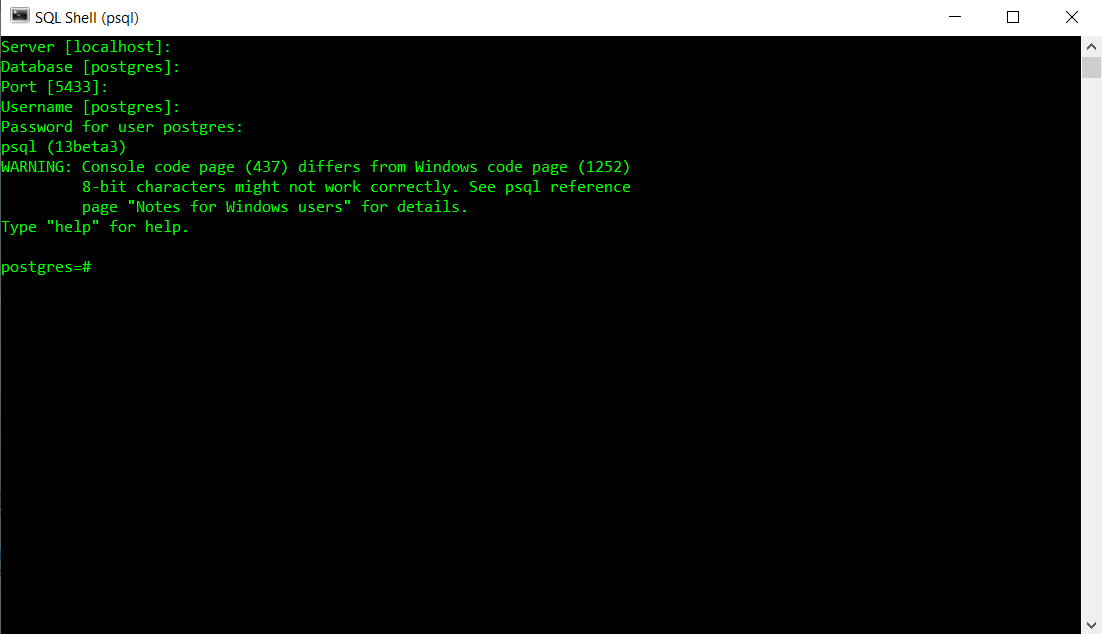
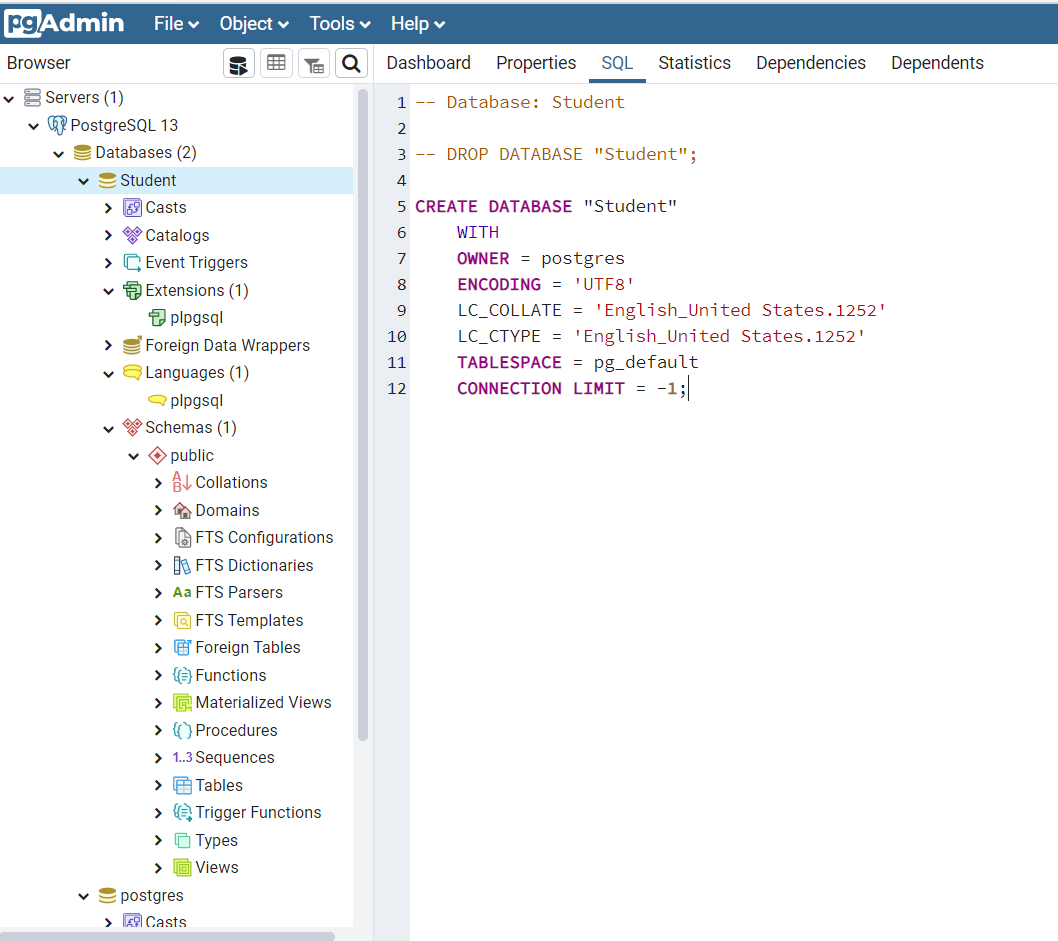
KloudOne Assignment 20-08-2020

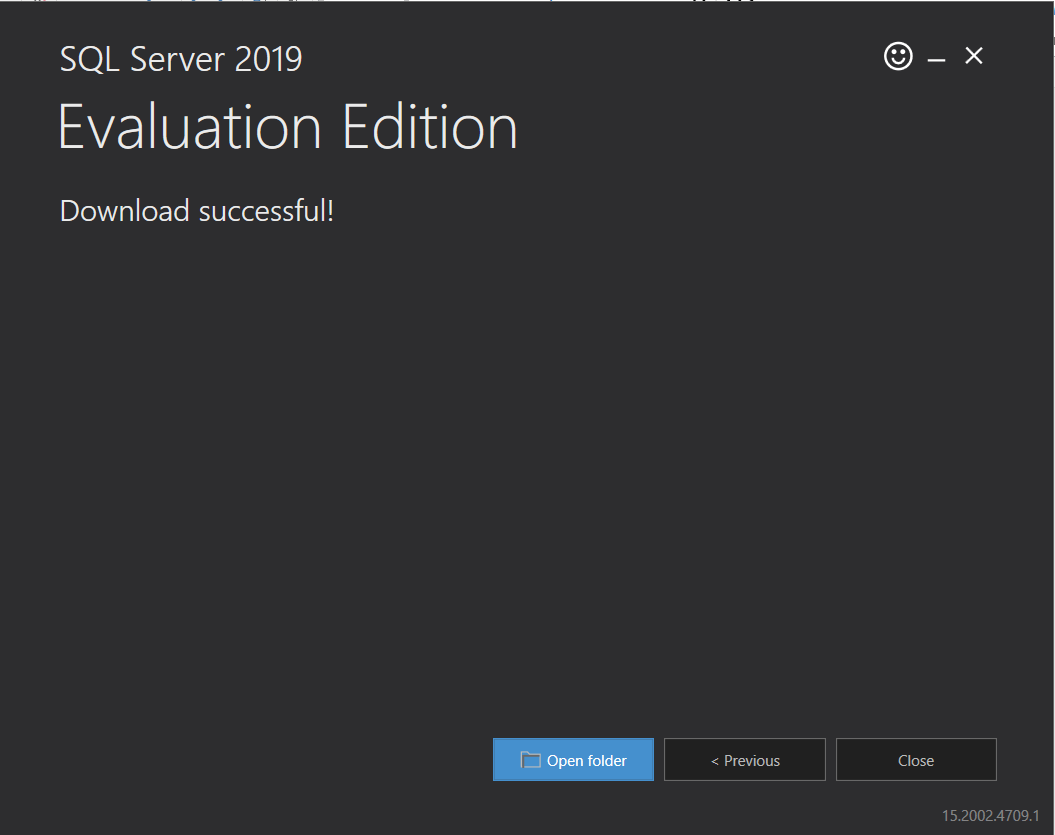
Installed PostgreSQL in windows 10

SQL Shell:



PgAdmin4:

Microsoft SQL installation:



DATA DEFINITION LANGUAGE:

It is used to define the structure of tables in the database. It contains the necessary statements for creating, manipulating, altering and deleting the tables.

DATA MANIPULATION LANGUAGE:

It is used to manipulate the data in the database. It contains statements to update, delete, insert and select data, stored in the database.

DDL(Data Definition Language) Commands in PostgreSQL:

1. Create Table

A table or relation can be created in a database by CREATE TABLE statement of SQL.

CREATE TABLE<table name>

( <attribute name> <datatype> [size] [constraint],

<attribute name> <datatype> [size] [constraint],

…..

);

Constraints are for example:

Type Description

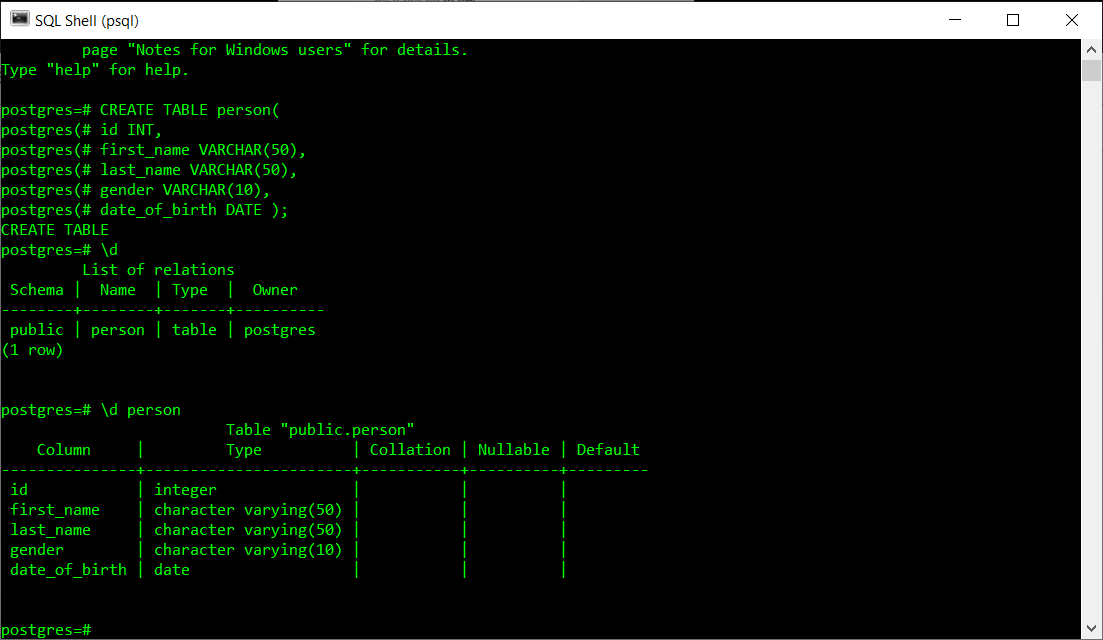
NULL/NOT NULL Specifies, if a column can or cannot have NULL values.

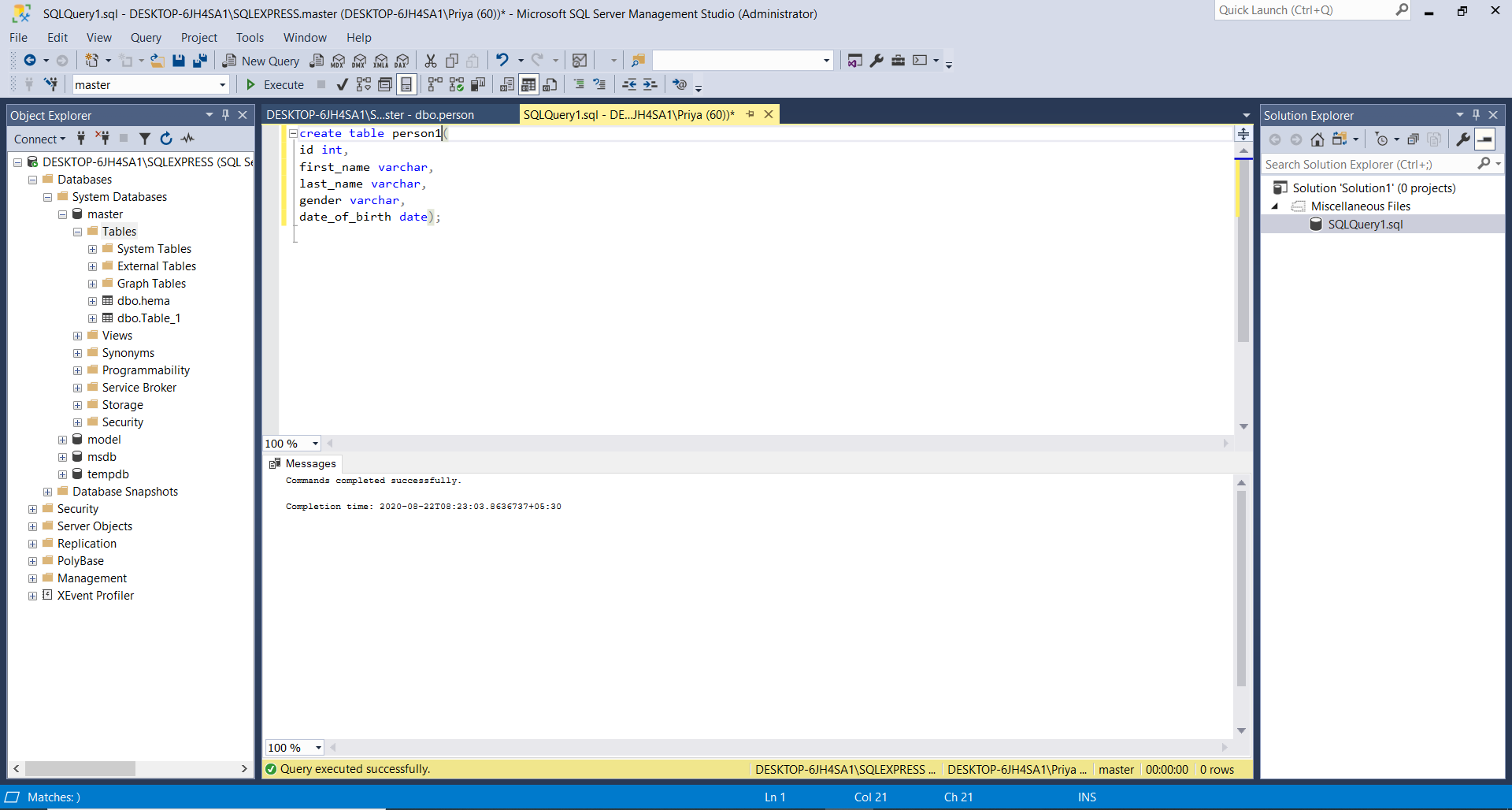
UNIQUE Value of a column have to be unique.

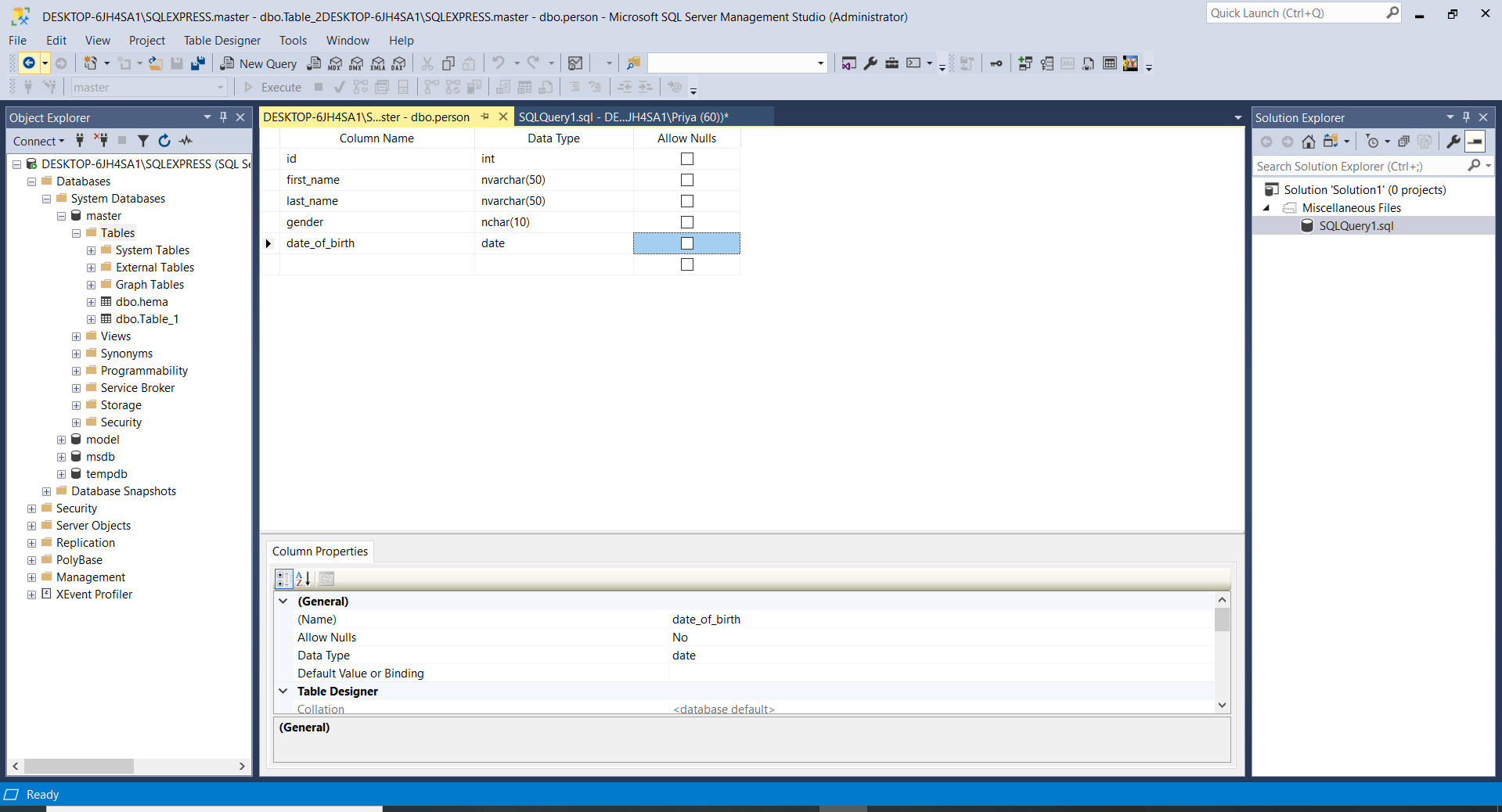
PRIMARY KEY Defines it to be a primary key.

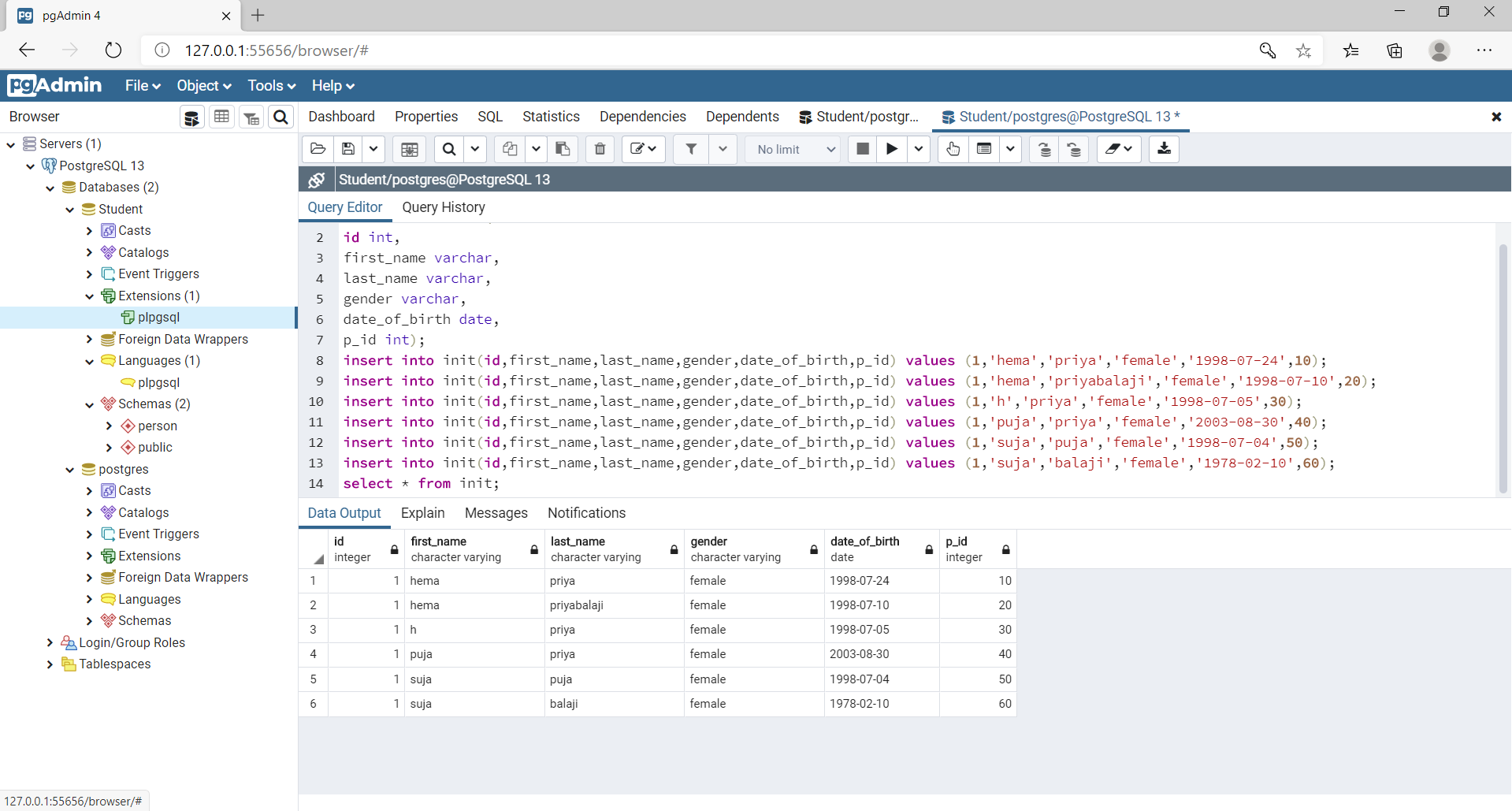
DEFAULT It prevents null values in a row.

CHECK Explicitly defines a condition that each row must satisfy.









1. Alter Table

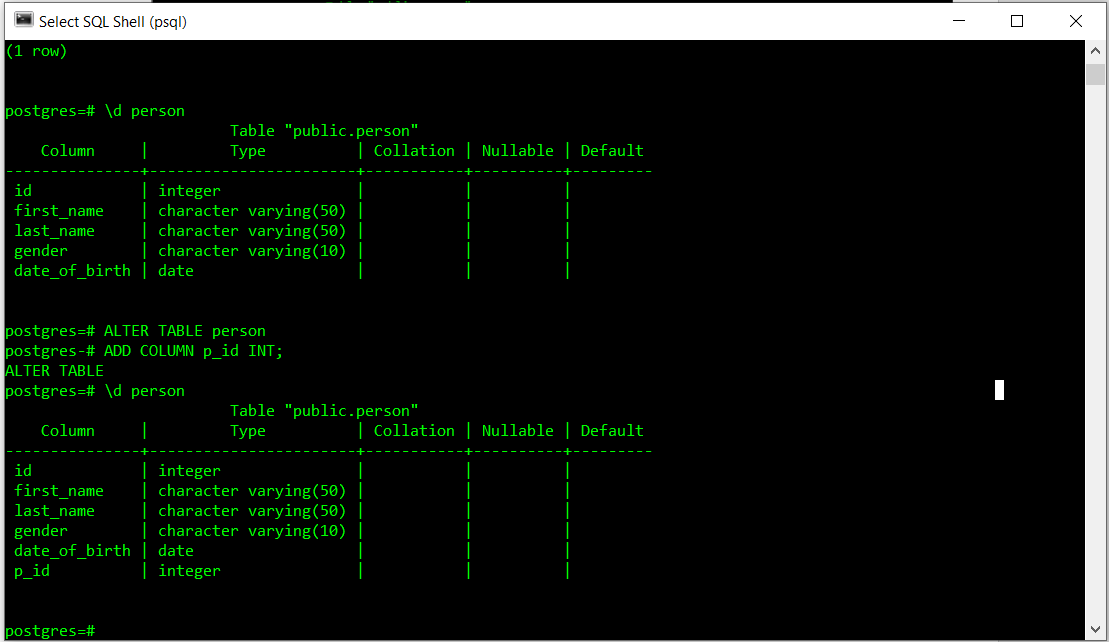
This command is used to change the table structure.

