| SEcomputer Roll number : 9914 | | | |
| --- | --- | --- | --- |
| Experiment no. : 3 part2 Date of Implementation : | | | |
| Aim : To implement data manipulation language(SQL) commands | | | |
| Tool Used : PostgreSQL | | | |
| Related Course outcome : Students should be able to  Write queries in SQL to retrieve any type of information from a database. | | | |
| **Rubrics for assessment of Experiment:**   | Indicator | Poor | Average | Good | | --- | --- | --- | --- | | Timeliness  Maintains Experiment deadline (3) | Experiment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness  Complete all parts of Experiment(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality  Extent of plagiarism(2) | Copied it from someone else(0) | At least try to implement but could not succeed (1) | Implemented (2) | | Knowledge  In depth knowledge of the Experiment(2) | Unable to answer any questions(0) | Unable to answer few questions (1) | Able to answer all questions (2) | | | | |
| **Assessment Marks :**   | Timeliness |  | | --- | --- | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | | | | |
| **Total : (Out of 10)** | | | |
| **Teacher's Sign :** | | | |
|  | ***EXPERIMENT 3*** | **DDL and DML Commands** |
|  | Aim | To implement DDL with integrity constraints and DML – Data manipulation language command |
|  | Tools | PostgreSQL/MySql |
|  | Theory | Data Definition Language-1) Create 2) Alter 3) Drop 4) Rename 5) Truncate   * [**CREATE**](https://www.geeksforgeeks.org/sql-create/) – is used to create the database or its objects (like table, index, function, views, store procedure and triggers). * [**DROP**](https://www.geeksforgeeks.org/sql-drop-truncate/) – is used to delete objects from the database. * [**ALTER**](https://www.geeksforgeeks.org/sql-alter-add-drop-modify/)-is used to alter the structure of the database. * [**TRUNCATE**](https://www.geeksforgeeks.org/sql-drop-truncate/)–is used to remove all records from a table, including all spaces allocated for the records are removed. * [**COMMENT**](https://www.geeksforgeeks.org/sql-comments/) –is used to add comments to the data dictionary. * [**RENAME**](https://www.geeksforgeeks.org/sql-alter-rename/) –is used to rename an object existing in the database.   1) Create table  create table tablename  (column1 data type,  column2 data type,  column3 data type,  ...  columnN data type );  2) **DROP object object\_name**  Examples:  DROP TABLE table\_name;  table\_name: Name of the table to be deleted.  DROP DATABASE database\_name;  database\_name: Name of the database to be deleted. |

**3) TRUNCATE**

TRUNCATE statement is a Data Definition Language (DDL) operation that is used to mark the extents of a table for deallocation (empty for reuse). The result of this operation quickly removes all data from a table, typically bypassing a number of integrity enforcing mechanisms. It was officially introduced in the standard.

The TRUNCATE TABLE mytable statement is logically (though not physically) equivalent to the DELETE FROM mytable statement (without a WHERE clause).

Syntax:

TRUNCATE TABLE table\_name;

table\_name: Name of the table to be truncated.

DATABASE name - student\_data

* **cannot** be rolled back, so it must be used wisely.

**DROP vs TRUNCATE**

* Truncate is normally ultra-fast and its ideal for deleting data from a temporary table.
* Truncate preserves the structure of the table for future use, unlike drop table where the table is deleted with its full structure.

Table or Database deletion using DROP statement

* To delete the whole database

DROP DATABASE student\_data;

After running the above query whole database will be deleted.

* To truncate Student\_details table from student\_data database.

TRUNCATE TABLE Student\_details;

After running the above query Student\_details table will be truncated, i.e, the data will be deleted but the structure will remain in the memory for further operations.

**Alter**

alter command is used for altering the table structure, such as,

* to add a column to existing table
* to rename any existing column
* to change data type of any column or to modify its size.
* to drop a column from the table.

ALTER TABLE table\_name ADD(

column\_name datatype);

| Procedure | **B)Data Manipulation Language**  A Data Manipulation Language enables programmers and users of the database to retrieve insert, delete and update data in a database. e.g. INSERT, UPDATE, DELETE, SELECT.  **INSERT:**  INSERT statement adds one or more records to any single table in a relational database.  INSERT INTO tablename VALUES (expr1,expr2……..);  **UPDATE:**  UPDATE statement that changes the data of one or more records in a table. Either all the rows can be updated, or a subset may be chosen using a condition.  UPDATE table\_name SET column\_name = value [, column\_name = value ...] [WHERE condition]  **DELETE:**  DELETE statement removes one or more records from a table. A subset may be defined for deletion using a condition, otherwise all records are removed.  DELETE FROM tablename WHERE condition |
| --- | --- |

|  | Task1: 1. Create following tables:  Table name : client\_master   | Column Name | Data type | Size |  | | --- | --- | --- | --- | | Client\_no | varchar | 6 | Primary key | | Name | varchar | 20 | Not null | | Address | varchar | 30 |  | | City | varchar | 15 |  | | Pincode | numeric | 8 |  | | State | varchar | 15 |  | | Bal\_due | numeric | 10,2 | >0 |   Table name: Product\_master   | Column Name | Data type | Size |  | | --- | --- | --- | --- | | product\_no | varchar | 6 | Primary key | | description | varchar | 15 | Not null | | Profit\_percent | numeric | 4,2 |  | | Unit\_measure | varchar | 10 |  | | Qty\_on\_hand | numeric | 8 | >0 | | Reorder\_level | numeric | 8 |  | | Sell\_price | numeric | 8,2 |  | | Cost\_price | numeric | 8,2 |  |   2. Insert 5-6 records in each table.  3. Find out the names of all clients  4. Retrieve the entire contents of the client\_master table.  5. Retrieve the list of names and cities of all the clients  6. List the various products available from the product\_master table  7. List all the clients who are located in mumbai.  8. Change the city of client\_no C001 to mumbai  9. Change the bal\_due of client\_no C005 to Rs. 1000  10. Change the cost price of 'hard disk' to Rs. 3000  11. Delete all the products from product\_master where the qty\_on\_hand is less than 100  12. Delete from client\_master where the column state holds the value 'Tamil Nadu'  Task2: Create the tables for the EER diagram of EXPT. no 2 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Post Lab Questions:** | 1. Explain different data types of Mysql/postgresql 2. Perform delete and truncate in lab and Differentiate delete and truncate |

Task 1:

Q1. create table client\_master(

client\_no varchar(6) primary key,

client\_name varchar(20) not null,

address varchar(30),

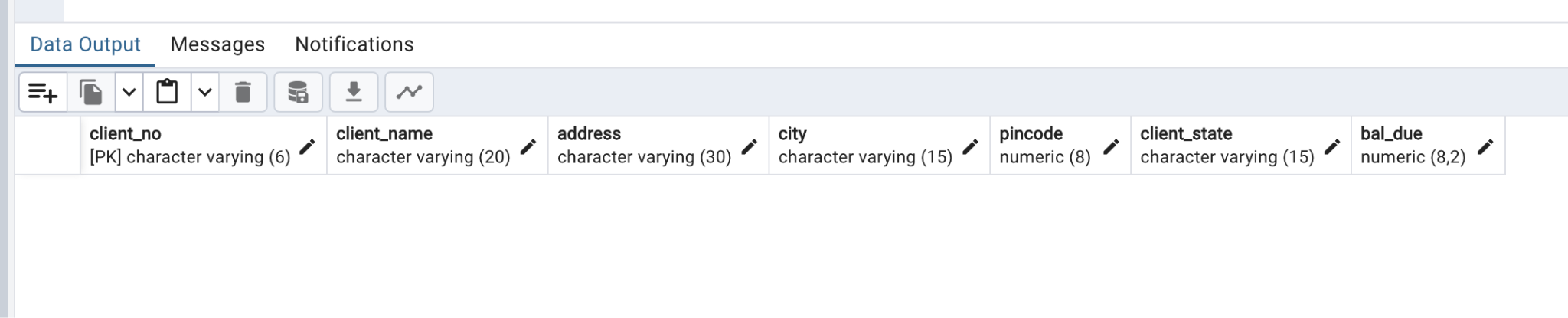
city varchar(15),

pincode numeric(8),

client\_state varchar(15),

bal\_due numeric(8, 2) check (bal\_due > 0));

select \* from client\_master;



create table Product\_master(

product\_no varchar(6) primary key,

description varchar(15) not null,

Profit\_percent numeric(4,2),

Unit\_measure varchar(10),

Qty\_on\_hand numeric(8) check (Qty\_on\_hand >0),

Reorder\_level numeric(8),

Sell\_price numeric(8, 2),

Cost\_price numeric(8, 2));

select \* from Product\_master;



Q2.

insert into client\_master

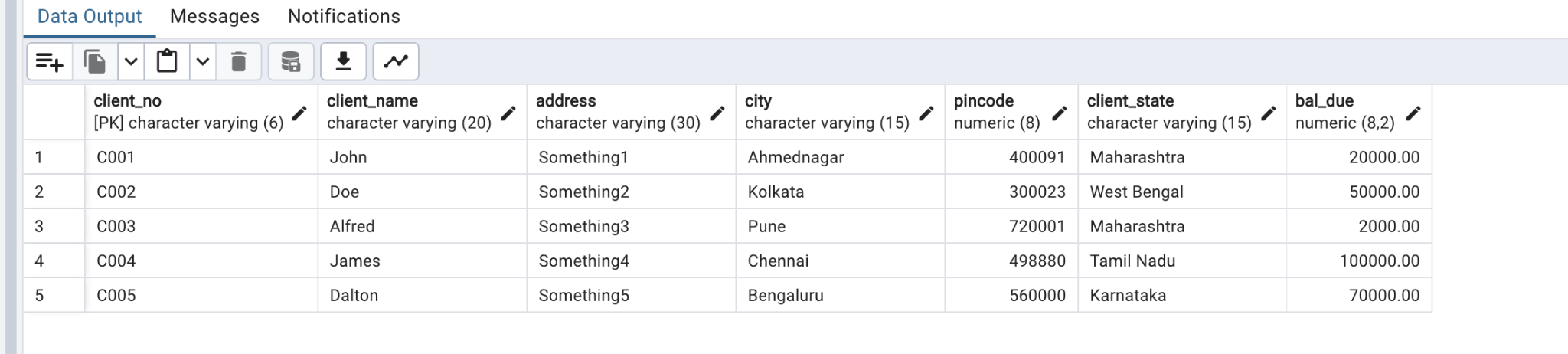
values ('C001', 'John', 'Something1','Ahmednagar', 400091, 'Maharashtra', 20000.00),

('C002', 'Doe', 'Something2', 'Kolkata', 300023, 'West Bengal', 50000.00),

('C003', 'Alfred', 'Something3', 'Pune', 720001, 'Maharashtra', 2000.00),

('C004', 'James', 'Something4', 'Chennai', 498880, 'Tamil Nadu', 100000.00),

('C005', 'Dalton', 'Something5', 'Bengaluru', 560000, 'Karnataka', 70000.00);



insert into Product\_master

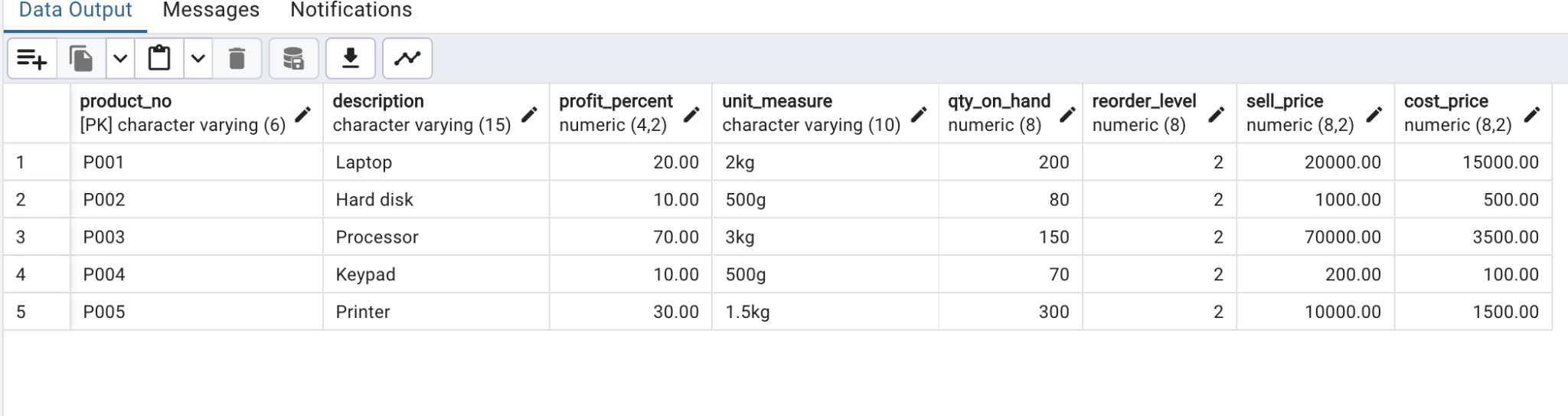
values ('P001', 'Laptop', 20.00, '2kg', 200, 2, 20000.00, 15000.00),

('P002', 'Hard disk', 10.00, '500g', 80, 2, 1000.00, 500.00),

('P003', 'Processor', 70.00, '3kg', 150, 2, 70000.00, 3500.00),

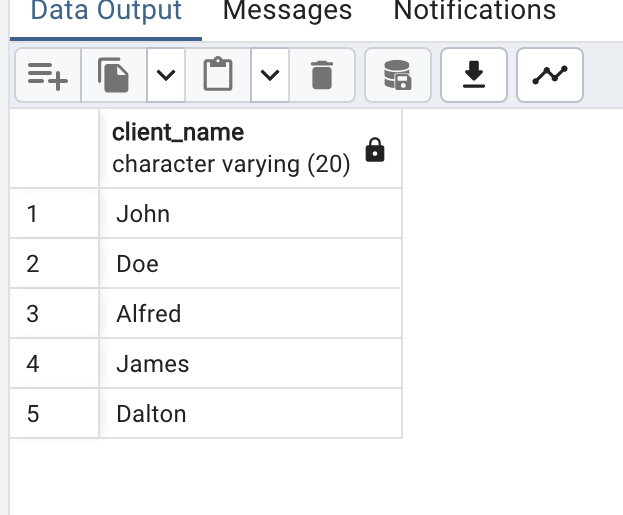
('P004', 'Keypad', 10.00, '500g', 70, 2, 200.00, 100.00),

