# Unit-1
# INTRODUCTION TO DBMS

**1. What is first normal form?**

The domain of attribute must include only atomic (simple, indivisible) values.

**2. What is 2NF?**

Relation schema R is in 2NF if it is in 1NF and every non-prime attribute An in R is fully functionally dependent on primary key.

**3. Define functional dependency.**

Functional dependency is a constraint between two sets of attributes from the database.

**4. What is a data model? List the types of data models used?(April/May 2011, 2012)**

A data model is a collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints.

E-R Model, Network Model, Object Oriented Data model

**5. Define full functional dependency.**

The removal of any attribute A from X means that the dependency does not hold any more.

**6. Explain about partial functional dependency?**

X and Y are attributes. Attribute Y is partially dependent on the attribute X only if it is dependent on a sub-set of attribute X.

**7. What you meant by transitive functional dependency?**

**Transitive dependency** is a functional dependency which holds by virtue of transitivity. A transitive dependency can occur only in a relation that has three or more attributes. Let A, B, and C designates three distinct attributes (or distinct collections of attributes) in the relation.

Suppose all three of the following conditions hold:

1. A→ B
2. It is not the case that B→A
3. B→C

Then the functional dependency A→C(Which follows from 1 and 3 by the axiom of transitivity) is a transitive dependency.

**8. What is meant by domain key normal form?**

Domain/key normal form (DKNF) is a normal form used in database normalization which requires that the database contains no constraints other than domain constraints and key constraints.

**9. Define database management system?**

Database management system (DBMS) is a collection of interrelated data and a set of programs to access those data.

**10. List any five applications of DBMS.**

Banking, Airlines, Universities, Credit card transactions, Tele communication, Finance, Sales, Manufacturing, Human resources

**11. What are the disadvantages of file processing system?**

The disadvantages of file processing systems are

a) Data redundancy and inconsistency
b) Difficulty in accessing data
c) Data isolation
d) Integrity problems
e) Atomicity problems
f) Concurrent access anomalies

**12. Define instance and schema?**

**Instance:** Collection of data stored in the data base at a particular moment is called an Instance of the database.

**Schema:** The overall design of the data base is called the data base schema.

**13. Define data model?**

A data model is a collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints.

**14. What is an entity relationship model?**

The entity relationship model is a collection of basic objects called entities and relationship among those objects. An entity is a thing or object in the real world that is distinguishable from other objects.

**15. What are attributes? Give examples.**

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

Example: possible attributes of customer entity are customer name, customer id, customer street, customer city.

**16. What is relationship? Give examples**

A relationship is an association among several entities. Example: A depositor relationship associates a customer with each account that he/she has.

**17. What are stored and derived attributes?**

Stored attributes: The attributes stored in a data base are called stored attributes.

Derived attributes: The attributes that are derived from the stored attributes are called derived attributes.

**18. What are composite attributes?**

Composite attributes can be divided in to sub parts. The degree of relationship type is the number of participating entity types.

**19. Define weak and strong entity sets?**

Weak entity set: entity set that do not have key attribute of their own are called weak entity sets.

Strong entity set: Entity set that has a primary key is termed a strong entity set.

**20. What does the cardinality ratio specify?**

Mapping cardinalities or cardinality ratios express the number of entities to which another entity can be associated. Mapping cardinalities must be one of the following:

- One to one
- One to many
- Many to one
- Many to many

**21. Define- relational algebra.**

The relational algebra is a procedural query language. It consists of a set of operations that take one or two relation as input and produce a new relation as output.

**22. What is a data dictionary?**

A data dictionary is a data structure which stores meta data about the structure of the database ie. the schema of the database.

**23. Explain the two types of participation constraint.**

- Total: The participation of an entity set E in a relationship set R is said to be total if every entity in E participates in at least one relationship in R.
- Partial: if only some entities in E participate in relationships in R, the participation of entity set E in relationship R is said to be partial.

**24. Define tuple variable?**

Tuple variables are used for comparing two tuples in the same relation. The tuple variables are defined in the from clause by way of the as clause.

**25. Define denormalization**

Demoralization is the process of attempting to optimize the performance of a database by adding redundant data or by grouping data. In some cases, demoralizations helps cover up the inefficiencies inherent in relational database software. A relational normalized database imposes a heavy access load over physical storage of data even if it is well tuned for high performance.

**26.Mention the codd's rule**

**Rule 0:** The Foundation rule
**Rule 1:** The information rule
**Rule 2:** The guaranteed access rule
**Rule 3:** Systematic treatment of null values
**Rule 4:** Active online catalog based on the relational model
**Rule 5:** The comprehensive data sublanguage rule

**Rule 6:** The view updating rule
**Rule 7:** High-level insert, update, and delete
**Rule 8:** Physical data independence
**Rule 9:** Logical data independence
**Rule 10:** Integrity independence
**Rule 11:** Distribution independence
**Rule 12:** The nonsubversion rule

**27.Define relational data model**

Relational model use a collection of tables to represent both data and the relationships among those data. Each table has a multiple columns and each columns has unique name.

**28. Define Single and Multi-valued Attributes**

- Single valued : can take on only a single value for each entity instance
  E.g. age of employee. There can be only one value for this
- Multi-valued:can take many values
  E.g. skill set of employee

**29. List out the operations of the relational algebra**

- It has Six basic operators
  1) select:
  2) project:
  3) union:
  *4)* set difference
  5) Cartesian product:
  6) Rename

**30. Define Object based data model**

Object based data model can be seen as extending the E-R model with notion of encapsulation, methods and object indentify

**31. Explain Semi structured data model**

- Specification of data where individual data item of same type may have different sets of attributes
- sometimes called schemaless or self-describing
- XML is widely used to represent this data model

**32. Explain Hierarchical data model**

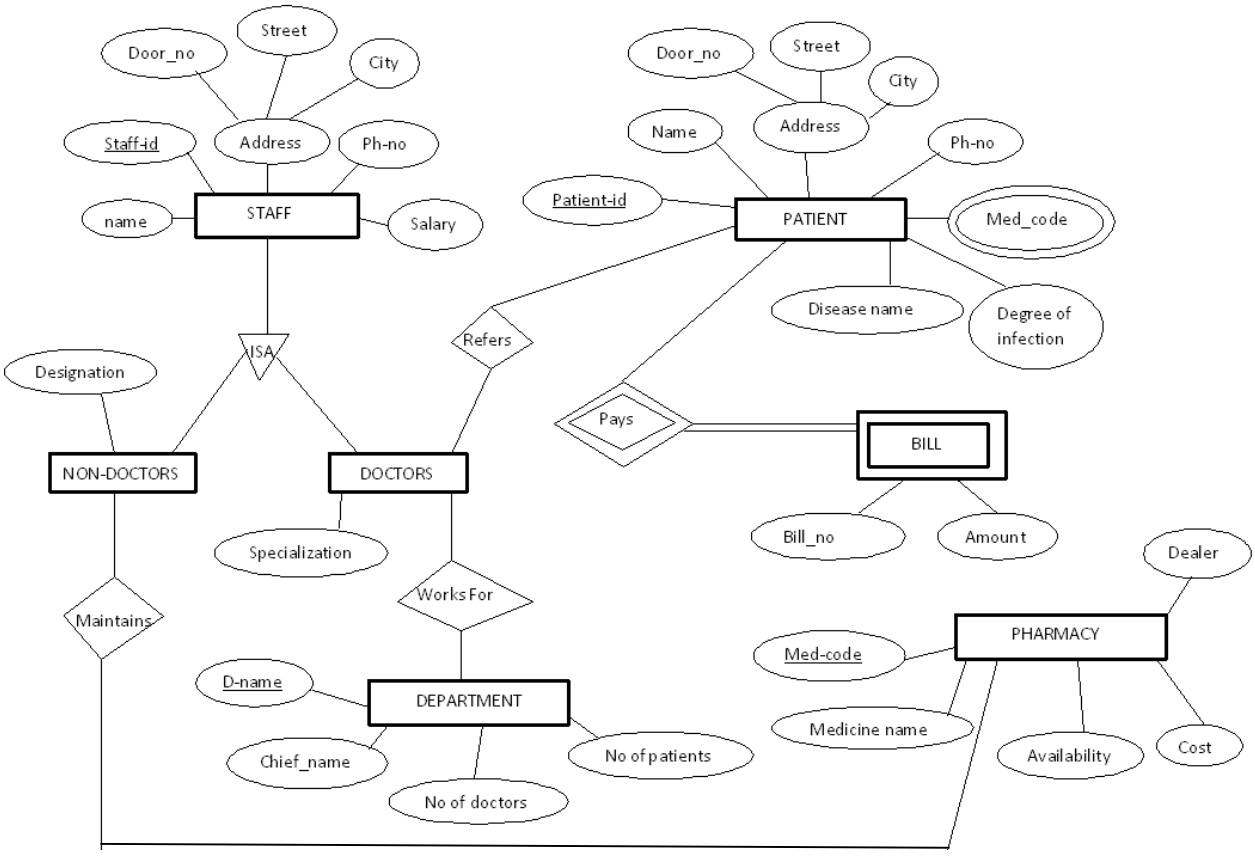
- The Hierarchical data model organizes data in a tree structure. There is hierarchy of parent and child data segments.
- This model uses parent child relationship.
- 1:M Mapping between record type

**33. Define Network Model**

- Some data were more naturally modeled with more than one parent per child.
- This model permitted the modeling of M:N relationship

## Part-B

1. **Explain ER model by taking Hospital management /University Database Management as case study**




**2. Explain the various components of ER diagram with examples.**

**ENTITY-RELATIONSHIP(ER) MODELING:**

- ER modeling: A graphical technique for understanding and organizing the data independent of the actual database implementation
- Entity: Any thing that may have an independent existence and about which we intend to collect data.Also known as Entity type.
- Entity instance: a particular member of the entity type e.g. a particular student
- Attributes: Properties/characteristics that describe entities
- Relationships: Associations between entities

**Attributes**

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

Example:

  - The domain of attribute marital status is just the four values: single, married, divorced, widowed
  - The domain of the attribute month is the twelve values ranging from January to December

- Key attribute: The attribute (or combination of attributes) that is unique for every entity instance
  - E.g the account number of an account, the employee id of an employee etc.
- If the key consists of two or more attributes in combination, it is called a composite key

**Simple Vs composite attribute**

• Simple attribute: cannot be divided into simpler components
E.g age of an employee
• Composite attribute: can be split into components
E.g Date of joining of the employee.
– Can be split into day, month and year

**Single Vs Multi-valued Attributes**

• Single valued : can take on only a single value for each entity instance
E.g. age of employee. There can be only one value for this
• Multi-valued:can take many values
E.g. skill set of employee

**Stored Vs Derived attribute**

• Stored Attribute:Attribute that need to be stored permanently.
• E.g. name of an employee
• Derived Attribute: Attribute that can be calculated based on other attributes
• E.g. : years of service of employee can be calculated from date of joining and current date

**Regular Vs. Weak entity type**

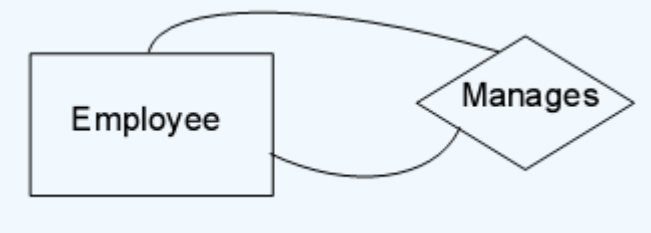
• Regular Entity: Entity that has its own key attribute.
E.g.: Employee, student ,customer, policy holder etc.
• Weak entity:Entity that depends on other entity for its existence and doesn't have key attribute of its own
E.g. : spouse of employee

**RELATIONSHIPS**

•A relationship type between two entity types defines the set of all associations between these entity types
• Each instance of the relationship between members of these entity types is called a Relationship instance

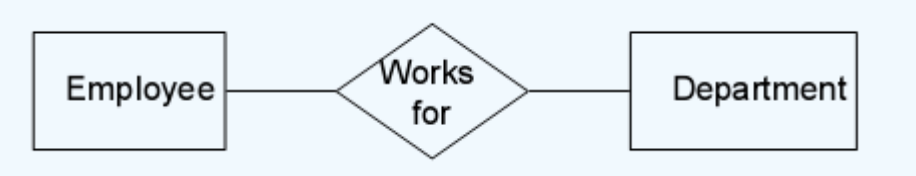
**Degree of a Relationship**

• Degree: the number of entity types involved
•One: Unary Relationship



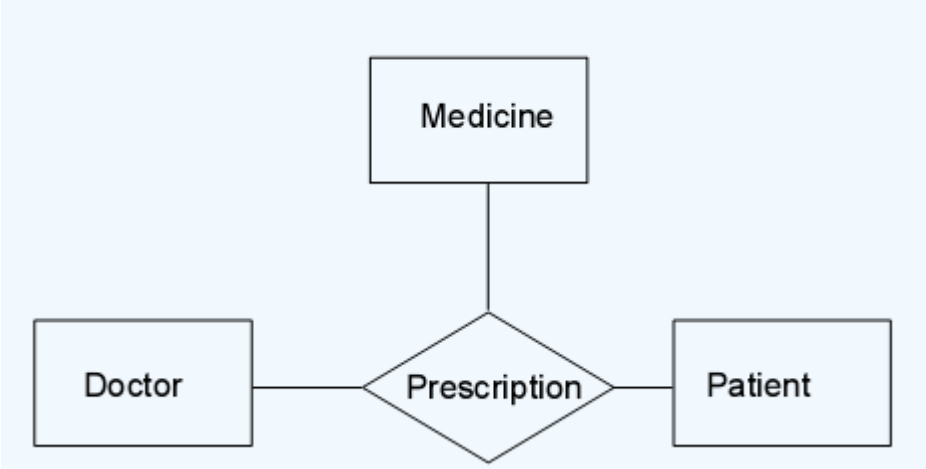

- ✓ A unary relationship is represented as a diamond which connects one entity to itself as a loop.
- ✓ The relationship above means, some instances of employee manage other instances of Employee.
- ✓ E.g.: employee manager-of employee is unary

•Two: Binary Relationship




- ✓ A relationship between two entity types
- ✓ E.g.: employee works-for department is binary

• Three: Ternary Relationship




✓ customer purchase item, shop keeper is a ternary relationship

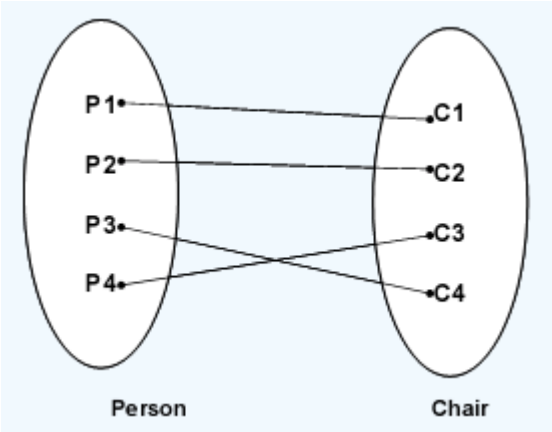
**Cardinality**

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

E.g.: Employee head-of department (1:1).
Lecturer offers course (1:n) assuming a course is taught by a single lecturer Student enrolls course (m:n).

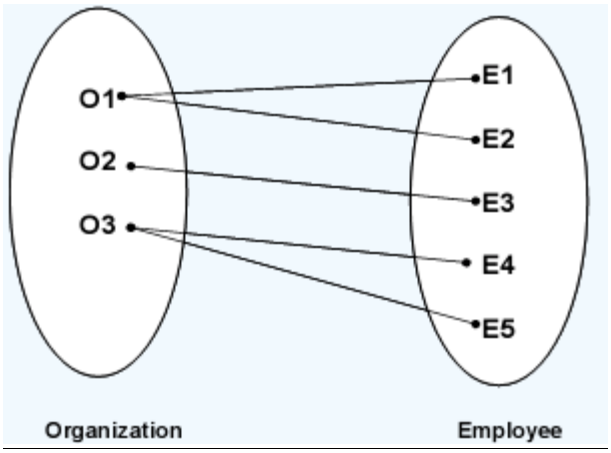
The minimum and maximum values of this connectivity is called the cardinality of the relationship.

**Cardinality – One – To – One**



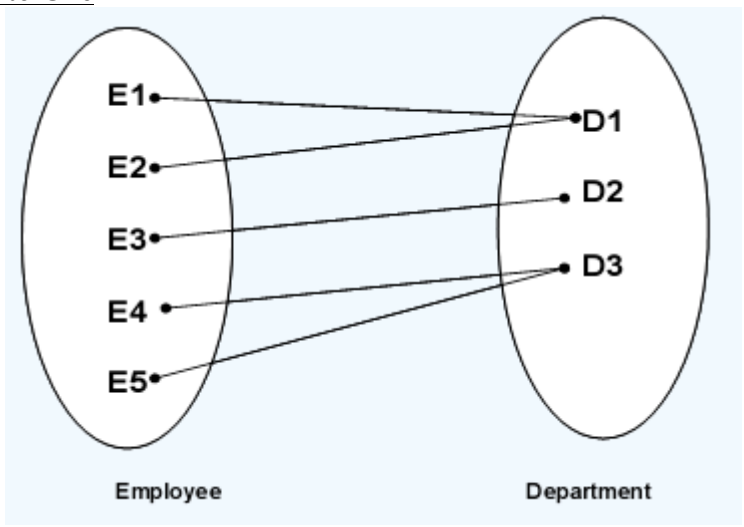

One instance of entity type Person is related to one instance of the entity type Chair.

**Cardinality – One –to- Many**



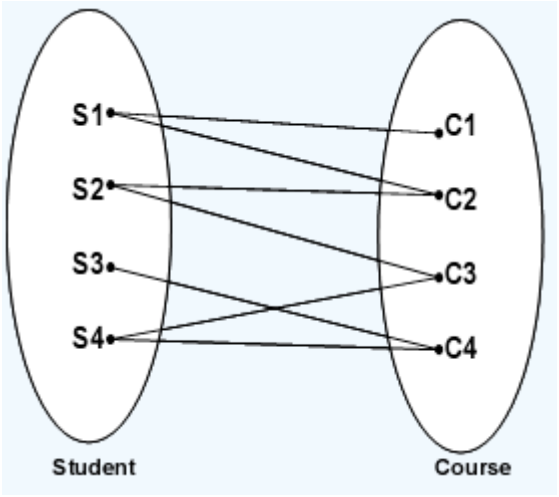

One instance of entity type Organization is related to multiple instances of entity type Employee.

