Book : Database System Concepts
- Abraham Silberschatz

Data, Database, and DBMS

- **Data** is simply a value of real world things or objects (concrete form such as a person or abstract form such as an account number). (Entity - সত্তা)
  Example - Karim, A-101, 5000
  Rahim, H-314, 01712345678 etc

- **Information** is the processed form of data which is meaningful and more useful to the user. That is, data is the raw material of information and information is the final product of processing of data.

  That is,
  Data $\xrightarrow{\text{Processed}}$ Information

  Example: Result sheet of an Exam.

[N.B : "Data" is a plural noun - it is plural form of the noun "datum." However it is used with both singular and plural verb.]

Data is raw form of information.

- Database is a collection of interrelated data of a particular enterprise (service based or product based).
  Example - AgraniBank.accdb
  Grameenphone.accbd

- A DBMS is one or more database and a set of programs (instruction) to access (manipulate) data from those database.

  DBMS = Database(s) + Programs
  Example - Oracle, MySQL, MongoDM
  Database operation:
  → Insertion
  → Deletion
  → Update
  → Query

SQL = Structured Query Language
MS Access is a DBMS.
A query language, in a general sense, is a computer programming language specifically designed to request and retrieve data from database and information systems. It allows user to extract and manipulate information stored within these systems.

CamScanner

» DBMS :

- Collection of programs that manages database structure and controls access to data.
- Possible to share data among multiple application or user.
- Makes data management more efficient and effective

DBMS = Database(s) + Programms

» Database Applications:

- Banking : all transactions
- Airliner : reservations, schedules
- Universities : registration, grades
- Sales : customers, products, purchases
- Manufacturing : production, inventory, orders, supply chain
- Human resources : employee records, salaries, tax deductions

Database touch all aspects of our life.

এটা উত্তর দিবো না
তবে সংজ্ঞাটা নিবে

○ What is DBA? (Database Administrator)

→ In the realm of Database Management Systems (DBMS), a Database Administrator (DBA) is a crucial role responsible for the overall health, performance, security, and maintenance of an organization's databases.

Responsibilities of a DBA:

- Database Design and Implementation
- Performance Optimization
- Security and Access Control
- Backup and Recovery
- User Support and Training
- Staying Updated.

1.12.2 Database Administrator

One of the main reasons for using DBMSs is to have central control of data and the programs that access those data. A person who has such central control over the system is called a Database Administrator (DBA). The functions of DBA include:

- Schema definition
- Storage structure and asscess defination.

- Schema and physical-organization modification.
- Granting of authorization for data access.
- Routine maintenance

Types of Database

- Single-user:
  - Support only one user at a time
- Desktop:
  - Single-user database running on a personal computer
- Multi-user:
  - Supports multiple users at the same time
- Workgroup:
  - Multi-user database that supports a small group of users on a single department.
- Enterprise:
  - Multi-user database that supports a large group of user or an entire organization.

■ Can be classified by location:

- Centralized
  - Supports data located at a single site.
- Distributed
  - Supports data distributed across several sites

CamScanner

- Can be classified by use
  - Transactional (or production):
    - Supports a company's day-to-day operations
  - Data warehouse:
    - Stores data used to generate information required to make tactical or strategic decisions
    - Often used to store historical data
    - Structure is quite different.

This typical "file-processing system" is supported by a conventional operating system. The system stores permanent records in various files, and it needs different application programs to extract records form, and add records to, the appropriate files. Before DBMS were introduced, organizations usually stored information in such systems.

Major Drawbacks of Typical or Conventional or Traditional File Processing System:

1. Data Redundancy (অতিরিক্তা বা বাহুল্যতা) and inconsistency (অসামঞ্জস্যতা)

The same information may be duplicated in several places (files). This redundancy leads to higher storage and access cost. In addition, it may lead to data inconsistency; that is, the various copies of the same data may no longer agree.

2. Difficulty in accessing data

The point here is that conventional file-processing environments do not allow needed data to be retrieved in convenient and effective manner. More responsive data-retrieval systems are required for general use.

3. Data isolation (বিক্ষিপ্ত)

Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropri

4. Integrity problem (সততা সমস্যা)

The data values stored in the database must satisfy certain types of consistency constraints. (সামঞ্জস্যের সীমাবদ্ধতা). In a DBMS, data integrity refers to the accuracy, consistency, and trustworthiness of the data stored within the database.

5. Atomicity Problems

Atomicity is a fundamental property of transactions in a DBMS. It ensures that a transaction is treated as a single, indivisible unit of work. An atomicity problem occurs when a transaction fails to be atomic. This means the transaction is not completed entirely, and the database is left in an inconsistent state.

6. Concurrent-access anomalies

Concurrent access anomalies occur in DBMS (Database Management System) when multiple transactions access and modify the same data simultaneously. These anomalies can lead to inconsistencies in the data, resulting in inaccurate information and unexpected outcomes. Example: Two people booking the last ticket on a flight at the same time might both get a confirmation, but only one can actually get the seat.

7. Security Problem

Not every user of the database system should be able to access all the data. For example, in a university, payroll personnel need to see only that part of the database that has financial information. They do not need access to information of academic records. But since application programs are added to the file-processing system in an ad hoc manner, enforcing such security constraints is difficult.

CamScanner

File Processing < DBMS < Data Mining < Big Data
Small Data < large Data < Huge Data < Huge Huge Data

- Data mining কাজ করে Data এর মাঝে থাকা Hidden pattern নিয়ে
- DBMS কাজ করে structured Data নিয়ে
- Data Mining / Big Data কাজ করে Unstructured Data নিয়ে।

Field

Student

Name	Roll	Mark
Karim	101	67
Rahim	102	85
Kamal	103	58
Jamal	109	75

→ Record / Tuple

There are 3 field with 4 records

Tabulated form of data in Relational Model RDBMS

RDBMS
↳ Present days database model, Tabular form of Data

4 Type of Database model.
- Relational Model
- Hierarchical Model
- Network Model
- Object-Oriented Model

CamScanner

E-R Model: An Entity-Relationship (ER) Model is a conceptual data modeling technique used for designing relational database. It provides a visual representation of the entities (real-world object) you want to store data about, the attributes of those entities, and the relationships that exist between them.

Topic	RDBMS	E-R Model
Column	Field	Attribute
Row	Record or Tuple	Entity
Table	Relation	Entity Set

Entity: An entity is a thing or object that is distinguishable from all other objects in the real world.
Example: A person or a bank account number.

Attribute: Descriptive properties of an entity is called attribute. (গুন বা বৈশিষ্ট্য)
Example: Name, Roll, Marks of a student.

CamScanner

1.3 View of Data (পরীক্ষায় আসে 5-marks)

Data Abstraction : (সারসংক্ষেপ, সারমর্ম, নির্যাস)

In the context of database systems, data abstraction refers to hiding the complexity of how data is physically stored and managed from users. It provides users with different levels of views to interact with the database based on their needs.

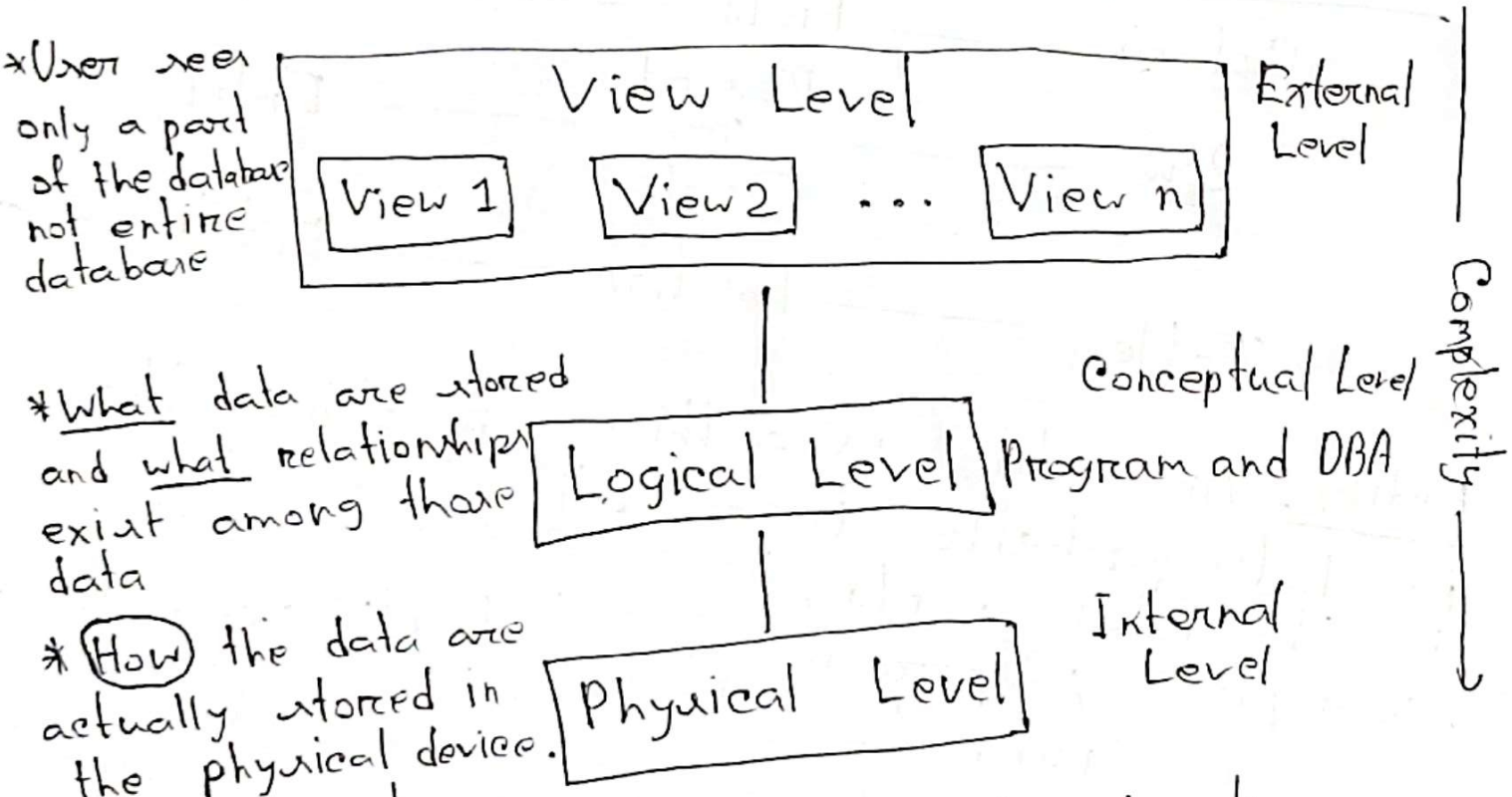


Fig: Three level of abstraction

Why Data Abstraction?
- Efficiency
- Usability
- Maintainability

Benefits of Data Abstraction
- Simplified User Interaction
- Physical Data Independence
- Improved Security.

Schema and Instance (2-3 marks)

Schema - নীল নকশা

Schema: The overall design of a database is called its schema. A database schema corresponds to the variable declaration in a program.
Student (Name, Roll, Marks)
কোন data insert করার পূর্বে table তৈরির command execute করলে, যে blank table তৈরি হয় তাকে Schema বলে।

Instance: The collection of information stored in the database at a particular moment is called an instance of the database.
- Schema তে Data insert করলে সেটা Instance হয়ে যায়।

CamScanner

1.3.3 Data Models

Underlying the structure of a database is the data model: a collection of conceptual tools for describing data, relationships, data semantics, and consistency constraints.

There are a number of different data models:

- Relational Model: The relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name. Table are also known as relations.
- Entity-Relationship Model: The entity-relationship (E-R) data model uses a collection of basic objects, called entities, and relationships among these objects. An entity is a "thing" or "object" in the real world that is distinguishable from other objects.
- Object-Based Data Model
- Semistructured Data Model (XML)

1.9 Database Architecture

The architecture of database system is greatly influenced by the underlying computer system on which the database system runs. Database systems can be centralized, or client-server, where one server machine executes work on behalf of multiple client machines.

