



Atma Ram Sanatan Dharma College

University of Delhi



Database Management Systems

Assignment – 1

(Odd Roll No.)



Submitted By -

Anshul Verma

Roll No. – 19/78065

B.Sc. (Hons.) Computer Science

Submitted To -

Mrs. Manisha Bagri

Department of Computer Science

Answer: 1

Union Type / Category: In multiple inheritance, a subclass will represent a collection of entities that is a subset of UNION of entities from distinct entity types, we call such a subclass a union type or a category.

A category has two or more superclasses that may represent collections of entities from distinct entity types, whereas other superclass/subclass relationships always have a single superclass. e.g. ENGINEERING_MANAGER in Fig 1 is a subclass of each of the three ^{super}classes ENGINEER, MANAGER and SALARIED_EMPLOYEE, this represents a constraint that an engineering manager must be an ENGINEER, a MANAGER, and a SALARIED_EMPLOYEE that is ENGINEERING_MANAGER is a subset of intersection of the 3 entity sets.

On the other hand, a category is a subset of union of its superclasses.

In Fig 2, an entity that is a member of OWNER must exist in only one of the superclasses, that represents a constraint that an OWNER may be a company, a BANK or a PERSON.

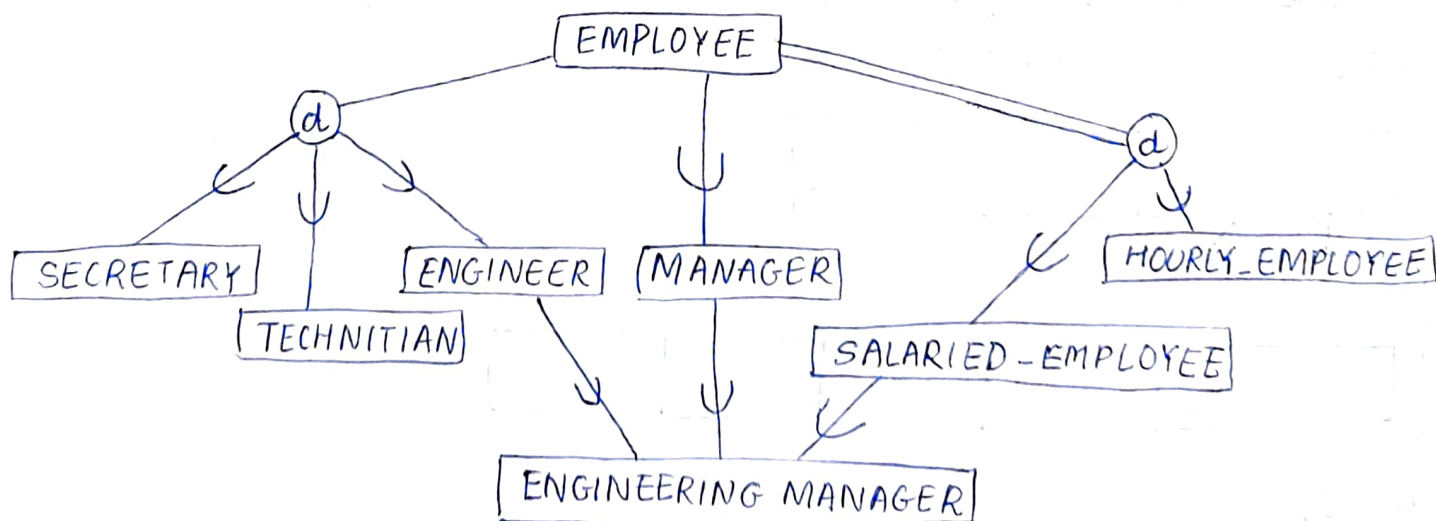
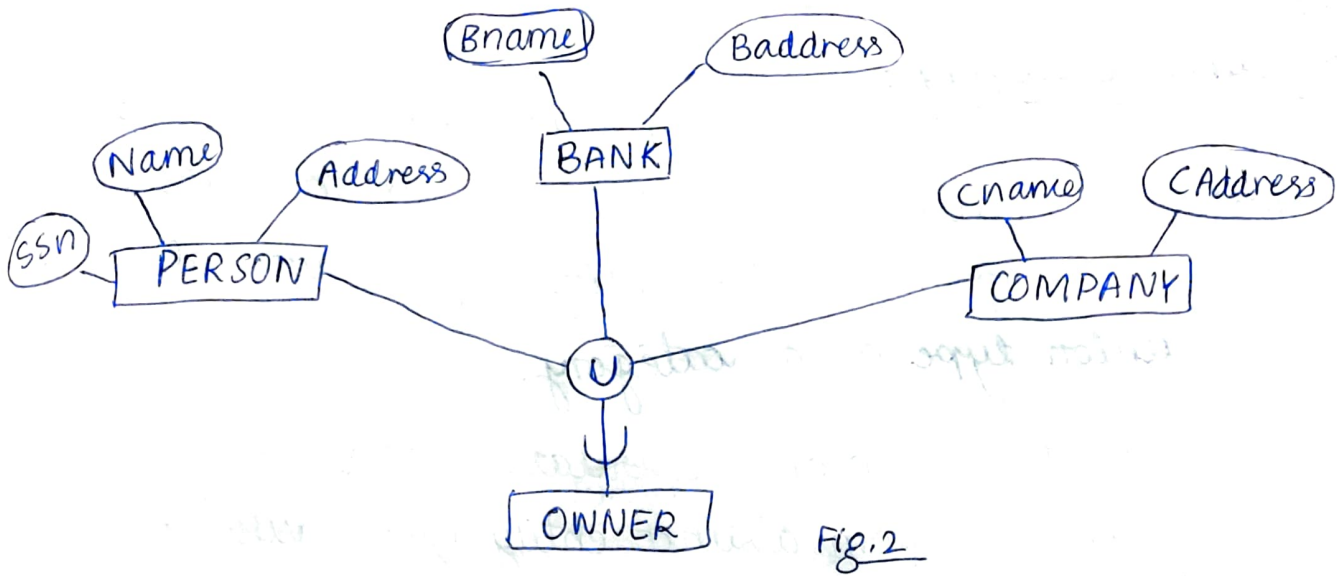
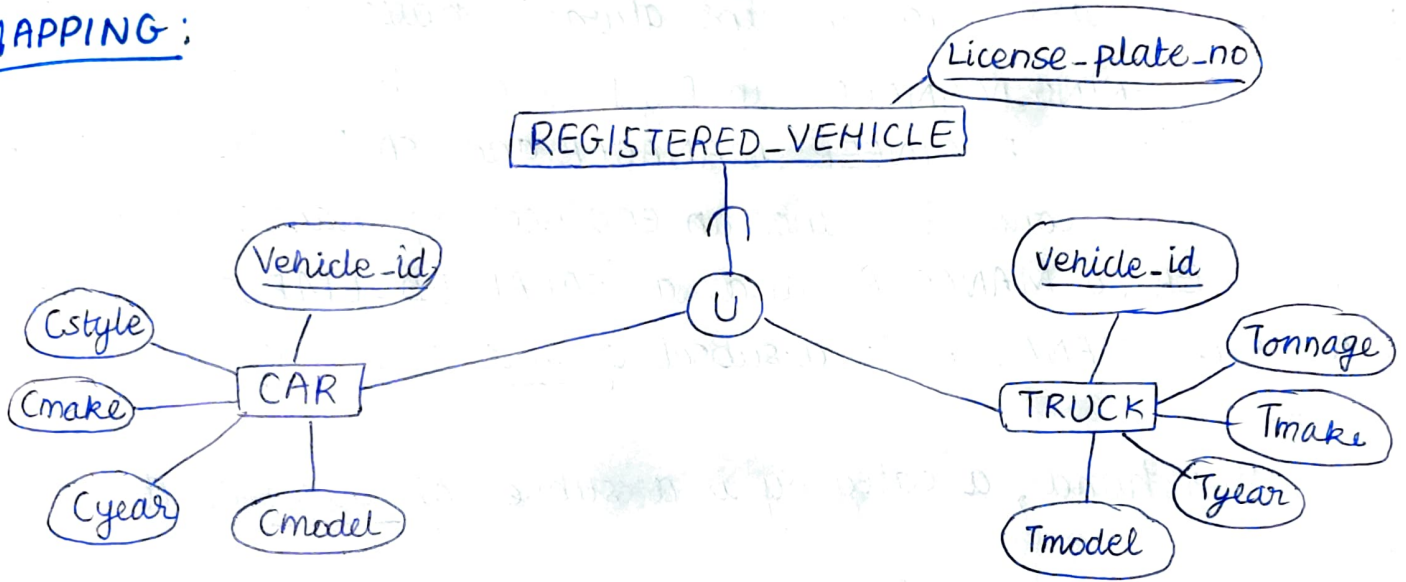


Fig. 1.