**Cardinality – Many-to-One**




Reverse of the One to Many relationship

**Cardinality – Many-to-Many**




Multiple instances of one Entity are related to multiple instances of another Entity.

**Relationship Participation**

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

• Partial: All instances need not participate
E.g.: Employee Head-of Department
Employee: partial
Department: total
All employees will not be head-of some department. So only few instances of employee entity participate in the above relationship. But each department will be headed by some employee. So department entity's participation is total and employee entity's participation is partial in the above relationship.

**1.1**

**3. Explain Different types of file systems organization in detail.**

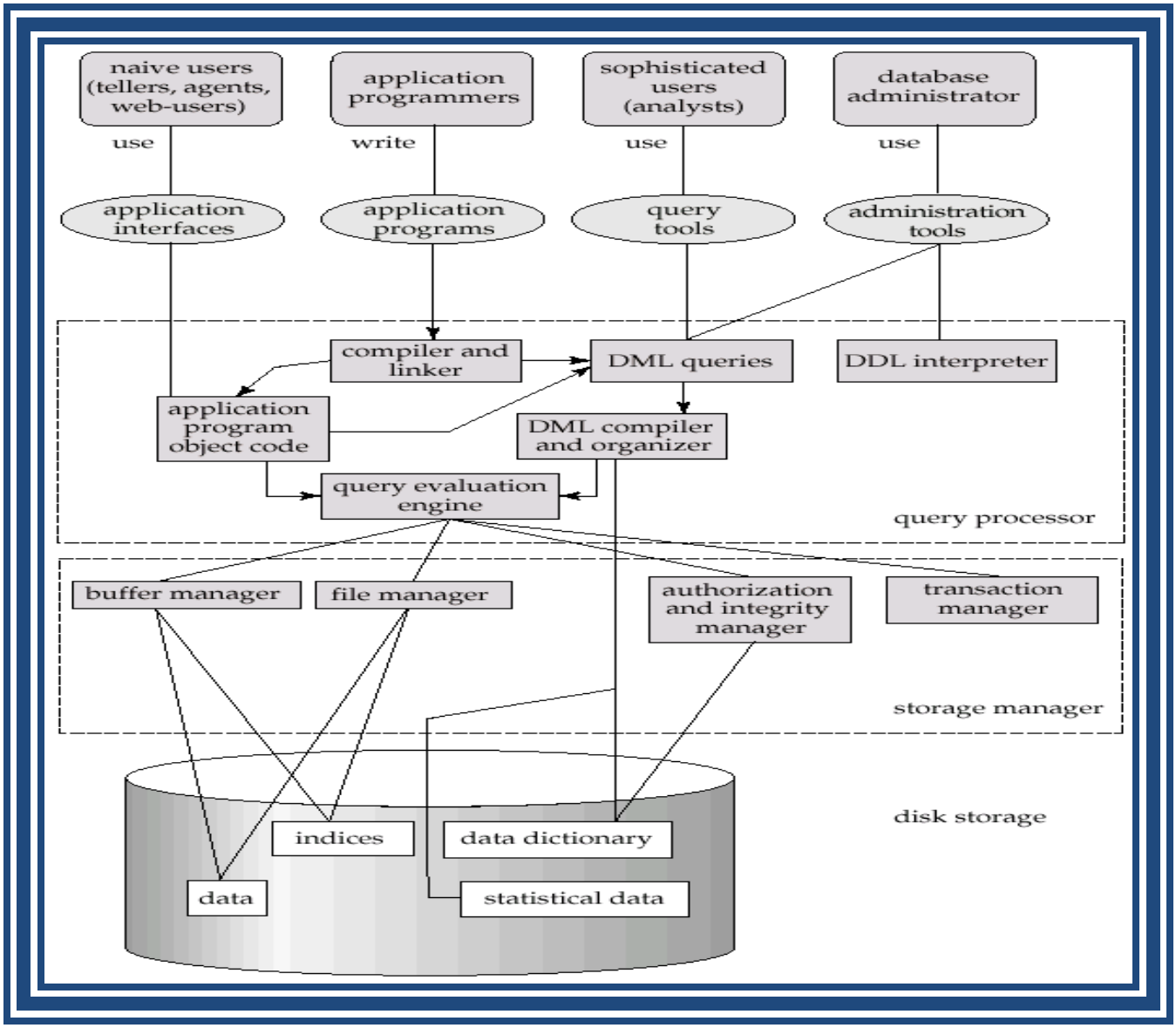
**Heap File Organization**: When a file is created using Heap File Organization mechanism, the Operating Systems allocates memory area to that file without any further accounting details. File records can be placed anywhere in that memory area. It is the responsibility of software to manage the records. Heap File does not support any ordering, sequencing or indexing on its own.
**Sequential File Organization**: Every file record contains a data field (attribute) to uniquely identify that record. In sequential file organization mechanism, records are placed in the file in the some sequential order based on the unique key field or search key. Practically, it is not possible to store all the records sequentially in physical form.
**Hash File Organization**: This mechanism uses a Hash function computation on some field of the records. As we know, that file is a collection of records, which has to be mapped on some block of the disk space allocated to it. This mapping is defined that the hash computation. The output of hash determines the location of disk block where the records may exist.
**Clustered File Organization**: Clustered file organization is not considered good for large databases. In this mechanism, related records from one or more relations are kept in a same disk block, that is, the ordering of records is not based on primary key or search key. This organization helps to retrieve data easily based on particular join condition. Other than particular join condition, on which data is stored, all queries become more expensive

**4. Explain the purpose and components of DBMS in detail.** **1.17-20**

**Database Users**
Users are differentiated by the way they expect to interact with the system

- Application programmers
- Sophisticated users
- Naïve users
- Database Administrator
- Specialized users etc,.

- **Application programmers:**
  – Professionals who write application programs and using these application programs they interact with the database system
- **Sophisticated users :**
  – These user interact with the database system without writing programs, But they submit queries to retrieve the information
- **Specialized users:**
  – Who write specialized database applications to interact with the database system.
- **Naïve users:**
  – Interacts with the database system by invoking some application programs that have been written previously by application programmers
  Eg : people accessing database over the web




**Database Administrator:**

- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise's information resources and needs.
    - Schema definition
    - Access method definition
    - Schema and physical organization modification
    - Granting user authority to access the database
    - Monitoring performance

**Storage Manager**

- The Storage Manager include these following components/modules
    - Authorization Manager
    - Transaction Manager
    - File Manager
    - Buffer Manager
- Storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
- The storage manager is responsible to the following tasks:
    - interaction with the file manager
    - efficient storing, retrieving and updating of data

- Authorization Manager
  - Checks whether the user is an authorized person or not
  - Test the satisfaction of integrity constraints
- Transaction Manager
  - Responsible for concurrent transaction execution

It ensures that the database remains in a consistent state despite of the system failure

**Query Processor**

- It is also a collection of components and used to interprets the queries which is submitted by the user.
- The query processor includes these following components
  - DDL interpreter
  - DML Compiler
  - Query evaluation engine
- DDL interpreter

  -Interprets DDL statements and records the definition in the data dictionary
- DML Compiler
  - Translate the DML Statements into an evaluation plan that contain low level instructions and the query evaluation engine can understand these instruction
  - Query can be translated into many alternative evaluation plans
  - Query optimization –picks up the lowest cost evaluation plan from the alternatives

5**. List out the disadvantages of File system over DB & explain it in detail.**

- In the early days, ***File-Processing system*** is used to store records. It uses various files for storing the records.
- Drawbacks of using file systems to store data:
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Data isolation — multiple files and formats
  - Integrity problems
    - Hard to add new constraints or change existing ones
  - Atomicity problem
    - Failures may leave database in an inconsistent state with partial updates carried out
    - E.g. transfer of funds from one account to another should either complete or not happen at all
  - Concurrent access anomalies
    - Concurrent accessed needed for performance
  - Security problems
- Database systems offer solutions to all the above problems

## 1.9

**6. Discuss about (i) Data Models (ii) Mapping cardinalities.**

**(i) Data Models**

- It is underlying structure of database
- It is collection of conceptual tolls for describing data, relationship and constraints.

Categories:

- Entity-Relationship model
- Relational model
- Object based data model

- Semi structured data model

1) Relational model

It uses a collection of tables to represent the data and the relationships among those data.
Each table contains many columns and each has a unique name.
It is the widely used data model and most of the database system is based on relational model.

2) Object based data model

Object based data model can be seen as extending the E-R model with notion of encapsulation, methods and object indentify.

3) Semi structured data model

- Specification of data where individual data item of same type may have different sets of attributes
- Sometimes called schema less or self-describing
- XML is widely used to represent this data model

4) Entity-Relationship model

The entity-relationship (ER) data model allows us to describe the data involved in a real-world enterprise in terms of objects and their relationships. It is widely used to develop an initial database design.

**(ii) Mapping cardinalities**

**Text**

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

An entity in A is associated with at most one entity in B and an entity in B is associated with at most one entity in A.

- **One to many(1:M)**

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

- **Many to one(M:1)**

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

- **Many to many(M:N)**

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.

**1.21**

**7.List out the operations of the relational algebra and explain with suitable examples.**

**Relational Algebra:**

- Procedural language
- The Operations can be classified as
  - Basic Operations
  - Additional Operations
  - Extended Operations

**Basic Operations**

  - select

- project
- union
- set difference
- Cartesian product
- rename

**Select Operation**

- It is used to select tuples from a relations
- Notation: $\sigma_p(r)$
  $p$ is called the selection predicate
- Defined as:
  $\sigma_p(r) = \{t \mid t \in r \text{ and } p(t)\}$
- Each term is one of:
  <attribute> *op* <attribute> or <constant>

where *op* is one of: =, ≠, >, ≥. <. ≤

- Example
  $\sigma_{branch\text{-}name=\text{“}Perryridge\text{”}}(account)$

**Project Operation**

- It is used to select certain columns from the relation.
- Notation:

  $\prod_{A1, A2, \ldots, Ak}(r)$

  Where $A_1$, $A_2$ are attributing names and $r$ is a relation name.
- The result is defined as the relation of $k$ columns obtained by erasing the columns that are not listed
- Duplicate rows removed from result, since relations are sets
- E.g. To eliminate the *branch-name* attribute of *account* relation
  $\prod_{account\text{-}number,\ balance}(account)$

**Union Operation**

- For $r \cup s$ to be valid.
  1. *r, s* must have the *same arity* (same number of attributes)
  2. The attribute domains must be *compatible* (e.g., 2nd column of *r* deals with the same type of values as does the 2nd column of *s*)
- E.g. to find all customers with either an account or a loan
- $\prod_{customer\text{-}name}(depositor) \cup \prod_{customer\text{-}name}(borrower)$

**Set Difference Operation**

- It is used to find tuples that are in one relation but are not in another relation.
- Notation $r - s$
- Defined as:
  $r - s = \{t \mid t \in r \text{ and } t \notin s\}$
- Set differences must be taken between *compatible* relations.
  - *r* and *s* must have the *same arity*

**Cartesian-Product Operation**

- It is used to combine information from two relations.
- Notation $r \times s$
- Defined as:
  $r \times s = \{t\, q \mid t \in r \text{ and } q \in s\}$

**Rename Operation**

- Allows us to name, and therefore to refer to, the results of relational-algebra expressions.
- Allows us to refer to a relation by more than one name.

Example:

$\rho_x(E)$

returns the expression $E$ under the name $X$

**2.21**

8**) Explain functional dependency in database design with its properties.**

**Functional dependency**

In a given relation R, X and Y are attributes. Attribute Y is **functionally dependent** on attribute X if each value of X determines **EXACTLY ONE** value of Y, which is represented as X -> Y (X can be composite in nature).

• We say here "x determines y" or "y is functionally dependent on x" X→Y does not imply Y→X

• If the value of an attribute "Marks" is known then the value of an attribute "Grade" is determined since Marks→ Grade

**Full functional dependency**

X and Y are attributes X Functionally determines Y Note: Subset of X should not functionally determine Y

Marks is fully functionally dependent on STUDENT# COURSE# and **not on sub set of** STUDENT# COURSE#. This means Marks can not be determined either by STUDENT# **OR** COURSE# alone. It can be determined only using STUDENT# **AND** COURSE# together. Hence Marks is fully functionally dependent on STUDENT# COURSE#.

**Partial functional dependency**

X and Y are attributes. Attribute Y is partially dependent on the attribute X only if it is dependent on a sub-set of attribute X.

Course Name, IName, Room# are partially dependent on composite attributes STUDENT# COURSE# because COURSE# alone defines the Course Name, IName, Room#.

**Transitive functional dependency**

**X Y and Z are three attributes. X -> Y, Y-> Z => X -> Z**

Room# depends on IName and in turn IName depends on COURSE#. Hence Room# transitively depends on COURSE#.

## 9.Define anomalies and explain 1st normal form & 2nd normal form with example.

A relation schema is in 1NF:

– If and only if all the attributes of the relation R are atomic in nature.

– Atomic: The smallest levels to which data may be broken down and remain meaningful

Second normal form: 2NF

A Relation is said to be in Second Normal Form if and only if:

– It is in the First normal form, and

– No partial dependency exists between non-key attributes and key attributes.

• An attribute of a relation R that belongs to any key of R is said to be a prime attribute and that which doesn't is a non-prime attribute. To make a table 2NF compliant, we have to remove all the partial dependencies

Note: - All partial dependencies are eliminated

**10. Discuss third normal form & BCNF with example.**

Third normal form: (3 NF) A relation R is said to be in the Third Normal Form (3NF) if and only if

− It is in 2NF and

− No transitive dependency exists between non-key attributes and key attributes.

• STUDENT# and COURSE# are the key attributes.

• All other attributes, except grade are non-partially, non-transitively dependent on key attributes.

• Student#, Course# - > Marks

• Marks -> Grade

Note : - All transitive dependencies are eliminated

Boyce-Codd Normal form – BCNF

A relation is said to be in Boyce Codd Normal Form (BCNF)

- if and only if all the determinants are candidate keys.BCNF relation is a strong 3NF, but not every 3NF relation is BCNF.

**2.35**

**11. Explain the 4NF and 5NF in detail with example.**

Fourth NF:

**Fourth normal form** (**4NF**) is a normal form used in database normalization. Introduced by Ronald Fagin in 1977, 4NF is the next level of normalization after Boyce–Codd normal form (BCNF). Whereas the second, third, and Boyce–Codd normal forms are concerned with functional dependencies, 4NF is concerned with a more general type of dependency known as a multivalued dependency. A table is in 4NF if and only if, for every one of its non-trivial multivalued dependencies $X \twoheadrightarrow Y$, $X$ is a super key—that is, $X$ is either a candidate key or a superset thereof.

**Multivalued dependencies**

If the column headings in a relational database table are divided into three disjoint groupings $X$, $Y$, and $Z$, then, in the context of a particular row, we can refer to the data beneath each group of headings as $x$, $y$, and $z$ respectively. A multivalued dependency $X \twoheadrightarrow Y$ signifies that if we choose any $x$ actually occurring in the table (call this choice $x_c$), and compile a list of all the $x_cyz$ combinations that occur in the table, we will find that $x_c$ is associated with the same $y$ entries regardless of z.A **trivial multivalued dependency** $X \twoheadrightarrow Y$ is one where either $Y$ is a subset of $X$, or $X$ and $Y$ together form the whole set of attributes of the relation.A functional dependency is a special case of multivalued dependency. In a functional dependency $X \rightarrow Y$, every $x$ determines *exactly one* $y$, never more than one.

**Pizza Delivery Permutations**

| Restaurant | Pizza Variety | Delivery Area |
|---|---|---|
| A1 Pizza | Thick Crust | Springfield |
| A1 Pizza | Thick Crust | Shelbyville |
| A1 Pizza | Thick Crust | Capital City |
| A1 Pizza | Stuffed Crust | Springfield |
| A1 Pizza | Stuffed Crust | Shelbyville |
| A1 Pizza | Stuffed Crust | Capital City |
| Elite Pizza | Thin Crust | Capital City |
| Elite Pizza | Stuffed Crust | Capital City |
| Vincenzo's Pizza | Thick Crust | Springfield |
| Vincenzo's Pizza | Thick Crust | Shelbyville |
| Vincenzo's Pizza | Thin Crust | Springfield |
| Vincenzo's Pizza | Thin Crust | Shelbyville |

**Varieties By Restaurant**

| Restaurant | Pizza Variety |
|---|---|
| A1 Pizza | Thick Crust |
| A1 Pizza | Stuffed Crust |
| Elite Pizza | Thin Crust |
| Elite Pizza | Stuffed Crust |
| Vincenzo's Pizza | Thick Crust |
| Vincenzo's Pizza | Thin Crust |

**Delivery Areas By Restaurant**

| Restaurant | Delivery Area |
|---|---|
| A1 Pizza | Springfield |
| A1 Pizza | Shelbyville |
| A1 Pizza | Capital City |
| Elite Pizza | Capital City |
| Vincenzo's Pizza | Springfield |
| Vincenzo's Pizza | Shelbyville |

**Fifth normal form**

**Fifth normal form** (**5NF**), also known as **Project-join normal form** (**PJ/NF**) is a level of database normalization designed to reduce redundancy in relational databases recording multi-valued facts by isolating semantically related multiple relationships. A table is said to be in the 5NF if and only if every join dependency in it is implied by the candidate keys.

