

## **DATABASE MANAGEMENT SYSTEM**

### **DATABASE**

A **database** is an organized **collection of structured information**, or data, typically **stored electronically in a computer system**.

### **DATABASE MANAGEMENT SYSTEM(DBMS)**

DBMS is a **collection of programs** which enables its **users to access a database**, manipulate data and reporting/representation of data.

### **FILE PROCESSING SYSTEM**

A file processing system is a **collection of programs** that **store and manage files in computer hard-disk**.

### **DISADVANTAGES**

#### **Data redundancy and inconsistency**

- The same data is **stored at various locations** in a file processing system. This redundancy of data leads to **higher storage and access cost**.
- it may happen that **when data is modified** then it may **not be modified at all the places at a time**. This leads data to be in inconsistent stage.

#### **Difficulty in accessing data**

The data is scattered in **various files at different locations**. The files also may be stored in **the different formats**.

#### **Integrity problem**

- The data stored should **satisfy some conditions** before **stored in the files**.
- Whenever the **new constraints are added**, it is difficult to change the programs to enforce them.

### Concurrency problems

- In many applications, it is very essential that **multiple user's access and updates** the **same data simultaneously**.
- The concurrent updates may result in **inconsistent data**.

### Security problem

- There is always **sensitive data stored** in the database such as pay, account balance, etc.
- In the file processing system, application programs are added to the system in an ad hoc manner so that it becomes **difficult to enforce such security constraints**.

## DATA ABSTRACTION

- Data Abstraction is a process of **hiding unwanted or irrelevant details from the end user**.
- This is achieved through several **levels of abstraction** to **simplify user's interaction with the system**.

### Physical level

The lowest level of abstraction **describes how the data are actually stored**.

### Logical level

- The next higher level of abstraction describes **what data are stored in the database and relationships among those data**.

### View level

The highest level of abstraction describes **only part of the entire database**.



## **DATABASE ADMINISTRATOR**

The person who has such **central control Over the database** is called database administrator, referred as DBA.

### **FUNCTIONS**

- Implementing **data policies, procedures, standards**
- **Planning and development** of an organization's **enterprise data model**
- Resolving **data conflicts**
- Disaster recovery planning
- **Training employees** in database management and use
- **Backing up database systems** for recovery scenarios
- Create **appropriate storage structures and access methods.**
- **Grant access to users** as decided by the data owner. Users access data as per their rights.

### **DATA MODELS**

- Data models define **how the logical structure of a database is modelled.**
- Data models define **how data is connected to each other** and how they are **processed and stored inside the system.**

### **ENTITY-RELATIONSHIP MODEL**

- This Model is **based** on the notion of **real-world entities and relationships among them.**
- ER Model **deals with entities and their attributes and relationships among entities.**





- ER Model creates entity set, relationship set, general attributes and constraints.

## RELATIONAL MODEL

- It is a most popular and scientific a model than others.
- A relational database stores data in the form of relations (tables)
- Relations can be normalized so each row in a relation contains a unique value.

**Relational Data Model**

Table:

Column 1	Column 2	Column 3	Column 4
Roll No.	Name	Age	Gpa
1	Aryan	21	3
2	Sachin	25	4
3	Prince	20	2.5
4	Anuj	21	3.5

Row 1  
Row 2  
Row 3  
Row 4

## STRUCTURE OF RELATIONAL DATABASES

### Query Language

Query languages are computer languages used to query the Database or Information System.





## Types Query Language

### Procedural

In procedural languages, the user would specify **what has to be done and how it can be done.**

### Non-procedural

The programs written in non-procedural languages specify **what is to be done and do not state exactly how a result is to be evaluated.**

## PARTS OF SQL

### Data Definition Language (DDL)

DDL commands are used to **add, modify or delete tables in database.**

### Data Manipulation Language (DML)

The DML includes commands to **insert the data into the table** and the **extract data from the database.**

### Embedded DML

This form of **SQL is used in the general-purpose** programming language such as C, COBOL.