('P005', 'Printer', 30.00, '1.5kg', 300, 2, 10000.00, 1500.00);



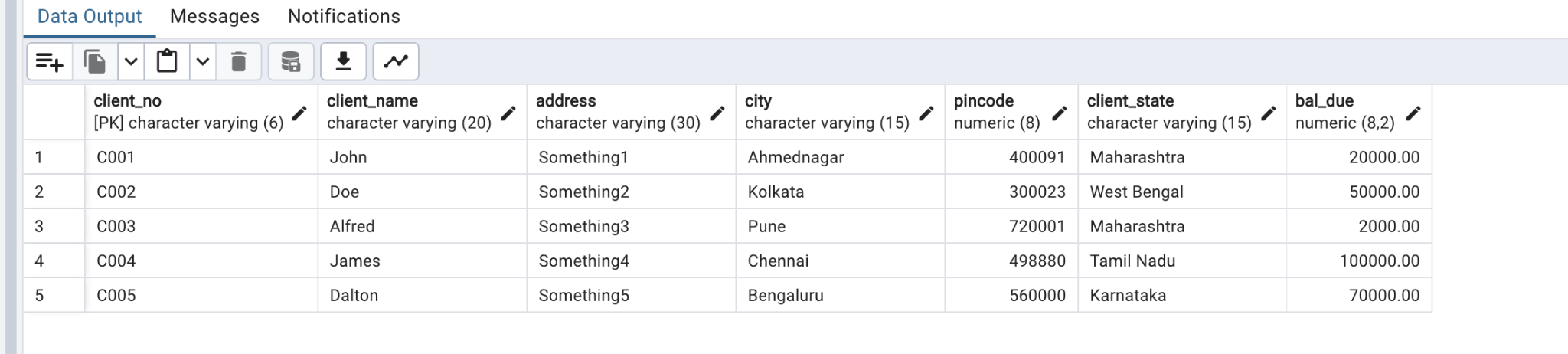
Q3.

select client\_name from client\_master;



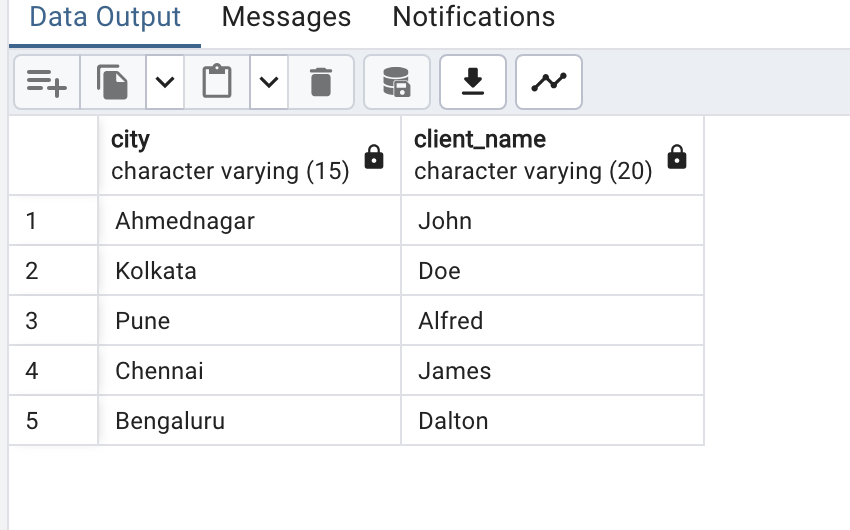
Q4.

select \* from client\_master;



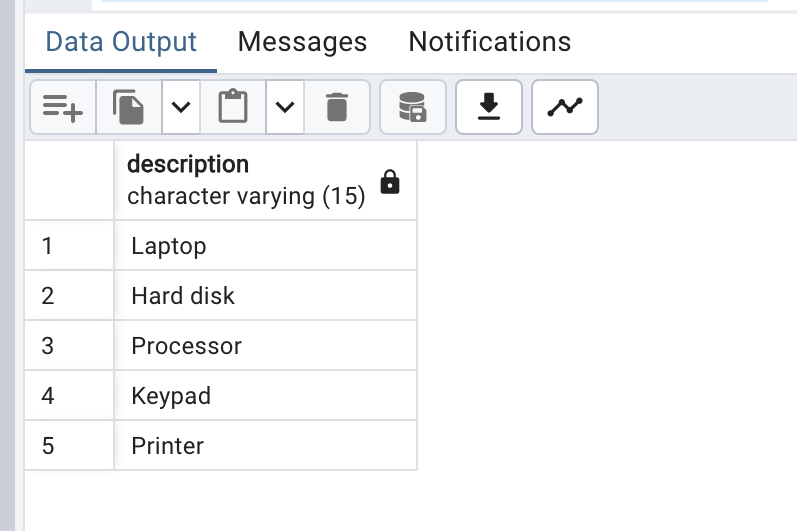
Q5.

