



**Database:** A database is an organised collection of data.

**DBMS:** Software used to manage databases is called Data Base Management System (DBMS).

**Relational Database:** A database in which the data is stored in the form of relations (also called tables) is called a Relational Database.

**RDBMS:** A DBMS used to manage Relational Databases is called an RDBMS (Relational Data Base Management System). Some popular RDBMS software available are: MySQL, Ingres, Postgres, Oracle DB2, MySQL Server, Sybase etc. Common Database management tools for mobile devices are SQL Lite, Postgres SQL.

#### Advantage of DBMS:

1) **Data Redundancy is reduced :**

Redundancy means duplication of data. Sometimes it is possible that the same piece of information may be stored in two or more files.

2) **Data Integration:**

In DBMS, data in database is stored in tables. A single database contains multiple tables and relationships can be created between tables. This makes easy to retrieve and update data.

3) **Sharing of data :**

The same data can be shared among different users and applications.

4) **Data Security**

**Data security** is the protection of the database from unauthorized users. Only the authorized persons are allowed to access the database.

#### Disadvantage:

1) Cost

2) Expensive hardware and software requirement

3) Required Experts to handle the database

#### Database Terminology:

	column	column	column
	↓	↓	↓
	id	joketext	jokedate
row →	1	Why did the chicken ...	2012-04-01
row →	2	Knock-knock! Who's ...	2012-04-01

**Data:** Data are collections of raw facts.

**Row/Record/Tuple:** A tuple is a record in a database.

**Field/Column/Attribute:** A field/Column is the names which are placed vertically in a table.

**Table/Relation:** A table is two-dimensional array, which contains intersection of rows and column.

**Degree:** The number of columns in a table is called the degree of the table. *Degree of above table is 3.*

**Cardinality:** The number of Tuples (rows) in the table is called its cardinality. If there are six rows in a table, so the cardinality of table is six. *Cardinality of above table is 2.*



- ✓ MySQL is an open source database management system.
- ✓ MySQL is developed, distributed and supported by Oracle Corporation.

The data in a MySQL database are stored in tables. A table is a collection of related data and it consists of columns and rows.

Databases are useful for storing information categorically. A company may have a database with the following tables:

- ❖ Employees
- ❖ Products
- ❖ Customers
- ❖ Orders

**Relationship:** An association established between common fields (columns) in two tables. A relationship can be one-to-one, one-to-many, or many-to-many.

**Relationship** means- relationship between two or more tables.

#### Types of Relationship

- 1) **One to One:** In a one-to-one relationship, each row in one database table is linked to one and only one other row in another table. In a one-to-one relationship between Table A and Table B, each row in Table A is linked to another row in Table B.  
**Example:** 1) One Student can opt for one stream only in a school. 2) Each employee is associated with one department.
- 2) **One to Many:** In a one-to-many relationship, each row in the related table can be related to many rows in the relating table.  
**Example:** One Order can carry many items to issue.
- 3) **Many to Many:** This type of relationship describes the situation where linking value can appear in multiple records on both side of a relationship.  
**Example:** 1) In a company many employees are associated with multiple projects (completed/existing), and at the same time, projects are associated with multiple employees. 2) For example, all the customers belonging to a bank is stored in a customer table while all the bank's products are stored in a product table. Each customer can have many products and each product can be assigned to many customers.

**Data Type** is the type of value that will be entered into the fields.

- 1) **CHAR(size)** - A fixed-length string from 1 to 255 characters in length right-padded with spaces to the specified length when stored. Values must be enclosed in single quotes or double quotes.
- 2) **VARCHAR(size)** - A variable-length string from 1 to 255 characters in length; for example VARCHAR(25). Values must be enclosed in single quotes or double quotes.
- 3) **DECIMAL(size,d)**
- 4) **INT OR INTEGER**
- 5) **DATE** - A date. Format: YYYY-MM-DD
- 6) **TIME** - A time. Format: HH:MM:SS

**Concept of Keys**

**Definition:** A database key is an attribute utilized to sort and/or identify data in some manner.

1. **Primary Key:** Primary key is a set of attributes that uniquely identifies the tuples within the relation.

