PIE Tech
POLLACHI INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Approved by AICTE and Affiliated to Anna University)
sky is the limit

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Regulation 2021

II Year- III Semester

AD3391 DATABASE DESIGN AND MANAGEMENT

PIE Tech

POLLACHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE and Affiliated to Anna University)

sky is the limit

AD3391 DATABASE DESIGN AND MANAGEMENT

AD3391 DATABASE DESIGN AND MANAGEMENT

UNIT I CONCEPTUAL DATA MODELING 8

Database environment – Database system development lifecycle – Requirements collection – Database design -- Entity-Relationship model – Enhanced-ER model – UML class diagrams.

UNIT II RELATIONAL MODEL AND SQL

Relational model concepts -- Integrity constraints -- SQL Data manipulation – SQL Data definition – Views -- SQL programming.

UNIT III RELATIONAL DATABASE DESIGN AND NORMALIZATION

ER and EER-to-Relational mapping – Update anomalies – Functional dependencies – Inference rules – Minimal cover – Properties of relational decomposition – Normalization (upto BCNF).

UNIT IV TRANSACTION MANAGEMENT

Transaction concepts – properties – Schedules – Serializability – Concurrency Control – Two-phase locking techniques.

UNIT V OBJECT RELATIONAL AND NO-SQL DATABASES

Mapping EER to ODB schema – Object identifier – reference types – rowtypes – UDTs – Subtypes and supertypes – user-defined routines – Collection types – Object Query Language; No-SQL: CAP theorem – Document-based: MongoDB data model and CRUD operations; Column-based: Hbase data model and CRUD operations.

Unit I

Relational Database

Database

⇒ Database is a collection of data in a systematic way so that data can be accessed, managed and updated in a effective manner.

⇒ DBMS is a software, helps the user to interact with the data to analyze and use the data for application.

⇒ DB is developed by to manage the computerized data stored in a software

⇒ In earlier days, DB helps to manage computerized data for a commercial application

⇒ It plays a important role in banking, business, education and so on.

⇒ Handling of vast data and managing the business needs or data in the main purpose of DBS

⇒ Security in data handling is a key role of DBMS

Advantages of DBMS

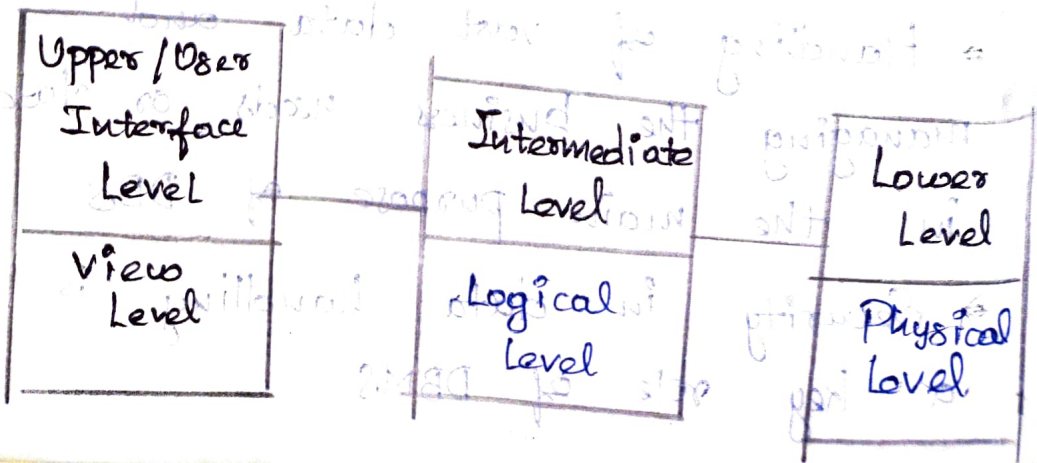
*. Retenary control *. Efficient Query Processing
*. Security from unauthorized access
*. Persistent storage *. Reduced Time
*. Backup and recovery *. Multiple User Interface
*. Updated Information *. Flexibility

Different Uses of Data

DBS gives a abstract view about data. Abstract view shows the external view of data in the uses of their utility Purpose and hide the internal information Such as how data formed, maintained and so on.

Data Abstraction

Data Abstraction defined as hiding internal information about data to the user. Internal information includes how data are stored, managed and so on aligned)



Based on hiding internal information, developer explain abstraction at three levels

- ⇒ Physical Level
- ⇒ Logical level
- ⇒ View level

Physical Level

It is the lower level of data abstraction describes about data structure and explain how data can be stored in DB.

Logical Level

⇒ It is the middle level of the data abstraction, it describes what data to be shown. The Administrator decides what information on DB

⇒ This level describes the relation among data structure and data used in middle level

View Level

* This is the user interface where describes only the part of entire database
* This level helps the user to access the data from the database

Eg

```
type employee record
    employee_id : int
    employee_name : String
    employee_salary : Double
end;
```

Instances

* Instance is defined as a single occurence or record of a particular entity in a database.
* The collection of information stored in a database at a moment is called instance
* The data in the database which is being entered at a particulas time is called State of a DB

Scheme

The overall design of the entire description is called database scheme.

Scheme diagram:

Employee

id	name	designation

Department

Code	exp	Salary

Grade

Senior	Assistant	HR

The concept of database scheme corresponds to the variable declared in the program like each variable has a particular value at a given instance, object like employee, department, grade are called as scheme construction.

Schema classification

*. Physical schema

*. Logical schema

*. Subschema

Physical Schema

Database designed at physical level

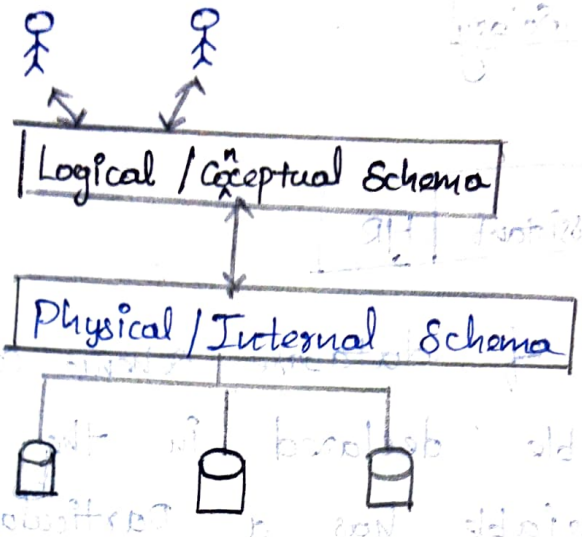
Logical Schema

Database designed at logical level

Three Schema Architecture

Universal database MS from IBM uses relational model to describe a conceptual schema gives a object oriented model to

describe the external schema



Data Independency in 3 Schema Architecture

Data Independency is defined as ability to change the schema gives a object oriented model to describe the external schema

Types

1. Logical data independency
2. Physical data independency

1] Ability to change the conceptual schema without changing on external schema. Conceptual schema can be changed the constraint

2] Ability to change the internal schema without having change the conceptual schema

Data Model

*. Data model is a collection of concept

describe the structure of database to understand the means of abstraction

* Datamodel describe a design of a database at different level of abstraction such as Physical, logical and view level

* Data constraints, schematic that holds the data it also includes operations to specify data updates and retrieval.

Eg:

It compute the employee salary, user-defined operation and database object, and generic operation perform like insert, delete, modify are basic model operations.

Four categories of data models are.

* Relational
* Entity relationship
* Object based
* Semi structured

Relational Model

Relation is a table of values of records. It means each row in a table have related to data value. The relational model has collection of table that represent the relationship among the data in a table.

Each table has row and column. Each Column has unique name. It is also known as record based model.

S.No	Name	Marks
1	Abi	9.9
2	Asrap	95

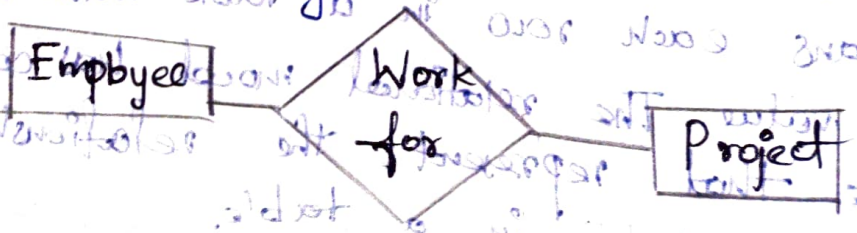
* It is called record based since each tab describes the info about particular record.
* It is the basic model comprises of no. of fields and attributes.
* The attributes refer to the column of table and tuple refers to the row of the table

Entity Relationship

It is popular high level conceptual data model. It is based on the real world base of object is called entities.

Attributes and relationship

⇒ A entity is a real world entity. Eg, Person
⇒ Array Attributes represent describe entity
⇒ A relationship is the association among two or more entity



Object based

* It is an extending model with encapsulation, methods and object entity
* It combines features of both object Oriented and relational data model.

Database System Architecture

It is the component of database System and connection among them

Database can be centralized (or) Client - Server which can be worked on the Parallel computer Architecture

There are 3 level of Architecture

1] Internal Level
2] External Level
3] Conceptual Level

Internal Level

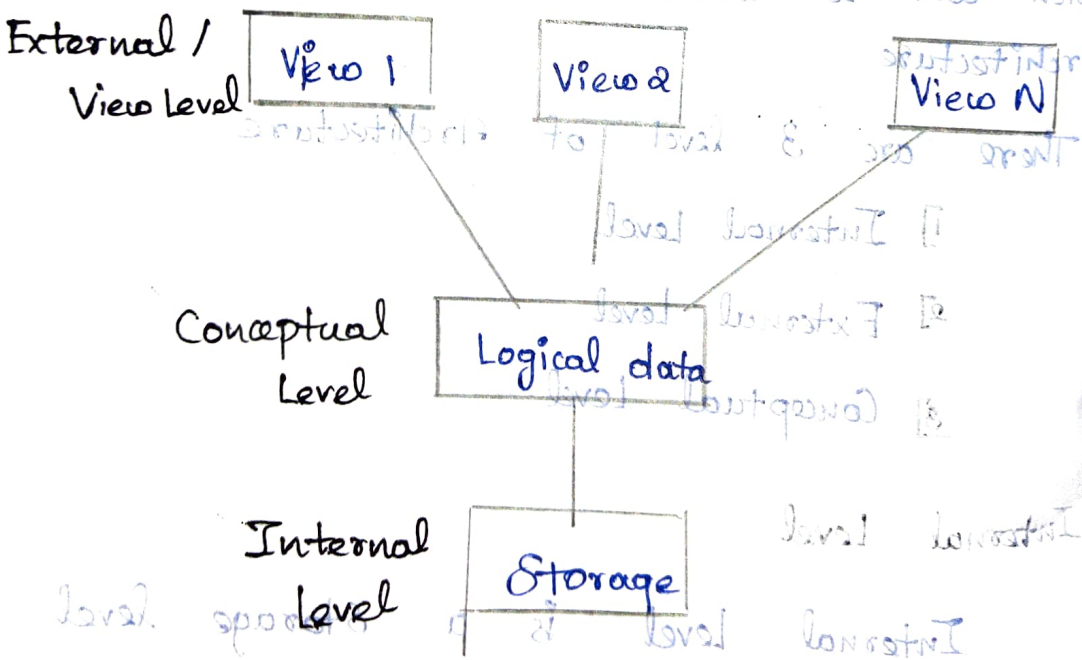
Internal Level is a storage level and Consider on what way the data Stored inside the System

External Level

External Level is a user level consider about how data being fetched and used by the user

Conceptual Level

Conceptual Level is a logical level which as the middleware between the Internal Level and External Level. It Performs all logical level of computation to maintain the data at the database



Eg. of 3 Level

External - Struct student {

int roll no;

char name[10];

float marks;

}

Conceptual Level

Student

Student_number	int [5]
name	character [15]
mark	Decimal [5]

Internal Level

Student

number # Bytes = 6 offset = 6, Index num

name # Bytes = 8 offset = 4

marks # Bytes = 6 offset = 4

⇒ External view consists of many occurance of external records and it is defined as external scheme.

