Database Management Systems

Date No

What is a Database?

Collection of related data.

Database Management System

DBMS is a software application to manage databases.

Ex:

My SQL
PostgreSQL
Oracal
Microsoft SQL Server.

* Store and manage data.
* Data is generated manually and automatically.
* Data is used (consumed) by end-users and different applications.

Example for Data.

Text, Audio, Numbers, Files, Video, Amounts, Images, Dates

Database System.

Database and database management system together is called a database system.

ProMate

Scanned with CamScanner

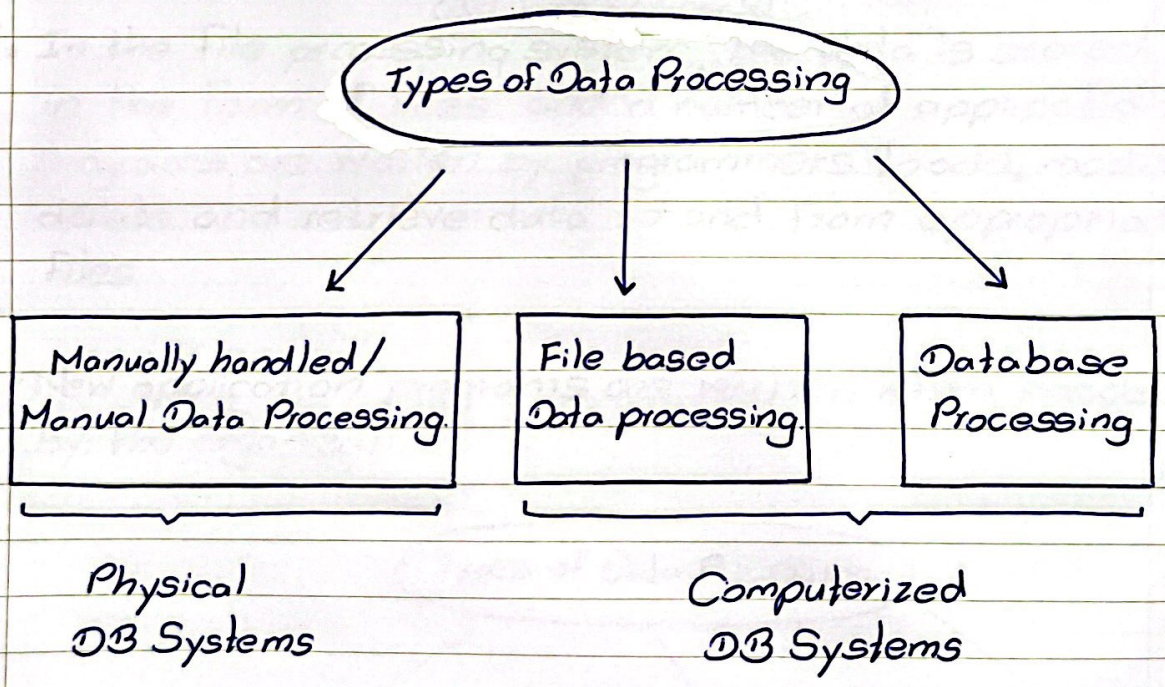
Date No

Advantages of Using DBMS

Size of data
Ease of updating
Accuracy
Scalability
Multiple Access
Validation.

Security
Redundancy
Incomplete
Access/Privacy
Multiple copies
Integrity.

Types of Data Processing.



ProMate

Scanned with CamScanner

Date No

Disadvantages of File Processing.

01.) Limited data Mapping and Access.
02.) Data Redundancy.
03.) Data Dependence.
04.) Data Inconsistency.
05.) Data Isolation.
06.) Security is less.
07.) Intergrity is limited.
08.) Concurrent access is limited.

Scanned with CamScanner

Date No

Examples of Database Applications.

01. Purchase from the market.
02. Purchases using your credit card.
03. Booking a holiday at the travel agents.
04. Using the local library.
05. Taking out insurance.
06. Using the internet.
07. Studying at university.

What is Data?

* Known facts that can be recorded and have an implict meaning; raw.

What is Database?

* A highly organized, interrelated and structured set of data about a particular enterprise controlled by a database management system (DBMS)

What is DBMS?

* Set of programs to access the data.
* An enviroment that is both convenient and efficient to use.

* Database systems are used to manage collections of data that are:
  - Highly valuable.
  - Relatively large.
  - Accessed by multiple users and applications, often at the same time.

ProMate

Scanned with CamScanner

Date No

* A modern database system is a complex software system whose task is to manage a large, complex collection of data.

* Databases touch all aspects of our lives.

Types of Databases and Database Applications.

* Traditional Applications.

01. Numeric and textual databases.

* More recent Applications.

01. Multimedia databases.
02. Geographic Information Systems. (GIS)
03. Biological and genome databases.
04. Data warehouses.
05. Mobile databases.
06. Real-time and active databases.

* DBMS is a software or it's an engine that enables users/ applications to access, manipulate and report on data stored in databases.

When not to use DBMS.

01. Main inhibitors (costs) of using a DBMS:
    * High initial investment and possible need for additional hardware.
    * Overhead for providing generality, security, concurrency control, recovery and intergrity functions.

ProMate

Scanned with CamScanner

Date No

02. When a DBMS may be unnecessary:
* If the database and applications are simple, well defined and not expected to change.
* If access to data by multiple users is not required.

03. When a DBMS may be infeasible:
* In embedded systems where a general-purpose DBMS may not fit in available storage.

Recent Developments.

01.) Social Networks. started capturing a lot of information about people and about communications among people-posts, tweets, photos, videos in system such as;
- Facebook.
- Twitter.
- Linked-In

All of the above constitues data.

02.) Search Engines, Google, Bing, Yahoo; collect their own respository of web pages for searching purposes.

03.) New technologies are emerging from the so-called non SQL, non-database software vendors to manage vast amounts of data generated on the web:
- Big Data storage systems involving large clusters of distributed computers.
- NOSQL (Non-SQL, Not Only SQL) systems.

04.) A large amount of data now resides on the cloud which means it is in huge data centers using thousands of machines.

ProMate

Scanned with CamScanner

Date No

Database Concepts.

* The need of database system arose in response to the traditional file processing system.

* In the database approach (file processing system), the data is stored in the form of files a number of application programs are written by programmers to add, modify, delete and retrieve data to and from appropriate files.

* Traditional database applications vs Big data storage systems or NOSQL systems provide cloud storage.

* New application programs are written when needed by the organization.

Things to be remembered...

A database...

- Is a miniworld.
- Is a logically coherent collection of data with some inherent meaning.
- Is designed, built and populated with data for a specific purpose.

Collection of related data.

Scanned with CamScanner

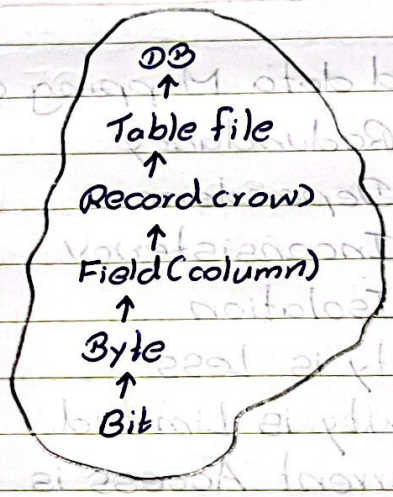
Date No

Characteristics of the Database Approach.

* Self-describing nature of a database system.
* Insulation between programs and data, and data abstraction.
* Support of multiple views of data.
* Sharing of data and multiuser transaction processing.

Data Hierarchy.

01. Database.
02. Files.
03. Records.
04. Fields.
05. Characters (Bytes)
06. Bits.



Data Modeling and Data Models.

Model

Abstraction of a real-world object or event.

Data Models.

Simple representations of complex real-world data structures.

- Useful for supporting a specific problem domain.

ProMate

Scanned with CamScanner

Date No

Data Modeling

Iterative and progressive process of creating a specific data model for a determined problem domain.

Importance of Data Models.

01. Are a communication tool.
02. Give an overoll view of the database.
03. Organize data for various users.
04. Are an abstraction for the creation of good database.

Categories of Data Models.

* High-level or conceptual data models:

Provide concepts that are close to the way many users perceive data

Ex: Entity-Relationship Model.

* Low-level or physical data models:

Provide concepts that describe the details of how data is stored on the computer storage media, typically magnetic disks.

* Representational (or implementation) data models:

Provide concepts that may be easily understood by end users but that are not too far removed from the way data is organized in computer storage.

ProMate

Scanned with CamScanner

Ex: Relational Model
Hierarchical Model
Network Model
OO Model

Types of Database Models.

01. Traditional Files - 60s
02. Hierarchical Database Model - 70s
03. Network Database Model - 70s
04. Relational Database Model - 80s
05. Object-oriented Database Model - 90s
06. Object-relational Database Model - 90s

Hierarchical and Network	* Difficult to represent M:M relationships. (hierarchical only) * Structural level dependency. * No ad hoc queries. (record-at-a-time access) * Access path predefined. (navigational access)
Relational	* Conceptual simplicity. (structual independence) * Provides ad hoc queries. (SQL) * Set-oriented access.
Entity Relationship	* Easy to understand. (more semantics) * Limited to conceptual modeling. (no implementation component)

ProMate

Scanned with CamScanner

Semantic → • Object Oriented → • Extended Relational (O/R DBMS)	* More semantics in data model. * Support for complex objects. * Inheritance (class hierarchy) * Behaviour. * Unstructured data (XML) * XML data exchanges.
NOSQL (Big Data)	* Addresses big data problems. * Less semantics in data model. * Based on schema-less key-value data model * Best suited for large sparse data stores.

Flat Model

* The flat model consists of a single, two-dimensional array of data elements, where all members of a given column are assumed to be similar values, and all members of a row are assumed to be related to one another.

* Columns of the table often have a type associated with them, defining them as character data, date or time information, integers or floating point numbers

Data Redundancy.

* Unnecessarily storing same data at different places.

* Islands of information: Scattered data locations.
  - Increases the probability of having different versions of the same data.

Scanned with CamScanner

Date No

* Implications.
  - Poor data security.
  - Data inconsistency.
  - Increased likelihood of data-entry errors when complex entries are made in different files.
  - Data anomaly : Develops when not all of the required changes in the redundant data are made successfully.

Types of Data Anomaly.