A join dependency *{A, B, … Z} on R is implied by the candidate key(s) of R if and only if each of A, B, …, Z is a super key for R

**Travelling Salesman Product Availability By Brand**

| Travelling Salesman | Brand | Product Type |
|---|---|---|
| Jack Schneider | Acme | Vacuum Cleaner |
| Jack Schneider | Acme | Breadbox |
| Willy Loman | Robusto | Pruning Shears |
| Willy Loman | Robusto | Vacuum Cleaner |
| Willy Loman | Robusto | Breadbox |
| Willy Loman | Robusto | Umbrella Stand |
| Louis Ferguson | Robusto | Vacuum Cleaner |
| Louis Ferguson | Robusto | Telescope |
| Louis Ferguson | Acme | Vacuum Cleaner |
| Louis Ferguson | Acme | Lava Lamp |
| Louis Ferguson | Nimbus | Tie Rack |

**Product Types By Travelling Salesman**

| Travelling Salesman | Product Type |
|---|---|
| Jack Schneider | Vacuum Cleaner |
| Jack Schneider | Breadbox |
| Willy Loman | Pruning Shears |
| Willy Loman | Vacuum Cleaner |
| Willy Loman | Breadbox |
| Willy Loman | Umbrella Stand |
| Louis Ferguson | Telescope |
| Louis Ferguson | Vacuum Cleaner |
| Louis Ferguson | Lava Lamp |
| Louis Ferguson | Tie Rack |

**Brands By Travelling Salesman**

| Travelling Salesman | Brand |
|---|---|
| Jack Schneider | Acme |
| Willy Loman | Robusto |
| Louis Ferguson | Robusto |
| Louis Ferguson | Acme |
| Louis Ferguson | Nimbus |

**Product Types By Brand**

| Brand | Product Type |
|---|---|
| Acme | Vacuum Cleaner |
| Acme | Breadbox |
| Acme | Lava Lamp |
| Robusto | Pruning Shears |
| Robusto | Vacuum Cleaner |
| Robusto | Breadbox |
| Robusto | Umbrella Stand |
| Robusto | Telescope |
| Nimbus | Tie Rack |

# Unit-2
# SQL & QUERY OPTIMIZATION

**1. Define query language?**

A query is a statement requesting the retrieval of information. The portion of DML that involves information retrieval is called a query language.

**2. What are the categories of SQL command?**

SQL commands are divided in to the following categories:

1. Data - definitition language
2. Data manipulation language
3. Data Query language
4. Data control language
5. Data administration statements
6. Transaction control statements

**3. List the string operations supported by SQL?**

1) Pattern matching Operation
2) Concatenation
3) Extracting character strings
4) Converting between uppercase and lower case letters.

**4. What is the use of Union and intersection operation?**

Union: The result of this operation includes all tuples that are either in r1 or in r2 or in both r1 and r2.Duplicate tuples are automatically eliminated.

Intersection: The result of this relation includes all tuples that are in both r1 and r2.

**5. What are aggregate functions? And list the aggregate functions supported by SQL?**

Aggregate functions are functions that take a collection of values as input and return a single value.

Aggregate functions supported by SQL are

- Average: avg
- Minimum: min
- Maximum: max
- Total: sum
- Count: count

**6. List out some date functions.**

To_date
To_char(sysdate,'fmt')
– d,dd,ddd,mon,dy,day,y,yy,yyy,yyyy,year,month,mm

**7. What is the use of sub queries?**

A sub query is a select-from-where expression that is nested with in another query. A common use of sub queries is to perform tests for set membership, make set comparisions, and determine set cardinality.

**8. List the SQL domain Types?**

SQL supports the following domain types.

1) Char(n) 2) varchar(n) 3) int 4) numeric(p,d) 5) float(n) 6) date.

**9. What is the use of integrity constraints?**

Integrity constraints ensure that changes made to the database by authorized users do not result in a loss of data consistency. Thus integrity constraints guard against accidental damage to the database.

**10. Mention the 2 forms of integrity constraints in ER model?**

Key declarations, Form of a relationship

**11. List some security violations (or) name any forms of malicious access.**

1) Unauthorized reading of data 2) Unauthorized modification of data 3) Unauthorized destruction of data.

**12. What is called query processing?**

Query processing refers to the range of activities involved in extracting data from a database.

**13. What is called a query evaluation plan?**

A sequence of primitive operations that can be used to evaluate ba query is a query evaluation plan or a query execution plan.

**14. What is called as an N-way merge?**

The merge operation is a generalization of the two-way merge used by the standard in-memory sort-merge algorithm. It merges N runs, so it is called an N-way merge.

**15. What is a primary key?**

Primary key is a set of one or more attributes that can uniquely identify record from the relation; it will not accept null values and redundant values. A relation can have only one primary key.

**16. What is a super key?**

A super key is a set of one or more attributes that collectively allows us to identify uniquely an entity in the entity set.

**17. What is foreign key?**

A relation schema r1 derived from an ER schema may include among its attributes the primary key of another relation schema r2.this attribute is called a foreign key from r1 referencing r2.

**18. What is the difference between unique and primary key?**

Unique and primary key are keys which are used to uniquely identify record from the relation. But unique key accepts null values; primary key does not accept null values.

**19. What is the difference between char and varchar2 data type?**

Char and varchar2 are data types which are used to store character values. But Char is static memory allocation; varchar2 is dynamic memory allocation.

**20. How to add primary key to a table with suitable query?**

Alter table <table name> add primary key(column);

**21. Define query optimization.**

Query optimization refers to the process of finding the lowest –cost method of evaluating a given query.

**22. Define Candidate key**

A Candidate key is a set of one or more attributes that can uniquely identify a row in a given table. A relation can have more than one candidate keys.

**23. How to create composite primary key with suitable query?**

Create table <table name>(a number, b number, primary key(a,b))

**24. List out DDL and DML Commands**

Data Definition Language:

1) Create
2) Alter
3) Rename
4) Drop

Data Manipulation Language:

1) Insert
2) Selec
3) Update
4) Delete

**25.What is embedded SQL? What are its advantages? (April/May 2011)**

The SQL standard defines embeddings of SQL in a variety of programming languages such as C, Java, and Cobol. A language to which SQL queries are embedded is referred to as a host language, and the SQL structures permitted in the host language comprise embedded SQL.

The basic form of these languages follows that of the System R embedding of SQL into PL/I.

EXEC SQL statement is used to identify embedded SQL request to the preprocessor

EXEC SQL <embedded SQL statement > END_EXEC

**26.What is dynamic sql?**

-The dynamic SQL components of SQL allows programs to construct and submit SQL Queries at runtime

-Using dynamic SQL the programs can create SQL queries at runtime and can either have them executed immediately or have them prepared for subsequent use.

**27. List out DCL and TCL Commands**

**TCL:**

**1.**Commit

2.Rollback
3.Savepoint

**DCL**:

1.Grant
2.Revoke

## Part-B

**1. Explain about data definition language in SQL with examples.**

**DDL COMMANDS (DATA DEFINITION LANGUAGE)**

- Create
- Alter
  - ✓ Add
  - ✓ Modify
  - ✓ Drop
- Rename
- Drop

**SYNTAX:**

**CREATE:**

Create table <table name> (column name1 datatype1 constraints, column2 datatype2 . . .);

**ALTER:**

**ADD:**

Alter table <table name> add(column name1 datatype1);

**MODIFY:**

Alter table <table name> modify(column name1 datatype1);

**DROP:**

Alter table <table name> drop (column name);

**RENAME:**

Rename <old table name> to <new table name>;

**DROP:**

Drop table <table name>;

**2. Explain about data manipulation language in SQL with examples.**

**DATA MANIPULATION LANGUAGE**
**DML Commands:**

- Insert
- Select
- Update
- Delete

**SYNTAX:**

**INSERT:**

**Single level:**

Insert into <table name> values (‘attributes1’, ’attributes2’……);

**Multilevel:**

Insert into <table name> values (‘&attributes1’,’&attributes2’….);

**SELECT:**

**Single level:**

Select <column name> from <table name>;

**Multilevel:**

Select * from <table name> where <condition>;

**UPDATE:**

**Single level:**

Update <table name> set <column name>=’values’ where <condition>;

**Multilevel:**

Update <table name> set <column name>=’values’;

**DELETE:**

**Single level:**

Delete from <table name> where <column name>=’values’;

**Multilevel:**

Delete from <table name>;

**3. Explain about data control language in SQL with examples.**

**DCL (Data Control Language)**

**GRANT**

Used to grant privileges to the user

```
GRANT privilege_name
ON object_name
TO {user_name |PUBLIC |role_name}
[WITH GRANT OPTION];
```

- **privilege_name** is the access right or privilege granted to the user. Some of the access rights are ALL, EXECUTE, and SELECT.
- **object_name** is the name of an database object like TABLE, VIEW, STORED PROC and SEQUENCE.
- **user_name** is the name of the user to whom an access right is being granted.
- **user_name** is the name of the user to whom an access right is being granted.
- **PUBLIC** is used to grant access rights to all users.
- **ROLES** are a set of privileges grouped together.
- **WITH GRANT OPTION** - allows a user to grant access rights to other users.

**REVOKE**

Used to revoke privileges from the user

```
REVOKE privilege_name
ON object  name
FROM {user_name |PUBLIC |role_name}
```

**1) System privileges** - This allows the user to CREATE, ALTER, or DROP database objects.
**2) Object privileges** - This allows the user to EXECUTE, SELECT, INSERT, UPDATE, or DELETE data from database objects to which the privileges apply.

**4. Explain about TCL in SQL with examples.**

**TCL (Transaction Control Language)**

**COMMIT**

Used to made the changes permanently in the Database.

**ROLLBACK**

Similar to the undo operation.

```
SQL> delete from branch;
6 rows deleted.
SQL> select * from branch;
no rows selected
SQL> rollback;
Rollback complete.
SQL>  select * from branch;
BRANCH_NAME     BRANCH_CITY            ASSETS
--------------- -------------------- ----------------------------------------
tambaram                chennai-20              50000
adayar                  chennai-20              100000
tnagar                  chennai-17               250000
saidapet                chennai-15              150000
chrompet                chennai-43              450000
guindy                  chennai-32              150000

6 rows selected.
```

---

**SAVE POINT**

```
SQL> select * from customer;
   CUSTID    PID       QUANTITY
    ----------  ----------  ----------
      100       1234        10
      101       1235        15
      102       1236        15
      103       1237        10
SQL> savepoint s1;
Savepoint created.
SQL> Delete from customer where custid=103;
   CUSTID   PID       QUANTITY
    ----------  ----------  ----------
      100       1234        10
      101       1235        15
      102       1236        15
SQL> rollback to s1;
Rollback complete.
SQL> select * from customer;
   CUSTID   PID      QUANTITY
   ----------  ---------- ----------
      100       1234        10
      101       1235        15
      102       1236        15
      103       1237        10
SQL> commit;
```

**5. Explain Integrity constraints with example.**

- Constraints

  Constraints within a database are rules which control values allowed in columns.
- Integrity

  Integrity refers to requirement that information be protected from improper modification.
- Integrity constraints

  Integrity constraints provide a way of ensuring that changes made to the database by authorized users do not result in a loss of data consistency
- Constraints
  - Primary key
  - Referential integerity
  - Check constraint
  - Unique Constraint
  - Not Null/Null
- Primary key

  The primary key of a relational table uniquely identifies each record in the table
  -Unique
  -Not null
- Referential integrity

  We ensure that a value appears in one relation for a given set of attribute also appears for a certain set of attribute in another relation.

eg : branch_name varchar2(10) references branch(branch_name)

- The foreign key identifies a column or set of columns in one (referencing) table that refers to a column or set of columns in another (referenced) table

- Check contraint

Eg :
Create table branch(
Balance number check (balance >500));

**6. With suitable example discuss the following conceptual queries in SQL.**

**a. Mapping of select clause**

- It is used to select tuples from a relations
- **Notation:** $\sigma_p(r)$

  $p$ is called the selection predicate
- Defined as:

  $$\sigma_p(r) = \{t \mid t \in r \text{ and } p(t)\}$$
- Each term is one of:

  <attribute> *op* <attribute> or <constant>

  where *op* is one of: =, ≠, >, ≥. <. ≤
- Example of selection:

  $\sigma_{branch\text{-}name=\text{“}Perryridge\text{”}}(account)$

**b. Tuple variables**

A tuple variable is variable that takes on tuples of a particular relation schema as values. That is, every value assigned to a given tuple variable has the same number and type of fields.

**c. Comparison string operation**

Strcmp()-used to compare two string

d. **Set operation**

**Set-Intersection Operation**

Notation: $r \cap s$
Defined as:
$r$ **and** $t \in s$ }
Note: $r \cap s = r - (r - s)$

**Set Difference Operation**

- It is used to find tuples that are in one relation but are not in another relation.
- Notation $r - s$
- Defined as:

$$r - s = \{t \mid t \in r \textbf{ and } t \notin s\}$$

- Set differences must be taken between *compatible* relations.
  - $r$ and $s$ must have the *same arity*

**e. Aggregate function**

**Aggregation function** takes a collection of values and returns a single value as a result.

**avg**: average value
**min**: minimum value
**max**: maximum value
**sum**: sum of values
**count**: number of values

**7. Design an employee detail relation and explain referential integrity using SQL queries.**

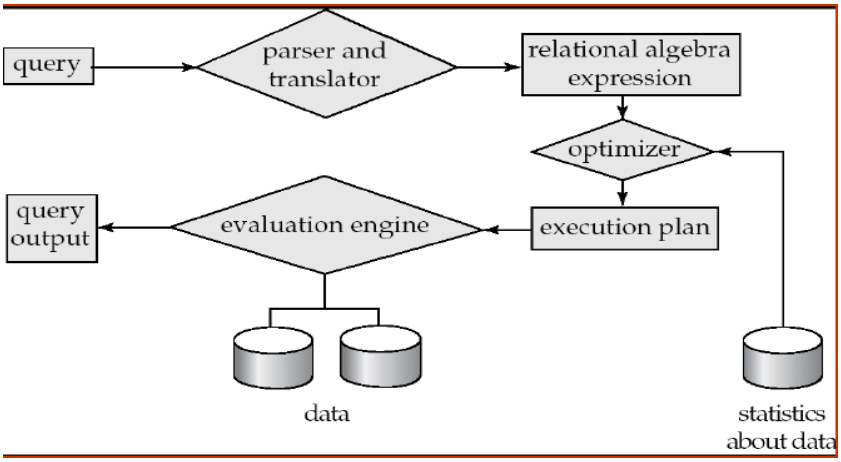
Employee(emp_id pk,emp_Name,Addr,phone_no,dept_id FK)
Dept(dept_id PK,dept_name)

**Dept table**
Create table dept(dept_id varchar2(20) primary key, dept_name varchar2(20))

**Employee table**
Create table employee(emp_id varchar2(20) primary key,emp_name varchar2(20),phone_no number,dept_id varchar2(20) **references** emp(dept_id));

**8. Explain about Query optimization with neat Diagram.**

- **Query Optimization:**
  Among all equivalent evaluation plans choose the one with lowest cost.
  - Cost is estimated using statistical information from the database catalog
    - e.g. number of tuples in each relation, size of tuples, etc.



**9. Explain about Embedded SQL,Static and Dynamic SQL**

**Embedded SQL**

-The SQL standards defines embedding of SQL in variety of programming language such as VB,C,C++,Java.

-A language to which SQL queries are embedded is referred to as a host language and the SQL structures permitted in the language comprise embedded SQL.

**Two reasons**

- Not all queries expressed in SQL,since SQL does not provide the full expressive power of a general purpose language.
- Nondeclarative actions- printing a report,intracting with user

**Dynamic SQL**

- The dynamic SQL components of SQL allows programs to construct and submit SQL Queries at runtime.
- In contrast the embedded SQL statement must be completely present at compile time they are compiled by the embedded SQL pre processor
- Using dynamic SQL the programs can create SQL queries at runtime and can either have them executed immediately or have them prepared for subsequent use.

**10. Explain about different data types in data base language with an example.**

User defined types useful in situations where the same data type is used in several columns from different tables.

The names of the user-defined types provides the extra information.

**Syntax**

- Creation

  Create or replace type <type_name> as object <representation >
- Description:

  Desc <type_name>;
- Delete type:

  Drop type <typename>;

**11. Explain in detail about Heuristics and Cost estimates in Query Optimization**

Process for heuristics optimization

- The parser of a high-level query generates an initial internal representation
- Apply heuristics rules to optimize the internal representation
- A query execution plan is generated to execute groups of operations based on the access paths available on the files involved in the query
- The main heuristic is to apply first the operations that reduce the size of intermediate results.

E.g., Apply SELECT and PROJECT operations before applying the JOIN or other binary operations.

Cost Based Query Optimiztion:

- ✓ Cost-based optimization is expensive, even with dynamic programming.
- ✓ Systems may use *heuristics* to reduce the number of choices that must be made in a cost-based fashion.
- ✓ Heuristic optimization transforms the query-tree by using a set of rules that typically (but not in all cases) improve execution performance:
  - ★ Perform selection early (reduces the number of tuples)
  - ★ Perform projection early (reduces the number of attributes)
  - ★ Perform most restrictive selection and join operations before other similar operations.
  - ★ Some systems use only heuristics, others combine heuristics with partial cost-based optimization.

Steps in Typical Heuristic Optimization

1. Deconstruct conjunctive selections into a sequence of single selection operations
2. Move selection operations down the query tree for the earliest possible execution .
3. Execute first those selection and join operations that will produce the smallest relations.
4. Replace Cartesian product operations that are followed by a selection condition by join operations
5. Deconstruct and move as far down the tree as possible lists of projection attributes, creating new projections where needed
6. Identify those subtrees whose operations can be pipelined, and execute them using pipelining.

## Unit 3
## TRANSACTION PROCESSING AND CONCURRENCY CONTROL

**1. Give the reasons for allowing concurrency?**

The reasons for allowing concurrency is if the transactions run serially, a short transaction may have to wait for a preceding long transaction to complete, which can lead
to unpredictable delays in running a transaction. So concurrent execution reduces the unpredictable delays in running transactions.

**2. What is average response time?**

The average response time is that the average time for a transaction to be completed after it has been submitted.

**3. What are the two types of serializability?**

The two types of serializability is Conflict serializability, View serializability

**4. Differentiate strict two phase locking protocol and rigorous two phase locking protocol.**

In strict two phases locking protocol all exclusive mode locks taken by a transaction is held until that transaction commits.

Rigorous two phase locking protocol requires that all locks be held until the Transaction commits.

**5. How the time stamps are implemented**

• Use the value of the system clock as the time stamp. That is a transaction's time stamp is equal to the value of the clock when the transaction enters the system.

• Use a logical counter that is incremented after a new timestamp has been assigned; that is the time stamp is equal to the value of the counter.

**6. What are the time stamps associated with each data item?**

• W-timestamp (Q) denotes the largest time stamp if any transaction that executed WRITE (Q) successfully.

• R-timestamp (Q) denotes the largest time stamp if any transaction that executed READ (Q) successfully.

**7. Define blocks?**

The database system resides permanently on nonvolatile storage, and is partitioned into fixed-length storage units called blocks.

**8. What are the different modes of lock?**

The modes of lock are: Shared, Exclusive

**9. Define deadlock?**

Neither of the transaction can ever proceed with its normal execution. This situation is called deadlock.

**10. Define the phases of two phase locking protocol**

Growing phase: a transaction may obtain locks but not release any lock.

