Unit - 1

Page :
Date: / /

Relational Model Confirmity & Integrity:-

Relational data model is the primary data model, which is used widely around the world for data storage and processing.

This model is simple but it has all the properties and capabilities required to process data with storage efficiency.

* Concepts:

Tables:- In a relational data model, relations are saved in front of tables. The table stores the relation among the entities. A table has rows & columns where rows represent records & columns represent the attributes.

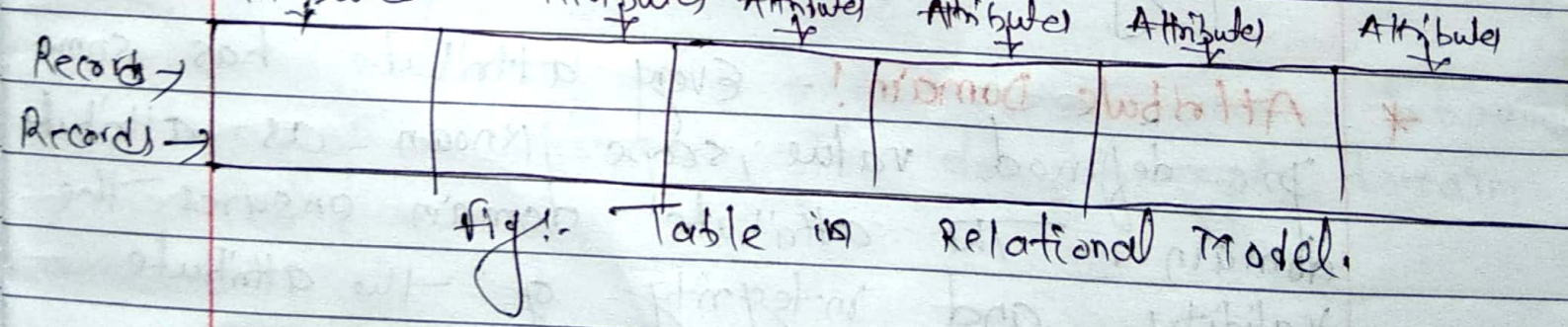


fig:- Table in Relational Model.

* Tuple:-

It is a single row of a table which contains a single record for that specific relation.

Page :
Date: / /

* **Relation instance!-** A finite set of tuples in the relational database system represents relation instance. Relation instances doesnot have duplicate tuples.

* **Relation schema!-** A relation schema describes the structure and properties of the relation such as the relation name, attributes and their names.

* **Relation Key!-** Each row has one or more attributes known as Relation Key which can identify the row in the relation uniquely

* **Attribute Domain!-** Every attribute has some pre-defined value, scope, known as attribute domain. The attribute domain ensures the validity and integrity of the attribute value

* **Constraints!-** Every relation has some conditions that must hold for it to be valid relation. These conditions are called relational integrity constraint. They are:

Page:
Date: / /

- Key constraints :- There must be atleast one subset of attributes that can identify a tuple uniquely. This subset is key for that relation.

If more than one such subsets exist then they are called candidate keys. Key constraints force that :

(i) in a relation with key attributes, no two tuples can have identical values for key attributes.

(ii) A key attributes cannot have null values.

- Domain Constraints :- Attributes are bound to some values range in real-world scenario. for eg : Age can only be a positive integer.

To enforce such range and to bound every attributes to domain range, domain constraints are used.

- Referential Integrity Constraints :- It comes on the concept of foreign keys.

It states that if a relation refers to a key attributes of a different or same relation, then that key element must exist

Page :
Date: / /

* Query optimization :-

A Query is a request for information from a database. A query plan is an ordered set of steps used to access data in a SQL relational database management system.

Given a query, there are many plans that a DBMS can follow to process it and produce it's answer. All the plans are equivalent in terms of their final o/p but vary in their cost. The idea of query optimization is to choose the most optimized plan.

Query optimization is a process of identifying how on RDBMS can improve on the performance of a query by re-ordering or exchanging the operations. The query optimization attempts to determine the most efficient way to execute a given query by considering the possible query plans.

A single query can be executed through different algorithms or re-written in different forms and structures. The query optimize attempts to determine the most efficient way to execute a given query by consider-

Page:
Date: / /

ring the possible query plans. The structure of query optimizer is as :-

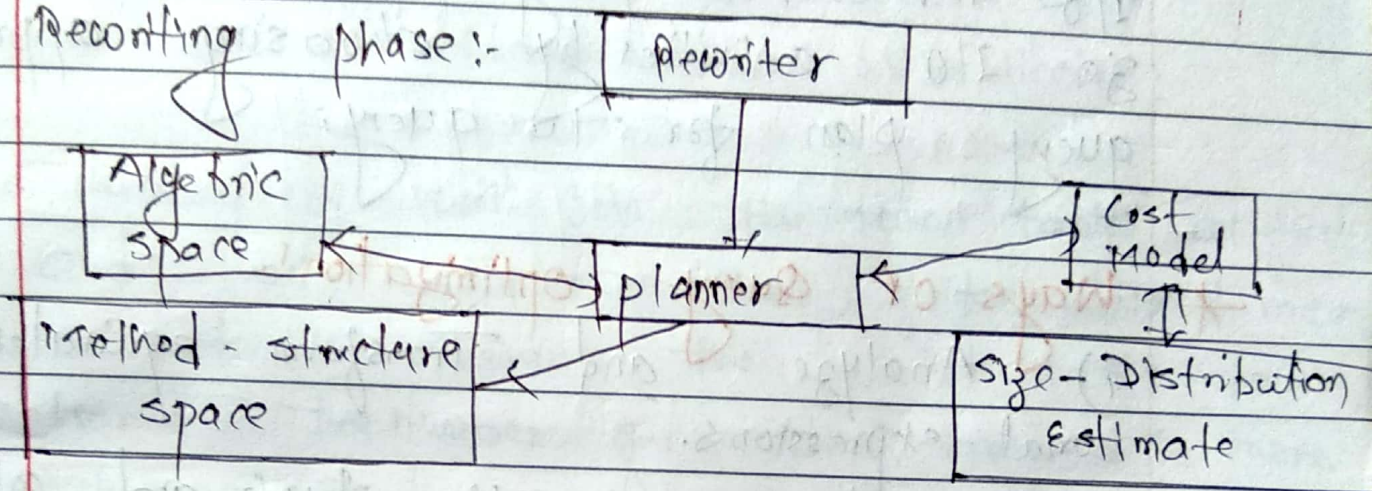


Fig :- Query optimizer Architecture.

* Need for Query optimization :-

(i). To perform automatic Navigation : A relational database system allows users to simply state what data they require and leave the system to locate and procss that data in the database.

(ii) To achieve acceptable performance : There may be various plans to perform a single user query and the query optimizer aims to select and execute the most efficient query plan based on information available to the system.

Page :
Date: / /

iii. To minimize existing discrepancies : Due to existing discrepancy in speed between CPU & I/O devices, a query optimizer aims to minimize IO activities by choosing appropriate query plan for the query.

* Ways of Query optimization :-

i) Analyze and Transform equivalent relational expressions.

Try to minimize the tuple and column counts of the intermediate and final query processes.

ii) Using different algorithms for each operation :- The underlying algorithms determine how tuples are accessed from the data structures they are stored in, indexing, hashing, data retrieval and hence influence the number of disk and block accesses.

Example :-

A query to retrieve students with age 18 and studying in Class 'Architecture'. The query is as follows:

Page :
Date: / /

```
SELECT std.* FROM STUDENT std, CLASS cls
WHERE std.CLASS-ID = cls.CLASS_ID
AND std.AGE = 18
AND cls.CLASS-NAME = 'Architecture';
```

This query will join the two tables at first and then applies the filters. That means ~~that~~ it traverses the whole table to join, hence the number of records involved is more. The optimized query is as follows:-

```
SELECT std.* from
(SELECT * FROM STUDENT WHERE AGE=18) std,
(SELECT * FROM CLASS WHERE CLASS.NAME=
                    'Architecture') cls
WHERE std.CLASS-ID = cls.class_ID
```

This query applies filters on each table first hence it reduces the number of records in each table then joins these intermediatory tables. Hence the cost of the second query is comparitively low.

Page :
Date:

Concurrency Control and Transaction Management :

→ Database system typically allows multiple users to access shared data concurrently. Here concurrency control is critical in ensuring that concurrent operations are carried out correctly and effectively

→ A transaction typically involves a series of I/O operations and other processing operations.

Assume that transactions T_A and T_B performs a series of operations T_{A1}, T_{A2}, T_{A3} and T_{B1}, T_{B2}, T_{B3} respectively. Then,

- In single transaction environment T_A is ran in the following sequence.

$T_{A1} \rightarrow T_{A2} \rightarrow T_{A3}$

- In Mutti-transaction environment T_A & T_B may run in any order such as:

$T_{A1} \rightarrow T_{B1} \rightarrow T_{B2} \rightarrow T_{A2} \rightarrow T_{A3} \rightarrow T_{B3}$

→ Concurrency is the process of managing simultaneous operations on the database without having them to interfere with each other. It prevents error when two or more users are accessing database simultaneously and at least one is

updating data. Although two transactions may be correct in themselves, interleaving of operations may produce an incorrect result.

Problems that occur when concurrent execution is uncontrolled:-

(a) The Lost Update Problem:- This problem occurs when two transactions that access the same database items have their operations interleaved is a way that makes the value of some address database item incorrectly.

T1	T2
read_item (x);	
x = x - N;	
	read_item (x)
	x = x + y;
write_item(x);	
read_item(y);	
	write item (x);
y = y + N;	
Write-Item (y);	

Date: / /

if X=80, N=5, Y=4 then our result must be 79 but in the above case due to interleaved operations our output is 84. Since the update made by the statement X = X - N ; is lost.

(b) The Temporary update problem :-
This problem occurs when one transaction updates a database item and then the transaction fails for some reason. The updated item is accessed by another transaction before it is changed back to its original value.

T_1	T_2
read-item(X);	
X = X + M;	
write-item(X);	
	read-item(X);
	X = X - Y;
	write-item(X);
read-item(Y)	

In above example if transaction T_2 fails

Page :
Date: / /

after T_2 has completed and operation T_1 is rolled back then operation T_2 reads the temporary value of x. Since the temporary value has been created by the rolled back operation it is temporary update problem,

(c) The Incorrect Summary Problem :-

If one transaction is calculating an aggregate summary function on a number of records while other transactions are updating some records, the aggregate function may calculate some values before they are updated and others they are updated.

T_1	T_2

	sum = 0
	read-item(A);
	sum := sum + (A);
	:

read-item(x) x := x - N write-item(x);	

	read-item(x);
	Sum := Sum + x;
	read-item(y);
	sum := sum + y
	:

read-item(y);
y := y + N;
write-item(y);	

Page :
Date: / /

T_2 reads x after N is subtracted and reads y before N is added, a wrong summary result is produced.

Solutions to the Concurrency Problems:-

The problems that arise during concurrent execution can be solved by two approaches:

(1). Serialization:-

~~The problems that arise during concurrent execution can be solved by~~ Serializability is a classical concurrency scheme. It ensures that a schedule for any existing concurrent transactions is equivalent to one that ~~exe~~ executes the transactions serially in some order.

It assumes that all accesses to the database are done using read and write operations. A schedule is called 'correct' if we can find a serial schedule that is equivalent to it.

Given a set of transactions $T_1, T_2 \ldots T_n$, two schedules S_1 and S_2 of these transactions are equivalent if the following conditions are satisfied.

Page :
Date: / /

Read-write synchronization :- If a transaction reads a value written by another transactions in one schedule, then it also does so in the other schedules.

Write-Write ~~transaction~~ synchronization :- If a transaction overwrites the value of another transaction.

(ii) Concurrency control protocols :-

These protocols allow concurrent execution of transactions in a controlled way. They can be divided into two major classes.

(a) Lock-Based Protocols :- Database systems equipped with lock-based protocols use a mechanism by which any transaction cannot read or write data until it acquires appropriate lock on it. Locks are of two types :

- Binary locks : A lock on data item can be in two states ; it is either locked or unlocked.

