Date: 21-6-18
MON TUE WED THU FRI SAT

- Database : It is a collection of data.
- Database Magnement System : DBMS Consist of collection of interrelate data and a set of programs to access those data

Application of DBMS. [BEAUT]

- Banking [BCF]
  - → Banking Information
  - → Credit Card Tranjaction
  - → Financial
- Enterprice Information [SOHAM]
  - → Sales
  - → Online Retailers
  - → Human Resource
  - → Accounting
  - → Manufacturing
- Airlines
- University
- Telecommunication

Scanned by CamScanner

Date:
MON TUE WED THU FRI SAT

- Purpose of DBMS [SAIDDDC]
(Adv. of DBMS, DisAd. of File System)

→ Security Problems.
→ Automicity Problem (All or nothing)
→ Integrity Problem
→ Data Isolation
→ Data redundancy & Inconsistency
→ Difficulty in accessing data.
→ Concurrent access.

DATA	INFORMATION
Data can be any text, character, word, numbers, pictures, sound or video that means little or nothing to human.	when data is processed, organized, structured or presented in a given context so as to make it useful is called information.
Data is raw, unorganized fact that need to be processed.	Information is useful and usually formated in a manner that allows it to be understood by human

- View of data

→ DATA Abstraction

Many database system users are not computer trained, developers hide the complexity from user through several levels of abstraction. To simplify users interaction with the system.

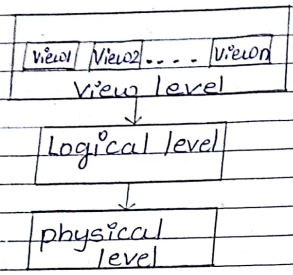
1: physical level: The lowest level of abstraction describes how the data are actually stored. It describes complex low level data structure in detail.

2: logical level: The next higher level of abstraction describes what data are stored in database. and what relationship exist among those data.

- physical data independance:
Implimentation of simple structures and logical levels may envolve complex physical level structure, the user of logical level doesn't need to be awaare of this complexity. This is refferred to as physical data independance

3: View level: The highest level of abstraction describes only part of entire database.

Scanned by CamScanner



- Schema: The overall design of database is called schema.

1. physical Schemphysical schema describes the db design at physical level
2. logical Schema logical Schema describes the db design at logical level
3. Sub Schema: The database may involve several schema at view level is called Sub Schema.

- Intence: The collection of information stored in database at particular moment is called intence.

Eg: CE/IT
faculty (Name, Id)
Student (en, Name, Add)
Course (Cu.Id, Name, Year)
division (Id, branch).

- Data Model
Underline the structure of database is the data model. It includes a collection of tools for describing data, data relationship, data symendix and constraints.

A data model provides a way to describe the design of database at physical, logical and view levels. Data model can be classified into four categories. (ROSE)

- Relational Model
- Object-based Data model
- Semi-structured data model
- Entity Relationship model

Database Languages

1. DML (Data Manipulation Language)
It express db queries & updates.
2. DDL (Data Definition Language)
It express db schema/specified db schema

DML: DML is a language that enables use of to manipulate / access data organized by data model.
The types of access are

Scanned by CamScanner

- Retrieval of information stored in the db.
- Insertion of new information into db.
- Deletion of information from the db.
- Modification of information stored in the db.

There are basically two type of DML

1. Procedural DML: It require user to specify what data are needed & how to get those data.

2. Declarative / Non-Procedural DML: It require a user to specify what data are needed without specifying how to get those data.

Query - Query is a statement requesting the retrieval of information

Query language - The portion of DML that involves information retrieval is called Query language

DDL: The set of definition of db expressed by special language is called DDL. DDL is also used to Specify additional properties of data.

Database users and DBA

The people to work with database can categorize as.

i) Database users
ii) Database administrator

There are four types of database user. (NASS)

a) Name users → They are unsophisticated users to interact with system by invoking one of the application program that have been written previously.

b) Application programmers → They are computer professionals who write application programs Rapid Application Development (RAD) tools are tools that enable application programmers to construct forms & reports.

