PHYSICS PRACTICAL SHEETS

Date: ..... CAMPUS .....
Class: .....
Roll No.: Purusotham Adhikari
Shift: .....
Object of the Experiment (Block Letter)

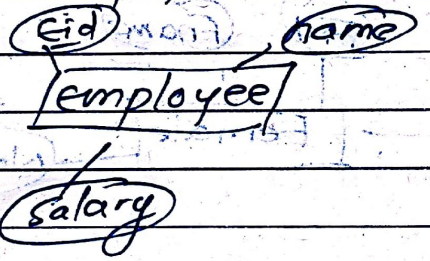
Experiment No.: .....
Group: .....
Sub.: .....
Set: .....

VARIETY PRODUCT

1. Explain the process to convert EER to relational model.

To reduce the ER or EER diagram into table simply we create a table for each entity set and for each relationship sets. To reduce given ER or EER diagram into tables normally we divide ER or EER diagram into following sections.

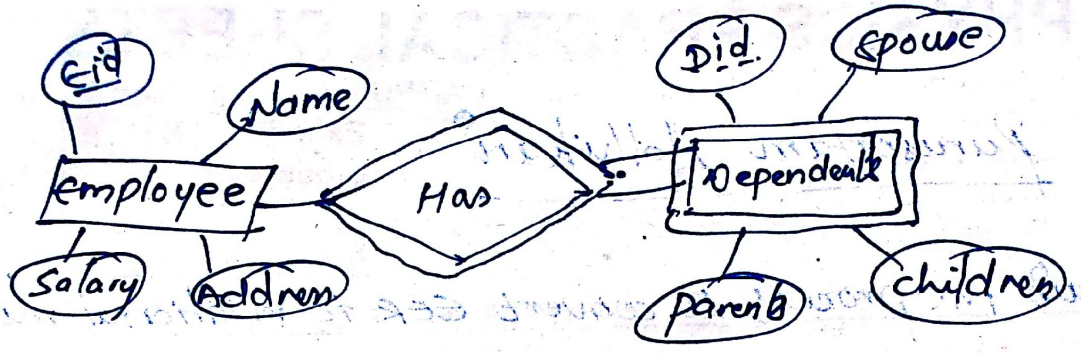
i) Mapping strong entitysets to ER
- create table for each of strong entity
- entity's attributes should become fields of tables with their respective data types.
- Declare key attribute of strong entity set as primary key of table.



employee

Eid	name	salary

ii) Mapping Weak entity sets to ER
A weak entity set does not have its own primary key and always participate in one-to-many relationship with owner entity set and has total participation.
- create table for weak entity set
- Add all its attributes to table as field
- Add Primary key of strong entity for identifying entity set
- Declare all foreign key constraints



employee

Eid	Name	Salary	Address

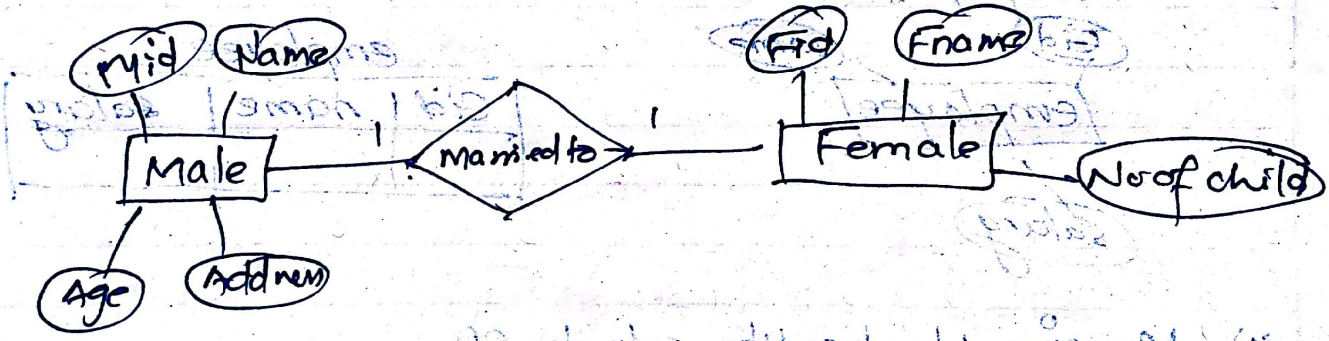
Dependents

Eid	Did	parents	Spouse	Childrens

iii) Mapping Relation sets to ER

a) Mapping Binary 1:1 relation type to ER

Here for relationship we set primary key of any one of entity set as foreign key.



