Chapter- 1

Database Concepts

Period – 01

Contents

- Database
- Advantages of database system
- Database abstraction
- Levels of database
- Database users
- Data models
- Relational data model
- Keys
- Relational algebra

Database

A database is a collection of interrelated data and a database system is basically a computer based record keeping system.

The intention of a database is that the same collection of data should serve as many applications as possible. Hence, a database is often conceived of as the repository of information needed for running certain functions in a corporation or organization.

Purpose of database

In a typical file-processing system, permanent records are stored in various files. A number of different application programs are written to extract records from and add records to the appropriate files. But this scheme has a number of major limitations and disadvantages, such as data redundancy (duplication of data), data inconsistency, unsharable data, unstandardized data, insecure data, incorrect data etc. A database management system is answer to all these problems as it provides a centralized control of the data.

Advantages provided by the database systems

1. Databases reduce the data redundancy to a large extent
2. Databases can control data inconsistency to a large extent
3. Databases facilitate sharing of data
4. Databases enforce standards
5. Databases can ensure data security
6. Integrity can be maintained through databases

DATABASE ABSTRACTION

A major purpose of a database system is to provide the users only that much information that is required by them. This means that the system does not disclose all the details of data, rather it hides certain details of how the data is stored and maintained.

A good database system ensures easy, smooth and efficient data structures in such a way so that every type of database user :end user, application system analyst, and physical storage system analyst, is able to access its desired information efficiently.

Database users

End users

An end user is a person who is not a computer-trained person but uses the database to retrieve some information.

For example, in a bank database, a customer, who wants to know how much balance remains in his account, is an end-user.

Application System Analyst

An application system analyst is the one who is concerned about all of

The database of logical level i.e., what all data constitute the database, what are the relationships between the data-entities etc. without considering the physical implementation details.

Physical Storage System Analyst

The physical storage system analyst is concerned with the physical implementation details of the database i.e., how would the database be stored on which storage device? What will be the starting address of the database? What will be the storage technique? Etc.

Since the requirements of different users differ from one another, the complexity of the database is hidden from them.

Hence the database is implemented through following 3 levels of abstraction.

Period – 02

Various Levels of Database Implementation

A database is implemented through three general levels :

1. Internal level
2. Conceptual level
3. External level

1. Internal Level (Physical Level)

The lowest level of abstraction, the internal level, is the one closest to physical storage. This level is also sometimes termed as physical level. It describes how the data are actually stored on the storage medium. At this level, complex low-level data structures are described indetails.

2. Conceptual Level (logical level)

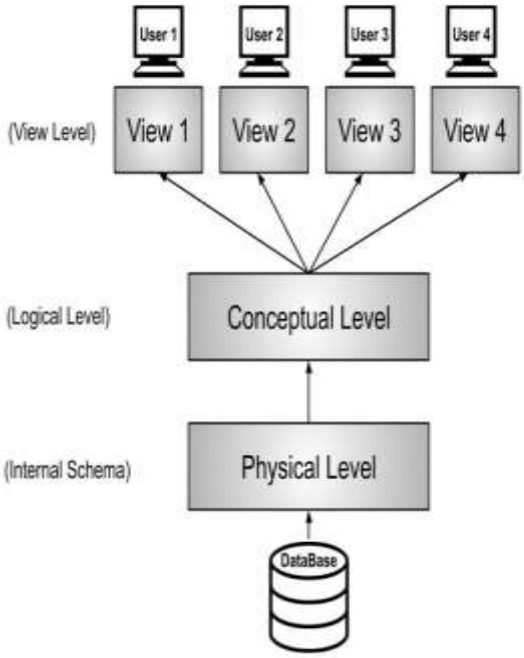
This level of abstraction describes what data are actually stored in the database. It also describes the relationships existing among data. At this level, the database is described logically in terms of simple data-structures. The users of this level are not concerned with how these logical data structures will be implemented at the physical level. Rather, they just are concerned about what information is to be kept in the database.

