DATABASE (1)

3RD CLASS

COMPUTER SCIENCE DEPARTMENT

1st Lecture – Introduction to Database:

Sunday 22nd of September 2024

LECTURER :

DR. RAYAN YOUSIF ALKHAYAT

DATABASE

References :

- **Modern Database Management Systems** ,Fred R. McFadden, 10th ed , Addison –Wesly , 2015
- **Database system concepts,** by Silberschatz, Korth and Sudarshan, 7th ed, McGraw-Hill, 2019

Introduction:

Concept of a Database

- A **database** is a collection of data and a set of rules that organize the data by specifying certain relationships among the data.
-
- Through these rules, the user describes a logical format for the data. The data items are stored in a file, but the precise physical format of the file is of no concern to the user. A database administrator is a person who defines the rules that organize the data and also controls who should have access to what parts of the data. The user interacts with the database through a program called a database manager or a database management system (DBMS), informally known as a front end.

Introduction:

Database Management System (DBMS):

- The DBMS is the application that manages the data included within the database, it contains information about a particular enterprise, it provides an environment that is both convenient and efficient to use.
- DBMS consists of :
- 1. Collection of interrelated data
- 2. Set of programs to access the data .

File System :

- In the early days, database applications were built on top of file systems which has many drawbacks to store and manipulate data such as:
- 1.	Data redundancy and inconsistency
- -Multiple file formats, duplication of information in different files
- 2.	Difficulty in accessing data .
- -Need to write a new program to carry out each new task
- 3.	Data isolation — multiple files and formats
- 4.	Integrity problems
- •	Integrity constraints (e.g. account balance > 0) become part of program code
- •	Hard to add new constraints or change existing ones

File System :

- 5. Atomicity of updates
- • Failures may leave database in an inconsistent state with partial updates carried out.
- E.g. transfer of funds from one account to another should either complete or not happen at all
- 6. Concurrent access by multiple users
- • Concurrent accessed needed for performance
- • Uncontrolled concurrent accesses can lead to inconsistencies.
- E.g. two people reading a balance and updating it at the same time
- 7. Security problems

Advantages of Using Databases

- The logical idea behind a database is this:

- A database is a single collection of data, stored and maintained at one central location, to which many people have access as needed.

- The essence of a good database is that the users are unaware of the physical arrangements; the unified logical arrangement is all they see.

Advantages of Using Databases

1. shared access, so that many users can use one common, centralized set of data
2. minimal redundancy, so that individual users do not have to collect and maintain their own sets of data
3. data consistency, so that a change to a data value affects all users of the data value
4. data integrity, so that data values are protected against accidental or malicious undesirable changes
5. controlled access, so that only authorized users are allowed to view or to modify data values

Database Applications: .

1)Banking: all transactions

2)Airlines: reservations, schedules

3)Universities: registration, grades

4)Sales: customers, products, purchases

5)Manufacturing: production, inventory, orders, supply chain

6)Human resources: employee records, salaries, tax deductions

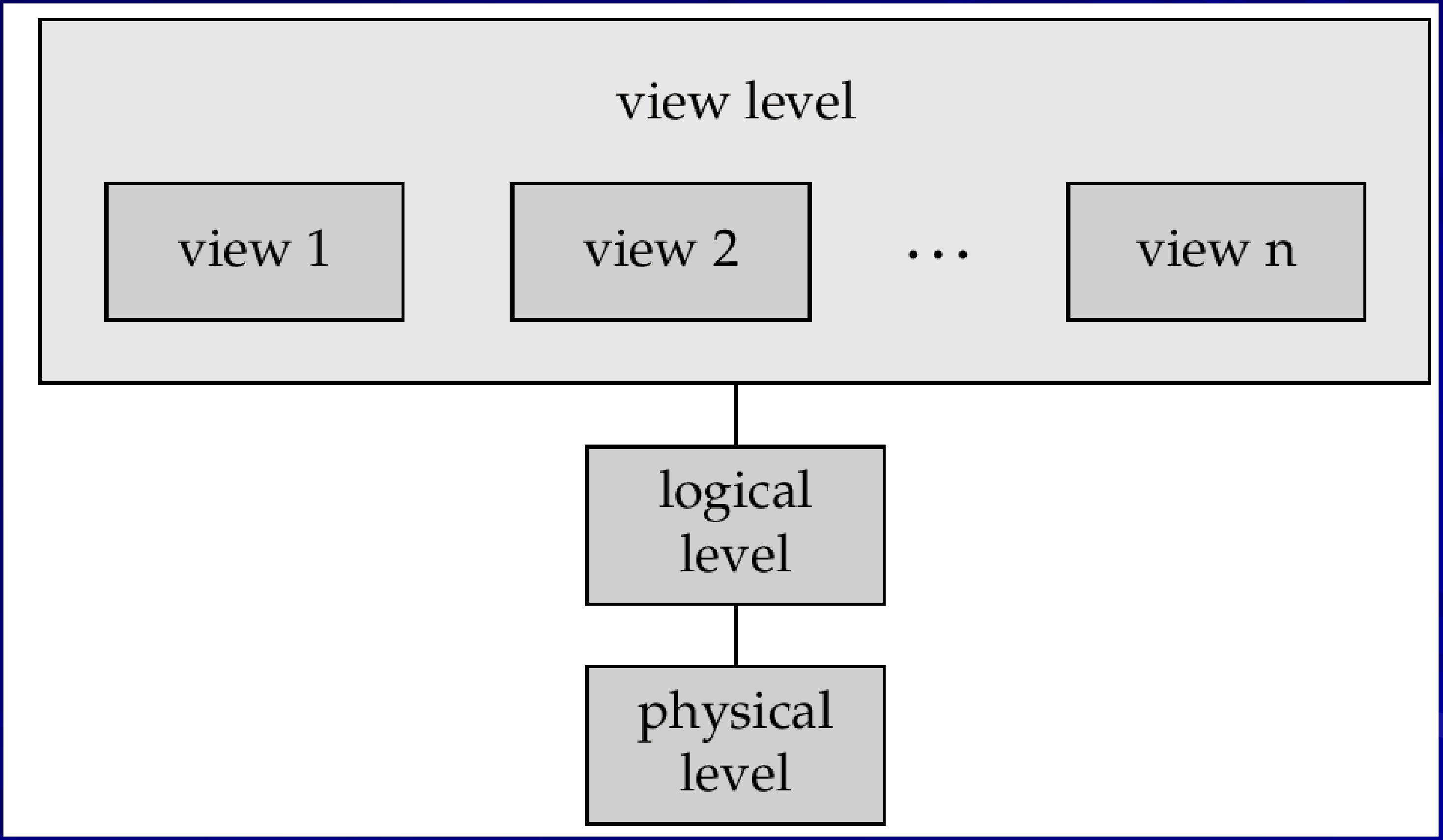
Data Levels of Abstraction:

- Physical level describes how a record (e.g., customer) is stored.
- Logical level: describes data stored in database, and the relationships among the data.

```
type customer = record
                    name : string;
                    street : string;
                    city : integer;
                end;
```

- View level: application programs hide details of data types. Views can also hide information (e.g., salary) for security purposes.

Data Levels of Abstraction:



Schemas and Instances:

- In terms of databases schemas and instances are like types and variables in programming languages.
- ***1. Schema*: is the logical structure of the database ,** it is analogous to type information of a variable in a program, **it has two types:**
- **Physical schema:** database design at the physical level
- **Logical schema:** database design at the logical level
- **e.g., the database consists of information about a set of customers and accounts and the relationship between them)**
-
- ***2. Instance*: is the actual content of the database at a particular point in time , it is** analogous to the value of a variable
- **Physical Data Independence:** the ability to modify the physical schema without changing the logical schema.

Data Models

Data Models : A collection of tools for describing

1. **Data**
2. **Data Relationships**
3. **Data Semantics**
4. **Data Constraints**

1. Types of Data Models:

1. **Entity-Relationship Model**
2. **Relational Model**
3. **Object-Oriented Model**
4. **Semi-Structured Data Models**

- ***Older models:***

1. **Network model**
2. **Hierarchical model**

Data Definition Language (DDL)

- Specification notation for defining the database schema
  - E.g.

    create table *account* (
    account-number **char**(10),
    balance **integer**)
- DDL compiler generates a set of tables stored in a *data dictionary*
- Data dictionary contains metadata (i.e., data about data)
  - database schema
  - Data *storage and definition* language
    - language in which the storage structure and access methods used by the database system are specified
    - Usually an extension of the data definition language

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - DML also known as query language
- Two classes of languages
  - Procedural – user specifies what data is required and how to get those data
  - Nonprocedural – user specifies what data is required without specifying how to get those data
- SQL is the most widely used query language

SQL

- SQL: widely used non-procedural language
  - E.g. find the name of the customer with customer-id 192-83-7465

```
select   customer.customer-name
from     customer
where    customer.customer-id = ‘192-83-7465’
```

  - E.g. find the balances of all accounts held by the customer with customer-id 192-83-7465

```
select   account.balance
from     depositor, account
where    depositor.customer-id = ‘192-83-7465’ and
         depositor.account-number = account.account-number
```

- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application program interface (e.g. ODBC/JDBC) which allow SQL queries to be sent to a database

Information and knowledge :

- Information is the Data that have been processed in such way of the person who use the Data. Since there are three types of data:

1. Input Data
2. Output Data
3. Operational Data

- Knowledge : is the gained facts and predictions after processing the information.

Database Users

- Users are differentiated by the way they expect to interact with the system
- 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
- Naïve users – invoke one of the permanent application programs that have been written previously
  - E.g. people accessing database over the web, bank tellers, clerical staff

Database Administrator

- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise's information resources and needs.
- Database administrator's duties include:
  - Schema definition
  - Storage structure and access method definition
  - Schema and physical organization modification
  - Granting user authority to access the database
  - Specifying integrity constraints
  - Acting as liaison with users
  - Monitoring performance and responding to changes in requirements

Transaction Management

- A *transaction* is a collection of operations that performs a single logical function in a database application
- Transaction-management component ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.

