Spring 2020 Semester Final Answer Sheet

Name: Syeda Nowshin Ibnat

ID: 17183103020

Intake: 39 (1)

Shift: Day

Semester: Spring 2020

Answer to the question: 1 (a)

Soln: An Entity-relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram).

An ER-model is a design or blueprint of a database that can later be implemented as database.

The main components of E-R model are: entity set and relationship set.

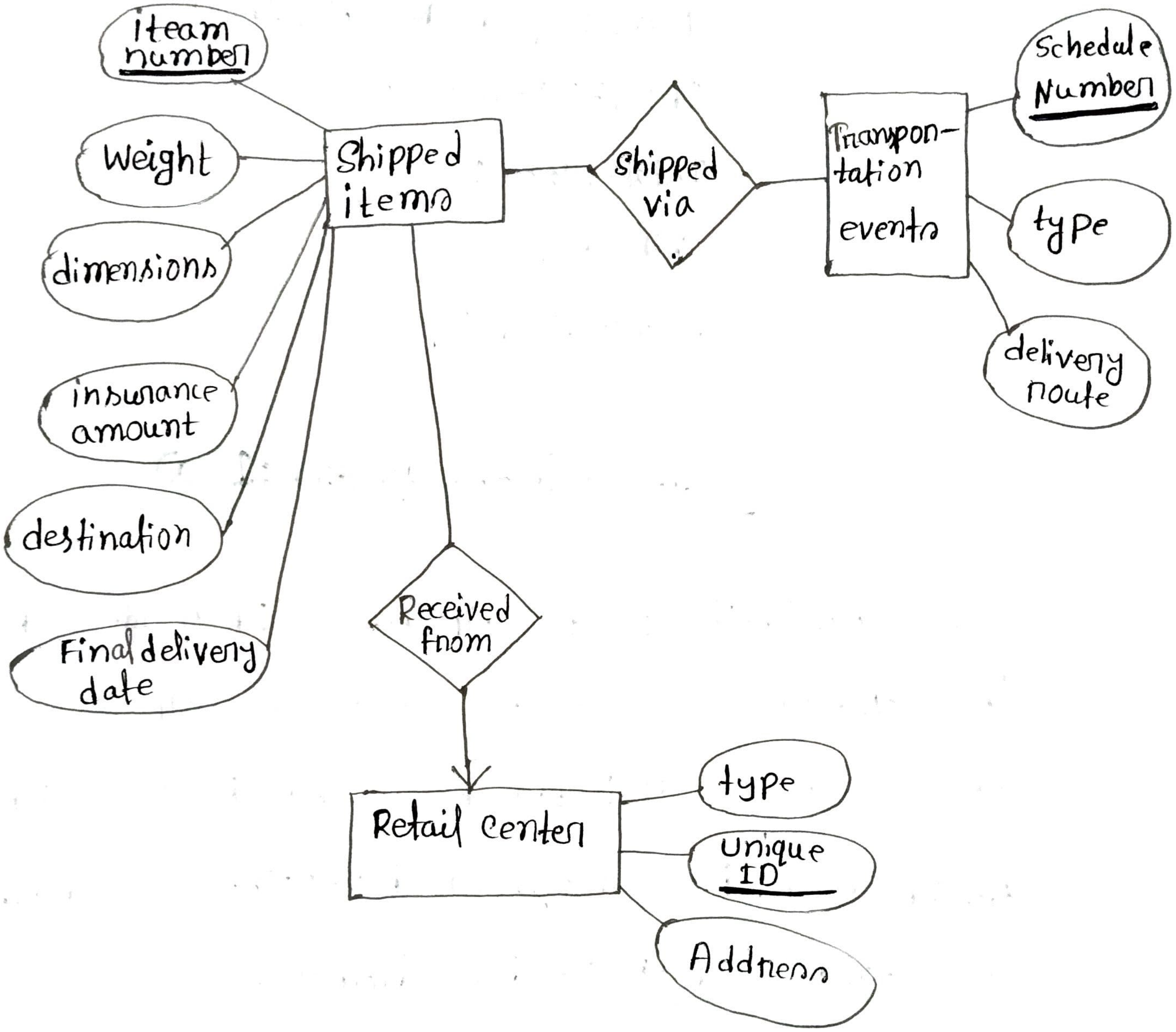
From the given scenario, an ER Diagram can be created.