select city, client\_name from client\_master;

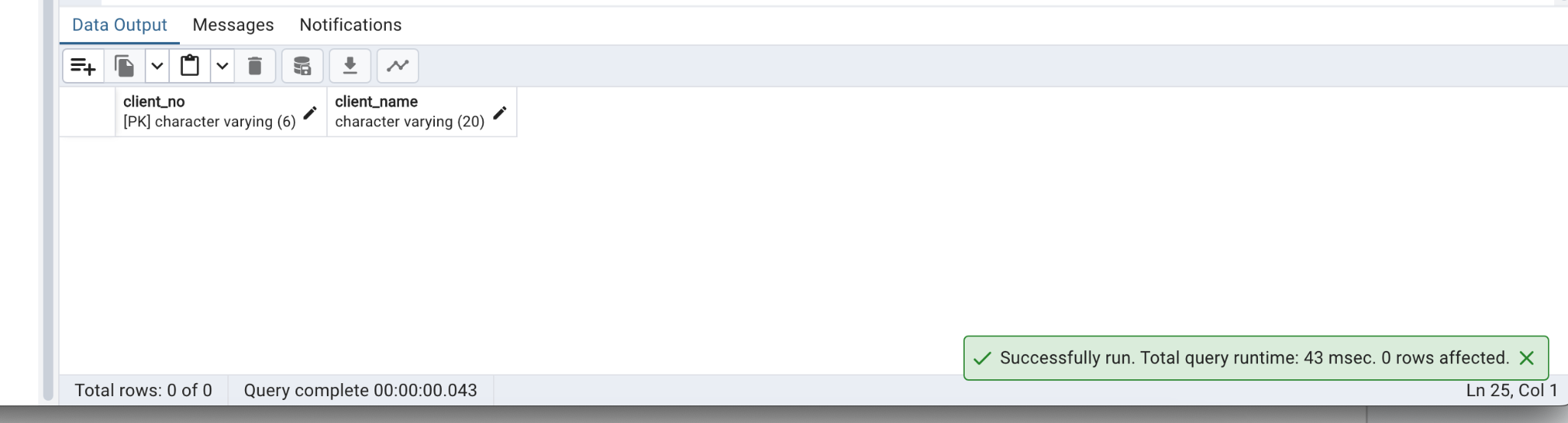


Q6.

select description from Product\_master;



Q7.

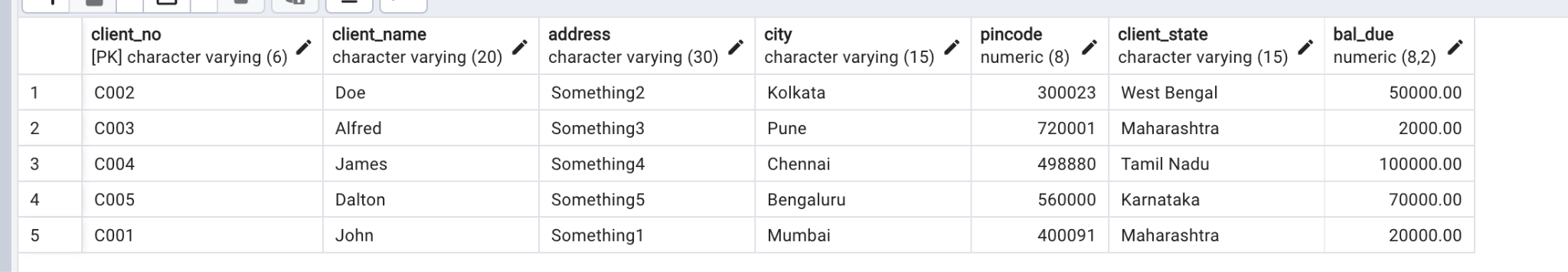
select client\_no, client\_name from client\_master where city = 'Mumbai';

Q8.

update client\_master

set city = 'Mumbai'

where client\_no = 'C001';

