02-07-2024 0287

Data: Raw facts

Information: Processed Data (or) Meaningful Data

Data Base: Collection of interrelated data

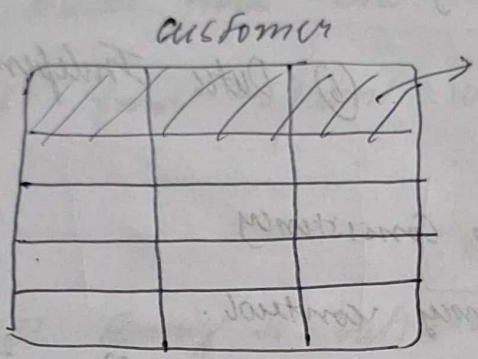
DBMS: A set of programs to store and retrieve in a convinient and efficient manner.

(Ex) MySQL, Oracle SQL, NoSQL

Relational DBMS: Relational Data Model

1450's

Table



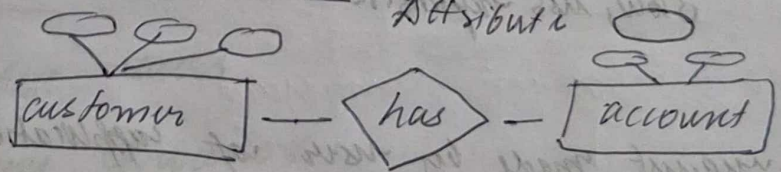
Applications of DBMS

1. Sales
2. Manufacturing
3. Banking
4. Educational
5. Finance
6. Transport Reservation
7. E Commerce
8. Social Media
9. Booking

E R

- Entity — Real World Objects
- Relationship
- Attribute

(Ex)



Data Models

Entity-Relationship, Relational, Hierarchical, Network, Object Oriented.

04-07-2024

*** File System Vs DBMS

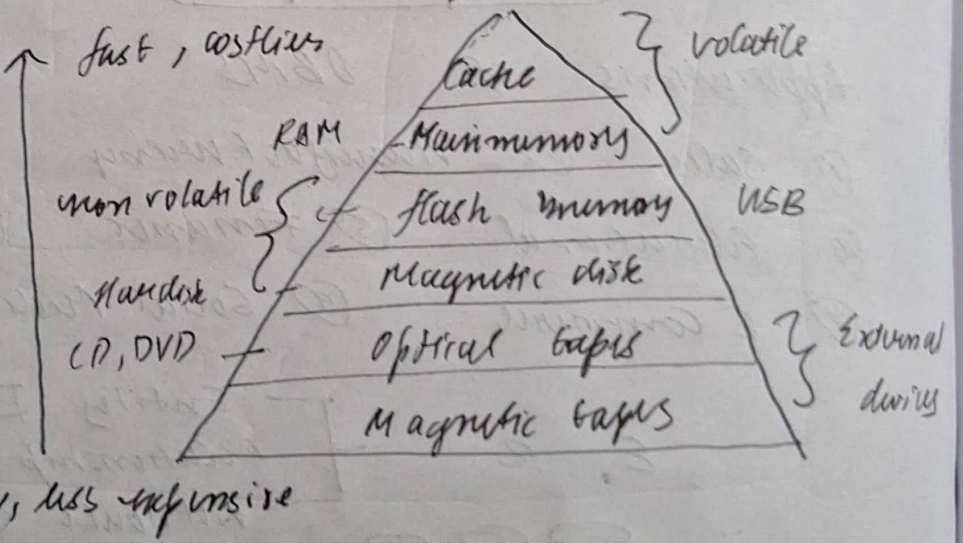
1. Data Redundancy & Inconsistency
2. Difficulty in accessing Data
3. Data Isolation
4. Atomicity Problems → completely done or not done at all
5. Integrity Problems
6. Concurrent Access Anomalies → Locking Protocols
7. Security Problems → Constraints & Data Abstraction

Advantages of DBMS

1. Reduced Data Redundancy
2. Backup & Recovery
3. Multiple User access
4. Maintaining Atomicity & Consistency
5. Maintaining concurrency control.
6. Data Independence.

Storage Data

1. Primary
2. Secondary
3. Tertiary



Queries

A Query is a request made by user or application program to store and retrieve the data from DB.

Basic Queries: 1. Select. 2. Insert 3. Update 4. Delete

① Select Select * / specific columns from table name;
where condition;

(a) Select * from Student where marks >= 92;
Select sid, sname from student;

② Insert Insert into table_name values (val1, val2, ...)

