

Ex.No. 1

Date:

DDL Commands – CREATE, ALTER, DROP

Aim:

To Create, Alter and Drop the table using Data Definition Language**SYNTAX:**

CREATE TABLE

```
CREATE TABLE table_name  
(  
column_name1 data_type,  
column_name2 data_type,  
column_name3 data_type,  
....  
);
```

ALTER A TABLE

To add a column in a table
ALTER TABLE table_name
ADD column_namedatatype;

To delete a column in a table
ALTER TABLE table_name
DROP COLUMN column_name;

DROP TABLE

DROP TABLE table_name;

TRUNCATE TABLE

TRUNCATE TABLE table_name;

```
mysql> CREATE Table FACULTY(FacNo VARCHAR(4), FacName VARCHAR(15),  
Gender CHAR(1), DOB DATE, DOJ DATE, MobileNo int(10));
```

```
mysql> CREATE Table FACULTY(FacNo VARCHAR(4), FacName VARCHAR(15),  
Gender CHAR(1), DOB DATE, DOJ DATE, MobileNo int(10));
```

Questions:

1) Create a table name STUDENT with following structure.

```
mysql> create table student(regno int(3),name varchar(15),gender char(1),dob date,mobileno  
int(1),city varchar(15));
```

Field	Type	Null	Key	Default	Extra
FacNo	varchar(4)	YES		NULL	
FacName	varchar(15)	YES		NULL	
Gender	char(1)	YES		NULL	
DOB	date	YES		NULL	
DOJ	date	YES		NULL	
MobileNo	int	YES		NULL	

2) Create a table name FACULTY with following structure

```
mysql> CREATE Table FACULTY(FacNo VARCHAR(4), FacName VARCHAR(15), Gender  
CHAR(1), DOB DATE, DOJ DATE, MobileNo int(10));
```

Field	Type	Null	Key	Default	Extra
FacNo	varchar(4)	YES		NULL	
FacName	varchar(15)	YES		NULL	
Gender	char(1)	YES		NULL	
DOB	date	YES		NULL	
DOJ	date	YES		NULL	
MobileNo	int	YES		NULL	

3) Create a table name DEPARTMENT with following structure.

```
mysql> CREATE Table DEPARTMENT(DeptNo VARCHAR(4), DeptName VARCHAR(15),  
DeptHead VARCHAR(4));
```

Field	Type	Null	Key	Default	Extra
DeptNo	varchar(4)	YES		NULL	
DeptName	varchar(15)	YES		NULL	
DeptHead	varchar(4)	YES		NULL	

4) Create a table name COURSE with following structure.

```
mysql> CREATE Table COURSE(CourseNo VARCHAR(3), CourseDesc VARCHAR(14),  
CourseType CHAR(1), SemNo CHAR(1), HallNo VARCHAR(4), FacNo VARCHAR(4));
```

Field	Type	Null	Key	Default	Extra
CourseNo	varchar(3)	YES		NULL	
CourseDesc	varchar(14)	YES		NULL	
CourseType	char(1)	YES		NULL	
SemNo	char(1)	YES		NULL	
HallNo	varchar(4)	YES		NULL	
FacNo	varchar(4)	YES		NULL	

5) Modify the table FACULTY by adding a column name DeptNo of datatype VARCHAR(4)

```
) mysql> Alter Table FACULTY add DeptNo VARCHAR(4);
```

Field	Type	Null	Key	Default	Extra
FacNo	varchar(4)	YES		NULL	
FacName	varchar(15)	YES		NULL	
Gender	char(1)	YES		NULL	
DOB	date	YES		NULL	
DOJ	date	YES		NULL	
MobileNo	int	YES		NULL	
DeptNo	varchar(4)	YES		NULL	

Ex.No. 2

Date:

DDL Commands with Constraints – PRIMARY, FOREIGN KEY, UNIQUE, CHECK

AIM:

To add the constraints like primary key, foreign key, unique key and check using DDL commands.

PRIMARY:

```
ALTER TABLE table_name  
ADD PRIMARY KEY(primary_key_column);
```

FOREIGN KEY:

```
ALTER TABLE table_name  
ADD CONSTRAINT constraint_name  
FOREIGN KEY foreign_key_name (columns)  
REFERENCES parent_table(columns)  
ON DELETE action  
ON UPDATE action
```

UNIQUE:

```
CREATE TABLE table_1(  
...  
column_name_1 data_type,  
...  
UNIQUE(column_name_1)  
);
```

CHECK

```
CREATE TABLE IF NOT EXISTS parts (  
part_no VARCHAR(18) PRIMARY KEY,  
description VARCHAR(40),
```

cost DECIMAL(10 , 2) NOT NULL CHECK(cost > 0), price
 DECIMAL (10,2) NOT NULL
);

Questions:

1) Alter the table STUDENT with following structure.

