

Module 17: MySQL BR & IE Data

Backup & Restore Database:

- ☐ It's important for DBA to regularly back up the database so that if the database becomes corrupted then DBA can use the backup to restore the database.
- ☐ Backup includes the structure and content of a database.
- ☐ We often include the database named "mysql" in backups since this database stores information about the users and the privileges for all databases on the server.

Backup:

- ☐ We use mysqldump program to backup or dump one or more databases into a SQL script file.

- ☐ **For single database:**

- ☐ `mysqldump db_name > file_path/file_name.sql -u root -p`

- ☐ **For multiple databases:**

→ ☐ `mysqldump`

- ☐ **For all databases:**

- ☐ `mysqldump --all-databases > file_path/file_name.sql -u root -p`

Restore:

- ☐ We can use mysql program to restore database / databases by running the SQL script file that contains the database backup.

- ☐ **For single database:**

→ ☐ `mysql`

- ☐ **For multiple databases:**

- ☐ `mysql < file_path/file_name.sql -u root -p`

Export and Import Data:

- ☐ Backing up a database can be referred as exporting a database and restoring a database can be referred as importing a database.
- ☐ Sometimes we may need just to export data from database to a file or import data from a file to a database.

Export data:

- ☐ We can add an INTO OUTFILE clause to a SELECT statement to save the result set into an output file.
- ☐ The syntax is:
 - ☐ **SELECT** column_list
 - ☐ **INTO OUTFILE** 'file_path'
 - ☐ [**FIELDS** **[TERMINATED BY string]**
 [ENCLOSED BY char]
 [ESCAPED BY char]]
 - ☐ **FROM** table_name
 - ☐ **[WHERE search_condition]**
 - ☐ **[ORDER BY order_by_list]**
- ☐ We can use FIELD clause to specify the character that will be used to delimit columns and enclose the data.
- ☐ We can use LINES clause to specify the new line character.

Important Points:

- ☐ Use of INTO OUTFILE changes the operation of the SELECT statement in several ways:
- ☐ The output produced by a SELECT ... INTO OUTFILE statement never leaves the server host. Instead of sending the result over the network to the client, the server writes it to a file on the server host.
- ☐ To prevent files from being overwritten, either accidentally or maliciously, the server requires that the output file not already exist.
- ☐ The statement causes the server to write a new file on the server host, so you must connect to the server using an account that has the FILE privilege. The file is created with file system access permissions that make it owned by the MySQL server but world-readable.
- ☐ The output file contains one line per row selected by the statement. By default, column values are delimited by tab characters and lines are terminated with newlines, but you can control the output format by adding format specifiers after the filename

Importing Data:

- ☐ LOAD DATA INFILE provides an alternative to INSERT for adding new records to a table.
- ☐ It reads the data from a separate data file.
- ☐ The syntax is:
 - ☐ **LOAD DATA [LOCAL] INFILE 'file_name'**
 - ☐ **[IGNORE | REPLACE]**
 - ☐ **INTO TABLE *table_name***
 - ☐ ***format_specifiers***
 - ☐ **[IGNORE *n* LINES]**
 - ☐ **[(*column_list*)]**

LOCAL:

- ☐ MySQL assumes the file is located on the server host and server reads the file directly.
- ☐ LOCAL means local to the client host from which statement is issued and hence the client program reads the data file and sends its contents over the network to the server.

IGNORE AND REPLACE:

- ☐ By default, an input record that causes a duplicate-key violation results in an error and the rest of the data file isn't loaded. Records processed up to that point are loaded into the table.
- ☐ IGNORE: If you specify the IGNORE keyword following the filename, new records that cause duplicate-key violations are ignored and no error occurs. LOAD DATA INFILE processes the entire file, loads all records not containing duplicate keys, and discards the rest.
- ☐ REPLACE: If you specify the REPLACE keyword after the filename, new records that cause duplicate-key violations replace any records already in the table that contain the duplicated key values. LOAD DATA INFILE processes the entire file and loads all its records into the table.
- ☐ IGNORE and REPLACE are mutually exclusive.

format_specifiers:

- ☐ LOAD DATA INFILE and SELECT ... INTO OUTFILE assume a default data file format in which column values are separated by tab characters and records are terminated by newlines.
- ☐ If a data file to be read by LOAD DATA INFILE has different column separators or line terminators, you must indicate what the format is so that MySQL can read the file contents correctly.
- ☐ For LOAD DATA INFILE,
 - ☐ format specifiers are listed after the table name.

- ☐ For SELECT ... INTO OUTFILE,
 - ☐ they follow the output filename.
- ☐ The syntax for format specifiers is the same for both statements and looks like this:
- ☐ FIELDS → formatting of data values within a line.
 - ☐ TERMINATED BY 'string'
 - ☐ ENCLOSED BY 'char'
 - ☐ ESCAPED BY 'char'
- ☐ LINES → defines line ending sequence.
 - ☐ TERMINATED BY 'string'

IGNORE n LINES:

- ☐ To ignore the initial part of the data file, use the IGNORE *n* LINES clause, where *n* is an integer that indicates the number of input lines to skip.
- ☐ This clause commonly is used when a file begins with a row of column names rather than data values.

column_list:

- ☐ By default, LOAD DATA INFILE assumes that data values in input lines are present in the same order as the columns in the table.
- ☐ If the data file contains more columns than the table, MySQL ignores the excess data values.
- ☐ If the data file contains too few columns, each missing column is set to its default value in the table. (This is the same way MySQL handles columns that aren't named in an INSERT statement.)
- ☐ If input lines don't contain values for every table column, or the data values are not in the same order as table columns, add a comma-separated list of column names within parentheses at the end of the LOAD DATA INFILE statement.

MCQs:

Q1) Which of these commands is used for database backup purposes in MySQL:

Options:

- | | |
|---------|----------------|
| A. dump | B. mysqldump |
| C. save | D. mysqlbackup |

Solution:

Q2) Which of these commands is used for database restore purposes in MySQL:

Options:

- | | |
|-----------------|--------------|
| A. load | B. mysqldump |
| C. mysqlrestore | D. mysql |

Solution:

Q3) which are the signs used for backup and restore operations. Select all that apply.

Options:

- A. >=
- B. >
- C. <--
- D. <

Solution:

Q4) which is the command for exporting data ?

Options:

- A. SELECT column_list INTO filepath
- B. SAVE column_list INTO OUTFILE filepath
- C. SELECT column_list INTO OUTFILE filepath
- D. EXPORT column_list INTO OUTFILE filepath

Solution:

Q5) in the export file the column values are delimited by:

Options:

- A. tab character
- B. , character
- C. # character
- D. \$ character

Solution:

Q6) Which of the following command is used for importing the data into a table.

Options:

- A. IMPORT DATA INTO TABLE table_name
- B. LOAD DATA INTO TABLE table_name
- C. IMPORT DATA INFILE file_name INTO TABLE table_name
- D. LOAD DATA INFILE file_name INTO TABLE table_name

Solution: