JAC BOARD 2026-27

COMPUTER SCIENCE -12th

DATABASE

Notes & MCQ

Study Smart • Score High • Build Strong Concepts



Class By

AJAY SIR

(BCA)

10+ Years Exp.

WHAT YOU WILL GET

- Complete Database Notes
- Important Definitions
- Relational Algebra with Examples
- Written Questions (50)
- MCQ (Chapter Wise)
- Model Questions
- JAC Board Pattern

RELATIONAL ALGEBRA

- σ Selection (Rows)
- π Projection (Columns)
- ∪ Union (Combine)
- − Set Difference (A - B = A not in B)
- × Cartesian Product (Every row × Every row)
- ∩ Intersection (Common rows)

MCQ TOPICS COVERED

- Database Basics
- Relational Model
- Keys & Constraints
- Relational Algebra
- SQL Basics
- Normalization
- Joins
- Integrity Rules
- More...

BEST NOTES
BEST PRACTICE
BEST RESULTS

CONTACT US
7870735176
9334134557

WEBSITE
uceti.com

PLAYSTORE
UCETI
Computer Education

ADDRESS
Babugaon Chowk,
Shiv Mandir Road,
Hazaribag

UJJWAL COMPUTER

EDUCATION AND TRAINING INSTITUTE

Right Skills • Right Direction • Right Career



UNIT- 3 DATABASE

What is Database?

A **Database** is a collection of related data stored in an organized manner so that it can be easily **accessed, managed, updated, and retrieved**.

- **Collection of Related Data:** A database stores related information together.
- **Organized Storage:** Data is arranged in a proper and systematic way.
- **Easy Access:** Data can be accessed quickly whenever required.
- **Easy Retrieval:** Required information can be searched and retrieved easily.
- **Easy Update & Management:** Data can be added, modified, deleted, and managed efficiently.
- **Safe Storage:** It stores a large amount of data securely.

Examples: School, Bank, Hospital, Library, Railway Reservation System.

What is the Purpose of Databases?

The **purpose of a database** is to store, organize, manage, and retrieve data efficiently and securely.

- **Store Data:** Stores a large amount of data in one place.
- **Organize Data:** Keeps data in a proper and systematic manner.
- **Easy Access:** Allows users to access data quickly.
- **Easy Retrieval:** Helps to search and retrieve required information.
- **Easy Update:** Data can be added, modified, or deleted easily.
- **Reduce Data Redundancy:** Avoids duplicate (repeated) data.
- **Improve Data Security:** Protects data from unauthorized access.
- **Data Sharing:** Allows multiple users to use the same database.
- **Backup and Recovery:** Helps recover data if it is lost.
- **Better Decision Making:** Provides accurate information for making decisions.

Examples

- **School:** Student Records , **Bank:** Customer Accounts , **Hospital:** Patient Records , **Library:** Book Records , **Railway:** Ticket Reservation

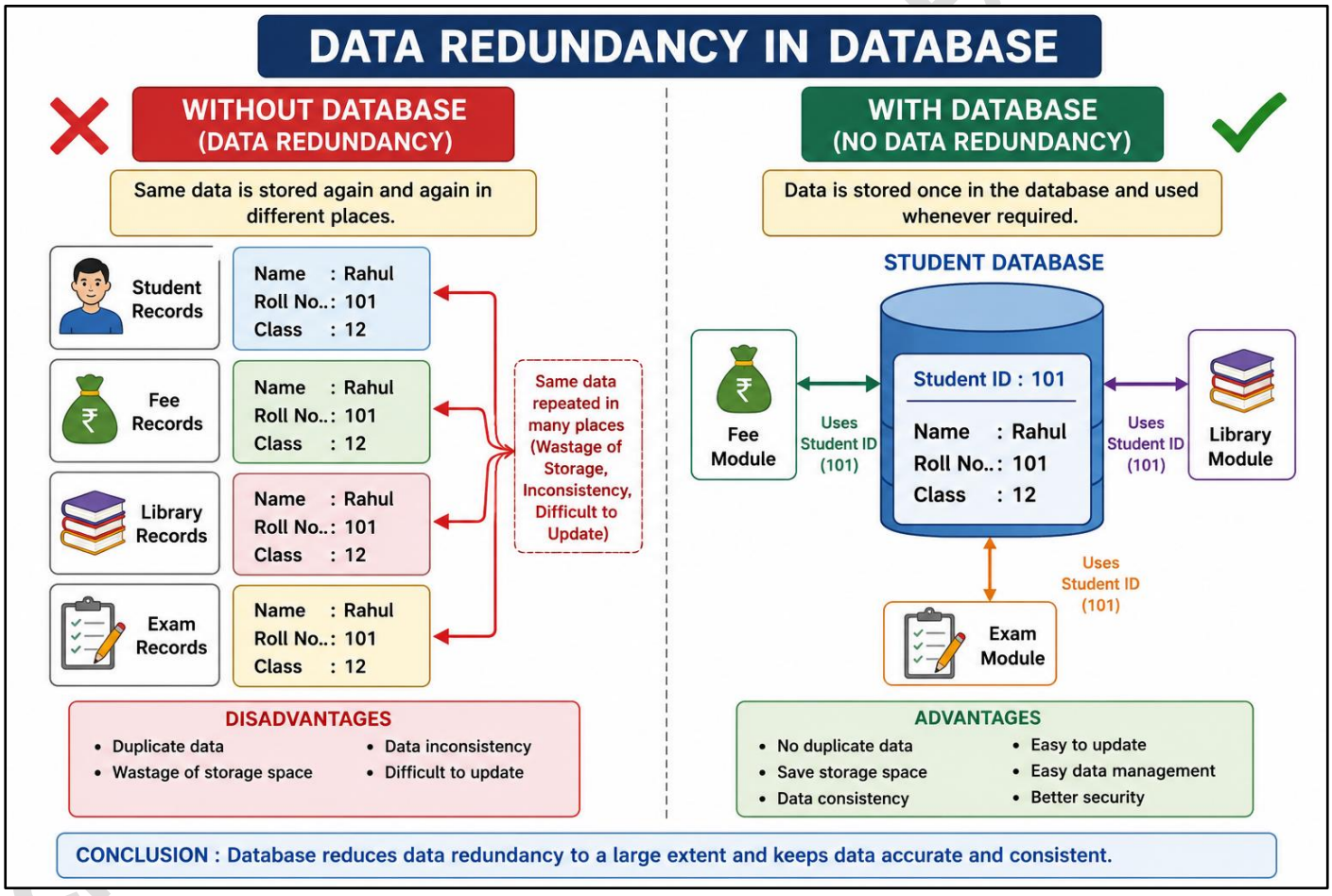
Class By :- Ajay Sir (10 - Years Experience)

What is Data Redundancy?

Data Redundancy is the unnecessary duplication (repetition) of the same data in different places within a database.