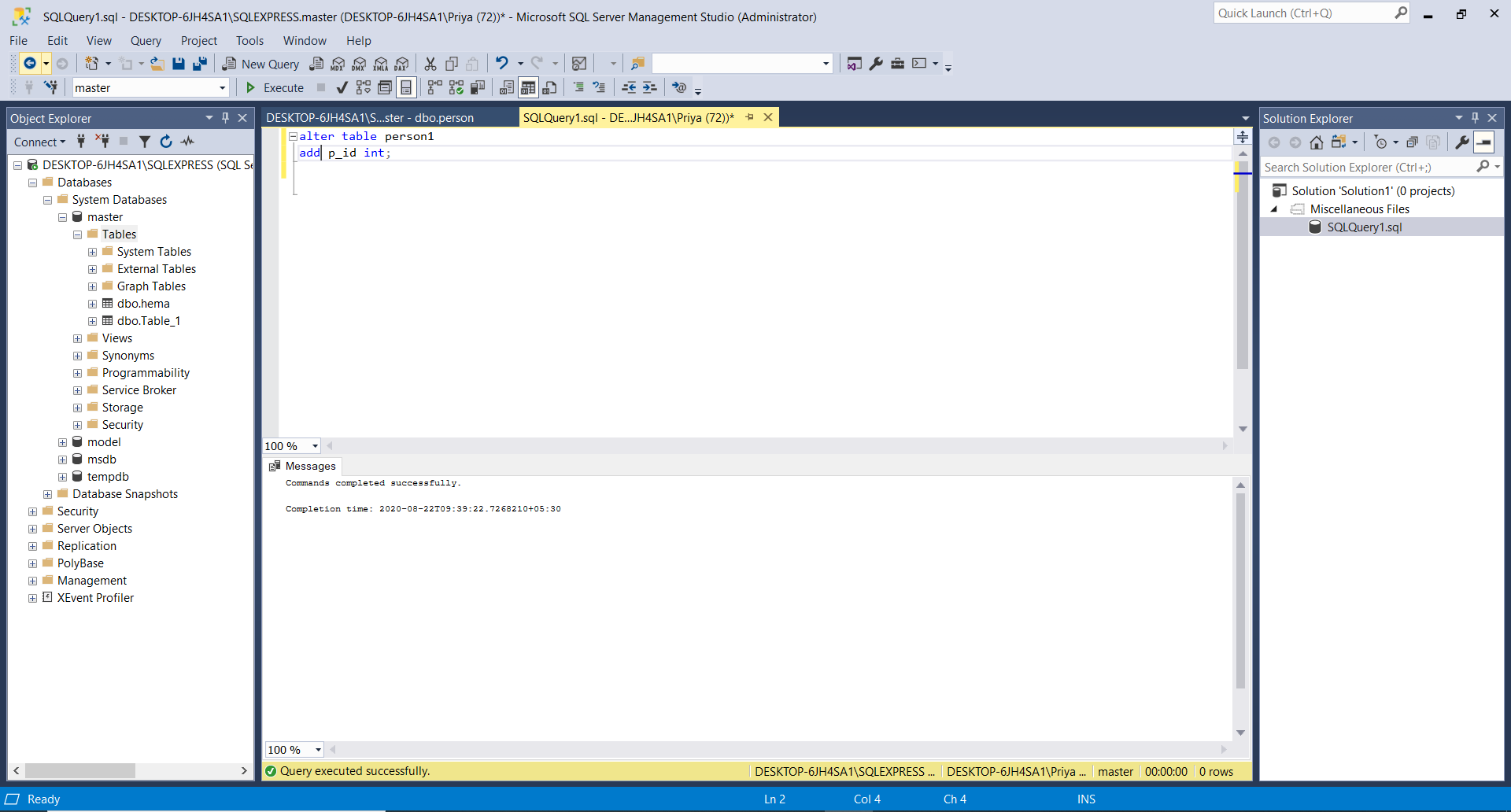
ADD To add new column

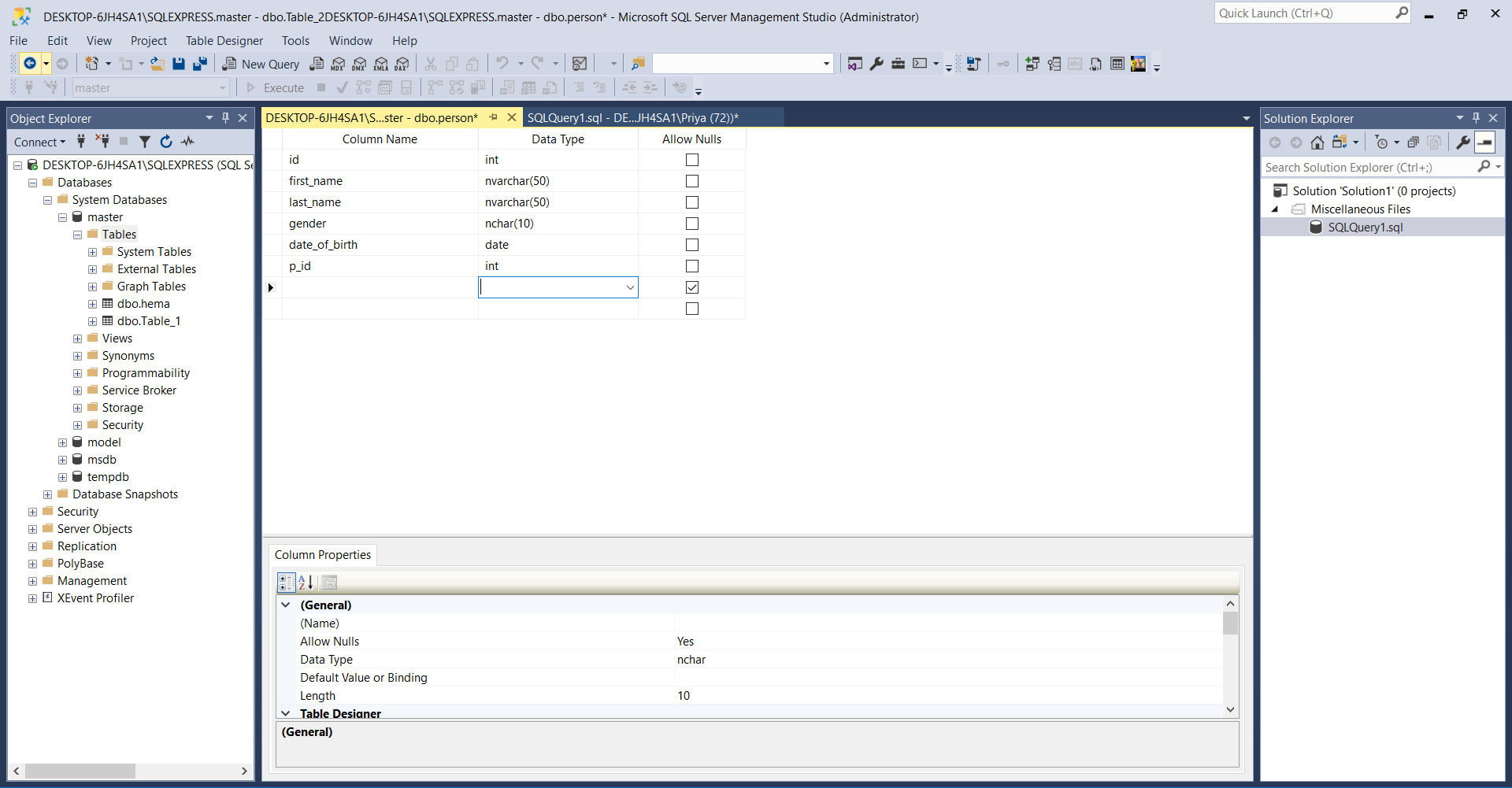
MODIFY To change the definition of an existing column

ALTER TABLE <table name>

ADD or MODIFY or DROP COLUMN <attribute name> <data type>;