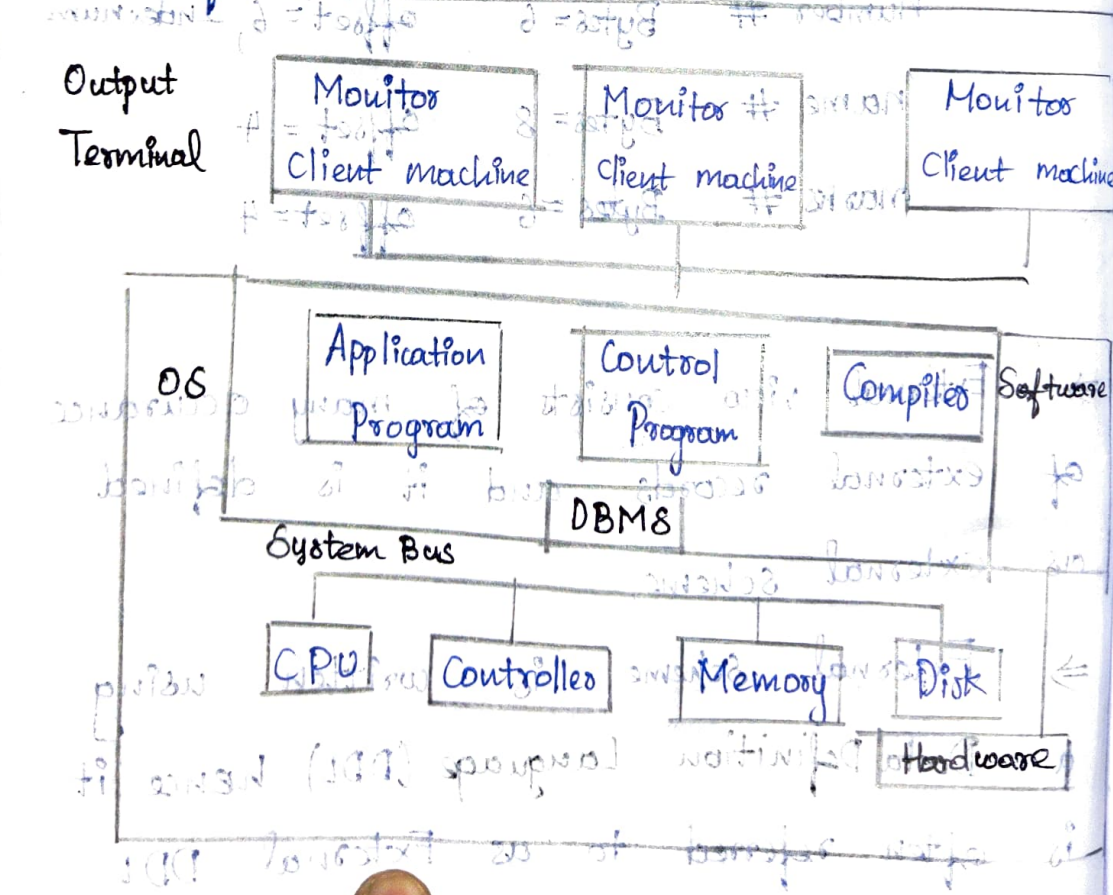
⇒ External scheme is written using a Data Definition Language (DDL) hence it is often referred to as External DDL

⇒ Conceptual View is a overall representation of an entire information content in the database, It is defined as conceptual Schema

⇒ It also include additional features such as security and integrity constraints.

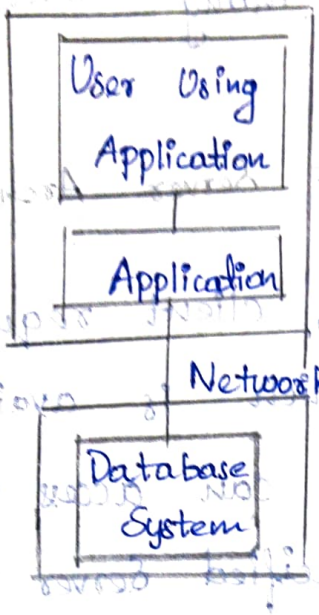
⇒ Internal View is a low level representation of a database contain stored record of the database. It is described as Internal Schema

⇒ Internal Scheme is written using DDL is referred to as Internal DDL



Generalized DBMS Architecture

Two Tier DBMS Architecture



⇒The applications is partitioned into components that decides as a client machine. The Interfaces Like ODBC and JDBC used for interaction between client and Server

⇒Centralized DBMS perform application Program execution, user interface, control of DBMS in a single machine, user access the database centralized Single mission.

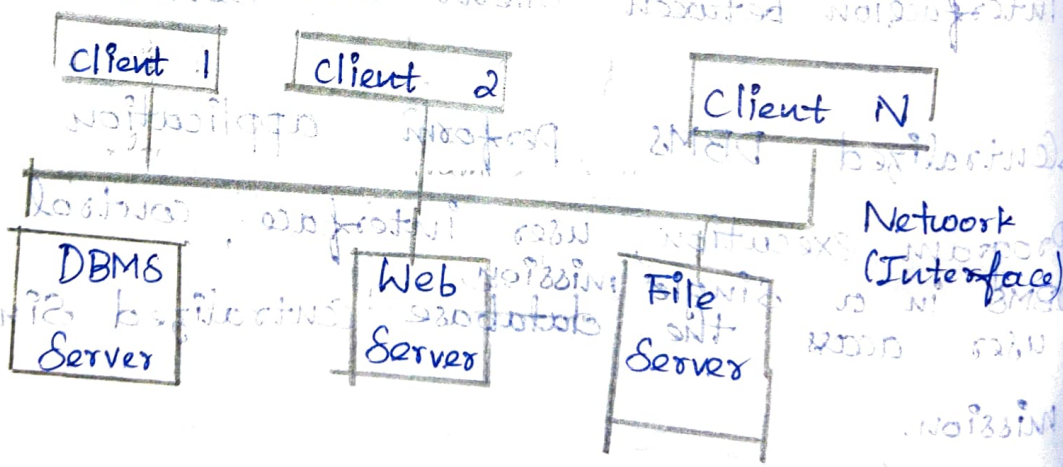
⇒But in a client Server architecture computing enviroument like Work station, file Server, Database Server, Web Server connected to the network

⇒ resources provided by the specific Server can be accessed by many client machine

Logical Two Tier Client Server Architecture

⇒ In Client Distribution, client require additional facility which is available in another server which can access the service by connecting to specified server to the network

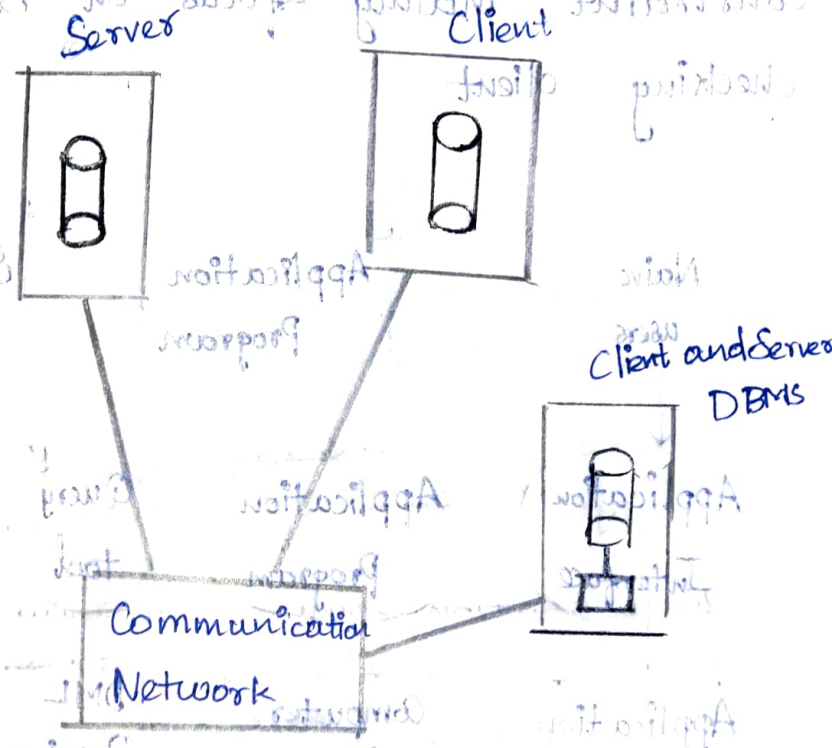
⇒ Server also a machine which holds the software and Hardware support provide service to the client machine



Physical Two Tier client Server Architecture

The machine would be dedicated server which is connected through the network

⇒ RDMS, most of the machines started as centralized system. In a client server architecture the application program executed in client side when the client need to access the database which is on the server side. Then a client start communicating through a program called open database connectivity Provideded by application program interface