	Column #Name	Constraints
		PRIMARY
1	RegNo	KEY
2	MobileNo	NOT NULL

Field	Type	Null	Key	Default	Extra
RegNo	int	NO	PRI	NULL	
Name	varchar(15)	YES		NULL	
Gender	char(1)	YES		NULL	
DOB	date	YES		NULL	
MobileNo	int	NO		NULL	
City	varchar(15)	YES		NULL	

2) Alter the table name FACULTY with following structure. The DeptNo in this table refers the DeptNo in the DEPARTMENT table.

	Column # Name	Constraints
1	FacNo	PRIMARY KEY
2	Gender	CHECK 'M' or 'F'

Field	Type	Null	Key	Default	Extra
FacNo	varchar(4)	NO	PRI	NULL	
FacName	varchar(15)	YES		NULL	
Gender	char(1)	YES		NULL	
DOB	date	YES		NULL	
DOJ	date	YES		NULL	
MobileNo	int	YES		NULL	
DeptNO	varchar(4)	YES		NULL	

3) After the FACULTY table is successfully created, test if you can add a constraint FOREIGN KEY to the DeptNo of this table.

Field	Type	Null	Key	Default	Extra
FacNo	varchar(4)	NO	PRI	NULL	
FacName	varchar(15)	YES		NULL	
Gender	char(1)	YES		NULL	
DOB	date	YES		NULL	
DOJ	date	YES		NULL	
MobileNo	int	YES		NULL	
DeptNO	varchar(4)	YES		NULL	

4) Alter the table name DEPARTMENT with following structure.

#	Column Name	Constraint
1	DeptNo	PRIMARY KEY

Field	Type	Null	Key	Default	Extra
DeptNo	varchar(4)	NO	PRI	NULL	
DeptName	varchar(15)	YES		NULL	
DeptHead	varchar(4)	YES		NULL	

5) Alter the table name COURSE with following structure.

#	Column Name	Constraint
1	CourseNo	PRIMARY KEY
2	SemNo	1 to 6

Field	Type	Null	Key	Default	Extra
CourseNo	varchar(3)	NO	PRI	NULL	
CourseDesc	varchar(14)	YES		NULL	
CourseType	char(1)	YES		NULL	
SemNo	char(1)	YES		NULL	
HallNo	varchar(4)	YES		NULL	
FacNo	varchar(4)	YES		NULL	

Ex.No. : 3

Date:

DML Commands – INSERT, SELECT, UPDATE, DELETE

Aim:

To perform Data Manipulation Language (DML) Commands such as INSERT, SELECT, UPDATE, DELETE in the table.

INSERT:

INSERT INTO table_name

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

(or)

INSERT INTO table_name (column1, column2, column3,...)

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

UPDATE:

UPDATE table_name

SET column1=value, column2=value2,...

WHERE some_column=some_value;

DELETE:

DELETE FROM table_name
WHERE some_column=some_value;

SELECT:

SELECT column_name(s)
FROM table_name;

Questions:

1. Populate all the five tables with your own data.
-

```
mysql> SELECT * FROM STUDENT;
```

RegNo	Name	Gender	DOB	MobileNo	City
1922211123	RAMU	M	2004-12-15	987654329	TIRUPATI
1922211125	ROSY	F	2004-08-24	987654323	NELLOR
1922211156	SITA	F	2004-02-14	876543297	KADAPA
1922211174	SUNNY	M	2004-12-16	765432123	GUNTOOR
1922211198	SRINIVAS	M	2004-06-17	986534256	VIZAG

2. Update the value of student name whose register number is '191711342'

```
mysql> SELECT * FROM FACULTY;
```

facno	facname	gender	DOB	DOJ	mobno	dep
1	kiran	M	2001-06-10	2006-03-07	2777777777	CODING
2	mahi	M	2001-03-01	2006-06-07	5555555557	CODING
3	mounika	F	2000-05-11	2013-03-04	553456789	CODING
4	keerthi	F	1987-08-10	2012-02-03	129863453	CODING
5	arun	M	1988-12-23	2015-09-13	123863453	CODING

