Database Management System

EG2201CT

Fourth Semester

Prepared by Santoshi kumari Pandit

Date 2080-06-15
Page

1.) Write a program to find the entered number in an array.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int ary[10], i, n, a;
printf("Enter total no. of elements");
scanf("%d", &n);
printf("\n Enter %d elements ", n);
for(i=0; i<n; i++)
{
scanf("%d", &ary[i]);
}
printf("\n Enter a no. to be searched");
Scanf("%d", &a);
for(i=0; i<n; i++)
{
if (a==ary[i])
{
printf("found");
}
else
{
printf("Not found");
}
}
}
```

Page

2.) Write a program to display the contents of an array in reverse order.

=>
```
#include <stdio.h>
#include <conio.h>
void main()
{
int ary[5], i, a, n;
printf("Enter total no. of element");
scanf("%d", &n);
printf("\n Enter %d elements", n);
for(i=0; i<n; i++);
{
scanf("%d", &ary[i]);
}
printf("\n Enter a no. to be searched"
for(i=n-1; i>=0; i--)
{
    printf("%d", ary[i]);
}
```

0	5
1	16
2	15
3	6
4	8

Date
Page

3.) Write a program to find total occurance of an entered number in an Array.

```
#include <stdio.h>
#include <conio.h>
void main()
{
int a[5], i, n, count=0;
printf("Enter the total no of elements");
scanf("%d", &n);
printf("\n Enter %d element", n);
for(i=0; i<5; i++)
{
scanf("%d", &a[i]);
}
printf("Enter the number of which we have to
        check occurance\n");
Scanf("%d", &n);
For(i=0; i<5; i++)
{
if(a[i]==n)
{
count++;
}
printf("The total no. of occurance is %d", count);
getch();
}
```

WAP to accept any no. from user and display no. is even/odd.

```
#include <stdio.h>
#include <conio.h>
void main()
{
int a;
printf("Enter any number\n");
scanf("%d", &n);
if(a%2==0)
{
printf("The given number is even");
}
else
{
printf("The given number is odd");
}
getch();
}
```

Q.No.2 WAP to accept any three no from user and display greater number.

```
#include<stdio.h>
#include<conio.h>
 void main()
 {
 int a, b, c;
printf("Enter any three number\n");
scanf("%d %d %d", &a, &b, &c);
if(a>b && a>c)
{
printf("%d is the greater number", a);
}
else if(b>a && b>c)
{
printf("%d is the greater number", b);
}
else
{
printf("%d is the greater number", c);
}
getch();
}
```

Date
Page

Q. NO.3 WAP to accept any no from user and find factorial value.

```
#include <stdio.h>
#include <conio.h>
  void main()
  {
int n;
printf("
```

DBMS

Date 2080-06-10
Page

V.V.I

Introduction to DBMS

* Data :- All the raw facts and figures which are collected to provide a meaningful collection is known as data.
for example :- Roll.no, Name, Age etc of a student are refered as data.

* Information :- The final result of collected data after processing is refered as after information. which helps to increase the knowledge of the person who uses it.

* Data Base :- A Database is well organized collection of data stored in standarized format which is designed to be shared by multiple users and can be ~~used by~~ easily accessed.

* DBMS :- ~~Bababa~~ Database management system is a slw that enables users to create and maintain a database.
DBMS defines a database, Stores the data supports a query language and produces reports

Date
Page

1.1 * Evolution of DataBase

(i.) Traditional DataBase :- In Traditional file oriented approach all records are stored in a particular single file. Such type of record keeping system is called flat file system. In early data processing system the following this advantages can occur :

a.) Data Redundancy (The same piece of information may be stored in two or more files.)

b.) Lack of flexibility (Information retrival possibilities would be limited).

(ii.) personal Computer DataBase :- The database which is itself server and client, is refered as personal computer database or, desktop database. This type of database doesn't ~~working~~ run in network environment. MsAccess is the example of such database which is suitable for single user software development.

(iii.) Client Server DataBase :- if a server serves the data according to request of client, such type of database is refered as client server database. All the facilities are provided by server and it maintence the

different level of security. SQL server is the example of client server database.

(iv.) Distributed DataBase :- The database which is distributed among different computer is known as distributed database. oracle 8i is the example of Distributed DataBase.

(v.) Network DataBase :- DataBase that supports different level of networking and sharing of data among different computer, such database is refered as Network DataBase. SQL server is the example of such database.

(vi.) Relational DataBase :- DataBase that supports different type of relationship, such data base is refered as relational database. oracle, SQL server are the example of such database.

Date 2080-06-17
Page

1.2 * Characteristics of the database.

In the database approach data is maintained in a systematic way that is defined once and then accessed by various users.

The main characteristics of database approach verses the file processing are the following:

- Self describing nature of database system.
- Insulation between programs and data.
- Support of multiple views of data.
- sharing of data and multiuser transation process.

* Advantages of DBMS.

DBMS are useful for:

i.) Controlling Redundancy (storing the same data multiple times).
ii.) Restricting unauthorized accessed.
iii.) Storing large amount of data.
iv.) providing storage structures for efficient query process.
v.) providing back up and recovery facilities.
vi.) providing multiple user interfaces.
vii.) Representing and managing complex relationship amount data.
viii.) Enforcing Integrity Constraints.
ix.) Reducing application development time.
x.) providing flexibility and availability of up-to-date information.

Date
Page

* Disadvantages of DBMS.

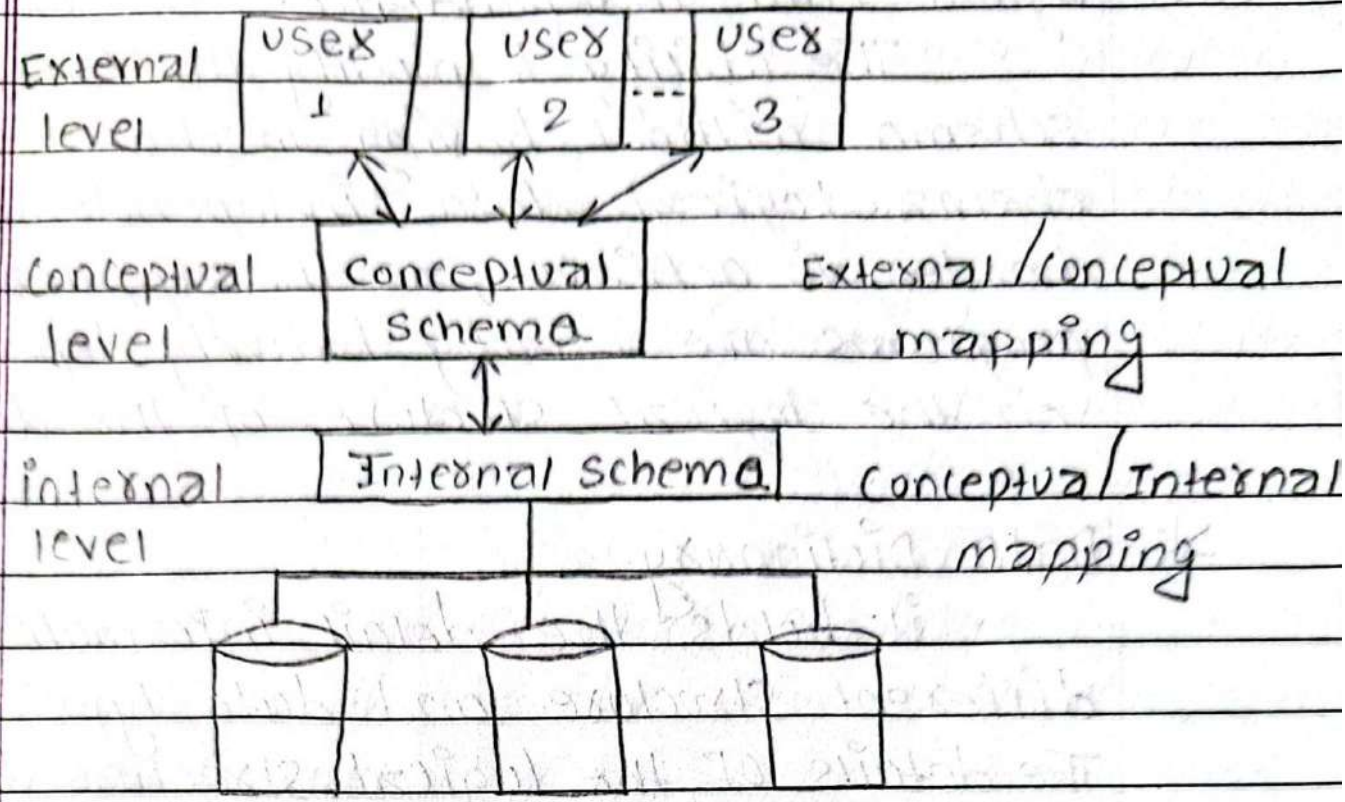
i) High initial investment in hardware, software and training.
ii) Qualified professionals should be required.
iii) Uses alot of memory to install program.
iv) Chances of breaking of security and crashing the database.
v) Data is abstract from the user.

1.3 * DBMS Architecture.

A commonly used view of database approach is the three-level architecture suggested by ANSI/SPARC (American National Standard Institute / standard planning and Requirement Committee). The reports proposed an architectural frame work for database. Under this approach the three level of the architecture are three different views of the data:

i) External Level (Individual user view).
ii) Conceptual Level (community user view).
iii) Internal Level (physical or storage view).

Date 06-18
Page



1.4* Data Independence

Data independence can be defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level. There are two types of data independence:

(i) physical data independence:

The ability to modify the physical schema without causing application program to be re-written. Modifications at this level are usually to improve performance. A physical

(ii.) Logical Data Independence.

The ability to modify the conceptual schema without having to change external schema. Logical data independence is harder to achieve, as the application programs are usually heavily dependent on the logical structure of the data.

* Data Dictionary.

It holds the detail information about different structure and data types such as: The details of the logical structure that are mapped into different structures, details of relationship between data item, details of all users privileges and access right, performance of resources with details.

1.4 * Data Abstraction

The major purpose of a database system is to provide users an abstract view of the system. The system hides certain details of how data is stored, created and maintained. Complexity should be hidden from database users.

There are several levels of abstraction:

(i) physical level

An physical level it is defined that how the data are stored.

Date
Page

hashing etc. it is lowest level of abstraction.

(ii.) Conceptual level

it is next highest level of abstraction it describes what data are stored and the relationship among them. it is database administrator level.

(iii.) View level

it is highest level of abstraction. it describes part of the data base for particular group of users. for e.g. ~~tailors~~ tellers in a bank get a view of customer accounts but not a payroll data.

* DataBase Administrator (DBA)

The database must be able to meet the demands of various users effectively, it is possible only if it is maintained and managed properly. A person who is responsible for administring all resources of a database is called DBA. ~~The~~

The DBA is responsible for :

(i.) Installing and operating software and hardwares as required.

(ii.) Authorizing access to the database.

(iii.) Co-ordaniting and monitoring the actual use of database.

(iv.) Maintaining the security of database.
(v.) providing proper back up and recovery.

06-19

1.6 * Classification of DBMS.

DBMS can be classified categorized into Centralized and Distributed Database.

	Centralized Database		Distributed Database
i.	A Centralized Database has all it's data store a single CPU in a single physical location.	i.	A distributed database is a database i.e. under the control of a central DBMS in which storage devices are not all attached to a common CPU. It may be stored in multiple computers located in different location.
ii	Database are easier to maintained, up-to-date than Distributed.	ii.	If data is lossed, retriving is easier because there is always a copy of data in a different location.
iii	If data is lossed, Retriving it would be much harder.	iii	Distributed Database systems are based on hierarchical structure.
iv	Designing a centralized database is generally much less complex.		

Unit-2
Data Models

Date
Page

1 * Data Models:

DBMS can also be classified on the basis of database.

Data Model are a collection of conceptual tools for describing data, there relationship, Semantic and constraints. Simply we can say that data models defines how data is connected to each other and how they are process and stored inside the system.

There are three different loops:

(i.) physical Data Models
(ii.) Record-Based Data Models
(iii.) object-Based Data Models

Record-Based Data Models
→ The Network Model
→ The hexarchical Model
→ The Relational Model

Object-Based Data Models
→ The object oriented Model
→ The ER Model

(i.) physical Data Models :- physical Data Models describe how data is stored in the computer, representing information, record ordering and access paths.

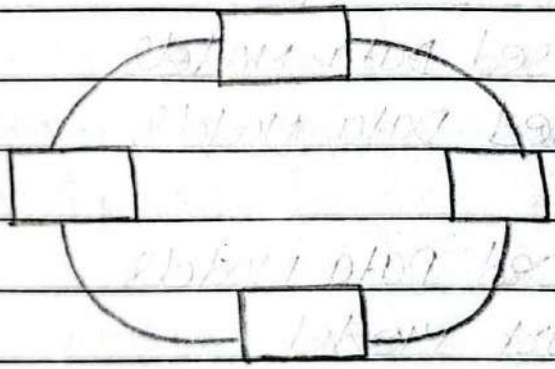
Date
Page

(ii.) Record-Base Data Models

In this Model the database is structured in fixed formal records of several types. They are used in describing data and the logical and view levels and use to specify the over all logical structure of the data base.

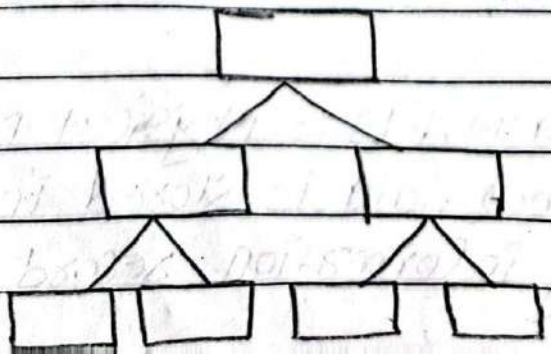
→ Network Model

In the Network Model data are interconnected to each other for the purpose of sharing the data in the database.



→ Hierarchical Model

In this Model data are organized as a tree structure in which parent and child relationship is created.



Date
Page

-> Relational Model

The database model which supports different levels of relationship among data such as one-to-one, one-to-many and many-to-many is called relational model

i.) one-to-one (1:1)

The relationship in which one entity is associated with another one entity is know as one-to-one relationship.

ii.) One-to-Many (1:M)

If one entity is associated with more than one entities such type of relationship is called one-to-many relationship.

iii.) Many-to-Many (N:M)

If many entities are associated with many other entities such type of relationship is called Many-to-Many relationship.

06-21

(iii) DBject Base Data Model

The object base models uses the concept of entities (objects) and relationships among them. object based logical models provide flexible structuring capabilities and allow data constraints to specified explicitly.

Date
Page

-> Object oriented model

The object oriented data model is an adaptation of the object oriented programming language concept to database system. Some concept and features should be followed to be an object oriented approach

- Encapsulation :- Encapsulation is the packing of data and function into a single component. it also protect data from accidental ~~Cooption~~. Corruption.

- polymorphism :- if an object can perform different hetrogenius implementation or operations, such characteristics is known as polymorphism.

- Data Abstraction :- it describes the behaviour of the data but not the functionality. Simply, we can say that only effect can be seen.

- Inheritance :- Inheritance is the major property of an object oriented programing in which child class adopts the propery of parent class.

Date
Page

→ E-R Model

The Entity Relationship model is based on a collection of basic objects (entities) and relationships among these objects.

A relationship is an association among several entities.

* Entity :- An entity is a person, place or thing in this world, that physically exist and distinguishable from other entities. book, student, company, plant etc are example of entity.

* ~~Entity set :-~~

2.3 * Entity set :- An entity set is a set of entities of the same time that share the same properti for example, students, students, companies, plants

2.4 * Attribute :- An attribute is a property of an entity. An entity is represented by a set of attributes that describe the properties of all members of an entity set. Example,

V.V.I

Student = Roll no, Name, Address, phone no

↓ entity

Attribute

Date
Page

Types of Attribute

- Simple Attribute

The value that cannot be devided further is known as simple attribute.
for example :- phone.no, Roll.no etc.

- Composite Attribute

The attribute that can be devided into more than one called simple attribute.
for example :- Name may have first name and last name.

- Derived Attribute

The attributes whose values are derived from other attributes.
for example :- average salary, Age (can be derived from DOB).

- Single value Attribute

The attribute that contain single value. for example :- citizeship.no, Registration no, roll.no etc.

- Multi Value Attribute.

The attribute that may contain more than one value. for example :- A person can have more than one mobile number.

Date 06-22
Page

2.5 * Relationship :-

Relationship is an association between two or more entities. for example :- An employ works in a department or A student enrolls in a faculty. Here, works and enrolls define the relationship.

2.5 * Relationship set :-

A relationship set is a group of similar kind of relationship among one or more entities.

* Roles of Relationship in E-R Model.

The main role of Relationship in E-R Model is to define link between the entities which can be represented by diamond symbol.

2.6 * E-R Diagram.

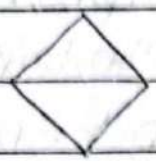
An E-R Diagram is a visual representation of data that describes how data is related to each other.

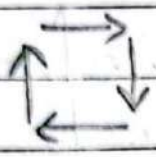
E-R Diagram uses some symbols to represent the types of information. They are given below.

 ⟹ Entity (place, person, thing)

Date
Page

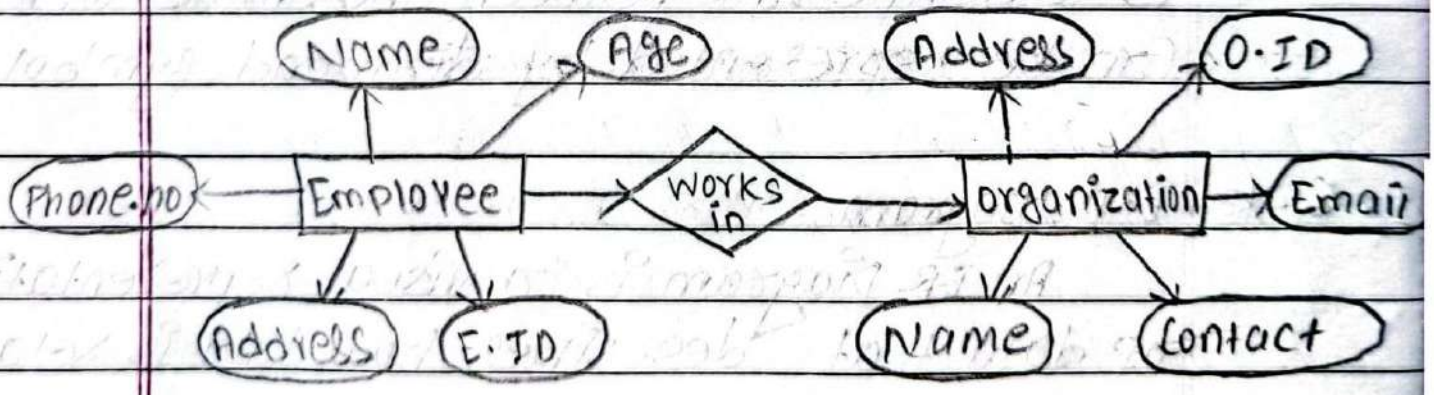
 ⟹ Attribute (properties of the entities)

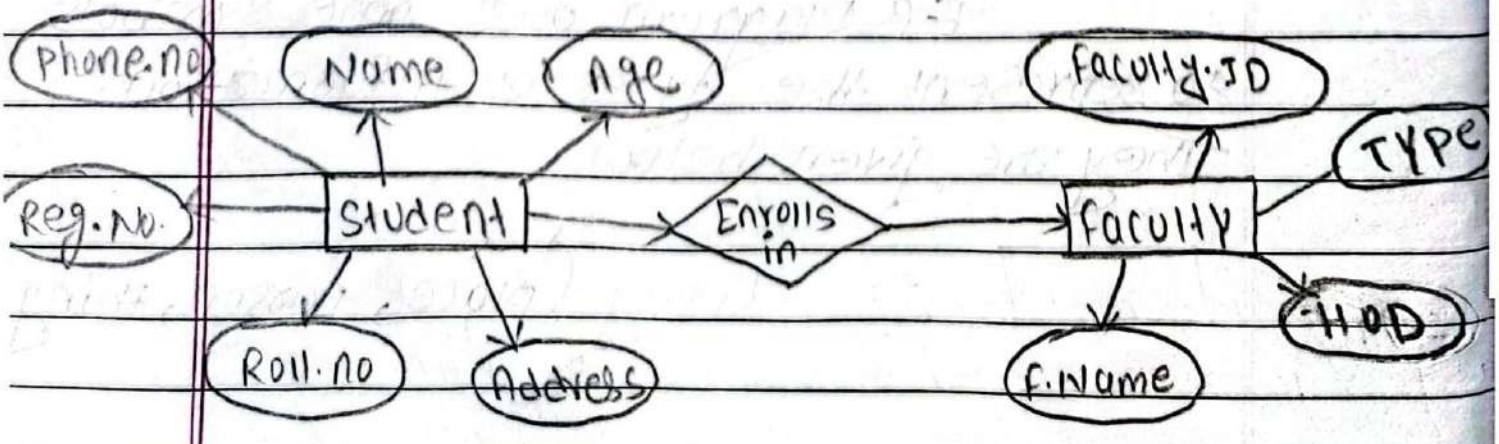
 ⟹ Relationship

 ⟹ flowline

(i.) 1-1 (one to one notation)
(ii.) 1-∞ (one to many notation)
(iii.) ∞-∞ (many to many notation)

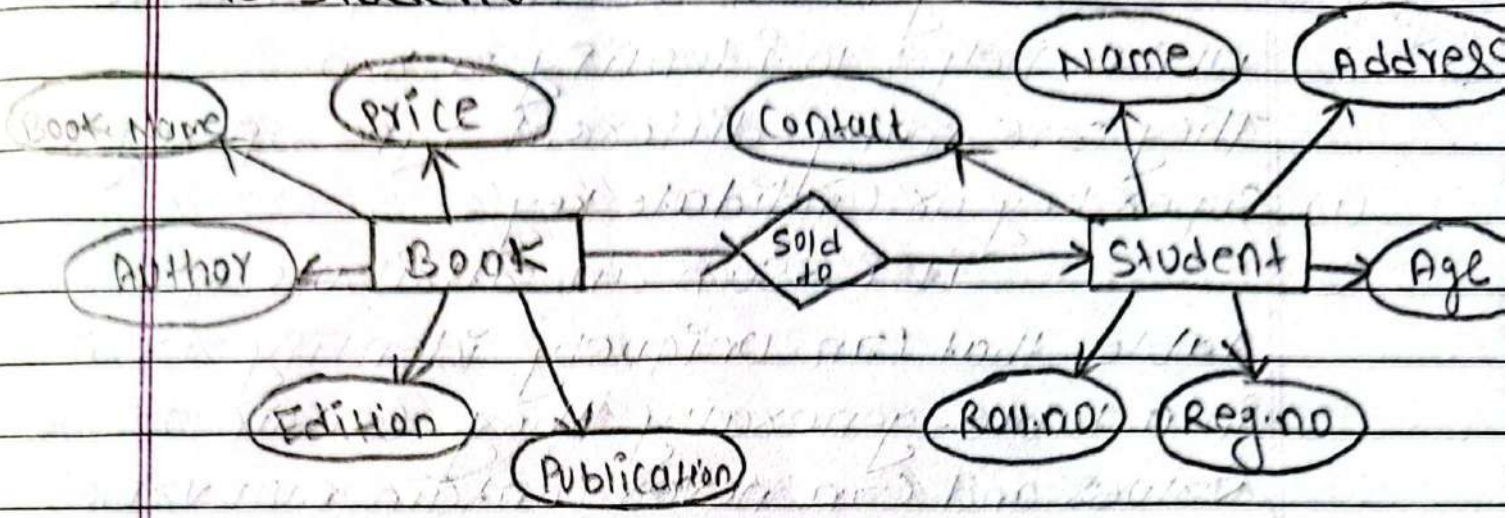
→ Some examples of E-R Diagram.



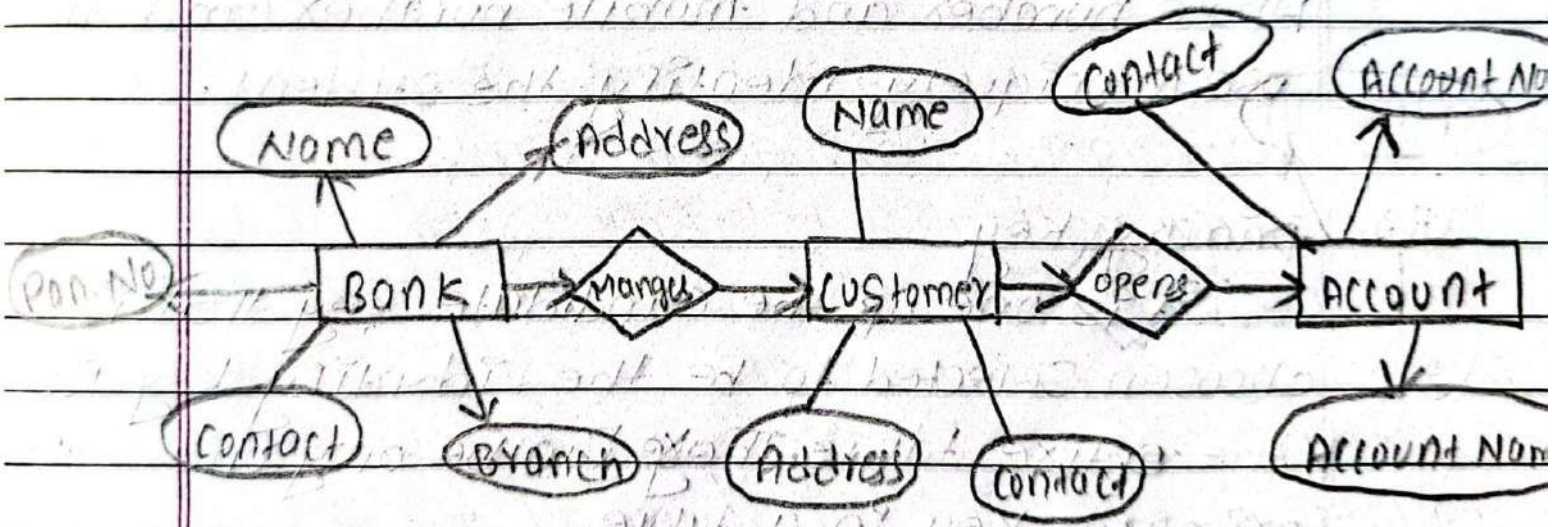


Date 06-03
Page

1.) Draw an E-R Diagram in which book solds to student



2.) Draw an E-R Diagram for banking Management:



Date 06-25
Page

2.4 The Concept of Keys

A key is an attribute of a table which helps to identify a row.
They are many different type of keys:

(i.) Super key or Candidate key

It is such an attribute of a table that can uniquely identify a row in a table. generally they contain unique values and can never contain null values. They can be more than one candidate key in a table. for example, in a student table ~~Whole~~ roll numbers and mobile numbers can both use to uniquely identify the student.

(ii.) primary key

It is one of the candidate key that is choosen selected to be the identify key for the entire table. There can be only one primary key in a table.

(iii.) Alternate key

This is the Candidate key which is not choosen as the primary key of the table.

(iv.) foreign key

Sometimes we may have to work with an attribute that doesnot have a primary key of its own. to identify its rows, we have to use

the primary attribute of a related table. Such a copy of another related table's primary key is ~~for~~ called foreign key.

(v.) Composite key

Sometimes one attribute is not enough to uniquely identify a row. So a combination of two or more attribute is used to create a unique combination of values, such key is known as composite key. for example, class 6 and Roll no 5 i.e. class 5 and Roll no 1 i.e (class + Roll)

* Strong and weak entities

V.V.I

Based on the concept of foreign key, there may arise a situation when we have to relate an entity having a primary key of its own and an entity not having a primary key of its own. In such case the entity having its own primary key is called a strong entity and the entity not having it's own primary key is called weak entity.

Date 06-26
Page

* Instances and Schemas

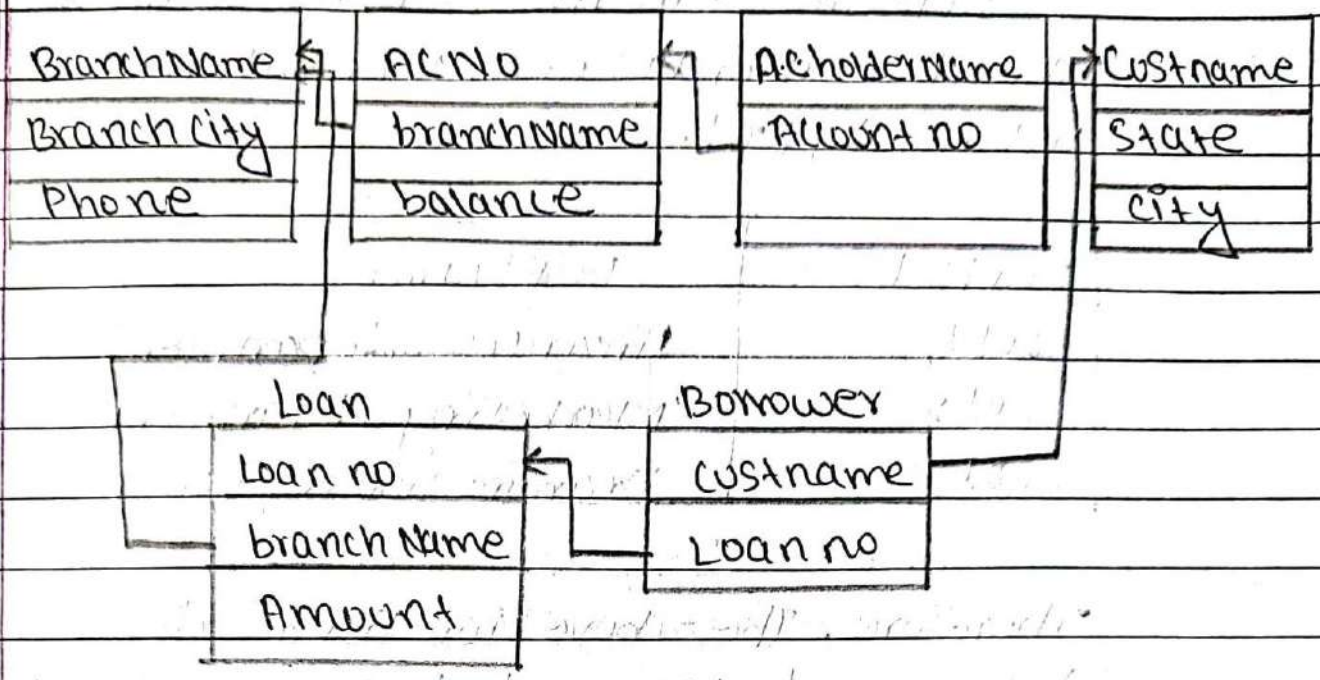
Databases change overtime as information is inserted modify and deleted. The collection of information stored in the database at a particular moment is called an "instances" of the database.

The over all design of the database is called the database schema. The schema describes the logical structure of the database.

A database schema, along with primary key and foreign key dependence is, can be represented pictorially by schema diagrams. The figure given below shows the schema diagram of a banking system. Each relation (Table) appears as a box, with the attributes listed inside it and the relation name above it. Foreign key dependence is appear as arrows from the foreign key attribute of the refrencing relation to the primary key of the referenced relation.

Date
Page

Branch Account depositor Customer



06-29

* Functional Dependency

Functional dependency in DBMS is a relationship between attributes of a table dependent on eachother. it helps in preventing data Redudency for example let us consider a table 'T' with attribute A and B. functional dependency is represented by (→) arrow sign. Then the following will represent the functional dependency between these attributes: A → B.

where B is functionally dependent on A.

A → is determinant set.

B → is dependent attribute.

example; consider we have a table department with two attributes. The deptId is our primary

Date
Page

key, which uniquely identified the ~~depart~~ deptName attribute. If we want to know the departMentName, then at first we need to have deptId.

DeptId	DeptName
0011	finance department
0012	Marketing department
0013	Examination department

Therefore, The above functional dependency between deptID and deptName can be determined as deptName is functionally depndent on deptId.

DeptId ⟶ DeptName

Date 06-30
Page

Relational DataBase

A Relational DataBase Consist of a collection of tables, Each of which have a unique name. A row in a table represent a relationship among a set of values. A table is an entity set and a row is an entity. for example, Considered an account table given below.

Account :-

Account No.	Branch Name	Balance
A101	Birgunj	5000
A102	pokhara	10000
A103	kathmandu	20000
A104	Butwal	2000

- it has three attributes (columns).