②

Name: Syeda Nowshin Ibnat

ID: 1718310302O

E-R Diagram



Entities: Shipped Iteams, Retail center, Transportation events.

Relationships: Received from (Retail center, shipped iteam).
shipped via (Shipped iteams, Transportation ~~en~~ events).

Name: Syeda Nowshin Ibnat
ID: 17183103020

Primary keys: Iteam number, Unique ID, Schedule number.

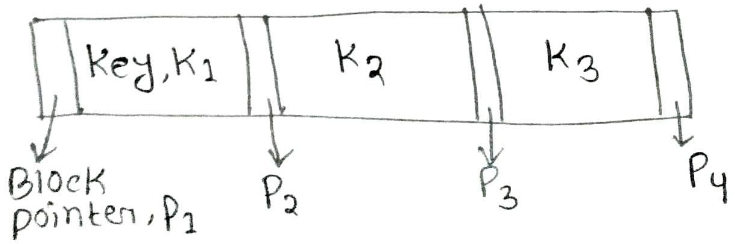
Here, between shipped iteams and Retail center have many-to-many relation.

And between Retail center and Shipped iteams have one-to-one relation.

Answer to the question: 1(b)

Soln: B+ tree is an extension of B tree which allows efficient insertion, deletion and search operations. In B+ tree, data can only be stored on the leaf nodes while internal nodes can only store the key values. The leaf node of a B+ tree are linked together in the form of singly linked lists to make the search queries more efficient.

structure of each node is:



(4)

Name: Syeda Nowshin Ibnat

ID: 17183103020

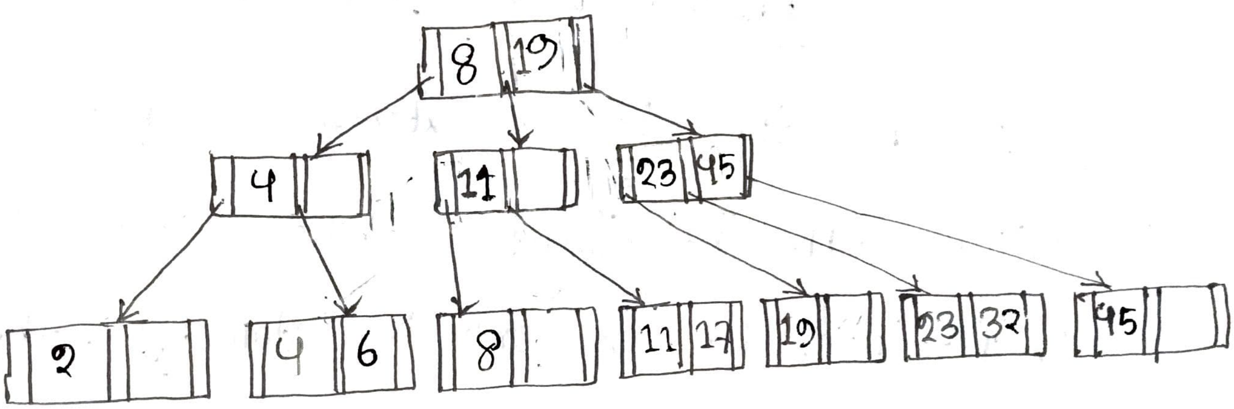
Given, Order = 3

Key values = 4, 6, 2, 8, 32, 17, 23, 19, 11, 45

Sorted values = 2, 4, 6, 8, 11, 17, 19, 23, 32, 45

Here, each node contains=(n-1) = (3-1) = 2 keys

B+ tree



Ans:

Name: Syeda Nowshin Ibnat
ID: 17183103020

Answer to the question: 2(a)

