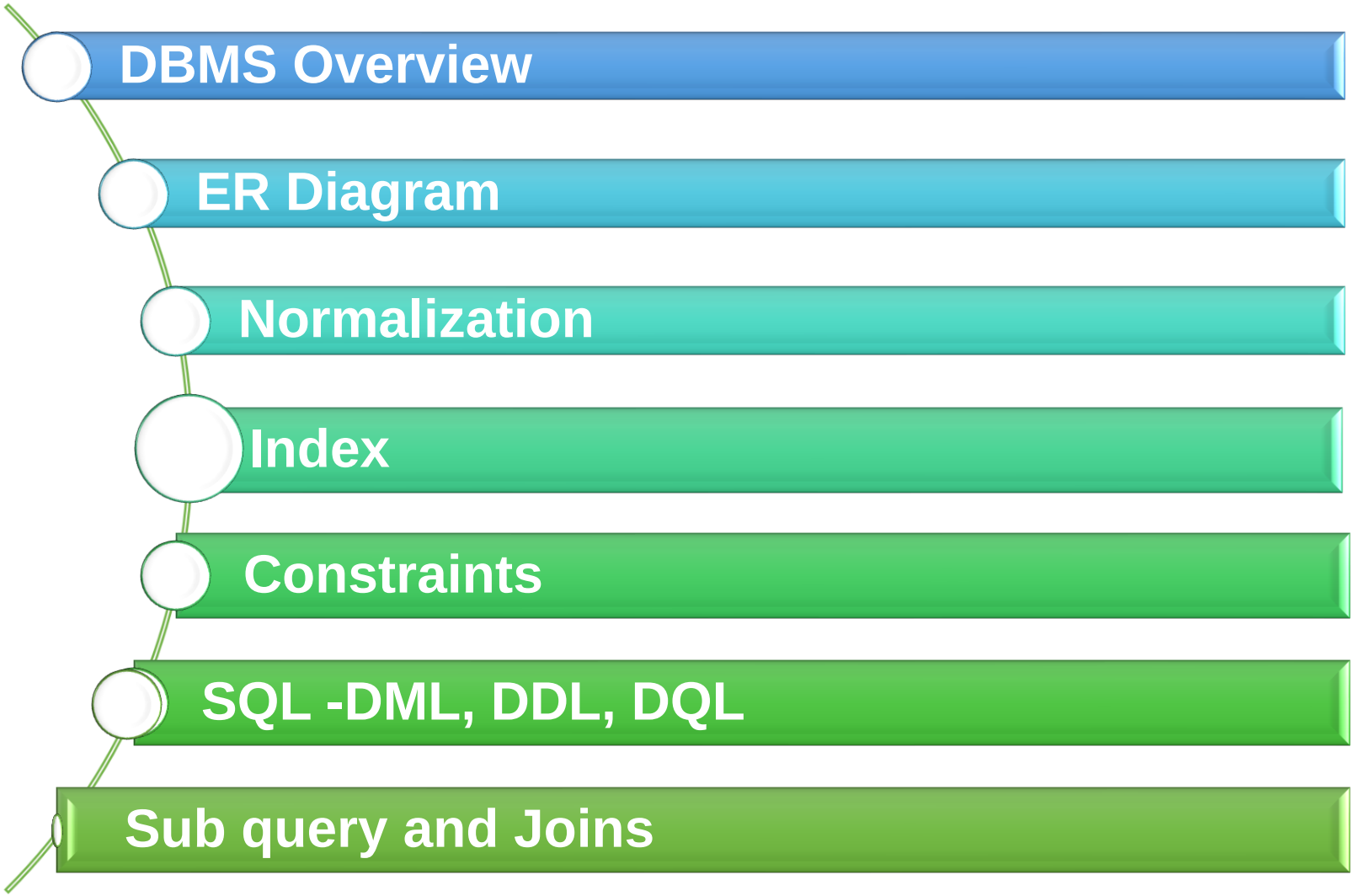
Database Fundamentals

standard chartered

axess academy

axess academy

Course Content



axess academy

DBMS Overview

DBMS

axess academy

- A database is an organized collection of structured information or data stored in a computer system.
- A database is usually controlled by a DBMS.
- Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data.
- A DBMS serves as an interface between an end-user and a database.

RDBMS

axess academy

- A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model.
- **Examples**
  - SQL
  - MS SQL Server
  - PostgreSQL
  - ORACLE
  - My-SQL
  - Microsoft Access, etc..

Terminologies

axess academy

- Schema
- Table or Relation
- Rows or Record or Tuple
- Column or Field or Attribute
- Data value

Schema and Table

axess academy

- **Schema**
  - Schema is a collection of one or more tables of data.
- **Table/Relation**
  - The RDBMS database uses **tables** to store data.
  - A table is a collection of related data entries and contains **rows and columns** to store data.
  - Each table represents some real-world objects such as person, place, or event about which information is collected.

Customer_Id	Customer_Name	Age	Location
01	Ajeet	24	India
02	Ratan	20	China
03	Lai	36	Singapore
04	Aaryan	32	Taiwan

Rows, Columns and Data

axess academy

- **Rows / Record / Tuple**
  - a single group of related data within a table

Customer_Id	Customer_Name	Age	Location
01	Ajeet	24	India

→ Row

- **Columns / Field / Attribute**
  - A column is a list of values, usually belonging to a particular field, displayed vertically in a table.
  - In Customer table, Customer_Id, Customer_Name, Age and Location are the columns

Customer_Id	Customer_Name	Age	Location
01	Ajeet	24	India
02	Ratan	20	China
03	Lai	36	Singapore

- **Data / Value** - information which database just stores.

