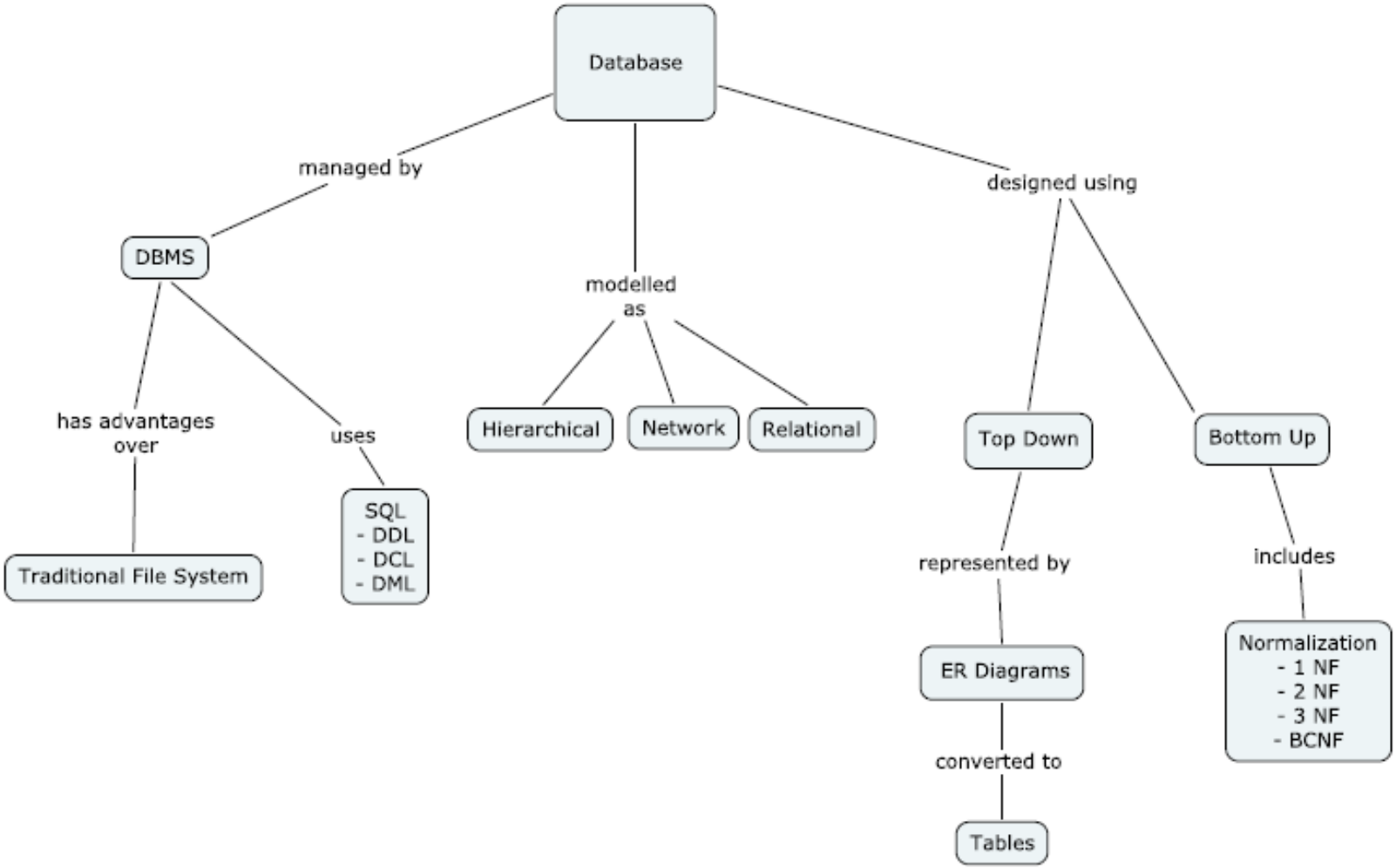
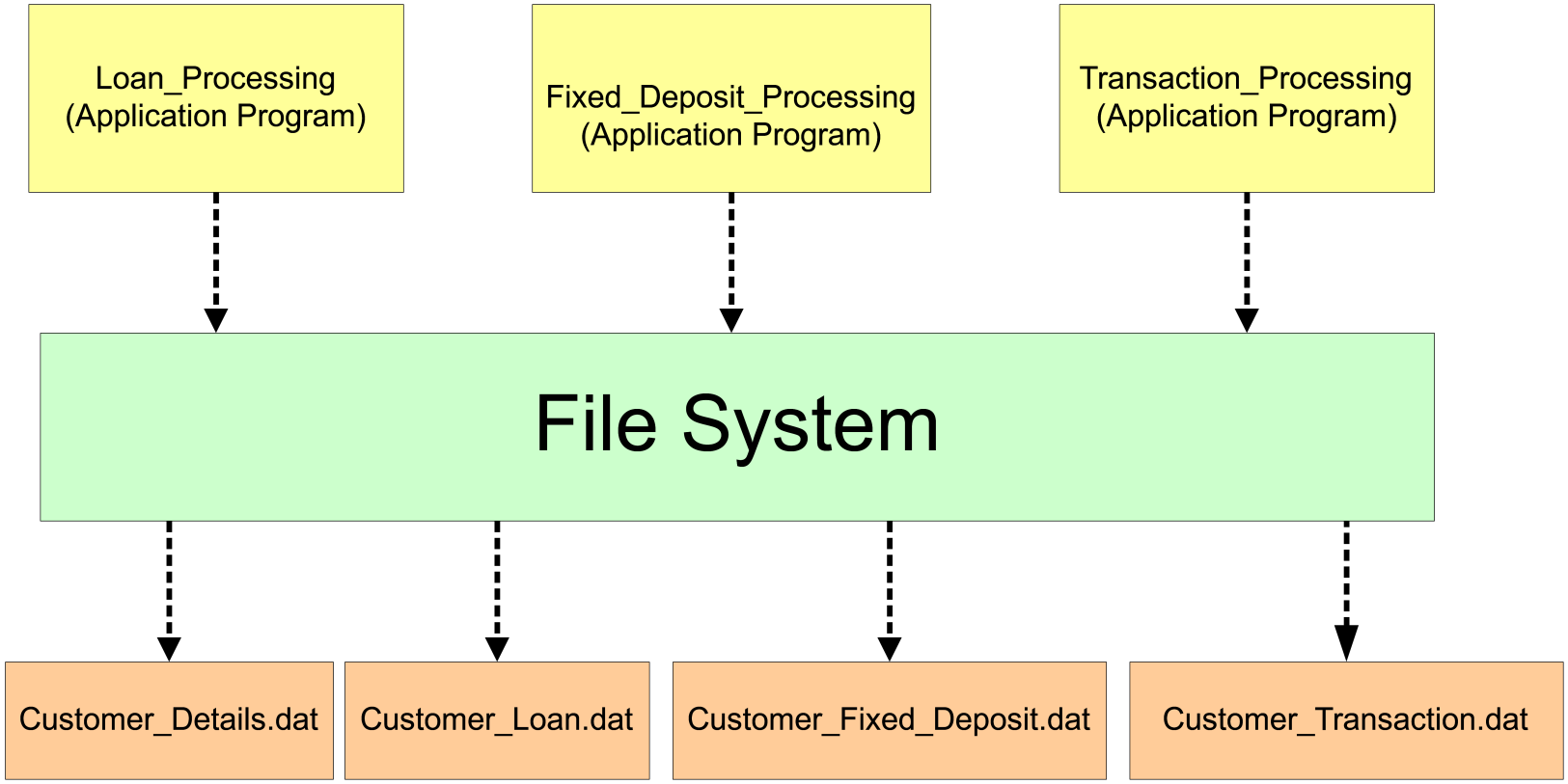
C Map- RDBMS Core



