Nadar Saraswathi College of Engineering & Technology

NSCET E-LEARNING PRESENTATION

LISTEN … LEARN… LEAD…

COMPUTER SCIENCE AND ENGINEERING

II YEAR / IV SEMESTER

CS8492 – DATABASE MANAGEMENT SYSTEMS

A.AMBIKA,M.E,MISTE
ASSISTANT PROFESSOR
Nadar Saraswathi College of Engineering & Technology,
Vadapudupatti, Annanji (po), Theni – 625531.

UNIT- I

RELATIONAL DATABASES

Nadar Saraswathi College of Engineering & Technology

Contents

- Purpose of Database System
- Views of data
- Data Models
- Database system Architecture
- Introduction to relational databases
- Relational Model
- Keys
- Relational Algebra
- SQL fundamentals
- Advanced SQL features
- Embedded SQL
- Dynamic SQL.

INTRODUCTION

- A Database Management System(DBMS) is a collection of interrelated data and various programs that are used to handle that data.
- The primary goal of DBMS is to prove a way to store and retrieve tha required information from the database in convenient and efficient manner.

INTRODUCTION

- Two important tasks:

i)Define the structure for storage information

ii)Provide Mechanism for manipulation of information.

Database System Applications:

Accounting,Manufacturing,Banking,Universities ,Reservation Systems,Telecommunication etc.

PURPOSE OF DATABASE SYSTEM

- The typical file processing system is supported by a conventional operating system.
- The system stores permanent records in various files, and it needs different application programs to extract records from, and add records to, the appropriate files.
- A file processing system has a number of major disadvantages.

PURPOSE OF DATABASE SYSTEM

Data redundancy and inconsistency:

In file processing, every user group maintains its own files for handling its data processing applications.

Data isolation:

Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult.

PURPOSE OF DATABASE SYSTEM

Integrity problems:

The data values stored in the database must satisfy certain types of consistency constraints.

Atomicity problems:

Atomic means the transaction must happen in its entirety or not at all. It is difficult to ensure atomicity in a conventional file processing system.

PURPOSE OF DATABASE SYSTEM

Concurrent access anomalies:

For the sake of overall performance of the system and faster response, many systems allow multiple users to update the data simultaneously.

Security problems:

Enforcing security constraints to the file processing system is difficult.

VIEWS OF DATA

A major purpose of a database system is to provide users with an abstract view of the data.
(i.e) The system hides certain details of how the data are stored and maintained.

- **External level / logical level:**

External level The users view of the database External level describes that part of the database that is relevant to each user.

VIEWS OF DATA

Conceptual level

It describes what data is stored in the database and the relationships among the data.

- The constraints on the data
- Semantic information about the data
- Security and integrity information

VIEWS OF DATA

Internal level / physical level

Internal level describes how the data is stored in the database.

- Storage space allocation for data and indexes.
- Record descriptions for storage.
- Record placement.
- Data compression and data encryption techniques.
- Below the internal level there is a physical level that may be managed by the operating system under the direction of the DBMS

INSTANCES AND SCHEMAS

Schema is the logical structure of the database E.g., The database consists of information about a set of customers and accounts and the relationship between them) analogous to type information of a variable in a program.

Physical schema: database design at the physical level.

Logical schema: database design at the logical level.

DATA MODELS

The data model is a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.

1.Record-based data models

The database consists of a number of fixed format records possibly of differing types.

There are three types of record-based logical data model.

- Hierarchical data model
- Network data model
- Relational data model

DATA MODELS

2.Object-based data models

Object-based data models use concepts such as entities, attributes, and relationships.An entity is a distinct object in the organization that is to be represents in the database.

3.Physical-data models.

The first two are used to describe data at the conceptual and external levels, the latter is used to describe data at the internal level.