3. External Level (View Level)

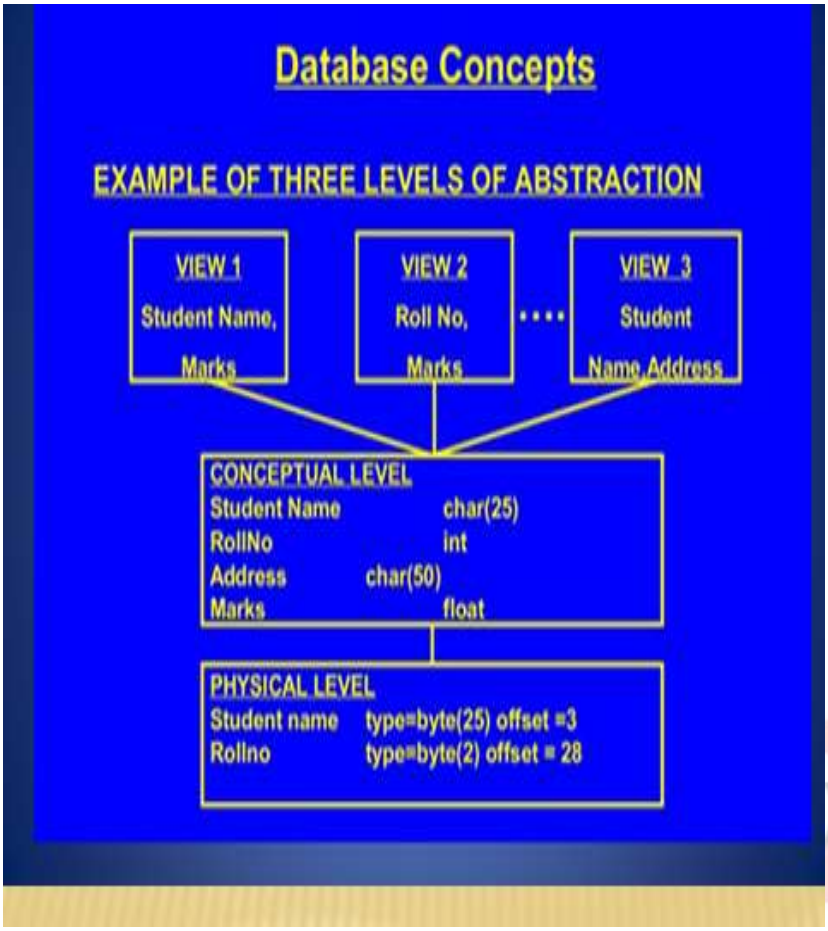
This is the level closest to the users and is concerned with the way in which the

data are viewed by individual users. Most of the users of the database are not concerned with all the information contained in the database. Instead, they need only a part of the database relevant to them.

For example, even though the bank database stores a lot many information, an account holder (a user) is interested only in his account details and not with the rest of the information stored in the database.



Levels of Data Abstraction



Data Independence

The ability to modify a scheme definition in one level without affecting a scheme definition in the next higher level is called Data Independence.

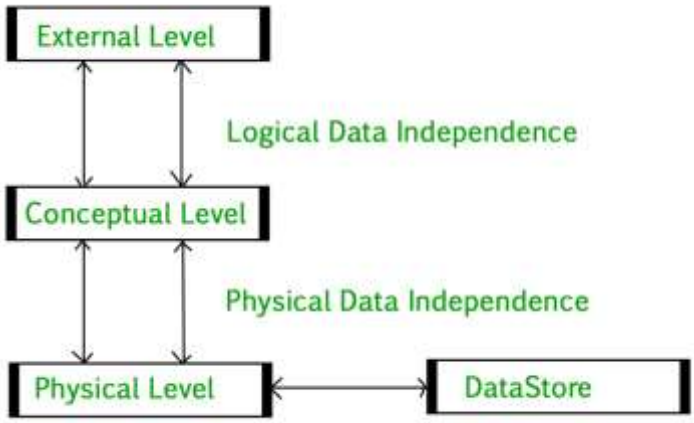
There are two levels of data independence: physical and logical.

Physical Data Independence

The ability to modify a scheme definition in physical level without affecting a scheme definition in the conceptual level is called physical Data Independence.