Scanned by CamScanner

c) Sophisticated users → They interact with system without writting any program. They form their request either using database query or by using table.

d) specialized users → They are Sophisticated users who write specialized database application that do not fit into traditional data proccesing network.

DBA

A person who has central control over the system is called DBA. The functionalities of DBA are (GRASS).

i) Granting of Authorization for data access
ii) Routine Maintenance → RBM
iii) Schema Definition
iv) Schema & Physical organization modification

Storage Manager (4 MARKS)

Storage manager is a component of database system that provide interface between low level data stored in database and the application program. it translate various DML Statements into low level file system command. There are various component of Storage Manager. (FB AT DID).

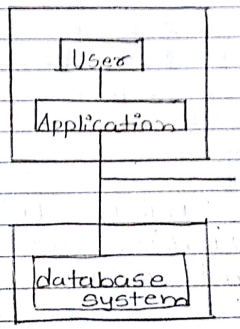
i) File Manager:
ii) Buffer Manager
iii) Authorization & Integrity Manager
iv) Transaction Manager
v) Data files
vi) Indices
vii) Data dictionary.

Query Processor

The component of query Processor are

i) DDL interpreter
ii) DML compiler
iii) Query evaluation engine.

2-tier (2 MARKS)

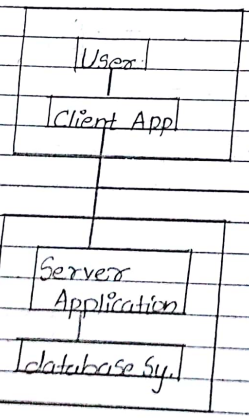


Scanned by CamScanner

In 2-tier architechure database application are partision into two parts, one at client side another at server site.

The application resites at client machine where it invokes database system functionality at server machine through query statements.

3-tier



In 3-tier database architechure the client machine act as front end and doesnot contain any direct access with server. Instead the client end communicates with application server usually to form interface. The application server communicate with database system to access data.

Questions

1. Define foll. terms
   a) Data
   b) Database
   c) DBMS
   d) Schema
   e) Intence
   f) DBA
   g) Physical data independence
   e) Data Abstraction
2. Enlist and Explain Different Application of DBMS
3. Write a short note on purpose of DBMS. (4 Marks)
   or
   What are the disadvantages of file system over DBMS.
   or
   Explain advantages of DBMS over file system.
4. Explain different labels in Data Abstraction.

Scanned by CamScanner

Date:
MON TUE WED THU FRI SAT

5. What is data model? Explain diff. Categories of data model.
6. Explain View of data. (Data Abstraction, Data Model, Schema)
7. Explain diff. uses of Data Definition Language.
8. Explain diff. types of database users.
9. Discuss role of DBA.
   or
   Enlist and Explain functions of DBA.
10. Enlist Components of DBMS Structure. Explain Storage manager & its components.
11. Explain Components of Query processor.
12. Explain 2-tier database architechure.
13. Explain 3-tier database architechure

Date: 12-7-18
MON TUE WED THU FRI SAT

Unit-2

Relational Model

Relational Algebra

A set of opperators that take one or more relations and return a relation as an output.

Fundamental Operator

i) Union
ii) intersection
iii) difference
iv) Cartesian product

The result of union operation returns all tuples appearing in either or both of given relation. Duplicate are eliminated.

For ex: Find all the subject & their credicts thaught in B.Tech CE & B.Tech IT from below relation

B.Tech_C.E

Subject	Credits
DBMS	4
DFS	5

B.Tech_IT

Subject	Credits
DBMS	4
OOP	5

B.Tech_C.E U B.Tech_IT

Subject	Credits
DBMS	4
OOP	5
DFS	5

The result of intersection operations

Scanned by CamScanner

return all tuples those are in both relation.

B.Tech_CE ∩ B.Tech_I.T

Subject	Credits
DBMS	4

Difference

The result of difference operation it returns all tuples in first relation but not in second relation.

B.Tech_CE - B.Tech_I.T

Subject	Credits
DFS	5

Cartesian Product

It returns all the tuples in all the relation. each tuple in first relation paired with all the tuples in second relation.

