Database Management System

Part-2

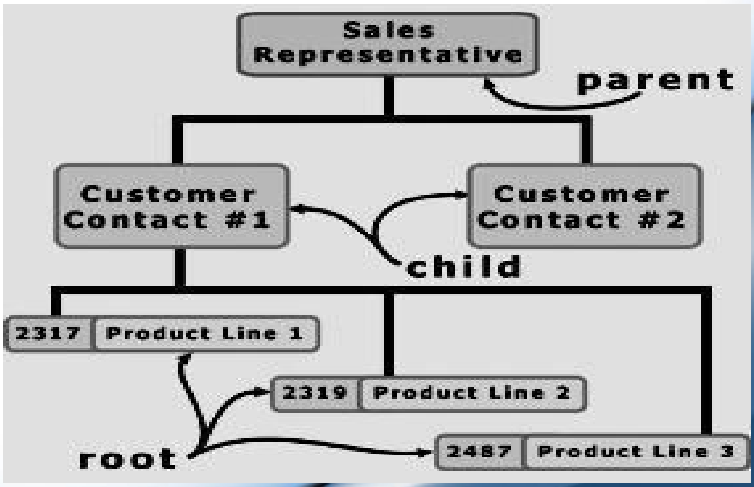
- In 1991 Sun's Green Team started the development of a new programming language which was loosely based on C++. The language was named Oak after the trees outside the office window of the language designer - James Gosling.
- In 1992 Sun turned Green Team into a fully owned company, called First Person Inc. National Center for Supercomputing introduced Mosaic in 1993, a WWW browser, and the Internet began to bustle with traffic. Soon other WWW browser followed.
- In 1994 First Person built an Oak-ready browser called WebRunner and Sun backed the decision to give the language (Oak) away for free, but first Oak was renamed to Java and WebRunner to HotJava. Java became available to millions of people due to Netscapes bundling of Java, and soon others followed (Bank 1995).

4.3 TYPES OF DATA MODEL

There are four different types of data models

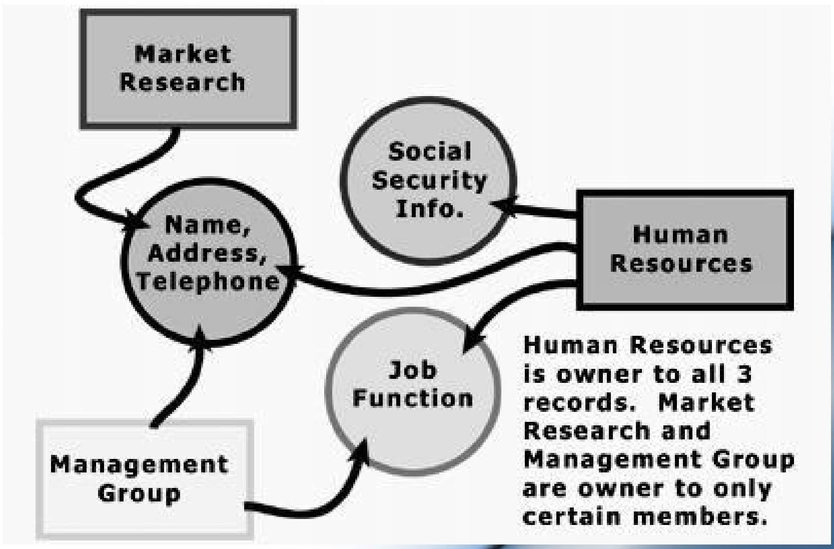
4.3.1 Hierarchical databases
4.3.2 Network databases
4.3.3 Relational databases
4.3.4 Object oriented databases

4.3.1 Hierarchical databases



- **Hierarchical Databases** is most commonly used with mainframe systems.
- It is one of the oldest methods of organizing and storing data and it is still used by some organizations for making travel reservations.
- A hierarchical database is organized in pyramid fashion, like the branches of a tree extending downwards.
- In this model, related fields or records are grouped together so that there are higher-level records and lower-level records, just like the parents in a family tree sit above the subordinated children.
- Based on this analogy, the parent record at the top of the pyramid is called the **root record**.
- A child record always has only one parent record to which it is linked, just like in a normal family tree.
- In contrast, a parent record may have more than one child record linked to it. Hierarchical databases work by moving from the top down.
- A record search is conducted by starting at the top of the pyramid and working down through the tree from parent to child until the appropriate child record is found. Furthermore, each child can also be a parent with children underneath it.

4.3.2Network databases


Market Research
Social Security Info.
Name, Address, Telephone
Human Resources
Job Function
Management Group
Human Resources is owner to all 3 records. Market Research and Management Group are owner to only certain members.

- **Network databases** are similar to hierarchical databases by also having a hierarchical structure. There are a few key differences, however.

- Instead of looking like an upside-down tree, a network database looks more like a cobweb or interconnected network of records. In network databases, children are called **members** and parents are called **owners**.

The most important difference is that each child or member can have more than one parent (or owner).

- Similar to hierarchical databases, network databases are principally used on mainframe computers.
- Since more connections can be made between different types of data, network databases are considered more flexible. However, two limitations must be considered when using this kind of database.
- Similar to hierarchical databases, network databases must be defined in advance. There is also a limit to the number of connections that can be made between records.

4.3.3 Relational databases

- Pre-relational models depended upon being able to determine explicitly where and how individual records were stored.

- Early relational proponents argued that the relational data model viewed information *logically* rather than *physically*, but this is not quite correct.

- Earlier data models associated the logical and physical aspects of information together; logically-related information was stored in physical proximity within a data file. The relational data model first separated the logical from the physical aspects.

- The relational data model looks at information as an *unordered* collection of "relations."

- Each relation is populated with *unordered* "tuples" of the same *unordered* "field" structure.

- Fields may only contain values of a well-defined ("atomic") domain or the *null* value. The unordered aspect needs to be emphasized. For expository purposes, relations are often viewed as "tables".

- The tuples constitute the "rows" of the table; values for a specific field constitute "columns". However, the "table data model" tends to impose a very non-relational ordering on both tuples and fields. Relations are an abstraction of how data is stored; tables are just one of many possible implementations.

- Some of the relational terms are crafted to emphasize the distinction between logical and physical features, to avoid confusing one concept with another. However, vocabulary leakage from other disciplines has sprinkled into the conversation of relational proponents.
- There is a strong tendency to refer to an individual tuple/row as a "record" because collections of fields in other models are called records. "Attribute" is often used synonymously with field.
- To be sure, "unordered" implies neither "chaotic" nor "random". Relations and Fields are named uniquely and identified easily. Distinguishing between tuples is more subtle since the order is not pre-defined.
- Rather than depending upon relative (as in hierarchy) or absolute (as in network) locations, tuples may only be differentiated according to their contents.
- Consequently, duplicate tuples are not permitted within a single relation. Even more strongly, distinct tuples must have a unique "key" (some combination of a relation's named fields).
- The set of minimal keys includes one "primary key"; the rest are "candidate keys". Within a tuple, references to other tuples are expressed as a "foreign key," which should contain the values of the referenced tuple's primary key.
- Relational theory provides a firm mathematical foundation for data management. Set theory could be applied to relations using relational algebraic operations (union, intersection, join, projection, etc.).
- Assertions about the existence or non-existence of some condition with a data base could be proven with a rigor unachievable with earlier models.

4.3.4 Object oriented databases

- A data model is a logic organization of the real world objects (entities), constraints on them, and the relationships among objects. A DB language is a concrete syntax for a data model. A DB system implements a data model.

o A core object-oriented data model consists of the following basic object-oriented concepts:

(1) **object and object identifier**: Any real world entity is uniformly modeled as an object (associated with a unique id: used to pinpoint an object to retrieve).

(2) **attributes and methods**: Here every object has a state (the set of values for the attributes of the object) and a behavior (the set of methods - program code - which operate on the state of the object). The state and behavior encapsulated in an object are accessed or invoked from outside the object only through explicit message passing.

An attribute is an instance variable, whose domain may be any class: user-defined or primitive. A class composition hierarchy (aggregation relationship) is orthogonal to the concept of a class hierarchy. The link in a class composition hierarchy may form cycles.

(3) **class**: a means of grouping all the objects which share the same set of attributes and methods. An object must belong to only one class as an instance of that class (instance-of relationship). A class is similar to an abstract data type. A class may also be primitive (no attributes), e.g., integer, string, Boolean.

(4) **Class hierarchy and inheritance**: derive a new class (subclass) from an existing class (superclass). The subclass inherits all the attributes and methods of the existing class and may have additional attributes and methods. single inheritance (class hierarchy) vs. multiple inheritance (class lattice).

4.4 ADVANTAGES AND DISADVANTAGES OF DATA MODELS

Advantages

1.Simplicity: Since the database is based on the hierarchical structure, the relationship between the various layers is logically simple.

2.Data Security: Hierarchical model was the first database model that offered the data security that is provided by the DBMS.