Logical Data Independence

The ability to modify a scheme definition in conceptual level without affecting a scheme definition in the view level is called logical Data Independence.



DIFFERENT DATA MODELS

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

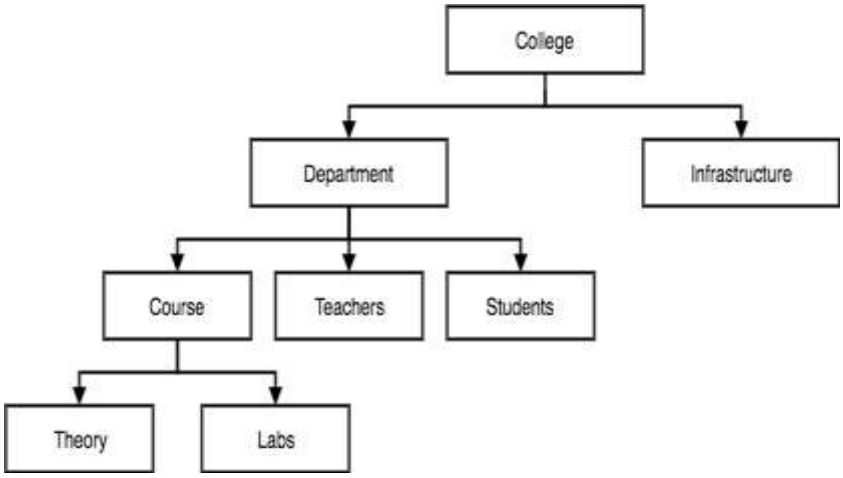
Data models are followed to create database.

The most widely used database models are :

1. Hierarchical Data Model
2. Network Data Model
3. Relational Data Model

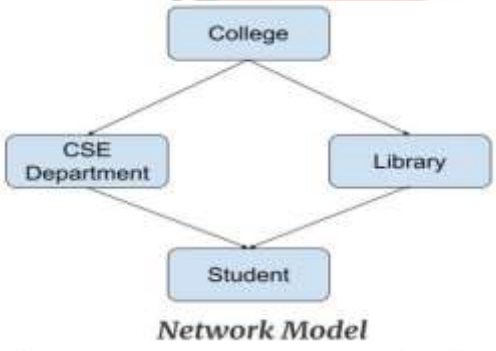
Hierarchical Data Model

In hierarchical model, data is organized into tree-like structure with one-to-many relationship between two different types of data.



Network Data Model

This model is an extension of the hierarchical model. This model is the same as the hierarchical model; the only difference is that a record can have more than one parent. It replaces the hierarchical tree with a graph.



Network Model

The Relational Data Model

The relational model was propounded by E.F. Codd of the IBM. In relational data model, the data is organized into tables.

- Terminology
- Relation
- Tuple
- Attribute
- Degree
- Cardinality
- Domain

- Relation

In general, a relation is a table i.e., data is arranged in rows and columns.

A relation has the following properties:

1. In any given column of a table, all items are of the same kind where as items in different columns may not be of the same kind.
2. For a row, each column must have an atomic (indivisible) value and also for a row, a column cannot have more than one value.
3. All rows of a relation are distinct. That is, a relation does not contain two rows which are identical in every column. That is, each row of the relation can be uniquely identified by its contents.
4. The ordering of rows within a relation is immaterial. That is, we cannot retrieve anything by saying that from row number 5, column name is to be accessed. There is no order maintained for rows inside a relation.
5. The columns of a relation are assigned distinct names and the ordering of these columns is immaterial.

Tuple

The rows of tables (relations) are generally referred to as Tuples .

Attributes

The columns/fields of tables (relations) are generally referred to as attributes.

Degree

The number of attributes in a relation determine the degree of a relation. A relation having 3 attributes is said to be a relation of degree 3. Similarly, a relation having n attributes is said to be a relation of degree n.

Relations of degree one are said to be unary, relations of degree two are binary, relations of degree three are ternary, ..., and relations of degree n are n-ary.

Cardinality

The number of tuples (rows) in a relation is called the cardinality of the relation .

Domain

A domain is a pool of values from which the actual values appearing in a given column are drawn.