Male

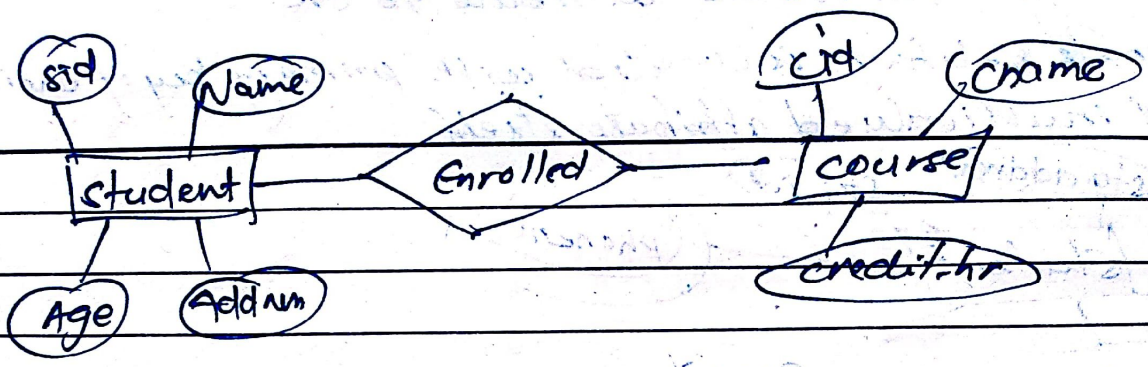
Mid	Name	Age	Address

Female

Fid	mid	Fname	No of child

b) Mapping 1:N relation types to ER

Here In many or N-side we include the primary key of one side entity-set as foreign key.



Student

~~Sid Name~~

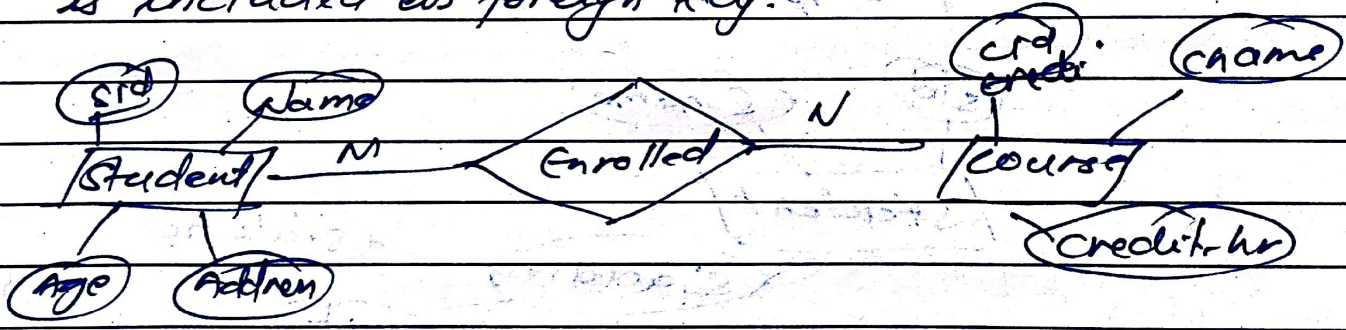
sid	cid	Name	Age	address

Course

cid	cname	credit_hr

c. Mapping M:N relationship types to ER

Here seperate relations is created for relationship type where primary key of each participating entity set is included as foreign key.



Student

sid	Name	Age	Address

Course

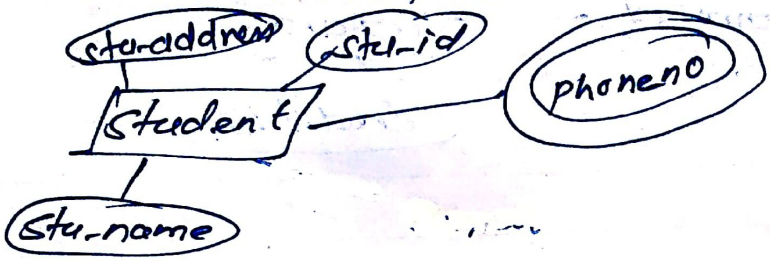
cid	cname	credithr

Enrolled

sid	cid

iv) Mapping Multivalue attribute to ER

Seperate Relation is created with primary key of entity set and multivalued attribute itself