3.Data Integrity: Since it is based on the parent child relationship, there is always a link between the parent segment and the child segment under it.

4. **Efficiency**: It is very efficient because when the database contains a large number of 1:N relationship and when the user require large number of transaction.

Disadvantages

1.Implementation complexity: Although it is simple and easy to design, it is quite complex to implement.

2.Database Management Problem: If you make any changes in the database structure, then you need to make changes in the entire application program that access the database.

3.Lack of Structural Independence: there is lack of structural independence because when we change the structure then it becomes compulsory to change the application too.

4.Operational Anomalies: Hierarchical model suffers from the insert, delete and update anomalies, also retrieval operation is difficult.

4.4.2 Network Model

Advantages

1. Conceptual Simplicity: just like hierarchical model it also simple and easy to implement.

2. **Capability to handle more relationship types**: the network model can handle one to one1:1 and many to many N: N relationship.

3. **Ease to access data:** the data access is easier than the hierarchical model.

4.Data Integrity: Since it is based on the parent child relationship, there is always a link between the parent segment and the child segment under it.

5.Data Independence: The network model is better than hierarchical model in case of data independence.

Disadvantages

1. System Complexity: All the records have to maintain using pointers thus the database structure becomes more complex.

2.Operational Anomalies: As discussed earlier in network model large number of pointers is required so insertion, deletion and updating more complex.

3.Absence of structural Independence: there is lack of structural independence because when we change the structure then it becomes compulsory to change the application too.

4.4.3 Relational Model

Advantages

1.Conceptual Simplicity: We have seen that both the hierarchical and network models are conceptually simple, but relational model is simpler than both of those two.

2.Structural Independence: In the Relational model, changes in the structure do not affect the data access.

3.Design Implementation: the relational model achieves both data independence and structural independence.

4.Ad hoc query capability: the presence of very powerful, flexible and easy to use capability is one of the main reason for the immense popularity of the relational database model.

Disadvantages

1.Hardware overheads: The relational database systems hide the implementation complexities and the physical data storage details from the user. For doing this, the relational database system need more powerful hardware computers and data storage devices.

2.Ease of design can lead to bad design: The relational database is easy to design and use. The user needs not to know the complexities of the data storage. This ease of design and use can lead to the development and implementation of the very poorly designed database management system.

4.5 BUSINESS RULES

- Business rules are the rules that are created to affect the way your business works. Usually, these are rules that involve employees or staff and are rules that specify what they can and cannot do.
- A great example of a business rule involves marriages. For many companies, a boss is not allowed to marry an employee or an accountant at a company is usually not allowed to marry another accountant.

- In this case, the accountants are not allowed to be married because there is a more likely chance that the spouses can change financial information and then cover for one another.
- These rules are intended to prevent disruption in a company or business.
- Business Rules are used every day to define entities, attributes, relationships and constraints.
- Usually though they are used for the organization that stores or uses data to be an explanation of a policy, procedure, or principle.
- The data can be considered significant only after business rules are defined, without them it's just records, but to a business they are the characteristics that are defined and seen by the company.
- Business Rules help employees focus on and implement the actions within the organizations environment.
- Some things to think about when creating business rules are to keep them simple, easy to understand, keep them broad so that everyone can have a similar interpretation. To be considered true, business rules must be in writing and kept up to date.
- Identifying business rules are very important to the database design. Business rules allow the creator to develop relationship participation rules and constraints and to create a correct data model.
- They also allow the creators to understand business processes, and the nature, role and scope of the data.
- They are a communication tool between users and creators, and they also help standardize the company's view of the data.
- It is important to keep in mind that some business rules cannot be modeled.
- Business Rules give the proper classification of entities, attributes, relationships, and constraints.
- Sources of business rules are managers, policy makers, department managers, written documentation, procedures, standards, operation manuals, and interviews with end users.

Some examples of business rules:

Departments------offers---------Course
Course----------generates---------Class
Professor --------teaches----------Class

- There are several protocols to the way business rules are written. Not every protocol has to be followed, but in general, a well-written set of business rules consist of having a unique identifier, describes one and only one concept, are written in plain language, are written, and are from a single source.

- In terms of a unique identifier, business rules should come with an identifier that may consist of the rule number and the department it affects. And example would be ‘BRacc01’. In this case, this business rule (BR) is directly related to the accounting department.
- Another important aspect of business rules consist of how the rules are shared within the company.
- A protocol for business rules that many follow is that the business rules are written down. However, with many businesses sharing information directly over the internet, some are opting to place their business rules online in company blogs, wikis, and websites.
- This shares the business rules with all employees faster and easier. In relation to how business rules are shred, it is very important that business rules are written in plain language.
- If business rules are written at a high level language, there is an increased chance that not every person will understand what the business rules cover or what is acceptable and what is not.

4.6 SUMMARY

- A *data model* is a picture or description which shows how the data is to be arranged to achieve a given task.
- The data structures and access techniques provided by a particular DBMS are called its data model.
- In 1964 the first commercial database management system (DBMS) was developed widely known as Integrated Data Store (IDS).
- A hierarchical database is organized in pyramid fashion, like the branches of a tree extending downwards.
- In hierarchical model, the parent record at the top of the pyramid is called the **root record** and the leaf node is called the **child record.**
- **Network databases** are similar to hierarchical databases by also having a hierarchical structure.
- The relational model organizes the records and stores the records in rows and columns.

7. REVIEW QUESTIONS

1) Explain the need for the data model.
2) Write in detail about the history of data model.
3) Write a short notes on

    a. Hierarchical Model
    b. Network Model
    c. Object Oriented Model
    d. Relational Model

4) Explain the merit and demerits of hierarchical model.
5) Explain the merit and demerits of network model
6) Explain the merit and demerits of Relational model.

❖❖❖❖

5

Unit Structure

5.0 OBJECTIVES

The database design process consists of a number of steps listed below. We will focus mainly on step 2, the conceptual database design, and the models used during this step.

Step 1: Requirements Collection and Analysis

- Prospective users are interviewed to understand and document data requirements
- This step results in a concise set of user requirements, which should be detailed and complete.
- The functional requirements should be specified, as well as the data requirements. Functional requirements consist of user operations that will be applied to the database, including retrievals and updates.

- Functional requirements can be documented using diagrams such as sequence diagrams, data flow diagrams, scenarios, etc.

Step 2: Conceptual Design

- Once the requirements are collected and analyzed, the designers go about creating the conceptual schema.
- Conceptual schema: concise description of data requirements of the users, and includes a detailed description of the entity types, relationships and constraints.
- The concepts do not include implementation details; therefore the end users easily understand them, and they can be used as a communication tool.
- The conceptual schema is used to ensure all user requirements are met, and they do not conflict.

Step 3: Database Implementation

- Many DBMS systems use an implementation data model, so the conceptual schema is transformed from the high-level data model into the implementation data model.
- This step is called logical design or data model mapping, which results in the implementation data model of the DBMS.

Step 4: Physical Design

- Internal storage structures, indexes, access paths and file organizations are specified.
  Application programs are designed and implemented

ER Model

In software Engineering, an entity relational model is an abstract and conceptual representation of dataEntity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often arelational database, and its requirements in a top-down fashion. Diagrams created by this process are called **entity-relationship diagrams**, **ER diagrams**, or **ERDs**.

In 1976, Entity relationship model developed by Chen,

ER Model is high level Conceptual model which used Conceptual design of database where as relational model are used to logical design of database

ER Diagram

- A database can be modeled as

  A collection of entities

  Relationship among the entities
- An entity is an real world object that exist and it is distinguishable from other entities

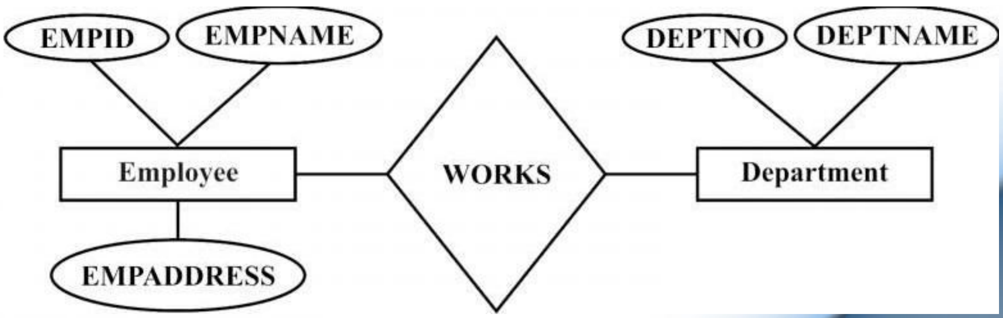
  Example Person, company, event, plant
- All the entities in the data model have attributes as known as properties of an entities

  Example: people have names and addresses

An Entity set is a set of an entities of all same type that share the same properties.