Advantages of Two Tier Architecture

1] Simplicity

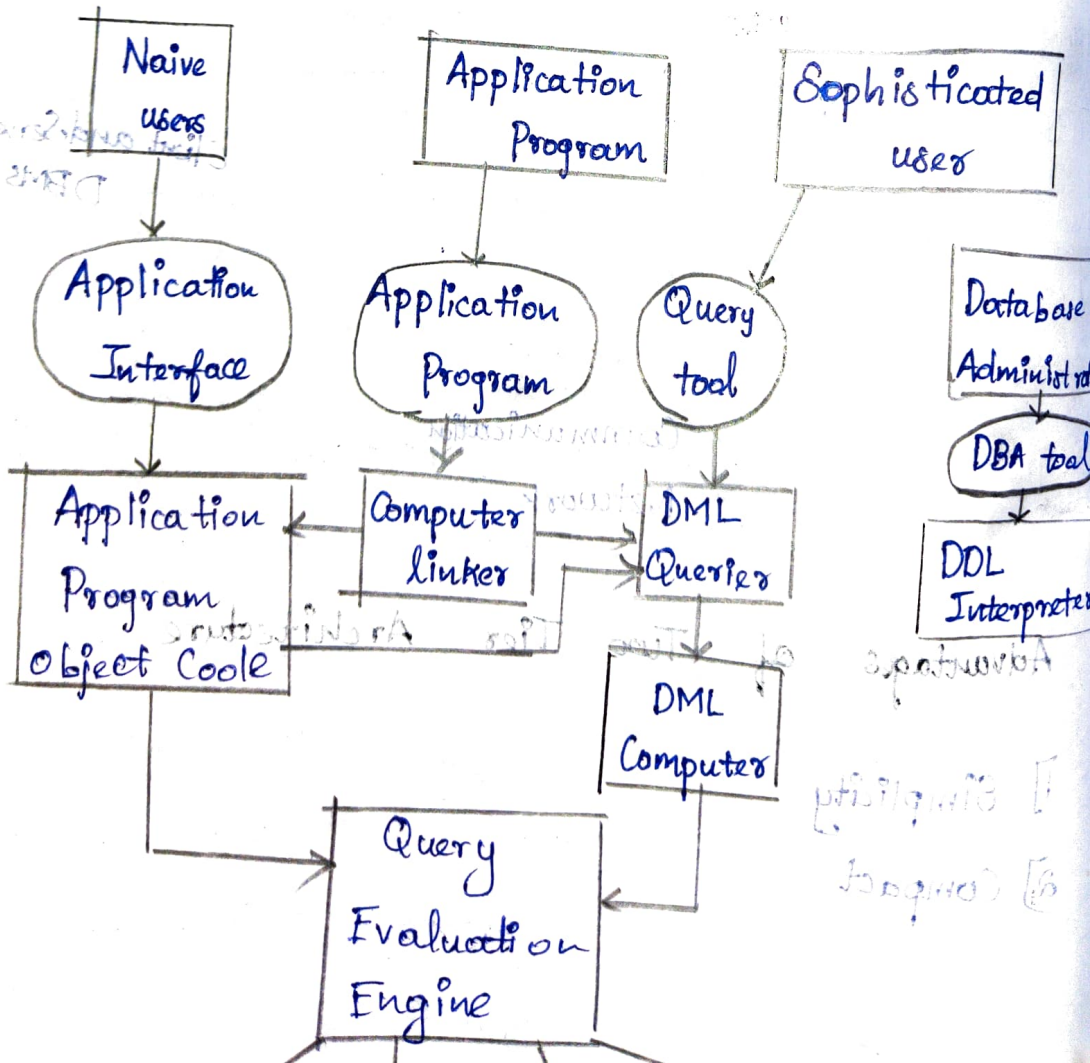
2] Compact

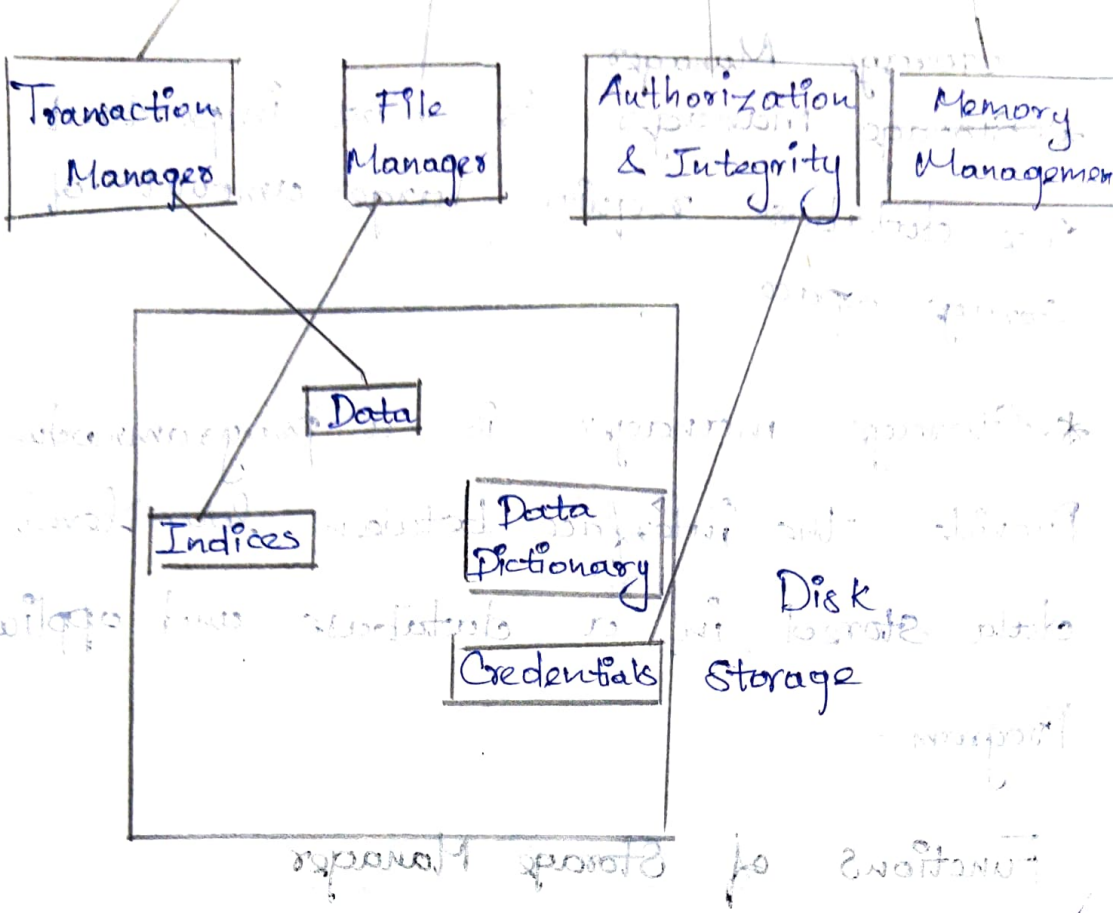
Three Tier Architecture

Emergency of web changed the role of client and server that needed the development of Three Tier Architecture

⇒ Middle Tier is Application / Web Server

⇒ Intermediate Server plays an intermed role by storing business procedure and Constraint. mainly focus on Security by checking client





*. Different query processing component Plays a vital role to make the database function.

*. Query processor fetch the require from the disk storage.

*. This system structure focus on how to store a data, ensure atomicity, perform concurrency control, distribute the Query Processing and so on.

*. The functional component of the database is divided into Storage manager and Query Processor.

Storage Manager

*. Storage manager is most important since database require large amount of storage space.

*. Storage manager is a programmodu Provide the interface between low level data stored in a database and applic Program.

Functions of Storage Manager

*. Interact with File Manager

*. Responsible for storing, Retrieving and updating data in the database

Components of Storage Manager

* Authorization and Integrity manager

*. Transaction Manager.

*. File Manager

*. Buffer Manager.

Query Processor

* Query processor perform DDL and DML commands.
* DDL Interpreter comprises of component
* DML compiler to translate DML command in a query language to query evaluation engine.
* Query evaluation engine execute low level instruction.

Relational Database

* Relational database is a collection of data interms of row and column, assigned unique name. row is called Tupk, column is called Attribute and the table is called a relation
* A datatype format is also specified for each domain.

Atomic value

Each value in the domain is indivisible as far as the relational model is concerned

Relational Schema (R)

* Relational Schema (R) denoted by $R(A_1, A_2, \ldots A_n)$ is made up of relation name R with list of Attributes $A_1, A_2, \ldots A_n$.

* Each Attribute role is play a some domain.

Characteristics of relation

* Ordering of Tuple in a relation
* Ordering of values with the tuple, values and NULL in the tuple.
* A Tuple is a set of pair i.e

(<attribute>, <values>)

Keys

* Key is used to identify any record or tuple or attribute in a relation
* Key can be a Single Attribute or Group of attribute.
* Key hold a unique value.

*. Keys are used to fetch a particular data from the huge volume of relation.

Types of key.

1] Super key
2] Primary key
3] Composite key
4] Candidate key
5] Secondary key
6] Non key Attribute
7] Non primary Attribute

Super key

Super key is a super set of Candidate key

Candidate key

It refers to minimal set of field which uniquely identify the record in a relation

Primary key

It is main key of any table that can uniquely identify a record in a table

Composite key

It has more than one attribute grou to uniquely identify a record in a relation table is called Composite key

Foreign key

It is a field in a table that identifies a row in a another table

Secondary key

The candidate key which is not selected as a primary key is called Secondary key

Relational Algebra

Select, Project, union, difference, cartesian product, rename, set insertion, nactural join, division assignm

Emp Id	Emp_name	emp_dept	emp_phoneno
1.	AA	IT	1234
2.	BB	CSE	5678
3.	CC	EEE	9101
4.	DD	ECE	1234

Unary Operation

σ <Selection Condition> R

σ <dept = IT>

Output: AA
1234
IT

Projection Operation

emp-ID	emp_address
1	XX
2	YY
3	ZZ
4	&8

Π <attribute> R_1

Rename Operation

$C_S (B_1, B_2, \ldots B_n) (R)$

Binary relation Operation

1] Natural Join operation
2] Division operation
3] Assignment operation

Natural Join Operations

Natural Join exports a common column along different relation whereas it is not applicable in case of Cartesian Product. Join Symbal $\bowtie$

Inner Join

Theta Join, Equi join, Natural Join are called Inner join.

Outer Join

Left Outer Join, Right Outer Join and Full outer join

Course	
A	B
100	DB
101	OS
102	DM

⋈

Head	
100	aaa
101	bbb
103	ccc

Left Outer Join

Course		⟕ Head	
100	DB	100	aaa
101	OS	NULL	NULL
102	DM	102	bbb

Right outer Join

Course		⟖ Head	
100	DB	100	aaa
102	DM	102	bbb
NULL	NULL	103	ccc

Full Outer Join

Courses ⋈ Head			
A	B	C	D
100	DB	100	aaa
101	OS	NULL	NULL
102	DM	102	bbb
NULL	NULL	103	ccc

SQL Fundamentals

* Structure Query Language (SQL) is one of the fundamental language in database architecture.

* It is domain specific language used to manage the database.

* SQL has several parts

1] Data Definition language

2] Data Manipulation Language

3] Data Control Language

4] Embedded SQL

5] Dynamic SQL

6] Transaction Control

7] View definition

8] Authorisation

1. Data Definition Language

Create, alter, drop, truncate, desc etc.

Eg,

1. create table employee (emp_no number(5), empname varchar(30));
2. insert into employee values (1, aa);
3. alter table employee add (dept varchar(10));
4. drop table employee;
5. delete from employee where emp no: 2;
6. desc employee;

Output:

Name	Type
emp_no	number[5]
emp_name	varchar[20]

truncate is used to delete the entire

2. Data Manipulation Language

insert, update, rename

1. insert into employee values(1, 'IT', chennai);

2. update employee set exp=2 where emp_no=2;

3. rename employee to emp;

3. Data Control Language

grant and revoke

Eg,

SQL > Grant all on employee to individual

SQL > Revoke all on employee from individual

4. Transaction Control Language

* 'Commit' it says the task completed
* 'Savepoint' help to go back to previous transaction
* 'rollback' restore the database upto last committed place.
* 'Set Transaction' changes the transaction point

Implementation of Built in functions in SQL

1. upper - changes the character into uppercase
2. Lower - changes the character into lowercase
3. initcap 4. Length 5. Rpad 6. Lpad

7. translate 8. replace 9. ltrim

9. rtrim 10. 'datatype' function displ

the datatype.

11. "view definition" function

'virtual table' obtained based on Query

1] Updatable view

2] Readonly view

Eg,

1. Create view customer as (select branch_na

customer_name, from 'loan account

where loan account_no = account. account_no

Advanced SQL

User defined is two forms namely

1. distanct type

2. Structure datatype

Large Object datatype

lob, clot - character data, blot - binar

data

Catalog

Schema in turn contain data object and relationship among the data.

Integrity Constrain

* It is important to ensure the originality of data in the database

Eg, Every student in a student relation must be same as a student name in a formative test relation.

Constraining table creation

Normal Unique check. NULL is defined as domain for every attribute

Roll No numeric(10) not null;

Unique

Unique values of attribute form a candidate key. Primary key always have a unique constrain

Eg, create tabb student (rollno int not null unique, name char[33] not null)

* Primary key constrain is used to identify each record in a relation

Eg, Create table student (rollno int no
unique, name char[33], Primary key (roll no))

* Primary key of one relation is a foreign key of Another relation.

⇒ Check constrain is used to ensure attribute value are in a specific condi

Eg, Create table employee (emp_no int not n
empname char2[50], Salary_number [20],
check (Salary > 0));

Set default

⇒ Set Default constrain is used to insert a default value in the attribute

Eg, Create table employee (emp_no int not nul
emp_name char2[50], Salary_number [20], default 2000

Referential Integrity

Condition of set of tuples in one relation appear in certain set of tuples in another relation which is termed as Referential Integrity.

Eg

1. create table dept (dep_id int notnull, dept_name char[255], Primarykey (dept_id));
2. Create table employee (emp_iol int notnull, emp_name char[255], dept_id int foreign key (dept_id) reference department (dept_id) on delete cascade);

⇒ Assertion

Assertion is a statement in SQL that always expect to be exist in a database

Eg, create assertion balance check (select * from customer where balance < 1000);

Authorization

Authorization to read, insert, create, update, delete data In SQL DDL include grant and revoke privileges.

