

MCS-023 solved assignment

july 2017-january 2018 session

1. List and describe briefly all the possible applications of a database management system for any IGNOU's Study Centre.

A.1. DataBase Management System

A Database management system is a computerized record-keeping system. It is a repository or a container for collection of computerized data files. The overall purpose of DBMS is to allow the users to define, store, retrieve and update the information contained in the database on demand. Information can be anything that is of significance to an individual or organization.

All the possible applications of a Database Management System in any IGNOU'S Regional Center :-

(i) Database is used to store students' details in a place

We know Database is used for storing Data and Information. In IGNOU'S Regional Center all data and Information related to student, staff and employees are stored from where we can access data and Information, update and also Delete the Data and Information.

(ii) Easy Query processing and management

From DBMS we can easily generate Query and we can easily manage it. And we can easily access Query in our database.

For Ex: `SELECT * FROM Students;`

where `SELECT` is a Query Statement,

(*) All information can access through *(astrik sign)

And, student is the name of Table.

(iii) Easy to understand and user friendly

The language we use in DBMS is easily understandable. Query Statements like `SELECT`, `DELETE` & `UPDATE` is very easy for users. Because we use a simple english words or Statements.

(iv) Store Students marks evaluation details

We easily get our Evaluation marks details that is stored in DBMS of Regional Center applications that is easily available for Enrolled Students.

(v) Store Employee details.

If we want to Know the details about the any of the Employee then we can easily get from the DBMS that is already stored in DBMS. To know which Employee is responsible for the which work that is assigned by Administrator.

(vi) Store final year project Submission.

In IGNOU programmed application the students have to create Project on any topics and the project is Submitted by Students in Regional Center. If student want to see the details of submitted project, they get from there.

(vii) Easy to implement Security and integrity of data

DBMS has four methods to maintain data integrity: normalizing data, defining business rules, providing referential integrity and data validation. Data normalization is a process where unnecessary dependencies and repeating groups of data are eliminated. Business rules for data access control data handling and sharing between applications. Referential integrity is used to protect data from corruption. Data validation uses multiple processes to validate data in the application.

(viii) Fast access of information

In DBMS, all data of an organization is integrated into a single database file. The data is recorded in only one place in the database and it is not duplicated. Through which we can access of Information fast.

(ix) Data sharing in other department of IGNOU.

In DBMS, data can be shared by authorized users of the organization. The database administrator manages the data and gives rights to users to access the data. Many users can be authorized to access the same piece of information simultaneously. The remote users can also share same data. Similarly, the data of same database can be shared between different application programs.

(x) Better data accuracy

By controlling the data redundancy, the data consistency is obtained. If a data item appears only once, any update to its value has to be performed only once and the updated value is immediately available to all users. If the DBMS has controlled redundancy, the database system enforces consistency and accurate.

Q.2. Identify all the associated entities for a Regional Centre Management System, their corresponding attributes, relationships and cardinality and design an Entity-Relationship (ER) diagram for it.

2. Identify all the associated entities for a Study Centre Management System, their corresponding attributes, relationships and cardinality and design an Entity-Relationship (ER) diagram for it.

A.2.

Entity:-

An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. All these entities have some attributes or properties that give them their identity.

An entity set is a collection of similar types of entities. An entity set may contain entities with attribute sharing similar values. For example, a Students set may contain all the students of a school; likewise a Teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

Attributes

Entities are represented by means of their properties, called attributes. All attributes have values. For example, a student entity may have name, class, and age as attributes.

There exists a domain or range of values that can be assigned to attributes. For example, a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative, etc.

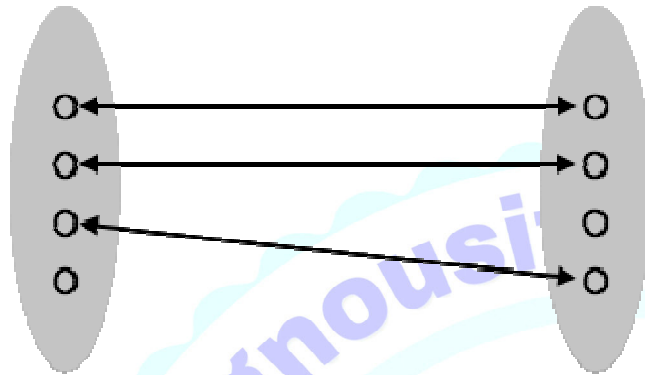
Relationship

The association among entities is called a relationship. For example, an employee works_at a department, a student enrolls in a course. Here, Works_at and Enrolls are called relationships.

Cardinality

Defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

One-to-one – One entity from entity set A can be associated with at most one entity of entity set B and vice versa.

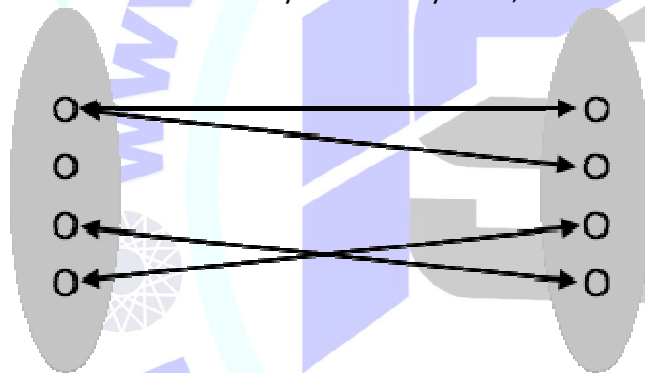


A

Sunil Poonia

B

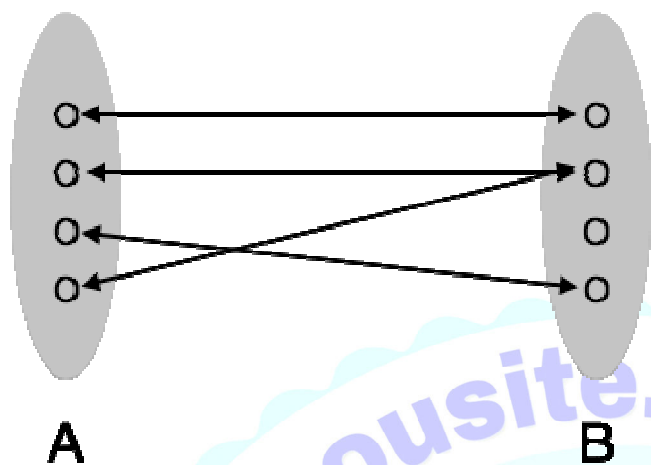
One-to-many – One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.



