# Unit-2: File Organization and Relational Algebra Concepts

## 1. File Organization

File organization refers to how data is stored in a file system to optimize retrieval, modification, and storage efficiency. Different methods help in improving performance.

## 1.1 Types of File Organization

## 1.1.1 Indexed Sequential Access (ISAM)

- Data is stored in order, and an index is used to find it quickly.  
- Uses primary and secondary indexes for faster access.  
- Example: If student records are sorted by roll number, an index helps in jumping directly to a specific roll number.

## 1.1.2 B-Trees & B++ Trees

- \*\*B-Trees:\*\* Balanced tree structures used for indexing large datasets, ensuring fast access.  
- \*\*B++ Trees:\*\* An improved version where all keys are stored at leaf nodes, making range queries more efficient.

## 1.1.3 Hashing

- A technique that maps keys to storage locations for fast data retrieval.  
- \*\*Hashing Functions:\*\* Functions that compute the storage location for a given key.

## 1.1.4 Collision Resolution Techniques

- \*\*Chaining:\*\* Uses linked lists to store multiple records in the same hash bucket.  
- \*\*Open Addressing:\*\* Finds the next available slot when a collision occurs.

## 1.1.5 Extendible Hashing

- A dynamic hashing method where the directory expands dynamically, preventing excessive collisions.

## 1.1.6 Dynamic Hashing

- Expands and shrinks hash tables dynamically based on data volume, improving performance and storage utilization.

## 2. Relational Databases

A relational database stores data in tables (relations), which consist of rows (tuples) and columns (attributes).

## 2.1 Relational Model Concepts

- Data is organized into tables with well-defined relationships.  
- Each table has a \*\*Primary Key\*\*, which uniquely identifies each row.  
- Relationships between tables are maintained using \*\*Foreign Keys\*\*.

## 2.2 Relational Constraints

1. \*\*Domain Constraint:\*\* Ensures each column has a specific type of data.  
2. \*\*Key Constraint:\*\* Each row must have a unique key (Primary Key).  
3. \*\*Referential Integrity Constraint:\*\* Ensures foreign keys match primary keys in another table.

## 2.3 SQL & SQL Constraints

SQL (Structured Query Language) is used to manage relational databases.  
- `NOT NULL` – Ensures a column cannot have empty values.  
- `UNIQUE` – Ensures all values in a column are unique.  
- `PRIMARY KEY` – Combines NOT NULL and UNIQUE constraints.  
- `FOREIGN KEY` – Ensures referential integrity between two tables.

## 3. Relational Algebra Concepts

Relational algebra is a set of operations used to manipulate relational database tables. It helps in query processing.

## 3.1 Types of Relational Operations

- \*\*Selection (σ):\*\* Filters rows based on a condition.  
- \*\*Projection (π):\*\* Selects specific columns.  
- \*\*Union:\*\* Combines two tables with the same structure.  
- \*\*Intersection:\*\* Returns common rows between two tables.  
- \*\*Difference:\*\* Returns rows from one table that are not in another.  
- \*\*Cartesian Product:\*\* Combines all rows from two tables.  
- \*\*Join (⨝):\*\* Combines related rows from two tables.

## 3.2 Joins in SQL

1. \*\*Inner Join\*\* – Returns matching rows from both tables.  
2. \*\*Left Join\*\* – Returns all rows from the first table and matching rows from the second.  
3. \*\*Right Join\*\* – Returns all rows from the second table and matching rows from the first.  
4. \*\*Full Join\*\* – Returns all rows from both tables, with NULL for missing matches.

## 4. Static vs. Dynamic Hashing

| Feature | Static Hashing | Dynamic Hashing |  
|--------------|--------------|--------------|  
| Definition | Uses a fixed-size hash table | Hash table size changes dynamically |  
| Memory Utilization | May waste memory | Efficient memory use |  
| Performance | Can slow down | Maintains good performance |  
| Collision Handling | Chaining/Open addressing | Expanding buckets |  
| Scalability | Limited | Highly scalable |

## 5. Database Design using EER to Relational Model

A \*\*Enhanced Entity-Relationship (EER) model\*\* includes:  
- \*\*Entities (Objects like Students, Courses, etc.)\*\*  
- \*\*Attributes (Properties like Name, Age, Course\_ID, etc.)\*\*  
- \*\*Relationships (How entities relate to each other)\*\*

## Conclusion

This unit covers essential concepts of file organization, relational databases, and relational algebra for efficient database design and management.