Tier or Layer:

* Single layer → 1-Tier Architecture
* Multi Layer
  - → 2-Tier Architecture
  - → 3-Tier "
  - → N-Tier "

] Network based Architecture

* Single layer or 1-Tier Architecture:

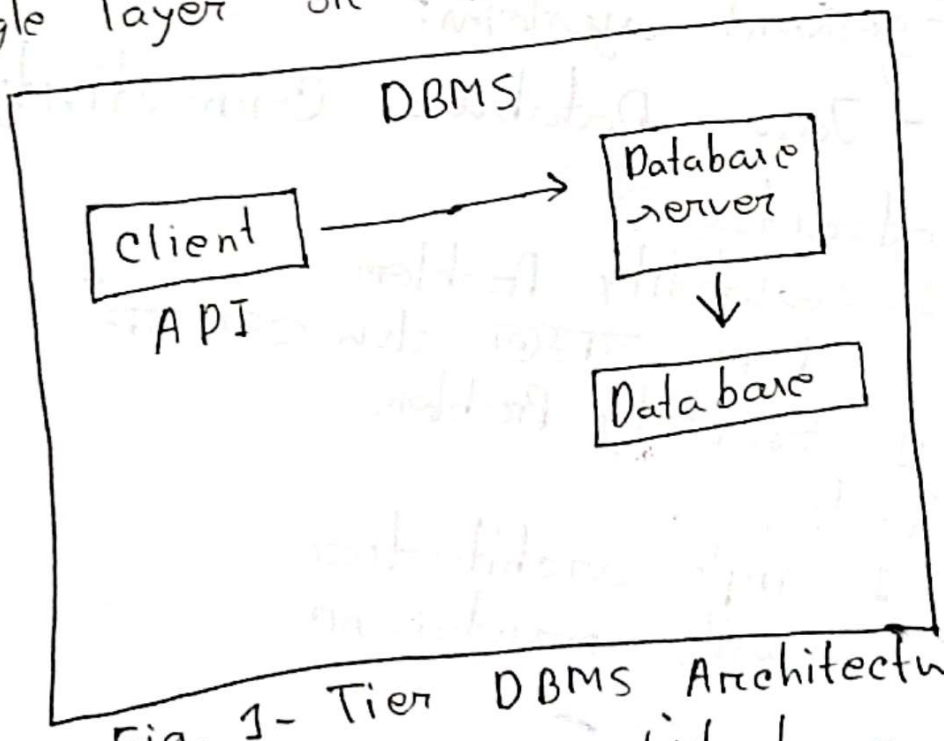
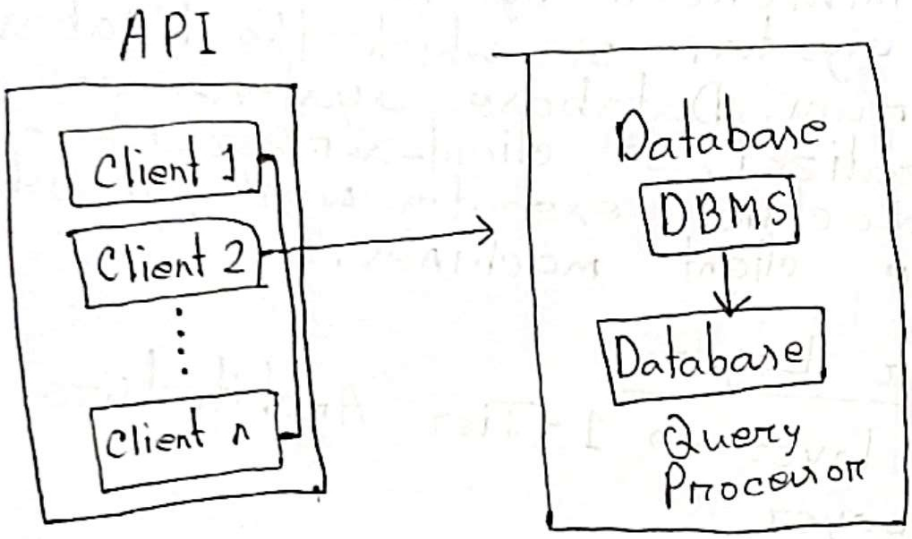


Fig. 1-Tier DBMS Architecture

* Personal computer এর architecture → 1

2 Layers or 2-Tiers Architecture



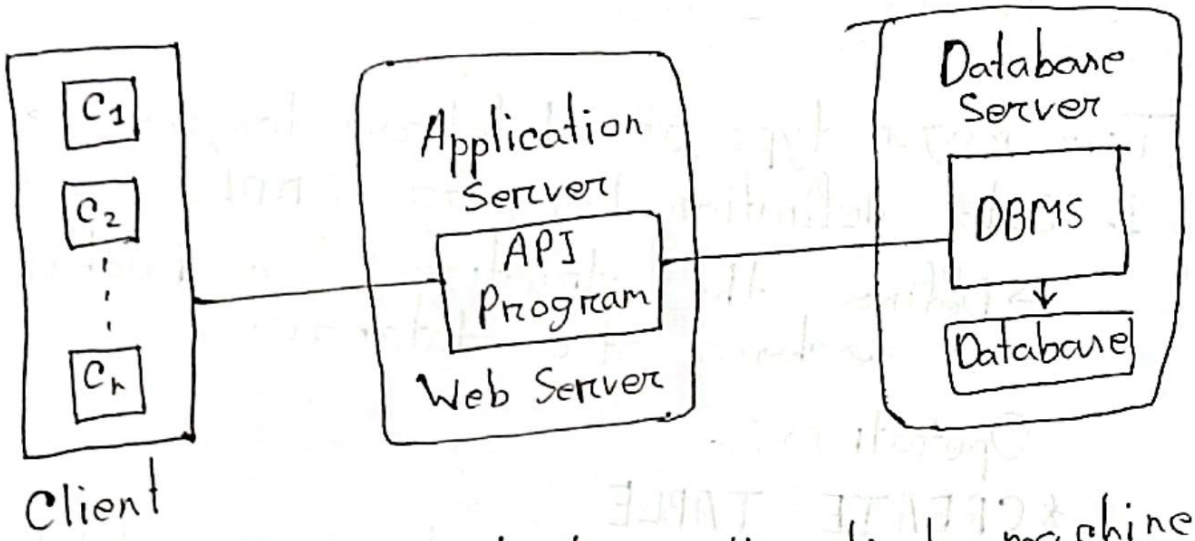
ODBC : Open Database Connectivity is a standard application programming interface (API) that allows applications to access data from various database management systems.

JDBC - Java Database Connectivity

* Disadvantage:
  1. Scalability Problem
     User বাড়লে slow হয়ে যায়
  2. Security Problem

* Advantage:
  1. Simple architecture
  2. Easily maintanance

* 3-Tier Architecture



In a 3-Tier Architecture, the client machine acts as merely a front end and does not contain any direct database calls. Instead, the client end communicates with an application server, usually through a forms interface. The application server in turn communicates with a database system to access data.

1.4 Database Language

Two major type of database languages:

1. Data definition Language (DDL)
   → Defines the structure of a database or schema of a database.

   Operations:
   * CREATE TABLE
   * ALTER TABLE (structure ক্ষে update)
   * DROP TABLE

2. Data Manipulation Language (DML)

   Operations:
   * Data insertion : INSERT INTO
   * Data Query : SELECT FROM WHERE
   * Data Update : UPDATE SET
   * Data Deletion : DELETE FROM

DDL (User)	DML (User)
* DBA * Database Designer	* End user

Minor types of DB languages:

1. Data Control language
   - Set Privilege to the users as well as DB.
2. Transaction Control Language
3. Operates on data transaction
4. Commit
5. Roll back

1.4.1 Data-Manipulation Language

A data-manipulation language (DML) is a language that enables users to access or manipulate data as organized by the appropriate data model.

- Retrieval of information stored in the DB.
- Insertion of new information into the DB.
- Deletion of information from the DB.
- Modification of information stored in the DB.

There are basically two types of DML:

- Procedural DMLs require a user to specify what data are needed and how to get those data.
- Declarative DMLs (nonprocedural DMLs) require a user to specify what data are needed without specifying how to get those data.

We specify the storage structure and access methods used by the database system by a set of statements in a special type of DDL called a data storage and definition language.

CREATE TABLE

Syntax:

```
create table table_name (
    column1   Data_type (size),
    column2   Data_type(size),
       :
    columnN   Data_type(size))
```

Code:

```
CREATE TABLE student (
Name varchar(20),
Roll int ; ← default 11 digit
GPA float(3,2)
);
```

Student

Name	Roll	GPA

NOTE:

char(n) → fixed length n
varchar(n) → variable length char at most n

data type:
1 * String
2 * Numeric
3 * Date Time

* Text type অনেক, বড় type.

Date Time: Date type:

1. date : yyyy-mm-dd | 2024-05-14
2. Time : hh:mm:ss | 11:59:49
3. date time : yyyy-mm-dd hh:mm:ss
4. year : yyyy

Insert Value / Record in a Table:

```
INSERT INTO TABLE_Name
(column1, column2, ... columnN) ← optional
VALUES (value1, value2, ... valueN);
```

Ex:

```
INSERT INTO Student (Name, Roll, GPA)
VALUES (KARIM, 101, 3.75);
```

Query:

Syntax:

```
SELECT Field1, Field2, ... FieldN
FROM Table_Name
WHERE Condition;
```

Ex-1:

```
SELECT * FROM Table-Name;
```

↳ সব দেখার জন্য

Database keys:

Key: A key is one or a set of attributes used to uniquely identify records or tuples of a table.

Roll	Name	Age	City
101	Karim	20	Dhaka
102	Rahim	30	Rajshahi
103	Karim	20	Dhaka

1. Super key
2. Candidate key
3. Primary key
4. Alternate key
5. Foreign key
6. Composite key

☐ **Super key:**

- A super key is a set of all key combinations those can be used to uniquely identify records in a table.

বইয়ে আছে, A super key is a set of one or more attributes that, taken collectively, allow us to identify uniquely a tuple in the relation.

{Roll} ← SK, {Roll, Name} ✓ SK, {Roll, Age} ← SK, {Roll, City} ← SK
{Roll, Name, Age}, {Roll, Name, City},
{Roll, Age, City}, {Roll, Name, Age, City}

- যদি সব attribute primary key হয়, তবে super key = $2^n - 1$ টি।

□ Candidate key:

- A candidate key is a minimal super key that is as minimum as possible super key is a candidate key. A table can have more than one candidate key.
- Proper subset of a super key is not a super key is called a candidate key.

$A = \{1, 2, 3\}$ $B = \{1, 2, 3\}$

$\Rightarrow A \subseteq B$ and $B \subseteq A$

$\therefore A = B$

$C = \{1, 2\}$

$\therefore C \subset A$

$\Rightarrow C \subset A$ and $A \not\subset C$

$\Rightarrow C \subset A$

Example:

Let, Super key A = A

A, AB, AC, ABC

A is candidate key.

- একটি টেবিলের অনেক Candidate key এর মধ্যে একটি হবে primary, বাকি গুলোকে Alternate key বলা হয়।

Primary key:

A primary key is one or more attributes used to uniquely identify ~~record~~ records in a table.

* It cannot contain duplicate value.
* It cannot contain NULL value.
  - That is, the candidate key which does not contain NULL value is called a primary key.

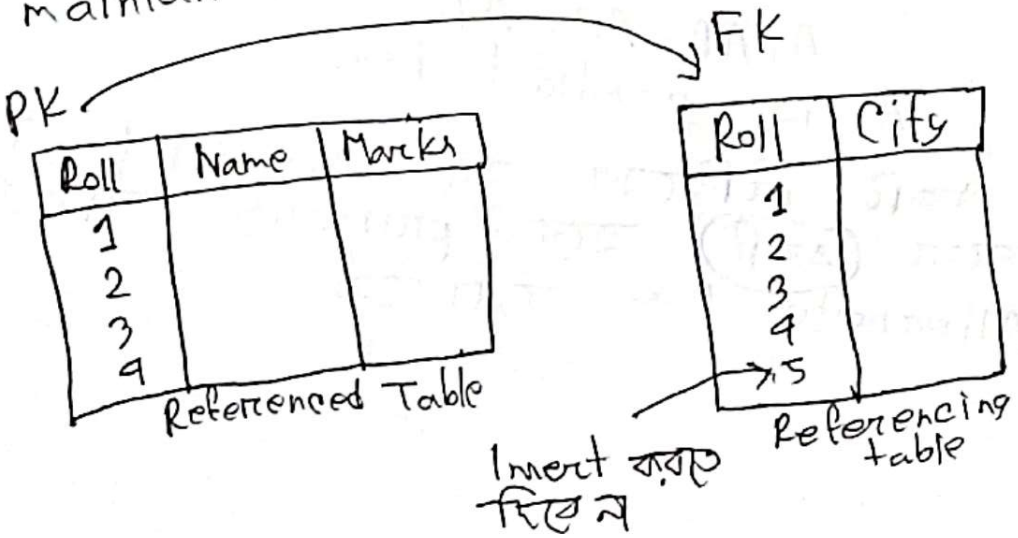
Composite key:

Composite key is a combination of keys.

