INTRODUCTION TO DBMS

UNIT-1

Introduction

★ DBMS :-

1- DBMS (Data Base Management System) is a software system that allow user to define, create and maintain a database and provides controlled access to the data.

2- A data base management system is basically a collection of programs that enables user to store, modify and extract information from a database as per requirement. DBMS is an intermediate layer between programs and the data. Programs access the database management system, which access the data.

There are different types of DBMS ranking from small system that run on personal computer to a huge system that run on mainframes. The following are the main examples of database applications —

(i) Computeral library system
(ii) Automated teller machine
(iii) Flight reservation system
(iv) Railway reservation system

3- A DBMS is a piece of software that provides services for accessing our data-base, while maintaining all the required features of the data.

4- Commercially available DBMS in the market are - Dbase, Foxpro, IMS, Oracle, Mysql, Sqlserver.

5- These systems allow users to create, update and extract information from their databases. Compare to manual filling system, the biggest advantages to a computerised system are speed, accuracy and accessibility.

* Components of DBMS :-

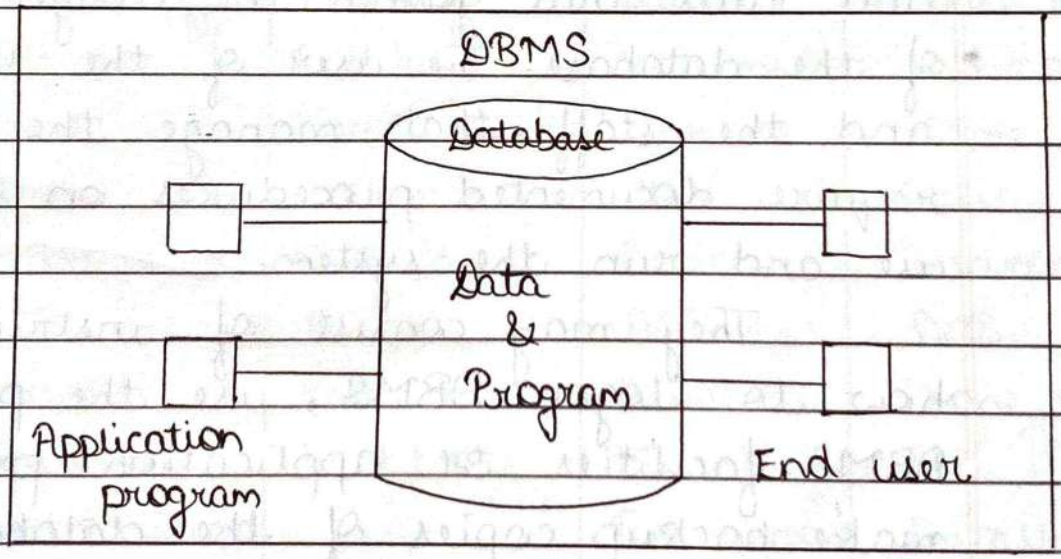
There are 5 major components in the 'database system environment and their interrelationship are -

1- Hardware
2- Software
3- Data
4- User
5- Procedures

1- Hardware - The hardware is the actual computer system used for keeping and accessing the database. Conventional DBMS hardware consist of secondary storage device, usually harddisk on which the database physically resides, device controllers. Databases runs on a range of machine, from microcomputer to large mainframes. Other hardware issues for a DBMS includes database machines, which is

hardware designed especially for support of database system.

2- **Software** — The software is the actual DBMS between the physical database itself and the user of the system. This a layer of software usually called DBMS. One general function provided by the DBMS is shielding the database users from complex hardware level detail. The DBMS allows the user to communicate with the database. It is the mediator between the database and the users. The DBMS control the access and help to maintain the consistency of the data. Utilities are usually included as part of the DBMS.



Components of DBMS

3- **Data** — It is the most important part of DBMS environment from end user point of view. The data act as a bridge between the

machine component and user component. The database contains the operational data. The database should contain all the data needed by the organisation. One of the major feature of database is that the actual data are seperated from program that use the data.

4- User — There are number of users who can access or retrieve data on demand using the application and the interface provided by the DBMS. Each type of user needs different software capability.

5- Procedures — Procedures are instructions and rules that govern the design and use of the database. The user of the system and the staff that manage the database require documented procedures on how to use and run the system.

They may consist of instructions how to logon DBMS, use the particular DBMS facilities or application program, make backup copies of the database, handle hardware and software failure.

★ Advantages of database :-

1. Redundancy — In filesystem, each application has its own private files, which cannot be shared between multiple applications. This can often leads to considerable redundancy in the stored data, which results in wastage of storage space. By having centralised database, most of disc be avoided. It is not possible that all redundancy can be eleminated. Sometimes there are sound business and technical reason for maintaining multiple copies of same data.

Example-

File system

General office	Library	Hostel	Account office
Roll no.	Roll no	Roll no	Roll no.
Name	Name	Name	Name
Class	Class	Class	Class
Father name	Address	Address	Address
DOB	DOB	DOB	DOB
Address	Phone no.	Phone no	Phone no.
Phone no.	No. of books issue	Mess bills	Fees
Previous record		Room no.	Installments
Attendence	Fine	etc.	Discount
Marks	etc.		Total
			Balance
			etc.

Page
Date / /

Database

General office	Library	Hostel	Account office
Roll no.	Roll no.	Roll no	Roll no
Name	No. of books issue	Mess bills	Fee
Class		Room no	Installments
Father name	Fine	etc.	Discount
DOB			Balance
Address			Total
Phone no.			
Previous record			
Attendance			
Marks			

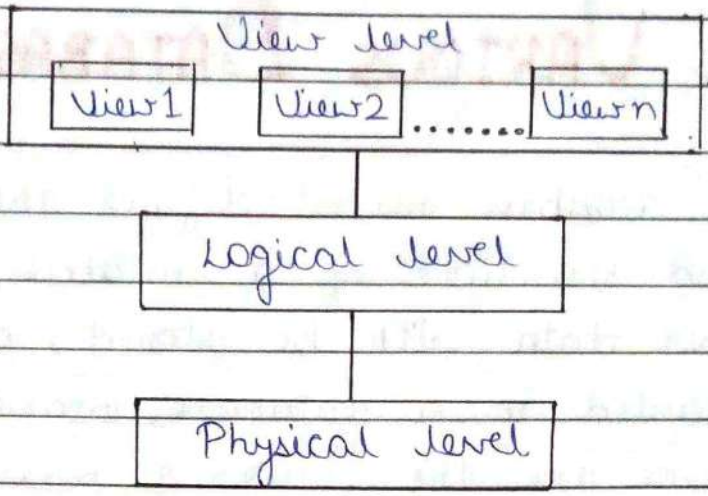
2. **Integrity** — Integrity of data means that data in database is always accurate, such that incorrect information cannot be stored in database. In order to maintain integrity of data, some integrity constants are enforced in database. A database should provide capabilities of defining and enforcing the constants.

3. **Inconsistancy** — Inconsistancy can be remove with the help of DBMS.

4. **Sharing of data** — Sharing of data allow the existing applications to use the data in database. It is also useful in developing new applications, which will

use the same data stored.

★ Levels of Database / Architecture of DBMS :-



1- Physical level — This is the lowest level of abstraction. The physical level describes complex low level data-structure in detail.

2- Logical level — This is the next higher level of abstraction and it describes how the data are actually stored and what relationship exist among those data. The logical level describes the whole data.

3- View level — This is the highest level of abstraction. The view level of abstraction exist to simply their interaction with the system.

Assignment

VARIOUS DATABASE MODELS

A database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system. There are the following models –

1. Hierarchical model
2. Network model
3. Object - oriented data model
4. Entity relationship model
5. Relational model
6. Flat data model

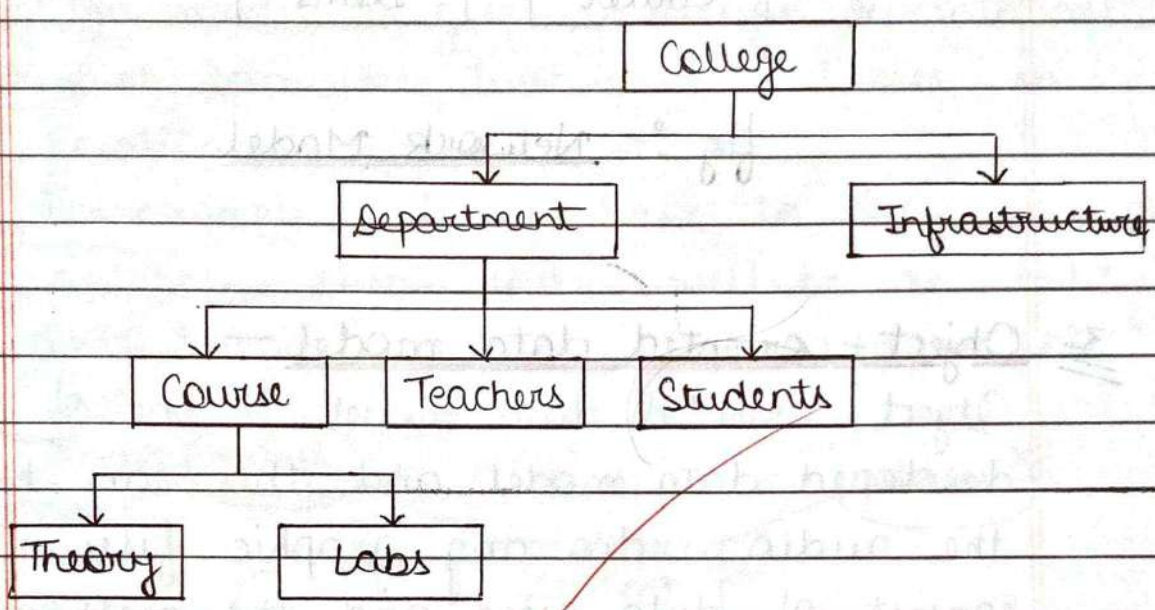
1. Hierarchical Model –

This database model organises data into a tree-like-structure, with a single root, to which all the other data is linked. The hierarchy starts from the root data, and expands like a tree, adding child nodes to the parent node. In this model a child node will only have a single parent node.