- **shared / Exclusive :-** Here The locking mechanism differentiates the locks based on their uses. If a lock is acquired on a data item to perform a write operation it is an exclusive lock while read locks are shared a no data value is being changed.

  Eg:-

  Simplistic lock protocol
  pre-claiming lock protocol
  2-phase locking protocol (2PL)
  strict 2PL.

(b). **Time stamp - Based Protocols:-**

The most common used concurrency control protocol is the timestamp based protocol. This protocol either uses a system time or logical counter as a time stamp.

lock based protocols manage the order between the conflicting pairs among transactions at the time of execution while time-stamp based protocols start working as soon as a transaction is created.

Page :
Date: / /

Every transaction has a timestamp associated with it and the ordering is determined by the age of transaction A transaction created at 0002 clock has higher priority than a transaction created 0004 clock.

- Time stamp ordering protocol: The conflicting pair of tasks should be executed according to time stamp values of transactions.
- Thomas write Rule: Improves time stamp ordering protocol by modifying it to make the schedule view serializable. Instead of rolling back any transaction Ti, the 'write' operation is ignored.

Database Performance Tuning:-

The process of continuing to revise/adjust the physical database design by monitoring resource utilization as well as internal DBMS processing to reveal bottlenecks such as contention for some data or devices.

Date: / /

The main goal of database tuning is
→ To make application run faster.
→ To lower the response time of queries/ transactions.
→ To improve the overall throughput of transactions.

Performance is Affected by:-
- The resources available in the system.
- How well these resources are used and shared.

Database tuning describes a group of activities used to optimize and standardize the performance of a database. Database tuning aims to maximize the use of system resources to perform work as efficiently and rapidly as possibly.

Methods of DB performance tuning:-
- **Tuning indexes:-** Certain queries may take long time to run due to lack of index and some indexes may not be utilized at all. Further, some indexes may cause excessive overhead due to the frequent change in index attribute. These all problems may

Page :
Date: / /

be considered. Some solutions are:-

* Drop and/or build new indexes.
* change a non-clustered index to a clustered-index.
* Rebuilding the index.

• **Tuning the Database Design :-** The basic blocks of database are tables, hence developing efficient tables is of great help in performance.

Similarly, to meet the changing processing requirements, modification of conceptual schema, logical schema and physical schema design is necessary.

Actions such as normalization and denormalization can help when attributes from two or more tables are frequently needed.

Reducing the number of read & write operation greatly affect the performance of database which can be done by eliminating redundancy and potential anamolies.

* **Tuning Queries :-** A Query may issue to many disk access, the query must be

optimized by comparing all possible algorithms. A query plan may not use the relevant indexes even if they can be used due to presence of arithmetic expressions, numerical comparisions of attributes of different sizes & precision, NULL comparisions and sub-string comparisions.

* Checking H/W and S/W incompatibility and problems are reduces.

* Distributed Relational System and Data Replication

A distributed database systems allows applications to access data from local and remote databases. A distributed database is a database in which storage devices are not attached to a common processing unit such as a CPU and which is controlled by a distributed database management system.

It may be stored in multiple computers, located in same physical location, or may be distributed over a N/W of interconnected computers.

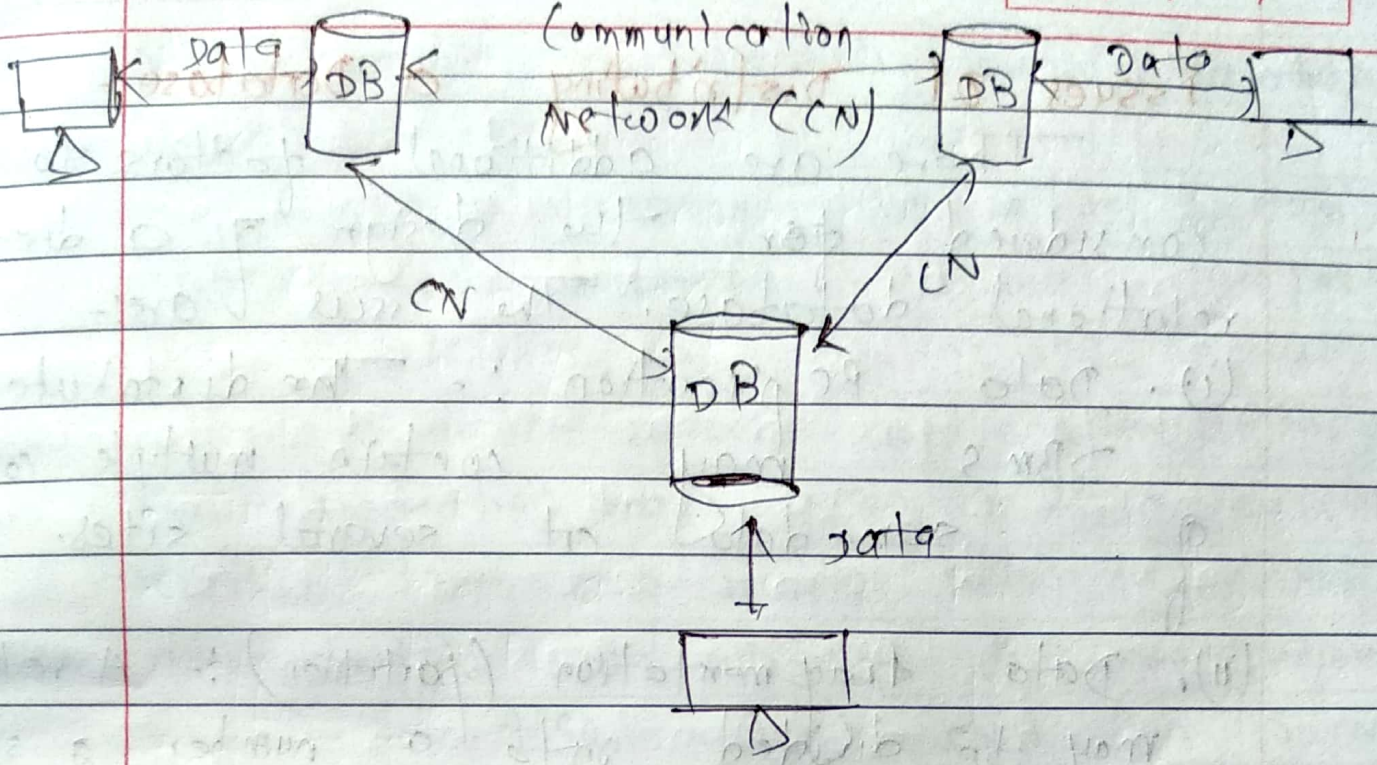


Fig: Distributed database system.

An examples of distributed system is airline reservation systems with operations in several locations all over the world.

* Advantages:-
- Increased availability & reliability.
- Lower communication cost.
- fast response & improved performance.
- Local and/or remote access & control

* Disadvantages:-
- Slw cost & complexity.
- Communication / processing overheads.
- Data integrity & consistency.

Date: / /

Issues of Distributing a Database:-

There are additional factors to be considered for the design of a distributed relational database. The issues are:-

(i). Data Replication :- The distributed DBMS may contain multiple copies of same data at several sites.

(ii). Data fragmentation (partition) :- A relation may be divided into a number of sub-relations (fragments) which are then distributed.

(iii) Distribution Transparency :- It allows the users to perceive the distributed database as a single, logical entity.

* Data Replication:-

The term replication refers to the operation of copying and maintaining database objects in multiple databases belonging to a distributed system.

While replication relies on distributed database technology, database replication offers application benefits that are not

Page :
Date: / /

possible within a pure distributed database system (DDBMS).

Most commonly, Replication is used to improve local database performance and protect the availability of applications because alternate data access options exist. for eg:- An application may access a local database rather than a remote server to minimize nlw traffic and achieve maximum performance.

Also, Any application can continue to function in case of local server failure, while other servers with replicated data remain accessible.

Replication involves using special slw that looks for changes in the distributed database, once the changes are identified, the replication process makes the necessary modification to make all the database same.

However replication may be complex & time consuming depending on the size and number of the databases. This process can also require alot of time & resources.

Page :
Date: / /

* Types of Replication:-

* Replicated Database:- It is a distributed database where some of the data is duplicated at various sites.

* fully Replicated Database:- It is a distributed database where all stored data are duplicated and allocated to all sites.

Replication has a different effect on read and update applications:-

- Read only Applications:- They take advantage of replication, which makes it more likely that they can reference data locally.

- Update Applications:- may present problems due to replication, since they must update all copies in order to preserve data consistency.

Data fragmentation:-

It is the process of dividing relations into fragments for distribution

Page :
Date: / /

Advantages :

Usages:- Since application usually work with views rather than entire relations, it makes sense to use subsets of relations as the unit of distribution.

Efficiency:- If the relation can be decomposed into fragments, it is possible to allow a no. of transactions to execute concurrently

parallelism:- Parallel execution can be performed

Disadvantages

Integrity:- Integrity checking can be made more complex if data and functional dependencies are fragmented and distributed to different sites.

Performance:-

It can be slower to process some global applications which require data from fragments located at different sites.

There are 3 fragmentation Rules:-

1. **Completeness:-** Each data item from a global relation R must appear in at least one

Page :
Date: / /

of it's fragments. This rule ensures no loss of data during fragmentation.

(ii) Reconstruction :- It must always be possible to reconstruct each global relation from it's fragments. This rule ensures no loss of functional dependencies.

(iii) Dis Joint less :- Each data item from a global relation should appear in only one of its fragments except for vertical fragmentation where primary key attributes must be repeated to allow the reconstruction. This rule ensures minimal replication.

There are generally three fragmentation options.

- Horizontal fragmentation (Restrict).
- Vertical " (Project)
- Mixed " (Join)

- Transparency in DBMS :-

An important aspect of a DDBMS is to hide the detail of data distribution from it's users. This allows them to

perceive the DDBMS as a single logical entity.

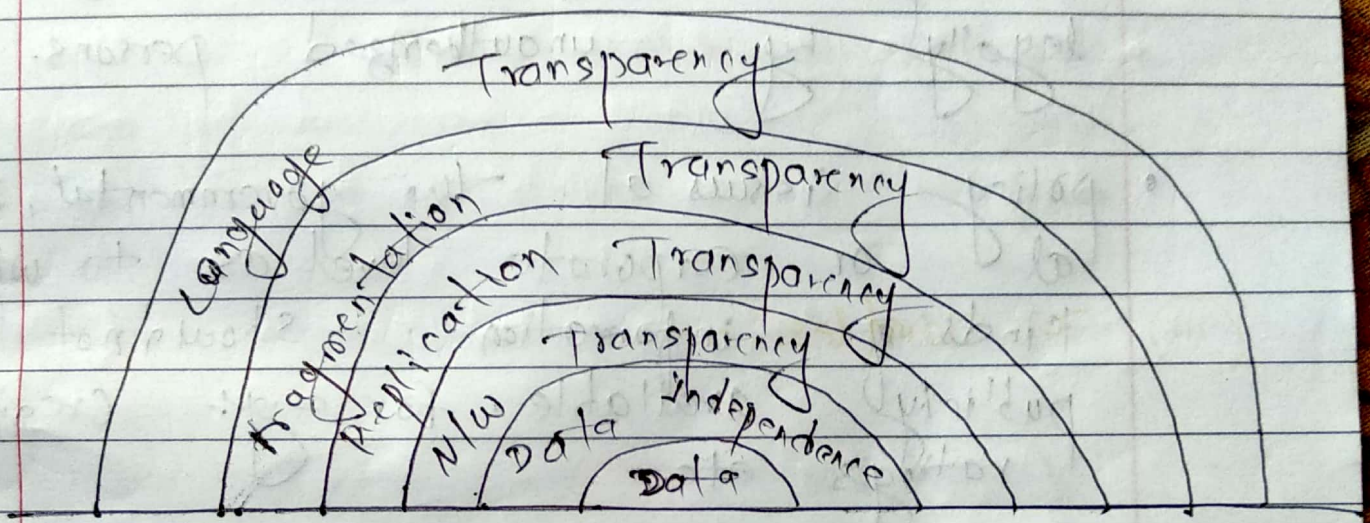


fig:- Types of transparency in DBMS.