DATABASE SYSTEM ARCHITECTURE

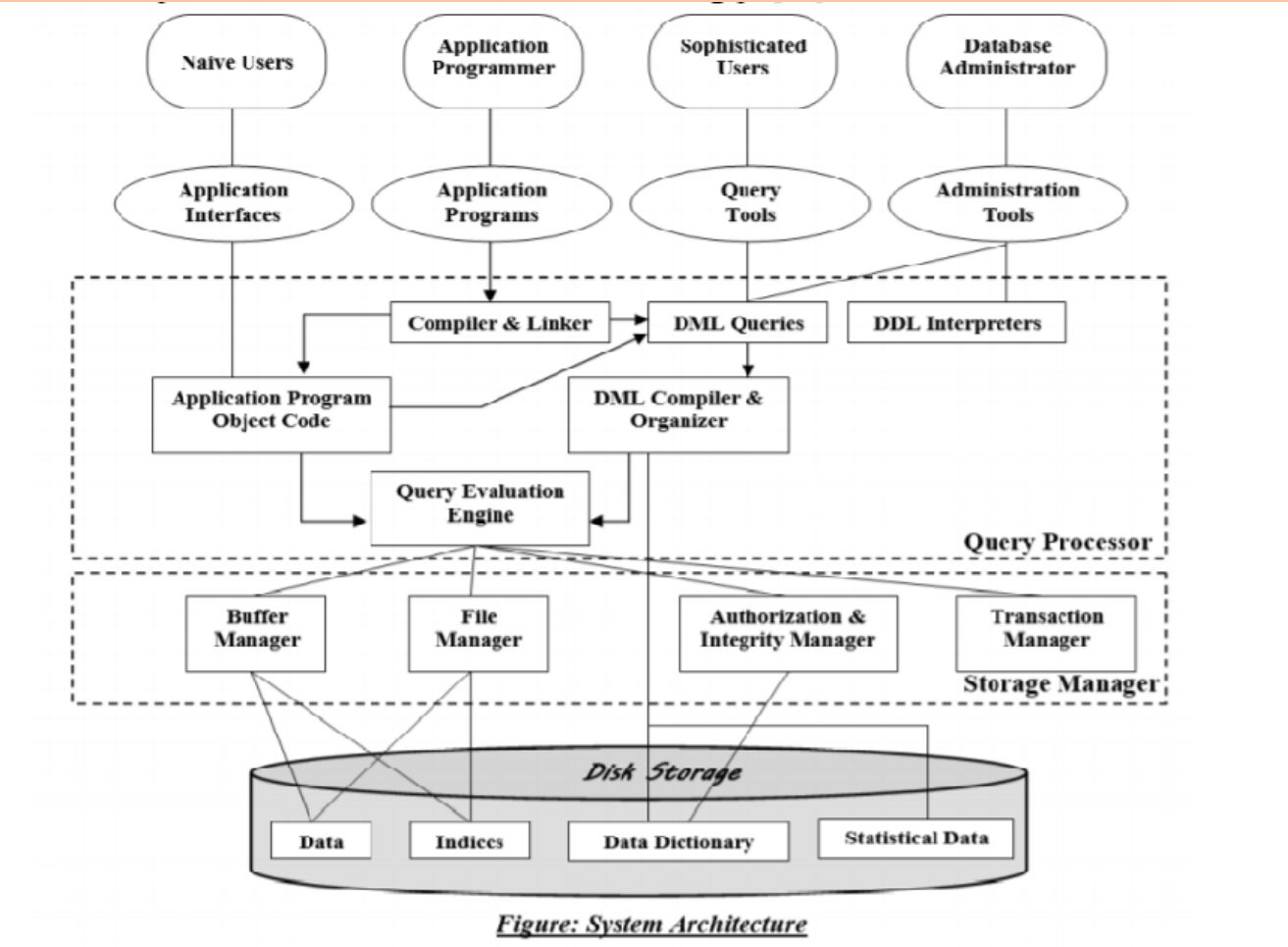


Figure: System Architecture

DATABASE SYSTEM ARCHITECTURE

Transaction Management:

A transaction is a collection of operations that performs a single logical function in a database application.

