DBMS

Database Management System

Data base

Data → • Raw facts, unprocessed facts
• Refers to what is actually stored

Information → • Result of processing raw data
• Refers to meaning of the data, understood by the user.

Data management focuses on the generation, storage and retrieval of data.

Problems / Limitations in file System :-

i) Data Redundancy
↳ repetitive data, wastage of memory

ii) Data inconsistency
↳ incorrectness of data

iii) Data Integration
↳ complex and time consuming

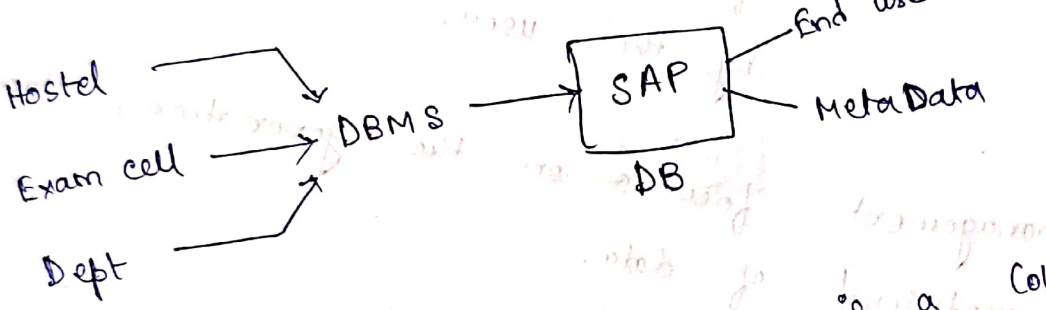
Scanned with OKEN Scanner

Database :- • Database is a collection of interrelated data.

• Database is the collection of End user Data and Metadata.

↳ provides the structure of data (data about data)

Database is an organized collection of data of an organisation or enterprise.



DBMS :- → Database Management System is a Collection of programs that manages Structure & Controls access to the data Stored in the database.

→ It includes tools to add, modify, or delete data from the database, ask Questions (or Queries) about the data Stored in the database and produce reports

→ DBMS Serves as the intermediary b/w the users and database

Database Types :-

No. of users → Single user DB
A database system where one person can access it at a time

→ Multi user DB
↳ multiple people can access at a time

Location :-
- Centralised Database
  - system failure
  - Bottleneck problem
- Distributed Database
  - Replication
  - Fragmentation

⊛ Advantages of Database System over File System :-

i) Reduced Data Redundancy
ii) Reduced Data inconsistency
iii) Protects from unauthorised access
iv) Allows data sharing
v) Allows multiple user interfaces
vi) uses data structures for efficient access/query processir
vii) Ease in Scaling
viii) Enforces Entity Integrity
ix) Allows backup and recovery.

Scanned with OKEN Scanner

Data Model

Data Model is a Collection of Conceptual tools for describing data, data relationships, data Semantics and Consistency Constraints. That means a data model provides a way to describe the design of a database

→ It is relatively simple representation, usually graphical of Complex real-world data Structures.

→ Data modeling is Considered as the most important part of the database design process.

Entity → An entity represents a particular type of object in the real world.

Entity set → Collection of similar type of entities

Attribute → characteristics | Properties of entities

Domain → Set of permitable values for an attribute

Constraints → A Constraint is a restriction placed on the data.

NULL value → non-existing data

Relationship → association b/w two entities

Relationship set → Set of similar type of relationship

Scanned with OKEN Scanner

Relationship Types :-

Many-to-Many

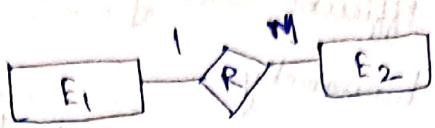
M : N

1 E_1 — N E_2

M E_1 — 1 E_2

One-to-Many

1 : M / M : 1



1 E_1 — M E_2

1 E_1 — 1 E_2

one-to-one

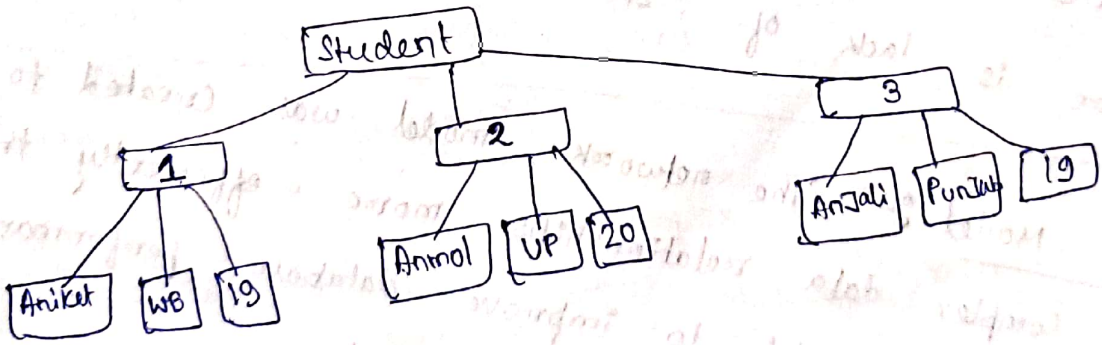
1 : 1

1 E_1 → 1 E_2

1 E_1 ← 1 E_2

⑧ Hierarchical Model :-

① developed in 1960s to manage large amount of data for complex manufacturing projects. Its basic logical structure is represented by a inverted tree



Scanned with OKEN Scanner

Hierarchical Model :-

Advantages :-

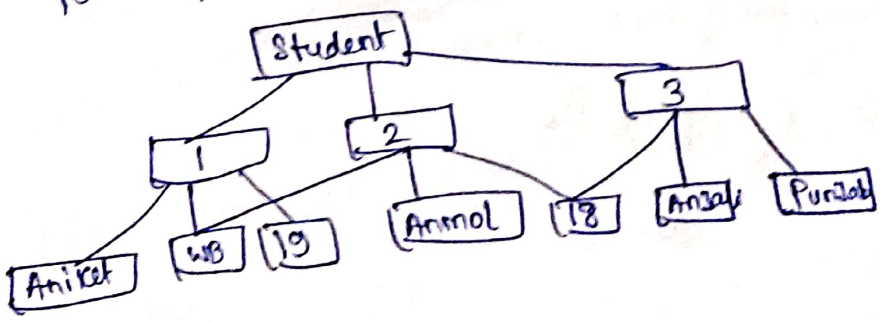
i) Easy to understand
ii) efficient with 1:M relationships
iii) Efficient Storage for data that have a clear hierarchy.
iv) promotes Conceptual Simplicity & data integrit
v) It promotes data Sharing.

Disadvantages :-

i) It is Complex to implement
ii) It is difficult to manage
iii) It Can't represent M:N relationships
iv) There is no DDL and DML
v) There is lack of standards.

---

② **Network Model :-** The network model was created to represent complex data relationships. more efficiently than the hierarchical model to improve database performance and to impose a database standard.



Scanned with OKEN Scanner

Advantages :-

i) It represents complex data relationships better than hierarchical models.

ii) It handles more relationship types such as M:N

iii) Data Redundancy is less

iv) Improved database performance

v) It includes DDL and DML

Disadvantages :-

i) System Complexity limits efficiency

ii) Navigation System yields complex implementation and management.

iii) Structural changes requires changes in all application programs.

iv) Database contains a Complex array of pointers that thread through a set of records.

v) Complex structure

vi) Networks can become chaotic unless planned carefully.

Scanned with OKEN Scanner

3) Relational Data Model :- 1970s - E.F. Codd

RDBMS table

It is implemented through RDBMS, which is easier to understand and implement.

The most important advantage of the RDBMS is its ability to hide the complexities of the relational model from the user.

Another reason for the relational data model's rise to dominance is its powerful and flexible query language. Generally, SQL is used for this purpose.

Advantages :-

i) changes in a table's structure do not affect data access or application programs.

ii) Tabular view substantially improves Conceptual Simplicity, thereby promoting easier database design, implementation, management and use.

iii) Have referential integrity Controls ensures data Consistency

iv) RDBMS isolates the end-users from physical level details and improves implementation and management simplicity

Disadvantages :-

i) Conceptual simplicity gives relatively untrained people the tools to use a good system poorly.

ii) It may promote islands of information problems as individuals and departments can easily develop their own applications.

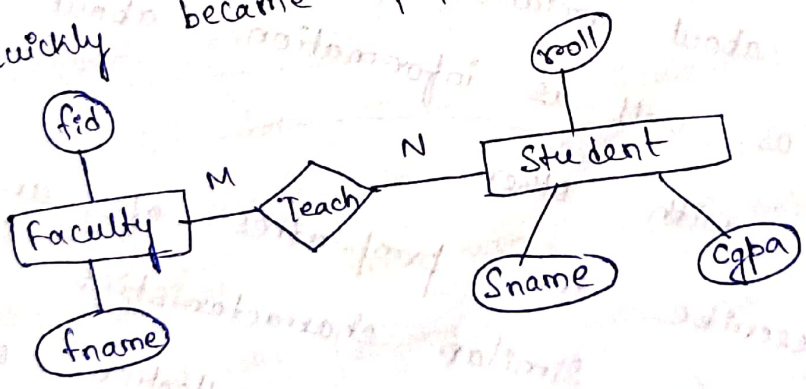
iii) Requires skilled knowledge to implement.

iv) Relational database software is expensive.

---

④ Entity-Relationship (ER) Model :- 1976 by peter chain

It was a graphical representation of entities and their relationships in a database structure that quickly became popular.



Advantages :-

i) Visual modelling yields exceptional conceptual simplicity.

ii) Visual representation makes it an effective communication tool.

iii) It is integrated with dominant relational model.

Scanned with OKEN Scanner

Disadvantages :- (ER model)

i) There is limited Contraint representation

ii) There is limited relationship representation.

iii) There is no DML.

iv) Loss of information Content when attributes are removed from entities to avoid Crowded displays.

---

⑤ Object Oriented (OO) Data MODEL :-

In object oriented data model, both data and their relationships are Contained in a single structure called an object. Like the relational model's entity an object is described by its factual Content. But quite unlike an entity, an object includes information about relationships b/w the facts within the object, as well as information about its relationships with other.

Attributes describe the properties of an object. Objects that share similar characteristics are grouped in classes. Thus a class is a collection of similar objects with shared structure and methods.

Scanned with OKEN Scanner

Advantages :-

i) Semantic Content is added.
ii) Support for Complex objects.
iii) visual representation includes Semantic Content.
iv) Inheritance promotes data integrity.

Disadvantages :-

i) It is a Complex navigational System.
ii) High System overheads Slow transactions.
iii) Slow development of standards caused vendors to supply their own enhancements, thus eliminating a widely accepted standard.

---

6) Object-Relational (OR) model :-

The object-oriented data model is somewhat spherical in nature, allowing access to unique elements anywhere within a database structures, with extremely high performance. But it performs extremely poorly when retrieving more than a single data item.

The relational data model is best suited for retrieval of groups of data, but can also be used to access unique data items fairly efficiently.

Object Relational = object oriented + relational

Scanned with OKEN Scanner

7) Semi Structured Model :-

The Semi-Structured data model permits the specification of data where individual data items of the same type may have different sets of attributes.

The XML (Extensible Markup Language) is widely used to represent Semi-Structured data. It supports unstructured data.

Scanned with OKEN Scanner

Level Abstraction of Database

The goal of ANSI/SPARC 3-level abstraction is to seperate the user applications and the physical database. It deals with the data, the relationship between them and the different access methods implemented on the database. The logical design of a database is called a **Schema**.

View level / External level

→ Highest level of abstraction

→ describes only specific part of database

→ At this level we have application programs which are accessed by user according to data required.

→ There are many different views of same data

Logical / conceptual Level

→ middle level of abstra".

→ describes what data is stored and what are their relationships

→ Programmers works at this level.

→ Data administrator use this level of abstraction to decide what information to keep in a database.

