

COURSE NAME: DBMS

COURSE CODE:23AD2102R

**Topic: INTRODUCTION TO SQL, DATA
DEFINITION LANGUAGE (DDL)**

Session - 9

AIM OF THE SESSION



To familiarize students with the basic concept of SQL languages and Create table command in detail.

INSTRUCTIONAL OBJECTIVES



This Session is designed to:

1. Discuss the types of SQL languages.
2. Various command under different types of SQL Languages.
3. Introduction of Create table command with its syntax and examples.

LEARNING OUTCOMES



At the end of this session, you should be able to understand the basic commands of SQL and learn how to write queries with SQL commands.

TABLE

Table also called Relation

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Primary Key (points to CustomerID)

Domain
Ex: NOT NULL (points to CustomerName)

Tuple OR Row (points to the three data rows)

Total # of rows is **Cardinality**

Column OR Attributes (points to the three column headers)

Total # of column is **Degree**

KEY

A DBMS Key is an attribute or set of attributes which helps us to identify a row (tuple) in a relation(table). It allows us to find the relation between two tables.

There are different types of keys:

1. Super Key
2. Unique Key
3. Primary Key
4. Candidate Key
5. Alternate Key
6. Foreign Key

SUPER KEY CANDIDATE KEY & NON-PRIME ATTRIBUTE

- **Super Key:** A super key is a set of one or more attributes(columns), which can uniquely identify a row in a table.
- **Candidate Key:** A super key with no redundant attribute is known as candidate key.
- **Non-prime Attribute:** An attribute that is not part of any candidate key is known as non-prime attribute.

Emp_Id	Emp_Number	Emp_Name
E01	2264	Steve
E02	2278	John
E03	2288	Martin
E04	2297	Robert

Super Key:

1. {Emp_Id}
2. {Emp_Number}
3. {Emp_Id} {Emp_Number}
4. {Emp_Number} {Emp_Name}
5. {Emp_Id} {Emp_Name}
6. {Emp_Id} {Emp_Number} {Emp_Name}

Candidate Key:

1. {Emp_Id}
2. {Emp_Number}

Non-prime attribute

{Emp_Name}

10

UNIQUE KEY & PRIMARY KEY

- A **Unique key** uniquely identifies each record in the database. This provides uniqueness for the column or set of columns and it can accept only one null value. Unique key is a subset of super key.
- A **primary key** is a combination of fields which uniquely specify a row. This is a special kind of unique key, and it has implicit NOT NULL constraint. It means, Primary key values cannot be NULL.
 - A Primary key constraint has automatic unique constraint defined on it. But not, in the case of Unique Key.
 - Primary Key can be chosen from Candidate key.
 - There can be many unique constraint defined per table, but only one primary key constraint defined per table.

ALTERNATE KEY FOREIGN KEY

A candidate key, that is not a primary key is called an **Alternate key**.

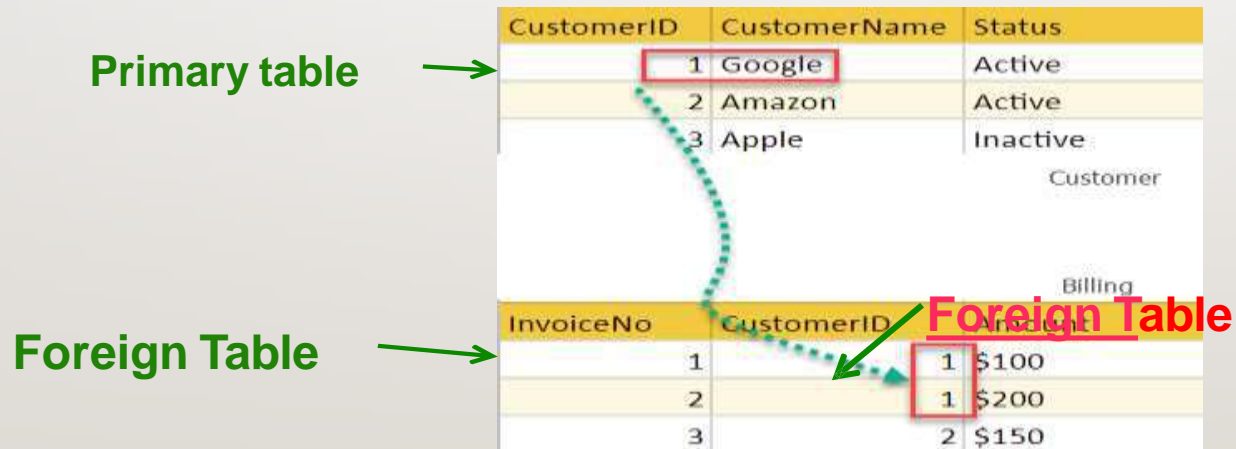
A non key attribute(or a group of non-key attributes) whose values are derived from primary key of some other table is known as **Foreign key** in its current table.