(a) Insert into student values (5, 'Smith', 85);

③ update update table_name set col = val where condition

(a) update Student set marks = 100 where srd = 2;

④ Delete Delete from table_name where condition;

(a) Delete from student where sname = 'Rai';

18-07-2024

(a) Write a SQL Query to retrieve all the details of students who scored more than 90%.

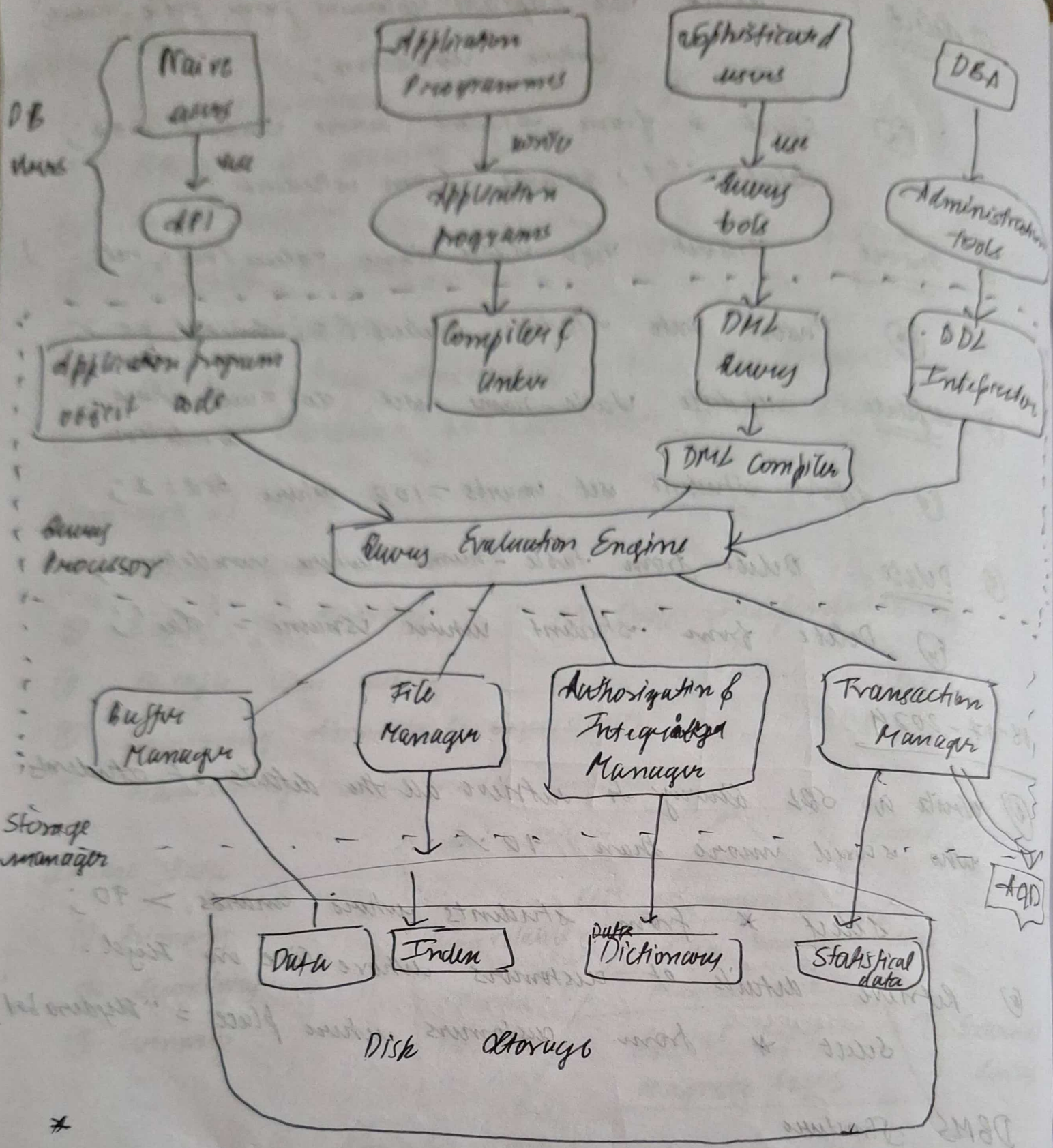
select * from students where marks > 90;

(b) Retrieve details of customers where live in Hyd.

select * from customers where place = "Hyderabad";

DBMS Structure

① Database users
② Query Processor
③ Storage Manager
④ Disc - storage.