Shrinking phase: a transaction may release locks but may not obtain any new locks.

**11. Define upgrade and downgrade?**

It provides a mechanism for conversion from shared lock to exclusive lock is known as upgrade.

It provides a mechanism for conversion from exclusive lock to shared lock is known as downgrade.

**12. What is a database graph?**

The partial ordering implies that the set D may now be viewed as a directed acyclic graph, called a database graph.

**13. What is meant by buffer blocks?**

The blocks residing temporarily in main memory are referred to as buffer blocks.

**14. What are uncommitted modifications?**

The immediate-modification technique allows database modifications to be output to the database while the transaction is still in the active state. Data modifications written by active transactions are called uncommitted modifications.

**15. Define shadow paging.**

An alternative to log-based crash recovery technique is shadow paging. This technique needs fewer disk accesses than do the log-based methods.

**16. Define page.**

The database is partitioned into some number of fixed-length blocks, which are referred to as pages.

**17. Explain current page table and shadow page table.**

The key idea behind the shadow paging technique is to maintain two page tables during the life of the transaction: the current page table and the shadow page table. Both the page tables are identical when the transaction starts. The current page table may be changed when a transaction performs a write operation.

**18. What are the drawbacks of shadow-paging technique?**

- Commit Overhead
- Data fragmentation
- Garbage collection

**19. Define garbage collection.**

Garbage may be created also as a side effect of crashes. Periodically, it is necessary to find all the garbage pages and to add them to the list of free pages. This process is called garbage collection.

**20. What is transaction?**

Collections of operations that form a single logical unit of work are called transactions.

**21. What are the properties of transaction?**

The properties of transactions are:

Atomicity,
Consistency,
Isolation,
Durability

**22. What is recovery management component?**

Ensuring durability is the responsibility of a software component of the base system called the recovery management component.

**23. When is a transaction rolled back?**

Any changes that the aborted transaction made to the database must be undone. Once the changes caused by an aborted transaction have been undone, then the transaction has been rolled back.

**24. What are the states of transaction?**

The states of transaction are Active, Partially committed, Failed, Aborted, Committed, Terminated

**25. What is a shadow copy scheme?**

It is simple, but efficient, scheme called the shadow copy schemes. It is based on making copies of the database called shadow copies that one transaction is active at a time. The scheme also assumes that the database is simply a file on disk.

**26.Define Serializability and mention its types**

A (possibly concurrent) schedule is serializable if it is equivalent to a serial schedule.

Different forms of schedule equivalence give rise to the notions of:

1.conflict serializability

2.view serializability

**27. Metion the approaches of deadlock recovery**

The common solution is to roll back one or more transactions to break the deadlock

- Selection of victim
- Rollback
  - Partial
  - Total
- Starvation

## Part-B

**1. Explain about two phase commit protocol with an example.**

-Two phase commit is important when the transaction is related with many resource managers.
-The system component called coordinator handles the COMMIT or ROLLBACK operation.
It has two phases
-Prepare
-Commit

- **Prepare**

It instruct all the resource manager to get ready to **go either way.** Then the resource manager now replies to the coordinator.

- **Commit**
  - The Coordinator receive reply from all the resource manager, if all reply were OK then it takes the decision **COMMIT.**
  - If it receives any reply as NOT OK then it takes the decision **ROLLBACK.**

**2. What is serializability? Explain its types?**

A (possibly concurrent) schedule is serializable if it is equivalent to a serial schedule. Different forms of schedule equivalence give rise to the notions of:

1. **conflict serializability**
2. **view serializability**

Instructions $l_i$ and $l_j$ of transactions $T_i$ and $T_j$ respectively, **conflict** if and only if there exists some item $Q$ accessed by both $l_i$ and $l_j$, and at least one of these instructions wrote $Q$.

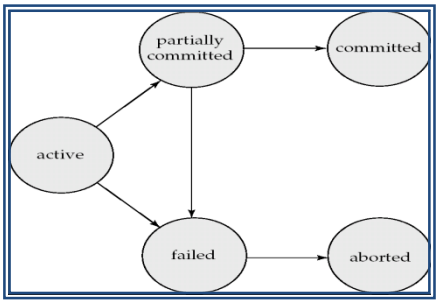
1. $l_i$ = **read**($Q$), $l_j$ = **read**($Q$).$l_i$ and $l_j$ don't conflict.
2. $l_i$ = **read**($Q$), $l_j$ = **write**($Q$). They conflict.
3. $l_i$ = **write**($Q$), $l_j$ = **read**($Q$). They conflict
4. $l_i$ = **write**($Q$), $l_j$ = **write**($Q$). They conflict

- If a schedule $S$ can be transformed into a schedule $S'$ by a series of swaps of non-conflicting instructions, we say that $S$ and $S'$ are **conflict equivalent**.
- We say that a schedule $S$ is **conflict serializable** if it is conflict equivalent to a serial schedule

**view serializability**

- Let $S$ and $S'$ be two schedules with the same set of transactions. $S$ and $S'$ are **view equivalent** if the following three conditions are met, for each data item $Q$,
  - If in schedule S, transaction $T_i$ reads the initial value of $Q$, then in schedule $S'$ also transaction $T_i$ must read the initial value of $Q$.
  - If in schedule S transaction $T_i$ executes **read**($Q$), and that value was produced by transaction $T_j$ (if any), then in schedule $S'$ also transaction $T_i$ must read the value of $Q$ that was produced by the same **write**(Q) operation of transaction $T_j$ .
  - The transaction (if any) that performs the final **write**($Q$) operation in schedule $S$ must also perform the final **write**($Q$) operation in schedule $S'$.

**3. Write short notes on Transaction State and discuss the properties of transaction.**




- **Active** – the initial state; the transaction stays in this state while it is executing
- **Partially committed** – after the final statement has been executed.
- **Failed --** after the discovery that normal execution can no longer proceed.
- **Aborted** – after the transaction has been rolled back and the database restored to its state prior to the start of the transaction. Two options after it has been aborted:
  - restart the transaction
  - kill the transaction
- **Committed** – after successful completion.

**ACID Properties**

- Atomicity
-Consistency
-Isolation
-Durability

4. **Briefly describe two phase locking in concurrency control techniques.**

This protocol requires that each transaction issue lock and unlock request in two phases

- Growing phase
- Shrinking phase

**Growing phase**

-During this phase new locks can be occurred but none can be released

**Shrinking phase**

-During which existing locks can be released and no new locks can be occurred

**Types**

- Strict two phase locking protocol
- Rigorous two phase locking protocol

**Strict two phase locking protocol**

- This protocol requires not only that locking be two phase, but also all exclusive locks taken by a transaction be held until that transaction commits.

**Rigorous two phase locking protocol**

This protocol requires that all locks be held until all transaction commits

**5. Explain the concepts of concurrent execution in Transaction processing system.**

The transaction-processing system allows **concurrent execution** of multiple transactions to improve the system performance. In concurrent execution, the database management system controls the execution of two or more transactions in parallel; however, allows only one operation of any transaction to occur at any given time within the system. This is also known as **interleaved execution** of multiple transactions. The database system allows concurrent execution of transactions due to two reasons.

First, a transaction performing read or write operation using I/O devices may not be using the CPU at a particular point of time. Thus, while one transaction is performing I/O operations, the CPU can process another transaction. This is possible because CPU and I/O system in the computer system are capable of operating in parallel. This overlapping of I/O and CPU activities reduces the amount of time for which the disks and processors are idle and, thus, increases the **throughput** of the system (the number of transactions executed in a given amount of time).

**6. Why concurrency control is needed? Explain with an example.**

Simultaneous execution of transactions over a shared database can create several data integrity and consistency problems.

- lost updated problem
-Temporary updated problem
-Incorrect summery problem

**Lost updated problem**

| T1 | T2 |
|---|---|
| Read(X) | Read(X) |
| X:=X-N | X:=X+M |
| Write(X) | Write(X) |
| Read(Y) | |
| Y:=Y+N | |
| Write(Y) | |

Temporary updated problem

| T1 | T2 |
|---|---|
| **Read(X)**<br>**X:=X-N**<br>**Write(X)**<br>**Rollback**<br>**Read(Y)** | **Read(X)**<br>**X:=X-M**<br>**Write(X)** |

**Incorrect summery problem**

| T1 | T2 |
|---|---|
| Read(B)<br>B:=B-N<br>Write(B)<br>Read(X)<br>X:=X-N<br>Write(X) | Sum:=0<br>Read(A)<br>Sum:=sum+A<br>Read(B)<br>Sum:=sum+B<br>.<br>.<br>. |

| | |
|---|---|
| | Read(X)<br>Sum:=sum+X |
| | |

**7. Define Dead Lock. Explain about dead lock prevention protocol.**

System is deadlocked if there is a set of transactions such that every transaction in the set is waiting for another transaction in the set.

***Deadlock prevention*** protocols ensure that the system will *never* enter into a deadlock state. Some prevention strategies :

- Require that each transaction locks all its data items before it begins execution (predeclaration).
- Impose partial ordering of all data items and require that a transaction can lock data items only in the order specified by the partial order (graph-based protocol).

**WAIT-DIE SCHEME:**

In this scheme, if a transaction request to lock a resource (data item), which is already held with conflicting lock by some other transaction, one of the two possibilities may occur:

- If $TS(T_i) < TS(T_j)$, that is $T_i$, which is requesting a conflicting lock, is older than $T_j$, $T_i$ is allowed to wait until the data-item is available.
- If $TS(T_i) > TS(t_j)$, that is $T_i$ is younger than $T_j$, $T_i$ dies. $T_i$ is restarted later with random delay but with same timestamp.

This scheme allows the older transaction to wait but kills the younger one.

**WOUND-WAIT SCHEME:**

In this scheme, if a transaction request to lock a resource (data item), which is already held with conflicting lock by some other transaction, one of the two possibilities may occur:

- If $TS(T_i) < TS(T_j)$, that is $T_i$, which is requesting a conflicting lock, is older than $T_j$, $T_i$ forces $T_j$ to be rolled back, that is $T_i$ wounds $T_j$. $T_j$ is restarted later with random delay but with same timestamp.
- If $TS(T_i) > TS(T_j)$, that is $T_i$ is younger than $T_j$, $T_i$ is forced to wait until the resource is available.

This scheme, allows the younger transaction to wait but when an older transaction request an item held by younger one, the older transaction forces the younger one to abort and release the item.

In both cases, transaction, which enters late in the system, is aborted.

**8. Explain about dead lock recovery algorithm with an example.**

The common solution is to roll back one or more transactions to break the deadlock.

Three action need to be taken

a. Selection of victim
b. Rollback
c. Starvation

Selection of victim

i. Set of deadlocked transations,must determine which transaction to roll back to break the deadlock.
ii. Consider the factor minimum cost

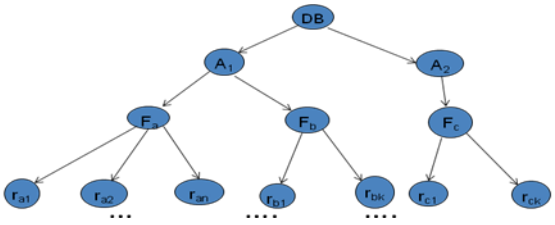
Rollback

- once we decided that a particular transaction must be rolled back, must determine how far this transaction should be rolled back
  - Total rollback
  - Partial rollback

Starvation

Ensure that a transaction can be picked as victim only a finite number of times.

**9. Illustrate Granularity locking method in concurrency control.**




Intent locking

- Intent locks are put on all the ancestors of a node before that node is locked explicitly.
- If a node is locked in an intention mode, explicit locking is being done at a lower level of the tree.

Intent shared lock(IS)

- If a node is locked in indent shared mode, explicit locking is being done at a lower level of the tree, but with only shared-mode lock
- Suppose the transaction $T_1$ reads record $r_{a2}$ in file $F_a$. Then,$T_1$ needs to lock the database, area $A_{1,and}$ $F_a$ in IS mode, and finally lock $r_{a2}$ in S mode.

Intent exclusive lock(IX)

if a node is locked in intent locking is being done at a lower level of the tree, but with exclusive mode or shared-mode locks.

Suppose the transaction $T_2$ modifies record $r_{a9}$ in file $F_a$. Then,$T_2$ needs to lock the database, area $A_{1,and}$ $F_a$ in IX mode, and finally to lock $r_{a9}$ in X mode

Shared Intent exclusive lock (SIX)

- If the node is locked in Shared Intent exclusive mode, the subtree rooted by that node is locked explicitly in shared mode, and that explicit locking is being done at lower level with exclusive mode.

Shared lock (S)

-T can tolerate concurrent readers but not concurrent updaters in R.

Exclusive lock (X)

-T cannot tolerate any concurrent access to R at all.

**10. Describe Database Recovery concepts.**

**Crash Recovery**

Though we are living in highly technologically advanced era where hundreds of satellite monitor the earth and at every second billions of people are connected through information technology, failure is expected but not every time acceptable.

DBMS is highly complex system with hundreds of transactions being executed every second. Availability of DBMS depends on its complex architecture and underlying hardware or system software. If it fails or crashes amid transactions being executed, it is expected that the system would follow some sort of algorithm or techniques to recover from crashes or failures.

**Failure Classification**

To see where the problem has occurred we generalize the failure into various categories, as follows:

**TRANSACTION FAILURE**

When a transaction is failed to execute or it reaches a point after which it cannot be completed successfully it has to abort. This is called transaction failure. Where only few transaction or process are hurt.

**Reason for transaction failure could be:**

- **Logical errors:** where a transaction cannot complete because of it has some code error or any internal error condition
- **System errors:** where the database system itself terminates an active transaction because DBMS is not able to execute it or it has to stop because of some system condition. For example, in case of deadlock or resource unavailability systems aborts an active transaction.

**SYSTEM CRASH**

There are problems, which are external to the system, which may cause the system to stop abruptly and cause the system to crash. For example interruption in power supply, failure of underlying hardware or software failure.

Examples may include operating system errors.

**DISK FAILURE:**

In early days of technology evolution, it was a common problem where hard disk drives or storage drives used to fail frequently.

Disk failures include formation of bad sectors, unreachability to the disk, disk head crash or any other failure, which destroys all or part of disk storage

**Storage Structure**

We have already described storage system here. In brief, the storage structure can be divided in various categories:

- **Volatile storage:** As name suggests, this storage does not survive system crashes and mostly placed very closed to CPU by embedding them onto the chipset itself for examples: main memory, cache memory. They are fast but can store a small amount of information.
- **Nonvolatile storage:** These memories are made to survive system crashes. They are huge in data storage capacity but slower in accessibility. Examples may include, hard disks, magnetic tapes, flash memory, non-volatile (battery backed up) RAM.

**Recovery and Atomicity**

When a system crashes, it many have several transactions being executed and various files opened for them to modifying data items. As we know that transactions are made of various operations, which are atomic in nature. But according to ACID properties of DBMS, atomicity of transactions as a whole must be maintained that is, either all operations are executed or none.

**When DBMS recovers from a crash it should maintain the following:**

- It should check the states of all transactions, which were being executed.
- A transaction may be in the middle of some operation; DBMS must ensure the atomicity of transaction in this case.
- It should check whether the transaction can be completed now or needs to be rolled back.
- No transactions would be allowed to left DBMS in inconsistent state.
- There are two types of techniques, which can help DBMS in recovering as well as maintaining the atomicity of transaction:
- Maintaining the logs of each transaction, and writing them onto some stable storage before actually modifying the database.
- Maintaining shadow paging, where are the changes are done on a volatile memory and later the actual database is updated.

**Log-Based Recovery**

Log is a sequence of records, which maintains the records of actions performed by a transaction. It is important that the logs are written prior to actual modification and stored on a stable storage media, which is failsafe.

Log based recovery works as follows:

- The log file is kept on stable storage media
- When a transaction enters the system and starts execution, it writes a log about it

  $<T_n, \text{Start}>$

- When the transaction modifies an item X, it write logs as follows:

  $<T_n, X, V_1, V_2>$

- It reads Tn has changed the value of X, from V1 to V2.
- When transaction finishes, it logs:

  $<T_n, \text{commit}>$

Database can be modified using two approaches:

**Deferred database modification:** All logs are written on to the stable storage and database is updated when transaction commits.
**Immediate database modification:** Each log follows an actual database modification. That is, database is modified immediately after every operation.

Recovery with concurrent transactions

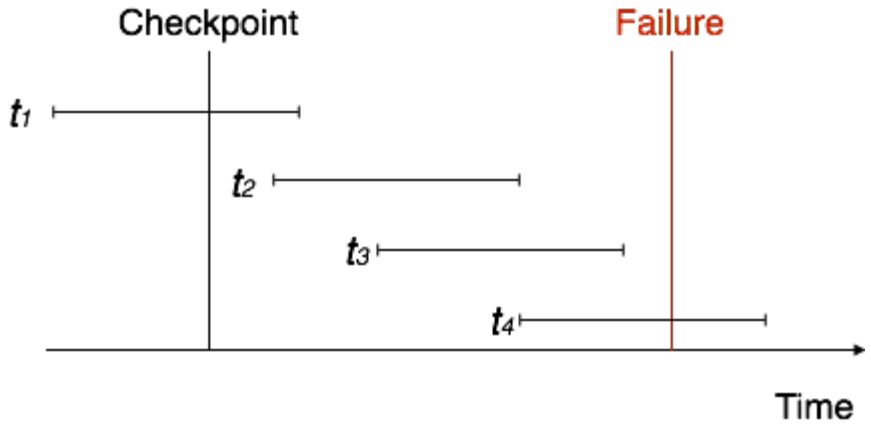
When more than one transactions are being executed in parallel, the logs are interleaved. At the time of recovery it would become hard for recovery system to backtrack all logs, and then start recovering. To ease this situation most modern DBMS use the concept of 'checkpoints'.

**CHECKPOINT**

Keeping and maintaining logs in real time and in real environment may fill out all the memory space available in the system. At time passes log file may be too big to be handled at all. Checkpoint is a mechanism where all the previous logs are removed from the system and stored permanently in storage disk. Checkpoint declares a point before which the DBMS was in consistent state and all the transactions were committed.

**RECOVERY**

When system with concurrent transaction crashes and recovers, it does behave in the following manner:




The recovery system reads the logs backwards from the end to the last Checkpoint.