Security Consideration !-

Database security involves allowing or disallowing user actions on the database and the objects with in it. It is also concerned with verifying and validating that the data being accessed and the person that is accessing the data.

Page :
Date: / /

Types Of Security:-

- Legal and ethical issues regarding the right to access certain information some information may be considered to be private and cannot be accessed legally by unauthorized persons.

- policy issues at the governmental, institutional or corporate level as to what kinds of information should not be made publicly available. for e.g:- credit card ratings etc.

- System related issues such as the system levels at which various security functions should be enforced for e.g:- either at the physical flow level, the operating system level or the DBMS level.

- The need to identify multiple security levels. e.g:- Top secret, secret, confidential & unclassified.

* Approaches to Database security:-

There are two main approaches to database security, depending on the type of objects through which the access control is enforced.

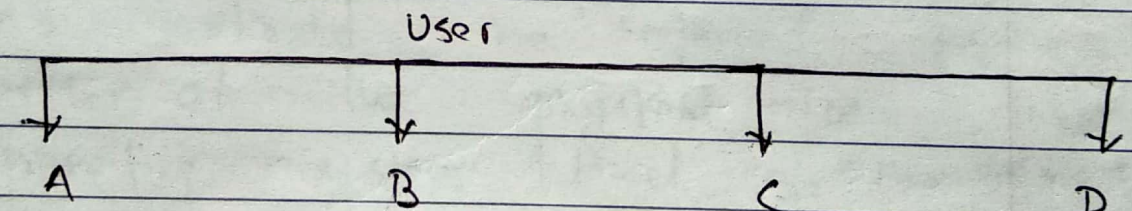
- **Discretionary Access Control:-**

Control is exercised by assigning users different access rights on different data objects.

There are two various methods used in the discretionary ~~model~~ control.

Access Matrix:-

- Use tables to specify access previlege for diff^n users on different data objects.



Data object

Table 1	select		Select	
Attribute 1		select	update	
Att. 2	Update			Update
Table 2		select		
Att 1.	Select			
Att 2.		update		Insert

Page :
Date: / /

Security Rules:-

- Supported by SQL standard and facilitated by two statements GRANT and REVOKE:

Syntax:-

GRANT operation
ON data object
To user(s)
violation response

REVOKE operation
ON data object
To user(s)

- **Mandatory Control:-**
  - Each data object is assigned a classification level and each user is assigned clearance level.
  - users with clearance level i can only access data objects whose classification level is is lower than or equal to i.
  - The classification & clearness level have the same number of possibilities, those levels form a strict ordering hierarchy.

Page:
Date: / /

Eg:-

User	Clearance Level
Directors	4 (Top secret)
Senior Manager	3
Managers	2
Authorized clerks	1
Any body	0

then, classification level,

	Select	Update	Delete
Table 1	2	4	4
att 1	1	3	1
att 2	1	3	0
table 2	3	3	4
att 2	3	3	4

* Database Specific Concerns:-

i) Unauthorized use of database knowingly & unknowingly
- unauthorized users attempting access to db.
- Authorized users attempting unauthorized operators on certain data object.

ii) Data Encryption provides additional protection for sensative data during transmission.

The Extended Entity Relationship Model & Object Model

Page:
Date: / /

UNIT-2:

The ER Model Revisited
Motivation for complex data types
User defined abstract data types & structured types
subclass, superclasses, Inheritence
Specialization & generalization
Constraints and characteristics of Specialization & generalization
Relationship types of degree higher than two

Page :
Date: / /

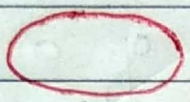
The ER Model Revisited :-

* The entity-Relationship model or (ER model) is a way of graphically representing the logical relationships of entities (or objects) in order to create a database.
* In ER Modeling, the structure of a database is described as a diagram called an entity-relationship diagram (or ER diagram).
* Entities are rendered as points, polygons, circles or ovals. Relationships are portrayed as lines connecting the points, polygons, circles or ovals. Any ER diagram has an equivalent relational table and any relational table has an equivalent ER diagram.
* An entity is a real-world item or concept that exists on it's own. for example : employees, departments, products or networks.
* An entity can be defined by means of it's properties, called attributes.
* The creation of an ER diagram, which is one of the first steps in designing a database, helps the designers to understand and to specify the desired components of the database and the relationships among these components.

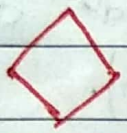
Date: / - /

Symbol and Notation used in ER diagram:-

Entity.



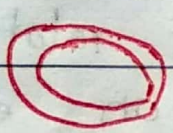
Attribute



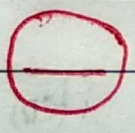
Relationship

Weak entity

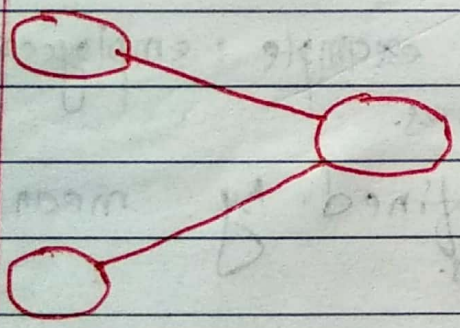
Weak Entity Relationship



multivalued attributes.



Key Attribute



Composite Attribute

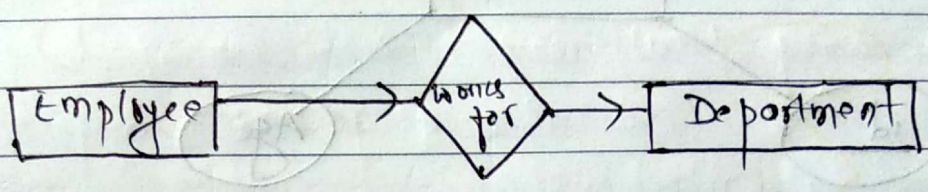
* Components of ER Diagram:-

The E-R diagram has three main components:

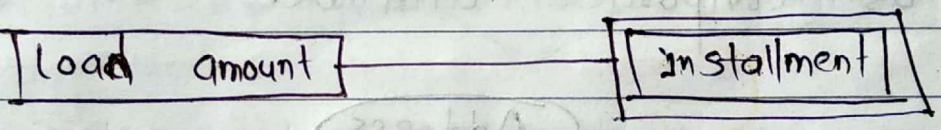
1. Entity:- An entity can be any object, place, person or class. In ER diagram, an entity is represented using rectangles. Consider

Page :
Date: / /

an example of an organization, Employee, manager, Department, product and many more can be taken as entities from an organization.



Weak entity:- Weak Entity is an entity that depends on another entity. weak entity dont have key attribute of their own. Double rectangle represents weak entity

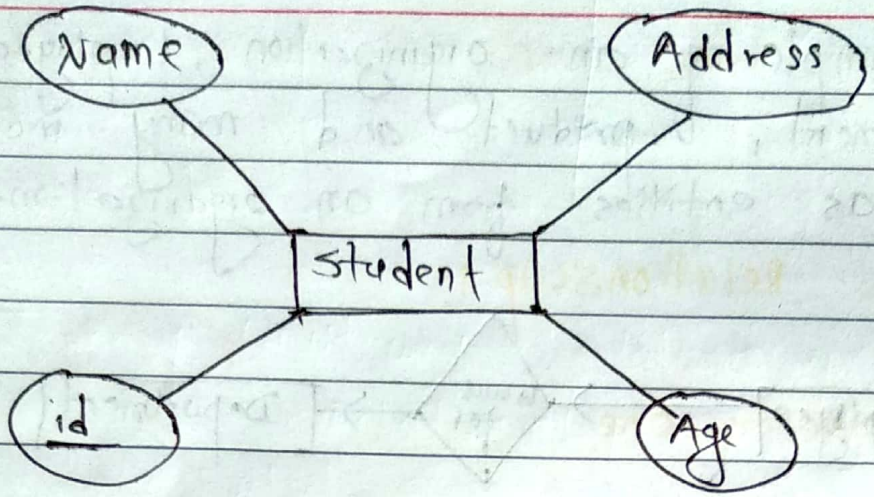


2. Key Attribute:-

An Attribute describes a property or characteristics of an entity. for eg: Name, Age, Address etc. can be attributes of a student, An attribute is represented using eclipse

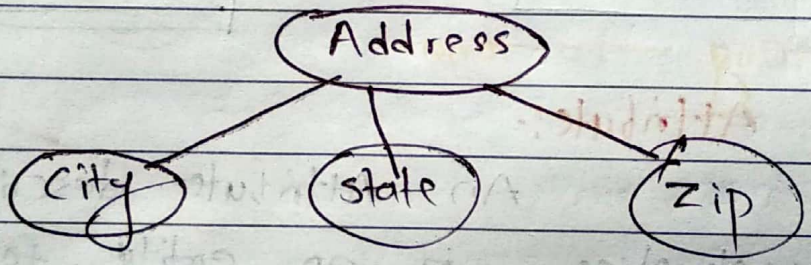
Key Attribute:-

Key attribute represents the main characteristic of an entity. It is used to represent primary key. Eclipse with underlying lines represent key attribute.



Composite Attribute :-

An attribute can also have their own attributes. These attributes are known as composite attribute.



3.) Relationship :-

Relationship describes relation s between entities. Relationship is represented using diamonds.

Types of Relationship :-

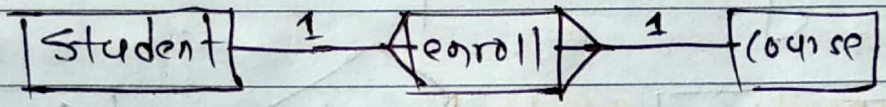
i) Binary Relationship

Page :
Date: / /

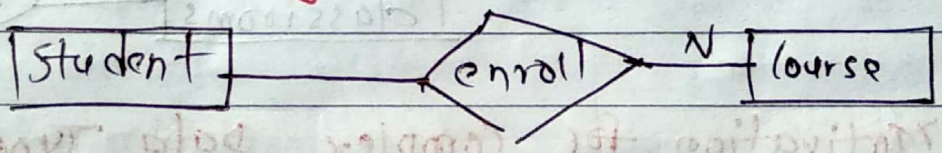
2) Recursive Relationship
3) Ternary Relationship

Binary Relationship :-

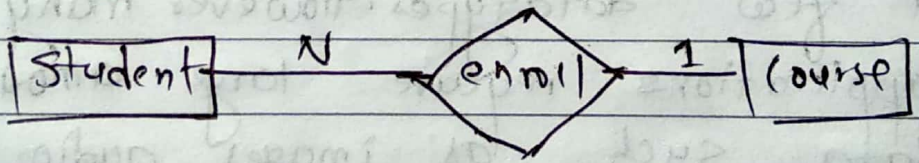
1: One-to-One



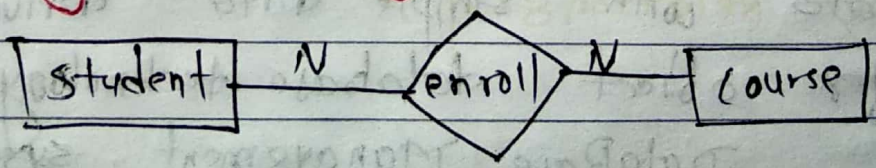
2: One-to-Many



3: Many-to-one

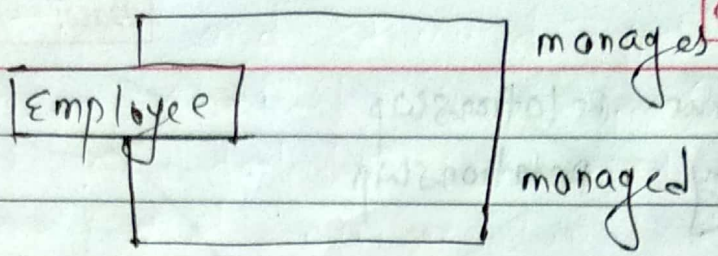


4: Many-to-many



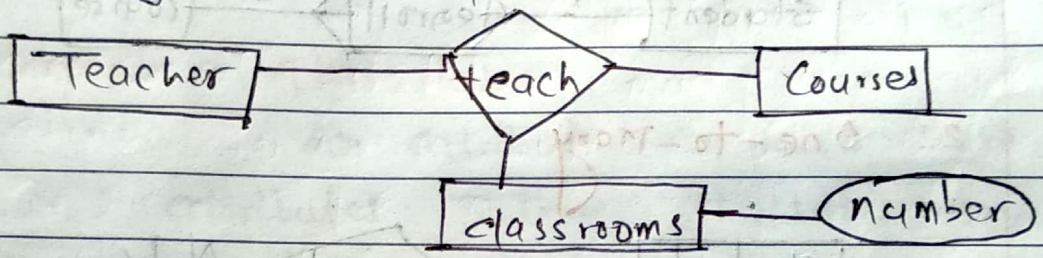
Recursive Relationship :-

When an entity is released with itself it is known as Recursive Relationship manages.