DDL - Data Definition Language

SQL

DDL | DML | DCL | TCL

DDL: Create, Alter, Drop, (Terminate), Truncate

DML: Insert, update, Delete

DCL: grant, revoke

TCL: commit, savepoint

ACID

Atomicity

Consistency

META DATA

Data about data

types of DBs

- Relational DB
- Centralized DB – DB maintained as centralized system ex - central lib.
- Object oriented DB
- NoSQL DB or NonSQL DB
- Hierarchical DB
- Cloud DB
- Network DB
- Operational DB – creates & updates Data in real time

Adv consistency, good data quantity, less cost.

Disadv response time is more.

stored in terms of objects and it follows all the OOPS principles.

- Unstructured DB → stores data in diff ways.
  - → It can store wide range of datasets.
  - → key-value storage
    stores every single item as a key
  - → Document oriented DB
    store data as JSON type doc.
  - → Graph DB
    stores data in a graph data structure
  - → wide column store
    Data is stored in large columns

Adv : High scalability, quick data access, it can manage and handle large data sets.

- → Executes & handles daily data operations
- → Manages per day transactions

L Enterprise DB

→ Manage massive amount of data → parallel query execution
→ Improved efficiency → supports multiprocessors

L XML DB

→ used to store large amount of data in XML format.
→ Data stored in this DB can be queried by using XQuery.

L→ XML enabled DB
L→ Native XML DB

Data Independance

The ability to modify one schema without affecting the higher level schema is called data independence.

Schema is the overall structure of the relation / DB.

Ex) Employee (Eid / Ename / Esal / Dept)
↓ no ↓ varchar ↓ numb ↓ numb

Instance is a collection of values at a particular point of time or snapshot of time.

View of data

Can be done in 3 levels

View level / External / user → more security → (views)

$V_1, V_2 \ldots V_n$

Logical level / middle level (structure)

Physical level / Internal level

Indices (Faster access)

Views in DBMS

create view Emp_view as select E.id, Ename, dept
from employee;

DB Design steps

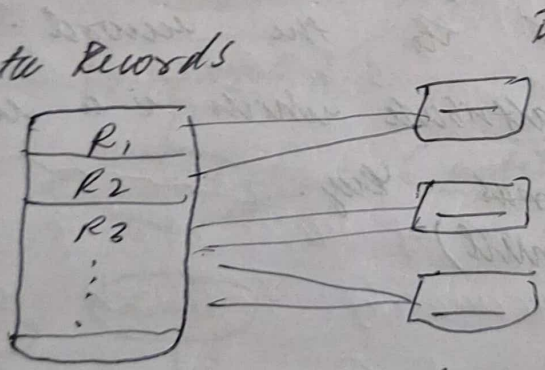
ERD Tool

① Requirement analysis ② Conceptual DB design
③ Logical ④ Schema refinement ⑤ Physical DBD
⑥ Application & security
Relational model (RDL) Implementation & testing

Overview of File Organization in DBMS

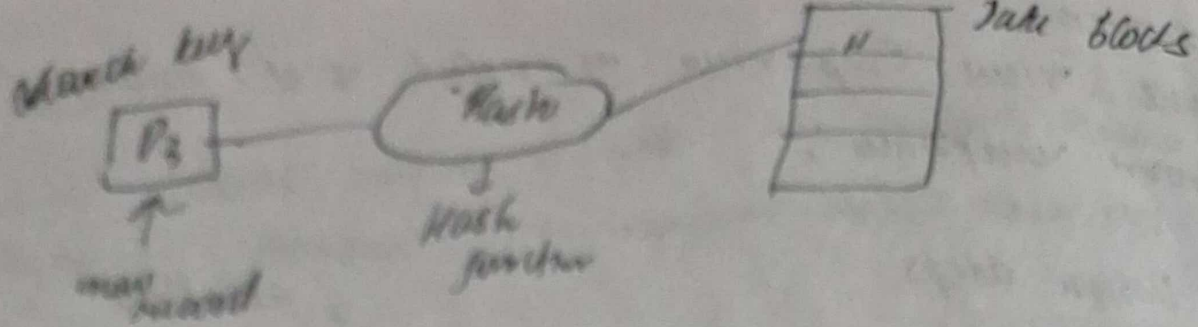
= File Organization methods:
- Sequential - Heap - Hash - Clustered
- ISAM
(Index Sequential Access Method)

HEAP FO - The files are not stored in any particular order. and new files are simply added at the end of the file.

