DATABASE AND INFORMATION RETRIEVAL

DR. ELIEL KEELSON

LECTURE 01 – INTRODUCTION

Introduction and Background

- Most computer applications require a means of holding persistent data i.e, data that preserves its value between successive invocations of the software that produces it and between successive switch-ons of the computer itself.

- The need for persistent storage has traditionally been met by the use of magnetic storage devices such as tape and disk.

- The storage space on disks is generally controlled by ‘file management’ software, which can be part of the OS, a layer of software above the OS, or both.

Data Storage In Computers

- The basic tool provided is the ‘file’ abstraction.

- A file is a group of binary digits of arbitrary length, recorded on the disk surface, which the OS treats as a unit of storage.

- To support the use of files as a storage medium, the system also provides a naming and categorisation mechanism and reading, writing and positioning facilities.

Data Storage In Computers

- The file can have an arbitrary complex internal structure, but this is generally recognised by the application software and is not visible to the OS.

- For instance, the file may consist of purely text characters, it may be a binary image of a graphical picture.

- The OS views this as a series of bits and is not aware of the interpretation placed on these bits by the application software.

Data Storage In Computers

- The file system provides the user with the means to store programs and associated data and as such is an indispensable part of the computer.

- End-user applications such as word processing, graphic design and spreadsheets extensively utilise the file system for storage of documents, designs, etc.

The Problem With File Systems

- The structure of records is defined in the application program. This produced two consequences:
  - In order to change a file format, every program using the file had to be modified.
  - At the same time, the file had to be rebuilt in the new format using ‘one-off’ conversion programs

- The files were designed to suit the application currently being developed. When attempts were made to integrate different applications, the files were found to be incompatible

The Problem With File Systems

- Because files were created to meet the requirements of each separate application, the same data were duplicated.

- The ability to create files of arbitrary complexity made it difficult to provide generalised querying and maintenance facilities.

THE DB CONCEPT

Why Databases

The Database Concept

- The database concept arose as an attempt to solve these problems.

- Typically file-based systems concentrated on the functionality of the programs, with files being constructed to serve the persistent storage needs of these problems.

- The alternative approach- the database approach – is to look first at the design of the application’s data and then write programs to process it.

Motivation for Database Systems Application Programs

- Data Sharing: User’s share the same data
- Concurrent Access: Users’ actions do not interfere with each other
- Data Integrity: The DBMS keeps data consistent and accurate
- Program-data Independence : Data structures can change without affecting applications
- Flexible querying: The DBMS can easily deal with new queries

BASIC DB TERMS

Intro to the Database Vocabulary

Basic Definitions – Database

- A Database is a collection of data, though not just any collection counts as a database.

- It is a persistent, self-describing, structured collection of related items of data.

- It can also be described as a collection of information organized to provide efficient retrieval.

Basic Definitions – DBMS

- A Database Management System (DBMS) is system software for creating and managing databases.
- It provides users and programmers with a systematic way to create, retrieve, update and manage data.
- It stores data in such a way that makes it easier to retrieve, manipulate, and produce information.
- Examples of DBMS are MySQL, PostgreSQL, Microsoft Access, SQL Server, FileMaker, Oracle, RDBMS, dBASE, Clipper, and FoxPro

Basic Definitions – DBMS

- Since there are so many database management systems available, it is important for there to be a way for them to communicate with each other.

- For this reason, most database software comes with an Open Database Connectivity (ODBC) driver that allows the database to integrate with other databases.

- For example, common SQL statements such as SELECT and INSERT are translated from a program's proprietary syntax into a syntax other databases can understand.

Basic Definitions – Metadata

- Metadata is the description of data stored in the database.
- It summarizes basic information about data, which can make finding and working with particular instances of data easier.
- For example, *author*, *date created* and *date modified* are examples of very basic document metadata.
- Having the ability to filter through that metadata makes it much easier for someone to locate a specific document.

Basic Definitions – Database Schema

- The Database Schema is the skeletal structure that represents the logical view of the entire database.

- It defines how the data is organized and how the relations among them are associated.

- It gives an idea of all the constraints that are to be applied on the data.

- The schematic diagram of the database is often drawn to aid implementers to design the database.

Basic Definitions – Database Instance

- A Database Instance is a collection of information stored in the database at a particular moment.
- It contains a snapshot of the database.
- Database instances tend to change with time.
- A DBMS ensures that its every instance (state) is in a valid state, by diligently following all the validations, constraints, and conditions that the database designers have imposed

Basic Definitions

- Access to the database application data and to the schemas or ‘meta-data’ (i.e. the stored specification data that describes application data formats) is vested entirely in the DBMS.

- The schemas are stored within the database itself, so the database effectively contains its own description.

Basic Definitions

- Different modes of access to the DBMS are possible:
  - We can write a program in a high-level language that ‘talks’ to the DBMS to provide the required services, or
  - We can use interactive facilities provided by the user interface of the DBMS

- Additional services can be provided by specially written maintenance and utility programs.

DATABASE USERS

Category of Users Who Use Databases

- A typical DBMS has users with different rights and permissions who use it for different purposes. Some users retrieve data and some back it up. The users of a DBMS can be broadly categorized as follows:

Category of Users Who Use Databases - Administrators

- Administrators maintain the DBMS and are responsible for administrating the database.

- They are responsible to look after its usage and by whom it should be used. They create access profiles for users and apply limitations to maintain isolation and force security.