- It maintains two lists, undo-list and redo-list.
- If the recovery system sees a log with $<T_n$, Start> and $<T_n$, Commit> or just $<T_n$, Commit>, it puts the transaction in redo-list.
- If the recovery system sees a log with $<T_n$, Start> but no commit or abort log found, it puts the transaction in undo-list.
- All transactions in undo-list are then undone and their logs are removed. All transaction in redo-list, their previous logs are removed and then redone again and log saved.

## Unit 4
## TRENDS IN DATABASE TECHNOLOGY

**1. What is B-Tree?**

A B-tree eliminates the redundant storage of search-key values .It allows search key values to appear only once.

**2. What is a B+-Tree index?**

A B+-Tree index takes the form of a balanced tree in which every path from the root of the root of the root of the tree to a leaf of the tree is of the same length.

**3. What is a hash index?**

A hash index organizes the search keys, with their associated pointers, into a hash file structure.

**4. Define seek time.**

The time for repositioning the arm is called the seek time and it increases with the distance that the arm is called the seek time.

**5. Define rotational latency time.**

The time spent waiting for the sector to be accessed to appear under the head is called the rotational latency time.

**6. What is called mirroring?**

The simplest approach to introducing redundancy is to duplicate every disk. This technique is called mirroring or shadowing.

**7. What are the two main goals of parallelism?**

Load –balance multiple small accesses, so that the throughput of such accesses increases.
Parallelize large accesses so that the response time of large accesses is reduced

**8. What are the factors to be taken into account when choosing a RAID level?**

Monetary cost of extra disk storage requirements.
Performance requirements in terms of number of I/O operations
Performance when a disk has failed.
Performances during rebuild.

**9. What is an index?**

An index is a structure that helps to locate desired records of a relation quickly, without examining all records

**10. What are the types of storage devices?**

Primary storage, Secondary storage, Tertiary storage, Volatile storage, Nonvolatile storage

**11. What is called remapping of bad sectors?**

If the controller detects that a sector is damaged when the disk is initially formatted, or when an attempt is made to write the sector, it can logically map the sector to a different physical location.

**12. What is meant by software and hardware RAID systems?**

RAID can be implemented with no change at the hardware level, using only software modification. Such RAID implementations are called software RAID systems and the systems with special hardware support are called hardware RAID systems.

**13. Define hot swapping?**

Hot swapping permits the removal of faulty disks and replaces it by new ones without turning power off. Hot swapping reduces the mean time to repair.

**14. What are the ways in which the variable-length records arise in database systems?**

Storage of multiple record types in a file.
Record types that allow variable lengths for one or more fields.
Record types that allow repeating fields.

**15. What are the two types of blocks in the fixed –length representation? Define them.**

- Anchor block: Contains the first record of a chain.
- Overflow block: Contains the records other than those that are the first record of a chain.

**16. What is hashing file organization?**

In the hashing file organization, a hash function is computed on some attribute of each record. The result of the hash function specifies in which block of the file the record should be placed.

**17. What are called index-sequential files?**

The files that are ordered sequentially with a primary index on the search key are called index-sequential files.

**18. Define Primary index and Secondary Index**

**It is** in a sequentially ordered file, the index whose search key specifies the sequential order of the file. Also called **clustering index.** The search key of a primary index is usually but not necessarily the primary key

**It is** an index whose search key specifies an order different from the sequential order of the file. Also called nonclustering index**.**

**19. Define Datamarts.**

A data mart is the access layer of the data warehouse environment that is used to get data out to the users. The data mart is a subset of the data warehouse that is usually oriented to a specific business line or team. Data marts are small slices of the data warehouse..

**20. Define Distributed Database Systems**

Database spread over multiple machines (also referred to as sites or nodes).Network interconnects the machines. Database shared by users on multiple machines is called Distributed Database Systems

**21. Define data mining and Data warehouse**

It refers to the mining or discovery of new information in terms of patterns or rules from vast amounts of data.

Data warehouses provide access to data for complex analysis, knowledge discovery, and decision making

**22. What are the advantages of web-DBMS?**

Platform independence, graphical user interface, transparent network access, scalable deployment

**23. What are the issues to be considered for mobile database?**

Data distribution, replication, recovery and fault tolerance

**24. What is spatial data?**

Spatial data include geographic data, such as maps and associated information and computer aided design data

**25. What is multimedia data?explain about multimedia database**

Multimedia data typically means digital images,audio, video, animation and graphics together with text data.

– The acquisition, generation, storage and processing of multimedia data in computers and transmission over networks have grown tremendously in the recent past

– consistency, concurrency, integrity, security and availability of data.

**26. Types of Distributed Database**

- Homogenous distributed DB
- Heterogeneous distributed DB

**27.Define fragmentation in Distributed Database**

The system partitions the relation into several fragment and stores each fragment at different sites

- Two approaches
  - Horizontal fragmentation
  - Vertical fragmentation

**Part-B**

**1. Describe File Organization.**

-The database is stored as a collection of ***files***.

- Each file is a sequence of ***records***.

-A **record** is a sequence of fields.

-Classifications of records

i. **Fixed length record**

ii. **Variable length record**

Fixed length record approach:

b. assume record size is fixed

c. each file has records of one particular type only

d. Different files are used for different relations. This case is easiest to implement.

- Simple approach:
  - Record access is simple
    - Modification: do not allow records to cross block boundaries

**Variable length record**

- Deletion of record *I:*
  alternatives*:*
  - move records $i + 1, \ldots, n$ to $i, \ldots, n - 1$
  - do not move records, but link all free records on a *free list*
- Variable-length records arise in database systems in several ways:
  - Storage of multiple record types in a file.
  - Record types that allow variable lengths for one or more fields.

- Byte string representation
  - Attach an *end-of-record* (⊥) control character to the end of each record
  - Difficulty with deletion

Slotted Page Structure

Slotted page header contains:

- number of record entries
- end of free space in the block
- location and size of each record

**Free Lists**

- Store the address of the first deleted record in the file header.
- Use this first record to store the address of the second deleted record, and so on

Pointer method

- A variable-length record is represented by a list of fixed-length records, chained together via pointers.
- Can be used even if the maximum record length is not known

**2. Define RAID and Briefly Explain RAID techniques.**

**RAID: Redundant Arrays of Independent Disks**

a. disk organization techniques that manage a large numbers of disks, providing a view of a single disk of
   i. high capacity and high speed by using multiple disks in parallel, and
   ii. high reliability by storing data redundantly, so that data can be recovered even if a disk fails

**RAID Level 0**: striping; non-redundant.

- Used in high-performance applications where data lost is not critical.

**RAID Level 1**: **Mirroring** (or **shadowing**)

- Duplicate every disk.
- Every write is carried out on both disks
- If one disk in a pair fails, data still available in the other
  - Data loss would occur only if a disk fails, and its mirror disk also fails before the system is repaired
    - Probability of combined event is very small

- **RAID Level 2**: Error-Correcting-Codes (ECC) with bit striping.
- **RAID Level 3**: Bit-Interleaved Parity
  - a single parity bit is enough for error correction, not just detection, since we know which disk has failed
    - When writing data, corresponding parity bits must also be computed and written to a parity bit disk
    - To recover data in a damaged disk, compute XOR of bits from other disks (including parity bit disk)
- **RAID Level 4:** Block-Interleaved Parity.
  - When writing data block, corresponding block of parity bits must also be computed and written to parity disk
  - To find value of a damaged block, compute XOR of bits from corresponding blocks (including parity block) from other disks.
- **RAID Level 5:** Block-Interleaved Distributed Parity; partitions data and parity among all $N + 1$ disks, rather than storing data in $N$ disks and parity in 1 disk.
- **RAID Level 6**: P+Q Redundancy scheme; similar to Level 5, but stores extra redundant information to guard against multiple disk failures.
  - Better reliability than Level 5 at a higher cost; not used as widely.
  -

**3. Explain Secondary storage devices.**

Can differentiate storage into:

b. **volatile storage:** loses contents when power is switched off
c. **non-volatile storage**:
    i. Contents persist even when power is switched off.
    ii. Includes secondary and tertiary storage.

**primary storage:** Fastest media but volatile (cache, main memory).
**secondary storage:** next level in hierarchy, non-volatile, moderately fast access time.
also called **on-line storage**
**tertiary storage:** lowest level in hierarchy, non-volatile, slow access timealso called **off-line storage**

E.g. magnetic tape, optical storage

**Cache** – fastest and most costly form of storage; volatile; managed by the computer system hardware.
**Main memory**:

- fast access
- generally too small
- capacities of up to a few Gigabytes widely used currently
- **Volatile** — contents of main memory are usually lost if a power failure or system crash occurs.

**Flash memory**

- Data survives power failure
- Data can be written at any location.
- The location can be erased and written to again
- Can support only a limited number of write/erase cycles.
- Reads are fast.
- But writes are slow (few microseconds), erase is slower
- also known as EEPROM (Electrically Erasable Programmable Read-Only Memory)

**Magnetic-disk**

a. Data is stored on spinning disk, and read/written magnetically
b. Primary medium for the long-term storage of data.
c. Data must be moved from disk to main memory for access, and written back for storage
    i. Much slower access than main memory
d. **direct-access** – possible to read data on disk in any order, unlike magnetic tape
    i. Much larger capacity than main memory/flash memory
    ii. disk failure can destroy data, but is very rare

**Read-write head**

-Positioned very close to the platter surface
-Reads or writes magnetically.

-Surface of platter divided into circular **tracks**
-Each track is divided into **sectors.**

-A sector is the smallest unit of data that can be read or written.

-Head-disk assemblies

-multiple disk platters on a single spindle (typically 2 to 4)
-ne head per platter, mounted on a common arm.

**Cylinder** *i* consists of $i^{th}$ track of all the platters
**Disk controller** – interfaces between the computer system and the disk drive hardware.

- accepts high-level commands to read or write a sector
- initiates actions such as moving the disk arm to the right track and actually reading or writing the data
- Ensures successful writing by reading back sector after writing it.

- **Optical storage**
    - non-volatile, data is read optically from a spinning disk using a laser
    - CD-ROM and DVD most popular forms
    - Write-one, read-many (WORM) optical disks are available (CD-R and DVD-R)
    - Multiple write versions also available (CD-RW, DVD-RW)
    - Reads and writes are slower than with magnetic disk

- **Tape storage**
  - non-volatile, Used mainly for backup, for storage of infrequently used information, and as an off-line medium for transferring information from one system to another..
- Hold large volumes of data and provide high transfer rates
  - **sequential-access** – much slower than disk
- Very slow access time in comparison to magnetic disks and optical disks
  - very high capacity (300 GB tapes available)
  - storage costs much cheaper than disk, but drives are expensive

**4. Explain about static and dynamic hashing with an example**

Static hashing:

A bucket is a unit of storage containing one or more records.

In a hash file organization we obtain the bucket of a record directly from its search-key value using a hash function.

Hash function h is a function from the set of all search-key values K to the set of all bucket addresses B.

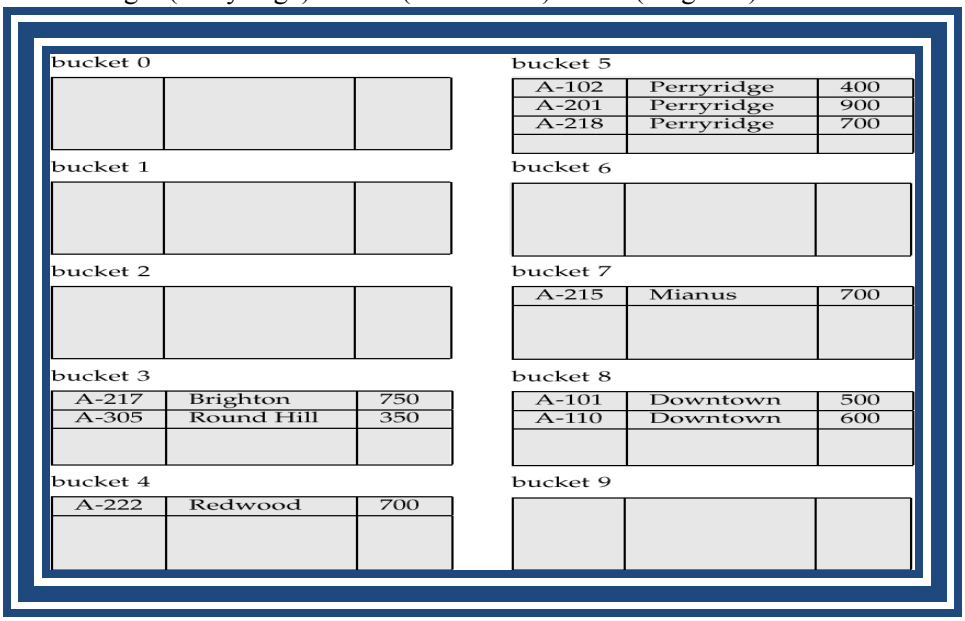
Hash function is used to locate records for access, insertion as well as deletion.

Example:

-There are 10 buckets,

-The hash function returns the sum of the binary representations of the characters modulo 10

- E.g. h(Perryridge) = 5 h(Round Hill) = 3 h(Brighton) = 3




Dynamic hashing:

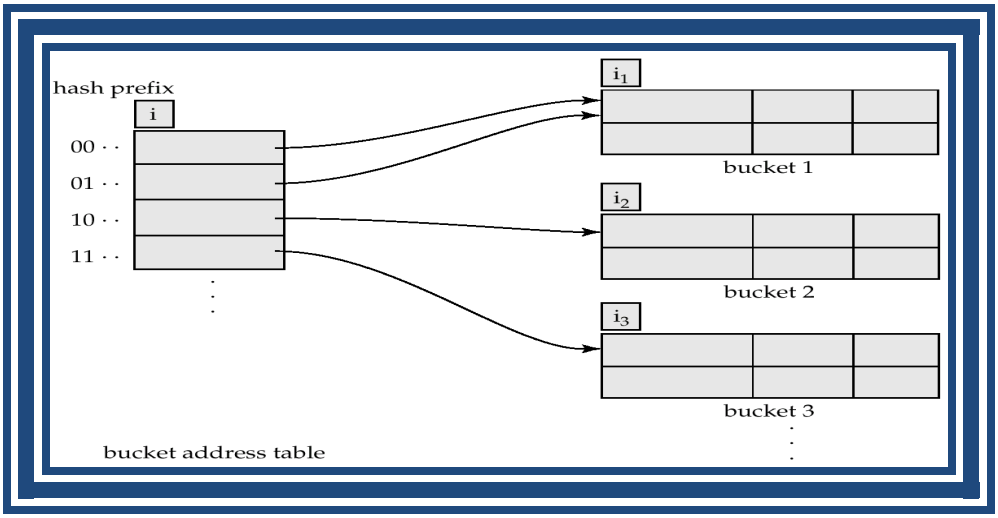
- Good for database that grows and shrinks in size
- Allows the hash function to be modified dynamically
- Extendable hashing – one form of dynamic hashing
  - Hash function generates values over a large range — typically b-bit integers, with

b = 32.

- At any time use only a prefix of the hash function to index into a table of bucket addresses.
- Let the length of the prefix be i bits, $0 \le i \le 32$.
- Bucket address table size = $2^{i}$. Initially i = 0
- Value of i grows and shrinks as the size of the database grows and shrinks.
- Multiple entries in the bucket address table may point to a bucket.
- Thus, actual number of buckets is $< 2^{i}$

– The number of buckets also changes dynamically due to coalescing and splitting of buckets.

**General Extendable Hash Structure**




In this structure, $i_2 = i_3 = i$, whereas $i_1 = i - 1$ (see next slide for details)

**Use of Extendable Hash Structure**

- To locate the bucket containing search-key $K_j$:
  1. Compute $h(K_j) = X$
  2. Use the first i high order bits of X as a displacement into bucket address table, and follow the pointer to appropriate bucket

**Updates in Extendable Hash Structure**

- To insert a record with search-key value $K_j$
  - follow same procedure as look-up and locate the bucket, say j.
  - If there is room in the bucket j insert record in the bucket.
  - Overflow buckets used instead in some cases.
- To delete a key value,
  - locate it in its bucket and remove it.
  - The bucket itself can be removed if it becomes empty
  - Coalescing of buckets can be done
  - Decreasing bucket address table size is also possible

**5. Explain about Multidimensional and Parallel with an example**

MULTIDIMENSIONAL DATABASES:

A multidimensional database (MDB) is a type of database that is optimized for data warehouse and online analytical processing (OLAP) applications. Multidimensional databases are frequently created using input from existing relational databases. Whereas a relational database is typically accessed using a Structured Query Language (SQL) query, a multidimensional database allows a user to ask questions like "How many Aptivas have been sold in Nebraska so far this year?" and similar questions related to summarizing business operations and trends. An OLAP application that accesses data from a multidimensional database is known as a MOLAP (multidimensional OLAP) application.

A multidimensional database - or a multidimensional database management system (MDDBMS) - implies the ability to rapidly process the data in the database so that answers can be generated quickly. A number of vendors provide products that use multidimensional databases. Approaches to how data is stored and the user interface vary.Conceptually, a multidimensional database uses the idea of a data cube to represent the dimensions of data available to a user. For example, "sales" could be viewed in the dimensions of

product model, geography, time, or some additional dimension. In this case, "sales" is known as the *measure attribute* of the data cube and the other dimensions are seen as *feature attributes*. Additionally, a database creator can define hierarchies and levels within a dimension (for example, state and city levels within a regional hierarchy).

PARALLEL DATABASES:

A parallel database is designed to take advantage of such architectures by running multiple instances which "share" a single physical database. In appropriate applications, a parallel server can allow access to a single database by users on multiple machines, with increased performance.

A parallel server processes transactions in parallel by servicing a stream of transactions using multiple CPUs on different nodes, where each CPU processes an entire transaction. Using parallel data manipulation language you can have one transaction being performed by multiple nodes. This is an efficient approach because many applications consist of online insert and update transactions which tend to have short data access requirements. In addition to balancing the workload among CPUs, the parallel database provides for concurrent access to data and protects data integrity.

