NAME: Sayed Muslim shah
ID#: 14856
MODULE: Bachelors {Software Engineering}
SEMESTER: 4
SECTION: B
SUBJECT: Database systems
INSTRUCTOR: Mam Rimsha Khan

1) **Which attribute in the following table is a candidate key? Assume that no more data will ever be added to this table.**

ID	Name	Semester	Department	Cell
1	Sania	1	CS	03334324234
2	Romaisa	1	CS	03335399123
3	Alina	1	CS	03150034224
4	Ayeza	3	CS	03455559822

ANSWER:

CONDIDATE KEY:

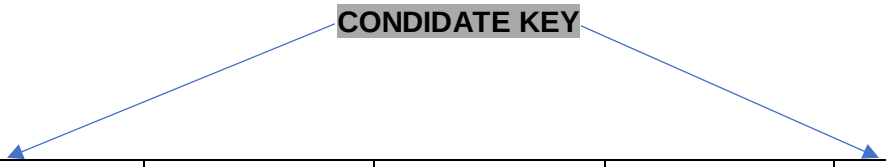
CANDIDATE KEY is a set of attributes that uniquely identify tuples in a table. Candidate Key is a super key with no repeated attributes. The Primary key should be selected from the candidate keys. Every table must have at least a single candidate key. A table can have multiple candidate keys but only a single primary key.

Properties of Candidate key:

a) It must contain unique values.
b) Candidate key may have multiple attributes.
c) Must not contain null values.
d) It should contain minimum fields to ensure uniqueness.
e) Uniquely identify each record in a table.

EXAMPLE:

In the given table Stud ID, Cell are candidate keys which help us to uniquely identify the student record in the table.



ID	Name	Semester	Department	Cell
1	Sania	1	CS	03334324234
2	Romaisa	1	CS	03335399123
3	Alina	1	CS	03150034224
4	Ayeza	3	CS	03455559822

2) **What is Data Redundancy and Data Integrity?**

ANSWER:

DATA REDUNDANCY: Data redundancy occurs when the same piece of data is stored in two or more separate places and is a common occurrence in many businesses. As more companies are moving away from soloed data to using a central repository to store information, they are finding that their database is filled with inconsistent duplicates of the same entry. Although it can be challenging to reconcile — or even benefit from — duplicate data entries, understanding how to reduce and track data redundancy efficiently can help mitigate long-term inconsistency issues for your business.

Advantages of data redundancy:

Alternative data backup method:

Backing up data involves creating compressed and encrypted versions of data and storing it in a computer system or the cloud. Data redundancy offers an extra layer of protection and reinforces the backup by replicating data to an additional system. It's often an advantage when companies incorporate data redundancy into their disaster recovery plans.

Better data security:

Data security relates to protecting data, in a database or a file storage system, from unwanted activities such as cyberattacks or data breaches. Having the same data stored in two or more separate places can protect an organization in the event of a cyberattack or breach — an event which can result in lost time and money, as well as a damaged reputation.

Faster data access and updates:

When data is redundant, employees enjoy fast access and quick updates because the necessary information is available on multiple systems. This is particularly important for customer service-based organizations whose customers expect promptness and efficiency.

Improved data reliability: Data that is reliable is complete and accurate. Organizations can use data redundancy to double check data and confirm it's

Disadvantages of data redundancy:

Possible data inconsistency:

Data redundancy occurs when the same piece of data exists in multiple places, whereas data inconsistency is when the same data exists in different formats in multiple tables. Unfortunately, data redundancy can cause data inconsistency, which can provide a company with unreliable and/or meaningless information.

Increase in data corruption:

Data corruption is when data becomes damaged as a result of errors in writing, reading, storage, or processing. When the same data fields are repeated in a database or file storage system, data corruption arises. If a file gets corrupted, for example, and an employee tries to open it, they may get an error message and not be able to complete their task.

Increase in database size:

Data redundancy may increase the size and complexity of a database — making it more of a challenge to maintain. A larger database can also lead to longer load times and a great deal of

headaches and frustrations for employees as they'll need to spend more time completing daily tasks.