- **Duplicate Data:** The same data is stored more than once.
- **Wastes Storage Space:** It uses extra memory unnecessarily.
- **Causes Inconsistency:** Different copies of the same data may contain different values.
- **Difficult to Manage:** Updating duplicate data takes more time and effort.
- **Database Reduces Redundancy:** A DBMS minimizes duplicate data and keeps data consistent.

- **Database reduces data redundancy to a large extent.**

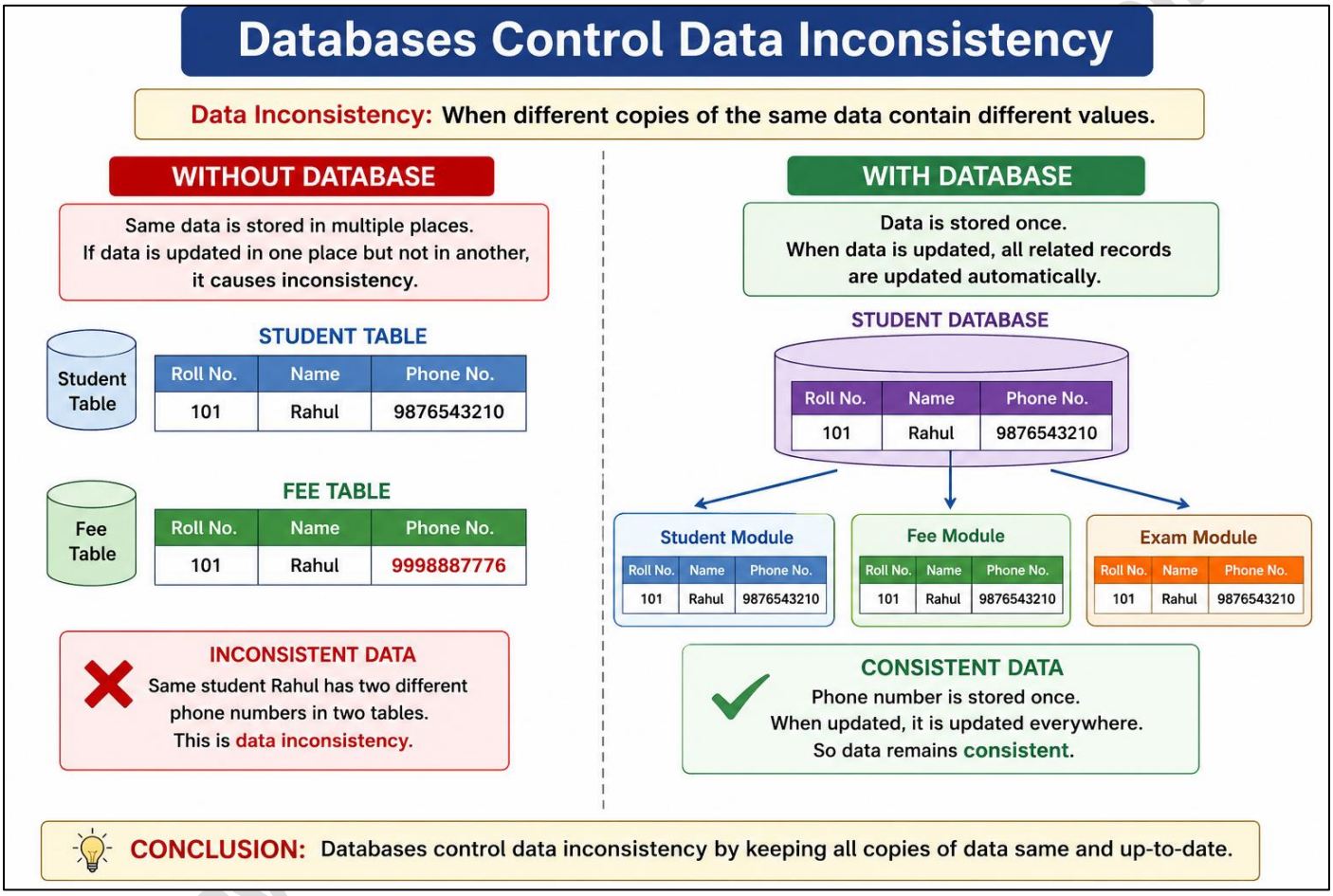




Databases control data inconsistency

Data Inconsistency means that the **same data is stored in different places, but the values are not the same.** This usually happens when one copy of the data is updated while another copy is not.

- The **same data** is stored in **more than one place**.
- One copy is **updated**, but another copy is **not updated**.
- As a result, **different values** are shown for the same data.
- This problem is called **Data Inconsistency**.



What do you mean by "Databases facilitate sharing of data"?

Databases facilitate sharing of data means that the same data can be accessed and used by **multiple users or applications at the same** time according to their permissions.

- **Multiple Users:** Many users can access the same database.
- **Data Sharing:** The same data can be shared among different users and applications.
- **Easy Access:** Users can access the required data quickly.
- **No Duplicate Files:** There is no need to create separate copies of the same data.



- **Improves Collaboration:** Different departments can work using the same database.

Example (Bank Database)

A bank has **one customer database**.

- **Cashier** checks account balance.
- **Loan Department** verifies customer details.
- **ATM** uses the same database for transactions.
- **Mobile Banking App** shows account information.
- **Bank Manager** views customer reports.

➡ **All services use the same database without creating separate copies of the data.** This is called **Databases facilitate sharing of data.**

What do you mean by "Databases enforce standards"?

Databases enforce standards means that a database stores and manages data in a **uniform (standard) format** so that all users follow the same rules and the data remains accurate and consistent.

Comparison: Without Standards vs With Database Standards

Without Standards ❌	With Database Standards ✅
Different data formats are used.	A single standard format is used.
Data may become inconsistent.	Data remains consistent.
Confusion and errors may occur.	Data is clear and accurate.
Difficult to search and manage data.	Easy to search and manage data.
Data exchange is difficult.	Data exchange is easy.

Example: Student Phone Numbers

Without Standards ❌	With Database Standards ✅
9876543210	9876543210

Class By :- Ajay Sir (10+ Years Experience)

+91-9876543210	9123456789
98765-43210	9988776655
(98765)43210	9876543211

How can Databases ensure data security?

Database Security means protecting the database from **unauthorized access, modification, destruction, or disclosure** of data.

- **Protects Data:** Protects data from unauthorized users.
- **Authorized Access Only:** Only authorized users can access the database.
- **Prevents Unauthorized Changes:** Stops unauthorized modification or deletion of data.
- **Maintains Privacy:** Keeps sensitive information private and confidential.
- **Ensures Safe Storage:** Stores data securely and protects it from misuse.

How can Integrity be maintained through Databases?

Data Integrity means ensuring that the data stored in a database is **accurate, correct, complete, and consistent** throughout its life.