01. Update Anomalies.
02. Insertion Anomalies.
03. Deletion Anomalies.

Hierarchial Model

* Data is organized into a tree-like structure, implying a single upward link in each record to describe the nesting, and a sort field to keep the records in a particular order in each same-level list.

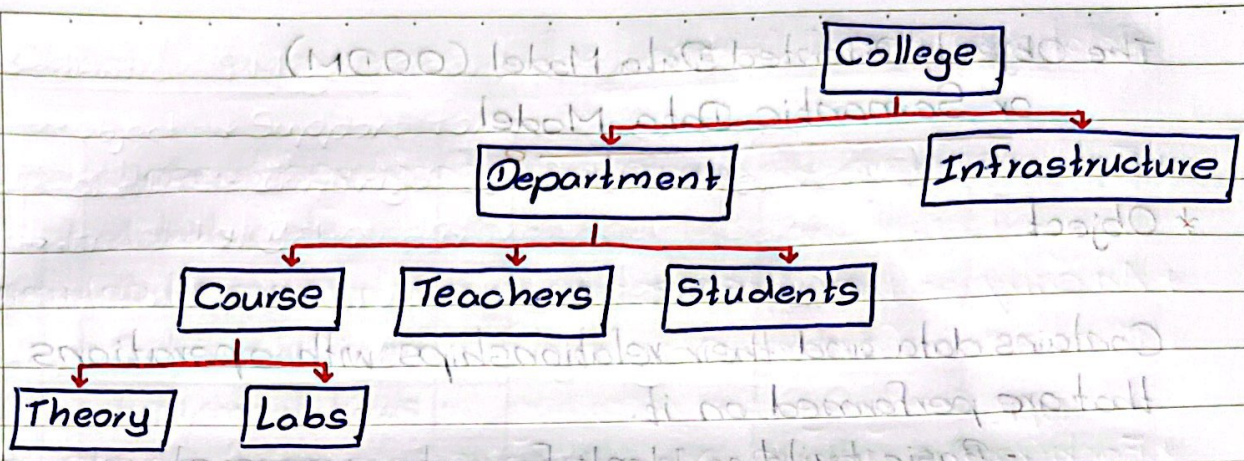
* Hierarchial structures were widely used in the early mainframe database mangement systems.

* This structure allows one 1:M relationship between two types of data.

* This structure is very efficient to describe many relationships in the real world.

ProMate

Scanned with CamScanner



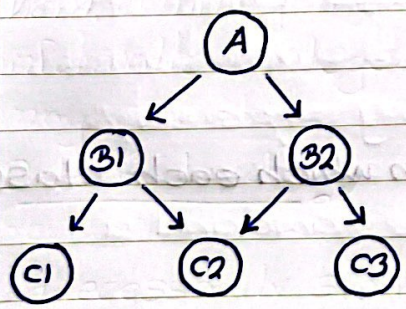
Network Data Model

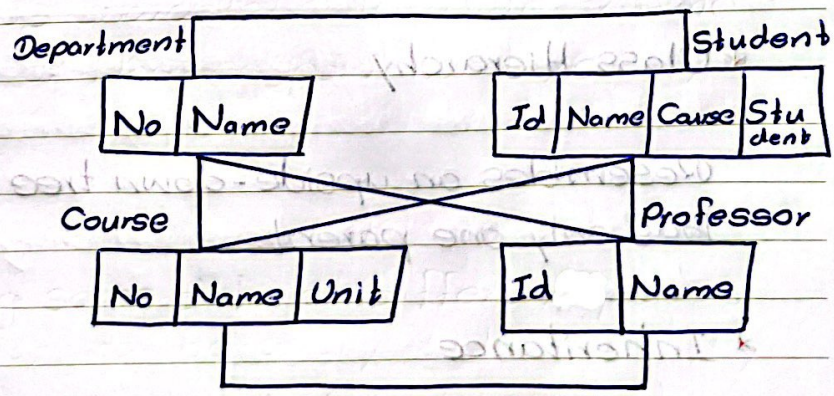
* An extension of the Hierarchial model. Data is organized more like a graph, and are allowed to have more than one parent mode.

* Improve database performance and impose a database standard.

* Depicts both 1:M and M:M relationships.

* This was the most widely used database model, before Relational Model was introduced.





Object-oriented programming + Relational Data Model → Object-oriented Data Model.

Date No

The Object-Oriented Data Model (OODM) or Semantic Data Model

* **Object**

Contains data and their relationships with operations that are performed on it.

- Basic building block for autonomous structures.
- Abstraction of real-world entity.

* **Attributes**

Describe the properties of an object.

* **Methods**

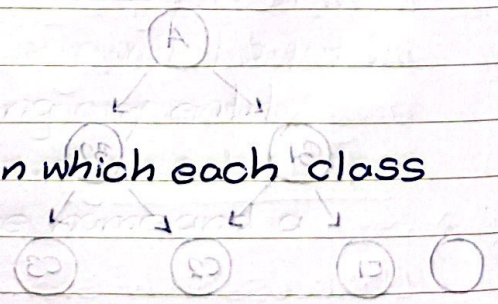
Describe the behavior of an object.

* **Class**

Collection of similar objects with shared structure and behavior organized in a class hierarchy.

* **Class Hierarchy.**

Resembles an upside-down tree in which each class has only one parent.



* **Inheritance**

Object inherits methods and attributes of parent class.

ProMate

Scanned with CamScanner

Date No

Object 1: Maintenance Report

Date	
Activity code	
Route no.	
Daily production	
Equipment hours	
Labor hours.	

Object 1 : Instance

01-12-01
24
I-95
2.5
6.0
6.0

Object 2 : Maintenance Activity

Activity code	
Activity name	
Production unit	
Average daily production rate	

The Relational Model

* Foundation is a mathematical concept known as a relatio

* Based on a relation.
  - Relation or table : Matrix composed of interestin tuple and attribute
    - tuple : Rows
    - attribute : Columns

* The relationship is maintained by storing a common field

student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss. C
3	C#	Mr. C.Hash
4	Php	Mr. P.H.P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

Scanned with CamScanner

Date No

Relational Model [Properties]

* Each relation (or table) in a database has a unique name

* An entry at the intersection of each row and column is atomic (or single-valued).

* Each tuple (row) is unique; No two rows in a relation are identical.

* Each attribute (or column) within a table has unique name

Relational Objects:

* Table are comprised of rows and a fixed number of named columns.

* Data is presented to the user as tables;

	Column 1	Column 2	Column 3
Row 1			
Row 2			
Row 3			

* Columns are attributes describing any entity.

* Each column must have an unique name and a data type.

ProMate

Scanned with CamScanner

Date No

Structure of a relation (Ex: Employee)
Employee (Name, Designation, Department)

Employee.

Name	Designation	Department

* Rows are records that present information about a particular entity occurrence.

Employee.

Name	Designation	Department
Jason	Software engineer	SE
Shavantha	DA engineer	DA
Roshini	Solution engineer	BS

Relational Model Terminology.

* Row is called a tuple.
* Column header is called an attribute.
* Table is called a relation.
* The data type describing the type of values that can appear in each column is called a domain.

Ex:

- Employee ages: value between 15 & 80 years old.
  The above is called logical definitions of domains.

- A data type or format can also be specified for each domain.
  Ex: The employee age is an integer between 15 and 80

ProMate

Scanned with CamScanner

Date No

Relational Model [Properties]

* The sequence of columns (left to right) is insignificant.

* The columns of a relation can be interchanged without changing the meaning or use of the relation.

* The sequence of rows (top to bottom) is insignificant.

* Rows of a relation may be interchanged or stored in any sequence.

Relational Constraints.

* ① Domain Constraints.

* Specifies that the value of each attribute "A" must be an atomic value.
* And from the specified domain.

* Key Constraints.

* There is a sub set of attributes of a relational schema with the property that no two tuples should have the same combination of values for the attributes.

ProMate

Scanned with CamScanner

Date No

Advantages of Relational Model.

01. Structural independence is promoted using independent tables.
02. Tabular view improves conceptual simplicity.
03. Ad hoc query capability is based on SQL.
04. Isolates the end user from physical-level details.
05. Improves implementation and management simplicity.

Disadvantages of Relational Model.

01. Requires substantial hardware and system software overhead.
02. Conceptual simplicity gives untrained people the tools to use a good system poorly.
03. May promote information problems.

ProMate

Scanned with CamScanner

Date No

Relational Database Management System (RDBMS)

* A collection of programs and capabilities that enable IT teams and others to create, update, administer and otherwise interact with a relational database.

* Most commercial RDBMS use Structured Query Language (SQL) to access the database.

* Makes the relational data model easier to understand and implement.

Standard Database Concepts

* Schema

Conceptual organization of the entire database as viewed by the database administrator.

* Subschema.

Portion of the database seen by the application programs that produce the desired information from the data within the database.

* Data Manipulation language. (DML)

Enviroment in which data can be managed and is used to work with the data in the database.

* Schema data definition language (DDL)

Enables the database administrator to define the schema components.

Scanned with CamScanner

ER, Relational Schema and Normalization

Date No

Steps.

Idea → ER Diagram → Relational Schema → RDBMS

ERD

ERD is a data modeling technology used in software engineering to produce a conceptual data model of a database system.

Data Modeling.

The data modeling revolves around discovering and analyzing organizational and users data requirements.

The requirements for the model development must be based on policies, stakeholders (Ex: survey, discussion, observation), business procedures, system specifications and so on.

* Identify what data is important.
* Identify what data should be maintained.

ProMate

Scanned with CamScanner

Procedure of ERD.

* Data modeling is iterative process.
* "complete" and "100% error free" model is not possible.
* Only "optimized" model is possible.

Steps to create an ERD.

Entity Identification.

↓

Relationship Identification.

↓

Cardinality Identification.

↓

Identify Attributes.

↓

Create ERD.

* The major activity of this phase is identifying entities, attributes and their relationships to construct model using the Entity Relationship Diagram.

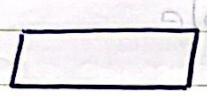
ProMate

Scanned with CamScanner

* Entity → table
* Attribute → column
* Relationship → line.

Symbols

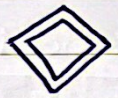
* Chen Notation


→ Strong Entity

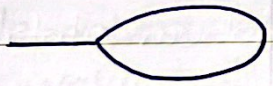
→ Weak Entity

} Entity.