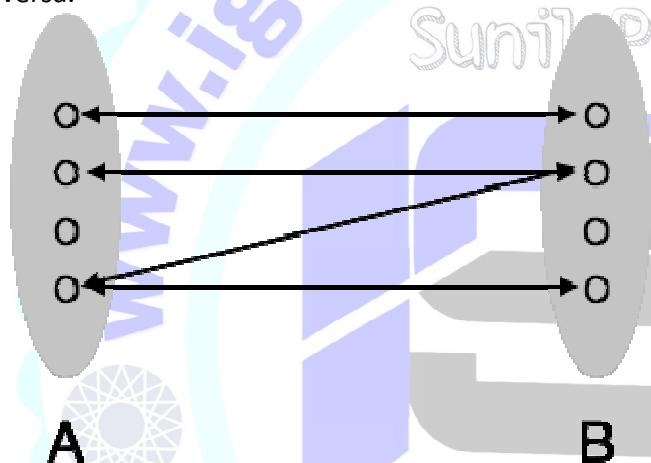
A

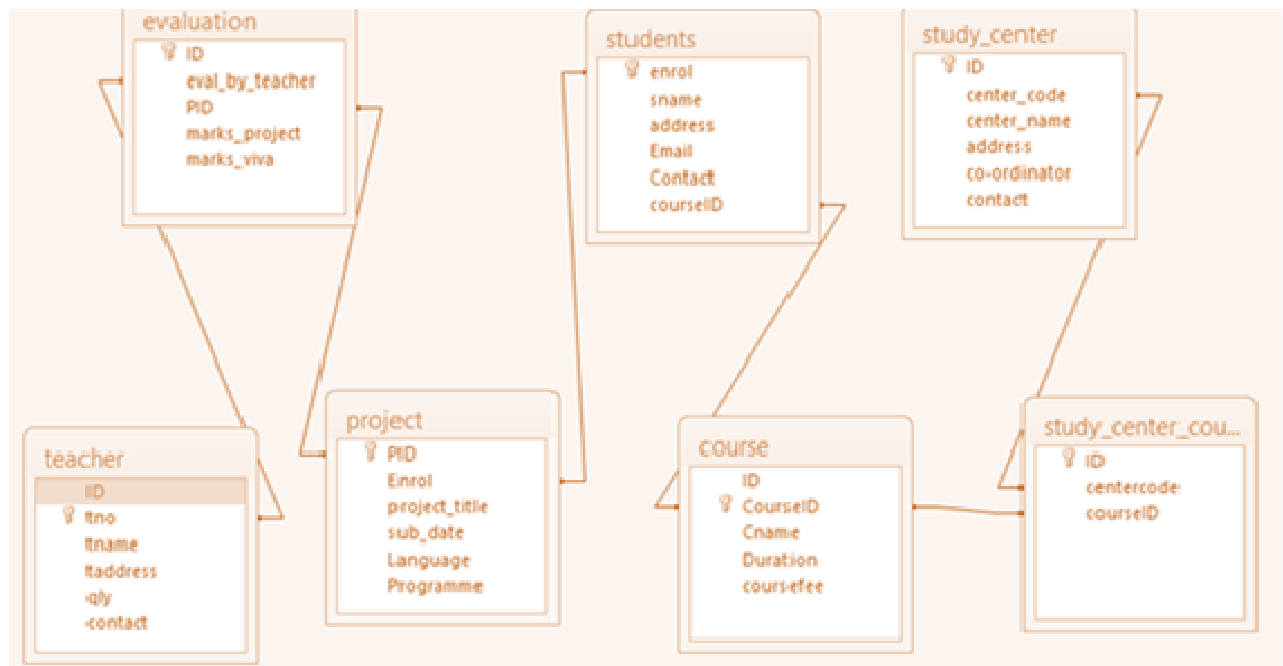
B

Many-to-one – More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.

