UNIT 3

CHAPTER 13: DATABASE FUNDAMENTALS - MYSQL REVISION TOUR

Database: Collection of logically related data stored in a structure format.

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

RDBMS: A DBMS used to manage Relational Databases is called an RDBMS (Relational Data Base Management System). Some popular RDBMS software available are: Oracle, MySQL, Sybase, and Ingress.

Benefits of using a DBMS are:

a. Redundancy can be controlled
b. Inconsistency can be avoided
c. Data can be shared
d. Security restrictions can be applied.

MySQL: It is an Open Source RDBMS Software. It is available free of cost.

Relation/Table: A table refers to a two dimensional representation of data arranged in columns (also called fields or attributes) and rows (also called records or tuples).

Key: A column or a combination of columns which have some specific characteristics in a relation e.g. are Primary Key, Candidate Key and Foreign Key etc.

Primary Key: The group of one or more attribute(s) used to uniquely identify each row/tuple of a relation/table is called its Primary Key.

Candidate Key: A group of columns which can be set as the primary key of a relation is called a candidate key because it is one of the candidates available to be the primary key of the relation.

Alternate Key: A candidate key of a table which is not set as primary key is called its Alternate Key.

Degree is the number of columns/attributes in the table.

Cardinality is the number of rows/tuples in a table.

SQL (Structured Query Language): It is the language used to manipulate and manage databases and tables within them using an RDBMS. There are following four types of SQL commands:

1. **DDL (Data Definition Language):** Deals with the Structure (create, remove, or modify) of databases and tables e.g. CREATE, DROP, ALTER.
2. **DML (Data Manipulation Language):** Used to manipulate data/ values within tables e.g. INSERT, UPDATE, DELETE.
3. **DCL (Data Control Language):** Used to control the access to the databases and tables e.g. GRANT, REVOKE.
4. **TCL(Transaction Control Language):** used to manage and control the transaction e.g. COMMIT , ROLLBACK, SAVEPOINT

Some Commonly used DDL Command are as follows:-

SNo	Command, Syntax and Purpose
1	Command : Create Database Syntax: Create database <Database Name>; Purpose: Creates a database with specified name
2	Command : Create Table Syntax: Create Table <Table Name> (<Column Name1>, Data Type1,<Colomn Name2>, Data Type2); Purpose: Creates a table with specified name

3	Command : Alter Table Syntax: Alter Table <Table Name> Add <Column Name> Data Type1; Alter Table <Table Name> Drop <Column Name>; Alter Table <Table Name> Modify <Column Name> <new_defintion>; Purpose: Modify the strcture of a table
4	Command : Use Syntax: Use <Database Name>; Purpose: Open the specified database for use
5	Command : Select Database( ) Syntax: : Select Database( ); Purpose: Show the name of current Database
6	Command : Show tables; Syntax: : Show tables ; Purpose: Show a list of tables in the current database
7	Command : Show databases; Syntax: : Show databases; Purpose: Show a list of databases
8	Command : Insert Syntax : Insert Into <Table Name> (<Column Name1>,<Column Name2>….,values (<value1, value 2….); Purpose: Insert Data into the table
9	Command : Select Syntax: Select * Column name, Expression,Column name From table Name Where Condition Order by Column Name Asc/Desc ; Purpose: To reterive selected data from the table
10	Command : Describe Syntax: : Desc <Table Name> ; Purpose: Show structure of table
11	Command : Update Syntax: : Update <table name> Set <Column name >=Value Where <condition>; Purpose: Update or Modify the data in tables

Following are the clauses which can be used with select command

SNo	CLAUSE	EXPLANATION
1	DISTINCT	Used to display distinct values from a column of a table
2	WHERE	Used to specify the condition based on which rows of a table are dispalyed
3	BETWEEN	Used to define the range of values within which the column values must fall to make a condition true. It include both upper and lower values.
4	IN	Used to select values that natch any values in a lsit of specified values
5	LIKE	Used for pattern matching of string data using wildcard characters % and _
6	IS NULL/ NOT	Used to select rows in which the specified column is NULL (or is NOT NULL)