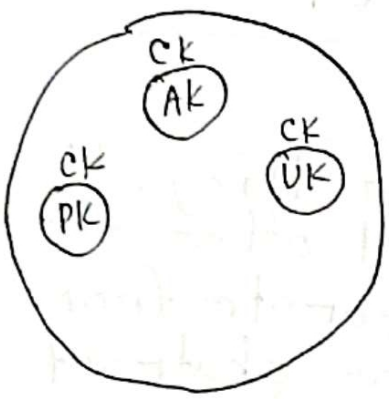
Foreign key:

A foreign key of a table is a key which references the primary key of another table or same table.
↑ exceptional case

- That is a foreign key is used to maintain referential integrity.



Super key



CK = Candidate key
PK = Primary key
UK = Unique key
AK = Alternat ke

Primary key থাকলে composite key দরকার হয় না।

- Primary = Unique + Not null
- Unique key = Unique + 1 Null Record.

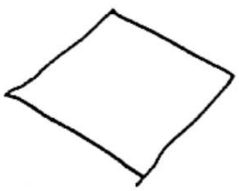
CHAPTER - 7

Database Design and the E-R Model

Entity:

Real world objects or things that is distinguishable from all other objects. An entity can be a concrete from such as a person or an abstract form, such as an account-no.

Table $\xrightarrow[\text{Conceptual}]{\text{Pictorial representation}}$ E-R model

Symbols	Meaning
	• Entity set (Table)
	• Attribute (Columns)
	• Link or connection between Entity set and attributes or Entity set and relationship set.
	• Relationship set

Attribute:

An entity is represented by a set of attributes. Attributes are descriptive properties possessed by each member of an entity set.

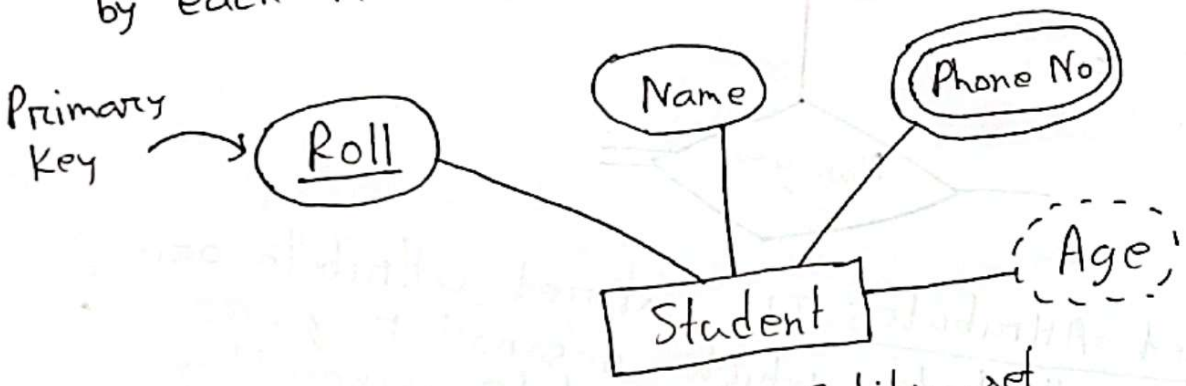
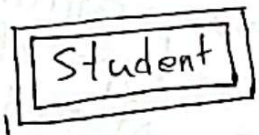
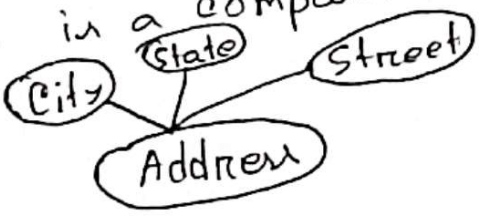


Fig: A student Entity set

NOTE: Primary key না থাকলে  এরকম হলে। একে weak entity set বলে।

- **Simple attribute**: Attributes that cannot be further subdivided are called simple/atomic attributes.
  Example: Phone number, PIN code
- **Multi-valued attribute**: Attributes having a set of values for a single entity is called a multi-valued attribute.
  Example: Mob_no, Email_id
- **Derived attribute**: When one attribute value is derived from the other is called a derived attribute.
  Example: Age
- **Composite attribute**: An attribute that can be split into components is a composite attribute.
  Example:



* Descriptive attribute : Descriptive attribute give information about the relationship set.

Example:

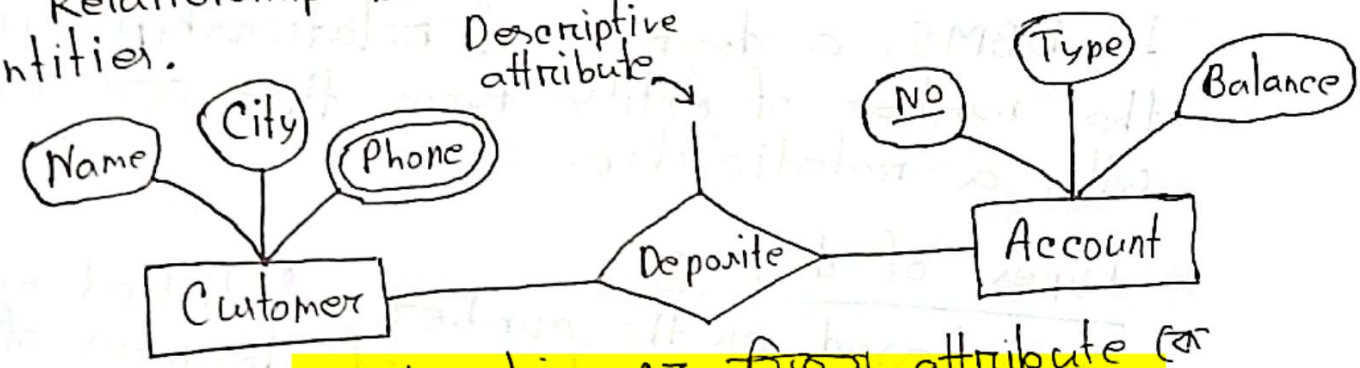
Stored Attribute: The stored attribute are those attribute which doesn't require any type of further update since they are stored in the database.

Example: DOB

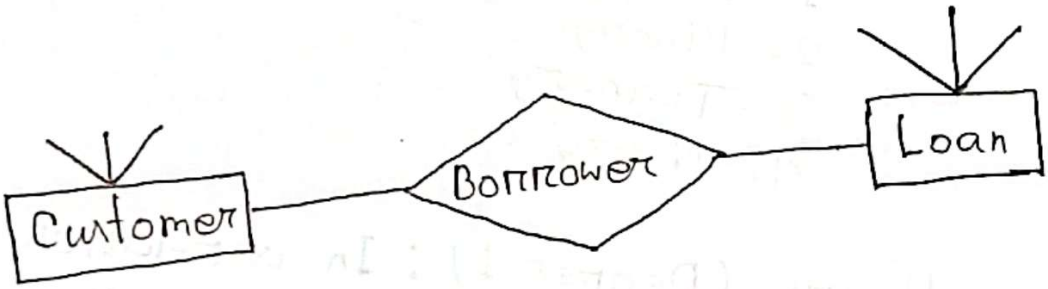
Entity are attribute.
Entity set are table.

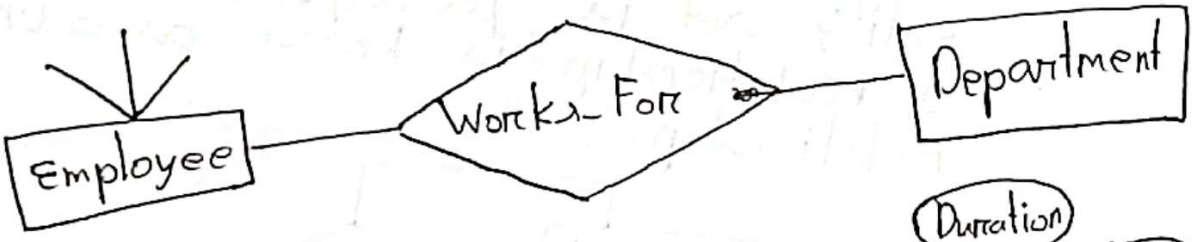
7.2.2 Relationship Sets

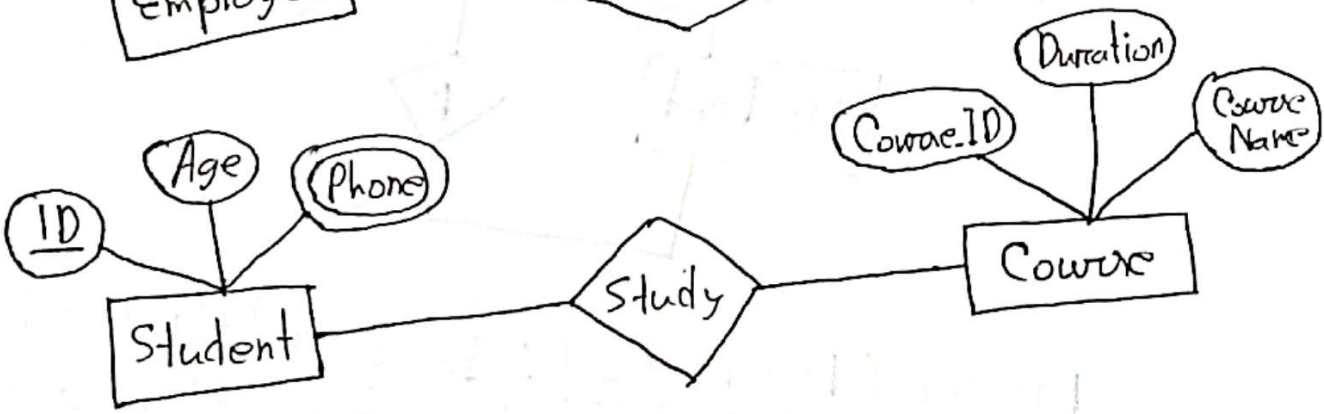
A Relationship is an association among several entities.



NOTE : Relationship এর নিজস্ব attribute কে descriptive attribute বলা হয়।







CamScanner

Degree of Relation in DBMS

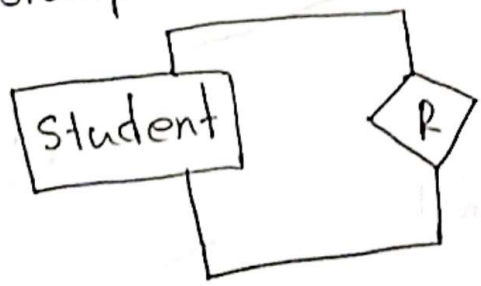
In DBMS, a degree of relationship represents the number of entity types that are associated with a relationship.

Types of degree

Now, based on the number of linked entity types, we have 4 types of degrees of relationships.

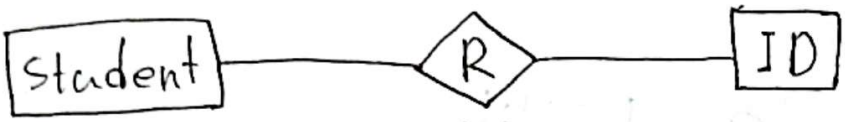
1. Unary
2. Binary
3. Ternary
4. N-ary

Unary (Degree 1): In a relation only one entity set is participating then such type of relationship is known as a unary relationship.

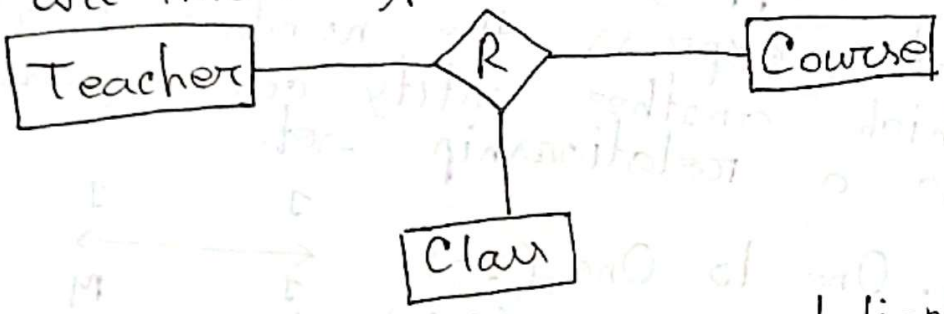


Binary (Degree 2): In a relation when two entity sets are participating then such of type is known as binary relationship. This is the most used relationship.