Student

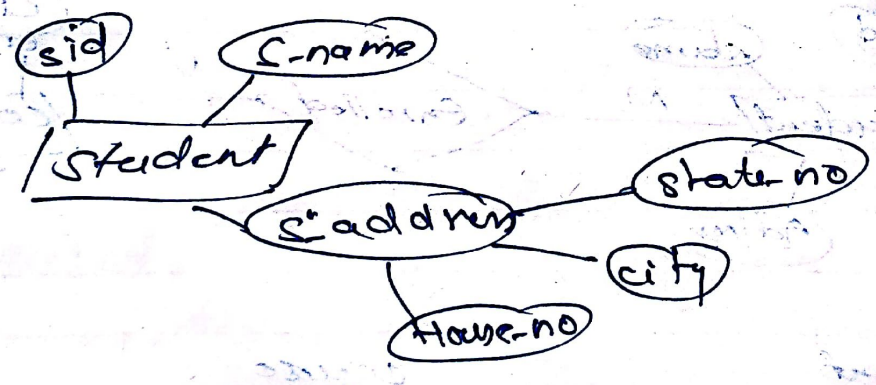
Stu-id	stu-name	stu-address

phone

Stu-id	phone-no

v) Mapping composite Attribute to ER

No seperate attribute is created for composite attribute itself rather attributes are created for component attribute of composite attribute