StudentID	StudentName	StudentPhone
C101	Chintan	9828672864
C102	Rimjhim	9413345965

In the table above, StudentID is the primary key.

The values placed in primary key columns must be **unique** for each row: no duplicates can be tolerated. In addition, nulls (Blank) are not allowed in primary key columns.

2. **Foreign Key:** It is a single field in the given table which directly references a primary key in another table.

Or

A foreign key is a field in a relational table that matches the primary key column of another table. The foreign key can be used to cross-reference tables.

3. **Alternate Key:** A candidate key that is not primary key is called alternate key.
4. **Candidate Key:** Any column or set of columns that uniquely identifies the rows of a table similar to primary key. When more than one primary key exists the alternative are called Candidate Key.

**Working with MySQL**

1. **DDL (Data Definition Language):** This is a category of SQL commands. All the commands which are used to create, destroy, or restructure databases and tables come under this category. Examples of DDL commands are - CREATE, DROP, ALTER.
2. **DML (Data Manipulation Language):** This is a category of SQL commands. All the commands which are used to manipulate data within tables come under this category. Examples of DML commands are - INSERT, UPDATE, DELETE.

**CREATE DATABASE**

```
CREATE DATABASE <database name>;
```

```
CREATE DATABASE School;
```

**USE DATABASE**

```
USE School;
```

**DISPLAY DATABASES**

```
Show databases;
```

**Creating Table**

After creating a database, the next step is creation of tables in the database. For this CREATE TABLE statement is used.

**Syntax:**

```
CREATE TABLE <TableName>(<ColumnName1> <Data Type1>,  
<ColumnName2> <Data Type2>,...,<ColumnNameN> <Data TypeN>);
```

**Example 1:**

```
CREATE TABLE Learner  
(  
RollNo INTEGER,  
Name VARCHAR(25)  
);
```

**Example 2:**

```
CREATE TABLE Student(  
Rollno INTEGER,  
Name VARCHAR(25),  
Gender CHAR(1),  
Marks1 DECIMAL(4,1));
```

**Insert Record****Syntax:**

```
INSERT INTO <TableName>
VALUES (<Value1>,<Value2>,...,<ValueN>);
```

**Example:**

```
INSERT INTO Learner VALUES (14,'Aruna Asaf Ali');
```

**Select Records**

```
SELECT * FROM <TableName>;
```

**Example:**

```
SELECT * FROM Learner;
```

```
SELECT name FROM Learner;
```

**WHERE clause (Retrieving specific rows)**

Tables usually contain many rows. Mostly, we do not want to display all the rows of a table. Certain rows can be displayed based on the criteria for selection of rows using the keyword WHERE. The WHERE clause is used to filter records. It is used to extract only those records that fulfill a specified criterion.

**Syntax:**

```
SELECT <column name1> [,<column name> ,...] FROM <table name>
```

```
WHERE <condition>;
```

Ms. Sujata wants to display the names and marks of all those students who have secured marks above 80, she enters:

```
SELECT Name,Marks1 FROM Student WHERE Marks1 > 80;
```

```
SELECT name FROM Learner WHERE id=101;
```

**UPDATE STATEMENT**

In the table student, Mr. Vikas entered a student's marks as 93. Suppose, that student found out that one of her answers was unchecked and got her marks increased by 1. How would Mr. Vikas change it in the table? She can use the UPDATE statement to modify existing data in the table.

**Syntax:**

```
UPDATE <table_name>
SET <column name> = <value> [,<column name> = <value> , ...]
[WHERE <condn>;]
```

**Example:**

```
UPDATE Student SET Marks1 = 94 WHERE name = 'Monica Rana';
```

**DELETE STATEMENT**

Sometimes students leave school or an employee leaves an organization. Their rows have to be deleted from the table. Deleting data from a table is very simple. DELETE statement is used to delete rows from a table. DELETE removes the entire row, not the individual column values. Care must be taken while using this statement as accidentally important data may get deleted.

