

پوهنتون کابل

پوهنځی کمپیوتر ساینس

دیارتمنت سیستم های معلوماتی

Structured Query Language (SQL) Fundamentals

Lecture 05

تهیه کننده : پوهنیار محمد شعیب "زرین خیل"
سال : 1389

Structured Query Language (SQL) 05

By: M Shuaib Zarinkhail

2010



SQL Basic Commands

- ▶ In command prompt window go to
 - c:\program files\mysql\mysql server 5.0\bin
 - This is when MySQL DBMS is installed in its default directory
 - Type →mysql -u root
- ▶ If the MySQL is password protected then:
 - Type →mysql -p -u root

SQL Basic Commands

- ▶ To create a log-file for this session
 - Type → \T path_file_name.txt
 - or
 - Type → tee path_file_name.txt
 - e. g. → tee e:\SQL88\day4.txt
 - In this example
 - “e” is the drive name
 - “db2practice” is the folder name which should already been created
 - “day4.txt” is the file name that may or may not exist, if it does not exist the MySQL will create it automatically

SQL Basic Commands

- ▶ To stop recording to the log-file for the remaining of the session
 - Type `→ \t`
- Or
 - Type `→ notee`
 - e. g. `→ notee`

SQL Basic Commands

To change the log-file for the remaining of the session to another log-file

- ▶ Retype the first command (s)
- ▶ e. g. →tee e:\SQL88\day5.txt
- ▶ If an existing path_file_name is used
 - New recording will be appended to the existing text file

SQL Basic Commands

To view databases

- ▶ Type → show databases;
 - This command lists the databases on the MySQL server host
 - SHOW SCHEMAS is a synonym for SHOW DATABASES as of MySQL 5.0.2
 - e.g. → Show schemas;
 - You see only those databases for which you have some kind of privilege

SQL Basic Commands

To activate DBs

- ▶ Type → use database_name
- ▶ To use a specific DB named **school**
 - → use school; (semicolon is optional)

Note: You can only use one activate database at a time

- ▶ To change to another DB in the system
 - Retype the previous command with the DB name

SQL Basic Commands

- ▶ To view tables in the active DB (e.g. school)
 - Type → show tables;
 - This command lists the non-TEMPORARY tables in a given database
- Non-Temporary_Tables equal to Base_Tables in DB

SQL Basic Commands

- ▶ To view a table's or view's structure in a DB
 - Type → DESCRIBE TableName or ViewName
 - e. g. → DESCRIBE tOne;
 - e. g. → DESCRIBE vOne;
 - This command provides information about the columns in a table or a view
 - It is a shortcut for SHOW COLUMNS FROM (As of MySQL 5.0.1)
 - DESC is the abbreviation of DESCRIBE

SQL Basic Commands (Drop Table)

- ▶ To delete a table from a DB
 - Type → `DROP TABLE TableName`
 - e.g. → `DROP TABLE tOne;`
- ▶ `DROP TABLE` removes one or more tables
 - You must have the `DROP` privilege for each table
- ▶ All table data and the table definition are removed
- ▶ If any of the tables named in the argument list does not exist
 - MySQL returns an error indicating by the name of the non-existing tables

SQL Basic Commands (Drop Table)

- ▶ Use IF EXISTS to prevent an error from occurring for tables that do not exist
 - A warning_NOTE is generated for each non-existent table when using IF EXISTS in DROP TABLE command
 - e.g. →drop table if exists tOne, tFive;

SQL-DDL (SQL Data Definition Language)

- ▶ You can create databases, tables, indexes, and columns in SQL
- ▶ To do so, you need to define them by names (identifiers)
- ▶ Database, table, index, column, and alias names are identifiers
- ▶ An identifier may be quoted or used unquoted
- ▶ An identifier should not have space, if it has, it must be quoted

SQL-DDL

- ▶ If an identifier contains special characters or is a reserved word, it must be quoted
 - Examples of reserved words are: SELECT, FROM, WHERE, etc
- ▶ Identifiers may begin, end or include digits but they may not consist solely of digits unless they are quoted
- ▶ Identifier quoting was introduced in MySQL 3.23.6

Note: The identifier quote character is the backtick (`)

SQL-DDL (Some restrictions on the characters that may appear in identifiers)

- ▶ Before MySQL 4.1, identifier quote characters should not be used in identifiers
- ▶ Database, table, and column names should not end with space characters
- ▶ Database and table names cannot contain “/”, “\”, “.”, or characters that are not allowed in file names

SQL-DDL

This table shows the maximum size of identifiers for MySQL (Before 4.1.5)

Identifier	Length/byte
Database	64
Table	64
Column	64
Index	64
Constraint	64
Alias	256

SQL-DDL (Create Database)

- ▶ To create a new DB
 - Type → create database DB_name
 - e. g. → create database class2;
 - → use class2
- ▶ This command creates a database with the given name
- ▶ To use this command, you need the CREATE privilege for the database
- ▶ An error occurs if the database exists and you did not specify the IF NOT EXISTS option

SQL-DDL (Create Database)

- ▶ To prevent errors you can type the above command as:
 - create database If Not Exists DB_name
 - e. g. → create database if not exists class2;
 - → use class2
- ▶ CREATE SCHEMA is a synonym for CREATE DATABASE (starting from MySQL 5.0.2)

Note: In case of creating and using a new database, the second command (USE DB_Name) is required!