## DIFFERENCE BETWEEN DBMS VS RELATIONAL DBMS

DBMS	RDBMS
<ul style="list-style-type: none"><li>• DBMS stands for "Database Management System".</li></ul>	<ul style="list-style-type: none"><li>• RDBMS stands for "Relational Database Management System".</li></ul>
<ul style="list-style-type: none"><li>• DBMS technology stores the data in the form of files.</li></ul>	<ul style="list-style-type: none"><li>• RDBMS stores the data in the form of tables.</li></ul>
<ul style="list-style-type: none"><li>• DBMS is designed to handle small amounts of data.</li></ul>	<ul style="list-style-type: none"><li>• RDBMS is designed to deal with vast amount of data.</li></ul>
<ul style="list-style-type: none"><li>• DBMS provides support only for a single user at a time.</li></ul>	<ul style="list-style-type: none"><li>• RDBMS provides support for multiple users at a time.</li></ul>

## ACID PROPERTIES

In order to **maintain consistency in a database**, before and after the **transaction**, certain properties are followed. These are called ACID properties

### Atomicity

Atomicity requires that each **transaction be "all or nothing"**: if one part of the **transaction fails, then the entire transaction fails**, and the database state is left unchanged.

### Consistency

This means that **integrity constraints** must be maintained so that the database is **consistent before and after the transaction**.

### Example

The total amount before and after the transaction must be maintained.

Total before T occurs = 2000 + 2000 = 4000

Total after T occurs = 1000 + 3000 = 4000

### Isolation

This property ensures that multiple **transactions can occur concurrently without leading to the inconsistency** of the database state.

### Durability

The durability property ensures that once a **transaction** has been **committed, it will remain So, even in the event of power loss, crashes, or errors.**

## NORMALIZATION OF A DATABASE

- It is a process of **organizing the data** in database to **avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.**
- It should **divide larger tables to smaller tables** and link them using relationships.

### ANOMALIES IN DBMS

In Database Management System (DBMS), anomaly means the **inconsistency occurred in the relational table** during the operations performed on the relational table.

emp_id	emp_name	emp_address	emp_dept
101	Krish	Delhi	D001
101	Krish	Delhi	D002
123	Malini	Agra	D890
166	Navin	Chennai	D900
166	Navin	Chennai	D004

## TYPES OF ANOMALY

### Update anomaly

When we **update some rows in the table**, and if it leads to the inconsistency of the table then **this anomaly occurs**.

#### Example

In the above table, if we want to update the address of Krish then we will have to update all the rows where Krish is present. If during the update we miss any single row, then there will be two addresses of Krish, which will lead to inconsistent and wrong databases.

### Insert anomaly

If there is a new row inserted in the table and it creates the inconsistency in the table then it is called the insertion anomaly.

#### Example

If in the above table, we create a new row of a worker, and if it is not allocated to any department then we cannot insert it in the table so, it will create an insertion anomaly.

### Delete anomaly

If we delete some rows from the table and if any other information or data which is required is also deleted from the database, this is called the deletion anomaly in the database.

#### Example

In the above table, if we want to delete the department number D001 then the details of Krish will also be deleted.



## FORMS OF NORMALIZATION

### CONCEPTS OF KEYS

It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationships between tables.

#### Types of Keys

##### Primary key

It is the first key used to **identify one and only one instance** of an entity uniquely.

##### Candidate key

A candidate key is an **attribute or set of attributes** that can **uniquely identify a tuple**.

##### Super Key

super key is a **set of attributes** that **uniquely identifies each tuple of a relation**.

##### Foreign key

Foreign keys are the column of the table **used to point to the primary key of another table**.



## FORMS OF NORMALIZATION

### FIRST NORMAL FORM (1NF)

As per the rule of **first normal form**, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

### EXAMPLE

emp_id	emp_name	emp_address	emp_mobile
10	Harish	New Delhi	8912312390
11	Rahul	Kanpur	8812121212 9900012222
12	Ravi	Chennai	7778881212
13	Mahesh	Bangalore	9990000123 8123450987

### SOLUTION

emp_id	emp_name	emp_address	emp_mobile
101	Harish	New Delhi	8912312390
102	Rahul	Kanpur	8812121212
102	Rahul	Kanpur	9900012222
103	Ravi	Chennai	7778881212
104	Mahesh	Bangalore	9990000123
104	Mahesh	Bangalore	8123450987

### SECOND NORMAL FORM (2NF)

- Table is in 1NF (First normal form)
- **No non-prime attribute** is dependent on the **proper subset** of any candidate key of table.

