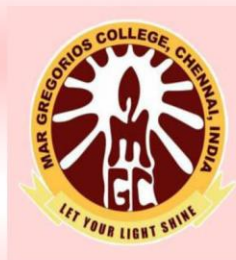# MAR GREGORIOS COLLEGE
OF ARTS & SCIENCE

Block No.8, College Road, Mogappair West, Chennai – 37

Affiliated to the University of Madras
Approved by the Government of Tamil Nadu
An ISO 9001:2015 Certified Institution

DEPARTMENT OF

COMPUTER APPLICATION

SUBJECT NAME: DATABASE MANAGEMENT SYSTEM

SUBJECT CODE: SAE5B

SEMESTER: V

PREPARED BY: PROF.D.STEPHY JOY

Unit 1: Advantages and Components of a Database Management Systems – Feasibility Study – Class Diagrams – Data Types – Events – Normal Forms – Integrity – Converting Class Diagrams to Normalized Tables – Data Dictionary.

Unit-2: Query Basics – Computation Using Queries – Subtotals and GROUP BY Command – Queries with Multiple Tables – Subqueries – Joins – DDL & DML – Testing Queries

Unit 3: Effective Design of Forms and Reports – Form Layout – Creating Forms – Graphical Objects – Reports – Procedural Languages – Data on Forms – Programs to Retrieve and Save Data – Error Handling.

Unit-4: Power of Application Structure – User Interface Features – Transaction – Forms Events – Custom Reports – Distributing Application – Table Operations – Data Storage Methods – Storing Data Columns – Data Clustering and Partitioning.

Unit-5: Database Administration – Development Stages – Application Types – Backup and Recovery – Security and Privacy – Distributed Databases – Client/Server Databases – Web as a Client/Server System – Objects – Object Oriented Databases – Integrated Applications.

UNIT - 1

1.Database

The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. It is also used to organize the data in the form of a table, schema, views, and reports, etc.

For example: The college Database organizes the data about the admin, staff, students and faculty etc.

Using the database, you can easily retrieve, insert, and delete the information.

1.1 Database Management System

- Database management system is a software which is used to manage the database. For example: MySQL, Oracle, etc are a very popular commercial database which is used in different applications.

- DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.

- It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

1.2 Advantages of DBMS

- **Controls database redundancy:** It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.

- **Data sharing:** In DBMS, the authorized users of an organization can share the data among multiple users.

- **Easily Maintenance:** It can be easily maintainable due to the centralized nature of the database system.

- **Reduce time:** It reduces development time and maintenance need.

- **Backup:** It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.

- **multiple user interface:** It provides different types of user interfaces like graphical user interfaces, application program interfaces

1.3 Components of DBMS
DBMS have several components, each performing very significant tasks in the database management system environment. Below is a list of components within the database and its environment.

Software

This is the set of programs used to control and manage the overall database. This includes the DBMS software itself, the Operating System, the network software being used to share the data among users, and the application programs used to access data in the DBMS.

Hardware

Consists of a set of physical electronic devices such as computers, I/O devices, storage devices, etc., this provides the interface between computers and the real world systems.

Data

DBMS exists to collect, store, process and access data, the most important component. The database contains both the actual or operational data and the metadata.

Procedures

These are the instructions and rules that assist on how to use the DBMS, and in designing and running the database, using documented procedures, to guide the users that operate and manage it.

Database Access Language

This is used to access the data to and from the database, to enter new data, update existing data, or retrieve required data from databases. The user writes a set of appropriate commands in a database access language, submits these to the DBMS, which then processes the data and generates and displays a set of results into a user readable form.

Data Dictionary

This is a reserved space within a database used to store information about the database itself. A data dictionary is a set of read-only table and views, containing the different information about the data used in the enterprise to ensure that database representation of the data follow one standard as defined in the dictionary.

1.4 Feasibility Study

Feasibility studies are covered in detail within system analysis texts.

The goal of feasibility study is to determine whether a proposed project is worth pursing.

The study examine two fundamental categories:

- Cost
- Potential Benefit

COSTS	BENEFITS
UP-FRONT/ONE-TIME COSTS • Software • Hardware • Communications • Data conversion • Studies and design • Training **ONGOING COSTS** • Personnel • Software upgrades • Supplies • Support • Software and hardware maintenance	**COST SAVINGS** • Software maintenance • Fewer errors • Less data maintenance • Less user training **INCREASED VALUE** • Better access to data • Better decisions • Better communication • More timely reports • Faster reaction to change • New products and services **STRATEGIC ADVANTAGES** • Lock out competitors

1.5 Class Diagram

- The basic information is used to create a model of data.

- A class diagram displays the key element and associations.

- A class diagram is used to communicate with user and with other designers.

TERM	DEFINITION	PET STORE EXAMPLES
Entity	Something in the real world that you wish to describe or track	Customer, Merchandise, Sale
Class	Description of an entity that includes its attributes (properties) and behavior (methods)	Customer, Merchandise, Sale
Object	One instance of a class with specific data	Joe Jones, Premium Cat Food, Sale #32
Property	A characteristic or descriptor of a class or entity	LastName, Description, SaleDate
Method	A function that is performed by the class	AddCustomer, UpdateInventory, ComputeTotal
Association	A relationship between two or more classes	Each sale can have only one customer

1.6 Association and Relationship

Relationships depict a connection between several things, such as structural, behavioral, or grouping things in the unified modeling language. Since it is termed as a link, it demonstrates how things are interrelated to each other at the time of system execution. It constitutes four types of relationships, i.e., **dependency, association, generalization,** and **realization.**

Dependency

Whenever there is a change in either the structure or the behavior of the class that affects the other class, such a relationship is termed as a dependency. Or, simply, we can say a class contained in other class is known as dependency. It is a unidirectional relationship.

Association

Association is a structural relationship that represents how two entities are linked or connected to each other within a system. It can form several types of associations, such as **one-to-one, one-to-many, many-to-one,** and **many-to-many.** A ternary association is one that constitutes three links. It portrays the static relationship between the entities of two classes.

An association can be categorized into four types of associations, i.e., bi-directional, unidirectional, aggregation (composition aggregation), and reflexive, such that an aggregation is a special form of association and composition is a special form of aggregation. The mostly used associations are unidirectional and bi-directional.
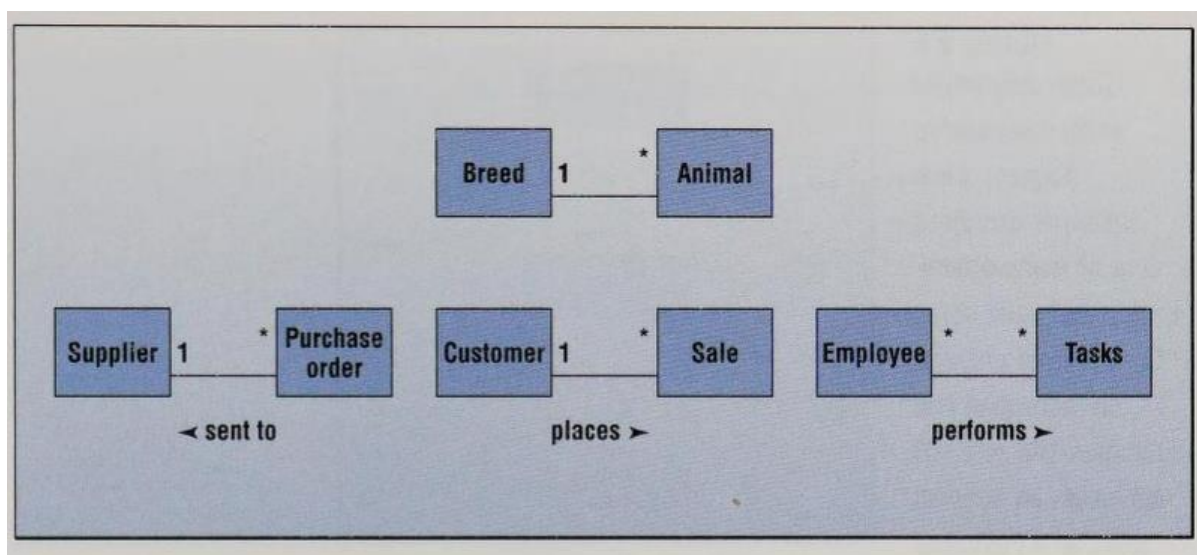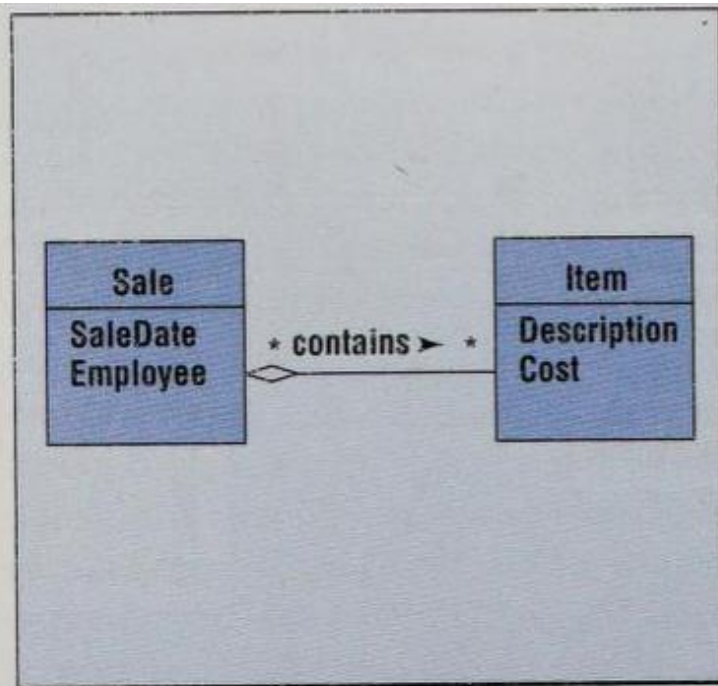

Aggregation

An aggregation is a special form of association. It portrays a part-of relationship. It forms a binary relationship, which means it cannot include more than two classes. It is also known as **Has-a relationship.** It specifies the direction of an object contained in another object. In aggregation, a child can exist independent of the parent.

FIGURE 2.9
Association aggregation. A Sale contains a list of Items being purchased. A small diamond is placed on the association to remind us of this special relationship.