- Foreign key is used to represent the relationship between two tables.
- Foreign key of a table is a primary key of some other table.

PRIMARY TABLE & FOREIGN TABLE

- The table in which foreign key attribute exists, is called a **Foreign Table or Detail table.**
- The table that defines the primary key, which is the foreign key of the detail table or Foreign table refers to, is called **Primary table or Master table.**

↙ Primary key



SQL

Structured Query Language

- **SQL** is a language which is used to create & operate database.
- **SQL** is the basic language used for all the databases.
- **SQL** can be used by both casual users as well as skilled programmer.

There are minor syntax changes amongst different databases, but the basic **SQL** syntax remains largely the same.

- According to ANSI (American National Standards Institute), **SQL** is the standard language to operate a relational database management system.
- **SQL** is used in the accessing, updating, and manipulation of data in a database. Its design allows for the management of data in an RDBMS, such as MySQL and PostgreSQL.

PROCESSING CAPABILITIES OF SQL

1. DATA DEFINITION LANGUAGE(DDL)

The SQL DDL provides commands for defining relation schemas, deleting relations, creating indexes and modifying relation schemas.

2.INTERACTIVE DATA MANIPULATION LANGUAGE(DML)

The SQL DML includes a query language based on both relational algebra and the tuple relational calculus. It includes the commands to insert, delete and modify tuples in the data base.

PROCESSING CAPABILITIES OF SQL

3. Embedded SQL

It is a method of combining the computing power of a programming language and the database manipulation capabilities of SQL.

4. View Definition

The SQL DDL also includes commands for defining views. (Views are the virtual tables that do not really exist in its own but is derived from one or more base table(s)).

5. Authorization

The SQL DDL also includes commands for specifying access rights to relation and views.

PROCESSING CAPABILITIES OF SQL

6.Integrity

The SQL provides limited forms of integrity checking. Future products and standards of SQL are likely to include enhanced features for integrity checking.

7.Transaction Control

SQL includes commands for specifying the beginning and ending of transactions along with the commands to have a control over transaction processing.

Data Definition Language

- Data Definition Language actually consists of the SQL commands that can be used to define the database structure or schema.
- It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.
- DDL is a set of SQL commands used to create, modify, and delete database structures but not data.
- These commands are normally not used by a general user, who should be accessing the database via an application.

List of DDL commands

- CREATE: This command is used to create the new database or its objects.
- ALTER: This is used to alter the structure of the database.
- DROP: This command is used to delete objects from the database.
- TRUNCATE: This is used to remove all records from a table, including all spaces allocated for the records are removed.
- RENAME: This is used to rename an object existing in the database.

1) CREATE TABLE command: This DDL command allows us to create the new table.

```
CREATE TABLE [IF NOT EXISTS] <table_name> ( <column1> <data_type(length)>
[column_constraint], <column2> <data_type(length)> [column_constraint], ...
<columnN> <data_type(length)> [column_constraint], [table_constraints] );
```

In the above syntax,

- After CREATE TABLE we can specify optional IF NOT EXISTS clause, which will create a table only if it does not exist. If a table already exists, Postgres will only give a warning instead of an error and skip creating a new table.
- The table_name is the unique name of a table which you want to create.
- In the bracket, specify the list of column names along with their data types. Specify the optional [column_constraint] such as NOT NULL, Check, Unique, Primary Key, Foreign Key etc.

DATA DEFINITION LANGUAGE (Cont..)

```
CREATE TABLE IF NOT EXISTS person ( Id INT PRIMARY KEY, first_name VARCHAR(50) NOT NULL, last_name VARCHAR(50) NOT NULL, gender CHAR(1), birthdate DATE, email_id VARCHAR(100) UNIQUE, country_of_birth VARCHAR(50) );
```