Soln: Given, Relation $R(A,B,C,D)$

$$F = \{ A \to BCD,\ B \to C,\ A \to B,\ AB \to C,\ B \to D \}$$

(i) Some members of $F+$:

$A \to C$

$A \to D$

$B \to CD$ (union)

(ii) Here, $B+ = BCD$ $\left| \begin{array}{l} B \to C \\ B \to D \end{array} \right.$

(iii) canonical cover of F:

Step 1: Applying decomposition.

$$\left.\begin{array}{l} A \to B \\ A \to C \\ A \to D \\ B \to C \\ A \to B \\ AB \to C \\ B \to D \end{array}\right\} \quad \begin{array}{l} A \to C \\ A \to B \\ A \to D \\ B \to C \\ B \to D \\ AB \to C \end{array}$$

Name : Syeda Nowshin Ibnat
ID : 1718310302O

Step : 2 if A→c is needed / or not

$A^+ = ABcD$
$-(A^+) = BDA$
not same

so, A→c is needed

Step : 3 if A→B is need / not

$A^+ = ABcD$
$-(A^+) = DA$
not same

A→B
A→D
B→c
B→D
AB→c

Step : 4 if A→D is not / not

$A^+ = AD$
$-(A^+) = A$
not same

So, A→D is needed.

A→D
B→c
B→D
AB→c

Step : 5 if B→c is need / not

$B^+ = BcD$
$-(B^+) = DB$
not same

So, B→c is needed.

B→c
B→D
AB→c

Name: Syeda Nowshin Ibnat

ID: 17183103020

Step: 6 if B→D is need/not

$$\left.\begin{array}{l} B^+ = BD \\ -(B^+) = B \end{array}\right\} \text{not same}$$

B→D
AB→C

So, B→D is needed.

Step: 7 if AB→C is need/not

$$\left.\begin{array}{l} AB^+ = ABC \\ -(AB^+) = ABC \end{array}\right\} \text{same}$$

So, AB→C is not needed.

The canonical cover is, $F = \left\{ \begin{array}{c} A \to C \\ A \to B \\ A \to D \\ B \to C \\ B \to D \end{array} \right\}.$

Ans:

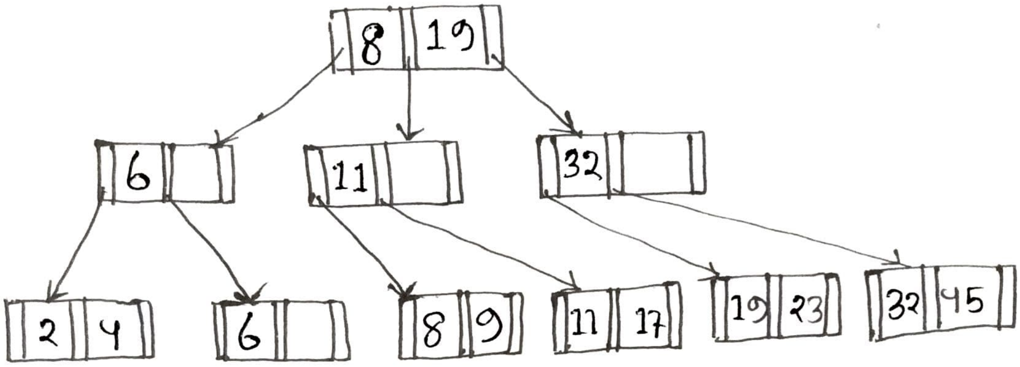
8

Name: Syeda Nowshin Ibnat.