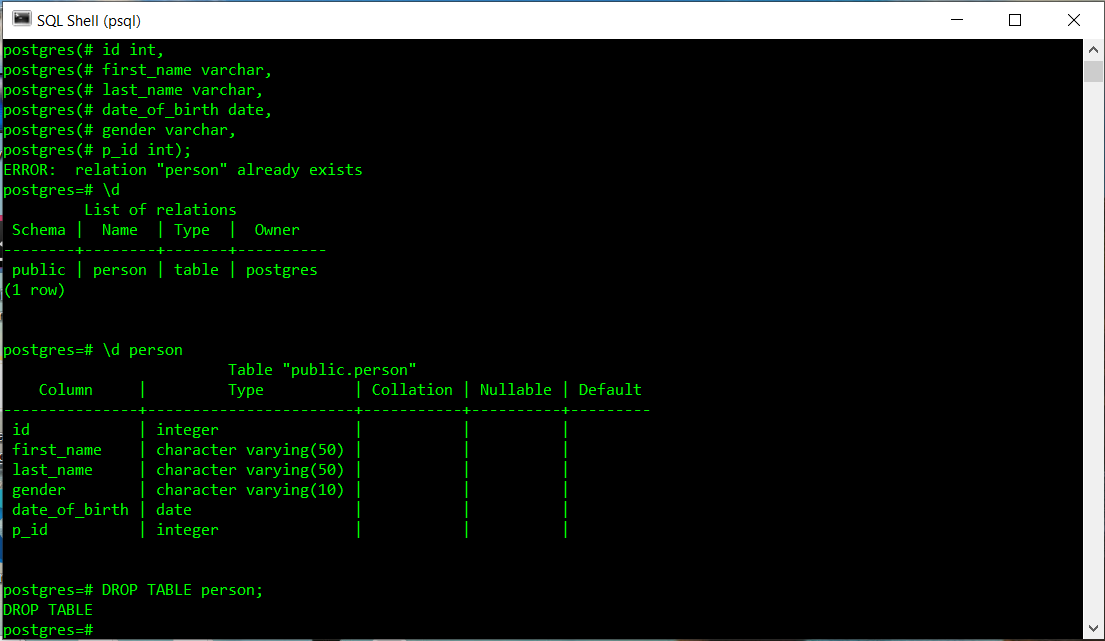




1. Drop Table

To remove the table physically, we use this command.

DROP TABLE <table name>;

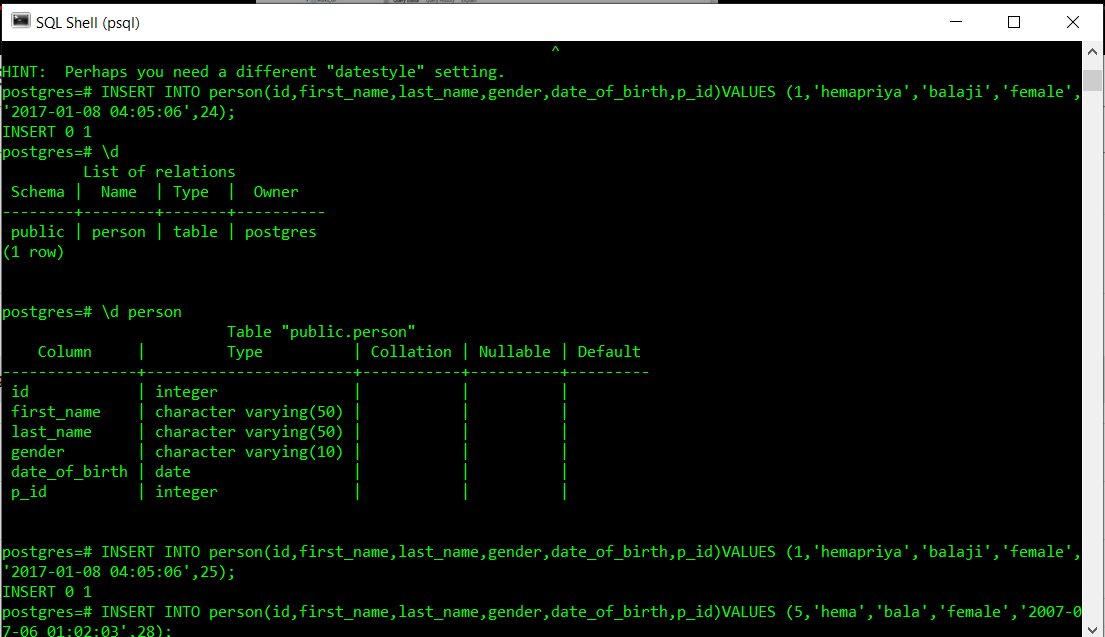


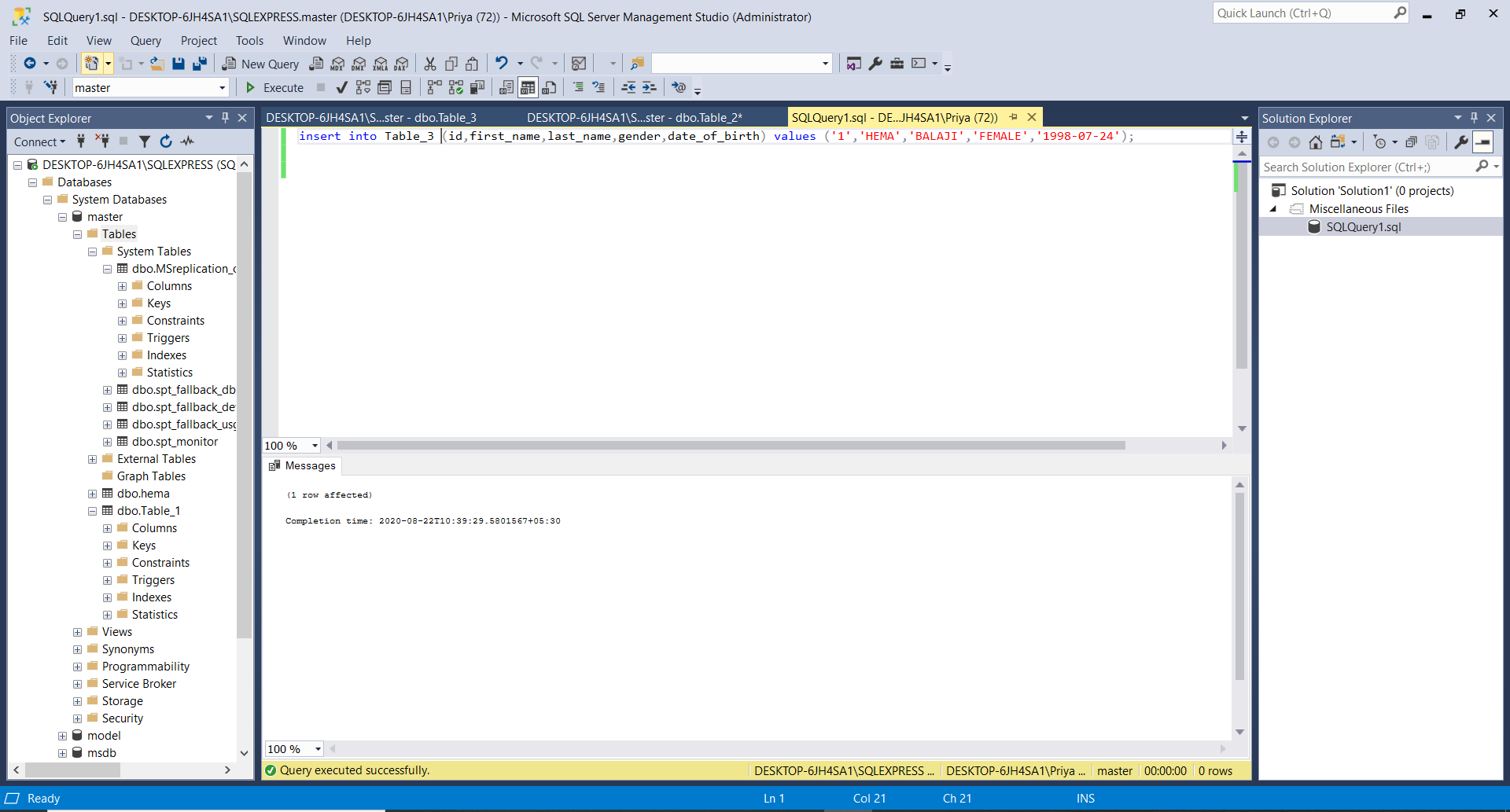
DML(Data Manipulation Language) Commands in PostgreSQL

