

Expl. DDL Commands - CREATE, ALTER, DROP

Aim - To create, Alter and Drop the table using Data definition L.

Description

Data definition language (DDL) ~~changes~~ are used to define ~~the~~ database structure or schema.

DDL - Create, Alter, Drop, rename,

Truncate.

Syntax

CREATE TABLE (Col-n, datatype);

ALTER TABLE add a column in a table;

DROP TABLE table-name;

TRUNCATE TABLE table-name;

Create a table name DEPARTMENT with following structure.

	Name	Description	datatype
1	Dept no	Department id	VARCHAR(10)
2	DeptName	Department Name	VARCHAR(50)
3	DeptHead	Department Head	VARCHAR(4)

Output

my sql > Create Table department (DeptNo int(10),

DeptName char(10), DeptHead char(10));

field	Type	Null	Key	RefDefault	Extra
DeptNo	int(10)	YES	NULL		
DeptName	char(10)	YES	NULL		
DeptHead	char(10)	YES	NULL		

RESULT:

Tables are created, altered and modified using DDL commands.

dim... To

Descriptio
n

PRIMARY

FOREIG

UNIQUE

Symbol

All

Check

Ad

For

Re

Or

Exp-2 PDL commands with Constraints

PRIMARY KEY, FOREIGN, UNIQUE, CHECK

AIM: To add the constraints like
Foreign, unique and check

Description:

PRIMARY KEY: uniquely identifies each record
in a database table

FOREIGN KEY: It is used to link two tables.

UNIQUE constraint: It ensures that all values are
~~different~~ in a column

CHECK constraint: To limit the value range that

can be placed in a column.

ALTER TABLE table-name

ADD CONSTRAINT constraint-name

FOREIGN KEY foreign-key-name

REFERENCES parent-table

ON DELETE action

ON UPDATE action

EXP

CREATE TABLE table-name (column-1, column-2, ..., column-n)

UNIQUE (Column-name-1)

Alter table name DEPARTMENT

#	Column Name	Constraint
1	Dept-No	PRIMARY KEY

Output

My sql> alter table department add pri key

Field	Type	Null	Key	Default	Extra
deptno	int(10)	No	pri	NULL	
deptname	char(10)	Yes		NULL	
depthead	char(10)	Yes		NULL	

RESULT :-

DDL commands with Primary, Foreign.

Unique, check constraints are updated and verified.

EXP-3 DML Commands - INSERT, SELECT,

UPDATE, DELETE

Dimension To perform Data manipulation language commands such as INSERT, SELECT, UPDATE, DELETE in table.

Description:

INSERT - insert data into a table.

UPDATE - updates existing data within a

DELETE - deletes all records from a table.

Select - retrieve data from a database.

Syntax

INSERT INTO table (Value1, Value2, ...)

UPDATE table

DELETE FROM table

SELECT column_name

FROM table;

and

Modify the date of birth for the faculty whose name is RAM with a value 1993-05-01

Fac No	Fac Name	gender	Pob	mobile no	Date
1	RAM	M	2000	9055252	2001

Result:

Not in DB. Job building - 1993-05-01

~~Writing~~
DML commands such as INSERT, SELECT, UPDATE, DELETE are performed.

SQL
ca
ec

Old job 2000

Select new 1993-05-01

Action after writing

Job 1993-05-01

EXP

EXP-4 SELECT with various clause -

WHERE, pattern matching

dim :- To view the records from the table using SELECT command with WHERE clause and pattern.

actually
(1983-85)

DESCRIPTION :-

SELECT :- allow you to get the data from tables.

SELECT

The result of select statement also called as result set that is a list of rows, each consisting of same number of columns.

SYNTAX :-

```
SELECT col_1, col_2  
FROM table_1
```

[INNER / LEFT / RIGHT] JOIN table_2 ON condition

WHERE

GROUP BY col_1

HAVING group-conditions

ORDER BY col_1

who are the boy students registered

for course with the course number CO1

Output

Mysql > select * from course;

CourseNo	CourseName	CourseType	Semester	Hall	Faculty
1	program	IT	2000	201	201

MySQL Database Help 10 Minutes 45

Result:

need to add some restrictions due

The records from the table ~~course~~

displayed using

~~SELECT command with~~

~~WHERE clause and pattern matching.~~

Additional notes (from 17/3/1999)

ANSWER

10) VB QUEST

Windows quest manually

10) VB ASPECT

EXP-5 SELECT with various clause - BETWEEN

(IN, Aggregate function)

Aim: To view the records from the tables
using ~~Select command~~ with BETWEEN,
IN, Aggregate functions

Description:-

The BETWEEN operators allows you to specify a range to test. We often use the BETWEEN operators in the WHERE clause of the SELECT, INSERT, UPDATE and DELETE statements.

Syntax

~~SELECT col1, col2...
from tablename~~

WHERE expr [NOT] BETWEEN begin_expr AND end_expr;

~~SELECT col1, col2...
from tablename~~

WHERE (expr1|cols) IN 'value1, 'value2'

How many courses did each student register for? use assessment table.

Output

mysql> select courseno, count(courseno), count(assessno)

from course where regno like '19111111'

courseno	coursedesc	Count (regno)
101001	Maths	6
101002	Science	4

Result

The records from the tables are displayed using select command's where clause and pattern matching. Note that

the table 'student' has been omitted.

dim

De
J
in
or

J

C

C

E

EXP-6 SELECT with various clauses

GROUP BY, HAVING, ORDER BY

dim :- To view the records from the table using SELECT commands with group by, having, order by

Description:-

The group By clause groups a set of rows into a set of rows into a set of summary rows by values of columns or expressions.