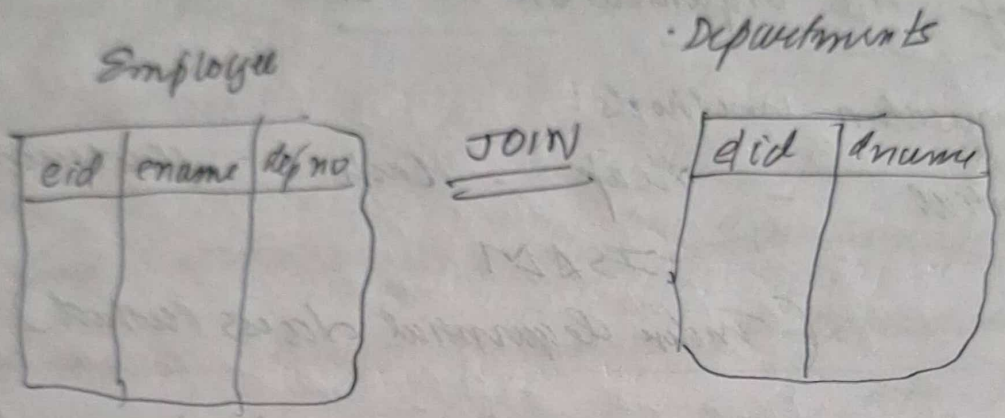


Record is placed where there is space
Data is stored in the form data blocks.

HASH FO Records are stored on table based on hash function. Hash indicates splitting of key into pieces. Records of each location are stored in seperate file.



Clustered FO :- In clustered FO, 2 or more related tables are stored within the same file, known as clusters. - It is used when there is a frequent need for joining the tables with the same condition.



ISAM FO : - IS an Advanced Sequential FO method. - In this, records are stored in file with the help of primary key. - For each primary key, an index value is created and mapped to the record.

Primary key is a key attribute which is a combination of unique and ~~NOT NULL~~ (not null) key.

DATA MODELS

- Data model describes how data is stored in DB. - It is an initial step in DB design to model a blueprint.

1. Flat model : Stored in one simple table.
   Adv: Easy
   Disadv: redundancy.

↳ ER ↳ Relational ↳ Hierarchical

↳ Network ↳ OO DM – the data in this data model is a collection of objects.

Additional features of ER Model

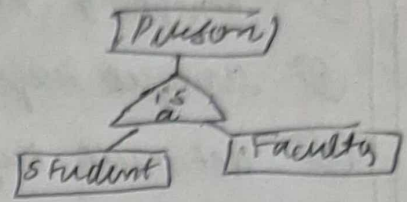
① Participation constraints. ⟶ Partial / Total participation

② Strong / Weak Entity

③ Generalization – Bottom up approach

④ Specialization

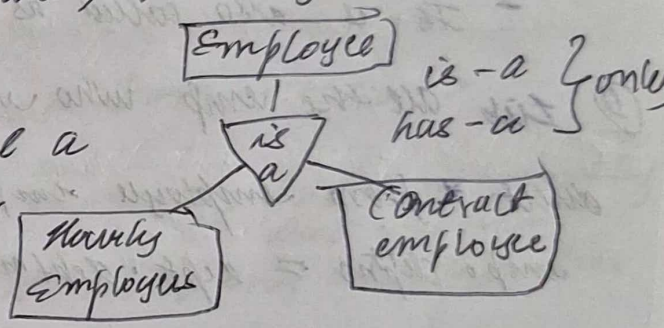
⑤ Aggregation.