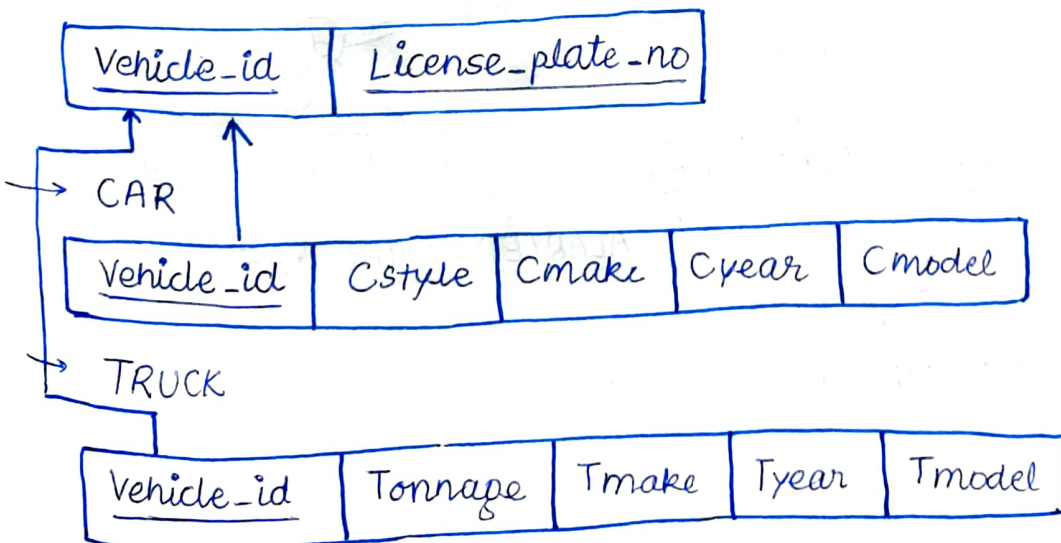


MAPPING :



For a category whose Since, defining superclasses CAR and Truck have a common key, we don't need a new surrogate key here.

→ REGISTERED-VEHICLE



Answer: 2

Attribute-defined specialization

→ If all subclasses in a specialisation have their membership condition on the same attribute of the superclass, the specialization is called an attribute-defined specialization.

→ Membership is determined by the condition on a predefined attribute known as defining attribute.

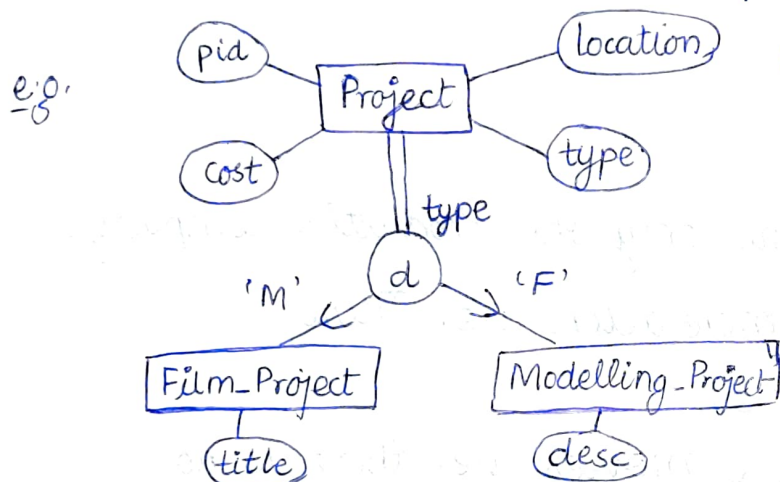


Fig 1

subclasses.

This is an attribute defined specialization.

Whereas, In fig 2, there is no condition for determining membership in a subclass.

When a user will apply the operation to add an entity to a subclass, then can determine the membership condition will be determined by the database user

and the membership will be specified individually for each entity by the user.

User-defined specialization

→ When we do not have a condition for determining membership in a subclass, the subclass is called user-defined and the specialization is called an user-defined specialization.

→ Membership is determined by database users when they apply the operation to add an entity to the subclass.