* Ternary / multiway Relationship :-

Relationship of degree three is Ternary relationship.



Motivation for Complex Data Types :-

Most relational DBMs supports only a few datatypes. However many business applications require large amount of complex data such as images, audio & video.

The need to integrate complex data with simple data derives the demand for object database technology.

DataBase management systems traditionally dealt with simple tabular data. In recent years, object-relational database systems were designed to support complex datatypes.

The datatypes that ORDBMS are

Page :
Date: / /

intended to support are Images, videos, textual objects are consider an example; to represent phone numbers, since people may have more than one phone it is multivalued attribute and is natural attribute.

The alternative of normalization by creating a new relation is expensive & artificial for this example.

With complex type systems we can model E-R model concepts, such as composite attribute multivalued attributes, generalization and specialization directly, without a complex translation to the relational model.

User Defined Abstract Data Types and structured Types:

Page :
Date: / /

Abstract Data Types (ADT) are user defined datatypes. Unlike traditional database management system which only provide for primitive data types such as INTEGER and CHARACTER, the object oriented programming language allow for the creation of abstract data types.

The data types offered in commercial database management system, CHAR, INTEGER, NUMBER, VARCHAR, BIT are sufficient for most relational database applicationns but ability to create and user defined data types can greatly simplify their database design.

In recent times, demand has grown for ways to deal with more complex data types. Consider for example: address.

While an entire address could be viewed as an atomic data item of type string, this view would hide details such as the street address, city, state and postal code, which could be of interest to queries.

On the other hand, if an address were represented by breaking into compo-

Page :
Date: / /

nents (street, adress, city, state and postal code), writing queries would be more complicated since they would have to mention each field. A better alternative is to allow structured data types that allow a type address with subparts.

Street address, city, state & postal-name

* Subclasses, Superclasses & Inheritence:

Superclass & subclass:

An entity type is used to represent both a type of entity and the entity set or collection of entities of that type that exist in the databases.

for eg:- The entities that are members of the EMPLOYEE entity type may be distinguished further into SECRETARY, ENGINEER, MANAGER, SALARIED_EMPLOYEE, HOURLY_EMPLOYEE and so on.

The set of entities in each of the latter groupings is a subset of the entities that belong to the EMPLOYEE entity set, meaning that every entity that is a member of one of these subgroupings is also an employee.

Page :
Date: / /

we call each of these subgroupings a subclass or subtype of the EMPLOYEE entity type, and the EMPLOYEE entity type is called the superclass or supertype of each of these subclasses.

Inheritence :- An entity cannot exist in the database merely being a member of a subclass. It also must be a member of the superclass

eg:- A salaried employee who is also an engineer belongs to the two subclasses ENGINEER and SALARIED_EMPLOYEE of the EMPLOYEE entity type. However, it is not necessary that every entity in a superclass is a member of some subclass.

An important concept associated with subclasses is inheritence. We say that an entity that is member of a subclass inherits all the entity that is member of a subclass inherits all the attributes of the entity as a member of the superclass. The entity also inherits all the attributes of the entity as a member of

Page :
Date: / /

the superclass. The entity also inherits all the relationships in which the superclass participates.

* **Specialization and Generalization!**

As the complexity of data increased, it becomes inefficient to use the traditional ER model for database modeling.

Hence some improvements or enhancements were made to the existing ER model to make it able to handle the complex applications better.

So, as part of the Enhanced ER model, along with other improvements, three new concepts were added to the existing ER model they are!

1) Generalization
2) Specialization
3) Aggregation

Generalization!- It is a bottom-up approach in which two lower level entities combine to form a higher level entity.

It is a process of extracting shared

characteristics from two or more classes and combining them into generalized superclass. shared characteristics can be attributes, associations or methods, i.e, sub classes are combined to form a superclass.

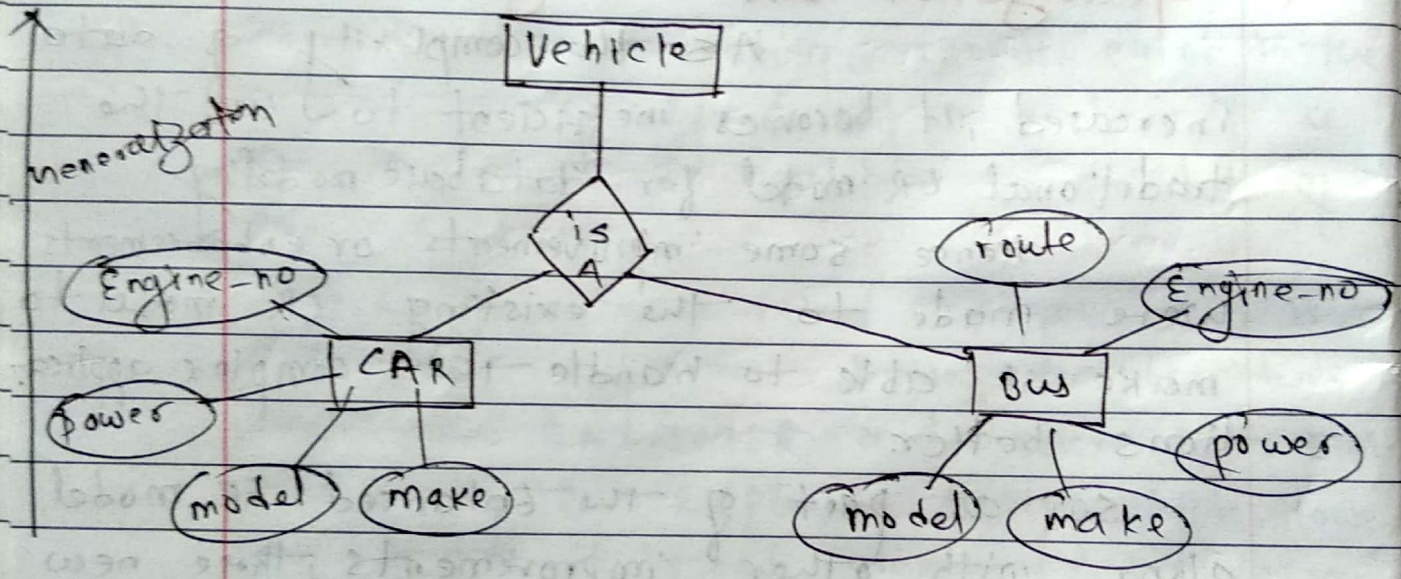


fig: Generalization

In the fig above the subclasses CAR & BUS share partially same attributes, During generalization, the shared attributes are used to create a new superclass vehicle with partially common attribute.

- Engine-no
- power
- model
- make

Page :
Date: / /

Specialization :- In contrast to generalization specialization means creating new subclasses from an existing system.

Opposite to generalization, it is a top-down approach in which one higher level entity can be broken down into lower level entity. If we can find out that certain attributes, associations, or methods only apply to some of the objects of the class, a subclass can be created.

The most inclusive class in generalization / specialization is called the superclass and is generally located at the top of the diagram. The more specific classes are called subclasses and are generally placed below the superclass.

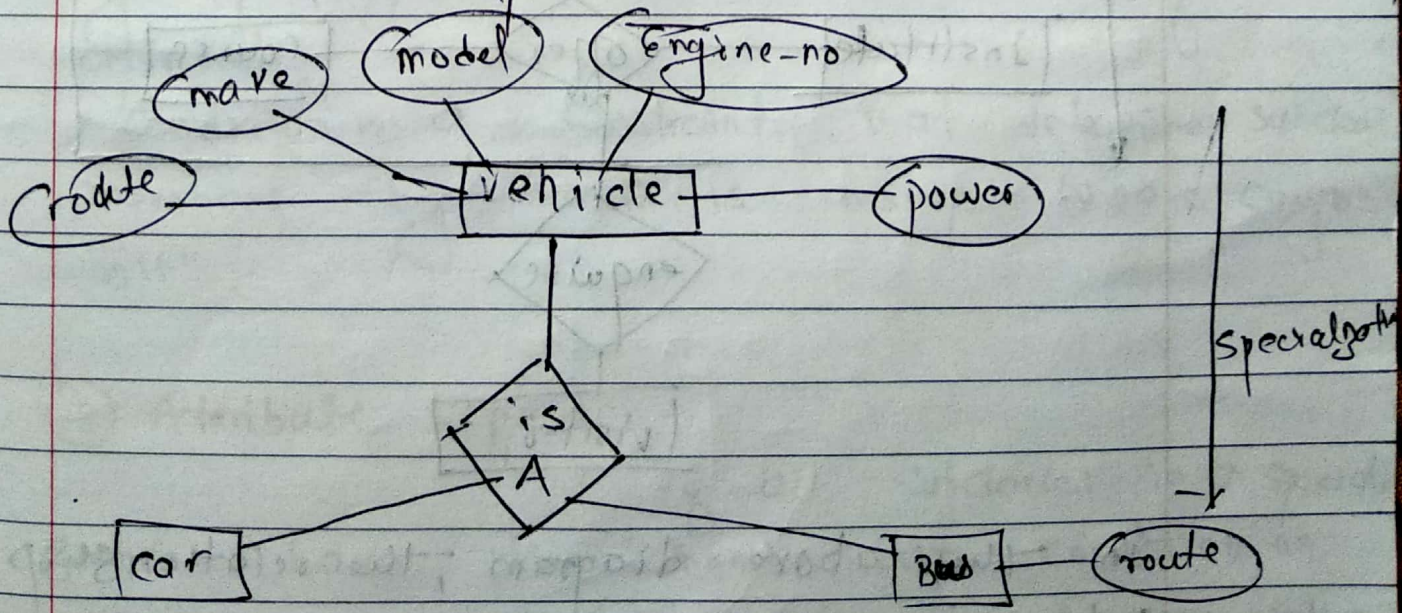


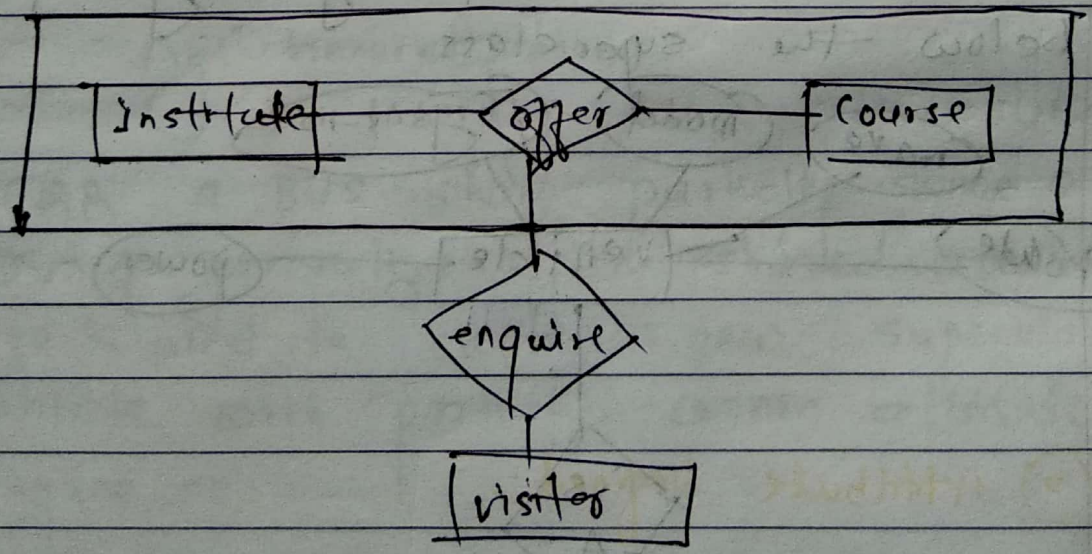
fig :- Specialization

Page :
Date: / /

The class vehicle has the attribute 'route' which is needed only for 'Bus' but not for 'car'.