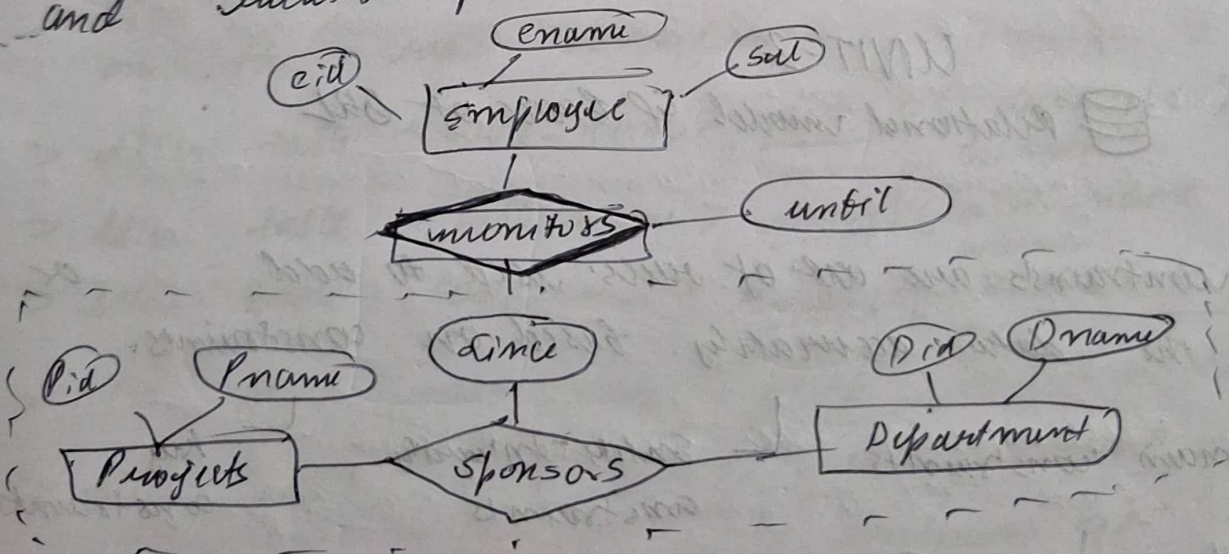


The common features of 2 entities are combined to form higher level entity.

It is a top-down approach – Here, the entity is divided into sub-entities.

Used when we want to model a relationship involving entity sets and relationship set.





Keys.

① super key : It is a super set. It can contain null values
② candidate key : It is a minimal set of super key.
③ Primary key : It is combination of not null key of unique key.
④ Alternate key : In the set of candidate key, the key other
⑤ Composite key : It can have a primary key with another attributes
⑥ unique key
⑦ Not null key
⑧ foreign key :- It is used when the user need information from diff relations.

Decided by DBA.

Then the primary key is called alternate key.

- It is used to join the tables based on common attributes.
- It is also called as (referential) integrity constraint

① List all the emp who work in sales dept.

select * from employee emp, department dept where emp.deptno = dept.deptno and dname = sales;

UNIT-2

Relational model of Basics of DBL

Integrity constraints are set of rules used to add or modify the data accurately based on constraints.

- Domain constraints
  - check
  - not null
- Entity-Integrity constraints
- key constraints
- Referential constraints.
  - foreign key

Basic DDL

└ DDL (Data Definition Lang) [Commited]

CREATE ALTER DROP RENAME TRUNCATE

└ used to create db or tables

> create database db_name;

> create table table_name (col1, datype, col2 datype, ...);

Write a query to create employee table with count of 8 attributes

> create database first;

> create table employee

(eid number, ename varchar(20), edob date,

ephone number, email varchar(50), equal varchar(20),

edesig varchar(30), esal integer);

copy of table

> create <table 2> as select * / cols from <table 1>

└ ALTER — {add, drop, modify, rename, add constraint, drop constraint}

The alter command is used to make any modification on the table structure.

> Alter table <table_name> add col datatype;

> Alter table <table_name> modify column col_name datatype;

> Alter table <table_name> rename column old_col to new_col;

> Alter table <table_name> drop column <col_name>;

> Alter table <table_name> add constraint <constraint_name> contraint type (col);

> Alter table <table_name> drop constraint <constra_name>;

└ DROP > drop table <table_name>;

└ RENAME > rename old_tablename to new_tablename;

└ TRUNCATE > truncate table <table_name>;

(drop only values, not structure)

> insert into <table-name> values (v1, v2) ... Vn);

(2) > insert into faculty values (1, 'raj', ...);

DML (Data manipulation Language) [uncommited]

INSERT — UPDATE — DELETE

INSERT
- used to insert values into the table.

> insert into <table-name> values (V1, V2, ... Vn);

1. insert values into specific columns

> insert into <table-name> (col1, col2, col3) values (v, v2 ... Vn

> insert into <table-name> select * from <table-name>;

Inserting specific columns from other table

> insert into <table-name> (col1, col2, coln) select col1, col2, coln from <table-name>;

2. Insert values at runtime

> insert into <table-name> values (&col1, &col2, ... &coln);

UPDATE
- used to update a single record or multiple record based on a condition.

> update <table-name> set column = value where condition;

DELETE used to delete a single record or multiple records based on a condition.

> ~~update~~ ~~table-name~~ ~~set~~ ~~column~~ = ~~value~~

> delete from <table-name> where condition;

- Difference b/w truncate & delete →
  - Delete statement is used to remove single or multiple records based on specific condition - [DML]
  - Truncate command removes complete data from the table but not the table itself. [DDL]

— Truncate of works much faster than delete which deletes row by row.