This model efficiently describes many real-

world relationships like index of a book, recipes etc.

For example – department is the parent entity called root and it has several children entities like students, professors and many more.



2. Network Model –

This is an extension of the hierarchical model. In this model data is organised more like a graph, and are allowed to have more than one parent node.

Network model has the entities which are organised in a graphical representation and some entities in the graph can be accessed through several paths. In this database model data is more related as more relationships are established in the database model.

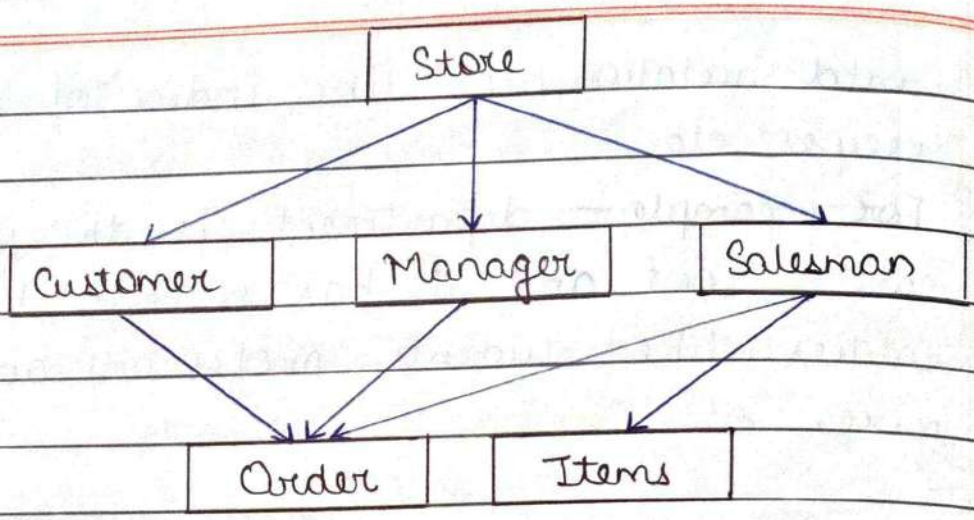


fig :- Network Model

3- Object-oriented data model —

Object-oriented data model is one of the developed data model and this can hold the audio, video ang graphic files. These consist of data piece and the methods which are the DBMS instructions.

Object 1: Maintenance Report

Date
Activity code
Route no.
Daily Production
Equipment Hours

Object 1 Instance

01-12-01
24
1-95
2.5
6.0

Object 2: Maintenance Activity

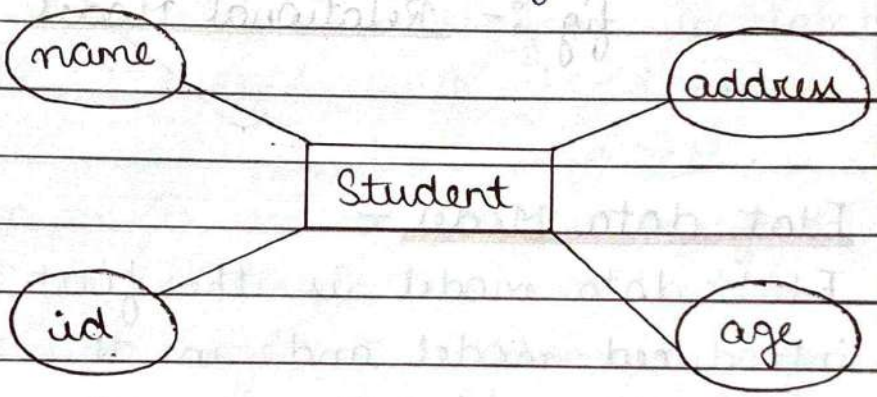
Activity code
Activity name
Production unit
Average Production Rate

4- Entity - relationship model —

In this database model, relationships are created by dividing object of interest into entity and its characteristics into attributes. Different entities are related using relationships.

This model is good to design a database, which can then turned into tables in relational model.

For example — If we have to design a school database, then Student will be an 'entity' with 'attributes' name, age, address etc.



5- Relational Model —

In this model, the data can be stored in the tables and this storing is called as relation. All the information related to a particular type is stored in rows of that table. Hence, tables are also known as 'relations' in relational model. Each column contains value from same domain and it is called as attribute.

student_Id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_Id	name	teacher
1	Java	Mr.J
2	C++	Miss C
3	C#	Mr.C Hash
4	Php	Mr.PHP

student_Id	subject_Id	marks
1	1	98
1	2	78
2	1	76
3	2	88

fig :- Relational Model

6- Flat data Model -

Flat data model is the first and foremost introduced model and in this all the data used is kept in the same plane. Since it was used earlier this model was not so scientific.

Roll no	Name	Course
5482	Mark	Web designing
5486	Steve	Java
5496	Smith	Oracle

fig :- flat data model

Richa written 04/09/19

* **Relational database Model** :- The relational database model is database based on first order predicate logic.

	Ad no	Name	Class	Section	Average
Tuple →	101	Anu	12	A	86
Key →	105	B	12	D	65
	203	Leena	11	B	95
	205	Madhu	10	C	75

In database, a relation means a table in which data are organised in the form of rows and columns, therefore in database, relation are equivalent to table.

- **Domain** — A domain is the original set of atomic values used to model data. In database management a domain refers to all possible unique value of particular column. For example— A domain of gender column has two set of values that is male and female.
  The domain of marital status has a set of four values, i.e married, unmarried, widows and divorced.

- Tuple — Horizontal subset / information in a table is called tuple. The tuple is also known as record which keeps particular

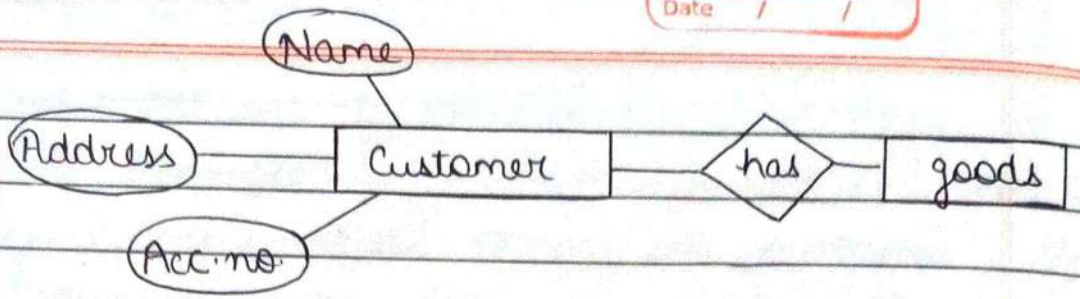
information of relation.
For example— In customer table, one row gives information about one customer. In student table one row gives information about one student.

- Key — Key are important part of relational database and a vital part of the structure of the table. They help enforce integrity and help to identify the relation between the tables. There are three main types of key- i.e, candidate key, primary key, foriegn key and alternate key.

* E-R data model :-

Entity relationship model is a high-level data model. It is based on perception of real world that consist of collection of basic objects called entities and relation between them.

- **Entities** — An entity is an object that has its existance in real world. It includes all those things about which data is collected. An entity may be a tangible object such as student, a place or a part. It may also be a non-tangible such as an event, a job title or a customer account.
  For ex— customer buy goods, customer has goods.



- **An entity set** — It is a set of entities of the same type that shares the same properties or attributes. Ex- The set of all persons who are customer at a bank.

- **Attributes** — Attributes are units that describes the characteristics or properties of entity. In a database entities are represented by table and attributes are represented by column.

- **Domain** — Domain of an attribute is a collection of all possible values an attribute can have.

- **Field / Column** — Field or column represent one related part of table and its smallest logical structure of storage in database. It holds a piece of information about an item of subject.

- **Record / Row** — A record is a collection of multiple related fields that can be treated as a unit.

- **Table** — A table is a name collection of

logically related multiple records.

