Principles of Database Design

Dr CK Raju

Muthoot Institute of Technology and Science
(Affiliated to KTU)

rajuck@mgmits.ac.in

February 6, 2019

Overview

Objectives

- To impart basic understanding of the theory and applications of database management systems.
- To give basic level understanding of internals of database systems.
- To expose to some of the recent trends in databases.

Syllabus

- Types of data, database and DBMS, Languages and users.
- Software Architecture, E-R and Extended E-R Modelling,
- Relational Model concepts and languages, relational algebra and tuple relational calculus,
- SQL, views, assertions and triggers, HLL interfaces,
- Relational db design, FDs and normal forms,
- Secondary storage organization, indexing and hashing,
- Query optimization, concurrent transaction processing
- Recovery principles, recent topics.

Outcomes: Students would be able to

- define, explain and illustrate the fundamental concepts of databases,
- construct an Entity-Relationship (E-R) model from specifications and to perform the transformation of the conceptual model into corresponding logical data structures,
- model and design a relational database following the design principles,
- develop queries for relational database in the context of practical applications,
- define, explain and illustrate fundamental principles of data organization, query optimization and concurrent transaction processing,
- appreciate the latest trends in databases.

Modules

- Module I - Introduction, Entity-Relationship Model
- Module II - Relational Model, Database Languages
- Module III - SQL, Views-assertions-triggers, Functions-Procedures-HLL-Interfaces
- Module IV - Relational Database Design
- Module V - Physical Data Organisation, Query Optimization
- Module VI - Transaction Processing Concepts, Recent Topics

Introduction

- Data: structured, semi-structured and unstructured
- DBMS: Concept and Overview
- Data Models
- Database Languages
- DB Admin, DB Users
- DBMS: three schema architecture
- DB architectures and classification

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Concepts

- Data : Facts that can be stored, processed and also possess some implicit meaning
- Databse : Collection of related data, representing some aspect of real world, univ-of-discourse
- Databases are designed, built and populated with data for some specific purpose
- Manual databases exist in some libraries
- DBMS is a software system used to create and maintain a database
- Meta-data : Database definition or descriptive database catalog information
- DBMSs help in designing, creating, manipulating and sharing of databases

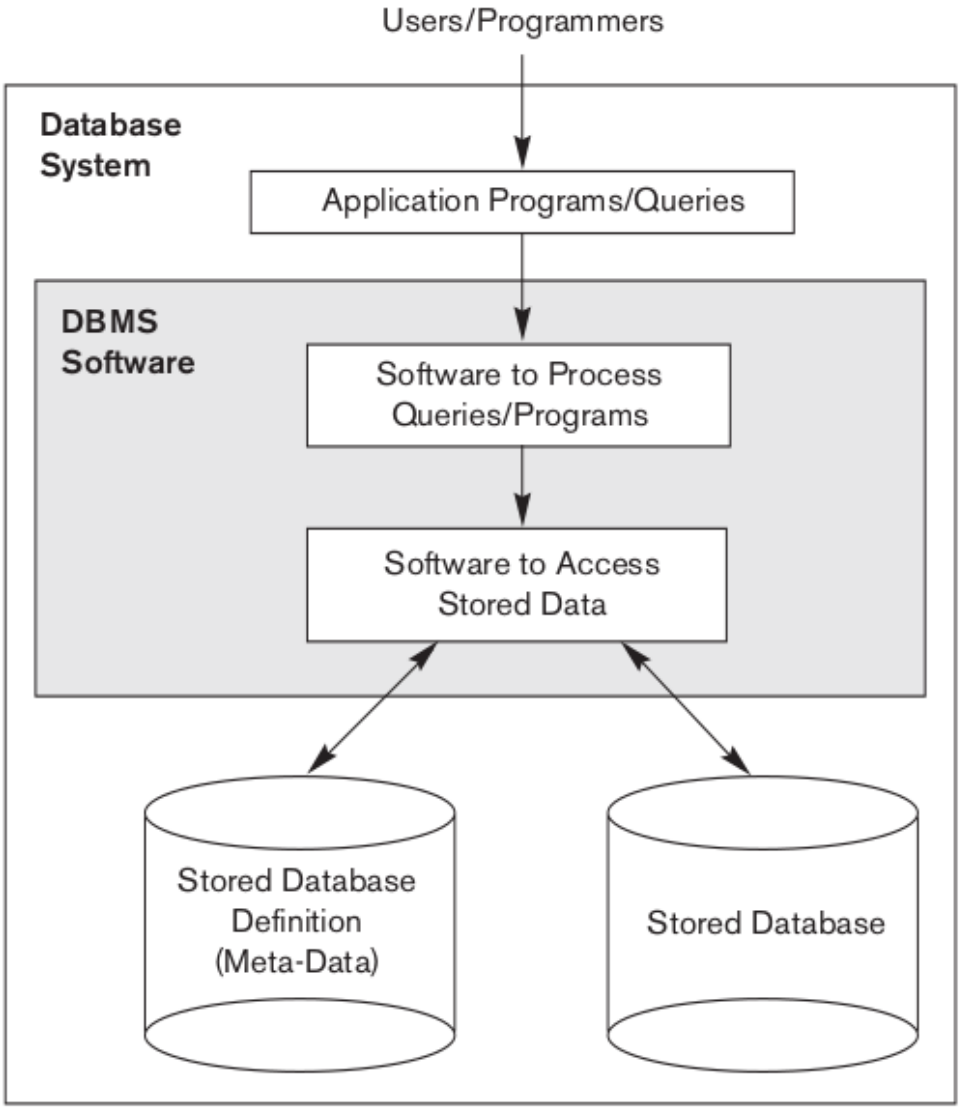
Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Concepts