Traditional Method of Data Storage



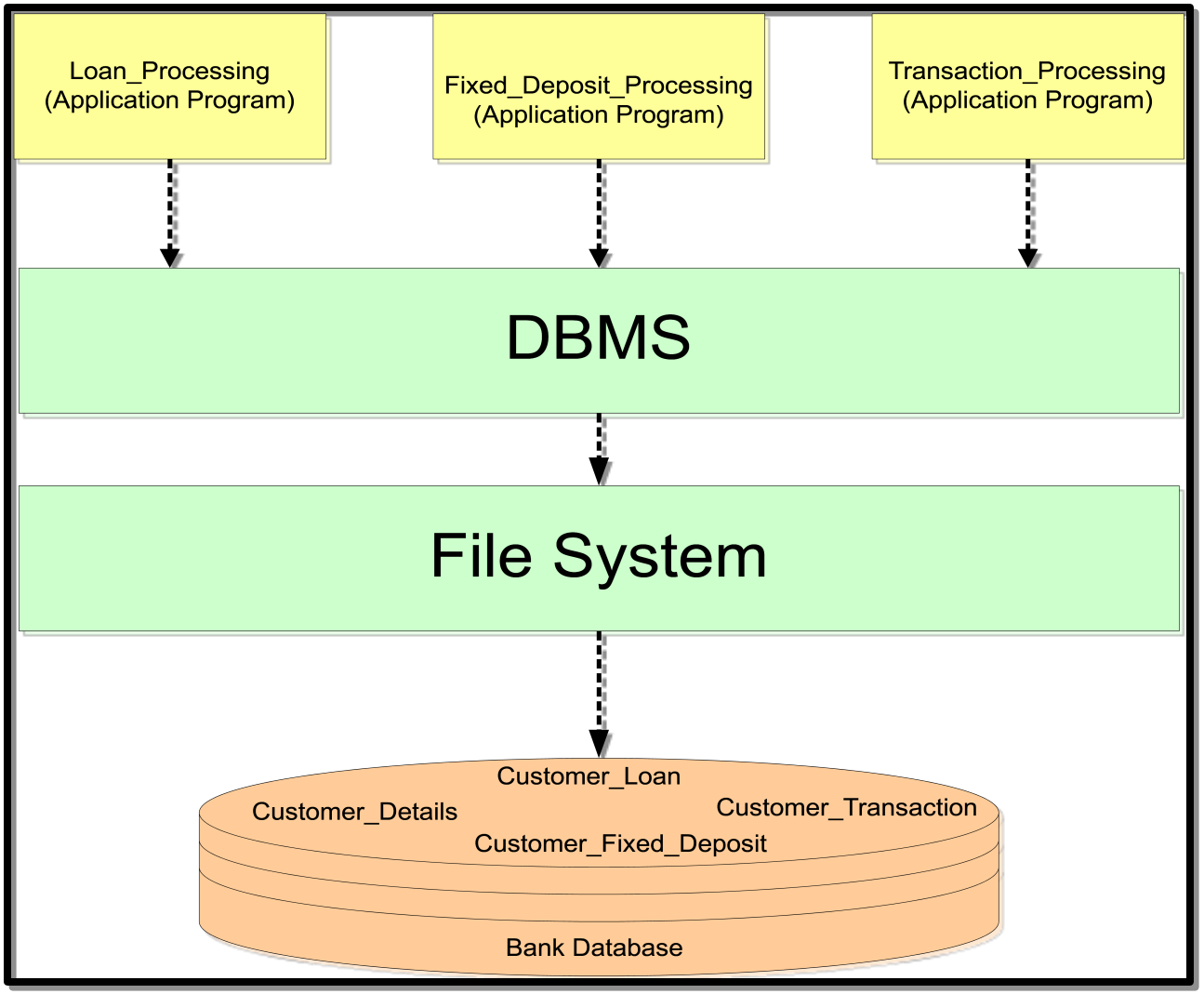
Problems: Traditional Approach

- Data Security
- Data Redundancy
- Data Isolation
- Program / Data Dependence
- Lack of Flexibility
- Concurrent Access Anomalies

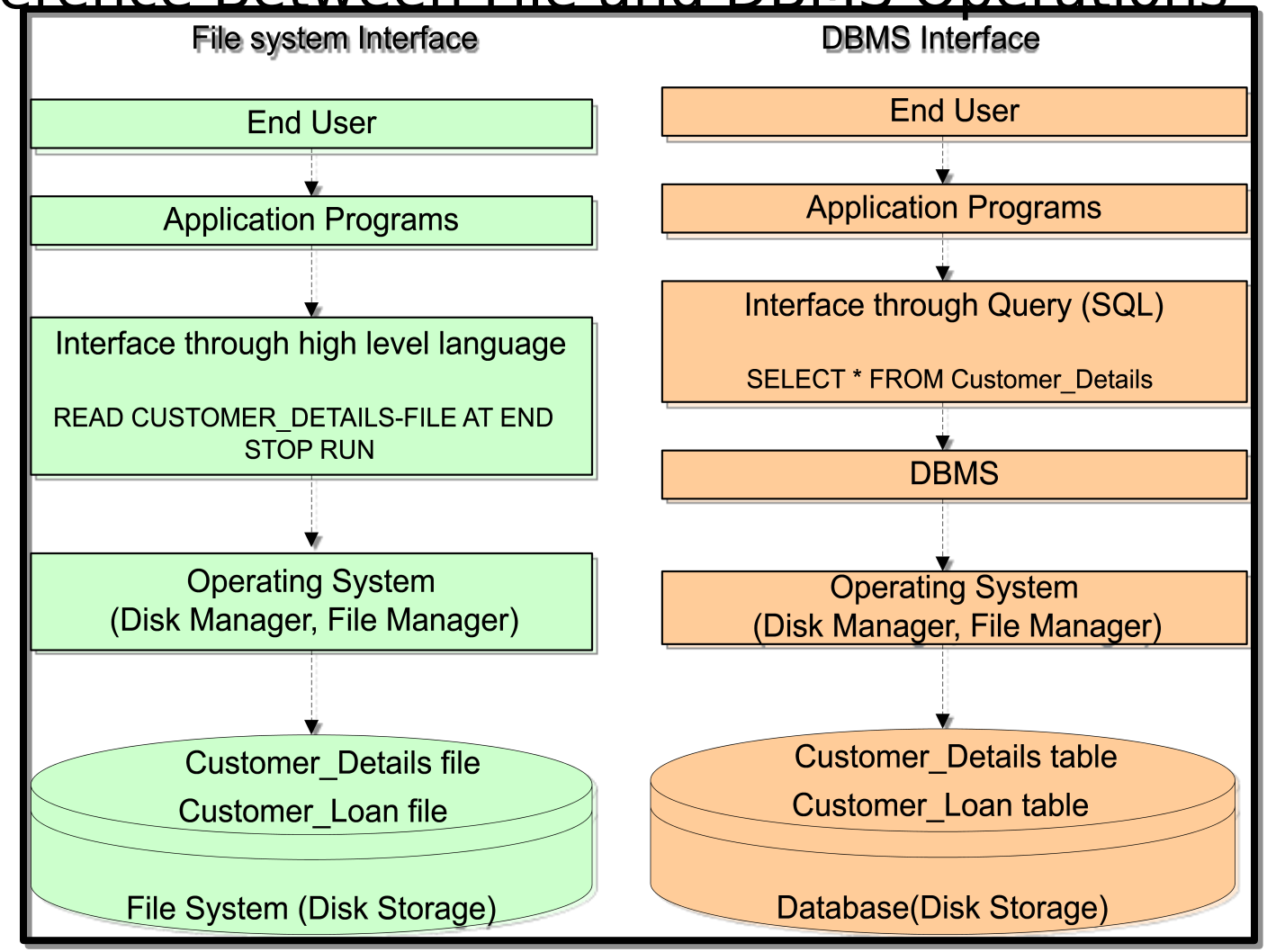
Database Management System

- DBMS is consist of collection of interrelated files and set of programs.
- These set of programs allow users to access and modify files
- *Primary Goal is to provide a convenient and efficient way to store, retrieve and modify information*
- Layer of abstraction between the application programs and the file system

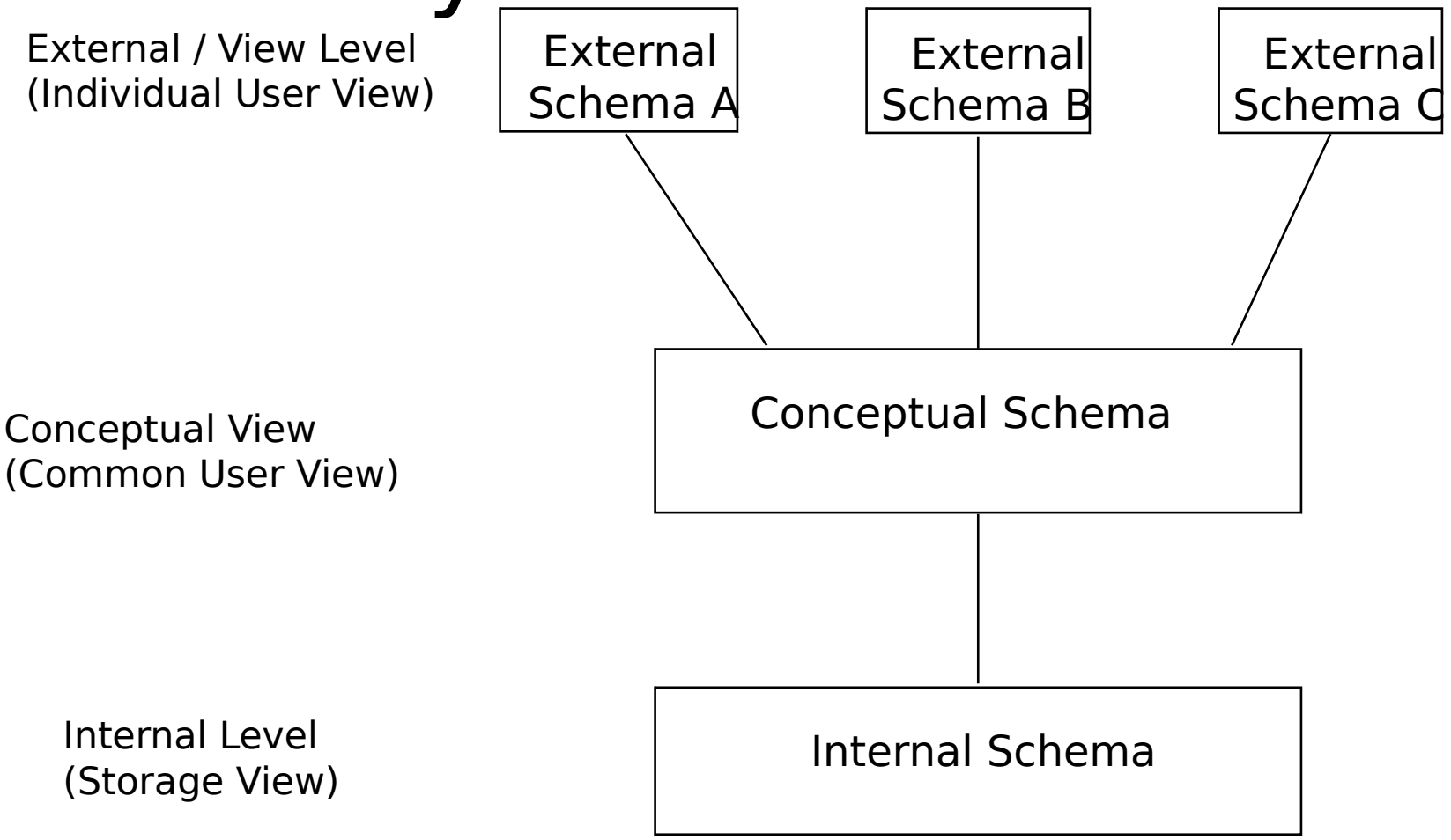
Where does the DBMS fit in?



