Databases

IT Learning &
Outsourcing Center

SQL (DML and DDL)

Lector: Hristo Topuzov
E-mail: Hristo.Topuzov@pragmatic.bg

www.pragmatic.bg

2018

Agenda

- Basic Data Types in MySQL
- Data manipulation language (DML)
- Data definition language (DDL)

DML and DDL

- SQL statements are divided into two major categories:
  - Data manipulation language (DML) - used for managing data within schema objects. The data manipulation language statements are – select, insert, update, delete.
  - Data definition language (DDL) - used to define the database structure or schema. The data definition language statements are – create, alter, drop

Basic Data Types

- MySQL includes a number of data types such as:
  - DOUBLE([size, [d]]) -a large number with floating point number
  - BIGINT - large integer (8 bytes)
  - INT - integer (4 bytes)
  - MEDIUMINT - medium sized integer (3 bytes)
  - SMALLINT - small integer (2 bytes)

Basic Data Types

- MySQL includes a number of data types such as:
  - TEXT -large block of text
  - VARCHAR(size) -variable-length string
  - CHAR(size) -fixed-length string
  - BIT - bit-field values
  - DATE -date
  - TIMESTAMP -date and time

DML – Inserting Data

- INSERT command
  - INSERT INTO <table> VALUES (<values>)
  - INSERT INTO <table>(<columns>) VALUES (<values>)
  - INSERT INTO <table> SELECT <values>

```
INSERT INTO COUNTRY
VALUES ('1', 'Bulgaria', 'Sofia')

INSERT INTO COUNTRY(NAME, CAPITAL)
VALUES ('Bulgaria', 'Sofia')

INSERT INTO COUNTRY(COUNTRY_ID, NAME, CAPITAL)
SELECT NULL, COUNTRY, CAPITAL FROM CAPITALS
```

DML – Inserting Data

- When some of the columns in an insert statement are omitted then they are given either:
  - A NULL value
  - A DEFAULT VALUE (if defined for the column)If a value for a column is not defined and the column has a NOT NULL constraint (and no default value is defined) - the insert statement **fails** to execute

DML – Updating Data

- UPDATE command
  - UPDATE <table> SET <column=expression> WHERE <condition>
  - **Note: Don't forget the WHERE clause!**

```
UPDATE PERSONS
SET NAME = 'Updated Name'
WHERE PERSON_ID = 1

UPDATE EMPLOYEES
SET SALARY = SALARY * 1.10
WHERE DEPARTMENT_ID = 3
```

DML – Updating Data

- Updating joined table:
  - UPDATE employees e

    JOIN titles t on t.Id = e.titleid

    SET salary = 1.1*salary

    WHERE t.name = 'Manager‘

Note: Don't forget the WHERE clause!

DML – Deleting Data

- Deleting rows from a table
  - DELETE FROM <table> WHERE <condition>

```
DELETE FROM PERSONS WHERE PERSON_ID = 1

DELETE FROM PERSONS WHERE NAME LIKE 'S%'
```

Note: Don't forget the WHERE clause!

- The DELETE statement conflicted with the reference constraint (FK)

DML – Deleting Data

- Delete all rows from a table at once
  - TRUNCATE TABLE <table>

```
TRUNCATE TABLE PERSONS
```

- Cannot truncate table if it is being referenced by a FOREIGN KEY constraint

DML – Deleting Data

- Deleting from joined tables:
  - DELETE e FROM employees e

    JOIN titles t on t.Id = e.titleid

    WHERE t.name = 'Junior Quality Assurance Engineer'

Note: Don't forget the WHERE clause!