Create table using constraints

Constraints are set of rules supplied to store the data

Constraints are defined at two levels
- Column level constraints
- Table level constraints

column level

① > create table <table_name> (col1 dt1 constraint type, col2 dt2 constraint type, ...);

table level

② > create table <table_name> (col1 dt1, col2 dt2, ... ; constraint type (col_name));

③ Write a command to create customer table using column level constraints.

④ write a command to create passenger table using table level constraints.

- create table wh (wid number primary key, wsize number not null, wtag varchar(20) not null);

- create table wh (wid number, wsize number, wtag varchar(20) references tags (wtag)), primary key (wid), not null (wsize));

create table sailors (sid integer primary key,
sname varchar(100) not null, rating number)

create table reserves (sid integer not null,
bid integer not null, day date),
references sailors(sid),
not null
references boats(bid)

create table boats (bid integer primary key,
bname varchar(100) not null, color varchar(20)

DCL : Data Control Language

- Grant - Revoke

CREATE USER

create user <user-name> identified by <pass>;

Grant syntax

> Grant privilege_type [(column-list)] on object_type
object_name to user;

grant select on student to u1;

Revoke syntax

> revoke [grant option for] privilege_type
[(column-list)] on [object_type] object_name
from <user_name>;

Revoke select on student from u1;

TCL - Transaction control

A transaction is sequence of steps which performs some tasks.

- Commit - Rollback - Save point
- To save permanently in database
- commit changes to database
- Goes to previous save-point

Savepoint <savepoint_name>

rollback to <savepoint_name>

LOGICAL DATABASE DESIGN

Logical Database Design (conversion of ER model to Relational Model)

(1) Convert strong entity to table / relation
(2) Convert weak entity to table / relation
(3) " Relationship " "
(4) " entity with cardinality ratio to tables
(5) " " " " " & total participation
(6) Convert specialization / Generalization to tables
(7) " Aggregation " "

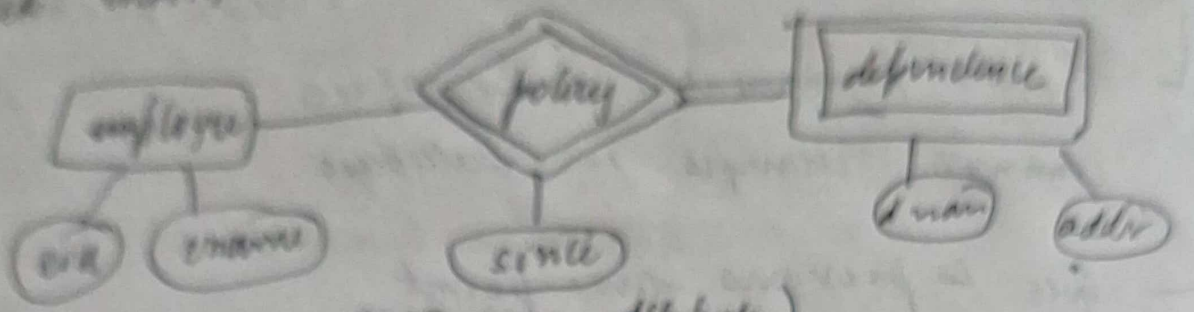
II

- Entity becomes relation
- Ignore derived attribute
- Simple valued or (simple) attribute will (become) become an attribute in a relation
- Key attribute is considered as primary key column in the relation. Ignore the composite attribute. Ignore the composite attribute, rename name and add

only simple attribute like fname, lname, addr.

- A weak entity should be considered for recording multivalue attribute.

[1] Weak entity to table

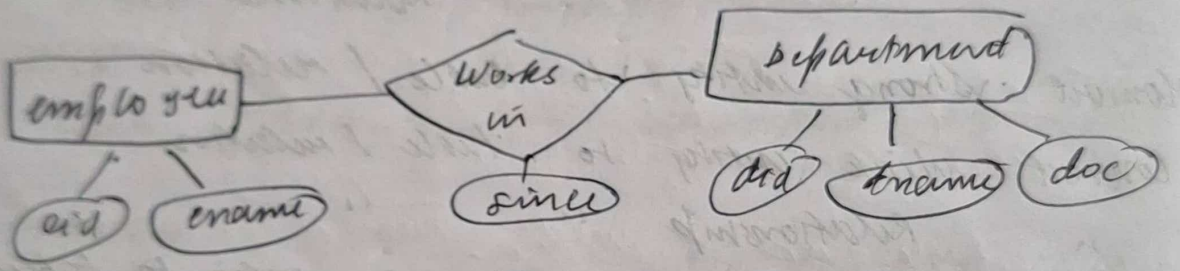


(dis[emphasize attribute)