ER Model and ER Diagram

axess academy

- The ER Model is a model for identifying entities to be represented in the database and representation of how those entities are related.
- ER diagrams are used to represent the ER model in a database, which makes them easy to convert into relations (tables).
- It is used to analyze to structure of the Database.
- It shows relationships between entities and their attributes.

axess academy

ER Diagram

Symbols used in ER Model

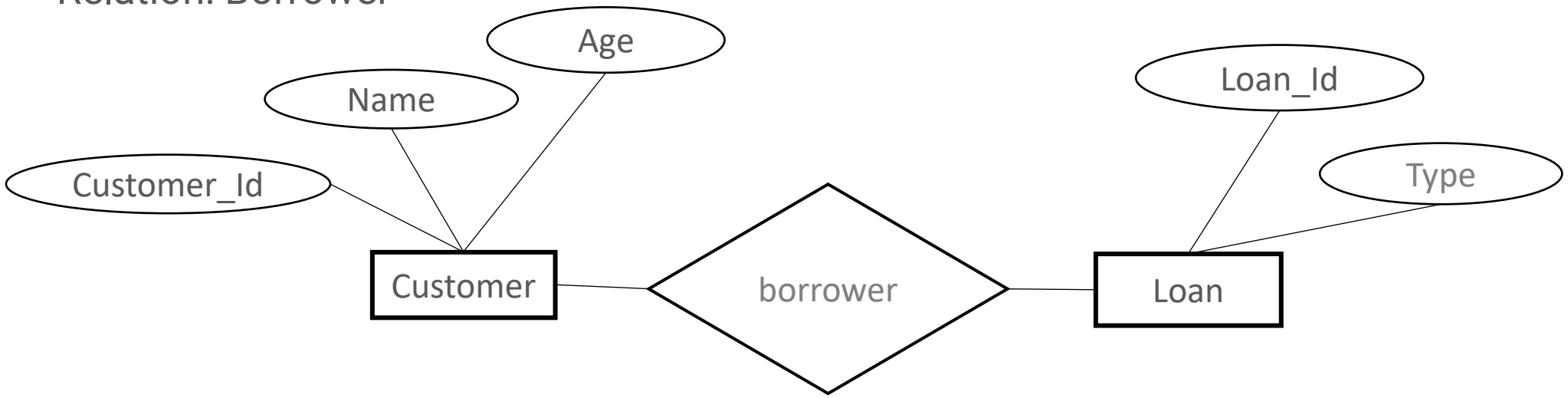
axess academy

- **Rectangles:** Rectangles represent Entities in the ER Model.
- **Ellipses:** Ellipses represent Attributes in the ER Model.
- **Diamond:** Diamonds represent Relationships among Entities.
- **Lines:** Lines represent attributes to entities and entity sets with other relationship types.

Example

axess academy

- In Loan Management,
  - Entities : customer and loan
  - Attributes : Customer_Id, Name, Age, Loan_Id and Type
  - Relation: Borrower



axess academy

Normalaization

Normalization

axess academy

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations.
- It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
- Normalization divides the larger table into smaller and links them using relationships.

Normal Forms

axess academy

- Normal forms are used to eliminate or reduce redundancy in database tables.
- In database management systems (DBMS), normal forms are a series of guidelines that help to ensure that the design of a database is efficient, organized, and free from data anomalies. There are several levels of normalization, each with its own set of guidelines, known as normal forms.

First Normal Form (1NF)

axess academy

- A relation will be 1NF if it contains an atomic value.
- It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.
- First Normal Form (1NF) does not eliminate redundancy, but rather, it's that it eliminates repeating groups

Employee Table without 1NF

Emp_Id	Emp_Name	Emp_Phone
14	John	7273445698, 9064738238
20	Harry	8574783832
12	Sam	7390372389, 8589830302

Employee Table with 1NF

Emp_Id	Emp_Name	Emp_Phone
14	John	7273445698
14	John	9064738238
20	Harry	8574783832
12	Sam	7390372389
12	Sam	8589830302

Second Normal Form (2NF)

axess academy

A table is said to be in 2NF, if

- The table meets all the conditions of 1NF
- Move redundant data to a separate table
- Create relationship between these tables using foreign keys.

Employee – Dept table after 1NF

Emp_Id	Emp_name	Salary	Dept_name	Location
101	Aarav	25000	IT	India
102	Sarah	30000	HR	China
103	Tyra	40000	HR	China
104	Lai	35000	IT	India

Employee Table after 2NF

Emp_Id	Emp_Name	Dept_Id
101	Aarav	201
102	Sarah	202
103	Tyra	202
104	Lai	201

Dept Table after 2NF

Dept_Id	Dept_Name	Loaction
201	IT	India
202	HR	China

*we need to break the table into 2, and move the redundant department data (DeptName, and Location) into it's own table. To link the tables with each other, we use the DeptId foreign key. The tables below are in 2NF.