Embedded SQL

* It is way of combining Programming language and SQL to support an application
* It is a method of inserting inline SQL Queries into the programming language

*. Embedded SQL is supported by o
database and not supported by MySQ

Embedded SQL compilation is done
two step process

1] SQL statement pre compiled
check for the correctness and execution p

2] After precompilation the executab
code embedded into C, then C compiler
compile and execute

Syntax:

```
EXEC SQL CONNECT database_name
EXEC SQL CONNECT user_name
```

Host Variable

Host variable capture the value
Passed by the query.

```
EXEC SQL BEGIN DECLARE SECTION
    int empno;
    char emp_name;
    char dept[10];
```

EXEC SQL, END DECLARE SECTION

Indicator Variable

This variable are two byte short type capture a NULL value

Two types of Embedded SQL

1. Static Embedded SQL
2. Dynamic Embedded SQL

Static Embedded SQL

* Entire SQL statement is executed during the precompilation process. take
* Host variable are not allowed to the table name or attribute

Dynamic Embedded SQL

Entire SQL statement is made to execute at run time using a Host Variable

Dynamic SQL

Dynamic SQL is a SQL printf during the run time.

Important features of SQL Server

⇒ In static SQL it will compile at Compile time and execute at run t

⇒ In Dynamic SQL both Compilation execution is done at the run time.

⇒ In a Dynamic SQL till runtime we dont know the file_name, rows and Columns. etc.

Advantages of Dynamic SQL

* Rescue code for different dynamically builten table.
* Allow parameters filtering within clau
* Sorting by a column using conditional Statement.

Disadvantages

* Performance Loss
* Temporary table cannot be accessed and it is declared as global
* Error Management is

* Maintenance is difficult
* Complex query tend to increase the execution time
* SQL injection attacker Possible

DB Embedded SQL and Dynamic SQL

Embedded SQL	Dynamic SQL
The SQL statement could not change at run time	The SQL statement are constructed during run time
Since this statements are static it can hard coded into an application	This statement built in during the run time
SQL database is pre determined to the SQL Query	Database is determined during the run time
It is not flexible to the General purpose SQL Queries	Flexible to the General Purpose SQL Queries
SQL statements are compiled during compile time	SQL statements are compiled during run time
It is less efficient since it is possible to access for the predetermined database	It is more efficient Since it is used for general application
Here data is uniformly distributed	Here data is Non Uniformly distributed

It make use of Execute immediate. Execute and prepare. Statements are not availabe	It make use of Execute immediate Execute and prepare Statement

Unit - 2

Database Design

* Database Design is a logical structure of a database. It is also called as Database model

E-R Model (Entity - Relation)

* Entity is a real world object distinguished from the other object.
* A Entity is described using a set of attributes.
* Entity Set is a collection of similar entities
  Ex, All employees
* A Entity is a Entity Set having same set of attributes
* Each entity has a key
* Each attribute has a domain

Relationship

* A Relationship is the association among the several entities

* A relationship set is a mathematical relation among $n \geq 2$

* Attribute have a value of specific domain or type of column

* Using a key attribute other details of relationship can be identified.

Types of Attribute

1] Simple and composite attribute

* Simple attribute cannot be divided into similar components

Ex. Age

* Composite attribute is the combination of different attribute. It can be divided into similar components

Ex, ~~Courses joined by Student~~, Date of birth

2] Single and Multivalue Attribute

⇒ Each Entity takes a single value for an instance.

Ex. Age

In Multivalue Attribute, entity take many attribute.

Ex, course joined by Student

3] Stored and Derived attribute

* Stored Attributes are Attributes which are needed to be stored permanently in the database

Ex. Name of the Student

* Derive Attribute are derived from other attribute. This attribute is obtained from manipulation of corresponding attribute

Ex. Age from a database

Descriptive Attribute

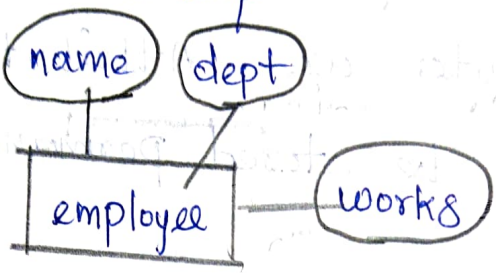
* Attribute comes with a relationship is termed as the descriptive attribute

Five degrees of relationship

* Unary
* Binary
* Ternary
* Quadnary
* Nuary

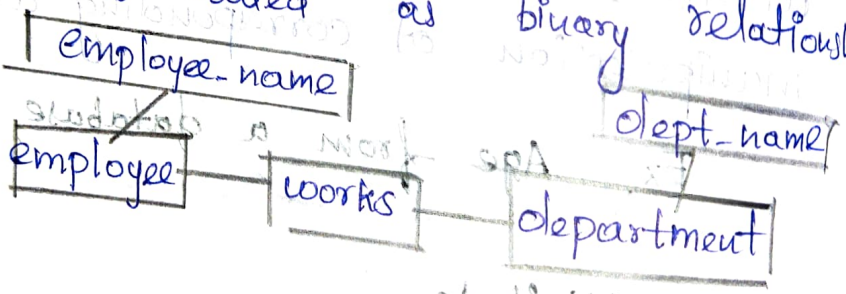
Unary relationship

* When a association of an attribute exist with a single entity is called as unary relationship



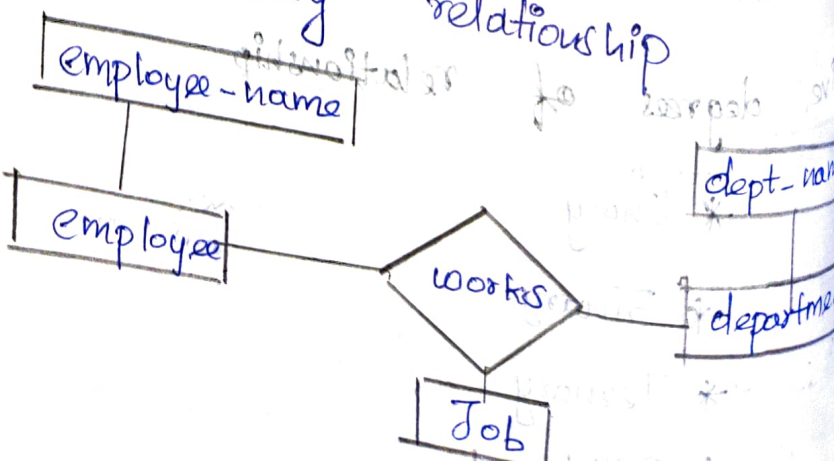
Binary relationship

When a two entity exist in a association which is called as binary relationship



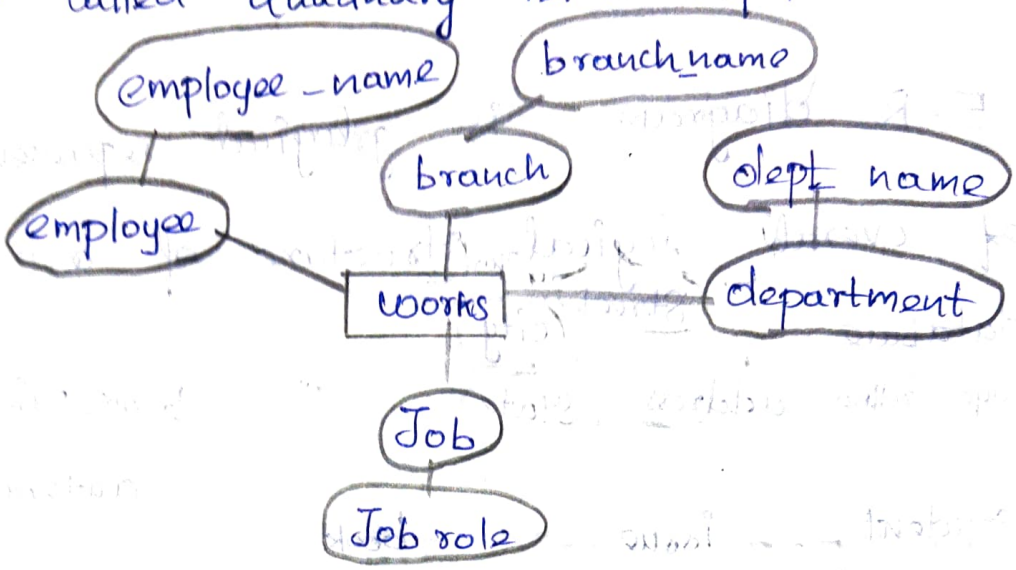
Ternary relationship

When a three entity takes place it called as Ternary relationship



Quadnary relationship

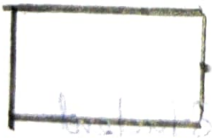
When a four entity takes place then it is called Quadnary relationship.



Nnary relationship

It is more than four entity exist then it is called N nary relationship

Component of E-R diagram


→ entity

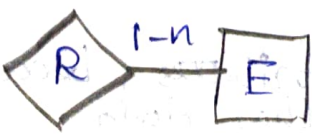
→ Specialization/ Generalization


→ relationship

→ links


→ Multivalued attributes


→ Attributes



→ Cardinality limits


→ derived attributes

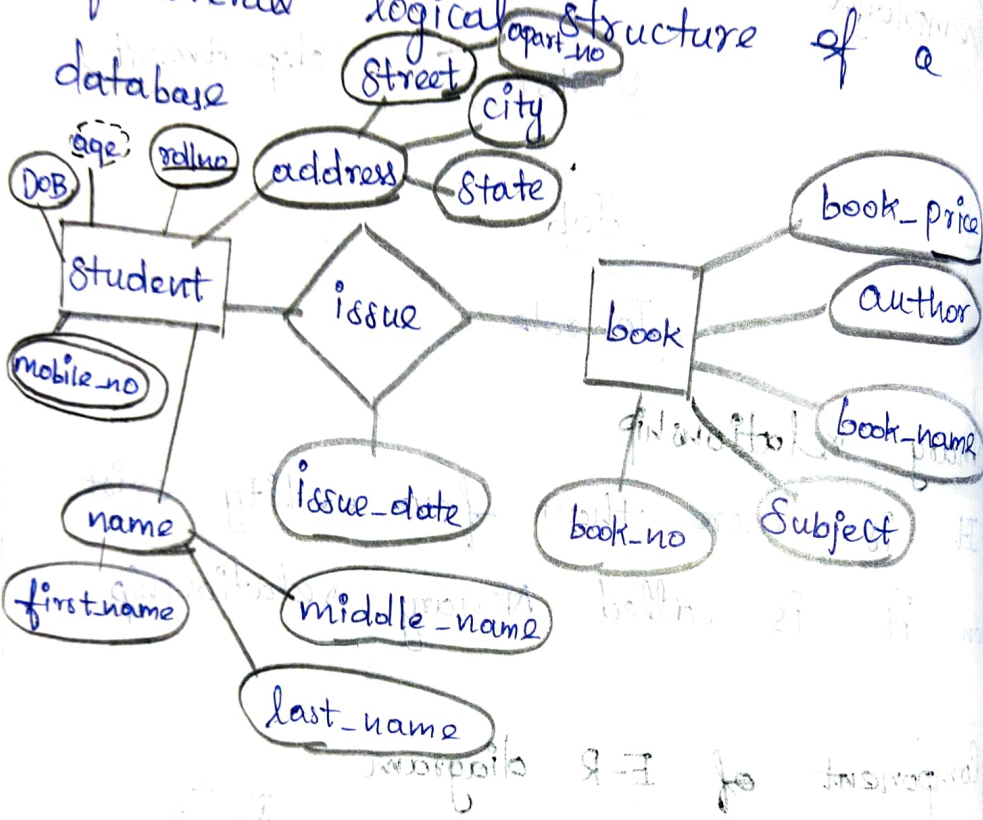
= → Total Participation

→ Weak e

Sex

E-R diagram is graphical repres

of overall logical structure of a database



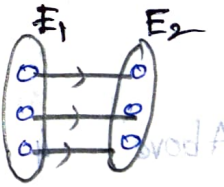
* Student and book are entity
* issue is a relationship between Student and book
* attributes in a entity are denoted by oval shape
* mobile number is a multivalue attribute one student can have more than one mobile number
* age is a derived attribute because it is derived from other data