- Administrators also look after DBMS resources like system license, required tools, and other software and hardware related maintenance.

Category of Users Who Use Databases – Designers

- Designers are the group of people who actually work on the designing part of the database.
- They keep a close watch on what data should be kept and in what format.
- They identify and design the whole set of entities, relations, constraints, and views.

Category of Users Who Use Databases – End Users

- End users are those who actually reap the benefits of having a DBMS.

- End users can range from simple viewers who pay attention to the logs or market rates to sophisticated users such as business analysts.

DATABASE ARCHITECTURE

The Database Architecture

- The design of a DBMS depends on its architecture.
- The architecture of a DBMS can be seen as either single tier or multi-tier.
- An n-tier architecture divides the whole system into related but independent ***n*** modules, which can be independently modified, altered, changed, or replaced.

The Database Architecture – 1 Tier

- In 1-tier architecture, the DBMS is the only entity where the user directly sits on the DBMS and uses it.
- Any changes done here will directly be done on the DBMS itself.
- It does not provide handy tools for end-users. Database designers and programmers normally prefer to use single-tier architecture

The Database Architecture – 2 Tier

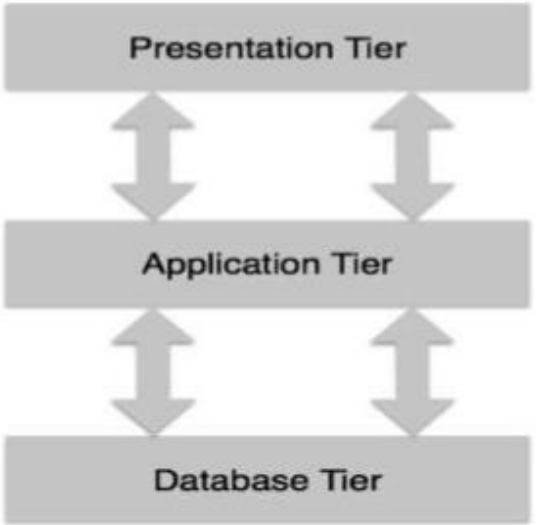
- If the architecture of DBMS is 2-tier, then it must have an application through which the DBMS can be accessed.

- Programmers use 2-tier architecture where they access the DBMS by means of an application.

- Here the application tier is entirely independent of the database in terms of operation, design, and programming.

The Database Architecture – 3 Tier

- A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.



The Database Architecture – 3 Tier – Database Tier

- At this tier, the database resides along with its query processing languages.

- We also have the relations that define the data and their constraints at this level.

The Database Architecture – 3 Tier – Application Tier

- This tier resides the application server and the programs that access the database.

- For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database (structure) beyond the application.

- At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.

The Database Architecture – 3 Tier – Presentation Tier

- End-users operate on this tier and they know nothing about any existence of the database beyond this layer.

- At this layer, multiple views of the database can be provided by the application.

- All views are generated by applications that reside in the application tier.

DATA MODELLING

Introduction to Data Models

Data Modelling

- In general usage, a model of some ‘real’ system is another representation that shares certain relevant features with the real system.

- A model can be a set of equations, an actual physical scale model, a computer program, etc.

- Models are useful in that the characteristics of the real system can be analysed by studying the nature and behaviour of the model.

Data Modelling

- In the study of databases, the interest is in data modelling.

- The role of data modelling is to provide techniques that allow us to represent, by graphical and other formal methods, the nature of data in real-world computer applications.

Data Modelling

The data models that are of interest are:

- The Hierarchical Model
- The Network Model
- The Relational Model
- The Object-Oriented Model
- The Entity-Relationship (ER) Model

Data Modelling

- The Hierarchical Model formed the basis of the earliest databases and, as the name implies, organised data in a hierarchical structure.

- The Network Model followed the hierarchical model and solved some of the problems of that model.

- The Relational Model (RM) was first introduced by Codd in 1970 and forms the basis of most current database management systems.

Data Modelling

- The Object-Oriented (OO) model has become more prominent in recent years, particularly in the field of programming and systems development.

- The Entity-Relationship (ER) model was devised by Chen in 1976. It is a diagrammatic technique that provides a generalised approach to the representation of data and which is particularly suitable and helpful in the design of relational database systems.

DATABASE APPLICATIONS

Places Where DB’s Are Needed

Areas of Application

- Databases today are essential to every business. Here are some representative applications:

- **Banking:** For customer information, accounts, and loans, and banking transactions.

- **Airlines:** For reservations and schedule information. Airlines were among the first to use databases in a geographically distributed manner—terminals situated around the world accessed the central database system through phone lines and other data networks.

Areas of Application

- **Universities:** For student information, course registrations, and grades.
- **Credit Card Transactions:** For purchases on credit cards and generation of monthly statements.
- **Finance:** For storing information about holdings, sales, and purchases of financial instruments such as stocks and bonds.
- **Telecommunication:** For keeping records of calls made, generating monthly bills, maintaining balances on prepaid calling cards, and storing information about the communication networks.

Areas of Application

- **Sales:** For customer, product, and purchase information.

- **Manufacturing:** For management of supply chain and for tracking production of items in factories, inventories of items in warehouses/stores, and orders for items.

THANKS!

Any questions?

You can find me at elielkeelson@gmail.com &
ekeelson@knust.edu.gh