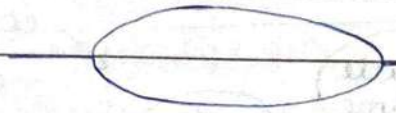
Attributes

Name	Address	Acc. no.
Renu	ABC	23
Neha		

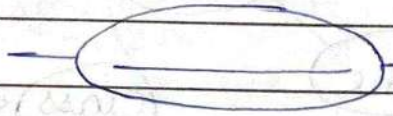
Row →

column ↑

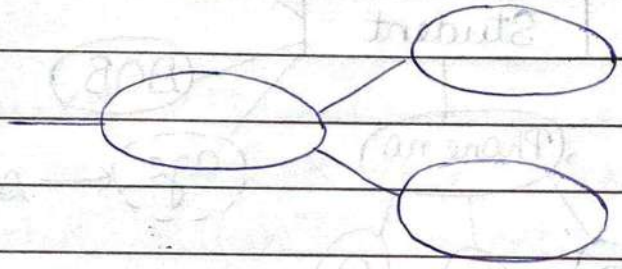
* Notations of E-R diagram :-



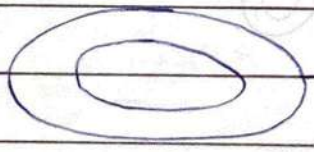
Attributes



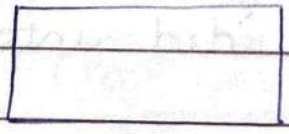
Key Attributes



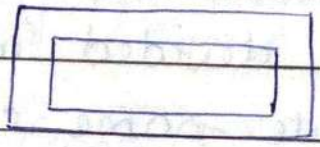
Composite attributes



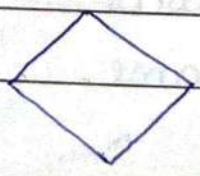
Multivalued attribute



Strong entity set

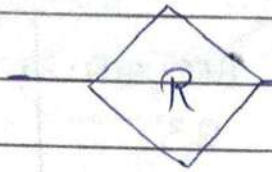


Weak entity set

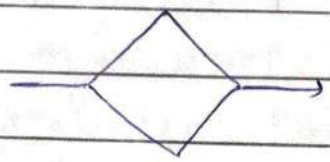


Relationship

——— links

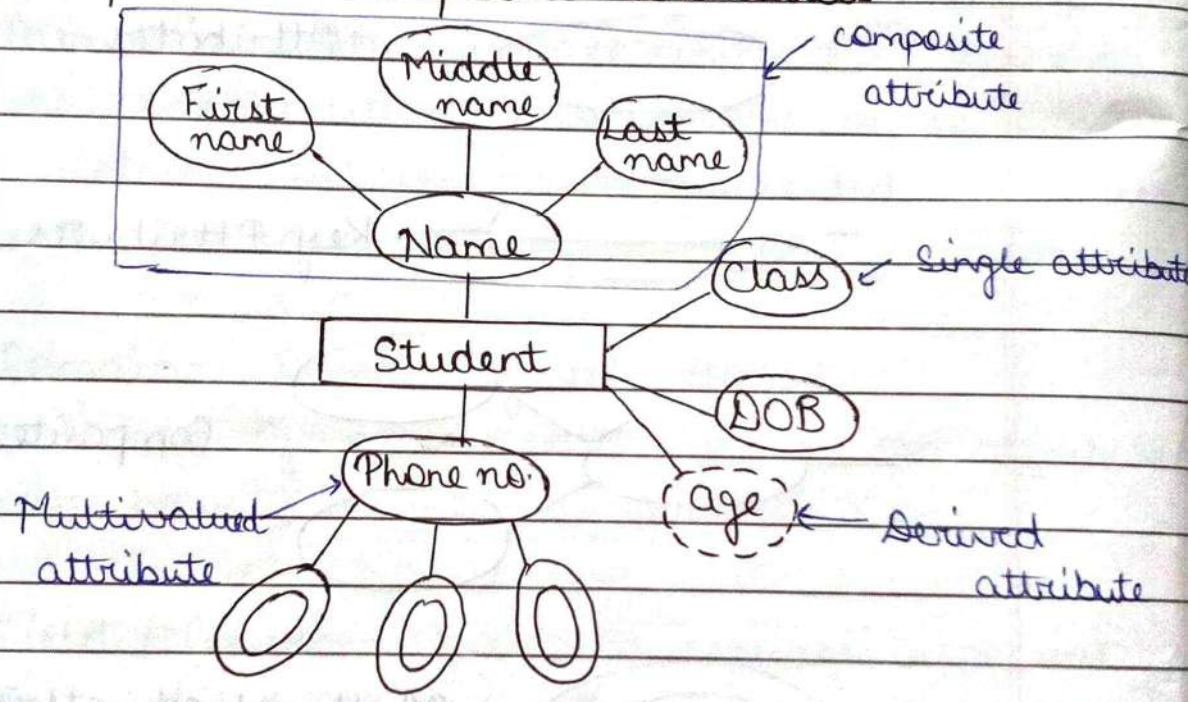


Many to many relationship



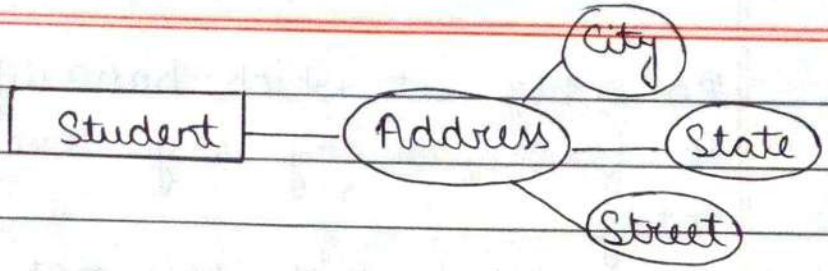
Many to one relationship

- Simple and composite attributes —



Simple attribute — Single attributes are those which cannot be divided into sub-parts.

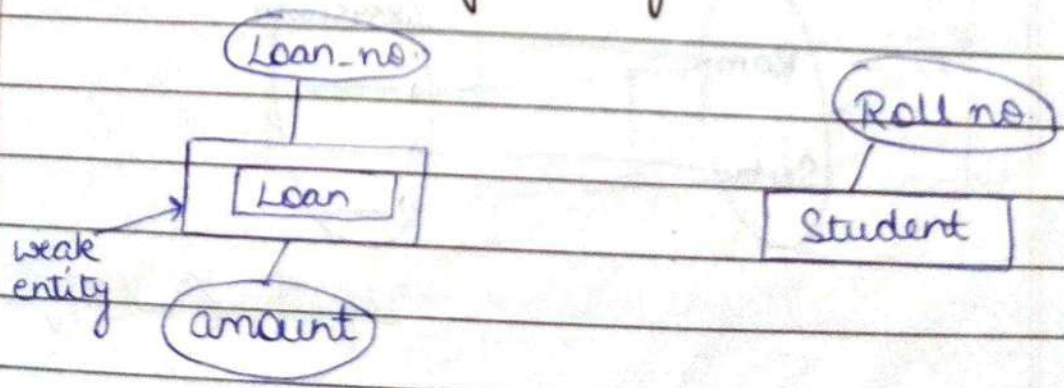
Composite attribute — Composite attributes are those which can be devided into sub-parts. For ex — The attribute name can be divided into 3 sub-parts — first name, middle name, last name.



- **Single valued or multi-valued attributes :-** The attribute which has single value for a particular entity is called single-valued attribute.
  Ex- DOB of an employee.
  Multivalued attributes has set of values for specific entity.
  Ex- Phone no. of an employee, customers in bank.

- **Derived attributes-** This type of attributes can be derived from the values of other related attributes or entity.
  For ex- Age of a student can be derived from his DOB (Date Of Birth).

- **Weak and strong entity-**

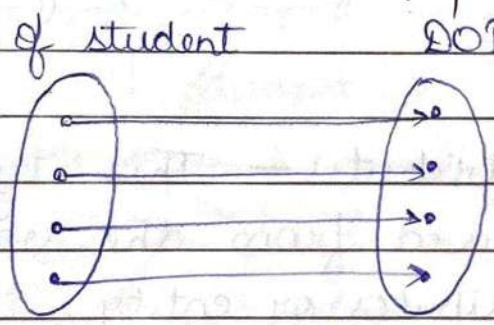


Pradeep
Date / /

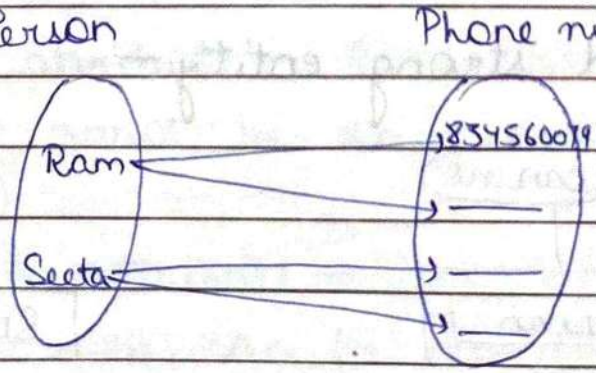
An entity set which has sufficient attribute to form primary key are called strong entity.
An entity which does not have primary key are called weak entity.

* Mapping constraints :-
Mapping constraints or mapping cardinalities or cardinality ratio express the number of entities to which another entity can be associated with a relationship set.

1- One to one relationship -

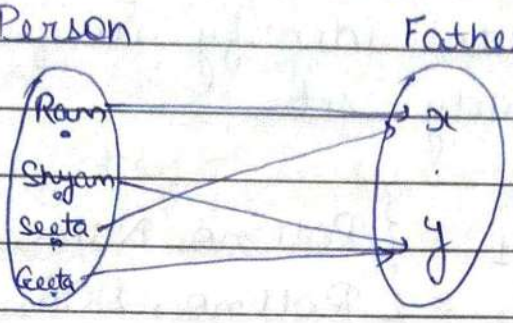


2- One to many -

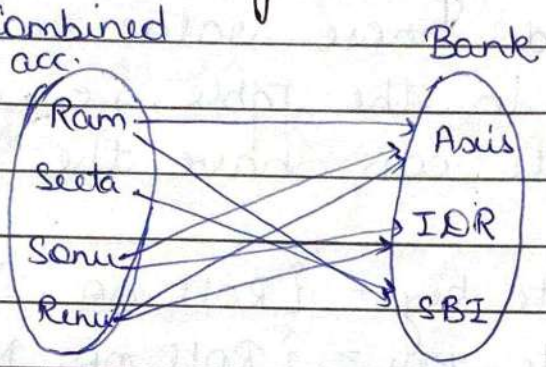


Page Pradeep
Date 19/Sep/2019

3- Many to one —



4- Many to many —



Key :- A key field is the column value in a table that is used to either uniquely identify a row of data in a table or stablish a relationship with another table.

Student

Roll no	Name	DOB	Course	Year

Types of keys —

1. Super key — A super key is a set of one

or more attributes that taken collectively allow us to identify uniquely an entity in the entity set.