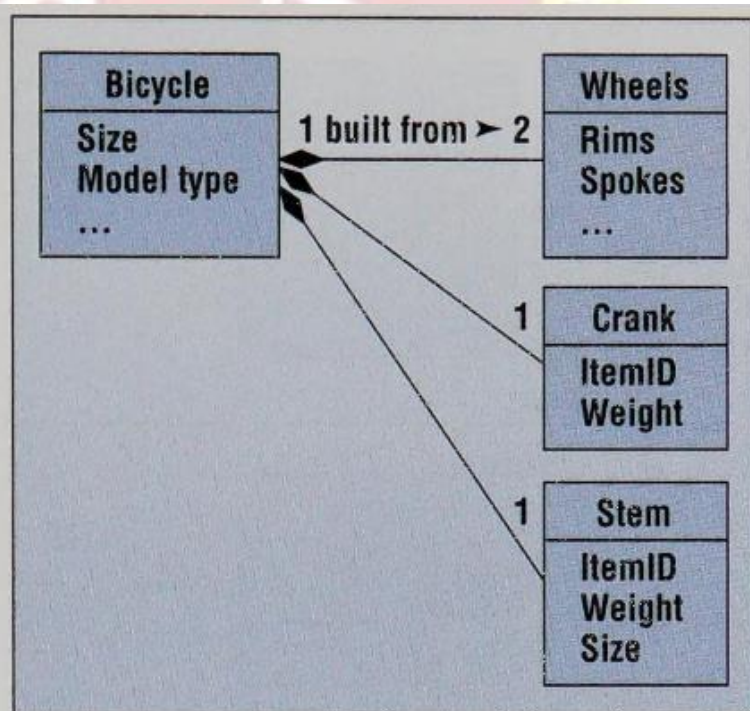| Sale		Item
SaleDate Employee	* contains ▶ *	Description Cost

Composition

In a composition relationship, the child depends on the parent. It forms a two-way relationship. It is a special case of aggregation. It is known as **Part-of** relationship.

FIGURE 2.10
Association composition. A bicycle is built from several individual components. These components no longer exist separately; they become the bicycle.
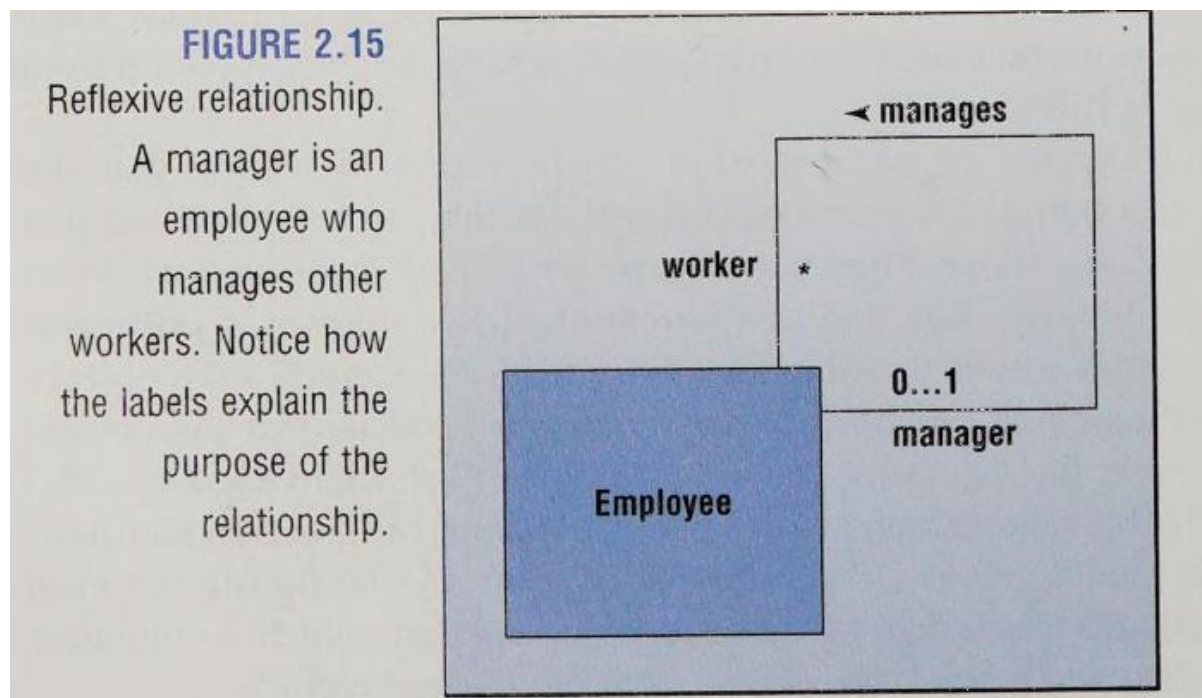
Bicycle		Wheels
Size Model type ...	1 built from ▶ 2	Rims Spokes ...

1	Crank
	ItemID Weight

1	Stem
	ItemID Weight Size

Generalization

The generalization relationship implements the object-oriented concept called inheritance or **is-a** relationship. It exists between two objects (things or entities), such that one entity is a parent (superclass or base class), and the other one is a child (subclass or derived class). These are represented in terms of inheritance. Any child can access, update, or inherit the functionality, structure, and behavior of the parent.
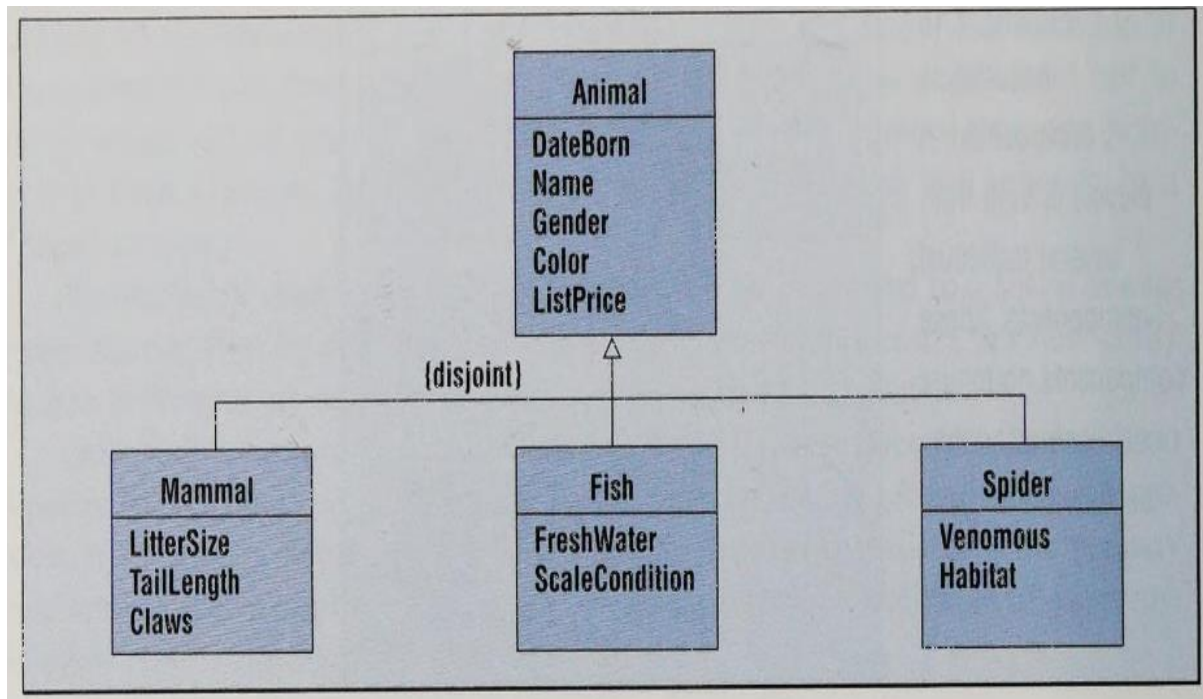
Reflexive

It is a kind of relationship in which one thing specifies the behavior or a responsibility to be carried out, and the other thing carries out that behavior. It can be represented on a class diagram or component diagrams. The realization relationship is constituted between interfaces, classes, packages, and components to link a client element to the supplier element.

FIGURE 2.15
Reflexive relationship. A manager is an employee who manages other workers. Notice how the labels explain the purpose of the relationship.

1.7 Data Type or Data Domain

- Text
- Numbers
- Date and Times
- Binary object
- Computed value

Data Domain – Text

- Text columns are often limited no more than 255 characters.
- Fixed Length – Fixed Length string always take up the amount of space you allocate and are useful to improve speed in handling short strings like identification of two letter state abbreviation.
- Variable Length String - Variable Length string are stored so they take only as much space as needed for each row.
- Eg- Name char(10)

Data Domain – Number

- Numeric data is also common and computers recognize several variation of numeric data.
- The most important decision you have to make about numeric data column is choosing between integer and floating point number.
- Eg Empid Integer(20)

Data Domain – Dates and Times

- All database need a special data type for data and times.

- Most system combines the two into one domain; some provide two sperate definition.

- Eg dob date

Data Domain – Binary Object

- A relatively new domain is a separate category for objects or binary large objects(BLOB).

- It enables to store any types of object created by computer.

- A useful example is to use a BLOB to hold images and files from other software packages.

Data Domain – Computed Values

- It's a derived value

1.8 Events

- Events are actions that are dependent on time.

- Events are another important components of modern database systems that you need to understand. Three basic types of events occurs in a database environment.

    - Business event that trigger some function, such as a sale triggering a reduction in inventory.

    - Data changes that signal some alert, such as an inventory that drops below present level, which triggers a new purchase order.

    - User interface events that trigger some action, such as user clicking on an icon to send a purchase order to a supplier.

1.9 Keys in DBMS

KEYS in DBMS is an attribute or set of attributes which helps you to identify a row(tuple) in a relation(table). They allow you to find the relation between two tables. Keys help you uniquely identify a row in a table by a combination of one or more columns in that table. Key is also helpful for finding unique record or row from the table. Database key is also helpful for finding unique record or row from the table.

Types of Keys in Database Management System

There are mainly seven different types of Keys in DBMS and each key has it's different functionality:

- **Super Key -** A super key is a group of single or multiple keys which identifies rows in a table.
- **Primary Key -** is a column or group of columns in a table that uniquely identify every row in that table.

- **Candidate Key -** is a set of attributes that uniquely identify tuples in a table. Candidate Key is a super key with no repeated attributes.
- **Alternate Key -** is a column or group of columns in a table that uniquely identify every row in that table.
- **Foreign Key -** is a column that creates a relationship between two tables. The purpose of Foreign keys is to maintain data integrity and allow navigation between two different instances of an entity.
- **Compound Key -** has two or more attributes that allow you to uniquely recognize a specific record. It is possible that each column may not be unique by itself within the database.
- **Composite Key -** An artificial key which aims to uniquely identify each record is called a surrogate key. These kind of key are unique because they are created when you don't have any natural primary key.
- **Surrogate Key -** An artificial key which aims to uniquely identify each record is called a surrogate key. These kind of key are unique because they are created when you don't have any natural primary key.

What is the Super key?

A superkey is a group of single or multiple keys which identifies rows in a table. A Super key may have additional attributes that are not needed for unique identification.

Example:

EmpSSN	EmpNum	Empname
9812345098	AB05	Shown
9876512345	AB06	Roslyn
199937890	AB07	James

In the above-given example, EmpSSN and EmpNum name are superkeys.

What is a Primary Key?

PRIMARY KEY is a column or group of columns in a table that uniquely identify every row in that table. The Primary Key can't be a duplicate meaning the same value can't appear more than once in the table. A table cannot have more than one primary key.

Rules for defining Primary key:

- Two rows can't have the same primary key value
- It must for every row to have a primary key value.
- The primary key field cannot be null.