* rollno is a primary attribute
* name is a composite attribute which is decomposed into multiple attributes such as first, last and middle name
* address and street are composite attribute
* issue is a relationship which as descriptive attribute

Cardinalities Mapping

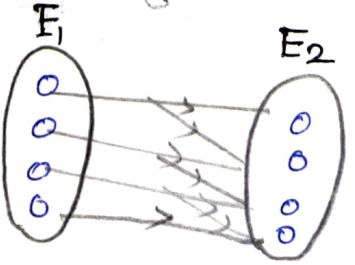
i] One to one relation

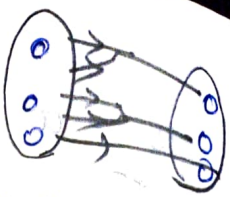
Here an entity e1 is associated with atmost 1 entity in e2



ii] One to many cardinality

An entity e1 is associated with one of more than 1 entity in e2 but e2 must associated with 1 entity in e1



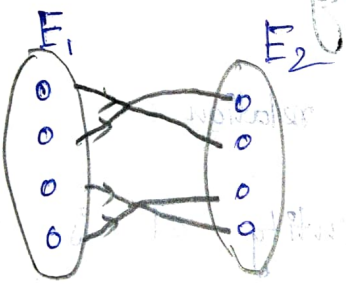


iii] **Many to One**

An entity E1 is associated with atmost 1 entity in E2. E2 must be associated with any no. of entity in

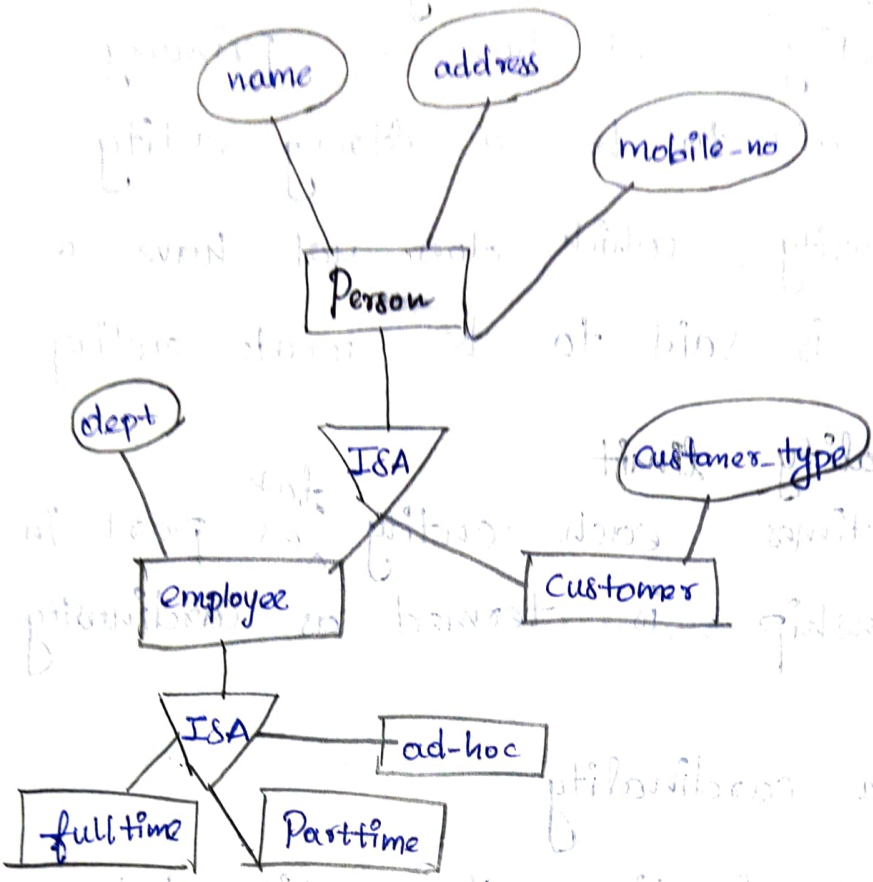
iv] **Many to Many**

An entity E1 is associated with 0 or more no. of entities in E2 and entity E2 must be associated with or more no. of entities in E2



Above 4 mapping constrain can be classified as total Participation constrain Partial Participation constrain.

Specialization and Generalization of E-R Diagram



* Person entity is specialized as employee and customer entity

* employee is further specialized as full time, Part time and ad-hoc.

Binary Relationship

Here two entities are connected to a relationship then it is called Binary relationship

Strong and Weak Entity

* An entity which have a primary key is said to be a strong entity
* An entity which does not have a Primary is said to be weak entity

Cardinality limit

No. of times each entity takes part in a relationship is termed as Cardinality limits.

Minimum cardinality

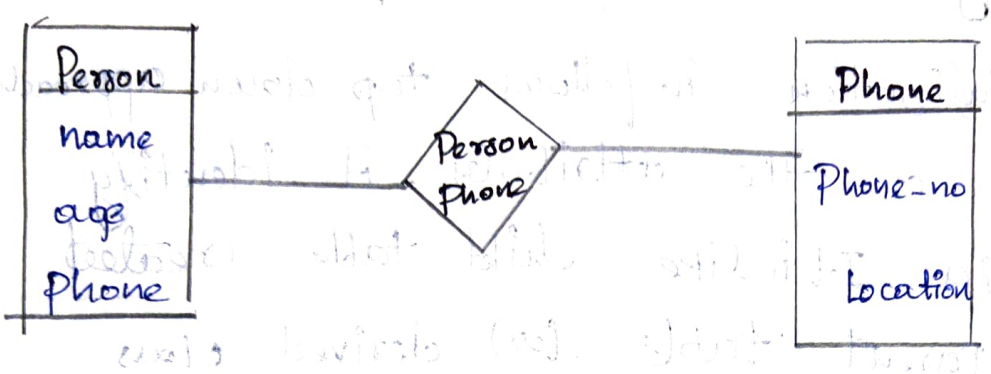
* Zero no. of times the entity takes Place in a relationship total no. of Participation of an entity in a relationship.
* Maximum value atmost one time Participation of an entity in a relationship. n no. of times the entity

E-R model design issue

Use of entity set and Attribute

* Consider an entity Person, attribute Phone number. that Person can have this phone number as office LAN number or Personal phone number. So we need

Create separate phone number entity to add an attribute phone no. and location



Use of Entity Set & Relationship set

one entity set can be related into too or more different entity set.

Ex, Registration of course related to Student and course available

Binary & Non Binary

Non binary relationship can be explained always by binary relationship. It is possible to have a distant Non binary relationship set by a distant binary relationship set

Placement of relationship attribute

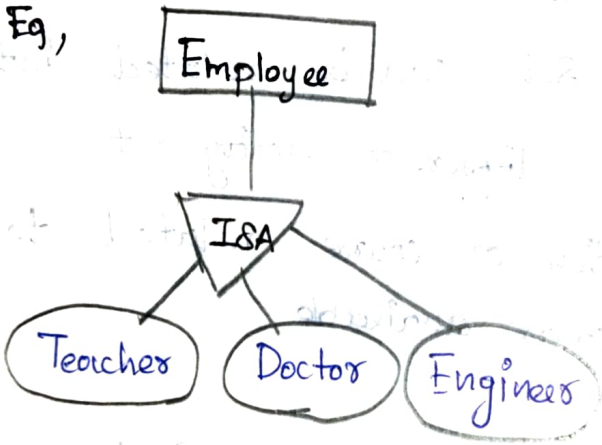
The design of placing descriptive attribute should reflect the characteristics of entity

been used by a model E-R diagr

Entity enhanced relationship

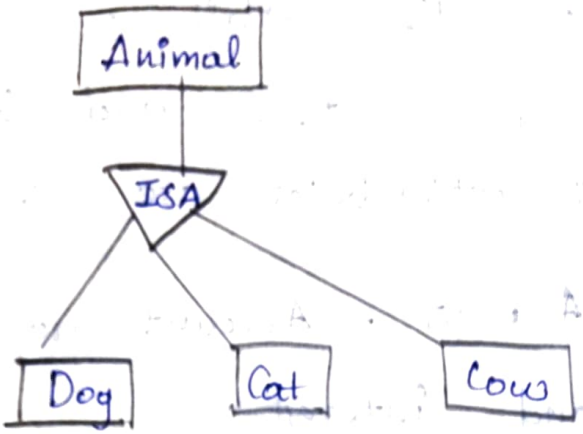
⇒ Specialization in follows top down ap based on the attributes. It identif entities. It is like child table creat by parent table (or) derived class from the existing class table

Eg,



* Generalization follows Bottom Up approa It works based on the child table finds the mother table It finds bas class from the derived class

* Child entity take a attribute from the higher level of entity it perfor the multilevel inheritance in the entity set

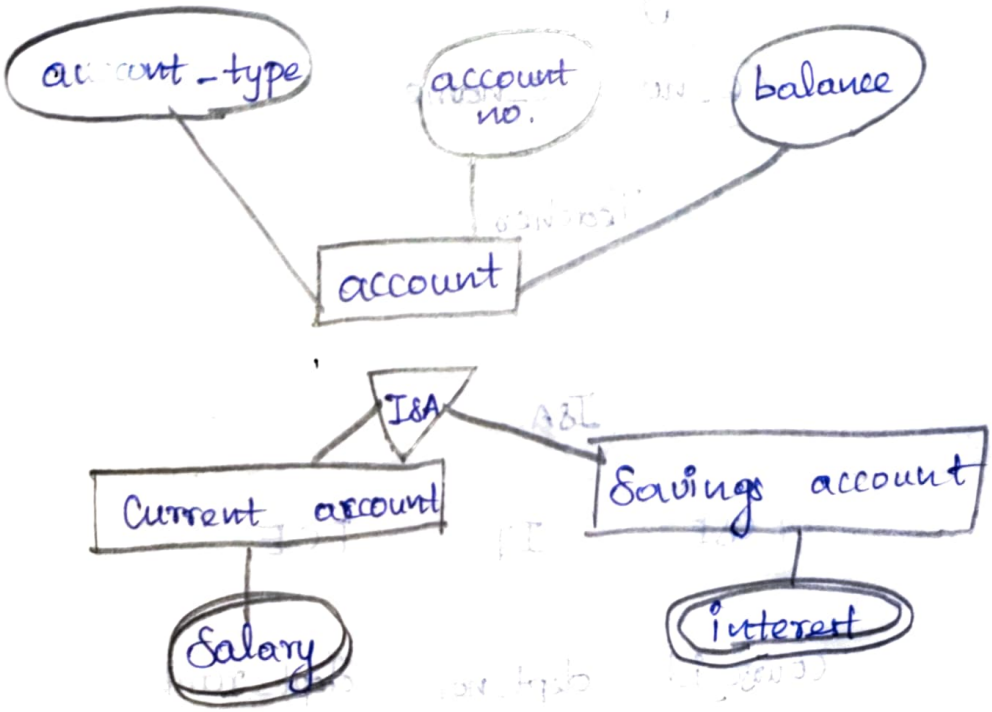


Design a constrain on Specialization and Generalization.

Condition: Defined constrain

*Lower level entities attributes are identified based on the condition satisfied by the Upper level entities.

*It can take Single level inheritance or Multilevel inheritance



⇒ Based upon the attributes such as account type, account_no and balance lower level attributes are identified.