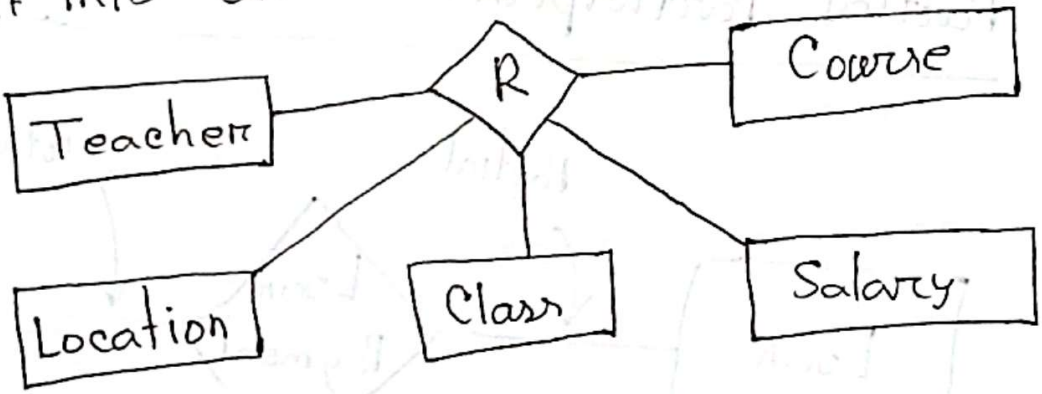
CamScanner



Ternary (Degree 3): In the Ternary relationship, there are three type of entity associates.



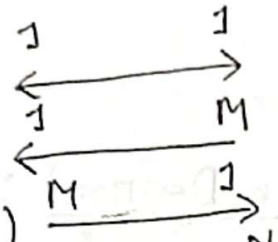
N-ary (n Degree) : In the N-ary relationship, there are n types of entity that associates. So, we can say that an N-ary relationship exists when there are n types of entities. There is one limitation of the N-ary relationship, as there are many entities so it is very hard to convert into an entity, relational table.



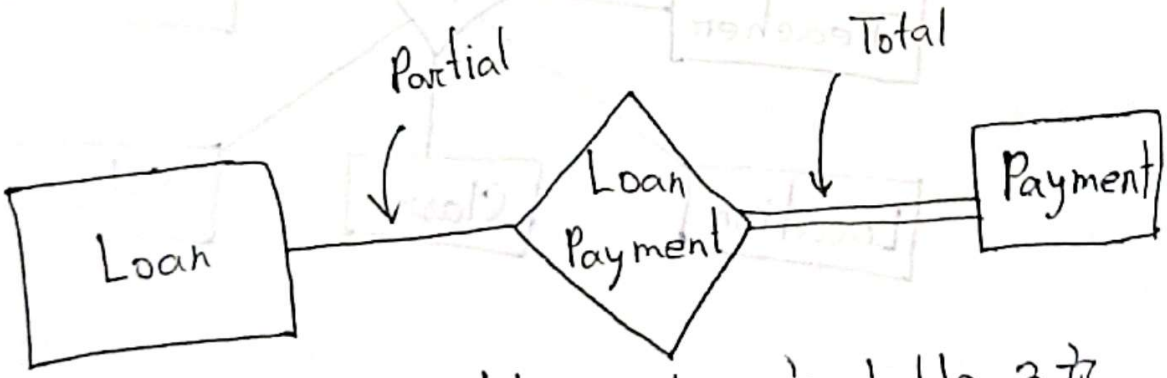
Mapping Cardinality

- Mapping cardinality or cardinality ratio express the number of entities to which another entity can be associated via a relationship set.

1. One to One (1-1)
2. One to Many (1-M)
3. Many to One (M-1)
4. Many to Many (M-N)



Partial Participation Vs Total Participation

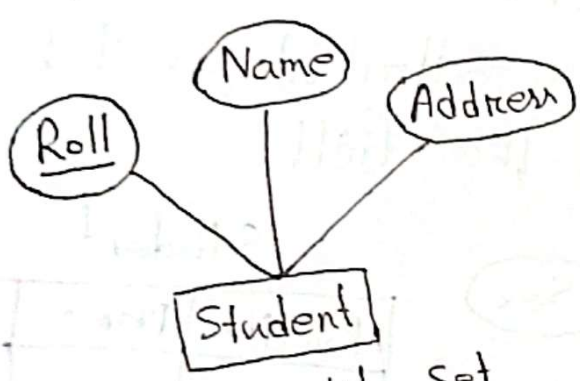


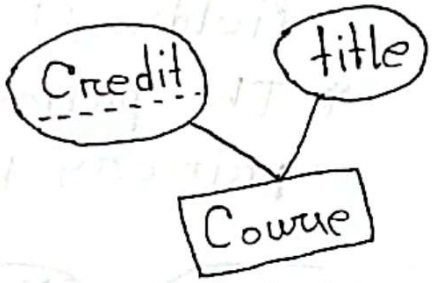
NOTE: এখানে entity set 2টা table 3টা

এখানে সব Payment এর ই individual Loan আছে। কিন্তু সমস্ত Loan এর payment সম্পন্ন হয়নি।

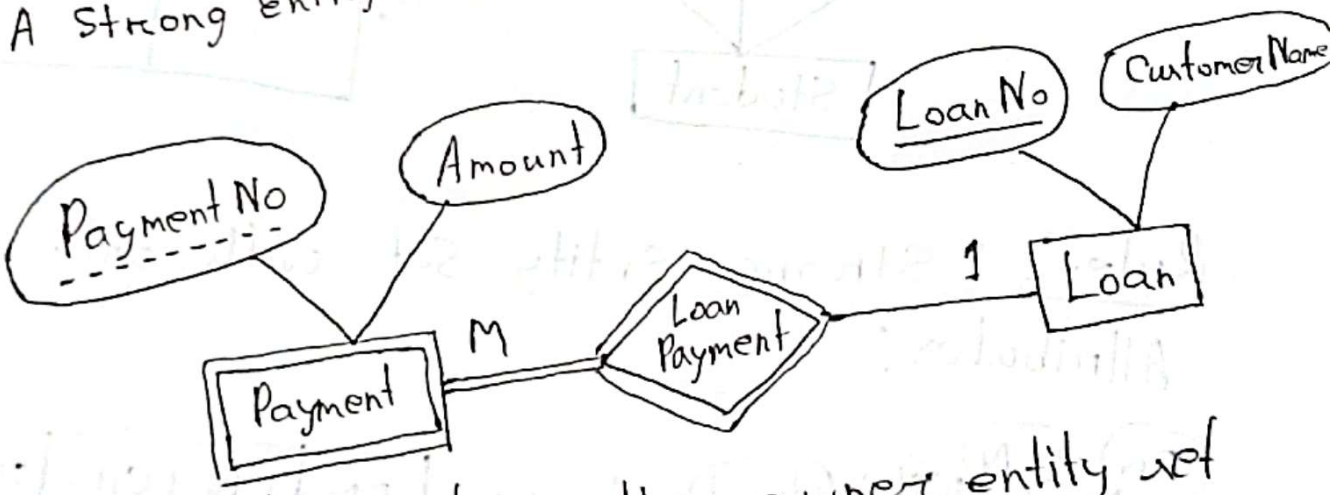
Strong Entity Set Vs. Weak Entity Set

- **Strong Entity Set:** An Entity set which has a primary key is called a strong entity set.
- An entity set which does not have a primary key is called a weak entity set.





A Strong Entity Set



- Loan entity set is the owner entity set of Payment entity set.
- Owner entity set / Indentifying entity set.
- Identifying relationship.
- Weak relationship এর descriptive attribute থাকবে না, strong এর থাকবে।

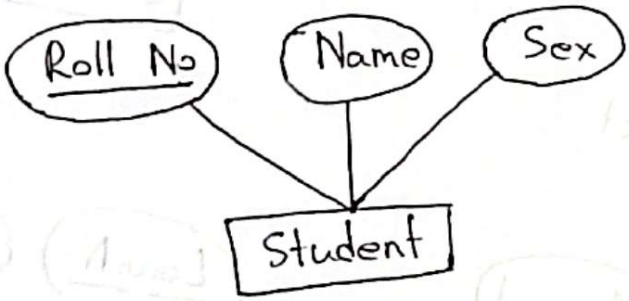
CamScanner

ER Model to Relational Model Conversion

Rules to Convert E-R Diagram into Tables

Rule 1: Strong Entity Set With only simple Attribute

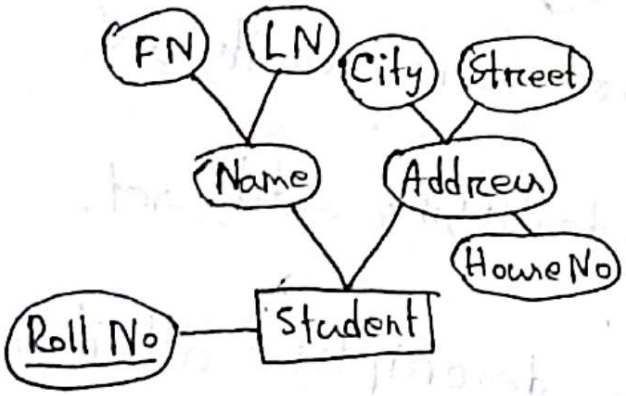
* Attributes of the entity set will be the fields of the table
* The primary key attribute will be the primary key of the field.



Student

Roll No	Name	Sex

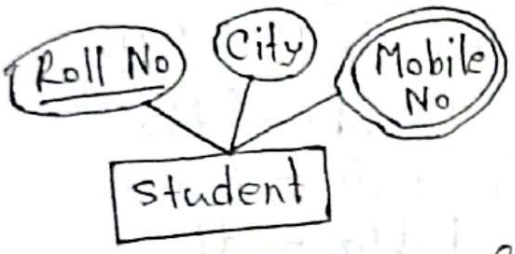
Rule-2: Strong Entity Set with only composite Attributes:



Roll No	FN	LN | City	Street	House

leaf level গুলো field হিসাবে use হবে

Rule-3: Strong Entity Set with Multivalued Attributes:



Student

Roll No	City

Student Mobile

Roll No	Mobile

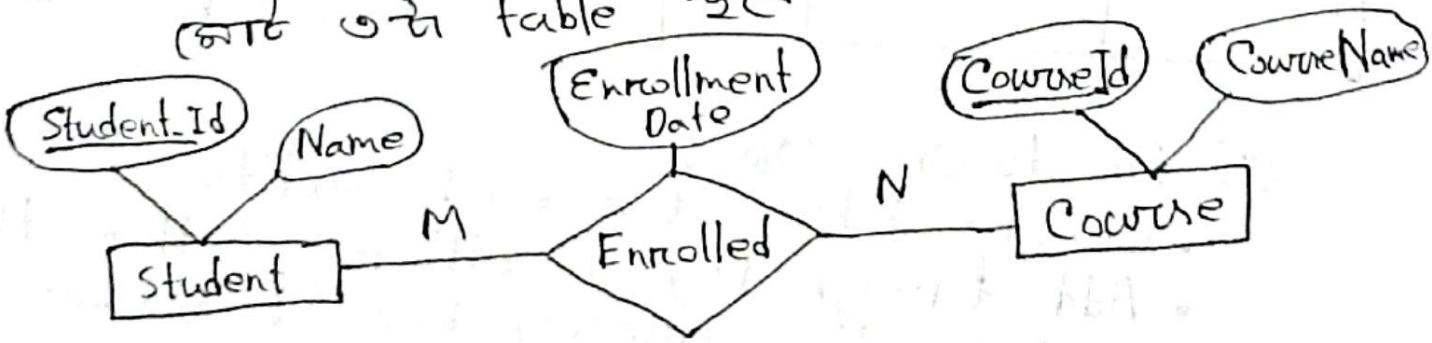
Separate table for each multivalued attribute.

Rule-4: Translating Relationship Set into a Table:

a) Converting many-to-many relationship:

- Create a new table for each many-to-many relationship, using the primary key from M-side and the N-side as foreign keys.
- Declare Primary for the new tables foreign key combination
- If your Relationship have properties, insert those in the new table as well.