Super key$_1$ = {Roll no, Name, DOB}
Super key$_2$ = {Roll no., DOB, Year}

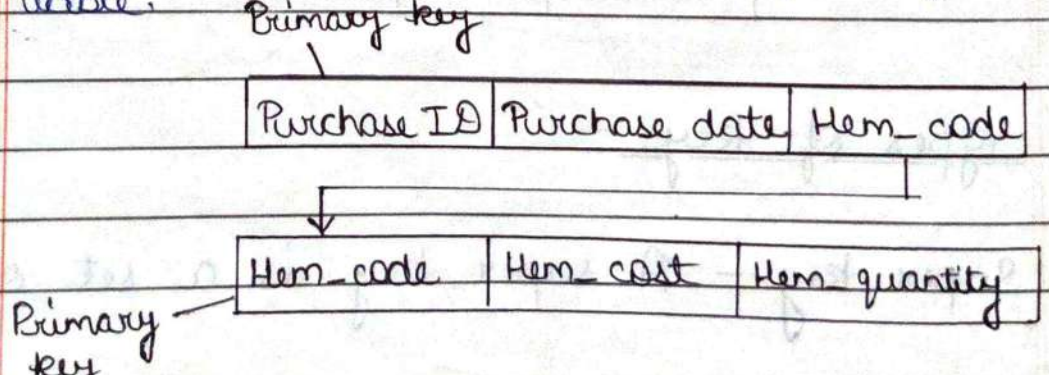
2- Candidate key - Combination of one or more field whose value uniquely identify a record in the table, i.e, no two record in a table can have the same key value.

Candidate key = {Roll no., DOB}
Candidate key = {Roll no, Name}

3- Primary key — Primary key uniquely identify record within a table. Primary key value cannot be changed.

Primary key = {Roll no.}

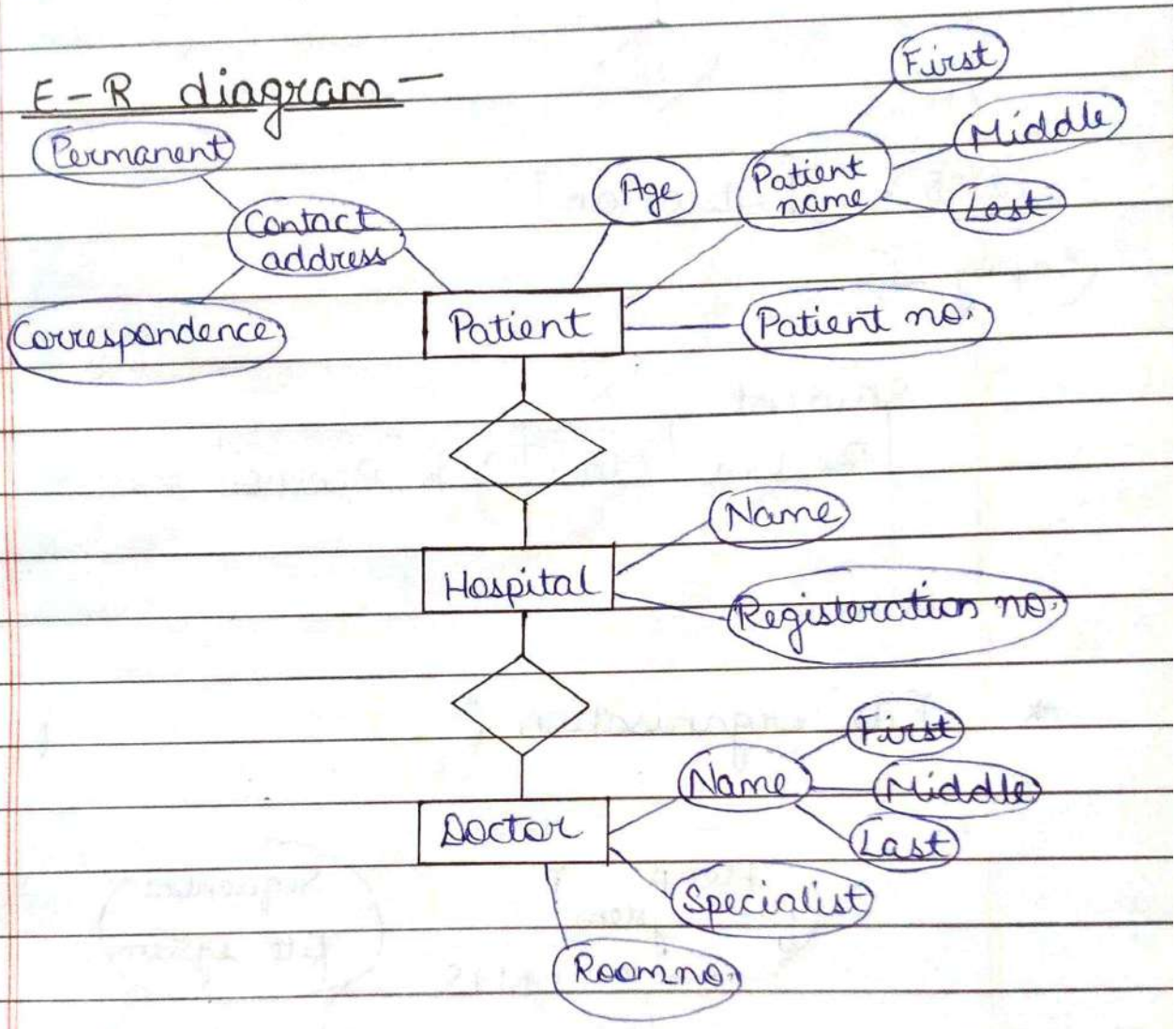
4- Foreign key — A foreign key is a combination of one or more column in a table ~~at~~ that reference a primary key in other table.



5- Alternate key – Any candidate key other than primary key are called alternate key.

6- Secondary key – Secondary key is used to increase speed and retrieval.

- E-R diagram –



Patient

Patient no.	Patient name	Age	Contact address	
	First, Middle, Last		Permanent	Correspondence

Hospital

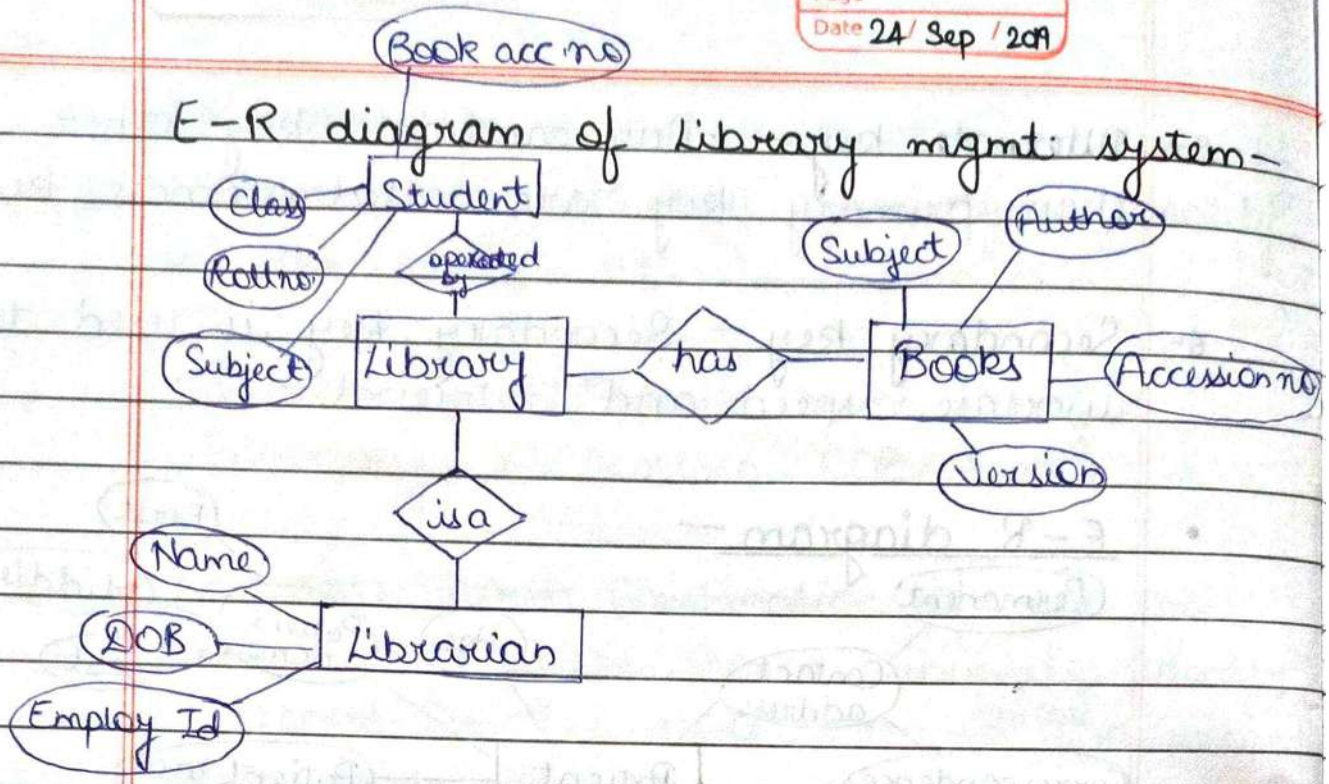
Registeration no.	Name

Doctor

Name	Specialist	Room no.

Pradeep
Page
Date 24/ Sep /209

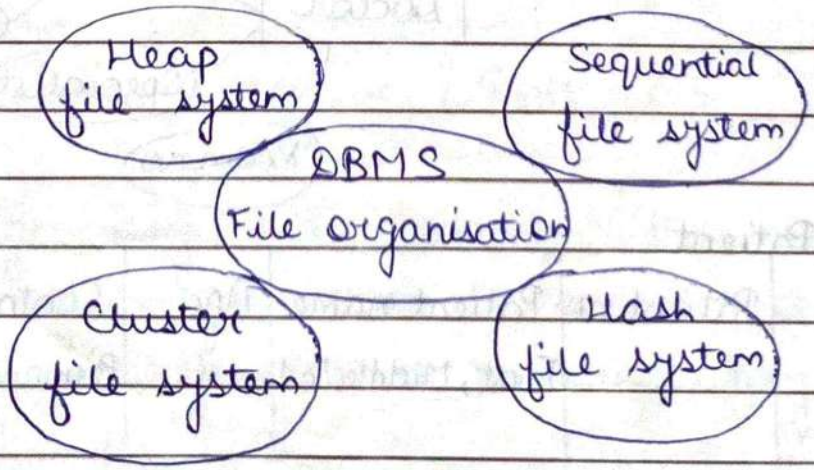
E-R diagram of Library mgmt system -



Student

Roll no.	Class	Sub	Acc. no.

