UNIT-3

RELATIONAL DATABASE MANAGEMENT SYSTEMS (BASIC)

SESSION 1: APPRECIATE THE CONCEPT OF DATABASE MANAGEMENT SYSTEM

Database:

A database is an organized collection of data. In other words, it is a container of information.
Example: Microsoft Access, OpenOffice.org Base, and MySQL.

Database Management System (DBMS):

A database management system is a software package with computer programs that controls the creation, maintenance and use of a database.

A DBMS allows different user application programs to concurrently access the same database.
Example: Oracle, IBM DB2, Microsoft SQL server, Microsoft Access, PostgreSQL, MySQL, FoxPro and SQLite.
Data can be organized into two types:

- **Flat File**: Data is stored in a single table. Usually suitable for less amount of data.
- **Relational**: Data is stored in multiple tables and the tables are linked using a common field. Relational is suitable for medium to large amount of data.

**Database Servers:**

Database servers are dedicated computers that hold the actual databases and run only the DBMS and related software.

Databases on the database servers are accessed through command line or graphic user interface tools referred to as **Frontends**; database servers are referred to as **Back-ends**. Such type of data access is referred to as Client-server model.
Relational Database Management Systems (RDBMS):

- RDBMS store data in the form of tables.
- These tables are connected to one another using key field.
- The RDBMS is the most popular database system among organizations across the world. It provides a dependable method of storing and retrieving large amounts of data while offering a combination of system performance and ease of implementation.
**Database Concepts:**
Database contains objects that are used for storing and managing information.

1. **Item:** - Item is about which information is stored in the database.
2. **Field:** - Each question that we ask about our item is a Field.
3. **Record:** - Record is a set of information (made up of fields) stored in your database about one of the items.
4. **Value:** - Value is the actual text or numerical amount or date that you put in while adding information to your database.

**For example,**
**Database :- Employee**

<table>
<thead>
<tr>
<th>EMP_ID</th>
<th>EMP_NAME</th>
<th>EMP_DESGNTN</th>
<th>EMP_DOB</th>
</tr>
</thead>
<tbody>
<tr>
<td>E001</td>
<td>ABC</td>
<td>Manager</td>
<td>12/8/1987</td>
</tr>
<tr>
<td>E002</td>
<td>XYZ</td>
<td>Team Lead</td>
<td>5/4/1980</td>
</tr>
<tr>
<td>E003</td>
<td>MNO</td>
<td>Head ops</td>
<td>6/8/1966</td>
</tr>
<tr>
<td>E004</td>
<td>LMN</td>
<td>Team Lead</td>
<td>5/9/1988</td>
</tr>
<tr>
<td>E005</td>
<td>UVW</td>
<td>Engg</td>
<td>23/1/1989</td>
</tr>
</tbody>
</table>

**Item :** Employee  
**Field :** Emp_Code , Emp_Name , Emp_Address , Emp_Designation , Emp_ContactNo , Emp_Salary  
**Value :** E001 , ABC , Meerut , Manager , 9876543210 , Rs. 50,000  
**Record :**
<table>
<thead>
<tr>
<th>EMP_ID</th>
<th>EMP_NAME</th>
<th>EMP_ADDRESS</th>
<th>EMP_DESGNTN</th>
<th>EMP_CONTACTNO</th>
<th>EMP_SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>E001</td>
<td>ABC</td>
<td>Meerut</td>
<td>Manager</td>
<td>9876543210</td>
<td>Rs. 50,000</td>
</tr>
</tbody>
</table>

**Key Field :-** Key Field is a value in a Field that uniquely identifies the record.  
Eg. E001 which is unique to every employee.
**PRIMARY KEY:** A primary key is a unique value that identifies a row in a table. These keys are also indexed in the database, making it faster for the database to search a record.

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Forename</th>
<th>Surname</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Simon</td>
<td>Jones</td>
</tr>
<tr>
<td>2</td>
<td>Emma</td>
<td>Price</td>
</tr>
<tr>
<td>3</td>
<td>Laura</td>
<td>Jones</td>
</tr>
<tr>
<td>4</td>
<td>Jonathan</td>
<td>Hale</td>
</tr>
<tr>
<td>5</td>
<td>Emma</td>
<td>Smith</td>
</tr>
</tbody>
</table>

When **primary** key constraint is applied on one or more columns then it is known as **Composite Primary Key**.