- The value in a primary key column can never be modified or updated if any foreign key refers to that primary key.

Example:

In the following example, <code>StudID</code> is a Primary Key.

StudID	Roll No	First Name	LastName	Email	
1	11	Tom	Price	abc@gmail.com	
2	12	Nick	Wright	xyz@gmail.com	
3	13	Dana	Natan	mno@yahoo.com	

What is the Alternate key?

ALTERNATE KEYS is a column or group of columns in a table that uniquely identify every row in that table. A table can have multiple choices for a primary key but only one can be set as the primary key. All the keys which are not primary key are called an Alternate Key.

Example:

In this table, StudID, Roll No, Email are qualified to become a primary key. But since StudID is the primary key, Roll No, Email becomes the alternative key.

StudID	Roll No	First Name	LastName	Email	
1	11	Tom	Price	abc@gmail.com	
2	12	Nick	Wright	xyz@gmail.com	
3	13	Dana	Natan	mno@yahoo.com	

What is a Candidate 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:

- It must contain unique values
- Candidate key may have multiple attributes
- Must not contain null values
- It should contain minimum fields to ensure uniqueness
- Uniquely identify each record in a table

Example: In the given table Stud ID, Roll No, and email are candidate keys which help us to uniquely identify the student record in the table.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

What is the Foreign key?

FOREIGN KEY is a column that creates a relationship between two tables. The purpose of Foreign keys is to maintain data integrity and allow navigation between two different instances of an entity. It acts as a cross-reference between two tables as it references the primary key of another table.

Example:

DeptCode	DeptName
001	Science
002	English
005	Computer

Teacher ID	Fname	Lname
B002	David	Warner
B017	Sara	Joseph
B009	Mike	Brunton

In this key in dbms example, we have two table, teach and department in a school. However, there is no way to see which search work in which department.

In this table, adding the foreign key in Deptcode to the Teacher name, we can create a relationship between the two tables.

Teacher ID	DeptCode	Fname	Lname
B002	002	David	Warner
B017	002	Sara	Joseph
B009	001	Mike	Brunton

This concept is also known as Referential Integrity.

What is the Compound key?

COMPOUND KEY has two or more attributes that allow you to uniquely recognize a specific record. It is possible that each column may not be unique by itself within the database. However, when combined with the other column or columns the combination of composite keys become unique. The purpose of the compound key in database is to uniquely identify each record in the table.

Example:

OrderNo	PorductID	Product Name	Quantity
B005	JAP102459	Mouse	5
B005	DKT321573	USB	10
B005	OMG446789	LCD Monitor	20
B004	DKT321573	USB	15
B002	OMG446789	Laser Printer	3

In this example, OrderNo and ProductIDcan't be a primary key as it does not uniquely identify a record. However, a compound key of Order ID and Product ID could be used as it uniquely identified each record.

What is the Composite key?

COMPOSITE KEY is a combination of two or more columns that uniquely identify rows in a table. The combination of columns guarantees uniqueness, though individually uniqueness is not guaranteed. Hence, they are combined to uniquely identify records in a table.

The difference between compound and the composite key is that any part of the compound key can be a foreign key, but the composite key may or maybe not a part of the foreign key.

What is a Surrogate key?

SURROGATE KEYS is An artificial key which aims to uniquely identify each record is called a surrogate key. This kind of partial key in dbms is unique because it is created when you don't have any natural primary key. They do not lend any meaning to the data in the table. Surrogate key is usually an integer. A surrogate key is a value generated right before the record is inserted into a table.

Fname	Lastname	Start Time	End Time
Anne	Smith	09:00	18:00

Jack	Francis	08:00	17:00
Anna	McLean	11:00	20:00
Shown	Willam	14:00	23:00

Above, given example, shown shift timings of the different employee. In this example, a surrogate key is needed to uniquely identify each employee.

Surrogate keys in sql are allowed when

- No property has the parameter of the primary key.
- In the table when the primary key is too big or complicated.

1.10 Normalization of Database

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

Normalization is used for mainly two purposes,

- Eliminating redundant(useless) data.

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

Problems Without Normalization

If a table is not properly normalized and have data redundancy then it will not only eat up extra memory space but will also make it 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 a **Student** table.

rollno	Name	Branch	hod	office_tel
401	Akon	CSE	Mr. X	53337
402	Bkon	CSE	Mr. X	53337

| 403 | Ckon | CSE | Mr. X | 53337 |
| 404 | Dkon | CSE | Mr. X | 53337 |

In the table above, we have data of 4 Computer Sci. students. As we can see, data for the fields branch, hod(Head of Department) and office_tel is repeated for the students who are in the same branch in the college, this is **Data Redundancy**.

Insertion Anomaly

Suppose for a new admission, until and unless a student opts for a branch, data of the student cannot be inserted, or else we will have to set the branch information as **NULL**.

Also, if we have to insert data of 100 students of same branch, then the branch information will be repeated for all those 100 students.

These scenarios are nothing but **Insertion anomalies**.

Updation Anomaly

What if Mr. X leaves the college? or is no longer the HOD of computer science department? In that case all the student records will have to be updated, and if by mistake we miss any record, it will lead to data inconsistency. This is Updation anomaly.

Deletion Anomaly

In our **Student** table, two different informations are kept together, Student information and Branch information. Hence, at the end of the academic year, if student records are deleted, we will also lose the branch information. This is Deletion anomaly.

Normalization Rule

Normalization rules are divided into the following normal forms:

1. First Normal Form

2. Second Normal Form

3. Third Normal Form

4. BCNF

5. Fourth Normal Form

First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single(atomic) valued attributes/columns.

2. Values stored in a column should be of the same domain

3. All the columns in a table should have unique names.

4. And the order in which data is stored, does not matter.

In the next tutorial, we will discuss about the **First Normal Form** in details.

Second Normal Form (2NF)

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.

2. And, it should not have Partial Dependency.

To understand what is Partial Dependency and how to normalize a table to 2nd normal for, jump to the **Second Normal Form** tutorial.

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.

2. And, it doesn't have Transitive Dependency.

Here is the **Third Normal Form** tutorial. But we suggest you to first study about the second normal form and then head over to the third normal form.

Boyce and Codd Normal Form (BCNF)

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

- R must be in 3rd Normal Form

- and, for each functional dependency ($X \rightarrow Y$), X should be a super Key.

To learn about BCNF in detail with a very easy to understand example, head to **Boye-Codd Normal Form** tutorial.

Fourth Normal Form (4NF)

A table is said to be in the Fourth Normal Form when,

1. It is in the Boyce-Codd Normal Form.

2. And, it doesn't have Multi-Valued Dependency.

1.11 Data Dictionary

Data Dictionary consists of database metadata. It has records about objects in the database.

What Data Dictionary consists of

Data Dictionary consists of the following information −

- Name of the tables in the database
- Constraints of a table i.e. keys, relationships, etc.
- Columns of the tables that related to each other
- Owner of the table
- Last accessed information of the object
- Last updated information of the object

An example of Data Dictionary can be personal details of a student −

Example

<StudentPersonalDetails>

Student_ID   **Student_Name**   **Student_Address**   **Student_City**

The following is the data dictionary for the above fields −

Field Name	Datatype	Field Length	Constraint	Description
Student_ID	Number	5	Primary Key	Student id
Student_Name	Varchar	20	Not Null	Name of the student
Student_Address	Varchar	30	Not Null	Address of the student
Student_City	Varchar	20	Not Null	City of the student

Types of Data Dictionary

Here are the two types of data dictionary −

Active Data Dictionary

The DBMS software manages the active data dictionary automatically. The modification is an automatic task and most RDBMS has active data dictionary. It is also known as integrated data dictionary.

Passive Data Dictionary

Managed by the users and is modified manually when the database structure change. Also known as non-integrated data dictionary.

UNIT - 2

2.Introduction to SQL

- SQL stands for Structured Query Language

- SQL lets you access and manipulate databases

- Some of the popular relational database example are: MySQL, Oracle, mariaDB, postgreSQL etc.

2.1 Some of The Most Important SQL Commands

- **SELECT** - extracts data from a database

- **UPDATE** - updates data in a database

- **DELETE** - deletes data from a database

- **INSERT INTO** - inserts new data into a database

- **CREATE DATABASE** - creates a new database

- **ALTER DATABASE** - modifies a database

- **CREATE TABLE** - creates a new table

- **ALTER TABLE** - modifies a table

- **DROP TABLE** - deletes a table

- **CREATE INDEX** - creates an index (search key)

- **DROP INDEX** - deletes an index

2.1.1 THE SQL SELECT STATEMENT

- The SELECT statement is used to select data from a database.
- The data returned is stored in a result table, called the result-set.

Select Statement Example – Retrieve single column

Lets say we have a table **EMPLOYEES** with the following data:

```
+------+----------+---------+----------+
|SSN  |EMP_NAME |EMP_AGE |EMP_SALARY|
+------+----------+---------+----------+
| 101 |Steve    | 23     | 9000.00 |
| 223 |Peter    | 24     | 2550.00 |
| 388 |Shubham  | 19     | 2444.00 |
| 499 |Chaitanya| 29     | 6588.00 |
| 589 |Apoorv   | 21     | 1400.00 |
| 689 |Rajat    | 24     | 8900.00 |
| 700 |Ajeet    | 20     |18300.00 |
```

Lets say we want to fetch a single column EMP_NAME from the given table – we can write a SQL statement select like this:

SELECT EMP_NAME FROM EMPLOYEE;

Output of the above Query:

```
+----------+
| EMP_NAME |
+------+---+
| Steve    |
| Peter    |
| Shubham  |
| Chaitanya|
| Apoorv   |
| Rajat    |
| Ajeet    |
+----------+
```

Select multiple columns

SELECT SSN, EMP_NAME FROM EMPLOYEES;
Select Every columns

SELECT * FROM EMPLOYEES;

2.1.2 SELECT statement Clause

In the above examples, we have seen how to fetch single column, multiple columns and entire table using SELECT statement. **What if we want to fetch only few rows or want the rows in an particular order or grouped rows etc.?** We can do so with the help of clauses in SQL select statement.

These clauses are optional but they are frequently used in SQL statements in the real world database operations.

WHERE Clause in SQL: WHERE clause filters the records, with the help of this clause we specify a condition in the SQL statement and only those records that fulfil the condition are retrieved from database.

ORDER BY Clause in SQL: ORDER BY clause is used to return the rows in ascending or descending order of the data.

GROUP BY Clause in SQL: GROUP BY clause groups the rows with same data, this is often used along with the aggregate functions.

HAVING Clause in SQL: HAVING clause filters the records, just like WHERE clause, however it is used along with the GROUP BY clause so it filters the records from the output set of rows produced by GROUP BY clause.

2.1.3 The SQL WHERE Clause

Where clause is used to fetch a particular row or set of rows from a table. This clause filters records based on given conditions and only those row(s) comes out as result that satisfies the condition defined in WHERE clause of the SQL query.