* File organisation :-



File organisation defines how file records are mapped onto this block. We have four types of file organisation to

organise file record.

1. Heap organisation — When a file is created using heap file organisation, the operating system allocates memory area to the file without any further accounting details. File record can be placed anywhere in that memory area. It is the responsibility of the software to manage the records. Heap file does not support any ordering, sequencing or indexing on its own.

2. Sequential file organisation — Every file record contain a data field to uniquely identify that record. In sequential file organisation records are placed in the files in some sequential order based ~~files~~ on the unique key field or search key.

* Database languages :-

1. Data definition languages
2. Data manipulation languages

1- Data definition languages — Data definition languages used by database administrator. It is a set of SQL command used to create, modify, delete database structure but not data. These commands are normally not used by general users.

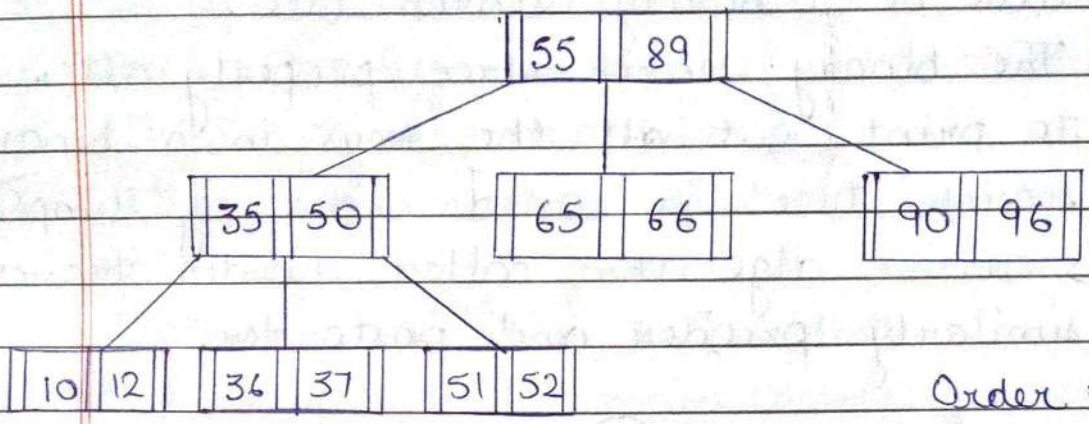
Data definition language is for standard command for different structures in the database. Data definition language statements are used to create structure of a table, modifying the existing structure of the table and remove the existing table. Some of the DDL statements are create table, alter table, draw table.

2. Data manipulation languages — A DML is a language that enables users to access or manipulate data or organised by the appropriate data model. The
The data manipulation language includes inserting data into tables, deleting data from the table, retrieving data and modifying the existing data.
The common DML statements are select, update, delete or insert.

★ DBA (Database Administrator) :-
The person having the central control over the system is called data base administrator.
The data base administrator is an information technology expert or well trained computer literate person who is responsible for the technical operations of database or all databases in an organisation.

UNIT-3

★ Index Sequential file access method :-



Order = 2

An index sequential access method is a file management technology developed by IBM and focus on fast retrieval of records which are maintained in the sorted order with the help of index.

When data are be stored in ISAM they are entered sequentially.

Unlike other database an ISAM method had small index making them search faster.

The methodology focused on direct access to specific record with the help of index.

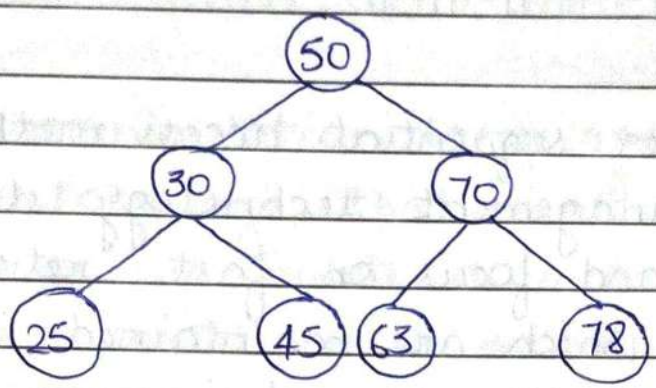
★ Binary search tree :- Binary search tree is a node based binary tree data-structure which has following properties —

1- The left sub-tree of a node contain only nodes with key lesser than the root nodes.

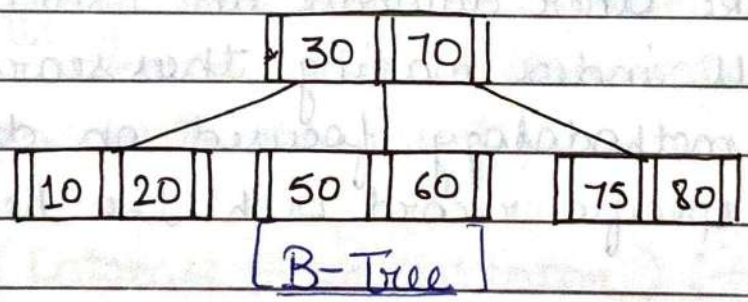
2- The right sub-tree of a node contains only nodes with key greater than the root nodes.

3- The left and right sub-tree each must also be a binary search tree.

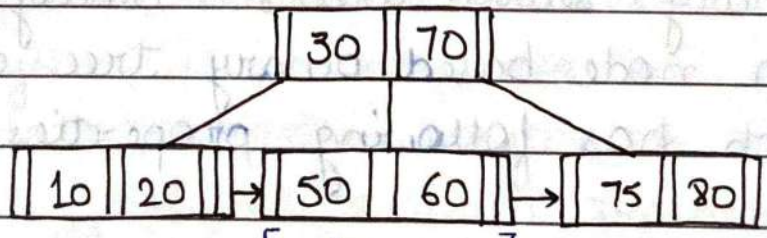
4- The binary search tree property allow us to print out all the keys in a binary search tree in sorted order by simple recursive algorithm called inorder traversal similarly preorder and postorder.



★ B- tree :-



B-Tree



B+ tree

1. B tree is a self balancing search tree. In
2. most of It is assume that everything is

in main memory.

3. To understand the use of B tree we must think that huge amount of that ~~that~~ cannot fit in main memory.

4. When the number of ~~tree~~ keys is high the data is read from disc in the form of blocks. ~~so~~ Disc this access time is very high compared to main memory access time.

5. The main idea using B tree is to reduce the number of disc accesses. Most of the tree operations (search, insert, delete, min, max).

6. The height of B tree is kept low by putting maximum number of possible keys in a B- tree node. Generally a B- tree node size is kept equal to the disc block size.

7. If height is low for B-tree, total disc access for most of the operations are reduced significantly compared to balanced binary search tree.

B$^+$ tree —

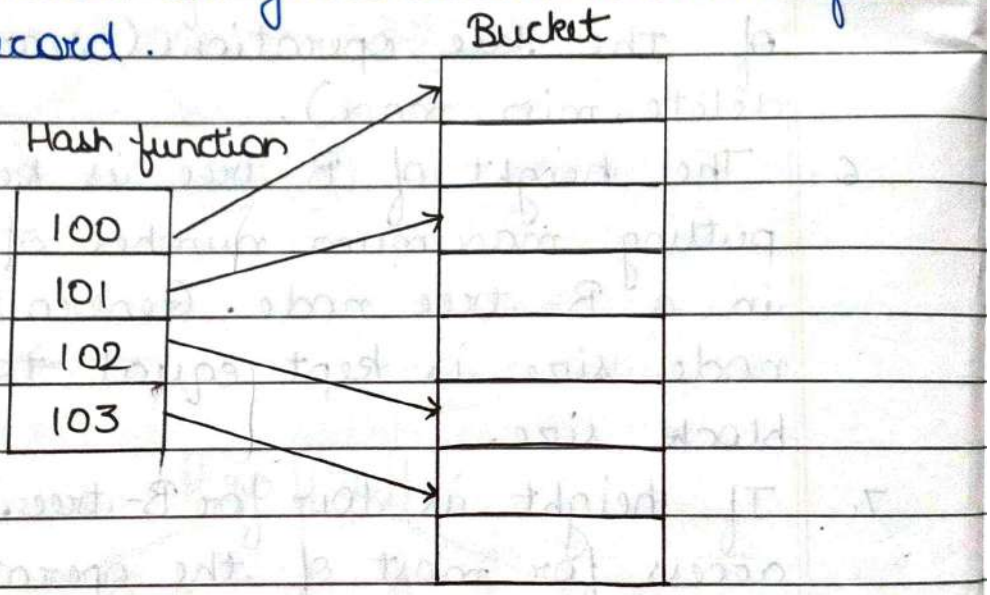
B$^+$ tree is an array tree with the variable of a large number of children per node. A B$^+$ tree consist of root, internal nodes and leaves. The root may be either a leave or a node with two or more children.

Pradeep
Page
Date 31 / Oct / 2019

* **Hashing :-** For use data-structure it can be almost next to impossible to search all the index values through all its level and then reach the destination data-block to retrieve the desired data.

Hashing is an effective technique to calculate the direct location of a data record on the disk.

Hashing uses hash function with search key as parameters to generate the address of a data-record.



* **Hash organisation :-**

Bucket — A hash file stores data in bucket format. Bucket is consider a unit of storage. A bucket typically stores one complete disk block, which in term can store one or more records.