Physical / Internal Level

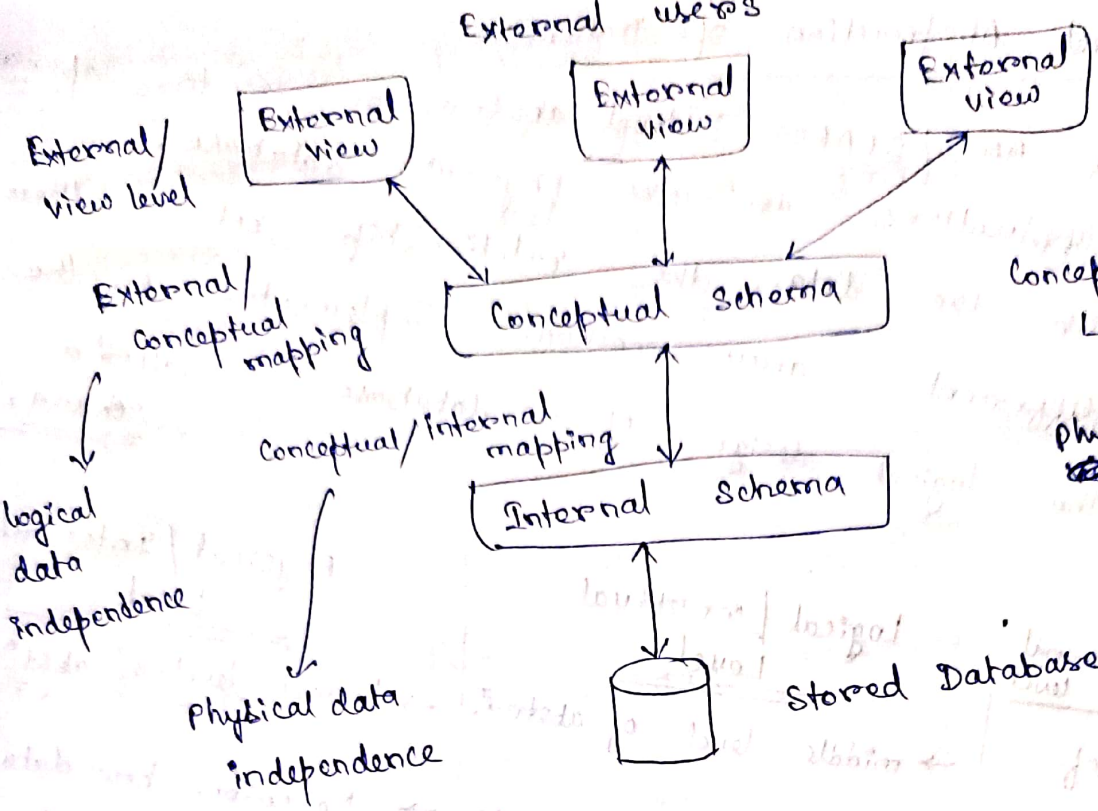
→ lowest level of abst".

→ Describes how data is actually stored

→ This level is kept hidden by compiler from programmer.

→ At this level the records can be described as block of storage (bytes, TB, GB)

Scanned with OKEN Scanner



Mapping → In a database system based on the 3-level architecture, each user group refers only to its own external schema. The process of transforming requests and results between different levels are called mapping.

Logical Data Independence :- It defines the correspondence between a particular external view and the conceptual view. Logical Data Independence indicates that the Conceptual Schema can be changed without affecting the existing external schemas.

Scanned with OKEN Scanner

Physical Data Independence :-

It defines the correspondence between the Conceptual view and the Stored database. Physical Data Independence indicates that the internal schema can be changed without any change to the Conceptual views and external views.

Database Users :-

Casual/Naive users :- They are the normal users who interact with the system by invoking application programs that have been written previously. (use external level)

Application Programmers :- They are computer professionals who write application programs to access data from the database. Application programmers can use different tools to develop user interfaces. (conceptual level)

Data Analyst/Sophisticated users :- They interact with the system without creating any application program. Rather, they form their requests in a database query language and submit each such query to a query processor. (Conceptual level)

Specialized users :- They are sophisticated users who write specialized database applications that don't fit into the traditional data processing framework.

Scanned with OKEN Scanner

⊛ Database Administrator (DBA) :-

The person who has central control of the whole database system is called DBA. The DBA coordinates all the activities of the database system.

- Roles of DBA are :-

i) DBA creates the original database schema by executing a set of DDL statements.

ii) DBA defines and controls the access methods for the different users.

iii) DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance.

iv) By granting different types of authorization, DBA can regulate, which parts of the database various users can access.

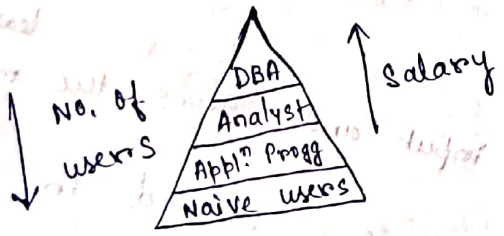
v) DBA specifies the different types of constraints to different tables or objects.

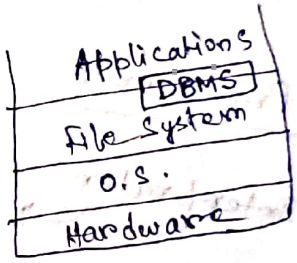
Scanned with OKEN Scanner

vi) DBA is responsible for the periodically backing up the database.

vii) DBA ensures that enough free disk space is available for normal operations and upgrading disk space as required.

viii) DBA monitors the Jobs running on the database and ensures that the performance is not degraded by very expensive tasks submitted by some users.





Database Architecture

Components :- * Storage Manager * Query Processor

AIBFT
- Authorization and Integrity Manager
- Buffer Manager
- File Manager
- Transaction Manager

Facts :-
- Data
- Index
- Data Dictionary
  - ↳ Contains the structure

Query Processor :-

* DDL Interpreter ↳ to read the high level code language
* DML Compiler ↳ takes the DML input and gives the machine learning language

★ The output of DDL interpreter is stored in the Data Dictionary. Data dictionary is never maintained by the user.

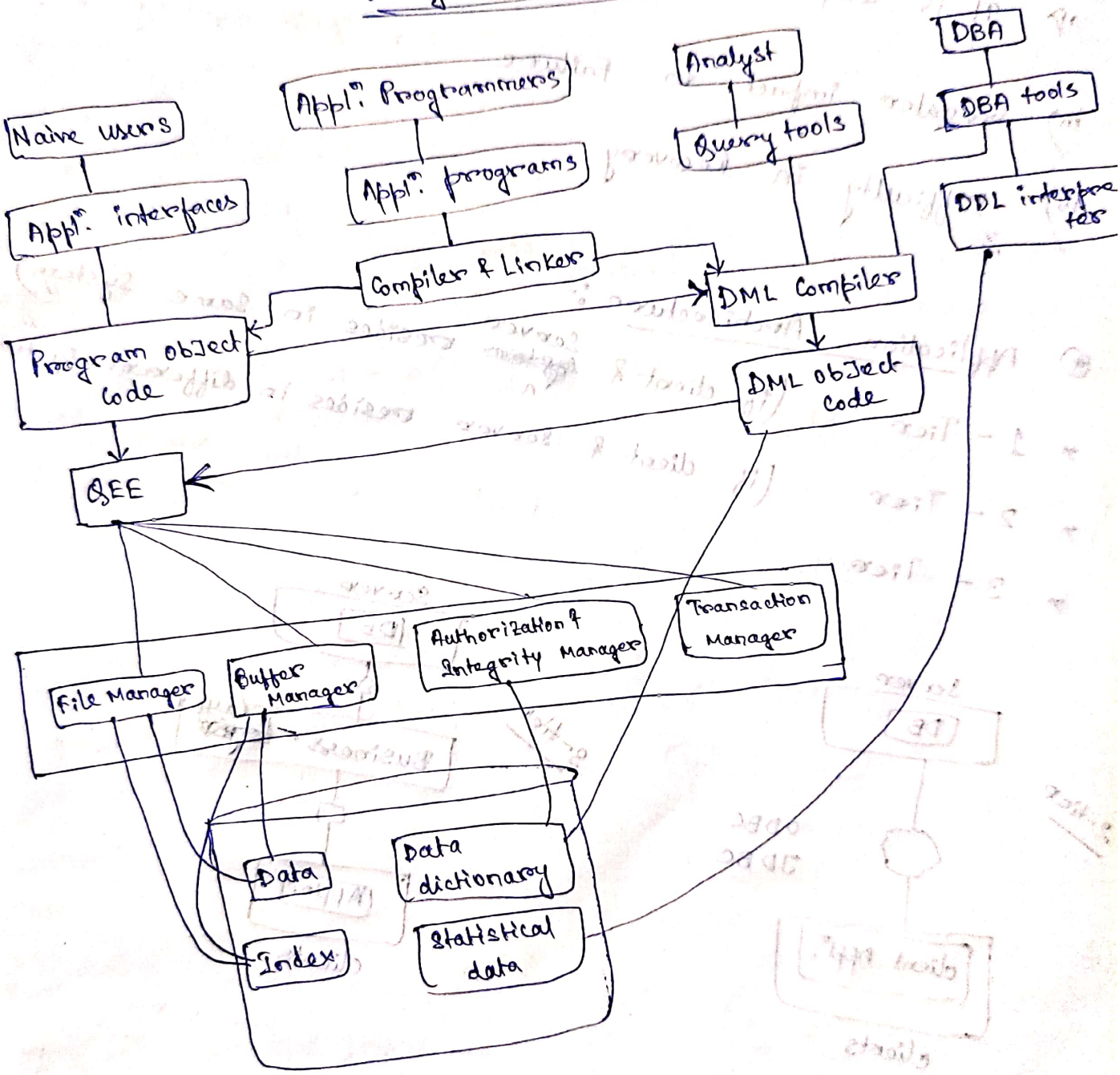
whenever any DML operation needs to be executed it will first consult with the data dictionary for the validitation purpose. If it is validated then the DML oper^{n} will be executed otherwise it will be declined.

Scanned with OKEN Scanner

* Query Evaluation Engine (QEE) :-

→ The optimal path is chosen for the execution of a Query.

Query Processor



Scanned with OKEN Scanner

Disadvantages of DBMS :-

i) Larger File Size (because we are storing both End user data and meta data)

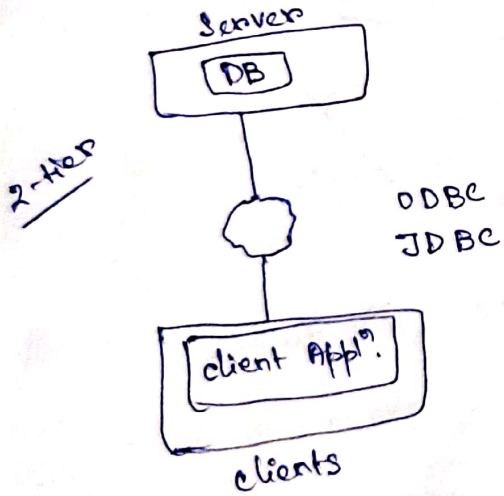
ii) It is Complex.

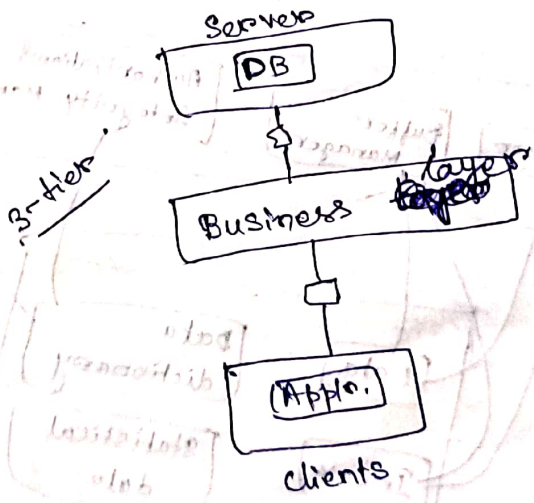
iii) Greater impact to Failure

iv) Difficulty in Recovery

Application Architecture :-

* 1 - Tier (if client & server resides in same system)
* 2 - Tier (if client & server resides in different system)
* 3 - Tier





Scanned with OKEN Scanner

⊛ Stages of DB design :-

Requirement (Text)

↓

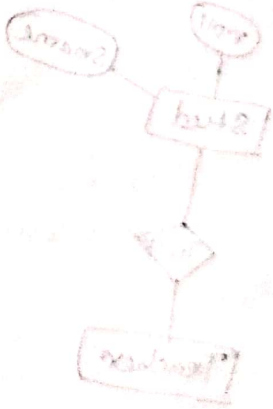
ER diagram

↓

Table / Schema

↺ Refinement (normalization)

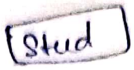
⊛ ER diagram must in Exam (VVI)



ENTITY - RELATIONSHIP DESIGN

Text → ER Diagram

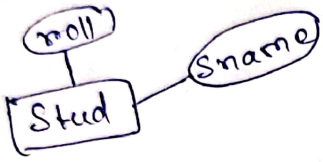
① Entity set :- In the text entity sets are the noun



Ex→ Teachers are teaching the students

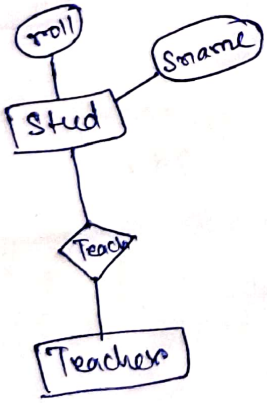
entity set

② Attributes :- represented by oval shape



③ Relationship :- It is identified by the verbs in the sentence.

represented by diamond shape

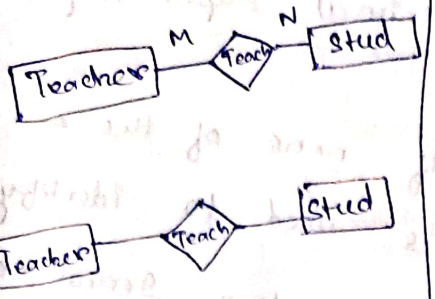


Employees are working in different departements; one employee works in a single department only

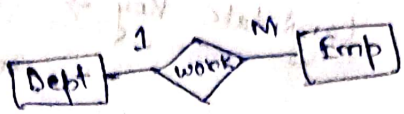
* Chain Notation
* Bechman "
  1: ↖
  M: —

Scanned with OKEN Scanner

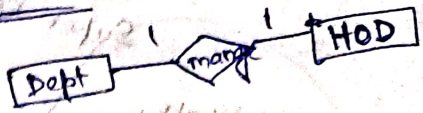
M : N

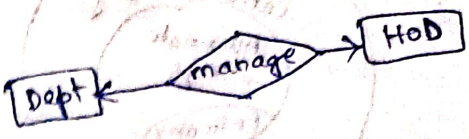


1 : M



1 : 1





KEY :-

roll	regd	name	city
1	12345	Adi	KOL
2	12346	dakesh	RMR
3	12347	rohit	Patna
91	12348	faizal	GKP
45	12349	rohit	Patna

A key allows us to identify a set of attributes that suffice (need) to distinguish entities from each other.