WHERE Syntax

SELECT *column1*, *column2, ...*
FROM *table_name*
WHERE *condition*;

Eg:
SELECT EMP_NAME
FROM EMPLOYEES
WHERE EMP_AGE >23;

Operators in The WHERE Clause

The following operators can be used in the WHERE clause:

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal. **Note:** In some versions of SQL this operator may be written as !=
BETWEEN	Between a certain range

LIKE	Search for a pattern
IN	To specify multiple possible values for a column

2.1.4 SQL ORDER BY Clause

ORDER BY clause is used to sort the returned records in an order. By using ORDER BY clause, we can sort the result in ascending or descending order. This clause can be used with multiple columns as well.

ORDER BY Syntax

SELECT *column1*, *column2, ...*
FROM *table_name*
ORDER BY *column1, column2, ...* ASC|DESC;

Eg: SELECT *
  FROM EMPLOYEE
ORDER BY CITY ASC, PINCODE DESC;

2.1.5 The SQL GROUP BY Statement

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

GROUP BY Syntax

SELECT *column_name(s)*
FROM *table_name*
WHERE *condition*
GROUP BY *column_name(s)*
ORDER BY *column_name(s);*

Example

SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;

2.1.6 The SQL HAVING Clause

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

HAVING Syntax

SELECT *column_name(s)*
FROM *table_name*
WHERE *condition*
GROUP BY *column_name(s)*
HAVING *condition*
ORDER BY *column_name(s);*

Example

SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5;

2.1.7 The SQL SELECT DISTINCT Statement

The SELECT DISTINCT statement is used to return only distinct (different) values.

Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

SELECT DISTINCT Syntax

SELECT DISTINCT *column1*, *column2, ...*
FROM *table_name*;

Eg:

SELECT DISTINCT Emp_City FROM employee;

2.1.8 The SQL AND, OR and NOT Operators

The WHERE clause can be combined with AND, OR, and NOT operators.

The AND and OR operators are used to filter records based on more than one condition:

- The AND operator displays a record if all the conditions separated by AND are TRUE.
- The OR operator displays a record if any of the conditions separated by OR is TRUE.

The NOT operator displays a record if the condition(s) is NOT TRUE.

AND Syntax

SELECT *column1*, *column2, ...*
FROM *table_name*
WHERE *condition1* AND *condition2* AND *condition3 ...*;

OR Syntax

SELECT *column1*, *column2, ...*
FROM *table_name*
WHERE *condition1* OR *condition2* OR *condition3 ...*;

NOT Syntax

SELECT *column1*, *column2, ...*
FROM *table_name*
WHERE NOT *condition*;

2.2 Aggregate functions in SQL

In database management an aggregate function is a function where the values of multiple rows are grouped together as input on certain criteria to form a single value of more significant meaning.

Aggregate Functions
1) **Count()** - The COUNT() function returns the number of rows that matches a specified criterion.

2) **Sum()** - The SUM() function returns the total sum of a numeric column.

3) **Avg()** - The AVG() function returns the average value of a numeric column.

4) **Min()** - The MIN() function returns the smallest value of the selected column.

5) **Max()** - The MAX() function returns the largest value of the selected column.

MIN() Syntax

SELECT MIN(*column_name*)
FROM *table_name*
WHERE *condition*;

Example

SELECT MIN(Price) AS SmallestPrice
FROM Products;

MAX() Syntax

SELECT MAX(*column_name*)
FROM *table_name*
WHERE *condition*;

Example

SELECT MAX(Price) AS LargestPrice
FROM Products;

COUNT() Syntax

SELECT COUNT(*column_name*)
FROM *table_name*
WHERE *condition*;

Example

SELECT COUNT(ProductID)
FROM Products;

AVG() Syntax

SELECT AVG(*column_name*)
FROM *table_name*
WHERE *condition*;

Example

SELECT AVG(Price)
FROM Products;

SUM() Syntax

SELECT SUM(*column_name*)
FROM *table_name*
WHERE *condition*;

Example

SELECT SUM(Quantity)
FROM OrderDetails;

2.3 SQL JOIN

The SQL **Joins** clause is used to combine records from two or more tables in a database. A JOIN is a means for combining fields from two tables by using values common to each.

Consider the following two tables −

Table 1 − CUSTOMERS Table

```
+----+----------+-----+-----------+----------+
| ID | NAME    | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|1|Ramesh|32|Ahmedabad|2000.00|
|2|Khilan|25|Delhi|1500.00|
|3|kaushik|23|Kota|2000.00|
|4|Chaitali|25|Mumbai|6500.00|
|5|Hardik|27|Bhopal|8500.00|
|6|Komal|22| MP      |4500.00|
|7|Muffy|24|Indore|10000.00|
```

```
+----+----------+-----+----------+----------+
```

Table 2 − ORDERS Table

```
+-----+--------------------+------------+--------+
|OID  | DATE               | CUSTOMER_ID | AMOUNT |
+-----+--------------------+------------+--------+
|102|2009-10-0800:00:00|3|3000|
|100|2009-10-0800:00:00|3|1500|
|101|2009-11-2000:00:00|2|1560|
|103|2008-05-2000:00:00|4|2060|
+-----+--------------------+------------+--------+
```

Now, let us join these two tables in our SELECT statement as shown below.

```
SQL> SELECT ID, NAME, AGE, AMOUNT
   FROM CUSTOMERS, ORDERS
WHERE  CUSTOMERS.ID= ORDERS.CUSTOMER_ID;
```

This would produce the following result.

```
+----+----------+-----+--------+
| ID | NAME     | AGE | AMOUNT |
+----+----------+-----+--------+
| 3 | kaushik  | 23 |  3000 |
| 3 | kaushik  | 23 |  1500 |
| 2 | Khilan   | 25 |  1560 |
| 4 | Chaitali | 25 |  2060 |
+----+----------+-----+--------+
```

Here, it is noticeable that the join is performed in the WHERE clause. Several operators can be used to join tables, such as =, <, >, <>, <=, >=, !=, BETWEEN, LIKE, and NOT; they can all be used to join tables. However, the most common operator is the equal to symbol.

There are different types of joins available in SQL −

- INNER JOIN − returns rows when there is a match in both tables.

- LEFT JOIN − returns all rows from the left table, even if there are no matches in the right table.

- RIGHT JOIN − returns all rows from the right table, even if there are no matches in the left table.

- FULL JOIN − returns rows when there is a match in one of the tables.

- SELF JOIN − is used to join a table to itself as if the table were two tables, temporarily renaming at least one table in the SQL statement.

- CARTESIAN JOIN − returns the Cartesian product of the sets of records from the two or more joined tables.
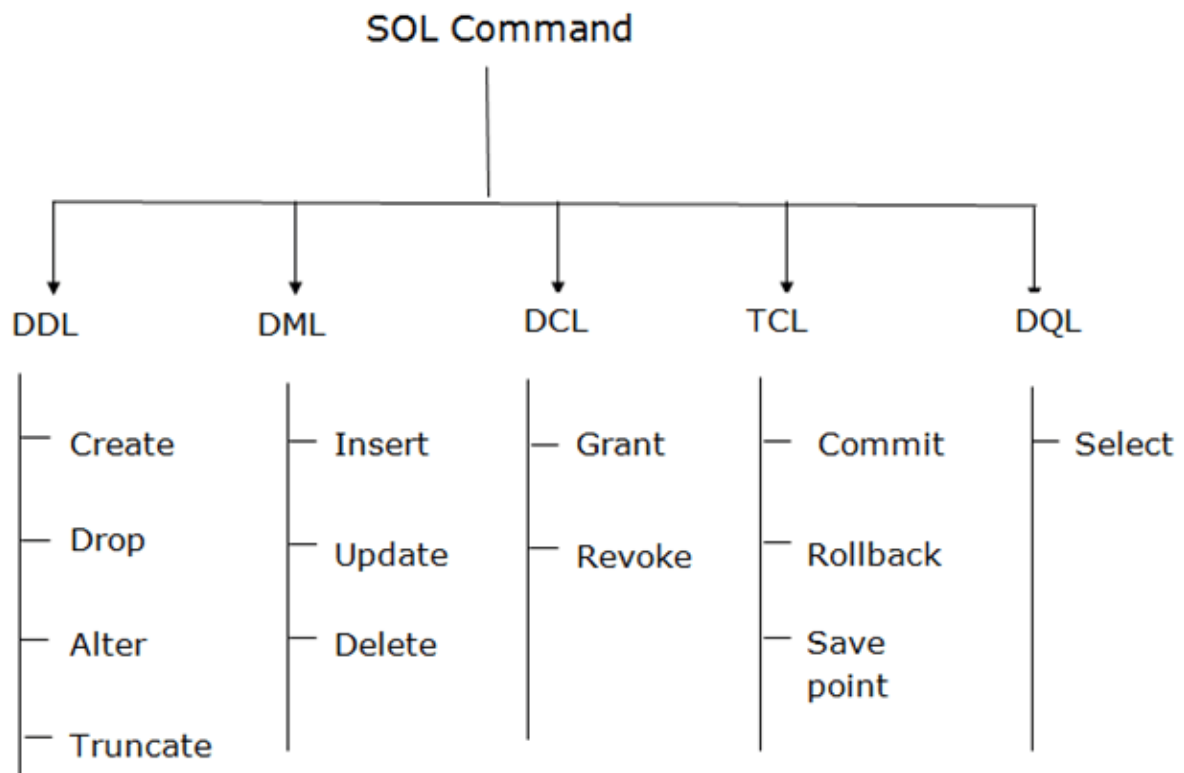
2.4 SQL Commands

- o SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.

- o SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

2.4.1 Types of SQL Commands

There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.

```
                          SOL Command
                              |
   ┌──────────┬──────────────┼──────────────┬──────────────┐
   ↓          ↓              ↓              ↓              ↓
  DDL        DML            DCL            TCL            DQL
   |          |              |              |              |
  ─ Create   ─ Insert       ─ Grant        ─ Commit       ─ Select
   |          |              |              |
  ─ Drop     ─ Update       ─ Revoke       ─ Rollback
   |          |                             |
  ─ Alter    ─ Delete                       ─ Save
   |                                          point
  ─ Truncate
```

Data Definition Language (DDL)
- o DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- o All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- o CREATE
- o ALTER
- o DROP
- o TRUNCATE

a. CREATE It is used to create a new table in the database.
Syntax:

CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES[,....]);
Example:
CREATE TABLE EMPLOYEE(Name VARCHAR2(20), Email VARCHAR2(100), DOB D
ATE);

b. DROP: It is used to delete both the structure and record stored in the table.
Syntax
DROP TABLE ;
Example
DROP TABLE EMPLOYEE;

c. ALTER: It is used to alter the structure of the database. This change could be either to
modify the characteristics of an existing attribute or probably to add a new attribute.
Syntax:
To add a new column in the table
ALTER TABLE table_name ADD column_name COLUMN-definition;
To modify existing column in the table:
ALTER TABLE MODIFY(COLUMN DEFINITION....);
Example
ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));
ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));

d. TRUNCATE: It is used to delete all the rows from the table and free the space containing
the table.
Syntax:
TRUNCATE TABLE table_name;
Example:
TRUNCATE TABLE EMPLOYEE;

Data Manipulation Language
   o DML commands are used to modify the database. It is responsible for all form of
     changes in the database.
   o The command of DML is not auto-committed that means it can't permanently save all
     the changes in the database. They can be rollback.