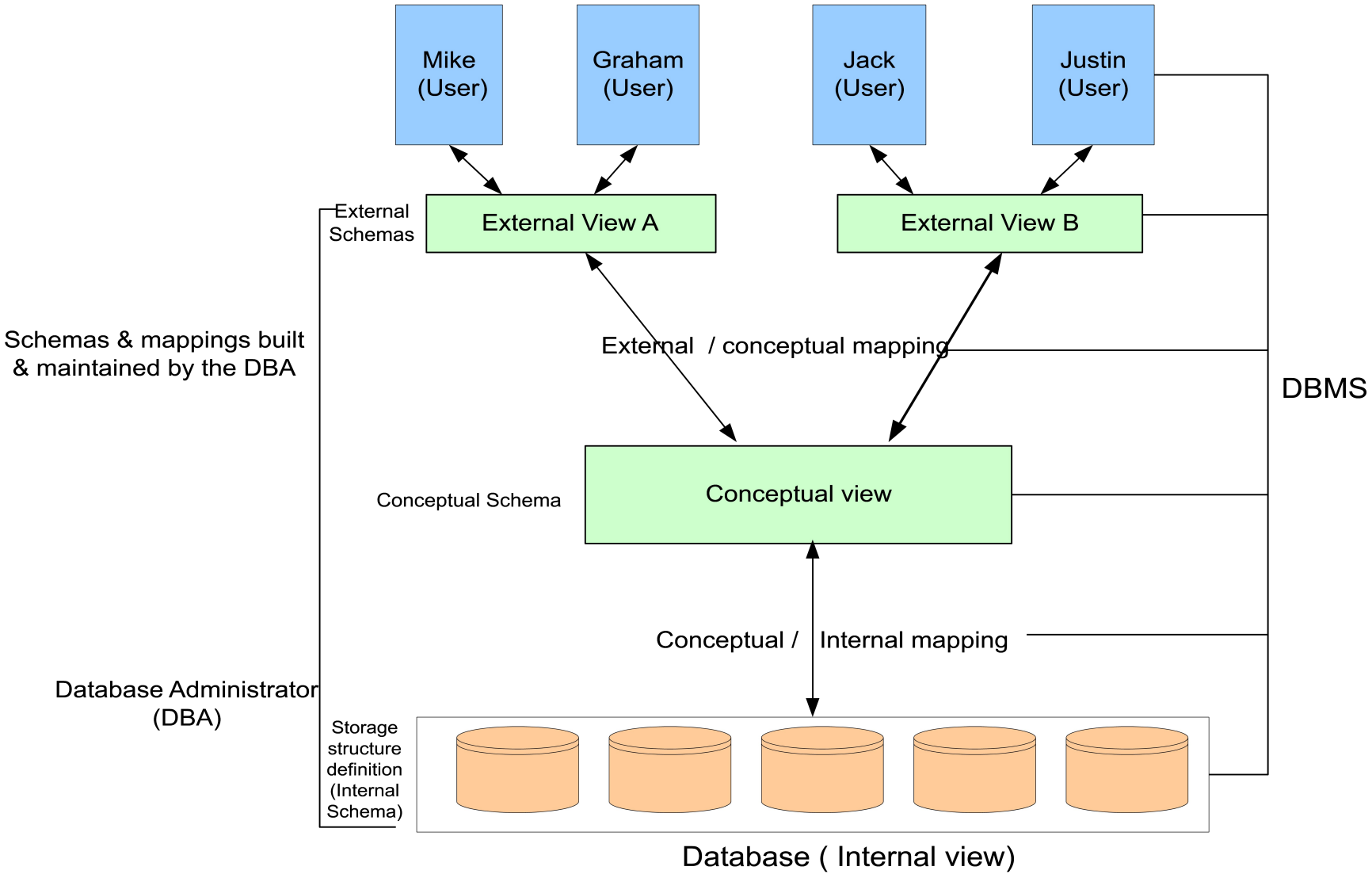
Difference Between File and DBMS Operations



Three-layer Architecture



Detailed System Architecture



An example of the three levels

External

```
Customer_Loan
Cust_ID         : 101
Loan_No             : 1011
Amount_in_Dollars      : 8755.00
```

Conceptual

```
CREATE TABLE Customer_Loan (
Cust_ID         NUMBER(4)
Loan_No             NUMBER(4)
Amount_in_Dollars  NUMBER(7,2))
```

Internal

```
Cust_ID                 TYPE = BYTE (4), OFFSET = 0
Loan_No                 TYPE = BYTE (4), OFFSET = 4
Amount_in_Dollars           TYPE = BYTE (7), OFFSET = 8
```

Users of a DBMS

- Database Administrator (DBA)
  - Managing information contents
  - Liaison with users
  - Enforcing security and integrity rules
  - Strategizing backup & recovery
  - Monitoring performance
- Database designers
- Application programmers
- End users

Advantages of a DBMS

- Data independence
- Reduction in data redundancy
- Better security
- Better flexibility
- Effective data sharing
- Enforces integrity constraints
- Enables backup and recovery

Relational model basics

- Data is viewed as existing in two dimensional tables known as relations
- A relation (table) consists of unique attributes (columns) and tuples (rows)
- Sometimes the value to be inserted into a particular cell may be unknown, or it may have no value. This is represented by a **NULL**
- Null is not the same as zero, blank or an empty string
- Relational Database: Any database whose logical organization is based on relational data model.
- RDBMS: A DBMS that manages the relational database.

Keys in relational model

- **Candidate key**

  A Candidate key is a set of **one or more attributes(minimal)** that can uniquely identify a row in a given table.

- **Primary Key**

  During the creation of the table, the Database Designer chooses one of the Candidate Key from amongst the several available, to uniquely identify row in the given table.

- **Alternate Key**

  The candidate key that is chosen to perform the identification task is called the *primary key* and the remaining candidate keys are known as alternate keys.

  No of Alternate Keys = No of Candidate Keys - 1

Key and Non-key Attributes in Relational Model

- Key Attributes

  The attributes that participate in the Candidate key are Key attributes

- Non-Key Attributes
  - The attributes other than the Candidate Key attributes in a table/relation are called Non-Key attributes.

    OR
  - The attributes which do not participate in the Candidate key.

Example

Given a relation

Trainee(Empno, FirstName, LastName, Email, PhoneNo)

Assumptions:

i. **Empno for each trainee is different.**
ii. **Email for each trainee is different**
iii. **PhoneNo for each trainee is different**
iv. **Combination of FirstName and LastName for each trainee is different**

Candidate key:

{Empno},{Email},{PhoneNo},{FirstName,LastName}

Primary key:

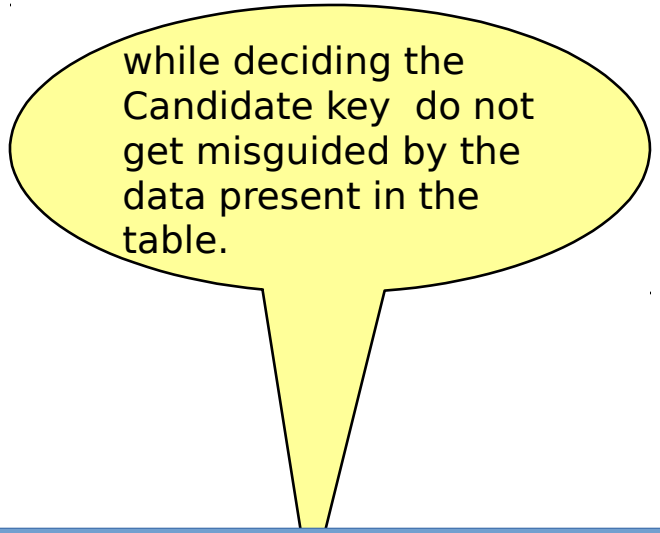
{Empno}

Alternate Key:

{Email},{PhoneNo},{FirstName,LastName}

What are the candidate keys?