Example: set of all persons ,companies,trees, holidays

ER Diagram



- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Underline indicates primary key attributes
- Ellipses represent an attributes
- Double Lines represent total participation of an entity in a relationship set
- Double rectangle represent a weak entity sets

Strong Entity type

An entity type which has own distinct primary key that used to identify specific uniquely from another entity type is called as Strong Entity type

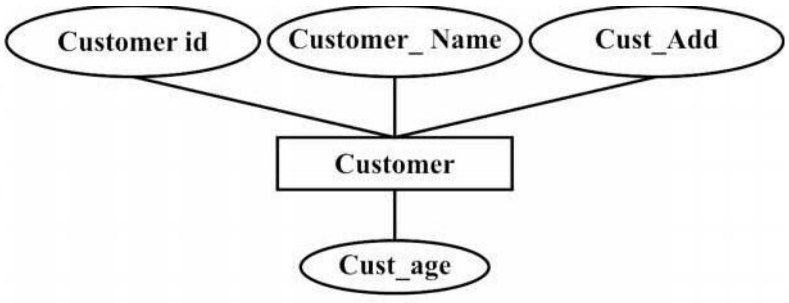
An Entity type which is independent on some other entity type icalled Strong Entity type

Example

In the Case of Client entity Client_no is the primary key of Client entity which is used to uniquely identified among the Client 's entity set

In the case of Customer Entity , Customer_id is the primary key of Customer Entity which is used to uniquely identified among the Customer's entity set

Strong Entity type is represented by rectangle Symbol



Customer Entity

Weak entity Type

Entity type which is dependent on some other entity type is called as Weak entity type

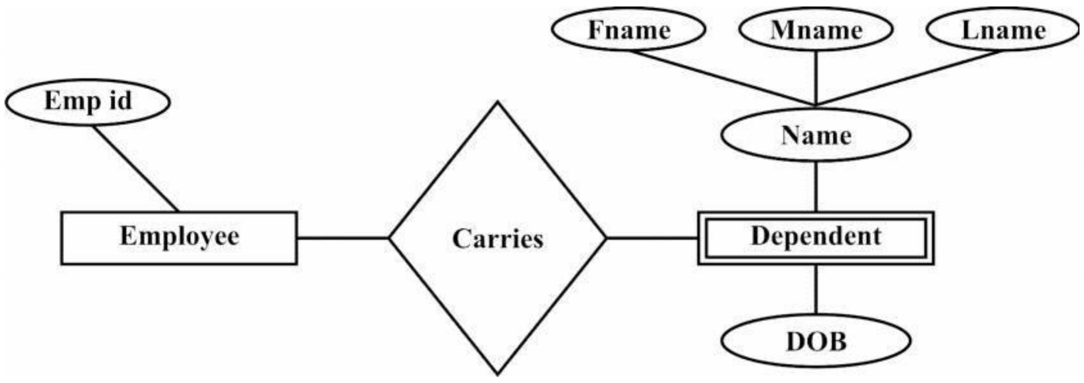
- Weak entity type is dependent on a strong entity and cannot exist on its own
- It does not have a unique identifier that has partial identifier
- Partial identifier is represented by double-line

Some weak entities assign partial identifiers and such partial identifiers of an weak entity called as discriminator

Weak entity type is represented by double rectangle.

Identify relationship

Strong entity type is link with the weak entity type



Dependent entity depend upon Employee entity for primary key

Attributes

Properties of an entity or relationship type is called as attribute Example Staffno, staffname,staff_designation is describes an entity Staff Value of an attribute play a major role of data strored in database Each entity will have the value which is assigned to its attributes Consider an example

Above stated example of Staff Entity which has the attribute named as staffno, the value which is assigned to the staff attribute is '101' and the staffname attribute has the value is 'Mahendra, and staff_desigination attribute has the value is 'Manager'

Attribute domains

The set of allowable values which is assigned to one or more attribute is knowns as Attribute domains

There are types of attributes has been classified Such as simple and Composite type,single valued and multi valued attributes Stored and derived attributes, null attributes and Key attributes

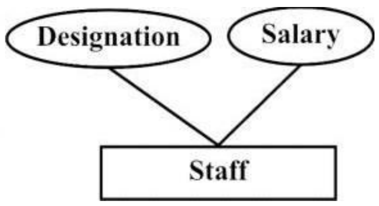
1) Simple Attributes

Simple attributes is an attributes which can further divided in to two parts

Or

An Attribute composed of single compoenent with an independent existence

For an example: Desgination of an staff and Salary of an staff



Simple Attributes

Composite Attribute

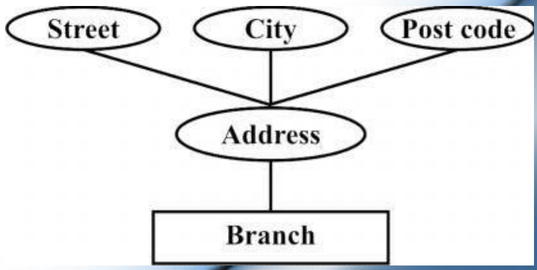
Composite Attribute is an attribute which is futher divided into many parts

Or

An attributed composed of multiple component, each component has its own independent existence

Example

Address attributes of an Branch entity that can be further divided in to sub parts i.e street, city and postalcode as an attributes



Composite Attributes

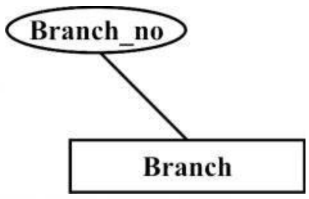
2) Single valued and Mutli Valued attributes

Single valued attribute is an attribute which as single value(atomic) for each entity.

Or

An attribute that holds a single valuefor each occurrence of an entity type

Example: Each branch has only single valued attributes is known as branch_no



Single Valued attributes

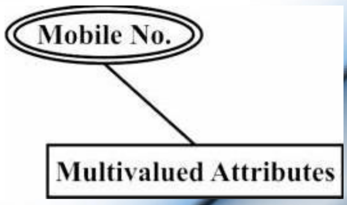
Mutli valued attributes

Mutli valued attribute is an attributes which as many values for each entity

Or

An attribute that holds multiple values for each occurrence of an entity type.

Example : Each staff member has multiple mobile numbers



Multivalued Attributes

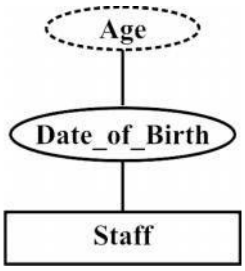
3) Stored and Derived attributes

Stored attributes is an attribute which is used supplied a value to the related attreibute

Example Date_of_Birth of an staff is a stored attributes

Derived attributes

The value from the derived attribute is derived from the stored attribute for an example Date_of_Birth is a stored attribute for an each staff member . The value for an Age can be derived from the Date_of_ Birth attributes I.e by subtracting the Date_of _Birth from the Current date, therefore the Stored attributes is used supplied a value to the related attributes



Null attribute

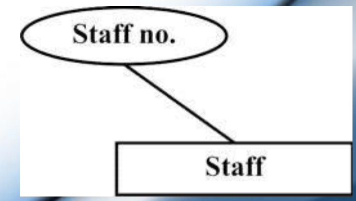
The attribute which take NULL value when entity does not have the value to it.

The Null attribute is an attribute their value is unknown, unassigned and missing information

Key Attributes

This attribute has the unique value for an entity which is used to identified given row in the table is called as key attribute of an entity

Example : Staff_ no is an key attribute which has an unique value which is used to identifies given row in the table

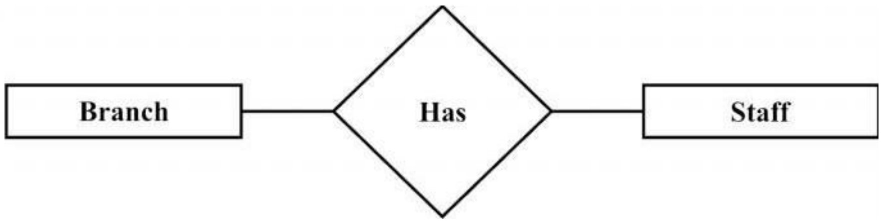


Relationships

A set of meaningful relationship among several entities

We used to inidicate the diamond symbol for Relationships among the several entities, it could read from left to right

Example : Branch has a staff



Degree of relationship

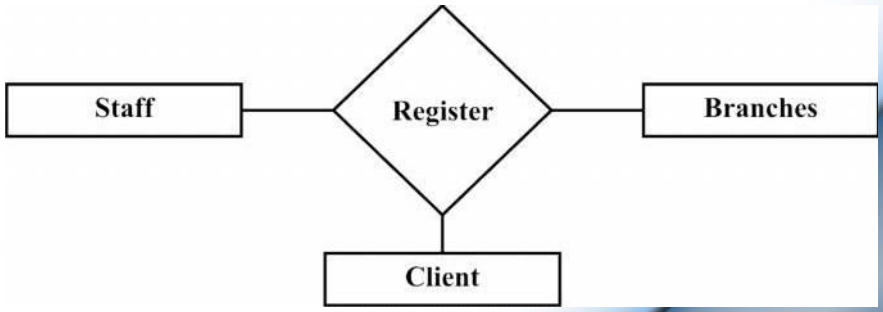
It is the number of entities participated in a particular relational model

There are two type of degree of relationship.

