



**GURUKUL INTERNATIONAL SCHOOL**

**JANJGIR**

**A DAY CUM BOARDING SR. SEC. ENGLISH MEDIUM SCHOOL**

**RUN BY MARS EDUCATION SOCIETY - JANJGIR**

Affiliation No: 3330265 | School Code: 15094 | U-Dise No: 22061700909

*We Love, We Share, With Care*



# HAPPY SUMMER HOLIDAYS

# ASSIGNMENT

2024-25

**Class : 12**

# AMATHS

- If A is a singular matrix then adj A is
  - non-singular
  - singular
  - symmetric
  - not define.
- The value of x and y is if  $\begin{bmatrix} 4, & 3 \\ X, & 5 \end{bmatrix} = \begin{bmatrix} Y & Z \\ 1 & 5 \end{bmatrix}$ ,
  - X=5, Y=3
  - X=1, Y=4
  - X=4, Y=1
  - none of these
- The number of all possible matrices of order 3x3 with each entry 0 or 1 is
  - 27
  - 15
  - 81
  - 512
- $A = [a_{ij}]$  m x n is a square matrices, if
  - m<n
  - m>n
  - m=n
  - none of these.
- If A , B are symmetric matrices of same order, then AB-BA is
  - skew symmetric matrix
  - Only symmetric matrix
  - zero matrix
  - Identity matrix.
- Let A be a square matrices. Of order 3 x 3, then  $|K A|$  is equal to
  - $K |A|$
  - $K^2 |A|$
  - $K^3 |A|$
  - $3K |A|$
- If  $A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$  then  $A^5$  is
  - 5A
  - 10 A
  - 16 A
  - 32 A
- If the area of a triangle is 35 squnits with vertices (2,-6), (5,4) and (K,4). then K is
  - 12
  - 2
  - 12,-2
  - 12,-2
- if  $\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} x & 3 \\ 2x & 5 \end{bmatrix}$  the value of x is
  - 4
  - 3
  - 2
  - none of these

10. IF  $A = \begin{bmatrix} 1 & 2 \\ 4 & 2 \end{bmatrix}$ , then show that  $|2A| = 4|A|$

11. using determinantsshow that the points  $A(a, b+c)$   $B(b, c+a)$ ,  
 $C(c, a+b)$  are collinear

12. If  $A = \begin{bmatrix} 0 & -\tan \frac{\alpha}{2} \\ \tan \frac{\alpha}{2} & 0 \end{bmatrix}$  and  $I$  is that identify matrix of order 2, show that

$$I + A = (I - A) \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cot \alpha \end{bmatrix}$$

13. If  $\begin{bmatrix} x + y + z \\ x + z \\ y + z \end{bmatrix} = \begin{bmatrix} 9 \\ 5 \\ 7 \end{bmatrix}$  find  $(x + y + z)$

14. Write and learn all Trigonometric formula.

15. Write and learn all differentiation formula.

16. For what value of  $x$  the matrices  $A = \begin{bmatrix} 1 & -2 & 3 \\ 1 & 2 & 1 \\ x & 2 & -3 \end{bmatrix}$  is singular

17. If  $A = \begin{bmatrix} 1 & 2 & -3 \\ 2 & 3 & 2 \\ 3 & -3 & -4 \end{bmatrix}$ . Hence solve the system of equations

$$x + 2y - 3z = -4$$

$$2x + 3y + 2z = 2$$

$$3x - 3y - 4z = 11$$

(i)  $|A|$

(ii) Find  $A^{-1}$

(iii) Find the value of  $x + y + z$

18. Find a matrix  $A$  such that  $2A - 3B + 5C = 0$ , where  $B = \begin{bmatrix} -2 & 2 & 0 \\ 3 & 1 & 4 \end{bmatrix}$  and

$$C = \begin{bmatrix} 2 & 0 & -2 \\ 7 & 1 & 6 \end{bmatrix}$$

19. Use matrix multiplication to divide Rs.30,000 in two parts such that the total annual interest at 9% on the first part and 11% on the second part amount Rs.3060

20. If  $A = \begin{bmatrix} 3 & -2 \\ 4 & -2 \end{bmatrix}$  and  $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$  find K so that  $A^2 = KA - 2I$

# Biology

1. Starting with the zygote, draw the diagrams of the different stages of embryo development in a dicot.
2. What are the possible types of pollinations in chasmogamous flowers. Give reasons.
3. With a neat, labelled diagram, describe the parts of a mature angiosperm embryo sac. Mention the role of synergids.
4. Draw the diagram of a microsporangium and label its wall layers. Write briefly on the role of the endothecium.
5. Embryo sacs of some apomictic species appear normal but contain diploid cells. Suggest a suitable explanation for the condition.
6. What role does pituitary gonadotropins play during follicular and ovulatory phases of menstrual cycle? Explain the shifts in steroidal secretion.
7. Meiotic division during oogenesis is different from that in spermatogenesis. Explain how and why?
8. The zygote passes through several developmental stages till implantation, Describe each stage briefly with suitable diagrams.
9. Draw a neat diagram of the female reproductive system and label the parts associated with the following (a) production of gamete, (b) site of fertilisation (c) site of implantation and, (d) birth canal.
10. With a suitable diagram, describe the organisation of mammary gland.
11. Corpus luteum in pregnancy has a long life. However, if fertilisation does not take place, it remains active only for 10-12 days. Explain.
12. What is foetal ejection reflex? Explain how it leads to parturition?
13. Except endocrine function, what are the other functions of placenta.
14. Why doctors recommend breast feeding during initial period of infant growth?
15. What are the events that take place in the ovary and uterus during follicular phase of the menstrual cycle



## Section – A

1. To measure the resistance and impedance of an inductor with or without iron core.
2. To measure resistance, voltage (AC/DC), current (AC) and check continuity of a given circuit using multimeter.
3. To assemble a household circuit comprising three bulbs, three (on/off) switches, a fuse and a power source.
4. To assemble the components of a given electrical circuit.
5. To study the variation in potential drop with length of a wire for a steady current.
6. To draw the diagram of a given open circuit comprising at least a battery, resistor/rheostat, key, ammeter and voltmeter. Mark the components that are not connected in proper order and correct the circuit and also the circuit diagram.

## Section – B

1. To identify a diode, an LED, a resistor and a capacitor from a mixed collection of such items.
2. Use of multimeter to see the unidirectional flow of current in case of a diode and an LED and check whether a given electronic component (e.g., diode) is in working order.
3. To study effect of intensity of light (by varying distance of the source) on an LDR.
4. To observe refraction and lateral deviation of a beam of light incident obliquely on a glass slab.
5. To observe diffraction of light due to a thin slit.
6. To study the nature and size of the image formed by a (i) convex lens, or (ii) concave mirror, on a screen by using a candle and a screen (for different distances of the candle from the lens/mirror).
7. To obtain a lens combination with the specified focal length by using two lenses from the given set of lenses.