Case 1
Assumptions
One customer can have only one account
An account can belong to only one customer



Cust_ID	Cust_Last_ Name	Cust_Mid _Name	Cust_First _Name	Account _No	Account_ Type	Bank_Branch	Cust_Email
101	Smith	A.	Mike	1020	Savings	Downtown	Smith_Mike@yahoo.com
102	Smith	S.	Graham	2348	Checking	Bridgewater	Smith_Graham@rediffmail.com
103	Langer	G.	Justin	3421	Savings	Plainsboro	Langer_Justin@yahoo.com
104	Quails	D.	Jack	2367	Checking	Downtown	Quails_Jack@yahoo.com
105	Jones	E.	Simon	2389	Checking	Brighton	Jones_Simon@rediffmail.com

Customer_Detail records from Customer_Details table

What are the candidate keys?

Case 2
Assumptions
One customer can have many accounts
An account can belong to only one customer

Cust_ID	Cust_Last_ Name	Cust_Mid _Name	Cust_First _Name	Account _No	Account_ Type	Bank_Branch	Cust_Email
101	Smith	A.	Mike	1020	Savings	Downtown	Smith_Mike@yahoo.com
102	Smith	S.	Graham	2348	Checking	Bridgewater	Smith_Graham@rediffmail.com
103	Langer	G.	Justin	3421	Savings	Plainsboro	Langer_Justin@yahoo.com
104	Quails	D.	Jack	2367	Checking	Downtown	Quails_Jack@yahoo.com
105	Jones	E.	Simon	2389	Checking	Brighton	Jones_Simon@rediffmail.com

Customer_Detail records from Customer_Details table

What are the candidate keys?

Case 3 :
Assumptions

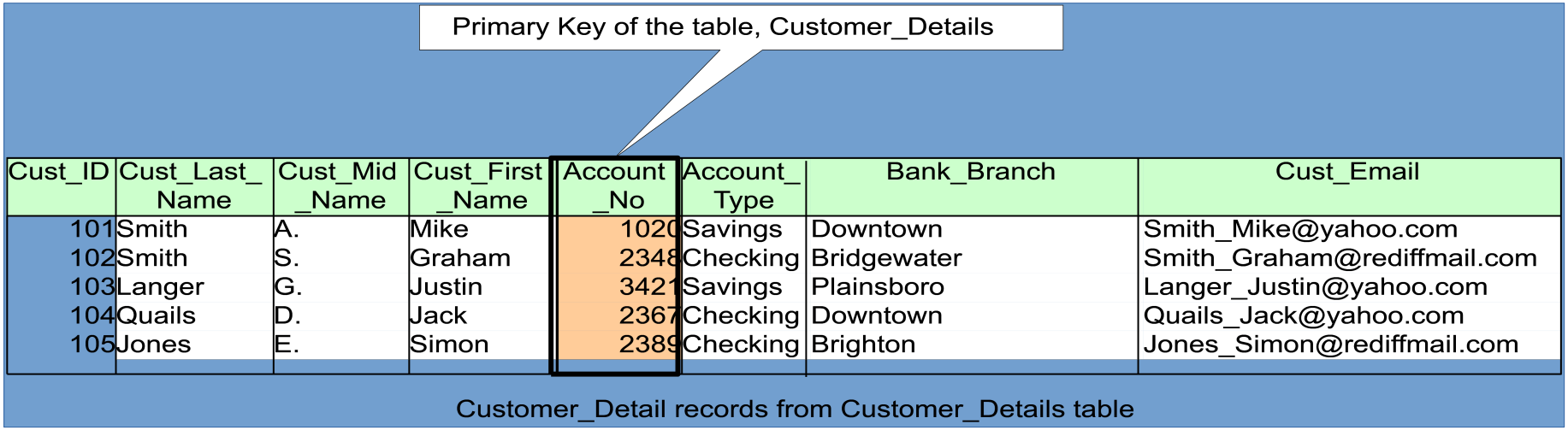
One customer can have many accounts.
An account can belong to more than one customer (joint account)

Cust_ID	Cust_Last_ Name	Cust_Mid _Name	Cust_First _Name	Account _No	Account_ Type	Bank_Branch	Cust_Email
101	Smith	A.	Mike	1020	Savings	Downtown	Smith_Mike@yahoo.com
102	Smith	S.	Graham	2348	Checking	Bridgewater	Smith_Graham@rediffmail.com
103	Langer	G.	Justin	3421	Savings	Plainsboro	Langer_Justin@yahoo.com
104	Quails	D.	Jack	2367	Checking	Downtown	Quails_Jack@yahoo.com
105	Jones	E.	Simon	2389	Checking	Brighton	Jones_Simon@rediffmail.com

Customer_Detail records from Customer_Details table

Choosing a Primary key from Candidate keys -Guidelines

- Give preference to numeric column(s)
- Give preference to single attribute
- Give preference to minimal composite key



Cust_ID	Cust_Last_ Name	Cust_Mid _Name	Cust_First _Name	Account _No	Account_ Type	Bank_Branch	Cust_Email
101	Smith	A.	Mike	1020	Savings	Downtown	Smith_Mike@yahoo.com
102	Smith	S.	Graham	2348	Checking	Bridgewater	Smith_Graham@rediffmail.com
103	Langer	G.	Justin	3421	Savings	Plainsboro	Langer_Justin@yahoo.com
104	Quails	D.	Jack	2367	Checking	Downtown	Quails_Jack@yahoo.com
105	Jones	E.	Simon	2389	Checking	Brighton	Jones_Simon@rediffmail.com

Customer_Detail records from Customer_Details table

Foreign Key

- **Foreign key**
  - A Foreign Key is a set of attribute (s) whose values are required to match values of a column in the same or another table.

DEPT **EMP**

(Parent /Master/Referenced Table) **(Child /Referencing Table)**

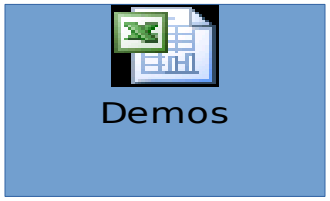
DeptNo	DName
D1	IVS
D2	ENR

EmpNo	EName	EDeptNo
1001	Elsa	D1
1002	John	D2
1003	Maria	Null
1004	Maida	D1

- Points to remember
  - Foreign key values do not (usually) have to be unique.
  - Foreign keys can also be *null* .
  - To enter the data in child table corresponding data must be present in master table or NULL is the default entry in child table in the referenced column (FK column)

Foreign Key

- Foreign key



- Points to remember
  - A Foreign Key is a set of attributes of a table, whose values are required to match values of some Candidate Key in the same or another table
  - Foreign Key column must match the values of the corresponding Candidate Key column. This is known as Referential constraint.
  - A table which has a Foreign Key referring to its own Candidate Key is known as Self-Referencing table

Database design Technique

Database Design Techniques

Top down Approach

In **Top down** approach we start defining the data set and then we go on defining data elements in those sets. This approach generally leads to redundant information in one or more table.

Some references call this **Entity - Relationship modeling**.

Bottom Up approach

In **Bottom up** approach we start defining required attribute first and then group these attribute to form the entities. Another term used for this method is **normalization** from functional dependencies.

ER modeling

- **ER modeling:** A graphical technique for *understanding* and organizing the data independent of the actual database implementation.

- **Entity:** Any thing that may have an independent existence and about which we intend to collect data. also known as **Entity type.**
  - e.g.: **Trainee**

- **Relationships:** Associations between entities.
  - e.g.: Trainee belongs to a Batch

- **Attributes:** Properties/characteristics that describe entities.
  - e.g.: Trainee Name, BatchName, DOB, Address, etc.

Entity Types

- **Regular Entity**: Entity that has its own key attribute (s).

  e.g.: Employee, student ,customer, policy holder etc.

- **Weak entity**: Entity that depends on other entity for its existence and doesn't have key attribute (s) of its own

  e.g. : spouse of employee

Attributes

- The set of possible values for an attribute is called the **domain** of the attribute

  e.g.:

  1. The domain of attribute ***marital status*** is having four values: single, married, divorced or widowed.

  2. The domain of the attribute month is having twelve values ranging from January to December.

- **Key attribute***:* The attribute (or combination of attributes) that is unique for every entity instance
  – e.g.: the account number of an account, the employee id of an employee etc.

