SE-Computer A Batch C	Roll number : 9913
Experiment no. : 3(Part-1)	Date of Implementation :
Aim : To implement data definition language (DDL) commands	
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 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 assignment(2)	Unable to answer 2 questions(0)	Unable to answer 1 question (1)	Able to answer 2 questions (2)

Assessment Marks :

Timeliness	
Completeness and neatness	
Originality	
Knowledge	
Total	

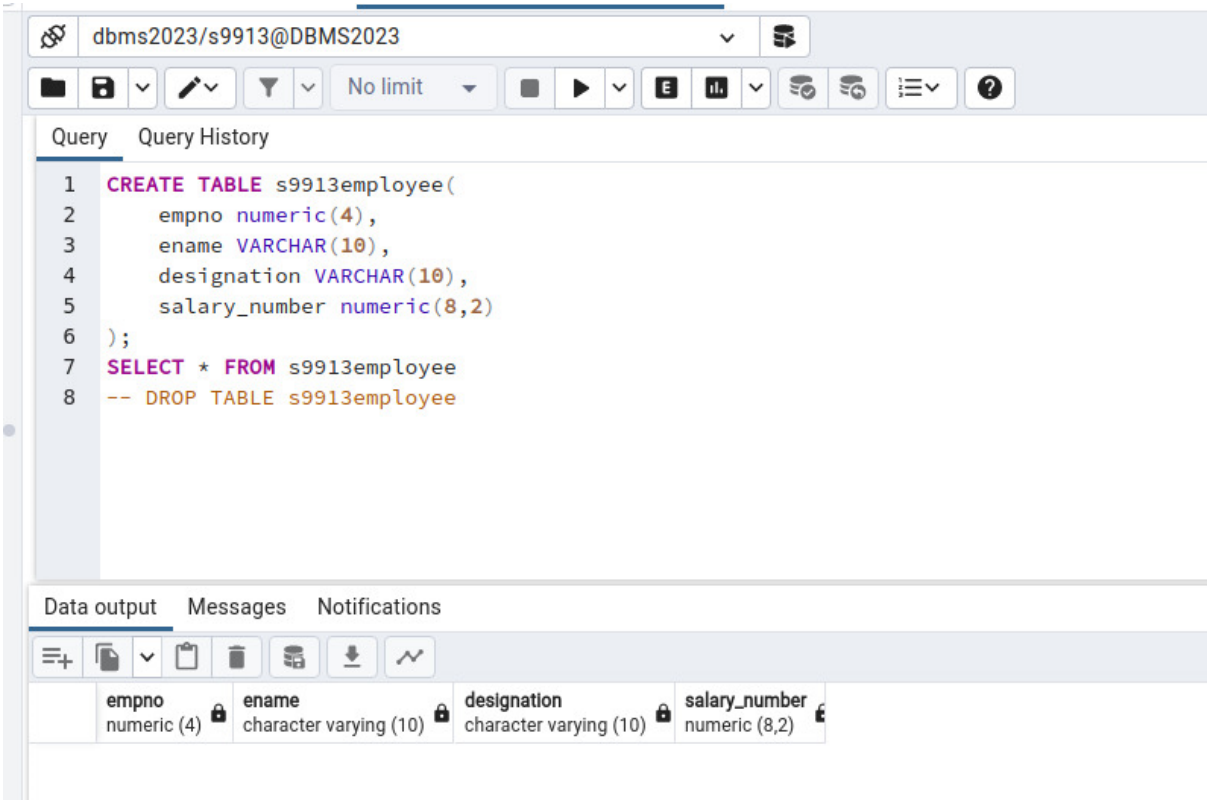
Total : (Out of 10)

Teacher's Sign :