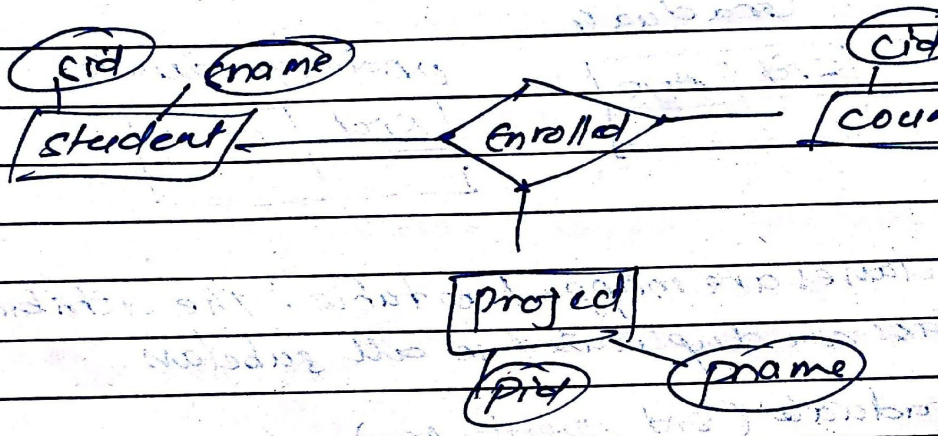


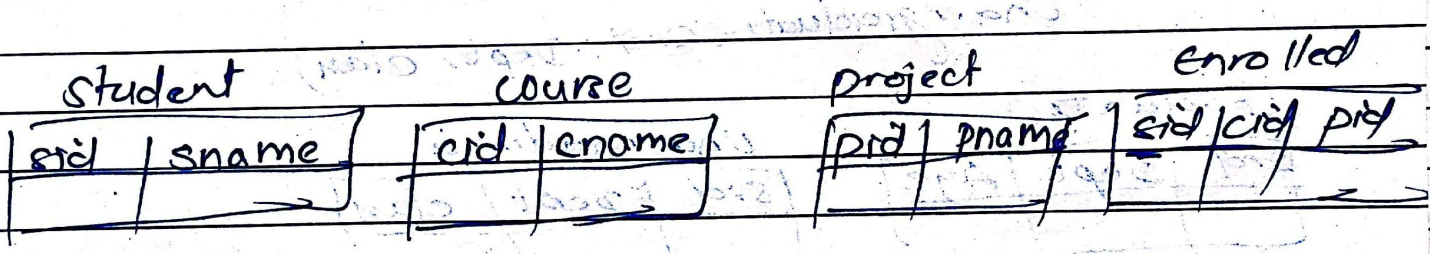
Student

Std	Cname	House-no	city	state-no

Mapping N-ary Relationship Type to ER

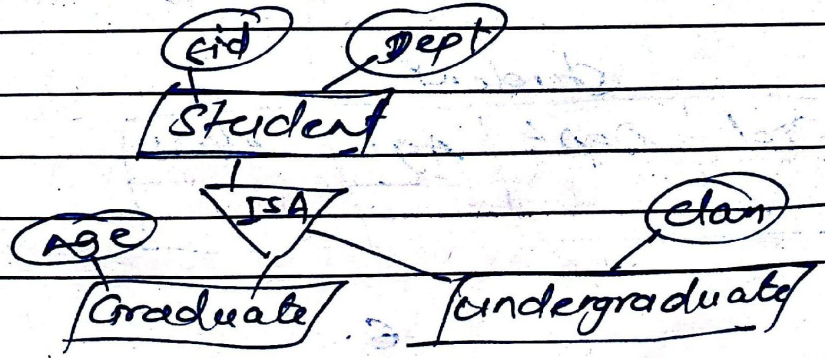
for each n-ary relationship set for $n > 2$, a new relation is created. (which is similar to M:N but seperate table is created for more entity set)





Mapping Specialization / Generalization to ER.

Here we set primary key of super class to their subclass as their foreign key. If subset class are disjoint and complete then relation for a subclass entity ~~for attr~~ set includes all attributes of superclass entity set and all of its attribute.



Method 1

All the entities in the Relationship are mapped to individual tables.

Student (Sid, Dept)
Graduate (Sid, Age)
Undergraduate (Sid, clan)

Student

Sid	Dept

Graduate

Sid	Age

Undergraduate

Sid	class

Method 2.

Only subclasses are mapped to table. The attributes in superclass are duplicated in all subclass

Graduate (Sid, Dept, Age)
Undergraduate (Sid, Dept, clan)

Graduate

Sid	Dept	Age

Undergraduate

Sid	Dept	class

Method 3

only superclass is mapped to a table. Here we introduce null value. If we use one subclass attribute of another subclass is null.

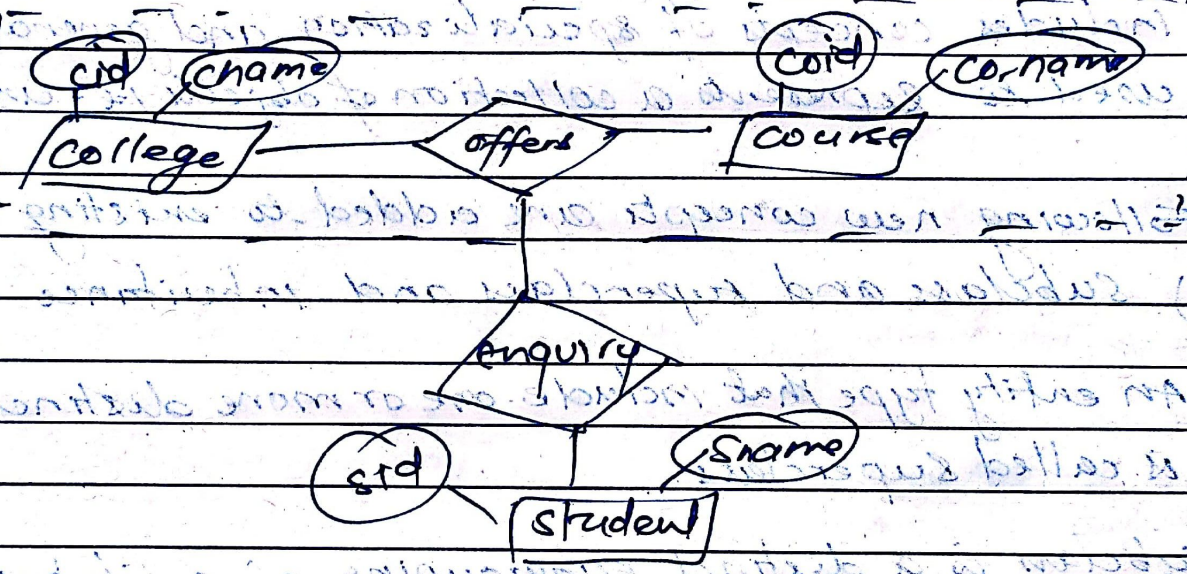
Student (Sid, Dept, Age, clan)

Student

Sid	Dept	Age	class

Mapping Aggregation to ER.