A Set of attributes used to uniquely identify the records (super keys)

Key

{roll}

{regd}

{roll, regd}

{roll, name}

{roll, city}

{regd, name}

{regd, city}

{roll, regd, name}

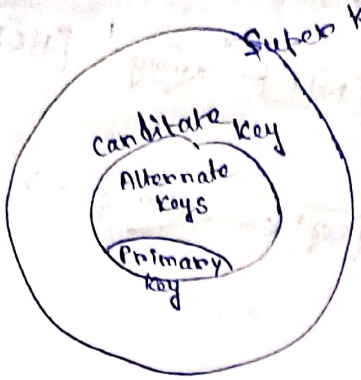
{roll, regd, city}

{roll, name, city}

{regd, name, city}

{roll, regd, name, city}

Scanned with OKEN Scanner



Candidate key → These are the minimal super keys

Surrogate key → If none of the attribute set is used to identify the records, a extra serial no is added which is known as Surrogate key.

Foreign key (Referential Integrity)

Foreign key is the set of attributes which is used for referring another entity set which have primary key.

PK

i) Atmost 1 PK.

ii) unique

iii) not NULL

iv)

FK

i) Any no. of FK

ii) not unique

iii) NULL

* Primary key is being represented by underline the attributes.

Foreign keys is never represented.

* Primary key is a candidate key that is chosen by the database designer as the principle means of identifying entities within a entity set.

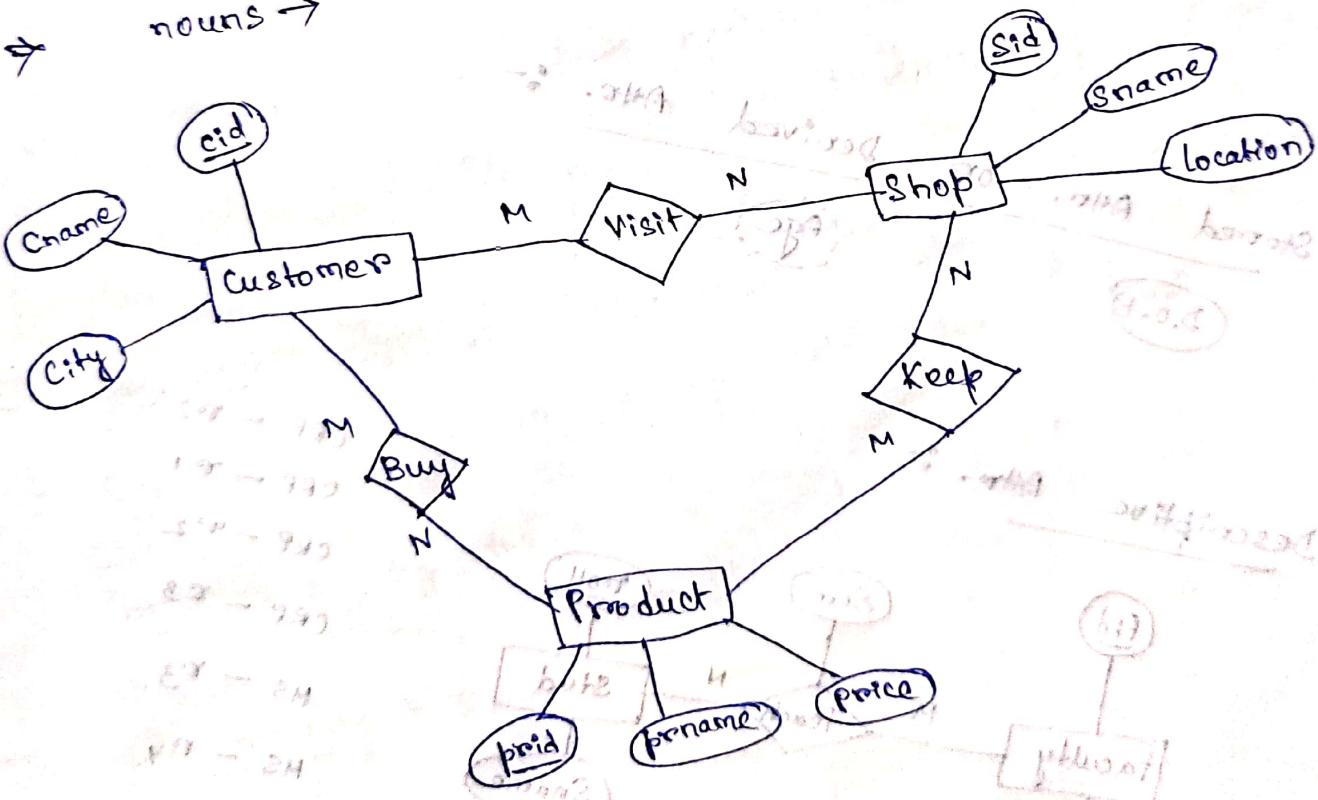
Scanned with OKEN Scanner

24/11

(*) Draw an ER diagram for the following -

1) There are multiple customers visiting the shops. Each customer has one unique CID along with Cname and City. similarly each shop has one unique Sid along with Sname and location. The customers are buying the products kept at different shops. for every product, unique pid, pname and price are stored.

* nouns → customers, shops, products

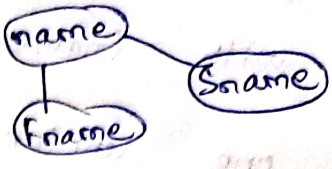


⊛ Types of Attributes :-

1) Simple Attribute or Composite Attribute :-

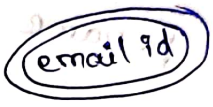
↳ which cannot be further decomposed

↳ which can be meaningfully decomposed



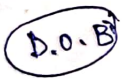
2) Single-valued Attr. or Multivalued Attr. :-

↳

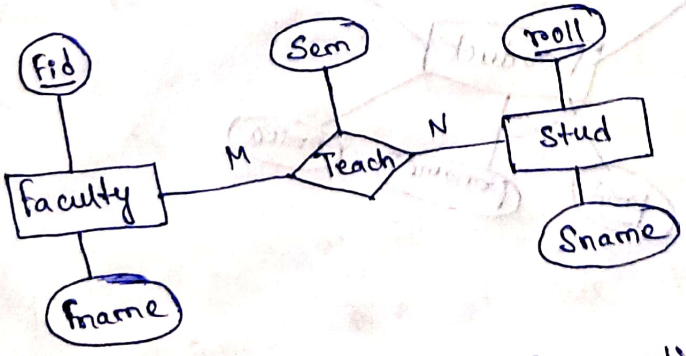


can have multiple value.

3) Stored Attr. or Derived Attr. :-



4) Descriptive Attr. :-

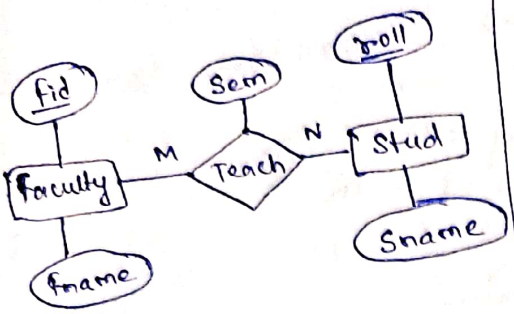


descriptive attr. describes more information about relation

CRP – r2
CRP – r1
CRP – r2
CRP – r3
MS – r3
MS – r4

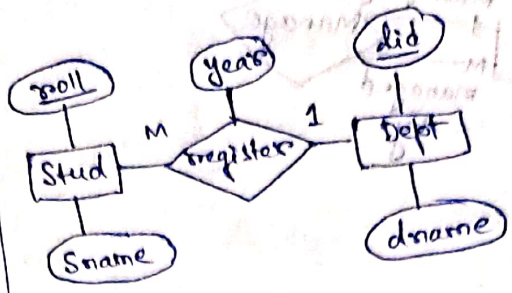
Keys of Relationships :-

M:N



uniquely identify union of primary key of associated set of entities.

1:M

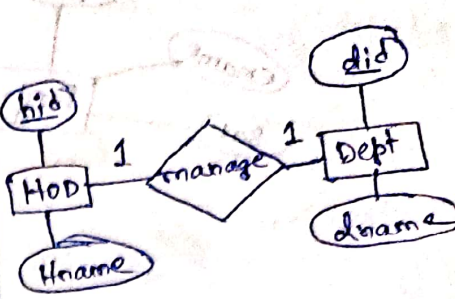


uniquely identified by roll

primary key will be on the many side.

If descriptive is there, you have to consider it

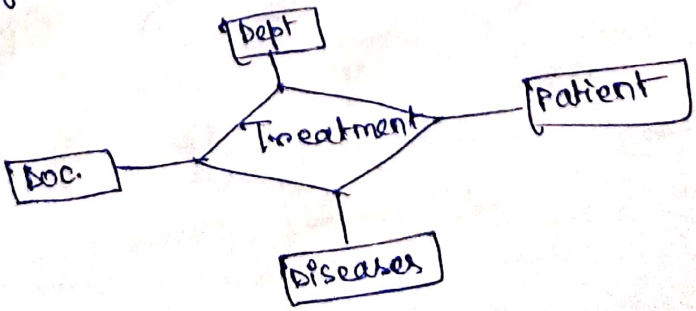
1:1



uniquely identified either by hid or by did.