Third Normal Form (3NF)

axess academy

A table is said to be in 3NF, if the table

- Meets all the conditions of 1NF and 2NF
- Does not contain columns (attributes) that are not fully dependent upon the primary key

Employee table

Emp_Id	Emp_name	Salary	Annual_salary	Dept_Id
101	Aarav	25000	300000	201
102	Sarah	30000	360000	202
103	Tyra	40000	480000	202
104	Lai	35000	420000	201

Employee Table after 2NF

Emp_Id	Emp_Name	Dept_Id
101	Aarav	201
102	Sarah	202
103	Tyra	202
104	Lai	201

Dept Table after 2NF

Dept_Id	Dept_Name	Loaction
201	IT	India
202	HR	China

*The AnnualSalary is also dependent on the Salary column. In fact, to compute the AnnualSalary, we multiply the Salary by 12. Since AnnualSalary is not fully dependent on the primary key, and it can be computed, we can remove this column from the table, which then, will adhere to 3NF.

axess academy

Indexing

Index

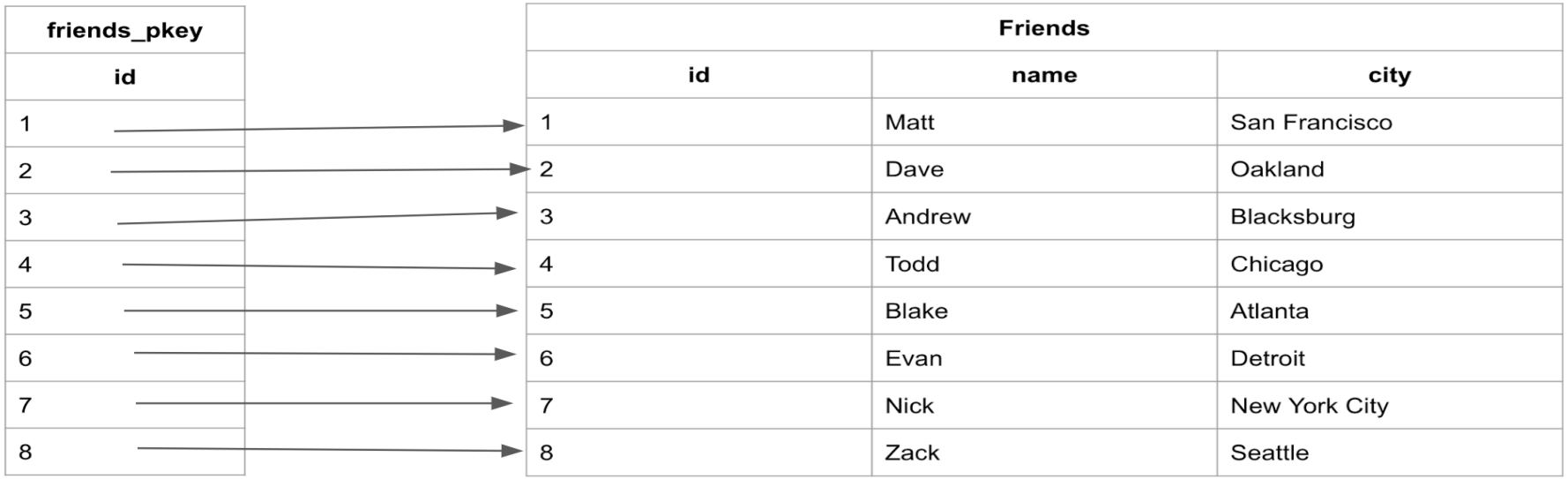
axess academy

- An index in SQL Server is a data structure associated with a table or view
- Index speeds up the retrieval of rows from the table based on the values in one or more columns.
- Indexes are put internally and are not visible to the user.
- There are two types of databases indexes:
  - Clustered
  - Non-clustered

Clustered Index

axess academy

- Clustered indexes are the unique index per table that uses the primary key to organize the data that is within the table.
- The clustered index ensures that the primary key is stored in increasing order, which is also the order the table holds in memory.
  - Clustered indexes do not have to be explicitly declared.
  - Created when the table is created.
  - Use the primary key sorted in ascending order.



friends_pkey
id
1
2
3
4
5
6
7
8

Friends		
id	name	city
1	Matt	San Francisco
2	Dave	Oakland
3	Andrew	Blacksburg
4	Todd	Chicago
5	Blake	Atlanta
6	Evan	Detroit
7	Nick	New York City
8	Zack	Seattle

Non Clustered Index

axess academy

- Non-clustered indexes are sorted references for a specific field, from the main table, that hold pointers back to the original entries of the table.

Index

friends_name_asc	
name	**id**
Andrew	3
Blake	5
Dave	2
Evan	6
Matt	1
Nick	7
Todd	4
Zack	8

points to

Table

Friends		
id	**name**	**city**
1	Matt	San Francisco
2	Dave	Oakland
3	Andrew	Blacksburg
4	Todd	Chicago
5	Blake	Atlanta
6	Evan	Detroit
7	Nick	New York City
8	Zack	Seattle

When Should Indexes be Created?

axess academy

- A column contains a wide range of values.
- A column does not contain a large number of null values.
- One or more columns are frequently used together in a where clause or a join condition.
- We can see here that the table has the data stored ordered by an incrementing id based on the order in which the data was added. And the Index has the names stored in alphabetical order.

axess academy

PostgreSQL

Overview of PostgreSQL

axess academy

- **PostgreSQL**, also known as **Postgres**, is a free and open-source relational database management system (RDBMS) emphasizing extensibility and technical standards compliance.
- PostgreSQL can run dynamic websites and web apps.
- PostgreSQL's write-ahead logging makes it a highly fault-tolerant database
- PostgreSQL source code is freely available under an open source license. This allows you the freedom to use, modify, and implement it as per your business needs.
- PostgreSQL supports geographic objects so you can use it for location-based services and geographic information systems

Supported Data Types

axess academy

- PostgreSQL supports the following data types:
  - Text Types (char(4),text)
  - Numeric Types (decimal, integer, numeric)
  - Dates and Times (timestamp/time with/ without time zone, date)
  - Boolean (boolean)
  - XML
  - JSON
  - Bits
  - Binary Data
  - Network
  - Arrays
  - Create your Data Type
  - Temporal
  - UUID
  - Array
  - JSON
  - Special Data types for storing a network address and geometric data.