In relational model seperate relation is created for this relationship set and relation contains primary key of associated entity set and relationship set & its own attributes.



college

cid	cname

course

coid	coname

enquiry

eid	ename	cid	coid

2. Describe the extended ER model in details.

EER is an improvement to the existing ER model to handle the complexity of data application eg. CAD, CASE, multimedia OIS etc.

features

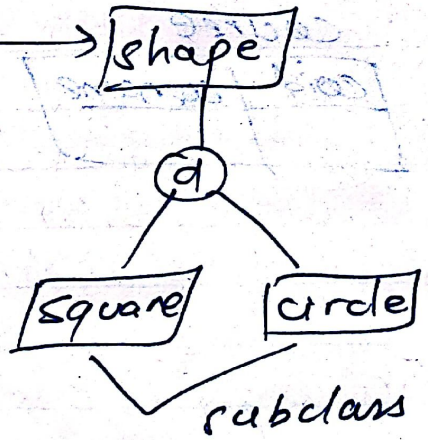
- create a design more accurate to database schemas.
- Reflects the data properties and constraints more precisely.
- includes all modelling concepts of ER model
- includes concepts of specialization and generalization
- used to represents a collection of objects i.e. union.

Following new concepts are added to existing ER model.

i) subclass and superclass and inheritance

An entity type that include one or more distinct subgrouping is called superclass.

subclass is a distinct subgrouping of entitytype.



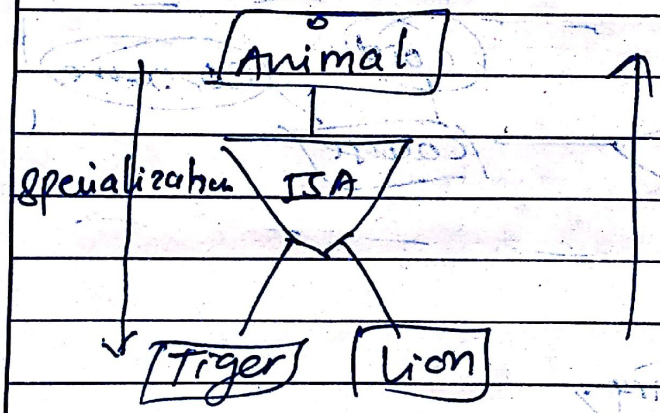
A subclass with more than one superclass is called shared subclass. The attribute of superclass are inherited by shared subclass which may also have

its own additional attributes this process is known as multiple inheritance.

ii) Specialization and Generalization.

The process of maximizing the differences between members of an entity by identifying their distinguishing characteristics is called specialization. (~~bottom up~~) (top down)

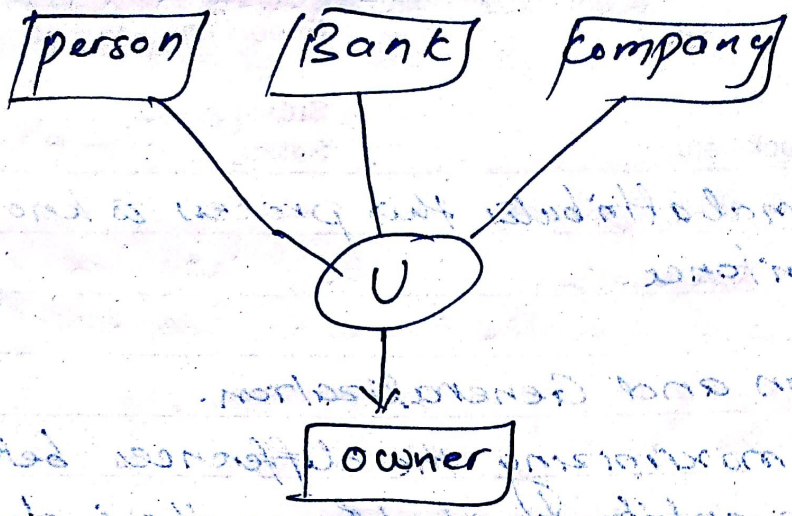
The process of minimizing the differences between entities by identifying their common characteristics is known as generalization (bottom up)