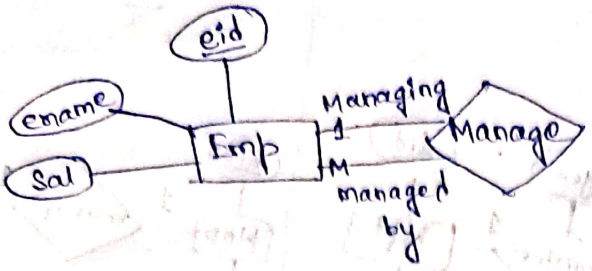
Relationship Types :-

* Binary → If two entity sets are associated
* Ternary → If three " " "
* Quaternary → If four " " "



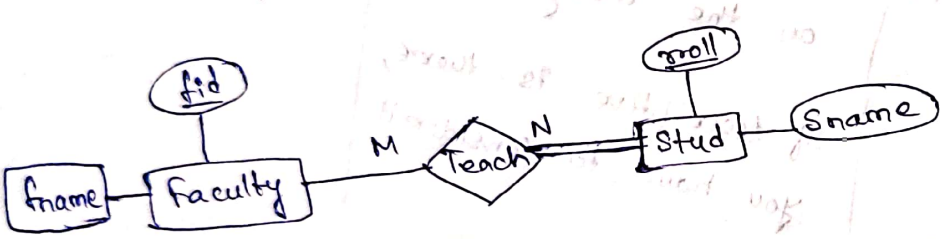
Scanned with OKEN Scanner

* Recursive Relationship :- Self referential integrity



* Participation Constraints :-

- Total / Full Participation
- Partial

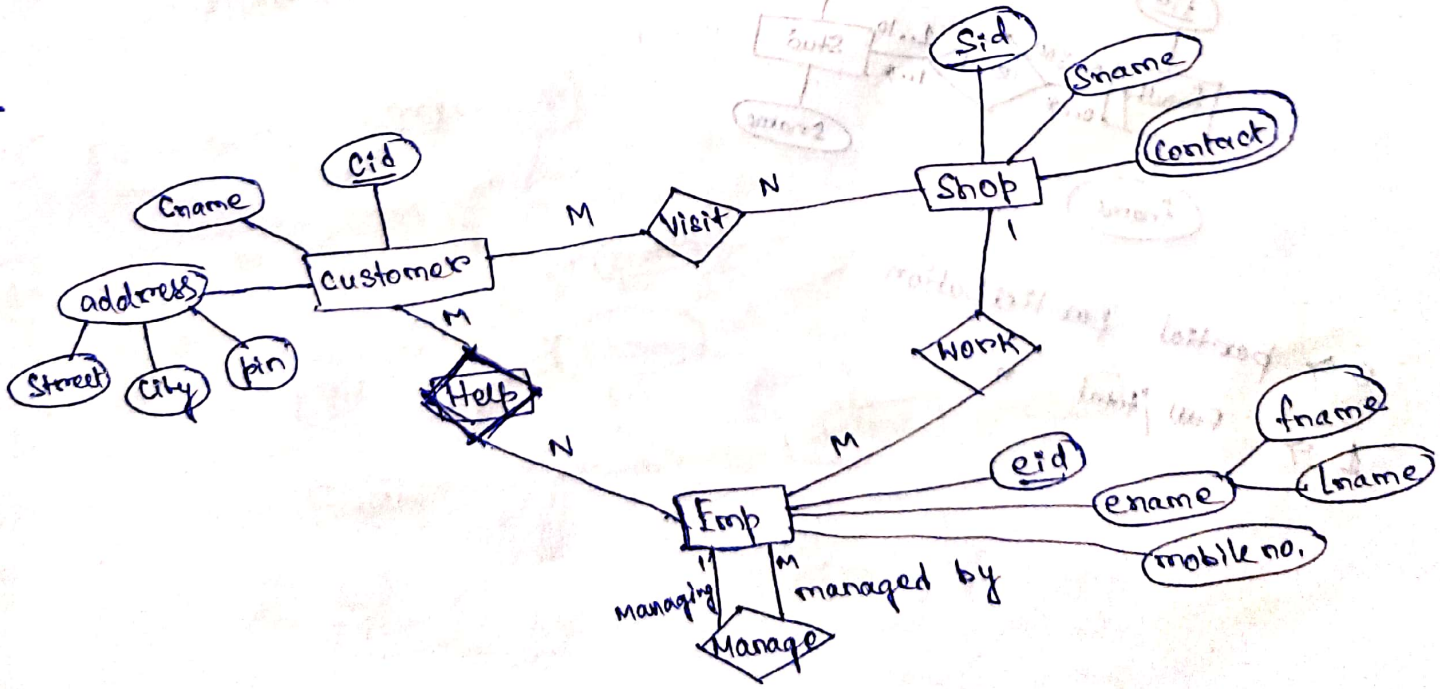


Total / Full → All entities of the entity set are participating.

Partial → All " " " " are not ".

⊛ Draw the ER diagram for the following :-

There are multiple customers visiting the shops. Each customer has one unique cid along with cname and address. (can be decomposed to street, city and pin) For every shop, unique sid, sname and contacts(s) are stored. Many employees (identified uniquely by eid, ename (can be decomposed to fname and lname) and mobile no.) are working in different shops; one employee can work in a single shop only. Out of the employees one employee is managing the other employees. Similarly the employees are helping the customers.

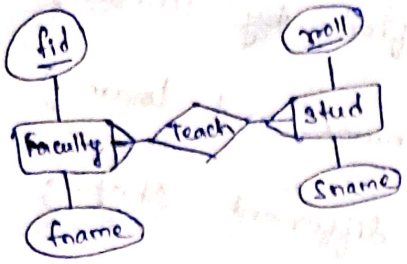


31/1/24

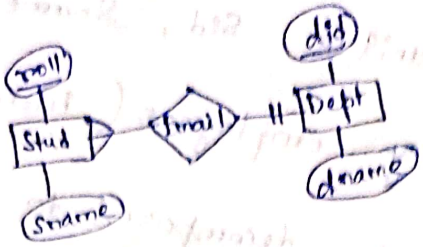
~~Codd's~~

* Crow's foot Notation :-

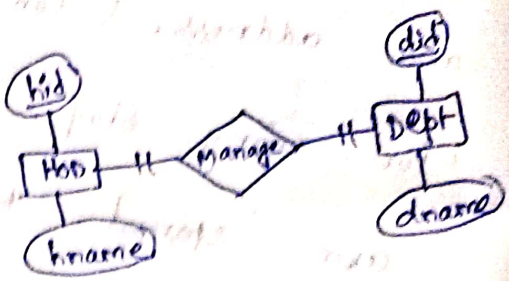
M:N



1:M



1:1



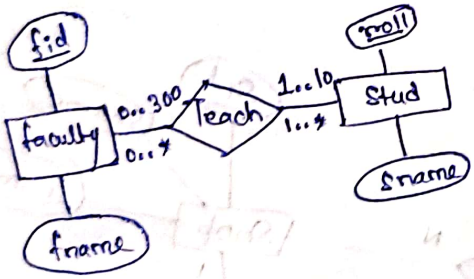
1 : ||

M : >

* Min representation :-

l..h

low ↙ ↘ high



0 ⇒ partial participation

1 ⇒ full / total "

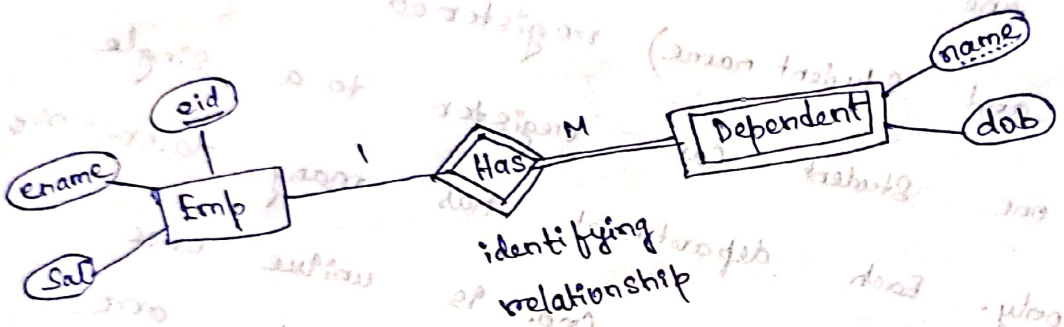
(*) Strong and weak entity Set :-

Strong entity set is the entity set which can exist alone, independtly.

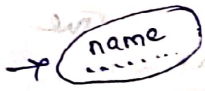
It is having a primary key.

weak entity set cannot exist alone. It is dependent.

There is no primary key in weak entity set.

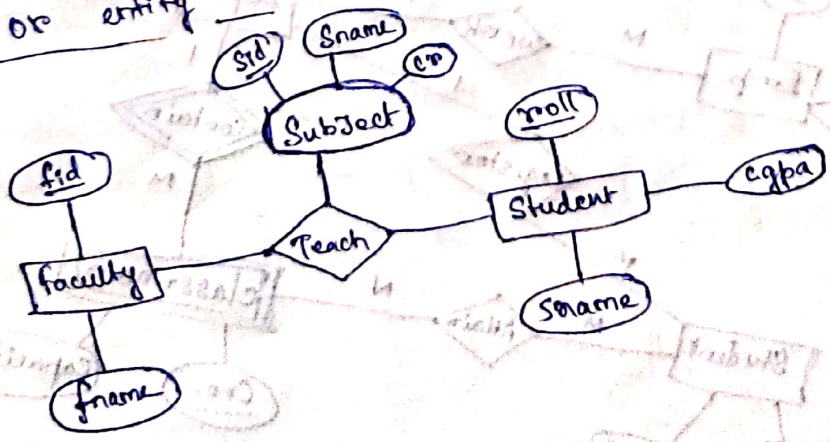


local key
or
Partial key
or
discriminator



(*) ER Design Issues :-

1. Attribute or entity set →

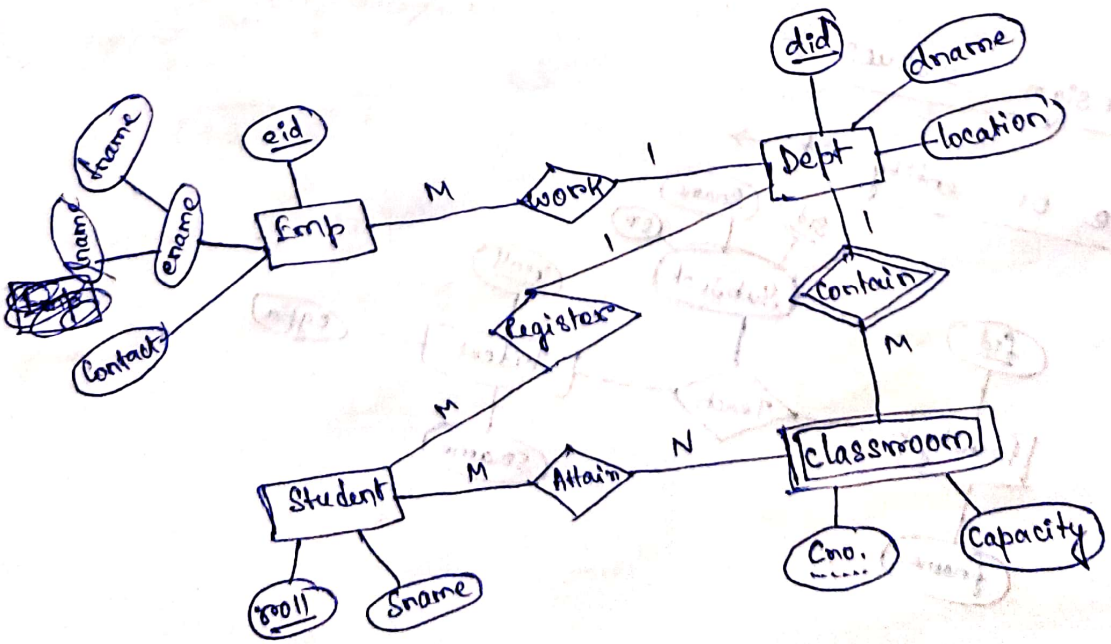


Scanned with OKEN Scanner

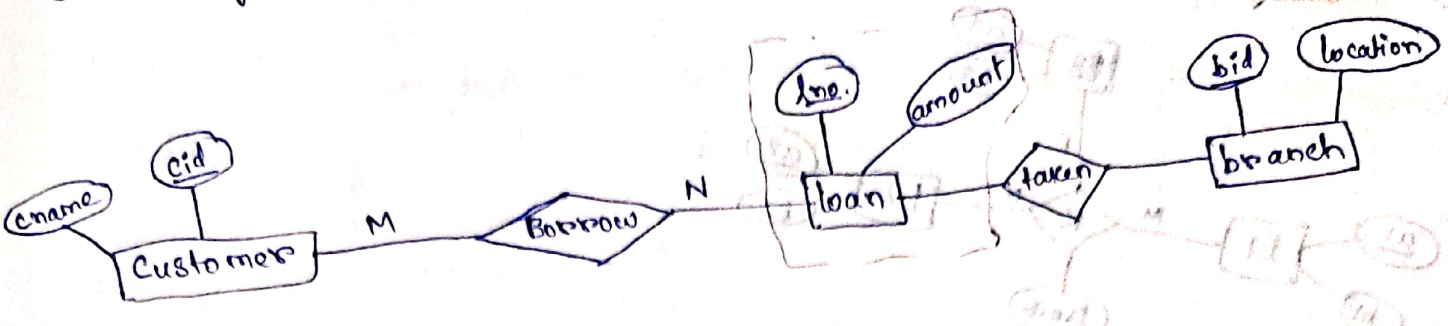
(*) Draw the ER diagram for one university. There are many employees working in different departments; one employee can work in a single department.

Each employee has one unique eid along with ename (can be decomposed into fname and lname) and Contact. Similarly each department has one unique did along with dname and location. Many students (identified by unique roll no. and student name) registered to different departments; one student can register to a single department only. Each department has many classrooms (with cno. and capacity); the cno. is unique once the department id is known. The students are attending classes in different classrooms.