Increase in cost:

When more data is created due to data redundancy, storage costs suddenly increase. This can be a serious issue for organizations who are trying to keep costs low in order to increase profits and meet their goals. In addition, implementing a database system can become more expensive.

DATA INTEGRITY:

The term data integrity refers to the overall accuracy, completeness, and reliability of data. It can be specified by the lack of variation between two instances or consecutive updates of a record, indicating that your information is error-free. It also corresponds to the security of data pertaining to regulatory compliance.

Data integrity is preserved by an array of error-checking and validation procedures, rules, and principles executed during the integration flow designing phase. These checks and correction procedures are based on a predefined set of business rules. For instance, the rules dictate to filter out the data with incorrect date or time values.

The importance of data integrity is also evident when creating relationships between disparate data elements. It ensures that the data transferring from one stage to another is accurate and error-free. This way, when the information is written into the database, its trustworthiness is implied regardless of the duration for which it is stored or the frequency of access.

advantages of Data Integrity:

Following are the benefits or advantages of Data Integrity

➡It ensures quality in the product and/or service.

➡It ensures safety and privacy of customers e.g. patients, social media users etc.

➡It increases confidence of consumers to use online digital applications and tools. This helps to increase businesses in digital economy.

➡The data integrity helps to protect data from end to end transfer over transmission medium.

➡Stored procedures can be used with ease in order to have complete control of data access.

disadvantages of Data Integrity:

➡The DBMS (Database management system) should have capability to enforce data integrity for all the applications which use the data.

➡It lacks in structural independence.

➡The programming is complex to implement.

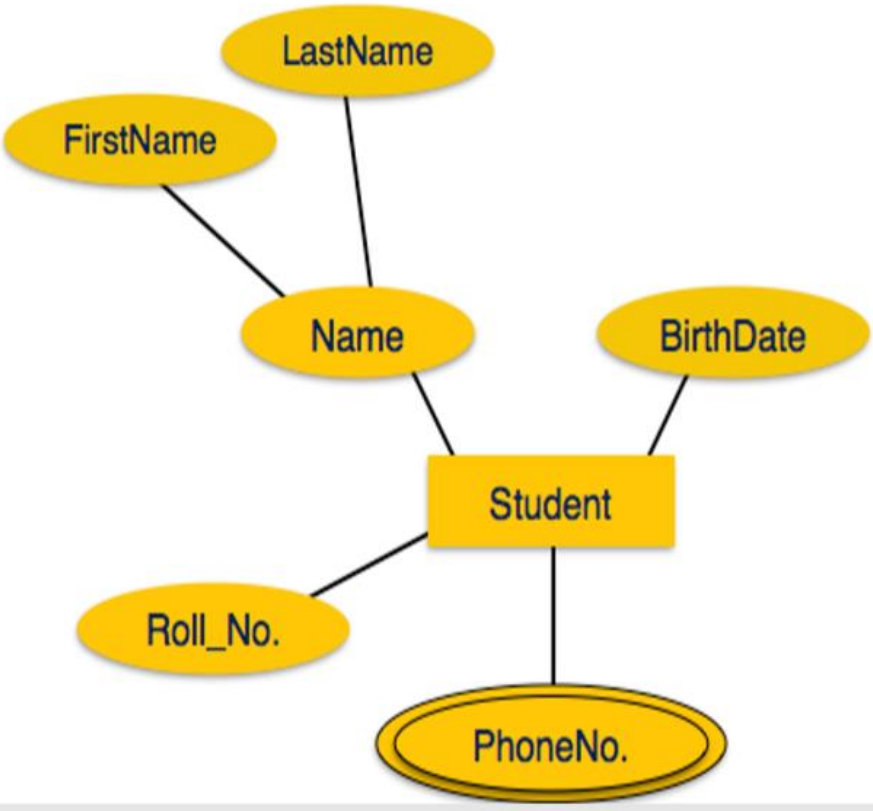
➡It is complex to implement across the entire system due to many new ICTs (Information and Communication Technologies).