Steps to database server connection

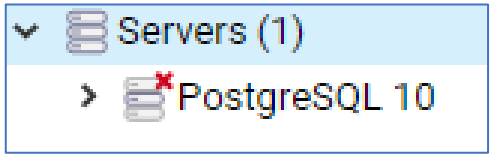
axess academy

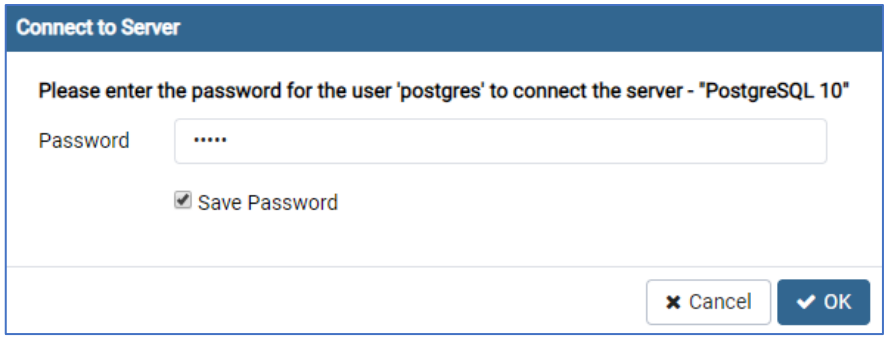
- To use PostgreSQL in your machine, you need to install:
  - PostgreSQL Database Server
  - A graphical tool to administer and manage the DB. pgAdmin is the most popular tool GUI Tool for PostgreSQL
- After successful installation, to launch PostgreSQL go to Start Menu and search pgAdmin 4 and once you select this, it will open pgAdmin homepage.

Postgres SQL

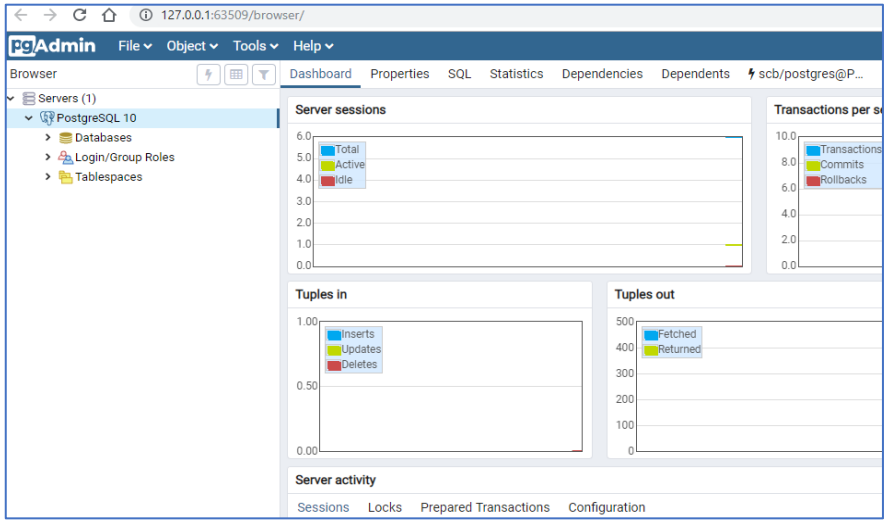
axess academy

- Click on Servers > Postgre SQL 10 in the left tree
- Enter super user password set during installation and click OK.