**FOREIGN KEY:** The foreign key identifies a column or set of columns in one (referencing) table that refers to a column or set of columns in another (referenced) table.

**Note:** The “one” side of a relation is always the parent, and provides the PK(Primary Key) Attributes to be copied. The “many” side of a relation is always the child, into which the FK(Foreign Key) attributes are copied.

**one parent, PK (Primary Key) ; many child , FK (Foreign Key)**
SESSION 2: CREATE AND EDIT TABLES USING WIZARD AND SQL COMMANDS

**TABLE:**
A table is a set of data elements that is organized using a model of vertical columns and horizontal rows. Each row is identified by a unique key index or the key field.

**COLUMNS OR FIELD OR ATTRIBUTES:**
A column is a set of data values of a particular simple type, one for each row of the table. For eg. Emp_Code, Emp_Name, Emp_Address etc.

**ROWS OR RECORDS OR TUPLES**
A row represents a single, data items in a table. Each row in a table represents a set of related data, and every row in the table has the same structure.

**DATA TYPES:**
Datatypes are used to identify the type of data we are going to store in the database.

**Categories of data types**- Data types can be broadly classified into five categories:

- Numeric Types
- Alphanumeric Types
- Binary Types
- Date Time
- Other variable Types
**NUMERIC TYPES:**

They are used for describing numeric values like mobile number, age, etc.

**The different types of numeric data types available are:-**

1. Boolean (Yes / No)  
2. TinyInt (Tiny Integer)  
3. SmallInt (Small Integer)  
4. Integer  
5. BigInt (Big Integer)  
6. Numeric  
7. Decimal  
8. Real  
9. Float  
10. Double

**ALPHANUMERIC TYPES:-**

**The list of different datatypes available in alphanumeric types are :-**

1. LongVarChar (Memo) (Long Text)  
2. Char (Text-fix) (Small Text)  
3. VarChar (Text) (Text of specified Length)  
4. VarChar_IgnoreCase (Text) (Comparisions are not case sensitive)

**BINARY TYPES:-**

Binary types are used for storing data in binary formats. It can be used for storing photos, music files or (in general file of any format) etc.

**The list of different datatypes available in Binary types are :-**

1. LongVarBinary (Image)  
2. Binary (Binary (fix) )  
3. VarBinary (Binary)

**DATE TIME:-**

Date time data types are used for describing date and time values for the field used in the table of a database. It can be used for storing information such as date of birth, date of admission etc.
The list of different data types available in Date Time type are :-
1. Date (Stores month, day and year information)
2. Time (Store hour, minute and second information)
3. Timestamp (Stores date and time information)

OTHER DATA TYPES

Other/ object

Launching Openoffice

Start → Programs → OpenOffice.org 3.4.1 → OpenOffice.org.

(OR)

Double click on shortcut icon

Creating Database:

- To create database select “Database” option in the open office launch window.
- Database wizard appears
- create a new database by selecting the option “Create a new database”.
- We can also open an existing database file that you have already created by selecting the option “Open an existing database” file. Click Next. And then click finish
- The Save As dialog box appears.
- Specify a name for the database in the File name: field and click Save.
- Database will be created
Creating Tables (Design view)

• Tables are the basic building blocks of a database. You store the data in the database in the form of tables

• After creating the database, in the database window. Click on “Create Table in Design View” option available under Tasks and a Table Design window appears.

• Specify the field name and data type of the field to be created by selecting the appropriate type available under Field type dropdown list.

• Then save the table. Specify the table name. The default name is Table1. Click OK.

• Then, a dialog box appears to set a primary key for the table created.
• we can select the appropriate option to set the primary key or leave the table without a primary key.

• If we click Yes, the application will set the primary key for the first field created automatically.

Inserting data to the table created in design view:

In the database view, we can see the name of the table created. Double click on the table name, table data view appears wherein we can enter the records.

Click on save and close the data view.
Create table using SQL DDL Command

SQL commands are also used to manage and manipulate data in a database.

The SQL commands are categorized as:

- 1. DDL (Data Definition Language)
- 2. DML (Data Manipulation Language)

**DATA DEFINITION / DESCRIPTION LANGUAGE:**

It is a standard for commands that define the different structures in a database. DDL statements create, modify and remove database objects such as tables, indexes and users.