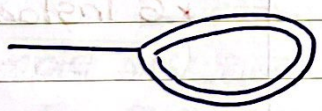
→ Relationship


→ Identifying Relationship

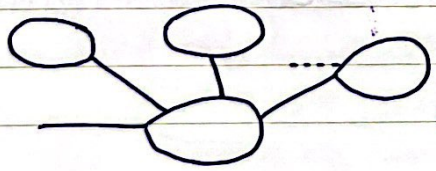
} Relationship


→ Simple Attribute

→ Key Attribute


→ Multi valued Attribute

→ Dervied Attribute.


→ Composite Attribute.

} Attribute

ProMate

Scanned with CamScanner

Date No

How to find entities?

* "anything (people, places, objects, events) about which we store information (Ex: supplier, machine tool, employee, utility pole, air line seat.)"
* Tangible: customer, product
* Intangible: order, accounting receivable.
* Look for singular nouns (beginner)
* BUT a proper noun is not a good candidate.

Entity Instance

* A single occurrence of an entity.

Student ID	Last Name	First Name.
2144	Arnold	Betty
3122	Taylor	John
3843	Simmons	Lisa
9844	Macy	Bill
2837	Leath	Heather
2293	Wrench	Tim.

* 6 instances.

Entity: student

instance

Scanned with CamScanner

Date No

How to find Attributes?

* Attributes are data objects that either identify or describe entities. (property of any entity)
* In other words, it is a descriptor whose values are associated with individual entities of a specific entity type.
* The process for identifying attributes is similar except now you want to look for and extract those names that appear to be descriptive noun phrases.

Classes of Attributes.

01. Simple attribute
02. Composite attribute
03. Derived attribute
04. Single-valued attribute
05. Multi-valued attribute.

(01.) Simple Attribute.

* A simple attribute cannot be subdivided.
  Ex: Age, Gender, Marital status.

(02.) Composite Attribute.

* A composite attribute can be further subdivided to yield additional attributes.
  Ex: • ADDRESS → Street, City, State, Zip
  • PHONE NUMBER → Area code, Exchange number

ProMate

Scanned with CamScanner

Date No

(03) Derived Attribute

* Is not physically stored within the database.
* Insted, it is derived by using an algorithm.

Example 01: Late Charge of 2%
- MS Access: Invoice Amt * 0.02

Example 02: AGE can be derived from the date of birth and the current date.
- MS Access: int(Date() - Emp_DOB)/365)

(04) Single-valued Attribute.

* Can have only a single (atomic) value.
  Ex:
  - A person can have only one social security number.
  - A manufactured part can have only one serial number.

* A single-valued attribute is not necessarily a simple attribute.
  Ex:
  - Part No: CA-08-02-189935
  - Location: CA, Factory #: 08, shift #: 02, part #: 189935

ProMate

Scanned with CamScanner

Date No

(05) Multi-valued Attribute.

* Can have many values.
  Ex: • A person may have several college degrees.
  • A household may have several phones with different numbers.
  • A car colour.

Key Attributes.

* An attribute of an entity type for which each entity must have a unique value is called a key attribute of the entity type.
  Ex: SSN of EMPLOYEE.

* A key attribute may be composite.
  Ex: VehicleTagNumber is a key of the CAR entity type with components (Number, State).

* An entity type may have more than one key.
  Ex: The CAR entity type may have two keys:
  • VehicleIdentificationNumber (popularly called VIN)
  • VehicleTagNumber (Number, State) (also know as license_plate number)

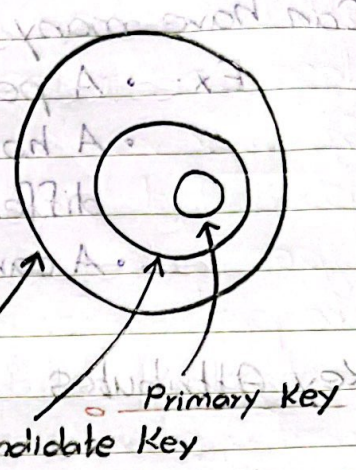
ProMate

Scanned with CamScanner

Date No

Concepts of Keys.

01.) Candidate Key.
02.) Primary Key.
03.) Super Key.
04.) Alternate Key. / Secondary Key.
05.) Composite Key.
06.) Foreign Key.



(01.) Candidate Key.

* A candidate key is a minimal set of attributes necessary to identify a tuple.
  Ex: employeeID

* Each table may have one or more candidate keys. One of these candidate keys is selected as the table primary key.

SID	S Name	Marks
S1	A	40
S2	A	40
S3	B	50
S2	B	50

ProMate

Scanned with CamScanner

Date No

(02) Primary Key.

* One of the candidate keys that has no NULL values.

  * NULL = Unknown.

* An attribute (or combination of attributes) that uniquely identifies each row in a relation.

  - Employee (Emp_No, Emp_Name, Department)

Ex:

Employee

E-No	E-Name	D-No
179	Silva	7
857	Perera	4
342	Dias	7

Primary Key

Salary

E-No	Eff-Date	Amt
179	1/1/98	8000
857	3/7/94	9000
179	1/6/97	7000
342	28/1/97	7500

← Primary Key →

Scanned with CamScanner

Date No

(03) Super Key.

Super Key = Candidate Key + 0 or more attributes

* Every candidate key is a super key.
* Every super key cannot be a candidate key.

* Minimal super key becomes the candidate key.

EID	Salary	E Name.
1	400	A
2	500	B
3	400	A
4	500	C

(04) Alternative Key / Secondary Key.

* An alternative key is any candidate key that is not the primary key.
* Alternative keys are sometimes referred to as secondary keys.

* Null values are allowed.
* More than one alternative keys are allowed.

ProMate

Scanned with CamScanner

SID	Bank ACC	Name	Votee ID
1	Acc1	A	V1
2	Acc2	B	V2
3	Acc3	A	V3
4	Acc4	B	V4
5	Acc5	A	null

(05) Composite Key.

* A primary key that consists of more than one attribute.

-Salary (Emp_No, Eff_Date, Amount)

(Ex from Primary Key)

(06) Foreign Key.

* An attribute in a relation of a database that serves as the primary key of another relation in the same database.

ProMate

Scanned with CamScanner

E-No	EmpName	Dep-No
12	Sarath	1
34	Kumara	2

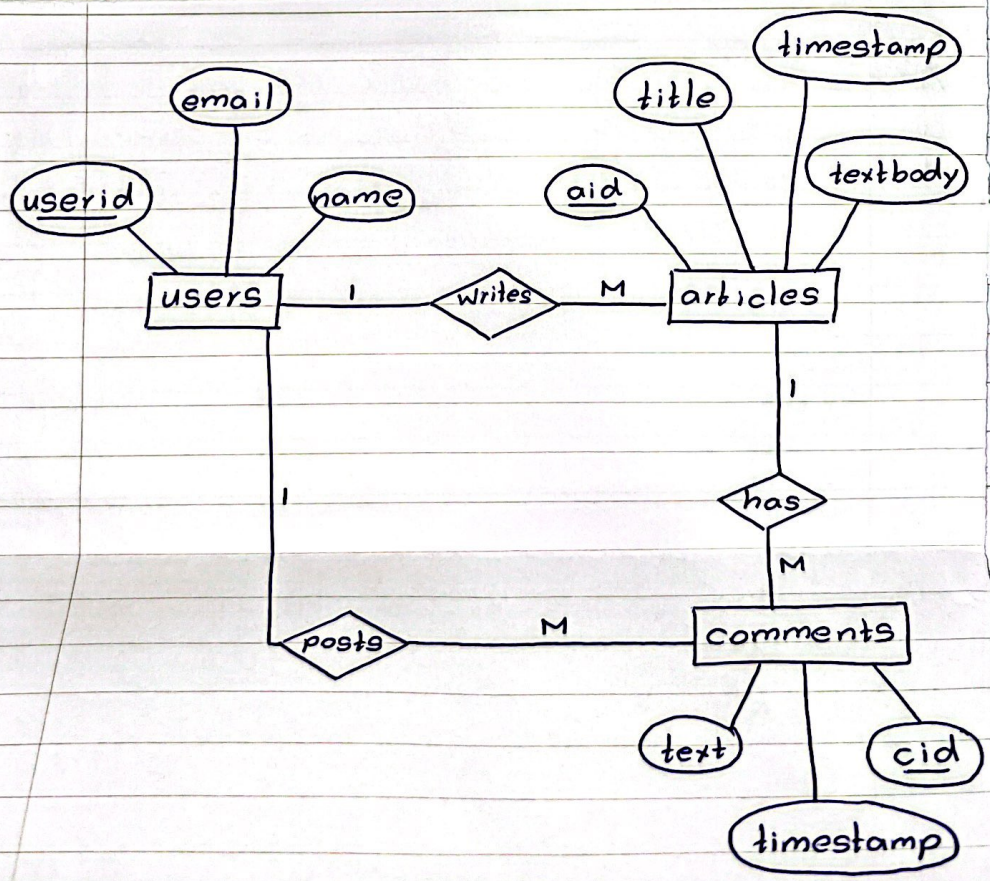
Primary Key

Foreign Key

Dep-No	Locations
1	Gampaha
2	Colombo

Primary Key

Sample ERD



Scanned with CamScanner

Date No

Relational Intergrity Constraints.

There're 3 types of Relational Intergrity Constraints.

01. Entity intergrity constraints. ← Key cons
02. Domain intergrity constraints. ← Domain cons
03. Referential intergrity constraints.

(01.) Entity intergrity constraints.

* All primary key entries are unique, and no part of a primary key may be null.

(02.) Domain intergrity constraints.

* All the values of an attribute must bound to a specific range.

(03.) Referential intergrity constraints.

* Referencing attribute should be a subset of referred attribute.
* Referential intergrity requires that a foreign key must have a matching primary key or it must be null.
* This constraint is specified between two tables (parent & child); it maintains the correspondence between rows in these tables.
* It means the reference from a row in one table to another table must be valid.

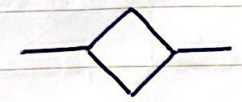
Scanned with CamScanner

Date No

Relationships.

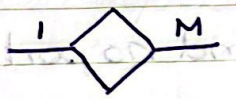
* Relationship is a association in between one or more entities.

*Chen Notation.

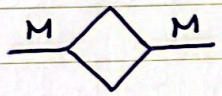


→ relationship.

→ one-to-one relationship.



→ one-to-many relationship.



→ many-to-many relationship.