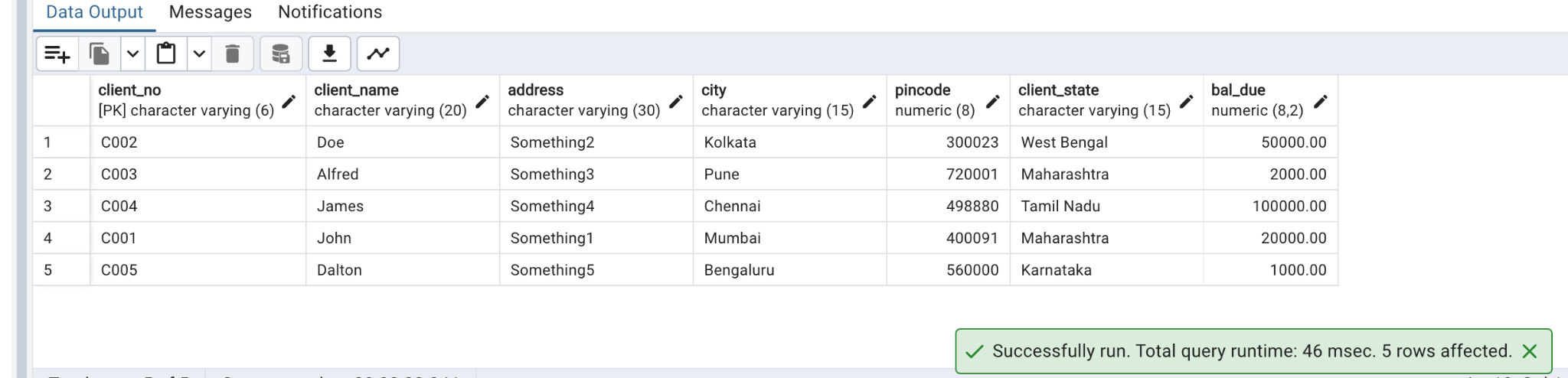


Q9.

update client\_master

set bal\_due = 1000.00

where client\_no = 'C005';

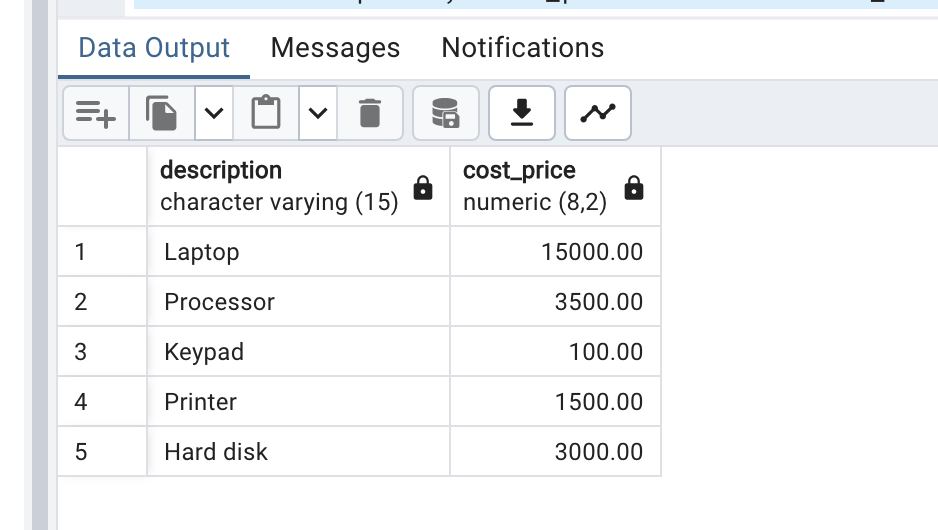


Q10.