Attributes Types

Types of Attributes	Definition	Example
Simple attribute	Cannot be divided into simpler components	**Gender** of the employee
Composite attribute	Can be split into components	**Address** of the employee
Single valued	Can take on only a single value for each entity instance	**Age** of the employee
Multi-valued	Can take up many values	**Skill set** of the employee
Stored Attribute	Attribute that need to be stored permanently	**Date of joining** of the employee
Derived Attribute	Attribute that can be calculated based on other attributes.	**Years of service** of the employee

Degree of a Relationship

- **Degree**: the number of entity types involved
  - One *Unary*
  - Two *Binary*
  - Three *Ternary*

e.g.:

1. *employee **manager-of** employee is unary*

2. *employee **works-for** department is binary*

3. *Customer purchases items from a shop keeper*
   - *Here customer **purchase** item, shop keeper is a ternary relationship*

Cardinality

- Relationships can have different *connectivity*
  - **one-to-one** (1:1)
  - **one-to-many** (1:N)
  - **many-to-one** (M:1)
  - **many-to-many** (M:N)

  e.g.:

  Employee **head-of** department (1:1)

  Lecturer **offers** course (1:N) assuming a course is taught by a single lecturer

  Student **enrolls** course (M:N)

Relationship Participation

- **Total** : Every entity instance must be connected through the relationship to another instance of the other participating entity types

- **Partial***:* All instances need not participate

e.g.: Employee **Head-of** Department

Employee: partial

Department: total

ER MODELING - NOTATIONS

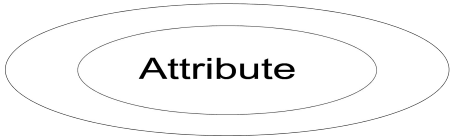
ER Modeling -Notations

An Entity can be defined as an object or concept about which user wants to store information.

A weak Entity requires another Entity for it's existence. Example Order Item depends upon Order Number for its existence. Without Order Number it is impossible to identify Order Item uniquely.

Properties or characteristics of an Entity is called Attributes of entity

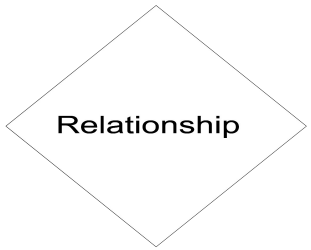
If an attribute is the unique or distinguishing characteristic of the Entity it is called Key Attribute



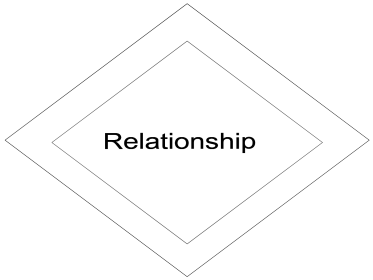
If an attribute can have more than one value then it is called multi-valued attribute.
For example, an employee Entity can have multiple skill values.

ER Modeling -Notations

If the value of an attribute can be derived from another attribute it is called derived attribute. For example, an employee's monthly salary is based on the employee's basic salary and House rent allowance.

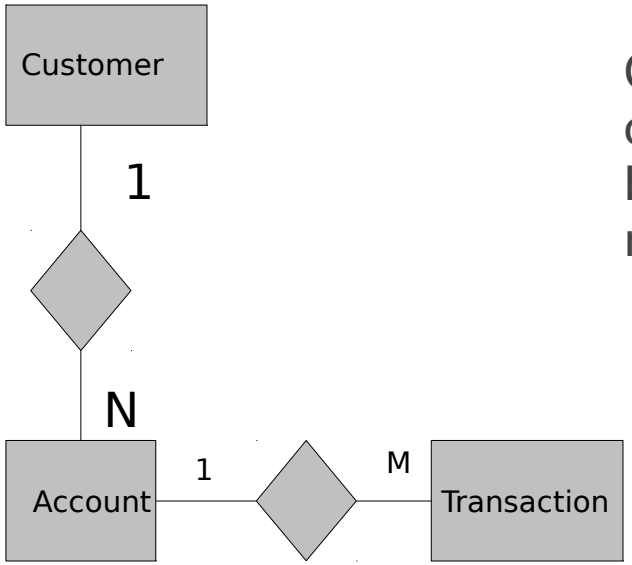


Relationships in ER Diagram illustrate how two entities of database share information.

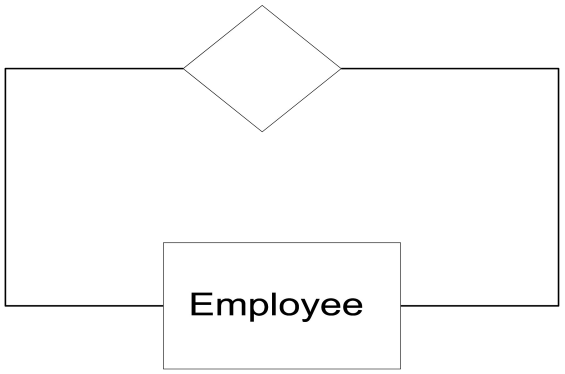


We connect a weak entity through a strong entity using a weak relationship notation.

ER Modeling -Notations

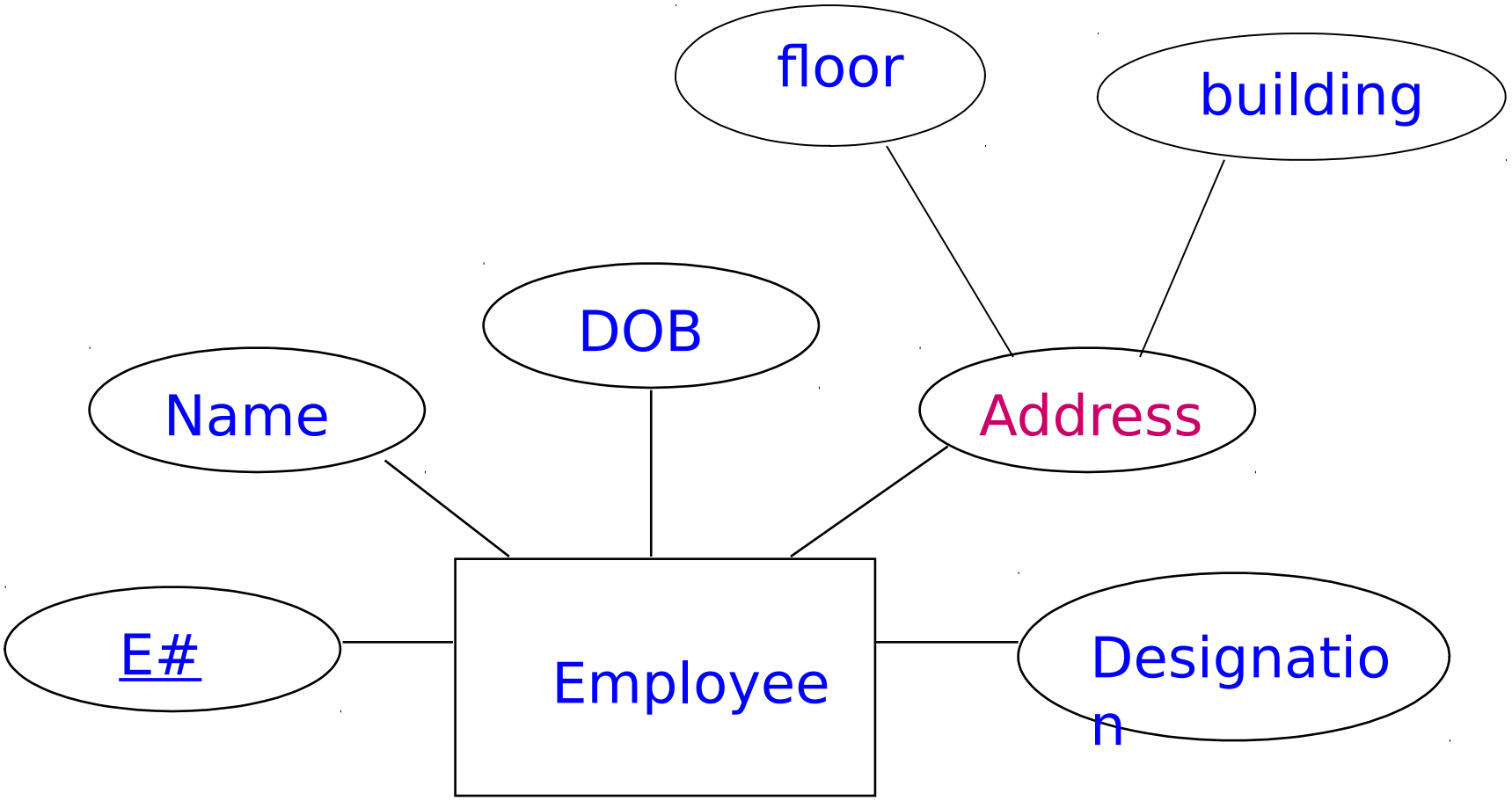


Cardinality of relationship tells how many instances of an Entity type is relate to one instance of another Entity Type. M,N both represent ‘MANY’ and 1 represents ‘ONE’ Cardinality

