POSTGRESQL CLASS 3

BASIC CRUD STATEMENTS

INTRO

- SQL input consists of a sequence of *commands*. A command is composed of a sequence of *tokens*, terminated by a semicolon (“;”). The end of the input stream also terminates a command. Which tokens are valid depends on the syntax of the particular command.
- A token can be a *key word*, an *identifier*, a *quoted identifier*, a *literal* (or constant), or a special character symbol. Tokens are normally separated by whitespace (space, tab, newline), but need not be if there is no ambiguity (which is generally only the case if a special character is adjacent to some other token type).

INTRO

- Tokens such as SELECT, UPDATE OR VALUES are examples of keywords. These are words that have a fixed meaning in the SQL language. They are usually written in UPPERCASE.

IDENTITY

- In version 10 of PostgreSQL, it has introduced a new feature named GENERATED AS IDENTITY constraint. The GENERATED AS IDENTITY constraint allows user to assign a unique value to the column automatically.
- data_type: The data type can be any of the following:
- SMALLINT,
- INT,
- or BIGINT.

IDENTITY

- **Generated Always:**
- If we have GENERATED ALWAYS defined which means the PostgreSQL has to always create new value for the identity column. In case of the column with GENERATED ALWAYS constraint if we try to insert or update a value for same then PostgreSQL throws an error or exception.
- **Generated By Default:**
- If we have GENERATED BY DEFAULT defined means the PostgreSQL will create a new value for the PostgreSQL identity column. In case of the column with GENERATED BY DEFAULT constraint if we try to insert or update a value for same then PostgreSQL uses same value and do not use any system-generated value also PostgreSQL does not throw any error or exception

ENUMS

- create type GENDER as enum(MALE', ‘FEMALE’);
- create table cars (id INT, sex GENDER);

CRUD OPERATIONS

- The basic CRUD operations are as follows
- CREATE - This is the INSERT INTO statement.
- READ – This is the SELECT statement
- UPDATE – This is the UPDATE statement
- DELETE – this is the DELETE FROM statement.

CREATE DATABASE

- CREATE DATABASE db_name;
- This cannot be executed inside a transaction block.
-

CREATE TABLE

- CREATE TABLE tbl_name(id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY, date TIMESTAMP DEFAULT CURRENT_TIMESTAMP);

INSERT INTO STATEMENT

- INSERT INT0 tbl_name(ID,name,date) VALUES(1,'Andrew','2022-05-22');

SELECT STATEMENT

- SELECT * FROM tbl_name;
- SELECT column_name_1, column_name_2 FROM tbl_name;

UPDATE STATEMENT

- UPDATE tbl_name SET column_name_1 = 1;

DELETE FROM STATEMENT

- DELETE FROM tbl_name WHERE id = 1;
- DELETE FROM tbl_name;

DROP STATEMENT

- DROP TABLE [IF EXISTS] name_of_the_table [CASCADE | RESTRICT];
- **IF EXISTS–** It is an optional parameter. This keyword can be used to prevent any error which is resulted while using the above command when there is no table that exists in the current database having the name “name_of_the_table”. If the “IF EXISTS” keyword is used in your query and none of the table named “name_of_the_table” is present in the current server then it will simply notify you about the same and won’t result in any error.
- **name_of_the_table–** It is the name of the table that you wish to drop and is located in the current database.
- **CASCADE–** This is used when a foreign key constraint comes into the picture. When the target table has any foreign constraints over other table or the target table is used in any Views then CASCADE property needs to be mentioned for handling the same. When CASCADE is used then all the related views to the target table are also deleted permanently while in the case of foreign key constraints the related table is not affected. All its contents remain intact in the table.
- **RESTRICT–** This is the default value in the DROP TABLE query. When this is specified or none out of CASCADE or RESTRICT are specified in the query statement then dropping the table is not allowed if any foreign constraint exists on any other table or the target table is used in any other views.

TRUNCATE TABLE STATEMENT

- TRUNCATE TABLE tbl_name

ALTER TABLE

- ALTER TABLE persons ADD phone_no VARCHAR(25);
- ALTER TABLE persons DROP phone_no VARCHAR(25);
- ALTER TABLE persons ALTER COLUMN phone_no VARCHAR(25);
- ALTER TABLE persons ADD {CONSTRAINT} (id);
- ALTER TABLE persons RENAME COLUMN date TO date_entry;