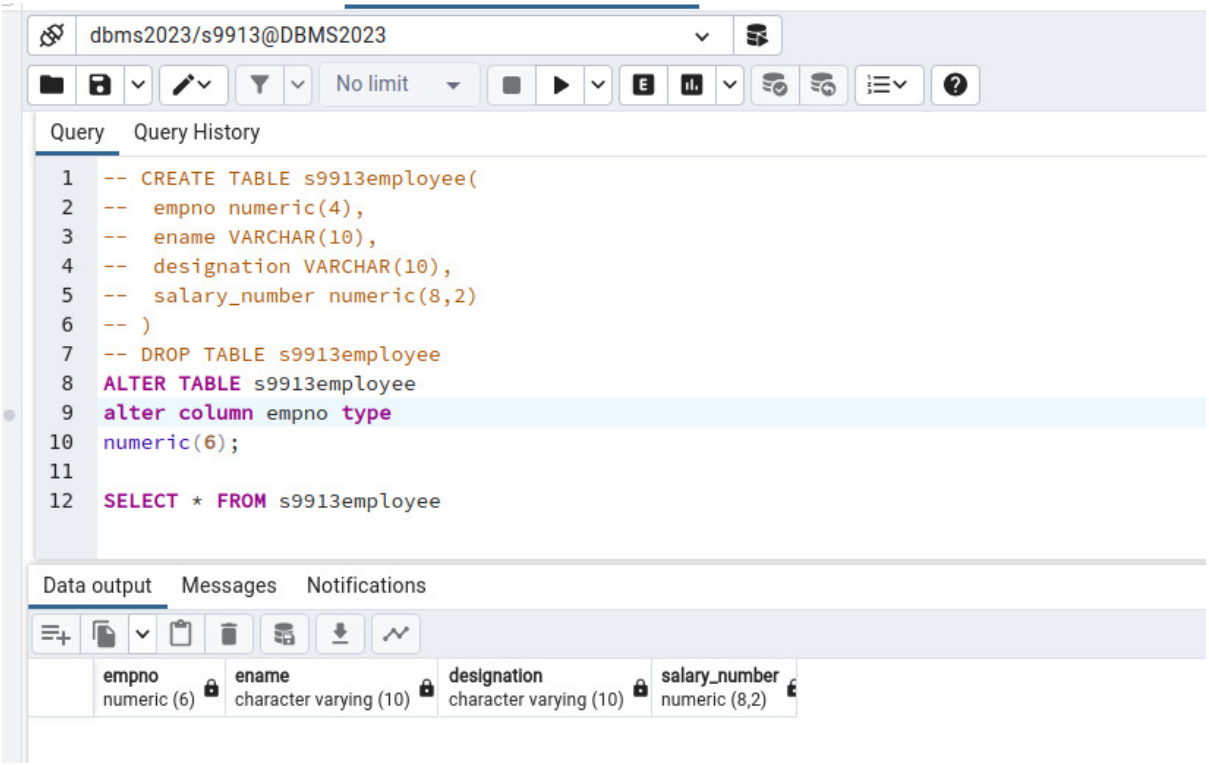
	EXPERIMENT 3	DDL Commands
	Aim	To implement DDL – Data definition language command
	Tools	PostgreSQL/MYSQL
	Theory	**SQL:** It is structured query language, basically used to pass the query to retrieve and manipulate the information from database **DDL:** The Data Definition Language (DDL) is used to create the database (i.e. tables, keys, relationships etc), maintain the structure of the database and destroy databases and database objects. Eg. Create, Drop, Alter, Describe, Truncate **1. CREATE statements: It is used to create the table.** CREATE TABLE table_name(columnName1 datatype(size), columnName2 datatype(size),……); 2. **DROP statements:** To destroy an existing database, table, index, or view. If a table is dropped all records held within it are lost and cannot be recovered. DROP TABLE table_name; 3. **ALTER statements:** To modify an existing database object. **Adding new columns:** Alter table table_name Add(New_columnName1 datatype(size), New_columnName2 datatype(size),……); **Dropping a columns from a table** : Alter table table_name DROP column columnName: **Modifying Existing columns:** Alter table table_name Modify (columnName1 Newdatatype(Newsize)); 4. **Describe statements:** To describe the structure (column and data types) of an existing database, table, index, or view. DESC table_name; 5. **Truncate statements:** To destroy the data in an existing database, table, index, or view. If a table is truncated all records held within it are lost and cannot be recovered but the table structure is maintained. TRUNCATE TABLE table_name;

	Procedure	1. Write a query to create a table employee with empno, ename, designation, and salary. Emp (empno number (4), ename varchar (10), designation varchar (10), salary number (8,2)); 2. Write a Query to Alter the column empno number (4) to empno number (6). 3. Write a Query to Alter the table employee with multiple columns (empno, ename.) 4. Write a query to add a new column in to employee as qualification varchar2(6) 5. Write a query to add multiple columns in to employee dob date , doj date 6. Write a query to drop a column 'doj' from an existing table employee 7. Write a query to drop multiple columns 'dob' and 'qualification' from employee 8. Truncate table EMP 9. Drop table EMP
	Post Lab Questions:	1. What is Data Dictionary? 2. What is Schema? 3. What are different data types in SQL?