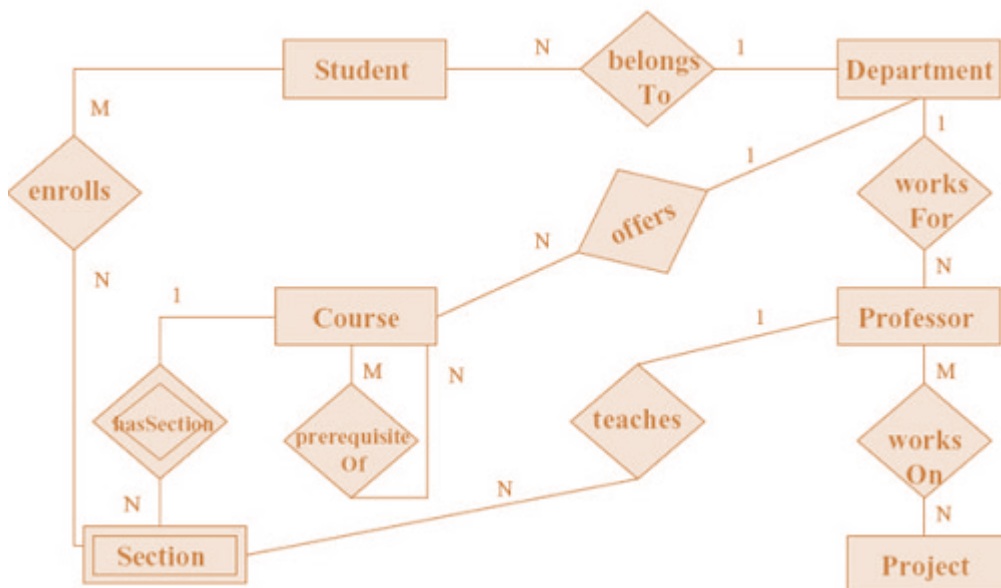


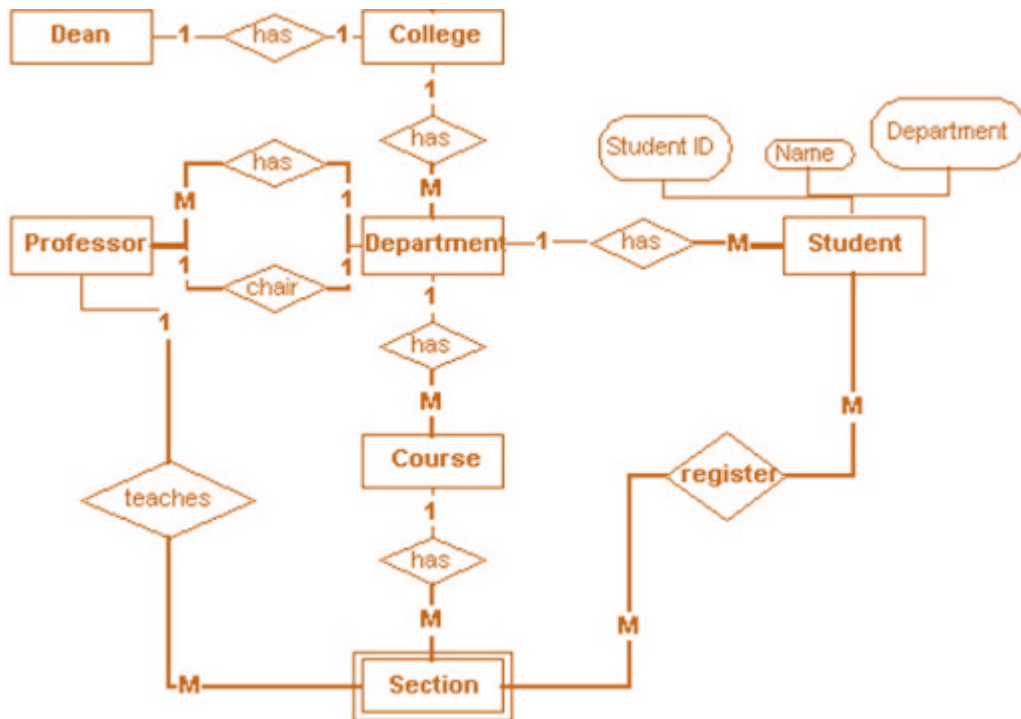
Many-to-many – One entity from A can be associated with more than one entity from B and vice versa.





Entity Relationship Diagram:-





3. Consider the E-R diagram of Question 2 and design the relational schema and the tables. Perform and show the Normalization till the required normal form. Implement the database using MS-Access and submit the screenshots along with your assignment response for this question.

A.3.
NORMALIZATION:-

Database Normalisation is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form by removing duplicated data from the relation tables. Normalization is used for mainly two purpose,

Eliminating redundant(useless) data.

Ensuring data dependencies make sense i.e data is logically stored.

Problem Without Normalization Without Normalization, it becomes difficult to handle and update the database, without facing data loss. Insertion, Updation and Deletion Anomalies are very frequent if Database is not Normalized. To understand these anomalies let us take an example of Student table.

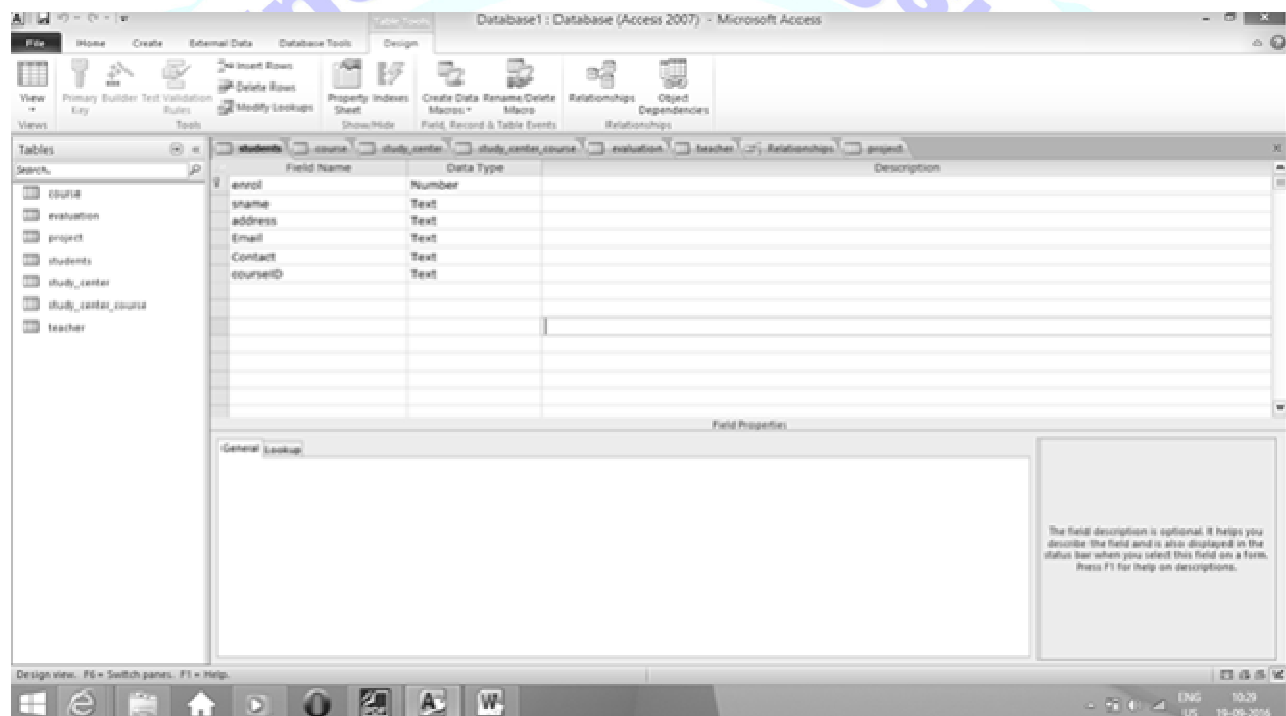
S_id	S_Name	S_Address	Subject_opted
401	Adam	Noida	Bio
402	Alex	Panipat	Maths