Here, In fig 1, defining attribute is 'type' and all the values ~~the~~ entities with 'type' 'M' will belong to 'Film_Project' subclass and all the entities with type 'F' will belong to 'Modelling_Project' subclass.

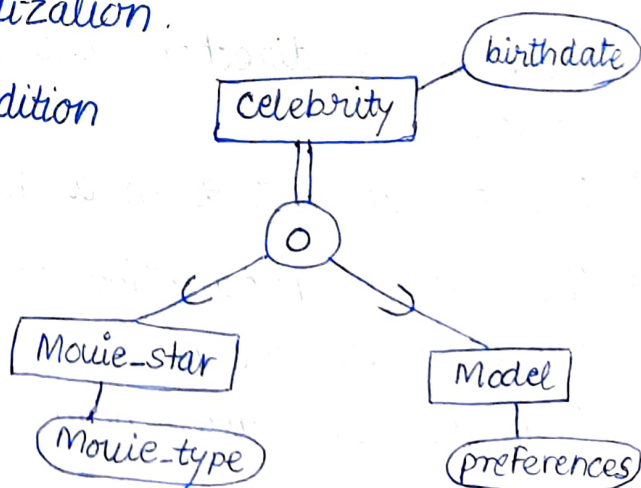
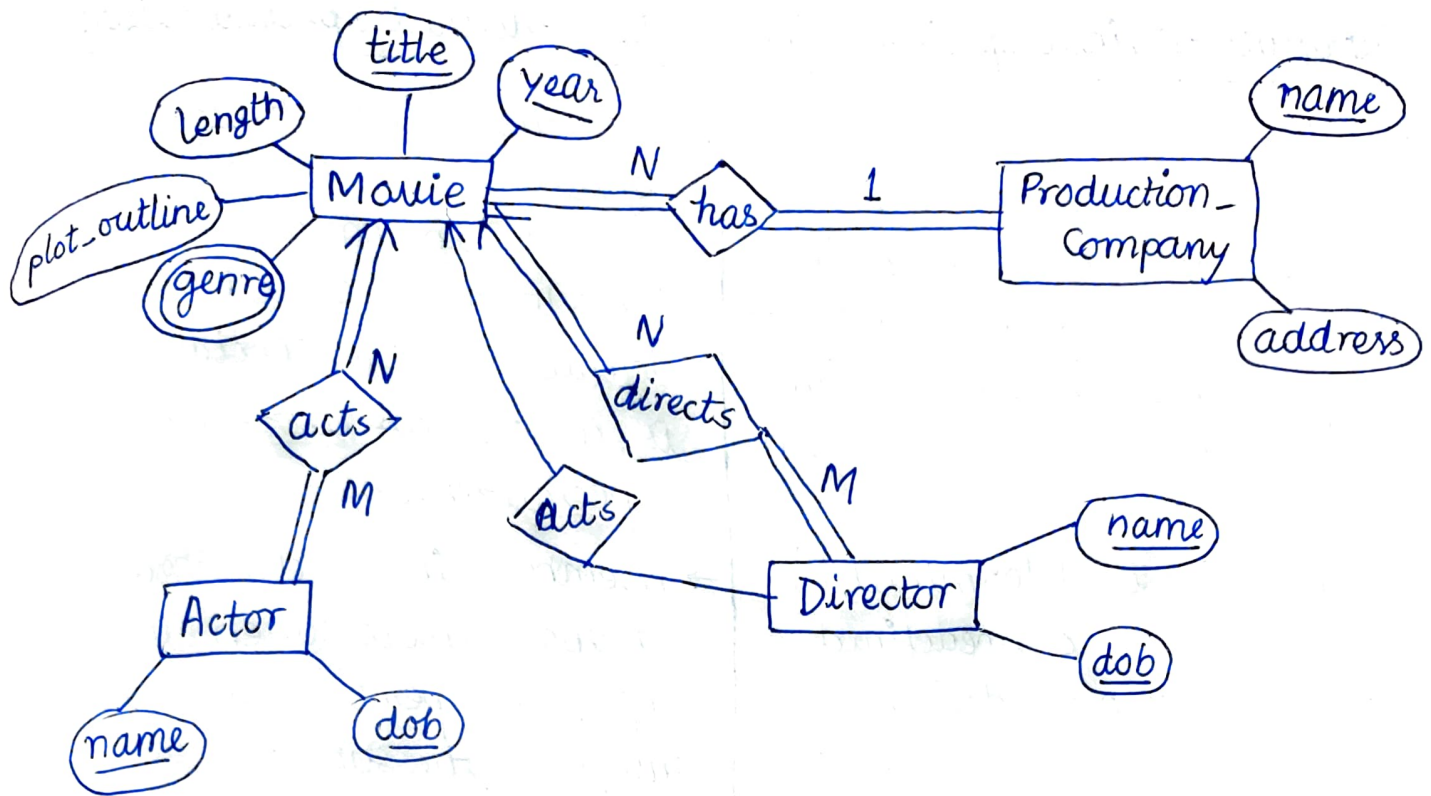