Emp_id	Emp_name	Job_name	Salary	Mobile_no	Dep_id	Project_id
AfterA001	John	Engineer	100000	9111037890	2	99
AfterA002	Adam	Analyst	50000	9587569214	3	100
AfterA003	Kande	Manager	890000	7895212355	2	65

EMPLOYEE TABLE

Relation: Employee

Attributes: Emp_id, Emp_name, job_name, Salary,Mobile_noDep_id, Project_id

Tuples:

AftreAoo1 john engineer 100000 9111037890 2 99

AftreAoo1 Adam Analyst 50000 9587569214 3 100

AftreAoo3 kande manager 890000 7895212355 2 65

Degree: 7

Cardinality: 3

Domain of Dep_id: 1,2,3,4,5

Period – 03

View

- A view is a virtual table that doesn't really exist in the database.
- A view can be derived from the contents of other tables based on certain condition.
- Just like other tables a view can be queried, updated, inserted into, deleted from

etc.

Keys

- They are used to establish and identify relationships between tables and also to uniquely identify any record or row of data inside a table.
- A Key can be a single attribute or a group of attributes, where the combination may act as a key.

Types of keys

- Primary key
- Candidate key
- Alternate key
- Foreign key

Primary key

A primary key is a set of one or more attributes that can uniquely identify tuples within the relation.

Every relation does have a primary key. In some tables, combination of more than one attribute provides a unique value for each row. In such tables, the group of these attributes is declared as primary key. In such cases, the primary key consists of more than one attribute, it is called composite-primary-key.

- Primary keys must contain unique values.
- A primary key can not have NULL values.

Customer ID	Forename	Surname
1	Simon	Jones
2	Emma	Price
3	Laura	Jones
4	Jonathan	Hale
5	Emma	Smith