- **Maintains Accuracy:** Ensures the data is correct and error-free.
- **Maintains Consistency:** Keeps the same data consistent throughout the database.
- **Prevents Invalid Data:** Stops incorrect or invalid data from being entered.
- **Maintains Relationships:** Preserves the correct relationship between related tables.
- **Provides Reliable Information:** Users always get accurate and trustworthy data.

Without Data Integrity ❌	With Data Integrity ✅
Incorrect data may be entered.	Only valid and correct data is accepted.
Duplicate or inconsistent data may occur.	Data remains accurate and consistent.
Relationships between tables may be broken.	Relationships are maintained correctly.
Reports may contain wrong information.	Reports are accurate and reliable.



How can Integrity be maintained through Databases?

Data Integrity means ensuring that the data stored in a database is **accurate, correct, complete, and consistent** throughout its life.

- **Maintains Accuracy:** Ensures the data is correct and error-free.
- **Maintains Consistency:** Keeps the same data consistent throughout the database.
- **Prevents Invalid Data:** Stops incorrect or invalid data from being entered.
- **Maintains Relationships:** Preserves the correct relationship between related tables.
- **Provides Reliable Information:** Users always get accurate and trustworthy data.

Without Data Integrity ❌	With Data Integrity ✅
Incorrect or invalid data may be entered.	Only valid and correct data is accepted.
Different tables may contain different values.	All related tables contain consistent data.
Data may become unreliable.	Data remains accurate and reliable.
Reports may contain incorrect information.	Reports are always correct and trustworthy.

What is Database Abstraction?

Database Abstraction is the process of **hiding the complex details of a database** and showing **only the required information** to the user.

- **Hides Details:** It hides the database details.
- **Shows Needed Data:** It shows only the needed data.
- **Easy to Use:** It makes the database easy to use.
- **Keeps Data Safe:** It protects important data.
- **Different Users, Different Data:** Each user sees only their own data.

Levels of Database Abstraction

Level	Description
Physical Level (Internal Level)	Describes **how data is stored** in the database.
Logical Level (Conceptual Level)	Describes **what data is stored** and the relationship between data.
View Level (External Level)	Shows **only the required data** to different users.

Class By Ajay Sir (10+ Years Experience)



What is Physical Level (Internal Level)?

The Physical Level (Internal Level) describes how data is actually stored in the database, including storage structures and access methods.

- **Stores Data:** It stores data in the database.
- **Internal Details:** It contains the internal storage details.
- **Hidden from Users:** Normal users cannot see this level.
- **Used by DBA:** It is mainly used by the **Database Administrator (DBA)**.
- **Improves Performance:** It helps store data efficiently for faster access.

Example

Suppose a **School Database** stores student records.

- **Student** sees only **marks and attendance**.
- **Teacher** sees **student details**.
- **Physical Level** stores the records on the **hard disk or server**.

What is Logical Level (Conceptual Level)?

The **Logical Level (Conceptual Level)** describes **what data is stored in the database and how different data is related to each other.**

- **Describes Data:** It tells what data is stored.
- **Shows Relationships:** It shows how different data is connected.
- **Hides Storage Details:** It does not show how data is physically stored.
- **Used by Designers:** It is used by database designers and programmers.
- **Easy to Understand:** It provides a clear structure of the database.

Example

School Database

Table	Stored Data
Student	Roll No, Name, Class
Marks	Roll No, Subject, Marks
Fee	Roll No, Fee Status

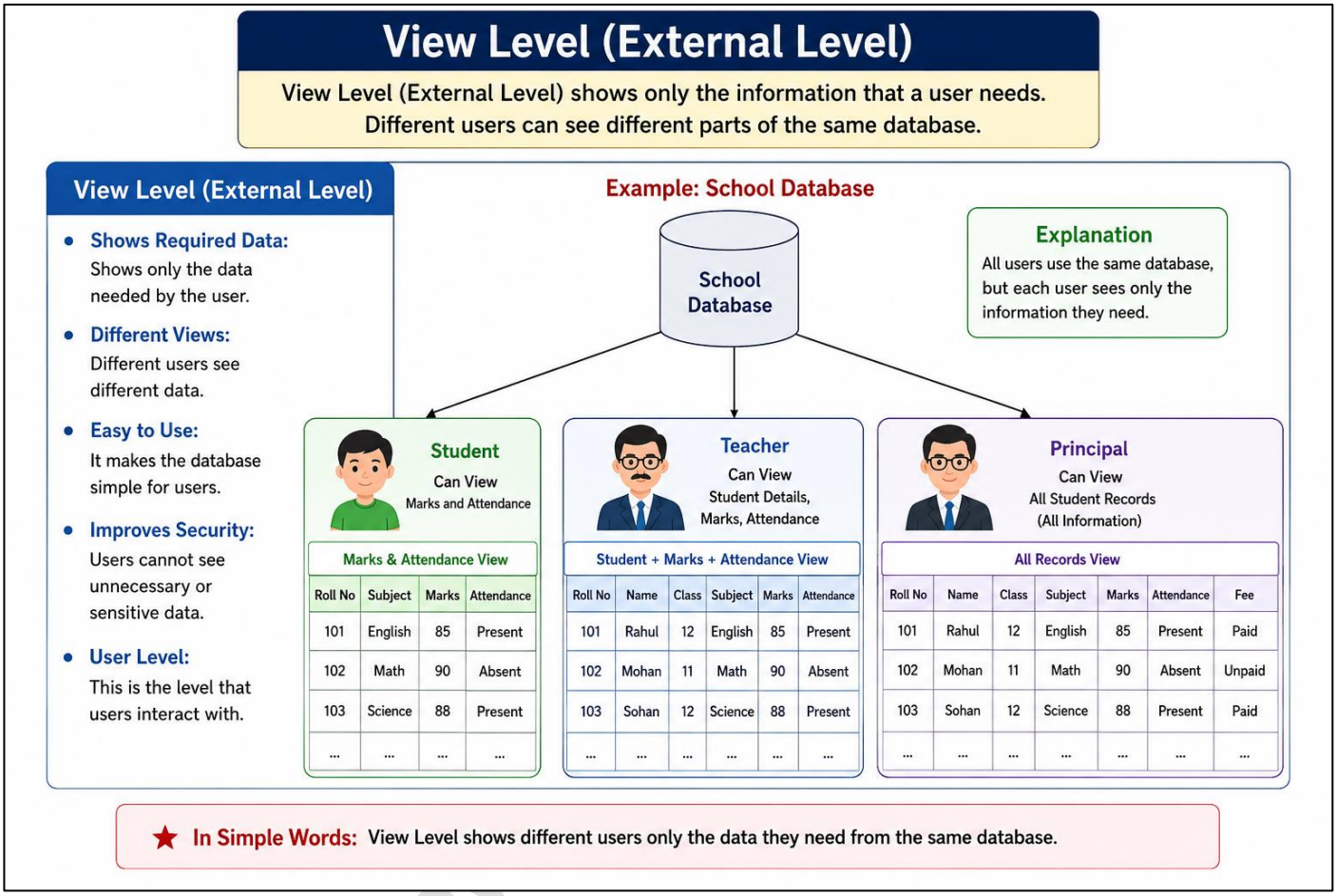
What is View Level (External Level)?