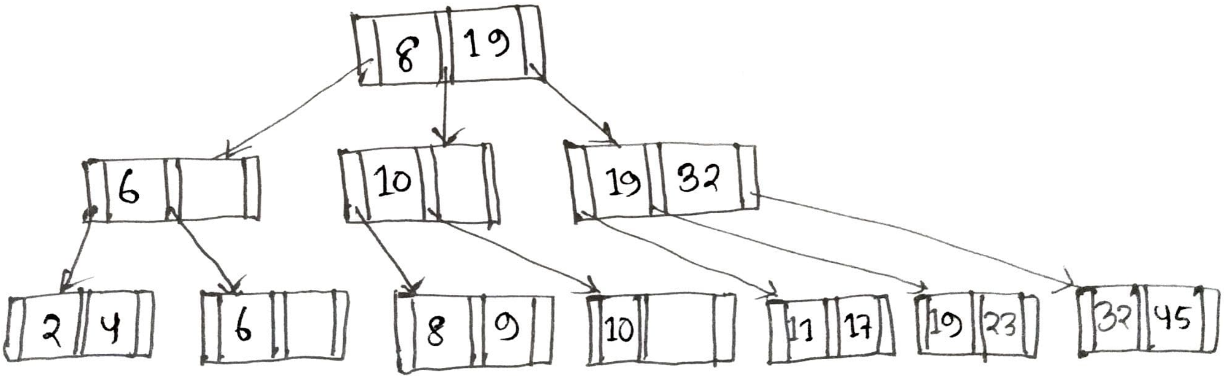
ID: 17183103020

Answer to the question: 2(b)

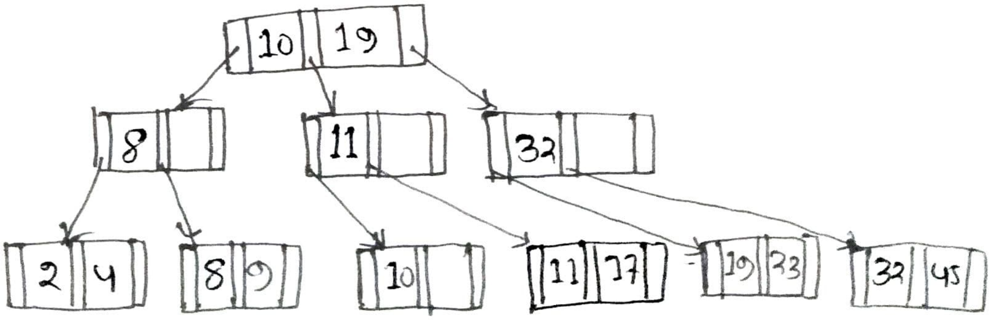
(i) Insert - 9



(ii) Insert - 10



(iii) Delete - 6



Name : Syeda Nowshin Ibnat
ID: 17183103020

Answer to the question : 3(a)

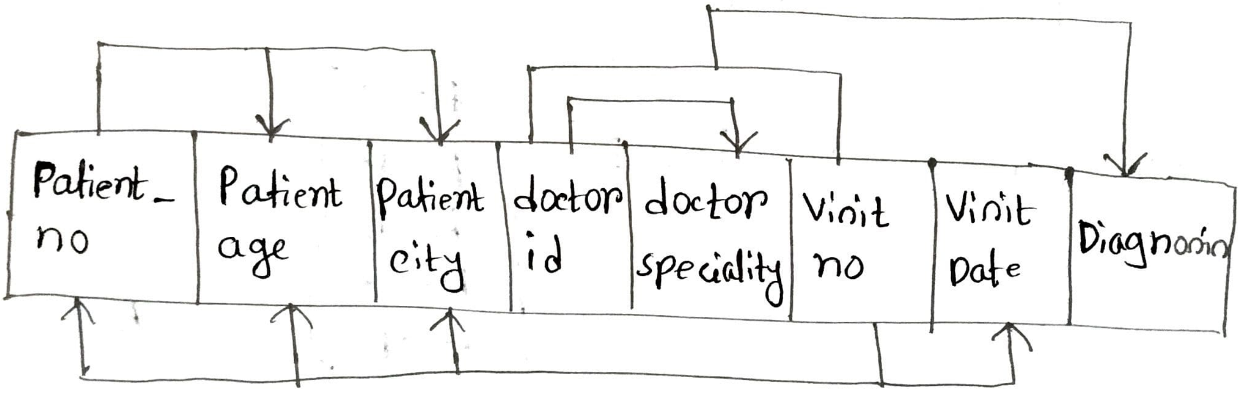
Given,

Patient-no → Patient_age, patient_city.