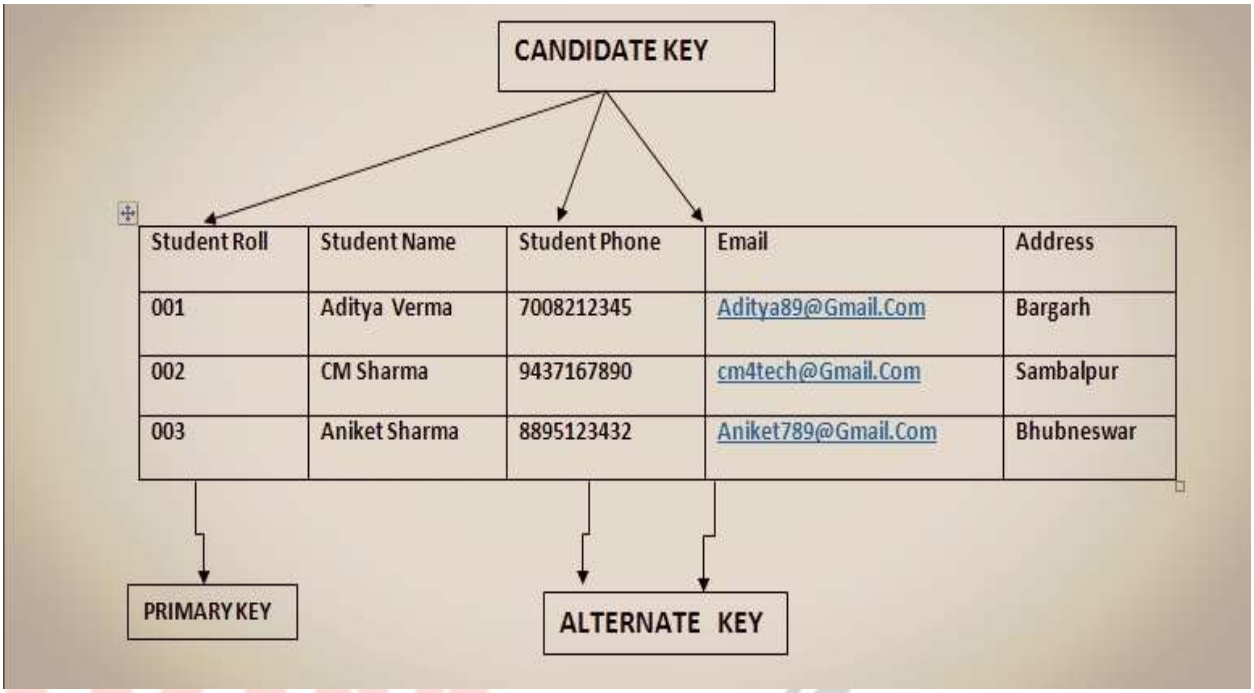
Simple primary key

Student	Date	Period	Present
J. Smith	01/06/09	1	Y
B. Rogers	01/06/09	1	Y
A. Black	01/06/09	2	N
H. Rose	02/06/09	1	Y
M. Wright	02/06/09	2	N

These three fields are combined to make a composite primary key

Candidate key

- A set of keys those are having the ability to be the primary key are called candidate keys.
- Out of all the candidate keys one is selected as the primary key.

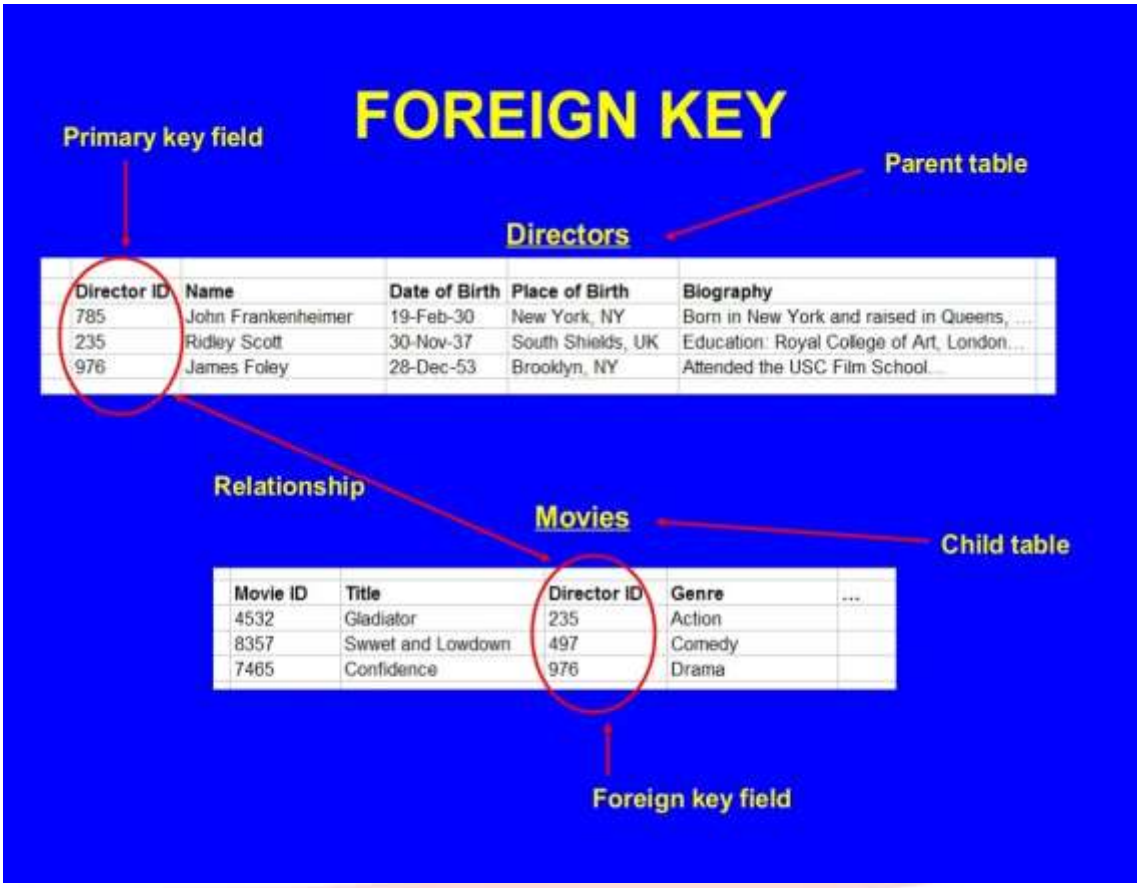


Alternate key

The candidate keys which are not selected as the primary key are referred as alternate keys.