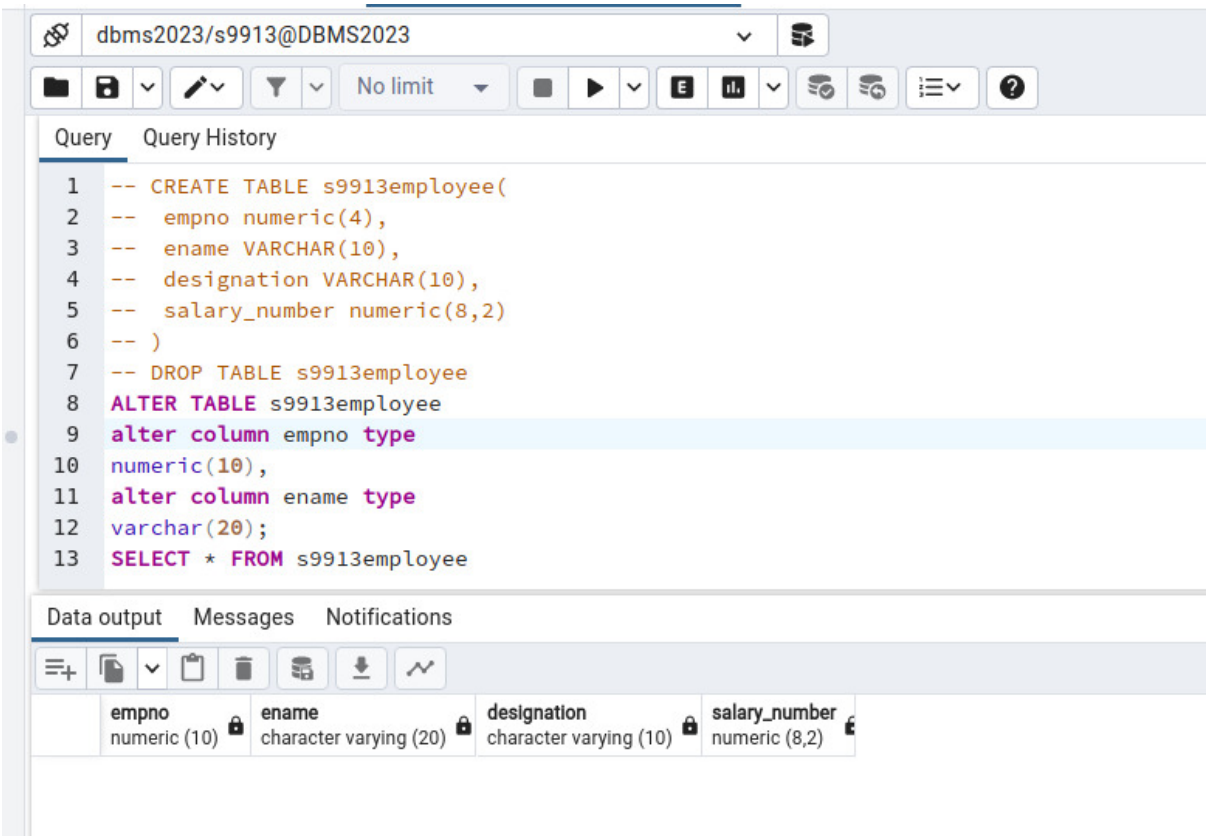
1



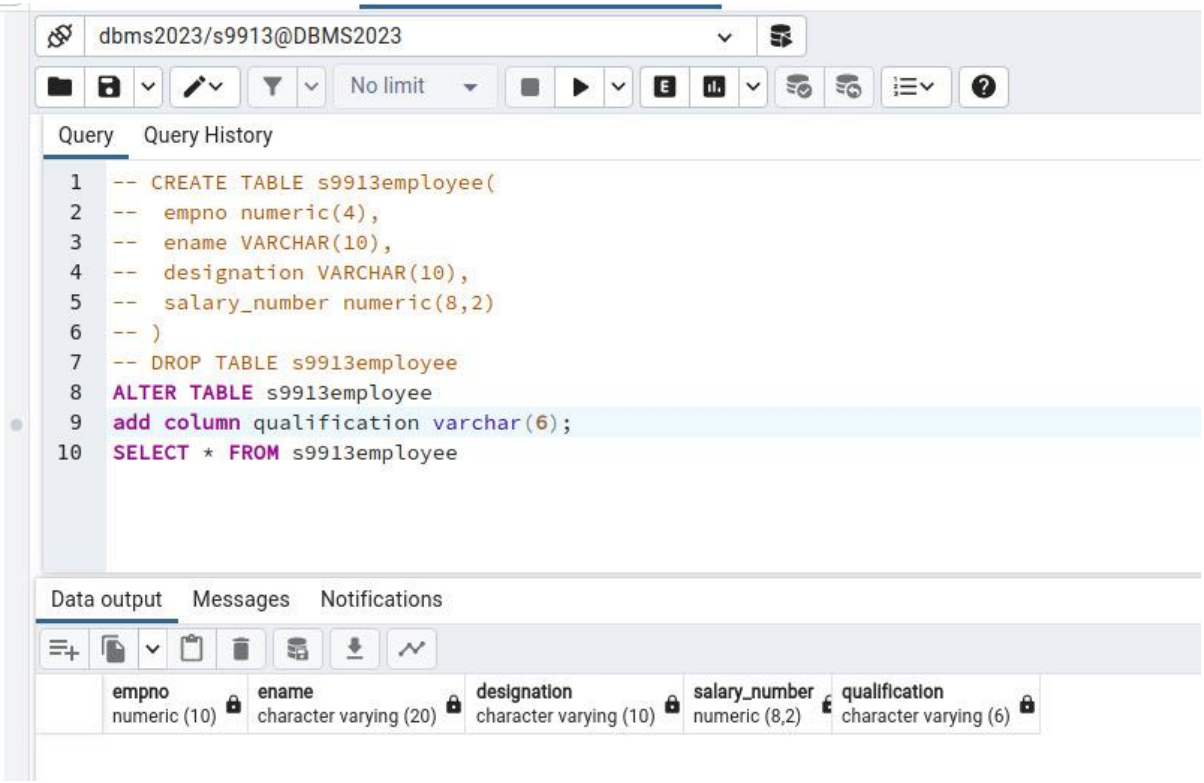
2


dbms2023/s9913@DBMS2023
No limit
Query
Query History
1 -- CREATE TABLE s9913employee(
2 -- empno numeric(4),
3 -- ename VARCHAR(10),
4 -- designation VARCHAR(10),
5 -- salary_number numeric(8,2)
6 --)
7 -- DROP TABLE s9913employee
8 ALTER TABLE s9913employee
9 alter column empno type
10 numeric(6);
11
12 SELECT * FROM s9913employee
Data output
Messages
Notifications
empno numeric (6)
ename character varying (10)
designation character varying (10)
salary_number numeric (8,2)