The **View Level (External Level)** shows **only the information that a user needs**. Different users can see different parts of the same database.

- **Shows Required Data:** Shows only the needed information.
- **Different Views:** Different users see different data.

Class By :- Ajay Sir (10+ Years Experience)

- **Easy to Use:** Makes the database simple for users.
- **Improves Security:** Users cannot see confidential data.
- **User Level:** This is the level that users interact with.



What is Database Independence?

Database Independence means that **changes made in one level of the database do not affect the other levels or application programs.**

Example: Bank Database

- A customer uses the **bank app** to check the account balance.
- The **DBA** changes the database storage or structure.
- The **bank app still works normally**, and the customer can check the balance.

➡ **This is Database Independence because database changes do not affect the user or application.**

Types of Database Independence

1. Physical Data Independence
2. Logical Data Independence



Physical Data Independence

Physical Data Independence means that **changes in the Physical (Internal) Level do not affect the Logical (Conceptual) Level.**

Example

- The DBA changes the **hard disk** or **storage method** of the database.
- Users continue using the application normally.

➡ **Application works without any changes.**

Logical Data Independence

Logical Data Independence means that **changes in the Logical (Conceptual) Level do not affect the View (External) Level.**

Example

- A **Bank Database** adds a new field called **Email** in the Customer table.
- Customers can still use the banking app normally.

➡ **Users do not notice any changes.**

Comparison

Physical Data Independence	Logical Data Independence
Changes in **Physical Level**	Changes in **Logical Level**
Does not affect **Logical Level**	Does not affect **View Level**
Easier to achieve	More difficult to achieve
Example: Changing storage device	Example: Adding a new column in a table

What are Different Data Models?

A **Data Model** is a way of **organizing, storing, and representing data** in a database.

Types of Data Models

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

Define Hierarchical Data Model

The **Hierarchical Data Model** is a database model in which data is organized in a **tree-like structure**. It follows a **Parent–Child relationship**, where each child has **only one parent**, but a parent can have **many children**.

Class By:- Ajay Sir (10+ Years Experience)

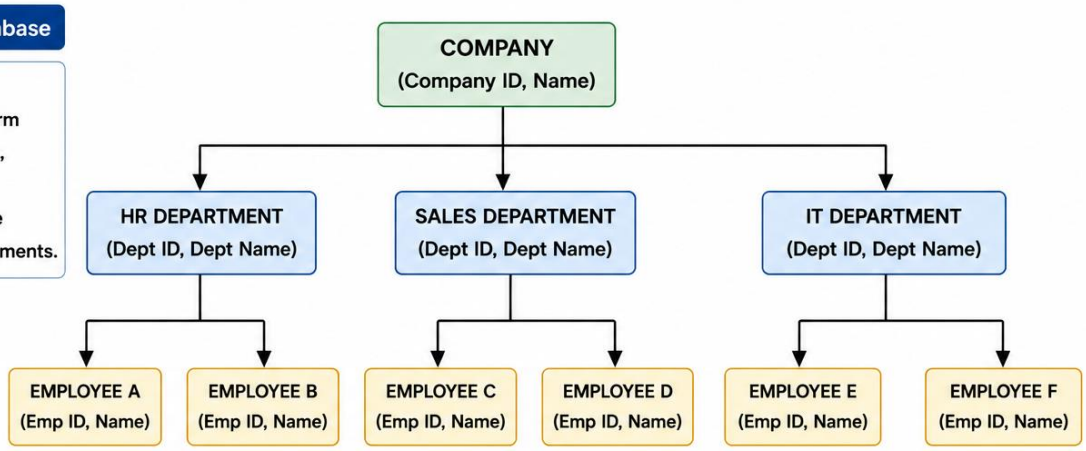
- **Tree Structure:** Data is organized in a tree form.
- **Parent–Child Relationship:** Each child has only one parent.
- **One-to-Many Relationship:** One parent can have many child records.
- **Easy Navigation:** Data is accessed from the top (root) to the bottom.
- **Fast Access:** Suitable for hierarchical data.

Hierarchical Data Model – Example

Hierarchical Data Model organizes data in a tree structure. It follows Parent–Child relationship.
Each child has only one parent, but a parent can have many children.

Example : Company Database

In a company, a structure is maintained in hierarchical form where Company is at the top, Departments are below the company, and Employees are below their respective departments.



Define Network Data Model.

Network Data Model is a database model that stores data in a **network structure,** where **one child can have multiple parents,** supporting **many-to-many relationships**.

Class By

Network Data Model – Example

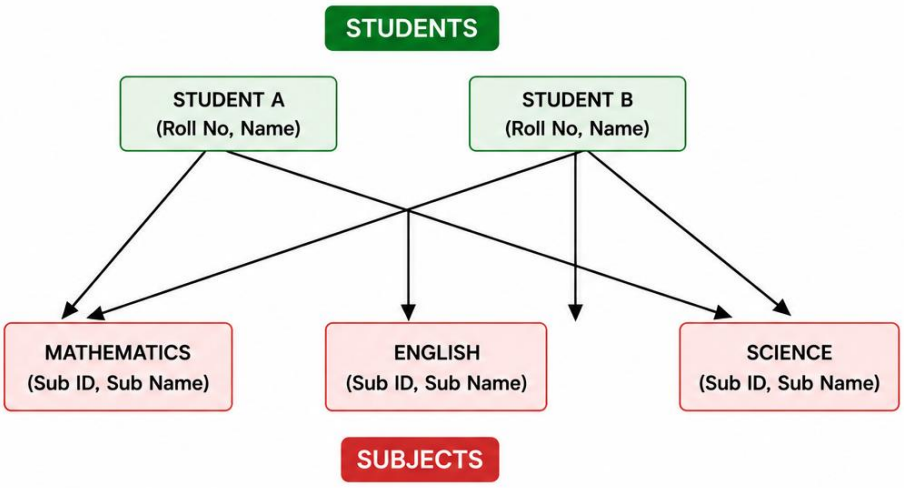
Network Data Model organizes data in a network structure. It allows a child record to have more than one parent, supporting many-to-many relationships.

Example : Student–Subject Database

In this database, Students and Subjects are connected in a network form.

- One student can study many subjects.
- One subject can be studied by many students.
- Each subject has more than one parent (student).
- Each student has more than one parent (subject).

This is a Many-to-Many relationship.



Explanation

- Students and Subjects are records.
- Lines (links) show relationships between them.
- A Subject can have many Students (multiple parents).
- A Student can have many Subjects.
- This model is more flexible than Hierarchical model.

Define Relational Data Model.

The **Relational Data Model** is a database model in which **data is stored in tables (rows and columns)**. The tables are connected using **Primary Key** and **Foreign Key**.