Here are some commands that come under DML:

   o INSERT
   o UPDATE
   o DELETE

a. INSERT: The INSERT statement is a SQL query. It is used to insert data into the row of a
table.

Syntax:

INSERT INTO TABLE_NAME

(col1, col2, col3,.... col N)

VALUES (value1, value2, value3, ... valueN);

Or

INSERT INTO TABLE_NAME

VALUES (value1, value2, value3, ... valueN);

For example:

INSERT INTO javatpoint (Author, Subject) VALUES ("Sonoo", "DBMS");

b. UPDATE: This command is used to update or modify the value of a column in the table.

Syntax:

UPDATE table_name SET [column_name1= value1,...column_nameN = valueN] [WHERE CONDITION]

For example:

UPDATE students

SET User_Name = 'Sonoo'

WHERE Student_Id = '3'

c. DELETE: It is used to remove one or more row from a table.

Syntax:

DELETE FROM table_name [WHERE condition];

For example:

DELETE FROM javatpoint

WHERE Author="Sonoo";

Data Control Language

DCL commands are used to grant and take back authority from any database user.

Here are some commands that come under DCL:

- o Grant
- o Revoke

a. Grant: It is used to give user access privileges to a database.

Example

GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;

b. Revoke: It is used to take back permissions from the user.

Example

REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;

Transaction Control Language

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

- o COMMIT
- o ROLLBACK
- o SAVEPOINT

a. Commit: Commit command is used to save all the transactions to the database.

Syntax:

COMMIT;

Example:

DELETE FROM CUSTOMERS

WHERE AGE = 25;

COMMIT;

b. Rollback: Rollback command is used to undo transactions that have not already been saved to the database.

Syntax:

ROLLBACK;

Example:

DELETE FROM CUSTOMERS

WHERE AGE = 25;

ROLLBACK;

c. SAVEPOINT: It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax:

SAVEPOINT SAVEPOINT_NAME;

5. Data Query Language

DQL is used to fetch the data from the database.

It uses only one command:

- o SELECT

a. SELECT: This is the same as the projection operation of relational algebra. It is used to select the attribute based on the condition described by WHERE clause.

Syntax:

SELECT expressions

FROM TABLES

WHERE conditions;

For example:

SELECT emp_name

FROM employee

WHERE age > 20;

UNIT - 3

3.Introduction to Forms andReport

Forms and reports are an important part of the database application. Designer use them to create an integrated application, making it easier for user to perform their task. Decision maker and clerical workers use forms and reports on a daily basis. Normally forms were used as input and report were used to display result. Nowadays, forms are also used to display result.

Basic uses of forms are,

- Collect data
- Display query data
- Display analysis and computation result
- Switch board
- Direct manipulation of object like Graphics

Reports are typically printed on paper, but they are increasingly begin created for direct display on the screen. Report can be used to format the data and present results from complex analysis. Forms and reports have several common features.

3.2.Effective Design of forms and Reports

When designing forms and reports, one must remember that they are the primary contact with the users. The key to effective design is to determine the needs of the user. Research into human factors design has also led to several hints and guidelines that designers should follow when building forms and reports.

There are some basic human factors design elements that system designers should incorporate in their applications. They are,

User Control

With current operating systems, the primary factor is that the users-not the programmer and not the application-should always have control. For example do not expect users to enter data in a particular sequence.

Application responds to the events or triggers by performing calculations, retrieving or storing data, and offering new choices.

Whenever possible, provide options for user customization.

Example: Changing the display features such as color, typeface, or size.

Consistency

To ensure consistency, there are two factors

- At the beginning of the application establish a design standard and basic template for all designers to use.
- Toward the end of the application development always go back and check for consistency.

Clarity

Clarity means keeping the application simple and well organized. If the application has multiple forms and reports, organize them according to user tasks. It helps to have a clear purpose for an application and to make sure the design enhances that purpose. Use precise terminology, avoid jargon, and stick with terms that are used within the organization.

Example: If a company refers to its employees as   Assocites , use that term, instead of Employees .

Aesthetics

Aesthetics also play an important role in the user interface. The goal is to use color and design to enhance the forms and reports. Although design and art are highly subjective, bad designs are immediately obvious to others. Graphics and art are important, and they provide an attractive and familiar environment for users.

Feedback

Uses of feedback include accepting input, acknowledging changes of data, highlighting completion of a task, or signifying the start or completion of some event.

Several options can be used to provide feedback. Visually, the cursor can be changed, text can be highlighted, a button can be pushed in, or a box may change color.

More direct forms of feedback, such as displaying messages on the screen, can be used in more complicated cases. Some systems use audio feedback, playing a musical theme or sound when the user selects a task or when the computer finishes an operation.

Forgiveness

Designers should provide the users with the possibility of,

- Anticipation and correction of errors and confirmation on deletions and updates
- Backup and recovery of data

3.3 Form Layout or types of forms

Forms are used to collect data, display results and organize the overall system. Examples for forms are Business forms, Electronic spread sheet, ATM transaction layout, etc.

Four basic types of forms are,

- Tabular forms
- Single-row forms
- Subform forms
- Switchboard forms

3.3.1 Tabular Form

- Tabular forms, which displays data in rows and columns.
- It can be used as a subform and is rarely used as a stand-alone form.
- In subforms, they collect and display a limited list of data that is related to the main form.
- The main feature of tabular form is that it displays multiple rows of data for editing.

Difference between the tabular and datasheet form is that the tabular form provides a little more control over the display-such as three-dimensional controls and individual color settings. As a subform, the datasheet view takes up less space than the tabular form and looks like a spreadsheet.

Advantage

Tabular forms display multiple rows of data for editing.

Disadvantage

Tabular form will not work very well to find a specific entry, when the table is too large in terms of number of rows and columns.

3.3.2 Single Row or Columnar Form

- A single row form displays data for one row at a time.
- The goal is to display every column.
- The designer can also use color, graphics, and command buttons to make the form easier to use.
- The single-row form is generally the most-used form layout.
- We can also include charts to help users make decisions.

- The greatest feature is that the designer can display the data at any location on the form.
- It is useful for designing a form that looks like a traditional paper form.
- The navigation buttons on the bottom of the form enable the user to display different rows.

3.3.3 Sub Forms

- Subform, which display data from two tables that have a one-to-many relationship.
- A subform is usually a tabular form embedded on the main form.
- The main form must be a single-row form, and the subform should be a tabular view.
- Most database system enables to create forms that have multiple subforms.
- The subforms can be either independent- as separate boxes on the main form- or nested- where each subform lies inside another.
- Subform contents are linked to the main form through a common column.

3.3.4 Switchboard Form

- Switchboard forms or menus which direct the user to other forms and reports in the application.
- Switchboard or menu forms provide the overall structure to an application.
- The switchboard form often contains images, and the design reflects the style of the company.
- Command buttons or links are the most important features of the switchboard form.
- When the user selects a button, a corresponding form or report is opened.
- The key to a successful application begins with the switchboard form not just its design but also its content.
- The forms should match the user's tasks.

3.3.5 Advantages of Forms

- A form provides an easy way to view data.
- Using forms, data can be entered easily. This saves time and prevents typographical errors.
- Forms present data in an attractive format with special fonts and other graphical effects such as color and shading.
- Forms offer the most convenient layout for entering, changing and viewing records present in the database.
- An entry field in a form can present a list of valid values from which users can pick to fill out the field easily.

3.4 Creating Forms

The first step in creating a form is that the user should understand its purpose and how it will be used. Its usage dictates the specific data that needs to be displayed then, identify the database tables that hold that data. A form should only attempt to update data to one table at a time.

Suppose if the data needs to be stored in four related tables, one should use the multiple linked forms or by creating sections within a form, where each section can hold data for a new linked table.

One of these sections could be a subform that contains repeating rows of data linked to the main form. Updateable queries are used to display data from multiple tables.

3.5 Computer tools

Dealing with totals and subtotals is a challenging problem in database application design. Consider the basic sales from example, where a subform displays purchases of individual items. The main advantage to storing totals is that it is faster to retrieve them later, and the queries are easier to write. If you do not store the totals, then your queries have to computer the value every time you need them. These calculations are slower than retrieving the totals because the process entails retrieving all of the underlying details rows, instead of just one total line.

The main drawback to storing calculated values is that the totals might become inconsistent.

To be safe, for all stored totals, you need to include a separate program that goes back to the underlying transaction details and computers the correct totals. Many of these calculations required the use of procedural languages, or chains of SQL statements.

3.6 Procedural language

Procedural language is the generic term for a traditional programming language such as BASIC, COBOL, C++, or Java that evaluates commands in sequential order and contains certain basic logic elements. To write programs, you need to learn the basic operations listed in below:-

- Variables
- Computations
- Standard functions
- Debug
- Output
- Input
- Conditions
- Loops
- Arrays

Programming ultimately reduces to two components: logic and syntax. Programming logic represents the structure of the program and the use of the programming elements to solve the underlying problem.

Syntax represents the specific commands and features in a programming language.

3.7 Programming environment

Even beyond basic structure and syntax, DBMS vendors have several choices about how to create a programming language. Perhaps the biggest choice is exactly where the language will be used. Three common methods is used:

1. within the database forms and reports.

2. within the query system as an extension to SQL

3. hosted within separate programs outside the DBMS(Embedded SQL)

The location or environment, of the language affects what it can do, how it is used and what you will have to learn. For example, to embed SQL within an external program, if you already known the host language (e.g., C++ or COBOL), you only must learn how to transfer data between SQL and the host language. On the other hand, languages inside the DBMS have their own logic and syntax, so you must learn an entirely new system.

3.7.1 Windows Environment

The windows interface has significantly changed way programs operate. In the "old days", each program had its own style. The programmer had to design every aspect of the user interface. Also, the program was in control form the start of the session to the end. The user and the operating system (Windows) have control of the computer. As a developer, you write procedures that are called by the user or the operating system to perform a specific task. When the task is accomplished, control is returned to the operating system. As the user performs some task, the event triggers your code to run. A key aspect of developing system is to identify which of the many events should trigger your code.

In many systems (e.g., Oracle and SQL Server), triggers can also be define directly on the data tables.

3.8 Data on Forms

When you need a user to enter data, you will create a form. Forms are also used to display data and results of computations. Each control on the form represents a type of data associated with that form.

You can also create unbound controls that will hold and display data on a form temporarily. When the form closed, the unbound data disappears. You can use unbound controls when you want users to enter raw data. Then your procedure performs computations on the data and stores it in the database.

3.8.1 Using programs to save and retrieve data in the data base

Once you can understand the basic operation of cursors, it is straightforward to retrieve or change the data. Just remember that you change data on row at a time. In Microsoft Access you essentially treat the columns as variables. For example, to obtain the value in the BalanceDue column of the current row, you would use rst("BalanceDue").