How to find relationships.

* Relationships are associations between entities.
* Typically, a relationship is indicated by a verb connecting two or more entities.
* Employees are assigned to projects.
* Relationships should be classified in terms of cardinality.

  Ex: One-to-one, One-to-many.

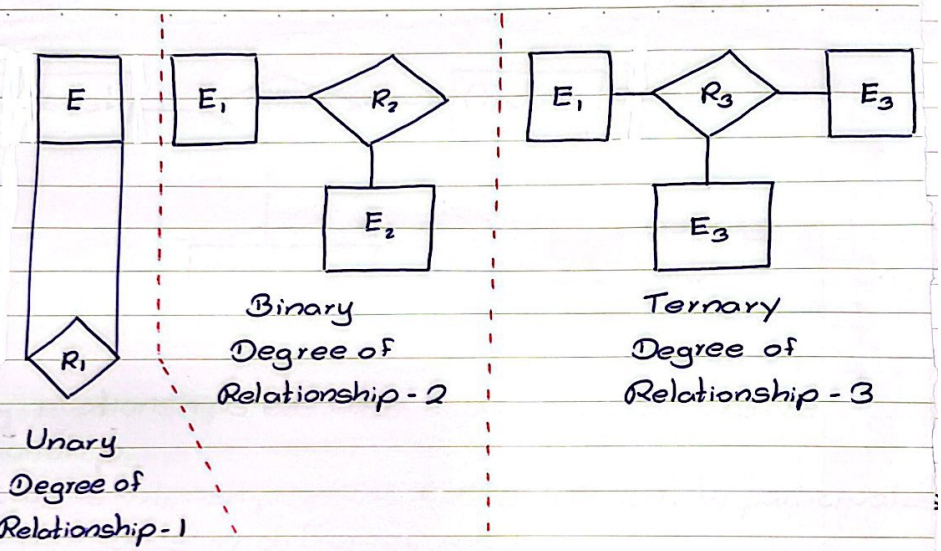
ProMate

Scanned with CamScanner

Degree of a Relationship.

* Number of entity types that participate in a relationship.

* If there're two entity types involved, it is a binary relationship type.
* If there're three entity types involved, it is a ternary relationship type.
* It is possible to have a n-array relationship.



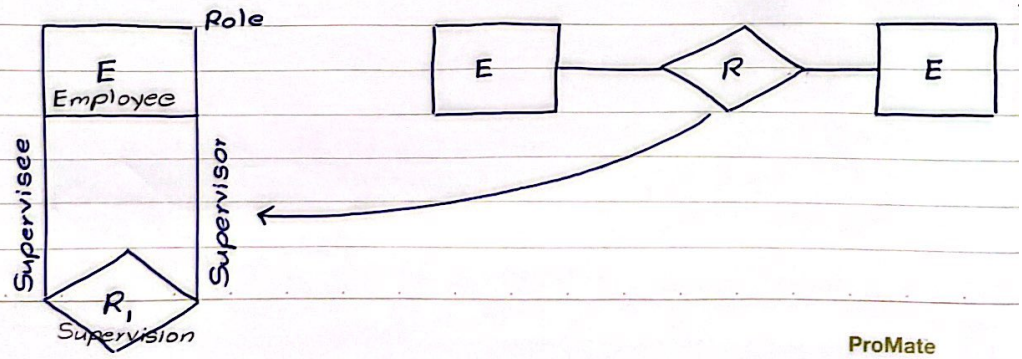
* Unary relationships are also known as a recursive relationship.

* It is a relationship where the same entity participates more than once in different roles.

* In ERD, need to display role names to distinguish participations.

Defining roles in a Unary Relationship.

Ex: EMPLOYEES are supervised by EMPLOYEES.



ProMate

Scanned with CamScanner

Date No

Connectivity

* The number of instances of one entity that can or must be associated with each instance of another entity.

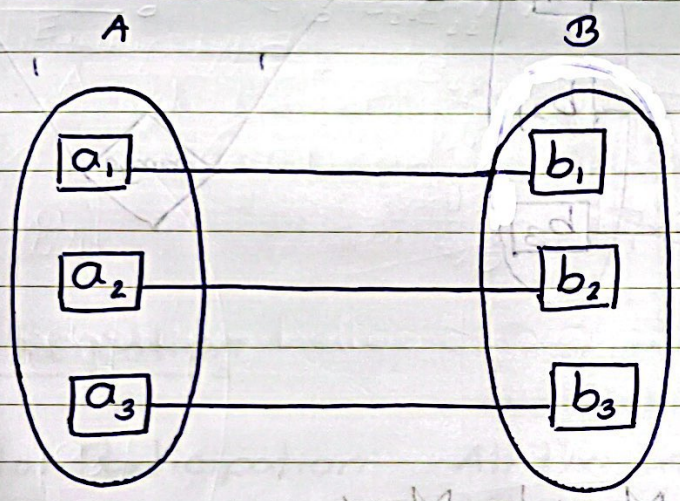
* If we have two entity types A and B, the cardinality constraints specifiers the number of instances of entity B that can (or must) be associated with entity A.

* Four possible categories are;
  01. One-to-one (1:1) relationship.
  02. One-to-many (1:m) relationship.
  03. Many-to-one (m:1) relationship.
  04. Many-to-many (m:m) relationship.