3. Delete the record in the table FACULTY, who resigned her job.
4. Modify the date of birth for the faculty whose name is 'RAM' with a value '1983-05-01'.
5. Remove all faculty who are having over 65 years
6. View all the records from the five tables. Exercise

3)

```
mysql> delete from faculty where resigned="yes";
Query OK, 3 rows affected (0.02 sec)
```

```
mysql> select * from faculty;
```

facno	facname	gender	DOB	DOJ	mobno	deptno	resigned
3	mounika	F	2000-05-11	2013-03-04	553456789	C003	no
5	arun	M	1988-12-23	2015-09-13	123863453	C005	no

4)

```
mysql> update faculty set DOB="1983-05-01" where facname="arun";
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from faculty;
```

facno	facname	gender	DOB	DOJ	mobno	deptno	resigned
3	mounika	F	2000-05-11	2013-03-04	553456789	C003	no
5	arun	M	1983-05-01	2015-09-13	123863453	C005	no

5)

```
mysql> delete from faculty where age>65;
Query OK, 1 row affected (0.04 sec)
```

```
mysql> select * from faculty;
```

facno	facname	gender	DOB	DOJ	mobno	deptno	resigned
5	arun	M	1983-05-01	2015-09-13	123863453	C005	no

Ex. No.: 4

Date:

SELECT with various clause – WHERE, pattern matching

AIM:

To view the records from the tables using SELECT commands with WHERE Clause and Pattern matching.

SELECT:

```
SELECT
column_1, column_2, ...
FROM
table_1
[INNER | LEFT | RIGHT] JOIN table_2 ON conditions
WHERE
conditions
GROUP BY column_1
HAVING group_conditions
ORDER BY column_1
LIMIT offset, length;
```

The SELECT statement consists of several clauses as explained in the following list:

- SELECT followed by a list of comma-separated columns or an asterisk (*) to indicate that you want to return all columns.
 - FROM specifies the table or view where you want to query the data.
 - JOIN gets related data from other tables based on specific join conditions.
 - WHERE clause filters row in the result set.
 - GROUP BY clause groups a set of rows into groups and applies aggregate functions on each group.
 - HAVING clause filters group based on groups defined by GROUP BY clause.
 - ORDER BY clause specifies a list of columns for sorting.
 - LIMIT constrains the number of returned rows.
-

Questions:

WHERE:

1. The student counsellor wanted to display the registration number, student name and date of birth for all the students.
2. The controller of examinations wanted to list all the female students
3. Who are the boy students registered for course with the course number “C001”
4. Display all faculty details joined before “November 2014”
5. Display all the courses not allotted to halls

LIKE:

6. List the students whose name ends with the substring “ma”
7. Display all students whose name contains the substring “ma”
8. Find all the students who are located in cities having “Sal” as substring
9. Display the students whose names do not contain six letters.

10. Find all the students whose names contains "th"

OUTPUTS:

1)

```
mysql> select RegNo,Name,DOB from student;
```

RegNo	Name	DOB
1922211123	RAKESH	2004-12-15
1922211125	ROSY	2004-08-24
1922211156	SITA	2004-02-14
1922211174	SUNNY	2004-12-16
1922211198	SRINIVAS	2004-06-17

2)

```
mysql> select * from student where gender = "F";
```

RegNo	Name	Gender	DOB	MobileNo	City
1922211125	ROSY	F	2004-08-24	987654323	NELLORE
1922211156	SITA	F	2004-02-14	876543297	KADAPUR

3)

```
mysql> select * from student where courseno="C001";
```

RegNo	Name	Gender	DOB	MobileNo	City	courseno
1922211123	RAKESH	M	2004-12-15	987654329	TIRUPATHI	C001
1922211125	ROSY	F	2004-08-24	987654323	NELLORE	C001
1922211198	SRINIVAS	M	2004-06-17	986534256	VIZAG	C001

4)

```
mysql> select * from FACULTY course where DOJ<"2014-11-01";
```

facno	facname	gender	DOB	DOJ	mobno	deptno	resigned
7	surekha	F	1981-08-18	2013-12-15	123456543	C004	yes
8	rahu1	M	1987-05-19	2012-10-17	123456678	C005	yes