- DB Query : S/w program that queries database for some information.
- DB Transaction : An activity that reads or manipulates data in a database.
- Protection : System protection and security protection
- System protection - against h/w, s/w malfunction; Security - against unauthorised access

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

A typical Database environment



A student database having course information

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Approach

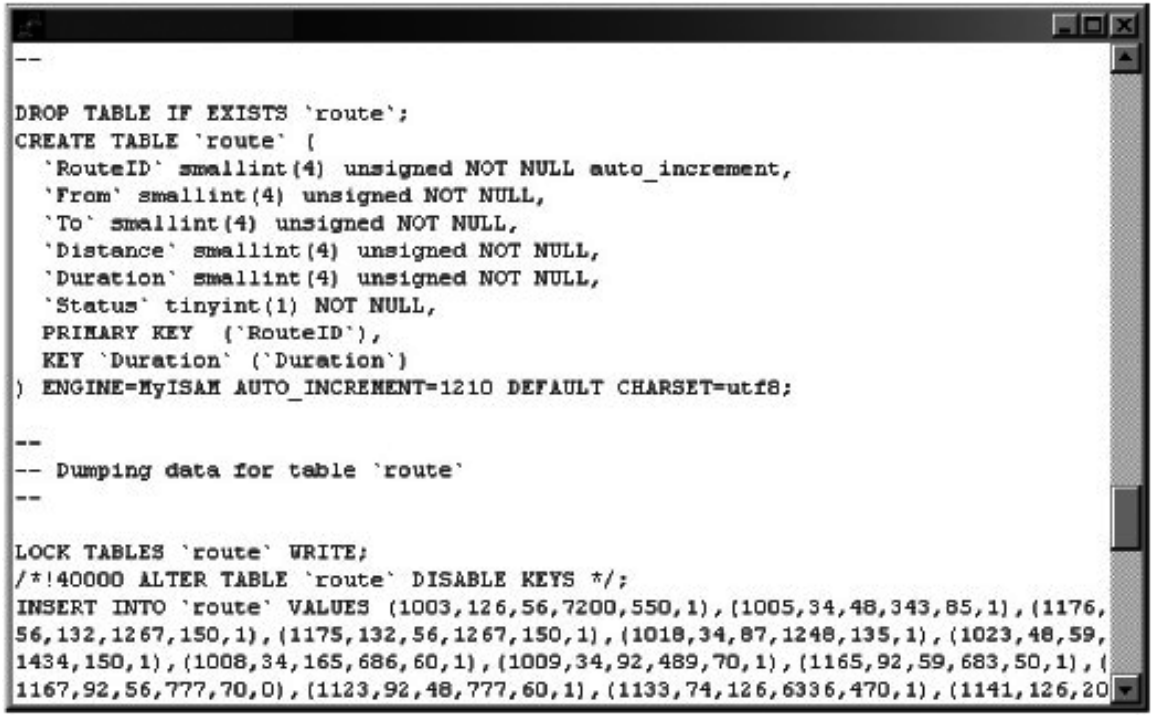
Special Features

- Self-describing notion of database system - meta-data
- Insulation between programs, data and data abstraction
- Support of multiple **views** of data
- Sharing of data and multiuser transaction processing

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Approach

- Self-describing notion of database system - meta-data



```
--

DROP TABLE IF EXISTS `route`;
CREATE TABLE `route` (
  `RouteID` smallint(4) unsigned NOT NULL auto_increment,
  `From` smallint(4) unsigned NOT NULL,
  `To` smallint(4) unsigned NOT NULL,
  `Distance` smallint(4) unsigned NOT NULL,
  `Duration` smallint(4) unsigned NOT NULL,
  `Status` tinyint(1) NOT NULL,
  PRIMARY KEY  (`RouteID`),
  KEY `Duration` (`Duration`)
) ENGINE=MyISAM AUTO_INCREMENT=1210 DEFAULT CHARSET=utf8;

--
-- Dumping data for table `route`
--

LOCK TABLES `route` WRITE;
/*!40000 ALTER TABLE `route` DISABLE KEYS */;
INSERT INTO `route` VALUES (1003,126,56,7200,550,1),(1005,34,48,343,85,1),(1176,
56,132,1267,150,1),(1175,132,56,1267,150,1),(1018,34,87,1248,135,1),(1023,48,59,
1434,150,1),(1008,34,165,686,60,1),(1009,34,92,489,70,1),(1165,92,59,683,50,1),(
1167,92,56,777,70,0),(1123,92,48,777,60,1),(1133,74,126,6336,470,1),(1141,126,20
```

- Insulation between programs, data and data abstraction
- Support of multiple **views** of data
- Sharing of data and multiuser transaction processing

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Approach

- Self-describing notion of database system - meta-data
- Insulation between programs, data and data abstraction
  1. Program-data independence - independence of data from programs
  2. Program-operation independence
  3. Characteristic of data abstraction - program-data and program-operation independence
- Support of multiple **views** of data
- Sharing of data and multiuser transaction processing

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Approach

- Self-describing notion of database system - meta-data
- Insulation between programs, data and data abstraction
- Support of multiple **views** of data
  1. Special perspective of database
  2. Might contain subset of database in conjunction with other information