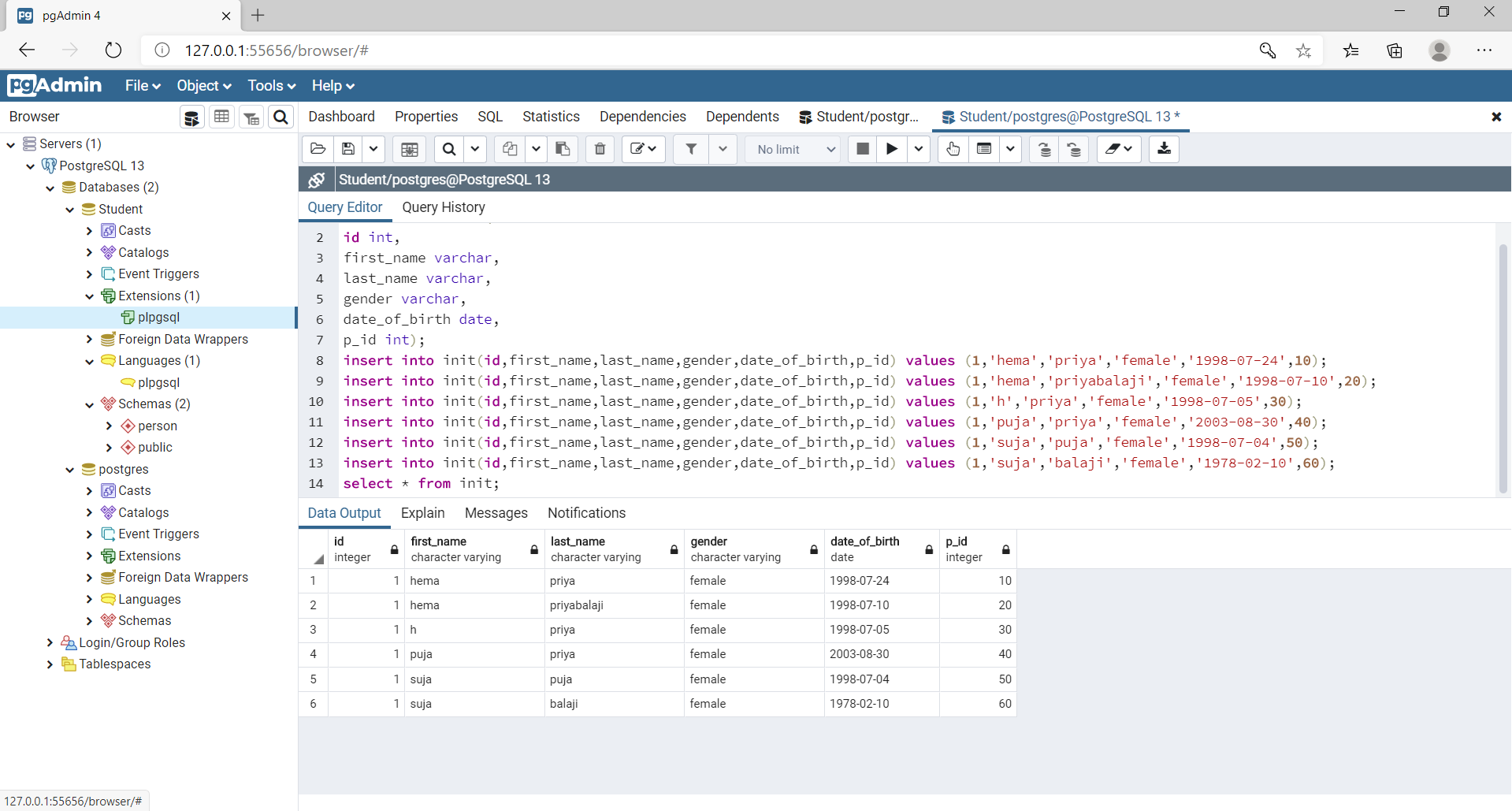
1. INSERT INTO

To add new records to the end of an existing database.

INSERT INTO <table name> (<column name1>,<coloumn name2>, …) VALUES (<value1>,<value2>, …);





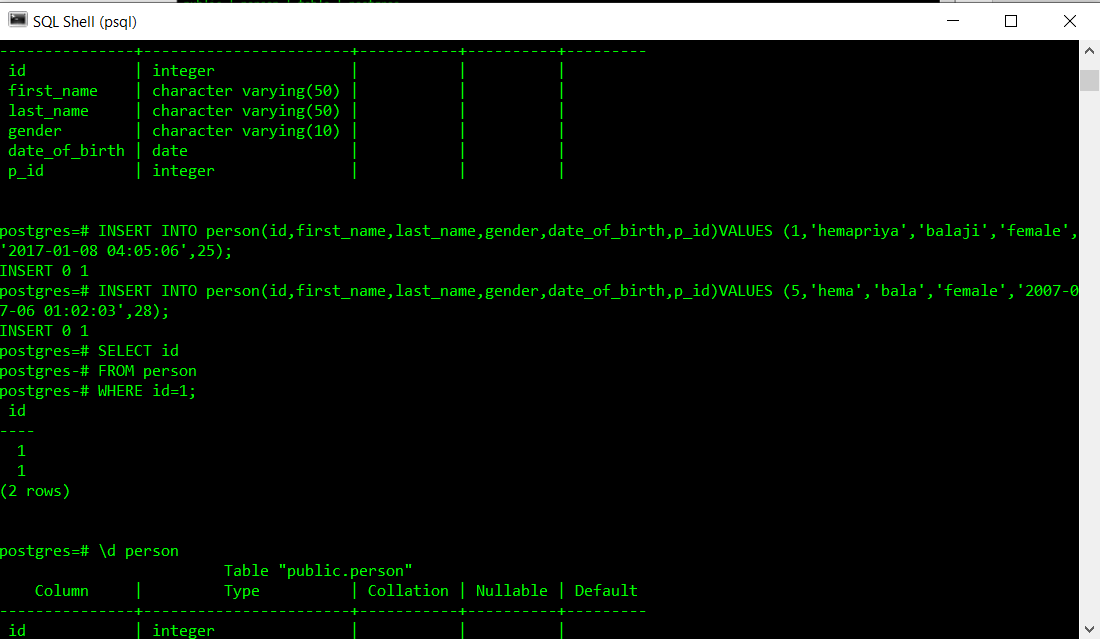


1. SELECT

SELECT <attribute name1>,…

FROM <table name>

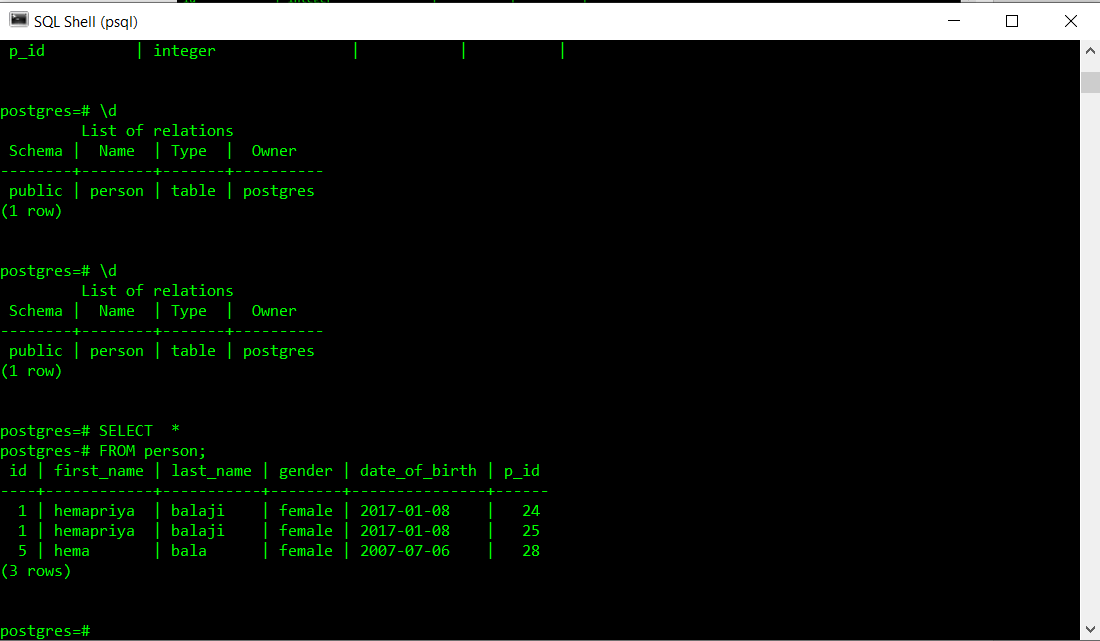
WHERE <condition>;