Binary relationship: A Relationship of degree two is called as binary relationship

Ternary Relationship: A relationship of degree three is called as Ternary relationship.

Example



Staff registers a Client at a branch

Relationship set

The collection of similar relationship is known as Relationship set

Constraints on relationship

1) Mapping Constraints / Cardinalities

The number (or range) of possible entity type that is associated to another entity type through a particular entity

Cardinalities indicates that a specific number of entity occurrence of related entity .

Type of Mapping Constraints

One-to-one (1:1)

One-to-many (1:*)

Many- to-one(*:1)

Many-to-many (*:*)

Type**One-to-one (1:1)**

In this type of Mapping Constrant One record of an entity is related to the one record of an another entity

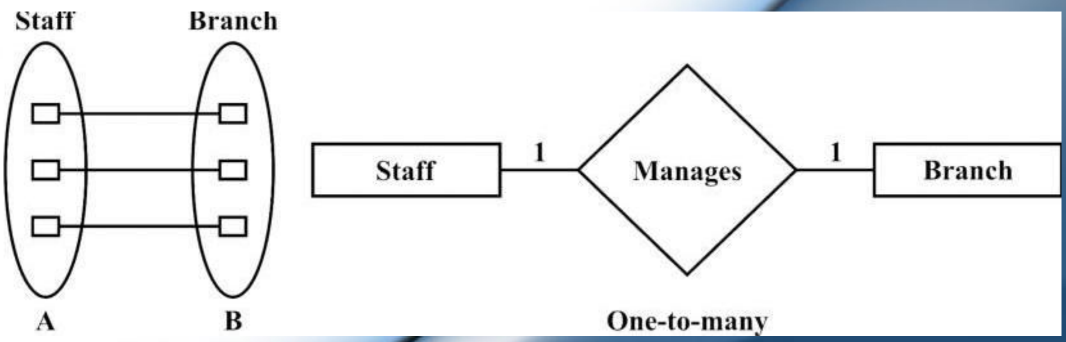
That is one row on an table is related to an one row of another table

i.e A is associated with at most one entity in B and Bis associated with at most one entity in A

Example

Each branch is managed by one member of the staff that’s means Branch Manager

A member of staff can manage zero or one branch



2) One- to- many

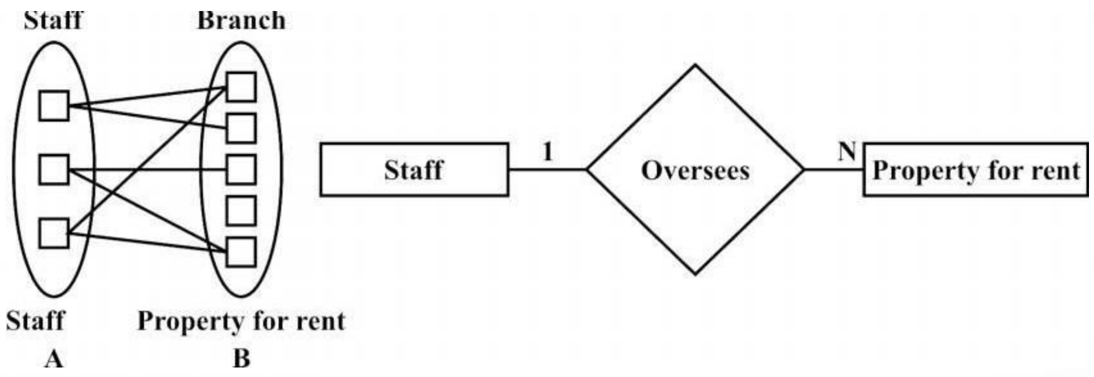
In this constraints, One record in the entity can be related with many record in other entity

A is associated with any number of entities in B

B is associated with at most one entity in A

E.g. each member of staff oversees zero or more prosperity for rent

Every row in the Staff table can have relationship with many rows in the properityforRent Table

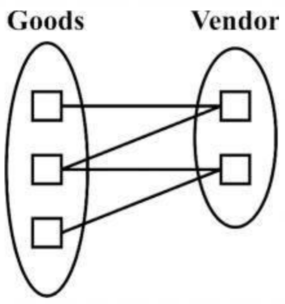


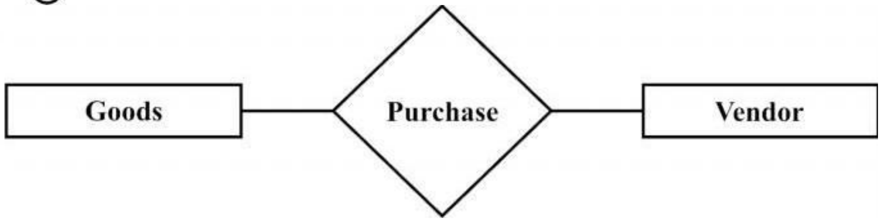
One To Many

In this type Mapping Constraints , Many records in the one enity is related to the only one records in the other entity

An entity in A is associated with at least one entity in B . an entity in B can be associated with any number of entities in A.

Example one vendors has many Goods and Many Goods is purchase by one Vendors



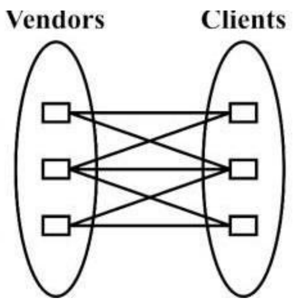


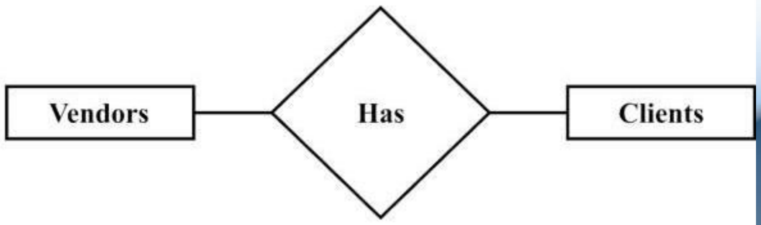
Many to Many

In this Mapping Constraints , Many records in the entity is related Many records in the other entity

An entity in A is associated with any number of entities in B. and an entity in B is associated with any number of entities in A.

Many Vendors Has Clients and Many Clients has may Vendors

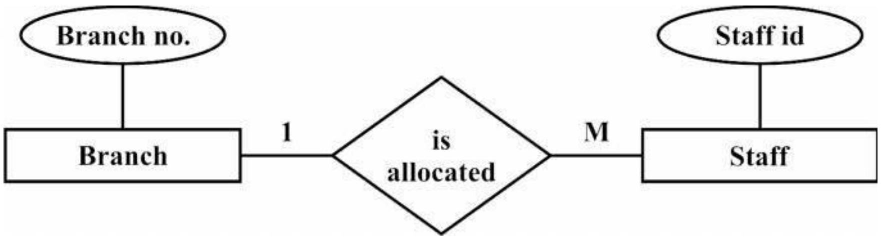




Participation Constraints

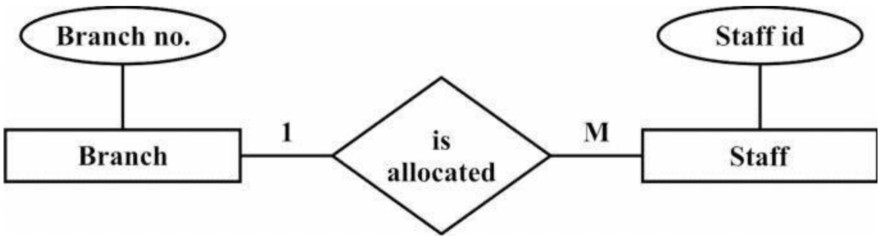
There are two types of participation constraints: Total Participation: Every Instance of the first Entity type must share with on or more instances of the relationship type with the other entity type.

The total participation is represented by a dark line or double line between the relationship and entity



Every Branch office is allocated members of Staff

Partial Participation: There exist an instance of the first entity type that don't share an instance of the relationship type with the other entity type.