- Sharing of data and multiuser transaction processing

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Approach

- Self-describing notion of database system - meta-data
- Insulation between programs, data and data abstraction
- Support of multiple **views** of data
- Sharing of data and multiuser transaction processing
  1. Multi-user DBMS involves **concurrency control**
  2. Special **online transaction processing** applications
  3. Process **isolation** amongst different processes
  4. Ensure **atomicity** property of transaction - either whole or none

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Users

Users may be divided into

- **Actors on the Scene** - Those who actually use and control the database content, and those who design, develop and maintain database applications
- **Workers Behind the Scene** - Those who design and develop the DBMS software and related tools, and the computer systems operators

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Actors on the scene

- **Database Administrators** - Responsible for authorizing access to the database, for coordinating and monitoring its use, acquiring software and hardware resources, controlling its use and monitoring efficiency of operations.
- **Database Designers** - Responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

End Users : Actors on the Scene

- **Casual** - those who access database occasionally when needed
- **Naive or Parametric**
- **Sophisticated**
- **Standalone**

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

End Users : Actors on the Scene

- **Casual**
- **Naive or Parametric** - they make up a large section of the end-user population. They use previously well-defined functions in the form of *canned transactions* against the database. Examples are bank-tellers or reservation clerks who do this activity for an entire shift of operations.
- **Sophisticated**
- **Standalone**

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

End Users : Actors on the Scene

- **Casual**
- **Naive or Parametric**
- **Sophisticated** - These include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities. Many use tools in the form of software packages that work closely with the stored database.
- **Standalone**

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

End Users : Actors on the Scene

- **Casual**
- **Naive or Parametric**
- **Sophisticated**
- **Standalone** - Mostly maintain personal databases using ready-to-use packaged applications. An example is a tax program user that creates its own internal database. Another example is a user that maintains an address book.