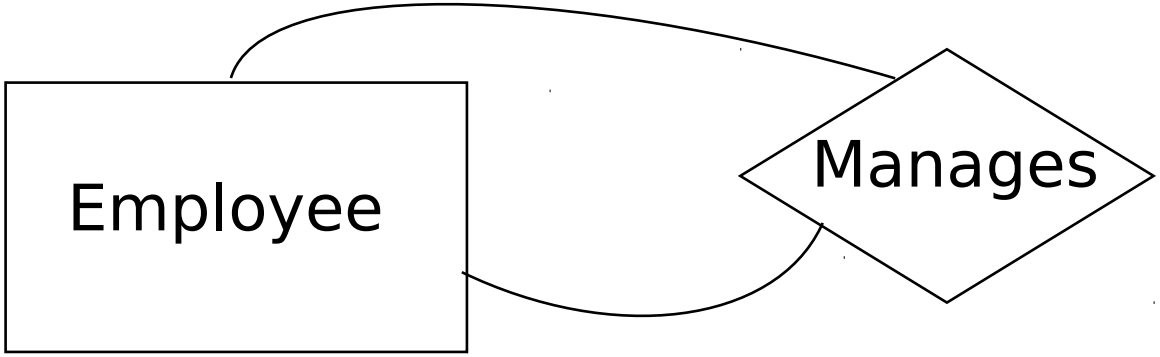


An entity can be self linked.
For example, employees can supervise other employees

Composite attribute

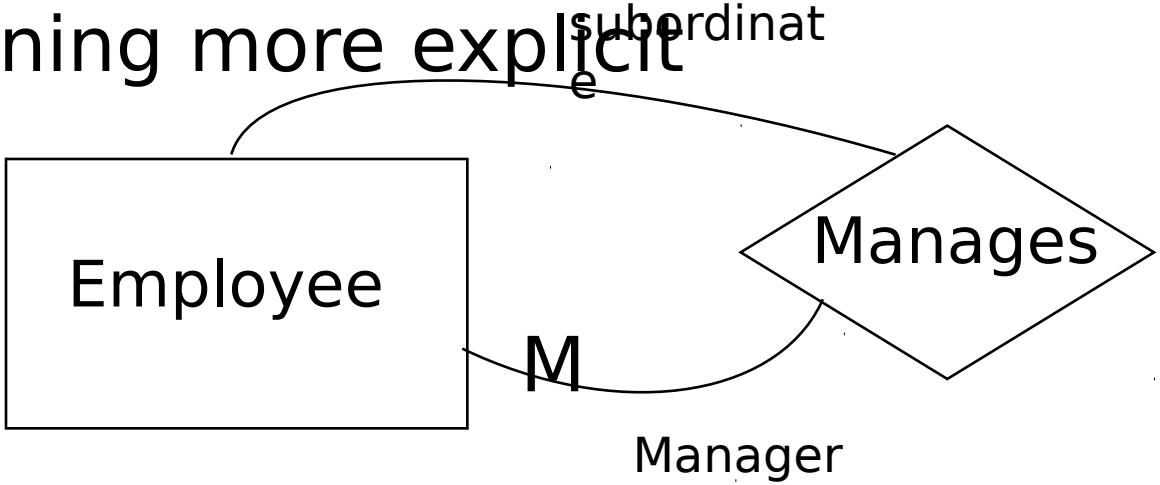


Unary Relationship

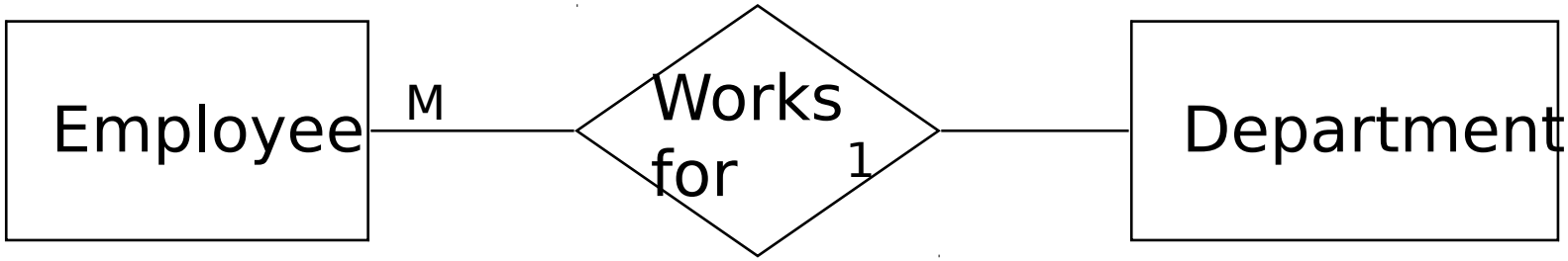


Role names

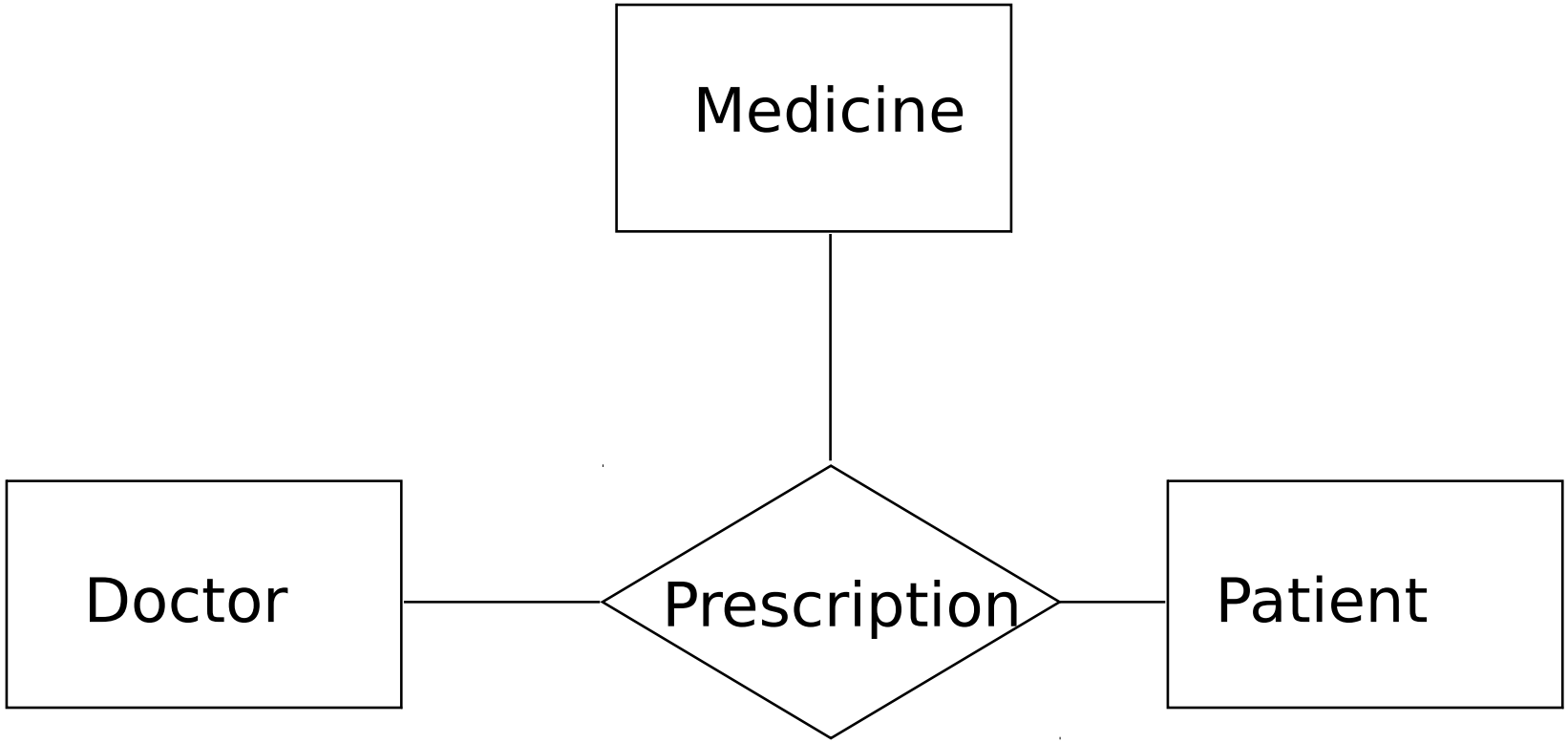
- Role names may be added to make the meaning more explicit