5)

```
mysql> delete from course where hallallot = "yes";
Query OK, 2 rows affected (0.02 sec)
```

```
mysql> select * from course;
```

CourseNo	CourseDesc	CourseType	SemNo	HallNo	FacNo	hallallot
1	DBMS	A	1	CO01	U001	no
2	PDSD	B	2	CO02	U002	no
3	BIO	C	3	CO03	U003	no

6)

```
mysql> select * from student where name like "%ma";
```

RegNo	Name	Gender	DOB	MobileNo	City	course
1922211123	PUMA	M	2004-12-15	987654329	TIRUPATHI	C001
1922211125	SUMA	F	2004-08-24	987654323	NELLORE	C001
1922211156	RAMA	F	2004-02-14	876543297	KADAPA	C002

7)

```
mysql> select * from student where name like "%ma%";
```

RegNo	Name	Gender	DOB	MobileNo	City	course
1922211123	PUMA	M	2004-12-15	987654329	TIRUPATHI	C001
1922211125	SUMA	F	2004-08-24	987654323	NELLORE	C001
1922211156	RAMA	F	2004-02-14	876543297	KADAPA	C002
1922211174	MAHESH	M	2004-12-16	765432123	GUNTOOR	C002
1922211198	HEMANTH	M	2004-06-17	986534256	VIZAG	C001

8)

```
mysql> SELECT * from student where City like "%SAL%";
```

RegNo	Name	Gender	DOB	MobileNo	City	course
1922211123	PUMA	M	2004-12-15	987654329	SALT	C001
1922211125	SUMA	F	2004-08-24	987654323	BASAL	C001

9)

```
mysql> SELECT * from student where Name not like "_____";
```

RegNo	Name	Gender	DOB	MobileNo	City	cours
1922211123	PUMA	M	2004-12-15	987654329	SALT	C001
1922211125	SUMA	F	2004-08-24	987654323	BASAL	C001
1922211156	RAMA	F	2004-02-14	876543297	KADAPA	C002
1922211198	HEMANTH	M	2004-06-17	986534256	VIZAG	C001

10)

```
mysql> SELECT * from student where Name like "%th%";
```

RegNo	Name	Gender	DOB	MobileNo	City	cours
1922211198	HEMANTH	M	2004-06-17	986534256	VIZAG	C001

Ex. No. : 5

Date:

SELECT with various clause – BETWEEN, IN, Aggregate function

AIM:

To view the records from the tables using SELECT commands with BETWEEN, IN, Aggregate functions.

BETWEEN operator:

```
SELECT
column1,column2,...
FROM
table_name
WHERE expr [NOT] BETWEEN begin_expr AND end_expr;
```

The *expr* is the expression to test in the range that is defined by *begin_expr* and *end_expr*.

IN operator:

```
SELECT
```

```
column1,column2,...
FROM
table_name
WHERE (expr|column_1) IN ('value1','value2',...);
```

Questions:

IN & BETWEEN

1. List the type of the courses “Statistics” and “Programming”
2. The instructor wants to know the CourseNos whose scores are in the range 50 to 80

AGGREGATE

1. Find the average mark of “C002”.
2. List the maximum, minimum mark for “C021”
3. List the maximum, minimum, average mark for each subject in 5th semester
4. List the name of the courses and average mark of each courses.
5. Calculate the sum of all the scores.
6. How many students are registered for each course? Display the course description and the number of students registered in each course.
7. How many courses did each student register for? Use Assessment table.

OUTPUTS:

1)

```
mysql> select avg(score) from student where curso = "C002";
+-----+
| avg(score) |
+-----+
|      81.5 |
+-----+
```

2)

```
mysql> select min(c021),max(c021) from student;
+-----+-----+
| min(c021) | max(c021) |
+-----+-----+
|      76   |      95   |
+-----+-----+
```

3)

```
mysql> select min(score),max(score),avg(score)from student where SemNo =
```

min(score)	max(score)	avg(score)
72	96	83.6

4)

5)

```
mysql> select sum(score) from student;
```

sum(score)
418

Ex. No.: 6

Date: 27/01/2023

SELECT with various clause – GROUP BY, HAVING, ORDER BY

GROUP BY - HAVING

1. How many students are registered for each course? Display the course description and the number of students registered in each course.
2. How many courses did each student register for? Use Assessment table.