এবার এর table হবে



Student

Student_Id	Name

Enrolled

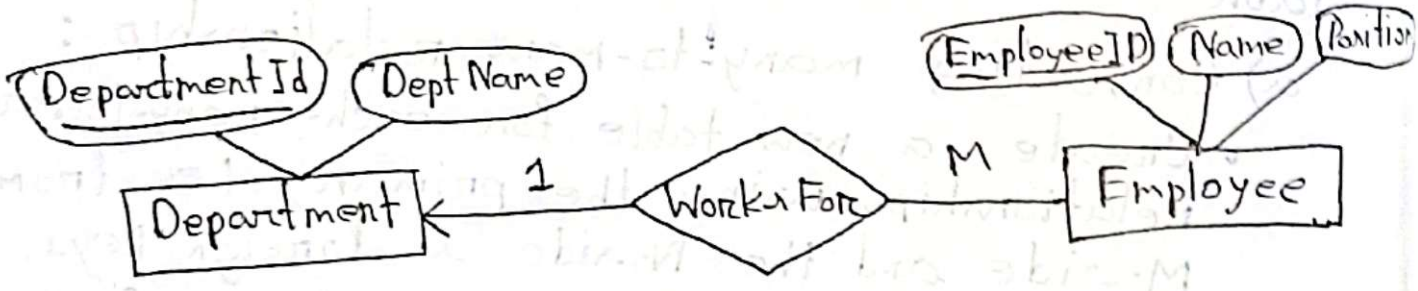
Student_Id	Course_Id	En. Date

Course

CourseId	Course Name

(b) Many-to-one হলে / One-to-Many হলে :

- Create Tables for each entity
- Add a foreign key column in the table representing the "many" side of the relationship. This column will reference the primary key of the table on the "one" side.
- If the relationship has attributes, include them in the table on "many" side.



Department

DeptId	Dept Name

Employee

EmployeeID	Dept Id	Name	Position

(c) One-to-One হলে :

- Create tables for each entity
- Add foreign key to either of them that refers the primary key of another table.
- Alternatively, combine the two entities into a single table if they are tightly coupled.

Either,

AR($\underline{a_1}$, a_2, a_3, b_1)

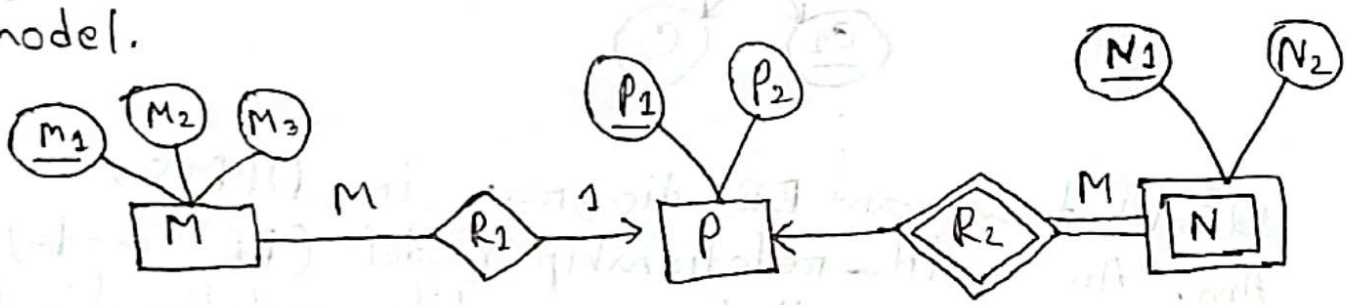
B(b_1, b_2, b_3)

Or,

BR($\underline{b_1}$, b_2, b_3, a_1)

A($\underline{a_1}$, a_2, a_3)

Q1. Find the minimum number of tables required for the following ER diagram in relational model.



Sol^n:

M($\underline{m_1}$, m_2, m_3)

R_1(m_1, P_1)

P($\underline{P_1}$, P_2)

R_2($\underline{P_1}$, N_1)

N($\underline{N_1}$, N_2)

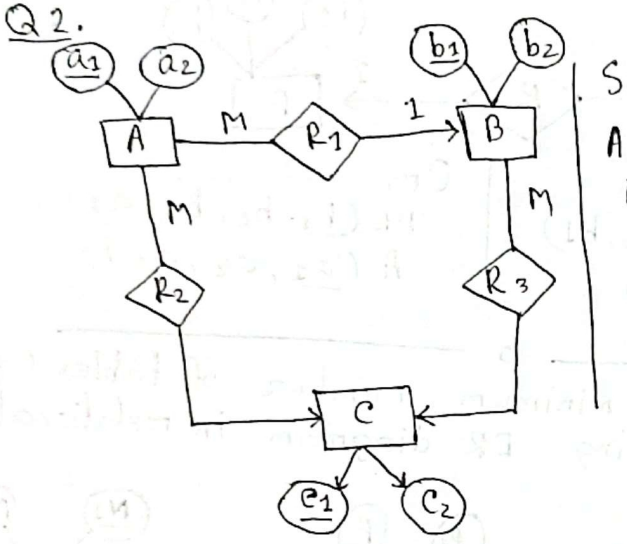
MR$_1$($\underline{m_1}$, m_2, m_3, P_1)

P(P_1, P_2)

R_2N($\underline{n_1}$, n_2, P_1)

∴ minimum number of table is - 3

Q 2.



Solve:

$AR_1R_2(\underline{a_1}, a_2, b_1, c_1)$

$B(\underline{b_1}, b_2)$

$C(\underline{c_1}, c_2)$

$R_3(b_1, c_1)$

Q3. What is an ER diagram in DBMS?

Ans. An entity-relationship model (ER mode) uses a diagram called an entity relationship diagram (ER Diagram) to illustrate the structure of a database.

Q4. How is an ER diagram reduced to a table?

Ans. The notations in an ER diagram can be used to represent the database, and these notations can be reduced to a set of tables.

Q5. What is the purpose of the ER-diagram?

Q6. Why is the ER diagram important?

Extended E

The Extended E-R mo
ER model by adding
model by adding a
more complex relatio
constraints. These fe
real-world scenario
ER model falls shor

Key Features of the

1. Specialization
2. Generalization
3. Aggregation

Specialization: The

single entity set in
entity sets based
characteristics. (The

- It is a top-down a
- Here attributes a
  are inherited by
- Here in schema
  produced.

Extended E-R Features

The Extended E-R model extends the traditional ER model by adding additional constructs to model by adding additional constructs to model more complex relationships, inheritance, and constraints. These features are useful in real-world scenarios where the basic ER model falls short.

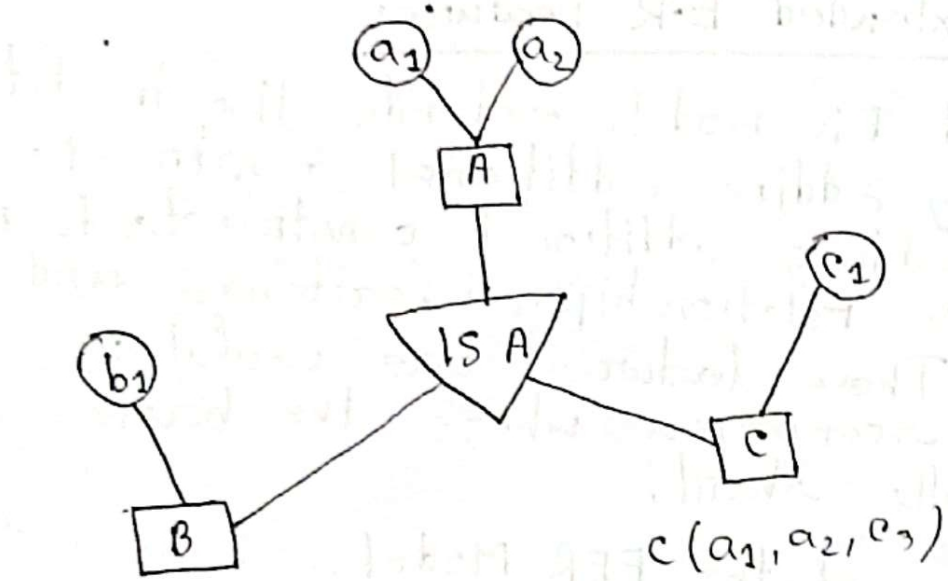
Key Features of the EER Model

1. Specialization
2. Generalization
3. Aggregation

Specialization: The process of dividing a single entity set into multiple, more specific entity sets based on some distinguishing characteristics. (Inheritaze)

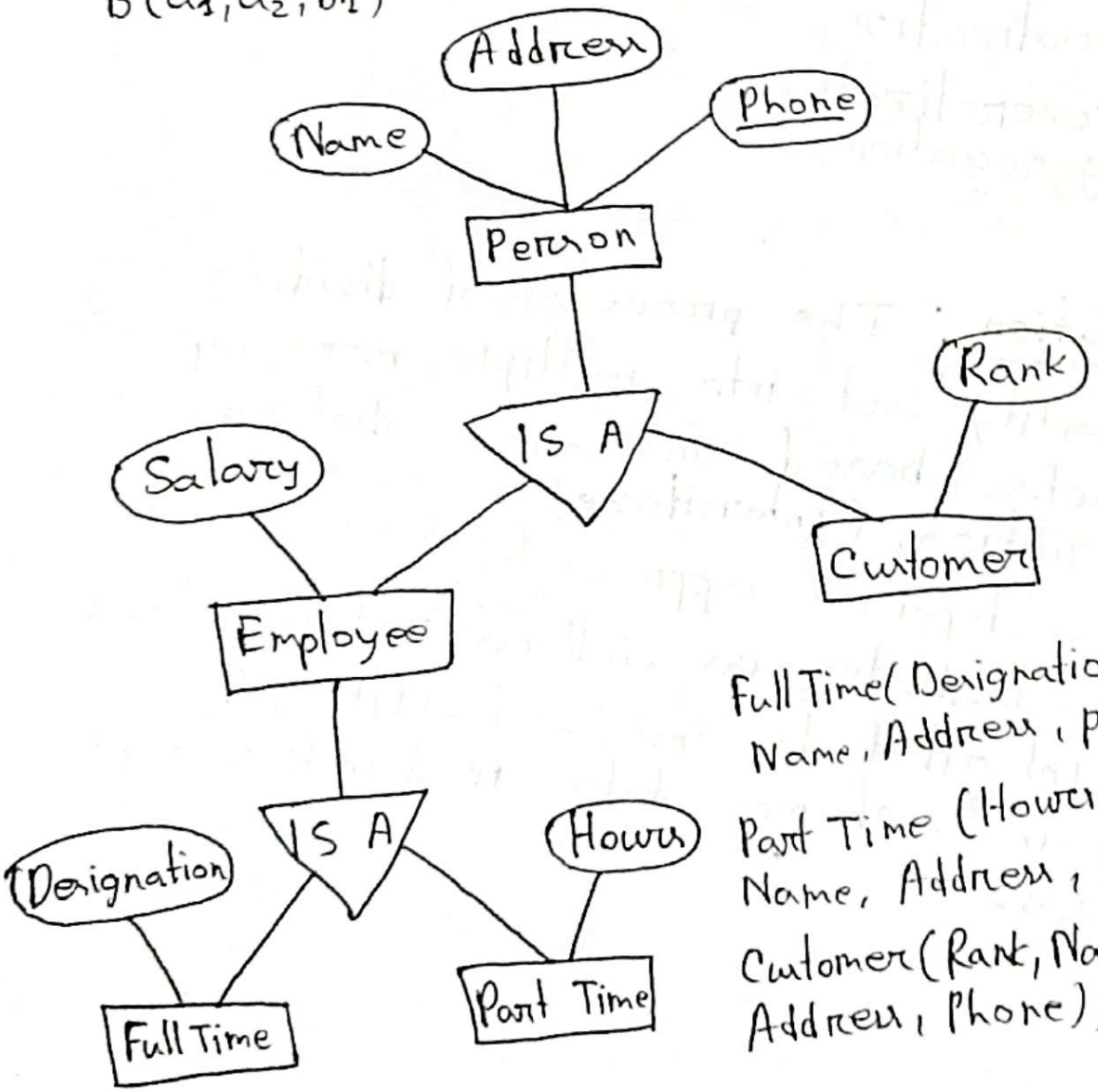
- It is a top-down approach.
- Here attributes as well as relationships are inherited by the sub-entity sets.
- Here in schema data redundancy is produced.

CamScanner



B (a_1, a_2, b_1)

C (a_1, a_2, c_1)



Full Time (Designation, Salary, Name, Address, phone);

Part Time (Hours, Salary, Name, Address, phone);

Customer (Rank, Name, Address, Phone);

Generalization: The process of combining two or more entity sets into a single, generalized entity set.

- Bottom up approach
- A higher entity set is made from two or more lower entity sets depending on common or gener attributes.