A member of Staff need not work at a Branch office

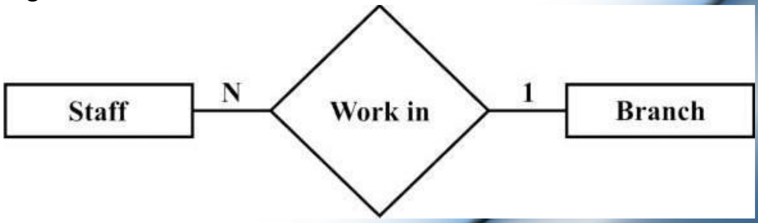
Notations used In ER Diagrams For Representing Relations

1) Cardinality Ratio Notation

In this method ,Cardinality ratio (of a binary relationship): 1:1, 1:N, N:1, or M:N

Shown by placing appropriate numbers on the relationship edges

Eg

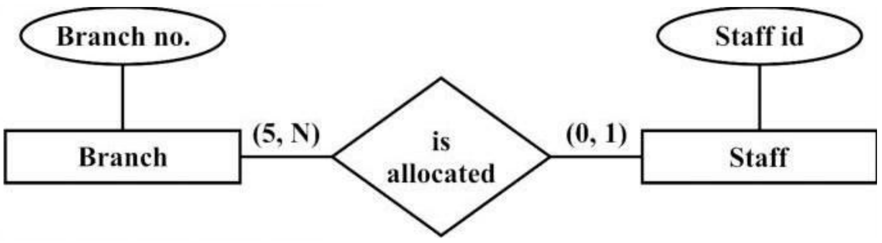


Number of Staffs working in Branch

2) Min –Max notation

The altermate of notation by specify the pair of integer, that used to specify the minimum and maximum participation of each entity type in the form of(min, max)

The Minimum participation of 0 indicate partial participation where as maximum participation of 1 or more indicates total participation



At least 5 staff is allocated to branch

Limitation Of Entity Relationship Model

Problems may arise when designing a conceptual data model called connection traps.
•Often due to a misinterpretation of the meaning of certain relationships.

Extended Entity Relation Relationship Model

Since 1980 there has been increase in the emergence of new database application with more demanding application

Basic concepts of ER modeling are not sufficient to represent the requirement of newer, more complex operation

To overcome the issue of ER modeling there is response in development of additional 'semantic' modeling concept

Semantic concept which is integrated into original ER Model is known as Extended Entity Relation Relationship Model (EER)

Additional Concept which is includes in the Extended Entity Relation Relationship Model are specialization/ generalization, categorization, superclass/subclass, attribute inheritance

Extended EER Model is used the concept of object oriented such as inheritance

Sub classes and Super classes

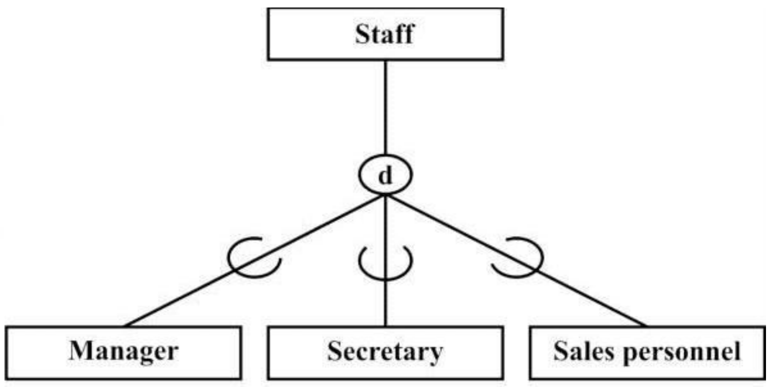
In some case , entity type has numerous sub-grouping of its entities because that are meaningful way for representation and need to be explicitly defined because of their importance

The set listed is a subset of the entities that belong to the staff entity which means that every entity that belongs to one of the sub sets is also an Staff

An entity type that includes distinct Subclasses that require to be represented in a data model is called as super class.

A Subclass is an entity type that has a distinct role and is also a member of the Superclass.

Staff is the super class where as manager, Secretary, sales personnel is the subclass



Superclass / Subclass Relationship

Superclass /Subclass Relationship

Superclass /Subclass Relationship

The relationship between super class and subclass is called Superclass /Subclass Relationship

In Superclass /Subclass Relationship, the encircled 'd' Indicates that there is Superclass /Subclass Relationship, it is denoted by the symbol

Hence Superclass /Subclass Relationship lead to the object oriented Concept is called as Inheritance

As the above diagram, Arc drawn above the line towards Subclass indicated inheritance Relationship

Type Inheritance

- The type of an entity is defined by the attributes it possesses, and the relationship types it participates in.

- Because an entity in a subclass represents the same entity from the super class, it should possess all the values for its attributes, as well as the attributes as a member of the super class.

- This means that an entity that is a member of a subclass inherits all the attributes of the entity as a member of the super class; as well, an entity inherits all the relationships in which the super class participates.

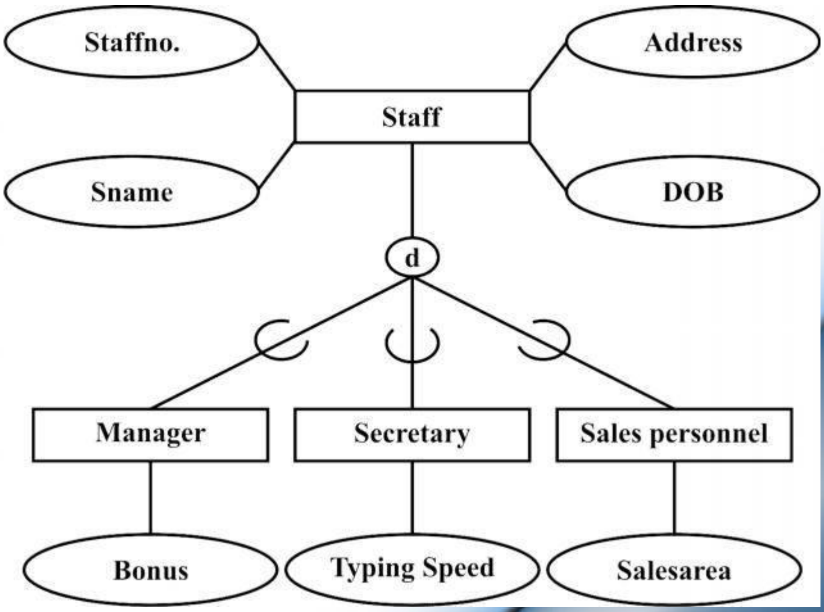
Specialization

The process of defining a set of subclasses of super class

The specia;ization is a top down approach of super class and subclasses

The set of sub classes is based on some distinguishing characteristic of the super class.

.



Notation for Specialization

- To represent a specialization, the subclasses that define a specialization are attached by lines to a circle that represents the specialization, and is connected to the super class.

- The subset symbol (half-circle) is shown on each line connecting a subclass to a super class, indicates the direction of the super class/subclass relationship.
- Attributes that only apply to the sub class are attached to the rectangle representing the subclass. They are called specific attributes.

A sub class can also participate in specific relationship types

Reasons for Specialization

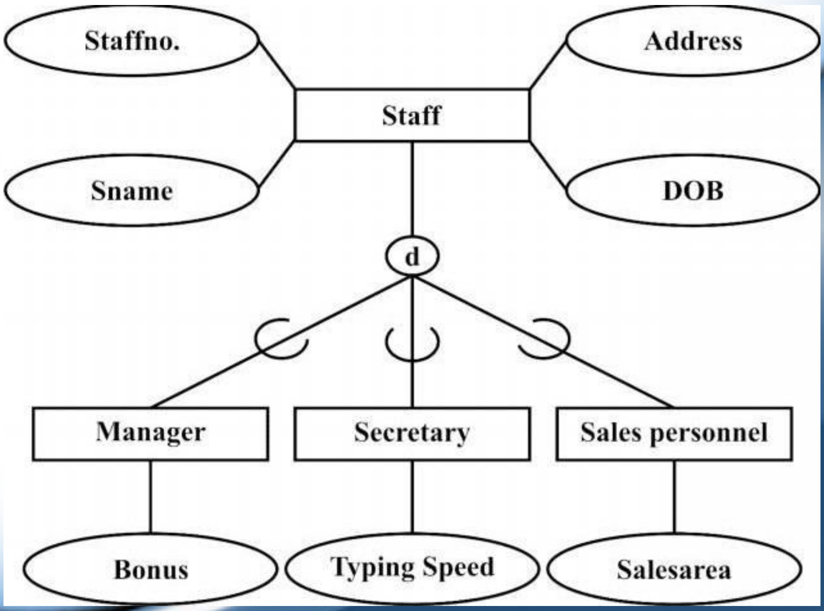
- Certain attributes may apply to some but not all entities of a super class. A subclass is defined in order to group the entities to which the attributes apply.
- The second reason for using subclasses is that some relationship types may be participated in only by entities that are members of the subclass.

Summary of Specialization
Allows for:

- Defining set of subclasses of entity type
- Create additional specific attributes for each sub class
- Create additional specific relationship types between each sub class and other entity types or other subclasses.

Generalization

- Generalization is the reverse of specialization and this is a bottom-up approach
  In Generalization, there are Several classes with common features and generalizing into a super class.