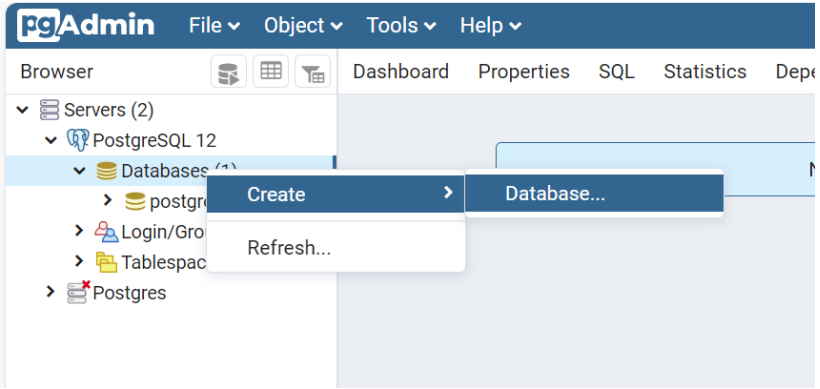
- You will see the Dashboard



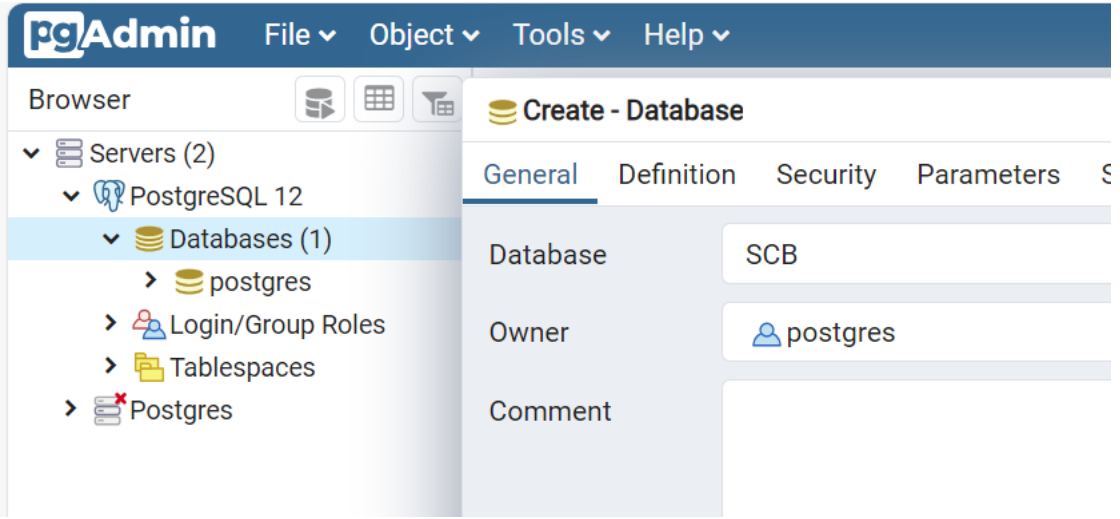
Creating DB

axess academy

- Right click on Databases and click create->Database



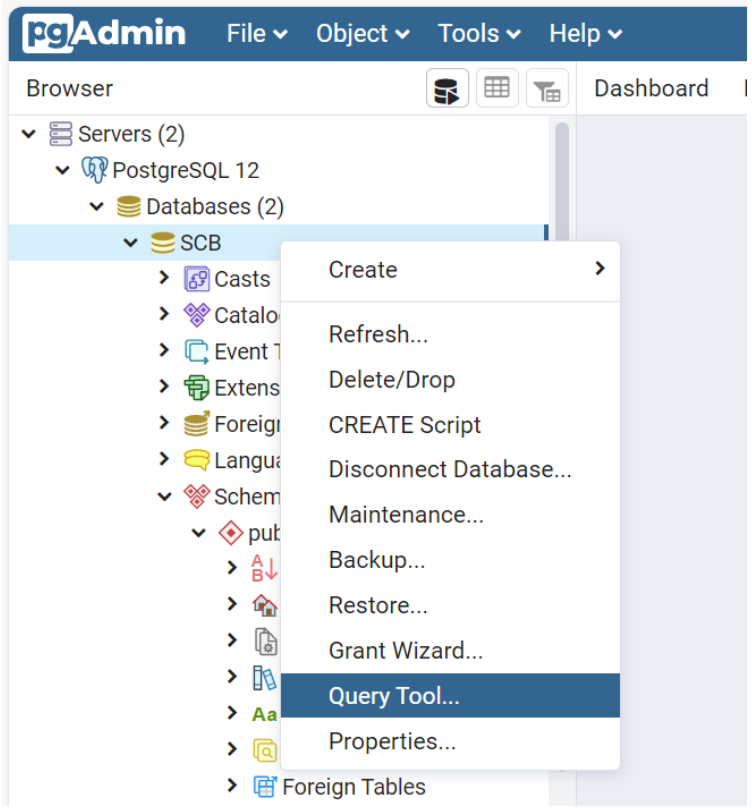
- Enter the Database name and click ok to create DB