3 How a multivalued composite attribute is represented in Conceptual Model. Show with example.

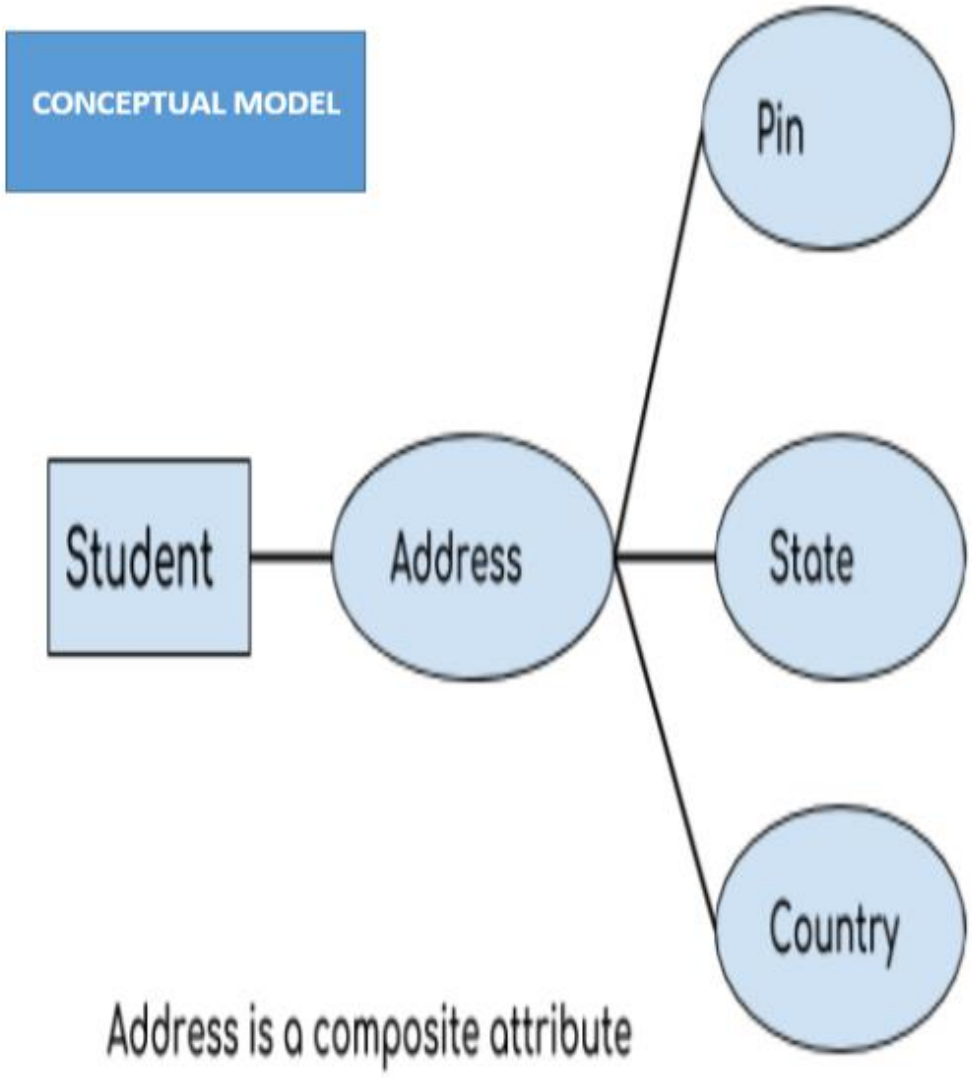
ANSWER:

Multivalued attribute:

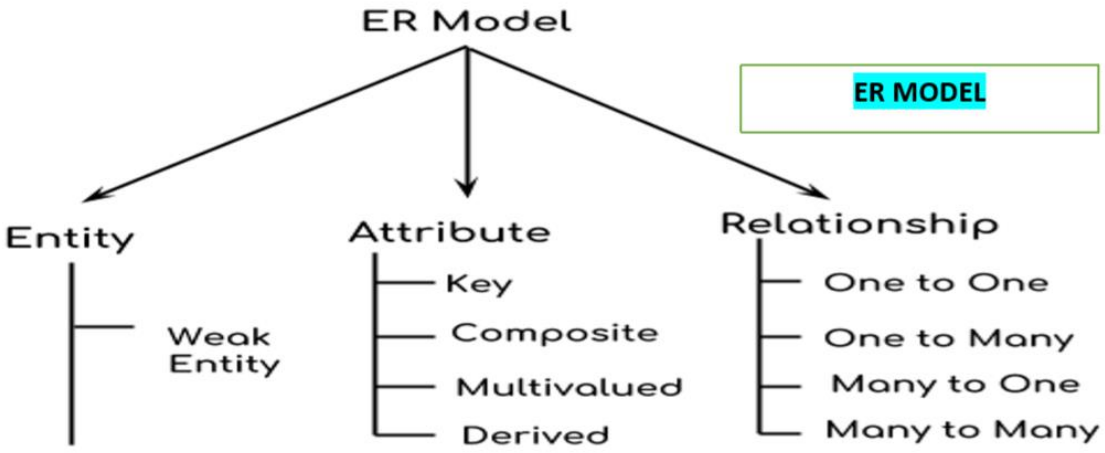
An attribute that can hold multiple values is known as multivalued attribute. It is represented with double ovals in an ER Diagram. For example – A person can have more than one phone numbers so the phone number attribute is multivalued.



Composite attribute is represented in Conceptual Model:



Most attributes are atomic attributes. Multivalued A multivalued attribute may have one or more value for a particular entity. ... A typical example of a composite attribute is a person's address, which is composed of atomic attributes, such as City, Zip, and Street.



Components of ER Diagram

4. How is there 'reduced maintenance' in database approach?

ANSWER:
→Database applications in a busy corporate setting, yet take considerable time over meticulous database design, extensive constraints, automated tests, error logs, and defensive coding. Why? Because it cuts down on the subsequent need for maintenance

It can cost more to maintain a mature software application in use than it did to develop it in the first place, and this is particularly the case when we are maintaining systems built on top of an RDBMS. To keep an application in use, both developers and DBAs may be forced to spend considerable time on maintenance tasks such as

- Fixing bugs and deploying the fixes
- Changing the functionality because the requirements have changed
- Cleaning up the data
- Dealing with concurrency issues; troubleshooting deadlocks, for example
- Speeding up slow queries

The Maintenance tasks
It can cost more to maintain a mature software application in use than it did to develop it in the first place, and this is particularly the case when we are maintaining systems built on top of an

RDBMS. To keep an application in use, both developers and DBAs may be forced to spend considerable time on maintenance tasks such as

Fixing bugs and deploying the fixes
Changing the functionality because the requirements have changed
Cleaning up the data
Dealing with concurrency issues; troubleshooting deadlocks, for example

Speeding up slow queries

If the system that we develop then experiences many of these problems in production, it then also erodes both the users' confidence and the teams' morale, on top of the raw cost. When our users encounter issues or deal with inconsistent behavior of the system, it prevents them from completing their tasks effectively. When we who are tasked with maintaining the system have to fix so many problems on top of developing new features, it is bad for our work-life balance. Also, the time spent troubleshooting and fixing is the time not spent on doing other, usually more interesting and productive, work. As a result, unreliable systems may eventually lose both users and developers.

Writing Maintainable Code

If an application is intended to provide a long-term solution, it must be maintainable. To achieve this, it usually makes sense to use development practices that ensure that our system is robust and easy to change, thereby reducing some of the maintenance burden. This is especially true in Agile environments: Here, we are always just a few days away from the next release. It is difficult to justify spending a day on the interesting challenge of tuning queries or troubleshooting deadlocks when all the pressure is to concentrate on tomorrow's deployment.

