

DBMS.

Chapter 3.

Unit 1. Theory.

Creating tables in SQL.

SQL - Structured query

language

- not case sensitive
- short, clear, flexible.

ANSI SQL. - C, C++

→ DT Numeric

- ↳ INT (TINYINT,
SMALLINT, MEDIUMINT, BIGINT)
- ↳ FLOAT (M,D)

MAD is size,

e.g., FLOAT (4,2)

now 99.99 as max value -

LL

(DECIMAL (M,D))

eg: DECIMAL (6,2)

① String — (1 and 255 char)
 ↳ CHAR(n)

eg: CHAR(10) — fixed length
 ↳ VARCHAR (n)

↳ VARCHAR(n) — variable length

Complex

③ Date

↳ DATE (YYYY-MM-DD)

↳ DATETIME

YYYY-MM-DD HH:MM:SS

↳ TIMESTAMP

→ n → without hiphen

↳ TIME (HH:MM: SS)

↳ YEAR(CM)

YEAR(4), YEAR(2)

DOMS

(∴ no rollback) LL

→ components.

① DDL

↳ CREATE

↳ DROP

↳ ALTER

{ (Deletes from memory)

↳ TRUNCATE

② DML → for table

↳ INSERT

↳ DELETE

↳ UPDATE

↳ SELECT

③ TCL → Transaction Control

↳ COMMIT

↳ ROLLBACK

④ DCL → Data Control

↳ GRANT

↳ REVOKE

→ DDL

Build and Modify tables
and other objects

Databases objects :-

- Database
- Table
- View

* Auto committed operation

① CREATE

To create database &
table(b)

(a)

(a) Meaning
CREATE DATABASE DB;
SHOW DATABASES;

Using

USE DB;
SELECT DATABASE();



(SHOW ACTIVE
database)

DOMS

11

(b)

• creating

- ↳ CREATE TABLE STUD (
- SRNO VARCHAR (6),
- SNAME — n —,
- PERC DECIMAL (8,2),
- DOB DATE
-);

(list)

- ↳ SHOW TABLES;

• Display

- ↳ DESC STUD

\$

②

ALTER

- Used to modify table

structure

• eg:-

ALTER TABLE STUD ADD COLUMN

Email VARCHAR (30);

— n — MODIFY Email

Email VARCHAR (50);

— n — DROP COLUMN Email;

DESC STUD;

DOMS

③ DROP

from

DB.

· `DROP TABLE STUD;`

④ TRUNCATE

- same as drop
- not used.

⑤ RENAME

DB

· `RENAME TABLE`

`STUP TO STUDENT;`

`SHOW TABLES;`

· Database constraints -

⑥ DOMAIN

· `NOT NULL` defined limits

 ↳ `CHECK` (to impose user-defined limits)

 ↳ `DEFAULT`

· `DATA type
(defined constraints)`

DOMS

creating table with domain
constraints;

CREATE TABLE STUD (

 - n - NOT NULL ,

 - n - NOT NULL ,

 - n - CHECK (FEE > 0) ,

 - n - "Unknown");

DESC STUD ;

Row level

unique

primary key

 ↓ key

(not null unique values)

{

 - n - PRIMARY KEY }

 - n -



(column level
constraints)

DOMS

Referential Integrity
Non-Null

11

FOREIGN KEY

(Refer to PK of other table)



PRIMARY KEY (SNO) DEPARTMENT
FOREIGN KEY DID REFERENCES (DID)

(Table level constraint)

DOMS'

— / —

Leet1 - solved practice

↳ AUTOINCREMENT

↳ AUTO_INCREMENT

(
syntax)

Leet @1. Practical session.

↳ MySQL practical.

— * — Leetmu1 — x — .

DOMS®

Lecture 2: DQL

- ① Insert
- ② Delete
- ③ Update
- ④ Display

(iv) - select * from STUD;
 - displays every row in the table

- select * from STUD WHERE
 column = value;

Row qualifier, e.g.: dep = 'IT'
 condition sid = 101
 problems with

- select column from ...
 - displays only the particular column.

- select distinct, same from STUD;
 - does not repeat.

DQLS

- SELECT * from STUD WHERE
DEPT IN ("CS", "IT");


DEPP IS NULL ,
DEPT IS NOTNULL

- SELECT -n- where SID
BETWEEN W02 AND W07

① Insert VALUES

- INSERT INTO STUDENTS
W21, 'AKHIL', '2003-02-07',
"BTL");
↳ Basic Inserting -

- INSERT INTO STUD(SID,
SNAME) VALUES ('WZ2', 'Vannu');

L only in side &
share, other terms are
will \leftarrow if constraints match

Backup table.
↳ create table → structure
should be same.

INSERT INTO BACKUP SELECT
* FROM STUD; ↑
(no values keyword)

③ Update.

- UPDATE STUD SET

SID = SID + 100;

— for all rows.

- — — — — WHERE DEPT = 'IT';

— — — — WHERE DATE_SAMP =

"2008-07-01";

④ DELETE

- DELETE from STUD;

Deletes all the rows ↗

- DELETE FROM STUD

WHERE DEPT IS NULL ;

NOTE,

SELECT SUM * D1 FROM EMP ;

Lecture 3: Theory. (display, filter & sort)

Row qualifier condⁿ - (FILTERING)

- ↳ comparison ($=, >, <, \leq$)
- ↳ $\text{LIM}(\text{col} \text{ in } ("", ""))$
- ↳ combining (AND, OR, NOT)
- ↳ BETWEEN (standing, ending)
- ↳ LIKE (set inclusion)

⑤ WHERE,

column LIKE Pattern

J

- g a - *
abc% (underscore represents)
%on-b% single char to ignore

j

-- (* or % represents zero or more char % ignore)

DOMS

SORT

L ORDER BY

L ORDER BY EName;

 -- EName ASC;

L --n-- DESC;

LIMIT/NL

L LIMIT n1 OFFSET n2

n1 - no. of rows in OP

n2 - starting row no.

Lecture 3: Practical

- `SELECT * FROM TBLMane;`
- only name, and sal
- `SELECT name, sal from Emp;`
 - \rightarrow distinct \rightarrow unique
 - \rightarrow (unique)
- filtering
 - `age > 30`

`SELECT * from emp WHERE age > 30;`

\rightarrow age not equal to 38
 \leftrightarrow is used.

\therefore condition = true & age > 35

`SELECT * from emp where age > 35 & and loc = 'Pune';`

\uparrow (can be \rightarrow or)
depends on the query.
DOMS

• location in file, min

Selects → from emp. whos
loc IN ('min', 'max')

• between

SELECT — n —

ARE BETWEEN 20 AND 30;

(and a NOT if question)
says,

not,

for string is BETWEEN
"Say" AND "Zog"

it will print all the rows
between 'S' and 'Z'

— / / —

• IN

— h —

— SORT

WM ORDER BY WLNNAME

— n — ORDER BY ENAME, AGE DESC;

{ parent Jay, then for Jay,
age in according
sort

— LIMIT

— n — LIMIT 5;

{} from 5

— n — LIMIT 5, 3;

{} from 5, 3 no.

DOMS