- **Table Structure:** Data is stored in **tables**.
- **Rows and Columns:** Each table contains **rows (records)** and **columns (fields)**.
- **Primary Key:** Each table has a **Primary Key** to uniquely identify each record.
- **Foreign Key:** Tables are connected using a **Foreign Key**.
- **Easy to Manage:** It is simple, flexible, and the **most widely used** database model.

Class By- Ajay Sir



Relational Data Model – Explanation with Example

Relational Data Model is a database model in which data is stored in tables (rows and columns).
The tables are connected using Primary Key and Foreign Key.

1 Table Structure
Data is stored in tables.

2 Rows and Columns
Each table contains rows (records) and columns (fields).

3 Primary Key
Each table has a Primary Key to uniquely identify each record.

4 Foreign Key
Tables are connected using a Foreign Key.

5 Easy to Manage
It is simple, flexible, and the most widely used database model.

Example : School Database

1. STUDENT TABLE

Student_ID (PK)	Student_Name	Age	Gender
101	Amit Kumar	20	Male
102	Priya Singh	19	Female
103	Rahul Verma	21	Male

Student_ID is the Primary Key (PK) of Student table.

2. COURSE TABLE

Course_ID (PK)	Course_Name	Duration
C01	Computer Science	3 Months
C02	Web Development	4 Months
C03	Tally with GST	2 Months

Course_ID is the Primary Key (PK) of Course table.

3. ENROLLMENT TABLE (Connecting Table)

Enrollment_ID (PK)	Student_ID (FK)	Course_ID (FK)	Enroll_Date
1	101	C01	01-06-2024
2	101	C02	05-06-2024
3	102	C02	07-06-2024
4	103	C03	10-06-2024
5	101	C03	12-06-2024

Student_ID in Enrollment table is a Foreign Key (FK) referencing Student_ID in Student table.

Course_ID in Enrollment table is a Foreign Key (FK) referencing Course_ID in Course table.

PK = Primary Key FK = Foreign Key

Advantages
- Easy to understand.
- Reduces data redundancy.
- Improves data integrity.
- Easy to maintain.
- Supports complex queries using SQL.

How It Works
Data is stored in different tables. → Each table has a Primary Key. → Foreign Key in one table relates it to another table. → Using these keys, related data can be retrieved. → This is called Relational Data Model.

Terminology of Relational Data Model

Term	Meaning
Relation	Table
Attribute	Column / Field
Tuple	Row / Record
Domain	Valid values of an attribute
Degree	Number of columns
Cardinality	Number of rows
Primary Key	Unique identifier
Foreign Key	Connects two tables

Class By :- Ajay Sir

Relation (Table) in Relational Data Model

A **Relation** is a **table** in a relational database. It stores data in the form of **rows (tuples)** and **columns (attributes)**.

Characteristics of Relation

- A **Relation is a table.**
- It consists of **rows (tuples)** and **columns (attributes)**.
- Each row represents **one record.**
- Each column represents **one attribute (field).**
- Every relation has a **unique name.**

Example
Student Relation (Student Table)

Student_ID	Student_Name	Age	Course
101	Amit	20	DCA
102	Priya	19	ADCA
103	Rahul	21	Tally

Explanation

- **Relation:** Student Table
- **Rows (Tuples):** 3
- **Columns (Attributes):** Student_ID, Student_Name, Age, Course

Attribute (Column/Field) in Relational Data Model

Example

Student Table

Student_ID	Student_Name	Age	Course
101	Amit	20	DCA
102	Priya	19	ADCA

Explanation

- **Student_ID** → Attribute
- **Student_Name** → Attribute
- **Age** → Attribute
- **Course** → Attribute

➡ **Each column in the table is called an Attribute.**

Class By :- Ajay Sir (10 - Years Experience)



Tuple (Row/Record) in Relational Data Model

A **Tuple** is a **single row (record)** in a relation (table). It contains the complete information about one entity.

Characteristics of Tuple

1. A **Tuple is a row (record)** in a table.
2. It contains values for **all attributes (columns)**.
3. Each tuple represents **one complete entity**.
4. A table can have **many tuples**.

Example

Student Table

Student_ID	Student_Name	Age	Course
101	Amit	20	DCA
102	Priya	19	ADCA
103	Rahul	21	Tally

Explanation

- **Tuple 1:** (101, Amit, 20, DCA)
- **Tuple 2:** (102, Priya, 19, ADCA)
- **Tuple 3:** (103, Rahul, 21, Tally)

➡ **Each row in the table is called a Tuple.**

Domain in Relational Data Model

A **Domain** is the **set of all valid (allowed) values** that can be stored in an attribute (column).

Characteristics of Domain

- A **Domain** defines the **allowed values** for an attribute.
- It helps maintain **data accuracy**.
- It prevents **invalid data** from being entered.
- Each attribute has its **own domain**.

Example

Student_ID	Student_Name	Age	Gender
101	Amit	20	Male
102	Priya	19	Female

Class By - Ajay Sir (10 - Years Experience)

Domain of Each Attribute

Attribute	Domain (Allowed Values)
Student_ID	Positive Integers (101, 102, 103, ...)
Student_Name	Alphabetic characters (Amit, Priya, Rahul, ...)
Age	1–100
Gender	Male, Female

Degree in Relational Data Model

The **Degree** of a relation is the **total number of attributes (columns)** in a table.

- **Characteristics of Degree**
- Degree is the **number of columns** in a table.
- It is also called the **number of attributes**.
- Degree is **fixed** for a table unless a new column is added or removed.
- Degree does **not depend on the number of rows**.

Example

Student_ID	Student_Name	Age	Course
101	Amit	20	DCA
102	Priya	19	ADCA
103	Rahul	21	Tally

Explanation

- **Student_ID** → 1st Attribute
- **Student_Name** → 2nd Attribute
- **Age** → 3rd Attribute
- **Course** → 4th Attribute

➡ **Total Columns (Attributes) = 4**

Degree = 4

Cardinality in Relational Data Model

The **Cardinality** of a relation is the **total number of tuples (rows/records)** in a table.

Characteristics of Cardinality

1. Cardinality is the **number of rows (records)** in a table.
2. It is also called the **number of tuples**.

Class By :- Ajay Sir (10 - Years Experience)



3. Cardinality changes when records are **added** or **deleted**.
4. It does **not depend on the number of columns**.

Example

Student_ID	Student_Name	Age	Course
101	Amit	20	DCA
102	Priya	19	ADCA
103	Rahul	21	Tally
104	Neha	20	DCA

Explanation

- **Tuple 1:** 101, Amit, 20, DCA
- **Tuple 2:** 102, Priya, 19, ADCA
- **Tuple 3:** 103, Rahul, 21, Tally
- **Tuple 4:** 104, Neha, 20, DCA

➡ **Total Rows (Tuples) = 4**
Cardinality = 4

Primary Key in Relational Data Model

A **Primary Key** is an **attribute (column)** or a combination of attributes that **uniquely identifies each record (row)** in a table. A Primary Key **cannot contain duplicate or NULL values**.