	NULL	
7	ORDER BY	Used to display the selected rows in ascending or desending order of the specified column

String Functions:		
SN	Name & syntax	Description
1	LENGTH()	Returns the length of a column or a string in bytes
2	CONCAT(Str1, Str2)	Returns a string that result from concatenating the arguments.
3	INSTR(Str, Substr)	Returns the position of the first occurance of substring (substr) in the string(str)
4	Lower(str) or LCASE(Str)	Returns the argument <str> in lowercase
5	Upper(str) UCASE(Str)	Returns the argument <str> in uppercase
6	LEFT(Str, n)	Returns the first n character from the string
7	RIGHT(Str,n)	Returns the last n character from the string

Numeric Functions:		
SN	Name & syntax	Description
1	POWER(X,Y) or POW(X,Y)	Returns the values raised to the power of y
2	ROUND(X)	Rounds the argument X to the nearest Integer
3	ROUND(x,d)	Rounds the argument X to d decimal places
4	Truncate(x,d)	Truncate the argunment x to d decimal places

Date and Time Functions:		
SN	Name & syntax	Description
1	CURDATE()	Returns the current date in YYYY-MM-DD format YYYYMMDD format, depending on whether the function is used in a string or numeric context
2	NOW ()	Returns the current date and time in YYYY-MM-DD HH:MM:SS
3	SYSDATE()	Returns the current date and time in YYYYMMDD HHMMSS.uuuuuuu format
4	DATE(expr)	Extracts the date part of date or datetime <exp>
5	MONTH(date)	Returns the numeric month from the specified date in the range 0 to 12. it returns 0 for dates such as '0000-00-00' or '2010-00-00' that have a zero month part
6	YEAR(date)	Returns the year for specified date in the range 0 to 9999. it returns 0 for the 'Zero' date. Returns values like 1998, 2016 etc.
7	DAYNAME(date)	Returns the name of the weekday for specified date
8	DAYOFMONTH(date)	Returns the day of month in the range 0-31
9	DAYOFWEEK(date)	Returns the day of week in number as 1 for Sunday, 2 for Monday and so on
10	DAYOFYEAR(date)	Returns the day of the year for given date in numeric format in the range 1 to 366.

CALCULATOR FUNCTIONALITY USING SELECT STATEMENT

```
SELECT 5+68;
SELECT 5+68 FROM DUAL;
SELECT ECODE, SAL_AMT*12 FROM SALARY;
```

TO KNOW CURRENT DATE

```
SELECT CURDATE();
```

DISPLAY DATA WITH OTHER HEADING

```
SELECT SAL_AMT*12 AS "ANNUAL SALARY" FROM SALARY
SELECT 22/7 AS PI;
```

HANDLING NULLS

```
SELECT NAME, BIRTH, DEATH FROM ABC;
SELECT NAME, BIRTH, IFNULL(DEATH, "ALIVE") FROM ABC;
```

PUTTING TEXT IN SELECT QUERY

```
SELECT EMPNAME, 'GETS THE SALARY PER MONTH' SAL_AMT FROM SALARY;
```

USE OF RELATIONAL OPERATORS: = , > , < , >= , <= , <>

```
SELECT * FROM EMPLOYEE WHERE ECODE<>2001;
```

USE OF LOGICAL OPERATORS: OR / || , AND / && , NOT / !

```
SELECT * FROM EMPLOYEE WHERE ECODE=2001 OR EMPNAME='RAVI KUMAR';
SELECT * FROM EMPLOYEE WHERE ECODE=2001 AND EMPNAME='RAVI KUMAR';
SELECT * FROM EMPLOYEE WHERE (NOT ECODE=2001);
```

BETWEEN

```
SELECT ECODE, EMPNAME FROM SALARY WHERE SAL_AMT BETWEEN 20000 AND 50000;
SELECT ECODE, EMPNAME FROM SALARY WHERE SAL_AMT NOT BETWEEN 20000 AND 50000;
```

IN

SELECT * FROM EMPLOYEE WHERE CITY IN ('DELHI','MUMBAI','BANGALORE');

LIKE

SELECT * FROM EMPLOYEE WHERE EMPNAME LIKE 'A%'; -- Starts with A.