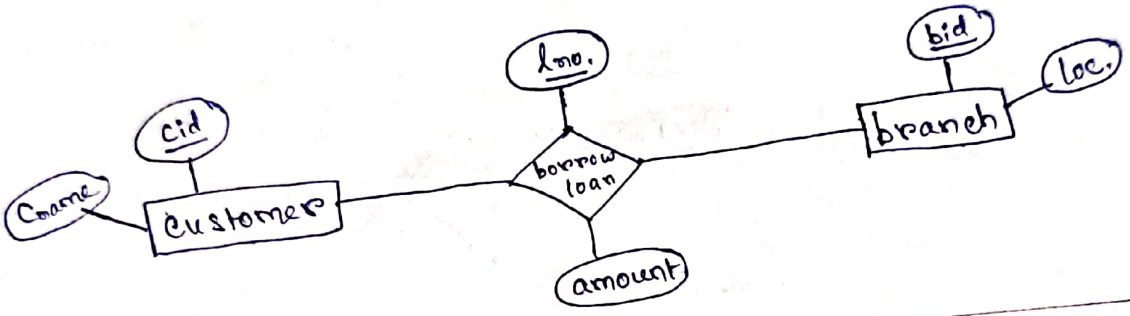
*



(2.) Entity set or Relationship :-

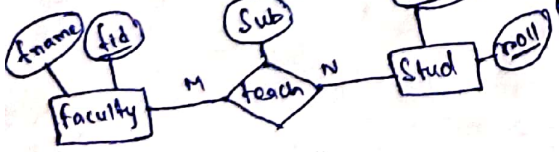


1 customer → 1 loan → 1 branch



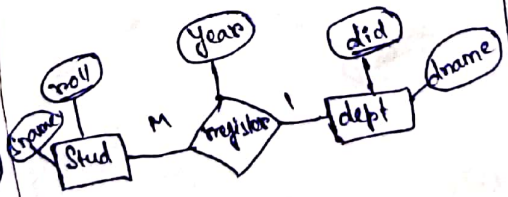
(3.) Placement or Descriptive Attributes :-

M:N



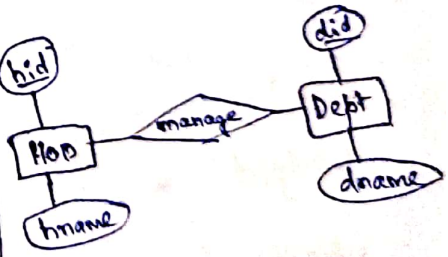
Descriptive attribute can't be placed in any of the entity set.

1:M



In the many side, the descriptive attribute can be placed.

1:1

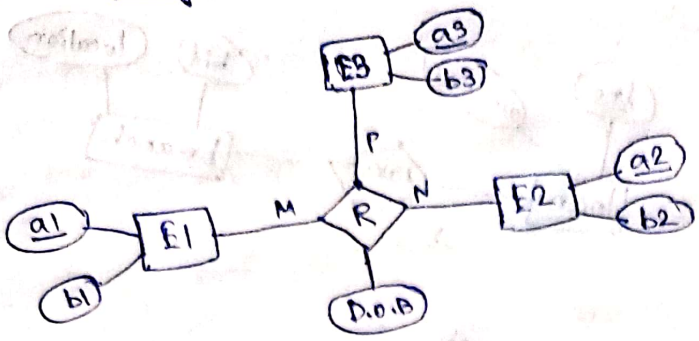


descriptive attribute can be put at either side

descriptive attribute is put along with the foreign key

Scanned with OKEN Scanner

Q. n-ary to binary relationship :-



(*) EER

↓

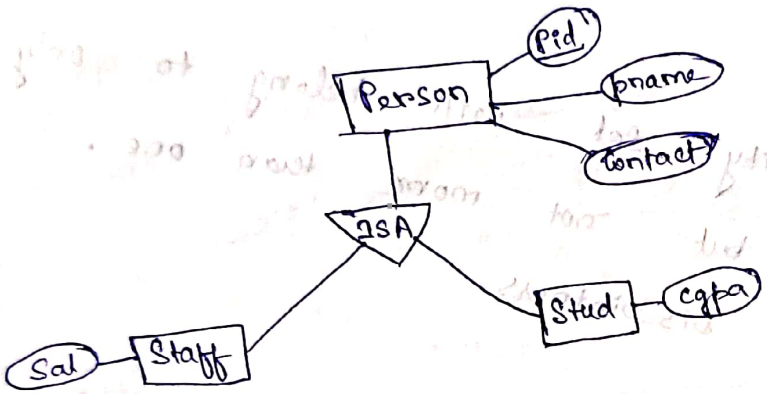
Extended or Enhanced

✓ Inheritance

✓ Abstraction

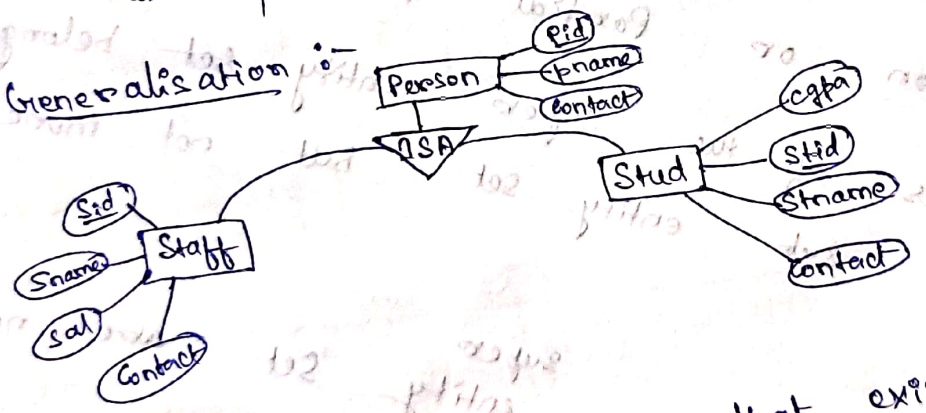
* Specialisation or Sub-grouping :-

The process of designating Sub-groupings within a entity set is known as Specialisation.



Sub entity set

* Generalisation :-



The Containment relationship that exists b/w a higher level entity set and one-or-more lower level entity set. is known as Generalisation.

Scanned with OKEN Scanner

Constraints on Generalisation / Specialization :-

* System defined Condition or * user defined

* Disjointness or overlapping

Overlapping means some entity of super entity set are belonging to more than one entity set.

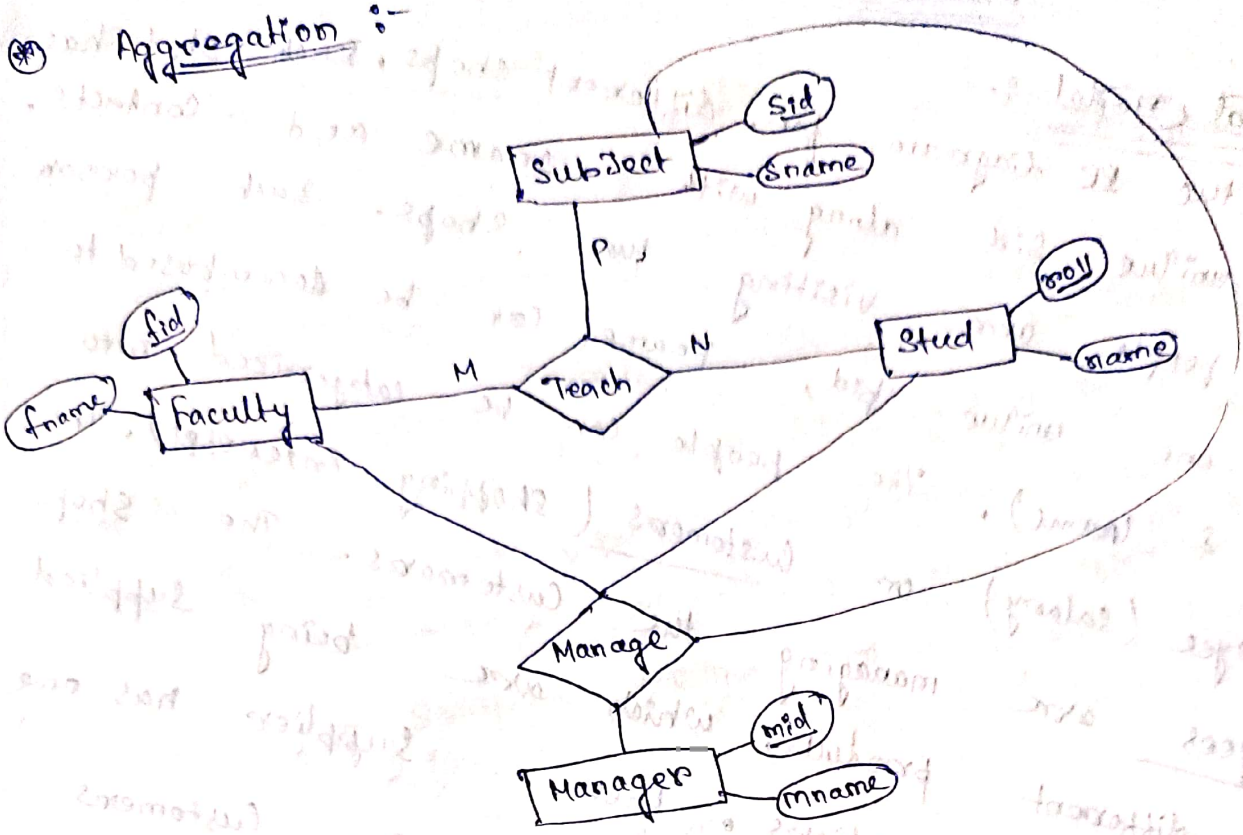
The entities of the super entity set will belong to only one sub entity set but not more than one. This is known as Disjointness.

* Total Participation or Partial Participation

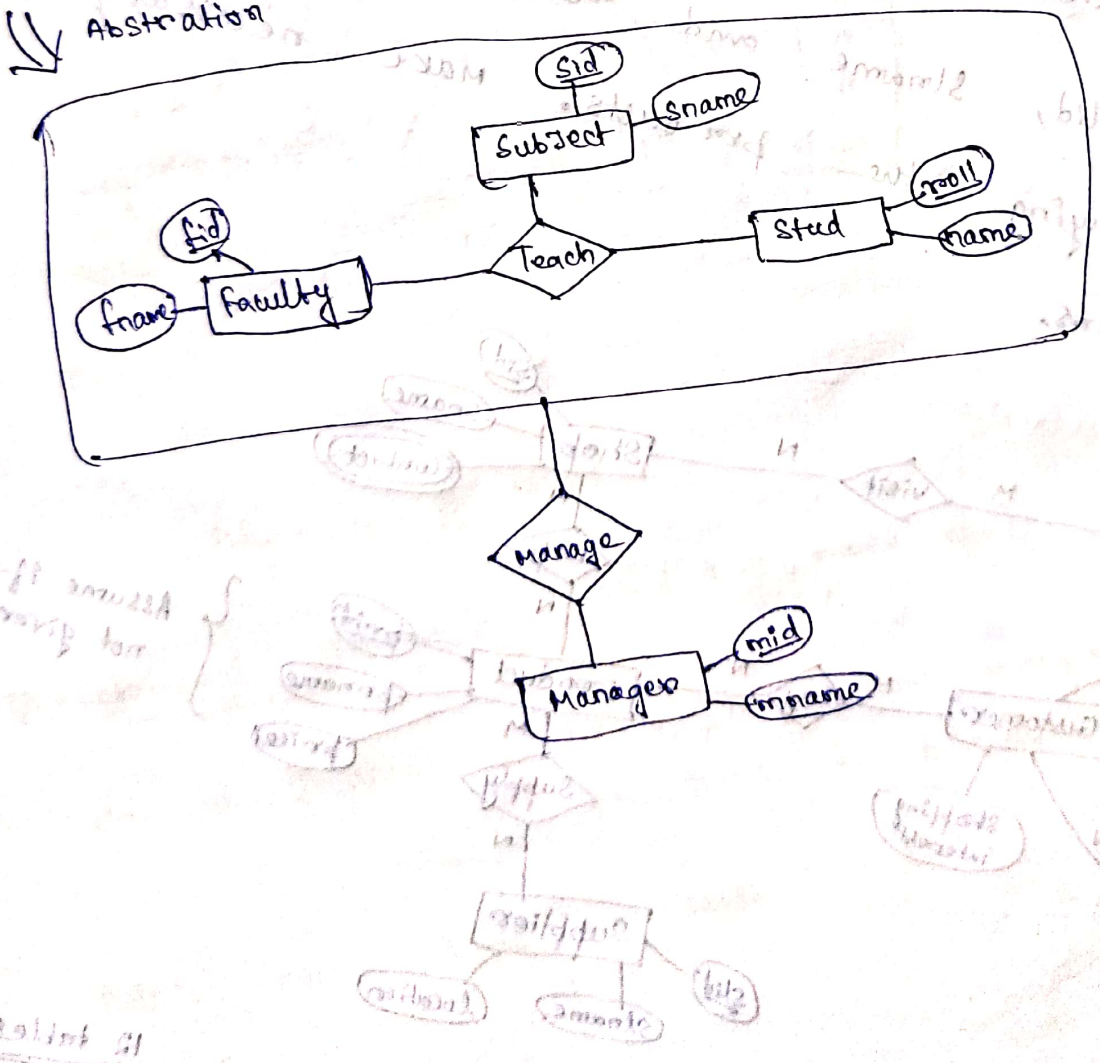
- All of the entities of the super entity set belongs to any of the sub entity set but not more than 1.