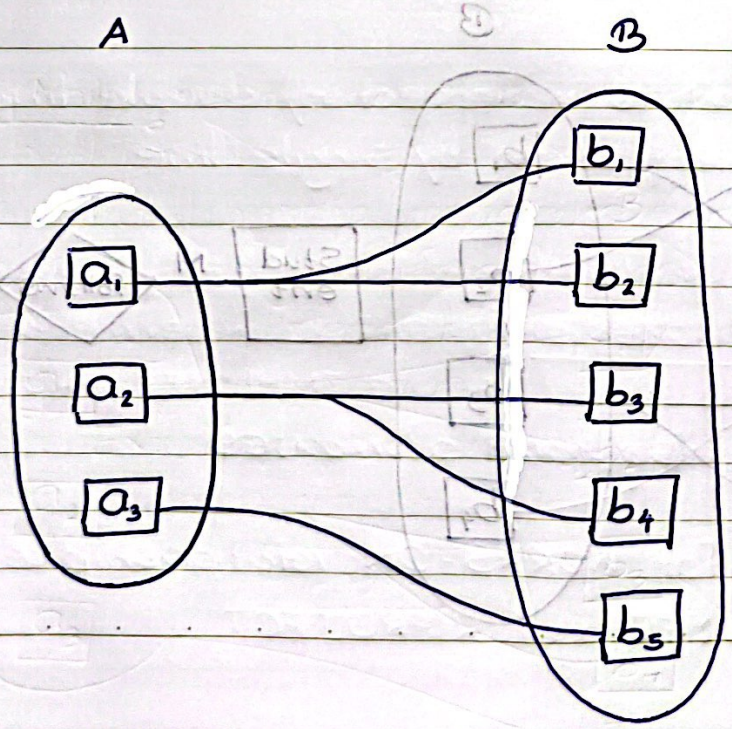
Scanned with CamScanner

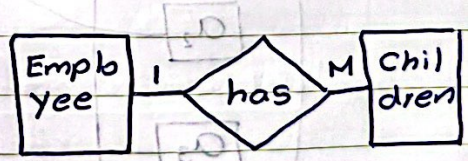
Date No

One-to-One



One-to-Many



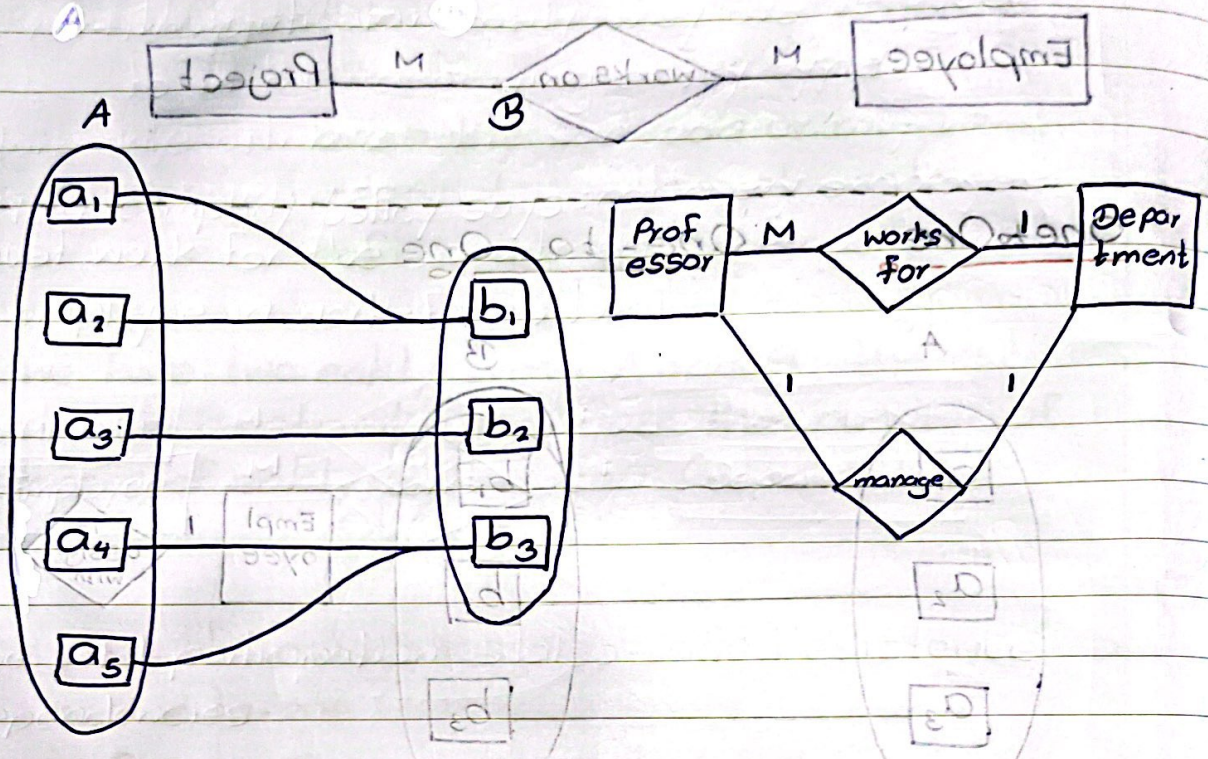


ProMate

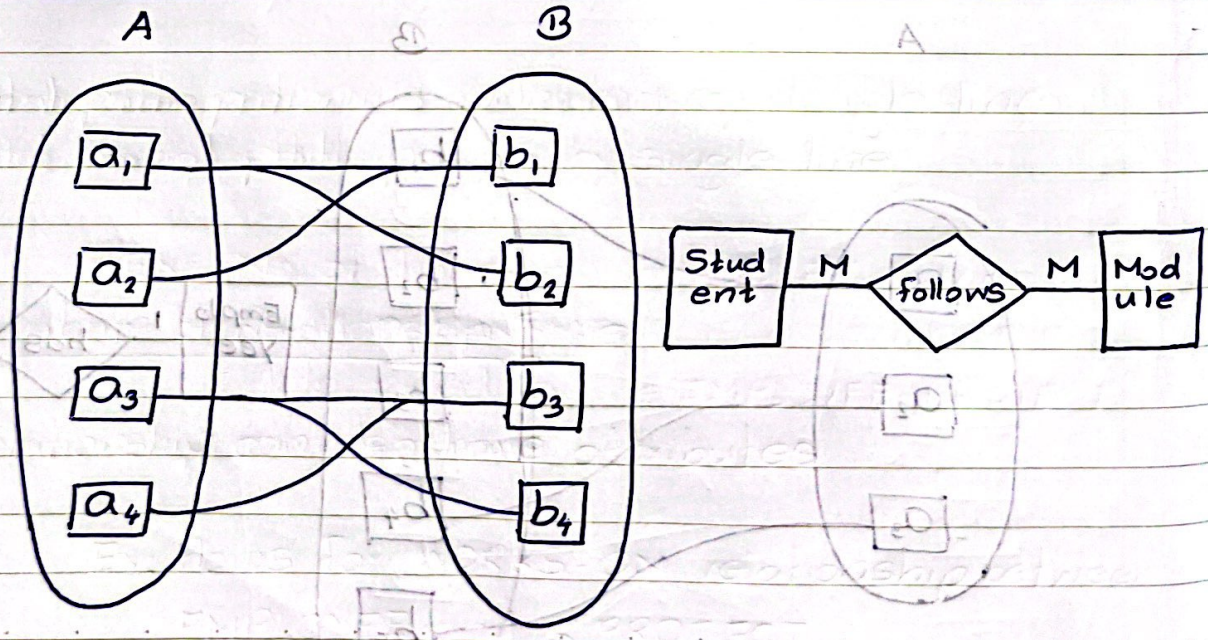
Scanned with CamScanner

Date No

Many-to-One.



Many-to-Many.



ProMate

Scanned with CamScanner

Weak Entity Types.

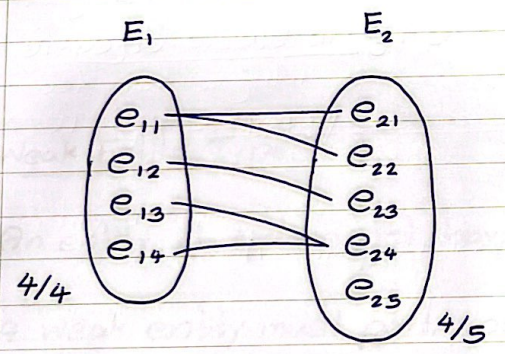
* An entity that does not have a key attribute.

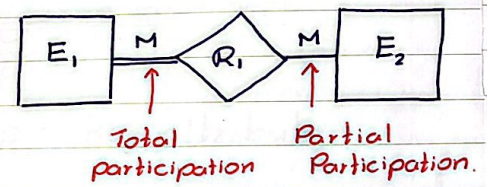
* A weak entity must participate in an identifying relationship type with an owner or identifying entity type.

* Entities are identified by the combination of:
  - A partial key of the weak entity type.
  - The particular entity they're related to in the identifying entity type.

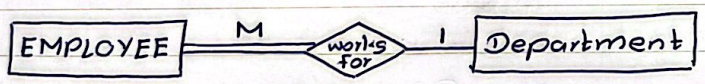
Participation.

* Total Participation - All the instances contribute to participation.
* Partial Participation - Few instances contribute to participation.





Ex: if company policy says that every employee must work for the department, then participation of employee in work-for is total.



* The total participation is also called existence dependencies.

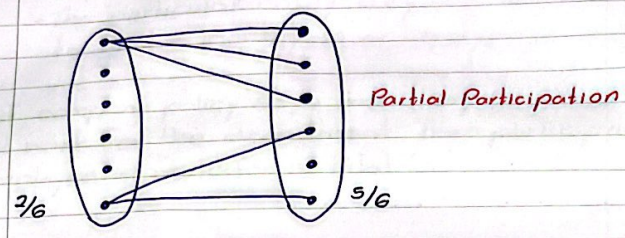
But we can't say that every employee must MANAGE a department. Hence relationship is partial.

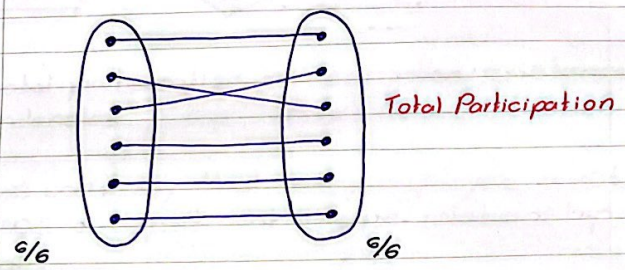
* Total participation → Double line.
* Partial participation → Single line.

ProMate

Scanned with CamScanner

Participation: Total vs Partial.



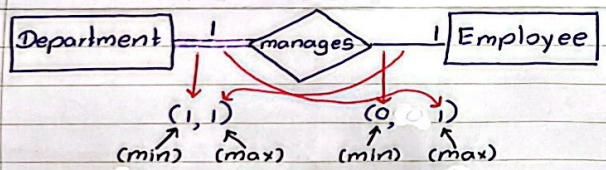


Cardinality.

Department is managed by an employee.

All the departments are managed by employess.

An employee can manage at most (max) 1 department.



Min

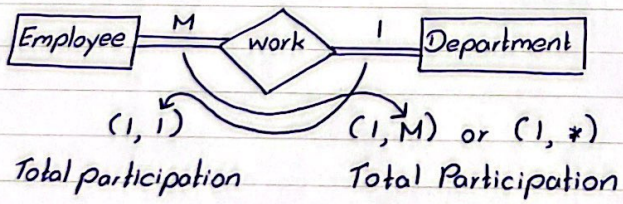
* Min denotes participation

1 → Total participation
0 → Partial participation

Max

* Max denotes connectivity of the opposite side.

Ex: An employee can work for excatly 1 department. But a department can have any no. of employees.

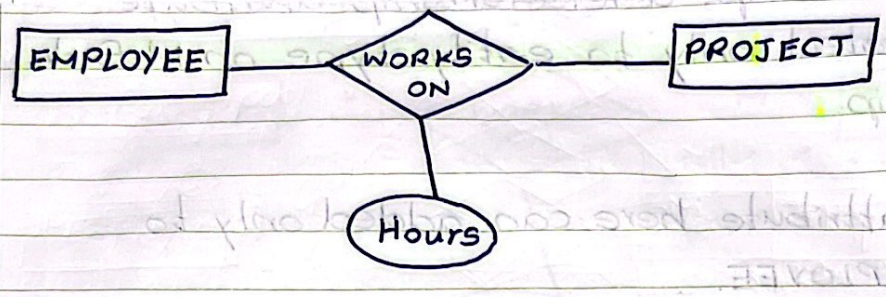


* Many denotes by "*" or "M"

Attribute of Relationship Type.

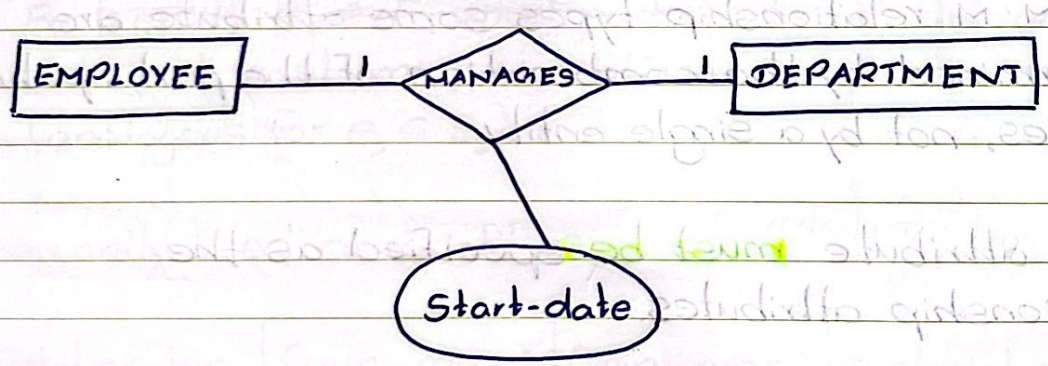
* Relationship can also have attributes.

  Ex: Hours for WORKS-ON relationship between EMPLOYEE and PROJECT.



* Attributes of 1:1 or 1:M relationship can be migrated to one of the participating entity types.

  Ex: Start-date attributes of MANAGES can be attribute of either DEPARTMENT or EMPLOYEE though conceptually it belongs to MANAGES.



* Because each EMPLOYEE MANAGES is a 1:1 relationship.
* So every DEPARTMENT / EMPLOYEE entity participate in atmost one relationship instance.
* So value of the start-date can be determined seperatly either by participating DEPARTMENT entity or participating EMPLOYEE entity.

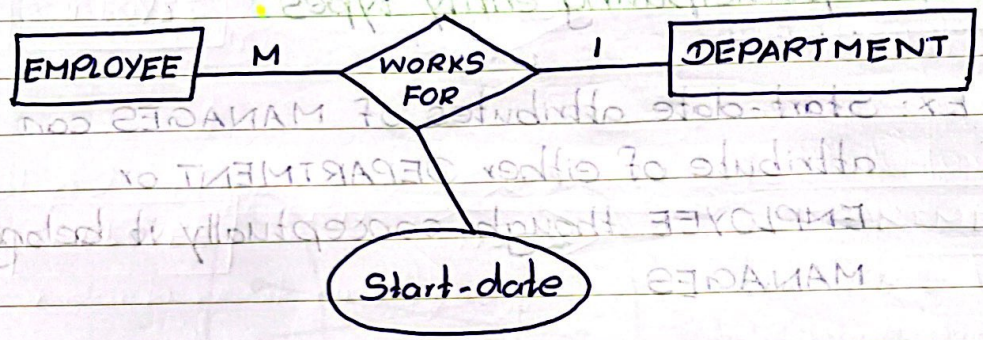
ProMate

Scanned with CamScanner

Date No

* For 1:M relationship, a relationship attribute can be migrated only to entity type on M-Side of relationship.