## Suggested Investigatory Projects (any 2 In File)

1. To study various factors on which the internal resistance/EMF of a cell depends.
2. To study the variations in current flowing in a circuit containing an LDR because of a variation in (a) the power of the incandescent lamp, used to 'illuminate' the LDR (keeping all the lamps at a fixed distance). (b) the distance of a incandescent lamp (of fixed power) used to 'illuminate' the LDR.
3. To find the refractive indices of (a) water (b) oil (transparent) using a plane mirror, an equiconvex lens (made from a glass of known refractive index) and an adjustable object needle.
4. To investigate the relation between the ratio of (i) output and input voltage and (ii) number of turns in the secondary coil and primary coil of a self-designed transformer.
5. To investigate the dependence of the angle of deviation on the angle of incidence using a hollow prism filled one by one, with different transparent fluids.
6. To estimate the charge induced on each one of the two identical Styrofoam (or pith) balls suspended in a vertical plane by making use of Coulomb's law.
7. To study the factor on which the self-inductance of a coil depends by observing the effect of this coil, when put in series with a resistor/(bulb) in a circuit fed up by an A.C. source of adjustable frequency.
8. To study the earth's magnetic field using a compass needle -bar magnet by plotting magnetic field lines and tangent galvanometer.



# JUST CHEMISTRY

## CHAPTER 6. HALOALKANES AND HALOARENES

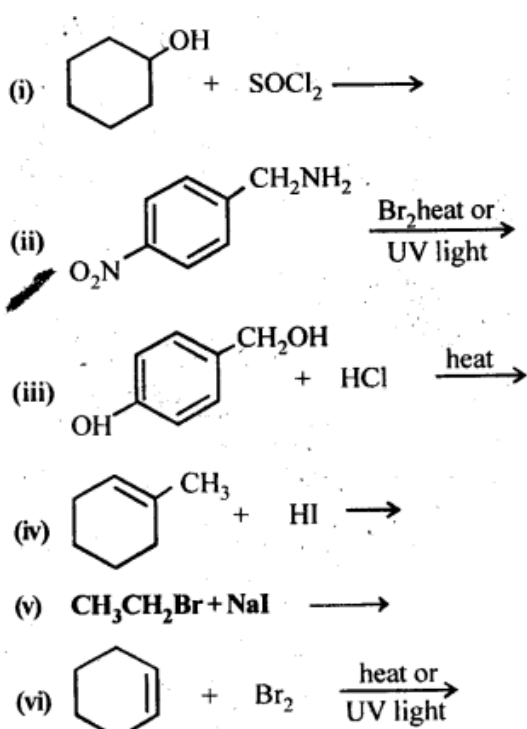
**Q.1. Name the following halides according to the IUPAC system and classify them as alkyl, allyl, benzyl (primary, secondary, tertiary), vinyl, or aryl halides:**

- (i)  $(\text{CH}_3)_2\text{CHCH}(\text{Cl})\text{CH}_3$
- (ii)  $\text{CH}_3\text{CH}_2\text{CH}(\text{CH}_3)\text{CH}(\text{C}_2\text{H}_5)\text{Cl}$
- (iii)  $\text{CH}_3\text{CH}_2\text{C}(\text{CH}_3)_2\text{CH}_2\text{I}$
- (iv)  $(\text{CH}_3)_3\text{CCH}_2\text{CH}(\text{Br})\text{C}_6\text{H}_5$
- (v)  $\text{CH}_3\text{CH}(\text{CH}_3)\text{CH}(\text{Br})\text{CH}_3$
- (vi)  $\text{CH}_3\text{C}(\text{C}_2\text{H}_5)_2\text{CH}_2\text{Br}$
- (vii)  $\text{CH}_3\text{C}(\text{Cl})(\text{C}_2\text{H}_5)\text{CH}_2\text{CH}_3$
- (viii)  $\text{CH}_3\text{CH}=\text{C}(\text{Cl})\text{CH}_2\text{CH}(\text{CH}_3)_2$
- (ix)  $\text{CH}_3\text{CH}=\text{CHC}(\text{Br})(\text{CH}_3)_2$
- (x) P- $\text{ClC}_6\text{H}_4\text{CH}_2\text{CH}(\text{CH}_3)_2$

**Q. 2. Write structures of the following compounds:**

- (i) 2-Chloro-3-methylpentane
- (ii) 1-Chloro-4- ethylcyclohexane
- (iii) 4-tert. Butyl-3-iodoheptane
- (iv) 1,4-Dibromobut-2-ene
- (v) 1-Bromo-4-sec. butyl-2-methylbenzene.

**Q. 3. Draw the structures of major monohalo products in each of the following reactions:**





Q. 4. Arrange each set of compounds in order of increasing boiling points :

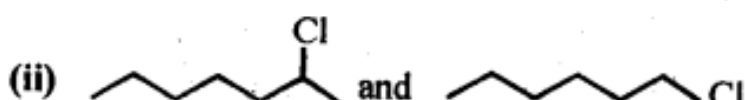
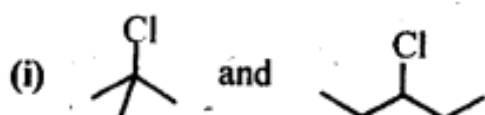
(i) Bromomethane, bromoform, chloromethane, dibromomethane

(ii) 1-Chloropropane, isopropylchloride, 1-chlorobutane.

Q. 5. p-dichlorobenzene has higher m.p. and lesser solubility than those of o- and m-isomers.

Discuss.

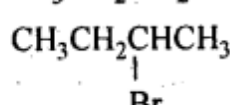
Q. 6. In the following pairs of halogen compounds, which compound undergoes faster  $\text{SN}^1$  reaction?

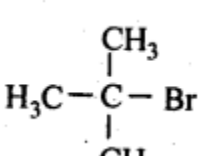


Q. 7. Which alkyl halide from the following pairs would you expect to react more rapidly by an  $\text{SN}^{2?}$

mechanism? Explain your answer.

(i)  $\text{CH}_3\text{CH}_2\text{CH}_2\text{CH}_2\text{Br}$  or



(ii)  $\text{CH}_3\text{CH}_2\text{CH}(\text{Br})\text{CH}_3$  or 

(iii)  $\text{CH}_3\text{CH}(\text{CH}_3)\text{CH}_2\text{CH}_2\text{Br}$  or  $\text{CH}_3\text{CH}_2\text{CH}(\text{CH}_3)\text{CH}_2\text{Br}$

Q. 8. How will you bring about the following conversions?

(i) Ethanol to but-1-yne.

(ii) Ethane to bromoethene

(iii) Propene to 1-nitropropane

(iv) Toluene to benzyl alcohol

(v) Propene to propyne

(vi) Ethanol to ethyl fluoride

(vii) Bromomethane to propanone

(viii) But-1-ene to but-2-ene

(ix) 1-Chlorobutane to n-octane

(x) Benzene to biphenyl

Q. 9. What happens when:

(i) n-butyl chloride is treated with alcoholic KOH.