Attribute Inheritance

•An entity in a Subclass may possess subclass specific attributes, as well as those associated with the Superclass

5.6 CODD'S RULE

Dr. E.F Codd was inventor of the relational database model. This model say that whether the Database management system follow the relational model or not and what extents model is relational.

The article mentioned by Dr.E.F.Codd that according to these rule, There is no database management system fully implements all the 12 rules what he has been specified

In 1990, The codd rules extended 12 to 18 rules that's includes catlog,datatypes,authorization etc.

OverView of codd's rule

Sr.NO	Rule	Description
1	The information rule:	All the information in the database should be represented in the term of relational or table.Information should be stored as an values in a tables
2	The guaranteed access rule	All data must be accessible. The Rule say that there is fundamental requirement of primay key for each record in table ,and there should be no ambiguity by stating the table name and its primary key of the each record in the table along with columns name to be acessed
3	Systematic treatment of null values:	Null values could not be treated as blank space or zero values, The null values are known as unknown values, unassigned values should be treated as missing information and inapplicable information that should be treated as systematic , distinct from regular values

4	Active online catalog based on the relational model:	The system must support an online catalog based data dictionary which hold the information or description about the table in the database
5	The comprehensive data sublanguage rule:	The system must support at least one relational language that through which the data in the database must be accessed 1 The language can be used both interactively and within application programs. The Languauge must Supports data definition operations (including view definitions), data manipulation operations (update as well as retrieval), security and integrity constraints, and transaction management operations (begin, commit, and rollback).
6	The view updating rule:	All the view must be theoretically updatable can be updated by the system
7	High-level insert, update, and delete:	This rules states that in the relational model, the structured query language must performed data manipulation such as inserting ,updating and deleting record on sets of rows in the table
8	Physical data independence:	Any change made in the data is physically stored in term of data is stored in the file system through array and link list must not effect application that access the data structure

9	Logical data independence:	This rule state that changes in the logical level(rows ,columns and so on) must not change to the application's structure
10	Integrity independence:	Data integrity constraints should be considered as separated from application program, the structured query language which defines data integrity constraint must be stored in the database in term of data in table that is, in the catalog and not in the application.
11	Distribution independence:	The rules states that the data can be stored centrally on the single machine or it can be stored in the various location(ditributed) on various machine but it should be invisible to the user i.e the user does not location of data is stored whether on the single machine(Centrally stored) or the distributed stored.If the location of database in change then the existing application must continually access the change database
12	The nonsubversion rule:	The system must not have features that allow you to subvert database structure integrity. Basically, the system must not include back doors that let you cheat the system for features such as administrative privileges or data constraints.

Codd’s rule in detail

1) The information rule:

I) All the information in the database should be represented in the term of relational or table.Information should be stored as an values in a tables

II) Data should be stored in form a table and no other means to stored the data

III) E.g.If want to stored data of student in the form of table.Consider name of Table is Students , it has four field(i.e column name) Roll_no, Firstname ,Lastname and date_of_birth and Consist five record mans Five rows

Students Table			
Roll_no	Firstname	Lastname	date_of_birth
101	Sachin	Godbole	17/07/1981
102	Mahavir	Jain	04/12/1985
103	Dinesh	Maheshwari	09/10/1987
104	Yogesh	Lad	06/11/1985
105	Mahesh	Thorat	07/06/1989

2) Guaranteed access rule

I) The guaranteed access All data must be accessible. The Rule say that there is fundamental requirement of primay key for each record in table ,and there should be no ambiguity by stating the table name and its primary key of the each record in the table along with columns name to be accessed rule

II) For accessing the data from the table , we must provides Table name , Primary key(ie Each unique value for each record(row) in the table) and other column names in the table to be accessed

III) Considered the above Students table, if we want to find the First name , Lastname and date_of_birth of student whose Roll_no is 103

IV) Here , the Roll_no is the primary key for the Students Table, This Roll_no Columns is distinct from all other columns,based upon the primary key, all the information present in the table must be guaranteed accessed

3) Systematic Treatment of Null values

I) Null values could not be treated as blank space or zero values, The null values are known as unknown values, unassigned values should be treated as missing information and inapplicable information that should be treated as systematic , distinct from regular values

II) Null values is very important concept is the database ,A null values must be represented as missin information in the table , it is not same as the blank space, dash, or zero, hash or any other representation

III) A null values means that we don't know what information must be provided or entered in to this field name

IV) Null values must be handled logically and consistent manner

4) Active online catalog based on the relational model:

I) The system must support an online catalog based data dictionary which hold the information or description about the table in the database

II) User Tables: The user table contains the data about the table which is created by any users in the database systems

III) System tables: The system table contains the data about the structure of the database and database object

IV) Metadata: The data which hold the description of table in the database, the table structure, database structure , the relationship among the tables, the queries and on , This data id often called as metadata , in short term, Metadata is data about the data

V) The collection of the system tables is known as the system catalogs or data dictionary

5) The comprehensive data sublanguage rule:

I)The system must support at least one relational language that through which the data in the database must be accessed

II) The language must support all the operation of the following items:

Data definition View
definition Data
manipulation
Integrity constraints
Authorization
Transaction boundaries (begin, commit and rollback)

6) The view updating rule:

I) All the view must be theoretically updatable can be updated by the system

II)There is ambiguity in this rule, the Structured query lanagauge support a single updation at a time suppose if we try combine two or more tables a for a complex views and try to update the views and the DBMS would fail to update the records to the respective tables, thereby violating this rule.

IV) If that view doesn't include the primary key columns in the view, then each record in the table cannot be updated, thereby violating this rule.

Eg. If Roll_no column is not present in the view then it is not possible update the view of the student table

7) High-level insert, update, and delete:

I) This rules states that in the relational model, the structured query language must performed data manipulation such as inserting ,updating and deleting record on sets of rows in the table

II) You expected from the RDBMS, that you can retrieves all the record from table applying single command on the set of tables,or by using single query statement, this rules state that not only retrieves all the record from table but also you can apply the delete , insert, and update multiple records should possible by using the single command

III) Considered an example, if you want to delete the record of the invoices table which are older than six years,you don't have locate postion each record and delete them individually , uou should able to delete set of records in the table using one single command

IV) The same concept can be apply to inserting and updating the record

8) Physical data independence:

i) Any change made in the data is physically stored in term of data is stored in the file system through array and link list must not effect application that access the data structure

This rule say that any change is made in the back end(SQLServer/oracle) must not effect front end application(Visual basic/Java)

If the database file renamed or database location is change, then this should not have effect on the application.

9) Logical data independence:

This rule state that changes in the logical level(rows ,columns and so on) must not change to the application's structure

This rule state that it should possible to change the database design or alter the database design without the user being aware of it.

Thse change could be to adding a new table in the datable or to delete the table from the database but the application must effect for accessing the datastructure

Consider an example if want the performance search the record in the table, for that reason you have split the Customer table in to part i.e Customer_India and Customer_Rest,This allows to search a recor in the Customer_India rapidly, but what about the exiting user who is referring to the Customer table.In practice it can be done by creating a view which will combines two table into the single table with the same name. so that there should be effect on the application.

10) Integrity independence:

Data integrity constraints should be considered as separated from application program, the structured query language which defines data integrity constraint must be stored in the database in term of data in table that is, in the catalog and not in the application.

Referential integrity and entity integrity is integral part of the relational database , in more specific term, the following two integrity should be apply to the relational database.

i)Entity integrity:The column which have the primary key value should not contain missing values or duplicate value.This mean the column should contain the null values and unique value in the each record set

ii) Referential integrity:The column which have the foreign key value, there must exist a matching primary key column value mean the foreign key column have duplicate value must be referential to primary key column value.

11) Distribution independence

The rules states that the data can be stored centrally on the single machine or it can be stored in the various location(ditributed) on various machine but it should be invisible to the user i.e the user does not location of data is stored whether on the single machine(Centrally stored) or the distributed stored.If the location of database in change then the existing application must continually access the change database

One of the important benefits of networking is that it allows multi-user access to a database; that is, the users can access the data which is distributed across the network.

However, it is also possible to distribute the data across the same network.

This rule also state that even if the table moves from one location to another location the user should aware of it, it should be transparent to the user, changing in the location mean that the application should not be rewriiten.

12 Nonsubversion rule

The system must not have features that allow you to subvert database structure integrity. Basically, the system must not include back doors that let you cheat the system for features such as administrative privileges or data constraints.

To understand another way, a user should not be allowed access the database by means of other way, other than SQL

❖❖❖❖

6

RELATIONAL DATABASE MODEL

Unit Structure

6.1 RELATIONAL DATABASE MODEL:

E.F.codd first proposed the realational database Model also he is known as the father of Realtional model.

