SchoolNova

IT102

Data Definition Language

(based on the materials from lasalle.edu)

DDL

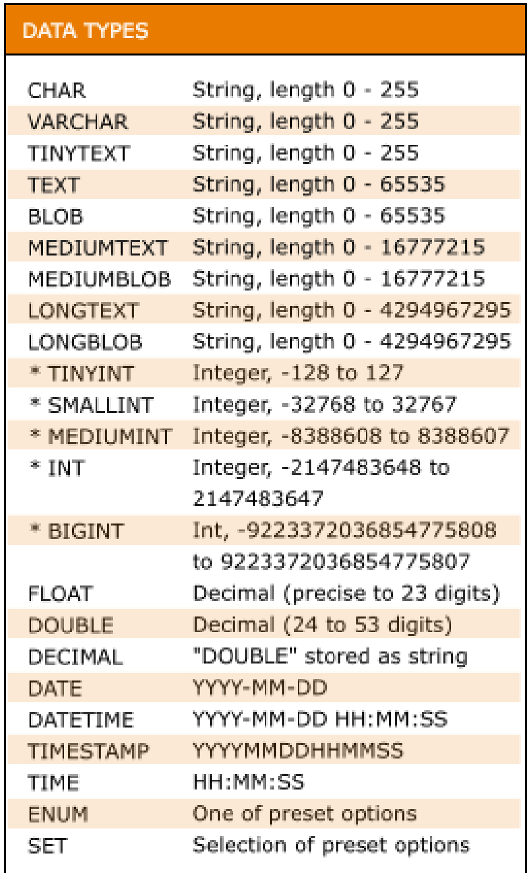
- The Data Definition Language (DDL) is the part of SQL that allows one to set up a database schema, that is, to define attributes, tables, views and relationships.

- One of the smallest defined units is an attribute.
- An attribute is given a name (identifier) and assigned a type.

Identifiers

- An **identifier** is the name assigned to a table, column, view, etc.
  - The identifier is a string of characters from some set.
  - The string is at most 128 characters long.
  - The standard set of characters consists of capital letters (A,B,..), small letters (a,b,…), digits (0,1,…) and the underscore character (_).
  - An identifier starts with a letter.
  - An identifier contains no spaces (allowed by MS Access but not by standard SQL).

Data Types



DATA TYPES	
CHAR	String, length 0 - 255
VARCHAR	String, length 0 - 255
TINYTEXT	String, length 0 - 255
TEXT	String, length 0 - 65535
BLOB	String, length 0 - 65535
MEDIUMTEXT	String, length 0 - 16777215
MEDIUMBLOB	String, length 0 - 16777215
LONGTEXT	String, length 0 - 4294967295
LONGBLOB	String, length 0 - 4294967295
* TINYINT	Integer, -128 to 127
* SMALLINT	Integer, -32768 to 32767
* MEDIUMINT	Integer, -8388608 to 8388607
* INT	Integer, -2147483648 to 2147483647
* BIGINT	Int, -9223372036854775808 to 9223372036854775807
FLOAT	Decimal (precise to 23 digits)
DOUBLE	Decimal (24 to 53 digits)
DECIMAL	"DOUBLE" stored as string
DATE	YYYY-MM-DD
DATETIME	YYYY-MM-DD HH:MM:SS
TIMESTAMP	YYYYMMDDHHMMSS
TIME	HH:MM:SS
ENUM	One of preset options
SET	Selection of preset options

- Use BIT for Boolean values (1 or 0);
- CHAR and VARCHAR are character strings. When defining an attribute of the character type, one includes a length. The CHAR variable must have the specified length, whereas the VARCHAR variable can be up to the specified length.
    - characterID CHAR(6)
    - fName VARCHAR(30)
- INT and SMALLINT are used for whole numbers, such as: 56, 0 or –77. SMALLINT is used if the numbers are fairly small and one wants to save space.
    - rooms SMALLINT
    - population INT
- FLOAT, DOUBLE and DECIMAL - in floating point numbers, the decimal place moves and digits on the right-hand side (least significant) might be dropped during various calculations. MySQL performs rounding when storing values, so if you insert 999.00009 into a FLOAT(7,4) column, the approximate result is 999.0001. DECIMAL stores exact numeric data values.
- DATE, TIME and TIMESTAMP are used for variables that track when something happened. DATE keeps track of year, month and day. TIME keeps track of hours, minutes and seconds. The TIMESTAMP data type is used for values that contain both date and time parts. TIMESTAMP has a range of '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC.

