

SEcomputer A batch-C		Roll number : 9913	
Experiment no. : 3 part2		Date of Implementation :	
Aim : To implement data manipulation language (DML) commands			
Tool Used : PostgreSQL			
Related Course outcome : Students should be able to Write queries in SQL to retrieve any type of information from a data base.			
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	<p>Data Definition Language-1) Create 2) Alter 3) Drop 4) Rename 5) Truncate</p> <ul style="list-style-type: none"> • <u>CREATE</u> – is used to create the database or its objects (like table, index, function, views, store procedure and triggers). • <u>DROP</u> – is used to delete objects from the database. • <u>ALTER</u>–is used to alter the structure of the database. • <u>TRUNCATE</u>–is used to remove all records from a table, including all spaces allocated for the records are removed. • <u>COMMENT</u> –is used to add comments to the data dictionary. • <u>RENAME</u> –is used to rename an object existing in the database. <p>1) Create table create table tablename (column1 data type, column2 data type, column3 data type, ... columnN data type);</p> <p>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.</p>

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	<p>B)Data Manipulation Language</p> <p>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.</p> <p><u>INSERT:</u></p> <p>INSERT statement adds one or more records to any single table in a relational database.</p> <p>INSERT INTO tablename VALUES (expr1,expr2.....);</p> <p><u>UPDATE:</u></p> <p>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.</p> <p>UPDATE table_name SET column_name = value [, column_name = value ...] [WHERE condition]</p> <p><u>DELETE:</u></p> <p>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.</p> <p>DELETE FROM tablename WHERE condition</p>
-----------	--

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 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

Q1

The screenshot displays a database management interface with two panels. The top panel shows the creation of a table named 'client_master' with the following SQL code:

```
1 -- CREATE TABLE client_master (  
2 --     client_no VARCHAR(6) PRIMARY KEY,  
3 --     name VARCHAR(20) NOT NULL,  
4 --     address VARCHAR(30),  
5 --     city VARCHAR(15),  
6 --     pincode NUMERIC(8),  
7 --     state VARCHAR(15),  
8 --     bal_due NUMERIC(10,2) CHECK (bal_due > 0)  
9 -- );  
10 select * from client_master
```

The bottom panel shows the creation of a table named 'Product_master' with the following SQL code:

```
1 CREATE TABLE Product_master (  
2     product_no VARCHAR(6) PRIMARY KEY,  
3     description VARCHAR(15) NOT NULL,  
4     Profit_percent NUMERIC(4,2),  
5     Unit_measure VARCHAR(10),  
6     Qty_on_hand NUMERIC(8) check (Qty_on_hand > 0),  
7     Recorder_level NUMERIC(8),  
8     Sell_price NUMERIC(8,2),  
9     Cost_price NUMERIC(8,2)  
10 );  
11 select * from Product_master  
12
```

Both panels include a 'Data output' section at the bottom, which lists the columns and their data types for the respective tables. The 'client_master' table has columns: client_no [PK] character varying (6), name character varying (20), address character varying (30), city character varying (15), pincode numeric (8), state character varying (15), and bal_due numeric (10,2). The 'Product_master' table has columns: product_no [PK] character varying (6), description character varying (15), profit_percent numeric (4,2), unit_measure character varying (10), qty_on_hand numeric (8), recorder_level numeric (8), sell_price numeric (8,2), and cost_price numeric (8,2).

Q2

PostgreSQL

1 INSERT INTO client_master

2 VALUES

3 ('C001', 'John', '123 Main', 'Mumbai', 400091, 'Maharashtra', 20000.00),

4 ('C002', 'Doe', '456 Elm', 'Kolkata', 300023, 'West Bengal', 50000.00),

5 ('C003', 'Alfred', '789 Oak', 'Pune', 720001, 'Maharashtra', 2000.00),

6 ('C004', 'James', '321 Map', 'Chennai', 498880, 'Tamil Nadu', 100000.00),

7 ('C005', 'Dalton', '567 Pine', 'Mumbai', 560000, 'Karnataka', 70000.00);

8

9 SELECT * FROM client_master

client_master - TABLE

client_no	name	address	city	pincode	state	bal_due
C001	John	123 Main	Mumbai	400091	Maharashtra	20000.00
C002	Doe	456 Elm	Kolkata	300023	West Bengal	50000.00
C003	Alfred	789 Oak	Pune	720001	Maharashtra	2000.00
C004	James	321 Map	Chennai	498880	Tamil Nadu	100000.00
C005	Dalton	567 Pine	Mumbai	560000	Karnataka	70000.00

