پوهنتون کابل

پوهنځی کمپیوترساینس

دیپارتمنت سیستم های معلوماتی

Structured Query Language (SQL) Fundamentals

Lectures 16

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

Structured Query Language (SQL) 16 + Lab 02

By: M Shuaib Zarinkhail 2010

SQL-DML (Delete Data)

- You can delete userdata from a DB:

Type → DELETE FROM TABLENAME
WHERE criteria (optional)

e.g. →delete from project
where MaxHours = 76.0;

- Data can be deleted from one table or more than one table at a time

Note: Be aware you can not undo a delete command in DB unless using *transaction … rollback*

Delete (Single-Table Syntax)

- To delete data from one table, you can use the following syntax

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
  FROM table_reference
[WHERE where_condition]
[ORDER BY ...]
[LIMIT row_count]
```

- *The keywords in braces are optional*

Delete (Multiple-Table Syntax)

- To delete data from many tables, you can use the following syntax

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
tbl_names
FROM table_references
[WHERE where_condition]
```

- *tbl_names and tbl_references should be equal and reference the same tables*

Delete (Single-Table)

- For the single-table syntax, the DELETE statement deletes rows from table_reference
- The WHERE clause specifies the conditions that identify which rows to delete
  - With no WHERE clause, all rows are deleted

e.g. DELETE FROM tblOne WHERE Dep = ‘Math’ ORDER BY Hire_Date LIMIT 1;

Delete (Single-Table)

- The ORDER BY deletes rows in the order that specified
- ORDER BY may also be useful in some cases to delete rows in an order required to avoid referential integrity violations
- ORDER BY can be used with DELETE beginning MySQL 4.0
- The LIMIT clause places a limitation on the number of rows that can be deleted

Delete (Multiple-Table)

- Multi-table delete is added from MySQL 4.0
- As already showed, you can specify many tables in the DELETE statement to delete rows from one or more of them
  - It depends on a particular condition in multiple tables
- You cannot use ORDER BY or LIMIT in a multiple-table DELETE

Delete (Multiple-Table)

- For the multiple-table syntax, DELETE deletes from each tbl_name the rows that satisfy the conditions
- In multi-table DELETE, the ORDER BY and LIMIT cannot be used
  - e.g. DELETE tOne, tFour FROM tOne, tFour WHERE tOne.Dep = tFour.Department AND tOne.Salary>5000;

SQL-DML (Delete Data)

- As stated, a DELETE statement with no WHERE clause deletes all rows
- A faster way to do this, when you do not need to know the number of deleted rows, you can use the "TRUNCATE TABLE" command
  - TRUNCATE TABLE TableName
  - i.e. truncate table tblOne;

SQL-DML (Delete Data)

- The affected table with TRUNCATE command can not be rolled back
  - While using the DELETE command in a transaction safe mode, you can rollback the deleted data
- DELETE is much slower than TRUNCATE, because it deletes one row at a time and TRUNCATE empties a table at once

SQL-DML (Delete Data)

- In MySQL 3.23, DELETE without a WHERE clause returns zero as the number of affected rows
- In MySQL 3.23, if you want to know the number of deleted rows in a DELETE command, you have to add a WHERE clause with an expression that is true for every row in DELETE command
  - e.g. DELETE FROM tbl_name WHERE 1>0;

SQL-DML (Delete Modifiers)

1. LOW_PRIORITY:

- The server delays execution of the DELETE until no other clients are reading from the table
- This affects only storage engines that use table-level locking (MyISAM, MEMORY, MERGE)

SQL-DML (Delete Modifiers)

2. QUICK:

- It may speed up some kinds of delete operations

3. IGNORE

- Causes MySQL to ignore all errors during the process of deleting rows
- Errors are returned as warnings
- This option first appeared in MySQL 4.1.1

SQL-DML (Delete Problems)

- If you are deleting many rows from a large table, you may exceed the lock table size for an InnoDB table

- To avoid this problem, or to minimize the time for delete, the following strategy (which does not use DELETE at all) might be helpful: … *NEXT SLIDE*

Delete Data (Without using DELETE)

1. Select the rows not to be deleted into an empty table that has the same structure as the original table:
   - INSERT INTO t_copy SELECT * FROM t WHERE ... ;
2. Use RENAME TABLE to move the original table out of the way and rename the copy to the original table:
   - RENAME TABLE t TO t_old, t_copy TO t;
3. Drop the original table:
   - DROP TABLE t_old;

Delete Data (Using Aliases)

As of MySQL 4.1.2, aliases can be used where:

- In the list of tables from which to delete rows, aliases will have a default database unless one is specified explicitly
- For example, if the current database is test, the following statement does not work because the unqualified alias a1 has a default database of test: *NEXT SLIDE*

Delete Data (Using Aliases)

  - DELETE a1, a2 FROM db1.t1 AS a1 INNER JOIN db2.t2 AS a2 WHERE a1.id=a2.id;

- To correctly match the alias, you must explicitly qualify it with the database of the table being aliased:
  - DELETE db1.a1, db2.a2 FROM db1.t1 AS a1 INNER JOIN db2.t2 AS a2 WHERE a1.id=a2.id;

Lab 02 – Movies Database

In this lab you have to:

- Create new tables in the movies database
- Load data to the databases from external data files

At the end of lab time:

- Record your answers and turn them to lab instructor
- Keep the database for future labs (lab02 and lab03)