Name:- Umar Farooq

Reg No:-14740

Subject:-Data Base system

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.

CONDIDATE KEY

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

Q2:-What is date Redundancy and Date integrity?

Ans:

Date Redundancy:

- Repetition or superfluity of date
- Reduces date consistency
- Negative impact

Date integrity:

- Maintenance and assurance of the accuracy and consistency of date over the entire life cycle
- Helps to improve date consistency
- Positives

Q3:- How a multivalued composite attribute is represent in conceptual model. Show with example?

Ans:- Multivalued Attributes. A multivalued attribute of an entity is an attribute that can have more than one value associated with the key of the entity.

For example, a large company could have many divisions, some of them possibly in different cities

Q4:- How is there 'reduced maintenance' in date base approach?

Ans:- →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.

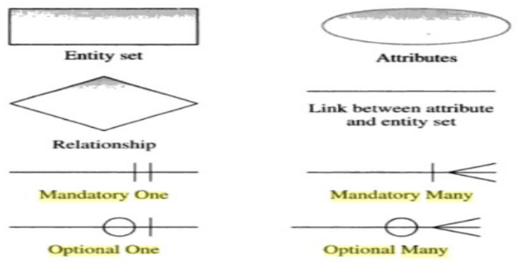
Without automated test harnesses, it is difficult for RDBMS systems to survive by adapting to changing environments

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.

Q5:-How are the following represent using ER diagram: Mandatory one, Mandatory many, optional one, optional Many?

Ans:-



Q6:- Why is there an explicit need to backup in date base approach?

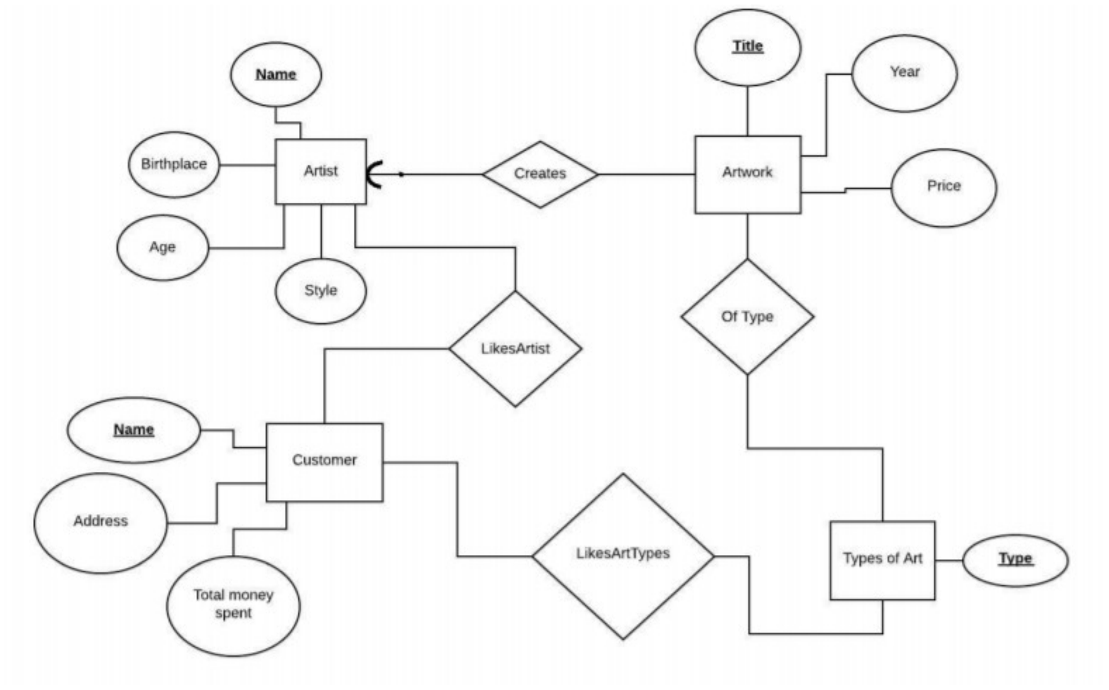
Ans:- The explicit need of backup in database approach is because for a centralized shared database to be accurate and available all times, comprehensive procedure is required to be developed and used for providing backup copies of data and for restoring a database when damage occurs.

Q2:-Draw an ERD from the following business rules: use the proper notations for the types of attributes

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

- The date base shall keep information about Artist , their name(which are unique), birthplace, age, and style of art.
- For each piece of artwork, the artist, the yeart it was made, its unique title, is types of the art(e.g. painting lithograph, sculpture, photograph) and its price must be stored.
- Piece of artwork are also classified into groups of various kind for the example, portraits, still lifes, works by piece.
- A given piece may belong to more than one group.
- Each group identified by a name that describe the group
- Finally galleries keep the customer's unique name, address, total amount of dollars spent in the gallery and the artist and group of the art that the customer tends to like.

Ans:-

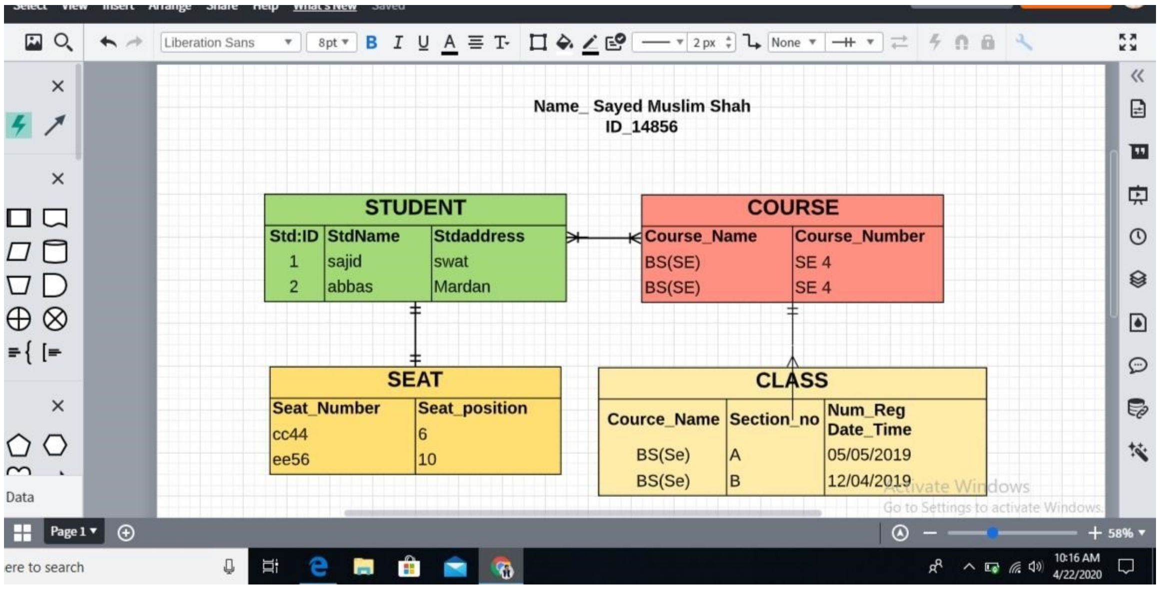


Q3:- convert the following Conceptual Model to Relational model?

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