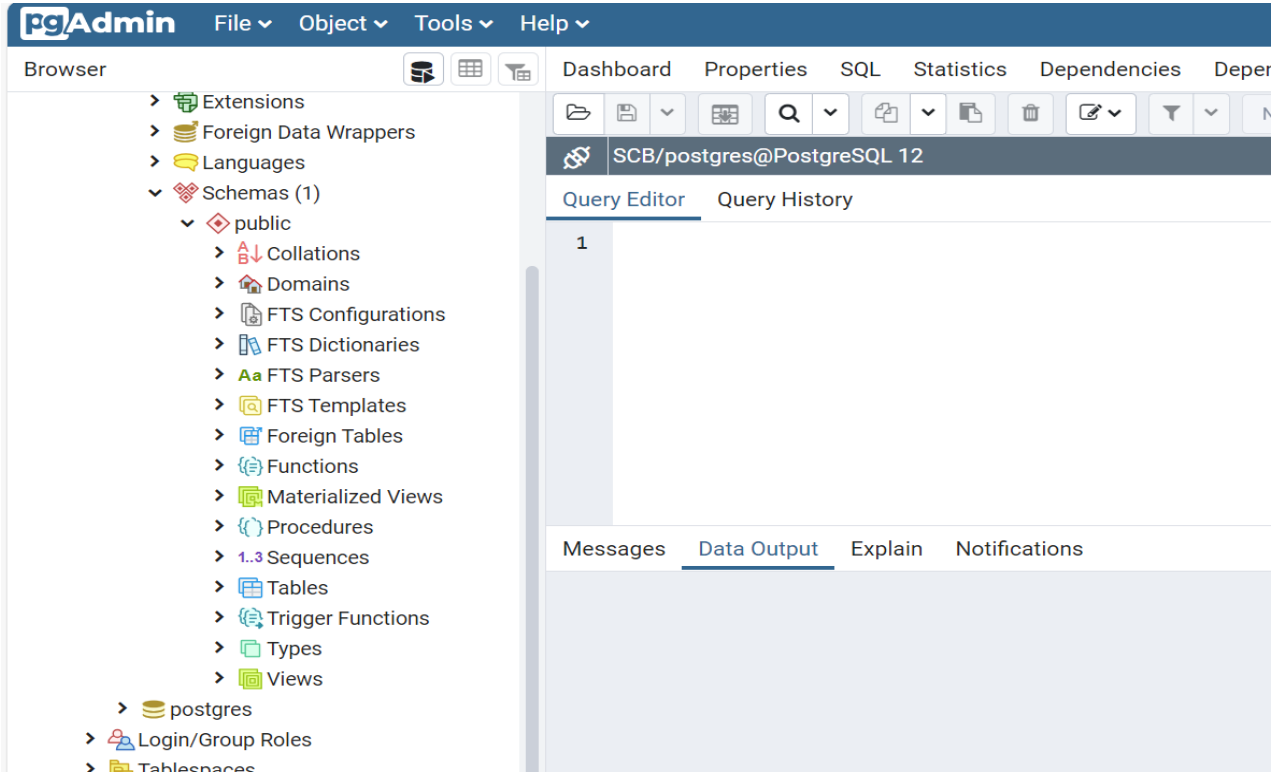


Accessing Query Tool

axess academy

- Right click on the data base and select Query tool to get SQL Query Editor





axess academy

SQL

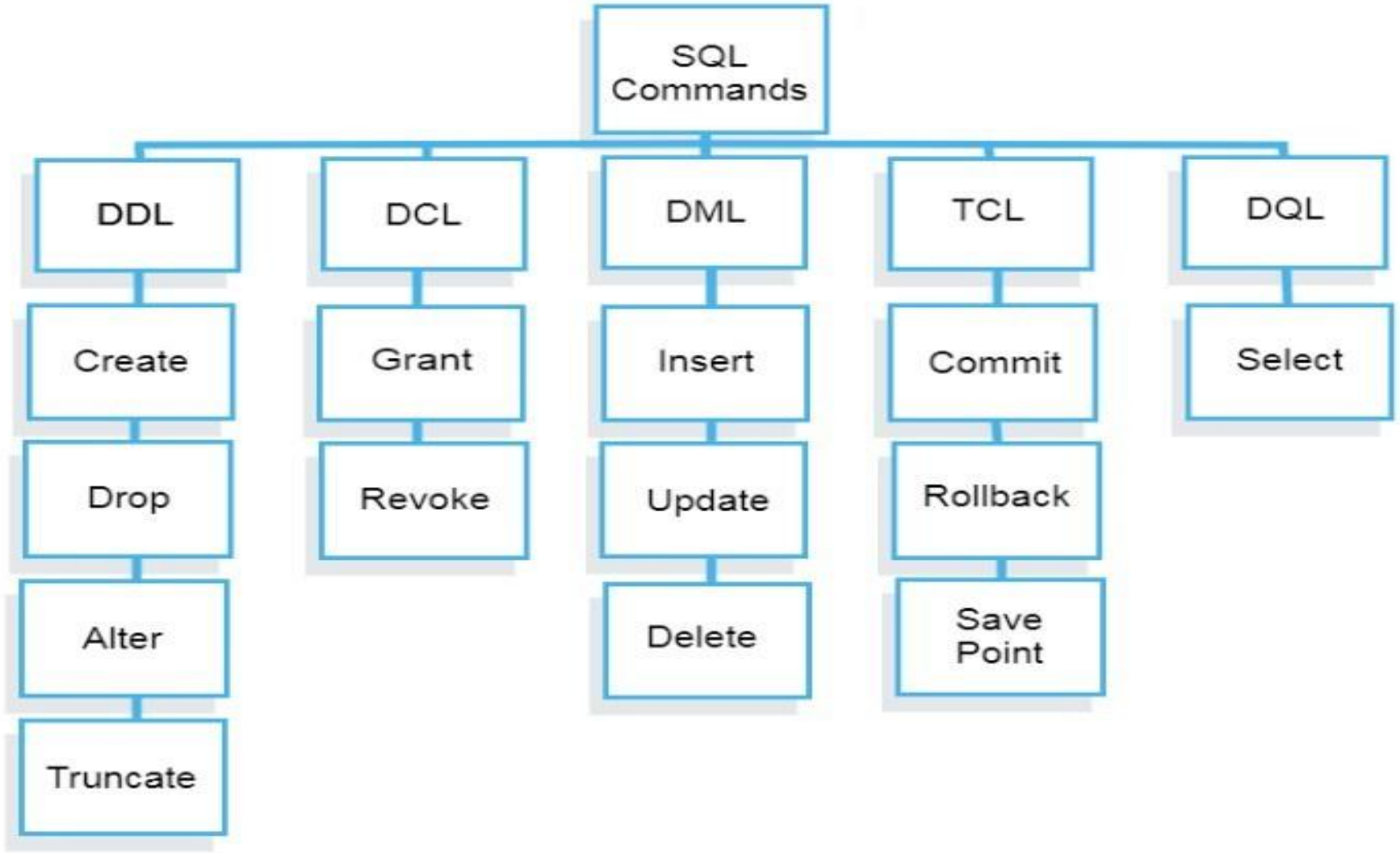
SQL

axess academy

- SQL(Structured Query Language) is a query language which is used to store and retrieve the data from a databases like MySQL, MS Access, SQL Server, Oracle, Sybase, Informix, Postgres etc.
- SQL Commands are divided into five broad categories
  - DDL (Data Definition Language)
  - DML (Data Manipulation Language)
  - DQL (Data Query Language)
  - DCL (Data Control Language)
  - TCL (Transaction Control Language)

SQL Commands

axess academy



axess academy

Constraints

Constraints

axess academy

- NOT NULL
  - Ensures that a column cannot have a NULL value
- UNIQUE
  - Ensures that all values in a column are different
- PRIMARY KEY
  - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY
  - Prevents actions that would destroy links between tables
- CHECK
  - Ensures that the values in a column satisfies a specific condition