doctor_id → doctor_speciality

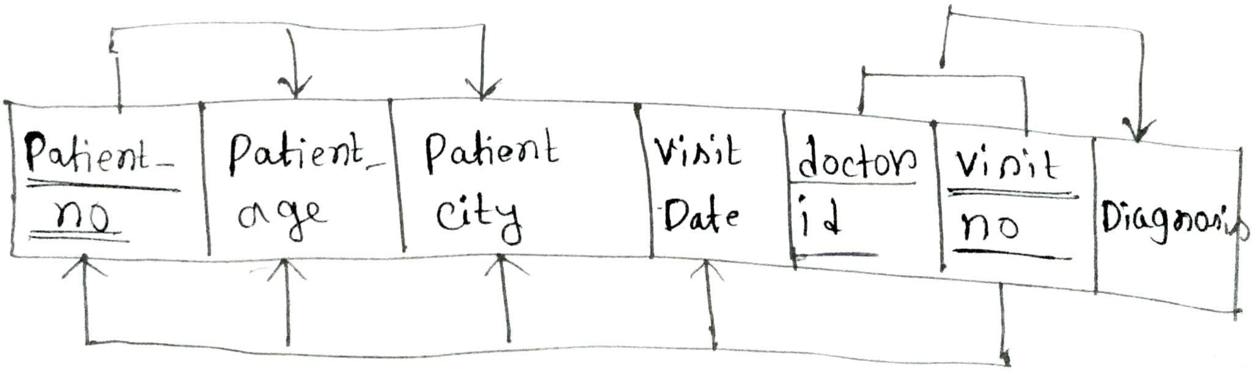
Visit_no → Patient_no, Patient-age, Patient_city
Visit-date

Visit_no, doctor_id → Diagnosis.



This table is not in 1NF

Making this table to 1NF:



Name: Syeda Nowshin Ibnat

ID: 17183103020

doctor_id	doctor-speciality

Now, this is in 1NF.

This 1NF table is consist with partial dependency.

So, now we need to remove all type of partial dependency. to ~~cont~~ convert it into 2NF.

Visit no	Patient no	Patient age	Patient city	Visit-date

Visit_no	doctor_id	Diagnosis

Doctor_id	doctor speciality

Now, this is in 2NF.

~~$~~ To make it 3NF we need to remove all transitive dependency.

Patient_no	Patient-age	Patient_city

visit-no	visit_Date	Patient-no

visit-no	doctor_id	Diagnosis

doctor_id	doctor_speciality

So, This is in 3 NF.

Ans:

Name: Syeda Nowshin Ibnat

ID: 17183103020

Answer to the question : 3(b)

→ Candidate key is a super key that is a proper subset of another super key. It must contain unique values. candidate key may have multiple attributes. It does n't contain null values.

→ A super key is a set of one or more attributes that taken collectively to identify uniquely a tuple in a relation. A candidate key is a super key but vice versa is not true.

From the functional dependency of Q 3(a) we get that,

Candidate key → Visit_no.

Super key → Patient_no, visit_no, doctor_id.

Ans:

Name: Syeda Nowshin Ibnat

ID: 17183103020

Answer to the question: 4(a)

Soln: Transactions is a set of operation, represents a single logical unit of task. Transactions access data using read and write operations. In order to maintain consistency in a database, before and after the transaction, certain properties are followed. These are called ACID properties.

From the given schedule. let, A and B are two different account.

Now, suppose the current values of accounts A and B are \$3000 and \$1000, respectively. And also that the two transactions are T_1 and T_2.