(ii) Bromobenzene is treated with Mg in the presence of dry ether.

(iii) Chlorobenzene is subjected to hydrolysis.

(iv) Ethyl chloride is treated with aqueous KOH.

(v) Methyl bromide is treated with sodium in the presence of dry ether,

(vi) Methyl chloride is treated with KCN.

**Q. 10. The treatment of alkyl chlorides with aqueous KOH leads to the formation of alcohols but in presence of alcoholic KOH, alkenes are major products. Explain.**

---

**Marking Scheme –**

**Initiative, cooperativeness and participation – 05 marks**

**Content accuracy and research work – 05 marks**

**Creativity, originality – 05 marks**

**Viva for content assimilation - 05 marks**

---

**Total marks = 20**



**Read the chapters and answer the following questions:**

- **A Letter to God**
- **Dust of snow**
- **Fire and Ice**
- **Answer the following questions (20-30 words):**
  - I. Who was Lencho? What were his main problems?
  - II. How would you describe Lencho?
  - III. What did Lencho compare the raindrop to and why?
  - IV. Why was Lencho's soul filled with sorrow?
  - V. Why did Lencho not want the money to be sent through mail?
  - VI. What does the poet Robert Frost want to convey through the poem 'Dust of Snow'?
  - VII. How has the poet observed 'nature' in the poem 'Dust of Snow'?
  - VIII. Why does the poet use such poetically uncommon bird and tree? What does it reflect?

❖ **Answer the following questions (100-120 words)**

- I. Give a character-sketch of Lencho
- II. How did the hailstones affect Lencho's field? What was Lencho's only hope?
- III. Small things in life make significant changes in our life. Elaborate with reference to the poem 'Dust of Snow'.
- IV. Discuss how extreme behaviour can hasten the end of the world with respect to 'Fire and Ice'.
- V. Our attitude towards a situation evokes both negative and positive response. Analyze this with reference to the poem, 'Dust of Snow' to bring out the inherent valuable lessons.

**Note: 1. complete your book back exercises of these chapters.**

**2. Do Holiday home work in English copy itself.**

## **PROJECT WORK**

### **INSTRUCTIONS**

1. The holiday homework will be considered as your English project work.
2. Project will be done in a file (plastic strip file with A4 sheet may also be used)

3. Students have to pick **one** topic from the prescribed book **prose /poetry. The topics selected should be different from one another.**
4. YOUR PAGE SEQUENCE SHOULD BE:

\*page 1: Write -English Project Work, name, class, roll number, topic etc.

### **Example**

#### **ENGLISH PROJECT WORK**

**NAME:**

**CLASS:**

**ROLL NUMBER:**

**TOPIC : (from prose / poetry)**

### **\*CERTIFICATE**

This is to certify that I..... of class.....  
have done the English project work on my own. It is my own original  
work as per the guidelines provided by the teacher.

Signature

Name:

### **\* ACKNOWLEDGEMENT**

I would like to express my special thanks of gratitude to my teacher  
(Name of the teacher) who gave me the golden opportunity to do this  
wonderful project on the topic (Write the topic name). I offer my sincere  
appreciation for the learning opportunities provided by my principal  
(Name of the principal) which also helped me in doing a lot of research  
and I came to know about so many new things. I am really thankful to  
them. I would also like to thank my parents and friends who helped me a  
lot in finalizing this project within the limited time frame

**ROLL NUMBER:**

**NAME:**

### **\*INDEX**

**\* INTRODUCTION**

**\* ABOUT AUTHOR**

**\* ABOUT THE CHARACTERS**

**\* SUMMARY**

**\* THEME**

**\* CONCLUSION**

**\* BIBLIOGRAPHY**

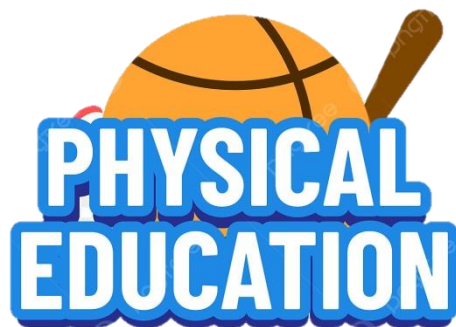
**\*USE PICTURES AND CAPTIONS WHEREVER POSSIBLE**

# ART INTEGRATED PROJECT WORK

**You have to prepare a project on your neighbouring state Kerala**

- ❖ Front page
- ❖ Acknowledgement
- ❖ Certificate
- ❖ Index
- ❖ Introduction : About neighbouring state
- ❖ Culture of Kerala – food , dress, tradition dance , people etc.
- ❖ language
- ❖ Religions of Kerala
- ❖ Marriage culture
- ❖ Festivals
- ❖ Art and craft
- ❖ Scholars
- ❖ National Award winners
- ❖ Bibliography
- ❖ Use attractive photographs wherever required.

**NOTE: THE PROJECT WORK MUST BE HANDWRITTEN. ONLY COVERPAGE CAN BE THE PRINT OUT WITH SCHOOL LOGO.**



# PHYSICAL EDUCATION

**CHAPTER 1. (MANAGEMENT OF SPORTING EVENTS)**

**CHAPTER 2. (CHILDREN AND WOMEN IN SPORTS)**

**(Short answer type question)**

**Q. 1. Write the basic functioning of sports event management?**

**Q. 2. Give any two objectives of intramural?**

**Q.3. Explain about Lordosis?**

**Q 4. Define:-**

**1) Knock Knee**

**2) Flat Foot**

**Q. 5. Explain pre, during and post responsibilities of various committees.**

**(Long answer type question)**

**Q 6. Write a short note on benefits of participation in sports?**

**Q 7. Compare any three committees that work together to organize sports Event with its responsibilities?**

**Q 8. Define directing and controlling.**

**Q 9. What do you mean by community sports?**

**Q 10. Define extramural? What is the significance of extramural?**



## MYSQL

It is freely available open source Relational Database Management System (RDBMS) that uses **Structured Query Language(SQL)**. In MySQL database , information is stored in Tables. A single MySQL database can contain many tables at once and store thousands of individual records.

### SQL (Structured Query Language)

SQL is a language that enables you to create and operate on relational databases, which are sets of related information stored in tables.

### DIFFERENT DATA MODELS

A **data model** refers to a set of concepts to describe the structure of a database, and certain constraints (restrictions) that the database should obey. The four data model that are used for database management are :

1. **Relational data model** : In this data model, the data is organized into tables (i.e. rows and columns). These tables are called relations.
2. **Hierarchical data model**
3. **Network data model**
4. **Object Oriented data model**

### RELATIONAL MODEL TERMINOLOGY

1. **Relation** : A table storing logically related data is called a Relation.
2. **Tuple** : A row of a relation is generally referred to as a tuple.
3. **Attribute** : A column of a relation is generally referred to as an attribute.
4. **Degree** : This refers to the **number of attributes** in a relation.
5. **Cardinality** : This refers to the **number of tuples** in a relation.
6. **Primary Key** : This refers to a set of one or more attributes that can uniquely identify tuples within the relation.
7. **Candidate Key** : All attribute combinations inside a relation that can serve as primary key are candidate keys as these are candidates for primary key position.
8. **Alternate Key** : A candidate key that is not primary key, is called an alternate key.
9. **Foreign Key** : A non-key attribute, whose values are derived from the primary key of some other table, is known as foreign key in its current table.