B.Tech_C.E × B.Tech_I.T

Subject	credits	Subject	Credits
DBMS	4	DBMS	4
DBMS	4	OOP	5
DFS	5	DBMS	4
DFS	5	OOP	5

Relational Algebra Queries (Displays rowwise data)

i) Selection operation

- The Select operation selects tuples that satisfy a given condition. The Select operation is represented as follow

$\sigma_{\text{<condition>}}(\text{Relation})$

The symbol 'σ' is used to denote the select operator and condition is an expression specified on the attributes of relation.

For eg: Subject

Subject Id	Subject_name	Credit
301	DBMS	25
302	DFS	23
303	OOP	24
304	DLD	22
305	Maths-3	24
306	FOL	23

Queries: Display details of subject whose credit is greater than 23

(subject)

$\sigma_{\text{credit} > 23}$

Queries: Display sub. details whose subject name start from D

(subject)

$\sigma_{\text{subject\_name} = 'D\%'}$

Scanned by CamScanner

Date:
MON TUE WED THU FRI SAT

Display subject details whose credit is 24 & Subject_name start from 'O'.

(subject)

$\sigma_{(credit=24)\ AND\ (Subject\_name='O\%')}$

ii) Projection Operation

- The project operation selects certain attributes from relation while discarding others. It removes any duplicate tuples from the result operation.
- The project operation is represented as follows:

$\Pi_{A1, A2 \cdots An}$ (R)

- The symbol 'π' is used to denote the project operation and $A_1, A_2, \ldots A_n$ is list of attributes from the attribute of relation 'R'

For eg:

Display all Subject Id's with Subject name.

(subject)

$\Pi_{subject\_Id,\ subject\_name}$

Date:
MON TUE WED THU FRI SAT

Fri 12
$\pi x \rightarrow 6$

Composition of Select And Project Operation

For eg:

Display the name of subjects having credit greater than 23.

(subject)

$\Pi_{subject\_name}\ (\sigma_{credit>23})$

Book (Book_Id, Title, Author, Publisher, Year, Price)

Display all book title with author & price.

(Book)

$\Pi_{Title, Author, Price}$

Display books published in Year 20801.

(Book)

$\sigma_{Year=2001}$

Display all the details of book whose publishing Year greater than 23 or price is less than 500.

(Book)

$\sigma_{(Year>23)\ OR\ (Price<500)}$

Scanned by CamScanner

Date:
MON TUE WED THU FRI SAT

Display the titles of books having price greater than 300 (BOOK).

$\Pi_{Title}(\sigma_{price>300}$ (BOOK))

Display Author of book with Book-Id B002.

$\Pi_{Author}(\sigma_{Book\text{-}Id='B002'}$ (BOOK))

Schema
Person (Id, Name, Add)
Car (Reg no, Year, Model)
Accident (Date, Driver, Car regno)
Owner (Id, Lic.)

Q→ Find the names of persons who are involved in an accident

Π_{Name} (Person) $\cap$ Π_{Driver} (Accident)

Find the registration no. of Cars which weren't involved in any accident

$\Pi_{Reg\text{-}no}$ (Car) $-$ $\Pi_{car\text{-}regno}$ (Accident)

Date:
MON TUE WED THU FRI SAT

Rename Operation

In relational algebra user can rename either the relation or the attributes or both. The general rename operation can take any of the foll. forms.

1) $\rho_{\langle new\text{-}name\rangle}(R) \Rightarrow \rho_{sub\_details(Id, Subname, Credit)}$

2) $\rho_{(newAttributename)}(R)$

3) $\rho_{\langle new\text{-}name\rangle(newAttributename)}$

Schema Subject (S_Id, name, credits)

i) Rename Subject relation with Sub details
$\rho_{sub\_details}$ (subject)

ii) Rename attributes name of Subject with Id, Sub_name, credit

$\rho_{(Id, Subjectname, credits)}$ (subject)

iii) Rename subject relation as well as attributes name of it.

$\rho_{sub\_details(Id, Subname, credit)}$ (subject)

Scanned by CamScanner

Additional Operation

i) Natural Join Operation

The natural joint is a binary operation that allows us to combine certain Selection & certain product into one operation.