Foreign key

- It establishes relationship between two tables.
- It is a non-key whose values are derived from the primary key of another table.
- The table in which the non-key or foreign key exists is called foreign table or detailed table.
- The table that defines the primary key is called the primary table or master table.



The Relational Algebra

The relational algebra is a collection of operations on relations. Each operation takes one or more relations as its operand(s) and produces another relation as its result.

The following operations can be performed on the relations.

- Select operation
- Project operation
- Cartesian product
- Union operation
- Intersection operation

Unary operations

Binary operations

- Set difference

Select operation

The SELECT operation is used for selecting a subset of the tuples according to a given selection condition. Sigma(σ) Symbol denotes it.

- Syntax:

$\sigma_{condition}(relation)$

Relational operators (<,<=,>,>=,=,≠) can be used in conditions. More than one conditions can be used by combining with AND, OR operators.

Example:

1. Selects tuples from Tutorials where topic = 'Database'.

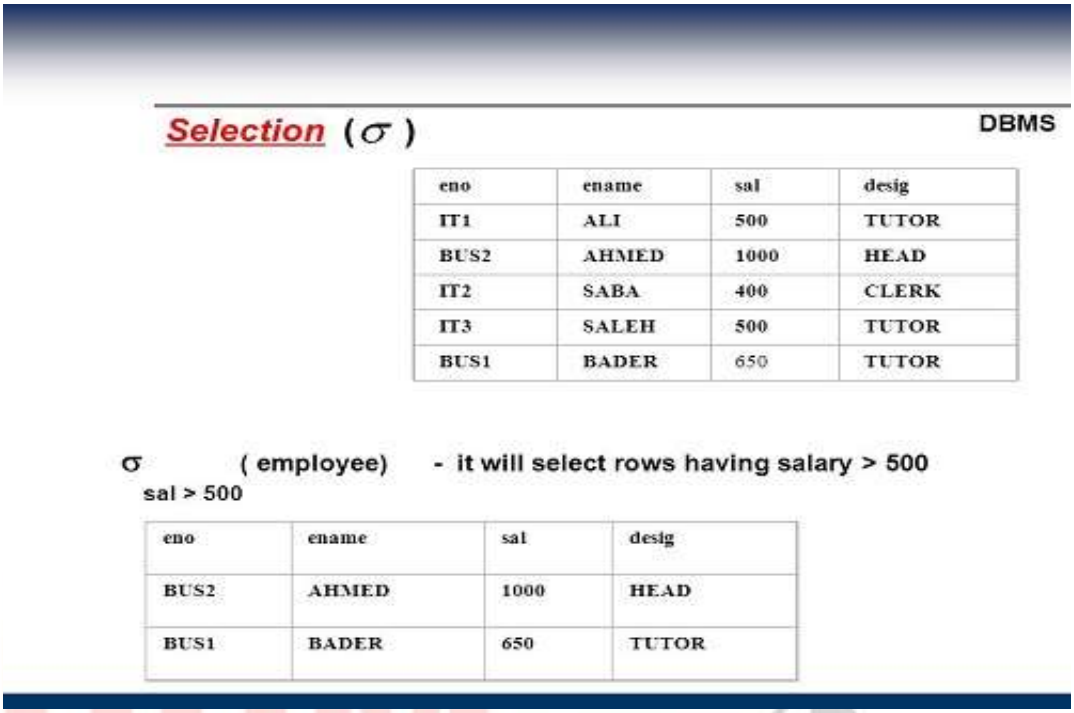
$$\sigma_{\text{topic = "Database"}} \textbf{ (Tutorials)}$$

2. Selects tuples from Tutorials where the topic is 'Database' and author is ‘guru99’.

$$\sigma_{\text{topic = "Database" and author = "guru99"}} \textbf{(Tutorials)}$$

3. Selects tuples from Customers where sales is greater than 50000

$$\sigma_{\text{sales > 50000}} \textbf{ (Customers)}$$



Project Operation

Projection Operator (π) is a unary operator in relational algebra that performs a projection operation. It displays the columns of a table based on the specified attributes.