Hash function – A hash function is a mapping function that maps all the set of search keys to the address where actual ~~add~~ record are placed. It is a function from search key to bucket addresses.

* Static hashing – In static hashing when a search key value is provided the hash function always compute the same address.
For example – mod-4 hash function is used then it shall generate only fixed values. The output address shall always be same for that function. The number of bucket provided remain unchanged at all times.

Operations –

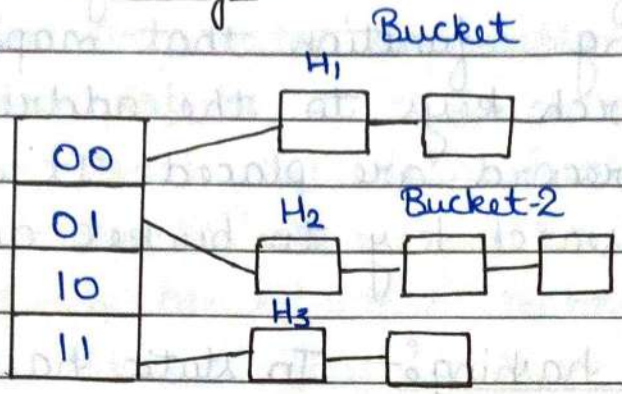
1.) Insertion – When a record is required to be entered using static hash, the hash function h computes the bucket address for search key K where the record will be stored

2.) Search – When a record need to be retrieve the same hash function can be used to retrieve the address of the bucket where the data is stored.

3.) Delete – This is simply a search followed by a deletion operation.

Pradeep
Page
Date 1 / Nov / 2019

* Dynamic hashing :-



→ The problem in static hashing is that it cannot expand or shrink dynamically as the size of database grow or shrink.

→ Dynamic hashing provides a mechanism which data bucket are added and removed dynamically

→ Dynamic hashing is also known as extended hashing.

→ Hashing function in dynamic hashing is made to produce large number of values and only few are used initially.

UNIT – 4

Relational Database Model

→ Relational database management system is a database model based on first order predicate logic.

→ In database a relation means a table in which data are organised in the form of row and columns therefore in database relations are equivalent to table.

→ An important feature of relational system is that a single database can spread across several tables.

→ Today, various commercial database are available i.e. Oracle, Microsoft SQL Server.

★ Properties of Relational table –

1. Values are atomic.
2. Column values are of the same kind.
3. Each row is unique.
4. Each column has a unique name.
5. The sequence of row is insignificant.
6. The sequence of column is insignificant.

★ E.F. CODD Rule for RDBMS :–

CODD published 12 rules that define an ideal relational database and has provide a guideline for all relational database system.

1- The information rule- This rule requires that all data should be presented in tabular form. This is the basic of relational model.

2. Guaranteed access rule - All data should be accessible without ambiguity. This can be accomplished by combination of table, table name, primary key and column name.

3. Systematic treatment of null values - The RDBMS must support the null values to represent the missing information. There must be distinct from 0 and space.

4. Dynamic online catalogue Based on Relational model - System catalogue is a collection of tables that the DBMS maintain for its own use. These tables ~~called~~ hold the description of structure of the database.

5. Comprehensive data sub-language rule - The database must support one defined language for that include functionalities for data definition, data manipulation, data ambiguity and database transaction control.
For example - SQL structured query language

6. View updating rule - Data can be presented in different logical combination called views.

Page
Date / /

Assignment

* 12 RULES OF CODD :-

7. High-level insert, update and delete rule — A database must support high-level insertion, updation and deletion. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records.

8. Physical data independence — The data stored in a database must be independent of the applica-tions that access the database. Any change in the physical structure of a database must not have any impact on how the data is being accessed by external applications.

9. Logical data independence — The logical data in a database must be independent of its user's view (application). Any change in logical data must not affect the applications using it. For example — if two tables are merged or one is split into two different tables, there should be no impact or change on the user application. This is one of the most difficult rule to apply.

10. Integrity independence – A database must be independent of the application that uses it. All its integrity constraints can be independently modified without the need of any change in the application. This rule makes a database independent of the front-end application and its interface.

11. Distribution independence – The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the foundation of distributed database systems.

12. Non-Subversion rule – If a system has an interface that provides access to low-level records, then the interface must not be able to subvert the system and bypass security and integrity constraints.

* SHORT NOTE ON RDBMS :-

RDBMS stands for "Relational Database Management System." An RDBMS is a DBMS designed specifically for relational databases. Therefore, RDBMSes are subset of DBMSes.

All modern database management systems like SQL, MS SQL Server, IBM DB2, ORACLE, My-SQL and Microsoft Access are based on RDBMS.

It is called Relational Database Management System (RDBMS) because it is based on relational model introduced by E.F. Codd.

How it works — Data is represented in terms of tuples (rows) in RDBMS. Relational database is most commonly used database. It contains number of tables and each table has its own primary key. Due to a collection of organised set of tables, data can be accessed easily in RDBMS.
It is "relational" because the values within each table are related to each other. The relational structure makes it possible to run queries across multiple tables at once. An RDBMS also provide a visual representation of the data.
For example: it may display data in a tables like a spreadsheet, allowing you to view and even edit individual values in the table. Some RDBMS programs allows to create forms that can streamline entering, editing and deleting data.

RDBMS is an advanced version of DBMS system. It came into existence during 1970's. RDBMS also allows the organisation to access data more efficiently than DBMS. RDBMS is a powerful data management system and is widely used across the world.

Pradeep
Date / /

* DIFFERENCE BETWEEN DBMS AND RDBMS :-

	DBMS	RDBMS
1.	DBMS applications store data as file.	RDBMS applications store data in a tabular form.
2.	In DBMS, data is generally stored in either a hierarchical form or a navigational form.	In RDBMS, the tables have an identifier called primary key and the data values are stored in the form of tables.
3.	Normalization is not present in DBMS.	Normalization is present in RDBMS.
4.	DBMS does not apply any security with regards to data manipulation.	RDBMS defines the integrity constraints for the purpose of ACID (Atomocity, Consistency, Isolation and Durability) property.
5.	DBMS has to provide some uniform methods to access the stored information.	RDBMS system supports a tabular structure of the data and a relationship between them to access the stored information.
6.	DBMS does not support distributed database.	RDBMS supports distributed database.
7.	DBMS is meant to be for small organisation and deal with small data. It supports single user.	RDBMS is designed to handle large amount of data. It supports multiple users.
8.	No relationship between data	Data is stored in the form of tables which are related to

	each-other.
9. Data redundancy is common in this model.	Keys and indexes do not allow data redundancy.
10. Data fetching is slower for large amount of data.	Data fetching is fast because of relational approach.
11. Low software and hardware necessities.	Higher software and hardware necessities.
Examples: XML, Microsoft Access etc.	Examples: MySQL, Oracle, PostgreSQL, SQL Server etc.

* RELATIONAL CONSTRAINTS :-

Relational constraints are the restrictions imposed on the database contents and operations. They ensure the correctness of data in the database.

Relational Constraints →
- Domain Constraint
- Tuple Uniqueness Constraint
- Key Constraint
- Entity Integrity Constraint
- Referential Integrity Constrai

1. **Domain Constraint** – Domain constraint defines the domain or set of values for an attribute. It specifies that the value taken by the attribute must be the atomic value from its domain.

Example –

Consider the following Student table –

STU_ID	Name	Age
S001	Akshay	20
S002	Shashank	21
S003	Rahul	A

Here, value 'A' is not allowed since only integer values can be taken by the age attribute.

2. Tuple Uniqueness Constraint – Tuple uniqueness constraint specifies that all the tuples must be necessarily unique in any relation.

Example –

STU_ID	Name	Age
S001	Akshay	20
S001	Akshay	20
S003	Shashank	20
S004	Rahul	21

This relation does not satisfy the tuple uniqueness constraint since here all the tuples are not unique.

3. Key Constraint – Key constraint specifies that in any relation – All the values of primary key must be unique. The value of primary key must not be null.

Example –

STU ID	Name	Age
S001	Akshay	20
S001	Abhishek	21
S003	Shashank	21
S004	Rahul	20

This relation does not satisfy the key constraint as here all the values of primary key are not unique

4. Entity Integrity Constraint – Entity integrity constraint specifies that no attribute of primary key must contain a null value in any relation. This is because the presence of null value in the primary key violates the uniqueness property.

Example –

STU_ID	Name	Age
S001	Akshay	20
S002	Abhishek	21
S003	Shashank	21
	Rahul	20

This relation does not satisfy the entity integrity constraint as here the primary key contains a NULL value.

5. Referential Integrity Constraint – This constraint is enforced when a foreign key references the primary key of a relation. It specifies that all the values taken by the foreign key must either be available in the relation of the primary key or be null.

Example –

Student

STU_ID	Name	Dept_no
S001	Akshay	D10
S002	Shashank	D11
S003	Rahul	D14

Department

Dept no	Dept name
D10	ASET
D11	ALS
D12	ASFL

Here, the relation 'Student' does not satisfy the

referential integrity constraint. This is because in relation 'Department', no value of primary key specifies department no. 14.

Richa
06/11/19

★ Relational Algebra :-

1- Selection- Selection in relational algebra return those tuple (record) in a relation that fulfill a condition (produce table contain subset of row)

Syntax :

? condition (relation)

Example- The table S (for Student)

Relation : Student

? class = 12 (S)

Student

Ad No.	Name	Class	Section	Average
101	Anu	12	A	85
105	Balu	12	D	65
205	Leena	11	B	95

Output

AdNo	Name	Class	Section	Avg
101	Anu	12	A	85
105	Balu	12	D	65

2- Projection- Projection in relational algebra return those column in a relation that given in the

attribute list.