* Start-date attribute here can added only to EMPLOYEE.



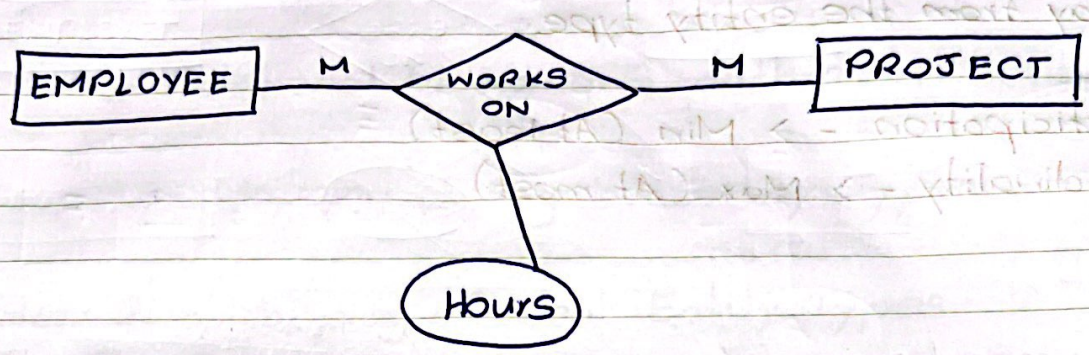
* For M:M relationship types some attribute are determined by the combination of the participating entities, not by a single entity.

* Such attribute must be specified as the relationship attributes.

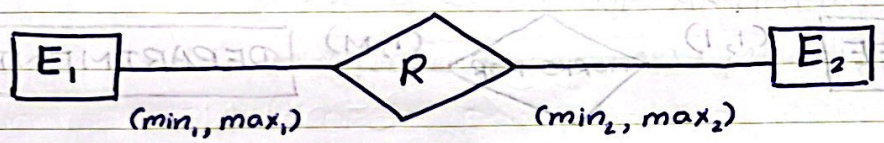
Ex: No. of hours an EMPLOYEE works on in DEPARTMENT is determent is determined by the EMPLOYEE-PROJECT combination.

ProMate

Scanned with CamScanner



Alternative Notation: Cardinality.



Each entity $e_1 \in E_1$ must participate in relationship set R at least min_1 and at most max_1 times. (analogous for $e_2 \in E_2$)

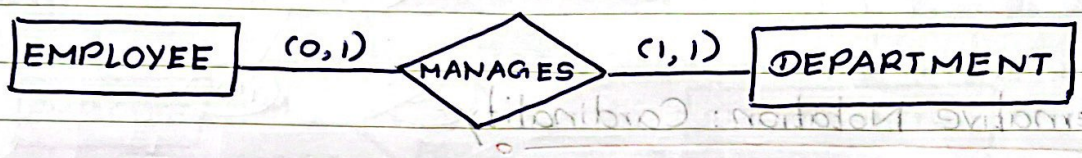
* Frequently used cardinalities.

Relationship	(min_1, max_1)	(min_2, max_2)	pictorial notation
many-to-many	(0, *)	(0, *)	
many-to-one	(0, 1)	(0, *)	
one-to-one	(0, 1)	(0, 1)	

Date No

* Read min, max values next to the entity type looking away from the entity type.

Participation -> Min (At least)
Cardinality -> Max (At most)



ProMate

Scanned with CamScanner

Date No

Relational Schema

ER-to-Relational Mapping Algorithm.

Step 1 - Mapping of Regular Entity Types.

Step 2 - Mapping of Weak Entity Types.

Step 3 - Mapping of Binary 1:1 Relation Types.

Step 4 - Mapping of Binary 1:M Relationship Types

Step 5 - Mapping of Binary M:M Relationship Types.

Step 6 - Mapping of Multivalued attributes.

Step 7 - Mapping of N-ary Relationship Types.

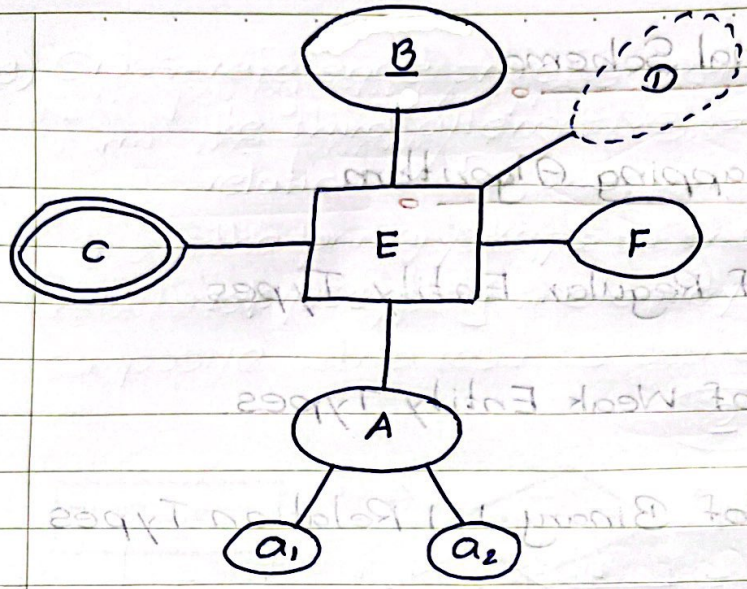
Step 1 - Mapping of Regular Entity Types.

* For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.

* Choose one of the key attributes of E as the primary key for R. If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

ProMate

Scanned with CamScanner



E			
a_1	a_2	B	F

Step 2 - Mapping of Weak Entity Sets.

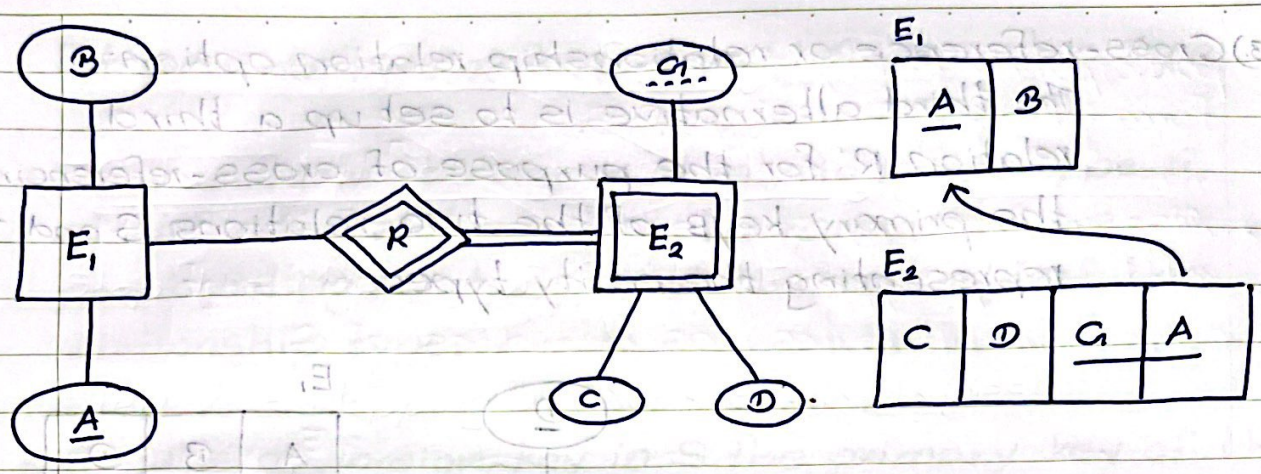
* For each weak entity type W in the ER schema with owner entity type E, create a relation R and include all simple attributes (or simple components of composite attributes) of W as attributes of R.

* In addition, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).

* The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

ProMate

Scanned with CamScanner



Step 3 - Mapping of Binary 1:1 Relation Types:

* For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.

* There're 3 possible approaches:

(01.) Foreign Key approach:

Chose one of the relations - S, say and include a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S.

(02.) Merged Relation option:

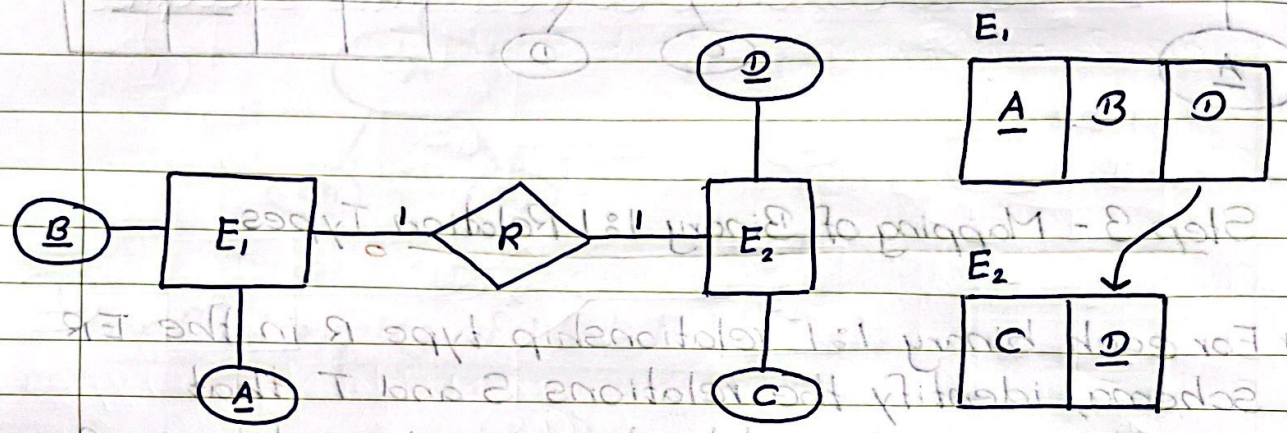
An alternate mapping of a 1:1 relationship type is possible by merging the two entity types and the relationship into a single relation. This may be appropiate when both participations are total.