Characteristics of Primary Key

- A **Primary Key uniquely identifies** each record.
- It **cannot contain duplicate values**.
- It **cannot contain NULL values**.
- Every table should have **only one Primary Key** (it may consist of one or more columns).
- It helps in **connecting tables** and maintaining data integrity.

Example

Student_ID (PK)	Student_Name	Age	Course
101	Amit	20	DCA
102	Priya	19	ADCA
103	Rahul	21	Tally

Class By :- Ajay Sir (10+ Years Experience)

Explanation

- **Student_ID** is the **Primary Key**.
- Each student has a **different Student_ID**.
- No two students can have the same **Student_ID**.
- **Student_ID** cannot be **NULL**.

Foreign Key in Relational Data Model

A **Foreign Key** is an **attribute (column)** in one table that **references the Primary Key of another table**. It is used to **connect two tables**.

- **Characteristics of Foreign Key**
- A **Foreign Key connects two tables**.
- It **references the Primary Key** of another table.
- It helps maintain **relationships** between tables.
- A Foreign Key **can contain duplicate values**.
- A Foreign Key **may contain NULL values** (if allowed).

Example

Student Table

Student_ID (PK)	Student_Name
101	Amit
102	Priya
103	Rahul

Fee Table

Fee_ID	Student_ID (FK)	Amount
F01	101	5000
F02	102	5000
F03	103	6000

Class By :- Ajay Sir (10 - Years Experience)



Explanation

- **Student_ID** is the **Primary Key** in the **Student Table**.
- **Student_ID** in the **Fee Table** is the **Foreign Key**.
- The **Foreign Key** connects the **Student Table** and the **Fee Table**.

Types of Keys in DBMS

Keys are used to **identify records uniquely** and **maintain relationships** between tables.

Types of Keys

1. Candidate Key
2. Primary Key
3. Alternate Key
4. Foreign Key
5. Unique Key

Explanation

- **Candidate Key:** A minimal Super Key (no unnecessary attributes).
- **Primary Key:** The Candidate Key selected to uniquely identify records.
- **Alternate Key:** The remaining Candidate Keys that are not selected as the Primary Key.
- **Foreign Key:** An attribute that references the Primary Key of another table.
- **Unique Key:** Ensures that all values in a column are unique.

MCQ Questions Exercise

1. A Database is a collection of:
A) Programs B) Related Data C) Hardware D) Networks
2. DBMS is mainly used to:
A) Delete computers B) Manage databases C) Design websites D) Write programs
3. Which of the following reduces duplicate data?
A) Data Redundancy B) Database C) Compiler D) Browser
4. Data Redundancy means:
A) Unique data B) Duplicate data C) Secure data D) Encrypted data

Class By : Ajay Sir (10+ Years Experience)



5. Data Inconsistency occurs when:
A) All copies are updated B) Different copies have different values C) Data is deleted
D) Data is encrypted
6. Database Abstraction means:
A) Showing all details B) Hiding complex details
C) Deleting data D) Compressing data
7. Which is the highest level of Database Abstraction?
A) Physical Level B) Logical Level C) View Level D) Storage Level
8. Physical Level describes:
A) What data is stored B) How data is stored C) User view D) Reports
9. Logical Level describes:
A) Database storage B) Data and relationships C) User screen D) Hardware
10. View Level is also called:
A) Internal Level B) Conceptual Level C) External Level D) Physical Level
11. Database Independence means:
A) Data cannot change B) One level changes without affecting others
C) Data is deleted D) Database is offline
12. Physical Data Independence affects:
A) Logical Level B) Physical Level only C) View Level D) None
13. Which Data Model uses a tree structure?
A) Relational B) Network C) Hierarchical D) Object-Oriented
14. Which Data Model supports many-to-many relationships?
A) Hierarchical B) Network C) Sequential D) File
15. Which Data Model stores data in tables?
A) Network B) Hierarchical C) Relational D) Tree
16. A Relation is:
A) Column B) Row C) Table D) Record
17. An Attribute is:
A) Row B) Column C) Table D) Database
18. A Tuple is:
A) Column B) Table C) Row D) Key
19. Domain means:
A) Number of rows B) Allowed values C) Primary Key D) Table name
20. Degree means:
A) Number of rows B) Number of columns

Class By :- Ajay Sir (10+ Years Experience)



C) Number of tables D) Number of keys

21. Cardinality means:

A) Number of rows B) Number of columns

C) Number of databases D) Number of attributes

22. Primary Key:

A) Allows duplicate values B) Contains NULL values

C) Uniquely identifies records D) Connects tables

23. Foreign Key:

A) Is always unique B) References another table's Primary Key

C) Is the same as Candidate Key D) Cannot repeat

24. Candidate Key is:

A) Selected key only B) Minimal Super Key C) Duplicate key D) Foreign Key

25. Alternate Key is:

A) Primary Key B) Remaining Candidate Key C) Foreign Key D) Composite Key

26. Which key connects two tables?

A) Candidate Key B) Foreign Key C) Alternate Key D) Unique Key

27. Degree depends on:

A) Rows B) Columns C) Records D) Tuples

28. Cardinality depends on:

A) Columns B) Rows C) Keys D) Domains

29. Which key cannot contain NULL values?

A) Foreign Key B) Primary Key C) Alternate Key D) Unique Key

30. Which is the most widely used Data Model?

A) Hierarchical B) Network C) Relational D) File

Written Questions Exercise

1. What is Database?
2. What is the purpose of a Database?
3. Define Data Redundancy.
4. What is Data Inconsistency?
5. Explain Database Abstraction.
6. What are the three levels of Database Abstraction?
7. Define Physical Level (Internal Level).
8. Define Logical Level (Conceptual Level).
9. Define View Level (External Level).
10. What is Database Independence?

Class By - Ajay Sir (10 - Years Experience)

11. What are the types of Database Independence?
12. Define Physical Data Independence.
13. Define Logical Data Independence.
14. What is a Data Model?
15. Explain the Hierarchical Data Model.
16. Define the Network Data Model.
17. Define the Relational Data Model.
18. What is a Relation?
19. Define Attribute.
20. What is a Tuple?
21. Define Domain.
22. What is Degree in a Relational Data Model?
23. What is Cardinality in a Relational Data Model?
24. Define Primary Key.
25. Define Foreign Key.
26. What is a Candidate Key?
27. What is an Alternate Key?
28. Differentiate between Primary Key and Foreign Key.
29. Differentiate between Degree and Cardinality.
30. Differentiate between Candidate Key and Alternate Key.

Relational Algebra

Relational Algebra is a **query language** used in **DBMS** to get data from a database. It tells the database **what operation to perform** on the tables to find the required information.

1. Selection (σ)

Selection (σ) is a relational algebra operation that **selects only those rows (records)** from a table that satisfy a given condition.