Steps towards making databases more maintainable

I am not going to make any blanket recommendations or hard-and-fast rules – those solutions and approaches that make sense in our environment might be an overkill or just wrong in some other circumstances. Yet I hope that some of the advice in this article might be useful for somebody else, so let me share the steps we have made to reduce maintenance costs by using better development practices.

Insulate the database behind a well-established interface.

We do not expose tables and views to our users. Instead, we provide an API of stored procedures. This allows us to change the procedures or the underlying tables without the risk of breaking applications. A little bit of extra work needed to wrap a DML statement in a procedure is a highly useful insurance against possible changes later on, making the changes down the road much easier?

Further in this article we shall discuss many examples when this insulation proves to be extremely useful.

Note: We decided against exposing views because we want to be able to break a complex query into smaller and simpler parts when needed, storing intermediate results in table variables or temporary tables.

Automate database testing, and maintain good test coverage

If longevity is required of a system that is built on top of an RDBMS, it is essential to have a rigorous harness of automated tests. Our automated tests allow us to dramatically reduce the number of bugs. Also they enable us to change our system much more easily, further reducing maintenance costs. For more detail, refer to this article: http://www.simple-talk.com/content/print.aspx?article=1419.

According to Darwin's theory, it is not necessarily the ones who are currently the strongest and the fastest who survive in the long term. It is the most adaptable species that eventually wins because they retain the ability to evolve and adapt to changing requirements.

Without automated test harnesses, it is difficult for RDBMS systems to survive by adapting to changing environments

Design tables defensively

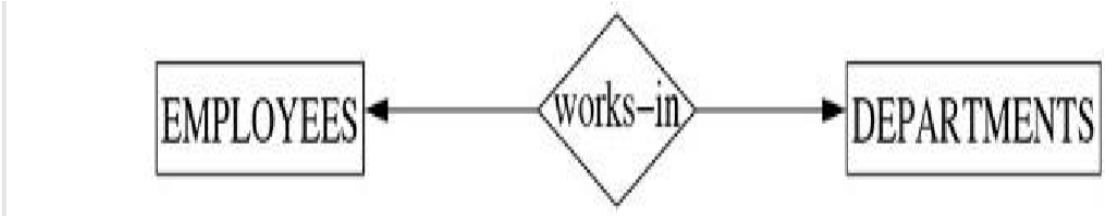
In some cases, it is possible to design tables defensively, so that they are less likely to need change, and it is easier to change them when the need comes. Suppose, for example, that right now the following table meets our current needs perfectly well (for brevity, we have skipped the check constraint which verifies that the number is in the proper range, as well as other columns not relevant to our example.

5. How are the following represented using ER Diagram: Mandatory one, Mandatory many, Optional one, Optional Many?

ANSWER:

Mandatory one:

→relationship exists when zero or one instance of entity A can be associated with zero or one instance of entity B, and zero or one instance of entity B can be associated with zero or one instance of entity A.

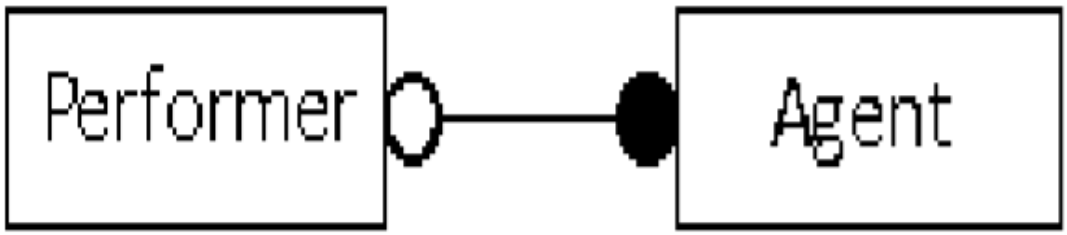


Mandatory many:

→ In relational database design, a one-to-many (1:N) relationship exists when, for one instance of entity A, there exists zero, one, or many instances of entity B; but for one instance of entity B, there exists zero or one instance of entity A.