SQL-DDL (Create Database)

- ▶ As of MySQL 4.1.1, *create_specification* options specify database characteristics
- ▶ Database characteristics are stored in the db.opt file in the database directory
- ▶ A database in MySQL is implemented as a directory
 - The directory containing files that correspond to tables in the database

SQL-DDL (Create Database)

- ▶ The CREATE DATABASE statement only creates a directory under the MySQL data directory
- ▶ If you manually create a directory under the data directory, the server considers it a database directory and it shows up in the output of SHOW DATABASES command
- ▶ You can also use the mysqladmin program to create databases

SQL-DDL (Drop Database)

- ▶ To delete a DB
 - Type → `DROP DATABASE DatabaseName`
 - e. g. → `DROP DATABASE school;`
- ▶ This command deletes all tables in the database and deletes the database
- ▶ To use `DROP DATABASE`, you need the `DROP` privilege on the database
- ▶ `DROP SCHEMA` is a synonym for `DROP DATABASE` (MySQL 5.0.2)

SQL-DDL (Drop Database)

- ▶ IF EXISTS is used to prevent any error from occurring if the database does not exist
 - → Drop database if exists dbName;

Note: Be careful with the DROP command, when you drop a table or a database it can not be restored unless you use “Transactions”

SQL-DDL (Drop Database)

- ▶ If other files or directories remain in the database directory, the database directory cannot be removed
- ▶ In this case, you must remove any remaining files or directories manually and issue the DROP DATABASE statement again
- ▶ You can also drop databases with mysqladmin program

SQL-DDL (Create Table)

- ▶ To create a new table within a DB
 - Type → create table NewTableName (Three-Part Column Description, Three-Part Column Description, Three-Part Column Description, Optional table constraints);
- ▶ CREATE TABLE creates a table with the given name
- ▶ The CREATE privilege for the table is required

SQL-DDL (Create Table)

- ▶ By default, the table is created in the active database
- ▶ An error occurs if
 - The table already exists
 - There is no active database
 - The database does not exist

Note: Database_names, tables_names, etc are already explained

SQL-DDL (Create Table)

- ▶ You can create table(s) for non-active databases
 - The table name should be specified as *db_name.tbl_name* (The DB must exist)
 - e.g. → create table dbOne.tOne (...);
- ▶ If you use quoted identifiers, quote the database and table names separately
 - For example, write: `mydb`.`mytbl`, not as `mydb.mytbl`

SQL-DDL (Create Table)

- ▶ DB temporary tables can be created by adding the TEMPORARY keyword between CREATE and TABLE keywords in create table command
 - e.g. CREATE TEMPORARY TABLE tableName (table Definitions)
- ▶ IF NOT EXISTS option can also be added to this command to show warnings instead of errors
 - e.g. → CREATE TABLE IF NOT EXISTS tableName (table Definitions)

SQL-DDL (Create Table)

- ▶ Similarly a table could be created from one or more existing tables and/or views
- ▶ All options in regular query could be implemented in this case
- ▶ For example we can select, project, negate, product, divide, and join existing tables
 - e.g. → `CREATE TABLE tThree SELECT * FROM tOne, tTwo;`

SQL-DDL (Create Table)

- ▶ MySQL represents each table by an .frm table format (definition) file in the database directory
- ▶ The storage engine for the table might create other files as well
- ▶ For each table *tbl_name*, there are three disk files
 - *tbl_name.frm* (Table format – Definition file)
 - *tbl_name.MYD* (Data file)
 - *tbl_name.MYI* (Index file)

SQL-DDL (Examples)

- ▶ To create table PROJECT, Type:

```
→ CREATE TABLE PROJECT (  
    ProjectID    Integer    Primary Key,  
    Name         Char(25)   Not Null,  
    Department   VarChar(100)  Null,  
    MaxHours     Numeric(5,2) Default 100  
);
```

SQL-DDL (Examples)

- ▶ To create table EMPLOYEE, Type:
→ CREATE TABLE EMPLOYEE (
EmployeeNumber Integer Primary Key,
Name Char(25) Not Null,
Phone Char(8),
Department VarChar(100)
);

SQL-DDL (Examples)

- ▶ To create table ASSIGNMENT, Type:
CREATE TABLE ASSIGNMENT (
 ProjectID Integer Not Null,
 EmployeeNum Integer Not Null,
 HoursWorked Numeric(5,2) Default 10
);