The advantages of the programming arise when the calculations are more complex. In particular, the code within the loop might have several conditions (If/Then/Else) and use data that was entered on the form.

3.8.2 Handling error

There are many sources of errors like mistake in code, faulty data entered by users, hardware or network failures, operating system glitches, and so on. You need to create your programs so that they can deal with errors. If you do not, the program might crash and the user will be left with a blank screen and lost data.

The two primary philosophies exit for handling errors:-

a. Check whether every major operation is performed correctly.
b. Write a special error-handling routine that is called only when any error arise.

Eg:-

On Erroe Go to ErrSub1
Program code
:
:
ErrSub:

Program to handle error

UNIT 4

4.1 Power of Applications

An application is a program or group of programs designed for end users. Applications are complete systems that perform specific tasks. The task is defined by the user.

Goal

The goal of an application is to provide information and help users to make decisions.

Applications serve two primary functions. They are,

They establish and control the user interface. They ensure data integrity.

4.1.1. User Interface

A user interface is the system by which people (users) interact with a machine.

The user interface includes hardware (physical) and software (logical) components. User interfaces exist for various systems, and provide a means of:

Input, allowing the users to manipulate a system

Output, allowing the system to indicate the effects of the users' manipulation

A good user interface provides a "user-friendly" experience, allowing the user to interact with the software or hardware in a natural and intuitive way.

The basic concept of improving the user interface is that the application should reflect the way the user works.

A primary purpose of an application is to integrate the input forms and reports. Users should not have to go searching for input forms and reports. They should be available at the click of a mouse.

Example: If users customarily print order forms when they are entered, the order- entry form should have a button to print the corresponding report.

4.1.2. Ensure Data Integrity

Data integrity means reliability and accuracy of data. Data integrity is a fundamental component of information security. A key purpose of applications is to ensure data integrity.

Example: To maintain data integrity numeric columns/cells should not accept alphabetic data.

The standard features to ensure data integrity are,

Data validation

Data validation is the process of ensuring that a program operates on clean, correct and useful data.

Example

Price field should always be greater than 0.

Automatic computations

Application should also be used to automatically perform all computations.

Example

Computation of sales tax automatically. These business rules are generally written as code within the forms.

Verification of totals

After computation and storing totals in a database, we need to write additional code that verifies and corrects the totals.

Example

Quantity on Hand (QOH) is updated when an item is sold or purchased. A separate program is required to validate the current totals by examining all sales and purchases.

Control of user access

An application should have the power of control by accessing from the user.

Transaction integrity

Transaction processing is used to maintain database integrity by ensuring that set of SQL operations execute completely or not at all.

The degree to which a transaction flowing through a network reaches its intended destination without impairment of its function, content or meaning.

Backup and recovery

Backup and recovery refer to the strategies and procedures involved in protecting the database against data loss and reconstructing the data in case of such data loss.

4.2.Application Structure

The layout or structure of an application defines how the user will deal with the application.

The first step in designing the structure or architecture of the application is to design each form.

The objective of application structure is to organize all of the forms and reports to produce a complete application.

4.2.1. Designing Application

The first step in designing the overall application structure is to identify the various users and outline the tasks that will be performed with the applications. A central switchboard form can be used to identify the user and direct him or her to the appropriate section. This segmentation reduces complexity.

4.3 User interface features

Modern windows based applications have several user interface feature that are designed to standardize the look and feel of applications and to make your applications relatively.

Three features are particularly important: Menus, toolbars, help system.

Menus:-

The menu is simply a list of choices that perform some action when selected by the user. Most menus are hierarchical. The windows interface standard specifies that menus should be displayed at the top of the application.

Toolbar:-

A toolbar generally contains a collection of buttons and menu items. When the user click a toolbar button, a predefined operation. A toolbar can contain traditional buttons, and it can contain texture menus. Most toolbars are dockable, when means that users ca drag them to any place on the application windows.

Helps system:-

The goal of this system is to provide the background information and the specific instructions that a user might need to effectively use the application system. Help files can contains text-based descriptions, figures and hypertext links to related topics. The user should be presented with information that is designed to help with the specific task they are working on at the time.

4.4 Transactions

A transaction is defined as a set of changes that must all be made together. Consider the example; you are working on a system for a bank. A customer goes to the ATM and instructs it to transfer $1,000 from savings to a checking account. This simple transaction required two steps: (1) subtracting the money from the savings account balance and (2) adding the money to the checking account balance.

The code to create this transaction will required two updates to the database. For example, there will be two SQL statements: one UPDATE command to decrease the balance in savings and a second UPDATE command to increase the balance in the checking account. If both IPDATE performed successful then transaction is 'COMMIT' otherwise 'ROLLBACK'.

4.5 Forms to help user

You can do several things to make your forms easier to use. The key is to understand the business operations and how the forms will be used. For example, you can control combo boxes with the WHERE clause and the NotInList event. You can reduce user keystrokes by user keystrokes by using mouse clicks and double clicks to automatically enter data and perform advanced lookups. Additionally, you can add decisions support features such as statistical analysis, optimization, and simulation. By adding more logic and control to your code, you can add expert system features that automatically computer or estimate key values.

4.6 Custom reports

Report writers are good at formatting data for common reports. Some report writers are powerful enough to handle complex reports. Some times you might encounter reports that are difficult to create with a report writer. Sometimes you want to conditional formatting, where the output changes based on the data. So, we should use a report writer to display the final report. Creating reports with the programming is time-consuming.

The fist step is to create a query that provides the data on the report. Then you can write a program that start at top of the query and examines each row. Your code becomes more complex if you want to keep all elements within a group on one page. Then you have to do termine the size of the group before it is printed.

You code must identify the breaks for each group. You should also keep track of the number of lines so that you can identity the end of the page.

4.7 Table operation

Database performance can be controlled by telling the DBMS how to store and retrieve the data in each table. To understand the difference between storage method, you must first

understand how the DBMS will use the data. Then by evaluating how each storage method affects the various table operation, you can choose the best method for your particular application.

Three major categories of operation affect table

- Retrieving data
- Storing data
- Reorganizing the database

Retrieve Data

Retrieving data constitutes some of the most common activities in a database application. These operations also present the best opportunity to improve performance. Application commonly performs three types of data retrieval. They read entire table, read the next row in a sequence, and find and retrieve an arbitrary row.

Store Data

A DBMS has to perform three basic operation involved with storing data: inserting a new row, deleting a row, modifying the data in a row. Most system implement a fast delete operation-they do not actually remove the deleted data.

Reorganize the Database

Largely because of the deletion method, database can become disorganized over time. Data that is flagged as deleted is still hiding in the table space. Empty hole of storage space are too small to hold new data, and data rows that are used together are no longer stored near each other.

These problems are particularly challenging with relational databases. In a relational databases the system data is also stored in tables.

4.9Storage System in DBMS

A database system provides an ultimate view of the stored data. However, data in the form of bits, bytes get stored in different storage devices.

In this section, we will take an overview of various types of storage devices that are used for accessing and storing data.

4.10Data Storage Method

For storing the data, there are different types of storage options available. These storage types differ from one another as per the speed and accessibility.

- Sequential Storage
- Indexes
- Pointer

- Linked List
- Direct or Hashed Access
- Bitmap Index

Sequential storage

Every file record contains a data field (attribute) to uniquely identify that record. In sequential file organization, records are placed in the file in some sequential order based on the unique key field or search key. Practically, it is not possible to store all the records sequentially in physical form.

Indexes – Access Method

- Index is the small table having two columns. The first columns consist of primary key of the table and second column consists of a set of pointers holding the address of the disk, where the particular key (value) can found.
- The indexes are very useful to improve the search operation in the DBMS system.

B+ Tree

- The B+ tree is a balanced binary search tree. It follows a multi-level index format.
- In the B+ tree, leaf nodes denote actual data pointers. B+ tree ensures that all leaf nodes remain at the same height.
- In the B+ tree, the leaf nodes are linked using a link list. Therefore, a B+ tree can support random access as well as sequential access.

Hash File Organization

Hash File Organization uses the computation of hash function on some fields of the records. The hash function's output determines the location of disk block where the records are to be placed.

When a record has to be received using the hash key columns, then the address is generated, and the whole record is retrieved using that address. In the same way, when a new record has to be inserted, then the address is generated using the hash key and record is directly inserted. The same process is applied in the case of delete and update.

In this method, there is no effort for searching and sorting the entire file. In this method, each record will be stored randomly in the memory.

Bitmap indexes

- Bitmap index is used to work with well for low-cardinal (refers to columns few

  unique values) columns in tables.
  For example: boolean data which has only two values true or false.

Storing data column

For basic business data consisting of numbers and short text, it rarely matters how individual

columns are stored. However, business applications are being developed that need to store

more complex data such as large amounts of text, graphics, sound, and even vido clips. This data is releatively complex and requires significantly more storage space.

Text and numbers

Fixed-width or positional-column storage is the simplest means of storing a row of data. Each column is allocated a fixed number of bytes and the data is stored in a set position. If data widths so not vary much, this method is a fast, efficient means to store columns. If descriptions can be short or very long, then you will have to allocate space for the longest possible description, which wastes spaces for the short descriptions.

Image and binary data

Most current database now provide the ability to store binary data within the database itself. For example, you can create a column to hold a picture for each row. Unfortunately, no standards exist for defining these columns or using this data. Hence, if you sue these features, it is difficult to convert your database to another vendor's format. The data type varies depending on the dbms: Access uses on OLE object column. Oracle uses LONG RAW or BLOB(Binary large object). SQL server uses image. More importantly the internal data format, and the storage and retrieval methods are different for each vendor.

The alternative to storing binary files within the DBMS is to store them in a separate subdirectory and then store only the file name within a text column in the database.

4.11Data clustering and Partitioning

Another way to improve database performance is to control the location of individual components of the table. For example, some parts of your application may always be retrieved together, so performance might improve if the two sets of data are retrieved together.

Data clustering

To improve general system performance, most computers retrieve data in chunks. They try to anticipate the next demand and read ahead of the current request. If the system guesses correctly, the next data request can be filled from RAM, which is substantially faster. Database systems designers have used this concept to improve performance of database applications. Some parts of an application are generally used at the same time. If you are using DBMS that supports data clustering, you can improve performance by identifying data that is commonly accessed together.

Data Partitioning

Another situation that commonly arises in business application is that some data is used more frequently than other data. Even in the same table, you might collect data that is used only occasionally.

Horizontal partitioning

Horizontal Partitioning divides a large table into smaller manageable parts without having to create separate tables for each part.

Data in a partitioned table is physically stored in groups of rows called a partition. Each partition can be accessed and maintained separately.

Vertical partitioning

Vertical partitioning divides a table into multiple tables that contain fewer columns.

The two types of Vertical partitioning are Normalization and Row splitting.

Normalization is a process of removing redundant columns from a table and putting them in other tables that linked with foreign key relationships.

Row Splitting divides the original table vertically into tables with fewer columns.