**Common DDL Statements are:**

1. Create :- Used to create database objects.
2. Alter :- Used to modify database objects.
3. Drop :- Used to delete database objects.

**Example:**

To create a student table, SQL command used is as follows:

```sql
Create table Student(Stud_Id integer Primary Key, Stud_Name varchar(20), Stud_Class varchar(3), Stud_Address varchar(30));
```
SESSION 3: Perform Operations on Table

In a database we can define the structure of the data and manipulate the data using some commands. There are two types of languages for this task. These are:

- **Data Definition Language (DDL)**
- **Data Manipulation Language (DML)**

**DATA MANIPULATION LANGUAGE:**

It is a standard for commands that enables users to access and manipulate data in a database.

**Common DML Statements are:**

1. **SELECT** :- Used for retrieval of information from the database.
2. **INSERT** :- Used for insertion of new information into the database.
3. **DELETE** :- Used for deletion of information in the database.
4. **UPDATE** :- Used for modification of information in the database.

A query language is a part of DML involving information retrieval only.

A popular data manipulation language is Structured Query Language (SQL). This is used to retrieve and manipulate data in a relational database. Other forms of DML are those used by IMS/DLI, CODASYL databases, such as IDMS and others. Data manipulation language comprises the SQL data change statements, which modify stored data but not the schema or database objects.
Types of DML:

1. **Procedural:** The user specifies what data is needed and how to get it.

2. **Non Procedural:** The user only specifies what data is needed. This is easier for the user but may not generate code as efficient as that produced by procedural languages.

*Note:* A popular data manipulation language is SQL (Structured Query Language.)

**Insert statement:**

syntax: `insert into <table_name> <column1, column2, column3...> values <value1, value2, value3 ...>;`

Example: `insert into Sdetails ("ID","Name","Rollno","DOB","Class","Phone","Email","Color","Location")values ('8', 'Ranjith Singh', '67', '12-03-99','X','435363', 'ranjth99@gmail.com', 'White', 'Bihar');`

**Delete statement:**

Syntax: `DELETE FROM <table_name> [WHERE] <condition>;`

Example: `delete from SDetails where ID=8;`
SORTING DATA

Sorting means to arrange the data in either ascending order of descending order. Select the column then click on sort buttons. Records in the table will be sorted.

Referential Integrity

*Referential integrity* is used to maintain accuracy and consistency of data in a relationship.

In Base, data can be linked between two or more tables with the help of primary key and foreign key constraints.

Referential integrity helps to avoid:

- Adding records to a related table if there is no associated record available in the primary key table.
- Changing values in a primary if any dependent records are present in associated table(s).
- Deleting records from a primary key table if there are any matching related records available in associated table(s).

Creating and Editing Relationships between Tables

A relationship refers to an association or connection between two or more tables. When we relate two tables, we don't need to enter the same data in separate tables.
Relationships between tables helps to:

- Save time as there is no need to enter the same data in separate tables.
- Reduce data-entry errors.
- Summarize data from related tables.

We can create a relationship between any two tables by selecting Relationships... option from the Tools menu.

Add the tables in amongst which you want to create the relationship. Select the tables and click on Add button.

There are three types of relationships which can be created in tables:

1. **ONE to ONE**
2. **ONE to MANY** OR **MANY to ONE**
3. **MANY to MANY**
There are two ways to create the relationships between the tables:

a. Click on Insert option and select New Relation... option in Relation Design window.

Select the options as required:

b. Drag the primary key column from one table and drop it on the key column of another table.
One to One Relationship

In this relationship, both the tables must have primary key columns.
Example: In the given tables EMP and DEPT, EMP_ID in EMP table and DEPT_ID in DEPT table are the primary keys.

One to Many Relationship

In this relationship, one of the table must have primary key column.
It signifies that one column of primary key table is associated with all the columns of associated table.
Example: In the given tables EMP and DEPT, EMP_ID in EMP table is the primary key.
Many to Many Relationship

In this relationship, no table has the primary key column.

It signifies that all the columns of primary key table are associated with all the columns of associated table.

Example: In the given tables EMP and DEPT, there is no primary key.