ORDER BY

1. Retrieve Name, Gender, MobileNo of all the students in ascending order of RegNo.

```
mysql> select Name, Gender, MobileNo from student order by RegNo;
+-----+-----+-----+
| Name      | Gender | MobileNo |
+-----+-----+-----+
| Hisham    | M      | 923748923 |
| Shaman    | M      | 923748923 |
| Tina      | F      | 923768323 |
| Ragul     | M      | 923745423 |
| Rehaankh  | M      | 923748923 |
| Fathima   | F      | 923448923 |
| Ananya    | F      | 987654312 |
| Abinaya   | F      | 987654321 |
+-----+-----+-----+
8 rows in set (0.00 sec)
```

2. List the faculty members in the order of older faculty first.

```
mysql> select FacName, DOB from faculty1 order by DOB;
+-----+-----+
| FacName | DOB      |
+-----+-----+
| Ram     | 1983-05-01 |
| Bill    | 1992-11-23 |
| Mary    | 1996-03-02 |
+-----+-----+
3 rows in set (0.00 sec)
```

Ex. No.: 7

Date:

SubQuery& Correlated Query

Sub-Query and Correlated Sub-Query:

1. Which of the student's score is greater than the highest score?

```
mysql> select Name, Score from student where Score in(select Score from student where Score>50);
+-----+-----+
| Name      | Score |
+-----+-----+
| Tina      | 55    |
| Ragul     | 60    |
| Rehaankh  | 65    |
| Fathima   | 70    |
| Ananya    | 75    |
| Abinaya   | 80    |
+-----+-----+
6 rows in set (0.01 sec)
```


2. Which of the students' have written more than one assessment test?

```
mysql> select Name, TestCount from student where TestCount in(select TestCount from student where TestCount>1);
```

Name	TestCount
Hisham	2
Shaman	4
Fathima	2
Ananya	3

4 rows in set (0.00 sec)

3. Which faculty has joined recently and when?

```
mysql> select FacName, DOJ from faculty1 where DOJ in(select max(DOJ) from faculty1) order by DOJ desc;
```

FacName	DOJ
Ram	2020-12-12
Mary	2020-12-12

2 rows in set (0.00 sec)

Ex. No.: 8

Date:27/01/2023

Joins – EquiJoin, InnerJoin, OuterJoin

Questions:

1. List the departments where the faculty members are working.

```
mysql> select faculty1.FacNo, faculty1.FacName, department.DeptNo, department.DeptName from faculty1 cross join department order by FacNo;
```

FacNo	FacName	DeptNo	DeptName
1	Mary	102	AIDS
1	Mary	101	CSE
2	Ram	102	AIDS
2	Ram	101	CSE
3	Bill	102	AIDS
3	Bill	101	CSE

6 rows in set (0.00 sec)

2. Find the student who has no score in any of the courses. List student name and course number.

```
mysql> select student.Name, student.Score, course.CourseNo from student inner join course on student.CourseNo=course.CourseNo;
```

Name	Score	CourseNo
Hisham	0	C001
Shaman	0	C003
Tina	0	C003

3. The office clerk needs the names of the courses taken by the faculty belonging to 'ECE department' whose name is 'Kamal'

```
mysql> select faculty1.FacName, faculty1.Courses, department.DeptNo, department.DeptName from faculty1, department where
FacName="Kamal" and DeptName="ECE";
+-----+-----+-----+-----+
| FacName | Courses      | DeptNo | DeptName |
+-----+-----+-----+-----+
| Kamal   | Microprocessor | 103    | ECE      |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Ex. No.: 9

Date:27/01/2023

VIEW, INDEX, SEQUENCE

Questions:

Create a view with name 'std_view' using STUDENT table which holds the value of register number, name and DOB of student.

```
mysql> create view std_view as select RegNo, Name, DOB from student;
Query OK, 0 rows affected (0.01 sec)
mysql> select * from std_view;
+-----+-----+-----+
| RegNo  | Name      | DOB      |
+-----+-----+-----+
| 191123489 | Hisham    | 2004-09-23 |
| 191123789 | Shaman    | 2004-05-28 |
| 191124389 | Tina      | 2004-09-13 |
| 191128489 | Ragul     | 2004-09-14 |
| 191129489 | Rehaankh  | 2004-12-31 |
| 191154589 | Fathima   | 2005-02-13 |
| 191711342 | Ananya    | 2005-07-09 |
| 192211241 | Abinaya   | 2004-11-23 |
+-----+-----+-----+
8 rows in set (0.00 sec)
```