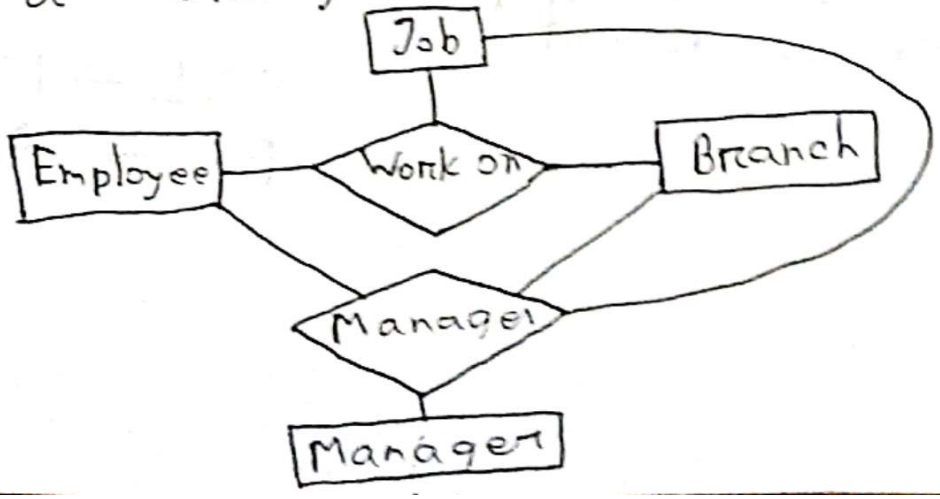
Full Time (Name, Address, phone, Salary, Designation)
Part Time (Nam, Addres, phone, Salary, Hour)

↓

Person (Name, Address, phone)
Fulltime (Phone, Salary, Design)
Customer (Rank, phone)

Aggregation: Aggregation is an abstraction that allows modelling a relationship as an entity.

- Relationship among relationships
- Here a relationship is treated as a entity set.

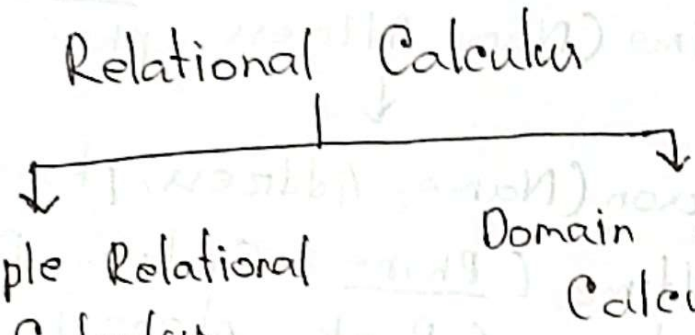


Chapter-3 | Relational Model

Relational Algebra

Query language: A query is a request to a database for retriving information.

i) Procedural (What and How)] → SQL
ii) Non-procedural (What)



Note: Relational algebra → থেকে SQL আসবে
প্রতীক ব্যবহৃত হয়
σ, π, ρ, ∩ | select, from, where

- Relational algebra:
A relational algebra is a procedural query language in which one or two relations are given as input and produce a relation as result.

SQL → 1980s
RA → 1970

Operations of Relational Algebra

Basic Operation
On Fundamental Operation

- Select (σ) 1
- Project (π) 1
- Union ($\cup$) 2
- Set difference ($-$) 2
- Cartesian Product ($\times$) 2
- Rename (ρ) 1
  - Output table এর নাম দিতে ব্যবহৃত হয়।

Additional Operations
Derived Operations

- Intersection ($\cap$)
- Division ($/$ or $\div$)
- Join ($\bowtie$) *
  - Inner Join
    1. Natural
    2. Theta
    3. Equi
  - Outer Join
    1. Left out.
    2. Right out.
    3. Full out.
- Assignment (দেয়া operation)

Basic Operation

Note:

1. Natural Join made from Cartesian Product
2. Set difference $\Rightarrow$ Intersection $(A-(A-B))$
3. এখানে Operation গুলো সেট rule অনুসারী distinct হবে।
4. Fundamental operation দিয়েও additional operation implement করা যাবে (universal gate) এর মতো।

CamScanner

Basic Operations

1. **Select (σ) :** (SQL এর where এর মতো)

- The select operation selects tuples that satisfy a given predicate.

Notation: $\sigma_p(r)$ p = condition r = relation

Example:

RA	SQL
$\sigma_{marks>60}^{(Result)}$	Select * From Result where marks > 60

Result

	Name	Roll	Marks	Dept
1	Karim	101	67	CSE
2	Rahim	102	95	EEE
3	Kamal	103	84	ICE
4	Jamal	104	56	CSE
5	Rafiq	105	90	EEE

Ans : row 1, 3, 5

NOTE:

SQL	RA
Select	Project
From	( )
Where	Select (σ)

⇒ $\sigma_{Dept = "CSE" \wedge marks > 60}(Result)$

CamScanner

2. Project (π):

- This operation shows the list of those attributes that we wish to appear in the result. Rest of the attributes are eliminated from

=> π_{NAME} (Result) Output : ৫ টা নাম দেখাবে

=> π_{Dept} (Result) ≈ SQL: Select Distinct Dept

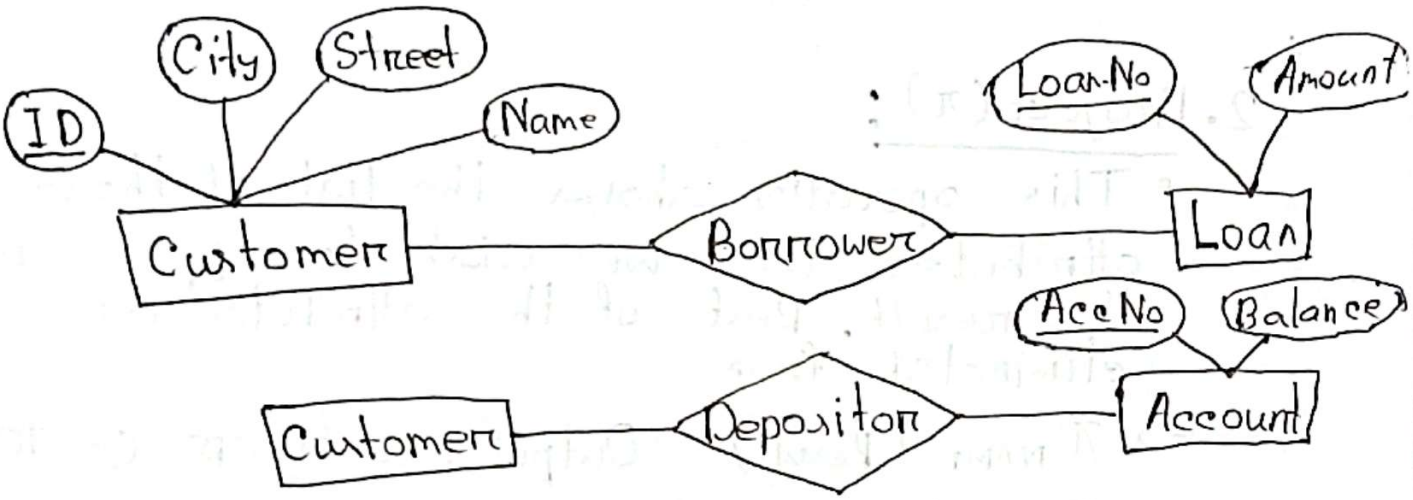
Output: CSE
EEE
ICE

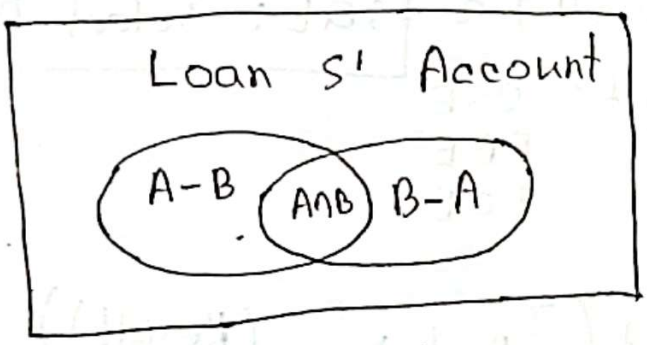
NOTE:

1. $\pi_{Dept}(\pi_{Dept, Name}(Result))$
   $!= \pi_{Dept, Name}(\pi_{Dept}(Result))$
   অর্থাৎ, এটি commutative operator নয়।
2. Vertical subset.

3. Union:

- Suppose there are two tuples R and S. The union operation contains all the tuples that are either in R or S or both in R & S.
- It eliminates the duplicate tuples. It is denoted by $\cup$.





(⇒) Find out the customer name who have a loan or account or both?

π_{Name} (Borrower) ∪ π_{Name} (Depositor)

NOTE:

- Depositor = Customer ◇ Account
- Borrower = Customer ◇ Loan

CamScanner

Cartesian Product or Cross Product

* Let r is a relation with m fields and p records
* Let s is a relation with n fields and q records.

Then,
$r \times s$ = is a relation with $m+n$ fields and $p \times q$ records

Example:

Let;

$r =$

A	B
a_1	b_1
a_2	b_2
a_3	b_3

and, $s =$

A	C	D
a_1	c_1	d_1
a_2	c_2	d_2
a_3	c_3	d_3
a_4	c_4	d_5

Then,
$r \times s =$

A	B	A	C	D
a_1	b_1	a_1	c_1	d_1
a_1	b_1	a_2	c_2	d_2
a_1	b_1	a_3	c_3	d_3
a_1	b_1	a_4	c_4	d_4
a_2	b_2	a_1	c_1	d_1
a_2	b_2	a_2	c_2	d_2
a_2	b_2	a_3	c_3	d_3
a_2	b_2	a_4	c_4	d_4
a_3	b_3	a_1	c_1	d_1
a_3	b_3	a_2	c_2	d_2
a_3	b_3	a_3	c_3	d_3
a_3	b_3	a_4	c_4	d_4

CamScanner

borrower × loan

borrower (customer_name, loan_number)

loan (loan_number, branch_name, amount)

↳ $\pi . A = \sigma . A$

↳ borrower.loan_number = loan.loan_number

Natural Join:

$\pi \bowtie s = \pi \times s$ and a selection on common field

borrower ⋈ Loan = $\pi_{\text{customer.nam}}$ (borrower × Loan)

(σ borrower.lone_numer = loan.loan_number)

```
Select customer.name
From borrower Natural Join Loan
where loan.amount >7000;
```

অথবা ;

```
Select customer.name
From borrower, loan
where loan.amount >7000 and
borrower.loan_number=loan.loan_number
```

Division Operation:

* $R_1 \div R_2$ or R_1 / R_2

Condition:

$R_1 \div R_2$ is possible when $R_2 \subset R_1$ and $R_1 \neq R_2$

* When use?

Ans. When in a query all or every word is avaiable.

Enroll

Sid	Cid
S_1	C_1
S_2	C_1
S_1	C_2
S_3	C_2

Course

Cid
C_1
C_2

Query: Find the students who have enrolled all course.

Ans: S_1

. Enroll ÷ course = S_1

Procedure:

$\pi_{sid}(Enroll) \times \pi_{cid}(course)$

S_1
S_2
S_3

×

C_1
C_2

=

~~S_1~~	~~C_1~~
~~S_1~~	~~C_2~~
~~S_2~~	~~C_1~~
S_2	C_2
S_1	C_1
S_3	C_2

CamScanner

Now :

Let : $A = \pi_{sid}\left(\left(\pi_{sid}(Enroll) \times \pi_{cid}(course) - \pi_{sid,cid}(Enroll)\right)\right)$

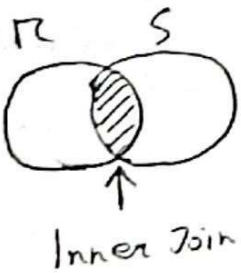
$= \begin{array}{|c|}\hline s_2 \\ \hline s_3 \\ \hline \end{array}$

এবং, $\pi_{sid}(Enroll) - A = \begin{array}{|c|}\hline s_1 \\ \hline s_2 \\ \hline s_3 \\ \hline \end{array} - \begin{array}{|c|}\hline s_2 \\ \hline s_3 \\ \hline \end{array} = s_1$

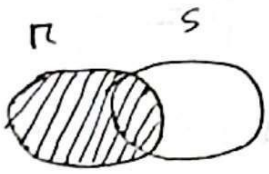
• Inner Join

NOTE: • Join বলতে inner join কেই বুঝায়, outer join, inner join এর extension
• X → cross join, cross product, cartesian product
• ⋈ → Natural Join (Inner join)
Theta → conditional join
• θ join এর special case হলো natural join and equi join.

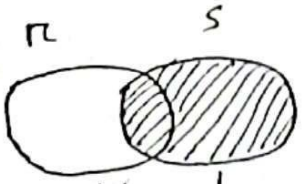
Def:
• An inner join is an association between two relations depending on some conditions. Here, the records will be displayed only when the condition is true.
• θ join is called conditional join. Natural join and Equi join are special case of θ join.