- to convert weak entity to relation, it requires two tables.

employee (eid, ename) dependents (eid, dname, addr, since)

[2] Convert relationship into table



employee (eid, ename)

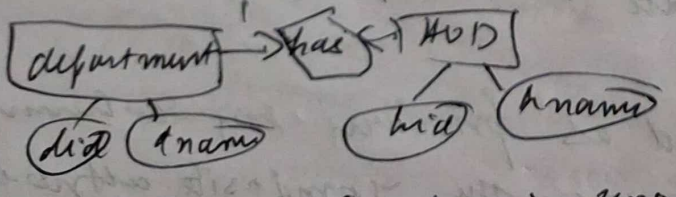
department (did, dname, doc)

work_in (eid, did, since)

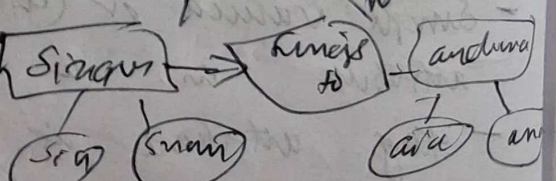
[4] Entity with cardinality ratio to table

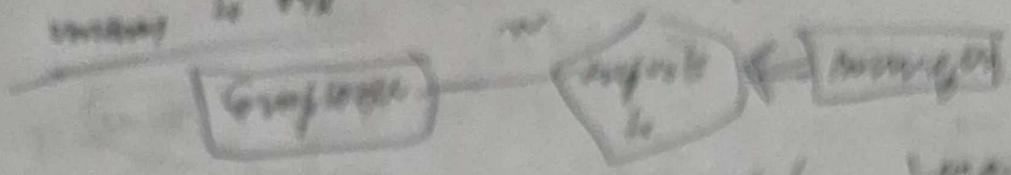
(1) one to one (2) one to many (3) many to one

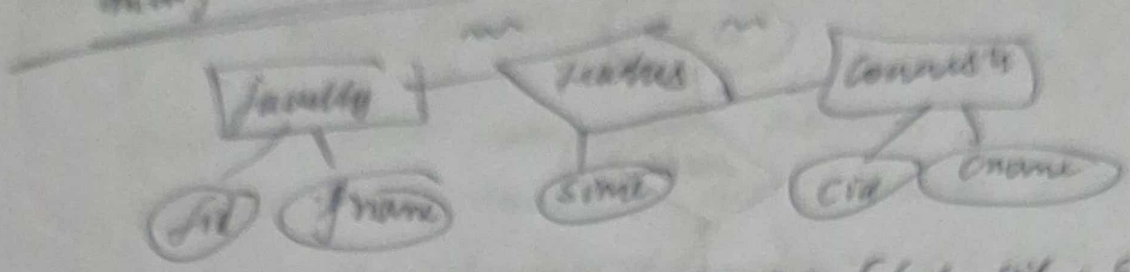
(4) many to many



department (did, dname), HOD (hid, hname)



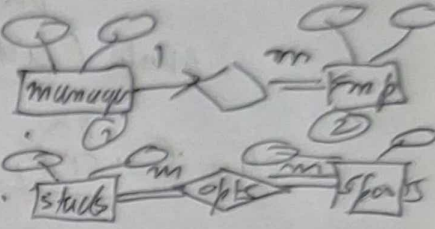




faculty (fid, fname) teaches (fid, cid, sem)
course (cid, cname)

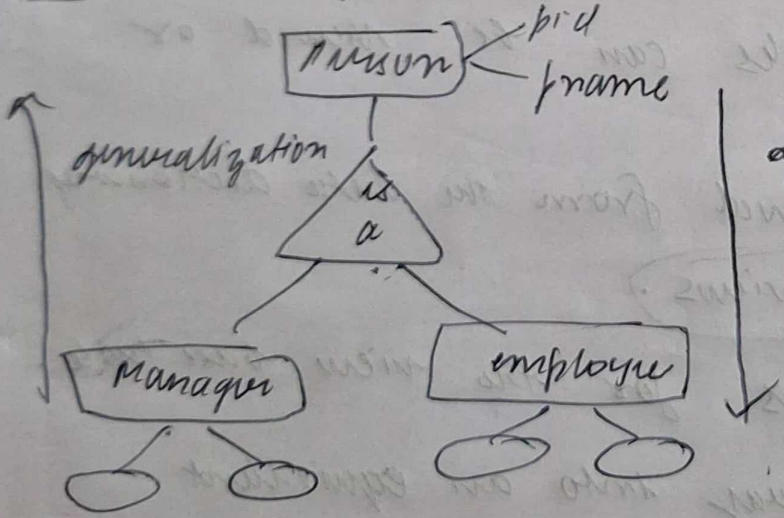
5 Convert entities with cardinality 1 total participation to tables.