4.12 Disk stripping and RAID storage

RAID refers to redundancy array of the independent disk. It is a technology which is used to connect multiple secondary storage devices for increased performance, data redundancy or both. It gives you the ability to survive one or more drive failure depending upon the RAID level used.

It consists of an array of disks in which multiple disks are connected to achieve different goals.

RAID technology

There are 7 levels of RAID schemes. These schemas are as RAID 0, RAID 1, ..., RAID 6.

These levels contain the following characteristics:

- o   It contains a set of physical disk drives.

- o In this technology, the operating system views these separate disks as a single logical disk.
- o In this technology, data is distributed across the physical drives of the array.
- o Redundancy disk capacity is used to store parity information.
- o In case of disk failure, the parity information can be helped to recover the data.

UNIT- 5

5.DBA (Database Administration)

Database management is among the fundamental processes in the software field of computing. Databases are used in almost every enterprise and the bigger the enterprise is, the bigger and more complicated the databases will have to be. Although facilitated to the greatest extent possible by database management system solutions, it still needs continuous human intervention. And while a system administrator can, in most cases, take care of the database, there are a lot of times when a specialist is needed, someone dedicated to database management.

5.1 Database administrator (DBA)

The people responsible for managing databases are called database administrators. A database administrator (DBA) directs or performs all activities related to maintaining a successful database environment.

Responsibilities include designing, implementing, and maintaining the database system; establishing policies and procedures pertaining to the management, security, maintenance, and use of the database management system; and training employees in database management and use.

The DBA position is generally staffed by a specialist who is trained in the administration of a particular DBMS.

5.2 Role of DBA

- A database administrator's responsibilities can include the following tasks:
- Installing and upgrading the database server and application tools
- Allocating system storage and planning future storage requirements for the database system
- Creating primary database storage structures (table spaces) after application developers have designed an application
- Creating primary objects (tables, views, indexes) once application developers have designed an application
- Modifying the database structure, as necessary, from information given by application developers
- Enrolling users and maintaining system security
- Ensuring compliance with database vendor license agreement Controlling and monitoring user access to the database Monitoring and optimizing the performance of

the database Planning for backup and recovery of database information Maintaining archived data on tape

- Backing up and restoring the database Contacting database vendor for technical support.

5.3 Database tasks by development stages

The tasks of a database are performed by the application developers.

Data base planning Database design Database implementation

Database operation &maintenances

Database Planning

- The first step is to estimate the data storage requirements.
- There initial estimates helps to determine the size and capacity of the hardware needed to support the application.
- The initial investigation should also provide some idea of the number of forms and reports that will be needed, as well as their complexity.
- These numbers will be used to estimate the time and cost required to develop the system.

Database Design

- The goal of design stage is to identify the user needs and design the appropriate data tables.
- Team work coordination &project management are important administrative task at this stage.
- Team work is supported with data standards as defined by the DA.
- Communication is enhanced through a shared data repository, networked tools, E-mail, Computer aided software engineering (CASE) tools.
- The project is often split by assigning forms and reports to individual team members. Each person is then responsible for identifying the business assumptions and defining the normalized tables needed for the assigned forms.

Database Implementation

Developments of the application & user interface are the major steps in database implementation.

Standards for application programming

- User interface must be carefully chosen.
- Programming structure and test procedures help ensures compatibility of the components and provide quality control.
- Programming variable &objects.
- Test procedure. Data access and ownership Loading database

5.4 Backup and recovery system

Perhaps most critical database management task is backup. No matter how well you plan, no matter how sophisticated your security system. Something well go wrong. Database manager and developer have an obligation to plan for disasters. The most critical aspect of planning to make sure that a current copy of database is easily accessible. Any types of disasters like fire, flood, terrorist attack, power failure computer virus, disk drive crash or accidental deletion required backup data. If there is a problem the database has to be restored from the backup.

DBAs are responsible for developing, implementing, and periodically testing a backup and recovery plan for the databases they manage. Even in large shops where a separate system administrator performs server backups, the DBA has final responsibility for making sure that the backups are being done as scheduled and that they include all the files needed to make database recovery possible after a failure. When failures do occur, the DBA needs to know how to use the backups to return the database to operational status as quickly as possible, without losing any transactions that were committed. There are several ways the database can fail, and the DBA must have a strategy to recover from each. From a business standpoint, there is a cost to doing backups, and the DBA makes management aware of the cost/risk trade-offs of various backup methods.

5.5 Security and privacy issue of database

Computer security is an issue with every company today and any computer application files security problem. The purpose of database is to share data. In a security context you want to control who can share the data and what user can do with it. There are basic categories computer securities.

• Physical security:-

It is the concerned with physical protecting the computing resources and preparing for avoiding physical disaster that might damage equipment or data. The must important task is to make sure that you always have current backup.

• Logical security:-

It consists of protecting the data and controlling access to the data. The essence of logical security is that you want to allow each user to have some access to the data but you want to control exactly what types of access the user will have.

5.6 Distributed Database

- Distributed database is a system in which storage devices are not connected to a common processing unit.

- Database is controlled by Distributed Database Management System and data may be stored at the same location or spread over the interconnected network. It is a loosely coupled system.

- Shared nothing architecture is used in distributed databases.

Distributted Dtatabase system
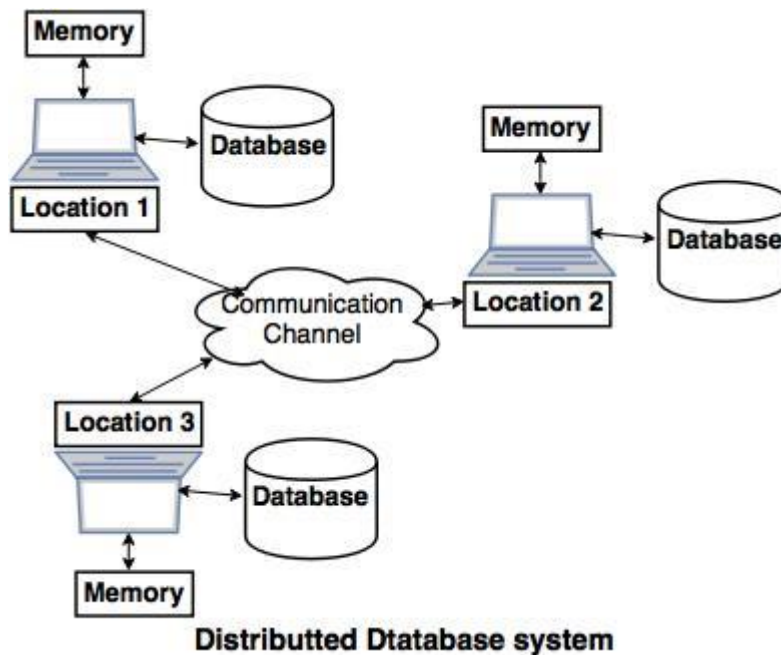
- The above diagram is a typical example of distributed database system, in which communication channel is used to communicate with the different locations and every system has its own memory and database.

5.6.1 Goals of Distributed Database system.

The concept of distributed database was built with a goal to improve:

Reliability: In distributed database system, if one system fails down or stops working for some time another system can complete the task.
Availability: In distributed database system reliability can be achieved even if sever fails down. Another system is available to serve the client request.
Performance: Performance can be achieved by distributing database over different locations. So the databases are available to every location which is easy to maintain.

5.6.2 Types of distributed databases.

The two types of distributed systems are as follows:

1. Homogeneous distributed databases system:

- Homogeneous distributed database system is a network of two or more databases (With same type of DBMS software) which can be stored on one or more machines.
- So, in this system data can be accessed and modified simultaneously on several databases in the network. Homogeneous distributed system are easy to handle.

Example: Consider that we have three departments using Oracle-9i for DBMS. If some changes are made in one department then, it would update the other department also.
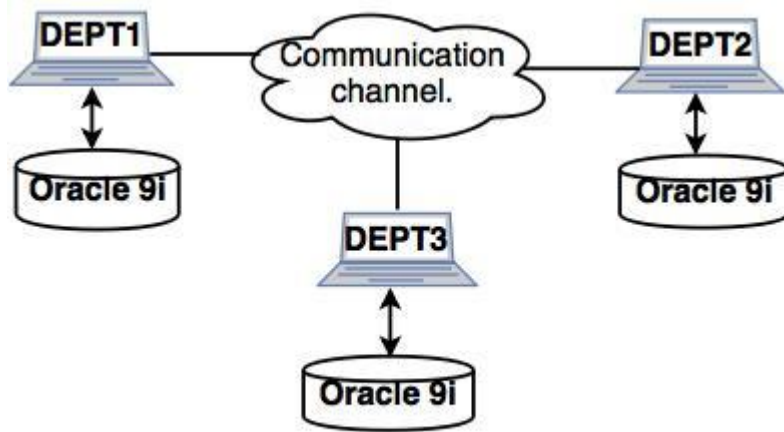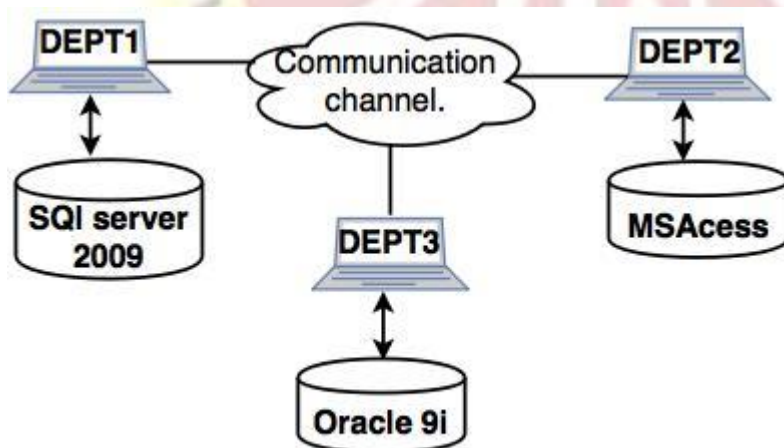
Homogeneous distributed system

2. Heterogeneous distributed database system.

- Heterogeneous distributed database system is a network of two or more databases with different types of DBMS software, which can be stored on one or more machines.

- In this system data can be accessible to several databases in the network with the help of generic connectivity (ODBC and JDBC).

Example: In the following diagram, different DBMS software are accessible to each other using ODBC and JDBC.

Heterogeneous distributed system

5.7 Data replication

Data replication is the process in which the data is copied at multiple locations (Different computers or servers) to improve the availability of data.

5.7.1 Goals of data replication

Data replication is done with an aim to:

- Increase the availability of data.
- Speed up the query evaluation.