It is denoted by ⋈

For ex:

Subject

Sub_id	Subna
5001	DFS
5002	DBMS
5003	OOP
5004	Em-3

class.

Sub_id	Class
5001	3-CE
5002	3-IT
5003	3-CE
5005	5-IT

Display all the details of Subject with its class.

Subject ⋈ Class

Sub_id	Subname	class
5001	DFS	3-CE
5002	DBMS	3-IT
5003	OOP	3-CE

List out Subject name with their class

(Subject ⋈ class)

Π subname, class

Schema

Salesperson (no, name, startyear, deptno)
trip (no, city, date, trip_Id)
expense (trip_Id, acc, amt)

Display city & amount for trip that exceed 10000 in expence.

Π city, amount (σ amount > 10000 (trip ⋈ expense))

Display the no. of Salesman who took trips to delhi.

Π no (σ city = "delhi" (trip))

Display the trip expences details with city where no. is 123

Π city, Amount, acount, Trip_Id (σ no = 123 (trip ⋈ expense))

1) Schema

Vehicle (reg_no, Colour, model)
person (E_no, name, add)
Owner (E_no, reg_no)

a) List the reg_no of Vehicles owned by John

b) List the names of person who owned maruti car.

c) List all the red coloured vehicle.

Scanned by CamScanner

1. $\Pi_{reg\_no}(\sigma_{name='John'}(Person \bowtie_{Person.name = Owner.name} Owner \bowtie_{Owner.vehicle\_id = Vehicle.vehicle\_id} Vehicle))$

2. $\Pi_{name}(\sigma_{model='maruti'}(Person \bowtie_{Person.name = Owner.name} Owner \bowtie_{Owner.vehicle\_id = Vehicle.vehicle\_id} Vehicle))$

3. $\Pi_{V.} \sigma_{colour='red'}(Vehicle)$

Division Operation

It is suited to queries that include the phrase for all. For ex:

r

A	B	C
α	1	γ
γ	1	β
α	2	γ
β	2	α
γ	2	β

s

B
1
2

r ÷ s

A	C
α	γ
γ	β

Assignment Operation

The assignment operation is denoted as (←). The result of expression to the right of Assig. Operator is assigned to the relation variable on the left of Assi. Operator.

For ex: (Student)

temp ← $\sigma_{studId=04}$

Extended Relational Algebra Operation

The basic relational algebra operation have been extended in several ways.

1> Generalised projection
2> Aggregate function
3> After join

1> ⇒ The Generalised projection operation extend the projection operation by allowing the arithmetic fun^n to be used in the projection list. The general form of generalised projection is $\Pi_{F_1, F_2, \ldots F_n}(E)$ where

$F_1, F_2, \ldots F_n$ are arithmetic expre. involving Constants & attributes.

For ex: Display name & % of Students (student)

$\Pi_{name, Total/5 \text{ as percen.}}$

2> ⇒ Aggregate fun^n take a collection of value & return single value as a result. Aggregate fun^n are sum, Avg, Count, Dis: Count Distinct, Min, Max.

For ex: Calculate Total Salary paid to employees (employee)

$\Pi_{sum(salary)}$

Date:
MON TUE WED THU FRI SAT

Display avg. salary of employees
(Employee)

$\mathcal{G}$ Avg(salary)

Count of number of unique depart. in employee relation
(Employee)

$\mathcal{G}$ count distinct (dept).

3) ⇒ Outer Join

After join operation is an extension of join operation to deal with missing information. There are three types of outer join

1. Left Outer
2. Right Outer
3. Full Outer

1) ⇒ It is denoted by '⟕', It takes all tuples in the left relation that didn't match with any tuples in right relation, It puts null value for all other attributes from right relation.