- total participation on one side.
- Total participation on both sides.



manages (mid, mname), employee (eid, ename, mid)
(sid, sname, eid, ename)

6 Converting generalization or specialization to tables.



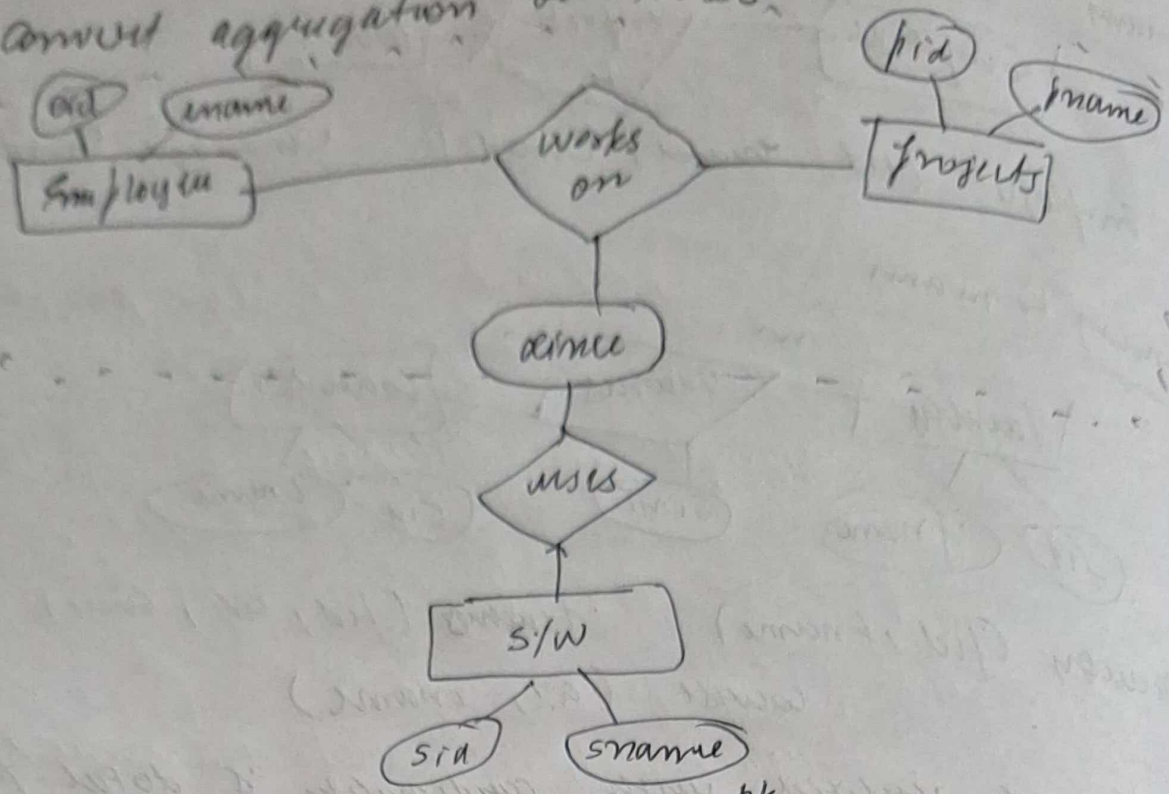
1 way

manages (mid, mname, pid)
employee (eid, ename, pid)

2nd way

person (pid, fname)
emp (eid, ename, pid)
mgrs (mid, mname, pid)

[7] convert aggregation to tables.



emp (eid, ename) projects (pid, pname)

s/w (sid, sname) works (eid, pid, since)

uses (eid, pid, sid)

Views

- A view is like a window created on base table through which data from tables can be viewed or changed.
- View definition is retrieved from the data dictionary tables. Ex- user_views.
- Checks access privileges for the view base table.
- Converts the view query into an equivalent operation on the underlying base table.

syntax create view <view_name> as <select query>

Ex create view emp_view as

select eid, ename, addr from employee;

- View is of two types.
  - Simple view: created on one base table.
  - Complex view: created on more than one base table.

drop
drop view <view_name>

create view emp_dept as
select eid, ename, did, dname from
employee e, department d where e.did = d.did.
and e.did = 30;