All the entities of the super entity set are not belonging to all the sub-entity set.

Aggregation :-



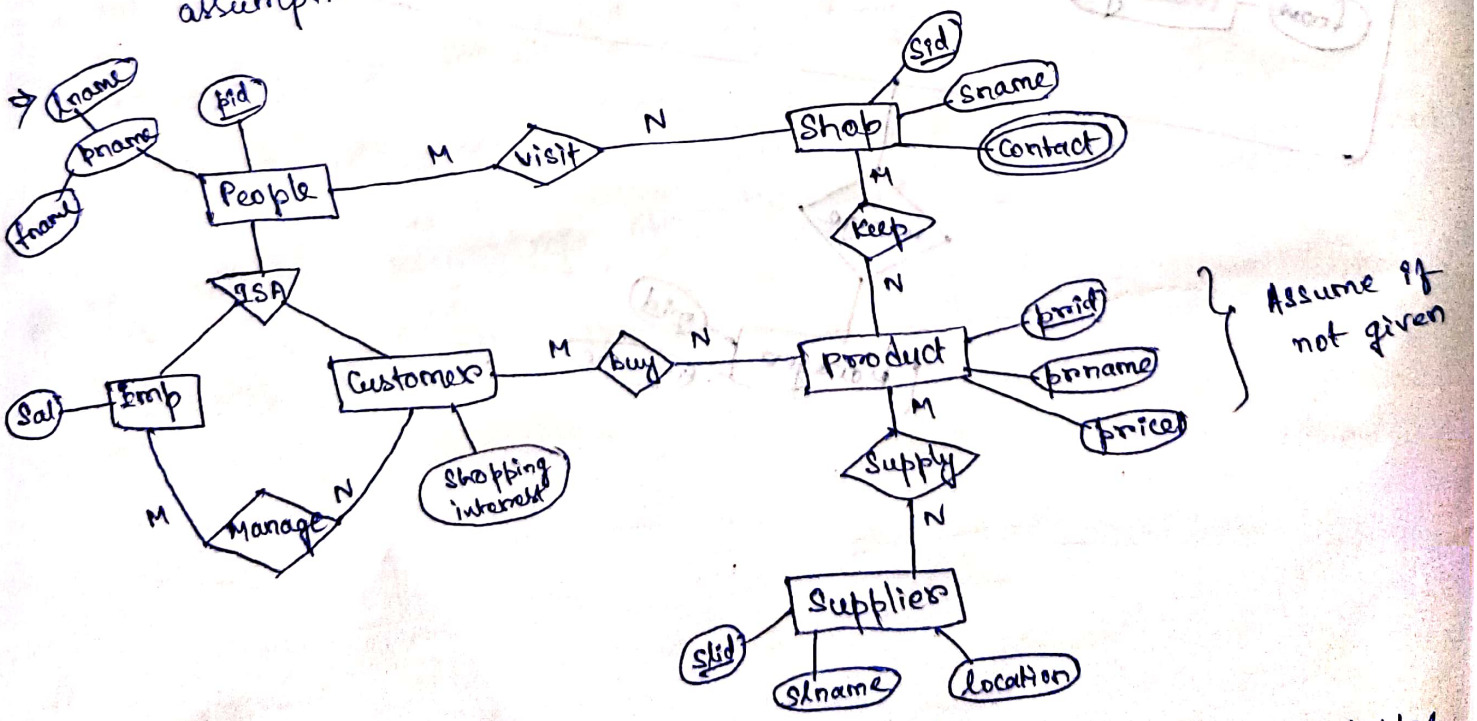
Abstration



Module - 2

Relational Model

* Draw the EE diagram for different shops. Each shop has one unique Sid along with Sname and contacts. Many people are visiting the shops. Each person has one unique pid, pname (can be decomposed to fname & lname). The people can be categorized into employee (salary) or customers (shopping interest). The employees are managing the customers. The shops keep different products, which are being supplied by different suppliers. Each supplier has one unique Slid, Slname and location. The customers are buying the products. Make necessary assumptions.



12 tables

7/2/24

Module-2

Relational Model :-

DDL

DML

DCL — GRANT (select permission granted for viewing only)
REVOKE (only select and deleted if certain user are revoked)

TCL — Commit (save it for permanent purpose for server not for client system)
Rollback (insertion for client buffer)
Savepoint (Rollback to S1, it will not go to initial state)

Entity - row / record

Entity set - Table

Attributes - columns

Domain -

Degree - No. of columns

Cardinality - No. of rows

Stud

roll	name	age
1	A	19
2	B	19
3	C	20

* Relational schema :- Stud(roll, name, age)

Relational Instance :- At a time how many records are there or a snapshot.

* Relation means it is the table

Relational Database :- E.F. codd developed 13 rules

i) for the relationship foreign key has to be used.

Relational Constraints :-

Candidate key → * uniqueness
* It should be minimal (Minimality)

Theory | Lab

i) Domain Constraint → by declaring the datatype and size

ii) Entity Integrity → by using primary key (unique key + not NULL)

iii) Referential Integrity → to maintain the relationship b/w two tables, using foreign key

iv) Business/operational Constraint → by using check constraint

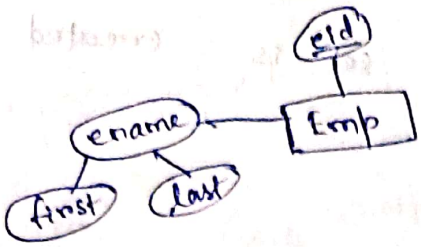
ER diagram to relational schema :-

i) Strong entity set →

Emp(eid, ename)

Scanned with OKEN Scanner

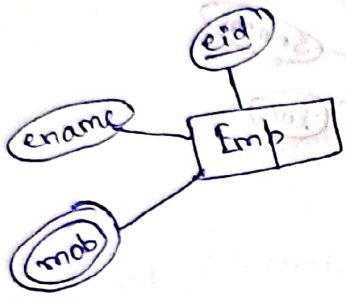
ii) Composite Attribute :-



Emp (eid, ename_first, ename_last)

or Emp (eid, first, last)

iii) Multivalued Attribute :- we have to create extra table for multivalued attribute.

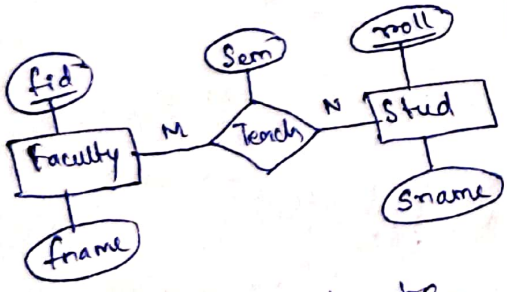


Emp (eid, ename)

Emp_mob (eid (fk), mob)

iv) Relationship :-

M:N



3 tables are to be constructed

Faculty (fid, fname)

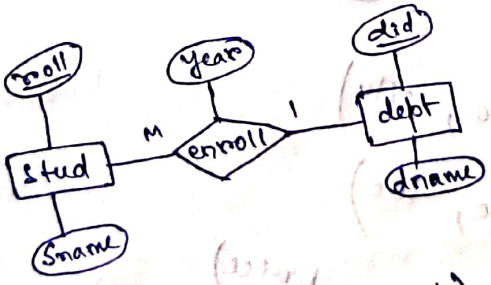
Stud (roll, Sname)

Teach (fid, roll, Sem)

f.k.

f.k

1:M



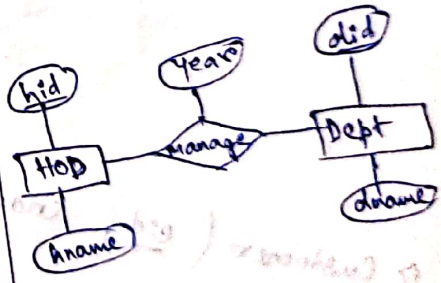
Stud (roll, Sname, did)

f.k.

dept (did, dname, year)

primary key in the one side is the foreign key for the many side.

1:1



HoD (hid, hname, did, year)

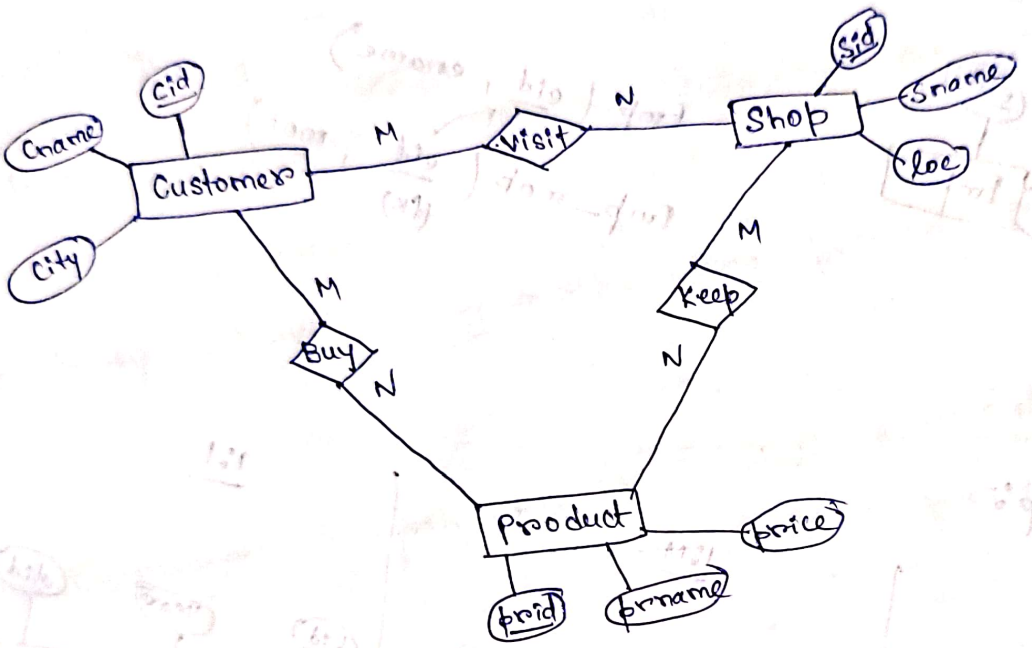
Dept (did, dname)

Scanned with OKEN Scanner

Special case :- If the table is 1:1 and is not connected to only one side, then a single entity set is created

Dept (did, dname, hid, hname)

(*)



i) Customer (cid, cname, city)

ii) Shop (sid, sname, loc)

iii) product (prid, prname, price)