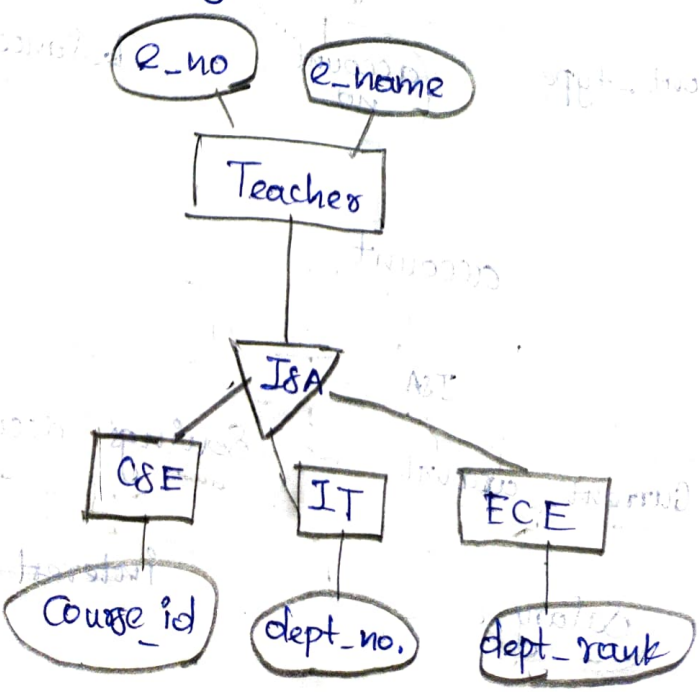
* Savings Account - Account no, account balance and interest

* Current Account - Account no, account balance and Salary

User Defined Constrain

* based on the user level the lower le entities attributes are identified

* Constrain based on whether attribute o an entity belong to one entity or mo than one entity



Constrain based on completeness

It specifies occurance of an each entity supertype may be atleast member of one subtype.

Partial Completeness

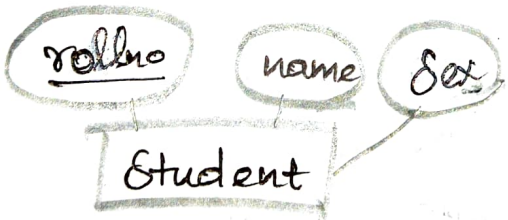
Not all the member of supertype will be member of subtype

Full Completeness

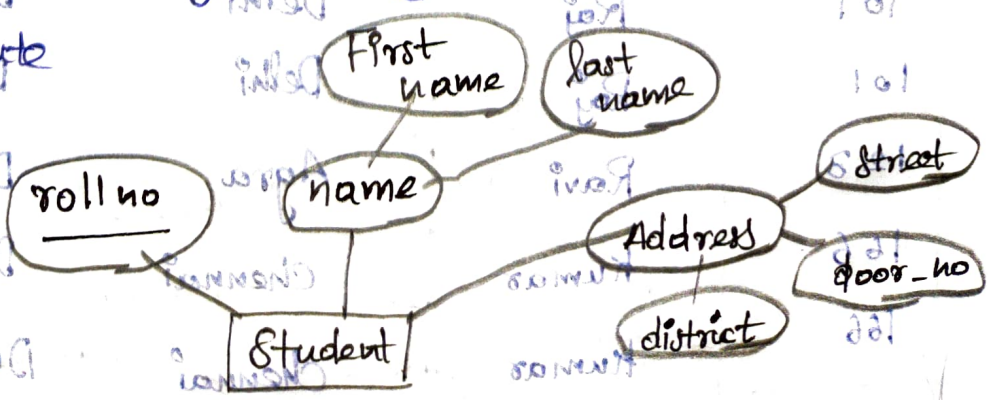
Every supertype occurance must be a member of one subtype

E-R to relational mapping

⇒ Strong entity with single attribute



⇒ For a strong entity with composite attribute



* For a strong entity set with any no. of composite attribute will require only one table in a relational model. While conversion simple attributes of the composite attribute are taken into the account and not the composite attribute itself.

Need for Normalization

→ **Anomalies in DBMS.**

There are 3 types of anomalies that occur when the database is not normalized. These are insertion, updation and deletion anomally.

Eg

Emp_id	Emp_name	Emp_address	Emp_dept
101	Raj	Delhi	D001
101	Raj	Delhi	D002
123	Ravi	Agra	D890
166	Kumar	Chennai	D900
166	Kumar	Chennai	D004

1] Update Anomaly

2] Insert Anomaly

3] Delete Anomaly

Properties of Normalization

i] Loseless decomposition

ii] Dependency Preservation

i] When the decomposition of relation are into r1 and r2 produce the same relation r Joining r1 and r2 is called <u>loseless decomposition</u>.

Ex Student

Reg_No	Student_name	dept
1	bala	cse
2	nandha	It
3	Pandi	cse

Reg_No	Student_name
1	Bala
2	Nandha
3	Pandi

Reg_No	dept
1	CSE
2	IT
3	CSE

iii] Dependency Preservation

⇒ The decomposition of relation r functional dependency $f = r_1 r_2$ to functional dependency f_1 and f_2 respectively. It is known as dependancy preservation.

R(ABCDEG)

$F = \{ AB \rightarrow C \quad B \rightarrow D \quad AC \rightarrow B \quad BC \rightarrow A \quad AD \rightarrow E \quad E \rightarrow G \}$

Decompose into $R_1(AB)$ $R_2(Bc)$ $R_3(ABDE)$ and $R_4(EG$

First Normal Form (1NF)

⇒ All attributes (Column) in the entity (tabb) must be single valued

⇒ Repeating or multi valued attributes are mov into a separate entity (tabb)

Eg,

Customer

Cid	Name	Contact No
C01	aaa	{123456789}
C02	bbb	{123,333,456}
C03	ccc	

Soln:

Customer

Cid	Name
C01	aaa
C02	bbb
C03	ccc

Customer Contact

Cid	Contact_No
C01	123456789
C02	123
C02	333
C02	456

The above tables are not in normalized form (1NF)

Properties of 1NF

* Each attribute must be atomic
* Value in a attribute must be of same domain
* All attributes should have unique name

Second Normal Form (2NF)

* Each entity must be 1NF
* Each attribute within the entity must solely depend on unique identifier of the entity.

Properties of 2NF

* It must be 1NF. It should not have a Partial dependency.

Consider the following relation

Teacher_id	Subject	Teacher_age
111	Maths	38
111	Physics	38
222	Biology	38
333	Physics	40
333	Chemistry	40

→ The above table is in 1NF, However it is not in 2NF, because Non-prime attribute (teacher_age) is dependent on (teacher_id) alone, which is a Proper subset of candidate key.

→ To make the table satisfied 2NF, the relation is split into two tables like this

Teacher Details

Teacher_Id	Teacher_age
111	38
222	38
333	40

Teacher Subject

Teacher_id	Subject
111	maths
111	physics
222	Biology
333	Physics
333	Chemistry

Prime and Non-prime * attribute is part of a Candidate key is

The attribute is not part of a candidate key is known as Non-prime Attribute

Third Normal Form (3NF)

A table is said to be in 3NF, if it contains the following conditions.

* It should be in 2NF
* Transitive functional dependency should be removed. (In other words non prime attribute should be no dependent on Non-prime attribute.

Transaction

* Transaction is defined as the group of task Performed by a user or application
* It is a set of changes executed together to update or read the content of the database

Properties

A - Atomicity (rollback, abort, Commit)

C - Consistency

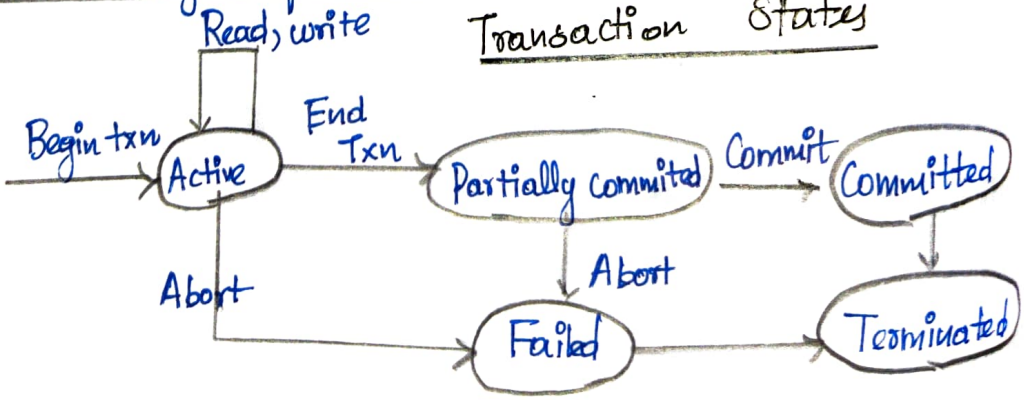
I - Isolation

D - Durability

Durability

* The changes applied to a database by a committed transaction should be permanently saved in database
* The changes must not be lost because of any failure.

Transaction States



* A txn goes into an active state immediately after it starts execution, where it can issue READ and WRITE operations. when the txn ends it moves to the Partially committed state.

* At this point, some recovery protocols need to ensure that a system failure will not result in an inability to record the changes of the txns permanently. Once this check is successful, the txn is said to have reached its commit point and enters committed state

*. However, a txn can go to the failed state if one of the checks fails or if the txn is aborted during its active state.

Eg,

Show how ACID Property can achieved during a system failure in case of money transaction from A to B.

Soln: Solution is to maintain every executed task of T_1 in A in a Volatile storage, only when all the tasks has been

completed, commit can be performed in the same way T2. of B takes place.

Student		
Reg_No	Name	Marks
1	aaa	80
2	bbb	85
3	ccc	90

SQL > Begin

SQL > insert into student values (4,'ddd',9

SQL > select * from student

SQL > commit

Schedules in Transactions

* A Schedule S for n transactions T1, T2 Tn, is ordering of a transaction operation in chronological Order.

* To run a multiple transaction con the scheduling of an instruction in transaction is termed as schedule.

Eg,

T_1	T_2
R(x)	
	W(x)

Characteristics of Schedule

* Each transaction must have complete set of instructions.
* Order of execution of instruction for a transaction must always remains same.
* Schedule can be done serially or concurrently

Different types of Schedule

* Serial Schedule
* Complete Schedule
* Recoverable Schedule
* Cascadeless schedule
* Strict schedule

Serial schedule

The operation of one transaction does not interleaved the operation of other transaction.

Eg,

T_1	T_2
A = 50	
read(A)	
A = A + 50	
write(A)	
	read(A)
	A = A + 50
	write(A)

Recoverable Schedule

If there are two transactions T_1 and T_2
If T_2 is accessing a data which is previously written by T_1, Then the commit operation of T_1 must appear before the commit operation of T_2 for each pair of transaction

T_1	T_2
A = 50	
Read(A)	
A = A + 50	
Write(A)	
	Read(A)
	A = A + 100
	Write(A)
Read(B)	
Write(B)	commit
Commit	

Cascadeless Schedule

If T_1 and T_2 are two transaction, T_2 access a data from T_1 then commit operation of T_1 must occur before the read operation of T_2

⇒ Cascading means effect of migration of one transaction to others.

Cascading Abort

T_1	T_2	T_3
Read1(x)		
Write 1 (x)		
	Read 2(x)	
	Write 2(x)	
		Read 3(x)
		Write 3(x)
Abort	Abort	Abort

Cascadeless Schedule

T_1	T_2	T_3
Read 1(A)		
Write 1(A)		
Commit	Read 2(x)	
	Write 2(x)	
	Commit	Read 3(x)
		Write 3(x)
		commit

⇒ In General, committed transaction does not come under the concept of abort, because transaction is already done or over.

