**What is Database?**

We create data every day and we don’t even realize it. When we wake up in the morning to check our notifications on our phones, transactions that we do in payment apps or browse social media it is all stored somewhere. It’s stored on a database.

# Database Management System (DBMS)

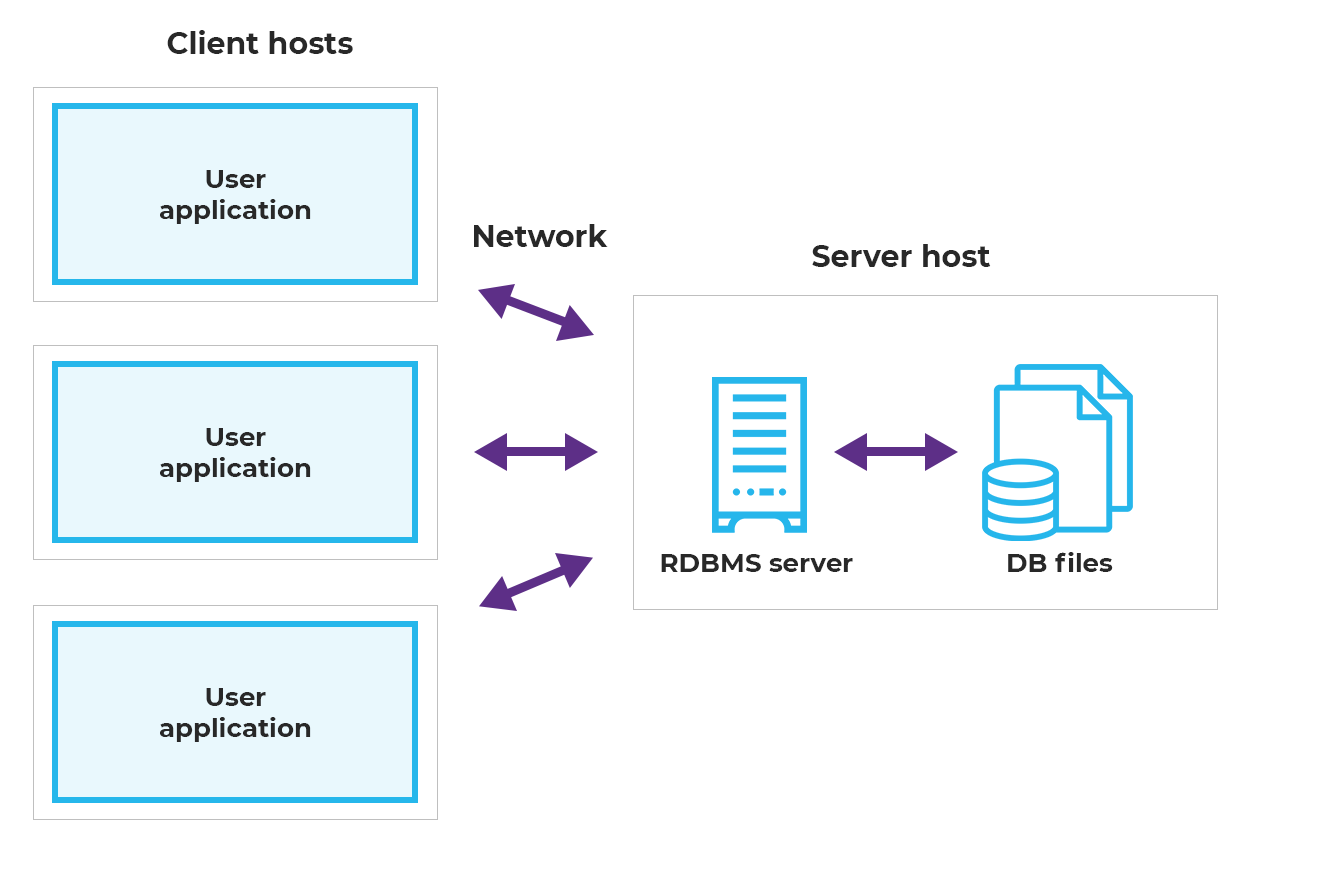
# DBMS is the software that would be installed on your personal computer or on a server, then you would use it to manage one or more databases