update Product\_master

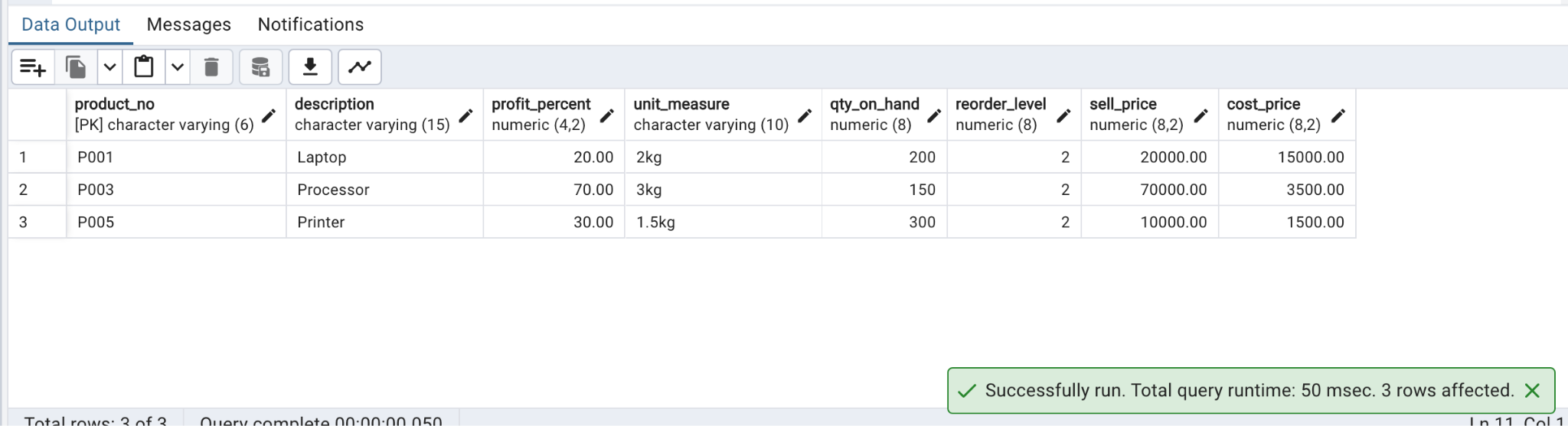
set Cost\_price = 3000.00

where description = 'Hard disk';



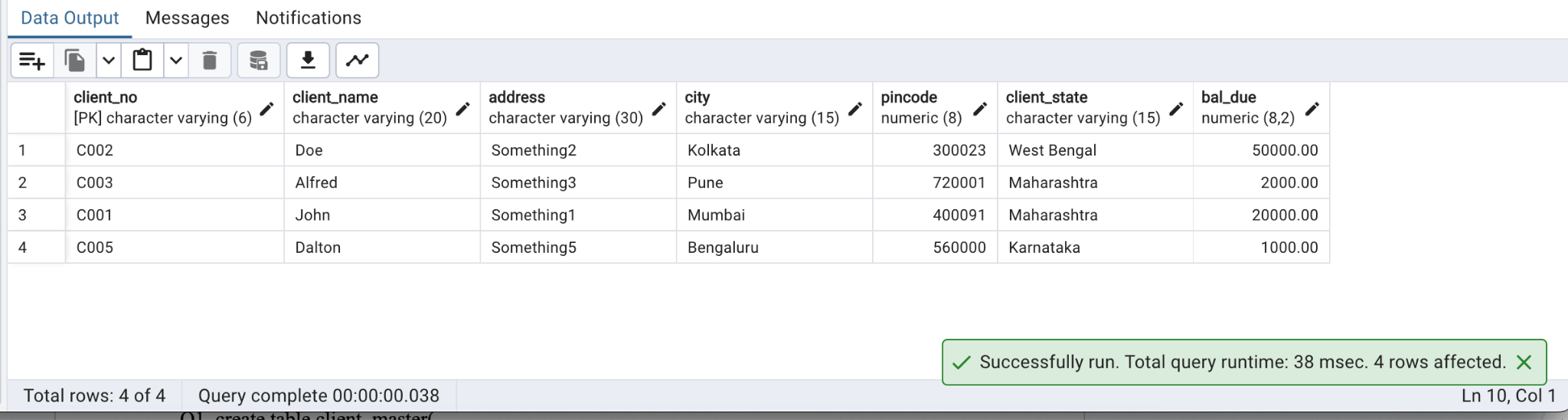
Q11.

delete from Product\_master where Qty\_on\_hand < 100;



Q12.

delete from client\_master where client\_state = 'Tamil Nadu';



Task 2:

CREATE TABLE customer (

customer\_name VARCHAR(255) PRIMARY KEY

);

CREATE TABLE branch (

branch\_id INT PRIMARY KEY,

branch\_name VARCHAR(255) NOT NULL,

branch\_address VARCHAR(255) NOT NULL

);

CREATE TABLE invoice (

invoice\_id INT PRIMARY KEY,

invoice\_date DATE NOT NULL,

customer\_name VARCHAR(255) REFERENCES customer(customer\_name),

branch\_id INT REFERENCES branch(branch\_id),

mode\_of\_payment VARCHAR(255),

invoice\_type VARCHAR(255),

gross\_amount DECIMAL(10,2) NOT NULL,

taxes DECIMAL(10,2) NOT NULL,

total\_amount DECIMAL(10,2) NOT NULL

);