Example

axess academy

```
Create table Dept(
        Deptno numeric(5) constraint pkey_depno primary key,
        Deptname varchar(10) constraint ukey_depname unique constraint nnull_deptname not null);
```

```
Create table employee (
        Empno numeric(5) constraint pkey primary key,
        firstname varchar(15) constraint E_nn1 not null,
        lastname varchar(20),
        age numeric constraint E_chk1 check(age >18 and age < 60) ,
        salary numeric,
        city varchar(20) constraint E_nn2 not null,
        state varchar(20) constraint E_nn3 not null,
        Deptno numeric constraint fkey references Dept(Deptno));
```

axess academy

DDL

DDL

axess academy

- The Data Definition Language is made up of SQL commands that can be used to design the database structure.
- It simply handles database schema descriptions and is used to construct and modify the structure of database objects in the database.
- It can be used to define the database schema. It allows adding/modifying/deleting the logical structures which contain the data
  - CREATE : to create a database and its objects
  - ALTER : alters the structure of the existing databases
  - DROP : delete objects from the databases
  - TRUNCATE : remove all records from a table, including memory allocated for the records are removed

DDL - Create

axess academy

- For creating a table, we must define the structure of a table by adding n ames to columns and providing data type and size of data to be stored i n columns.

Syntax:

```
create table "tablename" (
        "column1" "data type",
        "column2" "data type",
        "column3" "data type“
        );
```

Example:

```
Create table employee (
        Empno numeric(5) constraint pkey primary key,
        firstname varchar(15) constraint E_nn1 not null,
        lastname varchar(20),
        age numeric constraint E_chk1 check(age >18 and age < 60) ,
        salary numeric,
        address varchar(20) constraint E_nn3 not null,
        Deptno numeric constraint fkey references Dept(Deptno));
```

DDL - Alter

axess academy

- Used to modify or alter the structure of the table
- Alter table command is used to:
  - Add, drop, Modify table columns,
  - add, drop or modify constraints

- **Add a Column**:

  Syntax: Alter Table <Table Name> Add Column_name Column_definition

  Example: ALTER TABLE employee ADD experience numeric(3);

- **Modify a column**:

  Syntax: Alter Table <Table Name> Alter column column_name type column_definition;

  Example: ALTER TABLE employee alter salary type numeric(15,2);

  Syntax: Alter Table <Table Name> rename column <column_name> to <new_name>;

  Example: ALTER TABLE employee alter salary type numeric(15,2);

DDL – Alter

axess academy

- **Drop a column:**

  Syntax: Alter Table <TableName> Drop column column_name

  Example: ALTER TABLE employee DROP location

- **Add constraint**

  Syntax: ALTER TABLE Employee ADD CONSTRAINT Pkey1 PRIMARY KEY (empno);

  Example: Alter table Employee add constraint fkey1 foreign key(fkey_deptno) references DEPT(deptno)

- **Drop Constraints**

  Syntax: Alter Table Employee Drop constraint pkey2

  Example: Alter table Employee drop constraint fkey_deptno

DDL – drop & truncate

axess academy

- Delete:
  - This statement will delete the structure of the table
  - The table and its contents will not be recovered after delete
  - Use this statement carefully

Syntax:

Drop Table tablename

Example :

Drop table Dept

- Truncate:
  - Deletes all the records of the table
  - Changes are permanent

Syntax:

TRUNCATE Table tablename

Example :

TRUNCATE table dept;

axess academy

DML

DML

axess academy

- The SQL commands that deal with manipulating data in a database are classified as DML (Data Manipulation Language), which covers the majority of SQL statements.
- It's the part of the SQL statement that regulates who has access to the data and the database
- It helps us work with data that goes into the database such as
  - INSERT : Insert data into a Table/Database
  - UPDATE : Updates existing data within a Table/Database
  - DELETE : Deletes all records from a database Table/Database

DML - Insert

axess academy

- insert data to the table
  - With out specifying column names
  - Specifying column names
  - Using Select statement

- **Directly With out specifying column names**

```
insert into employee values ('Luke', 'Duke', 45, 50000.00, 'Hazard Co', 'Georgia');
```

- **Directly With specifying column names**

```
insert into employee(firstname, lastname, age, city, state) values ('Luke', 'Duke', 45,
'Hazard Co', 'Georgia');
```

- **Using Select statement**

```
insert into employee as select * from emp;
```

DDL – update & delete

axess academy

- Update statement is used to modify the existing records in a table.

Syntax : UPDATE table_name SET column1 = value1, column2 = value2, ...WHERE condition;

Example: update employee set salary=45000 where firstname='Luke'

- Delete statement is used to delete the records of a table

- **Deletes all records:**

  delete from employee;

- **Delete records based on Condition:**

  delete from employee where firstname=‘Luke’

Use case

axess academy

- Create table called accounts with the fields

Attribute	Data	Restriction
accno	Numeric	primary key
empId	Numeric	foreign key references empNo of employee table
balance	numeric	Should be greater than 0
branch	varchar	Not null

- Add new column called location with the type of varchar
- Rename the column accno to accountNo
- Insert data into the accounts table

axess academy

DQL

DQL

axess academy

- DQL statements are used to query the data contained in schema objects.
- The DQL Commands' goal is to return a schema relation depending on the query supplied to it.
- DQL can be defined as follows: It’s a part of a SQL statement that lets you get data from a database and put it in order.
- The most used code by Data scientists and analysts.
  - SELECT: It helps to do data analysis