Fig 2

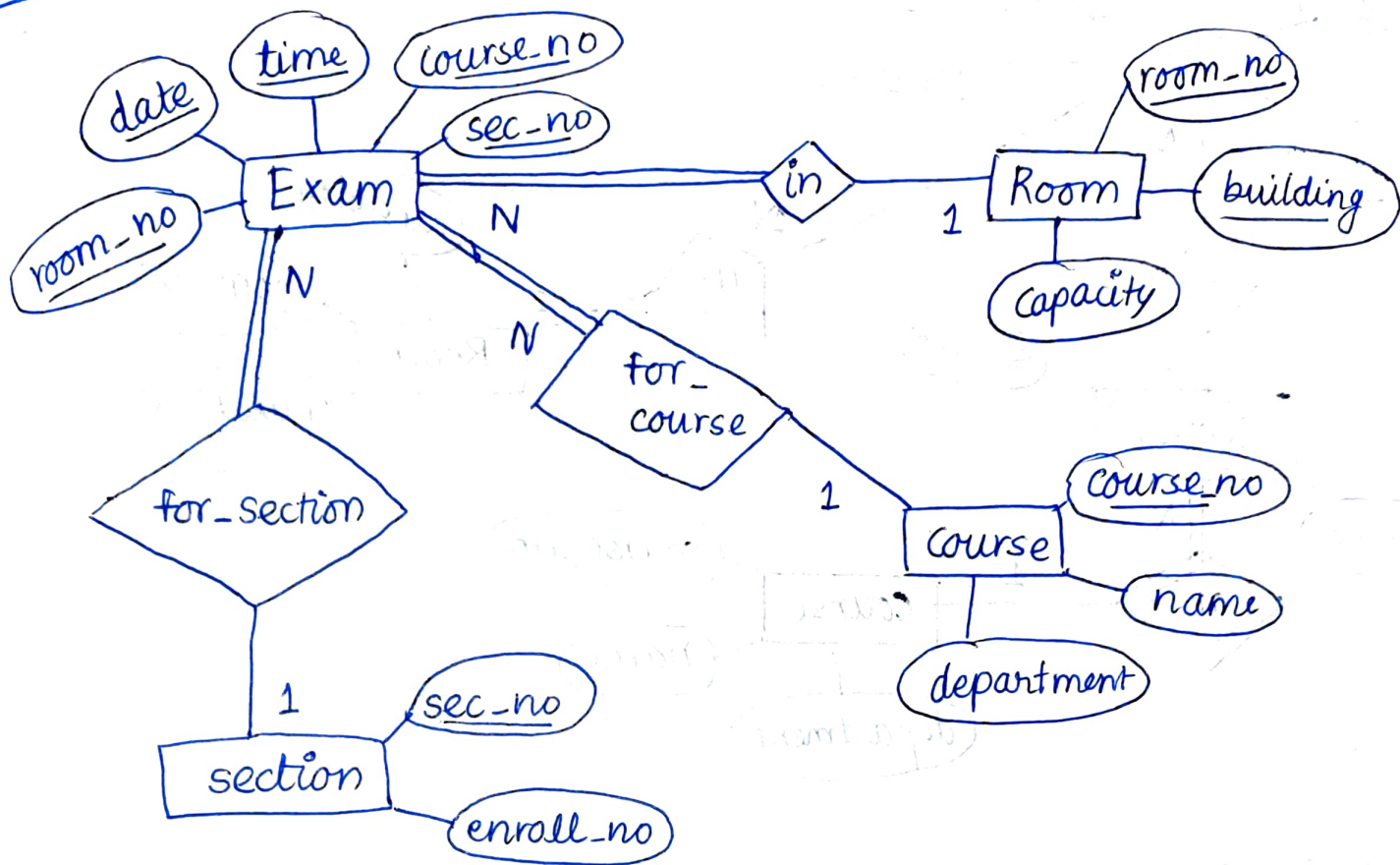
Answer: 3



Assumptions:

- ① Every movie has one and only one Production company.
- ② A movie can have one or more actors i.e. There is no movie with zero actors.
- ③ An actor ^{acts in} have one or more movies i.e. there is no actor who has not acted in zero movies.
- ④ There is no director who have not directed even a single movie.
- ⑤ Director can act & in a movie.
- ⑥ There is no production company that have not produced even a single movie.
- ⑦ A movie can have more than one directors.