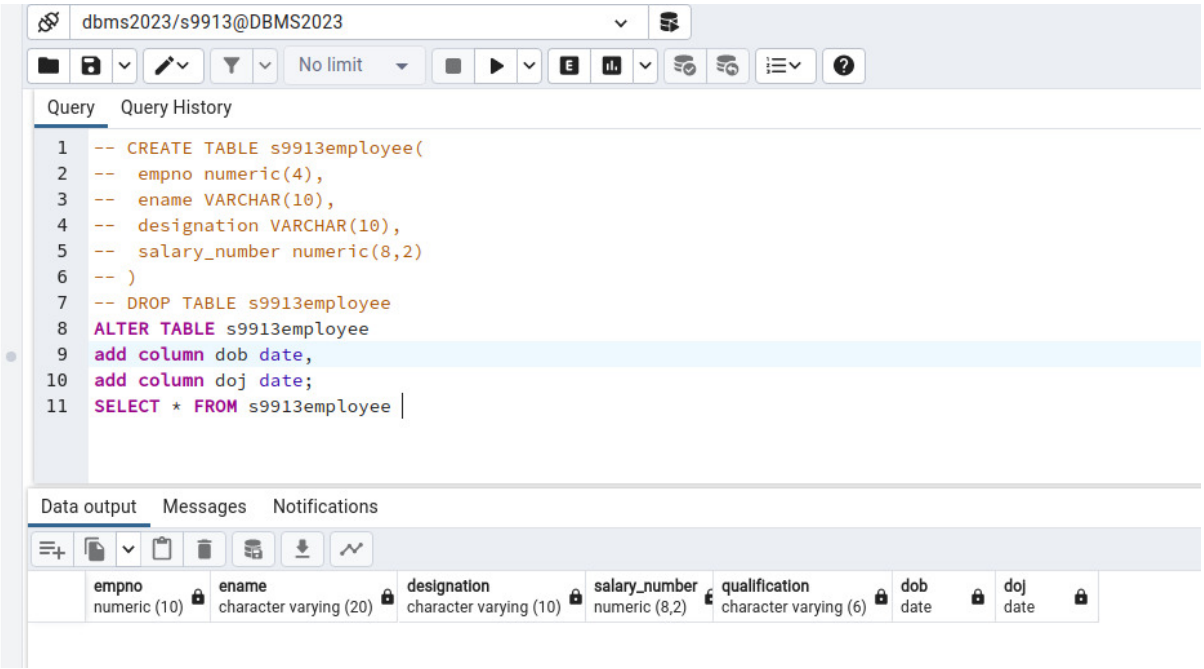
3


dbms2023/s9913@DBMS2023
No limit
Query
Query History
1 -- CREATE TABLE s9913employee(
2 -- empno numeric(4),
3 -- ename VARCHAR(10),
4 -- designation VARCHAR(10),
5 -- salary_number numeric(8,2)
6 --)
7 -- DROP TABLE s9913employee
8 ALTER TABLE s9913employee
9 alter column empno type
10 numeric(10),
11 alter column ename type
12 varchar(20);
13 SELECT * FROM s9913employee
Data output
Messages
Notifications
empno numeric (10)
ename character varying (20)
designation character varying (10)
salary_number numeric (8,2)