Syntax:

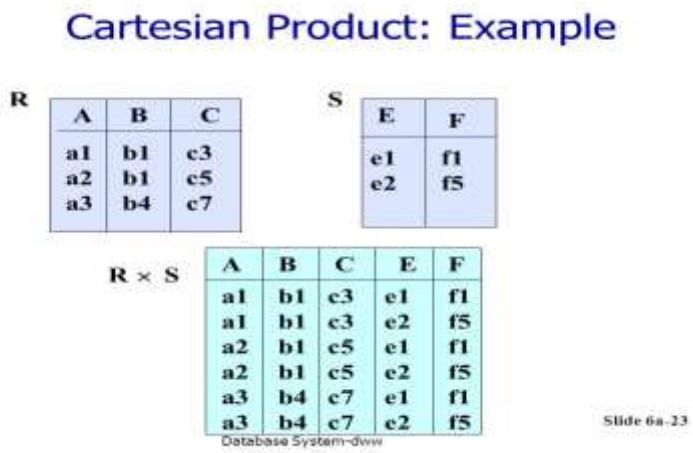
π<attribute list>(Relation)

ID	Name	Subject	Age
100	Ashish	Maths	19
200	Rahul	Science	20
300	Naina	Physics	20
400	Sameer	Chemistry	21

$\Pi_{Name,\ Age}$ **(Student)**

Cartesian Product (X)

The **Cartesian Product** called as the **CROSS PRODUCT**. It combines the tuples of one relation with all the tuples of the other relation. In the resultant relation the columns of both the input relations are added and the rows will be multiplied.

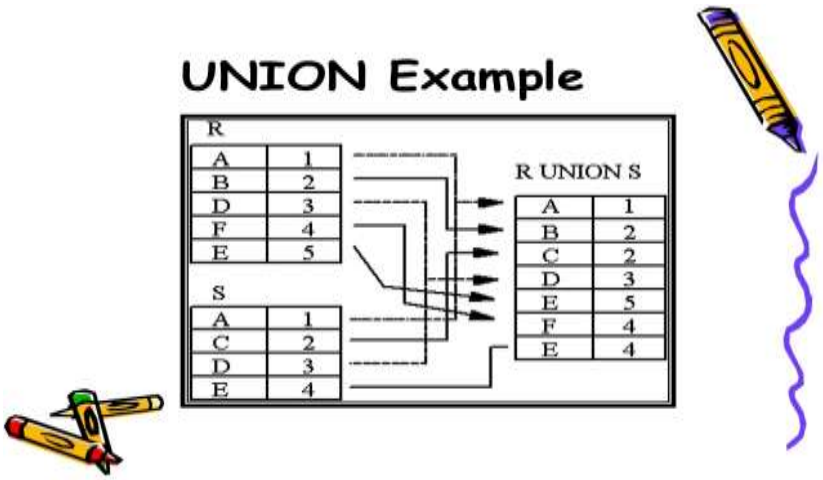


Union operation

The union operation between two relations produces a third relation that combines tuples from both the relations. The union operation between A and B is denoted by **AU B** that combines tuples from both the relations A and B.

Two things need to keep in mind while applying union operation are :

- Both the relations compulsory to have same number of attributes.
- Both the relations compulsory to have same domain for attributes.