Inner Join



Left outer Join



Right outer join

CamScanner

- θ = condition which will use $<, >, =, \geq, \leq$
- Join = Cartesian Product + Condition ↑ Equi Join

$$\pi \bowtie \pi = \Pi_{R \cup S}\left(\sigma_{\pi.A_1 = S.A_1 \wedge \pi.A_2 = S.A_2 \wedge \ldots \pi A_n = S A_n}^{(\pi \times S)}\right)$$

where, $R \cap S = A_1, A_2, A_3 \ldots A_n$
এখানে, Field নামও এক হতে হবে।

S (roll, name, Address)
π (roll, name, Mobile)

Example:

Student

Roll	Name	Reg
101	Karim	200
102	Rahim	300
109	Kamal	500

Result

Roll	Marks
101	600
103	100

Q. যে সকল student এর মার্কস 500 এর বেশি তাদের নাম।

RA. $\text{Student} \bowtie_\theta \text{Result} = \text{student} \underset{\text{condition}}{\bowtie} \text{result}$

$= \Pi_{\text{Name}}\left(\sigma_{\text{student.Roll} = \text{Result.Roll} \wedge \text{Result.marks} > 500}^{(\text{student} \times \text{Result})}\right)$

SQL:

```
select Name From student natural join result
on student.Roll = Result.Roll where
Result.marks > 500;
```

CamScanner

Natural join => Common Fields will be written only one time

θ join and Equi join => common fields will be written as many as it appear. (m+n)

Loan

Loan. No	Branch Name	Amount
L-101	Dhaka	9000
L-102	Rajshahi	7000
L-103	Khulna	8000

Borrower

Loan. No	Customer Name
L-101	Karim
L-102	Rahim
L-104	Kamal

Loan $\bowtie$ Borrower :

Loan. No	Branch Name	Amount	Customer
L-101	Dhaka	9000	Karim
L-102	Rajshahi	7000	Rahim
L-103	Khulna	8000	Null

Loan $\bowtie$ Borrower:

Loan. No	Branch Name	Amount	Customer Name
L-101	Dhaka	9000	Karim
L-102	Rajshahi	7000	Rahim
L-104	Null	Null	Kamal

CamScanner

Full Outer Join : ⟗

Loan ⟗ Borrower = Loan ⋈ Borrower
∪ Loan ⟕ Borrower

Loan_No	Branch_Name	Amount	Name
L-101	Dhaka	9000	Karim
L-102	Rajshahi	7000	Rahim
L-103	Khulna	8000	Null
L-104	Null	Null	Kamal

VVI

Constraints: Primary key, Foreign key, Unique, Not null, default, check

Aggerigate Function: Max, min, sum, overage

order by, group by

1. DCL (Data Control Language):
   - Manage user access and permission in the database.

   GRANT : Give specific permission.
   GRANT SELECT, INSERT ON Employee TO User1;

   REVOKE: Removes specific permission from user or roles.
   REVOKE Select on Employee FROM User1;

2. Data Definition Language (DDL):

- Defines and modifies the structure of database objects such as tables, indexes, views, and schemas.

CREATE, ALTER, DROP, TRUNCATE

3. DML (Data Manipulation Language):

INSERT, UPDATE, DELETE, SELECT

4. TCL (Transaction Control Language):

Manages database transactions, ensuring consistency and atomicity.

COMMIT; ROLLBACK, SAVEPOINT, SET TRANSACTION

DBMS → Section B

Functional Dependancy

নাম হওয়া উচিত Attribute dependency.

The functional dependency is a relationship that exists between two attributes. It typically exists between the primary key and non-key attribute within a table.

$$X \longrightarrow Y$$

x determines y

where x is called determinant attribute and y is called dependent attribute.

বাম সাইডের যেগুলো ডান সাথের সাথে রয়েছে সেগুলো অভবাবেই আসুক একই আসতে হবে

X	Y
1	2
2	4
3	6
4	7
1	2

Types of Functional Dependencies in DBMS :

1. Trival FD
2. Non-Trival FD
3. Multivalued FD
4. Transitive FD

1. **Trival FD:** In Trival FD, a dependent is always a subset of the determinant. If $X \rightarrow Y$ and Y is the subset of X.

If $X \rightarrow Y$ then $Y \subseteq X$.

2. **Non-trivial FD**: In Non-trivial FD, the dependent is strictly not a subset of the determinant. if $X \to Y$ then $Y \not\subseteq X$.

3. **Multivalued FD**: In Multivalued FD, entities of the dependent set are not dependent on each other. i.e. if $a \to \{b, c\}$ and there exists no functional dependency between b and c, then it is called a multivalued functional dependency.

4. **Transitive FD**: In transitive FD, dependent is indirectly dependent on determinant. i.e. if $a \to b$ & $b \to c$, then according to axiom of transitivity, $a \to c$. This is a transitive FD.

Example:

roll	Name	age
92	abc	17
93	pqr	18
94	xyz	18

(Roll, Name) → Name (Trival)
(Roll, Name) → Age (Non-trival)
(Roll) → (Name, Age) Multi-valued
(Roll, Name) → (Name, Age) semi-trival

CamScanner

Armstrong's Axioms:

The term Armstrong Axioms refer to the sound and complete set of inference rules or axioms, introduced by William W. Armstrong, that is used to test the logical implementation of FD.

Primary Rules:

1. Reflexivity Rule: if $Y \subseteq X$ then $X \to Y$

~~if $XB \subseteq A$ then $XA \to B$~~

2. Transitivity Rule:
   if $X \to Y$ and $Y \to Z$ then $X \to Z$
3. Augmentation Rule:
   if $X \to Y$ then $XA \to YA$

- Determinant part কমানো যাবে না, dependent part বেড়ে যায়।

Secondary Rules:

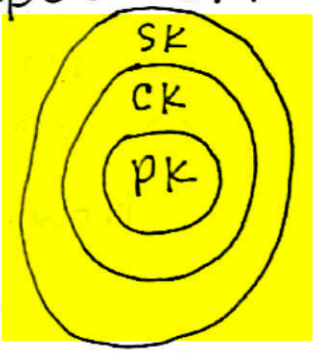
1. Union Rule:
   if $X \to Y$ and $X \to Z$ then
   $X \to YZ$
2. Decomposition / Splitting Rule:
   if $X \to YZ$ then $X \to Y$ and $X \to Z$
3. Pseudo Transitivity Rule:
   if $X \to Y$ and $YZ \to A$ then $XZ \to A$
4. Composition Rule:
   if $X \to Y$ and $A \to B$ then $XA \to YB$

CamScanner

Attribute Closure / Closure Set of Attributes

① If F is a set of functional dependencies then the closure of F, denoted as F^+, is the set of all functional dependencies logically implied by F.

Q. Find attribute closure, super key, candidate key?



=> R (A, B, C, D)

FD = { A → B, B → C, C → D }

A → A (Reflixivity)

→ AB (A থেকে B তে যাওয়া যায়)

→ ABC (B থেকে C " " ")

→ ABCD (C থেকে D " " ")

$A^+ = \{A, B, C, D\}$

$B^+ = \{B, C, D\}$

$C^+ = \{C, D\}$

$D^+ = \{D\}$

Note:

X^+ = combines set of attributes determined by X

ex: $A^+ = \{A, B, C, D\}$

$B^+ = \{B, C, D\}$

$C^+ = \{C, D\}$

$D^+ = \{D\}$ ∴ Super key = A

NOTE:

Super key : Set of attributes whose closure contain all the attributes of the given relation.

Candidate key : Proper subset of a super key is not a super key. Minimal super key is candidate key.

Now, consider,

$AB^+ = \{A, B, C, D\}$

AB হল super key but not candidate key

NOTE:

A যেহেতু super key, AB, ABC অর্থাৎ A এর সাথে যত কিছু Augment করা হোক না কেন সবাই super key হবে।

SK: A, AB, AC, AD, ABC, ABD, ACD, ABCD

□ Problem-2:

R(A, B, C, D)

∴ FD = {A → B, B → C, C → D, D → A}

Soln:

$A^+ = \{A, B, C, D\}$

$B^+ = \{B, C, D, A\}$

$C^+ = \{C, D, A, B\}$

$D^+ = \{D, A, B, C\}$

CamScanner

∴ Super key is A, B, C, D as well as candidate key.

Prob 4: How many candidate key exist?

$R(A, B, C, D, E)$

$FD = \{A \rightarrow B,\ D \rightarrow E\}$

Soln: Using shortcut trix:

$ABCDE^+ = \{A, B, C, D, E\}$

$ACDE^+ = \{A, C, D, E, B\}$

$ACD^+ = \{A, C, D, B, E\}$

Here ACD^+ is super key

Proper subset of ACD^+ are:

A, C, D, AC, AD, CD

Here,

$A^+ = \{A, B\}$

$C^+ = \{C\}$

$D^+ = \{D, E\}$

$AC^+ = \{A, C, B\}$

$AD^+ = \{A, D, B, E\}$

$CD^+ = \{C, D, E\}$

} Not super key

∴ ACD is super key as well as candidate key.

∴ Prime Attributes:

$\{A, C, D\}$

এখানে, A, C, D কে FD এর → x বামে না থাকলে } স্লাস করে দিতে।

Non Prime Attributes $\{B, E\}$
Here, only one (ACD) candidate key exist.

Problem 5: $R(A, B, C, D)$

$F.D = \{A \to B, B \to C, C \to A\}$

Soln:

$AB^+CD = \{A, B, C, D\}$

$A^+D = \{A, C, D, B\}$

$AD^+ = \{A, D, B, C\}$

Here AD^+ is SK.

$A^+ = \{A, B, C\}$ So, AD is CK

$D^+ = \{D\}$

Again,

AD
↓
CD

এখানে যেসপাশে আছে FD তে একসাথে replace করব

Now test CD is C_K or not

$C^+ = \{C, A, B\}$

$D^+ = \{D\}$

∴ CD is C_K

Again,

CD
↓
BD

Now test BD is C_K or not

$B^+ = \{B, C, A\}$

$D^+ = \{D\}$

∴ BD is also C_K.

So, C_K is $\{AD, CD, BD\}$

Prime attributes: $\{A, D, C, B\}$

Non-Prime attributes $\{\phi\}$

Problem 6: $R(A, B, C, D, E, F)$;

$F.D = \{AB \rightarrow C, C \rightarrow DE, E \rightarrow F, D \rightarrow A, C \rightarrow B\}$

CamScanner

Normalization

Normalization is used to reduce or minimize the data redundancy in table or relation.

Why normalization is needed?

→ Insertion Anomalies
→ Update "
→ Deletion "

Student

Sid	S Name	Credit	Dept	Building	Room No
1	Karim	160	CSE	B1	101
2	Rahim	140	EEE	B2	102
3	Kamal	150	MSE	B1	104
4	Jamal	160	CSE	B1	101
5	Rafiq	140	EEE	B2	102
6	Shafiq	160	CSE	B1	101
7	Salam	140	EEE	B2	102

CamScanner

- **Insertion Anomaly**: Insertion Anomaly refers to when one cannot insert a new tuple into a relationship due to lack of data.

If we want to add new dept to the following table this will happen:

Null	Null	Null	UP	B3	103

↑ because of primary key insert anomaly Problem

- **Updatation Anomaly**: The update anomaly is when an update of a single data value requires multiple rows of data to be updated.