DQL - Select

axess academy

- Retrieve data from Database

- **To select all columns from a table**

  Select * from employee

- **To select Specific columns from a table**

  select firstname, lastname, city from employee;

- **Creating a new name for a column for display purpose**

  Select firstname as FName, lastname as LName, city from Employee

- **To select all values of Age column**

  Select All age from employee

- **To display distinct values of Age column**

  Select distinct age from empinfo

Operators

axess academy

- Relational Operators:
  - Different Relational operators are = , < , > , <= , >= , != or < >

```
select last, city, age from empinfo where age > 30;
select last, city, age from empinfo where age <= 30;
select last, city, age from empinfo where first= ‘raj’;
```

- Logical Operators:
  - Different Logical operators are AND, OR and NOT

```
Select * from empinfo where age >20  AND age <80
Select * from empinfo where age <20  or age >60
Select * from empinfo where state in (‘Arizona’ , ‘Michigen’);
Select * from empinfo where age not in(10,20,34,67)
```

Like and null keyword

axess academy

- Pattern Search with like keyword
  - % - Matching any character or any numeric of Characters
  - _ - Matching one character only

```
Select * from empinfo where first like ‘m%’
Select * from empinfo where  first like ‘m%y’;
Select * from empinfo where first like ‘_ _ _ y’
Select * from empinfo where first like ‘_a_’
```

- Null Keyword
  - To fetch record whose column is Having NULL, is operator is used.

```
Select * from empinfo where last IS NULL
Select * from empinfo where last IS NOT NULL
```

Sorting records

axess academy

- Order by clause is used with Key words ASC and Desc for sorting records
- Default sorting key word - ASC

- Select * from Employeeinfo order by AGE
- Select * from Employeeinfo order by AGE asc
- Select * from Employeeinfo order by AGE desc
- Select first,last, age, address from employeeinfo order by 3

Aggregate Functions

axess academy

- Used to perform basic mathematical operations like SUM(),AVG(),COUNT(),MIN() and MAX()

- **Min & Max**

  Select min(age) from Empinfo

  Select Max(age) from EmpInfo

- **Avg & Sum**

  Select AVG(Age) from empinfo

  Select SUM(AGE) from empinfo

- **Count** (will not count null values)

  Select count(*) from Empinfo

  Select count(last) from empinfo

  Select count(distinc age) from empinfo

axess academy

Use Case

Use case

axess academy

- Display all employee details from employee table
- Display all employee details in sort it based on the empId.
- Display the total salary from employee table.
- Display firstname and lastname in single column as empName from employee table
- Display empId, salary, location from the employee table from specific location, whose salary greater than ‘n’ and sort based on name.
- Display empName, location from employee table whose name starts with ‘s’ in the specific location.

axess academy

Grouping & Having

Grouping data

axess academy

- The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country
- The GROUP BY statement is often used with aggregate functions to group the result-set by one or more columns.

	address character varying (20)	count bigint
1	Singapore	1
2	China	2
3	India	3

Syntax:

SELECT column_name(s) FROM table_name WHERE condition GROUP BY column_name(s)

Example:

select address,count(address) from employee group by address

Having

axess academy

- The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

Data Output | Explain | Messages | Not

	address character varying (20)	count bigint
1	China	2
2	India	3

Syntax:

SELECT column_name(s) FROM table_name WHERE condition GROUP BY column_name(s)
HAVING con

Example:

select address,count(address) from employee group by address having count(address)>=2

axess academy

Sub Query & Joins

Sub Query

axess academy

- In SQL a Subquery can be simply defined as a query within another query.
- In other words we can say that a Subquery is a query that is embedded in WHERE clause of another SQL query.

Syntax:

```
SELECT column_name FROM table_name WHERE column_name expression operator (
    SELECT COLUMN_NAME from TABLE_NAME WHERE ... );
```

Example:

```
select * from employee where salary = (select max(salary) from employee)
```

	empno [PK] numeri	firstname character varying (15)	lastname character va	age numeric	salary numeric	address character	deptno numeric
1	105	Sofia	Robbins	54	800000	China	203
2	109	Stetson	Best	54	800000	China	203

Joins

axess academy

- SQL Join statement is used to combine data or rows from two or more tables based on a common field between them.

Syntax:

```
SELECT table1.column1,table1.column2,table2.column1,.... FROM table1 JOIN table2
    ON table1.matching_column = table2.matching_column;
```

Example:

```
select * from dept join employee on dept.deptno = employee.deptno
```

	deptno numeric (5)	deptname character va	empno numeric (	firstname character va	lastname character	age numeric	salary numeric	address character vary	deptno numeric
1	201	TSA	101	Jones	smith	40	500000	India	201
2	202	CCIB	102	Eric	Cohen	30	450000	China	202
3	203	HR	103	Azrael	Marin	45	360000	India	203
4	204	Security	104	Jared	Miller	28	650000	Singapore	204
5	203	HR	105	Sofia	Robbins	54	800000	China	203

Types of joins

axess academy

- INNER JOIN
  - Returns records that have matching values in both tables
- LEFT OUTER JOIN
  - Returns all records from the left table, and the matched records from the right table
- RIGHT OUTER JOIN
  - Returns all records from the right table, and the matched records from the left table
- FULL OUTER JOIN
  - Returns all records when there is a match in either left or right table

Use case

axess academy

- Display the number of employees in each department.
- Display employee data who receives highest salary.
- Create tables as given below and insert the records.
- Display number of customers in each location
- Display the number of loans each customer applied