Answer: 4



Assumptions:

- ① Each exam will have one and only one section, one and only one course and ⁱⁿone and only one room.
- ② Each section can have zero or more exams.
- ③ Each course can have zero or more exams.
- ④ Each room can have zero or more exams i.e. there can be a room in which no exam is being conducted.
- ⑤ No exam can have zero sections, zero courses or zero rooms.

Answer: 5 (i) M join N where $M.X = N.P$

X	Y	Z	P	Q	R
5	3	6	5	10	6
10	7	9	10	7	12
5	2	7	5	10	6

(ii) M right^{outer} join N where $M.Y = N.Q$

X	Y	Z	P	Q	R
5	3	6	null	null	null
10	7	9	10	7	12
5	2	7	15	2	7

(iii) M full outer join N where $M.X = N.P$ and $M.Y = N.Q$

X	Y	Z	P	Q	R
5	3	6	null	null	null
10	7	9	10	7	12
5	2	7	null	null	null
null	null	null	5	10	6
null	null	null	15	2	7

Answer: 6 Duplicate tuples are not allowed in a relation because:

- (i) they create redundancy in database which makes the data processing like querying, inserting, deleting, updating etc. slow.
- (ii) the body of a relation is a set (a set of tuples) and sets in mathematics do not contain duplicate elements.
- (iii) it violates the specifications of the referential integrity constraints.

Answer: 7

- (i) SELECT B.Name FROM BORROWER B
WHERE NOT EXIST
(SELECT * FROM BOOK-LOANS BL
WHERE B.Card_no = BL.cardnumber) ;
- (ii) SELECT B.Name, B.Address, COUNT(*) AS "No. of books"
FROM BORROWER B, BOOK-LOANS BL
WHERE B.Card_no = BL.Cardnumber
GROUP BY BL.Cardnumber
HAVING COUNT(*) > 5 ;
- (iii) SELECT B.Title FROM BOOK B, BOOK-AUTHORS BA
WHERE B.PublisherName = "SSSS"
AND B.BookId = BA.Book_id
AND BA.AuthorName = "AAA" ;

Answer: 8

- (i) SELECT SP.S_Name
FROM Sales_Person SP, Sales S
WHERE SP.S_NO = S.S_No
AND S.P_Id = 71 ;
- (ii) SELECT C.C_Name
FROM Customer C, Sale S, Product P
WHERE P.Description = "Table Fans"
AND P.P_Id = S.P_Id
AND C.C_No = S.C_No ;
- (iii) SELECT SUM(Qty) AS "Total products sold"
FROM Sale
WHERE Date = "2012-03-15" ;

(iv) SELECT C.CName, COUNT(*) AS "No. of products purchased"
FROM Sales S, Customer C
GROUP BY S.C-No
HAVING S.C-No = C.C-No;

Answer: 9

(i) SELECT fname, lname, MAX(salary)
FROM WORKER
GROUP BY department;

(ii) SELECT Worker_ID, fname, lname FROM Worker
GROUP BY depart
WHERE salary IN (

SELECT salary
FROM Worker
GROUP BY salary
HAVING COUNT(*) > 1
) ;

(iii) SELECT department FROM Worker
GROUP BY department
HAVING COUNT(*) < 5;

Answer: 10 Examples where use of null values would be appropriate:

- (i) Suppose in a relation containing STUDENT details, not all students have office phones, ~~me~~ thus null value will be appropriate here, where the actual data is ~~not known~~ ~~or~~ does not exist.
- (ii) A student in above relation can have NULL for home phone, presumably either he does not have a home phone or he has one but we do not know it. Here, the data is ~~not known~~, and thus NULL value would be appropriate.

(iii) Suppose in a relation containing details for employees of a company, and an employee from that company has changed ~~at~~ his address but his address is not known, thus new address should be retained with a null value until next update.