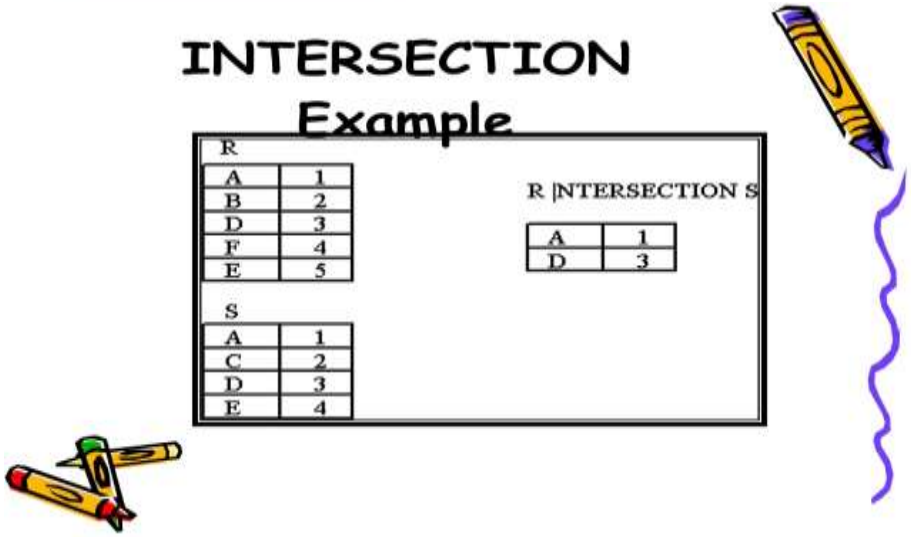
A U B=B U A

Intersection Operation

The intersection operation between two relations produces a third relation that selects the common tuples from both the relations. The Intersection operation between A and B is denoted by **A ∩ B**.

Two things need to keep in mind while applying intersection operation are :

- Both the relations compulsory to have same number of attributes.
- Both the relations compulsory to have same domain for attributes.



$$A \cap B = B \cap A$$

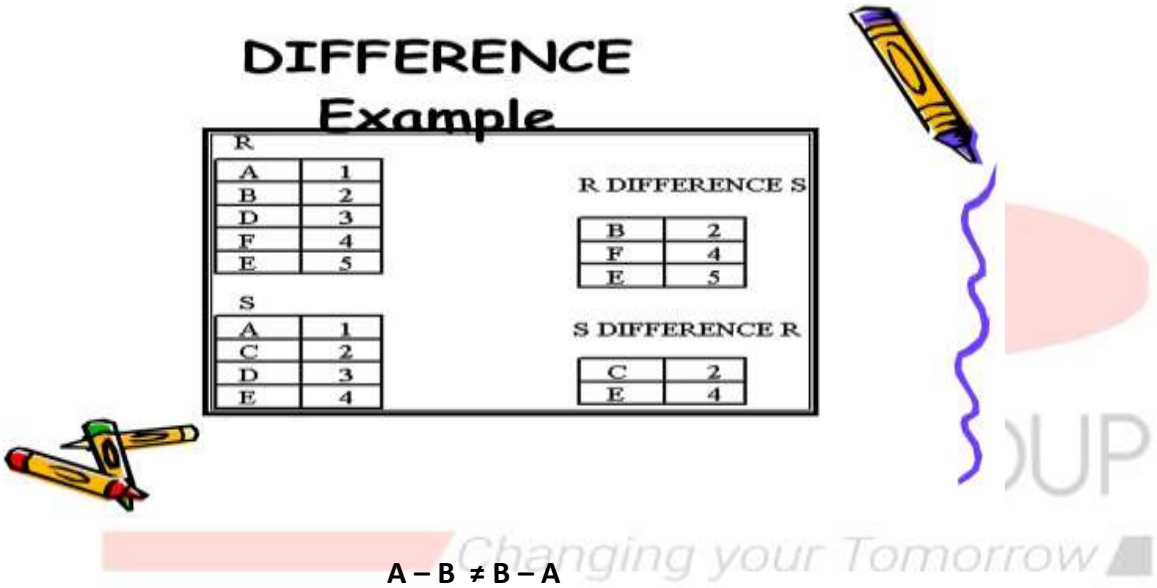
Difference operation

Difference operation between two relations A and B produces the resultant relation as

A - B that selects tuples present in A but not in B.

Two things need to keep in mind while applying difference operation are :

- Both the relations compulsory to have same number of attributes.
- Both the relations compulsory to have same domain for attributes.



$$A - B \neq B - A$$

Disadvantages of database system

- **Security may be compromised without good controls.**
- **Integrity may be compromised without good controls.**
- **Extra hardware may be required.**
- **Performance overhead may be significant.**
- **System is likely to be complex.**