- An attribute that is **not part of any candidate key** is known as non-prime attribute.

### EXAMPLE

teacher_id	subject	teacher_age
111	Maths	38
111	Physics	38
222	Biology	38
333	Physics	40
333	Chemistry	40

- **Candidate Keys** Teacher\_id and subject
- **Non-prime attribute:** Teacher\_age
- The table is in 1 NF because **each attribute has atomic values**.
- it is not in 2NF because **non-prime attribute teacher age is dependent on teacher\_id alone** which is proper subset of candidate key.

To make the table complies with 2NF we can break it in two tables like this

#### Teacher\_details

teacher_id	teacher_age
111	38
222	38
333	40

### Teacher\_subject

teacher id	subject
111	Maths
111	Physics
222	Biology
333	Physics
333	Chemistry

### THIRD NORMAL FORM (3NF)

- Table must be in 2NF
- **Transitive functional dependency of non-prime attribute** on any **super key** should be removed.

### TRANSITIVE DEPENDENCY

It refers to some **non-prime attribute** other than the candidate key that **depends on another non-prime attribute** that is dependent entirely on the candidate key.

emp_id	emp_name	emp_zip	emp_state	emp_city	emp_district
1001	John	282005	UP	Agra	Dayal Bagh
1002	Ajeet	222008	TN	Chennai	M-City
1006	Lora	282007	TN	Chennai	Urrapakkam
1101	Lilly	292008	UK	Pauri	Bhagwan
1201	Steve	222999	MP	Gwalior	Ratan

### BOYCE CODD NORMAL FORM (BCNF)

- It is in 3NF



- For every **functional dependency**  $X \rightarrow Y$ , X should be the **super key of the table**.

#### **FOURTH NORMAL FORM (4NF)**

- It builds on the **first three normal forms (1NF, 2NF and 3NF)** and the **Boyce-Codd Normal form (BCNF)**.
- In addition, table meeting the requirements of BCNF, **it must not contain more than one multivalued dependency**.

#### **FIFTH NORMAL FORM (5NF)**

- If we can **decompose table** further to **eliminate redundancy** and anomaly, and when we re-join the decomposed tables by means of **candidate keys**,
- In simple words, **joining two or more decomposed table** should not **lose records nor create new records**.

#### **ORACLE**

- It is an Object - **Relational database** from **Oracle Corporation with first commercial release in 1979**.
- The **latest version** of Oracle database is **Oracle 21c**, a database cloud service
- **ORACLE** provides full **support for communication networking** and distributed databases with **SQLnet and SQLstar products**.

#### **COMPONENTS**

**Forms** are used for the creating and running of custom **designed interface** for the **intersection, update, deletion and querying** of the tables.

**Menu** is a part of forms allows the **standard options like cut, paste**, etc. It can also include the **radio buttons in the menus**.



**Oracle Report** it is used to produce a **formatted output**. These **reports** can be produced with many levels of **nesting, multiple breaks on columns, subtotals, and grand totals** etc.

## BENEFITS OF ORACLE DATABASE

### Performance

- Oracle has **procedures and principles** that help us to get high levels of **database performance**.
- We can increase query execution time and operations with the use of **performance optimization techniques** in its database.

### Portability

- The Oracle database can be ported on **all different platforms**.
- We can use this database on **around 20 networking protocols** as well as **over 100 hardware platforms**.

### Backup and Recovery

- The Oracle database makes it easy to **accomplish backup and recovery quickly** by using the. **RMAN (Recovery Manager) functionality**.
- It can **recover or restore** database files **during downtime or outages**.

### PL/SQL

Oracle database supports **PL/SQL extension** for **procedural programming**.

### Multiple Database

Oracle database **allows several database** instances management on a single server.

### Flashback Technology

It allows us to **recover those data** that are **incorrectly deleted or lost** by **human errors**.