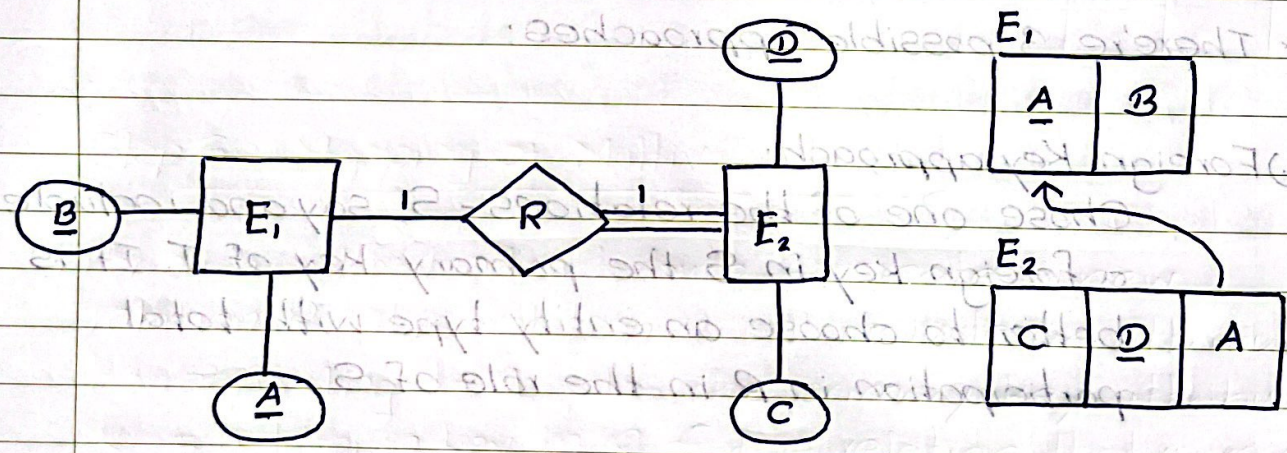
Scanned with CamScanner

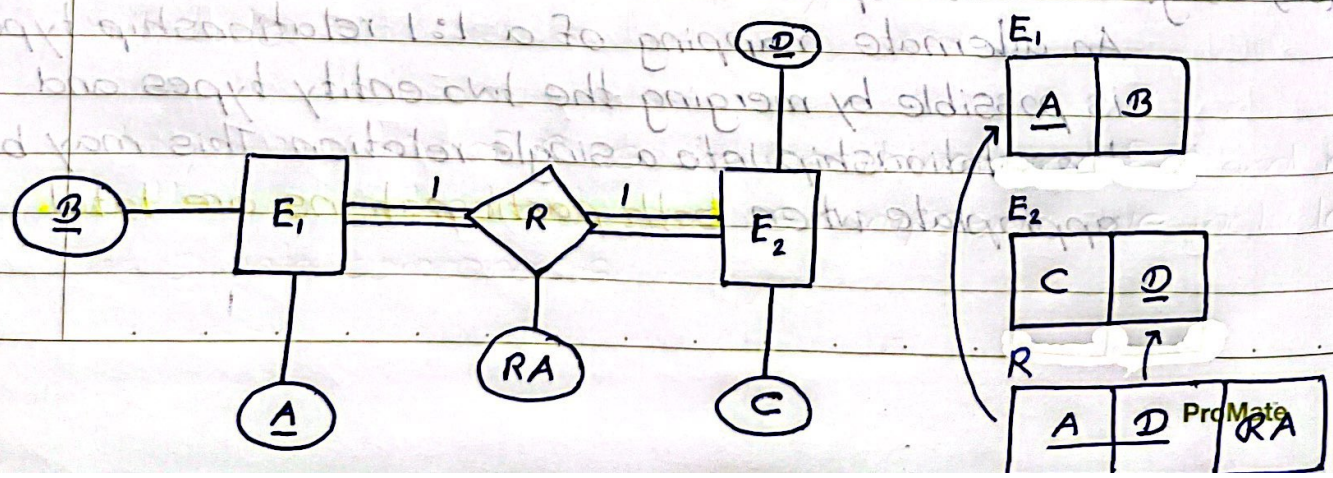
Date No

(03) Cross-reference or relationship relation option:

The third alternative is to set up a third relation R for the purpose of cross-referencing the primary keys of the two relations S and T representing the entity types.







Scanned with CamScanner

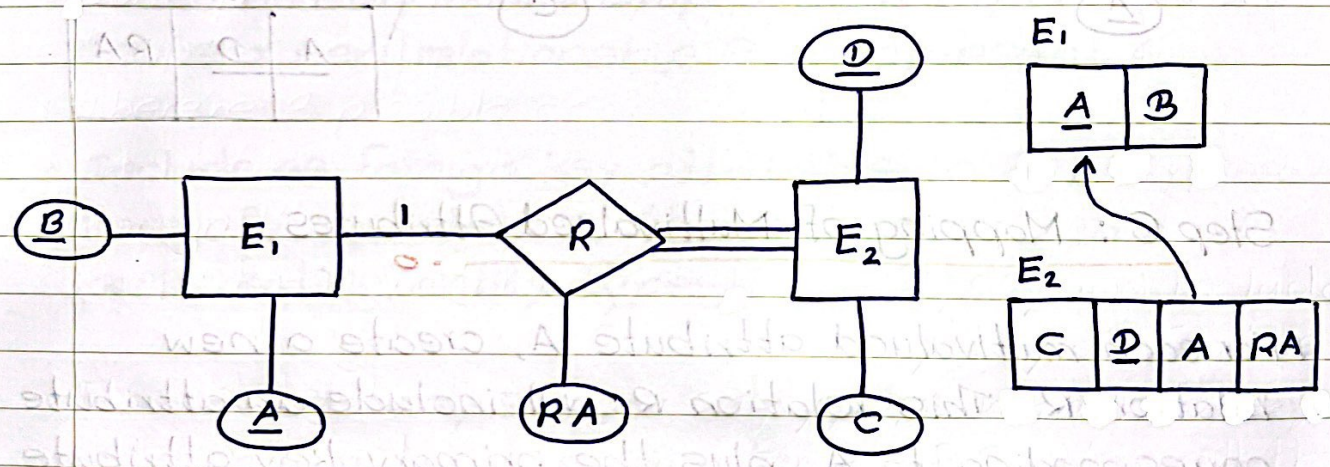
Date No

Step 4: Mapping of binary 1:M Relationship Type.

* For each regular binary 1:M relationship type R, identify the relation S that represent the participating entity type at the N-Side of the relationship type.

* Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R.

* Include any simple attributes of the 1:M relation type as attributes of S.



Step 5: Mapping of Binary M:M Relationship Types.

* For each regular binary M:M relationship type R, create a new relation S to represent R.

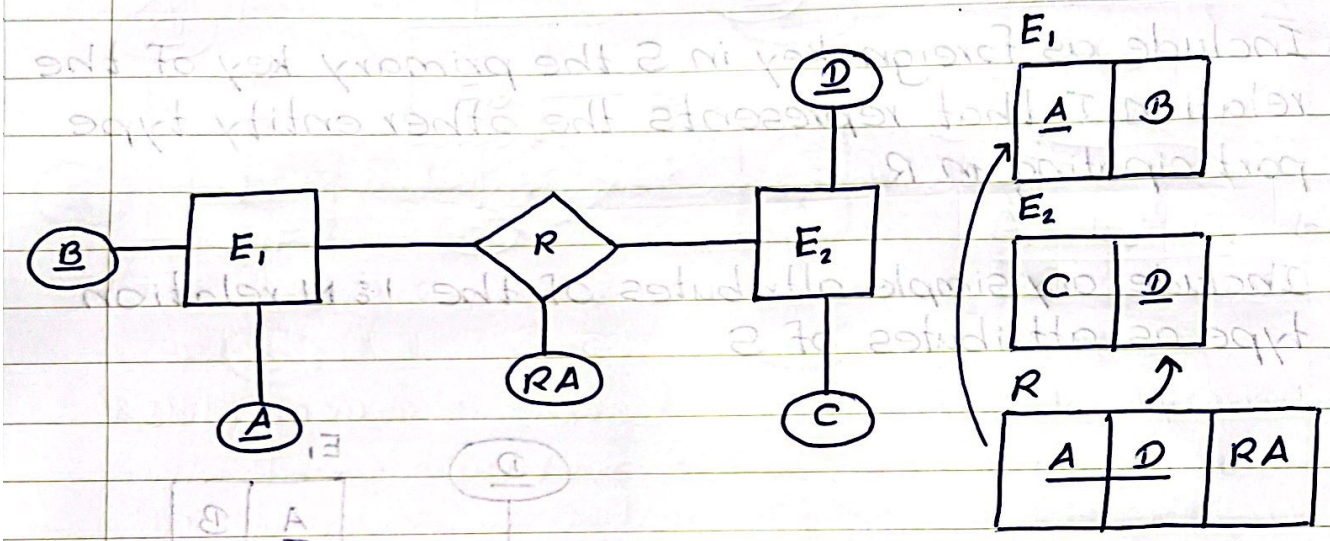
* Include as foreign key attributes in S the primary keys of the relations that represent the

ProMate

Scanned with CamScanner

participating entity types; their combination will form the primary key of S.

* Also include any simple attributes of the M:M relationship type (or simple components of composite attributes) as attributes of S.



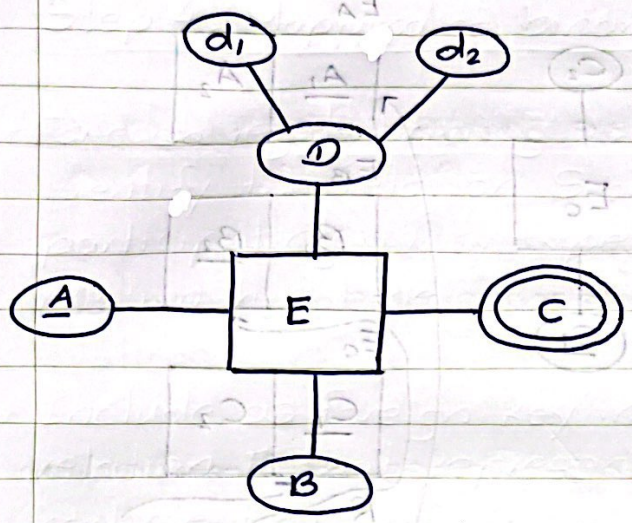
Step 6 : Mapping of Multivalued Attributes.