Storage Management:

A storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.

The storage manager is responsible for the Interaction with the file manager.

DATABASE SYSTEM ARCHITECTURE

Database Users:

- **Application programmers:-**Interact with system through DML calls.
- **Sophisticated users**:-Form requests in a database query language
- **Specialized users:–** Write specialized database applications that do not fit into the traditional data processing framework.

DATABASE SYSTEM ARCHITECTURE

Naive users:-Invoke one of the permanent application programs that have been written previously.

Database Administrator

- Coordinates all the activities of the database system
- Storage structure and access method definition
- Schema definition and physical organization modification

DATABASE SYSTEM ARCHITECTURE

- Granting user authority to access the database
- Specifying integrity constraints
- Acting as liaison with users
- Monitoring performance and responding to changes in requirements

File manager

Manages allocation of disk space and data structures used to represent information on disk.

.

DATABASE SYSTEM ARCHITECTURE

Database manager

The interface between low level data and application programs and queries.

Query processor

Translates statements in a query language into low-level instructions the database manager understands. (May also attempt to find an equivalent but more efficient form.)

DATABASE SYSTEM ARCHITECTURE

DML precompiler:

Converts DML statements embedded in an application program to normal procedure calls in a host language. The precompiler interacts with the query processor

DATABASE SYSTEM ARCHITECTURE

DDL compiler

converts DDL statements to a set of tables containing metadata stored in a data dictionary. In addition, several data structures are required for physical system implementation.

Data files:

store the database itself.

DATABASE SYSTEM ARCHITECTURE

Data dictionary:

stores information about the structure of the database. It is used heavily. Great emphasis should be placed on developing a good design and efficient implementation of the dictionary.

Indices:

Provide fast access to data items holding particular values

RELATIONAL DATABASES

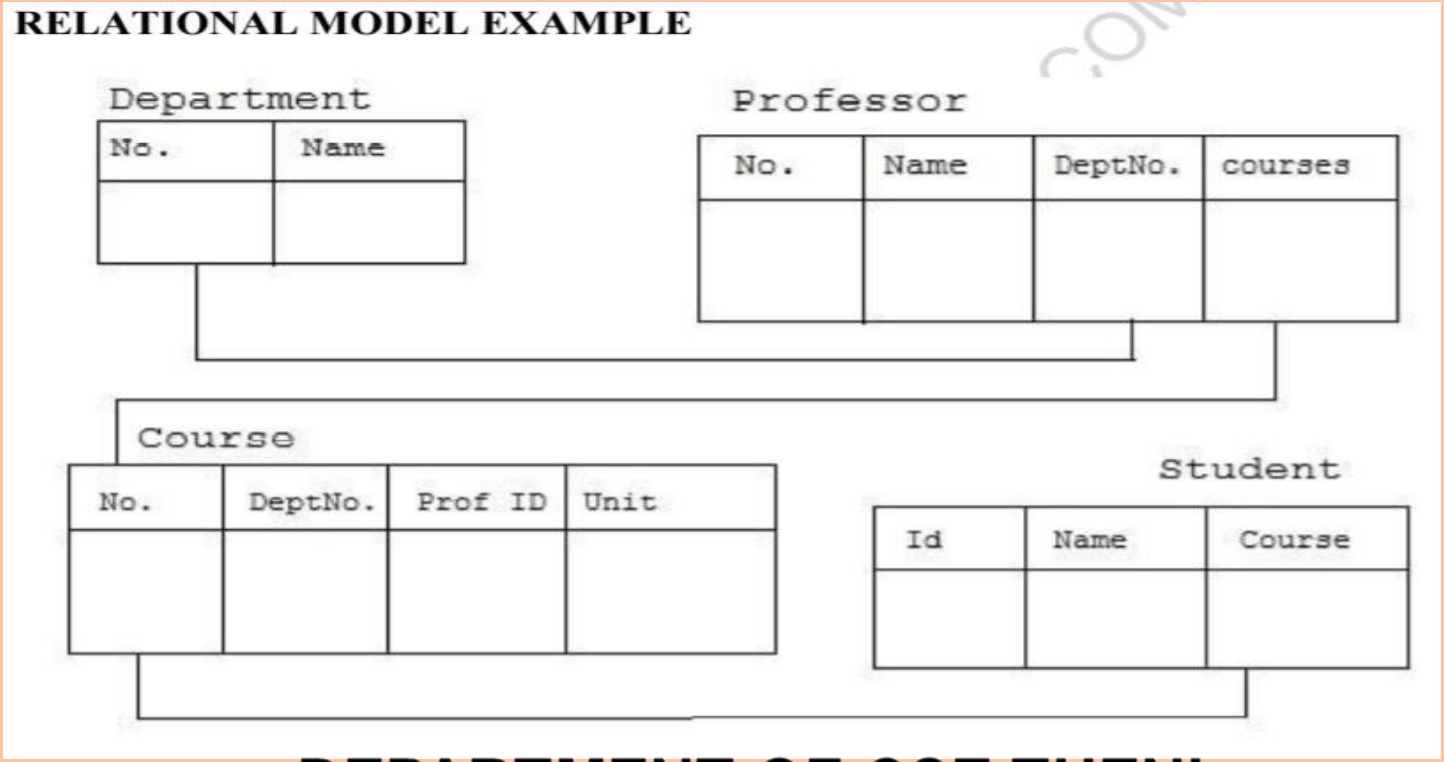
A relational database is based on the relational model and uses a collection of tables to represent both data and the relationships among those data.