### REFERENTIAL INTEGRITY

- A referential integrity is a system of rules that a DBMS uses to ensure that relationships between records in related tables are valid, and that users don't accidentally delete or change related data. This integrity is ensured by foreign key.

### CLASSIFICATION OF SQL STATEMENTS

SQL commands can be mainly divided into following categories:

#### **1. Data Definition Language(DDL) Commands**

Commands that allow you to perform task, related to data definition e.g;

- Creating, altering and dropping.
- Granting and revoking privileges and roles.
- Maintenance commands.

## 2. Data Manipulation Language(DML) Commands

Commands that allow you to perform data manipulation e.g., retrieval, insertion, deletion and modification of data stored in a database.

## 3. Transaction Control Language(TCL) Commands

Commands that allow you to manage and control the transactions e.g.,

- Making changes to database, permanent
- Undoing changes to database, permanent
- Creating savepoints
- Setting properties for current transactions.

## MYSQL ELEMENTS

1. Literals

2. Datatypes

3. Nulls

4. Comments

### LITERALS

It refers to a fixed data value. This fixed data value may be of character type or numeric type. For example, 'replay', 'Raj', '8', '306' are all character literals.

**Numbers not enclosed in quotation marks are numeric literals.** E.g. 22, 18, 1997 are all numeric literals.

Numeric literals can either be integer literals i.e., without any decimal or be real literals i.e. with a decimal point e.g. 17 is an integer literal but 17.0 and 17.5 are real literals.

### DATA TYPES

Data types are means to identify the type of data and associated operations for handling it. MySQL data types are divided into three categories:

- Numeric
- Date and time
- String types

#### Numeric Data Type

1. int – used for number without decimal.
2. Decimal(m,d) – used for floating/real numbers. m denotes the total length of number and d is number of decimal digits.

#### Date and Time Data Type

1. date – used to store date in YYYY-MM-DD format.
2. time – used to store time in HH:MM:SS format.

#### String Data Types

1. char(m) – used to store a fixed length string. m denotes max. number of characters.
2. varchar(m) – used to store a variable length string. m denotes max. no. of characters.

#### DIFFERENCE BETWEEN CHAR AND VARCHAR DATA TYPE

S.NO.	Char Datatype	Varchar Datatype
1.	It specifies a <b>fixed length</b> character String.	It specifies a <b>variable length</b> character string.
2.	When a column is given datatype as CHAR(n), then MySQL ensures that all values stored in that column have this length i.e. n bytes. If a value is shorter than this length n then blanks are added, but the size of value remains n bytes.	When a column is given datatype as VARCHAR(n), then the maximum size a value in this column can have is n bytes. Each value that is stored in this column store exactly as you specify it i.e. no blanks are added if the length is shorter than maximum length n.



## NULL VALUE

If a column in a row has no value, then column is said to be **null** , or to contain a null. **You should use a null value** when the actual value is not known or when a value would not be meaningful.

## DATABASE COMMNADS

### 1. VIEW EXISTING DATABASE

To view existing database names, the command is : **SHOW DATABASES ;**

### 2. CREATING DATABASE IN MYSQL

For creating the database in MySQL, we write the following command : **CREATE DATABASE <databasename> ;**

e.g. In order to create a database Student, command is :

**CREATE DATABASE Student ;**

### 3. ACCESSING DATABASE

For accessing already existing database , we write :

**USE <databasename> ;**

e.g. to access a database named Student , we write command as :

**USE Student ;**

### 4. DELETING DATABASE

For deleting any existing database , the command is :

**DROP DATABASE <databasename> ;**

e.g. to delete a database , say student, we write command as ; **DROP DATABASE Student ;**

### 5. VIEWING TABLE IN DATABASE

In order to view tables present in currently accessed database , command is : **SHOW TABLES ;**

## CREATING TABLES IN MYSQL

- Tables are created with the CREATE TABLE command. When a table is created, its columns are named, data types and sizes are supplied for each column.

**Syntax of CREATE TABLE command**

is : **CREATE TABLE <table-name>**

( <column name> <data type> ,

<column name> <data type> ,

..... ) ;

E.g. in order to create table EMPLOYEE given below :

ECODE	ENAME	GENDER	GRADE	GROSS
-------	-------	--------	-------	-------

We write the following command :

**CREATE TABLE employee**

( ECODE integer ,

ENAME varchar(20) ,

GENDER char(1) ,

GRADE char(2) ,

GROSS integer );

## INSERTING DATA INTO TABLE

- The rows are added to relations(table) using INSERT command of SQL. Syntax of INSERT is : **INSERT INTO <tablename> [<column list>] VALUE ( <value1> , <value2> , ..... ) ;**

e.g. to enter a row into EMPLOYEE table (created above), we write command as :

```
INSERT INTO employee  
VALUES(1001, 'Ravi', 'M', 'E4', 50000);
```

OR

```
INSERT INTO employee (ECODE , ENAME , GENDER , GRADE , GROSS)  
VALUES(1001, 'Ravi', 'M', 'E4', 50000);
```

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000

In order to insert another row in EMPLOYEE table , we write again INSERT command :

```
INSERT INTO employee  
VALUES(1002 , 'Akash' , 'M' , 'A1' , 35000);
```

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000
1002	Akash	M	A1	35000

### INSERTING NULL VALUES

- To insert value NULL in a specific column, we can type NULL without quotes and NULL will be inserted in that column. E.g. in order to insert NULL value in ENAME column of above table, we write INSERT command as :

```
INSERT INTO EMPLOYEE  
VALUES (1004 , NULL , 'M' , 'B2' , 38965 ) ;
```

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000
1002	Akash	M	A1	35000
1004	NULL	M	B2	38965

### SIMPLE QUERY USING SELECT COMMAND

- The SELECT command is used to pull information from a table. Syntax of SELECT command is : SELECT <column name>,<column name>  
FROM <tablename>  
WHERE <condition name> ;

### SELECTING ALL DATA

- In order to retrieve everything (all columns) from a table, SELECT command is used as : SELECT \* FROM <tablename> ;

e.g.