Obviously, here two similar but diff^n domain concepts are combined into one class. Through specialization special cases of vehicles are formed : 'car' & 'bus'. The attribute 'route' is placed where it belongs ie, in 'bus'. The attributes of the class vehicle apply to both bus & car.

Aggregation:- Aggregation is a process when relation between two entities is treated as a single entity.



In the above diagram, the relationship between Institute & course together is acting

Page :
Date: / /

as an entity, which is in relationship with another entity visitor. Now in real world, if a visitor or a student visits an institute, he/she will never only enquire about institute only or course only, rather he/she enquire both.

* Constraints and Characteristics of Specialization & Generalization:

There are three constraints that may apply to a specialization/generalization.

(1). Membership constraints:-

(a) Condition / predicate Defined:- In some specialization, we can determine exactly the entities that will become members of each subclass by placing a condition on the value of some attribute.

- Condition is a constraint that determines subclass members Eg:- tanker is a ship where cargo = "oil".

(b) Attribute Defined:-

If all subclasses in a specialization have their membership condition on the same attribute of the superclass, the

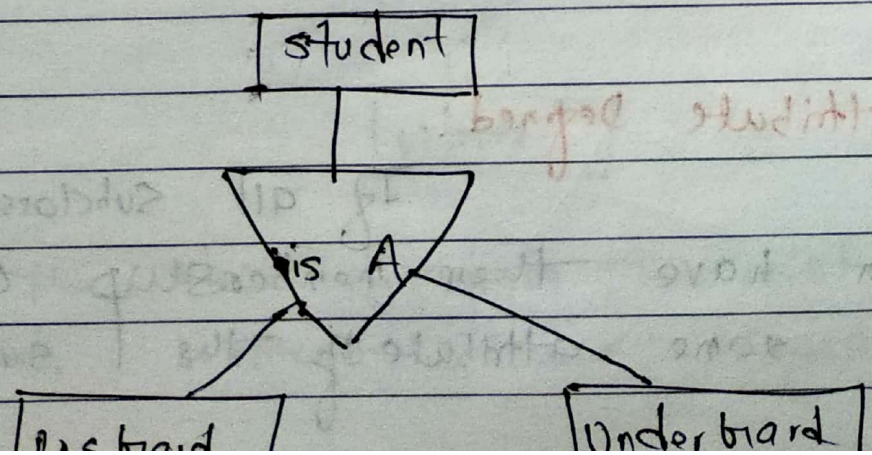
Page :
Date: / /

specialization itself is called an attribute-defined specification and the attribute is called the defining attribute.

(c) User defined :- When we don't have a condition for determining membership in a subclass the subclass is called user-defined. Membership in such a subclass is determined by the database users when they apply the operation to add an entity to the subclass.

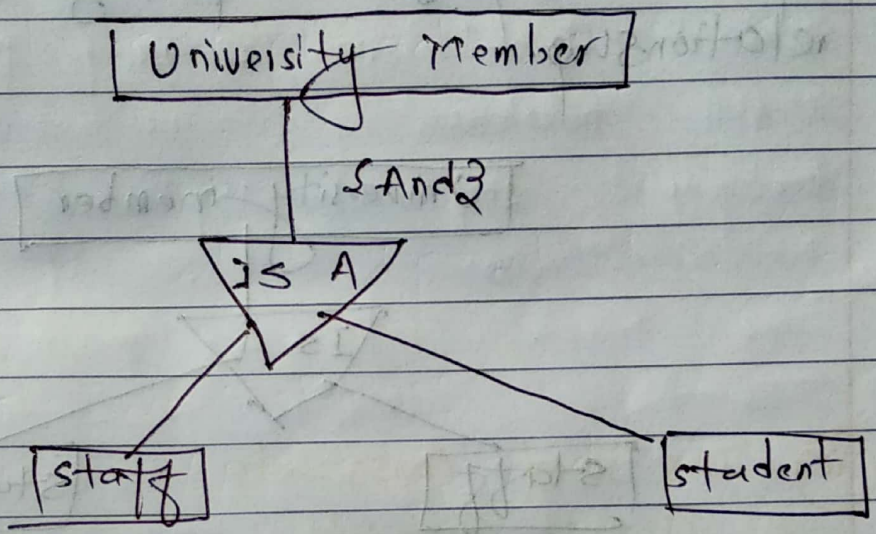
(ii) Dis Joint Constraints :-

a) Disjoint Constraints :- It specifies that the subclass of the specialization must be disjoint, this means that an entity can be member of at most one of the subclasses of the specialization 'or' is used to denote disjoint constraints.



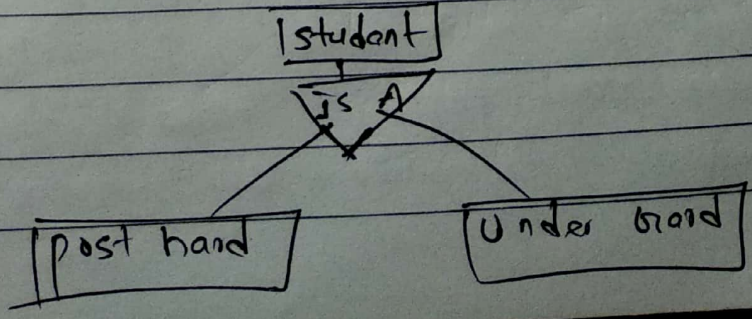
Page :
Date: / /

(b). Overlapping :- This applies when any entity can be a member of more than one subclasses of the specialization. {And} is used to represent the overlapping specialization/generalization relationship in the ER diagram.



(iii) Completeness / Totalness Constraint :-

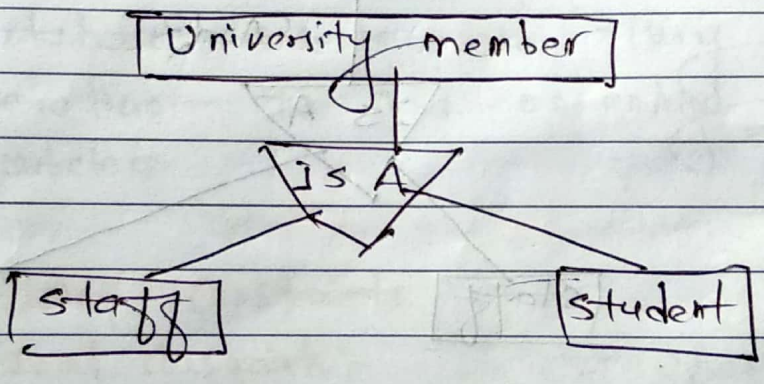
a) Total Specialization Constraint :- A total specialization constraint specifies that every entity in the super class must be the member of at least one subclass in the specialization. To represent completeness constraint, the keyword 'mandatory' is used.



Page :
Date: / /

(b) partial constraint :-

Some superclasses may not belong to subclasses Eg:- some people at university are neither student nor staff. The keyword 'optional' is used to represent a partial specialization / generalization relationship.



→ Relationship types of degree higher than two
Relational database design by EER to relational mapping.

Unit-3

Page: / /

Emerging Data Base Management System Technologies:

- Object oriented database concepts.
- Object Relational database concepts.
- Active database concepts
- Temporal database concepts.
- Spatial database concepts & Architecture.
- Deductive databases & query processing
- Mobile Databases
- Geographic Information System.

069/3

Page :
Date: / /

What is OID? How persistent objects are maintainance in OO Database?

⇒ An object oriented (OO) database system provides a unique identity to each independent object stored in the database. This unique identity is typically implemented via a unique, system-generated object identifier (OID).

The value of an OID is not visible to the external user but it is used internally by the system to identify each object uniquely and to create and manage inter-object references.

The main property required of an OID is that it be immutable; i.e. the OID value of a particular object should not change. This preserves the identity of the real-world object being represented. Hence, an OO database system must have some mechanism for generating OIDs and preserving the immutability property. It is also desirable that each OID be used only once; that is even if an object is removed from the database, its OID should not be assigned to another object.

Those two properties imply that the OID should not depend on any attribute values of

the object, since the value of an attribute may be changed or corrected.

An object is typically created by some executing application program by invoking the object constructor operation. Not all objects are meant to be stored permanently in the database. Transient object exists in the executing program and disappear once the program terminates while the persistent objects are stored in the database and persist after program termination. The typical approach to maintain persistent objects are naming and reachability. The naming mechanism involves giving an object an unique persistent name through which it can be retrieved by this or any other program. This persistent object name can be given via a specific statement or operation in the program.

All the names given to the objects must be unique with a particular database. Hence the named persistent objects are used as entry points to the database through which users and applications

Page :
Date: / /

can steal their database access. Obviously it is not practical to give names to all the objects in a large database. So, we we a second mechanism works by making the object reachable from some persistent object

An object B is said to be reachable from object A if the object B references object A in any way.

If we create a named persistent object N, with some class C then we can make objects of C persistent by adding them to the set or list, making them reachable from N. Hence, N defines a persistent collection of objects of class C.

Q7)2: Distinguish between multiple inheritance & selective inheritance OO concepts.

(i) Multiple inheritance is a type hierarchy occurs when a certain subtype T is a subtype of two or more types and hence inherits the functions [attributes and methods] of both supertypes.

Selective Inheritance occurs when a subtype inherits only some of the functions of a super-type. In this case an except clause may be used to list the functions in a supertype that are not to be inherited by subtype.

(ii) In single inheritance a derived class inherits a single base class while in multiple inheritance a derived class may inherit two or more base class.

(iii) Eg:- We may create a subtype ENGINEERING MANAGER that is a subtype of MANAGER and ENGINEER.

We may create subtype TEACHER, HOD from Subtype STAFF - UNIVERSITY = selective inheritance

Date: / /

Multiple Inheritence and selective Inheritence

→ Multiple inheritence in a type hierarchy occurs when a certain subType T of two or more types inherits the functions of both supertypes.

→ e.g. we may create a subtype ENGINEERING MANAGER that is a subtype of both MANAGER & ENGINEER.

→ One problem that may occur in multiple inheritance is that subtypes from which the subtype inherits may have distinct functions of the same name creating an ambiguity.

eg:- Both the MANAGER and ENGINEER may have a function called salary that may create ambiguity.

→ The general rule is that if a function is inherited only once. In such case, there is no ambiguity; the problem only arises if functions are distinct in two supertypes.

→ There are several techniques to deal with ambiguity.

The first solution is to have the system check for ambiguity when the subtype is created and let the user explicitly choose the function.

Page :
Date: / /

→ Another solution is to use some system default.

→ Third solution is to allow multiple inheritance if name ambiguity occurs instead forcing the user to change the names.

→ Selective inheritance occurs when a subtype inherits only some of the functions of supertypes. Other functions are not inherited. In this case an EXCEPT clause may be used to list the functions in a supertype that are not to be inherited by the subtype.

→ The mechanism of selective inheritance is not typically allowed in OODBMS but it is used more frequently in AI applications.

Page :
Date :

Object Relational Database Concepts (ORDBMS)

→ An ORDBMS is a database management system that is similar to relational database, except that it has an object-oriented database model. The ORDBMS supports objects, classes & inheritance in database schemas & query languages.

→ ORDBMS provide a middle between relational and object oriented databases.

In an ORDBMS, data is manipulated using queries in a query language. These systems bridge the gap between conceptual data modelling techniques such as entity relationship diagram and object relational mapping using classes and inheritance

→ ORDBMS also support data model extensions with custom datatypes and methods. This allows developers to raise the abstraction levels at which problem domains are viewed.

→ An ORDBMS is a combination of Relational and object-oriented database with the simplicity of relational database and extensibility of object oriented databases.

→ ORDBMS are the systems that attempt to extend relational databases systems with the functionality necessary to support a broader class of

application & in many ways provide a bridge between the relational & object-oriented paradigms.