5.7.2 Types of data replication

There are two types of data replication:

1. Synchronous Replication:
In synchronous replication, the replica will be modified immediately after some changes are made in the relation table. So there is no difference between original data and replica.
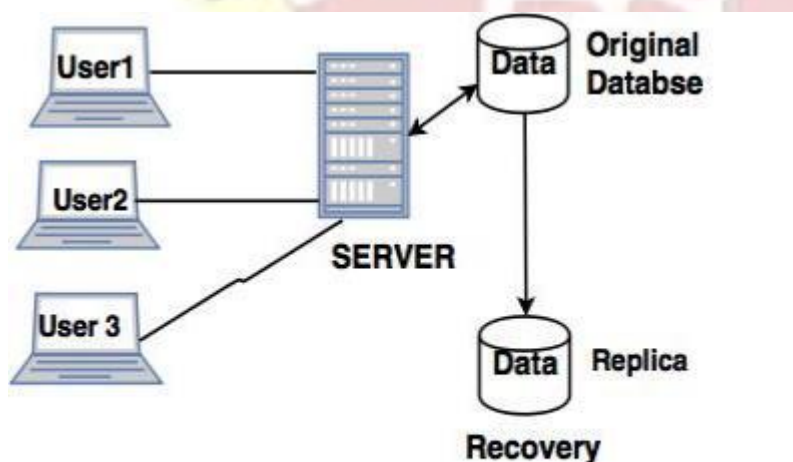
2. Asynchronous replication:
In asynchronous replication, the replica will be modified after commit is fired on to the database.

5.7.3 Replication Schemes

The three replication schemes are as follows:

1. Full Replication

In full replication scheme, the database is available to almost every location or user in communication network.

Full Replication Process In Distributed System

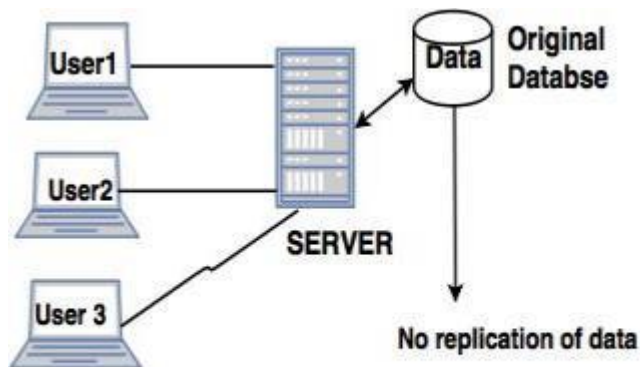**Advantages of full replication**

- High availability of data, as database is available to almost every location.
- Faster execution of queries.

Disadvantages of full replication

- Concurrency control is difficult to achieve in full replication.
- Update operation is slower.

2. No Replication

No replication means, each fragment is stored exactly at one location.

No Replication Process in Distributed Databases
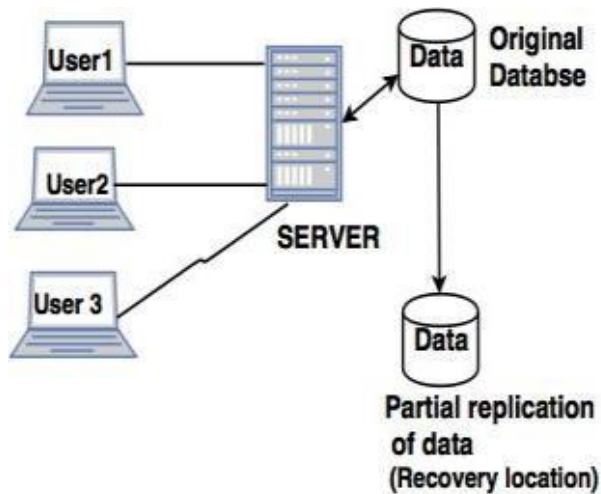
Advantages of no replication

- Concurrency can be minimized.
- Easy recovery of data.

Disadvantages of no replication

- Poor availability of data.
- Slows down the query execution process, as multiple clients are accessing the same server.

3. Partial replication

Partial replication means only some fragments are replicated from the database.

Partial Replication Process In Distributed System

Advantages of partial replication
The number of replicas created for fragments depend upon the importance of data in that fragment.

Distributed databases - Query processing and Optimization

DDBMS processes and optimizes a query in terms of communication cost of processing a distributed query and other parameters.

5.8 Various factors which are considered while processing a query are as follows:

5.8.1 Costs of Data transfer

- This is a very important factor while processing queries. The intermediate data is transferred to other location for data processing and the final result will be sent to the location where the actual query is processing.
- The cost of data increases if the locations are connected via high performance communicating channel.
- The DDBMS query optimization algorithms are used to minimize the cost of data transfer.

5.8.2 Semi-join based query optimization

- Semi-join is used to reduce the number of relations in a table before transferring it to another location.
- Only joining columns are transferred in this method.
- This method reduces the cost of data transfer.

5.8.3 Cost based query optimization

- Query optimization involves many operations like, selection, projection, aggregation.
- Cost of communication is considered in query optimization.
- In centralized database system, the information of relations at remote location is obtained from the server system catalogs.
- The data (query) which is manipulated at local location is considered as a sub query to other global locations. This process estimates the total cost which is needed to compute the intermediate relations.

5.9 Distributed Transactions

- A Distributed Databases Management System should be able to survive in a system failure without losing any data in the database.
- This property is provided in transaction processing.
- The local transaction works only on own location(Local Location) where it is considered as a global transaction for other locations.
- Transactions are assigned to transaction monitor which works as a supervisor.
- A distributed transaction process is designed to distribute data over many locations and transactions are carried out successfully or terminated successfully.
- Transaction Processing is very useful for concurrent execution and recovery of data.

### 5.9.1 Client-server architecture of Distributed system.

- A client server architecture has a number of clients and a few servers connected in a network.
- A client sends a query to one of the servers. The earliest available server solves it and replies.
- A Client-server architecture is simple to implement and execute due to centralized server system.
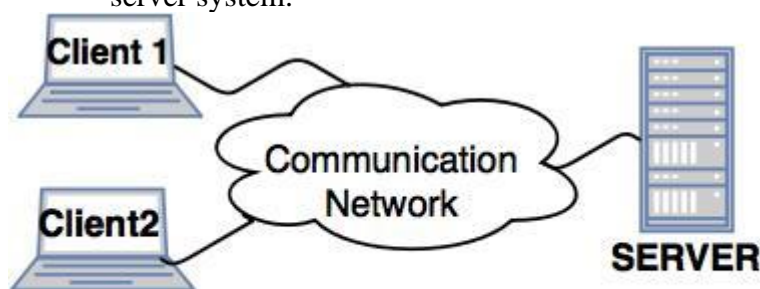
Client-server architecture

5.9.2 Collaborating server architecture.

- Collaborating server architecture is designed to run a single query on multiple servers.
- Servers break single query into multiple small queries and the result is sent to the client.
- Collaborating server architecture has a collection of database servers. Each server is capable for executing the current transactions across the databases.

Collabrorating server architecture
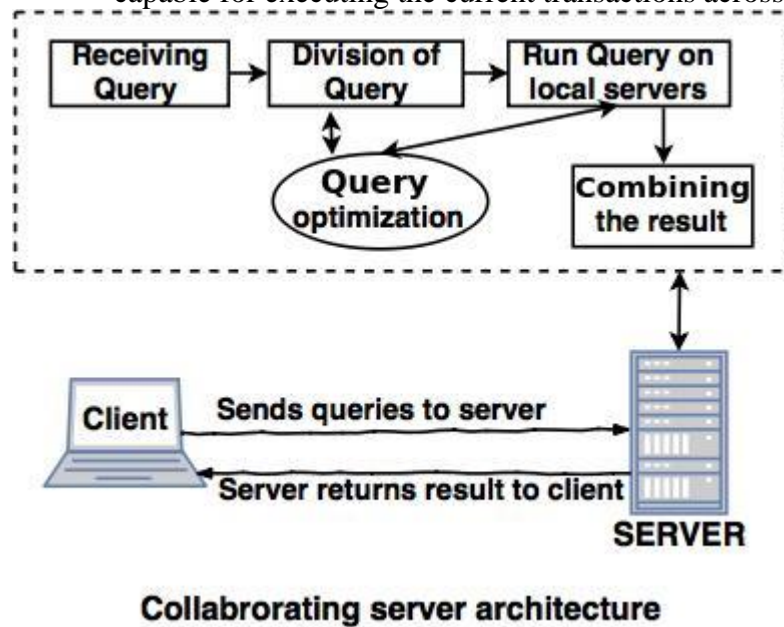
5.9.3 Middleware architecture.

- Middleware architectures are designed in such a way that single query is executed on multiple servers.
- This system needs only one server which is capable of managing queries and transactions from multiple servers.
- Middleware architecture uses local servers to handle local queries and transactions.
- The softwares are used for execution of queries and transactions across one or more independent database servers, this type of software is called as middleware.

5.10 Object Oriented Database

The **ODBMS** which is an abbreviation for **object oriented database management system**, is the data model in which data is stored in form of objects, which are instances of classes. These classes and objects together makes an object oriented data model.

Components of Object Oriented Data Model:
The OODBMS is based on three major components, namely: Object structure, Object classes, and Object identity. These are explained as following below.

Object **Structure:**
The structure of an object refers to the properties that an object is made up of. These properties of an object are referred to as an attribute. Thus, an object is a real world entity with certain attributes that makes up the object structure. Also an object encapsulates the data code into a single unit which in turn provides data abstraction by hiding the implementation details from the user.

The object structure is further composed of three types of components: Messages, Methods, and Variables. These are explained as following below.

1. **Messages –**
   A message provides an interface or acts as a communication medium between an object and the outside world. A message can be of two types:
   - **Read-only message:** If the invoked method does not change the value of a variable, then the invoking message is said to be a read-only message.

   - **Update message:** If the invoked method changes the value of a variable, then the invoking message is said to be an update message.

2. **Methods –**
   When a message is passed then the body of code that is executed is known as a method. Every time when a method is executed, it returns a value as output. A method can be of two types:
   - **Read-only method:** When the value of a variable is not affected by a method, then it is known as read-only method.

   - **Update-method:** When the value of a variable changes by a method, then it is known as an update method.

3. **Variables –**
   It stores the data of an object. The data stored in the variables makes the object distinguishable from one another.

5.10.1 Object Classes:

An object which is a real world entity is an instance of a class. Hence first we need to define a class and then the objects are made which differ in the values they store but share the same class definition. The objects in turn corresponds to various messages and variables stored in it.

Example –

class CLERK

{ //variables

   char name;

   string address;

   int id;

   int salary;

  //methods

  char get_name();

  string get_address();

  int annual_salary();

 };

In above example we can see, CLERK is a class that holds the object variables and messages.

An OODBMS also supports inheritance in an extensive manner as in a database there may be many classes with similar methods, variables and messages. Thus, the concept of class hierarchy is maintained to depict the similarities among various classes.

The concept of encapsulation that is the data or information hiding is also supported by object oriented data model. And this data model also provides the facility of abstract data types apart from the built-in data types like char, int, float. ADT's are the user defined data types that hold the values within it and can also have methods attached to it.

Thus, OODBMS provides numerous facilities to it's users, both built-in and user defined. It incorporates the properties of an object oriented data model with a database management system, and supports the concept of programming paradigms like classes and objects along with the support for other concepts like encapsulation, inheritance and the user defined ADT's (abstract data types).

*****