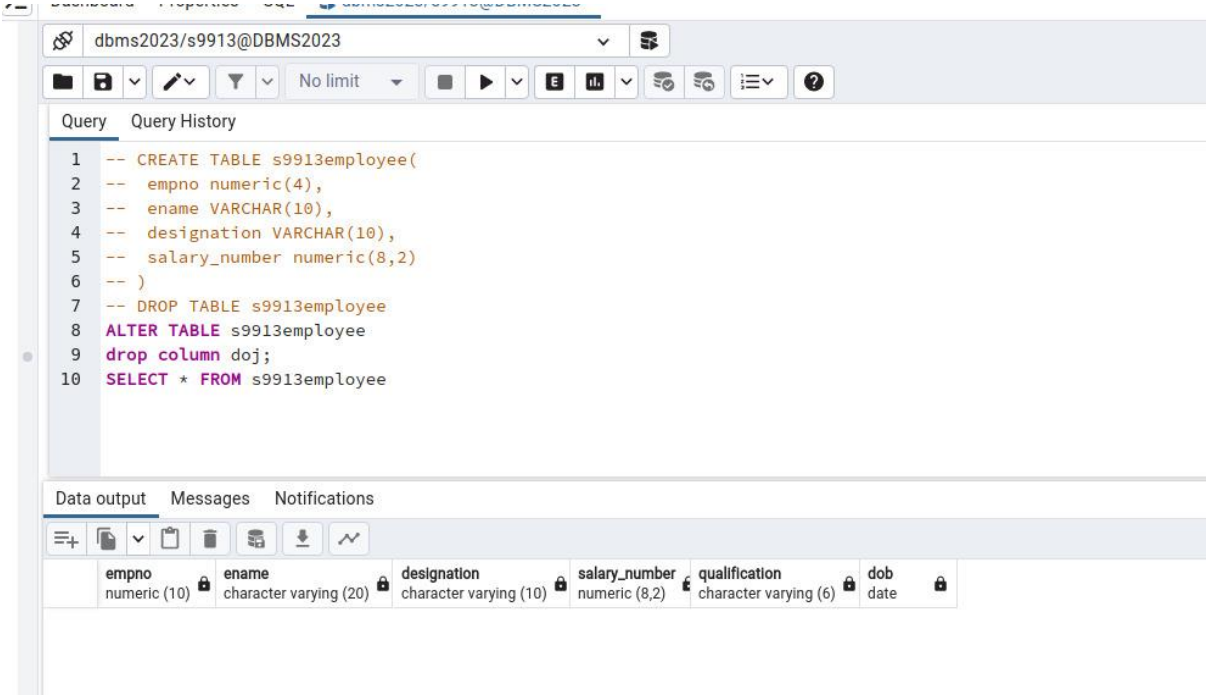
4


dbms2023/s9913@DBMS2023
No limit
Query
Query History
1 -- CREATE TABLE s9913employee(
2 -- empno numeric(4),
3 -- ename VARCHAR(10),
4 -- designation VARCHAR(10),
5 -- salary_number numeric(8,2)
6 --)
7 -- DROP TABLE s9913employee
8 ALTER TABLE s9913employee
9 add column qualification varchar(6);
10 SELECT * FROM s9913employee
Data output
Messages
Notifications
empno numeric (10)
ename character varying (20)
designation character varying (10)
salary_number numeric (8,2)
qualification character varying (6)