Every relation value has two pairs

1)A set of column-name: type-name pairs.

2)A set of rows

RELATIONAL DATABASES



Nadar Saraswathi College of Engineering & Technology

DEPARTMENT OF CSE,THENI

Keys

- DBMS has Five types of Keys in it and they all has different functionality.
- The Keys are as follows:
  - Super Key
  - Candidate Key
  - Primary Key
  - Foreign Key
  - Composite Key

Keys

- Super Key is a set of attributes whose set of values can uniquely identify an entity instance in the entity set.
- A **Super Key** can contains one or more than one attributes.
- **Super Key** is the broadest definition of unique identifiers of an entity in an entity set.

Keys

- **Candidate key** is a set of one or more attributes whose set of values can uniquely identify an entity instance in the entity set.
- Any attribute in the candidate key cannot be omitted without destroying the uniqueness property of the **Candidate key**.
- It is minimal **Super Key**.

Nadar Saraswathi College of Engineering & Technology

Keys

- The Primary Key is an attribute or a set of attributes that instance of an entity.
- Every entity in the data model must have a primary key whose values uniquely identify instances of the entity.

Keys- Foreign key

- Every dependent and category (subtype) entity in the model must have a foreign key for each relationship in which it participates.
- Foreign keys are formed in dependent and subtype entities by migrating the entire primary key from the parent or generic entity. If the primary key is composite, it may not be split

Keys-Composite key

- When a primary key is created from a combination of 2 or more columns, the primary key is called a composite key.
- Each column may not be unique by itself within the database table but when combined with the other column(s) in the composite key, the combination is unique.

RELATIONAL ALGEBRA

Let E1and E2be relational-algebra expressions;

The following are all relational-algebra expressions:

- E1nE2
- E1- E2
- E1x E2
- p(E1), Pis a predicate on attributes in E1

RELATIONAL ALGEBRA

- s(E1), Sis a list consisting of some of the attributes in E1
- x (E1), x is the new name for the result of E1.

Selection (or Restriction) (σ):

The selection operation works on a single relation R and defines a relation that contains only those tuples of R that satisfy the specified condition (predicate).