Remove the Relationships

The relationships applied on the tables can be removed also with the help of Delete option. Right Click on the relationship thread and select Delete option.
SESSION 4: RETRIEVE DATA USING QUERY

• **QUERY:**

A database query extracts data from a database and formats it into a human-readable form. A query must be written in the syntax the database requires (OR) Query is to collect specific information from the pool of data. A query helps us join information from different tables and filter that information. **Filtering** means that the query uses criteria you provide it to hide some data and present only what you want to see.

• **Query languages:**

Query languages are computer languages used to make queries into databases and information systems. Queries are commands that are used to define the data structure and also to manipulate the data in the database.
A SELECT statement retrieves zero or more rows from one or more database tables or database views. In most applications, SELECT is the most commonly used Data Manipulation Language (DML) command.

The SELECT statement has many optional clauses:

- **WHERE** specifies which rows to retrieve.
- **ORDER BY** specifies an order in which to return the rows.

To retrieve all the columns in a table the syntax is:

**Select Statement:**

```sql
syntax : select * from <tablename>;
```

In order to execute queries click on the **Queries** option available on the left side under database section, click **Create Query in SQL View**.

We can type the query in the window and execute it by using the F5 function key or by clicking the icon in the window.

For example, to display all the data in the table that we have created then the select statement will be:

**Example 1 : select * from SDetails;**

(This displays all the records present in the table)
Grouping of Data

To display the records containing the same type of values “WHERE” clause can be used with the Select SQL Command.

To get details about the list of students whose favorite color is blue, you can use:

```sql
select * from SDetails where Color='Blue';
```

To view records in ascending order of RollNo, from the table the select statement will be:

```sql
select * from SDetails order by “Rollno” ASC;
```

• **Update statement :**

Update statement is used for modifying records in a database. The general syntax of the update statement is as follows:

```sql
Syntax : UPDATE <table_name> SET <column_name> = value [, column_name = value ...] [WHERE <condition>]

Example : update SDetails set Location = ‘Bhubaneswar’ where Rollno = 14;
```
SESSION 5: CREATE FORMS AND REPORTS USING WIZARD

Creating forms

A form provides the user a systematic way of storing information into the database. It is an interface in a user specified layout that lets users to view, enter, and change data directly in database objects such as tables.

To create a form, Click on Forms option located under Database section

• Click Use Wizard to Create Form... option under Tasks group. The Form Wizard dialog box appears

• From which we can select selective fields to be sent onto the form by selecting the field name and clicking >button. We can select individual fields in a database or all fields in a database.

• To use all the fields in the table in a form, click the>> button.

• Click Next >. We can see the Set up a sub form step dialog box of the wizard

• You can select the option Add Subform if we need to insert the contents in the table in a separate form. Click Next>.

• Now to arrange selected fields in a form. Use different styles from the list displayed

• Once you have selected a style, click Next >

• A dialog box appears wherein we can select the data entry model.
• Click **Next >**. We can see a dialog box wherein we can specify the styles to be used in the form.

• Click **Next >**. In the dialog box specify the name of the form. Click Finish.

• A form window appears. Notice that the records in the table are displayed automatically within the form.
We can add new records to the table using the form by clicking the symbol located at the bottom of the form.

Once you click the symbol, you will be displayed with a window for creating records.

To view the saved records, Double-click **SDetails** (Table name) under **Tables** section. A window similar to the one below will be displayed; notice the record that you created using a form is also displayed along with other records.
Creating Reports

A report helps to display the data in a summarized manner. It is used to generate the overall work outcome in a clear format.

To create report, Click on Reports section under Database in the OpenOffice base application.

Now we can generate the report for the table created earlier.

- Click on Use Wizard to Create Report... option available under Tasks.

- Once you select the Use Wizard to Create Report... option.

- Report Wizard appears

- Wherein we have to select all the table fields by selecting the >> button, once we click the button>>

- Click Next >.

- Once you click Next>.

- We can redefine the label of the fields in the reports or else you can set the default name.
• Click Next > Now we can define grouping for the fields of the table.

• Click Next > sort the field variables in the report by selecting the appropriate field and sorting method.

• Click Next > select the layout of the report by selecting the appropriate option available under the Layout of data drop down list and we can also select the orientation of the report.

• Click Next > Define a name for the report or we can use the name of the table itself for the report a

• Click Finish.

• Once you click Finish. Report will be generated.