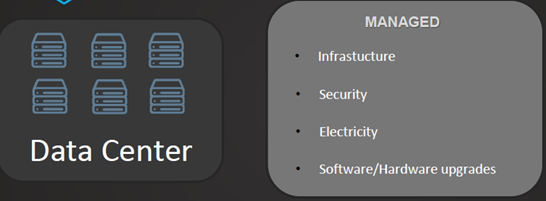
# Oracle, MySQL, SQL Server, PostgreSQL are couple of Relational Database Management Systems (RDBMS)

# Companies will install any of these Database Management Systems on their dedicated machines and users can access that data through network

# Description of Figure 16-1 follows https://miro.medium.com/max/800/1*ZbrQg86aqafQgtpofXJReQ.png

# Client => users machine Database server will have multiple databases





**On premises**

**Infrastructure=> Storage(Hard Disk) + Processor + RAM**

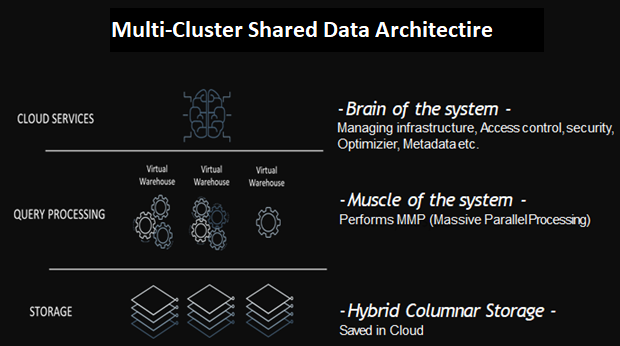
**1 TB + 2TB**

**Storage and compute will be in a single machine together (coupled)**

**So if we have to increase processor/storage or RAM capacity we can’t increase it beyond a limit**

**If you have to run a query, it requires Processor, RAM and Memory**

**Snowflake Architecture**



SNOWFLAKE LAYERS

Snowflake’s unique architecture consists of three key layers, all of them with High Availability. **The price is also charged separately for each layer**.

Each layer can independently scale : storage, compute, and services.

## Database

# A database is an organized collection of data. It is a collection, or a set of tables.

# Database System Concepts

# If we take example of banks, bigger banks will maintain multiple databases one for each department

# All accounts related data will be stored in one database, loan related data will be stored in another database, credit card data will be in another database

# How data will be stored in a database+

# Tables and Table Clusters - 11g Release 2 (11.2)

In RDBMS, the data is stored in database objects called **tables**. A table is a collection of related entries and consists of columns and rows.

# Everyone would have used Excel, if we correlate database with it. Excel is a collection of sheets. It contains data in rows and columns.

# https://miro.medium.com/max/1400/1*6-Eq75Vd67hq2pXbve_YpQ.jpeg

# Sheet we may correlate as a table

# Columns are also referred as Fields and rows as records

# If we have to interact with database and read the data, we will have to use SQL.

# SQL stands for ****Structured Query Language****. It is the language that translates all that data into understandable information

# There are different DBMS like Relational Database Management Systems, NoSQL Database Systems etc..

# RDBMS are the mostly used DBMS types, examples: Oracle, MySQL, SQL Server, DB2

# A relational database refers to a database that stores data in a structured format, using rows and columns. ... It is "relational" because the values within each table are related to each other. Tables may also be related to other tables.

# https://miro.medium.com/max/700/1*xY-atJW9j34HObJyceaV4A.png

# NoSQL database systems:

# MongoDB is one of NoSQL database where all the data will be stored in JSON documents

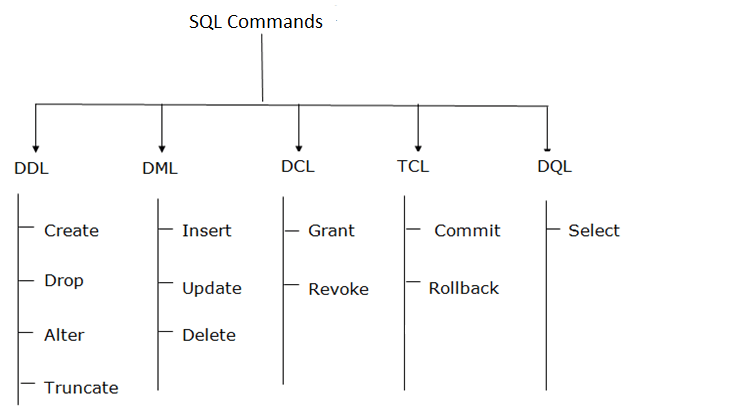
# NoSQL databases can be queried with SQL.

# SQL Commands

* SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.
* SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

## Types of SQL Commands

There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.