iii) Union type

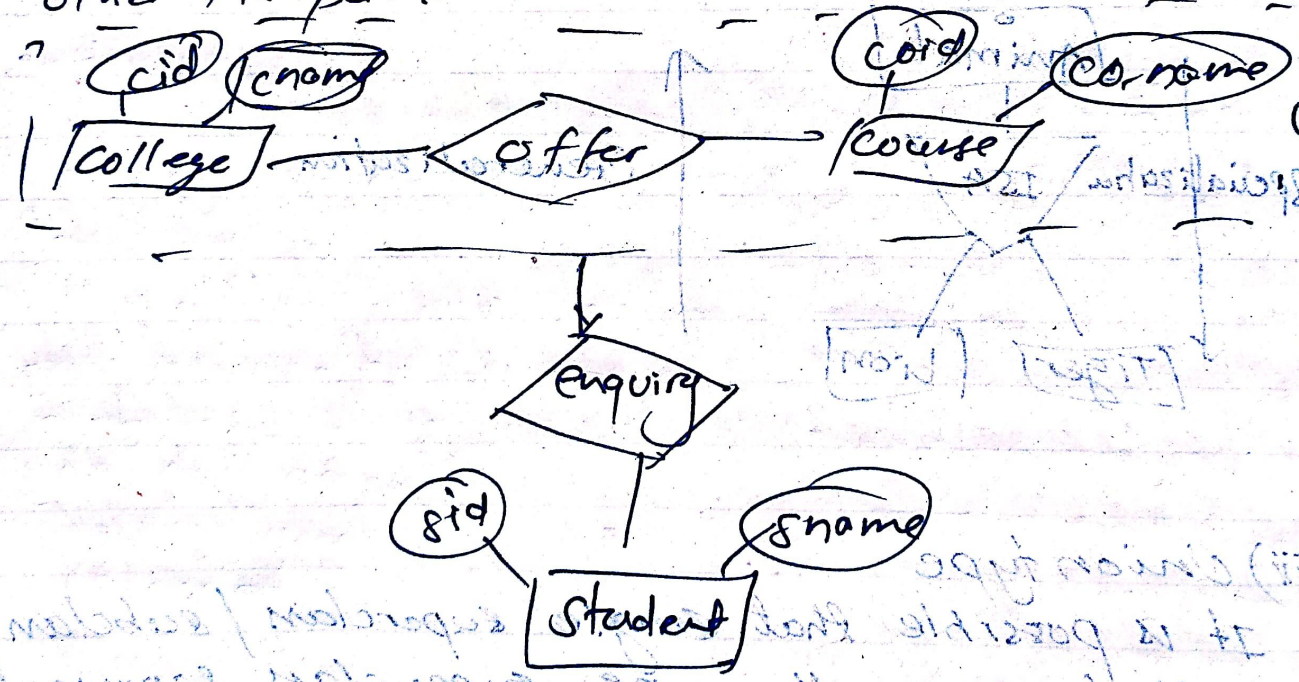
It is possible that single superclass / subclass relationship has more than one superclass representing different entity type. In this case, subclass will represent a collection of objects that is UNION of distinct types.

Union represents a single superclass or subclass relationship with more than one superclass.



iv) Aggregation.

Represent a 'has-a" or 'is-part-of' relationship between entity type, where one represents 'whole' and other the 'part'



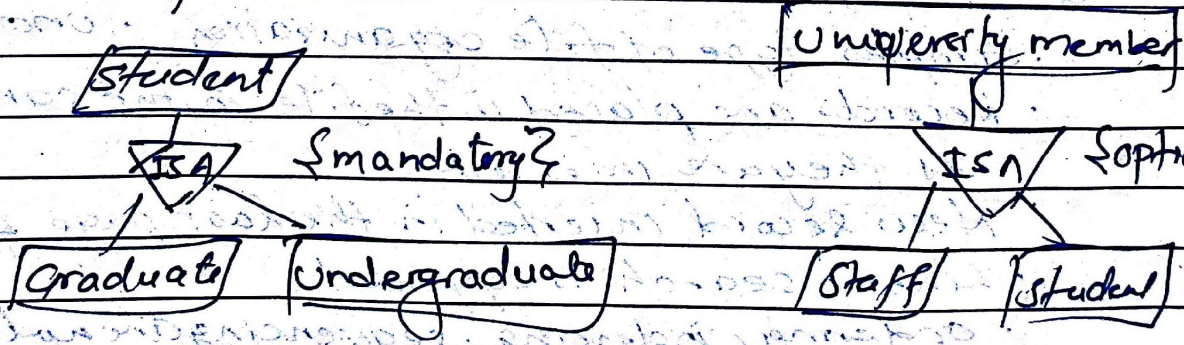
3. Differences between specialization and generalization.