Syntax:

σ Predicate

RELATIONAL ALGEBRA

Example:

List all staff with a salary greater than 10000.

Sol: salary > 10000 (Staff).

The input relation is staff and the predicate is salary>10000.

The selection operation defines a relation containing only those staff tuples with a salary greater than 10000.

RELATIONAL ALGEBRA

Projection (π):

The projection operation works on a single relation R and defines a relation that contains a vertical subset of R, extracting the values of specified attributes and eliminating duplicates.

Syntax: π al,. an(R)

Example:

Produce a list of salaries for all staff, showing only the staffNo, name and salary. Π staffNo.Name, Salary (Staff).

Nadar Saraswathi College of Engineering & Technology

RELATIONAL ALGEBRA

Rename (ρ):

Rename operation can rename either the relation name or the attribute names or both

Syntax:

ρs (BI.B2,.Bn) (R) Or ρs(R) Or p (B1.B2Bn) (R).

S is the new relation name and B1, B2,.Bn are the new attribute names.

The first expression renames both the relation and its attributes, the second renames the relation only, and the third renames the attributes only.

RELATIONAL ALGEBRA

Union:

The union of two relations R and S defines a relation that contains all the tuples of R or S or both R and S, duplicate tuples being eliminated. Union is possible only if the schemas of the two relations match.

Syntax: R U S

Example:

List all cities where there is either a branch office or a propertyforRent.

πCity (Branch) U π civ(propertyforRent)

RELATIONAL ALGEBRA

Set difference:

The set difference operation defines a relation consisting of the tuples that are in relation R, but not in S. R and S must be union-compatible.

Syntax :R-S

Example:

List all cities where there is a branch office but no properties for rent.

Soln:

Π city (Branch) –π city(propertyforRent).

RELATIONAL ALGEBRA

Intersection:

The intersection operation defines a relation consisting of the set of all tuples that are in both R and S. R and S must be union compatible.

Syntax:R ∩ S

Example : List all cities where there is both a branch office and at least one propertyforRent. πciity (Branch) ∩ πCjty (propertyforRent)

RELATIONAL ALGEBRA

Cartesian product:

The Cartesian product operation defines a relation that is the concatenation of every tuple of relation R with every tuple of relation S.

Syntax:

R X S

RELATIONAL ALGEBRA

Example:

List the names and comments of all clients who have viewed a property for Rent.

Soln:

The names of clients are held in the client relation and the details of viewings are held in the viewing relation. To obtain the list of clients and the comments on properties they have viewed, we need to combine two relations.

SQL FUNDAMENTALS
DATABASE LANGUAGES

The Data Manipulation Language (DML):

Manipulation operations usually include the following:

- Insertion of new data into the database
- Modification of data stored in the database
- Retrieval of data contained in the database
- Deletion of data from the database

SQL FUNDAMENTALS
DATABASE LANGUAGES

Advantages of SQL:

- Increased acceptance and availability of SQL.
- Applications written in SQL can be easily ported across systems.
- SQL as a language is independent of the way it is implemented internally.
- Simple and easy to learn.
- Set-at-a-time feature of the SQL makes it increasingly powerful than the record -at-a-time processing technique.

SQL FUNDAMENTALS
DATABASE LANGUAGES

SQL data types:

SQL supports the following data types.

- CHAR(n) -fixed length string of exactlyV characters.
- VARCHAR(n) -varying length string whose maximum length is 'n' characters.
- FLOAT -floating point number.

TYPES OF SQL COMMANDS

- **Data Definition Language (DDL):-** Used to create, alter and delete database objects.
- **Data Manipulation Language (DML):-** Used to insert, modify and delete the data in the database.

TYPES OF SQL COMMANDS

Data Query Language (DQL): Enables the users to query one or more tables to get the information they want.

Data Control Language (DCL): Controls the user access to the database objectsments.