Relation model was attempt to specify the database structure in term of matrix.ie the database should contain tables.The tables is in form of set of Columns and Rows. The relational model is set of 2 dimensional table consists of rows and columns

Tables in the database is known as relation and Columns in the table is called as attributes of an tables and rows in the table is called records or tuples

In the relational database model consists of set of tables having the unique name

One row in the table represents a relationships among the another table in the database.The set values in the one table is related to the set of values in another tables. Thus the table is represents a collection of relationship, The relationship among the tables in the form primary key –foreign key relationship

6.2 LOGICAL VIEW OF DATA:

1 introduction

Logical structure of tables is consist of 2- dimensional tables consist of numbers of horizontal rows and vertical columns

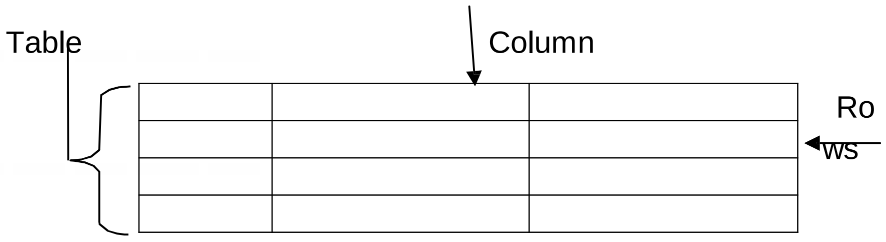


Table is an abstract entity which does not say how the data is stored in the physical memory of the computer system

Each table in the database has its own unique name trough which we can refer the content of the table by the unique name

2.Characteristics of an table

I)A tables in the database must be in the two-dimensional structure which consist number of rows and columns

II)Each row in table as called as record or tuple can represent as a single entity which is occur within the entity set i.e Customer record in the Customer table

III)Each column name in the table is called as attribute and each row in the table is called as record. Each column name in the table is unique namei.e no duplicate name in the same table cannot be repeated.

IV)Each rows/ coloumn interection represent a single data item.

V)All the value in the column must be represent in the same data format
VI)Each columns has the specific range of values, and also refer as the domain attribute

VII) The order of rows and columns is not limited to the DBMS.

3.Example

There is Customer Table contain all information about the Customer

Cust_id
Cust_Name
Cust_Age
Cust_Address
Cust_Mobile_No
Cust_Phone_No

Cust_id	Cust_ Name	Cust_ Age	Cust_ Address	Cust_Mobile _No	Cust_Phone_ No
1	Yogesh	20	Worli	9892456123	0224672345
2	Ramesh	23	Bandra	9320896742	0225678894
3	Ram	18	mahim	9819674534	0224678678
4	Pramod	24	Khar Road	9821673445	0223456478
5	Yatin	25	Dadar	9892396735	0222456783
6	Tushar	26	Matunga	9867458432	0226783452

Attribute

- Each column in the above table represent the data item in the database
- Each column in the table represent the attribute in the table
- Atleast one column consist in the table
- There must be one unique column in the table , this means that no two columns has the same name in the same table ,it is possible to have two column with same colmn name but it in the different table.
- The ANSI/SQL Standard does not specify a maximum numbers of rows and columns in the table.

Eg. Cust_id ,Cust_Name ,Cust_Age ,Cust_Address ,Cust_Mobile_No, Cust_Phone_No are the attributes of the Customer Table

Records/Tuples

- A single Record consist all the information of the single entity.
- Each horizontal row in the Customer table represented a single entity
- A Table consist any number of rows, The ANSI/SQL Standard doesnot specify the limits of rows in the table.
- Empty table is called when there is zero row consist in the table

6.3 KEY

Definition

A Column value in the table that uniquely identifies a single record in the table is called key of an table

A attribute or the set of attribute in the table that uniquely identifies each record in the entity set is called a key for that entity set

Types of keys

Simple Key: A key which has the single attribute is known as a simple key

Composite key: A key which consist two or more attributes is called a Composite Key.

Example:

Cust_id is a key attribute of Customer Table it is possible to have a single key for one customer i.e is Cust_id ie Cust_id =1 is only for the Cust_name =”Yogesh” please refer to the Customer Table which is mentioned above.

Types of key	Definition of Key
Super Key	A key is called super key which is sufficient to identify the unique record in the table
Candidate Key	A minimal super key is called Candidate key .A super key has no proper subset of candidate key
Primary Key	A candidate key is chosen as a principal to identify a unique
Secondary Key	
Foreign Key	an Column (or combination of Columns) in the one tables whose values is match the primary key in the another table

Types of key

1 Super Key

A key is called super key which is sufficient to identify the unique record in the table

Customer Table

Cust_id	Cust_ Name	Cust_ Age	Cust_ Address	Cust_Mobile_ No	Cust_Phone_ No
1	Yogesh	20	Worli	9892456123	0224672345
2	Ramesh	23	Bandra	9320896742	0225678894
3	Ram	18	mahim	9819674534	0224678678
4	Pramod	24	Khar Road	9821673445	0223456478
5	Yatin	25	Dadar	9892396735	0222456783
6	Tushar	26	Matunga	9867458432	0226783452

Example

Here Cust_id attribute of the entity set Customer is uniquely identify Customer entity from another so The Cust_id is the Super key. Another way is, the combination of Cust_id attribute and Cust_Name attribute is the Super key for the Customer Entity set. Only the Cust_Name is not called the Super Key because several customer may have the Same Name

2. candidate key

Defination:

A minimal super key is called Candidate key .A super key has no proper subset of candidate key

Here Minimum attribute of the super key is omitted unwanted attributed of an table that key is sufficient for identifying the unique record in the entity set so it is called as Candidate key

The Candidate key is also known as the primary key

Example

Cust_id	Cust_ Name	Cust _Age	Cust_ Address	Cust_Mobile _No	Cust_Phone_ No
1	Yogesh	20	Worli	9892456123	0224672345
2	Ramesh	23	Bandra	9320896742	0225678894
3	Ram	18	mahim	9819674534	0224678678
4	Pramod	24	Khar Road	9821673445	0223456478
5	Yatin	25	Dadar	9892396735	0222456783
6	Tushar	26	Matunga	9867458432	0226783452

From above statement say combination of Cust_id attribute and Cust_Name is a super key for the Customer entity set it is required to distinguish one record on the Customer entity from another record of sane set.

But Cust_id attribute of the Customer entity is asl known as minimal super key which also enough to distinguish one record from customer entity from another record from customer entity set, because Cust_Name is th additional attribute of the Csutomer table

2 Primary key

Defination

Primary key of the table is a columns or combination of the some columns whose values is uniquely identify a single record in the table.

Primary key state no two record of the table contain the same value in that column or Cobination of the column It state that a unique identifier for the entity set.

Cust_id	Cust_ Name	Cust_ Age	Cust_ Address	Cust_Mobile _No	Cust_Phone_ No
1	Yogesh	20	Worli	9892456123	0224672345
2	Ramesh	23	Bandra	9320896742	0225678894
3	Ram	18	mahim	9819674534	0224678678
4	Ram	20	Khar Road	9821673445	0223456478
5	Yatin	25	Dadar	9892396735	0222456783
6	Tushar	26	Matunga	9867458432	0226783452

In above table the Customer Age cannot act as primary key hence the customer age column contain repeated values and Customer Name also cannot act as primary key because it earlier state that several customer may have the same name hence Cust_Name column has the repeated values .

Hence there Cust_id can act as the primary key in the Customer table this is only column which contain a unique set of values.

3 Secondary key

Defination

Seconday key of the table consist the column and combination of the some columns which meant for data retrival purpose.

The secondary key not always required to primary key, other tah the pimary key there are some attribute which is required to retrieve data from the customer table using the another attribute such as Cust_Name and Cust_Age columns

Cust_id	Cust_ Name	Cust_ Age	Cust_ Address	Cust_Mobile_ No	Cust_Phone_ No
1	Yogesh	20	Worli	9892456123	0224672345
2	Ramesh	23	Bandra	9320896742	0225678894
3	Ram	18	mahim	9819674534	0224678678
4	Ram	20	Khar Road	9821673445	0223456478
5	Yatin	25	Dadar	9892396735	0222456783
6	Tushar	26	Matunga	9867458432	0226783452

In the above Customer Table Cust_Name and Cust_Age attribute act as the Secondary key

Foreign Key

A Column (or combination of Columns) in the one tables whose values is match the primary key in the another table is called as a foreign key

Foreign key can also have one or more column like as primary key A

single table may contain more than one foreign key which is related to the more tah one table, the table which used the foreign key is said the referiential integrity

What the referential integrity

Referential integrity say the column which contain foreign key in one table must be primary key of another table

In general term, Foreign key of Table A must be Primary key of Table B

Example

Customer Table

Cust_id	Cust_ Name	Cust_ Age	Cust_ Address	Cust_Mobile_ No	Cust_Phone_ No

Account Table

Account No	Cust_id	Account type	Balance	Description

In The above example Cust_id is the primary key for the customer Table while Cust_id is the foreing key for the Account table

Here the the datatype assigned to column and Numder of column in the foreign key is same as to the primary key.