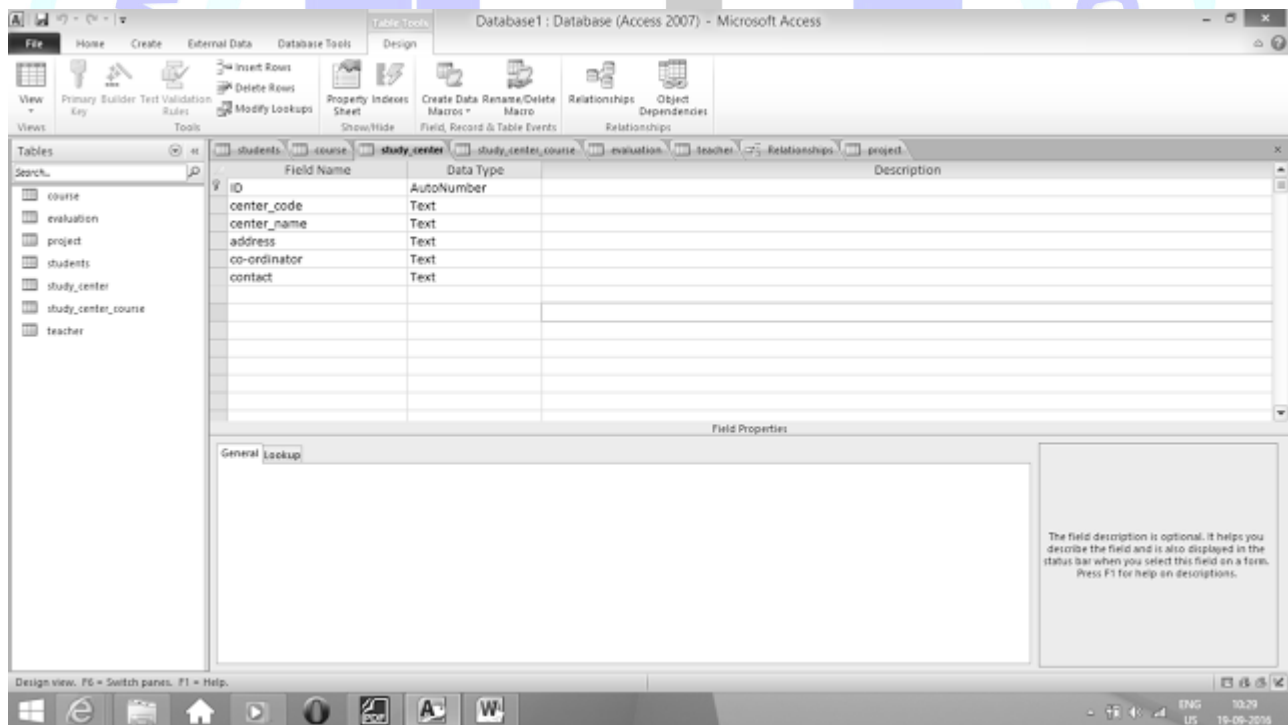
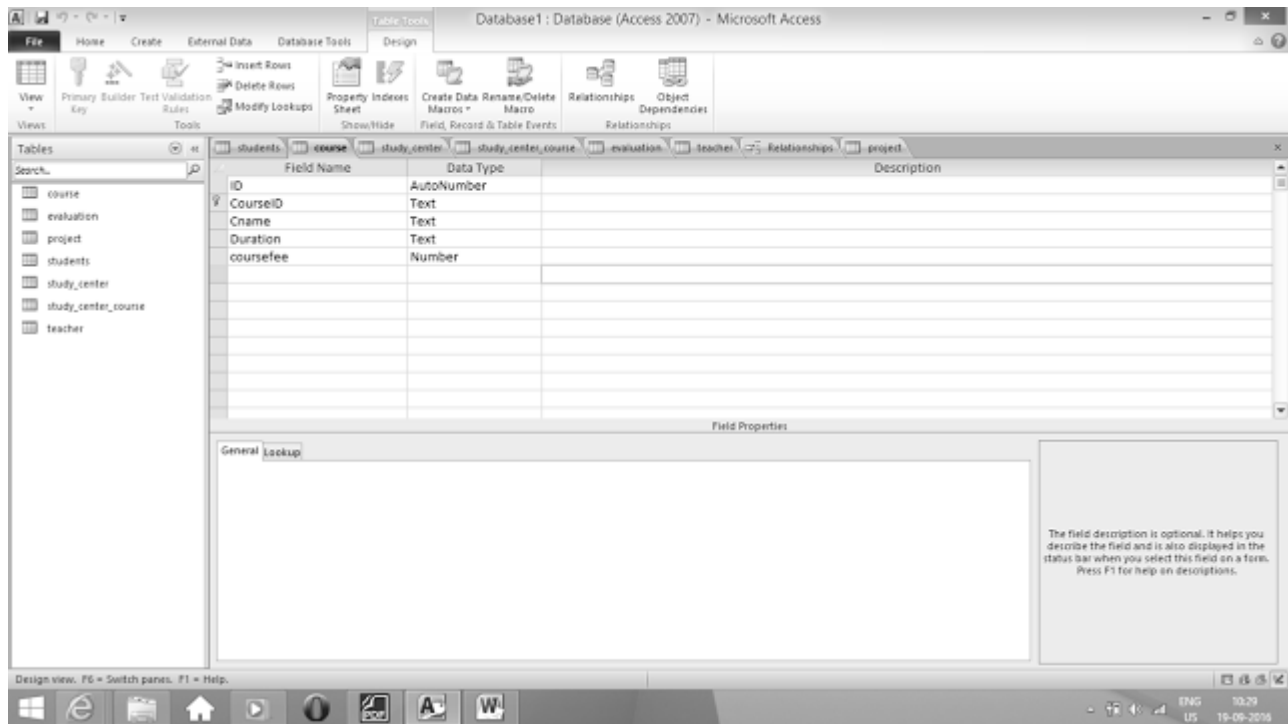
403	Stuart	Jammu	Maths
404	Adam	Noida	Physics

