Database fundamentals: DB concepts

Data organization – categorizing & classification of data to make it usable

Types of data:

Scalar – basic building blocks with only 1 value

Composite – made up of scalars or even other composite types

Abstract – data type that’s defined by its behavior. E.g:- tuple, set, stack, queue, graph

Data structures

Linear – arrays, lists

Tree – binary trees, heaps, space

Hash – distributed hash table, hash tree

Graphs – decision trees, directed, acyclic graphs

Storage of data throughout time:

Punch cards

Magnetic drums

Magnetic tapes & drives

Hard disks

Floppy disks

Compact disks (CD)

Zip drives

Digital video disks (DVDs)

Secure Digital (SD) cards

Universal Serial Bus (USB) flash drives

Cloud storage

Popular data storage methods

Flat files – eg: text files

Spreadsheets – excel, google sheets

Databases – MySQL, Microsoft Access

Popular DBs

MySQL – owned by Oracle but has GPL license

Oracle DB – one of the oldest & most reliable

SQL Server – Microsoft, competitor of Oracle DB

Postgres – free, open source & easy to work with

Advantages of DBMS

Data transfer – transfer data from 1 DB to another

Security

Integration – ensures all data is consistent & provides a bird’s eye view of the organization

Increases productivity

Decision making is lot easier

Ease of use

Features of DBMS

Multiple users

Data storage – interact with data storage easily

Security

Tasks are non-technical (mostly)

Duplication control – ensures that 2 users can’t modify the same record at the same time

Redundancy control – gives us uptime reliability to a certain extent

Integrity – ensures data is accurate, consistent

Metadata – provides us data about the database

DBMS types

Hierarchical – organizes data like the branches of a tree

Distributed (a.k.a network) – extension of hierarchical DBs where data has a many-to-many relationship & utilizes a network

Relational – organizes data as logically independent tables

Object – stores data as objects. So we can store not only the values & properties of the objects but also the operations on that objects.

Objects with related values & operations can be grouped together as classes

Hierarchical DB

Data is ordered in a tree-like structure

There is a one-to-many relationship between records

Eg: organizational chart – a parent record can have several child nodes

Distributed DB

Good for horizontal scaling

Availability, fault tolerance, throughput, latency & scalability is high

A distributed DB can be of 2 types:

Homogenous – copies of the main DB are distributed throughout the network

Heterogenous – Network can contain completely different schemas, data models & supported by different OS but are connected via the network

Relational DB

Data is stored in the form of tables

Links are created between data elements

Since the data is highly structured, they can easily be extracted and manipulated via queries that take less time

Object-oriented DB

Works well with OOP languages like Java, C++, C#

Implements OOP concepts like abstraction, encapsulation, inheritance & polymorphism

Characteristics of a good database

Availability - Must always be available whenever needed

Provide accurate, reliable & consistent data

Security – should only let authorized people to access it. Should prevent data from being corrupt or stolen.

Redundancy –

Good redundancy – When the DB is large & copies of our DB exist & are distributed across the network so that if one server fails, the other can take over

Bad redundancy – On the same server, redundancy must be minimized as they take up space where we could be storing good data

Performance – data must be retrieved & modified in a reasonable amount of time

Storage – must have adequate storage

DBMS users

DBA – Database administrators are gatekeepers, controllers of the DB.

Oversee software & hardware requirements of DB

Control access to DB for users

Take care of day-to-day operations – monitoring, performance, etc.

Controlling security

Determining content of DB

Database designers – Create schema for DB

Responsible for database design & architecture

Create storage structures – tables & relationships

Application developers – build apps that interact with the DB

End users – There are 4 different types of users:

Sophisticated – someone who is very familiar with the DB & knows the query language

Parametric – Not so familiar with technical knowledge of the DB & relies on UI

Casual - infrequently accesses the DB, could be a contractor

Standalone – people using a personal DB