In order to retrieve everything from Employee table, we write SELECT command as :

#### EMPLOYEE

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000
1002	Akash	M	A1	35000
1004	NULL	M	B2	38965

```
SELECT * FROM Employee ;
```

## SELECTING PARTICULAR COLUMNS

### EMPLOYEE

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000
1002	Akash	M	A1	35000
1004	Neela	F	B2	38965
1005	Sunny	M	A2	30000
1006	Ruby	F	A1	45000
1009	Neema	F	A2	52000

- A particular column from a table can be selected by specifying column-names with SELECT command. E.g. in above table, if we want to select ECODE and ENAME column, then command is :

```
SELECT ECODE , ENAME  
FROM EMPLOYEE ;
```

- E.g.2 in order to select only ENAME, GRADE and GROSS column, the command is :

```
SELECT ENAME , GRADE ,  
GROSS FROM EMPLOYEE ;
```

## SELECTING PARTICULAR ROWS

We can select particular rows from a table by specifying a condition through **WHERE** clause along with SELECT statement. E.g. In employee table if we want to select rows where Gender is female, then command is :

```
SELECT * FROM EMPLOYEE  
WHERE GENDER = 'F' ;
```

- E.g.2. in order to select rows where salary is greater than 48000, then command is :

```
SELECT * FROM EMPLOYEE  
WHERE GROSS > 48000 ;
```

## ELIMINATING REDUNDANT DATA

The **DISTINCT** keyword eliminates duplicate rows from the results of a SELECT statement. For example ,

```
SELECT GENDER FROM EMPLOYEE ;
```

GENDER
M
M
F
M
F
F

```
SELECT DISTINCT(GENDER) FROM EMPLOYEE ;
```

DISTINCT(GENDER)
M
F

## VIEWING STRUCTURE OF A TABLE

- If we want to know the structure of a table, we can use DESCRIBE or DESC command, as per following syntax :

```
DESCRIBE | DESC <tablename> ;
```

- e.g. to view the structure of table **EMPLOYEE**, command is :      **DESCRIBE EMPLOYEE ; OR DESC EMPLOYEE ;**

## USING COLUMN ALIASES

- The columns that we select in a query can be given a different name, i.e. column alias name for output purpose.

### Syntax :

```
SELECT <columnname> AS column alias , <columnname> AS column alias ....  
FROM <tablename> ;
```

e.g. In output, suppose we want to display ECODE column as EMPLOYEE\_CODE in output , then command is :

```
SELECT ECODE AS "EMPLOYEE_CODE"  
FROM EMPLOYEE ;
```

### CONDITION BASED ON A RANGE

- The **BETWEEN** operator defines a range of values that the column values must fall in to make the condition true. The range include both lower value and upper value.

e.g. to display ECODE, ENAME and GRADE of those employees whose salary is between 40000 and 50000, command is:

```
SELECT ECODE , ENAME ,GRADE  
FROM EMPLOYEE  
WHERE GROSS BETWEEN 40000 AND 50000 ;
```

Output will be :

ECODE	ENAME	GRADE
1001	Ravi	E4
1006	Ruby	A1

### CONDITION BASED ON A LIST

- To specify a list of values, IN operator is used. The IN operator selects value that match any value in a given list of values. E.g.

```
SELECT * FROM EMPLOYEE  
WHERE GRADE IN ('A1' , 'A2');
```

Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1002	Akash	M	A1	35000
1006	Ruby	F	A1	45000
1005	Sunny	M	A2	30000
1009	Neema	F	A2	52000

- The **NOT IN** operator finds rows that do not match in the list. E.g.

```
SELECT * FROM EMPLOYEE  
WHERE GRADE NOT IN ('A1' , 'A2');
```

Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000
1004	Neela	F	B2	38965

### CONDITION BASED ON PATTERN MATCHES

- LIKE operator is used for pattern matching in SQL. Patterns are described using two special wildcard characters:

1. percent(**%**) – The % character matches any substring.
2. underscore(**\_**) – The \_ character matches any character.

e.g. to display names of employee whose name starts with R in EMPLOYEE table, the command is :

```

SELECT ENAME
FROM EMPLOYEE
WHERE ENAME LIKE 'R%';

```

Output will be :

ENAME
Ravi
Ruby

e.g. to display details of employee whose second character in name is 'e'.

```

SELECT *
FROM EMPLOYEE
WHERE ENAME LIKE '_e%';

```

Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1004	Neela	F	B2	38965
1009	Neema	F	A2	52000

e.g. to display details of employee whose name ends with 'y'.

```

SELECT *
FROM EMPLOYEE
WHERE ENAME LIKE '%y';

```

Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1005	Sunny	M	A2	30000
1006	Ruby	F	A1	45000

### SEARCHING FOR NULL

- The NULL value in a column can be searched for in a table using IS NULL in the WHERE clause. E.g. to list employee details whose salary contain NULL, we use the command :

```

SELECT *
FROM EMPLOYEE
WHERE GROSS IS NULL;

```

e.g.

#### STUDENT

Roll_No	Name	Marks
1	ARUN	NULL
2	RAVI	56
4	SANJAY	NULL

to display the names of those students whose marks is NULL, we use the command :

```

SELECT Name
FROM EMPLOYEE
WHERE Marks IS NULL;

```

Output will be :

Name
ARUN
SANJAY

## SORTING RESULTS

Whenever the SELECT query is executed, the resulting rows appear in a predecided order. The **ORDER BY** clause allow sorting of query result. The sorting can be done either in ascending or descending order, the default is ascending.

The **ORDER BY** clause is used as :

```
SELECT <column name> , <column name>....  
FROM <tablename>  
WHERE <condition>  
ORDER BY <column name> ;
```

e.g. to display the details of employees in EMPLOYEE table in alphabetical order, we use command :

```
SELECT *  
FROM EMPLOYEE  
ORDER BY ENAME ;
```

Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1002	Akash	M	A1	35000
1004	Neela	F	B2	38965
1009	Neema	F	A2	52000
1001	Ravi	M	E4	50000
1006	Ruby	F	A1	45000
1005	Sunny	M	A2	30000

e.g. display list of employee in descending alphabetical order whose salary is greater than 40000.

```
SELECT ENAME  
FROM EMPLOYEE  
WHERE GROSS > 40000  
ORDER BY ENAME desc ;
```

Output will be :

ENAME
Ravi
Ruby
Neema

## MODIFYING DATA IN TABLES

you can modify data in tables using UPDATE command of SQL. The UPDATE command specifies the rows to be changed using the WHERE clause, and the new data using the SET keyword. Syntax of update command is :

```
UPDATE <tablename>  
SET <columnname>=value , <columnname>=value  
WHERE <condition> ;
```

e.g. to change the salary of employee of those in EMPLOYEE table having employee code 1009 to 55000.

```
UPDATE EMPLOYEE  
SET GROSS = 55000  
WHERE ECODE = 1009 ;
```

## UPDATING MORE THAN ONE COLUMNS

e.g. to update the salary to 58000 and grade to B2 for those employee whose employee code is 1001.

```
UPDATE EMPLOYEE  
SET GROSS = 58000, GRADE='B2'  
WHERE ECODE = 1009 ;
```