### 1. Data Definition Language (DDL)

* DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
* All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

* CREATE
* ALTER
* DROP
* TRUNCATE

**a. CREATE** It is used to create a new table in the database.

**Syntax:**

CREATE TABLE TABLE\_NAME (COLUMN\_NAME DATATYPES[,....]);

**Example:**

CREATE TABLE SKILLSCALER(Name VARCHAR2(20), Email VARCHAR2(100), DOB DATE);

**b. DROP:** It is used to delete both the structure and record stored in the table.

**Syntax**

DROP TABLE table\_name;

**Example**

DROP TABLE SKILLSCALER;

**c. ALTER:** It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

**Syntax:**

To add a new column in the table

ALTER TABLE table\_name ADD column\_name COLUMN-definition;

To modify existing column in the table:

ALTER TABLE table\_name MODIFY(column\_definitions....);

**EXAMPLE**

ALTER TABLE SKILLSCALER\_DETAILS ADD(ADDRESS VARCHAR2(20));

ALTER TABLE SKILLSCALER\_DETAILS MODIFY (NAME VARCHAR2(20));

**d. TRUNCATE:** It is used to delete all the rows from the table and free the space containing the table.

**Syntax:**

TRUNCATE TABLE table\_name;

**Example:**

TRUNCATE TABLE SKILLSCALER;

### 2. Data Manipulation Language

* DML commands are used to modify the database. It is responsible for all form of changes in the database.
* The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

Here are some commands that come under DML:

* INSERT
* UPDATE
* DELETE

**a. INSERT:** The INSERT statement is a SQL query. It is used to insert data into the row of a table.

**Syntax:**

INSERT INTO TABLE\_NAME  (col1, col2, col3,.... col N)  VALUES (value1, value2, value3, .... valueN);

Or

INSERT INTO TABLE\_NAME  VALUES (value1, value2, value3, .... valueN);

**For example:**

INSERT INTO SKILLSCALER (ID, NAME) VALUES (1, "Ramesh");

**b. UPDATE:** This command is used to update or modify the value of a column in the table.

**Syntax:**

UPDATE table\_name SET [column\_name1= value1,...column\_nameN = valueN] [WHERE CONDITION]

**For example:**

UPDATE students  SET User\_Name = 'Hari'    WHERE Student\_Id = '3'

**c. DELETE:** It is used to remove one or more row from a table.

**Syntax:**

DELETE FROM table\_name [WHERE condition];

**For example**

DELETE FROM SKILLSCALER WHERE SKILLSCALER\_NAME="Kiran";

### 3. Data Control Language

DCL commands are used to grant and take back authority from any database user.

Here are some commands that come under DCL:

* Grant
* Revoke

**a. Grant:** It is used to give user access privileges to a database.

**Example**

GRANT SELECT, UPDATE ON MY\_TABLE TO SOME\_USER, ANOTHER\_USER;

**b. Revoke:** It is used to take back permissions from the user.

**Example**

REVOKE SELECT, UPDATE ON MY\_TABLE FROM USER1, USER2;

### 4. Transaction Control Language

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

* COMMIT
* ROLLBACK

**a. Commit:** Commit command is used to save all the transactions to the database.

**Syntax:**

COMMIT;

**Example:**

DELETE FROM CUSTOMERS  WHERE AGE = 25;

COMMIT;

**b. Rollback:** Rollback command is used to undo transactions that have not already been saved to the database.

**Syntax:**

ROLLBACK;

**Example:**

DELETE FROM CUSTOMERS WHERE AGE = 25;

ROLLBACK;

### 5. Data Query Language

DQL is used to fetch the data from the database.

It uses only one command:

* SELECT

**a. SELECT:** This is the same as the projection operation of relational algebra. It is used to select the attribute based on the condition described by WHERE clause.

**Syntax:**

SELECT expressions  FROM TABLES WHERE conditions;

**For example:**

SELECT emp\_name  FROM employee  WHERE age > 20;

**DATA TYPES IN SQL**

Frequently used data types

1. **Integer.** A whole number without a fractional part. E.g. 1, 156, 2012412

2. **Decimal.** A number with a fractional part. E.g. 3.14, 3.141592654, 961.1241250

3. **Boolean.** A binary value. It can be either TRUE or FALSE.