Key elements of parallel processing:

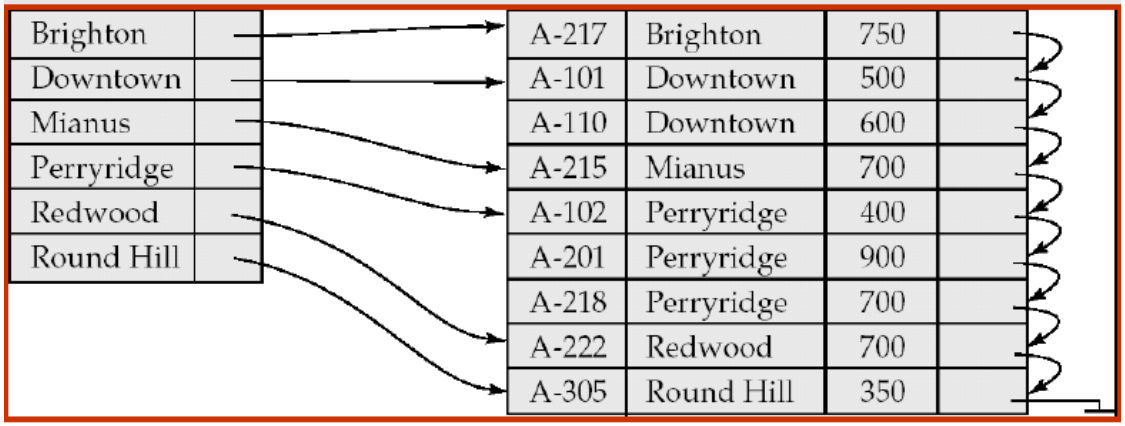
- Speedup and Scaleup: the Goals of Parallel Processing
- Synchronization: A Critical Success Factor
- Locking
- Messaging

**6. Explain about ordered indices with an example**

In an **ordered index,** index entries are stored sorted on the search key value.
**Primary index:** in a sequentially ordered file, the index whose search key specifies the sequential order of the file.
**Secondary index**: an index whose search key specifies an order different from the sequential order of the file.

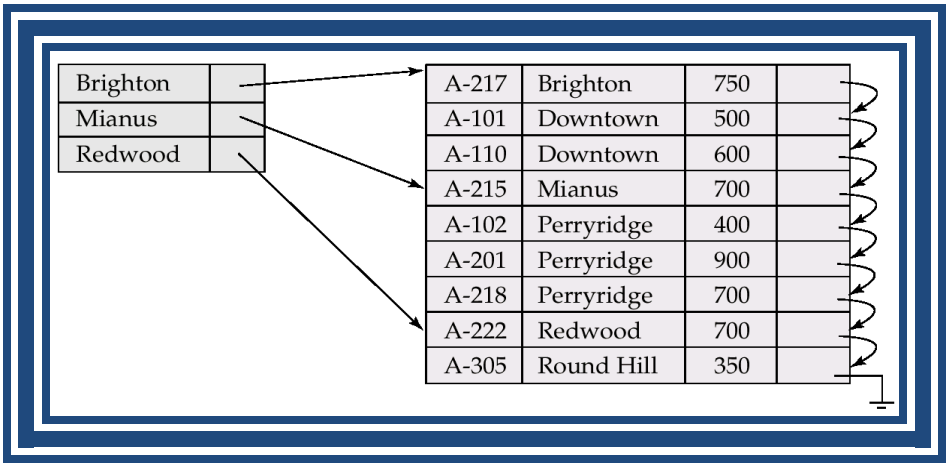
Types of Ordered Indices
- Dense index
- Sparse index

Dense index — Index record appears for every search-key value in the file.



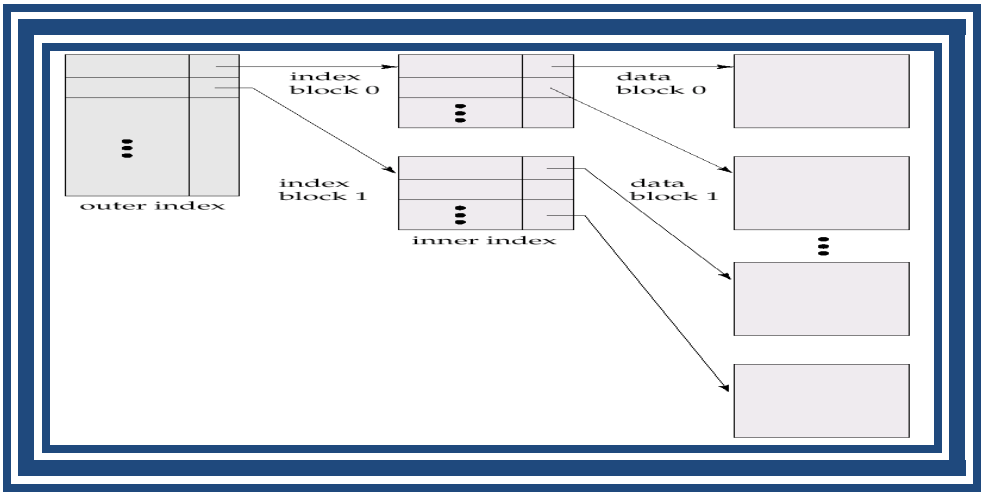

Sparse Index: contains index records for only some search-key values.
- Applicable when records are sequentially ordered on search-key

To locate a record with search-key value *K* we:
- Find index record with largest search-key value < *K*
- Search file sequentially starting at the record to which the index record points




Multilevel index

- If primary index does not fit in memory, access becomes expensive.
- To reduce number of disk accesses to index records, treat primary index kept on disk as a sequential file and construct a sparse index on it.
- outer index – a sparse index of primary index
- inner index – the primary index file
- If even outer index is too large to fit in main memory, yet another level of index can be created, and so on.




**Secondary Indices**

- Index record points to a bucket that contains pointers to all the actual records with that particular searchkey value.
- Secondary indices have to be dense

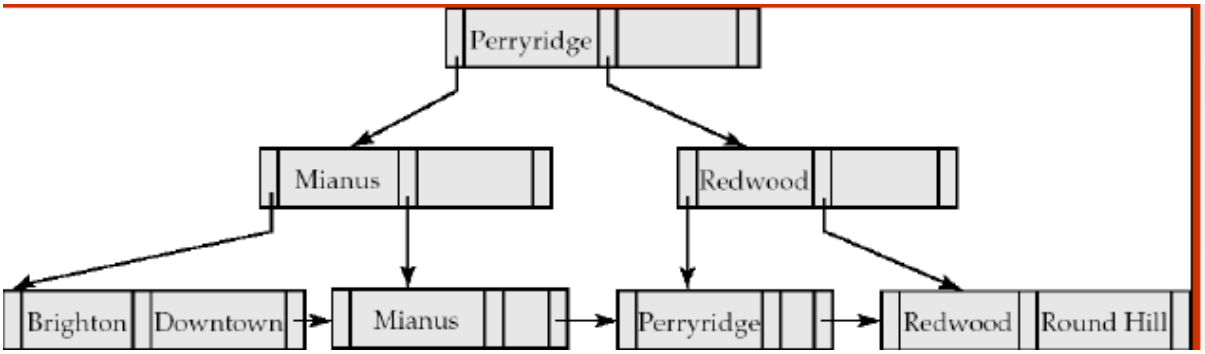
**7. Explain about B$^+$ trees indexing concepts with an example**

**Disadvantage of indexed-sequential files**: performance degrades as file grows, since many overflow blocks get created. Periodic reorganization of entire file is required.
**Advantage of B$^+$-tree index files**: automatically reorganizes itself with small, local, changes, in the face of insertions and deletions. Reorganization of entire file is not required to maintain performance.
**Disadvantage of B$^+$-trees:** extra insertion and deletion overhead, space overhead.

A B$^+$-tree is a rooted tree satisfying the following properties:

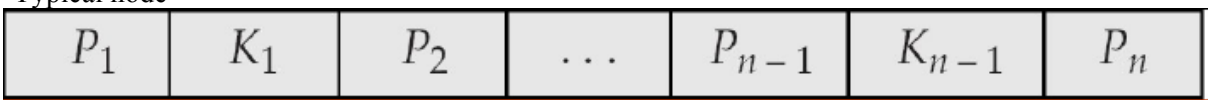
- All paths from root to leaf are of the same length
- Each node that is not a root or a leaf has between $[n/2]$ and $n$ children.
- Special cases:

- If the root is not a leaf, it has at least 2 children.
- If the root is a leaf, it can have between 0 and ($n$–1) values.




**B+Tree Node Structure**

-Typical node




*Ki are the search-key values

*Pi are pointers to children (for non-leaf nodes) or pointers to records or buckets of records (for leaf nodes).

-The searchkeys in a node are ordered

$K1 < K2 < K3 < . . . < Kn–1$

Properties of leaf node

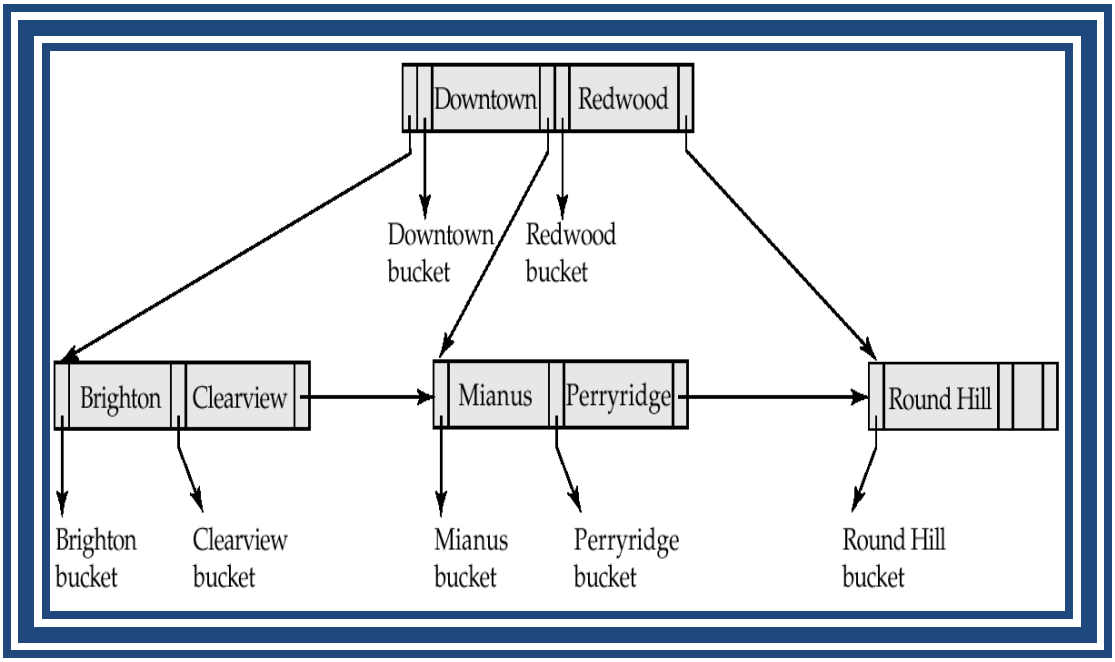
- For $i$ = 1, 2, . . ., $n$–1, pointer $P_i$ either points to a file record with search-key value $K_i$, or to a bucket of pointers to file records, each record having search-key value $K_i$.
- $P_n$ points to next leaf node in search-key order

Non-Leaf Nodes in $B^+$-Trees

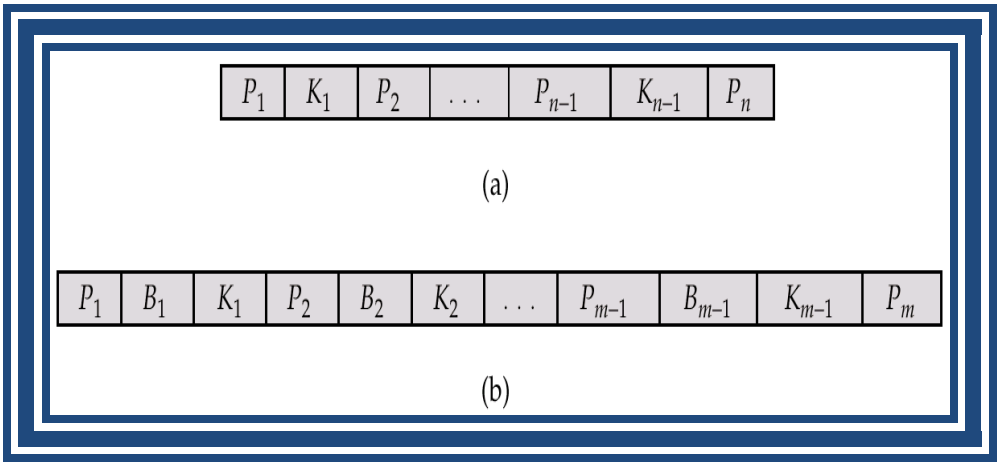
Non leaf nodes form a multi-level sparse index on the leaf nodes. For a non-leaf node with $m$ pointers:All the search-keys in the subtree to which $P_1$ points are less than $K_1$.

**8. Explain about B trees indexing concepts with an example**

- Similar to B+-tree, but B-tree allows search-key values to appear only once; eliminates redundant storage of search keys.
- Search keys in nonleaf nodes appear nowhere else in the B-tree; an additional pointer field for each search key in a nonleaf node must be included.




Generalized B-tree leaf node



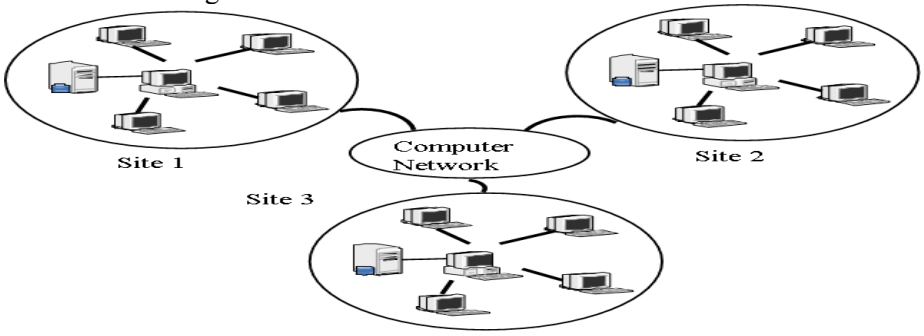

Nonleaf node – pointers $B_i$ are the bucket or file record pointers.

**9. Describe the concepts of Distributed Databases with neat Diagram**

In distributed database system data reside in several location where as centralized database system the data reside in single location

Classification

- Homogenous distributed DB
- Heterogeneous distributed DB




- Homogenous distributed DB

  - All sites have identical database management software, are aware of one another.
  - Agree to cooperate in processing users' request.
- Heterogeneous distributed DB
  - different sites may use different schemas and different DBMS software.
  - The sites may not be aware of one another
  - Provide only limited facilities for cooperation in transaction processing.
- Consider a relation r, there are two approaches to store this relation in the distributed DB.
  - Replication
  - Fragmentation
- Replication
  - The system maintains several identical replicas(copies) of the relation at different site.
  - Full replication- copy is stored in every site in the system.
- Advantages and disadvantages
  - Availability
  - Increased parallelism

  Increased overhead update
- Fragmentation
  - The system partitions the relation into several fragment and stores each fragment at different sites
  - Two approaches
    - Horizontal fragmentation
    - Vertical fragmentation

  Horizontal fragmentation

  Splits the relation by assigning each tuple of r to one or more fragments

  relation r is partitioned into a number of subsets, $r_1$ ,$r_2$,…..$r_n$ and can be reconstruct the original relation using union of all fragments, that is

$$r = r_1 \cup r_2 \cup \ldots\ldots \cup r_n$$

- Vertical fragmentation
  - Splits the relation by decomposing scheme R of relation and reconstruct the original relation by using natural join of all fragments. that is

$$r = r_1 \quad r_2 \quad \ldots\ldots \quad r_n$$

**10. Explain the concepts of Mobile and web databases.**

**Mobile database**

A **mobile database** is either a stationary database that can be connected to by a mobile computing device (e.g., smart phones and PDAs) over a mobile network, or a database which is actually stored by the mobile device. This could be a list of contacts, price information, distance travelled, or any other information

**Web databases**

**Web Database** is a web page API for storing data in databases that can be queried using a variant of SQL

11.**Explain Spatial and Multimedia Databases.**

**Multimedia DBMS**

A multimedia database management system (MM-DBMS) is a framework that manages different types of data potentially represented in a wide diversity of formats on a wide array of media sources.

Like the traditional DBMS, MM-DBMS should address requirements:

*Integration*

- Data items do not need to be duplicated for different programs

*Data independence*
- Separate the database and the management from the application programs

*Concurrency control*
- allows concurrent transactions

Requirements of Multimedia DBMS

*Persistence*
- Data objects can be saved and re-used by different transactions and program invocations

*Privacy*
- Access and authorization control

*Integrity control*
- Ensures database consistency between transactions

*Recovery*
- Failures of transactions should not affect the persistent data storage

*Query support*
- Allows easy querying of multimedia data

**Spatial Database**

- A SDBMS is a DBMS
- It offers spatial data types/data models/ query language
  - Support spatial properties/operations
- It supports spatial data types in its implementation
  - Support spatial indexing, algorithms for spatial selection and join

Spatial Database Applications

- GIS applications (maps):
  - Urban planning, route optimization, fire or pollution monitoring, utility networks, etc
- Other applications:
  - VLSI design, CAD/CAM, model of human brain, etc
- Traditional applications:
  - Multidimensional records

## Unit 5
## ADVANCED TOPICS

**1. Define Database security.**

Database security concerns the use of a broad range of information security controls to protect databases (potentially including the data, the database applications or stored functions, the database systems, the database servers and the associated network links) against compromises of their confidentiality, integrity and availability

**2. What are the Security risks to database systems?**

Unauthorized or unintended activity or misuse by authorized database users, database administrators, or network/systems managers, or by unauthorized users or hackers. For example inappropriate access to sensitive data, metadata or functions within databases, or inappropriate changes to the database programs, structures or security configurations

**3.List out the types of information security.**

- Access control
- Auditing
- Authentication
- Encryption
- Integrity controls
- Backups
- Application security
- Database Security applying Statistical Method

**4.Define Authentication.**

Authentication is the act of determining the identity of a user and of the host that they are using. The goal of authentication is to first verify that the user, either a person or system, which is attempting to interact with your system is allowed to do so. The second goal of authentication is to gather information regarding the way that the user is accessing your system

**5.What is Authorization**

Authorization is the act of determining the level of access that an authorized user has to behavior and data.

**6. Define Check Point.**

This is the place to validate users and to make appropriate decisions when dealing with security breaches. Also known as Access Verification, Validation and Penalization, and Holding off Hackers.

**7.Define Privileges.**

A separate database administrator (DBA) account should be created and protected that has full privileges to create/drop databases, create user accounts, and update user privileges. This simple means of separation of responsibility helps prevent accidental mis-configuration, lowers risk and lowers scope of compromise.