Before transaction, the total amount of money in accounts A and B is = A+B = \$3000 + \$1000 = \$4000

Now, if we put the amount of A ~~B~~ and B ~~on~~ in the schedule. we get:

Name: Syeda Nowshin Ibnat

ID: 17183103020

T1	T2
Read(3000)	
A:= 3000-50	
write(2950)	
	Read(1000)
	B:= 1000-10
	write(990)
Read(990)	
B:=990+50	
write(1040)	
	Read(2950)
	A=2950+10
	write(2960)

Here, ~~A~~after completing the transactions of account A and B, we get: A = ~~3~~$2960, B= $ 1040

Sum of A,B is= A+B= 2960+1040 =$4000

Atomicity

It means the entire transaction takes places at once or doesn't happend at all.

For example: Read (A);

A:= A-50;

write (A);

Read (B);

B:= B; → Transaction failed

write (B);

~~But~~ we can see from the schedule that entire transactions takes place.

Consistency

~~The dal~~ Before the transaction start and after transaction end the summation of both accounts must be same for maintaining consistency.

~~F~~ we ~~can~~ find out earlier that the summation is same, which is $4000.

Isolation

Isolation is a internal work. Mutiple transactions Occure ~~indp.~~ independently without interference. ~~For m this~~ ~~B~~ We have convert concurrent to serial transaction to maintain consistency. By using conflict serializibility we can convert the given schedule.

Name: Syeda Nowshin Ibnat

ID: 17183103020

Durability

The changes of a successfull transaction occurs even if the system failure occurs. From the schedule ~~is~~ we can see after happening a successful transaction, Change in the schedule remains same before the next transaction.

So, I can say the given schedule maintain the ACID properties of transaction.

Answer to the question: 4(b)

Soln:

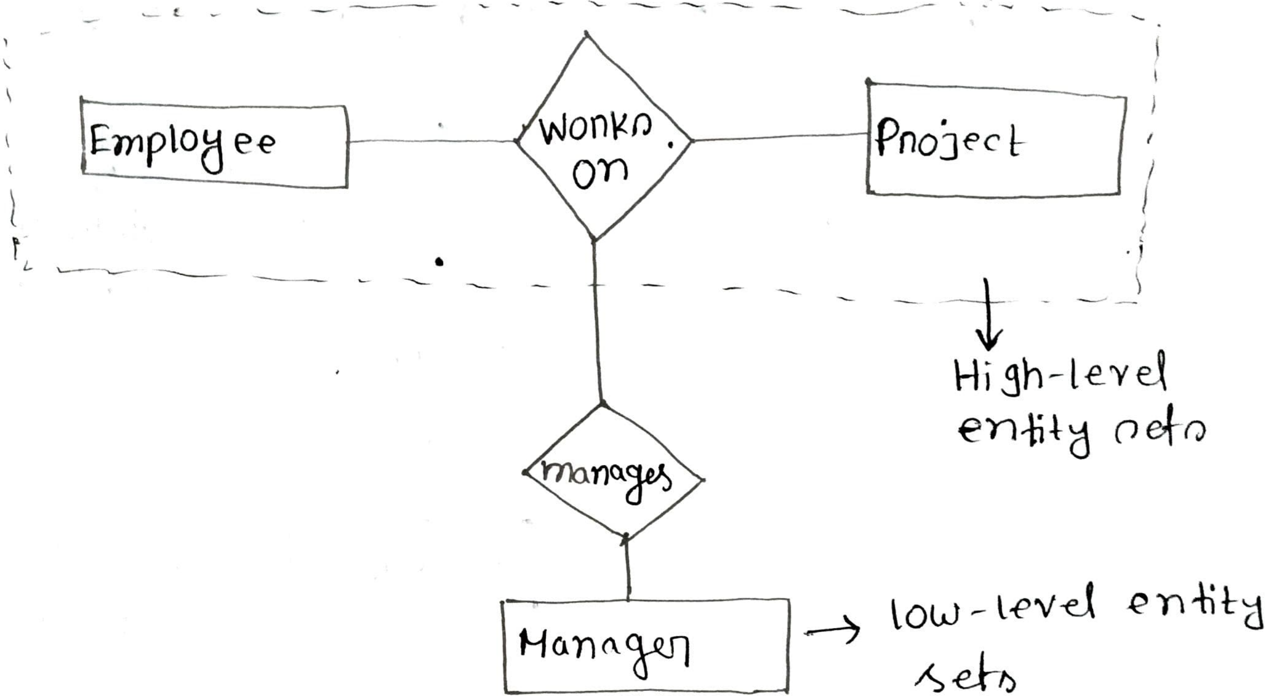
One limitation of E-R model is it cannot express relationships among relationships. To overcome this limitation, we use aggregation. Aggregation is an extended E-R feature. In aggregation, the relation between two entities is treated as a single entity. And relationship with its corresponding entities is aggregated into a higher level entity.