## OTHER EXAMPLES

e.g.1. Increase the salary of each employee by 1000 in the EMPLOYEE table.

```
UPDATE EMPLOYEE  
SET GROSS = GROSS +100 ;
```

e.g.2. Double the salary of employees having grade as 'A1' or 'A2' .

```
UPDATE EMPLOYEE  
SET GROSS = GROSS * 2 ;  
WHERE GRADE='A1' OR GRADE='A2' ;
```

e.g.3. Change the grade to 'A2' for those employees whose employee code is 1004 and name is Neela.

```
UPDATE EMPLOYEE  
SET GRADE='A2'  
WHERE ECODE=1004 AND GRADE='NEELA' ;
```

## DELETING DATA FROM TABLES

To delete some data from tables, DELETE command is used. The DELETE command removes rows from a table. The syntax of DELETE command is :

```
DELETE FROM <tablename>  
WHERE <condition> ;
```

For example, to remove the details of those employee from EMPLOYEE table whose grade is A1.

```
DELETE FROM EMPLOYEE  
WHERE GRADE ='A1' ;
```

## TO DELETE ALL THE CONTENTS FROM A TABLE

```
DELETE FROM EMPLOYEE ;
```

So if we do not specify any condition with WHERE clause, then all the rows of the table will be deleted. Thus above line will delete all rows from employee table.

## DROPPING TABLES

The DROP TABLE command lets you drop a table from the database. The syntax of DROP TABLE command is :

```
DROP TABLE <tablename> ;
```

e.g. to drop a table employee, we need to write :

```
DROP TABLE employee ;
```

Once this command is given, the table name is no longer recognized and no more commands can be given on that table. After this command is executed, all the data in the table along with table structure will be deleted.

S.NO.	DELETE COMMAND	DROP TABLE COMMAND
1	It is a DML command.	It is a DDL Command.
2	This command is used to delete only rows of data from a table	This command is used to delete all the data of the table along with the structure of the table. The table is no longer recognized when this command gets executed.
3	Syntax of DELETE command is: DELETE FROM <tablename> WHERE <condition> ;	Syntax of DROP command is : DROP TABLE <tablename> ;

## ALTER TABLE COMMAND

The ALTER TABLE command is used to change definitions of existing tables.(adding columns,deleting columns etc.). The ALTER TABLE command is used for :

1. adding columns to a table

2. Modifying column-definitions of a table.
3. Deleting columns of a table.
4. Adding constraints to table.
5. Enabling/Disabling constraints.

### ADDING COLUMNS TO TABLE

To add a column to a table, ALTER TABLE command can be used as per following syntax:

```
ALTER TABLE <tablename>
ADD <Column name> <datatype> <constraint>;
```

e.g. to add a new column ADDRESS to the EMPLOYEE table, we can write command as :

```
ALTER TABLE EMPLOYEE
ADD ADDRESS VARCHAR(50);
```

A new column by the name ADDRESS will be added to the table, where each row will contain NULL value for the new column.

ECODE	ENAME	GENDER	GRADE	GROSS	ADDRESS
1001	Ravi	M	E4	50000	NULL
1002	Akash	M	A1	35000	NULL
1004	Neela	F	B2	38965	NULL
1005	Sunny	M	A2	30000	NULL
1006	Ruby	F	A1	45000	NULL
1009	Neema	F	A2	52000	NULL

However if you specify NOT NULL constraint while adding a new column, MySQL adds the new column with the default value of that datatype e.g. for INT type it will add 0 , for CHAR types, it will add a space, and so on.

e.g. Given a table namely Testt with the following data in it.

Col1	Col2
1	A
2	G

Now following commands are given for the table. Predict the table contents after each of the following statements:

- (i) ALTER TABLE testt ADD col3 INT ;
- (ii) ALTER TABLE testt ADD col4 INT NOT NULL ;
- (iii) ALTER TABLE testt ADD col5 CHAR(3) NOT NULL ;
- (iv) ALTER TABLE testt ADD col6 VARCHAR(3);

### MODIFYING COLUMNS

Column name and data type of column can be changed as per following syntax :

```
ALTER TABLE <table name>
CHANGE <old column name> <new column name> <new datatype>;
```

If Only data type of column need to be changed, then

```
ALTER TABLE <table name>
MODIFY <column name> <new datatype>;
```



**e.g.1.** In table EMPLOYEE, change the column GROSS to SALARY.

```
ALTER TABLE EMPLOYEE
CHANGE GROSS SALARY INTEGER;
```

**e.g.2.** In table EMPLOYEE , change the column ENAME to EM\_NAME and data type from VARCHAR(20) to VARCHAR(30).

```
ALTER TABLE EMPLOYEE
CHANGE ENAME EM_NAME VARCHAR(30);
```

**e.g.3.** In table EMPLOYEE , change the datatype of GRADE column from CHAR(2) to VARCHAR(2).

```
ALTER TABLE EMPLOYEE
MODIFY GRADE VARCHAR(2);
```

### DELETING COLUMNS

To delete a column from a table, the ALTER TABLE command takes the following form :

```
ALTER TABLE <table name>
DROP <column name>;
```

e.g. to delete column GRADE from table EMPLOYEE, we will write :

```
ALTER TABLE EMPLOYEE
DROP GRADE ;
```

### ADDING/REMOVING CONSTRAINTS TO A TABLE

ALTER TABLE statement can be used to add constraints to your existing table by using it in following manner:

❖ TO ADD PRIMARY KEY CONSTRAINT  
ALTER TABLE <table name>  
ADD PRIMARY KEY (Column name);

e.g. to add PRIMARY KEY constraint on column ECODE of table EMPLOYEE , the command is :

```
ALTER TABLE EMPLOYEE
ADD PRIMARY KEY (ECODE) ;
```

❖ TO ADD FOREIGN KEY CONSTRAINT  
ALTER TABLE <table name>  
ADD FOREIGN KEY (Column name) REFERENCES Parent Table (Primary key of Parent Table);

### REMOVING CONSTRAINTS

- To remove primary key constraint from a table, we use ALTER TABLE command as : ALTER TABLE <table name>  
DROP PRIMARY KEY ;
- To remove foreign key constraint from a table, we use ALTER TABLE command as : ALTER TABLE <table name>  
DROP FOREIGN KEY ;

### ENABLING/DISABLING CONSTRAINTS

Only foreign key can be disabled/enabled in MySQL.

To disable foreign keys : SET FOREIGN\_KEY\_CHECKS = 0 ;

To enable foreign keys : SET FOREIGN\_KEY\_CHECKS = 1 ;

## INTEGRITY CONSTRAINTS/CONSTRAINTS

- A constraint is a condition or check applicable on a field(column) or set of fields(columns).
- Common types of constraints include :

S.No.	Constraints	Description
1	NOT NULL	Ensures that a column cannot have NULL value
2	DEFAULT	Provides a default value for a column when none is specified
3	UNIQUE	Ensures that all values in a column are different
4	CHECK	Makes sure that all values in a column satisfy certain criteria
5	PRIMARY KEY	Used to uniquely identify a row in the table
6	FOREIGN KEY	Used to ensure referential integrity of the data

## NOT NULL CONSTRAINT

By default, a column can hold NULL. If you do not want to allow NULL value in a column, then NOT NULL constraint must be applied on that column. E.g.

```
CREATE TABLE Customer
(
  SID integer NOT NULL ,
  Last_Name varchar(30) NOT NULL ,
  First_Name varchar(30) );
```

Columns **SID** and **Last\_Name** cannot include NULL, while **First\_Name** can include NULL.