* Advantages of ORDBMS:-

a) Reuse and sharing :-

→ The main advantages of ORDBMS is the ability to extend the DBMS server to perform standard functionality centrally, rather than have it coded in each applications.

b) Increased Productivity:-

→ ORDBMS provides increased productivity both for the developer & the end user.

c) Preserve Current Database functionality:-

→ The use of ORDBMS allows organizations to take advantage of the new extensions in an evolutionary way without losing the benefits of current database. features & functions.

d) Improved Performance:-

→ Using ORDBMS increases the ability to sort and locate files in the database faster. It also resolves many known weakness of RDBMS.

Page :
Date: / /

- Disadvantages of ORDBMS:-

(i) Complexity:-
The ORDBMS is complex by any means while compared with pure object oriented or relational model.

(ii) Increased Costs:-
The implementation and operation cost is increased as ORDBMS combines multiple DBMS together.

(iii) Simplicity & purity of Relational model are lost:
→ Since, ORDBMS supports data model extensions, the ease of using relational model are lost.

(iv) Object Applications are simply not as data centric as relational-based ones.

Implemention issues with ORDBMS

→ The ~~ORDBMS~~ obivious problem to consider in the context of an ORDBMS are:

- Object-Relational database design:- Object relational design is more complicated because we have to consider not only the underlying design considerations of application dependencies in the relational data model but also the object-oriented nature of the extended features.

- Storage and Access methods: The system must efficiently store ADT objects & structured objects and provide efficient indexed access to both. There is also complication with arrays.

- Query processing and optimization:- The use of ADTs and structured types call for new functionality in processing queries in ORDBMS.

To handle the new query processing functionality and optimizer must know

Date: / /

about the new functionality & use it appropriately

- Interaction of Rules with Transactions: Rule processing in RDBMS are implemented as triggers current RDBMS only implement immediate execution of triggers. A deferred execution of triggers involves additional processing.

- Variety of Data Types: Because of the variety in the datatypes in an ORDBMS and associated operations, efficient storage and access of the data is important.

The ORDBMS must dynamically link a user-defined function in it's address space only when it is required. Since numerous functions are required to operate on two-or-three dimensional spatial data, images text & so on. with static linking of all function libraries, the DBMS address space may increase by an order of magnitude.

Client-Server issues deal with the placement and activations of the functions. If the Server needs to executed

function, it is best to do so in the DBMS address space rather than remotely due to large amount of overhead.

Active DataBase Concepts:-

→ An active database is a database that includes an event-driven architecture, often in the form of the ECA rules, which can respond to conditions both inside and outside the database.

Here, the change in schema or data changes generate events that are monitored by active rules.

Active database management systems are involved by synchronous events generated by user or application programs as well as external asynchronous data change ~~that~~ event such as a change in sensor value or time.

The main distinctive feature is that an active database management systems support event monitoring. They store events in event history as an event type and time.

Date: / /

Generalized Model for Active DataBase & Oracle Triggers:-

The model that has been used for specifying active database rules is referred to as the Event-condition-Action or ECA model. A rule in the ECA model has three components.

(i) The Event(s) that trigger the rule:-

→ These events are usually database update operations that are explicitly applied to the database. However, in the general model, they could also be temporal events specified as a periodic time.

(Eg:- Trigger this rule every day at 5:30 am) or any other external events.

(ii) The condition that determines whether the Rule action should be executed:-

→ Once an event has been triggered, an optional condition may be evaluated. If no condition is specified, the action will be executed once that event occurs.

If a condition is specified, it is first evaluated and if the condition is true the action will be executed.

(ii) The Action to be Taken :-

→ The action is usually a sequence of SQL statements, but it could also be a database transactions or an external program that will be automatically executed.

Eg:- Let a simplified company database be:

EMPLOYEE:

Name, Ssn, Salary Dno Supervisor - ssn

DEPARTMENT:-

Dname Dno Total-sal manager-ssn

Here, Total-sal is a derived attribute whose value should be the sum of salaries of all employees who are assigned to particular department. Maintaining the correct value of such derived attribute can be done via a derived value.

The events that may change value of Total-sal are:-

(i) Adding new employees.
(ii) Changing salary of existing tuples.
(iii) Changing dept of employees
(iv) Deleting employees.

The Rules are:

Date: / /

R1: CREATE TRIGGER Total_sal1
AFTER INSERT ON EMPLOYEE
FOR EACH ROW
WHEN (NEW.Dno IS NOTNULL)
UPDATE DEPARTMENT
SET Total_sal = Total_sal + New_sal
Where Dno = New.Dno;

R2: CREATE TRIGGER Total_sal2
AFTER UPDATE OF SALARY ON EMPLOYEE
FOR EACH ROW
WHEN (NEW.Dno IS NOT NULL)
UPDATE DEPARTMENT
SET Total_sal = Total_sal + New.Salary
- old salary
WHERE Dno = NEW.Dno.

R3: CREATE TRIGGER Total_sal3
AFTER UPDATE OF Dno ON EMPLOYEE
FOR EACH ROW
BEGIN
UPDATE DEPARTMENT
SET Total_sal = Total_sal + NEW.
salary.
WHERE Dno = NEW.Dno

Page :
Date: / /

```
UPDATE DEPARTMENT
SET Total_sal = Total_sal - old.Sala
 WHERE Dno = OLD.Dno
END;
```

Q4:

```
CREATE TRIGGER Total_sal4
AFTER DELETE ON employee
FOR EACH ROW
 WHEN( OLD.Dno is NOT NULL)
   UPDATE DEPARTMENT
  SET Total_sal = Total_sal - OLD.
      Salary
  WHERE Dno = OLD.Dno.
```

Design and Implementation issues for Active DBs:

Date: / /

→ The first issue is concerned about activation deactivation & grouping of rules. In addition to creating rules, an active database system should allow users to activate, deactivate & drop rules by referring to their rules name. A deactivated rule will not be triggered by the triggering event. This feature allows users to selectively deactivate rules for certain periods of time when they are not needed, the activate command will make the rule active again. The drop command deletes the rule from the system.

→ The second issues concerns whether the triggered action should be executed before, after or concurrently with the triggering event.

→ A related issues is whether the action being executed should be considered as a separate transaction or whether it should be part of the same transaction that triggered the rule. let us assume that the triggering event occurs as part of transaction execution. There are three main possibilities for rule consideration.

Date: / /

(i) Immediate Consideration :-

→ The condition is evaluated as part of the same transaction as triggering event and is evaluated immediately. This case can be further categorized into three options:.

- Evaluate the condition before executing the triggering event.
- Evaluate the condition after executing the triggering event.
- Evaluate the condition instead of executing the triggering event.

(ii) Deferred Considerations :-

→ The condition is evaluated at the end of the transaction that included the triggering event. In this case there could be many triggered rules waiting to have their conditions evaluated.

(iii) Detached Considerations :-

→ The condition is evaluated as a separate transaction, spawned from the triggering transactions.

Date: / /

- The next set of options concern the relationship between evaluating the rule condition and executing the rule action. In this case also three options are possible:- Immediate, deferred & detached execution.
- Another issue that have limited the widespread use of active rules, inspite of their potential to simplify database & s/w development is that, there are no easy-to use techniques for designing, writing & verifying rules.

- **Potential Applications for Active DataBases:**
→ One important application of Active database is to allow notification of certain conditions that occur. for eg:- an active db may be used to monitor the temp of an industrial furnace.

The application can periodically insert the temperature reading records directly from the temperature sensors and active rules can be used to trigger an action whenever a temperature record is inserted and temp exceeds the danger level.

Date: / /

→ Active rules can also be used to enforce integrity constraints by specifying the types of events that may cause the constraints to be violated & than evaluating appropriate conditions that check whether the constraints are actually violated by the event or not. Here complex application constraints, often known as business rules may be enforced that way.

Eg.- in the 'university' database application one rule may monitor the grade point average of students whenever new grade is entered and it may alert HOD if GPA of student falls below certain threshold; another rule may check that course prerequisites are satisfied before allowing a student to enroll in a course & so on.

→ Other applications include the automatic maintainance of derived data. A similar application is to use active rules to maintain the consistency of materialized views

Date: / /

wherever the base relations are modified.
This application is also relevant to the new data warehousing technologies.
→ other possible applications include security monitoring, alerting, statistics gathering and authorization.

- **Temporal Data Bases Concepts :-**

→ A Temporal database is generally understood as a database capable of supporting storage & reasoning of time-based data.
A temporal database stores data relating to time instances. It offers temporal data types and stores information relating to past, present & future time.

→ The temporal database has two major notions or attributes (time dimensions).
(i) valid time
(ii) Transaction time.
These components can be combined to form a bitemporal data.
→ There are many examples of applications where same aspect of time is needed to

maintain the information in a database. They include healthcare insurance, reservation systems, scientific research & soon where decisions solely depend upon the time.

eg:-
In 'University' database, time is associated with 'semester' & 'year' of each 'section' or 'course'.

→ valid time is the time period during which a fact is true in the real world.

→ Transaction time period is the time period during which a fact stored in the database was known.

⇒ Bitemporal data combines both valid time and transaction time.

- **Time Representation :-**

→ for Temporal databases is considered to be an ordered sequence of points in some granularity that is determined by the application.

eg:- Consider that some temporal application never requires time units that are

less than one second. Then each time point represents one second in time using this granularity.

→ In reality, each second is a short time duration, not a point, since it may be further divided into multiseconds & so on.

→ The term 'chronon' defines the shortest duration of time supported, the purpose of choosing chronon is that, events occurring within the same second will be considered to be simultaneous events.

→ Since there is no known begining or ending of time, one needs a reference point from which to measure specific time points. A calender organizes time into different time units for convenience.

→ In SQL2, the temporal data types include DATE (specifying year, month and day as YYYY-MM-DD) TIME (specifing Hour, minute and second as HH:MM:SS).

TIMESTAMP (specifying date/Time combinations, with options for sub-second division if needed).

Page :
Date: / /

INTERVAL (a relative time duration such as 10 days or 250 minutes)

PERIOD (a time duration with fixed starting point, such as 10 day period from Jan 1 1991).

Date: / /

Time Dimensions:-

→ Time Dimensions are generally classified into valid-time and transaction-time dimensions. Both dimensions can be defined separately and hence they are orthogonal.

→ valid Time:- Any particular event or fact that is associated with a particular time point or time period in the database may be interpreted to mean different things.

The most natural interpretation is that the associated time is the time that the event occured or the fact was true in the real world. If this interpretation is used, the associated time is often referred to as the valid time. A temporal database using this interpretation is called a valid time database.

→ Transaction Time:- A different interpretation can be made, where the associated time refers to the time when the info was actually stored in the database i.e. it is the value of the system time clock when the information is valid in the system.

Page :
Date: / /

In this case, the associated time is called the transaction time. A Temporal database using this information is called a transaction time database.

→ Bi Temporal ~~Data Base~~ Relation :- Some database applications may require both valid time and transaction time in such case, the temporal database is called the Bitemporal database. It combines both the valid time and transaction time.

Page :
Date: / /

Spatial DataBase Concepts and ~~Architecture~~

→ A spatial database is a database that is enhanced to store and access data. A spatial data is defined in terms of geometric space.

→ These data are often associated with geographic locations and features like cities, Data on spatial databases are stored as coordinates, points, lines, polygons and topology. Some spatial databases handle more complex data like 3D-objects, topological coverage & linear n/w's.

→ Spatial databases keep track of objects in a multi-dimensional space. These databases are used in many applications such as environmental emergency & bottle management.

/* [Regular spatial database]

→ The main extensions that are need for spatial databases are models that can interpret spatial characteristics..

- The basic extensions needed are to include 2D ~~concepts~~ geometrical concepts, such as points, lines, circles, polygons and arcs in order to specify spatial characteristics of objects.

- Additionally, spatial operations are needed to create on the objects spatial characteristics.

E.g.:- To compute difference between objects

E.g.:- A database used for emergency management applications has static spatial characteristics such as street, highways, waterpumps, police station & dynamic spatial characteristics such as police vehicles, fire trucks & ambulances. Hence a spatial database must enforce all the spatial characteristics & spatial operations.