Syntax:

π attribute list (Relation)

Ex- π Ad no, Name (S)

S. Student

Ad No.	Name	Class Section
101	Anu	
102	Balu	
103	Leena	
104	Neena	
105	Seema	
106	Ram	

3- Union- The union of operator is used to combine two or more tables. Each table within the union should have the same number of columns, similar data-types and also the column must be in same order.

Syntax-

σ (Student 1)

Union

σ (Student 2)

Or

π ad no, Name (Student 1)

Union

π ad.no, Name (Student 2)

Pradeep
Page
Date 5 / Nov / 2019

Table : Student

Ad No.	Name	Class
101	Ram	12
102	Shyam	12
103	Seeta	12

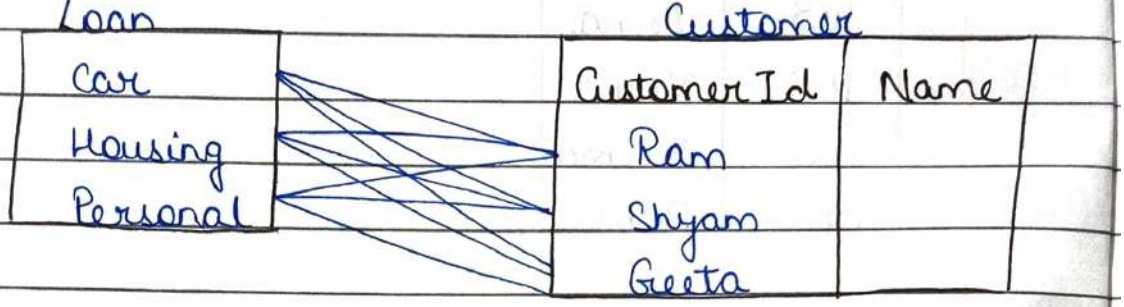
Table : Student

Ad. No	Name	Class
104	Seeta	12
105	Reeta	11
106	Shyam	12

Output

Ad No.	Name	Class
101	Ram	12
102	Shyam	12
103	Seeta	12
104	Reeta	11

★ 4- Cartesian product –



★ SQL :- SQL is structured query language. SQL is standard language for accessing and manipulating database. It is non-procedural in nature. The SQL standard specifies data definition, data manipulation and other associated facility of a DBMS that support the relational data model.

- SQL is a comprehensive language for controling and interacting with database mgmt. system.
- SQL command are used to implement the following –

1- SQL can retrieve data from a database.
2- SQL can insert record in database.

3. SQL can update, delete and create record in database.
4- It can create new tables and views in a database.

*1- Create table command - Create table command is used to create table structure. In this command we need to give information about tables such as- number of columns, types of each column and constraints.
Create table command requires name of the table, name of the field, definition and constraints for each field.

Constraints - In SQL we have following constraints-

1. NOT-NULL → to check a column cannot store a null value.
2. Primary key → to check that a column have a unique identity which help to find a particular record in table.

Syntax:

CREATE TABLE < table name >
< Column name 1 > < datatype > [Size] [Constraints]
< Column name 2 > < datatype > [Size] [Constraints]

Ex- Create the following table -

Column Name	Datatype	Size	Constraints
Ad no	Numeric	3	Primary
Name	Varchar	20	Not-null
Class	Numeric	2	
Section	Char	1	
Fees	Numeric	10,2	

Command –

```
CREATE TABLE student
(Ad no Numeric (3) Primary key,
 Name Varchar (20) Not null,
 Class Numeric (2),
 Section Char (1),
 Fees Numeric (10,2));
```

2- INSERT Into Command –

Syntax:

```
INSERT INTO<table name>[Column name 1,
                        Column name n...]
```

or

```
INSERT INTO<table name> Values (Value 1, Value 2
                        .... Value n)
```

Insert the following information in table –

Adno.	Name	Class	Section	Fees
111	Anu Jain	12	A	2500
222	Mohit Sharma	11	B	4500
3	K.P. Gupta	12	B	3000
4	Ajit Kumar	10	A	2000
5	Nandini	12	C	3000
6	Rohan Sharma	11	B	2500

INSERT INTO Student VALUES (1, "Anu Jain", "12", "A", "2500");
INSERT INTO Student VALUES (2, "Mohit Sharma", "11", "B", "4500");

3- SELECT Command -

SELECT (*/field list)
FORM < table name >
[Where < condition >]

→ Example- 1) Display student table information

SELECT *
FORM student;

This command is used to view table information from SQL database. By using SELECT command we can get one or more field information while using *, we can get all field information.
We can specify any condition using where clause. Where is optional. ~~This will~~

→ This will display all information of the particular table (student) in the database.

Example- 2.) To display name and class information of a student table -

SELECT name, class
FORM student;

Ex- 3.) To display name of 10th class student table information -

SELECT name
FORM student
where class = 10;

* Operators used in SQL command :-

1- **Arithmatic operator** — Arithmatic operator takes two command and perform a mathematical calculation on them. However they can be used only in select command. The arithmatic operator used in SQL are addition, subtraction, multiplication, division.

Example-

1.) Table : Name

First Name	Second Name
Anu	Jain
Madhu	Bhattja

Display first name with second

```
SELECT First name + Second name
  FROM Name;
```

Output :-

First name + Second name
Anu Jain
Madhu Bhattia

2.) Salary

```
SELECT Basic +DA
  FROM Salary
```

Basic	DA	Basic + DA
25000	5000	30,000
35000	7000	42,000

3.)

```
SELECT DA-100
  FROM Salary
```

DA - 100
4900
6900

4) SELECT DA/100
FROM Salary

DA/100
50
70

5) SELECT DA*100

DA*100
500000
700000

2- Relational Operator :- Relational operators are used to implement comparision between two operands. These operands can be used only in where clause.

Less than	<
Greater than	>
Less than equal to	≤
Greater than equal to	≥
Equal to	=
Not equal to	≠

1) Display student 'name' who are paying below 3000 Fees

```
SELECT name
FROM student
WHERE fees < 3000;
```

2) Display student name who are paying above equal to 3000 fees.

```
SELECT name
FROM Student
WHERE fees >= 3000;
```

Output :-

Name
Mohit Sharma
Nandini

3) Display student information who are not in class 10.

```
SELECT *
FROM student
WHERE class! = 10;
```

class
11
12
11
12

3- Logical operator :- Logical operators are also possible only in WHERE clause and are used to merge more than one condition.

AND

OR

NOT

Ex- 1.) Display information of student in class 11 B.

```
SELECT *
FROM student
WHERE class = "11" and section = "B";
```

AdNo	Name	class	Section	Fees
2	Mohit Sharma	11	B	4500
6	Rohan Sharma	11	B	2500

2.) Display 11th and 12th class student information

```
SELECT *
FROM student
WHERE class = '11' and class = '12';
```

Ad no.	Name	Class	Section	Fees
1	Anu Jain	12	A	2500
2	Mohit Sharma	11	B	4500
3	K.P. Gupta	12	B	3000
5	Nandini	12	C	3000
6	Rohan Sharma	11	B	2500

3) Display student information, who are not in class 10th.

```
SELECT *
FROM Student
WHERE NOT Class = 10;
```

Adno	Name	class	Section	Fees
1	Anuj	12	A	2500
2	Mohit	11	B	4500
3	K.P.	12	B	3000
5	Nandini	12	C	3000
6	Rohan	11	B	2500

4.) In operator specify multiple values in a where clause
For ex
Display students information who are in section A and B.

```
SELECT *
FROM Student
where class IN ("A", "B");
```

4= BETWEEN Operator - BETWEEN operator is used to test whether or not a value (stated before the keyword BETWEEN) is between the two values.

Ex- Display student information who are paying fees between 2500, 3500

```
SELECT *
FROM Student
WHERE fees BETWEEN 2500 and 3500;
```

* ORDER BY Command- This command is used to arrange values in ascending and descending order.

Ex-(i) SELECT *
FROM Student
ORDER BY fees ASC

(ii) SELECT *
FROM Student
ORDER By fees DESC

* Aggregate function:- Aggregate functions are used to implement calculations ~~for~~ based upon a particular column.
These function always returns a single value.
Sum(), COUNT(), AVG(), MIN(), MAX()

* PL/SQL:- PL/SQL is a procedural extension of Oracle SQL that offer language construct similar to those in programming languages. PL/SQL provides user and designer to develop complex database applications that require the uses of control structure ~~and~~ such as procedural functions and modules.

→ The basic construct in PL/SQL is block. Block allow designer to combine logically related SQL unit. In block, constants and variables can be declare, and variables can be used to store query results.

→ Statements in a PL/SQL block include SQL statement, control structures (loops), conditional statements (if, else), exceptional handling and call of the other PL/SQL block.

→ PL/SQL blocks specify procedures and functions can be group into packages. A package is similar to module and has an interface and implementation part. Oracle offered several predefined packages, For ex – Input-Output routines, file handling, job scheduling etc.

→

Advantage of PL/SQL –

1. Support to SQL
2. Block structures (functions & procedures).
3. Highly productivity.
4. Better performance (an SQL executes 1 stunt at a time, it can handle an entire block thus reduce comm^n between SQL and server).
5. Control structures
6. Modularity
7. Portability

PL/SQL Block structure :–

(1) Declare – Contain all variable constant cursor and user-defined exception that will be referenced within executable section.

(2) Executable Section – Contain SQL statements to

manipulate data in the database and PL/SQL statement to manipulate data in the block.

(3) Exception handling section – Specifies the action to perform when error and abnormal condition arise within the executable section.

(4) Sub-program – These are named PL/SQL block. They may be declared as procedural functions or packages.