An attempt to execute the following SQL statement,

```
INSERT INTO Customer
VALUES (NULL , 'Kumar' , 'Ajay');
```

will result in an error because this will lead to column SID being NULL, which violates the NOT NULL constraint on that column.

## DEFAULT CONSTRAINT

The DEFAULT constraint provides a default value to a column when the INSERT INTO statement does not provide a specific value. E.g.

```
CREATE TABLE Student
( Student_ID integer ,
  Name varchar(30) ,
  Score integer DEFAULT 80);
```

When following SQL statement is executed on table created above:

```
INSERT INTO Student
VALUES (10 , 'Ravi' );
```

no value has been provided for score field.

Then table Student looks like the following:

Student_ID	Name	Score
10	Ravi	80

score field has got the default value

## UNIQUE CONSTRAINT

- The UNIQUE constraint ensures that all values in a column are distinct. In other words, no two rows can hold the same value for a column with UNIQUE constraint.

e.g.

```
CREATE TABLE Customer
(   SID integer Unique ,
    Last_Name varchar(30) ,
    First_Name varchar(30) );
```

Column SID has a unique constraint, and hence cannot include duplicate values. So, if the table already contains the following rows :

SID	Last_Name	First_Name
1	Kumar	Ravi
2	Sharma	Ajay
3	Devi	Raj

The executing the following SQL statement,

```
INSERT INTO Customer
VALUES ('3', 'Cyrus', 'Grace');
```

will result in an error because the value 3 already exist in the SID column, thus trying to insert another row with that value violates the UNIQUE constraint.

### CHECK CONSTRAINT

- The CHECK constraint ensures that all values in a column satisfy certain conditions. Once defined, the table will only insert a new row or update an existing row if the new value satisfies the CHECK constraint.

e.g.

```
CREATE TABLE Customer
(   SID integer CHECK (SID > 0),
    Last_Name varchar(30) ,
    First_Name varchar(30) );
```

So, attempting to execute the following statement :

```
INSERT INTO Customer
VALUES (-2, 'Kapoor', 'Raj');
```

will result in an error because the values for SID must be greater than 0.

### PRIMARY KEY CONSTRAINT

- A primary key is used to identify each row in a table. A primary key can consist of one or more fields(column) on a table. When multiple fields are used as a primary key, they are called a **composite key**.
- You can define a primary key in CREATE TABLE command through keywords PRIMARY KEY. e.g.

```
CREATE TABLE Customer
(   SID integer NOT NULL PRIMARY KEY,
    Last_Name varchar(30) ,
    First_Name varchar(30) );
```

Or

```
CREATE TABLE Customer
(
  SID integer,
  Last_Name varchar(30) ,
  First_Name varchar(30),
  PRIMARY KEY (SID) );
```

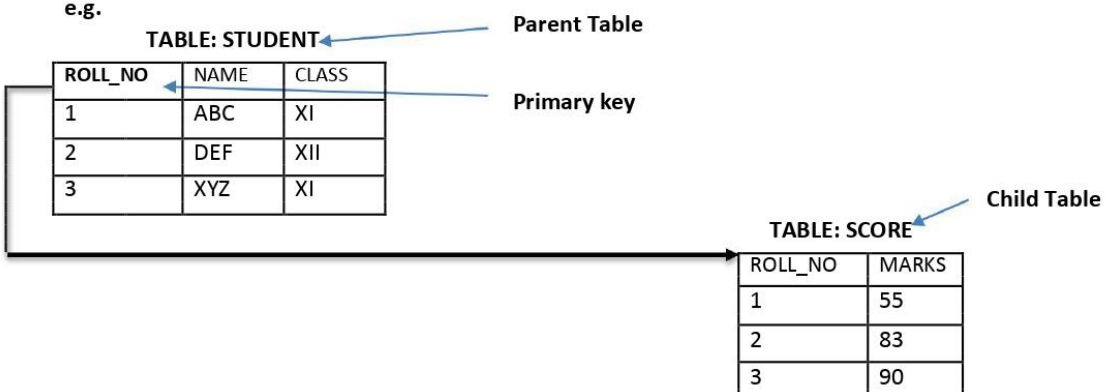
- The latter way is useful if you want to specify a composite primary key, e.g.

```
CREATE TABLE Customer
(
  Branch integer NOT NULL,
  SID integer NOT NULL ,
  Last_Name varchar(30) ,
  First_Name varchar(30),
  PRIMARY KEY (Branch , SID) );
```

### FOREIGN KEY CONSTRAINT

- Foreign key is a non key column of a table (**child table**) that draws its values from **primary key** of another table(**parent table**).
- The table in which a foreign key is defined is called a **referencing table or child table**. A table to which a foreign key points is called **referenced table or parent table**.

e.g.



Here column Roll\_No is a foreign key in table SCORE(Child Table) and it is drawing its values from Primary key (ROLL\_NO) of STUDENT table.(Parent Key).

```
CREATE TABLE STUDENT
(
  ROLL_NO integer NOT NULL PRIMARY KEY ,
  NAME VARCHAR(30) ,
  CLASS VARCHAR(3) );
```

```
CREATE TABLE SCORE
(
  ROLL_NO integer ,
  MARKS integer ,
  FOREIGN KEY(ROLL_NO) REFERENCES STUDENT(ROLL_NO) );
```

*\* Foreign key is always defined in the child table.*

### Syntax for using foreign key

FOREIGN KEY(column name) REFERENCES Parent\_Table(PK of Parent Table);

### REFERENCING ACTIONS

Referencing action with ON DELETE clause determines what to do in case of a DELETE occurs in the parent table.

Referencing action with ON UPDATE clause determines what to do in case of a UPDATE occurs in the parent table.

### Actions:

1. **CASCADE** : This action states that if a DELETE or UPDATE operation affects a row from the parent table, then automatically delete or update the matching rows in the child table i.e., cascade the action to child table.
2. **SET NULL** : This action states that if a DELETE or UPDATE operation affects a row from the parent table, then set the foreign key column in the child table to NULL.
3. **NO ACTION** : Any attempt for DELETE or UPDATE in parent table is not allowed.
4. **RESTRICT** : This action rejects the DELETE or UPDATE operation for the parent table.

### Q: Create two tables

Customer(customer\_id, name)

Customer\_sales(transaction\_id, amount , **customer\_id**)

Underlined columns indicate primary keys and bold column names indicate foreign key.

Make sure that no action should take place in case of a DELETE or UPDATE in the parent table.