1 INSERT INTO product_master

2 VALUES

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

4 ('P002', 'Hard disk', 10.00, '500g', 80, 2, 1500.00, 500.00),

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

6 ('P004', 'Keypad', 10.00, '500g', 70, 2, 2000.00, 100.00),

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

8

9 SELECT * FROM product_master

product_no	description	profit_perc...	unit_measure	qty_on_hand	recorder_le...	sell_price	cost_price
P001	Laptop	20.00	2kg	200	2	20000.00	15000.00
P002	Hard disk	10.00	500g	80	2	1500.00	500.00
P003	Processor	70.00	3kg	150	2	70000.00	3500.00
P004	Keypad	10.00	500g	70	2	2000.00	100.00
P005	Printer	30.00	1.5kg	300	2	10000.00	1500.00

3

PostgreSQL

```
1 SELECT NAME FROM client_master
```

name
John
Doe
Alfred
James
Dalton

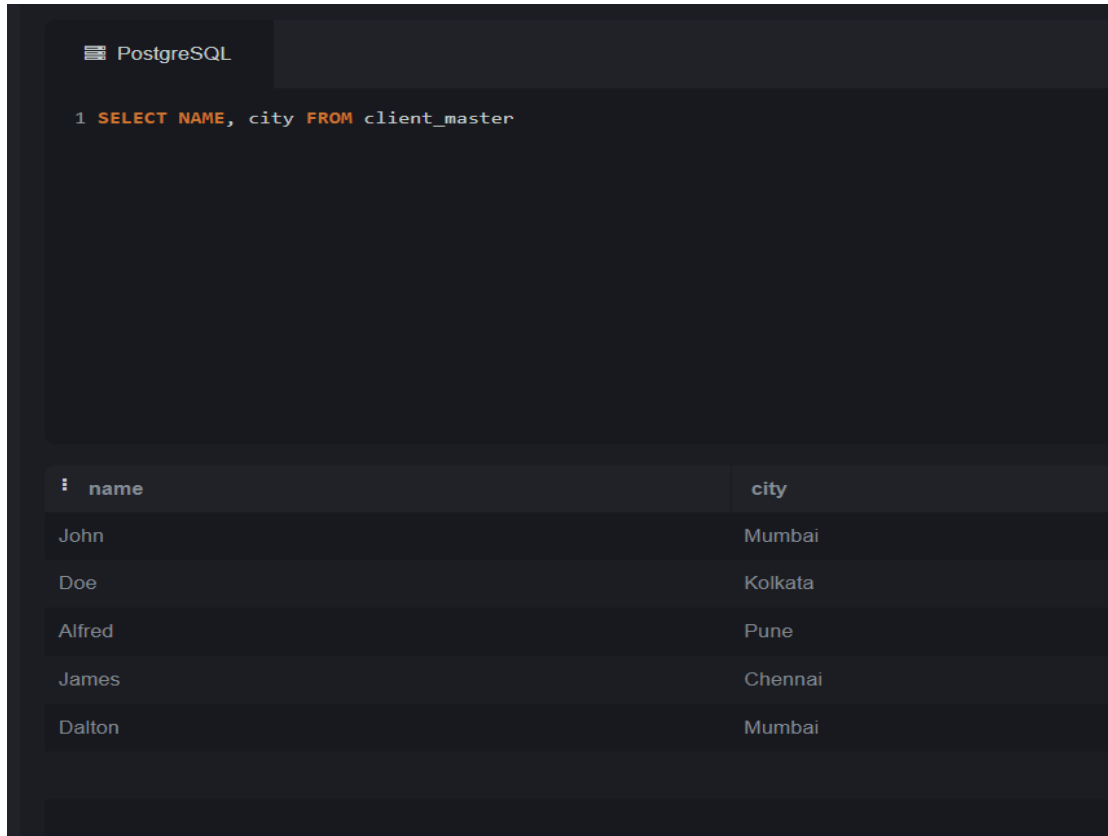
4

PostgreSQL

```
1 INSERT INTO client_master
2 VALUES
3 ('C001', 'John', '123 Main', 'Mumbai', 400091, 'Maharashtra', 20000.00),
4 ('C002', 'Doe', '456 Elm', 'Kolkata', 300023, 'West Bengal', 50000.00),
5 ('C003', 'Alfred', '789 Oak', 'Pune', 720001, 'Maharashtra', 2000.00),
6 ('C004', 'James', '321 Map', 'Chennai', 498880, 'Tamil Nadu', 100000.00),
7 ('C005', 'Dalton', '567 Pine', 'Mumbai', 560000, 'Karnataka', 70000.00);
8
9 SELECT * FROM client_master
```