Data Definition Language (DDL)

- Types of commands
  - Defining / editing objects
    - CREATE
    - ALTER
    - DROP
  - Managing access permissions
    - GRANT
    - REVOKE

DDL – Create table

- In order to create a new table:
  - Define the table name
  - Define the columns and their types
  - Define the table constraints (including primary/foreign keys) and default value
- Types of constraints:
  - NOT NULL (in MySQL it is not defined as a constraint)
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK (not enforced by MySQL)

DDL – Create table

- A table is created with the CREATE TABLE statement
  - CREATE TABLE <name> (<fields definitions>)

```
CREATE TABLE PEOPLE(
   PERSON_ID NUMBER AUTO_INCREMENT NOT NULL,
   NAME VARCHAR(100) NOT NULL,
   CONSTRAINT PERSONS_PK PRIMARY KEY(PERSON_ID)
);
```

DDL – Create table

- Constraints may be specified:
  - as part of the column definitions
  - in the CREATE TABLE statement (outside the column definitions)
  - outside the CREATE TABLE statement using ALTER TABLE statements

DDL – Create table

- Example (constraints are part of the column definitions):

```
CREATE TABLE LOCATIONS1 (
  Id INT AUTO_INCREMENT PRIMARY KEY,
  City VARCHAR(100) NOT NULL UNIQUE,
  Country VARCHAR(100) DEFAULT 'Bulgaria',
  DateAdded DATE,
  Status VARCHAR(10) CHECK (Status in ('OPENING',
'OPENED')),
  ManagerId INT REFERENCES Employees(Id)
);
```

DDL – Create table

- Example (constraints are outside the column definitions):

```
CREATE TABLE LOCATIONS2 (
  Id INT AUTO_INCREMENT,
  City VARCHAR(100) NOT NULL,
  Country VARCHAR(100) DEFAULT 'Bulgaria',
  DateAdded DATE,
  Status VARCHAR(10),
  ManagerId INT,
  CONSTRAINT c_locations2_PK PRIMARY KEY(Id),
  CONSTRAINT c_locations2_City_Unq UNIQUE (City),
  CONSTRAINT c_locations2_Status CHECK (Status in
('OPENING', 'OPENED')),
  CONSTRAINT loc2_fk_empl FOREIGN KEY(ManagerId)
REFERENCES Employees(Id));
```

DDL – Create table

- Example (constraints are outside the table definitions):

```
CREATE TABLE locations3 (
  Id INT AUTO_INCREMENT,                    NO CONSTRAINTS YET
  City VARCHAR(100),
  Country VARCHAR(100) DEFAULT 'Bulgaria',
  DateAdded DATE,
  Status VARCHAR(10),
  ManagerId INT
);
```

DDL – Create table

- Example (constraints are outside the table definitions):

CONSTRAINTS ARE ADDED AFTER THE TABLE DEFINITION

```
alter table Locations3 add constraint c_locations3_PK
PRIMARY KEY(Id);
alter table Locations3 add constraint
c_locations3_City_Unq UNIQUE (City);
alter table Locations3 add constraint
c_locations3_Status CHECK (Status in ('OPENING',
'OPENED'));
alter table Locations3 add constraint c_locations3_fk
FOREIGN KEY(ManagerId) REFERENCES Employees(Id);
alter table Locations3 modify City varchar(100) NOT
NULL;
```

DDL - Alter

- The ALTER TABLE statement is used to add, delete, or modify columns in an existing table

- To add a column in a table, use the following syntax:

```
ALTER TABLE table_name
ADD column_name datatype
```

DDL - Alter

- To delete a column in a table, use the following syntax

```
ALTER TABLE table_name
DROP COLUMN column_name
```

- To change the data type of a column in a table, use the following syntax

```
ALTER TABLE table_name
MODIFY COLUMN column_name datatype
```

DDL - Drop

- The DROP statement is used to delete a table
- DROP command:
  - DROP TABLE <name>
- Example:

```
DROP INDEX C_LOCATIONS4_CITY_UNQ ON LOCATIONS3;
ALTER TABLE LOCATIONS3 DROP PRIMARY KEY;
ALTER TABLE LOCATIONS3 DROP FOREIGN KEY
C_LOCATIONS4_FK;
```

- In MySQL a unique constraint is dropped by dropping the corresponding index

Questions