Generalization	specialization
(i) Bottom-up approach	i) top-down approach
(ii) size of schema gets reduced	(ii) size of schema gets increased.
iii) Normally applied to group of entitys	iii) applied to a single entity
iv) procen of creating groups from various entity sets	iv) procen of creating subgrouping within an entity set
v) No inheritance	v) inheritance

4. Discuss the different constraints of specialization and generalization.

i) partipa participation constraints

Determine whether every members in the superclass must participate as a member of subclan. A superclan/subclan relationship with mandatory participation specifies that every member in superclan must also be a member of a subclan



ii) Disjoint constraints

Describes the relationship between members of the subclasses and indicates whether it is possible for member of superclass to be member of one or more than one subclass

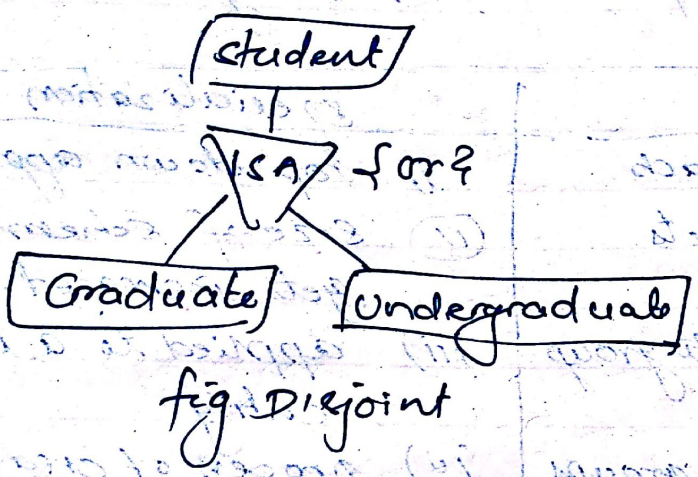


fig Disjoint

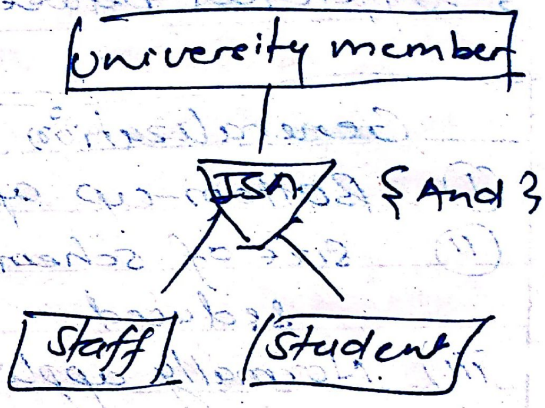


fig overlapping

5. What is file ~~era~~structure in DBMS? explain different types of file organization.

A file is a sequence of records stored in binary format. A disk drive is formatted into several blocks that can store records.

File organization is a logical relationship among various records and defines how files are mapped into disk blocks.

Types of file organizations:-

a. Heap files:-

- Simples type of file organization
- unordered.
- Records are placed in the file in the same ordered as they are inserted.
- New record inserted in the last page of file.
- Linear search is applied
- ordering, indexing, sequencing are not applied.

b. Sequential file:
- ordered - Binary search is applied
- Insertion/Deletion problem is maintained

c. Hash files:-
- has function calculated the address of page in which record is to be stored based on one or more fields in record.
- Hashfield is based field and hash key is the base field if it is also a key field.
- Randomly distributed

d. B+ tree files:
- Advanced method of and indexed sequential access method - Uses a tree like structure to store records
- key-index concept is used.
- All records are stored only at leaf node.
- Intermediate nodes acts as pointers to leaf node

e. cluster file organization
- Not considered good for large database
- ordering of records is not based on primary key or search key.

6. What is Indexing? Explain its types.
Indexing is used to optimize the performance of database by minimizing the no of disk accesses required when a query is processed.
The file containing logical records is data file file containing the index records is index files.

It retrieves the data fast and gives the better performance.

Types of Indexing :-

A. primary Indexing :-
Data is sequentially ordered by an ordering key field and indexing field is built on the ordering key field which is guaranteed to have a unique value in each records.

i. Dense indexing :-
A record is created for every search key valued in the database. It needs more space to store index records.

ii) Sparse Indexing :-
An index record that appears for only some of value. In this method a range of index columns stores the same data block address and when data needs to be retrieved, block address will be fetched.

B. Secondary Indexing :
Secondary index can be generated by a field which has a unique values for each record and it should be a candidate key.
Improves the performance of queries than primary. Every performance and maintaining indexes must be balanced.