CREATE TABLE items (

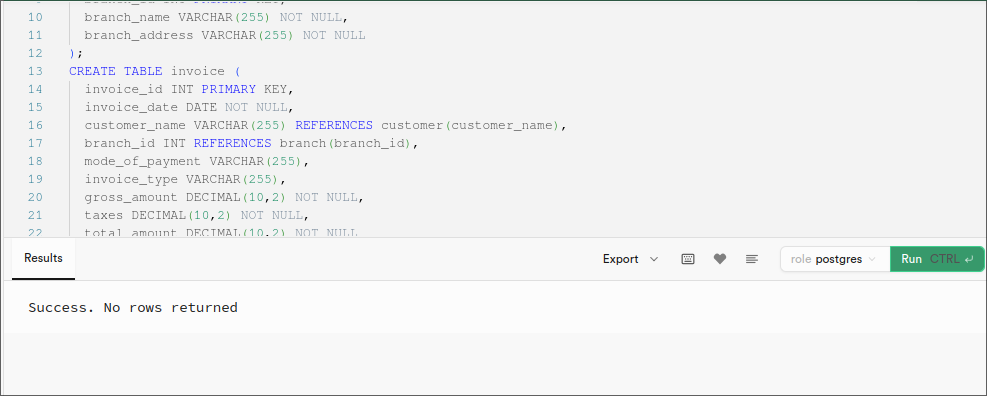
item\_code VARCHAR(255) PRIMARY KEY,

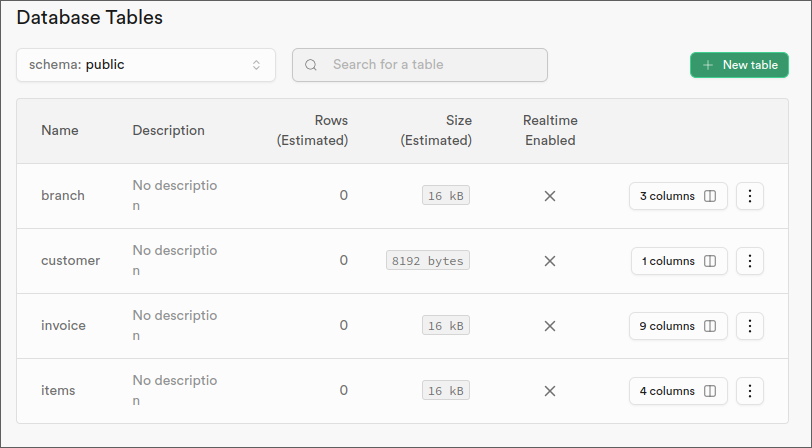
item\_name VARCHAR(255),

invoice\_id INT REFERENCES invoice(invoice\_id),

quantity INT NOT NULL

);





Postlab:

Q1.

MySQL:

1. INT: Stands for integer, used to store whole numbers without decimal points. It can store values from -2147483648 to 2147483647 for signed INT and 0 to 4294967295 for unsigned INT.

2. VARCHAR: Stands for variable-length character string. It is used to store alphanumeric characters of variable length. The maximum length of VARCHAR can be specified, e.g., VARCHAR(255).

3. DATE: Used to store dates in the format YYYY-MM-DD. It can store dates ranging from '1000-01-01' to '9999-12-31'.

4. DECIMAL: Also known as DEC or NUMERIC, it is used to store exact numeric values with decimal points. It requires two parameters: precision and scale. Precision represents the total number of digits, and scale represents the number of digits after the decimal point.

5. BOOLEAN: Used to store boolean values, which can be either true or false. In MySQL, boolean values are represented as 0 (false) or 1 (true).

PostgreSQL:

1. INTEGER: Similar to MySQL's INT, it is used to store whole numbers without decimal points. It can store values from -2147483648 to 2147483647.

2. TEXT: Used to store variable-length character data. It can hold strings of any length.

3. DATE: Similar to MySQL, it's used to store dates in the format YYYY-MM-DD.

4. NUMERIC: Similar to MySQL's DECIMAL, it is used to store exact numeric values with decimal points. It requires two parameters: precision and scale. Precision represents the total number of digits, and scale represents the number of digits after the decimal point.

5. BOOLEAN: Used to store boolean values, which can be either true or false. In PostgreSQL, boolean values are represented as 'true' or 'false'.

Q2.

1. DELETE:

- In PostgreSQL, ‘DELETE’ is a SQL command that removes rows(record) from a table based on a specified condition using a WHERE clause.

- ‘DELETE’ is a DML (Data Manipulation Language) statement.

- It can be rolled back using a transaction if it is within a transaction block.

- When using ‘DELETE’, PostgreSQL marks the rows as deleted and space can be reused later.

- ‘DELETE’ triggers are fired if defined for the table.

- It maintains the integrity constraints such as foreign keys and triggers associated with the table.

Example:

DELETE FROM table\_name WHERE condition;

2. TRUNCATE:

- In PostgreSQL, ‘TRUNCATE’ is a SQL command that removes all rows from a table.

- ‘TRUNCATE’ is a DDL (Data Definition Language) statement.

- It removes all rows without considering any condition. No WHERE clause can be specified.

- Unlike ‘DELETE’, ‘TRUNCATE’ cannot be rolled back using a transaction.

- ‘TRUNCATE’ resets sequences used by the table to their initial value.

- It is faster than `DELETE`, especially for large tables, as it deallocates the storage space occupied by the table.

- ‘TRUNCATE’ does not invoke triggers.

- It resets any associated auto-increment sequences.

Example:

TRUNCATE TABLE table\_name;