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Approach - Advantages

1. Controlling redundancy in data storage and in development and maintenance efforts. Sharing of data among multiple users.
2. Restricting unauthorized access to data.
3. Providing persistent storage for program Objects - in Object-oriented DBMSes
4. Providing Storage Structures (e.g. indexes) for efficient Query Processing

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Approach - Advantages continued

1. **Potential for enforcing standards:** - This is very crucial for the success of database applications in large organizations. Standards refer to data item names, display formats, screens, report structures, meta-data (description of data), Web page layouts, etc.
2. **Reduced application development time:** - Incremental time to add each new application is reduced.

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Approach - Implications

1. Providing backup and recovery services.
2. Providing multiple interfaces to different classes of users.
3. Representing complex relationships among data.
4. Enforcing integrity constraints on the database.
5. Drawing inferences and actions from the stored data using deductive and active rules

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Approach - Implications continued

1. **Flexibility to change data structures:** - Database structure may evolve as new requirements are defined.
2. **Availability of current information:** - Extremely important for on-line transaction systems such as airline, hotel, car reservations.
3. **Economies of scale:** - Wasteful overlap of resources and personnel can be avoided by consolidating data and applications across departments.

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Models

1. **Early Database Applications:** - The Hierarchical and Network Models were introduced in mid 1960s and dominated during the seventies.A bulk of the worldwide database processing still occurs using these models, particularly, the hierarchical model.
2. **Relational Model based Systems**
3. **Object-oriented**
4. **Others**

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Models

1. **Early Database Applications**
2. **Relational Model based Systems:** - Relational model was originally introduced in 1970, was heavily researched and experimented within IBM Research and several universities.Relational DBMS Products emerged in the early 1980s.
3. **Object-oriented and others**
4. **Others**

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Models

1. **Early Database Applications**
2. **Relational Model based Systems**
3. **Object-oriented Models** - Object-Oriented Database Management Systems (OODBMSs) were introduced in late 1980s and early 1990s to cater to the need of complex data processing in CAD and other applications. Their use has not taken off much. Many relational DBMSs have incorporated object database concepts, leading to a new category called object-relational DBMSs (ORDBMSs)
4. **Others**

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Database Models

1. **Early Database Applications**
2. **Relational Model based Systems**
3. **Object-oriented Models**
4. **Others:** - Extended relational systems add further capabilities (e.g. for multimedia data, XML, and other data types.) Web contains data in HTML (Hypertext markup language) with links among pages. This has given rise to a new set of applications and E-commerce is using new standards like XML (eXtended Markup Language). Script programming languages such as PHP and JavaScript allow generation of dynamic Web pages that are partially generated from a database. Also allow database updates through Web pages.

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

Extending DB Capabilities

Additional functionalities are getting added to DBMSes to cater to new needs of

1. Scientific Applications
2. XMS (eXtensible Markup Language)
3. Image Storage and Management
4. Audio and Video Data Management
5. Data Warehousing and Data Mining
6. Spatial Data Management
7. Time Series and Historical Data Management

This calls for new R and D to support newer data types, complex data structures, new functions, storage and indexing schemes etc.

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1

When NOT to use DBMS

1. Some inhibitors - high investment in infrastructure and people
2. When database application is simple, where changes are less frequent
3. If stringent real-time requirements are present
4. If access to multiple users is not required
5. If system is too complex to be modeled to a database system
6. If database users need special operations, unsupported by DBMS

Ref: Elmasri - Ch-1 and Ch-2,
Korth - Ch-1