C. cluster Indexing :
Records themselves are stored in the index and not pointers. Two or more columns can be grouped together to get Loco identifier and

index is created. cluster key defined order of data certa is table. Table records are physically reordered to match the index.

6. Multilevel Indexing:
- Created when a primary index doesnot fit in memory
- When an index file becomes large and extends over many pages, the search time for the required index increases, multilevel indexing attempts to overcome this issue.

7. Discuss the Hashing concept in DBMS.

Hashing is an effective technique to calculate the direct location of a data record on the disk without using sequential index structure which results decrease in data retrieval time. It uses hash functions with search keys are as parameters to generate the address of a data records. Memory location where the records are stored is known as data bucket or data block.

Types of hashing:-
- Static hashing:-
  Hash function always compute the same address when a search key value is given. For instance if mod 5 hash function is used, then only 5

values are generated. For that function, the output address must always be same.

- Dynamic Hashing
offers a process through which data buckets are dynamically and on-demand added and removed. Also keffered to as extending hashing.

Applications
- password verification
- compiler operation
- data structure
- message digesting

8. Differences between Hashing and Indexing.

Indexing	Hashing
i) It is a technique that allows to quickly retrieve records from database file	i) It is a technique that allows to search location of desired data on disk without using index structure.
ii) It is generally use to optimize or increase performance of database simply by minimizing no of disk accesses that are required when a query is proceed.	ii) It is generally used to index and retrieve items in database as it is faster to search that specific item using shorter hashed key rather than using its original value.
iii) Does not work well in large database	iii) work well for large database
iv) Uses data reference that holds the address of disk block with the value corresponding to key	iv) uses mathmatical function called hash function to calculate direct location

v) Types of indexing are primary, secondary, cluster and multilevel indexing	v) Types of hashing are static and dynamic hashing.

9. Differentiate the between static and dynamic hashing.

Dynamic Hashing	Static hashing
i) It allows the number of buckets to vary dynamically	i) A fixed number of bucket is allocated to a file to store the records.
ii) It uses a second stage of mapping to determine the bucket associated with some search key value.	ii) It uses a fixed hash function to partition the set of all possible search key value into subsets and then maps each subset to a bucket
iii) Performance does not degrade as the file grows.	iii) performance degrades due to bucket overflow
iv) Index entries are randomized in a way that the number of index entries in each bucket is roughly same	iv) It uses a dynamically changing function that allows the addressed space to grow and shrink with varying database files
v) More efficient	v) Less efficient
vi) Data bucket change depending on the records	vi) Resultant data bucket address is always same

10. Explain the concept of bucket overflow in hashing.

When the same address is generated for two or more records, a collision is said to have occured.

- Overflow chaining (Closed Hashing)

A new bucket is allocated for the same hash result when the bucket are complete and is linked after the previous one

eg. If h(k) = 3 and if some data is already present in this location, now a new bucket is linked to the existing location 3

- Linear Probing (open hashing):

When a hash function generates an address at which data is already stored, the next free bucket is allocated to it.

for eg: If h(k) = 6 and is already occupied now check next possible available bucket that is free.

11. Discuss any five advance sql in DBMS.

1. Union

The purpose of sql UNION query is to combine the results of two queries together while removing duplicates.

Syntax [sql statement 1]
UNION
[sql statement 2]

eg. SELECT Txn-Date FROM Store_information
UNION
SELECT Txn-Date FROM Internet_sales;

2. SQL UNION ALL
Same as union but it includes duplication
syntax
[sql statement 1]
UNION ALL
[sql statement 2]

3. SQL intersect
The INTERSECT command in SQL combines the results of two sql statement and only return data that are present in both sql statement

Syntax
[sql statement 1]
INTERSECT
[sql statement 2]

4. SQL EXISTS
EXISTS is a boolean operator used in subquery to test whether the inner query returns any rows. If it does, then outer query proceeds. If not returns nothing
Syntax SELECT 'col_name' FROM 'tablename'
WHERE EXISTS (SQL query);

5. Data definition language (DDL):
It contains of commands which defines data.
Commands are:

- Create: It is used to create a table
  Syntax
  tablename (attr1, datatype -- attrn datatype);
- drop: It is used to delete table including all attributes
  Syntax
  drop table tablename;
- alter: alter is a reserve keyword which modifies structure of table
  Syntax
  alter table tablename add (new col, datatype -- new colx datatype);