Example

Student Table

StudentID	Name	Age	City	Marks
101	Rahul	20	Ranchi	85

Class By :- Ajay Sir (10 - Years Experience)

102	Priya	19	Delhi	75
103	Amit	21	Ranchi	90
104	Neha	18	Patna	80

Query

σ City = 'Ranchi' (Student)

Result Table

StudentID	Name	Age	City	Marks
101	Rahul	20	Ranchi	85
103	Amit	21	Ranchi	90

Example 2

Query

σ Marks > 80 (Student)

Result Table

StudentID	Name	Age	City	Marks
101	Rahul	20	Ranchi	85
103	Amit	21	Ranchi	90

Example 3

Query

σ Age < 20 (Student)

Result Table

StudentID	Name	Age	City	Marks
102	Priya	19	Delhi	75
104	Neha	18	Patna	80

Class By :- Ajay Sir (10 - Years Experience)



Exercise Questions – Selection (σ)

Student Table

StudentID	Name	Age	Gender	City	Marks
101	Rahul	20	Male	Ranchi	85
102	Priya	19	Female	Delhi	75
103	Amit	21	Male	Ranchi	90
104	Neha	18	Female	Patna	80
105	Rohan	22	Male	Hazaribagh	65
106	Pooja	20	Female	Ranchi	95

Write the relational algebra query for the following:

1. Display all students from **Ranchi**.
2. Display all **Female** students.
3. Display students whose **Age is greater than 20**.
4. Display students whose **Marks are greater than 80**.
5. Display students from **Delhi**.
6. Display students whose **Age is less than 20**.
7. Display students whose **Marks are less than 70**.
8. Display all **Male** students.
9. Display students from **Patna**.
10. Display students whose **Marks are equal to 95**.
11. Display students whose **Age is equal to 20**.
12. Display students from **Hazaribagh**.
13. Display students whose **Marks are greater than or equal to 80**.
14. Display students whose **Age is less than or equal to 19**.

Class By :- Ajay Sir (10 - Years Experience)

15. Display students whose **City is not Ranchi**.

The Project Operator (π)

Projection **(π)** is a relational algebra operation that selects only the required **columns (attributes)** from a table. It does not display the **other columns**.

Syntax

π Column1, Column2, ... (Table Name)

Where:

- **π (Pi)** = Projection Operator
- **Column1, Column2** = Required columns
- **Table Name** = Name of the table

Example

Student Table

StudentID	Name	Age	City	Marks
101	Rahul	20	Ranchi	85
102	Priya	19	Delhi	75
103	Amit	21	Ranchi	90

Query

π Name, City (Student)

Name	City
Rahul	Ranchi
Priya	Delhi
Amit	Ranchi

Example 2

Query

π StudentID, Name (Student)

Result

Class By :- Ajay Sir (10 + Years Experience)

StudentID	Name
101	Rahul
102	Priya
103	Amit

Projection (π) – Practice Questions

Practice Set – 1

Student Table

StudentID	Name	Age	Gender	City	Marks
101	Rahul	20	Male	Ranchi	85
102	Priya	19	Female	Delhi	75
103	Amit	21	Male	Ranchi	90
104	Neha	18	Female	Patna	80
105	Rohan	22	Male	Hazaribagh	65

Write the relational algebra query for the following:

1. Display only the **Name** of all students.
2. Display only the **StudentID** and **Name**.
3. Display only the **Name** and **City**.
4. Display only the **Age** and **Marks**.
5. Display only the **Gender**.
6. Display only the **City**.
7. Display only the **StudentID**, **Name**, and **Marks**.
8. Display only the **Name**, **Gender**, and **City**.

Class By :- Ajay Sir (10+ Years Experience)

9. Display only the **StudentID** and **Age**.
10. Display only the **Marks**.

Practice Set – 2

Employee Table

EmpID	Name	Department	Salary	City
201	Ajay	HR	25000	Ranchi
202	Ravi	IT	40000	Patna
203	Pooja	Finance	35000	Delhi
204	Ankit	IT	45000	Ranchi
205	Neha	HR	28000	Patna

Write the relational algebra query for the following:

1. Display only the **Employee Name**.
2. Display only the **Department**.
3. Display only the **Name** and **Salary**.
4. Display only the **EmpID**, **Name**, and **Department**.
5. Display only the **City**.
6. Display only the **Department** and **Salary**.
7. Display only the **EmpID** and **City**.
8. Display only the **Name**, **Department**, and **City**.
9. Display only the **Salary**.
10. Display only the **EmpID** and **Name**.

Class By :- Ajay Sir (10 - Years Experience)

Practice Set – 3

Book Table

BookID	BookName	Author	Price	Category
1	DBMS	Navathe	450	Database
2	Python	Guido	550	Programming
3	C++	Bjarne	500	Programming
4	Java	James	650	Programming
5	SQL	Oracle	400	Database

Write the relational algebra query for the following:

1. Display only the **Book Name**.
2. Display only the **Author**.
3. Display only the **Book Name** and **Price**.
4. Display only the **Book Name** and **Category**.
5. Display only the **Author** and **Category**.
6. Display only the **BookID**, **Book Name**, and **Price**.
7. Display only the **Price**.
8. Display only the **BookID** and **Book Name**.
9. Display only the **Category**.
10. Display only the **Book Name**, **Author**, and **Category**.

UNION (∪) in Relational Algebra

- ✓ The **UNION** operation is used to **combine the rows of two relations (tables)** into one relation.
- ✓ It **removes duplicate rows** from the result.

Class By : Ajay Sir (10 - Years Experience)



RELATIONAL ALGEBRA – UNION (U)

UNION (∪) is used to combine the rows of two relations (tables) into a single relation.
It removes duplicate rows from the result.

TABLE A (Class A)

Roll	Name
1	Rahul
2	Amit
3	Neha

∪

TABLE B (Class B)

Roll	Name
3	Neha
4	Pooja
5	Ravi

→

A ∪ B (Union)

Roll	Name
1	Rahul
2	Amit
3	Neha
4	Pooja
5	Ravi

Neha is present in both tables, but appears only once in the result.

KEY POINTS

1. Both relations must have the same number of columns.
2. The data types of corresponding columns must be compatible.
3. The result will contain all unique rows from both relations.

ANOTHER EXAMPLE

TABLE C

ID	City
101	Ranchi
102	Patna
103	Delhi

∪

TABLE D

ID	City
103	Delhi
104	Kolkata
105	Bokaro

→

C ∪ D

ID	City
101	Ranchi
102	Patna
103	Delhi
104	Kolkata
105	Bokaro

Remember:
UNION (∪) = Combine rows of two tables + **Remove** duplicates.

Set Difference (−) in Relational Algebra

Set Difference is used to find the rows that are **present in the first table but NOT present in the second table.**

A − B = Rows in A that are not in B



RELATIONAL ALGEBRA – SET DIFFERENCE (–)

Set Difference (A – B) returns all the rows that are present in **A** but **NOT** present in **B**.