→ The Regular database systems use indexes for faster & more efficient search & access of data. This index, however is not fit for spatial queries.

Spatial database use a unique index called a spatial index to speed up database performance.

Spatial indexing is required because system should be able to retrieve data from a large collection of objects without searching the whole bunch. E.g.- Quadtree, octree, hrid

→ Aside from indexes, spatial databases also offer spatial datatypes in their data model and query language.

These data types are required to provide a fundamental abstraction and model the structure of the geometric objects with their corresponding relationship and operations in the spatial environment.

- **Deductive Database & Query Processing:**

→ A Deductive database is a database system that can conclude additional facts based on rules & facts stored in the database.

→ In a deductive database system, we typically specify rules through a declarative language, a language in which we specify what to achieve rather than how to achieve it.

→ An inference engine/ deduction mechanism within the system can deduce new facts from the database by interpreting these rules.

The model used for deductive db is closed, related to the field of logic programming and the prolog language.

→ A deductive database uses two main types of specification:- facts & rules

Page :
Date: / /

- facts are specified in a manner similar to the way relations are specified except that it is not necessary to include the attribute names.

Rules are somewhat similar to relational views. They specify virtual relations that can be formed from the facts by applying inference mechanisms based on the rule specifications.

Multimedia Database Concepts:-

Date: / /

- Multimedia database provides features that allow users to store and query different types of multimedia information such as:- images, videoclips, audioclips and documents.
- The main type of database queries involves locating multimedia sources that contain certain object of interest. for Eg:- One may also want to locate all the videoclips in a database that include a certain person say, ABC.
- The above type of queries are referred to as content-based retrieval, because the multimedia source is being retrieved based on its containing certain objects or activities. Hence a multimedia database must use some model to organize and index the multimedia sources based on their contents Identifying the contents of multimedia sources is different and time consuming task.

* There are two main approaches :- the first is based on automatic analysis of the multimedia sources to identify certain mathematical used different techniques depending on the type of multimedia source.

Date: / / /

The second approach depends on manual identification of the objects and activities of interest in each multimedia source and use this information to index the source.

Image	Video	Text	Audio
Stored in 2D representation form as pixel seq. of frames or cell values where each frame is a image		Alphanumeric characters	Recorded messages, songs, speeches, data transfer.

Mobile Databases:

→ A mobile database is a database that is transportable, portable and physically separate or detached from the corporate database server but have the capability to communicate with those servers from remote sites allowing the sharing of various kinds of data.

* Typical examples might include traffic police, taxi dispatchers and weather reporting services as well as financial market reporting and information brokering information.

* There are a number of HW as well as SW problems that must be resolved to fully utilize the capability of mobile computing. E.g. data management, transaction management & database recovery.

* Narrow bandwidth of wireless communication channel, short life of power supply of mobile units, changing location by required information are the major challenges in mobile database.

The mobile ~~company~~ computing architecture is a distributed architecture where a number of computers, generally referred to as fixed hosts (FS) and Base Section (BS), are interconnected through a high speed wired n/w.

→ fixed hosts are general purpose computers & donot manage mobile units but can be configured to do so.

Base stations are equipped with wireless interfaces and can communicate with mobile units to support data access.

→ Mobile units (hosts) & base stations communicate through wireless channels having bandwidths significantly lower than those of wired n/w.

→ A downlink channel is used for sending data from a BS to an MU and an uplink channel is used for sending data from an MU to BS.

→ Mobile units are battery-powered portable computers that move freely in a geographic mobility domain, an area that is restricted by the limited bandwidth of wireless communication channels.

→ for efficient mobility of units, the entire geographic mobility domain is divided into smaller domains called cells, this ensures that inter cell movement of mu will have no effect on data retrieval process.

→ The mobile computing platform can be effectively described under client-server paradigm which means we may sometimes refer a mobile unit as a client or sometimes as a user and the base station as servers.

→ client & servers communicate through wireless channels. The communication link between a client and a server may be modeled as multiple data channels or as a single channel.

* Characteristics of Mobile Environment :-

i) Rapid Data change :-

In mobile database environment, data generally changes very rapidly. Users often query servers to remain up-to-date. More specifically, they often want to track every broadcast for their data item of interest. E.g. Stock Market information, weather data, airline information.

ii) Random Usage :-

Since the users are mobile, they randomly enter and exit from cells. The average duration of a user's stay in the cell is referred to as a Residence Latency (RL), It is computed by users residence time in cells. Thus, each cell has an RL value. Servers maintain neither client arrival and departure patterns nor client-specific data request information.

(iii) Communication Latency :-

It is primarily due to data conversion into wireless, tracking and filtering of data on the receiver & transaction time.

iv) Limited Battery Life:-

The mobile devices may have to suffer with limited battery life. Applications with high power consumption may face some problems during operation.

v) Discontinuous Wireless Connectivity:-

Mobile stations may not always be connected to the stations. Since they may move out of geographic mobility domain.

vi) Performance Issues:-

An inactive device may not actively listen on the channel for data. This happens when a device tries to conserve energy.

- Data Management Issues:-

→ from data management stand point, mobile computing may considered a variation of distributed computing. Two possible scenario of distribution are:

i) The entire database is distributed mainly among the wired components possibly with full or partial replication. A BS manages its own db & may have additional functionality

to locate mobile units, query & transaction management.

ii) The database is distributed among wired and wireless components. Data management responsibility is shared among base stations & mobile units.

The additional issues are:

i) Data distribution & Replication:-

Data is unevenly distributed among the base station & mobile units. The consistency constraints generate the problem of cache management as they may be busy to process their own transaction & get disconnected.

ii) Transaction Models:-

A mobile transaction is executed sequentially through several stations & multiple data sets depending upon movement of mobile unit. central ~~department~~ coordination of transactions is lacking. Hence, traditional ACID property may need to be modified.

iii) Query Processing:-

Awareness of the data location & mobile unit is important & affects the cost/benefit analysis of query processing. The query

response needs to be retained to mobile units that may be in transit or cross cell boundaries as yet must receive correct result.

iv) Recovery & fault Tolerance :-
The mobile database environment must deal with site, media, transaction & communication failure.

v) Mobile DataBase Design :- The global name resolution problem for handling queries is compounded because of mobility & frequent shut down. Mobile database design must consider many issues of metadata management.

- Geographic Information System (GIS) :-

→ A GIS is a system designed to capture, store, manipulate analyze and present spatial or geographic data.
They describe the physical properties of the geographic world

→ The GIS broadly incorporate two types of data

1) Spatial data :- originating from maps, digital images, administrative & political boundaries roads, physical data such as rivers, climate regions, land elevation.

Page :
Date: / /

ii) Non-Spatial data : such as census counts, economic data and sales or marketing information that offers highly innovative approaches to meet some challenging technical demands.

GIS Applications :

→ It is possible to divide GIS application into three categories.

(i) Cartographic Applications :- In Cartographic & terrain modeling applications variations in spatial attribute are captured. for Eg:- soil characteristics, crop density, air quality etc,

(ii) Digital Terrain Modelling :- It involves the simulation of geographical activities such as landslide, avalanche, flood.

(iii) Geographic Object Application :- Here, objects of interest are identified from a physical domain such as :- power plants, electrical districts, product distribution

The first two categories of GIS applications require a field-based representation, where as the third category requires an object-based one. The cartographic approach

involves special functions that can include the overlapping of layers of maps to combine attributed data.

* Complex objects :-

→ They are built from simpler ones by applying constructors to them. The simplest objects are objects such as integers, characters, booleans and floats.

→ There are various complex object constructs. tuples, sets, bags, lists and arrays are examples.

- There are two main types of complex objects.

i) Unstructured Complex Object :-

The structure of these objects is not known to the DBMS. The unstructured data is not properly organized and this increase the compilation time of a program.

→ The unstructured complex object facility provided by a DBMS permits the storage and retrieval of large objects that are needed by the database application.

→ Only the application programs can interpret the objects. They are called unstructured in the sense that the DBMS does not know

Page :
Date: / /

what their structure is.

→ The objects are considered complex because they require a large area of storage & are not part of the standard data types provided by the traditional DBMS.

→ Since the object size is quite large, a DBMS may retrieve a portion of the object and provide it to the application program before the whole object is retrieved.

→ The DBMS may also use buffering & caching techniques to prefetch portions of the object before the application program needs to access them.

→ The DBMS doesn't have the capability to directly process selection conditions based on values of these objects unless the application program provide the code.

→ In OODBMS, this is done by defining an Abstract Data Type with operations for selection comparision etc.

→ This feature allows the ODBMS to have an extensible type system i.e, new types can be created & hence libraries of new types

Page :
Date: / /

(ii) Structured Complex Object :-

→ The object structure is defined & known to the DBMS, the structured data is organized in such a way that relational database can easily search the data.

→ The main difference is that the object structure is defined by repeated application of the type constructor provided by the OODBMS.

→ Since the database knows the object structure it is called structural complex object.

→ Two types of reference semantic exist between a complex object & its component.

- Ownership :-

→ The sub-objects are encapsulated with in a complex object and are considered a part of the complex object.

- Reference :-

→ The subobjects are independent objects but the same time may be considered a part of the complex object.

Unit-4

Page :
Date: / /

Data Mining:-

It is also refered to as knowledge discovery in databases. In computer science, it is the process of discovering useful patterns and relationships in large volumes of data.

→ Data mining combine tools from statistic and AI with database management to analyze large digital collection known as data sets.

→ In simple it refers to the process of extraction of interesting knowledge or patterns of knowledge from huge amount of data.

* Knowledge Discovery process / Data-mining process:-

i) Data Cleaning:- The first step is to remove the noise and inconsistent data from the data sets. It excludes all the unwanted data.

ii) Data Integration:- The multiple data sources that are of interest are integrated at this phase. The data cleaning and integration steps are preprocessing steps and the resulting data are stored in datawarehouse.

Page :
Date: / /

iii) Data Selection :- In this phase, the data relevant to the analysis task are retrieved from the database

iv) Data Transformation :- The selected data are transfered and consolidated into forms appropriate for mining by perfor-ming summary or aggregation operation. Sometimes, data transfor-mation & consolidation are performed before the data selection process particularly in case of data warehousing

v) Data Mining :- This is the main phase where various appropriate statistical & intelligent methods are applied to extract the required data patterns.

vi) Pattern Evaluation :- The obtained data pattern are evaluated to identify the pattern of interest representing knowledge

vii) Knowledge Representation :- The final step is to use visualization & knowledge representation techniques to present the mined

Page :
Date: / /

knowledge to the users. The steps 1 to 4 are different forms of data preprocessing, where data are prepared for mining.

The data mining step may interact with the user or a knowledge base.