**Sol :** CREATE TABLE Customer (  
customer\_id int Not Null Primary Key ,  
name varchar(30) );

```
CREATE TABLE Customer_sales (  
transaction_id Not Null Primary Key ,  
amount int ,  
customer_id int ,  
FOREIGN KEY(customer_id) REFERENCES Customer (customer_id)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION );
```

### Q: Distinguish between a Primary Key and a Unique key in a table.

S.NO.	PRIMARY KEY	UNIQUE KEY
1.	Column having Primary key can't contain NULL value	Column having Unique Key can contain NULL value
2.	There can be only one primary key in Table.	Many columns can be defined as Unique key

### Q: Distinguish between ALTER Command and UPDATE command of SQL.

S.NO.	ALTER COMMAND	UPDATE COMMAND
1.	It is a DDL Command	It is a DML command
2.	It is used to change the definition of existing table, i.e. adding column, deleting column, etc.	It is used to modify the data values present in the rows of the table.
3.	Syntax for adding column in a table: ALTER TABLE <tablename> ADD <Column name><Datatype> ;	Syntax for using UPDATE command: UPDATE <Tablename> SET <Columnname>=value WHERE <Condition> ;

## AGGREGATE / GROUP FUNCTIONS

Aggregate / Group functions work upon groups of rows , rather than on single row, and return one single output. Different aggregate functions are : COUNT ( ) , AVG ( ) , MIN ( ) , MAX ( ) , SUM ( )

Table : EMPL

EMPNO	ENAME	JOB	SAL	DEPTNO
8369	SMITH	CLERK	2985	10
8499	ANYA	SALESMAN	9870	20
8566	AMIR	SALESMAN	8760	30
8698	BINA	MANAGER	5643	20
8912	SUR	NULL	3000	10

### 1. AVG()

This function computes the average of given data. e.g. SELECT AVG(SAL)

FROM EMPL ;

Output

AVG(SAL)
6051.6

### 2. COUNT()

This function counts the number of rows in a given column.

If you specify the COLUMN name in parenthesis of function, then this function returns rows where COLUMN is not null.

If you specify the asterisk (\*), this function returns all rows, including duplicates and nulls.

e.g. SELECT COUNT(\*)

FROM EMPL ;

Output

COUNT(*)
5

e.g.2 SELECT COUNT(JOB)

FROM EMPL ;

Output

COUNT(JOB)
4

### 3. MAX()

This function returns the maximum value from a given column or expression.

e.g. SELECT MAX(SAL)

FROM EMPL ;

Output

MAX(SAL)
9870

#### 4. MIN()

This function returns the minimum value from a given column or expression.

e.g. SELECT MIN(SAL)  
FROM EMPL ;

Output

MIN(SAL)
2985

#### 5. SUM()

This function returns the sum of values in given column or expression.

e.g. SELECT SUM(SAL)  
FROM EMPL ;

Output

SUM(SAL)
30258

#### GROUPING RESULT – GROUP BY

The GROUP BY clause combines all those records(row) that have identical values in a particular field(column) or a group of fields(columns).

GROUPING can be done by a column name, or with aggregate functions in which case the aggregate produces a value for each group.

Table : EMPL

EMPNO	ENAME	JOB	SAL	DEPTNO
8369	SMITH	CLERK	2985	10
8499	ANYA	SALESMAN	9870	20
8566	AMIR	SALESMAN	8760	30
8698	BINA	MANAGER	5643	20

e.g. Calculate the number of employees in each grade.

```
SELECT JOB, COUNT(*)  
FROM EMPL  
GROUP BY JOB ;
```

Output

JOB	COUNT(*)
CLERK	1
SALESMAN	2
MANAGER	1

e.g.2. Calculate the sum of salary for each department.

```
SELECT DEPTNO , SUM(SAL)  
FROM EMPL  
GROUP BY DEPTNO ;
```

Output

DEPTNO	SUM(SAL)
10	2985
20	15513
30	8760

e.g.3. find the average salary of each department.

Sol:

**\*\* One thing that you should keep in mind is that while grouping , you should include only those values in the SELECT list that either have the same value for a group or contain a group(aggregate) function. Like in e.g. 2 given above, DEPTNO column has one(same) value for a group and the other expression SUM(SAL) contains a group function.**

### NESTED GROUP

- To create a group within a group i.e., nested group, you need to specify multiple fields in the GROUP BY expression. e.g. To group records **job wise** within **Deptno wise**, you need to issue a query statement like :

```
SELECT DEPTNO , JOB , COUNT(EMPNO)
FROM EMPL
GROUP BY DEPTNO , JOB ;
```

Output

DEPTNO	JOB	COUNT(EMPNO)
10	CLERK	1
20	SALESMAN	1
20	MANAGER	1
30	SALESMAN	1

### PLACING CONDITION ON GROUPS – HAVING CLAUSE

- The **HAVING** clause places conditions on groups in contrast to WHERE clause that places condition on individual rows. While **WHERE** conditions cannot include aggregate functions, **HAVING** conditions can do so.
- e.g. To display the jobs where the number of employees is less than 2,

```
SELECT JOB, COUNT(*)
FROM EMPL
GROUP BY JOB
HAVING COUNT(*) < 2 ;
```

Output

JOB	COUNT(*)
CLERK	1
MANAGER	1

### DATABASE TRANSACTIONS

#### TRANSACTION

A Transaction is a logical unit of work that must succeed or fail in its entirety. This statement means that a transaction may involve many sub steps, which should either all be carried out successfully or all be ignored if some failure occurs. A Transaction is an atomic operation which may not be divided into smaller operations.

#### Example of a Transaction

Begin transaction

```
Get balance from account X
Calculate new balance as X – 1000
Store new balance into database file
Get balance from account Y
Calculate new balance as Y + 1000
Store new balance into database file
```

End transaction



## TRANSACTION PROPERTIES (ACID PROPERTIES)

1. **ATOMICITY**(All or None Concept) – This property ensures that either all operations of the transaction are carried out or none are.
2. **CONSISTENCY** – This property implies that if the database was in a consistent state before the start of transaction execution, then upon termination of transaction, the database will also be in a consistent state.
3. **ISOLATION** – This property implies that each transaction is unaware of other transactions executing concurrently in the system.
4. **DURABILITY** – This property of a transaction ensures that after the successful completion of a transaction, the changes made by it to the database persist, even if there are system failures.

## TRANSACTION CONTROL COMMANDS (TCL)

The TCL of MySQL consists of following commands :

1. **BEGIN** or **START TRANSACTION** – marks the beginning of a transaction.
2. **COMMIT** – Ends the current transaction by saving database changes and starts a new transaction.
3. **ROLLBACK** – Ends the current transaction by discarding database changes and starts a new transaction.
4. **SAVEPOINT** – Define breakpoints for the transaction to allow partial rollbacks.
5. **SET AUTOCOMMIT** – Enables or disables the default auto commit mode