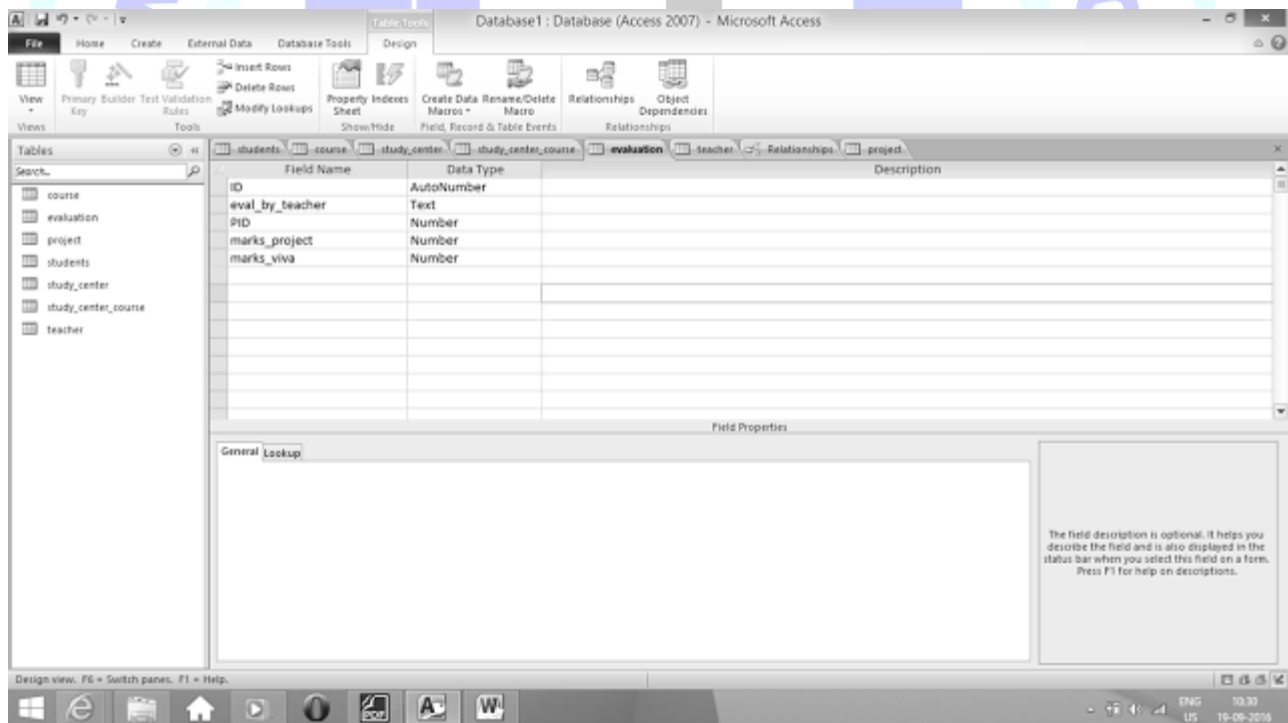
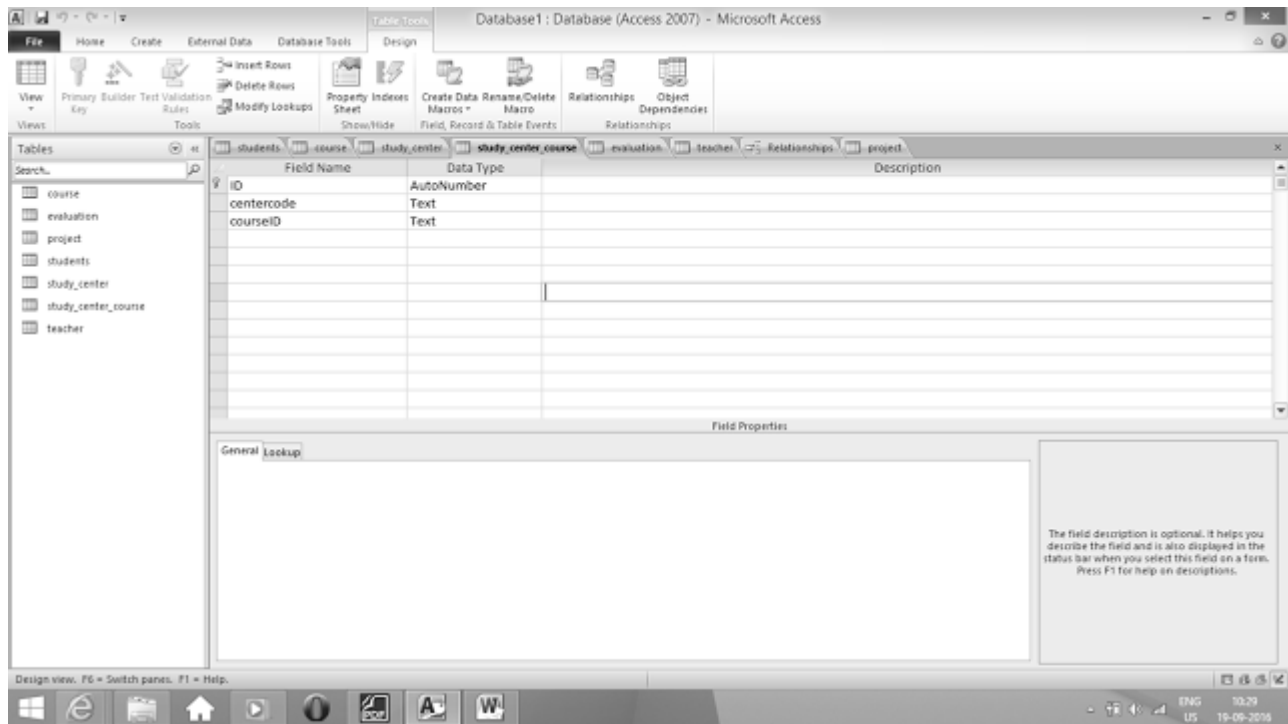
Updation Anamoly : To update address of a student who occurs twice or more than twice in a table, we will have to update S_Address column in all the rows, else data will become inconsistent.