The screenshot shows a database management tool interface. At the top, the connection is labeled 'MyDatabase/postgres@PostgreSQL 14'. Below the connection bar is a toolbar with various icons for file operations, query execution, and settings. The main area is divided into two tabs: 'Query' and 'Query History'. The 'Query' tab is active, displaying the following SQL statement:

```
1 CREATE TABLE IF NOT EXISTS person ( Id INT PRIMARY KEY,  
2 first_name VARCHAR(50) NOT NULL,  
3 last_name VARCHAR(50) NOT NULL,  
4 gender CHAR(1),  
5 birthdate DATE,  
6 email_id VARCHAR(100) UNIQUE,  
7 country_of_birth VARCHAR(50) );  
8
```

Below the query editor, there are three tabs: 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing the following output:

```
CREATE TABLE  
  
Query returned successfully in 68 msec.
```


2) ALTER TABLE command: This DDL command allows to alter the structure of the database. The commands are:

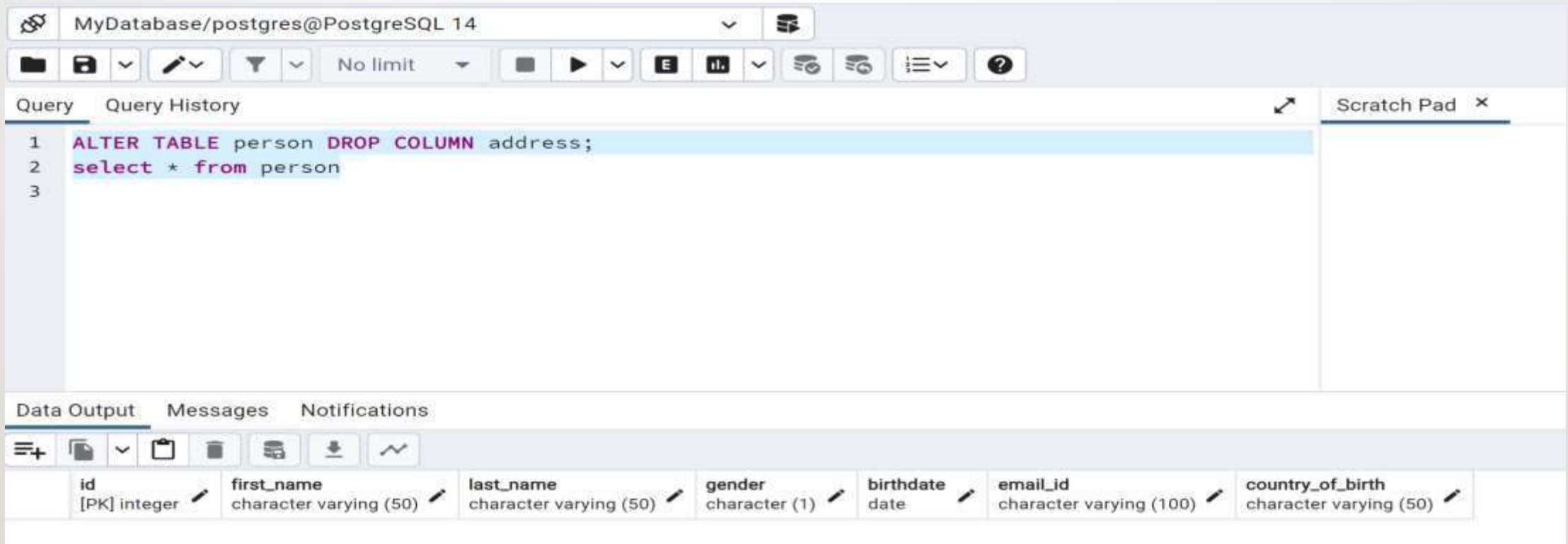
A) ADD COLUMN: It is used to add a new column in the table. For example, to add a column address of type varchar to an existing table:

```
ALTER TABLE person ADD COLUMN address varchar(50);
```



B) DROP COLUMN: It is used to drop an existing column from the table.

```
ALTER TABLE person DROP COLUMN address;
```



The screenshot shows a PostgreSQL database management tool interface. The top bar displays the connection name "MyDatabase/postgres@PostgreSQL 14". Below the connection bar is a toolbar with various icons for file operations, query execution, and settings. The main window is divided into two panes: "Query" and "Query History". The "Query" pane contains the following SQL code:

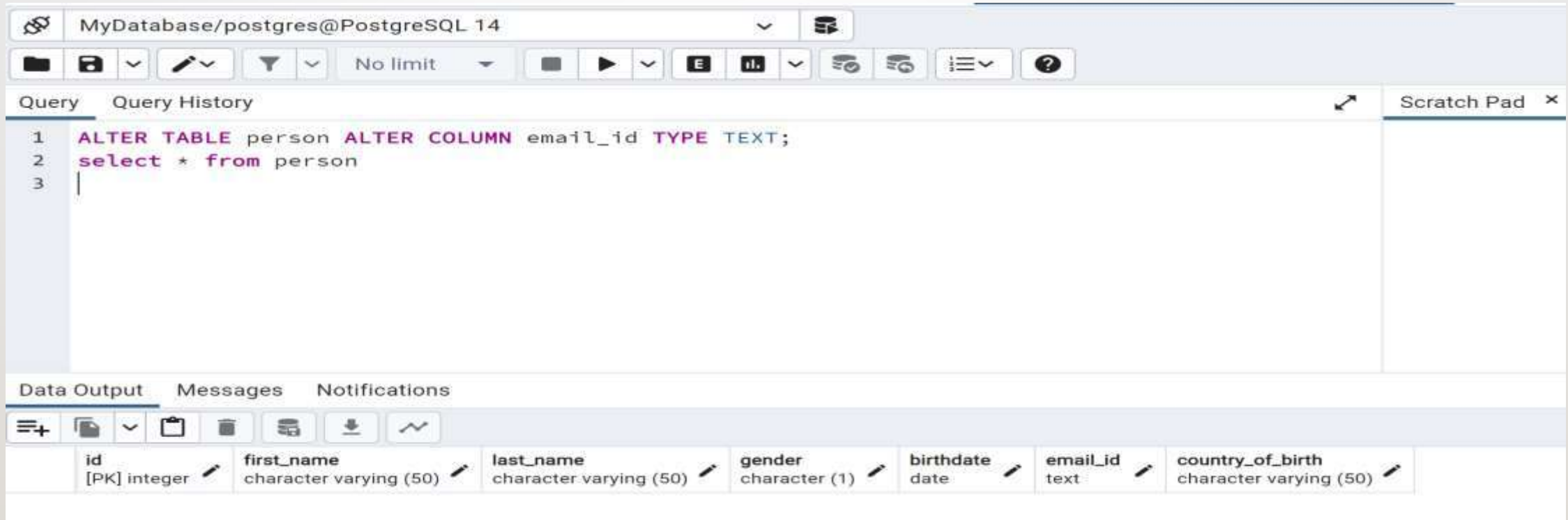
```
1 ALTER TABLE person DROP COLUMN address;
2 select * from person
3
```

The "Query History" pane is empty. To the right of the query panes is a "Scratch Pad" tab. Below the query panes is a "Data Output" tab, which is currently active. It displays the schema of the "person" table:

id	first_name	last_name	gender	birthdate	email_id	country_of_birth
[PK] integer	character varying (50)	character varying (50)	character (1)	date	character varying (100)	character varying (50)

C) ALTER COLUMN: It is used to alter the existing column. To change the datatype of email from VARCHAR to TEXT.

```
ALTER TABLE person ALTER COLUMN email_id TYPE TEXT;
```



The screenshot shows a PostgreSQL database management tool interface. The top bar indicates the connection is to 'MyDatabase/postgres@PostgreSQL 14'. Below the connection bar is a toolbar with various icons for file operations, query execution, and navigation. The main area is divided into two tabs: 'Query' and 'Query History'. The 'Query' tab is active, showing a SQL query editor with the following text:

```
1 ALTER TABLE person ALTER COLUMN email_id TYPE TEXT;
2 select * from person
3 |
```

Below the query editor is a 'Scratch Pad' tab. At the bottom of the interface, there are three tabs: 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, displaying a table schema for the 'person' table. The schema is as follows:

id	first_name	last_name	gender	birthdate	email_id	country_of_birth
[PK] integer	character varying (50)	character varying (50)	character (1)	date	text	character varying (50)

To add a NOT NULL constraint to a column:

```
ALTER TABLE person ALTER COLUMN gender SET NOT NULL;
```

To remove a NOT NULL constraint from a column:

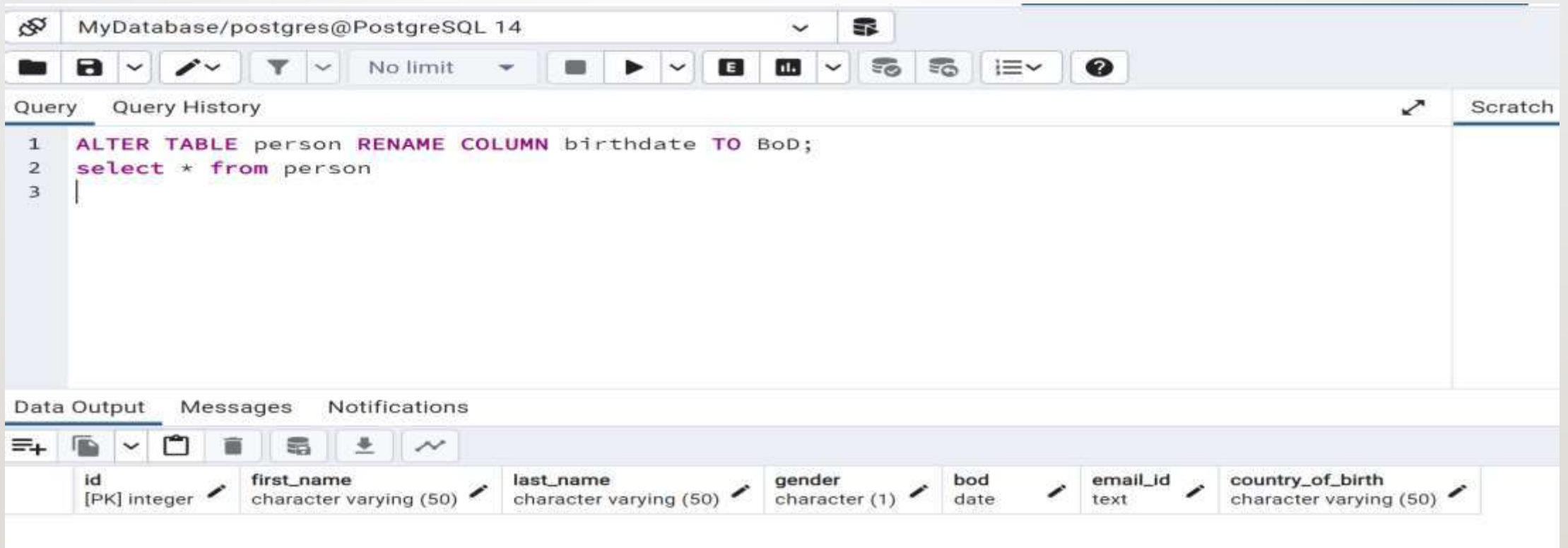
```
ALTER TABLE person ALTER COLUMN gender DROP NOT NULL;
```

To add a PRIMARY KEY constraint to a table:

```
ALTER TABLE person ADD PRIMARY KEY (id);
```

D) RENAME COLUMN: It is used to change the name of a column.

```
ALTER TABLE person RENAME COLUMN birthdate TO BoD;
```



The screenshot shows a PostgreSQL query editor interface. The top toolbar includes icons for connecting to a database, saving, editing, filtering, and executing queries. The main query area contains the following SQL code:

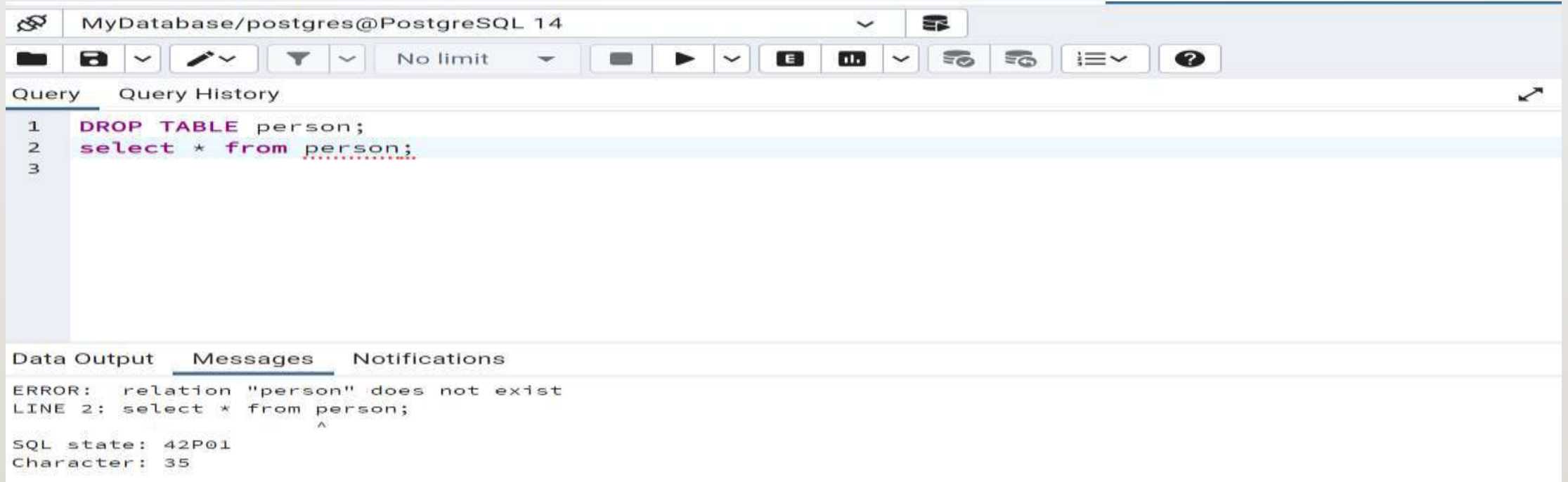
```
1 ALTER TABLE person RENAME COLUMN birthdate TO BoD;  
2 select * from person  
3 |
```