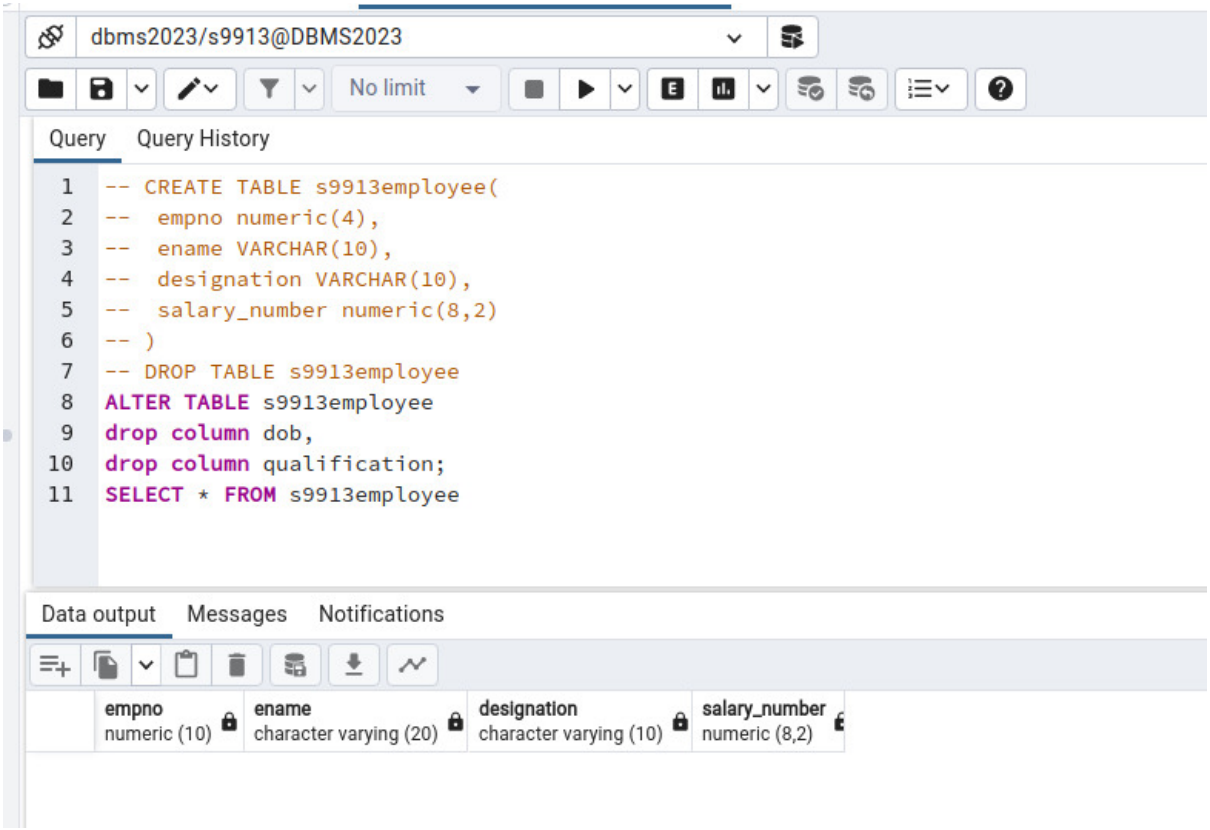
5


dbms2023/s9913@DBMS2023
No limit
Query
Query History
1 -- CREATE TABLE s9913employee(
2 -- empno numeric(4),
3 -- ename VARCHAR(10),
4 -- designation VARCHAR(10),
5 -- salary_number numeric(8,2)
6 --)
7 -- DROP TABLE s9913employee
8 ALTER TABLE s9913employee
9 add column dob date,
10 add column doj date;
11 SELECT * FROM s9913employee
Data output
Messages
Notifications
empno numeric (10)
ename character varying (20)
designation character varying (10)
salary_number numeric (8,2)
qualification character varying (6)
dob date
doj date