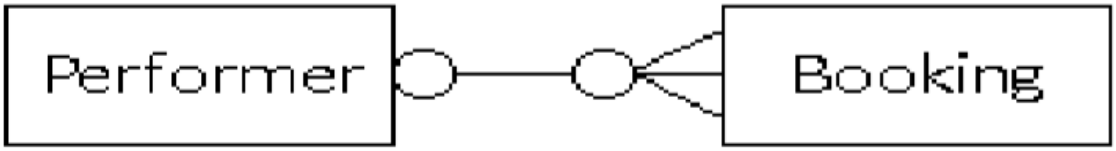
OPTIONAL ONE:

A member of staff does not have to be placed in a department, but all departments must have at least one member of staff.



Optional Many:

A performer might have one or more bookings; a booking might be associated with a performer.



In this case, a booking could be for an exhibition as it is optional for a booking to involve a performer, as indicated by the hollow circle. A performer might decline to accept any bookings; this is acceptable, as it is optional for a performer to have a booking (shown by the hollow circle).

6. Why is there an explicit need of backup in database approach?

ANSWER:

DATABASE BACKUP APPROACH:

Digital age has pushed data to the forefront of business functions, cementing data as one of the most valuable assets a business can have. Many businesses have adopted database software as a means to manage their data, including functionality to add, edit, and remove data as needed. Databases particularly excel in data querying; based on the database type, users can search for data using a massive and customizable range of parameters to get back exactly the information they need. They are also often integrated into a variety of business functions, doing everything from storing transactional information and user data to stock quantities and more.

Because the data stored in business databases is so important to daily functions, it's imperative that businesses have their databases backed up. Equipment failures, data corruption, user error, and other calamities can massively affect both internal users and customers. Database backup software helps ensure that companies always have more copies of their business data, just in case it's needed. These backups can be scheduled or manually run, and they can back up fully, differentially, incrementally, or any combination thereof.

Back up one or several database types to a chosen destination

Set up automated or manual backups at the needed extensiveness

Integration with other backup and IT infrastructure and management solutions

Responsibility over databases typically falls to a company's database administrator(s), or a database team if the company has those kinds of resources. so database backup solutions will likely be used most by those administrators or the database team. While databases themselves can be highly accessible for any member of a company, database backup software—as a part of the database maintenance ecosystem—should be handled by those explicitly focused on maintaining a company's database integrity.

Q 2: Draw an ERD from the following business rules: Use proper notations for the type of attributes?

A schema needs to capture all the information that An Art gallery need to maintain.

- The database shall keep information about Artists, their names (which are unique), birthplace, age, and style of art.
- For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g. painting lithography, sculpture, photograph), and its price must be stored.