The group by clause returns one row

ORDER BY clause sort a result set by a single column or multiple columns.

Syntax:-

```
SELECT c1, c2, ..., cn  
      [aggregation function (c1)]  
  FROM TABLE
```

WHERE

where - conditions

GROUP BY c1, c2 ... cn

HAVING Conditions.

How many courses did each student register for? Use assessed table

dim
GROUP - BY student_id

Output
student_id count(score)

Mysql > select course_no, count(score) from course_group by course_no;

course_no	Count(score)
1	1

Result:-

student_id score
course_no count(score)

The records from the tables score displayed using SELECT command with

GROUP BY, HAVING and ORDER BY

dim

- design score

student_id

student_no course_no

student_no course_no score

student_no course_no score

EXP-7:- Subquery & correlated Query

Aim:- To perform subquery and correlated query on the given relation.

Description :-

Subquery - A MySQL subquery is a query nested within another query such as ~~SELECT, INSERT, UPDATE or DELETE~~.

Correlated query - A correlated subquery is a subquery that uses the data from the outer query.

Syntax :-

~~SELECT C₁, C₂, ..., C_n~~
~~WHERE C_i IN (SELECT C₁, C₂, ..., C_n) FROM WHERE~~
~~conditions;~~

~~SELECT * FROM table WHERE exists (subquery);~~

which of the students have written more than one assessment test

Output

```
mysql> select name from stud where assess
```

name
manu
navi
inni

Result :-

The records from the tables were displayed using sub-Query and correlated Sub - Query.

~~Sub-Query~~

dim :-

Descr

Join

Cross

INN

LEI

STU

EXP-8 Joins - Equijoin, InnerJoin, OuterJoin

OuterJoin

Item name

dim : To perform JOIN using Equijoin, InnerJoin

OuterJoin on the given relation

Description :- It is used to find common rows between two relations.

JOIN - supports cross join, inner join,

Left join, Right join.

CROSS JOIN - makes a cartesian product of rows from multiple tables.

INNER JOIN - compare first table each row in

second table to find pairs of rows.

LEFT JOIN - returns all rows in left table

Syntax

Cross join:

SELECT t₁.id, t₂.id FROM t₁

inner join :

SELECT t₁.id, t₂.id FROM t₁

left join : SELECT t₁.id, t₂.id FROM t₁

right join : SELECT t₁.id t₂.id FROM t₁

List the departments where the faculty members are working.

Output

mysql> select faculty, facno, facname, facname, depno, deptname, deptname from faculty
cross join department.

facno	facname	depno	deptname
802	Rahman	11	sales.

Result:-

The records from the tables are displayed using JOIN using Equi Join, InnerJoin, OuterJoin.

Exp-9: VIEW, INDEX, SEQUENCE

Aim: To create view, index and sequence on the given relation.

Description:-

VIEW - In MySQL, almost features of view conform to the SQL: 2003 standard.

INDEX - A database index, or just index, helps speed up the retrieval of data from tables.

SEQUENCE - In MySQL, a sequence is a list of integers generated in the ascending order i.e. 1, 2, 3, ...

Syntax

VIEW : CREATE ALGORITHM= {MERGE|TEMPTABLE} [view]
[database-name].[view-name] AS
(SELECT statement)

INDEX : CREATE (UNIQUE | FULLTEXT | SPATIAL) index-name;

SEQUENCE : CREATE TABLE table-name

 COLUMNNAME1 AUTO_INCREMENT PRIMARY

 KEY, col_name2,
 col_name3;

Create a view stu - views a select regno, name and dob from sto;

Output

mysql> create view stu - views as select

regno, name and dob from stu;

Query ok

mysql> select * from stu - views;

Empty set (0 rows selected) time: 0.000000

stu	regno	name	dob
	19181164	0	1990-01-01
	19181144	0	1990-01-01
	19181134	0	1990-01-01
	1981124	0	1990-01-01
	1981122	0	1990-01-01

mysql> show index from stu;

stu	regno	A

Result

The records from the tables are displayed using JOIN using Equijoin, innerjoin, outerjoin.

Exp-10: Simple Programming exercise using (REPEAT, WHILE)

Aim: To learn how to use various MySQL loop statements including while, repeat to run a block of code repeatedly based on a condition.

Description:

WHILE loop :: The syntax of the WHILE statement as follows.

WHILE expression DO statements.

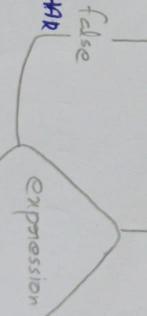
END WHILE

MySQL → BEGIN

CREATE PROCEDURE test-mysql.

DECLARE x INT;

DECLARE str VARCHAR



SET x=1;

SET str=' ';

WHILE x<=5 DO

SET str=CONCAT(str,x);

SET x=x+1;

END WHILE;

SELECT str;

END;

Query OK

you have used up all your wood. Try
and get some more wood.

Output

Output queue: slides, vibration, magnetizable
no wood, electrostatic, shoe, tool
no tools

Sto

1, 2, 3, 4, 5,

you have up to today all your
wood collected

Result: -
you have no resources left

Thus the simple programming exercise
using (REPEAT, WHILE) , executed
~~successfully~~ successfully.

WAKEUP = 2000

DECREASERATE

NUMBER OF SLIDES

1 = X T72

1 = K T72

DO 2 = X WHILE

(.1K(.12)T72)MOD = 402 T72

1 = X 122

END OF FILE

END OF FILE