- it has four ~~Toppe~~ Tuples (row).
- for each attribute they release a permitted a set of values, called the domain of that attribute.

4.7 Relational Algebra

Relational Algebra is a procedural query language, which takes instances for or relations as Input and gives instances of relations as output. it uses some uniary and Binary operators to perform queries.

The fundamental operations of relational algebra are as follows:

1. Selection
2. projection
3. Union
4. Set Difference
5. Cartesian product
6. ReName

1. Selection (select operation) (6)

it selects tuples that satisfy the given predicate (condition) from a relation.

Notation:

$\sigma_{P}(r)$

where, σ stands for selection predicate and r stands for relation. predicate is a logic formla which may use connectors like and, or, not.

These terms may use relational operators like $=, \neq, <, >, <=, >=$.

for example:-

(i) $\sigma_{subject = "database"}(books)$

output :- it select tuples from books table where subject is database.

(ii) $\sigma_{subject = "database" \text{ and } price = 450}(books)$

output :- it select tuples from books table where subject is database and price is 450.

2080-07-12

2.) projection (Π)

it projects columns that satisfied a given predicate.

Notation:

$\Pi_{A_1, A_2, \ldots, A_n}(R)$

where, $A_1, A_2, \ldots A_n$ are the attribut names of relation R.

Duplicate rows are automatically eliminated.

Example :-

(i) $\Pi_{subject, Author}(Books)$

output :- it select and project columns named as subject and Author from the relation books.

(3.) Union (∪)

it performs binary union between two given relations.

Notation:

$r_1 \cup r_2$

where, r_1 and r_2 are relations in a database.

for example:-

(i.) $\Pi_{Author}(Books) \cup \Pi_{Author}(Articles)$

output :- it projects the name of Author who have either written a book or an article or both.

4.) Set Difference (−)

The result of set difference operation are the tupples, which are present in one relation but ~~are~~ are not in the second relation.

Notation:

$r_1 - r_2$

finds all the tupples that are present in r_1 but not in r_2.

(i.) for Example:-
$\Pi_{Author}(Books) - \Pi_{Author}(Articles)$.
output:- it provides the name of Author who have written books but not Articles

5.) Cartesian product (x)
it combines information of two difference relation into one.

Notation:
$\gamma_1 \times \gamma_2$

(i.) for example:
$\Pi_{Author = "James"}(Books \times Articles)$
output:- it forms a table which shows all the books and Articles written by James.

6.) Rename (P)
The results of relational algebra expressio donot have a name that we use to refer the The rename operation is use to give them a name.

Notation:
$P_x(E)$
where, the result of expression (E) is saved with the name of 'x'.

* Some additional Relational Algebra operations.

(1) The set intersection operation

The set intersection operator is denoted by (∩). Suppose that we have to find all the customers who have both a loan and account. using intersection we can write

$\Pi_{customer\ name}(borrower) \cap \Pi_{customer}(depositer)$

Result :- it displays all the customers name from borrower and depositer tables who have both a loan and account in a bank.

(2) The Natural join operation

The Natural join is a binary operation that allows to combine certain selections and a cartesian product into one operation. It is denoted by the join symbol (⋈). It forms a cartesian product of it's two arguements performs, a selection forcing equality on those attributes that appear in both relation schemas and finally removes duplicate attributes.

for example, find the names of all customers who have a loan at the bank and find the amount of the loan.

Date
Page

Customer name, Loan number, Amount (Borrower Loan)

1.) Outer join :-

The outer join operation is an extension of the natural join operation to deal with missing information.

Suppose there is two table that contain data of full time employees.

⇒ Employees

Empname	Address	Contact
Robert	Birgunj	9823326280
John	Kalaiya	9822334560
Smith	Kathmandu	9811223456
Williams	Hetharda	9822336400

⇒ Full Time Employees (FTEMP)

Empname	Department	Salary
Robert	Computer	20,000/-
John	Mechanical	30,000/-
Gates	Electrical	50,000/-
Williams	Civil	40,000/-

Now, we have to generate a single table with all the informations (Address, Contact, Department, Salary) about Fully Time employees

Date
Page

we can use natural join for this whose result is given below as:

$\Pi_{Empname, Address, Contact, Department, Salary}$ (Employees ⋈ Full Time Employees)

⇒ Result of Natural join is:

Empname	Address	Contact	Department	Salary
Robert	Birgung	9822326280	Computer	20000
John	Kalaiya	9822334560	Mechanical	30000
Williams	Hethauda	9822333640	Civil	40000

we can notice that we have lost the information about smith and bates.

To avoid this situation we ~~have~~ can use the outer join operation. There are three forms of this operation:

(i.) Left outer join (⟕)
(ii.) Right outer join (⟖)
(iii.) Full outer join (⟗)

Ans 14

(i) Left outer join (⟕)

It takes all the tuples in the left table, setting all other attributes with null value, in the right table which doesn't match.

Page

Empname	Address	Contact	Department	Salary
Robert	Birgunj	9822232628	Computer	20000
John	Kalaiya	9822334560	mechanical	30000
williams	Hethauda	9822333640	civil	40000
Smith	Kathmandu	9811223456	NULL	NULL

Result of Employee ⋈ FTEmp

ii.) Right outer join (⟖)

EmpName	Address	Contact	Department	Salary
Robert	Birgunj	9822232628	Computer	20000
John	kalaiya	9822334560	mechanical	30000
~~bottigates~~	NULL	NULL	Electrical	50000
Williams	Hethauda	9822336400	civil	40000

Result of Employee ⟖ FTEmp

iii.) Full outer join (⟗)
it takes all tuples from left and right relations.

Empname	Address	Contact	Department	Salary
Robert	Birgunj	9822232628	Computer	20000
John	kalaiya	9822334560	mechanical	30000
Williams	Hethauda	9822333640	civil	40000
Smith	Kathmandu	9811223456	NULL	NULL
Gates	NULL	NULL	Electrical	50000

Result of Employee ⟗ FTEmp

Date 2080-08-10
Page

SQL

Structure query Language is developed by IBM in 1970, which was originally called "Sequel". In 1986, the ANSI (American National Standard Institute) and the ISO (International Organization for Standardization) published the another version of SQL.

SQL is a "query language" which is used for querying a database. i.e retrivin data and information from the database. It can also be used for defining the structure of the data, modifying data and specifying security constraints in the database.

The main components of SQL are:

(i.) Data Definition language (DDL)
(ii.) Data Manupulation Language (DML)
(iii.) Data Control Language (DCL)

(i.) Data Definition Language (DDL)

It provides command for creating, Modifying and deleting tables.

Date 2080-08-10
Page

(ii.) Data Manupulation Language (DML)

it includes command to insert modify, delete and retrive data in the database

(iii) Data Control Language (DCL)

it authorizes users to access and manupulate data.

* Basic Domain Types (Data Types)

SQL supports a variety of Built in data types:

(i) char(n):-

A fixed length characteristic string with user specified length (n) (character) can also be used.

(ii) varchar(n):-

A variable length characteristic with user specified maximum length 'n'.

(iii) Int :-

An integer

(iv) Small int :-

small integer.

(v.) float(n):-

A floating point number with precision precision

or atleast 'n' digit.

(vi) Numeric (P, D) :-
A fixed point number with user specified precision. The number consist of 'p' digits and 'D' is the right of the decimal point.

(vii) Date :-
It stores year, month and day values.

(viii) Time :-
It stores hours, minutes and second value

(ix) Timestamp :-
It stores year, month, day and hour, minutes, second values.

(j) DDL
* Create :- This command is used for creating a table in a database

syntax
~~Create table tablename (~~
~~name attribute name char,~~
~~Roll no int,~~
Create table tablename (
Attribute1 datatype1,
attribute2 datatype2,

Date
Page

eg: Create a table of Employee

```
create table Employee (
eid int,
ename char(10),
salary float(8));
```

eg: create a table of student

```
create table Student
( SRollno int,
  SName char(20),
  SFaculty char(20),
  SDOB Date,
  SAddress char(20),
  primary key (RollNo));
```

Note :- for defining primary key attribute.

* Alter :- This command is used for altering (modifying) an existing table. it allows a new column to be added.

