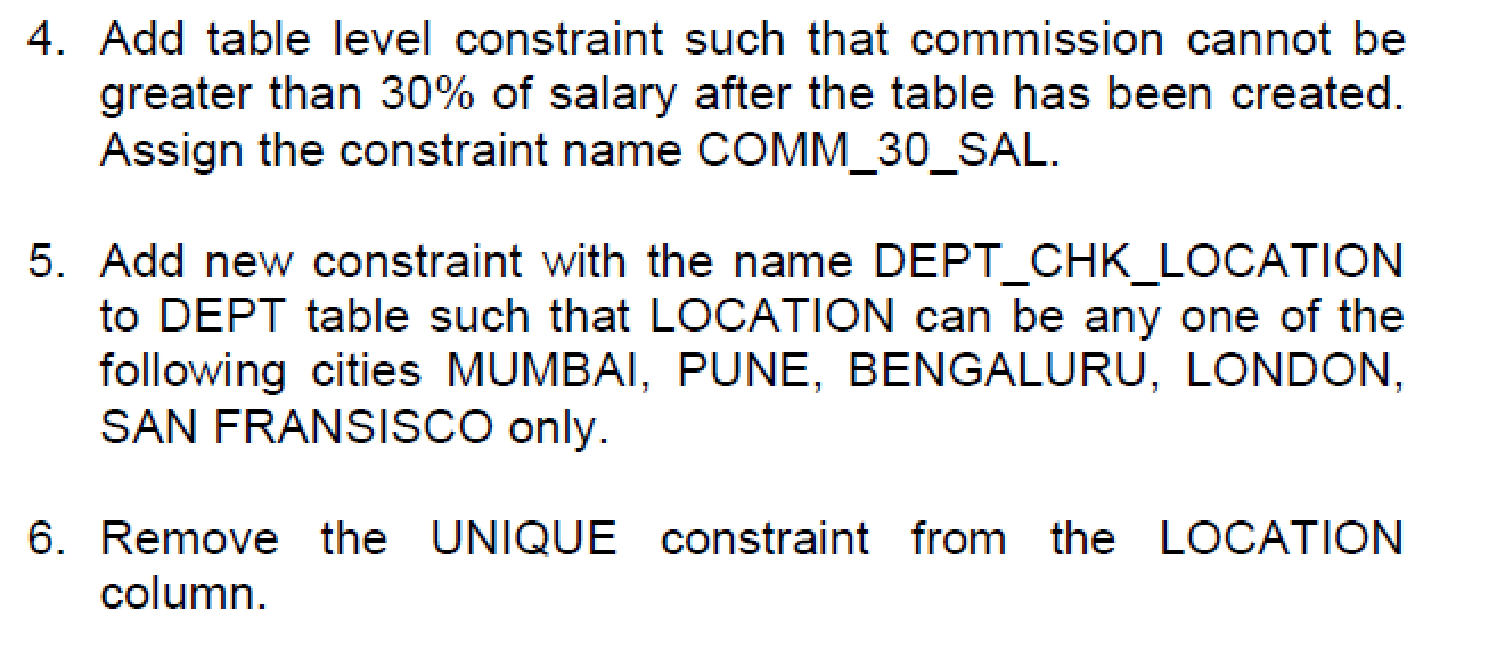


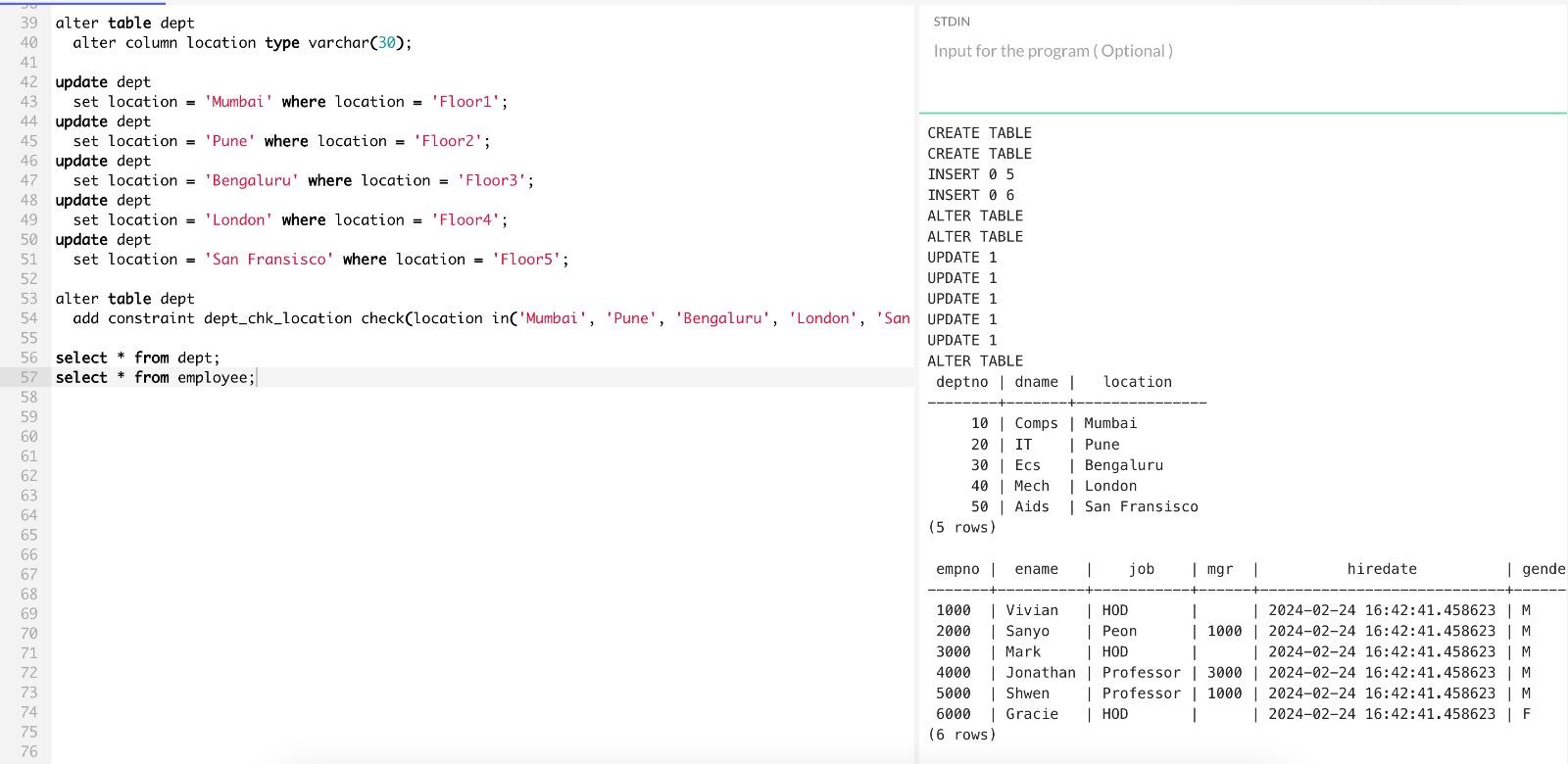














**Conclusion:**

Thus using the schema diagram from the previous experiment, the tables were created using CREATE DDL statement with the primary key and foreign key constraint. Other constraints like Check, Unique and Not Null were added to the appropriate column by using ALTER DDL statement.



**Post Lab Assignment:**



**1)What is NOT NULL constraint? What is DEFAULT constraint?**

**⇒**

* NOT NULL Constraint: A NOT NULL constraint prevents the insertion of NULL values into a specified column in a database table. It ensures that every time a record is inserted or updated, a non-NULL value must be provided for that column
* DEFAULT Constraint: A DEFAULT constraint specifies a default value for a column. Whenever a new record is inserted without explicitly providing a value for that column, the default value will be assigned instead

**2) What is primary key? What is PRIMARY KEY constraint?**

**⇒**

* Primary Key: A primary key is a column or a set of columns that uniquely identifies each row in a table. It enforces the entity integrity of the table by ensuring that each row has a unique identifier
* PRIMARY KEY Constraint: The PRIMARY KEY constraint is used to define a primary key for a table in a database. It ensures that the values in the specified column or columns are unique and not NULL. A table can have only one primary key, and all columns within the primary key must be defined as NOT NULL

**3) What is foreign key? How do you define a foreign key in your table?**

⇒ A foreign key is a column or group of columns in one table that refers to the primary key or unique key in another table. It establishes a link between these two tables, allowing for better organization and maintenance of data.

To define a foreign key in a table, you can use the following syntax:

ALTER TABLE child\_table\_name

ADD CONSTRAINT fk\_name

FOREIGN KEY(fk\_columns)

REFERENCES parent\_table\_name(pk\_column\_names)

[ON DELETE {ACTION}]

[ON UPDATE {ACTION}];

For e.g.:

ALTER TABLE orders

ADD CONSTRAINT fk\_orders\_customers

FOREIGN KEY (customer\_id)

REFERENCES customers (id);

| Signature of Faculty | Date of Completion: |
| --- | --- |
|  |  |
|  |  |