Below the query area, there are tabs for "Data Output", "Messages", and "Notifications". The "Data Output" tab is active, displaying a table with the following columns and data types:

id	first_name	last_name	gender	bod	email_id	country_of_birth
[PK] integer	character varying (50)	character varying (50)	character (1)	date	text	character varying (50)

3) DROP command: This DDL command allows us to delete the database objects. With the help of the “DROP” command we can drop (delete) the whole structure in one go.

```
DROP TABLE person;
```



The screenshot shows a PostgreSQL query editor interface. The title bar indicates the connection is 'MyDatabase/postgres@PostgreSQL 14'. The toolbar includes icons for saving, undo, redo, and other standard database operations. The 'Query' tab is active, showing a SQL script with two lines: 'DROP TABLE person;' and 'select * from person;'. The second line is highlighted in blue. Below the query editor, the 'Messages' tab is active, displaying an error message: 'ERROR: relation "person" does not exist LINE 2: select * from person;'. The error message is followed by the SQL state '42P01' and the character position '35'.

```
1 DROP TABLE person;  
2 select * from person;  
3
```

ERROR: relation "person" does not exist
LINE 2: select * from person;
SQL state: 42P01
Character: 35

4) TRUNCATE command: It is used to delete all the data from the table. With the help of the “TRUNCATE” command, we can’t delete the single row as here WHERE clause is not used.

Syntax of TRUNCATE TABLE command: **TRUNCATE TABLE** table_name;

5) ALTER TABLE RENAME command: This DDL command allows the users to change the name of the existing table. For example, To rename from person to employee, use the following ALTER TABLE statement:

```
ALTER TABLE person RENAME TO employee;
```


SUMMARY

1. In this section, we discussed the various types of SQL languages and commands of these languages that are used to work on the database and the data stored in the databases.
2. We also discussed the CREATE TABLE command in detail with its syntax and examples for creating the tables.

SELF-ASSESSMENT QUESTIONS

1. Commands that comes under DDL is/are –

- (a) CREATE
- (b) DROP
- (c) TRUCATE
- (d) ALLOF THE ABOVE

2. Command that comes under DML is/are –

- (a) ROLLBACK
- (b) GRANT
- (c) UPDATE
- (d) ALL OF THE ABOVE

SELF-ASSESSMENT QUESTIONS

3. Command that comes under DCL is/are -

- (a) GRANT
- (b) REVOKE
- (c) BOTH (a) AND (b)
- (d) NONE OF THE ABOVE

4. Following the completion of a transaction, it must be executed to save all the operations performed in the transaction. Here we are talking about which command?

- (a) REVOKE
- (b) COMMIT
- (c) ROLLBACK
- (d) SAVE

1. Describe various types of SQL Languages.
2. List out the commands of Data Definition Language with examples.
3. Analyze the DDL commands in PostgreSQL.
4. Summarize the create table command with its syntax and examples.

Reference Books:

1. Database System Concepts, Sixth Edition, Abraham Silberschatz, Yale University Henry, F. Korth
Lehigh University, S. Sudarshan Indian Institute of Technology, Bombay.
2. An Introduction to Database Systems by Bipin C. Desai
3. Fundamentals of Database Systems, 7th Edition, RamezElmasri, University of Texas at Arlington,
Shamkant B. Navathe, University of Texasat Arlington.

Sites and Web links:

1. <https://www.geeksforgeeks.org/postgresql-create-table/>
2. <https://www.tutorialsteacher.com/postgresql>

THANK YOU



Team – DBMS