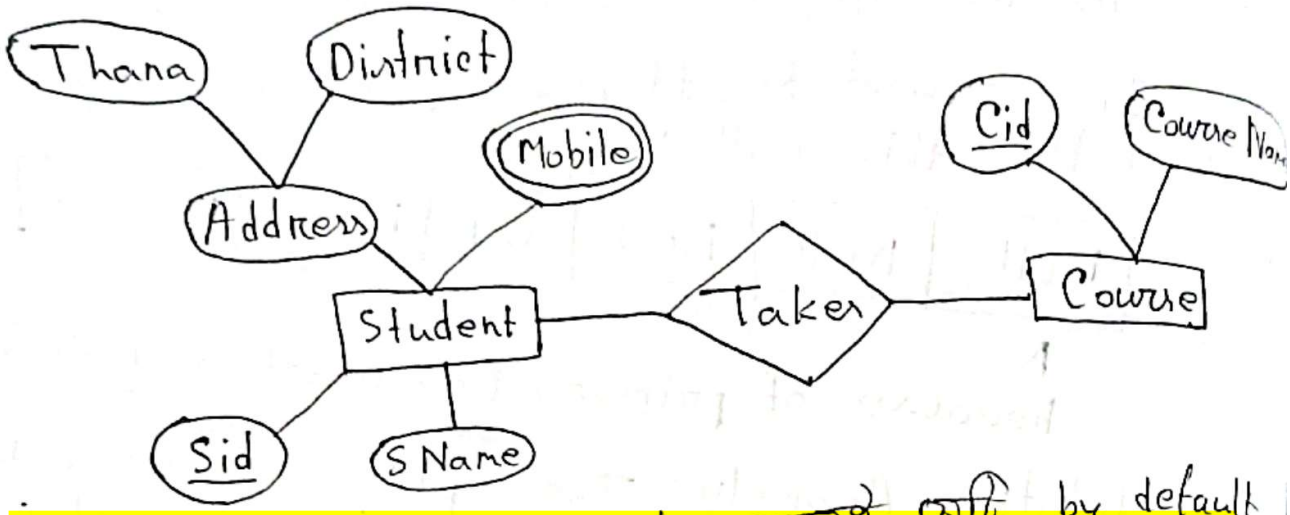
1	Karim	160	CSE	BT B3	HT 109

এই ভাবে সবগুলো tupple এ change করতে হবে।

- **Deletion Anomaly**: The deletion anomaly refers to the situation where the deletion of data results in the unintended loss of some other important data.

If we delete row-3, then we will also lost the information of "MSE" department.

First Normal Form (1NF)



NOTE : ER→Table convert করলেই সেটি by default 1 N.F তৈরি হয়।

Sid	SName	Addres	Mobile
1	Karim	Boalia, Raj	M1
2	Rahim	Badda, Dhaka	M2, M3
3	Kamal	Lalpur, Nat	M4

Not atomic

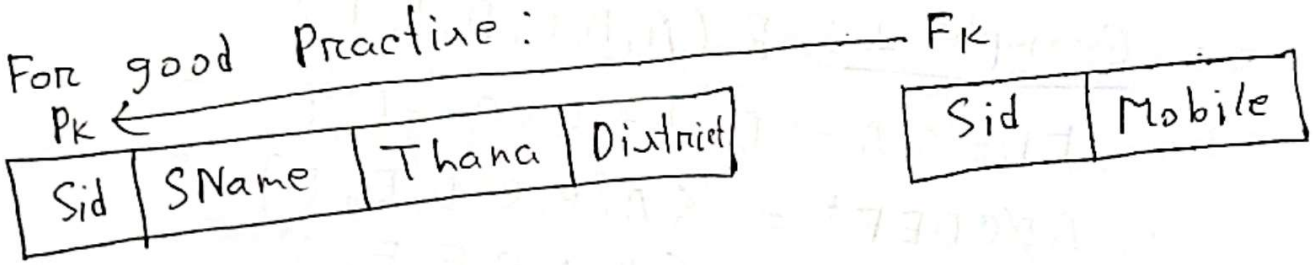
Condition to be 1NF

1. Data will be in atomic form. (Single valued)
2. Domain or Data Type will be maintained.
3. Column name will be unique.
4. No ordering of fields and records.
5. No two tuples will be same or identical.

1NF Table:

Sid	SName	Thana	District	Mobile
1	Karim	Boalia	Raj	M1
2	Rahim	Badda	Dhaka	M2
2	Rahim	Badda	Dhaka	M3
3	Kamal	Lalpan	Natore	M4

For good Practise:



Second Normal Form (2NF)

Rules:

1. Must be in 1NF.
2. Proper subset of $C_K \rightarrow$ Non-Prime Attributes(NPA)
   No partial dependency.

   Partial dependency থাকলে 2NF হবে না,
   না থাকলে হবে।

Example 2: $R(A, B, C, D, E, F)$

$FD = \{A \rightarrow B,\ B \rightarrow C,\ C \rightarrow D,\ D \rightarrow E\}$

$\therefore ABCDEF^+ = \{A, B, C, D, E, F\}$

$ACDEF^+ = \{A, C, D, E, F, B\}$

$ADEF^+ = \{A, D, E, F, B, C\}$

$AEF^+ = \{A, E, F, B, C, D\}$

$SK \rightarrow AF^+ = \{A, F, B, C, D, E\}$

Proper subset of AF^+

$A^+ = \{A, B, C, D, E\}$ $\quad C_K = \{AF\}$

$F^+ = \{F\}$

$\therefore$ Prime attributes $PA = \{A, F\}$

Non-Prime " $NPA = \{B, C, D, E\}$

Here C_K has partial dependency. $(A \rightarrow B)$
So it is not 2NF.

CamScanner

Note: A partial dependency occurs when a non-prime attribute depends on a proper subset of a composite candidate key.

If, ABC^+ in C_K then proper subset of C_K : A, B, C, AB, BC, CA

If, AB^+ " C_K " " " of C_K : A, B

If, A^+ " C_K " " " of C_K : $\{\phi\}$

Example 3:

$$R(A, B, C, D)$$

$$FD = \{A \rightarrow B, B \rightarrow C, C \rightarrow D\}$$

Solve:

$$ABCD^+ = \{A, B, C, D\}$$

$$ACD^+ = \{A, C, D, B\}$$

$$AB^+ = \{A, B, B, C\}$$

$$SK \rightarrow A^+ = \{A, B, C, D\}$$

Proper subset of A^+ in $\{\phi\}$

$\therefore C_K = A$, $PA = \{\phi\}$

$NPA = \{B, C, D\}$

For all FD there is no partial dependency. So it is 2NF.

Problem 1: Determine 2NF or not.

2NF

$R(A, B, C, D)$

$FD = \{AB \rightarrow CD, C \rightarrow A, D \rightarrow B\}$

3NF - 3rd Normalization Form

Rule:

1. The relation must be in 2NF.
2. No transitive dependency for NPA.

> NPA → NPA
> This is called Transitive dependency

* A relation is in 3NF if and only if the relation holds the any one of the following condition:
  - i) L.H.S is a Super key (SK)
    or.
  - ii) R.H.S is a Prime Attributes. (PA)

Problem 1:

$R(A, B, C, D)$

$FD = \{A \to B, B \to C, C \to D\}$

Soln:

$ABCD^+ = \{A, B, C, D\}$

$AC D^+ = \{A, C, D, B\}$

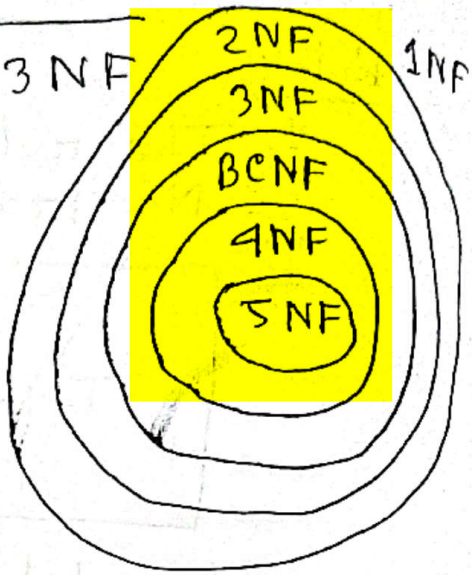
$AD^+ = \{A, D, B, C\}$

$A^+ = \{A, B, C, D\}$

$PA = \{A\}$ $CK = \{A\}$ $NPA = \{B, C, D\}$

∴ It is not 3NF because $D \to C$ which is a transative dependency.

Boyce-Code NF (BCNF) 3.5 NF



* BCNF is stronger than 3NF

Rules:

1. Must be in 3NF.
2. LHS is a super key.

Problem-1:

$R(A, B, C)$

$FD = \{A \to B, B \to C, C \to A\}$

Soln: $ABC^+ = \{A, B, C\}$

$AC^+ = \{A, B, C\}$

$SK \to A^+ = \{A, B, C\}$

$C_K = \{A, B, C\}$

$S_K = \{A, B, C\}$

$PA = \{A, C, B\}$

∴ it in BCNF.

Problem 2: Find out the highest nF:

$R(A, B, C, D, E)$

$F.D = \{A \to BCDE, BC \to ACE, D \to E\}$

Soln:

~~$ABCDE^+$~~ $= \{A, B, C, D, E\}$

$SK \to A^+ = \{A, B, C, D, E\}$

$\downarrow$

$BC^+ = \{A, B, C, D, E\}$

CamScanner

Here, $B^+ = \{B\}$

$C^+ = \{C\}$

$\therefore S_K = \{A, BC\}$, $C_K = \{A, BC\}$, $PA = \{A, B, C\}$

$NPA = \{D, E\}$

	$A \to BCDE$	$BC \to ACE$	$D \to E$	Rule
BCNF	✓	✓	✗	$S_K \to$ _
3NF	✓	✓	✗	! NPA → NPA
2NF	✓	✓	✗	No partial Dependency

So by default it is 1NF.

Problem 3:

$R(A, B, C, D, E)$

$FD = \{AB \to CDE, D \to A\}$

Soln: $ABCDE^+ = \{ABCDE\}$

$AB^+ = \{A, B, C, D, E\}$

$AB^+ \to S_K$

$C_K = \{AB\}$

$PA = \{A, B\}$

$NPA = \{C, D, E\}$

	FD 1	FD 2
BCNF	✓	✗
3NF	✓	✓

$\therefore$ It is 3NF.

CamScanner

4NF (4th Normalization Form)

Rules:

1. Relation must be in 3NF.
2. No multivalued dependancy

Condition for multivalued dependancy:

1. There must be atleast three attributes in relation R.
2. The depandancy will follow the condition.

   $A \twoheadrightarrow B$ and $A \twoheadrightarrow C$ and ther is no dependency between B and C. Here A, B and C are attributes.

A	B	C
Sid	Cource	Hobby
1	C DBMS	Football Cricket
2	C++	Reading
3	Java Network	Swimming

Sid	Cource
1	C
1	DBMS
2	C++
3	Java
3	Network

CamScanner

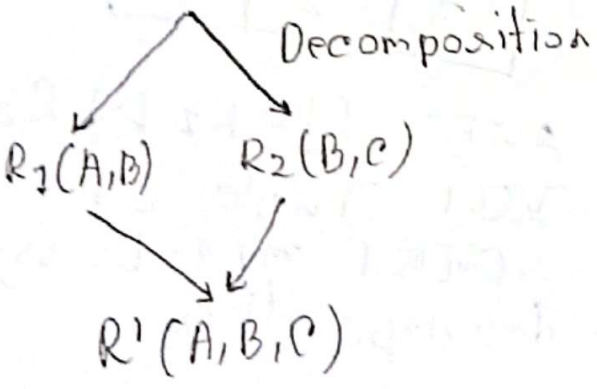
5NF (5th Normalization Form)

Rules:
1. Must be in 4NF
2. No Joint Dependancy

Decomposition: Decomposition refers to the division of tables into multiple tables to produce consistency in the data. We performed decomposition in DBMS when we want to process a particular data set. It is performed in a DBMS when we need to ensure consistency and remove anomalies and duplicate data present in the database.

Decomposition:
- Dependancy Preserving Decomposition.
- Lossy / Lossless Joint Decomposition

Lossy / Lossless Joint Decomposition



if $R = R'$ Then
it is lossless

Otherwise,
it is lossy decomposition

Ck, common field হিসেবে থাকলে lossless অন্যথায় lossy.

R

A	B	C
1	1	1
2	2	3
3	2	3

R_1

A	B
1	1
2	2
3	2

$\bowtie$

R_2

B	C
1	1
2	3
2	3

Query: Findout the value of C for which value of A is 1.

```
Select R2.C From
R2 Natural Join R1
where R1.A = 1;
```

```
Select R2.C From R2, R1
where R1.A=1 and
R1.B = R2.B.
```

$R_1 \bowtie R_2$

A	B	B	C
1	1	1	1
1	1	2	3
1	1	2	3
1	2	1	1
2	2	2	3
2	2	2	3
3	2	1	1
3	2	2	3
3	2	2	3

$R_1 \bowtie R_2$

A	B	C
1	1	1
2	2	3
2	2	3
3	2	3
3	2	3

← বেশি হয়ে

এখানে, $R1 = R_1 \bowtie R_2$ | মূলে Tuple বেশি চলে এসেছে। তাই Lossy decomposition.

Rules for Decomposition:

1. $R_1 \cup R_2 = R$
2. $R_1 \cap R_2 \neq \{\phi\}$
3. $R_1 \cap R_2 = R_1$ or $R_1 \cap R_2 = R_2$ → For lossless joint decomposition depending on C_F

এই নোটের মধ্যে Trigger Add করতে হবে