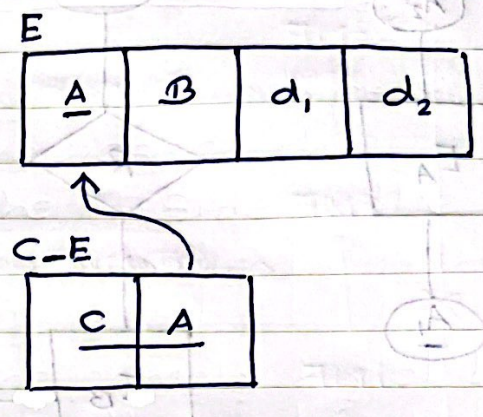
* For each multivalued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K as a foreign key in R of the relation that represents the entity type of relationship type that has A as an attribute.

* The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

ProMate

Scanned with CamScanner



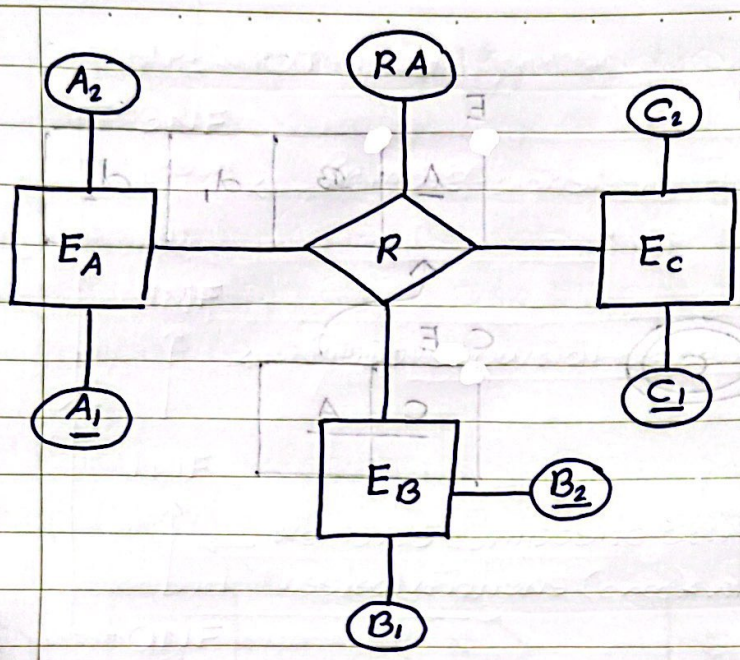


Step 7: Mapping of N-ary Relationship types.

* For each n-ary relationship type R, where $n > 2$, create a new relationship S to represent R.

* Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.

* Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.



E_A

A_1	A_2

E_B

B_1	B_2

E_C

C_1	C_2

R

A_1	B_2	C_1	RA

Normalization for Relational Databases.

What is Normalization?

The process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations.

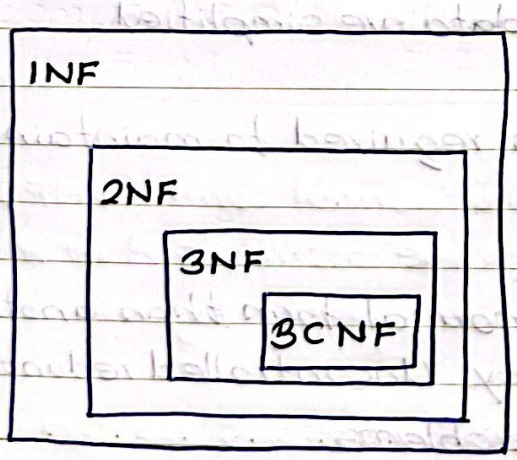
Why Normalize?

01) Reduce duplicates.
02) Organize data efficiently.
03) Reduce data anomalies. (Insertion, delete, update)

How to Normalize.

There is a sequence to normal forms:

* 1NF is considered the weakest.
* 2NF is stronger than 1NF.
* 3NF is stronger than 2NF.
* BCNF is considered the strongest.



- a relation in BCNF, is also in 3NF.
- a relation in 3NF, is also in 2NF.
- a relation in 2NF, is also in 1NF.

ProMate

Scanned with CamScanner

*3CNF (Higher and strict version of 3NF)

↑

3NF

↑ ← Remove transitive dependencies.

2NF

↑ ← Remove partial dependencies.

1NF

↑ ← Removes composite attributes, multi-valued and nested relations. (check whether values are atomic)

0NF

* Data anomalies are inconsistencies in the data stored in a database as a result of an operation such as update, insertion and/or deletion.

* Such inconsistencies may arise when have a particular record stored in multiple locations and not all of the copies are updated.

* The benifit of higher normal forms is that update semantics for the affected data are simplified.

* This means that applications required to maintain the database are simpler.

* A design that has a lower normal form than another design has more redundancy. Uncontrolled redundancy can lead to data intergrity problems.

ProMate

Scanned with CamScanner

Date No

First Normal Form

Disallows composite attributes, multi-valued attributes and nested relations; attributes whose values for an individual tuple are non-atomic.

Functional Dependency (FD)

Uses the concepts of FDs, primary key.

* Prime Attribute / Key attribute.

Attribute that is member of the primary key K.

* Full Functional Dependency.

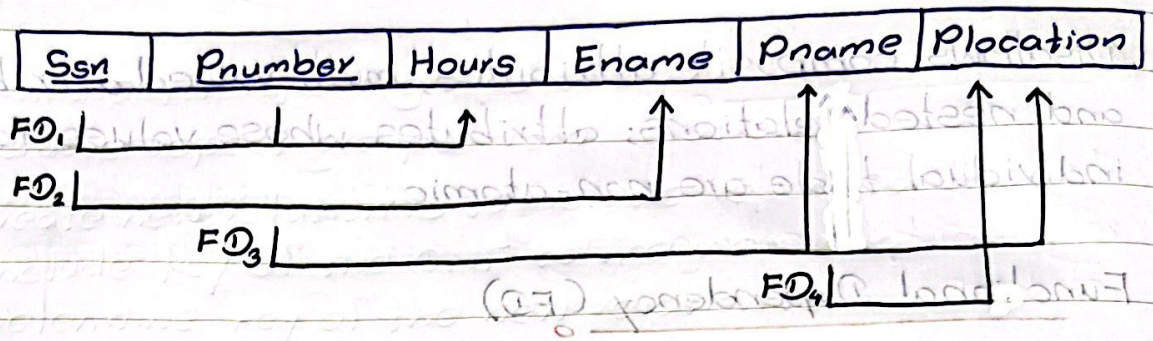
A FD $Y \rightarrow Z$ where removal of any attribute from Y means the FD does not hold anymore.

FD_1 : Full functional dependency.
FD_2 / FD_3 : Partial dependency.
FD_4 : Transitive dependency.

ProMate

Scanned with CamScanner

EMP_PROJ

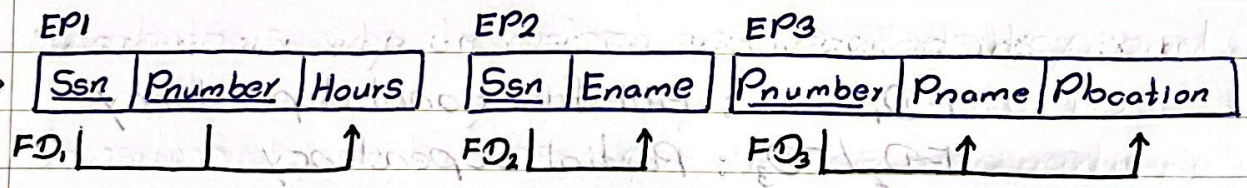


Normalization into 2NF

Second Normal Form

A relation schema R is in second normal form (2NF) if every non-prime attribute A in R is fully functionally dependent on the primary key.
(That is, we don't have any partial functional dependency.)

R can be decomposed into 2NF relations via the process of 2NF normalization.



Third Normal Form

A relation schema R is in third normal form (3NF) if it is in 2NF and no-non prime attribute A in R is transitively dependent on the primary key.

R can be decomposed into 3NF relations via the process of 3NF normalization.

ProMate

Scanned with CamScanner

Date No

* In $X \rightarrow Y$ and $Y \rightarrow Z$, with X as the primary key, we consider this a problem only if Y is <u>not</u> a candidate key.
When Y is a candidate key, there is no problem with the transitive dependency.

Exercise

PO-No	PO-Date	EMP-Code	SUPP-No	SUPP-Name	PART-No	PART-Des	PART-Qty

- Bring the above table to 1NF.

PO1

PO_No	PO_Date	EMP_Code	SUPP_No	SUPP_Name

PO2

PO_No	PART_No	PART_Des	PART_Qty

- Bring the above tables to 2NF.

PO1_1

PO_No	PO_Date	EMP_Code	SUPP_No	SUPP Name

PO2_1

PO_No	PART_No	PART_Qty

PO2_2

PART_No	PART_Des

- Bring the (PO1_1) table to 3NF.

O1_1_1

SUPP_No	SUPP_Name

PO1_1_2

PO_No	PO_Date	EMP_Code	SUPP_NO

ProMate

Scanned with CamScanner

Date No

1NF - Actions Required

01.) Examine for repeat groups of data.
02.) Remove repeat groups from relation
03.) Create new relation/s to include repeated data.
04.) Include key of the ONF to the new relation/s.
05.) Determine key of the new relation/s.

Second Normal Form - 2NF

Deals with the relationship between non-key and key fields.

It is relevant when the key is composite, i.e. consists of several fields.

Third Normal Form - 3NF

A relation is in 3NF if it is in 2NF and each non-key attribute is only dependent on the whole key, and not dependent on any non-key attribute. i.e. no transitive dependencies.

Problems

01.) Insert problem.
02.) Delete problem.
03.) Update problem.

ProMate

Scanned with CamScanner

Date No

* Check List.

- Identify all the entities and related attributes. (Make sure you recognize weak entities as well)
- Identify all the relationships. (Make sure you find relationship attributes, if any)
- Write your assumptions. (If any)
- Now draw your initial ER diagram.
- Now identify participation, connectivity and carnality constraints. (If any)
- Confirm whether you have identified identifying relationships. (If any)
- Cross examine your ER reading and reading the given description.
- Confirm whether you have correctly specified participation connectivity and carnality constraints.

ProMate

Scanned with CamScanner