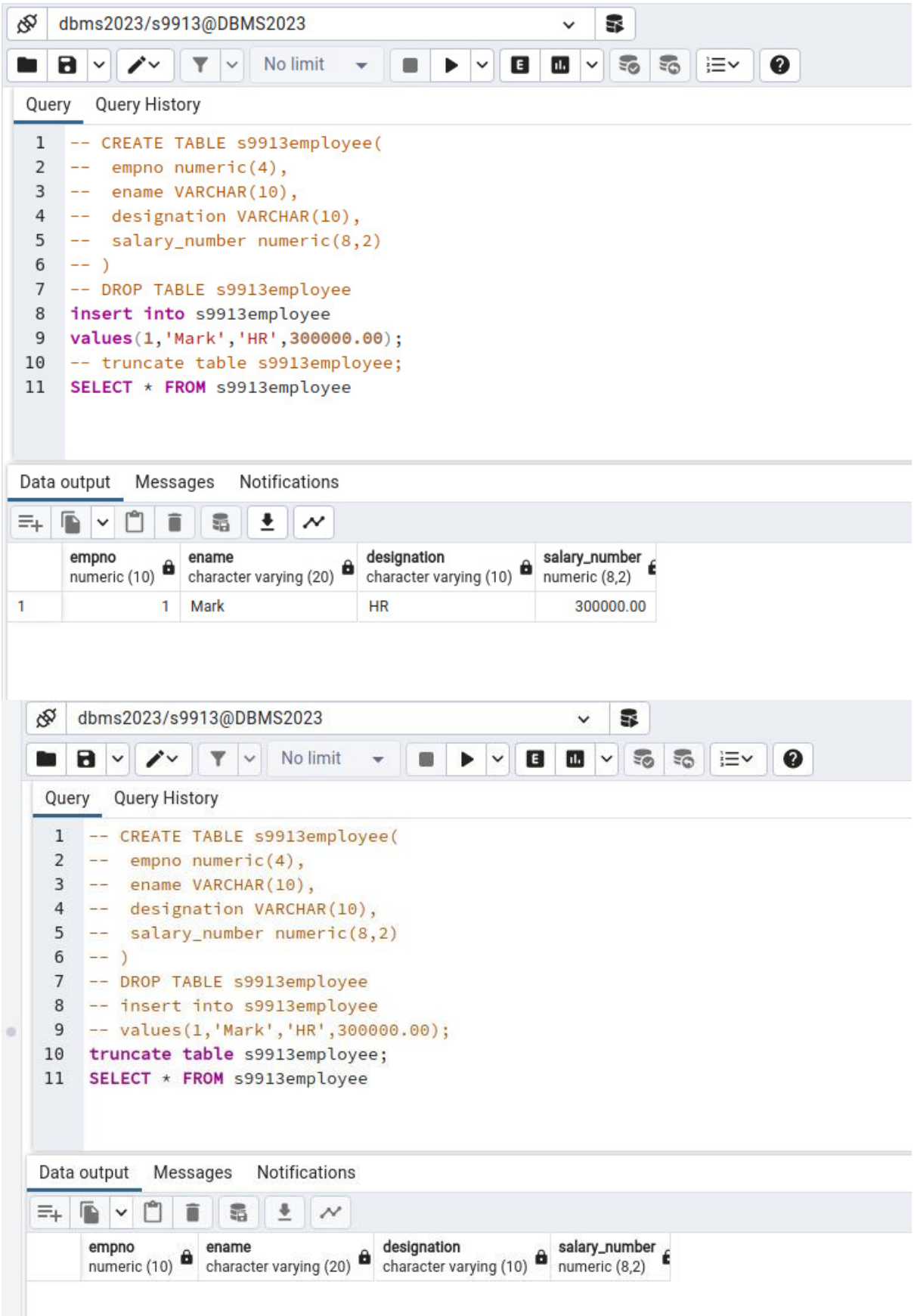
6


dbms2023/s9913@DBMS2023
No limit
Query
Query History
1 -- CREATE TABLE s9913employee(
2 -- empno numeric(4),
3 -- ename VARCHAR(10),
4 -- designation VARCHAR(10),
5 -- salary_number numeric(8,2)
6 --)
7 -- DROP TABLE s9913employee
8 ALTER TABLE s9913employee
9 drop column doj;
10 SELECT * FROM s9913employee
Data output
Messages
Notifications
empno numeric (10)
ename character varying (20)
designation character varying (10)
salary_number numeric (8,2)
qualification character varying (6)
dob date