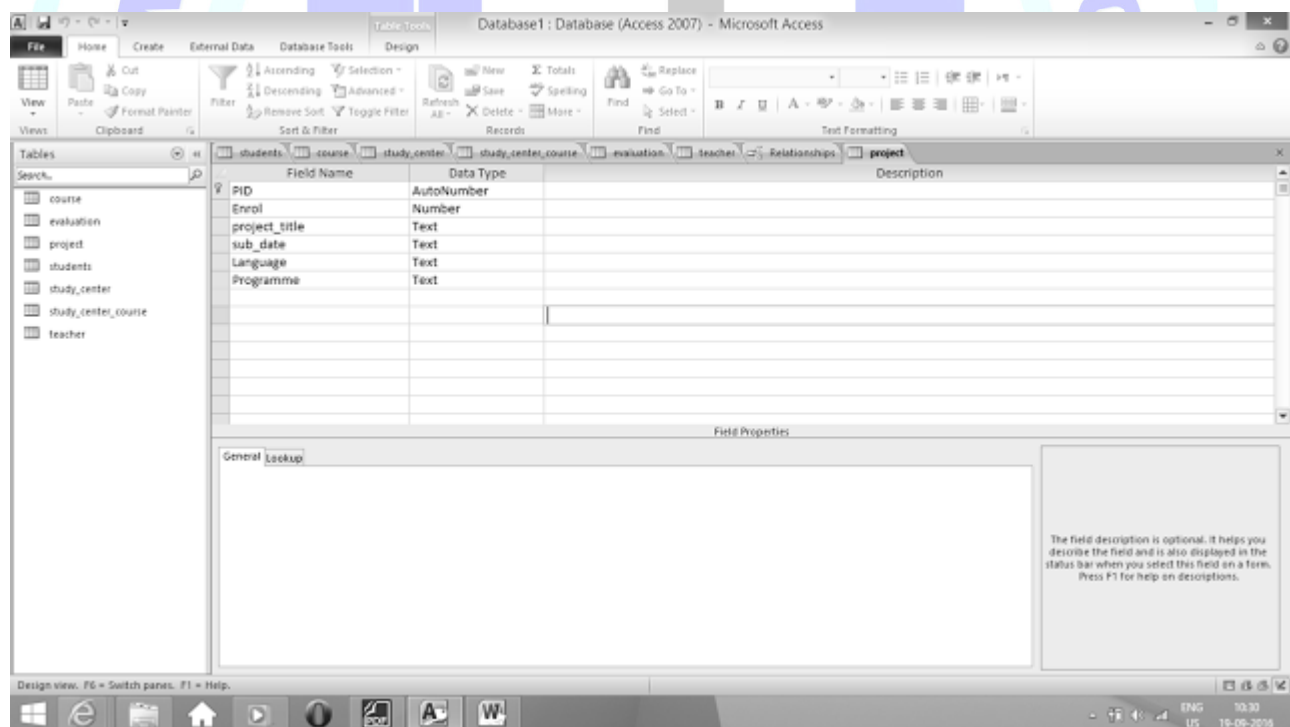
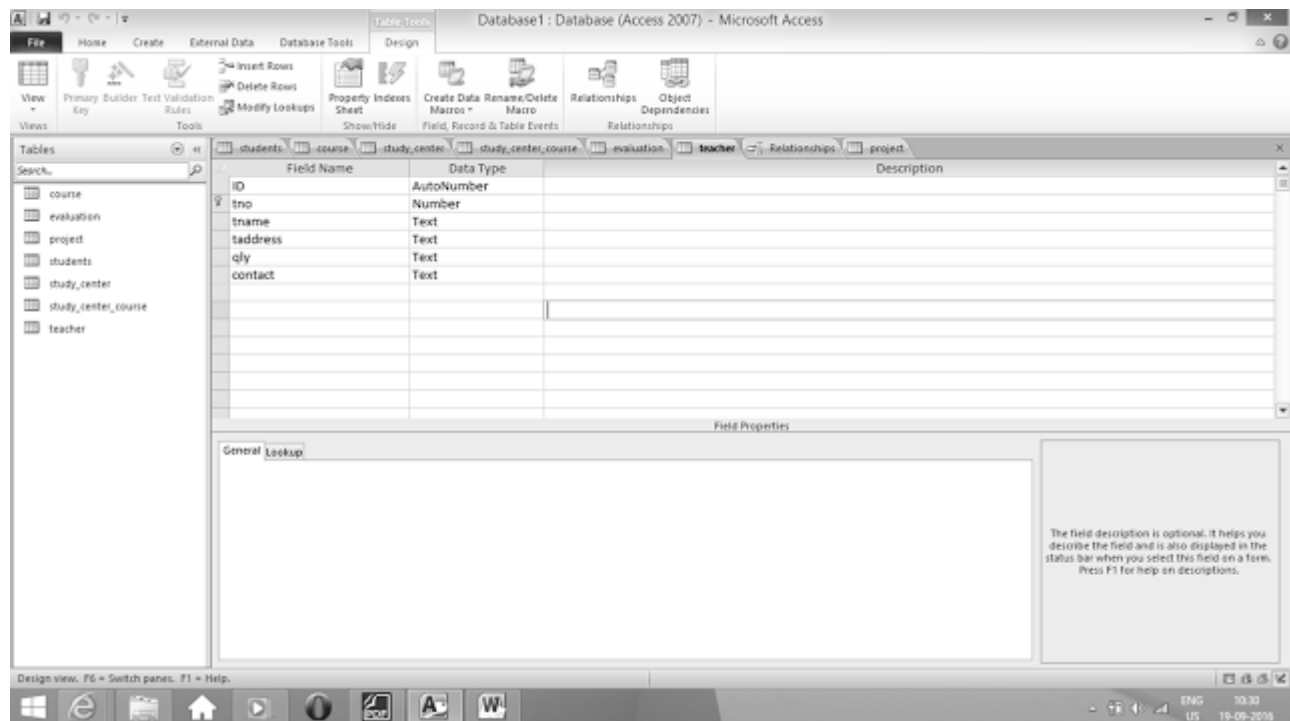
Insertion Anamoly : Suppose for a new admission, we have a Student id(S_id), name and address of a student but if student has not opted for any subjects yet then we have to insert NULL there, leading to Insertion Anamoly.