Syntax: Alter table existingtablename
Add Columnname datatype;

eg: To add phone.no in the employee table.

```
Alter table employee
Add phone_no char(10);
```

* Drop :- This command is used to delete table from the database.

Syntax :- Drop table existingtablename

egs :- Drop table employee

2080-08-11

(ii) Data Manupulation language (DML)

DML statements are used to manipulate in table. There are 4 standard DML commands :-

a.) Insert
b.) update
c.) Delete
d.) Select

Insert

This command is used to insert new tuples into the table.

Syntax :- Insert into tablename(attributes list)
values(values for attributes);

examples :(i) Inserting data into table employee

Insert into employee
(eid, ename, salary)
values(1, "Rajan", 5000);

~~eid int,~~
~~ename rajesh,~~
~~salary 5000.~~

Date
Page

2.) Inserting data in table student

⇒ SQL >insert into student
(Roll-no, Student_Name, DOB, Address)
values(1, "shyam", 2063-02-12, "Birgunj");

b.) Update

This command is used for modifying value in the table. For this Set clause is used with update statement. This clause specifies the Modification to be made to selected tuples.

E.g. 1. update employee
set salary = salary * 0.1;

2. update employee
set salary = salary * 2
where salary >= 5000;

3. update student
set address = "Birgunj"
where address = "simra";

c. Delete

This command is used for deleting data from the table.

Eg. 1. Delete from employee
{ Deleting all the tuples in employee table }

(2.) Delete from employee
where Name = "Rajan"
{ deleting tuple matching the condition }

3. Delete from employee
where salary between 5000 and 10,000.
{ deleting multiple tuples matching the condition }.

d.) Select
This command is used to retrive data desire as the result of a query one or more tables.

Eg. 1. SQL statement for selecting all data from table student.
⇒ select * from student.

2. SQL statement for listing only selected data from student.
⇒ SQL > select name, address from student.

Date
Page

3. SQL statement for selecting data that matches the condition.

=> SQL > select name from employee where Salary >10,000.

4. SQL statement for selecting data matching more than one condition.

=> SQL > select name from employee where address = "Birgunj" and Salary < 5000

5. SQL statement for selecting data in a given range.

=> SQL > select * from employee where Salary between 10,000 and 20,000.

6. SQL statement for selecting data of matching pattern.

=> SQL > select * from employee. where address like "Bir %".

7. SQL statement for selecting data is stored order (desc/asc).

=> SQL > select * from employee order by salary.

Date 2080-08-12
Page

8. SQL statement for selecting unique data
⇒ SQL > select distinct age from student

9. SQL statement for selecting a set of data using group by clause.
⇒ SQL > select age from student group by address.

* Using aggregate functions
These are five built in aggregate functions that takes multiple value as input and return a single value:

(i.) Sum
(ii.) Avg
(iii.) Max
(iv.) Min
(v.) Count

examples :-

(i.) SQL > select Sum (salary) from employee.

(ii.) SQL > select Avg (salary) from employee.

(iii.) SQL > select max (salary) from employee.

(iv.) SQL > select mix (salary) from employee.

(v.) SQL > select Count (name) from employee.

Date
Page

* The union operation.
To find all the bank customers having a loan, an account or both at the bank.
- SQL> select customer_name from depositer union select customer_name from Borrower.

* The intersect operation.
To find all customer who have both a loan and an account at the bank.
- SQL> select cust_name from depositer intersect select cust_name from Borrower.

* The except operation.
To find all customers who have both an account but no loan at the bank.
- SQL> select cust_name from depositer except select cust_name from Borrower.

* The Rename operation.
To Replace the attribute name i.e. loan number with the name loan_id.
- SQL> select cust_name, loannumber as loanid amount from borrower.

* Joining Two tables using where clause.

SQL> select cust-name, ~~loanno~~ loanid, amount from Boxxower, loan where Boxxower.loanid = Loan.Loanid.

* Joining Two tables using join clause.

SQL> select cust-name, loanid, amount from Boxxower join Loan ON Borrower.Loanid=Loan.Loan

17

* Transaction

A Transaction consist of a sequence of query i.e. any activities done with the database.

- Commit

Commit is similar to saving changes to a document. i.e. being editing that means it makes the updates performed by the transaction become permanent in the database.

- Rollback

Rollback is similar to quitting the edit session without saving changes, i.e. it undo all the updates perform by the SQL statement in the transaction.

Date
Page

Note :- Once a transaction has executed commit it effects can no longer be undone by Rollback

(iii) * Authorization (DCL statement)

We may assign a user several forms of authorization on parts of the database, such as

i. To Read data
ii. To insert New data
iii. To update data
iv. To delete data

This type of Authorization are called a privilege The SQL standard includes the privileges like select, insert, update and delete.

SQL includes commands to "grant" and "Revoke" privileges.

(1.) The grant statement is used to confirm authorizatio

Example :- (i) Grant select on account to "John", Merry

select → privilege
account → tablename
John, Merry → user name

(ii) Grant update (amount) on loan to John.

update → privilege
amount → attribute
loan → tablename
John → username

(2.) To Revoke an authorization, Revoke Statement is used which cancel the privilege i.e. granted previously.

Example:- (i) Revoke Select on Account from mery

(ii) Revoke update(amount) on loan from john.

* Subqueries

SQL provides a mechanism for nesting queries. A subquery is a select-from-where expression i.e. nested within another select-from-where expression. The expression following where clause can be either a simple predicate or it can be a query itself which is known as subquery.

eg: (i) Select Cust_Name from Borrower where Cust_Name in (select Cust_Name from Depositer)

* Views

A view is a virtual table that doesnot actually exist. It is made up of a query on other tables in the database. It could include only certain columns or rows from one or multiple tables.

Date
Page

* The advantages of creating views are :

(i.) Security
Users can be given access to only those data that concern them.

(ii.) Data integrity
Users can update only those data that the view shows.

(iii.) Simplicity
Even if a view is a multi table query, querying the view is simple like a single table query.

(iv.) protection from change
if the structure of the database changes, the users view of data can remain the same and vice-versa.

Syntax :- Create view viewname as query expression.

Eg :- Create view Empview as (select EmpNum, Name, post, Salary from employee).

* Constraints

In a relational database schema there are certain constraints to which the contents of a database must confirm. Constrains are the ways to define the validity of data. Constraints are also used to enforce referential integrity of data.

Types of constraints:

- primary key

A primary key is a column that uniquely identifies a particular row in a table.

- foreign key

A foreign key is a way to further constrain the allowable values of a column, to data that exist in another table. usually The table with the foreign key constraint is refering to another table by ~~tabl~~ that's table's primary key attributes

- Integrity constraints

A relational database schema is a set of relations with a set of integrity constraints which determines if the database is in a valid state or not.

- NULL value Constraints
  it specify that an attribute is NULL or Not

- Unique Constraints
  it specify that an attribute must have an unique value.

- check Constraints
  it specify some condition to be checked for an attribute.

Normalization
Unit - 3

V.V.I

* **Normalization**

Database Normalization is the process of organizing the attributes and tables of a relational database to minimize data redudancy.

Normalization involves decomposing (simplifying) A table into smallex tables without loosing informations.

Dr. E.f. Codd, The inventor of the Relational model, introduced the Concept of Normalization.

There are following Normal forms:

1. First Normal Form (1NF)
2. Second Normal Form (2NF)
3. Third Normal Form (3NF)
4. BCNF (Boyce Codd Normal Form)

1. First Normal Form (1NF)

In First Normal Form, Any row must not have a column in which more than one value is save. we must separate such data into multiple rows. Simply, we can say that, data should be atomic i.e. it can't be decompose into smaller pieces.

For example: Consider a table which is not in First Normal Form.

Student

Name Student	Age	Subject
Ram	14	Biology, Math
Shyam	16	Math
pawan	17	Math

Student table following 1NF will be:

Student

Name	Age	Subject
Ram	14	Biology
Ram	14	Math
Shyam	16	Math
pawan	17	Math

Date
Page

using the 1NF data redudancy increases as there will be many columns with same data in multiple rows but each rows as a whole will be unique.