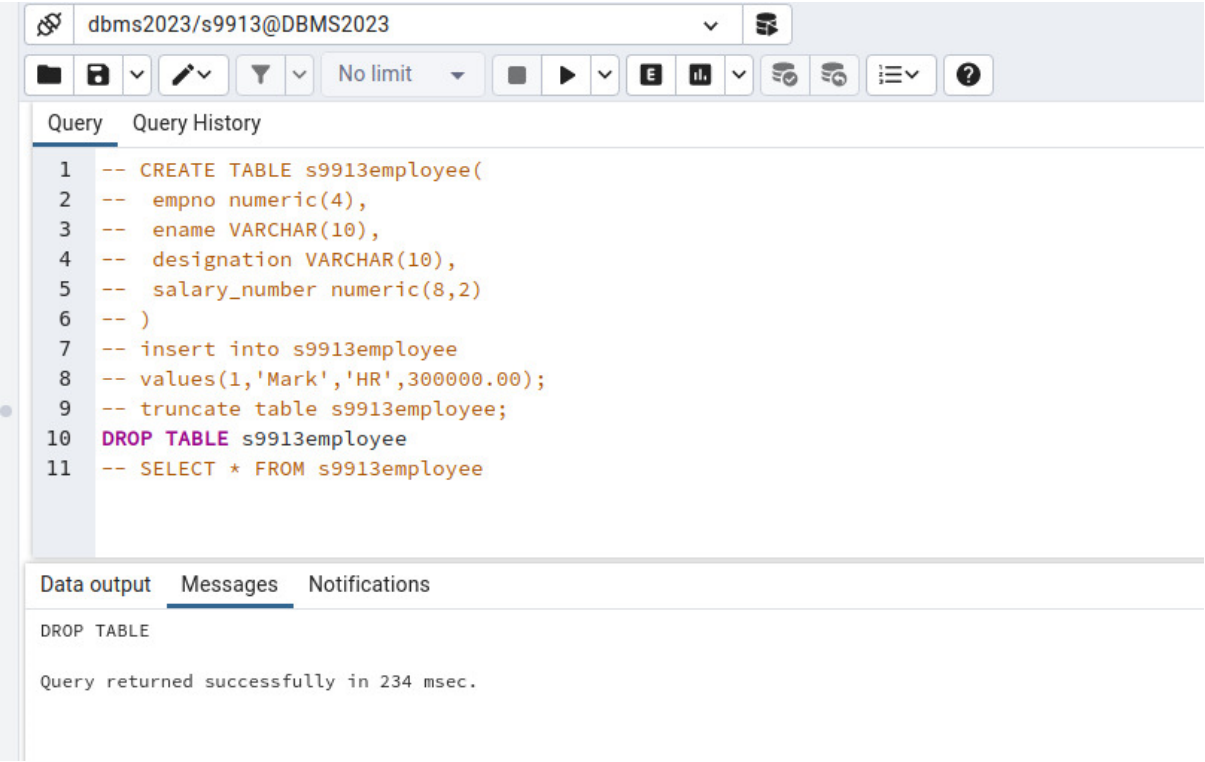
7


dbms2023/s9913@DBMS2023
No limit
Query
Query History
1 -- CREATE TABLE s9913employee(
2 -- empno numeric(4),
3 -- ename VARCHAR(10),
4 -- designation VARCHAR(10),
5 -- salary_number numeric(8,2)
6 --)
7 -- DROP TABLE s9913employee
8 ALTER TABLE s9913employee
9 drop column dob,
10 drop column qualification;
11 SELECT * FROM s9913employee
Data output
Messages
Notifications
empno numeric (10)
ename character varying (20)
designation character varying (10)
salary_number numeric (8,2)

8


dbms2023/s9913@DBMS2023
No limit
Query
Query History
1 -- CREATE TABLE s9913employee(
2 -- empno numeric(4),
3 -- ename VARCHAR(10),
4 -- designation VARCHAR(10),
5 -- salary_number numeric(8,2)
6 --)
7 -- DROP TABLE s9913employee
8 insert into s9913employee
9 values(1,'Mark','HR',300000.00);
10 -- truncate table s9913employee;
11 SELECT * FROM s9913employee
Data output
Messages
Notifications
empno numeric (10)
ename character varying (20)
designation character varying (10)
salary_number numeric (8,2)
1 1 Mark HR 300000.00
dbms2023/s9913@DBMS2023
No limit
Query
Query History
1 -- CREATE TABLE s9913employee(
2 -- empno numeric(4),
3 -- ename VARCHAR(10),
4 -- designation VARCHAR(10),
5 -- salary_number numeric(8,2)
6 --)
7 -- DROP TABLE s9913employee
8 -- insert into s9913employee
9 -- values(1,'Mark','HR',300000.00);
10 truncate table s9913employee;
11 SELECT * FROM s9913employee
Data output
Messages
Notifications
empno numeric (10)
ename character varying (20)
designation character varying (10)
salary_number numeric (8,2)

9



POSTLAB:-

Q1

A data dictionary is a centralized repository that stores metadata about a database, including definitions, data types, constraints, relationships, and other details. It serves as a reference guide for understanding and managing the structure and attributes of data within the database.

Q2

In a database, a schema is a logical container or namespace that holds a collection of database objects, including tables, views, indexes, and procedures. It provides a way to organize and manage database objects, allowing multiple users or applications to work independently within their designated schemas. Schemas help avoid naming conflicts and provide a structure for organizing and securing database elements.

Q3

Numeric Types:

INT, INTEGER: Integer.
SMALLINT: Small integer.
TINYINT: Very small integer.
BIGINT: Large integer.
DECIMAL(p, s), NUMERIC(p, s): Decimal number with a specified precision (p) and scale (s).
FLOAT: Floating-point number.
REAL: Real number.

Character/String Types:

CHAR(n): Fixed-length character string.
VARCHAR(n), VARCHAR(MAX): Variable-length character string with a maximum length of n characters or maximum allowed length.
TEXT: Variable-length character string with no specified maximum length.

Date and Time Types:

DATE: Date (year, month, day).
TIME: Time of day.
DATETIME, TIMESTAMP: Date and time.
INTERVAL: Time interval.

Boolean Type:

BOOLEAN, BOOL: Boolean