TABLE A

Roll	Name
1	Rahul
2	Amit
3	Neha
4	Pooja

4 rows

–

TABLE B

Roll	Name
3	Neha
4	Pooja
5	Ravi

3 rows

→

A – B (Set Difference)

Roll	Name
1	Rahul
2	Amit

2 rows (Result)

Rows 1 (Rahul) and 2 (Amit) are in Table A but not in Table B. So they appear in the result.

ANOTHER EXAMPLE

TABLE X

ID	City
101	Ranchi
102	Patna
103	Delhi
104	Kolkata

–

TABLE Y

ID	City
102	Patna
104	Kolkata
105	Bokaro

→

X – Y (Result)

ID	City
101	Ranchi
103	Delhi

Rows in X but not in Y

IMPORTANT POINTS

1. Both relations (tables) must have the same number of columns.
2. The data types of corresponding columns must be compatible.
3. The result contains only those rows that are in A but not in B.
4. Set Difference is NOT commutative. That means $A - B \neq B - A$

KEY IDEA: First table minus Second table → Keep only those rows which are not found in the Second table.

Cartesian Product (×) in Relational Algebra

The **Cartesian Product** operation combines **every row of the first table with every row of the second table**.

It is represented by the symbol **×**.

A × B = Every row of A is combined with every row of B.

Formula

If:

- Table A has **2 rows**
- Table B has **2 rows**

Then:

A × B = 2 × 2 = 4 rows

Class By :- Ajay S



RELATIONAL ALGEBRA – CARTESIAN PRODUCT (×)

DEFINITION: Cartesian Product (×) is an operation in which every row of the first table is combined with every row of the second table.

TABLE A : STUDENT

Roll	Name
1	Rahul
2	Amit

Total Rows = 2

×

TABLE B : COURSE

Course_ID	Course
C1	DCA
C2	ADCA

Total Rows = 2

→

A × B (CARTESIAN PRODUCT)

Roll	Name	Course_ID	Course
1	Rahul	C1	DCA
1	Rahul	C2	ADCA
2	Amit	C1	DCA
2	Amit	C2	ADCA

Total Rows = 2 × 2 = 4

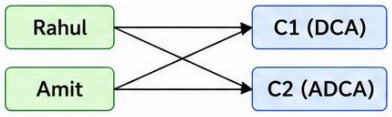
HOW IT WORKS ?

Each row of Table A is combined with each row of Table B.

- Rahul + C1 (DCA)
- Rahul + C2 (ADCA)
- Amit + C1 (DCA)
- Amit + C2 (ADCA)

VISUAL REPRESENTATION

Students (Table A) — Courses (Table B)



Every Student × Every Course = Combination

KEY POINTS

- If Table A has m rows and Table B has n rows, then A × B will have m × n rows.
- The result table contains all columns from both tables.
- Cartesian Product may produce many rows, so it is usually used before applying a condition (Join).

EASY TO REMEMBER: CARTESIAN PRODUCT = EVERY ROW × EVERY ROW

Intersection (∩)

Intersection is a relational algebra operation used to find the **common rows that are present in both relations (tables). Symbol: ∩**

A ∩ B = Rows that are present in both A and B.

Class By : Ajay Sir

RELATIONAL ALGEBRA – INTERSECTION (∩)

DEFINITION: Intersection (∩) is an operation in which we get the common rows that are present in both relations (tables).

TABLE A

Roll	Name
1	Rahul
2	Amit
3	Neha

Total Rows = 3

∩

TABLE B

Roll	Name
2	Amit
3	Neha
4	Pooja

Total Rows = 3

→

A ∩ B (INTERSECTION)

Roll	Name
2	Amit
3	Neha

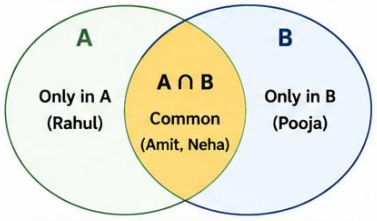
Total Rows = 2

HOW IT WORKS?

We compare rows of Table A with Table B. Rows that are common in both tables are selected.

- Rahul → only in A ✗
- Amit → in A and B ✓
- Neha → in A and B ✓
- Pooja → only in B ✗

VISUAL REPRESENTATION



Intersection = Common part of both

KEY POINTS

- Both tables must have the same number of columns and compatible data types.
- The result contains only those rows that are present in both tables.
- Duplicates are removed (set operation).
- Symbol: ∩ (Intersection)
- Example: A ∩ B

EASY TO REMEMBER: INTERSECTION (∩) = COMMON ROWS IN BOTH TABLES

Unit–3 Database: 50 Written Questions

A. Database Basics

1. What is a Database?
2. What is the purpose of a Database?
3. Define Data Redundancy.
4. What is Data Inconsistency?
5. Explain Database Abstraction.
6. What are the three levels of Database Abstraction?
7. Define Physical Level (Internal Level).
8. Define Logical Level (Conceptual Level).
9. Define View Level (External Level).
10. What is Database Independence?
11. What are the types of Database Independence?
12. Define Physical Data Independence.
13. Define Logical Data Independence.



14. What is a Data Model?
15. Explain the Hierarchical Data Model.
16. Define the Network Data Model.
17. Define the Relational Data Model.
18. What is a Relation?
19. Define Attribute.
20. What is a Tuple?
21. Define Domain.
22. What is Degree in a Relational Data Model?
23. What is Cardinality in a Relational Data Model?
24. Define Primary Key.
25. Define Foreign Key.
26. What is a Candidate Key?
27. What is an Alternate Key?
28. Differentiate between Primary Key and Foreign Key.
29. Differentiate between Degree and Cardinality.
30. Differentiate between Candidate Key and Alternate Key.

B. Relational Algebra

31. What is Relational Algebra?
32. What is the Selection operation in Relational Algebra?
33. What is the symbol used for Selection?
34. What is the purpose of the Selection operation?
35. What is the Projection operation in Relational Algebra?
36. What is the symbol used for Projection?
37. What is the purpose of the Projection operation?
38. Differentiate between Selection and Projection.
39. What is the UNION operation in Relational Algebra?
40. What is the symbol used for UNION?
41. What is Set Difference in Relational Algebra?
42. Explain **A – B = Rows in A that are not in B** with an example.
43. What is the Cartesian Product operation?
44. What is the symbol used for Cartesian Product?
45. Explain **A × B = Every row of A is combined with every row of B** with an example.
46. If Table A has 2 rows and Table B has 3 rows, how many rows will **A × B** produce?
47. What is the Intersection operation in Relational Algebra?

Class By :- Ajay Sir (10+ Years Experience)



48. What is the symbol used for Intersection?
49. Explain **A ∩ B = Rows that are present in both A and B** with an example.
50. Differentiate between **UNION (∪), Set Difference (−), Cartesian Product (×), and Intersection (∩)** with examples.

Class By :- Ajay Sir (10 - Years Experience)