2. Second Normal form (2NF)

A table is in a second Normal form if each Non-key column depends on the entire key. for example:

Student

Roll-No	Name	Marks	
1	Ram	Eng 40	Math 32
2	Shyam	Eng 50	Math 60
3	Hari	Eng 52	Math 70

Converting student table into 2NF.

P.K

Student

Roll_No	Name
1	Ram
2	Shyam
3	Hari

Student Marks

Roll_No	Eng	Math
1	40	32
2	50	60
3	52	70

Date 2080/08/15
Page

3.) Third Normal form (3NF)

A table is in third Normal Form if it is in second Normal form and every Non-key attribute of table non-transitively dependent on each candidate key of table.

if a column depends on the column i.e. not the part of a key then split those column into new table.

Converting 2NF into 3NF:-

Student

Roll_No	Name
1	Ram
2	Shyam
3	Hari

English

Roll-No	English
1	40
2	50
3	52

Math

Roll-No	Math
1	32
2	60
3	70

Date
Page

4.) Boyce Codd Normal Form (BCNF)

if an attribute of a composite key is dependent on an attribute of the other composite key, A Normalization BCNF is required.

For example:- Consider a table professor:

professor

(pro-code, department, head of department, Time

Now, Assume that:

-> A professor can work in more than 1 department
-> The time he spents in each department is given
-> Each department has only one head of department

The table professor is:

pro code	department	Head of department	time
P_1	physics	Ghoss	50 hrs
P_1	chemistry	Ram	50 hrs
P_2	Math	Shyam	30 hrs
P_2	physics	~~Has~~ Ghoss	40 hrs
P_3	Math	shyam	70 hrs

Converting table professor into BCNF.

Professor

pro-code	department	time
P_1	physics	50 hrs
P_1	chemistry	50 hrs
P_2	Math	30 hrs
P_2	physics	40 hrs
P_3	Math	70 hrs

Department

Department	Head of Department
physics	Ghoss
chemistry	Ram
Math	Shyam
~~physics~~	~~Ghoss~~
~~Math~~	~~Shyam~~

* Stored procedure

A stored procedure is a set of SQL statements with an assigned name, as a group, so it can be reused or shared by multiple programs.

Syntax:
```
Create procedure procedureName
AS
 SQL Statements
Go;
```

To execute Stored procedure.

Syntax:
```
Exec procedureName;
```

eg :-
```
Create procedure selectallStudent
AS
select * from student
Go;
Exec selectallStudent;
```

Date 2080/08/20
Page

* Trigger

A Trigger is a procedural code that is automatically executed in response to certain events on a particular table or view in a database. The Trigger is mostly used for maintaining the integrity of the information on the database. It is a special kind of stored procedure.

For example: If a new record is added to the employee table, new records should also be created in the tables of taxes, vaccations and salaries.

Some common events in trigger are:

(i.) Insert
(ii.) Update
(iii.) Delete

Example:
```
create trigger Stud_update
on student
for update
AS
if update (SID)
begin
print 'student ID shouldn't be change'.
print 'Rolling back'
Roll back
end
```

Date
Page

* Security

Database Security is a measure issue for any organization that includes the right to access certain information. Some informations must be protected and cannot be accessed wigely by unauthorized persons. The security can be classified ~~can be~~ into two categorized.

(i) physical security
(ii) Logical security

(i) physical Security

physical security is conserned with physically protecting resources from any kind of damage.

(ii) Logical Security

Logical security means protecting the data and informations and controlling access to the data stored in database.

Needs of Security:

- for controlling unauthorized access.
- for providing privileges to an authorized users.
- for protecting illegal modifications.
- for protect accidental loss of data.
- To stop pixacy.

* Integrity violation

The violation of data is concerned with those aspect of the system in which a particular security or integrity rule is broken. For example, if any entered data does not satisfies the integrity constraints that is defined or an unauthorized user gain access to the data, such conditions are known as Integrity violation.

* Authorization.

Authorization means to provide authority to users that either a person or user account is authorized for accessing the data or not. Authorization is completely associated with different type of rights which is given to the users. For example, if a user has given right read only data, he cannot insert or modify the data in the database.

* Encryption

V.V.I

Encryption is the process of translating data into a secure code. Encryption is the most effective way to achieve data security. To read an Encrypted file, user must have access to a secure key or password that enables to decrypt data. Un Encrypted data is called "plain text". Where as Encrypted data is known as "cipher text"

Date
Page

* **Decryption**

Decryption is the reverse process of Encryption. The process of decoding data that has been encrypted into a secure format is known as Decryption. Decryption also required a secure key or password to convert cipher text into plain text.

2080/08/24

* **Query processing (Transaction processing)**

* **Query**

A Query is an ingury to the database using the select statement which is used to ~~abstract~~ extract the data from the database in a readable format according to the user's request.

* **Query processing**

There are some techniques used by a DBMS To process, optimize and execute high level queries.

The figure given below illustrates the different steps (operations) of a query processing which is represented as a tree data structure called a query tree:

Date
Page

Query in a high level language (SQL)

↓

Scanning, Parsing and validating

↓

Immediate form of query

↓

Query optimizer

↓

Execution plan

↓

Query for generator

↓

Code to execute the query

↓

Run time database processor

↓

Result of query

* A query expressed in a high level query language such as SQL must be scanned, parsed and validated.

→ The scanner identifies the language token's such as SQL keywords, Attribute names, Relation name etc.

→ The parser checks the query syntax of the query language.

Date
Page

→ The Query must be validated, By checking all the attributes, values and relation names are valid or not.

The Query optimizer module has the task of producing and execution plan and the code generator generates the code to execute that plan. The runtime database processor runs the query code by using compiler or interpreter to produce the query result. if any errors will occur it generates an error message.

A query has many possible execution plans for retriving the result from the database files, And the process of selecting a suitable one for processing a query is known as Query optimization.

* Query Interpretation. (SQL Query ⇒ Compiler/Interpreter ⇒ Binary code ⇒ Computer understandable format ⇒ Result of query.

Query Interpretation is the process of translating the query (SQL statement) into binary language, by checking and correcting the errors, to make them understandable by the computer system.

which helps to produce the result.

* Query Decomposition.

Query decomposition is the process of breaking down (diving) a complex query into multiple query statement to make them simple and easy to execute.

A big query will take alots of time to execute and detecting errors is also difficult in such queries so query decomposition is required to overcome such situation

* Query cost estimation

Query cost estimation is concerned with a cost in which the enhancer consider what amount of time a query will require. Then the analyzer attempts to pick the most ideal query plan by taking a glance at the inquiry and insights of the information, attempting a few execution designs and choosing the most in-expensive of them.

* Evaluation of Expressions.

for evaluating an expression that carries multiple operations in it, we can perform the computation of each operation one by one.

Date
Page

However, In the query processing system we use two method for evaluating and expression carrying multiple operations. These methods are :

(i.) Matexialization
(ii.) pipelining

(i.) Materialization

In this method the given expression evaluates one relational operations at a time in an appropriate sequence (order). After evaluating all the operations the outputs are materialized in a temporary relation for their subsequent (further) uses which requires space in the disk for storage.

(ii) pipelining

It is an alternate method to the materialization. In this method each relational operation of the expression evaluate simultaneously in a pipeline. After evaluating one operations it's output is passed on to the next operating and the chain continues till all the relational operations are evaluated thoroughly. So there is no requirement of storing a tempory relation. Sun an advatage of pipelining makes it a better approach as

Compare to the materialization.

* Database performance tuning.
Database performance tuning refers to a group of activities DBAs perform to ensure databases operate smoothly and efficiently. It helps re-optimize a database system from top to bottom, from software to hardware, To improve over all performance

Unit-7
Recovery

Date 2080/08/25
Page

7.2 * Recovery

An improtant part op a database syster is a recovery schem that ~~than~~ can restore the database to it's consistant state that existed before. if the database is physically damaged such as a discrashed, The recovery method restore the past copy op the database.

The main technique used to handle such cases is database backup. i.e. The hole database and it's contents are copied into a storage medium such as magnetic tapes.

7.1 * failure classification.

There are several types op failure that may occur in a database system:

i.) Transaction failure

There are two types op error that may cause a transaction failure.

