Database Systems

Introduction

References:

1. Database Systems the Complete Book. Garcia-Molina et. al.
2. Database System Concepts: Silberschatz et. al.

Databases in today's world

- Databases is essential to every business, big or small.
- Many major Web sites: e.g., Google, Yahoo!, Amazon, Twitter…
  - Each have huge information.
  - Supports some queries and updates to data.
  - Provides you the information you request- finding the right pin in a huge haystack.
  - There is a database behind the scenes (web-interface).

Today: BIG databases

- Data can be very very huge.
  - Petabytes, more…
- E.g., Facebook, Google, Twitter, Amazon, WhatsApp …
- Data can consist of
  - Information in small units, such as name, DOB, Aadhar ID, etc.
  - Images (E-mail services,Facebook, Twitter, Amazon …)
  - Videos (--as above--)
  - Documents---pdf, html, …
  - Structured data:
    - struct data, or array of structures as in C.
      - Used in genome, protein and such databases.
  - 2d, 3d (possibly higher dimensional data)
    - Maps database (Google Maps, Mapquest , ….)
    - 3d chemical structure database.

World-Wide access

- Google, Facebook, Twitter, Amazon, Facebook and many many others are accessible world-wide.
- World-wide access: implications
  - Data should be replicated in multiple data centers world-wide.
    - Otherwise, a data center is down would imply global inaccessibility.
    - Partial data center outage implies partial inaccessibility.
  - Data is partitioned, for e.g., with respect to geographical proximity
    - E.g., facebook users in a country/geographical region typically access data of users in that region.
- Data is highly distributed, both partitioned and replicated.
- Is the data across copies consistent??
- Is the data Durable??

Scientific Databases

- Pharmaceutical database of active chemical compounds.
- Bio Sciences Databases
  - Genomes of humans and many organisms.
  - Individual human genomes/chromosome information.
  - Protein DB.
- Astronomy databases, such as
  - sent to earth by Hubble telescope,
  - sent by satellites in various orbitals, launched by companies, countries: many purposes.
  - sent by telescope observatories world over.

Classical Databases

- Corporation databases
  - Sales-Item-Location etc. information.
  - Employee Manager Skill Salary Accounts etc. information.
- University
  - course, student, offering, grades … information.
- Banking
  - Account, Branch, types of account, financial information.
- Transportation, like Railways IRCTC, Airlines.
- Billing Records database
  - Phone companies
  - Electric companies
  - Land records: Municipalities,Tehsils.

What do classical DBMSs provide?
Data Definition Language

- Purpose of classical DBMS
  - Allows ease of building ``typical'' and ``simple'' apps.
- App developers specify schema of their data.
  - Data item structures, types of each field etc.? Similar to C, Java etc.
  - DBMS provides a data definition language. High level.
  - App developers use it to specify schema.
  - Data definition language also allows for app developers to change schema---
    - Add fields or attributes to data items/class etc.
    - Create new data types, etc….

Data Manipulation Language

- DBMS provides a data manipulation language.
- App developers use the DBMS data manipulation language.
  - to write ``queries'' using query language.
  - App users will use these queries. E.g.,
    - Amazon: Give me a list of ``formal dress shirts''.
- Data manipulation language allows data modification, e.g.
  - Debit from your bank account.
  - Enter grades of course students at end of semester,…

Classical DBMSs provide Data Manipulation Language

- App developers use the DBMS data manipulation language.
  - Allows app developers to write ``queries'' using a query language. App users will use these queries. E.g., Amazon.
    - Find popular items bought under ``Sports->Cricket''.
  - Data manipulation language also allows ``modification'' of data.
    - Insert a new data item.
    - Update certain fields of existing data item.

Durable Data

- Data used by App has a long lifetime. E.g.,
  - University academic data.
  - Banking data.
  - Reservations data for trains, airlines.
- Users may use the app and leave, but data must remain consistent over a long period.
- Data should **remain in consistent state in the presence of failures**,
  - Software, hardware failures.
  - E.g., withdrawing money from ATM and a failure happens at bank server.

Classical DBMSs
Consistency, Concurrency, Durability

- Data has a long life, and so
  - Data must be consistent.
- Concurrency: access by app users can be highly concurrent, i.e., many users are accessing (reading/writing/modifying) the database. E.g.,
  - ATM accesses by SBI customers.
  - Railway reservation.
  - …many other applications
- DBMS provides consistency of data despite competitive, conflicting access.
  - E.g., two people trying to book the same seat in a plane/train should not end up reserving the same seat.
  - Concurrency Control.

Durability

- What is durability?
- Data must remain in a consistent state even in presence of failures.
- Failures can be the app software or hardware.
- E.g. A bank account to account transfer application.
  1. Deducts say Rs 100 from Account A.
  2. Adds say Rs 100 to Account B.
- Suppose after step 1 and before step 2, there is a hardware failure.
  - The resulting DB state would be inconsistent.
  - Atomicity: Either steps 1 and 2 both happen or neither happens.
  - If DB fails after step 1, the recovery module.

ACID properties of transactions

- Group one or more database operations into a transaction.
- `A' stands for atomicity, the all-or-nothing execution of transactions.
- `C' stands for consistency. Consistency properties between data items remains enforced at all times, even with failures:
  - E.g., Referential integrity: there is no student name in a University database which has no roll no.
  - E.g., Banking: Savings balance > 100.
  - Etc.
- `I' for Isolation: Each transaction executes as if no other transaction is executing at the same time.
  - Concurrently, two transactions attempting to reserve the same seat on the same flight: both cannot succeed.
- `D' for Durability. If a transaction completes, its effects on DB are permanent.

Query Processing

- Query Processor: portion of DBMS that is designed to strongly enhance its performance.
- It is typically classified into components.

1. Query Compiler: translates the query into an internal form called a query plan.
    1. Query parser: builds a tree structure from the textual query.
    2. Query pre-processor: Checks query and translates it into an internal tree/graph representation (plan) with algebraic operators.
    3. Query Optimizer: Transforms initial query plan into an efficient plan that answers the same query correctly and efficiently.

Notes and References

- The slides are based on Chapter 1 of the book by Garcia Molina et. al.

- It is recommended to study the database system architecture diagram given at the end of Chapter 1 of this book and the explanation given for the connections between components of the database system.