★ Data-types –

Binary - Integer	CHAR	TABLE
DEC	CHARACTER	
DECIMAL	LONG	
FLOAT	RAW	
INT	Raw ID	
INTEGER	STRING	
NUMBER	VARCHAR	
NUMERIC	DATA	
POSITIVE	BOOLEAN	
REAL	RECORD	

Declaring a variable –

SYNTAX : Var Name Type(s) [:= Value];

★ Normalisation :-

Transaction & Concurrency Control

★ Transaction System :- Collection of operation that form a single logical unit of ~~one~~ work is called transac

Transaction is a unit of program execution that accesses and possibly update various data items. It also define as a logical unit ~~its~~ of data processing that include one or more data access operation.

Transaction access data using two operations -

1- Read
2- Write

★ Properties of transaction - Transaction have 4 basic properties which are called ACID properties. These properties are -

1- Atomicity - A transaction is an atomic unit of processing. It is either performed completely or not performed at all. The update made by the transaction will be accessible to other transac-tion.

2- Consistency - The consistency property of transaction is operation performed by a transaction should be isolated from other transaction. i.e. the execution of transaction should not be interferenced (overlaped) by any other transaction.

2. Consistency — The consistency property of transaction implies that if data was in consistant state before the initiation of transaction then at the end of transaction the data will also be in consistant state.

3. Isolation — The isolation property of transaction indicates that the step by step ~~oper transaction~~ operation performed by transaction should be isolated from other transaction.

4. Durability — The change is applied to the database by the committed transaction must persist in the database. These changes must not be lost because of failure.

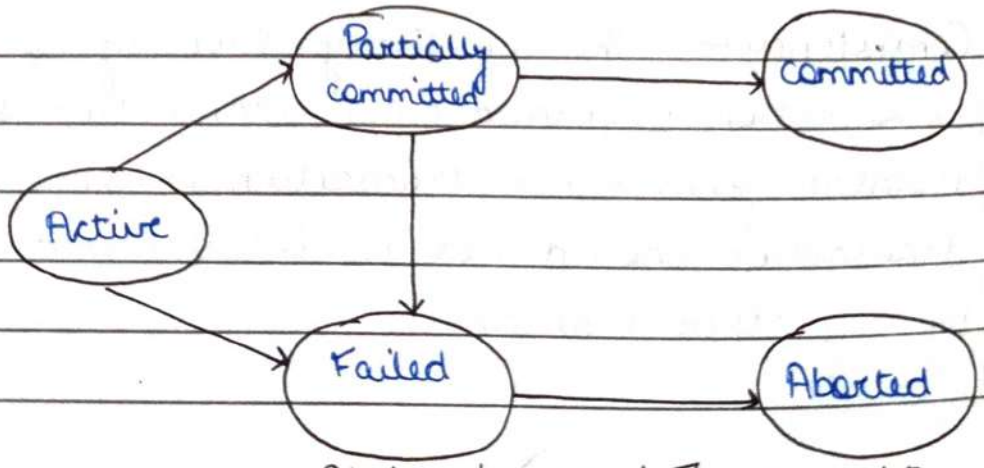
Example —

A transfer 50 Rs. ~~from~~ to B account.

A = 500 Rs
B = 300 Rs.

A(withdraw) = A − 50;
B (deposit) = B + 50;
A get balance = 450;
B get balance = 350;

* Transaction State :-



State diagram of Transaction state

1- Active state – It is the initial state the transaction stay in the state while it is executing.

2- Partially committed – After the final statement has been executed.

3- Aborted – After the transaction have been ~~overlapped~~ roll back and the database has been restored to its state prior to the state of the transaction.

4- Committed – A transaction that completes its execution successfully is said to be committed.

5- Failed – After the discovery that normal execution cannot no longer proceed.

* Recovery from Transaction failure :- The recovery process is an integral part of a database system, which is responsible for detection of failure and recovery of database.

1- Cascading roll back –

T_1	T_2	T_3
Read(X)		
Read(Y)		
Z = X + Y		
Write(Z)		
	Read(Z)	
	Write(Z)	Read(Z)

Transaction T_1 writes the value of Z that is read by transaction T_2. Transaction T_2 writes the value of z is read by transaction T_3.
If T_1 fails, T_1 must be roll back. T_2 is dependent on T_1, T_2 must be roll back. Since T_3 is dependent on T_2, T_3 must be roll back.

Log based recovery – To keep track of database transaction, the DBMS maintain special files called log files or generals that contain about all updates. The log files contains information like transaction identifier, type of log record.
Since the information contained in the log files are critical for database recovery, two seperate copies are maintained.
Since the log files are brought to failure the log files are periodicaly achieved and stored in offline storage.

★ Concurrency Control – Concurrency control is the management of concurrent transaction execution.
Concurrency control can be performed by various methods such as locking, time stamp, validation technique and optimistic method.
The objective of concurrency control is similar to the objective of multiuser computer system, where many user can perform different operation

at the same time.

1- Locking technique for concurrency control —

T_1	T_2	T_3
Read(X) } lock x		
Read(Y) } lock y		
Z = X + Y	lock(Z)	
Write(Z)	Read(Z) } lock z	
	Write(Z) } lock z	Read(Z) } lock(z)

Locking is a procedure used to control concurrent access to data.

When one transaction is accessing the data, a log may be deny access to the ~~event~~ other transaction to prevent in current result.

Locking is one of the most widely used mechanism for serializability. The basic rule for locking are

(i) If the transaction has a read lock on the data item, it can read the item but not update it.

(ii) If a transaction has a read lock on the data item, the other transaction can be obtain read lock on a data item but not write lock

(iii) If a transaction has a write lock it can read and update both.

(iv) If a transaction has write lock then the other transaction cannot obtain either a read lock or write lock.

* Two phase locking protocol — Two phase locking protocol ensures serializability. The protocol

requires that each transaction issues lock and unlock request in two phase –

(i) Growing phase
(ii) Shrinking phase

1- Growing phase – In this phase a transaction may obtain lock but not release any lock. Here the number of lock increases from 0 to the maximum for the transaction.

2- Shrinking phase – In this phase a transaction may release lock but may not obtain any new lock. Here the number of lock decreases from maximum to 0.

There are 3 types of two phase locking protocol

- Conservative two phase locking protocol – It requires a transaction to pre-declare all the locks. It requires before its execution.

→ If any pre-declared item cannot be locked the transaction wait until all items are lock.

→ The locking protocol is deadlock free but it is difficult to use in practise.

- Strict two phase locking –

In this protocol a transaction does not release any of its write lock until it commit or abort.

- Rigrous two phase locking — In this protocol a transaction does not release any of write lock and read lock until it commit or abort.

* Time stamp protocol for concurrency control :-
Time stamping is a concurrency control protocol in which the fundamental goal is to order transaction globally in such a way that older transaction get priority in the event of conflict.
In this time stamp ordering method, if a transaction attempt read or write data item, then the read or write operation is allowed only if the last update on the data item was carried out by an older transaction.
Otherwise, the transaction requesting the read-write is restarted and give a new time stamp. The restarting transaction is assigned a new time stamp to prevent it continuously aborting and restarting.

Example-

T_1 (10)	T_2 (20)	T_3 (30)	A (R_TS=0)	B (R-TS=0)
Read A			W-TS=0	W-TS=0
A = A - 100			R-TS=0	
Write A				
Read B	Read A		W-TS=10	
B = B + 100	A = A - 250		R-TS=20	R-TS=10
Write B	Write A			
	Read B	Read A	W-TS=20	W-TS=10
	B = B + 250		R-TS=30	
	Write B	Read B		W-TS=20
		Display A+B		R-TS=30

* Validation based protocol :- In validation based technique we do not update all the data items directly to the database rather it keeps ~~rather~~ local copies of all updates during transaction execution and as the transaction reaches its end.

A validation phase check the consistency of data item due to the updates, and if it is consistant then all data from the local copies goes to the database.

Reading is done when required and if at the end of transaction any data item is found to be inconsistant then the transaction is roll-back and restarted again.

There are 3 phases of validation based concurrency control protocol -

1- Read phase - Here the transaction is activated and read the last committed value from the database and put these value in local variables. All the updates are only in local copies of the database.

2- Validation phase - In this phase checking is performed for the read value during the first phase against the current values. Also it check if the data items are the modified data should remain consistant.

3- Write phase — If the transaction are validated and the data is in consistant mode then update made by the transaction are applied to the database permanently.
If the data are not validated or inconsistent phase it will violate the serializability so it discard a roll-back and all the updates and transaction is restarted.

UNIT - 6
Normalization

* Normalization — Database normalization is a technique to organise the content of the table for transactional database.
This includes creating table, establishing relationship between those tables and make table flexible by eliminating two factors —
1- Redundancy
2- Inconsistant dependency
Normalization is part of successful database design, database system may be inaccurate, slow and inefficient.
When we normalize the database we have four goals —
(i) Arranging the data into logical grouping such that each group describe a small part of the whole.
(ii) Minimizing the amount of duplicate data, called redundancy stored in database.
(iii) Organise the data such that, when you modify it.

you make the change only in one place.

(iv) Without compromising the integrity of data in storage we can manipulate it.

The main advantage of normalization are –

1- It frees the database from certain add, edit and delete anomlies.

2- It reduces the need of restructure of data as new kind of data are introduced.

3- It makes the data more informative to user, including different users making different queries.

★ Functional dependency –

$$t_1(x) = t_2(x)$$

An attribute in a relational model table is said to be functionaly dependent on another attribute in the table. If it can take one value for a given value of attribute upon which it is functionally dependend. It is denoted by $A \rightarrow B$ (A implies B).

$A \rightarrow B$, $ABCD \rightarrow E$ and $EF \rightarrow G$

★ First Normal Form :-

END

Hope the study material was helpful , to stay connected with us : Visit Us

www.dreamstudy.tk

https://www.facebook.com/ allbcaweb
https://www.facebook.com/dreamstudy

https://www.instagram.com/ allbcaweb

https://www.youtube.com/channel/UC8l0Dfy2YekfiigaXEtbAow or
search on youtube with - dreamstudy website

https://twitter.com/ allbcaweb