a.) Logical errors

The transaction can't be continue with it's normal execution because of some logical conditions such as wrong input, data not found, data overflow etc.

b.) System error.

The system has entered in an ~~undisirable~~ undesireable

State, as a result of which a transaction cannot continue with it's normal execution.

ii.) System crash

if there is a problem in computer hardware, database software or operating system that causes the loss of information, such condition is known as system crash.

iii.) Disc failure.

A Disc block losses it's content due to any type of failure during a data transfer operation that causes a transaction not possib

* Backup Recovery.

The main goal of recovery is to ensure atomicity property of a transaction. if a transaction fails before complety it's execution, the recovery mechanism has to make sure that the transaction has no lasting effects on the database.

Recovery from transaction failures usually means that the database is recovered to the most recent consistent state just before the time of failure. or if the database is physically damaged, The recovery method restor a past copy of the database. The technique must be used to handle such situation is keeping

Date
Page

database backup.

7.4 • Log Based Recovery

Log Based Recovery means to recover the data by using log files. As we perform any transaction in database file, A log file is created and maintained that contain all the information about database transaction. if the transactions fails we the help of some command to log file, the restoring of database is possible.

2080/08/27

• Shadow paging

7.5 (i) In shadow paging, A database is devided into multiple pages each of which represents a fixed size disc memory.

(ii) Actually shadow pages are replicas (duplicate) of the original database which are created at the start of a transaction where the database state is copied into the shadow pages.

(iii) Only the original database will be change during the transaction not the shadow pages.

(iv.) The updates in the shadow pages are made when the transaction reaches the Committed State.

(v.) In the case of system failure, Recovery procedure are carried out after comparing the database true pages (original pages) to it's shadow pages.

7-3 * In-place and out-of-place update.

In-place update involves installing or upgrading a system or application without altering or replacing it's existing hardware or software components why out-of-place update envolves replacing hardware or software components in-order to install or upgrade a system or application.

7.6 * Local Recovery Manager

The local recovery manage is the component that is responsible for processing commit and ~~avoid~~ abort operation. It is also responsible for the re-start operation which ~~ind~~ initiates recovery from a failure to bring the database back into a consistant state where it can process transaction again.

7.7 * UNDO and REDO protocols.

When a failure occurs the DBMS uses the transaction log to determine which transaction where incomplete the time of the failure. Then it performs a series of operations to undo the incomplete transaction and REDO the completed once.

- UNDO

it under the actions of transaction that didn't commit, so that the database reflects only the action of committed transaction.

- REDO

It Repeats all actions, starting from an appropriate point in the log and restores the database state to what it was at the time of the failure.

Unit - 6

Date 2080/09/03
Page

* Concurrency Control

Concurrency control is a mechanism to achieve isolation i.e. to control the interaction among the concurrent transaction in order to prevent them from destroying the consistency of the database.

* Transaction

The term transaction refers to a collection of operation that forms a single logic unit of work.

A transaction is a unit of program execution that access and updates various data address. For example: Transfering money from one account to another is a transaction

To ensure the integrity of the data the database system should maintain various properties. These properties of the transaction is known as ACID properties.

(i) A → Atomicity
(ii) C → Consistency
(iii) I → Isolation
(iii) D → Durability

(i) Atomicity

It is important that either all actions of a transaction be executed completely or

incase of any failure, partial effects of each incomplete transaction be undone.

(ii.) Consistency

Execution of a transaction in Isolation preserves the Consistency (accuracy) of the database.

(iii.) Isolation

Database system must provide mechanism to isolate each transaction from the effects of other concurrently executing transaction.

(iv.) Durability

After a transaction complete successfully, the changes it has made to the database remains even if there are any system failure.

* Concurrent Execution in DBMS.

In a multi user system, multiple user can access and use the same database at one time, which is known as the concurrent execution of the database. it means that the same database is executed simultaneously on a multiuser system by different users.

Date
Page

* Transaction Model

Generall transaction model describes those state of transaction in which it has to perform from begin to end block. A transaction must be in one of the following states:

(i.) Active

The initial state in which the transaction stays in, why it is executing.

(ii.) partially committed

After the final statement has been executed.

(iii.) failed

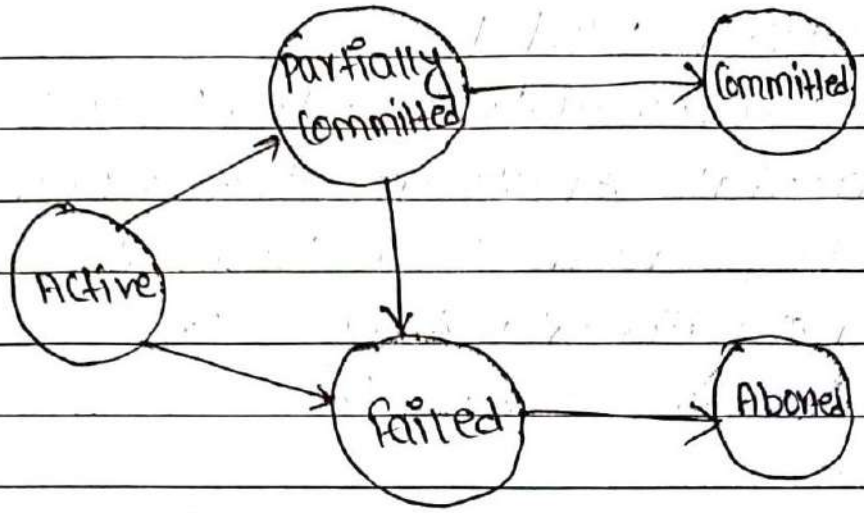
After the discovery that ~~some~~ normal execution can no longer proceed.

(iv.) Aboxted

After the transaction has been rolled back and the database has been restore to its state prior to the start of the transacti

(v.) Comitted

After Successful Completion.



* Sceduling & Serializability

* Sceduling

The execution sequences in which the transactions are executed one at a time in the ordered is called sceduling.

* Serializability

Each serial scedule consist of a sequence of instructions from various transactions where the instructions belonging to one single transaction appear together in that schedule is known as serializibility. for a set of n transaction their exist n! diffent valid serial schedules.

* **Lock**

A lock is a mechanism to control concurrent execution or access to a data item.

* **Lock based Concurrency Control.**

When performing any transaction, there is required some controls or protocols that means while one transaction is accessing a data item no other transaction can modify that data item. It is possible to allow a transaction to access a data item only if it is currently holding a lock on that item.

* **Types of Locks**

There are various modes in which a data item may be locked:

(i) **Shared mode**

If a transaction has obtained a shared mode lock on a item then it can read but cannot write.

(ii) **Exclusive Mode**

If a transaction has obtained an exclusive mode lock on a item then it can both read and write.

* Granting of lock

When a transaction request a lock on a dataitem in a particular mode and no other transaction has a lock on the same data item in a conflecting mode, the lock can be granted.

* Time Stamp based protocols.

Another method for determining the serializibility order is to select an ordering among transactions in advance is referred as time stamp ordering technique.

With each transaction T_i in the system, we associate a unique fixed time stamp($TS(T_i)$). This time stamp is assigned by the database system before the transaction starts execution. if a transaction T_i has been assigned time stamp and a new transaction T_j is enters the system then $TS(T_i) < TS(T_j)$. The time stamp of the transaction determined the serializibility order. There are two simple methods for implementing time stamp ordering technique (schem):

(i.) Use the value of the system clock as the time stamp

(ii.) use a logical counter is incremented after a new time stamp has been stamp.

Date
Page

* Two phase locking (2PL)

In database and transaction processing Two phase locking is a concurrency control method that guranties serializibility. the protocol utilizes locks, Applied by a transaction to data, which may block other transactions from accessing the same data during transaction's life.

Two phase locking has two phases, One is Growing and another is Shrinking

- Growing
  Where all the locks are been acquired by the transaction

- Shrinking
  Where the locks hold by the transaction are being released.

* Strict two phase locking (S2PL)

Strict 2PL is similar to 2PL as after acquiring all the locks, the transaction continue to execute normally. The only difference between 2PL and Strict 2PL is that, Strict 2PL doesn't realse a lock after using it. Strict 2PL waits until the hole transaction to Committee and then it releases all the locks at a time.

Date
Page

Strict 2PL doesn't have shrinking phase to realse locks.