Name: Syeda Nowshin Ibnat

ID: 17183103020

Two examples of aggregation:-

Example - 1



In this example-1, the relation between employee and project together is acting as an Entity. Which is in relationship with another entity Manager.

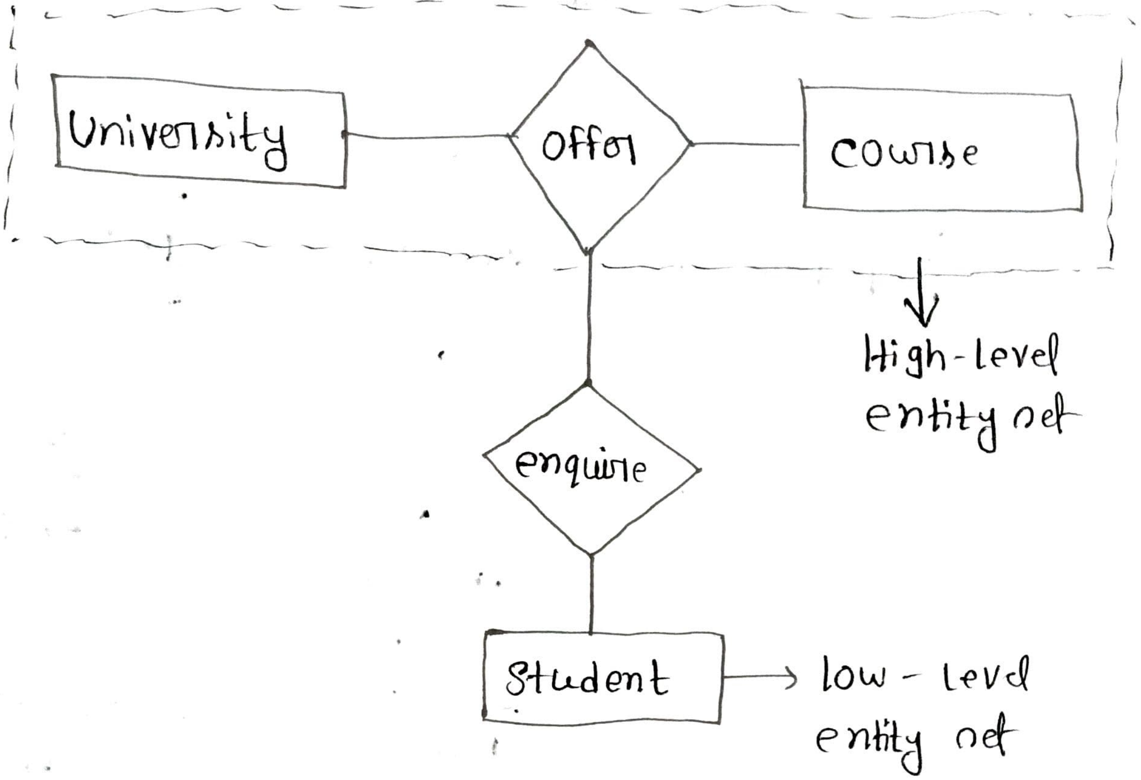
Here, Manager manages project and employee both. And it is a low-level entity set.

17

Name: Syeda Nowshin Ibnat

ID: 17183103020

Example-2



In this example-2, the relationship between University and course together, is acting as an Entity, which is in relationship with another entity Student. Now, in real world, if a student visits the university, he/she will never enquire about the university only on just about the course, rather he/she will ~~ask~~ enquire about both.