Cust_id	Cust_ Name	Cust_ Age	Cust_ Address	Cust_Mobile_ No	Cust_Phone_ No
1	Yogesh	20	Worli	9892456123	0224672345
2	Ramesh	23	Bandra	9320896742	0225678894
3	Ram	18	mahim	9819674534	0224678678
4	Ram	20	Khar Road	9821673445	0223456478
5	Yatin	25	Dadar	9892396735	0222456783
6	Tushar	26	Matunga	9867458432	0226783452

Account No	Cust_id	Account type	Balance	Description
101	1	Saving	10,000	
102	2	saving	20,200	
103	2	Saving	20,200	
104	3	Current	11,000	
105	4	Saving	50,000	

6.4 REALATIONAL INTEGRITY RULES

1) Entity Integrity

Entity Integrity ensure that there is no duplicate records in the table and each field that recognizes each record in the table must have unique value and not having null values

Entity Integrity specfies that every instance of entity have the unique values ie primary key must be kept and must have the values other than null values.

Entity Integrity is the mechanism the Database management system provides to maintain primary keys. The primary key is known as unique identifier for each rows in the table . Entity Integrity must have two properties for primary keys:

- The primary key must be unique for each row in the table that is no two primary key having the same value in the same table, The primary key values must be distinct i.e the value could not be repeated.
- The primary key values should not contain null values, primary key must be NOT NULL

The uniqueness property ensures that the primary key of each row uniquely identifies it; there are no duplicates. The second property ensures that the primary key has meaning, has a value; no component of the key is missing.

2. Referential Integrity

Referential integrity is a property of data which, when satisfied, requires every value of one attribute (column) of a relation (table) to exist as a value of another attribute in a different (or the same) relation (table).

For referential integrity to hold in a relational database, any field in a table that is declared a foreign key can contain only values from a parent table’s primary key or a candidate key. For instance, deleting a record that contains a value referred to by a foreign key in another table would break referential integrity.

Some relational database management systems (RDBMS) can enforce referential integrity, normally either by deleting the foreign key rows as well to maintain integrity, or by returning an error and not performing the delete.

Foreign key

A column or collection of column in one table whose values must match the primary key in the other table is known as a foreign key

Cust_id	Cust_ Name	Cust _Age	Cust_ Address	Cust_Mobile_ No	Cust_Phone_ No
1	Yogesh	20	Worli	9892456123	0224672345
2	Ramesh	23	Bandra	9320896742	0225678894
3	Ram	18	mahim	9819674534	0224678678
4	Ram	20	Khar Road	9821673445	0223456478
5	Yatin	25	Dadar	9892396735	0222456783
6	Tushar	26	Matunga	9867458432	0226783452

Account No	Cust_id	Account type	Balance	Description
101	1	Saving	10,000	
102	2	saving	20,200	
103	2	Saving	20,200	
104	3	Current	11,000	
105	4	Saving	50,000	

In above example

Cust_id column of Account Table is foreign key for the Account table while it is primary key for the Customer Table

3. Other integrity rules

NOT NULL. As the integrity rules states column which specify the NOT NULL values mean these column must contain some values which should not contain any NULL values

Unique. In this rules no two record or tuples have same values for the same attribute

Check. In this rule we can apply own integrity rules by applying CHECK Constraint.

6.5 RELATIONAL DATABASE DESIGN PROCESS

The Relational Database model was proposed by E.F.Codd in 1969.The Realtional Database Model is based on branch of mathematics called set theory and predicate logic. The idea behind to design the Relational Database model is that the database consist of series of unordered table or relation that can be manipulated using non-procedural process that return tables

Note: it is Commonly thought that word relational in the relational model comes from the fact that the tables is related together in the relational model, but it is inconvenient way to think of the term , but it is not accurate.. The table which codd is specifies while in writings was actually referred to as relation (a related set of Information).

While designing relational database model you have consider in the mind that how choose a best model in the real world and how this best model is fitted in the database. While designing the relational model you have o consider that which table you want to create, what column the table will consist, consider the relationship between the tables. While developing the relational model it would be nice you process was totally clear and intuitive or it can be even better to automated.

- The benefits of a relational Database Design process.
- Data entry, updating and deleting would be efficient and simple in manner
- Data retrieval, summarization and reporting will be efficient

- Database must follows a well designed model hence it behave predictably
- Large amount of information must stored in the database rather than in the application, the database must somewhat well documented
- Change to database structured are easy to make e.r creating database, tables , views.

6.5.1 Feature of Good Relational Database Design-Normalization

i) In the Relational Database Design, the process of organizing data to minimizing redundancy is known as Normalization The

ii) main aim of the Normalization is to decompose complex relation into smaller, well-structured relation

iii) Normalization is the process that involves dividing a large table into smaller table(which contain less redundant data) and stating the relationship among the tables.

iv) Data normalization or Database Normalization is also canonical synthesis is mean for preventing the inconsistent in a set of data by using unique values to reference common information

v) The main objective of the normalization is to isolate the data so that user can apply the operation such as addition, deletion and modification of a field in one table and then its propagated to the rest of the database through the well defined relationships

vi) The same set of data is repeated in multiple tables of database so there are chances that data in the database may lead to be inconsistent, so while updating , deleting or inserting the data into the inconsistent database which leads to problem of data integrity

vii) If we can apply the normalization on the table we can reduce the problem of data inconsistency for some extent

Definition of Normalization

In the Relational Database Design, the process of organizing data to minimizing redundancy is known as Normalization

Main aim of the Normalization

1. Ensure data integrity

i) The correct data should be stored in the database

ii) This can be achieved by applying integrity rules in the database

iii) Integrity rules prevent duplicate values in the database

2. Prevent Data Redundancy in database

i) Non-Normalized data is more vulnerable to data anomalies. The same set of information is present in the multiple rows, now if we applying the updating rule on the table then it lead to logical inconsistence this is known as update anomaly

An insufficiently normalized table might have one or more of the following characteristics:

update anomaly

The same set of information is present in the multiple rows, now if we applying the updating rule on the table then it lead to logical inconsistence. Consider an example of customer Table which contain set of attributes such as Cust_id ,Cust_Name, Cust_Address,

Cust_id	Cust_Name	Cust_Address
423	Pramod	Nerul
423	Pramod	Nerul
567	Manish	Vashi
567	Manish	Bhandup

Thus a change of address of a particular Customer will need update to multiple records. If the update is not carried out successfully—if, that is, the Customer's address is updated on some records but not others—then the table is remains in an inconsistent state. Specifically, the table provides conflicting answers to the question of what this particular customer's address is. This Known is known as an **update anomaly**.

The above Customer Table is Cust_id =567 having different address in the multiple records

An insertion anomaly

There are some circumstances in which certain fact cannot recorded at all

Example Consider a table Faculty and Course_code consist the Column name

Faculty_ID,Faculty_Name,Faculty_Hire_Date,Course_Code

Faculty_ID	Faculty_ Name	Faculty_Hire_ Date	Course_Code
386	Mahesh Lad	10/06/1994	ENG-207
197	Jayesh Shinde	12/06/1987	PP-205
197	Jayesh Shinde	12/06/1987	PP-206
234	Pramod Bhave	11/07/2005	?

Thus we can add the record the details of any faculty member who teaches at least one course, but we cannot record the details of a newly-hired faculty member who has not yet been assigned to teach any courses except by setting the Course Code to null. This known as an **insertion anomaly**.

In the above Table Until the new faculty member, Pramod Bhave , is assigned to teach at least one course, his details cannot be recorded.

An deletion anomaly.

There are circumstances in which the deletion of data representing certain facts necessitates the deletion of some unrelated data . The "Faculty and Courses" table suffers from this type of anomaly, for if a faculty member temporarily ceases to be assigned to any courses, we must delete the last of the records on which that faculty member appears, effectively also deleting the faculty member. This is known as a **deletion anomaly**.

Faculty_ID	Faculty_Name	Faculty_Hire_Date	Course_Code
386	Mahesh Lad	10/06/1994	ENG-207
197	Jayesh Shinde	12/06/1987	PP-205
197	Jayesh Shinde	12/06/1987	PP-206

Delete

All information about Mahesh Lad is lost when he temporarily ceases to be assigned to any courses.

Advantage of Normalization

1)Avoids data modification (INSERT/DELETE/UPDATE) anomalies as each data item lives in One place

2) Greater flexibility in getting the expected data in atomic granular

3)Normalization is conceptually cleaner and easier to maintain and change as your needs change

4) Fewer null values and less opportunity for inconsistency
5) A better handle on database security

6) Increased storage efficiency

7)The normalization process helps maximize the use of clustered indexes, which is the most powerful and useful type of index available. As more data is separated into multiple tables because of normalization, the more clustered indexes become available to help speed up data access.

Disadvantage of Normalization

1)Requires much more CPU, memory, and I/O to process thus normalized data gives reduced database performance

2)Requires more joins to get the desired result. A poorly-written query can bring the database down

3)Maintenance overhead. The higher the level of normalization, the greater the number of tables in the database

Thank You