8.What is Cryptography in database.

Cryptography is the art of "extreme information security." It is extreme in the sense that once treated with a cryptographic algorithm; a message (or a database field) is expected to remain secure even if the adversary has full access to the treated message. The adversary may even know which algorithm was used. If the cryptography is good, the message will remain secure.

**9. What is Benefits of Database Encryption?**

Ensure guaranteed access to encrypted data by authorized users by automating storage and back-up for mission critical master encryption keys.Simplify data privacy compliance obligations and reporting activities through the use of a security-certified encryption and key management to enforce critical best practices and other standards of due care.

**10.What is statistical database**

A statistical database is a database used for statistical analysis purposes. It is an OLAP (online analytical processing), instead of OLTP (online transaction processing) system. Statistical databases contain parameter data and the measured data for these parameters.

**11.Define OLTP.**

Online transaction processing, or OLTP, is a class of information systems that facilitate and manage transaction-oriented applications, typically for data entry and retrieval transaction processing.

**12.Define Database replication.**

Database replication can be used on many database management systems, usually with a master/slave relationship between the original and the copies. The master logs the updates, which then ripple through to the slaves. The slave outputs a message stating that it has received the update successfully, thus allowing the sending of subsequent updates.

**13. What is homogeneous distributed database and heterogeneous distributed database**

A homogeneous distributed database has identical software and hardware running all databases instances, and may appear through a single interface as if it were a single database.

A heterogeneous distributed database may have different hardware, operating systems, database management systems, and even data models for different databases.

**14.What are benefits of a data warehouse**

Congregate data from multiple sources into a single database so a single query engine can be used to present data.Mitigate the problem of database isolation level lock contention in transaction processing systems caused by attempts to run large, long running, analysis queries in transaction processing databases.

**15. Define offline data warehouse and Online database.**

Data warehouses at this stage are updated from data in the operational systems on a regular basis and the data warehouse data are stored in a data structure designed to facilitate reporting.

Online Integrated Data Warehousing represent the real time Data warehouses stage data in the warehouse is updated for every transaction performed on the source data

**16.Write about integrated data warehouse**

These data warehouses assemble data from different areas of business, so users can look up the information they need across other systems.

**17.Define spatial data mining.**

Spatial data mining is the application of data mining methods to spatial data. The end objective of spatial data mining is to find patterns in data with respect to geography.

**18.What is the advantage of OODB?**

An integrated repository of information that is shared by multiple users, multiple products, multiple applications on multiple platforms.

**19.Define Text mining.**

Text mining is also referred to as text data mining, it is equivalent to text analytics, refers to the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning.

**20.Define XML Database.**

An XML database is a data persistence software system that allows data to be stored in XML format. These data can then be queried, exported and serialized into the desired format. XML databases are usually associated with document-oriented databases.

**21.Define Association Rule Mining.**

Association rule mining is discovering frequent patterns, associations and correlations among items which are meaningful to the users and can generate strong rules on the basis of these frequent patterns, which helps in decision support system.

**22.Define Clustering.**

Clustering techniques consider data tuples as objects. They partition the objects into groups or clusters so that objects within a cluster are similar to one another and dissimilar to objects in other clusters.

**23. Define Information Retrival.**

It is an activity of obtaining information resources relevant to an information need from a collection of information resources.

**24.Define Crawling and indexing the web.**

Web Crawling is the process of search engines combing through web pages in order to properly index them. These “web crawlers” systematically crawl pages and look at the keywords contained on the page, the kind of content, all the links on the page, and then returns that information to the search engine’s server for indexing. Then they follow all the hyperlinks on the website to get to other websites. When a search engine user enters a query, the search engine will go to its index and return the most relevant search results based on the keywords in the search term. Web crawling is an automated process and provides quick, up to date data.

**25.Define Relevance Ranking.**

A system in which the search engine tries to determine the theme of a site that a link is coming from.

## Part-B

**1. Explain about Object Oriented Databases and XML Databases.**

**OODB = Object Orientation + Database Capabilities**

Object-Oriented Database Features:

-persistence
- support of transactions
- simple querying of bulk data
- concurrent access

- resilience
- security

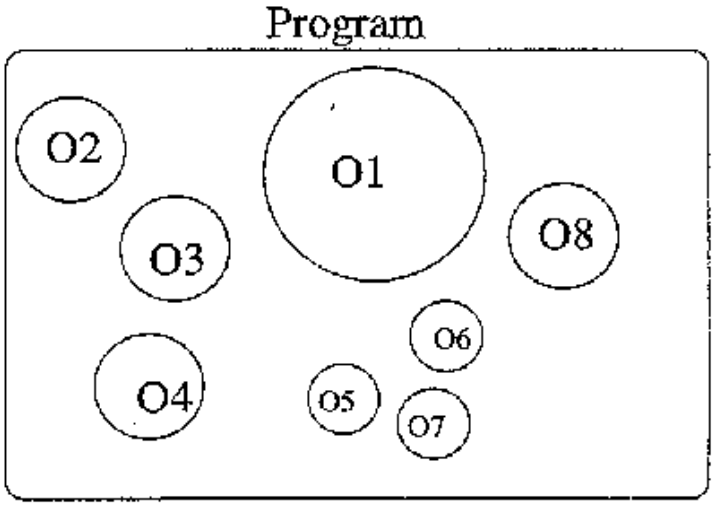
Integration and Sharing

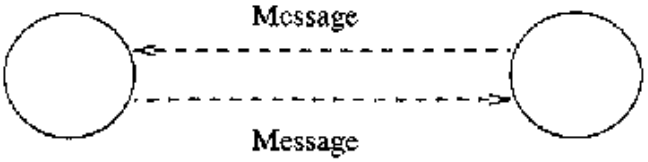
-Seamless integration of operating systems, databases, languages, spreadsheets, word processors, AI expert system shells.

-Sharing of data, information, software components, products, computing environments.

-Referential sharing:
Multiple applications, products, or objects share common sub-objects.

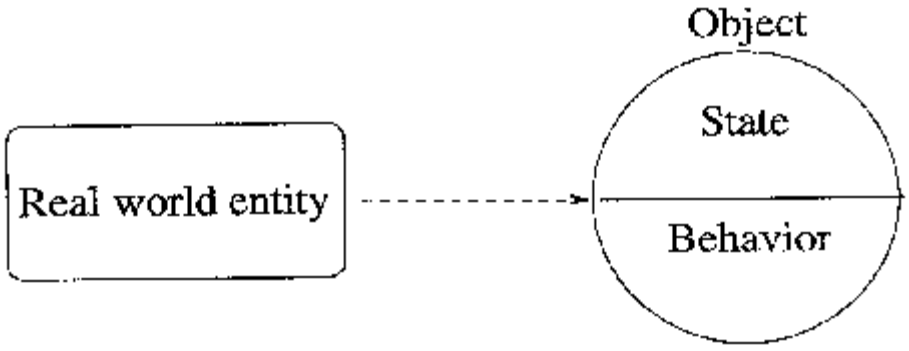
Object-oriented databases allows referential sharing through the support of object identity and inheritance.
The object-oriented paradigm is illustrated below:



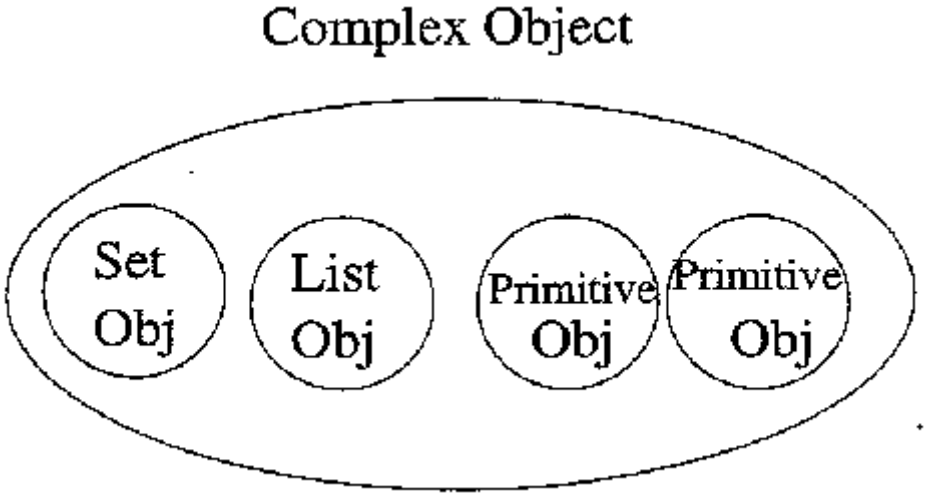

Objects and Identity
The following figure shows object with state and behavior. The state is represented by the values of the object's attributes, and the behavior is defined by the methods acting on the state of the object. There is a

unique object identifier OID to identify the object.



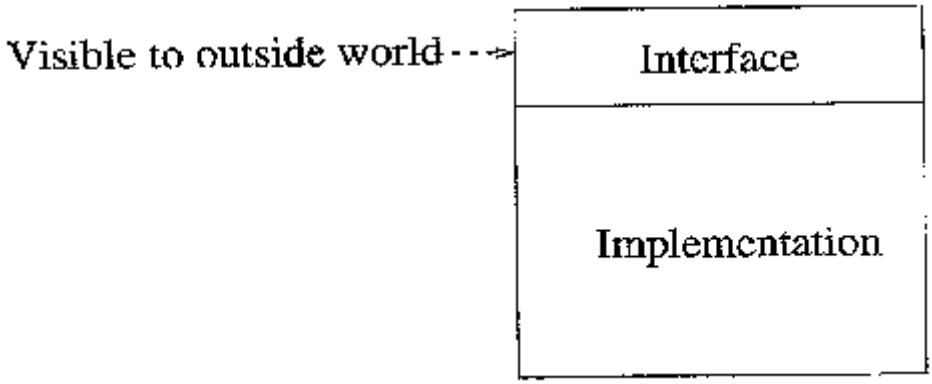

Complex Objects

Complex objects are built by applying constructors to simpler objects including: sets, lists and tuples. An example is illustrated below:




Encapsulation

Encapsulation is derived from the notion of Abstract Data Type (ADT). It is motivated by the need to make a clear distinction between the specification and the implementation of an operation. It reinforces modularity and provides a form of logical data independence.



**Class**

A class object is an object which acts as a template.

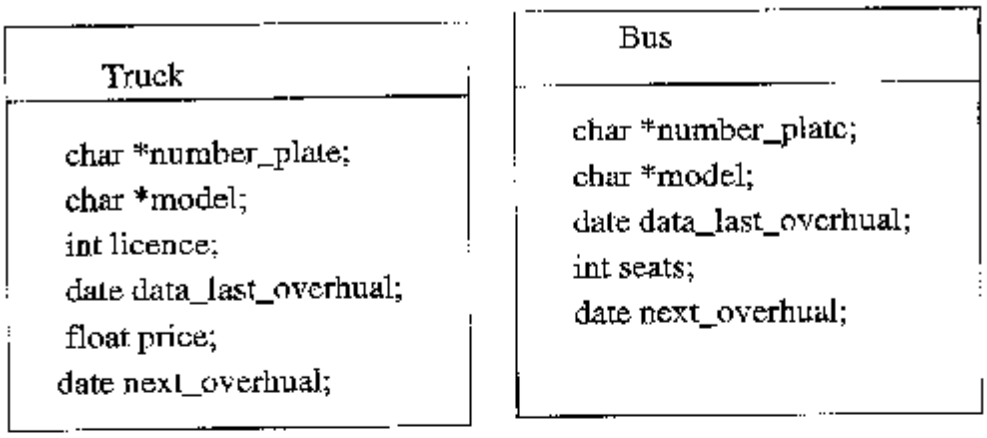
It specifies:
A structure that is the set of attributes of the instances
A set of operations
A set of methods which implement the operations

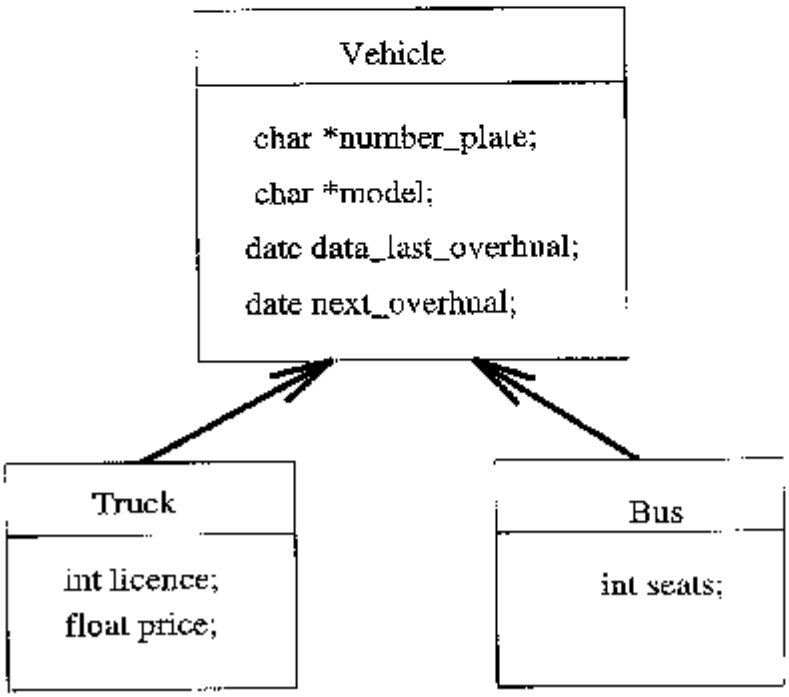
Instantiation means generating objects, Ex. 'new' operation in C++

Persistence of objects: Two approaches
An implicit characteristic of all objects
An orthogonal characteristic - insert the object into a persistent collection of objects

**Inheritance**

A mechanism of reusability, the most powerful concept of OO programming

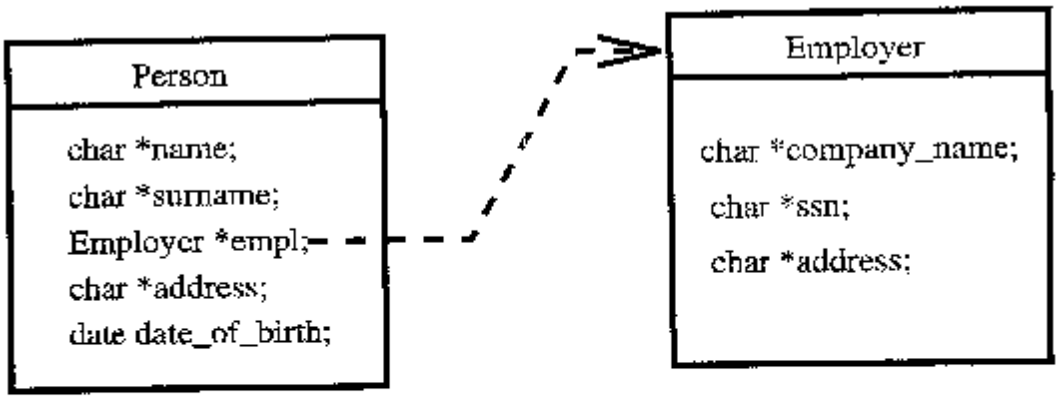


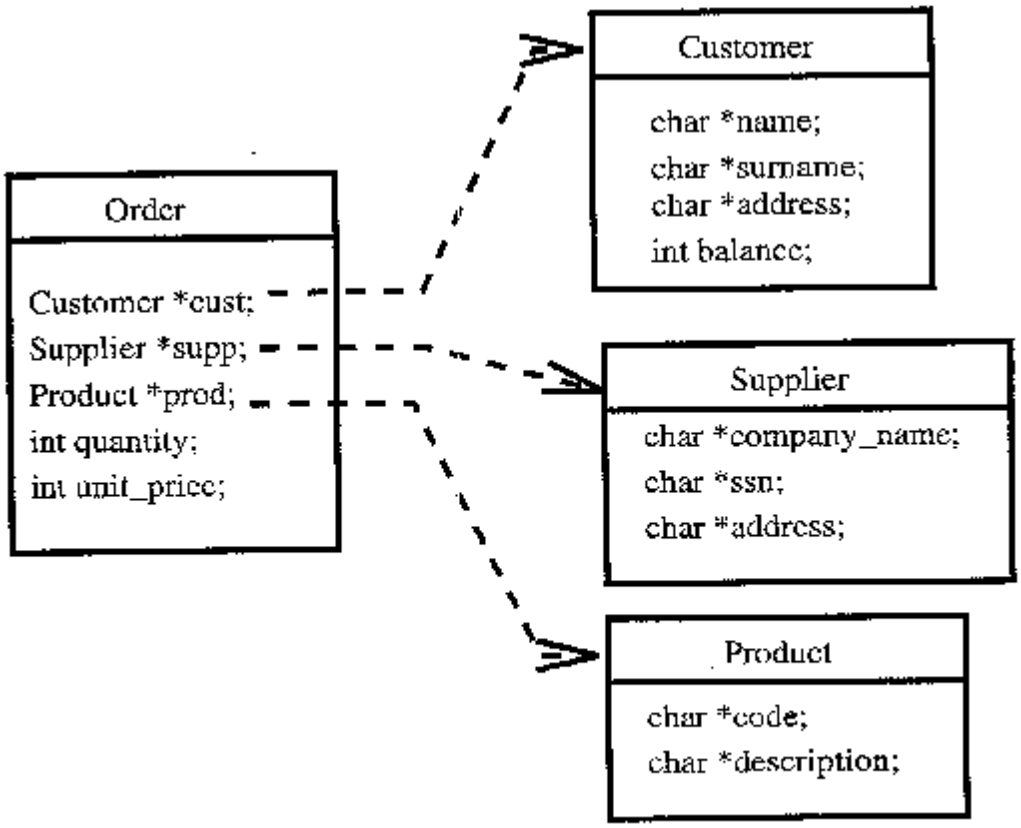
**Association**

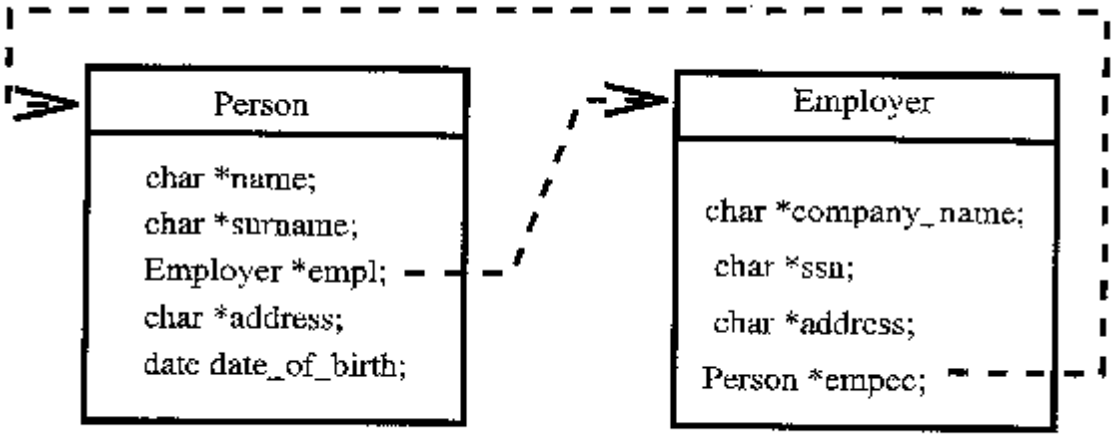
Association is a link between entities in an application

In OODB, associations are represented by means of references between objects
a representation of a binary association
a representation of a ternary association
reverse reference




Representation of a binary association






ADVANTAGES OF OODB

-An integrated repository of information that is shared by multiple users, multiple products, multiple applications on multiple platforms.