Strict Schedule

In cascadeless schedule, transaction is read from T_1 to T_2 must done after the commit in T_1 but there is no criteria for write() operation.

T_1	T_2
Read(x)	
Write(x)	
	Write(x)
Commit	Read(x)

Transaction Isolation levels in DBMS

Transaction isolation level is referred the degree at which one transaction is isolated from other transaction.

There are 10 factors determine isolation level

i] Dirty- Read

ii] Non Repeatable-read

iii) Phantom read

i) Dirty read

T_1	T_2
begin	begin
Rollback read(x)	
	read(x)

The activity of the read performed by T_2 before T_1 commit is called Dirty read

ii) Non repeatable read.

	T_1	T_2
1.	begin	begin
2.	$x = 2$	
3.	read(x)	
4.		read(x)
5.		$x = x + 2$
6.		Commit
7.	read(x)	

Phantom Read

When same queries executed more than once in a transaction then it is called Phantom read

Serializability

A schedule 'S' of 'n' txns is serializable if it is equivalent to some serial schedule of the 'n' txns. This property is called serializability.

T_1	T_2
R	R
W	W
R	W
W	R

There are 3 types of schedule equivalences

1. Result equivalence
2. Conflict equivalence
3. View equivalence.

1. **Result**

If a same result is produced by schedule then it is called result equivalence

2. **View**

If two schedule perform a same set of operations in a similar manner. then it is referred to as view equivalence

3. Conflict Equivalence

A pair of operations are said to be in conflict if they satisfy the following conditions

1] Both operations belong to different txns
2] They access the same data
3] At least one of the operations is write operation.

Eg T_1: R(A), T_2: W(A)

T_1	T_2
R(A)	
	W(A)

Conflict Serializable Schedule

A schedule is conflict serializable, if it is conflict equivalent to any of serial schedule.

View Serializable Schedule

A schedule is view serializable, if it is view equivalent to any serial Schedule

Conditions

1. Initial read
2. Read / Write sequence
3. Final update

2 Phase Locking

1. Growing Phase (Locking Phase)
2. Shrinking Phase (Relief Phase)

1. Growing Phase

In this phase, new locks on the desired data item can be acquired but none can be released

2. Shrinking Phase

In this phase, existing locks can be released, but no new locks can be acq

Lock

A Lock is a variable which is associated with a data item describes the status of data item with respect to the operation applied to it.

2 Phase Locking System

2 phase locking perform concurrency control

Problems occurs due to 2PLock

1] Early Lock
2] Dead lock
3] Cascading rollback

Types of 2 Phase Locking

1] Strict Two phase locking

2] Rigorous Two phase locking

Shadow Paging (No Undo No Redo)

It is a recovery scheme in which data base is considered to be made up of no. of fixed size disk pages

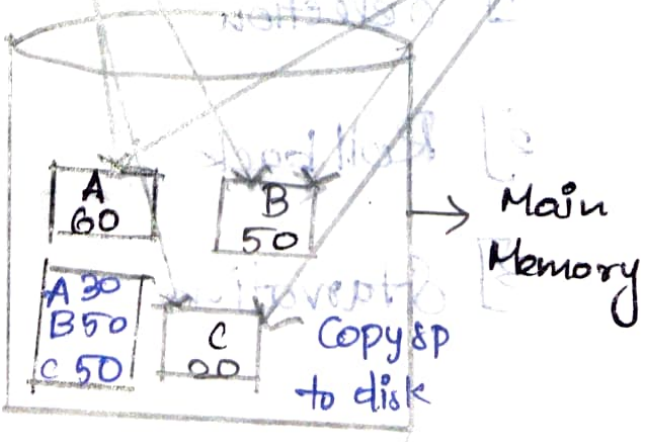
Structure :

Current page

B.No	Pointer
1	A 30
2	B 50
3	C 50

Shadow Page

B.No	Pointer
1	A 60-30=30
2	B 50
3	C 20+30=50



Advantages

* Log file is not necessary
* No UNDO No REDO operation

Disadvantages

* The use of pointer is overhead
* Data fragmentation

Transaction Recovery

To ensure atomicity despite, we first output information describing the modifications to stable storage without modifying the database itself

There are two approaches for recovery

1] Log based recovery
2] Shadow paging

Deadlock Recovery

1] Selection
2] Roll back
3] Starvation

Write Ahead Logging Rule (WAL)

Approaches in Transaction Recovery

1] Deferred database modification

2] Immediate database modification

ARIES Algorithm

⇒ It is a recovering algorithm

⇒ Algorithm for Recovery and Isolation Exploiting Semantics

⇒ It is based on Write Ahead Log protocol

⇒ ARIES has 3 phases in order

1] Analysis

2] Redo

3] Undo

4 Types of Log record

<START>

<UPDATE>

<COMMIT>

<ABORT>

⇒ Log is most commonly used structure for recording the modification that is to be

log file is maintained

<T, start>

<T, a=10,20>

<T, COMMIT>

Redo and Undo Operation

* There is an operation in which restore all the old value on to the disk called Undo operation or Rollback.

* There is an operation in which all the modified values are restored to the disk, This is called Rollfront or Redo operation

Write Ahead Logging Rule (WAL)

⇒ Before a block of data in a main memory is a output to the database. All log record pertaining to data in the block must have be output to the stable storage This rule is called WAL rule

Deferred Database Modification

No Redo / Undo Algorithm

T_1	T_2
Read(A,a)	Read(C,c)
a = a - 10 // 90	C = C - 20
Write(A,a)	Write(C,c)
Read(B,b)	// 280
b = b + 10 // 210	
Write(B,b)	

A = 100, B = 200 → T_1

C = 300 → T_2

Immediate Db Modification

In this technique the database is updated during the execution of a transaction even before it reaches the commit point

Log	Database
$<T_1$, start>	
$<T_1$ A 100 90>	A = 90
$<T_1$ B 200 210>	B = 210
$<T_1$ COMMIT>	
$<T_2$ START>	
$<T_2$ C, 300, 280>	C = 280
$<T_2$ COMMIT>	

Unit - 4

Implementation Techniques

Primary Memory - Cache and Main memory (Faster than others) (Expensive)

Secondary Memory - Magnetic Disk, Flash Memory (Non Volatile)

Tertiary Memory - Optical Disk, Magnetic Tape (Data Backup)

Features of a Memory

* Accessing Speed
* Performance per unit of data
* Reliability

According to features Storage is classified into different categories.

* Cache Memory
* Flash Memory
* Magnetic memory
* Optical Disk
* Main Memory
* Magnetic Tape

RAID (Redundant Array of Independent Disks)

→ It is a disk organization technique which manage a large no. of disks, Providing a view of a single disk of

i] High capacity and High speed by using multiple disks in Parallel

ii] High reliability by storing data redundantly so that data can be recovered even if a disk fails.

⇒ RAID employs the technique of disk mirroring or disk striping which involves partitioning each drive's storage space into units ranging from a sector (512 bytes) into several megabytes.

Organization of Records in Files

* It Stores an organized form
* Records must be in Organized form

5 types

i] Sequential File Organization
ii] Heap File Organization
iii] Hash File Organization
iv] B+ Tree File Organization
v] Clustered File Organization

1] Sequential File Organization

⇒ records are arranged in sequential order

Method's Type

1. Pile File Method
2. Sorted File Method

Adv

* Simple Design
* Cheaper Storage Mechanism

Disadv

* Fetching Particular Data

2] Heap File

Using some strategies in heap File

Adv

* Fetching of Record is easy

3] Hash File

Hash function is performed over the records

4] B+ Tree

is used

Tree structure to store data

5] Clustered File

Data Dictionary Storage

Data dictionary storage is mini database management system that manages met

Metadata

Metadata is a database information but not show the content

Roll No	Reg No	Name	Phone No.
22C8001	7236221	xxx	9050184841
22C8002	7236222	YYY	9750727203
22C8003	7236223	ZZZ	9872654310

Ex Data Dictionary

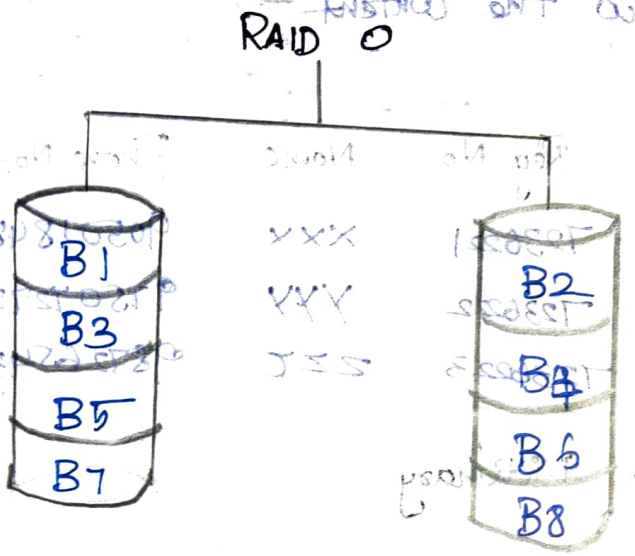
Attribute	Datatype	Information
RollNo	int	Primary key of the table
Reg No	int	Reg No of the Student
Name	varchar(20)	Name of the Student
Phone No	int	Phone No of the Student

RAID LEVELS

RAID

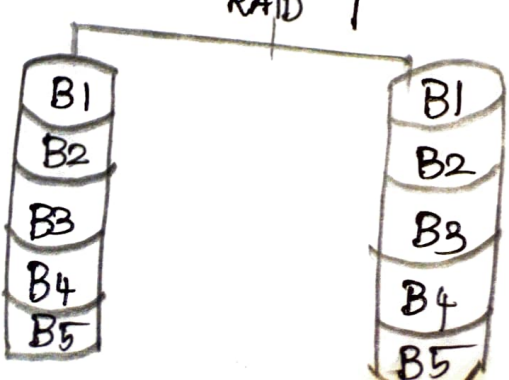
* Uses block level striping
* Data loss is not critical

RAID 0



RAID 1

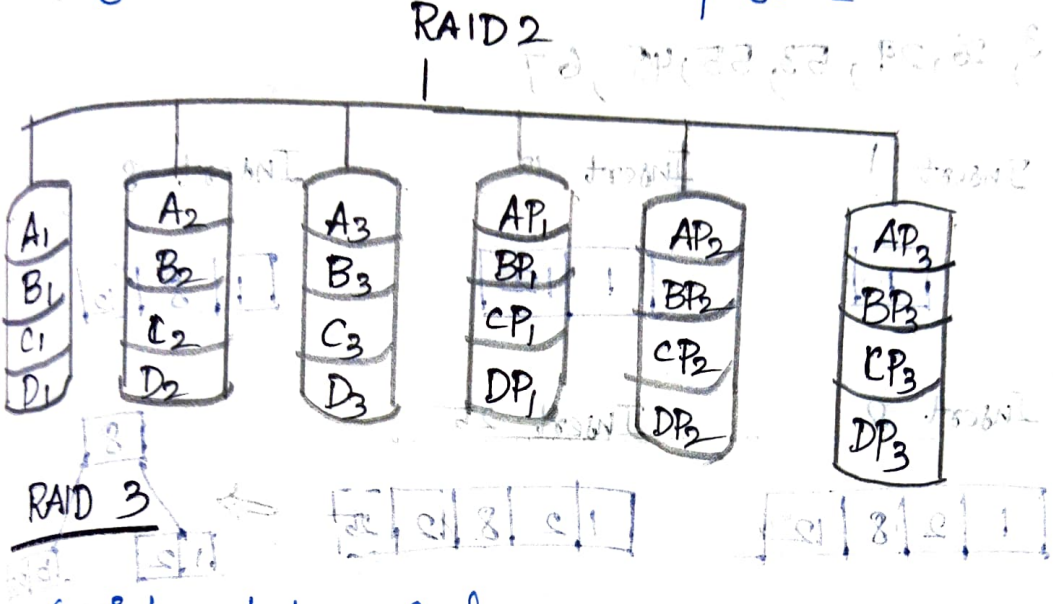
* Uses disk mirroring
* Read performance is improved
Since either disk can be read at the same time



RAID 2

⇒ Rarely used in practice data

⇒ High transfer rates are possible



RAID 3

Consists byte level striping with a dedicated parity disk

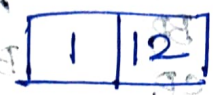
Faster data transfer.