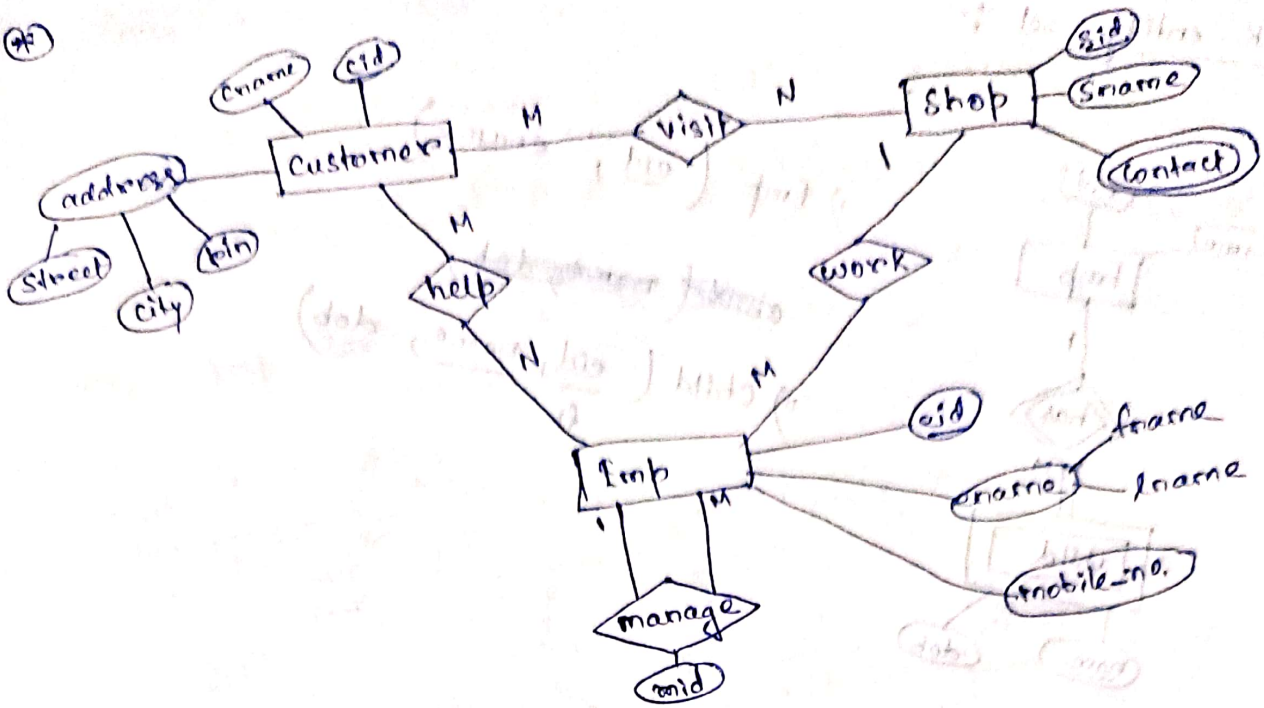
iv) visit (cid (fk), sid (fk))

v) Buy (cid (fk), prid (fk))

vi) Keep (sid (fk), prid (fk))

Scanned with OKEN Scanner

(*)



i) Customer(eid, Cname, Street, City, pin)

ii) Shop(Sid, Sname)

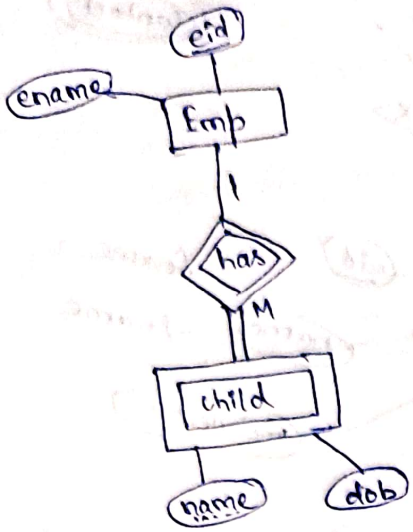
iii) Emp(eid, fname, lname, mob, mid f.k, Sid f.k)

iv) visit(cid f.k, Sid f.k)

v) help (cid f.k, eid f.k)

vi) Shop_Contact(Sid f.k, Contact)

v) weak entity Set :-

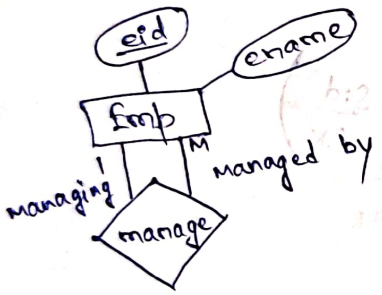


i) Emp (eid , ename)

~~child (name , dob)~~

ii) child (eid , name , dob)
fk

vi) Recursive Relationship :-



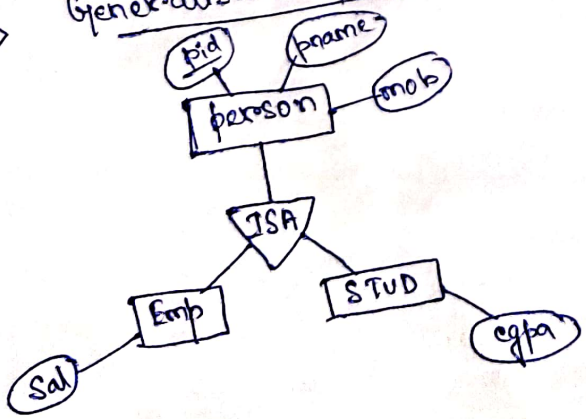
i) Emp (eid , ename)

ii) Manage (mid , eid)
fk fk

or we can go for a single table

• Emp (eid , ename , mid)
fk

vii) Generalisation | Specialization :-



i) person (pid , pname , mob)

ii) Emp (pid , sal) or (eid , sal)
fk (fk)

iii) STUD (pid , cgpa) or (std , cgpa)
(fk) (fk)

overlapping

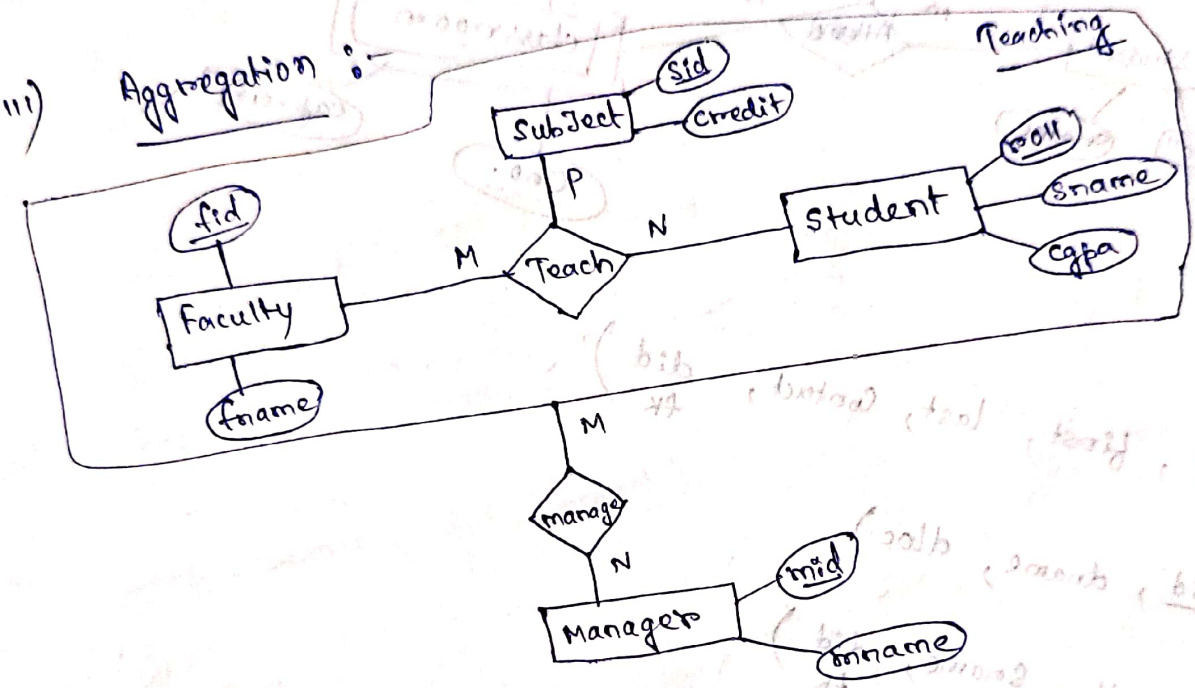
In case of disJoint

Emp(eid , ename, mob, sal)
Stud (sid , sname, mob, cgpa)

2 tables are sufficient

Here, Emp can't be a stud and stud can't be a Emp.

viii) Aggregation :-



i) Manager (mid , mname)

ii) manage (mid , fid , Sid , roll)
(fk) (fk) (fk) (fk)

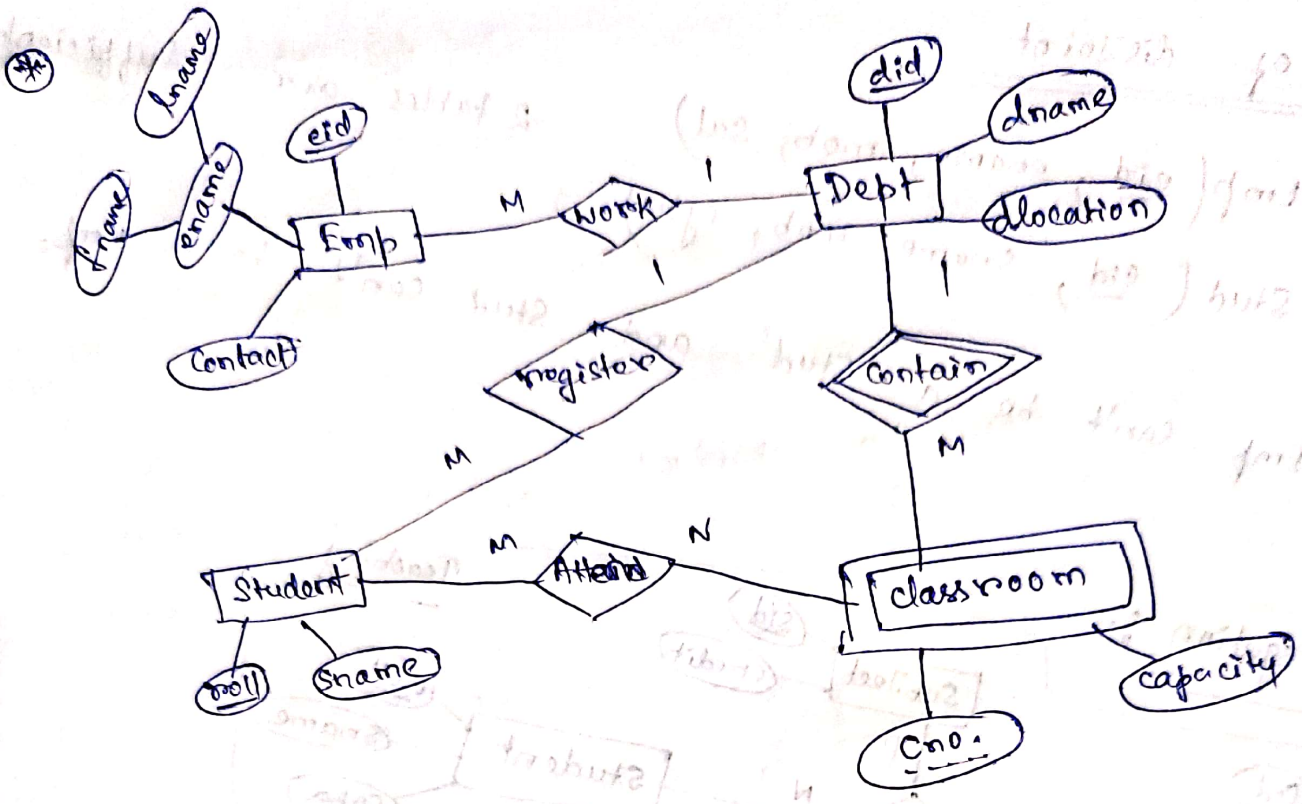
iii) faculty (fid , fname)

iv) student (roll , sname, cgpa)

v) SubJect (sid , credit)

vi) Teach (fid , sid , roll)
fk fk fk

Scanned with OKEN Scanner



Emp (eid , first, last, Contact, did)
fk

Dept (did , dname, dloc)

Student (roll , sname, did)
fk

classroom (did , cno. , capacity)
fk

Attend (did , cno , roll)
fk fk fk

(*) EER diagram

Tables :-

i) People (pid , first , last)

ii) Emp (eid , sal)
fk

iii) Customer (cid , interest)
fk

iv) manage (eid , cid)
fk fk

v) Shop (sid , Sname)

vi) Shop – Contact (sid , Contact)
fk

vii) Visit (pid , sid)
fk fk

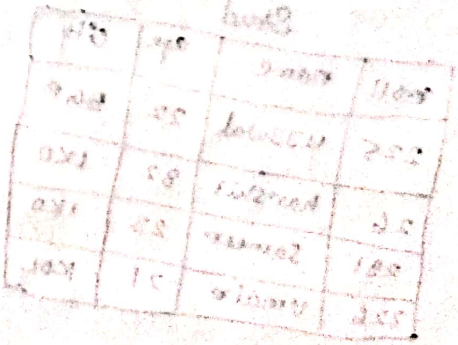
viii) Product (prid , proname , price)

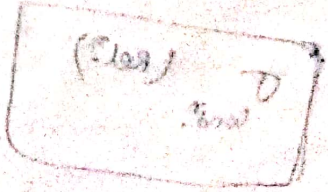
ix) keep (sid , prid)
fk fk

x) Buy (cid , prid)
fk fk

xi) Supplies (slid , slname , location)

xii) Supply (slid , prid)
fk fk





Scanned with OKEN Scanner

9/2/24

Query Language :-

- Procedural — Relational Algebra
- Non-Procedural — SQL
  - Relational calculus

Relational Algebra :-

Algebra operators
- Fundamental (6)
  - Unary (3) → (Selection, Projection, Rename)
  - Binary (3) → (union, Difference, cartesian product)
- Secondary (Intersection, Assignment, Join, Division)

① Selection (σ) :-

It is used to filter the records

Stud

roll	name	age	city
225	ujjwal	22	BGP
26	harshit	32	LKO
291	sameer	23	LKO
226	umair	21	KOL

$\sigma_{cond^n.}(Rel^n)$

Find the stud details where age is less than 25

→ select * from stud where age < 25

$\sigma_{age<25}(stud)$

$\sigma_{age<25 \wedge city = 'LKO'}$

(ii) Projection (π) :- used to filter the columns

Find the roll and name of students.

⇒ select roll, name from stud

$\pi_{attr}(rel^n)$ → $\pi_{roll,name}(stud)$

Composition :- Find the roll and name of students where age is less than 25.

⇒ select roll, name from stud where age < 25

$\pi_{roll,name,age}(\sigma_{age<25}(stud))$

* selection is always done before projection (optimal sol^n)

(iii) Rename :- (ρ) rho

to rename the column or table

changing the table name → $\rho_{Emp}(stud)$

" column → $\rho_{(eid, ename, age, city)}(stud)$

" both → $\rho_{Emp(eid, ename, age, city)}(stud)$

Scanned with OKEN Scanner

iv) Cartesian Product :- (X)

R x S (cross product)

Cardinality (R x S) = Cardinality (R) * Cardinality (S)

Degree (R x S) = degree (R) + degree (S)

R

a	b	c
1	A	10
2	B	20
3	C	30

S

a	e
10	ABC
50	DEF

R x S

R.a	b	c	S.a	e
1	A	10	10	ABC
1	A	10	50	DEF
2	B	20	10	ABC
2	B	20	50	DEF
3	C	30	10	ABC
3	C	30	50	DEF

Find the details of R x S where a values are same

R.a = S.a

↳ $\sigma_{R.a=S.a}(R \times S)$

Select * from R, S;

Select * from R, S where R.a = S.a;

↳ equiJoint ($\bowtie_=$)

Cond^n anything ThetaJoint ($\bowtie_\theta$)

Scanned with OKEN Scanner

v) Union :- (U)

Emp

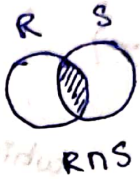
eid	ename	sal
1	A	10
2	B	20
3	C	15

Customer

cid	cname
2	A
4	D

RUS

R-S



RnS

Emp ∪ Customer ✗ we can't do that because the no. of data is not same

✓ $\Pi_{eid, ename}^{(Emp)} \cup \Pi_{cid, cname}^{(cus)}$

✓ $\Pi_{ename}^{(Emp)} \cup \Pi_{cname}^{(cus)}$

$\Pi_{eid, ename}^{(Emp)} \cup \Pi_{cname, cid}^{(cus)}$ ✗ → different datatype

Union Compatibility needs to be satisfied

R & S

↳ iff degree(R) = degree(S) and

domain(a_i) = domain(b_i) where $a_i \in R$, $b_i \in S$

Scanned with OKEN Scanner

* Find the ename along with the Cname together.

⇒ $\pi_{ename}(Emp) \cup \pi_{ename}(cus)$

Select ename from Emp UNION Select Cname from Cus;

vi) Difference :- (−)

Find the enames which are not present in Cname set.

⇒ $\pi_{ename}(Emp) - \pi_{cname}(cus)$

Select ename from Emp MINUS Select Cname from Cus;

vii) Intersection (∩) :-

• Find the enames which are also present in Cname set.

⇒ $\pi_{ename}(Emp) \cap \pi_{ename}(cus)$

Select ename from Emp INTERSECT Select Cname from Cus;

• Intersection can be derived from primary operators

$R \cap S = R - (R - S)$

$= S - (S - R)$

Scanned with OKEN Scanner

viii) Assignment (←) :- (assigning the value)

$A \leftarrow \pi_{ename}(Emp)$

$B \leftarrow \pi_{cname}(cus)$

$AnB \leftarrow \pi_{ename}(Emp) \cap \pi_{cname}(Cus)$

ix) Join ($\bowtie$) :-

Stud

Sid	Sname	age
1	A	19
2	B	20
3	C	20

Mark

Sid	Sem	sgpa
1	1	8.2
1	2	8.4
2	1	9.0
2	2	8.5

Join is a subset of cartesian product

Requirement for Join → Primary key & foreign key pairing must be present.

Stud ⋈ Mark

Sid	Sname	age	Sem	cgpa
1	A	19	1	8.2
1	A	19	2	8.4
2	B	20	1	9.0
2	B	20	2	8.5

Natural Join will give four records

• 12 records will be there (8×4) in cartesian Product

*) Division ($\div$) :-

$20 \div 6 = 3$ because 3 sets are present totally

(A)

Stud

Sid	Sname	age	city
1	A	19	BDS
2	B	20	KOL
3	C	20	BBS

(B)

Faculty

fid	fname
101	X
102	Y
103	Z
104	M

(C)

Teach

fid	sid
101	1
102	2
103	1
101	2

Result → 101
because it is involved with every student.

$C = A \div B$

* attr(A) = attr(C) + attr(B)
* records of A which are related to all records of B

⊛ find the fids who are teaching to every student.

→ $\Pi_{fid,sid}(Teach) \div \Pi_{sid}(stud)$

⊛ find the fnames who are teaching to every student.

→ $(faculty \bowtie Teach) \div \Pi_{sid}(stud)$

→ $\Pi_{fname,sid}$

Additional

xi) Generalised Projection :-

Find Sname and age+3 of stud.

⇒ Select Sname, age+3 from stud.

$\Pi_{Sname, age+3}$ (stud)

xii) Grouping :-

Find average age of stud.

⇒ Select AVG(age) from stud.

$\mathcal{G}_{AVG(age)}$ (stud)

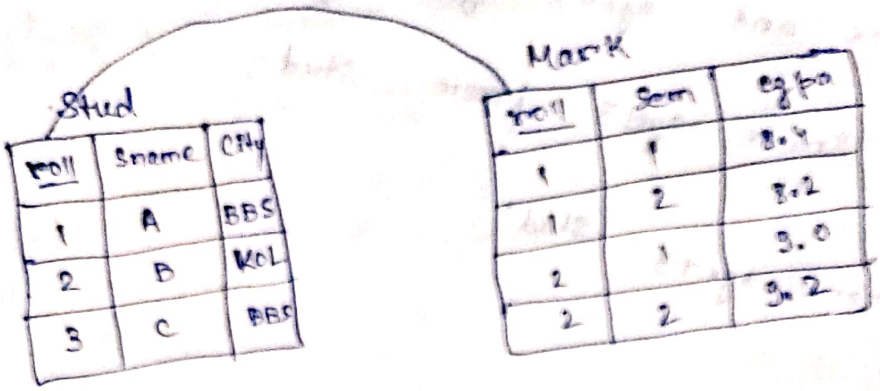
Find city-wise average age of stud.

⇒ $_{City}\mathcal{G}_{AVG(age)}$ (stud)

Select City, AVG(age) from stud GROUP BY City;

Scanned with OKEN Scanner

⊛ Join :-



Stud

roll	Sname	City
1	A	BBS
2	B	KOL
3	C	BBS

Mark

roll	Sem	cgpa
1	1	8.4
1	2	8.2
2	1	9.0
2	2	9.2

Stud X Mark

Stud-roll Sname City Mark-roll Sem cgpa

Inner Join → ↳ Theta Join ($\bowtie_\theta$)

$R \bowtie_\theta S = \sigma_\theta (R \times S)$

- Find the stud details along with their respective mark details.

⇒ Select * from Stud, mark where stud.roll = mark.roll

Stud ⋈ mark
stud.roll = mark.roll

$= \sigma_{stud.roll = mark.roll}$ (Stud x mark)

Scanned with OKEN Scanner

↳ Equi Join ($\bowtie_=$)

- Find Sname and their respective Sem & Sgpa.

⇒ Select Sname, Sem, Sgpa from Stud, mark where Stud.roll = mark.roll

$$\pi_{Sname, Sem, Sgpa}\left(\sigma_{Stud.roll = mark.roll}(Stud \times mark)\right)$$

↳ Natural Join ($\bowtie$)

Stud $\bowtie$ Mark Stud.roll = mark.roll

Equi Join

Stud.roll	Sname	city	mark.roll	Sem	Sgpa
1	A	BBS	1	1	8.4
1	A	BBS	1	2	8.2
2	B	KOL	2	1	9.0
2	B	KOL	2	2	9.2

Natural Join

roll	Sname	city	Sem	Sgpa
1	A	BBS	1	8.4
1	A	BBS	2	8.2
2	B	KOL	1	9.0
2	B	KOL	2	9.2

Scanned with OKEN Scanner

(*) Find the Sname and respective Sem and Sgpa of the students also include the students who do not have any marks.

⇒ for this we have to use outer Join outer Join is the superset of inner Join

Outer Join

- Left outer Join (⟕)
- Right " " (⟖)
- Full outer Join (⟗)

R ⟕ S

Select * from stud, mark where stud.roll = mark.roll(+);

or

Select * from stud left OUTER JOIN mark ON stud.roll = mark.roll;

Stud ⟕ mark

R ⟖ S

select * from stud, mark where stud.roll(+) = mark.roll;

R ⟗ S

select * from stud full OUTER JOIN mark ON stud.roll = mark.roll;

Self Join :- (Similar to theta Join)

Joining the same table to itself by creating two copies of it.

Emp

eid	ename	mid
1	BS	
2	CRP	1
3	AAA	1
4	MD	
5	MS	4

E_1 E_2

$E_1 \times E_2$

E_1.eid	E_1.ename	E_1.mid	E_2.eid	E_2.ename	E_2.mid
1	BS		1	BS	
1	BS		2	CRP	1
1	BS		3	AAA	1
1	BS		4	MD	
1					
2					

Q. Find the employee name who is managing 'CRP'.

→ select E_2.ename from Emp E_1, Emp E_2 where E_1.mid = E_2.eid and E_1.name = 'CRP';

$$\pi_{E_2.ename}\left(\sigma_{E_1.mid = E_2.eid \wedge E_1.ename = 'CRP'}\left(\rho_{E_1}(Emp) \times \rho_{E_2}(Emp)\right)\right)$$

or

$$\pi_{E_2.ename}\left(\sigma_{E_1.ename = 'CRP'}\left(\rho_{E_1}(Emp) \bowtie_{E_1.mid = E_2.eid} \rho_{E_2}(Emp)\right)\right)$$

Scanned with OKEN Scanner

⊛ Data Storage and Querying :-

The functional components of a database system can be broadly divided into :-

- The storage manager is important because databases typically require a large amount of storage space.
- The query processor is important because it helps the database system simplify and facilitate access to data.

⊛ Storage Manager :- A storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.

The storage manager is responsible for the interaction with the file manager. Storage manager is responsible for storing, retrieving and updating data in the database.

Scanned with OKEN Scanner

The storage manager component includes :-

i) Authorization and Integrity Manager :-

This module tests for the satisfaction of integrity constraints and checks the authority of users to access data.

ii) Buffer Manager :- Buffer manager is responsible for fetching data from the disk storage to MM. The buffer manager is a critical part of the storage system.

iii) File Manager :- This module manages the allocation of space on disk storage and data structures used to represent information stored on the disk.

iv) Transaction Manager :- Transaction Manager ensures that the database remains in a consistent state despite system failures and concurrent transaction executions proceed without conflicting.

Scanned with OKEN Scanner

- The storage manager implements several data structures as part of the physical system implementations:-
- Data files → These are files in the physical memory used to store the database itself.
- Data Dictionary → Data dictionary stores the metadata (data about data) that provides the information about the definitions of the data items and their relationships, authorizations and usage statistics.
- Indices → Indices are used to provide faster access to data items stored in the physical storage.

⊛ Query Processor :-

The work of query processor is to execute the query successfully.

- The major components of the query processor includes
  - DDL Interpreter
  - DML Compiler
  - Query Evaluation Engine

Scanned with OKEN Scanner

- DDL Interpreter → This is the interpreter used to interpret DDL statements and records the definitions in the data dictionary.

- DML Compiler → DML Compiler translates the DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands. When a user wants to perform a DML operation, the data dictionary has to be checked for the validation purpose.

- QEE :- This module executes the low-level instructions generated by the DML Compiler.

Scanned with OKEN Scanner