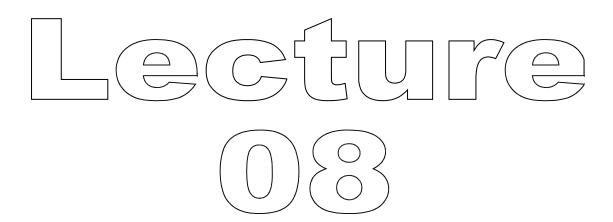
پوهنتون کابل پوهنځی کمپیوترساینس

دیپار تمنت سیستم های معلوماتی

Structured Query Language (SQL) Fundamentals



تهیه کننده : پوهنیار محمد شعیب "زرین خیل"

سال : 1389

Structured Query Language (SQL) 08

By: M Shuaib Zarinkhail

2010

Comments in MySQL

MySQL Server supports three comment styles:

- 2. From a "#" character to the end of line
- 3. From a "-- " sequence to the end of line
 - This style is supported as of MySQL 3.23.3
 - In MySQL, the "--" (double-dash) comment style requires the second dash to be followed by at least one whitespace or control character (such as a space, tab, newline, and so on)

Comments in MySQL

MySQL Server supports three comment styles:

- - This is the format using in standard SQL
 - Similar to the C programming language
 - This syntax allows a comment to extend over multiple lines
 - The beginning and closing sequences need not be on the same line

Comments in MySQL

The following example demonstrates all three styles for MySQL comments:

```
SELECT 1+1; # comment to the end of line

SELECT 1+1; -- comment to the end of line

SELECT 1 /* in-line comment */ + 1;

SELECT 1+

/* this is a

multiple-line comment

*/

1;
```

- ALTER TABLE enables you to change the structure of an existing table
- For example you can
 - add or delete columns
 - create or destroy indexes
 - change the type of existing columns
 - rename columns or a table itself
- Also you can change the comment for a table and type of a table

- The syntax for many of the allowable alterations is similar to clauses of the CREATE TABLE statement
- Some operations may result in warnings when attempted on a table
 - This include operations which are not supported by the storage engine
 - In MySQL 4.1 and up, these warnings can display by SHOW WARNINGS command

- If you use ALTER TABLE to change a column specification but the DESCRIBE command indicates that the column was not changed, it is possible and MySQL Server ignored your modification
- This occur when altering an irregular update on the structure of a table

- In most cases, ALTER TABLE works by making a temporary copy of the original table
 - The alteration is performed on the copy, and then the original table is doped and the new table is renamed and replaced
 - While ALTER TABLE is executing, the original table is readable by other sessions
 - Updates and writes to the table are stalled until the new table is ready

RENAME TABLE

- We can rename an existing table in MySQL
- One way is using the Alter Table command
- To do:

Type →ALTER TABLE TableOldName REANME TO TableNewName e.g. →alter table tOne rename to tTwo;

ADD COLUMN

You can add a column to an existing table in a database

→ ALTER TABLE TableName ADD COLUMN ThreePartColumnDefinition

e.g. → alter table tOne add column (salary integer(4) null);

ADD more than one COLUMN

- You can add more than one column to an existing table in MySQL
- This is possible by using the ALTER Table command

```
e.g. → alter table tOne add column (
Address1 char(12) not null,
Address2 char(12) null
```

ADD COLUMN

- As stated earlier, you can add a column to an existing table
- By default new column is added to the end of other columns
 - → ALTER TABLE TableName ADD COLUMN ThreePartColumnDefinition
 - e.g. → alter table tOne add column (salary integer(4) null);

ADD COLUMN

- You can add a column in a specific place at the beginning or among the columns
 - For this purpose one of the FIRST or AFTER keywords is used
 - e.g. → alter table tOne add column (ID int) first;
 - e.g. → alter table tOne add column (salary int) after address;

ADD INDEX

- You can set a column in a table as indexed
 - → ALTER TABLE TableName ADD INDEX IndexName (ColName)
 - → Alter table tOne add index iOne (colOne);

ADD INDEXes

- You can set more than one column in a table as indexed (composite)
 - → ALTER TABLE TableName ADD INDEX IndexName (ColNames)
 - → e.g. alter table tOne add index iTwo (colTwo, colSix);

CREATE INDEX

- Similar to the previous command, you can directly create an INDEX in a DB
 - → CREATE INDEX IndexName ON TableName (colNames)
 - → e.g. create index iThree on tTwo (colName);
- This command is also applicable for more than one column(s)

SHOW INDEX

- You can preview indexes in your tables
 - → SHOW INDEX FROM DatabaseName.TableName
 - → e.g. show index from dbOne.tOne;
- For active DB, database name is optional
- SHOW KEYS is a synonym for SHOW INDEX

DROP INDEX

- You can drop indexes using ALTER TABLE command
 - → ALTER TABLE TableName DROP INDEX IndexName
 - → e.g. alter table tOne drop index iOne;

DROP INDEX

- You can directly run the DROP INDEX command on a table
 - → DROP INDEX IndexName ON TableName
 - → e.g. drop index iTwo on tOne;

DROP COLUMN

- You can drop a column from a table using the ALTER TABLE command
 - → ALTER TABLE tableName DROP COLUMN ColumnName
 - → e.g. alter table tTwo drop column salary;

MODIFY COLUMN DEFINITION

- You can modify a column's definition by ALTER TABLE command
 - → ALTER TABLE TableName MODIFY COLUMN ColumnName NewDefinition
 - → e.g. alter table tOne modify column salary integer(22);

To convert a table from one storage engine to another, use an ALTER TABLE statement that indicates the new engine:

ALTER TABLE tOne ENGINE = MYISAM; ALTER TABLE tOne TYPE = BDB;