SELECT * FROM EMPLOYEE WHERE EMPNAME LIKE '----'; -- Exact four characters.

SELECT * FROM EMPLOYEE WHERE EMPNAME LIKE '---%'; -- At least three characters.

NULL

SELECT * FROM EMPLOYEE WHERE EMPNAME IS NULL;

ORDER BY

SELECT * FROM EMPLOYEE ORDER BY EMPNAME;

SELECT * FROM SALARY WHERE SAL_AMT >50000 ORDER BY ECODE DESC;

USING ALIAS

SELECT ECODE SAL_AMT*12 "ANNUAL SALARY" FROM SALARY ORDER BY "ANNUAL SALARY"

STRING FUNCTION

SELECT CHAR(65); -- To display the character against ASCII CODE 65.

SELECT CONCAT(ECODE, EMPNAME) AS "CODENAME" FROM EMPLOYEE; -- To merge two columns.

SELECT RTRIM(EMPNAME) FROM EMPLOYEE; -- Removes space from right side of empname.

SELECT LTRIM(EMPNAME) FROM EMPLOYEE; -- Removes space from left side of empname.

SELECT TRIM(EMPNAME) FROM EMPLOYEE; -- Removes space from both sides of empname.

NUMERIC FUNCTION

SELECT ECODE, MOD(15,6) FROM DUAL; -- display remainder 3.

SELECT MOD(SAL_AMT,1000) FROM SALARY;

SELECT POWER(5,3) FROM DUAL; -- Calculate 5 raise to power 3. i.e.125.

SELECT ROUND(15.193, 1) FROM DUAL; --- Display 15.2 (Round off)

DATE/TIME FUNCTION

CURDATE()/CURRENT)DATE()

SELECT CURDATE(); -- Display system date.

SELECT DATE('2010-04-13 01:02:33'); -- Display 2010-04-13 (date only).

SELECT MONTH('2010-04-13'); --- Display month only i.e. 04.

SELECT YEAR('2010-04-13'); --- Display year only i.e. 2010.

PRACTICE EXERCISES

Consider a database LOANS with the following table:

Table: LoanAccounts

AccNo	CustName	LoanAmount	Installment	Intrate	StartDate	Interest
1	R K Gupta	300000	36	12.00	19-07-2009	
2	S P Sharma	500000	48	10.00	22-03-2008	
3	K P Jain	300000	36	NULL	08-03-2007	
4	M P Yadav	800000	60	10.00	06-12-2008	
5	S P Sinha	200000	36	12.50	03-01-2010	
6	P Sharma	700000	60	12.50	05-06-2008	
7	K S Dhall	500000	48	NULL	05-03-2008	

Write SQL commands for the tasks 1 to 35 and write the output for the SQL commands 36 to 40:

Create Database and use it

1. Create the database LOANS.
2. Use the database LOANS.

Create Table / Insert Into

3. Create the table LoanAccounts and insert tuples in it.

Simple Select

4. Display the details of all the loans.
5. Display the AccNo, CustName, and LoanAmount of all the loans.

Conditional Select using Where Clause

6. Display the details of all the loans with less than 40 instalments.
7. Display the AccNo and LoanAmount of all the loans started before 01-04-2009.
8. Display the IntRate of all the loans started after 01-04-2009.

Using NULL

9. Display the details of all the loans whose rate of interest is NULL.
10. Display the details of all the loans whose rate of interest is not NULL.