**Syntax:**

```
DELETE FROM <tablename> [Where < condn>;]
```

One of the students with Roll number 14 has left the school and Ms. Sujata wants to delete his/her row. She uses the following statement to delete the row with roll number

```
DELETE FROM Student WHERE Rollno = 14;
```



Consider the following table named "GYM" with details about Fitness products being sold in the store.

Table Name : GYM

(UnitPrice is in Rs.)

PrCode	PrName	UnitPrice	Manufacturer
P101	Cross Trainer	25000	Avon Fitness
P102	TreadMill	32000	AG Fitline
P103	Massage Chair	20000	Fit Express
P104	Vibration Trainer	22000	Avon Fitness
P105	Bike	13000	Fit Express

Write SQL statements to do the following:

- Display the names of all the products in the store.
- Display the names and unit price of all the products in the store
- Display the names of all the products with unit price less than Rs.20000.00
- Display names of all products by the manufacturer "Fit Express"
- Add a new row for product with the details: "P106", "Vibro Exerciser", 23000, manufacturer: "Avon Fitness".
- Change the Unit Price data of all the rows by applying a 10% discount reduction on all the products.

Study the following data and answer the question given below:

Table: Customer

CustNo	CustName	CompanyNo	Phone
101	Rajat	P111	9828898288
102	Vinay	P222	9829998299
105	Sunil Nehra	P333	9828672864

Table: Company

CompanyNo	CompanyDetail	Business	Product
P333	Private	1000	10
P111	Govt	300	22
P222	Private	600	46

(a) Name the field, which can acts as the primary key for the table Company.

Ans: CompanyNo

(b) What type of relationship exists between table Company and Customer.

Ans: One to One.

Q2. The relation Products is used to maintain some data related to products sold in a Grocery Store. Study the table and answer the following questions:

Products				
Prod_ID	Prod_name	Price	Category	Availability
711	Shine Toothpaste	35	SOAPS	20
234	Sugar	45	EDIBLE	52
424	Chocolate Box	125	EDIBLE	12
543	Detergent	62	SOAPS	26

1. Name the field that can act as a primary key in the table Products. [1]
2. If a separate table Price needs to be created to maintain information about each product's price, Name any two fields from the above table that should be included in the new table. [1]
3. Name the type of relation that would exist between the table Products and the newly created table Price. [1]

Q3. Study the following data and answer the questions that follow:

Table : ACCOUNT

Acc_no	Cust_Name	Transaction	Amount	Email_id
A101	Sunil Nehra	Withdrawal	15,000	sknebra97@gmail.com
A102	Siddhant	Deposit	1,40,000	pkb@gmail.com
A103	Sandeep	Deposit	95,000	akm@yahoo.com

1. Suggest the data types that should be used for each of the fields in the above table.

Ans. The data types to be used for each of the fields are:

Field Name	Data Type
Acc_no	Text
Cust_Name	Text
Transaction	Text
Amount	Number or Currency
Email_id	Text or Hyperlink

2. If the Customer details have to be stored in a separate table (named CUSTOMER), then identify the two fields from the above table which should be included in the new table.

Ans. The two fields which should be included in the new table are:

1. Acc\_no
  2. Cust\_Name
  3. Email\_id
- Any two

Q4. Observe the two tables – ITEMS and SALES of a Database named Products given below and answer the questions that follow:

ITEMS		
Field Name	Data Type	Description (Optional)
ITEM_CODE	Number	Stores the Item Code
ITEM_NAME	Short Text	Stores the name of the item

SALES		
Field Name	Data Type	Description (Optional)
BILL_NO	Number	Stores the Bill Number
SALE_DATE	Date/Time	Stores the date of the sale
QUANTITY	Number	Stores the amount of item sold
ITEM_CODE	Number	Stores the Item code

1. Name the view in which the two tables are shown – Design or Datasheet.
2. Name an attribute from the table ITEMS.
3. Identify any one data type from the table SALES.
4. Name the foreign key for the table SALES.
5. Mention the degree of the table ITEMS.
6. What type of relationship will exist between these two tables?