For ex",

Date:
MON TUE WED THU FRI SAT

Student

name	city
abc	Surat
XYZ	Bardoli
pqr	Surat
mno	Valsad

Details

Name	Dept
XYZ	CE
def	IT
abc	EE
pqr	CE
klm	Mech

Student ⟕ Details

Name	city	Dept
abc	Surat	EE
XYZ	Bardoli	CE
pqr	Surat	CE
mno	Valsad	Null

2) ⇒ Right Outer join is denoted by ⟖. It takes / puts tuples from right relation that didn't match with any tuples from left relation with null. Then add them to the result,

Student ⟖ Detail

Name	Dept	City
XYZ	CE	Bardoli
def	IT	Null
abc	EE	Surat
pqr	CE	Surat
klm	Mech	Null

Scanned by CamScanner

3) ⇒ It is denoted by ~~DE~~

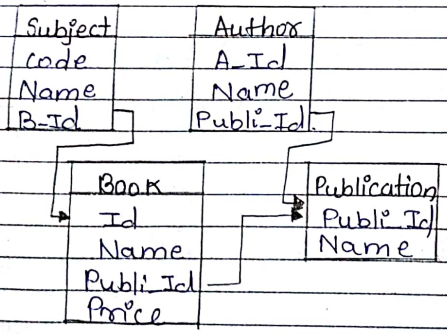
Student ~~DE~~ Details

Name	city	Dept
xyz	Bard.	CE
def	Null	IT
abc	Surat	EE
pqr	Surat	CE
klm	Null	Mech
mno	Valsad	Null

Schema Diagram.

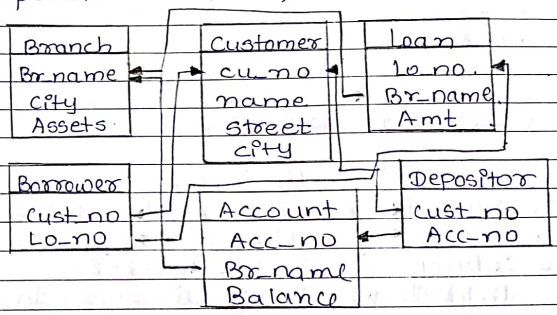
Library Management.

Book (Id, Name, Publi_Id, Name)
Publication (Publi_Id, Name)
Author (A_Id, Name, Publi_Id)
Subject (code, Name, B_Id)



Schema Diagram for Banking System

Branch (Br_name, city, Assets)
Customer (cu_no, name, street, city)
Loan (Lo_no, Br_name, Amt)
Borrower (cust_no, Lo_no)
Account (Acc_no, Br_name, Balance)
Depositor (Cust_no, Acc_no)



Schema of University

Dept (Dept_name, build, budget)
Course (Co_Id, Title, Dept_name, credits)
In Faculty (Id, Name, Dept_name, Salary)
Student (St_Id, Name, Dept_name, Marks)
Teaches (Co_Id, Sem, Year)
Councillor (St_Id, Id)

Scanned by CamScanner

Dept	Course	Faculty

Students	Teacher	Councillor

Relational Algebra.

Consider the foll. Schema.
Supplier (S_ID, S_Name, Address)
Parts (P_ID, P_Name, Color)
Catalouge (S_ID, P_ID, Cost)
Write the foll. queries in relational algebra.

1) Find the names of suppliers who supply some red parts

$\pi_{S\_Name}(\sigma_{color='red'}(Supplier \bowtie Catalog \bowtie Parts))$

2) Find the S_ID of suppliers who supplied some red or green parts

$\pi_{S\_ID}(\sigma_{(color='red') \text{ OR } (color='green')}(Supplier \bowtie catalog \bowtie parts))$

Find the S_ID of Supplier who supply every parts.

$\pi_{S\_ID}(catalog \div Parts)$

Find the P_ID of Parts of Supplied parts at least too diff. Suppliers

$\pi_{P\_ID}(\sigma_{C\_P\_ID >= 2}(\gamma_{distinct\ count(P\_ID)\ as\ C\_P\_ID}(catalog)))$

Find the parts detail whose price is greater than 10,000.