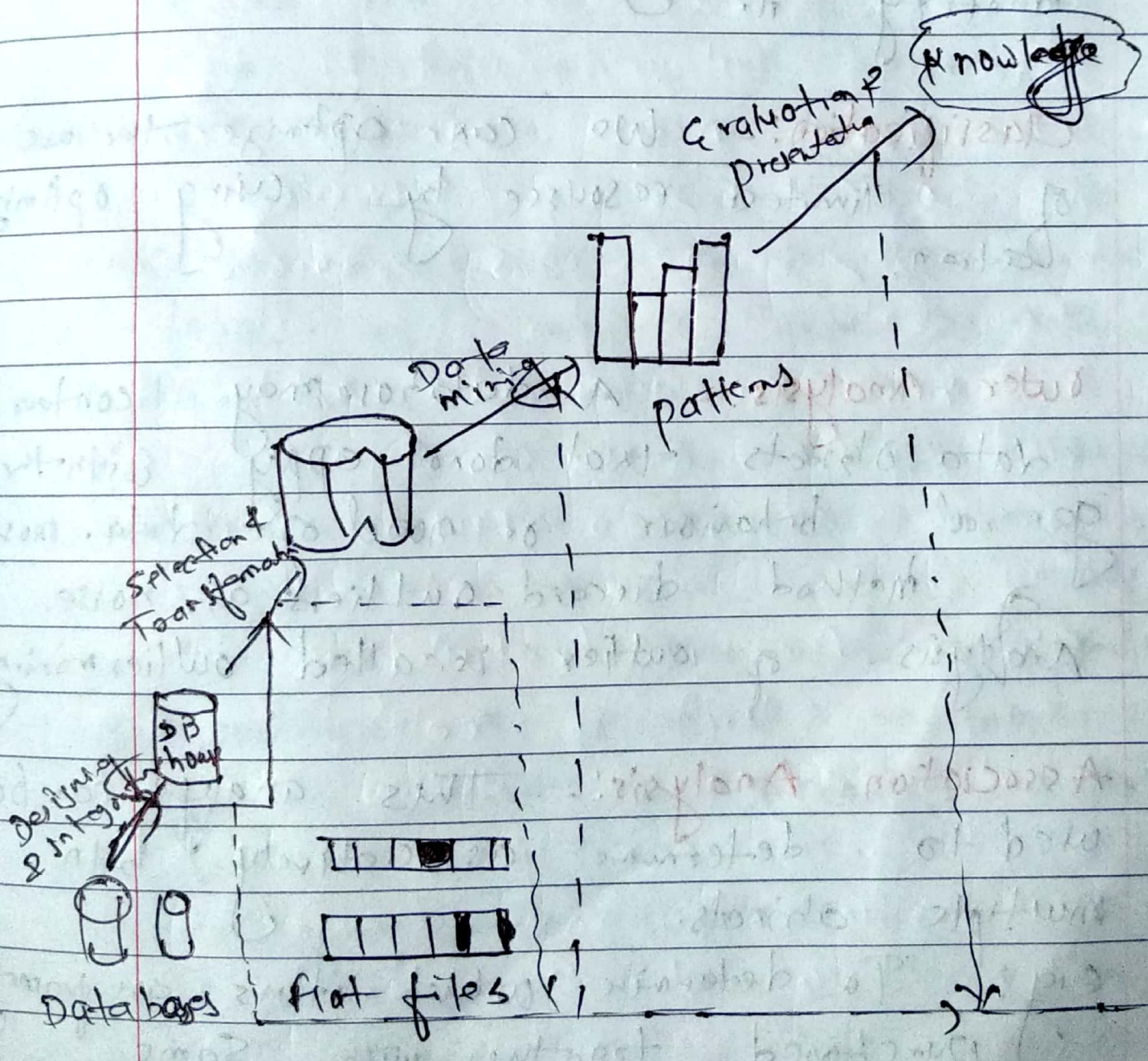


fig:- Data Mining in knowledge process discovery

Page :
Date: / /

* Goals of Data mining and Knowledge Discovery

- Prediction :- Data mining helps us to predict how certain attributes/events will behave in the future.

- Identifications :- We can use the data mining to determine the existence of an item, event or activity at certain period & identify it.

- Classification :- We can optimize the use of limited resource by using optimization.

- Outer Analysis :- A database may contain data objects that don't apply with the general behaviour or model of data. Most method discard outliers as noise. Analysis of outlier is called outlier mining.

- Association Analysis :- This analysis can be used to determine associatively betn multiple objects.
  Eg :- To determine which items are frequently purchased together with some transaction.

buys (x, "computer") ⇒ buys (x, "software")
(support = 1%, confidence = 50%)

- Classification & prediction:-

There are two forms of data analysis that can be used for extracting models, describing important classes or to predict future data trends. The two forms are:-

- classification
- prediction.

Classification models predict categorical class labels; and prediction model predict continuous valued functions.

for eg:- We can build a classification model to categorize bank loans applications as either safe or risky.

A prediction model to predict the expenditures of potential customers on computer equipment given their income & occupation.

Classification is facilated by constructing a model or classifier to predict the categories is used for classification.

Page :
Date: / /

The constructed model can be presented in many forms such as:

- If - THEN rules.
- Decision tree
- Neural Networks

Eg:- A credit card company to determine which prospects should receive a credit card offer.

Training Data:-

~~Age~~ Name	Age	Income	Card Offer
ABC	28	RS:14,00,000	Yes
XYZ	72	RS 2,30,000	NO

Then If - THEN Rules can be developed as: If (Age >18 OR Age <75) AND Income > 10,00,000 THEN Card offer = 'Yes'

- **Cluster Analysis / clustering:-**

Clustering is the task of grouping a set of objects in such a way that objects in a same group (cluster) are more similar, in some sense to each other than to those in other groups.

Page :
Date: / /

→ Cluster is a group of objects that belongs to the same class clustering is the process of making a group of abstract object into of classes of similar objects.

→ While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups.

→ The main advantages of clustering over classification is that it is adaptable to changes and helps single out useful features that distinguish different groups.

→ Clustering analysis is broadly used in many applications such as market research, pattern recognization, data analysis & image processing

→ clustering also helps in classifying documents on the web for information discovery. It can also be used for outlier detection application such as detection of credit card fraud.

→ Clustering methods can be classified into following categories:-

Partitioning Method
Hierarchical "
Density - Based "
Grid Based method & so on

Page :
Date: / /

Data Warehousing :-

Data warehousing is the process constructing and using a data warehouse. A data warehouse is constructed by integrating data from multiple heterogenious sources that support analytical reporting & decision making.

→ A datawarehouse is a database, which is kept separate from the organizations operational db. There is no frequent updating done in data warehouse.

→ A data warehouse processes consolidated historical data, which helps the organization to analyze its business.

→ A data warehouse helps executives to organize, understand and use their data to take strategic decisions. The decision support technologies are used to help utilize the data available in a data warehouse.

→ Any organization can gather historical data, analyze it and take decisions based on the info present in the warehouse.

→ A datawarehouse is a subject-oriented integrated-time variant and non-volatile collection of data in support of mgmts decision making process.

Page :
Date: / /

• Data Warehouse features :-

1) Subject-oriented :- A datawarehouse can be used to analyse a particular subject area. It focuses on the major subjects, such as customer, product and sales.
It focuses on modelling and analysis of data for decision makers not on daily operation or transaction processing.

ii) Integrated :- A data warehouse integrates data from multiple data sources. The data sources are heterogenious.
The data cleaning method is applied prior to data integration phase. If any two sources A & B may have multiple ways of identifying a product, however in a datawarehouse there will be only a single way of identifying a product.

iii) Time-variant :- The time limit for the data warehouse is significantly longer than that of operational system.
for eg :- one can retrieve data from months and years from the data warehouse. It differs from the operational data in the sense that operational data only stores most recent data.

Page :
Date: / /

iv. Non Volatile:- Once data is stored in the data warehouse the data remains fixed. the data can be retrieved multiple times but will never be altered.

A datawarehouse is separated from the operational environment the operational environment and the operational update of data doesnot occur in data warehouse environment.

A datawarehouse is a repository of information created from multiple sources stored under a unified schema.

* Typical functionality of a Data Warehouse:-

A data warehouse helps business executives to organize, analyze and use their data for decision making. Data warehousing is widely used in the following fields.

- financial services
- Banking services
- Customer needs
- Retail sectors
- Controlled manufacturing

The functions of data warehouse tools

and utilities are.

- **Data Extraction:-** This step involves gathering data from multiple heterogeneous sources. Then the data cleaning process may be applied to the gathered data to obtain extracted data.

- **Data Cleaning:-** It is the process of finding the desired appropriate data & correcting the errors in data.

- **Data Transformation:-** It involves converting the cleaned data from legacy format to the warehouse format.
  The warehouse format is the format in which the data is stored in the warehouse.

- **Data loading:-** It is the process of loading the warehouse data into datawarehouse. It involves various steps such as sorting, summarizing, consolidating, checking integrity and building indexes & partitions.

- **Refreshing:-** It involves updating from the

Page :
Date: / /

data sources to data warehouse.

A data warehouse maintains its functions in three layers.

Layer 1: Staging:- This layer is used to store raw data for use by developers.

Layer 2: Integration: The integration layer is used to integrate data and to have a level of abstraction from users.

layer 3: Access: The Access layer is for getting data out for users.

	DLAP DataWareHouse (Online Analytical process)	OLTP Operational DataBase (online Transaction process)
1)	It involves historical processing of information.	It involves day to day transaction (process).
2)	OLAP systems are used by knowledge workers such as executives, managers & analysts.	OLTP systems are used by clerks, DBAs or database professionals.

3.	It is used to analyze the buiness.	It is used to run the buiness.
4.	It focuses on information out.	It focuses on Data in.
5.	It is based on start schema, slw schema and flat constellation schema.	It is based on entity relationship models.
6.	It contains historical data.	It contains current data.
7.	It provides Summarized and consolidated data.	It provides primitive and highly detailed data.
8.	It provides Summarized and multidimensional view of data.	It provides detailed & flat relational view of data.
9.	The database size is very high.	The database size is low compared to that of OLAP.
10.	These are highly flexible.	It provides high performance.

Page :
Date: / /

11. There are limited users in terms of records accessed are in millions.

The number of users is in thousands and records accessed is in tens.

Unit - 5

(Old Question only)

Page :
Date: / /

069/4 : XML & HTML → explain [Kamal sir note]

069/7 : Attributes vs element in XML
attributes used in specifying elements in XML Schema. [Kamal sir].

070/1 : XPath, XQuery

070/6: Write a schema that provides tags for person's first name, last name, weight & shoe size, weight & shoe size tags should have attributes to designate measuring system.

071/10. XML schema Vs XML DTD [K.S].

072/6. Object Relational features included in SQL-99

072/8: Difference & Similarity between object & literal in OMG object model.

072/10: Explain XML Schema & XML DTD [K.S]

073/6. What is OR database. OR features of SQL.

Page :
Date: / /

* Path :-

→ * path address parts of an xml document by means of path expression. A path expression in xpath is a sequence of location steps separated by "/" instead of the "->" operator that separates location steps in SQL.

- The result of a path expression is a set of nodes. The xpath expression /university-3/instructor/name may return these elements.

<name> Ram </name>
<name> Shyama </name>

- The text() option can be used to discard the enclosing tags.
  e.g. /university-3/instructor/name/text()

- path expression are evaluated from left to right like a directory Hierarchy the initial '/' indicates the root of the document.

Page :
Date: / /

- The selection predicates can be used in xpath are contained in square brackets for eg,
  /University-3/ course [credits >=4]
  { Also refer to Kamal Sir note }

* XQuery :-
  XQuery is a language for finding and extracting elements and attributes from XML documents both the XQuery & xpath share the same data model & support same function and operation.

→ XQuery can be used to
- Extract information to use in a web service.
- Generate summary reports.
- Transform XML data to XHTML
- Search web document for relevant information.

→ XQuery queries are modeled after SQL queries but differ significantly from SQL. They are organized into 5 sections. for, let, where, order by ,

Page :
Date: / /

return (FLWOR)

a sample FLWOR expression is

```
for     $x  in /university-3/ course
Let     $course_id = $x/@course-id
Where   $x/credits > 3
return  <course-id> { $course_id }
                    </course_id>
```

Page :
Date: / /

* Objects & literals in ODMG:

→ Object and literals both are basic building blocks of the object model. However there are many similarities and differences between them.

→ The main difference between the two is that an object has both an object identifier and a state (current value) where as a literal has only a value but no object identifier.

→ In both cases the value ~~have~~ can have a complex structure the object state can change over time by modifying the object value. A literal is basically a constant value possibly having a complex structure that doesn't change.

→ A object is described by four characteristics:

i) Identifier ii) name
iii) Lifetime iv) structure.

The object identifier is a unique system wide identifier (Object-ID).

Page :
Date: / /

The name is an optional unique name for each object with in a particular database. It can be used to locate object in the database.

Any user also can locate various other objects that are referenced from the object by using the name.

- The life time of an object indicate, that either the object is transient or persistent object.

- The structure of an object specifies how the object is constructed by using the type constructor the structor specifies whether an object is atomic or a collection object.

- In the object model, a literal is a value that doesnot have an object identifier however the value may be a simple or complex structure.

- There are three types of literals.
  i) atomic
  ii) Collection
  iii) structured.

Page :
Date: / /

Atomic literals correspond to the values of basic datatypes and are predefined. The basic datatypes of the object model include long, short and unsigned integer, floats, Boolean, Char, string & so on. Structured literals correspond roughly to values that are constructed using the tuple constructor. They include Date, Interval, Time & Time stamp as built in structures, as well as any additional user defined type structure as needed by each application.