Deletion Anamoly : If (S_id) 401 has only one subject and temporarily he drops it, when we delete that row, entire student record will be deleted along with it.









4. Consider a “Project Management System” that maintains the database using the following tables:

Project (p_id, p_title, start_date, duration, end_date, c_id, cost)

Client (c_id, c_name, c_address, c_phone)

Employee (e_id, e_name, e_address, e_phone)

ProjectLeader (p_id, e_id)

Please note that an employee may be working on more than one project in the organization at a time. Write and run the following SQL queries on the tables:

- (a) Find the c_id and c_name of the clients who have given maximum number of projects to the company.
 - (b) Find the list of all the p_id's, their project details and project leaders of all the projects.
 - (c) Find the p_id's whose title starts with "a" or "A".
 - (d) Find the projects which will be completed by this month-end.
- (15 Marks)

(e) Find all the clients who have not given any project. Also find the person who is not working on any project at all.

Note: Make suitable assumptions, if any.

A.4.

Project (p_id, p_title, start_date, duration, end_date, c_id, cost)

Client (c_id, c_name, c_address, c_phone)

Employee (e_id, e_name, e_address, e_phone)

ProjectLeader (p_id, e_id)

(a) Find the c_id and c_name of the clients who have given maximum number of projects to the company.

(a) Select c_id, c_name from clients, from Project where Project.p_id=ProjectLeader.p_id ;

Explanation:- Select is Query Statement from where we can select all data or selected data from the table. Here the tables are Members from where we can access c_id and c_name attributes , and the table Project from where we can access actual Maximum number of projects.

Project.p_id=ProjectLeader.p_id , this is the way of accessing information from one(Member table) to another (Project)table.

(b) Find the list of all the p_id's, their project details and project leaders of all the projects.

(b) Select * from Project where Project.p_id = ProjectLeader = p.id;

Explanation:- Select is Query Statement from where we can select all data or selected data from the table. It is simply access all data or information from the table "Project", and employee details from the Project Leader.

(iii) Find the p_id's whose title starts with "a" or "A". .

(iii) Select * from Project where p_id ="a" or "A";

Explanation:- Select is Query Statement from where we can select all data or selected data from the table. Here the tables are Project from where we can access project with a specific character that start with "a" or "A".

(iv) Find the projects which will be completed by this month-end.

(iv) Select * from Project where end_date= 'month_name' ;

Explanation:- Select is Query Statement from where we can select all data or selected data from the table. Here the tables are Project from where we can access project with the end date.

(v) Find all the clients who have not given any project. Also find the person who is not working on any project at all.

(v) Select * from clients where c_id = Project.c_id;

and

Select * from clients where e_id = ProjectLeader.c_id;

Explanation:-

Select is Query Statement from where we can select all data or selected data from the table. It is simply access all data or information from the table "Members".

ok.isbn_no= Book_Accession.isbn_no and date_of_purchase=2014, this is the way of accessing information from one(Member table) to another (issue_return)table.

5. (a) Discuss the advantages and disadvantages of hierarchical database management system in comparison with RDBMS. Discuss types of applications suitable for hierarchical DBMS and RDBMS.

(b) Define the two principal integrity rules for the relational model. Discuss why it is desirable to enforce these rules.

A.5. (a)

The Hierarchical Data Model is a way of organizing a database with multiple one to many relationships. The structure is based on the rule that one parent can have many children but children are allowed only one parent. This structure allows information to be repeated through the parent child relations created by IBM and was implemented mainly in their Information Management System. (IMF), the precursor to the DBMS.