$\sigma_{cost > 10,000}(Parts)$

$\pi_{P\_ID, P\_Name, color}(\sigma_{cost > 10,000}(catalog) \bowtie Parts)$

Employee details
E_Details (E_ID, E_Name, Email, City)
Salary (E_ID, Salary, Catagory
Dept (D_ID, D_Name)
E_Dept (E_ID, D_ID)

1) List the names of employees whose city is either Mumbai or Banglore

$\pi_{E\_Name}(\sigma_{(City='Mumbai') \text{ OR } (City='Banglore')}(E\_Details))$

Scanned by CamScanner

2) List E_ID whose Salary is inbetween 50,000 to 80,000

$\pi_{E\_ID}(\sigma_{(Salary >= 50,000) \text{ AND } (Salary <= 80,000)}(Salary))$

3) List E_Name & their Email whose employee catagory is General Manager.

$\pi_{E\_NAME, Email}(\pi_{E\_ID}(\sigma_{category='G.M'}(Salary) \bowtie E\_Details))$

4) List E_ID who belongs to dept Customer Service

$\pi_{E\_ID}(\pi_{D\_ID}(\sigma_{Dname='C.S'}(Dept)) \bowtie E\_Dept)$

Entity-Relationship Model

1. **Entity**: Entity is thing or object in the real world that is distinguishable from other objects

2. **Entity Set**: Entity Set is set of entities of same type that share the same properties or attribute.

3. **Attributes**: An Entity is represented by set of Attributes. Attributes are classified as 1) Simple Attribute: Simple Attribute is an attribute composed of single component with independent existance, E.g → Roll No., Salary.

2) Composite Attribute: An attribute composed of multiple component each with independent existance is called Composite Attribute E.g → name composed of attribute like F_Name, M_name, L_Name.

3) Single Values Attribute: Single Value attribute is that holds single value for single entity.
eg:- C_ID, E_Roll.

4) Multiple Values Attribute: It is one that holds multiple values for single entity.
Eg:- hobby, which has multiple values as reading, singing, painting.

5) Derived attribute: It is one that represent that is derivable from value of related attribute.
Eg:- Age attribute can be derived from D.O.B.

Scanned by CamScanner

Date:
MON TUE WED THU FRI SAT

Relationship And Relationship Set

→ Relationship expresses association among several entities.

→ Relationship set is set of relationship of same type.

Types of Relationship

1) Unary
2) Binary
3) Ternary
4) Quaternery.

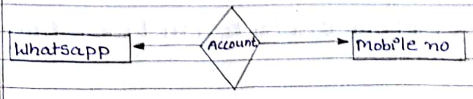
Constraints

Two main types of constraints are
1) Mapping cardinalities
2) Participation constraints

1) ⇒ Mapping Cardinalities express no. of entities to which another entity can be associated via relationship set. There are four types of mapping cardinalities

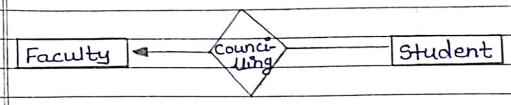
- one to one: An entity in 'A' is associated with at most one entity in 'B' and entity in 'B' is associated with at most one entity in 'A'.

for ex: A Whatsapp with single account with one single no.



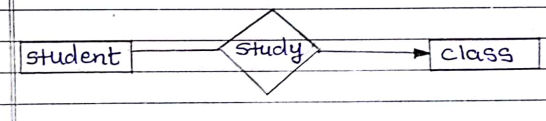
- one to many: An entity in 'A' is associated with any no. of entity in 'B' and entity in 'B' is associated with at most one entity in 'A'.

for ex: A faculty have many students for councilling.



- many to one: An entity in 'A' is associated with at most one entity in 'B' and entity in 'B' can be associated with any no. of entity in 'A'.

for ex: Many students studying in one class.

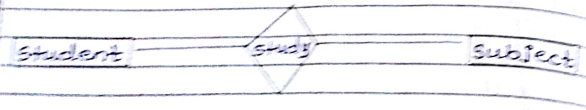


- many to many: An entity in 'A' is associated with any no. of entities in 'B' and entity in 'B' can be associated with any no. of entities in 'A'.

Scanned by CamScanner

Date :
MON TUE WED THU FRI SAT

For ex : Many Students studying many subjects.



SNAPTAIN

Date :
MON TUE WED THU FRI SAT

Scanned by CamScanner