B Tree Index

1, 12, 8, 2, 25, 6, 14, 28, 17, 7, 52, 16, 48, 68, 3, 26, 29, 53, 55, 45, 67

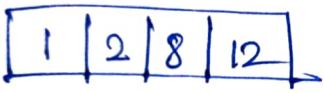
Insert 1

Insert 12

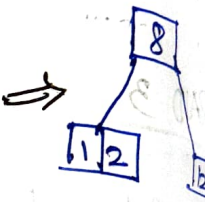


Insert 8

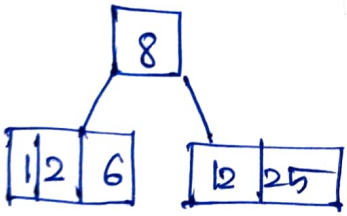
Insert 2



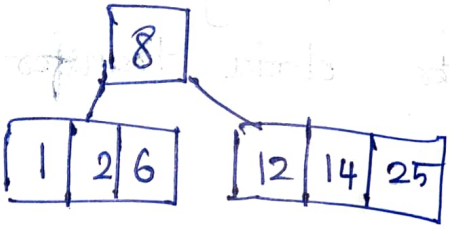
Insert 25



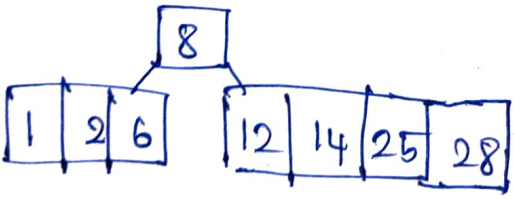
Insert 6

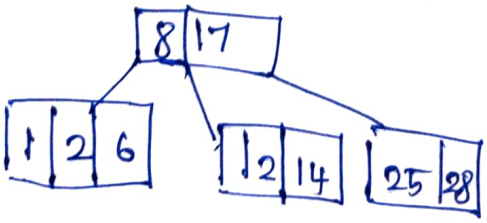


Insert 14

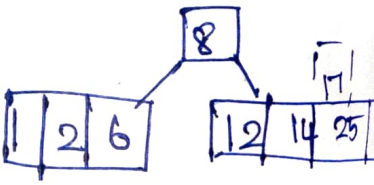


Insert 28

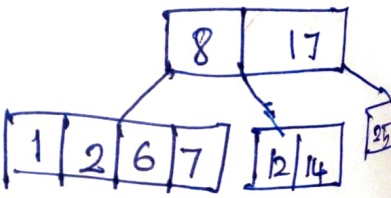




Insert 17



Insert 7



8 Tree Index File

1] Select Operation

S1 Linear Search (Brute Force)

$$\text{Cost} = \frac{br}{2} + 1 \text{ Seek}$$

cost for

S2 Primary Index

$$\text{Cost} = (h_i + 1) * (tT + tS)$$

S3 Non-key Attribute

$$\text{cost} = h_i(tT + tS) + tS + tT * b$$

S4 Secondary Index

$$\text{Cost} = (h_i + 1) * (tT + tS)$$

2] Join Operation

Types

1] J_1 Nested Loop Join

2] J_2 Single Loop Join

3] J_3 Sort - Merge Join

4] J_4 Hash-Join

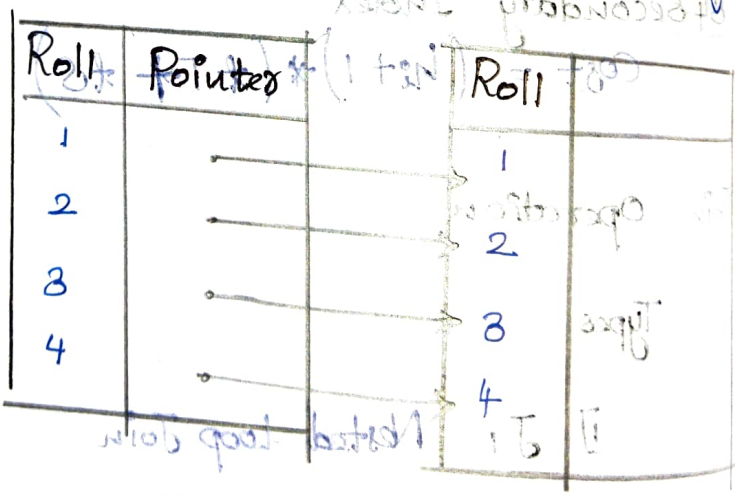
Primary Index

In primary index, there is a one one relationship between the entries in the index table and the records in the main table

Dense Primary Index

* The no. of entries in the index is the same as the no. of entries in the main table.

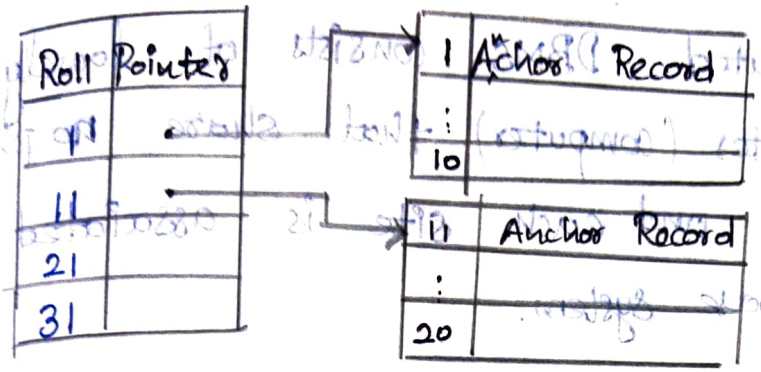
*. In other words, each and every record in the main table has an entry in the index.



Sparse (or) Non-dense primary index.

For larger tables, the dense primary index itself begins to grow in size. To keep the size of the index smaller,

instead of pointing to each and every record in the main table, the index points to the records in the main table in a gap.



Single Level Indexing

* A single value index is an auxillary file that makes it more efficient to search for a record in the datafile.

Multi level Indexing

There is an immense need to keep the index record in the main memory so as to speed up the operations. if a single level index is used then a large size index cannot be kept in the memory which leads to multiple disk accesses

Secondary Index

* In this technique two levels of indexing is used inorder to reduce the mapping size of the first level and in general

Unit - 5

Advanced Topics

Distributed Database

A distributed DBMS : consists of loosely coupled sites (computer) that share no physical components and each site is associated a database system.

Object Relational and NO SQL Database – 5

Mapping EER to ODB Schema :-

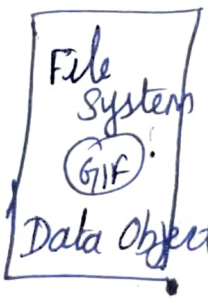
Step 1: Create an ODL class for each EER entity type or subclass

Step 2: Add relationship Properties (or) reference attributes for each binary relationship

Step 3: Include appropriate operation for each class

Object Identifier :-

In a database OID is a set of Integers that uniquely identifies each Row in a table. OID is typically is a sequence of numbers seperated by period



Properties of OID :-

* It is not visible to the external users
* It is Immutable
* OID value of a particular object should not change
* Even if an object is removed from the database the OID should be preserved

Reference Types :-

* The "Ref is System Generated" statement indicates that whenever a new person-type object is created, the System will assign it a unique system-generated identifier
* But it is also possible not have a system generated object identifier and use the traditional Key of the basic relational model is desired

Object Identifier using Reference types :-

* The user can specify the system-generated object identifier for the individual row by using

Ref is <old Attribute> <Value_Generation_method>;

* The Option for <value_Generation_method>
  System Generated : in which case the system will automatically generate a unique identifier
  Derived : in which the traditional method of using the user-provided Primary Key value to identify tuples is applied

Supertype and Subtype :-

Supertype :- is an entity type that has got relationship with one (or) more subtypes and it contains attributes that are common to its subtypes.

Subtype :- are subgroups of the supertype entity and have unique attributes but they will be different from each subtype.

Eg: People, Bank Account, Asset, Liability, Credit Card.

User defined Routines :-

The system creates the framework a form routine for you with the name Prefix for this routine is reusable, which means that you can use in other object in the Project.

Rowtype :-

Rowtype attribute provides a record type that represents a row in a database table. the record can store an entire row of data selected from the table (or) fetched from a cursor (or) cursor variable filled in a record and corresponding columns in a row have the same names and the datatypes.

UDT :-

* User defined data type (UDT) is a data type that derived from an existing data types. you can use UDTs to extend the built-in types already available and create your own customized data types

- → Distinct type
- → Structured type
- → Reference type
- → Array type
- → Row type
- → cursor type

Distinct type :-

* A distinct type is a UST that shares a internal representation with an existing built in types

PL/SQL Collection :

* A collection is an ordered group of element having the same data type. Each element is identified by unique subscript that represent its option in the collection
  * Nested table
  * Indesc by tables
  * Variable size array.

Indesc by table :-

Indesc by table is a set of key value pairs

Nested table :-

* Nested tables is like one-dimensional array with an arbitrary number of elements

Collection method :-

PL/SQL Provides the built in collection method that make collection easier to use.

Object query language :-

ObL is a version of the SQL that has been

designed for use in Network manager. The components create and intract with their database using OQL. use OQL to create new database (OR) insert data into existing database.

Convention and sample database:

* To illustrate the OQL keywords in use; a sample database has been used, the staff database

Features of OQL

* Database and table creation
* Inserting data into a table
* Selecting data from a table
* Counting row in a table

NoSQL:

* No SQL database store data differently than relational tables
* NoSQL database comes in a variety of types based on their data model.

Types of database - NoSQL

Document oriented database :

* It stores data in documents similar to JSON.
* Each document contain Pair of field and values.
* The values can typically variety of other object

key value database :

* Key value store is a simples type of database where each item contain key and values
* Each key is unique and associated with a single value

CAP Theorem :

* CAP Theorem is fundemental concept in distributed system Theory.

Consistency :

* It means all the nodes inside a network will have the same copies of a replicated data item visible for various transactions

Availability :-

* It means that each read (or) write request for a data item will either be processed sucessfully (or) will receive a message that the operation cant be completed.

Partion Tolerence :-

Partion tolerence means that the system can continue operating even if the network connecting the nodes has a fault that result in two or more partitions where the nodes in each partion can only communicate among each other.

Document based : mongo DB. Data model and

Crud Operations :-

(Create, Read, update and Delete) are the basic set of operation that allow user to interact with the mongoDB server.

Perform CRUD operation in MongoDB.

1. Create operation :-

* The create (or) insert operation used to insert (or) add new document in the collection

db.Create collection () It is used to create an empty collection

Column-based Hbase data model and CRUD operation :-

* Hbase is able to do what an RDMS cannot host very large, sparsely populated tables on cluster made from commodity hardware.

* HBase is a column-oriented database

* ~~The~~

Shared-nothing Architecture :-

* When building Scalable systems, we need to parallelize our computations

* This requires multiple processors

Sharding :-

* HBase Provides automatic sharding, with minimal duplication of state

Regions :-

* Tables are Partitioned vertically into different regions

The Hadoop Distributed file system :-

* In a production configurations HBase stores its underlying data on a file system called HDFS

* HDFS uses a cluster of computers of simulate a single

→ Resilance against hardware

→ Streaming Data access

→ Support for a large file