client_no	name	address	city	pincode	state	bal_due
C001	John	123 Main	Mumbai	400091	Maharashtra	20000.00
C002	Doe	456 Elm	Kolkata	300023	West Bengal	50000.00
C003	Alfred	789 Oak	Pune	720001	Maharashtra	2000.00
C004	James	321 Map	Chennai	498880	Tamil Nadu	100000.00
C005	Dalton	567 Pine	Mumbai	560000	Karnataka	70000.00

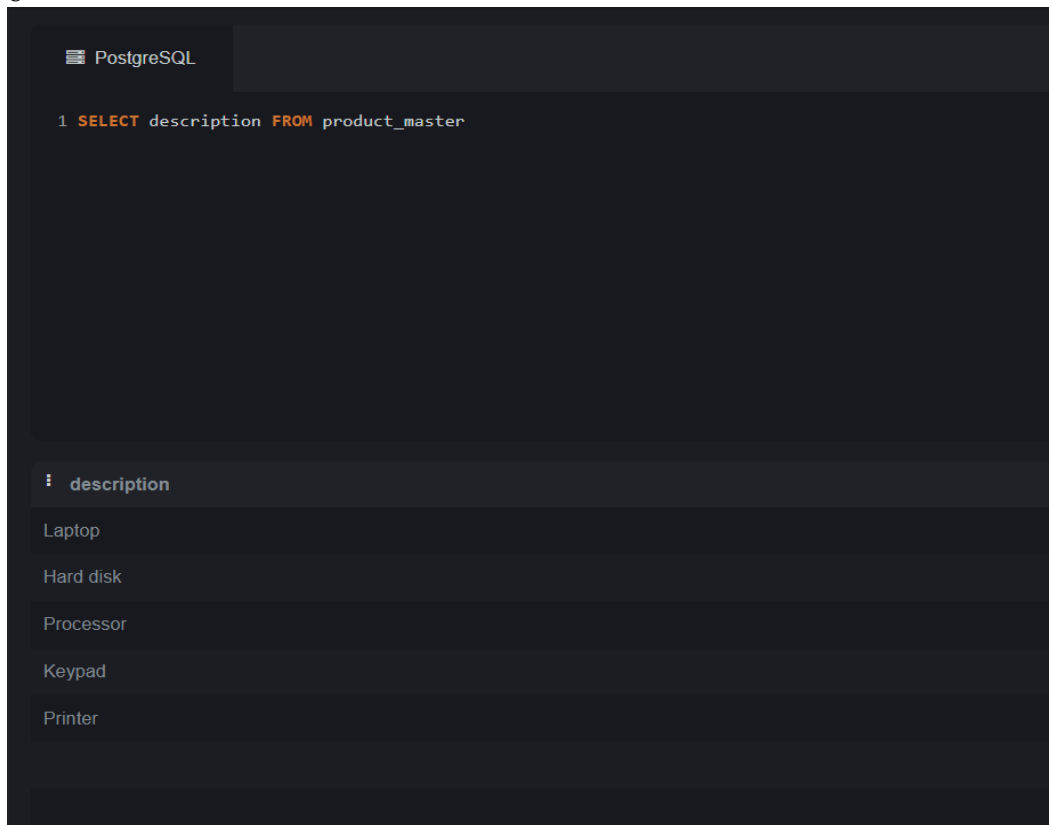
5



The image shows a PostgreSQL query editor interface. At the top, there is a tab labeled "PostgreSQL". Below the tab, a SQL query is entered: `1 SELECT NAME, city FROM client_master`. The query is executed, and the results are displayed in a table below. The table has two columns: "name" and "city". The results are as follows:

name	city
John	Mumbai
Doe	Kolkata
Alfred	Pune
James	Chennai
Dalton	Mumbai

6



The image shows a PostgreSQL query editor interface. At the top, there is a tab labeled "PostgreSQL". Below the tab, a SQL query is entered: `1 SELECT description FROM product_master`. The query is executed, and the results are displayed in a table below. The table has one column: "description". The results are as follows:

description
Laptop
Hard disk
Processor
Keypad
Printer

7

PostgreSQL

```

1 SELECT * FROM client_master WHERE city = 'Mumbai';
2

```

client_no	name	address	city	pincode	state	bal_due
C001	John	123 Main	Mumbai	400091	Maharashtra	20000.00
C005	Dalton	567 Pine	Mumbai	560000	Karnataka	70000.00