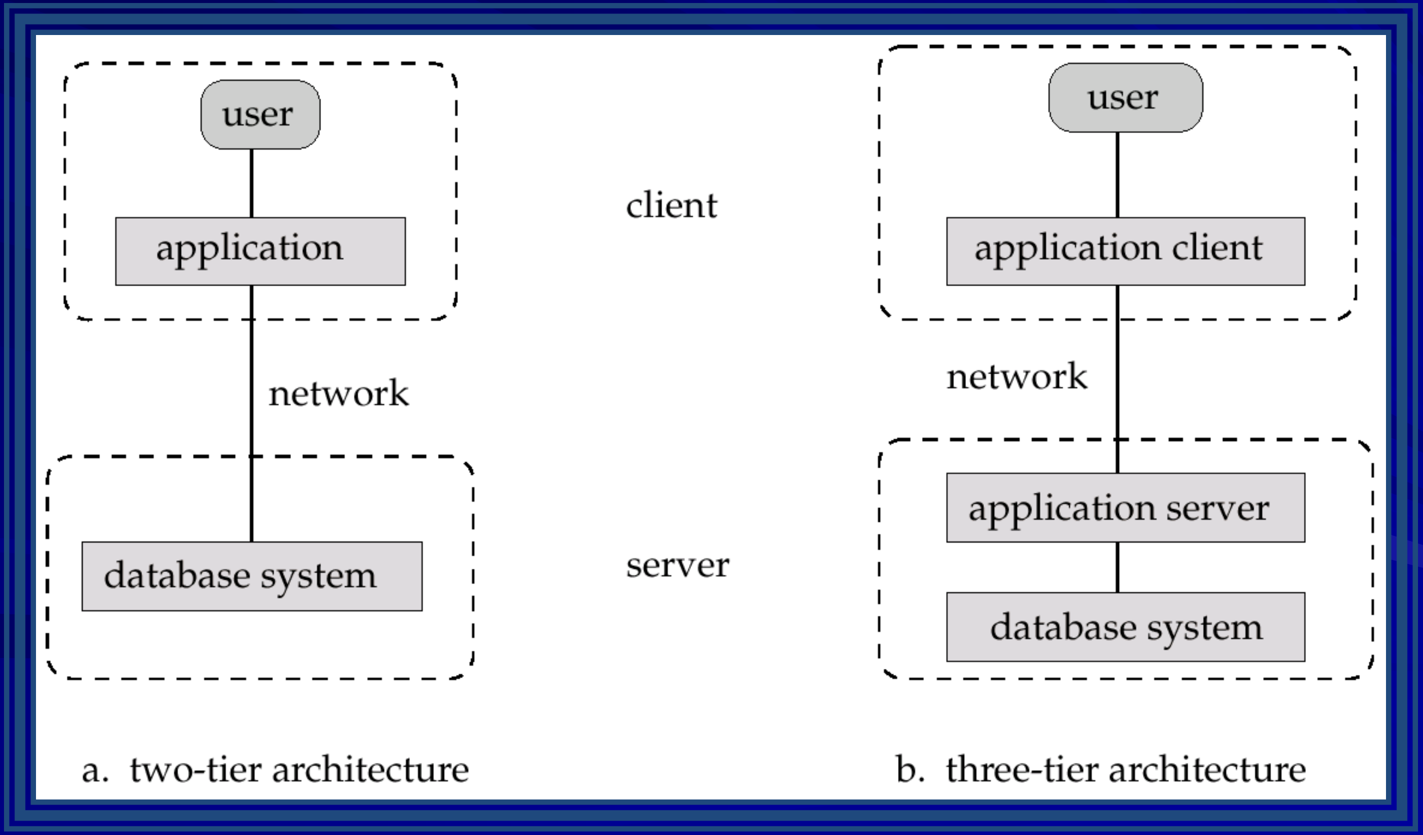
Storage Management

- 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 to the following tasks:

1. interaction with the file manager
2. efficient storing, retrieving and updating of data

Database Application Architectures :

1. **Two-tier architecture:**
2. **Three-tier architecture:**



General Aspects of Database:

1.**Entity :**is the object can be recognized from other objects depending on specific set of attributes. Such as Person, Student, Event, Plant.

2.**Relationship**: it is the union between entities

3.**Group Items**:the set of facts in the data base , it includes Entities + Relationships.

4.**Data Attributes**: a general and special properties or characteristic of an entity or relationship that is of interest to the organization.

General Aspects of Database:

5. Data Value: it is the information included by Data Attributes

6. Data Domain: the allowable domain used by the attributes to represent its data.

7. Data Structure :specification of the relationships among entities.

8. Records: (Instances): the set of values for an entity.

General Aspects of Database:

9.Primary Key : It is one or more than attributes included within an entity , it has a unique value for each instance of an entity , used for distinguishing between records .

- **e.g. : each bank customer has a fixed account_no could not be changed , and can used to determine the customer name and his account .**

10. Meta Data: The data that describes the properties of another data.

- **e.g.: declaring data structures.**

```
type customer = record
name : string;
street : string;
city : integer;
end;
```