Using DISTINCT Clause

11. Display the amounts of various loans from the table LoanAccounts. A loan amount should appear only once.
12. Display the number of instalments of various loans from the table LoanAccounts. An instalment should appear only once.

Using Logical Operators (NOT, AND, OR)

13. Display the details of all the loans started after 31-12-2008 for which the number of instalments are more than 36.
14. Display the CustName and LoanAmount for all the loans which do not have number of instalments 36.
15. Display the CustName and LoanAmount for all the loans for which the loan amount is less than 500000 or intrate is more than 12.
16. Display the details of all the loans which started in the year 2009.
17. Display the details of all the loans whose LoanAmount is in the range 400000 to 500000.
18. Display the details of all the loans whose rate of interest is in the range 11% to 12%.

Using IN Operator

19. Display the CustName and LoanAmount for all the loans for which the number of instalments are 24, 36, or 48.

Using BETWEEN Operator

20. Display the details of all the loans whose LoanAmount is in the range 400000 to 500000.
21. Display the details of all the loans whose rate of interest is in the range 11% to 12%.

Using LIKE Operator

22. Display the AccNo, CustName, and LoanAmount for all the loans for which the CustName ends with 'Sharma'.
23. Display the AccNo, CustName, and LoanAmount for all the loans for which the Cust_Name ends with 'a'.
24. Display the AccNo, CustName, and LoanAmount for all the loans for which the

Cust_Name contains 'a'

25. Display the AccNo, CustName, and LoanAmount for all the loans for which the Cust_Name does not contain 'P'.
26. Display the AccNo, CustName, and LoanAmount for all the loans for which the CustName contains 'a' as the second last character.

Using ORDER BY clause

27. Display the details of all the loans in the ascending order of their LoanAmount.
28. Display the details of all the loans in the descending order of their StartDate.
29. Display the details of all the loans in the ascending order of their LoanAmount and within LoanAmount in the descending order of their StartDate.

Using UPDATE, DELETE, ALTER TABLE

30. Change the interest rate 11.50% for all the loans for which interest rate is NULL.
31. Increase the interest rate by 0.5% for all the loans for which the loan amount is more than 400000.
32. For each loan replace Interest with (LoanAmount*IntRate*Instalments) 12*100.
33. Delete the records of all the loans whose start date is before 2007.
34. Delete the records of all the loans of 'K.P. Jain'
35. Add another column Category of type CHAR(1) in the Loan table.

SOLVED CBSE QUESTIONS

1. Mrs. Sharma is the class teacher of Class 'XII A' She wants to create a table 'Student' to store details of her class.

(i) Which of the following can be the attributes of Student table?

a) RollNo b) "Amit" c) Name d) 25

(ii) Name the Primary key of the table 'Student'. State reason for choosing it.

Ans. i. a) RollNo b) Name

ii. Primary Key: RollNo as it will be unique for each student of the class.

2. While creating the table Employee, Mr. John forgot to include the field EMPNO, now how to insert the EMPNO field with integer data type and 10 size into the Employee table?

Ans: Alter Table Employee
Add (EMPNO int(10));

3. While creating the table Student last week, Ms. Sharma forgot to include the column GamePlayed. Now write a command to insert the Gameplayed column with VARCHAR data type and 30 size into the Student table?

Ans. Alter Table Student Add (GamePlayed VARCHAR(30));

4. Sujata has created a table in MySQL. Later on she found that the width of name column is not sufficient for entering some long names. She wants to increase the width of the name column. Which command she should give to do this.

Ans: Alter Table.

5. While creating a table "MobDet", Kavita forgot to set primary key for the table. Write the statement to set the column MobileNo as the primary key of the table.

Ans.: Alter Table MobDet add primary key MobileNo;

6. Write a command to add a NOT NULL constraint on fees column of a student table.

Ans.: Alter table student MODIFY fees int(4) NOT NULL;