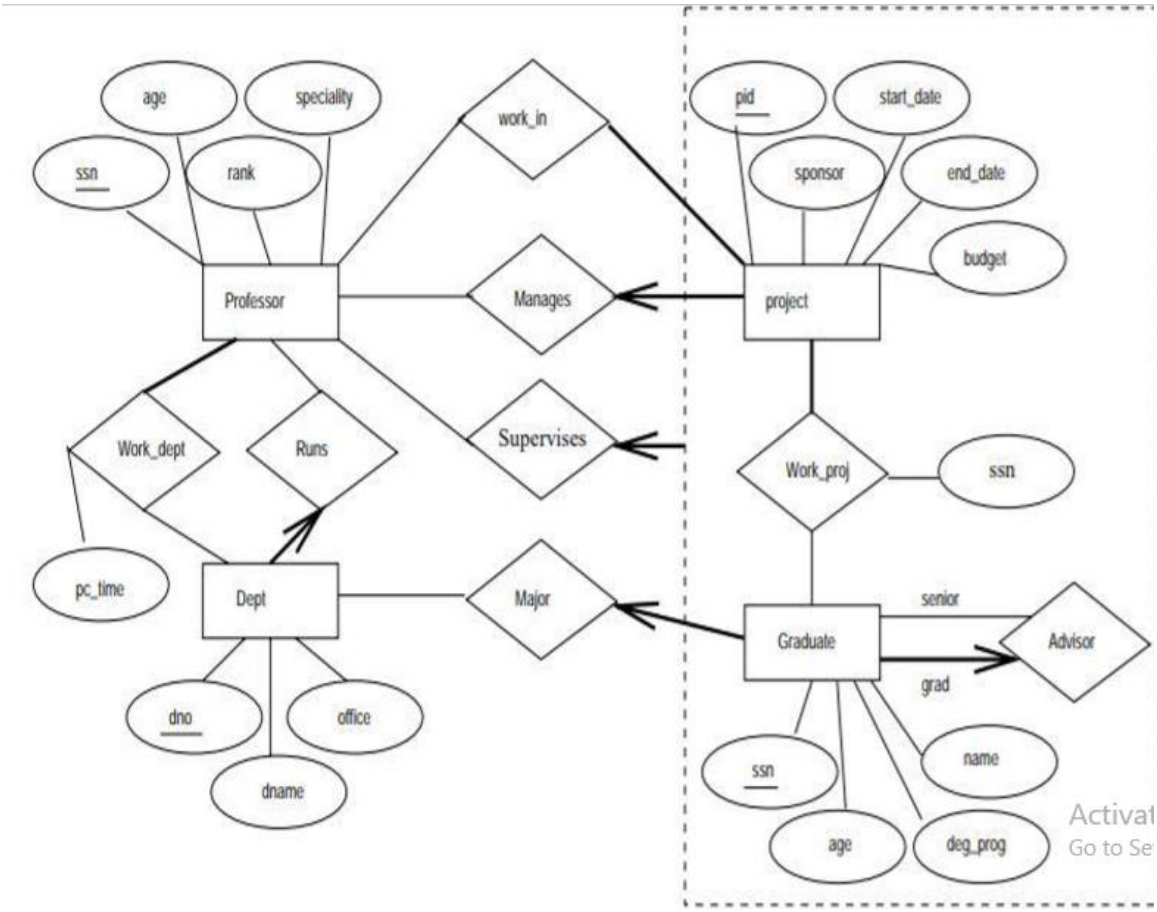
• Pieces of artwork are also classified into groups of various kinds for example, potraits, still lifes, works by Picasso, or works of the 19th century.

• A given piece may belong to more than one group.

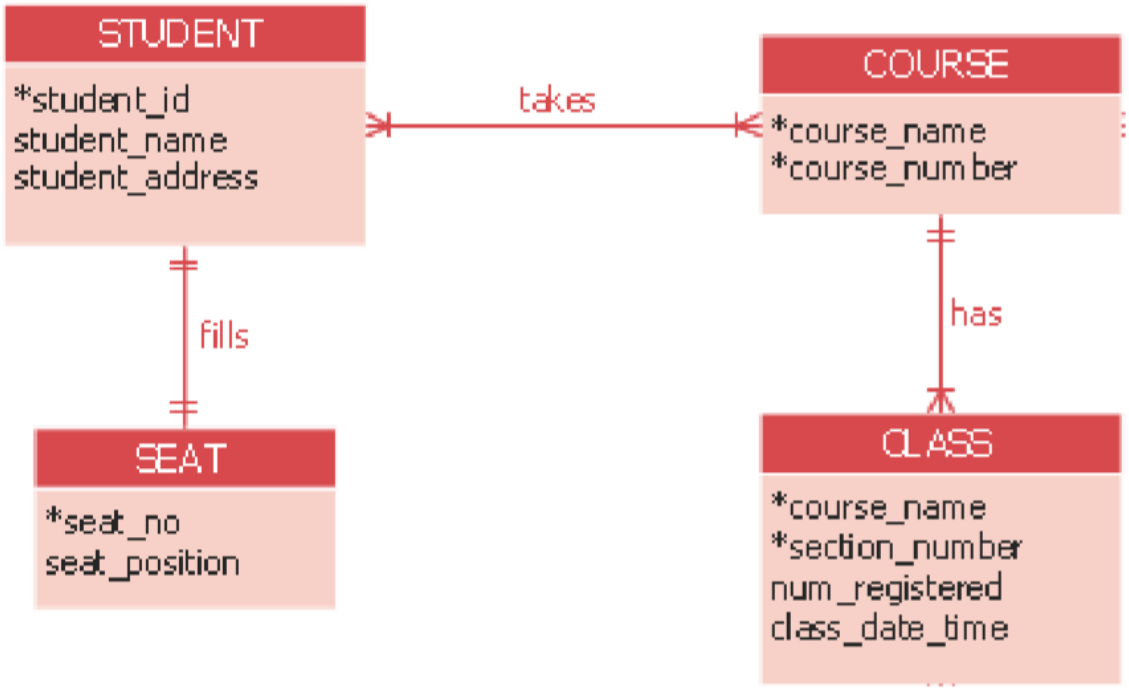
• Each group identified by a name that describes the group.