- It also solves the following problems:

1. The semantic gap: The real world and the Conceptual model is very similar.

2. Impedance mismatch: Programming languages and database systems must be interfaced to solve application problems. But the language style, data structures, of a programming language (such as C) and the DBMS (such as Oracle) are different. The OODB supports general purpose programming in the OODB framework.

3. New application requirements: Especially in OA, CAD, CAM, CASE, object-orientation is the most natural and most convenient.

**XML database**

An **XML database** is a data persistence software system that allows data to be stored in XML format. These data can then be queried, exported and serialized into the desired format. XML databases are usually associated with document-oriented databases.

Two major classes of XML database exist:[1]

1. **XML-enabled**: these may either map XML to traditional database structures (such as a relational database[2]), accepting XML as input and rendering XML as output, or more recently support native XML types within the traditional database. This term implies that the database processes the XML itself (as opposed to relying on middleware).
2. **Native XML** (NXD): the internal model of such databases depends on XML and uses XML documents as the fundamental unit of storage, which are, however, not necessarily stored in the form of text files.

**2. Explain in detail (i) Clustering (ii) Information Retrieval (iii) Transaction processing.**

**(i) Clustering**

Clustering, in the context of databases, refers to the ability of several servers or instances to connect to a single database. An instance is the collection of memory and processes that interacts with a database, which is the set of physical files that actually store data.

(ii) Information Retrieval

**Information retrieval** is the activity of obtaining information resources relevant to an information need from a collection of information resources. Searches can be based on metadata or on full-text indexing.

**(iii) Transaction processing.**

It is type of information system that collects stores, retrieves and modifies the data transaction of enterprises.

**3. Explain about Types of Privileges in database language**

A privilege is a right to execute a particular type of SQL statement or to access another user's object. Some examples of privileges include the right to:

- Connect to the database
- Create a table
- Select rows from another user's table
- Execute another user's stored procedure

You grant privileges to users so these users can accomplish tasks required for their jobs. You should grant a privilege only to a user who requires that privilege to accomplish the necessary work. Excessive granting of unnecessary privileges can compromise security. A user can receive a privilege in two different ways:

- You can grant privileges to users explicitly. For example, you can explicitly grant to user `SCOTT` the privilege to insert records into the `employees` table.
- You can also grant privileges to a role (a named group of privileges), and then grant the role to one or more users. For example, you can grant the privileges to select, insert, update, and delete records from the `employees` table to the role named `clerk`, which in turn you can grant to users `scott` and `brian`.

**Types:**

- System Privileges
- Schema Object Privileges
- Table Privileges
- View Privileges
- Procedure Privileges
- Type Privileges

**4. Explain about Threats and risks in database security.**

Any situation or event, whether international or uninternational, that will adversely effect a sys & consequently an org...

1: HARDWARE:

# Fire, Flood, Bomb

# Power Failure, Fluctuations

# Thrteats of equipments
# Physical Damage
# Radiation(mob etc)

2: COMMUNICATION NETWORK:
Ensure that no wiring break

3: DBMS:
# Theft
# Greater Access

4: USER:
# Hacking
# Viewing & Disclosing

5: Programing & Operators:
# Less Trained
# Creates Unsequre SW

Security access control (SAC) is an important aspect of any system. Security access control is the act of ensuring that an authenticated user accesses only what they are authorized to and no more. The bad news is that security is rarely at the top of people's lists, although mention terms such as data confidentiality, sensitivity, and ownership and they quickly become interested. The good news is that there is a wide range of techniques that you can apply to help secure access to your system. The bad news is that as Mitnick and Simon (2002) point out "…the human factor is the weakest link. Security is too often merely an illusion, an illusion sometimes made even worse when gullibility, naivette, or ignorance come into play." The go on to say that "security is not a technology problem – it's a people and management problem." Having said that, my experience is that the "technology factor" and the "people factor" go hand in hand; you need to address both issues to succeed.

This article overviews the issues associated with security access control within your system. Although it includes a brief discussion of authentication, the primary focus is on authorization, assuring that users have access to the functionality and information that they require and no more. The issues surrounding authorization are explored in detail as well as both database and object-oriented implementation strategies. As with other critical implementation issues, such as referential integrity and concurrency control, it isn't a black and white world. A "pure object" approach will likely prove to be insufficient as will a "pure database" approach, instead you will need to mix and match techniques.

**5. Explain about Data Warehousing with an example**

A data warehouse (DW, DWH), or an enterprise data warehouse (EDW), is a database used for reporting and data analysis. Integrating data from one or more disparate sources creates a central repository of data, a data warehouse (DW). Data warehouses store current and historical data and are used for creating trending reports for senior management reporting such as annual and quarterly comparisons.

**Benefits of a data warehouse**

A data warehouse maintains a copy of information from the source transaction systems. This architectural complexity provides the opportunity to :

- Congregate data from multiple sources into a single database so a single query engine can be used to present data.
- Mitigate the problem of database isolation level lock contention in transaction processing systems caused by attempts to run large, long running, analysis queries in transaction processing databases.
- Maintain data history, even if the source transaction systems do not.
- Integrate data from multiple source systems, enabling a central view across the enterprise. This benefit is always valuable, but particularly so when the organization has grown by merger.
- Improve data quality, by providing consistent codes and descriptions, flagging or even fixing bad data.
- Present the organization's information consistently.
- Provide a single common data model for all data of interest regardless of the data's source.
- Restructure the data so that it makes sense to the business users.
- Restructure the data so that it delivers excellent query performance, even for complex analytic queries, without impacting the operational systems.
- Add value to operational business applications, notably customer relationship management (CRM) systems.
- Making decision–support queries easier to write.

**Generic data warehouse environment**

The environment for data warehouses and marts includes the following:

- Source systems that provide data to the warehouse or mart;
- Data integration technology and processes that are needed to prepare the data for use;
- Different architectures for storing data in an organization's data warehouse or data marts;
- Different tools and applications for the variety of users;
- Metadata, data quality, and governance processes must be in place to ensure that the warehouse or mart meets its purposes.

In regards to source systems listed above, Rainer states, “A common source for the data in data warehouses is the company’s operational databases, which can be relational databases”.Regarding data integration, Rainer states, “It is necessary to extract data from source systems, transform them, and load them into a data mart or warehouse”. Rainer discusses storing data in an organization’s data warehouse or data marts.”.Metadata are data about data. “IT personnel need information about data sources; database, table, and column names; refresh schedules; and data usage measures“. Today, the most successful companies are those that can respond quickly and flexibly to market changes and opportunities. A key to this response is the effective and efficient use of data and information by analysts and managers. A “data warehouse” is a repository of historical data that are organized by subject to support decision makers in the organization. Once data are stored in a data mart or warehouse, they can be accessed.

**6. Explain about Data mining with an example**

**Data mining** an interdisciplinary subfield of computer science, is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use.

Data mining involves six common classes of tasks:

- Anomaly detection (Outlier/change/deviation detection) – The identification of unusual data records, that might be interesting or data errors that require further investigation.

- Association rule learning (Dependency modeling) – Searches for relationships between Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.

- Clustering – is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.

- Classification – is the task of generalizing known structure to apply to new data. For example, an e-mail program might attempt to classify an e-mail as "legitimate" or as "spam".

- Regression – attempts to find a function which models the data with the least error.

- Summarization – providing a more compact representation of the data set, including visualization and report generation.

- Data mining can be helpful to human resources (HR) departments in identifying the characteristics of their most successful employees. Information obtained – such as universities attended by highly successful employees – can help HR focus recruiting efforts accordingly. Additionally, Strategic Enterprise Management applications help a company translate corporate-level goals, such as profit and margin share targets, into operational decisions, such as production plans and workforce levels.

- Market basket analysis, relates to data-mining use in retail sales. If a clothing store records the purchases of customers, a data mining system could identify those customers who favor silk shirts over cotton ones. Although some explanations of relationships may be difficult, taking advantage of it is easier. The example deals with association rules within transaction-based data. Not all data are transaction based and logical, or inexact rules may also be present within a database.

- Market basket analysis has been used to identify the purchase patterns of the Alpha Consumer. Analyzing the data collected on this type of user has allowed companies to predict future buying trends and forecast supply demands.

- Data mining is a highly effective tool in the catalog marketing industry.Catalogers have a rich database of history of their customer transactions for millions of customers dating back a number of years. Data mining tools can identify patterns among customers and help identify the most likely customers to respond to upcoming mailing campaigns.

- Data mining for business applications can be integrated into a complex modeling and decision making process. Reactive (RBI) advocates a "holistic" approach that integrates data mining, modeling, and interactive visualization into an end-to-end discovery and continuous innovation process powered by human and automated learning.

- In the area of decision making, the RBI approach has been used to mine knowledge that is progressively acquired from the decision maker, and then self-tune the decision method accordingly.The relation between the quality of a data mining system and the amount of investment that the decision maker is willing to make was formalized by providing an economic perspective on the value of “extracted knowledge” in terms of its payoff to the organization This decision-theoretic classification framework was applied to a real-world semiconductor wafer manufacturing line, where decision rules for effectively monitoring and controlling the semiconductor wafer fabrication line were developed.

**Sensor data mining**

Wireless sensor networks can be used for facilitating the collection of data for spatial data mining for a variety of applications such as air pollution monitoring. A characteristic of such networks is that nearby sensor nodes monitoring an environmental feature typically register similar values. This kind of data redundancy due to the spatial correlation between sensor observations inspires the techniques for in-network data aggregation and mining. By measuring the spatial correlation between data sampled by different sensors, a wide class of specialized algorithms can be developed to develop more efficient spatial data mining algorithms.

**Visual data mining**

In the process of turning from analogical into digital, large data sets have been generated, collected, and stored discovering statistical patterns, trends and information which is hidden in data, in order to build predictive patterns. Studies suggest visual data mining is faster and much more intuitive than is traditional data mining. See also Computer vision.

**Music data mining**

Data mining techniques, and in particular co-occurrence analysis, has been used to discover relevant similarities among music corpora (radio lists, CD databases) for purposes including classifying music into genres in a more objective manner.

**Surveillance**

Data mining has been used by the U.S. government. Programs include the Total Information Awareness (TIA) program, Secure Flight (formerly known as Computer-Assisted Passenger Prescreening System (CAPPS II)), Analysis, Dissemination, Visualization, Insight, Semantic Enhancement (ADVISE), and the Multi-state Anti-Terrorism Information Exchange (MATRIX). These programs have been discontinued due to controversy over whether they violate the 4th Amendment to the United States Constitution, although many programs that were formed under them continue to be funded by different organizations or under different names

In the context of combating terrorism, two particularly plausible methods of data mining are "pattern mining" and "subject-based data mining".

**Pattern mining**

"Pattern mining" is a data mining method that involves finding existing patterns in data. In this context *patterns* often means association rules. The original motivation for searching association rules came from the desire to analyze supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. For example, an association rule "beer ⇒ potato chips (80%)" states that four out of five customers that bought beer also bought potato chips.

In the context of pattern mining as a tool to identify terrorist activity, the National Research Council provides the following definition: "Pattern-based data mining looks for patterns (including anomalous data patterns) that might be associated with terrorist activity — these patterns might be regarded as small signals in a large ocean of noise. Pattern Mining includes new areas such a Music Information Retrieval (MIR) where patterns seen both in the temporal and non temporal domains are imported to classical knowledge discovery search methods.

**Subject-based data mining**

"Subject-based data mining" is a data mining method involving the search for associations between individuals in data. In the context of combating terrorism, the National Research Council provides the following definition: "Subject-based data mining uses an initiating individual or other datum

that is considered, based on other information, to be of high interest, and the goal is to determine what other persons or financial transactions or movements, etc., are related to that initiating datum."

## 7. Explain the concepts of Database access Control

Each of the core OpenStack services (Compute, Identity, Networking, Block Storage) store state and configuration information in databases. In this chapter, we discuss how databases are used currently in OpenStack. We also explore security concerns, and the security ramifications of database backend choices.

### OpenStack database access model

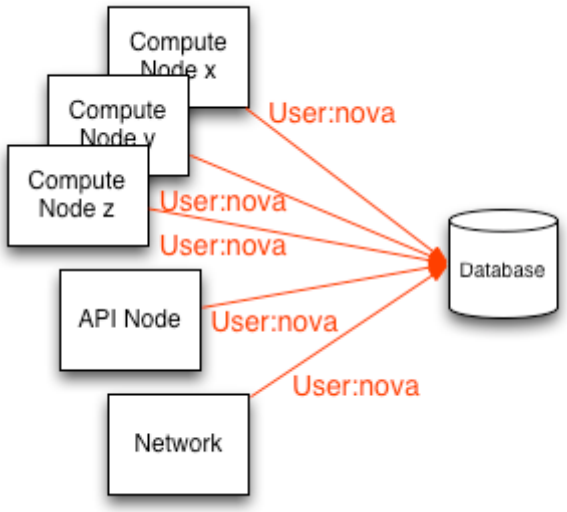
### Nova-conductor

All of the services within an OpenStack project access a single database. There are presently no reference policies for creating table or row based access restrictions to the database.

There are no general provisions for granular control of database operations in OpenStack. Access and privileges are granted simply based on whether a node has access to the database or not. In this scenario, nodes with access to the database may have full privileges to DROP, INSERT, or UPDATE functions.

### Granular access control

By default, each of the OpenStack services and their processes access the database using a shared set of credentials. This makes auditing database operations and revoking access privileges from a service and its processes to the database particularly difficult.

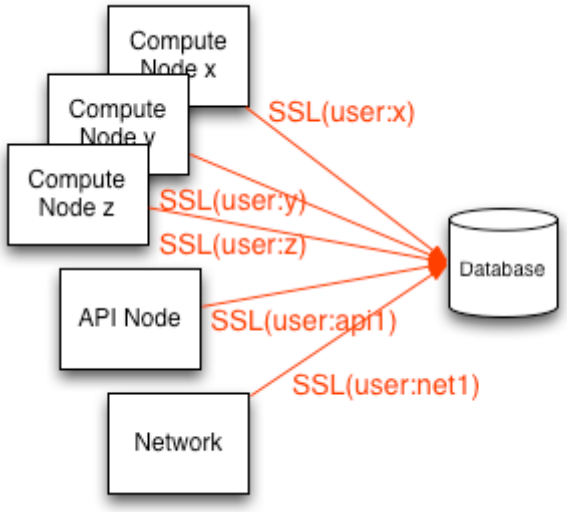



### Nova-conductor

The compute nodes are the least trusted of the services in OpenStack because they host tenant instances. The `nova-conductor` service has been introduced to serve as a database proxy, acting as an intermediary between the compute nodes and the database. We discuss its ramifications later in this chapter.

We strongly recommend:

- All database communications be isolated to a management network
- Securing communications using SSL
- Creating unique database user accounts per OpenStack service endpoint (illustrated below)




**8. Explain the concepts of Cryptography in database**

**Cryptography**

Cryptography is the art of "extreme information security." It is extreme in the sense that once treated with a cryptographic algorithm, a message (or a database field) is expected to remain secure even if the adversary has full access to the treated message. The adversary may even know which algorithm was used. If the cryptography is good, the message will remain secure.

**Symmetric Cryptography**

Symmetric key cryptography is so named because the cipher uses the same key for both encryption and decryption. Two famous ciphers, Data Encryption Standard (DES) and Advanced Encryption Standard (AES), both use symmetric keys.

**Public-Key Cryptography**

Public-key cryptography, also known as asymmetric cryptography, is a relatively recent invention. As you might guess from the name, the decryption key is different from the encryption key.

**Cryptographic Hashing**

A cryptographic hash, also known as a message digest, is like the fingerprint of some data. A cryptographic hash algorithm reduces even very large data to a small unique value. The interesting thing that separates cryptographic hashes from other hashes is that it is virtually impossible to either compute the original data from the hash value or to find other data that hashes to the same value.

**9. Explain the concepts of Crawling and Indexing the Web**

**Web indexing** (or **Internet indexing**) refers to various methods for indexing the contents of a website or of the Internet as a whole. Individual websites or intranets may use a back-of-the-book index, while

search engines usually use keywords and metadata to provide a more useful vocabulary for Internet or onsite searching. With the increase in the number of periodicals that have articles online, web indexing is also becoming important for periodical websites.

Back-of-the-book-style web indexes may be called "web site A-Z indexes". The implication with "A-Z" is that there is an alphabetical browse view or interface. This interface differs from that of a browse through layers of hierarchical categories (also known as a taxonomy) which are not necessarily alphabetical, but are also found on some web sites. Although an A-Z index could be used to index multiple sites, rather than the multiple pages of a single site, this is unusual.

Metadata web indexing involves assigning keywords or phrases to web pages or web sites within a meta-tag field, so that the web page or web site can be retrieved with a search engine that is customized to search the keywords field. This may or may not involve using keywords restricted to a controlled vocabulary list. This method is commonly used by search engine indexing.

**10. Explain Association rules and its types with an example**

**Association rule learning** is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness. Introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets.

For example, the rule $\{\text{onions}, \text{potatoes}\} \Rightarrow \{\text{burger}\}$ found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection, Continuous production, and bioinformatics. As opposed to sequence mining, association rule learning typically does not consider the order of items either within a transaction or across transactions.