SQL Operators

Arithmetic operators:-are used to add,subtract,multiply, divide and negate data value (+, -, *,/).

Comparison operators: -are used to compare one expression with another.

Some comparison operators are =, >, >=, <=, IN, ANY, ALL, SOME, BETWEEN, EXISTS, and so on.

SQL Operators

Logical operators:-are used to produce a single result from combining the two separate conditions. The logical operators are AND, OR and NOT.

Set operators:-Combine the results of two separate queries into a single result. The set operators are UNION, UNIONALL, INTERSECT, MINUS and so on.

SQL QUERIES

Creat e table:

The create table statement creates a new base table.

Syntax:

Create table table-name (col 1 –definition,[col2-definition]... [,coln-definition][primary- key-definition] [,alternate-key-definition][,foreignkey-definition]);

SQL QUERIES

Example:

SQL>create table Book(ISBN char(10) not null,Title char(30) not null with default, Author char(30) not null with default,Publisher char(30) not null with default,Year integer not null with default,Price integer null,Primary key (ISBN));
Table created

SQL QUERIES

Drop table:

An existing base table can be deleted at any time by using the drop table statement.

Syntax:

Drop table table-name;

Table dropped.

This command will delete the table named book along with its contents, indexes and any views defined for that table

SQL QUERIES

DESC:

Desc command used to view the structure of the table.

Syntax:

Desc table-name;

Example :

SQL>Desc book;

Truncate table:

If there is no further use of records stored in a table and the structure has to be retained then the records alone can be deleted.

SQL QUERIES

Syntax:

Truncate table tablename;

Example:

SQL>Truncate table book;

Table truncated.

This command would delete all the records from the table, book.

TRIGGER

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database.

The trigger is mostly used for maintaining the integrity of the information on the database.

Example:

when a new record is added to the employees table, new records should also be created in the tables of the taxes, vacations and salaries.

TRIGGER

Triggers are for

- Customization of database management;
- centralization of some business or validation rules;
- logging and audit.
- Overcome the mutating-table error w records should also be created in the tables of the taxes, vacations and salaries,etc…

TRIGGER

Security Authorization:

Forms of authorization on **parts of the database**:

Read - allows reading, but not modification of dat

Insert - allows insertion of new data, but not modification of existing data.

Update - allows modification, but not deletion of data.

Delete - allows deletion of data.

Forms of authorization **to modify the database schema**.

TRIGGER

Index - allows creation and deletion of indices.
Resources - allows creation of new relations.
Alteration - allows addition or deletion of attributes in a relation.

EMBEDDED SQL

The SQL standard defines embeddings of SQL in a variety of programming languages such as C,Java, and Cobol.

Embedded is referred to as a host language, and the SQL structures permitted in the host language comprise embedded SQL.

The basic form of these languages follows that of the System R embedding of SQL into PL/I.

EMBEDDED SQL

Specify the query in SQL and declare a cursor for it

```
EXEC SQL
    declare c cursor for select depositor.customer_name,
    customer_city from depositor, customer, account where
    depositor.customer_name = customer.customer_name and
    depositor account_number = account.account_number and
    account.balance > :amount
```

DYNAMIC SQL

Allows programs to construct and submit SQL queries at run time.

Example :

```
ar * sqlprog = “update account set balance = balance *
1.05 where account_number = ?”
EXEC SQL prepare dynprog from :sqlprog;char account [10] =
    “A-101”;
EXEC SQL execute dynprog using :account;
```

The dynamic SQL program contains a ?, which is a place holder for a value that is provided when the SQL program is executed.

DYNAMIC SQL

JDBC and ODBC API :

(Application-program interface) for a program to interact with a database server.

Application makes calls to Connect with the database server Send SQL commands to the database server.

Fetch tuples of result one-by-one into program variables. ODBC (Open Database Connectivity) works with C, C++, C#, and Visual Basic JDBC (Java Database Connectivity) works With Java.

THANK YOU