4. **Date.** Speaks for itself. You can also choose the format. E.g. 2017-12-31

5. **Time.** You can decide the format of this, as well. E.g. 23:59:59

6. **Timestamp.** The date and the time together. E.g. 2017-12-31 23:59:59

7. **Text.** This is the most general data type. But it can be alphabetical letters only,

or a mix of letters and numbers and any other characters. E.g. hello, R2D2

**Getting started**

1. Create Database

CREATE DATABASE database\_name

CREATE DATABASE hero\_database

2. Create Table

CREATE TABLE table\_name (  
 id INTEGER PRIMARY KEY,  
 name TEXT,  
 age INTEGER  
);

CREATE TABLE EMPLOYEE(  
 id INTEGER PRIMARY KEY,  
 name TEXT,  
 age INTEGER,  
 salary TEXT  
);

**Constraints**

Used when a table is created (or altered later) to limit the type of data put inside. There are column level and table level restraints.

**NOT NULL** => column cannot have a null value

**UNIQUE** => all values in column are different

**PRIMARY KEY** => combination of NOT NULL and

**FOREIGN KEY**=>uniquely identifies row in another table using primary key of parent table

**CHECK**=> ensures all values in column satisfy a specific condition

**DEFAULT**=> sets a default value for the column when no value is specified

**INDEX**=> Used to create and retrieve data from the database quickly

**Distinct**

Only selects unique values.

SELECT DISTINCT column\_name FROM table\_nameSELECT DISTINCT ability FROM employee

**Where** =>Selects data based on specified conditions.

SELECT column\_name FROM table\_name WHERE condition

SELECT name FROM employee WHERE name = "Kiran"

Can use AND and OR to specify more conditions as well.

SELECT name FROM Skillscaler WHERE name = "kiran" OR "Hari"

**Count** =>Gives you the amount of records that meets a specified condition.

SELECT COUNT(column\_name) FROM table\_name WHERE condition

SELECT COUNT(ability) FROM employee WHERE ability = "ESP"

**Order By** =>Orders the result in either ascending (default) or descending order of a column’s values.

SELECT "column\_name" FROM "table\_name" WHERE "condition" ORDER BY column\_name ASC|DESC

SELECT name FROM employee WHERE age > 16 ORDER BY age DESC

**Group By=>** Groups results by a specified column.

SELECT COUNT(column\_name) FROM table\_name GROUP BY column\_nameSELECT COUNT(ability) FROM employee GROUP BY ability

**Insert Into =>** Insert data into the table.

INSERT INTO table\_name (name, age, gender, ability) VALUES ("name\_value", age\_value, "gender\_value")

INSERT INTO employee (name, age, gender, ability) VALUES ("Kiran", 18, "male", " things")

**Update =>** Update the information for data inside the table.

UPDATE table\_name SET name = "new\_name" WHERE name = "old\_name";UPDATE employee SET name = "spoderman" WHERE name = "Spiderman";

**Delete=>** Delete a row from the database.

DELETE FROM table\_name WHERE conditionDELETE FROM employee WHERE name="spoderman"

**Select Into**=> Copies data from one table into another table.

SELECT column\_name(s) INTO new\_table FROM old\_table WHERE conditionSELECT name INTO people FROM employee WHERE name = "Kiran"

**Stored Procedure**

For when you have common code that you run, you can store it and just call it.

To store it:

CREATE PROCEDURE *procedure\_name*  
 AS  
*sql\_statement*  
 GO;

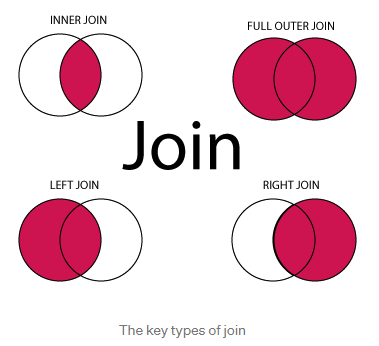
CREATE PROCEDURE *sp\_emp*   
 AS  
*SELECT \* FROM employee WHERE name= "Kiran"*  
 GO;

To call it:

EXEC *procedure\_name*

**Joins**

There are four types of joins.



* INNER JOIN — returns all rows when there is at least one match in BOTH tables
* LEFT JOIN — Returns all rows from the left table, and the matched rows from the right table
* RIGHT JOIN — Returns all rows from the right table, and the matched rows from the left table
* FULL JOIN — Returns all rows when there is a match in ONE of the tables

SELECT column\_name FROM table\_name\_1 INNER JOIN table\_name\_2 ON table\_name\_1.column\_name = table\_name\_2.column.name