• Finally, galleries keep the Customer's unique name, address, total amount of dollars spent in the gallery and the artist and groups of the art that the customer tends to like

ANSWER:



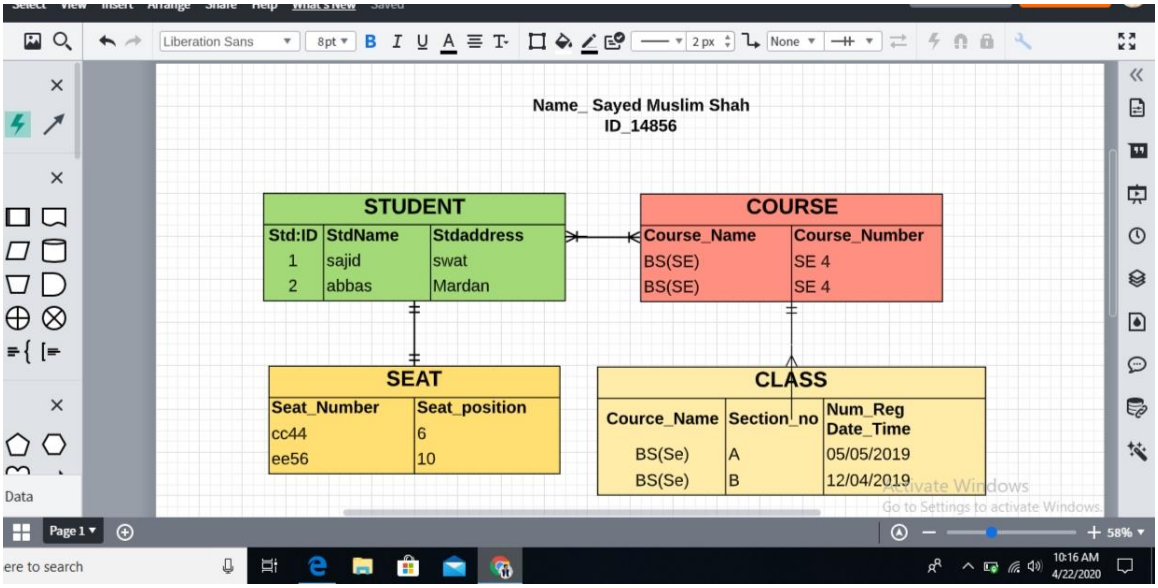
Q 3: Convert the following Conceptual Model to Relational Model.



ANSWER:

Mapping Process:

- Create table for weak entity set.
- Add all its attributes to table as field.
- Add the primary key of identifying entity set.
- Declare all foreign key constraints.


Name_ Sayed Muslim Shah
ID_14856
STUDENT
Std:ID
StdName
Stdaddress
1
sajid
swat
2
abbas
Mardan
COURSE
Course_Name
Course_Number
BS(SE)
SE 4
BS(SE)
SE 4
SEAT
Seat_Number
Seat_position
cc44
6
ee56
10
CLASS
Cource_Name
Section_no
Num_Reg
Date_Time
BS(Se)
A
05/05/2019
BS(Se)
B
12/04/2019
Activate Windows
Go to Settings to activate Windows.
Data
Page 1
58%
10:16 AM
4/22/2020