Advantages:

The model allows easy addition and deletion of new information. Data at the top of the Hierarchy is very fast to access. It was very easy to work with the model because it worked well with linear type data storage such as tapes. The model relates very well to natural hierarchies such as assembly plants and employee organization in corporations. It relates well to anything that works through a one to many relationship. For example; there is a president with many managers below them, and those managers have many employees below them, but each employee has only one manager. ga1111

Disadvantages:

This model has many issues that hold it back now that we require more sophisticated relationships. It requires data to be repetitively stored in many different entities. The database can be very slow when searching for information on the lower entities. We no longer use linear data storage mediums such as tapes so that advantage is null. Searching for data requires the DBMS to run through the entire model from top to bottom until the required information is found, making queries very slow. Can only model one to many relationships, many to many relationships are not supported. Clever manipulation of the model are required to make many to many relationships. For example; what if a professor teaches classes, and is also a graduate student?

Advantages of Relational DBMS:-

Relational databases allow the data to be clear-cut and uncluttered. Problems arise when all the data in the previous example is only contained in one table, unlike a relational database. Since each student has multiple grades, their contact information would be entered multiple times along with each grade.

Splitting data into a number of related tables brings many advantages over a flat file database. These include:

1. Data is only stored once. In the previous example, the city data was gathered into one table so now there is only one record per city. The advantages of this are

No multiple record changes needed

More efficient storage

Simple to delete or modify details.

All records in other tables having a link to that entry will show the change.

2. Complex queries can be carried out. A language called SQL has been developed to allow programmers to 'Insert', 'Update', 'Delete', 'Create', 'Drop' table records. These actions are further refined by a 'Where' clause. For example
`SELECT * FROM Customer WHERE ID = 2`

This SQL statement will extract record number 2 from the Customer table. Far more complicated queries can be written that can extract data from many tables at once.

3. Better security. By splitting data into tables, certain tables can be made confidential. When a person logs on with their username and password, the system can then limit access only to those tables whose records they are authorized to view. For example, a receptionist would be able to view employee location and contact details but not their salary. A salesman may see his team's sales performance but not competing teams.

4. Cater for future requirements. By having data held in separate tables, it is simple to add records that are not yet needed but may be in the future. For example, the city table could be expanded to include every city and town in the country, even though no other records are using them all as yet. A flat file database cannot do this.

5. Ease of use: The revision of any information as tables consisting of rows and columns is much easier to understand.

6. Flexibility: Different tables from which information has to be linked and extracted can be easily manipulated by operators such as project and join to give information in the form in which it is desired.

7. Precision: The usage of relational algebra and relational calculus in the manipulation of the relations between the tables ensures that there is no ambiguity, which may otherwise arise in establishing the linkages in a complicated network type database.

Summary - advantages of a relational database over flat file

- Avoids data duplication
- Avoids inconsistent records
- Easier to change data
- Easier to change data format
- Data can be added and removed easily
- Easier to maintain security.

Disadvantage of RDBMS:-

1. Performance: A major constraint and therefore disadvantage in the use of relational database system is machine performance. If the number of tables between which relationships to be established are large and the tables themselves effect the performance in responding to the SQL queries.
2. Physical Storage Consumption: With an interactive system, for example an operation like join would depend upon the physical storage also. It is, therefore common in relational databases to tune the databases and in such a case the physical data layout would be chosen so as to give good performance in the most frequently run operations. It therefore would naturally result in the fact that the lays frequently run operations would tend to become even more shared.
3. Slow extraction of meaning from data: if the data is naturally organized in a hierarchical manner and stored as such, the hierarchical approach may give quick meaning for that data.
4. Data Complexity: Data in an RDBMS resides in multiple tables, which are linked to each other through shared key values. An RDBMS does not force database designers to impose a coherent table structure; inexperienced programmers may design systems that create unnecessary complexity or limit the future development of the database through poorly chosen data types.

Application of Hierarchical and RDBMS

Applications where we use Database Management Systems are:

Telecom: There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain that huge amount of data that keeps updating every millisecond.

Industry: Where it is a manufacturing unit, warehouse or distribution centre, each one needs a database to keep the records of ins and outs. For example distribution centre should keep a track of the product units that supplied into the centre as well as the

products that got delivered out from the distribution centre on each day; this is where DBMS comes into picture.

Banking System: For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems.

Education sector: Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is a hell lot amount of inter-related data that needs to be stored and retrieved in an efficient manner.

Online shopping: You must be aware of the online shopping websites such as Amazon, Flipkart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

A.5.(b)

INTEGRITY RULES

- Integrity rules are needed to inform the DBMS about certain constraints in the real world.
 - Specific integrity rules apply to one specific database.
- Example: part weights must be greater than zero.
- General integrity rules apply to all databases.

Two general rules will be discussed to deal with: primary keys and foreign keys.

PRIMARY KEYS

- Primary key is a unique identifier for a relation.
- There could be several candidate keys, as long as they satisfy two properties:
 1. uniqueness
 2. minimality
- From the set of candidate keys, one is chosen to be the primary key.
- The others become alternate keys.

EXAMPLE: The relation R has several candidate keys.

ID SSN License_Number NAME

If we select ID to be the primary key, then the other candidate keys become alternate keys.

THE ENTITY INTEGRITY RULE

- No component of the primary key of a base relation is allowed to accept nulls.

WHAT ARE NULLS?

Null may mean "property does not apply". For example, the supplier may be a country, in which case the attribute CITY has a null value because such property does not apply.

Null may mean "value is unknown". For example, if the supplier is a person, then a null value for CITY attribute means we do not know the location of this supplier.

- Nulls cannot be in primary keys, but can be in alternate keys.

EXAMPLE: SSN may be null for one and only one person (why?)

FOREIGN KEYS

- A foreign key is an attribute of one relation R2, whose values are required to match those of the primary key of some other relation R1 (R1 and R2 can be identical)

EXAMPLE: SP relation has attribute S#, and S relation has primary key S#. Then S# in SP is considered a foreign key.

SP is called the "referencing relation". S is called the "referenced relation".

We can draw a "referential diagram"

SP ---S#---> S

or simply

SP -----> S