1. SELECT \*

SELECT \*

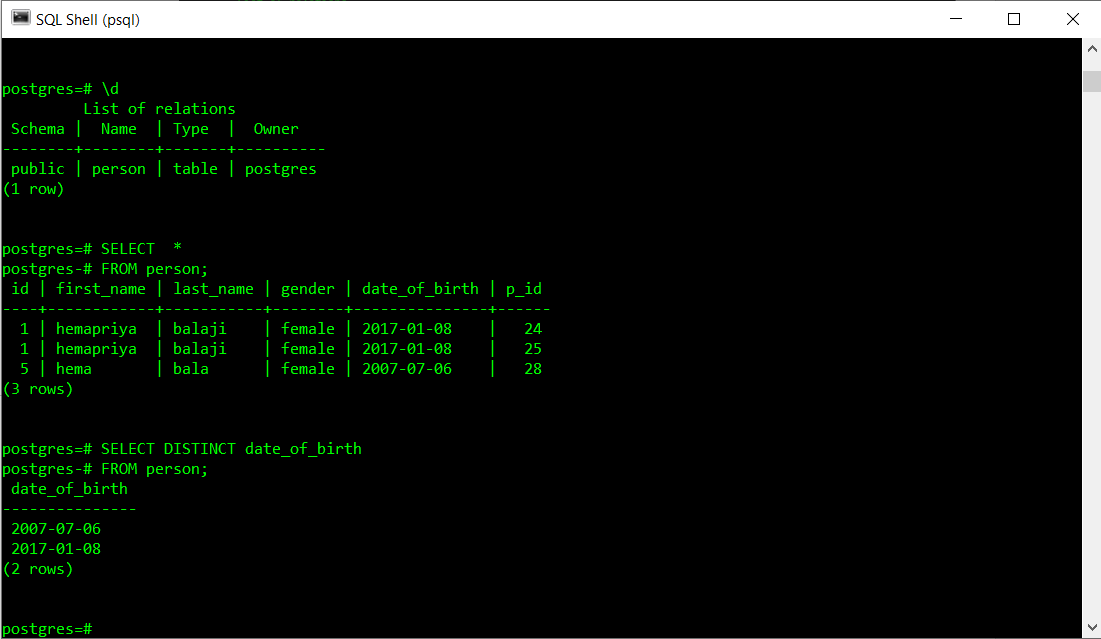
FROM<table name>;



1. SELECT DISTINCT

SELECT DISTINCT <attribute name>

FROM <table name>;

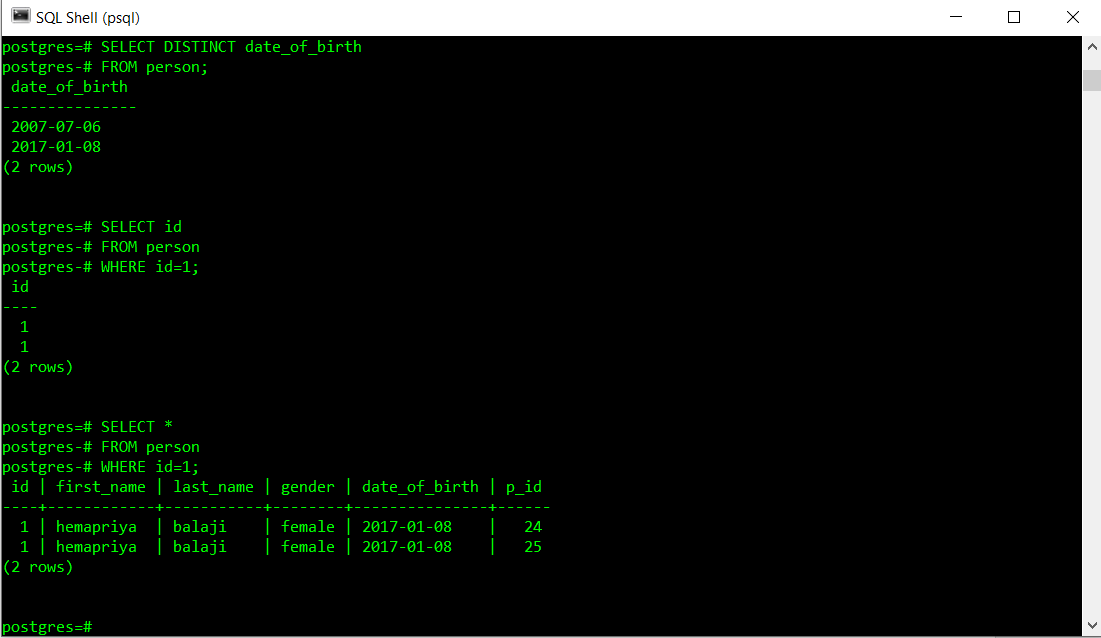


1. WHERE

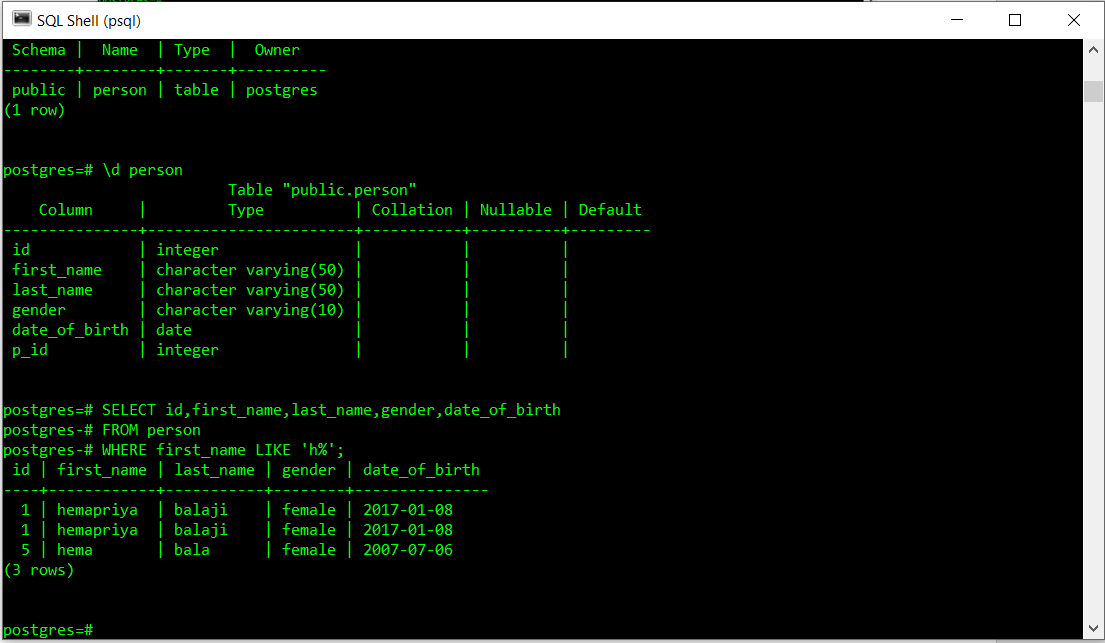
SELECT <attribute name>

FROM <table name>

WHERE <condition>;