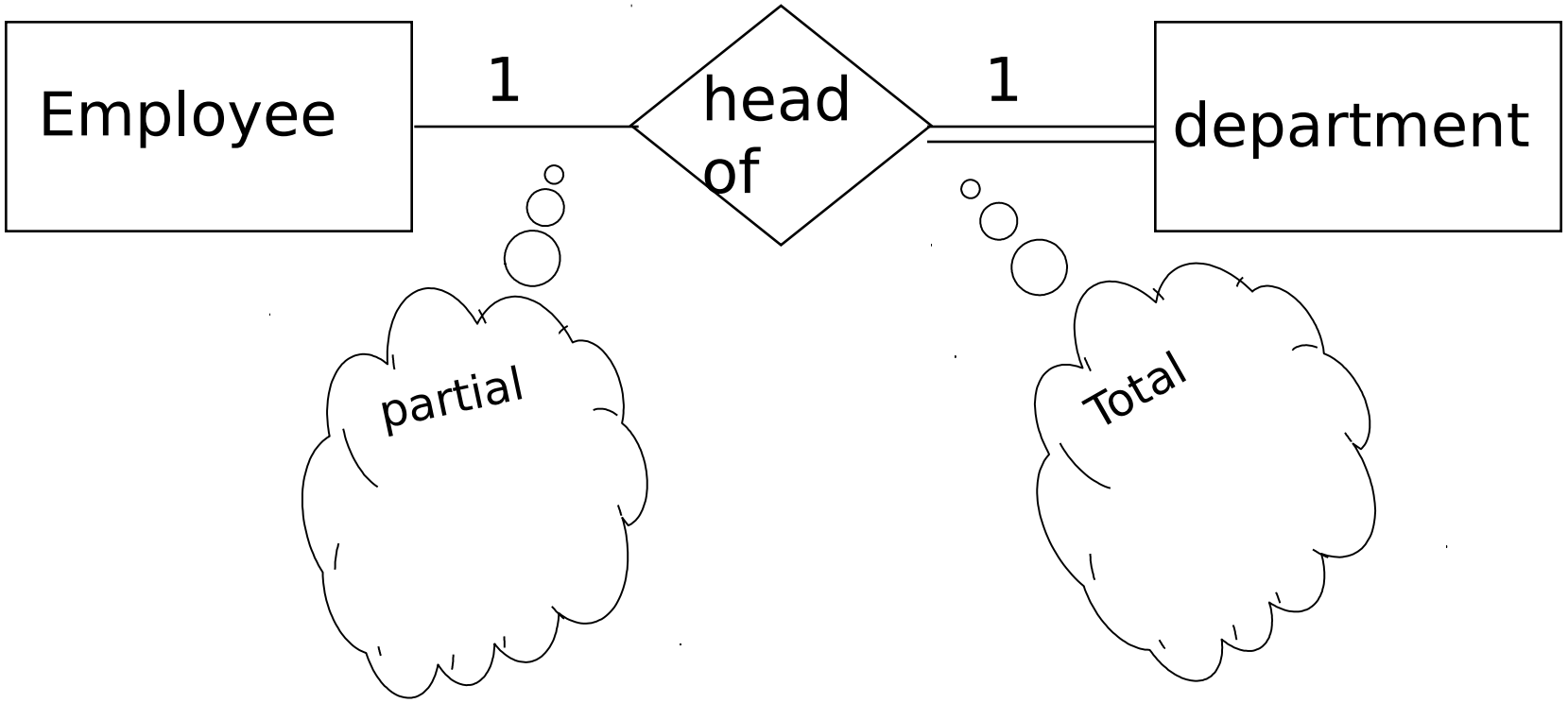
Binary Relationship



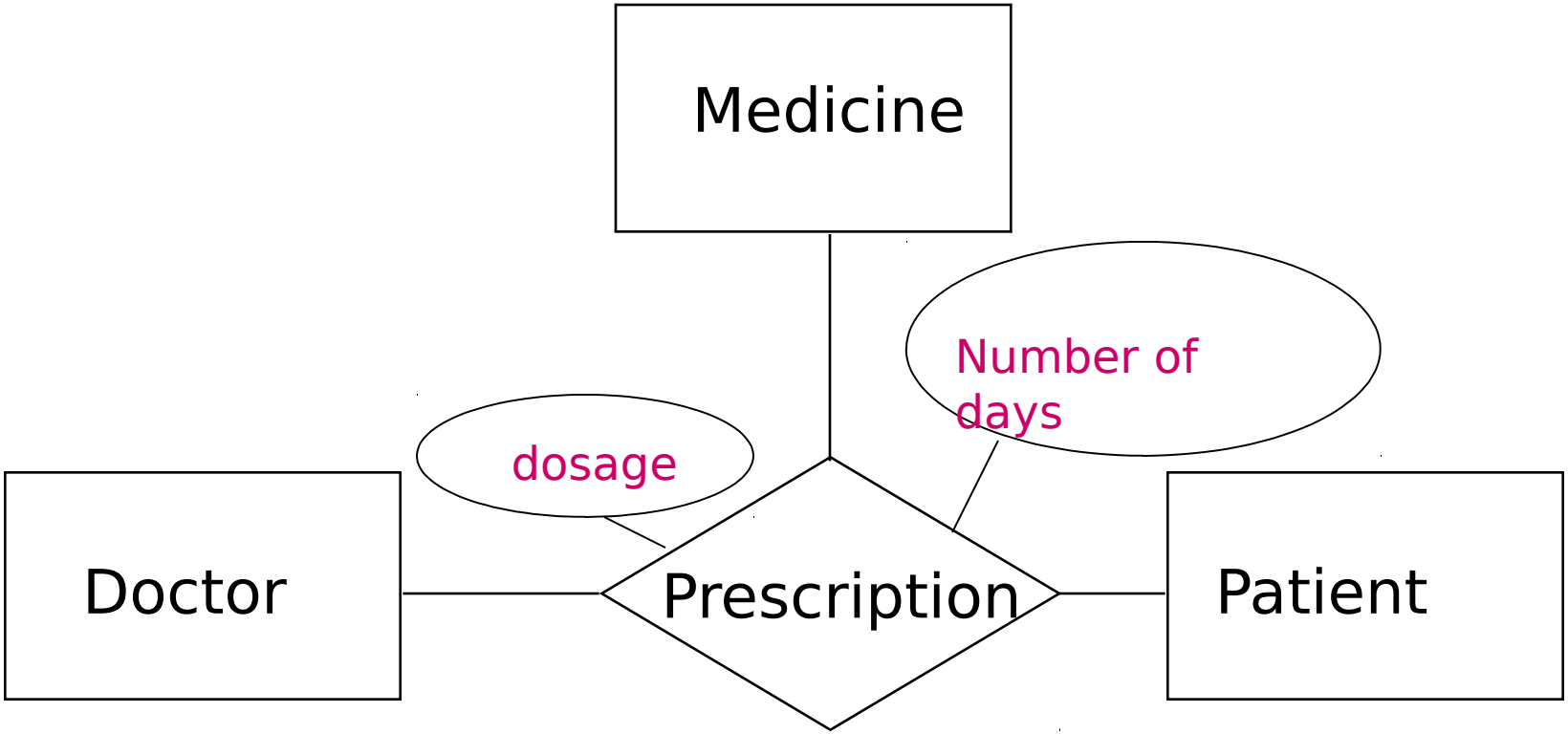
Ternary Relationship



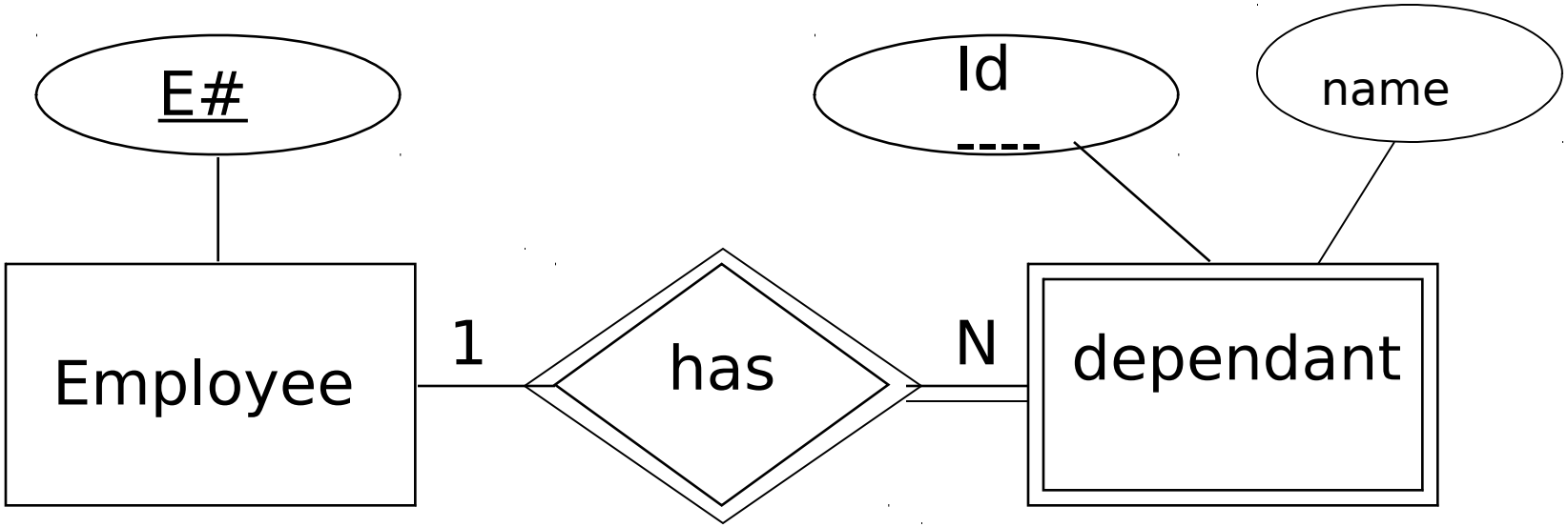
Relationship participation



Attributes of a Relationship



Weak entity



The dependant entity is represented by a double lined rectangle and the identifying relationship by a double lined diamond