8

```

1 UPDATE client_master SET city = 'Mumbai' WHERE client_no = 'C001';

```

History

Syntax | History

PostgreSQL

```
UPDATE client_master SET city = 'Mumbai'
```

20:54:09

9

```

1 UPDATE client_master SET bal_due = 1000 WHERE client_no = 'C005';
2 SELECT * FROM client_master

```

client_no	name	address	city	pincode	state	bal_due
C002	Doe	456 Elm	Kolkata	300023	West Bengal	50000.00
C003	Alfred	789 Oak	Pune	720001	Maharashtra	2000.00
C004	James	321 Map	Chennai	498880	Tamil Nadu	100000.00
C001	John	123 Main	Mumbai	400091	Maharashtra	20000.00
C005	Dalton	567 Pine	Mumbai	560000	Karnataka	1000.00

10

```

1 UPDATE product_master SET Cost_price = 3000 WHERE description = 'Hard disk';
2 SELECT * FROM product_master

```

product_...	description	profit_perc...	unit_measure	qty_on_hand	recorder_le...	sell_price	cost_price
P001	Laptop	20.00	2kg	200	2	20000.00	15000.00
P003	Processor	70.00	3kg	150	2	70000.00	3500.00
P004	Keypad	10.00	500g	70	2	2000.00	100.00
P005	Printer	30.00	1.5kg	300	2	10000.00	1500.00
P002	Hard disk	10.00	500g	80	2	1500.00	3000.00

11

```

1 DELETE FROM product_master WHERE qty_on_hand < 100;
2 SELECT * FROM product_master

```

product_...	description	profit_perc...	unit_measure	qty_on_hand	recorder_le...	sell_price	cost_price
P001	Laptop	20.00	2kg	200	2	20000.00	15000.00
P003	Processor	70.00	3kg	150	2	70000.00	3500.00
P005	Printer	30.00	1.5kg	300	2	10000.00	1500.00

12

client_no	name	address	city	pincode	state	bal_due
C002	Doe	456 Elm	Kolkata	300023	West Bengal	50000.00
C003	Alfred	789 Oak	Pune	720001	Maharashtra	2000.00
C001	John	123 Main	Mumbai	400091	Maharashtra	20000.00
C005	Dalton	567 Pine	Mumbai	560000	Karnataka	1000.00

Customer
<u>customer id</u>
name
number
address
credit card no

~~Cart contains~~

pays
customer id
transaction id

Cart contains
Transaction id
date and time
Total cost
quantity
item id

items_distributors

<u>item_id</u>	image	name	price	description
----------------	-------	------	-------	-------------

Shipper
Contact no
s name

1. Integer Types:

MySQL:

TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT

PostgreSQL:

SMALLINT, INTEGER, BIGINT

2. Decimal/Floating-Point Types:

MySQL:

DECIMAL, FLOAT, DOUBLE

PostgreSQL:

DECIMAL, NUMERIC, REAL, DOUBLE PRECISION

3. String/Character Types:

MySQL:

CHAR, VARCHAR, TEXT

PostgreSQL:

CHAR, VARCHAR, TEXT

4. Date and Time Types:

MySQL:

DATE, TIME, DATETIME, TIMESTAMP

PostgreSQL:

DATE, TIME, TIMESTAMP, INTERVAL

5. Boolean Type:

MySQL:

BOOLEAN

PostgreSQL:

BOOLEAN

DELETE:

The DELETE statement is used to remove specific rows from a table based on a condition specified in the WHERE clause.

It allows more flexibility as you can delete specific rows that meet certain criteria.

TRUNCATE:

The TRUNCATE statement is used to remove all rows from a table.

It removes all rows without considering any conditions. It effectively deletes all data from the table.