1. Using LIKE

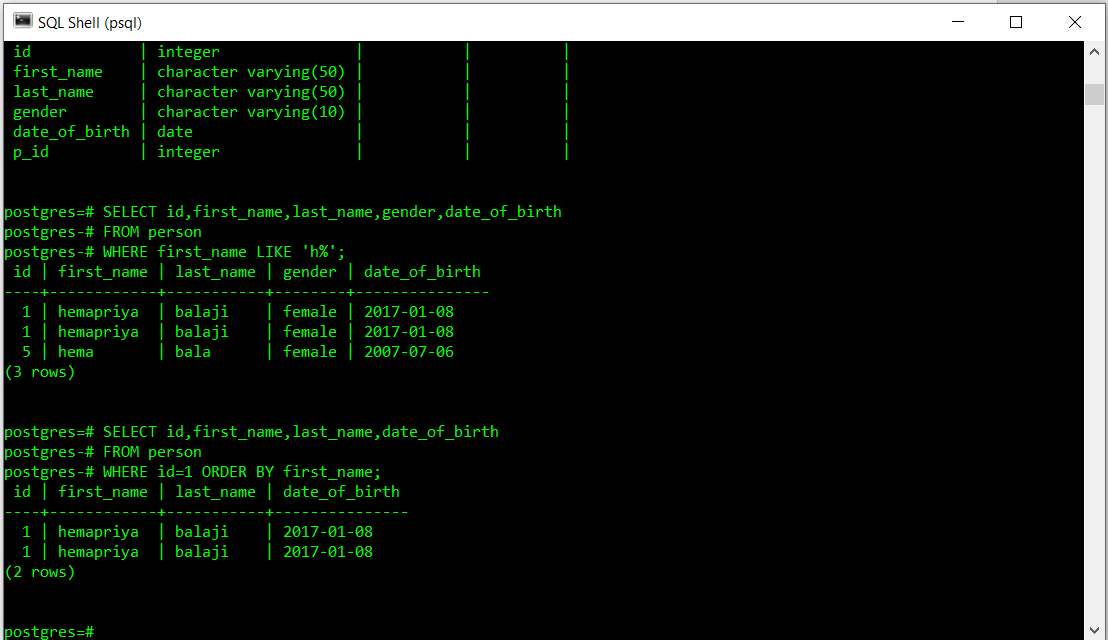


1. ORDER BY

SELECT <attribute name>

FROM<table name>

WHERE <condition>

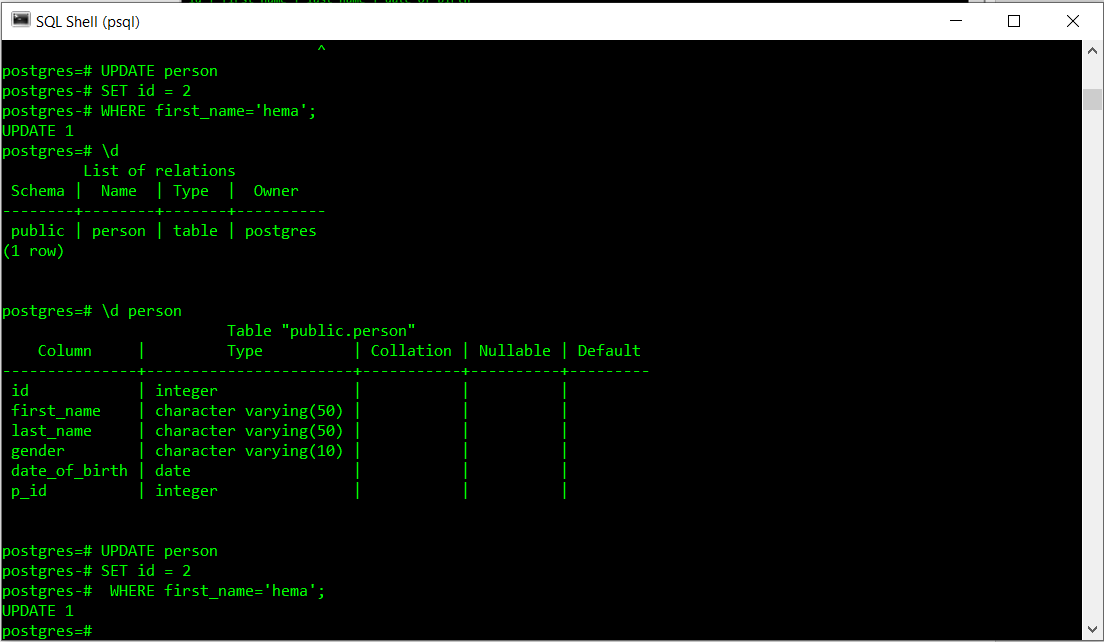


1. UPDATE

UPDATE <table name>

SET <attribute>=VALUE1

WHERE <attribute> =VALUE2



Comparison between postgresql and MS sql

|  |  |  |
| --- | --- | --- |
| Name | **Microsoft SQL Server**[**X**](https://db-engines.com/en/system/PostgreSQL) | **PostgreSQL**[**X**](https://db-engines.com/en/system/Microsoft+SQL+Server) |
| Description | Microsofts relational DBMS | Widely used open source [RDBMS](https://db-engines.com/en/article/RDBMS) info |
| Primary database model | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) info |
| Secondary database models | [Document store](https://db-engines.com/en/article/Document+Stores) [Graph DBMS](https://db-engines.com/en/article/Graph+DBMS) | [Document store](https://db-engines.com/en/article/Document+Stores) |
| |  |  | | --- | --- | | [DB-Engines Ranking](https://db-engines.com/en/ranking) info | [ranking trend](https://db-engines.com/en/ranking_trend/system/Microsoft+SQL+Server;PostgreSQL) | | [Trend Chart](https://db-engines.com/en/ranking_trend/system/Microsoft+SQL+Server%3BPostgreSQL) | | |  |  |  | | --- | --- | --- | | Score | 1075.87 | | | Rank | #3 | [Overall](https://db-engines.com/en/ranking) | |  | #3 | [Relational DBMS](https://db-engines.com/en/ranking/relational+dbms) | | |  |  |  | | --- | --- | --- | | Score | 536.77 | | | Rank | #4 | [Overall](https://db-engines.com/en/ranking) | |  | #4 | [Relational DBMS](https://db-engines.com/en/ranking/relational+dbms) | |
| Website | [www.microsoft.com/­en-us/­sql-server](https://www.microsoft.com/en-us/sql-server/) | [www.postgresql.org](https://www.postgresql.org/) |
| Technical documentation | [docs.microsoft.com/­en-ie/­sql/­sql-server/­sql-server-technical-documentation](https://docs.microsoft.com/en-ie/sql/sql-server/sql-server-technical-documentation) | [www.postgresql.org/­docs/­manuals](https://www.postgresql.org/docs/manuals/) |
| Developer | Microsoft | PostgreSQL Global Development Group info |
| Initial release | 1989 | 1989 info |
| Current release | SQL Server 2019, November 2019 | 12.4, August 2020 |
| License info | commercial info | Open Source info |
| Cloud-based only info | no | no |
| DBaaS offerings (sponsored links) info |  | * [Azure Database for PostgreSQL](https://azure.microsoft.com/en-us/services/postgresql/): A fully managed, scalable PostgreSQL relational database with high availability and security built in at no extra cost * [ScaleGrid for PostgreSQL](https://t.sidekickopen79.com/s1t/c/5/f18dQhb0SdYj8bGch0W2n0x6l2B9nMJW7t69v68pTbB4W63Bc1d16gGCMf3DJp1901?te=W3R5hFj4cm2zwW4cHbrv3K4dNZW3GGZrk1LCtCBW4fHrkG4cP21jf3R5h1204&si=370885007&pi=a68632e2-e84c-4cff-8541-4a0fd2702aba): Fully managed PostgreSQL hosting on AWS, Azure and DigitalOcean with high availability and SSH access on the #1 multi-cloud DBaaS. |
| Implementation language | C++ | C |
| Server operating systems | Linux Windows | FreeBSD HP-UX Linux NetBSD OpenBSD OS X Solaris Unix Windows |
| Data scheme | yes | yes |
| Typing info | yes | yes |
| XML support info | yes | yes info |
| Secondary indexes | yes | yes |
| SQL info | yes | yes info |
| APIs and other access methods | ADO.NET JDBC ODBC OLE DB Tabular Data Stream (TDS) | ADO.NET JDBC native C library ODBC streaming API for large objects |
| Supported programming languages | C# C++ Delphi Go Java JavaScript (Node.js) PHP Python R Ruby Visual Basic | .Net C C++ Delphi Java info JavaScript (Node.js) Perl PHP Python Tcl |
| Server-side scripts info | Transact SQL, .NET languages, R, Python and (with SQL Server 2019) Java | user defined functions info |
| Triggers | yes | yes |
| Partitioning methods info | tables can be distributed across several files (horizontal partitioning); sharding through federation | partitioning by range, list and (since PostgreSQL 11) by hash |
| Replication methods info | yes, but depending on the SQL-Server Edition | Source-replica replication info |
| MapReduce info | no | no |
| Consistency concepts info | Immediate Consistency | Immediate Consistency |
| Foreign keys info | yes | yes |
| Transaction concepts info | ACID | ACID |
| Concurrency info | yes | yes |
| Durability info | yes | yes |
| In-memory capabilities info | yes | no |
| User concepts info | fine grained access rights according to SQL-standard | fine grained access rights according to SQL-standard |

While switching from ms sql to postgresql syntax changes:

Which I faced while practicing:

* No TOP statement, so SELECT TOP 10 \* FROM table, becomes SELECT \* FROM table LIMIT 10 you can also use the maxrows attribute of CFQUERY to do this, if you want cross db code (which is good). MySQL also uses the LIMIT sytax, but Oracle uses yet another syntax
* LIKE statements are case sensitive in postgresql, they can be made case insensitive like this: SELECT \* FROM table WHERE LOWER(column) LIKE '%#LCase(var)#%' (Or you can use the ILIKE operator)
* The plus operator cannot be used for concatination so SELECT firstname + ' ' + lastname AS fullname becomes SELECT firstname || ' ' || lastname AS fullname this way works on both servers.