Case Study – ER Model For a college DB

Assumptions :

- A college contains many departments
- Each department can offer any number of courses
- Many instructors can work in a department
- An instructor can work only in one department
- For each department there is a Head
- An instructor can be head of only one department
- Each instructor can take any number of courses
- A course can be taken by only one instructor
- A student can enroll for any number of courses
- Each course can have any number of students

Steps in ER Modeling

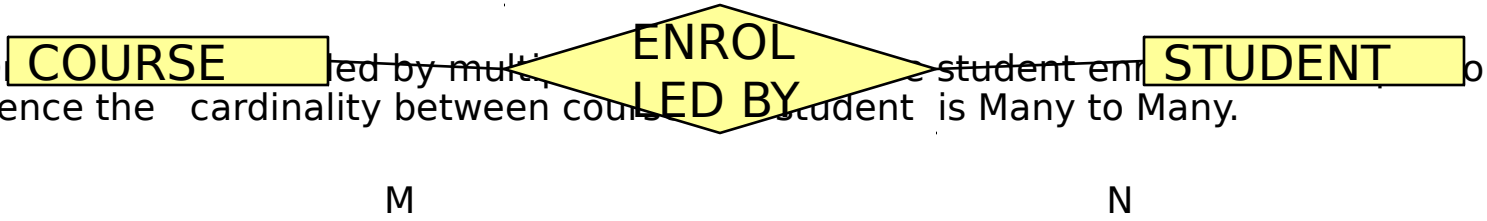
- Identify the Entities
- Find relationships
- Identify the key attributes for every Entity
- Identify other relevant attributes
- Draw complete E-R diagram with all attributes including Primary Key
- Review your results with your Business users

Step 1: Identify the Entities

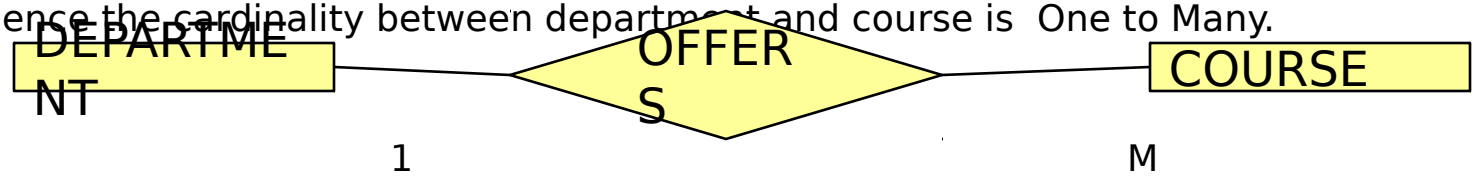
- DEPARTMENT
- STUDENT
- COURSE
- INSTRUCTOR

Step 2: Find the relationships

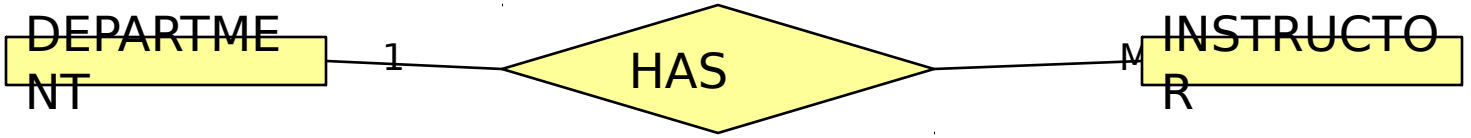
- O[illegible]ed by mult[illegible] student enr[illegible]ourses, hence the cardinality between cours[illegible]student is Many to Many.



- The department offers many courses and each course belongs to only one department, hence the cardinality between department and course is One to Many.

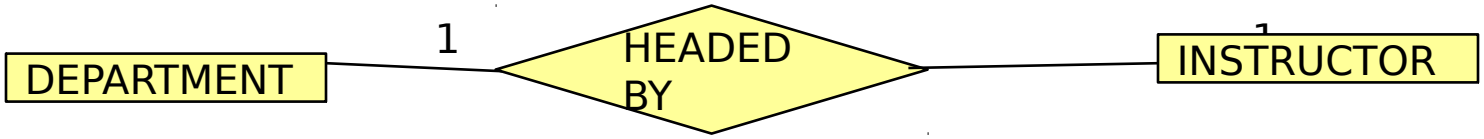


- One department has multiple instructors and one instructor belongs to one and only one department , hence the cardinality between department and instructor is one to Many.



Step 2: Find the relationships(Cont..)

- Each department there is a “Head of department” and one instructor is “Head of department “,hence the cardinality is one to one .



- One course is taught by only one instructor, but the instructor teaches many courses, hence the cardinality between course and instructor is many to one.

Step 3: Identify the key attributes

- Deptname is the key attribute for the Entity “Department”, as it identifies the Department uniquely.
- Course# (CourseId) is the key attribute for “Course” Entity.
- Student# (Student Number) is the key attribute for “Student” Entity.
- Instructor Name is the key attribute for “Instructor” Entity.

Step 4: Identify other relevant attributes

- For the department entity, the relevant attribute is location
- For course entity, course name, duration,prerequisite
- For instructor entity, room#, telephone#
- For student entity, student name, date of birth

Case Study – Online Retail Application(self study)

- Draw an ER diagram of Online Retail Application which allows customer to purchase items from a Retail shop.
- A customer can register to purchase an item. The customer will provide bank account number and bank name (the customer may have multiple account no).
- After registration each customer will have unique customer Id, user id and password.
- Customer can purchase one or more items in different quantities . The items can be of different classes based on their prices.
- Based on the quantity , price of item and discount(if any) on the purchased items, the bill will be generated. A bank account number is required to settle the bill.
- The application also mentions the information of suppliers who supply the items to the retail shop. The retail shop may give orders to supply the items based on some statistics they maintain about different items.

Steps in ER Modeling (self study)

Step 1: Identify the Entities

- CUSTOMER
- ITEM
- SUPPLIER
- BILL

Steps in ER Modeling(self study)

Step 2: Find the relationships

- Customer can purchase an item and each purchase will be corresponding to a bill. So it is a ternary relation ship.
- Items can be ordered to one or more suppliers. One supplier may take order of many items. So **many to many** relationship between **item** and **supplier**.
- One customer can pay many bill and one bill can be paid by only one customer. So **one to many** relation ship between **customer** and **bill**.

Steps in ER Modeling(self study)

Step 3: Identify the key attributes

- Customer entity will be identified by CustomerId
- Item entity will be identified by ItemId
- Supplier entity will be identified by SupplierId
- Bill entity will be identified by BillId

Steps in ER Modeling(self study)

Step 4: Identify other relevant attributes of Entities and Relationships

- For **Customer** entity the relevant attributes will be
  (CustomerId,CustomerName, DateOfRegistration, UserId,
  Password, AccountNo)

- For **Item** entity the relevant attributes will be
  (ItemId, ItemName, UnitOfMeasurement, UnitPrice, Discount,
  QuantityOnHand, SupplierId,ReOrderLevel,ReOrderQuantity,Class)

- For **Supplier** entity the relevant attributes will be
  (SupplierID, SupplierName, SupplierContactNo)

- For **Bill** entity the relevant attributes will be
  (BillId, AccountNo, BillAmount, BillDate)

Steps in ER Modeling(self study)

Step 4: Identify other relevant attributes of entities and Relationships (Cont..)

- For **Purchase** Relation the relevant attributes will be
  (QtyPurchased, NetPrice)

- For **OrderedTo** relation the relevant attributes will be
  (OtyOfOrder, OrderDate, DeliveryDate, DeliveryStatus)

- For **Pays** relation the relivent attributes will be
  (AccountNo)

Merits and Demerits of ER Modeling

Merits

- Easy to understand. Represented in Business Users Language. Can be understood by non-technical specialist.
- *Intuitive* and helps in Physical Database creation.
- Can help in database design.
- Gives a higher level description of the system.

Demerits

- Physical design derived from E-R Model may have some amount of redundancy which may lead to inconsistency.

- Sometime diagrams may lead to misinterpretations because of limited information present in the diagram.

Summary of ER Modeling

- Miscommunication between the application user and the designer is the major source of error.
- It is always better to represent business findings in terms of picture to avoid miscommunication
- It is practically impossible to review the complete requirement document by business users.
- An E-R diagram is one of the many ways to represent business findings in pictorial format.
- E-R Modeling will also help the database design
- E-R modeling has some amount of inconsistency and anomalies associated with it.