Complete documentation: http://dev.mysql.com/doc/refman/5.0/en/data-types.html

NULL, NOT NULL and DEFAULT

- A simple but important domain constraint is whether or not a value is allowed to be NULL.
- NULL is used if the data is unknown or not applicable.
- NULL is distinct from 0 for numbers and “” (the empty string) for characters.
- The SQL key words are NULL and NOT NULL
  - NULL means the data can have the NULL value
  - NOT NULL means it cannot

```
characterID CHAR(6) NOT NULL
```

- One can specify a value to use as a **default**.
- Example:

```
CREATE_TS TIMESTAMP DEFAULT CURRENT_TIMESTAMP

LAPTOP_OS VARCHAR(256) DEFAULT 'Windows 7'
```

Primary Keys

- No part of a primary key can be NULL and that the primary key must uniquely identify a record (known as Entity Integrity).
- The keyword PRIMARY KEY specifies that attribute(s) will serve this purpose.

```
PRIMARY KEY(characterID)

PRIMARY KEY(episodeID, characterID)
```

- Candidate key is an attribute or set of attributes having the same features as the primary key (uniqueness).
- To specify that a set of attributes must have this uniqueness property, use the keyword UNIQUE

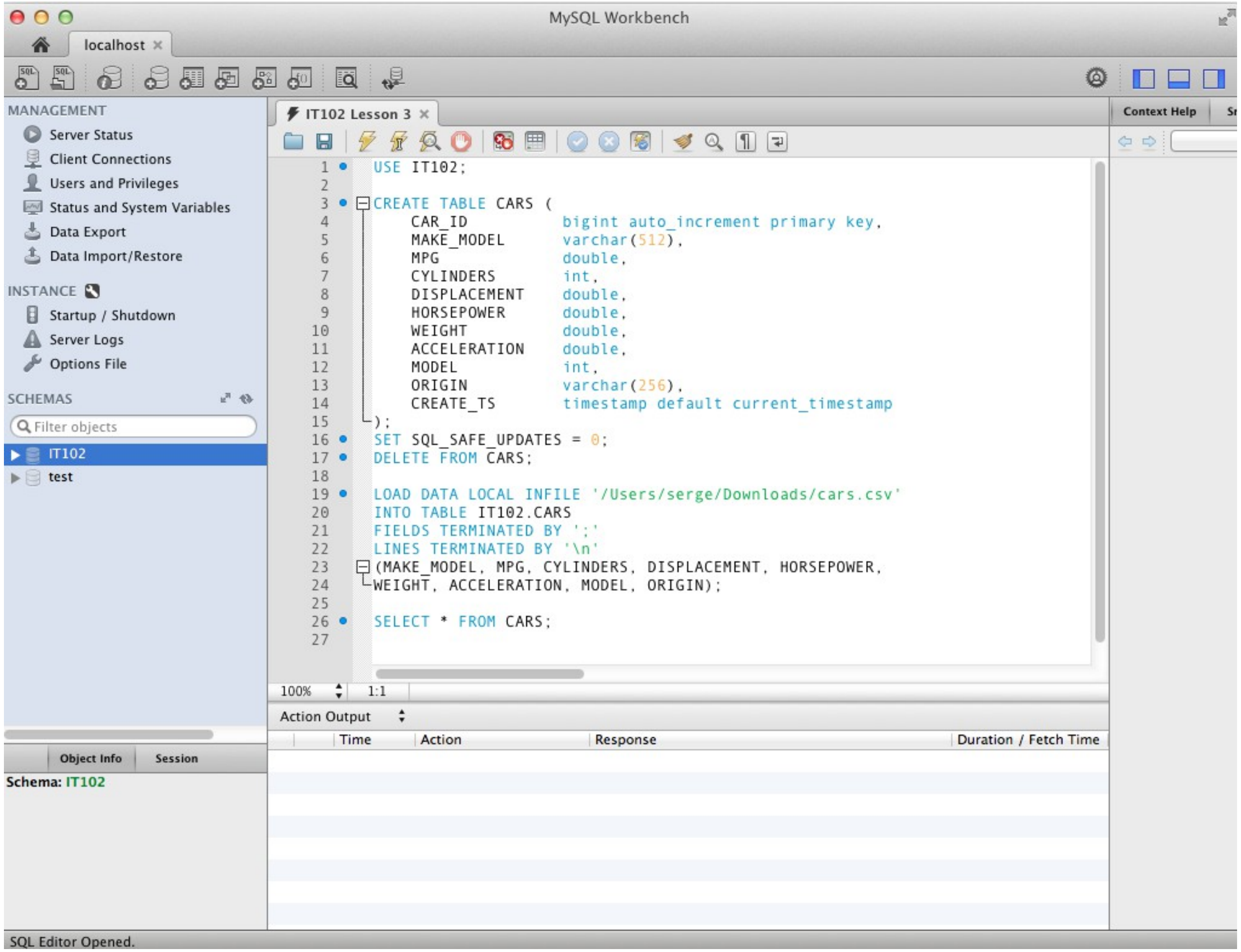
```
PRIMARY KEY(date, time, room)

UNIQUE(date, time, student_id)
```

CREATE TABLE Statement

```
CREATE TABLE table_name
(
column_name1 data_type(size),
column_name2 data_type(size),
column_name3 data_type(size),
....
);
```

Example

```
CREATE TABLE Persons
(
PersonID int,
LastName varchar(255),
FirstName varchar(255),
Address varchar(255),
City varchar(255)
);
```



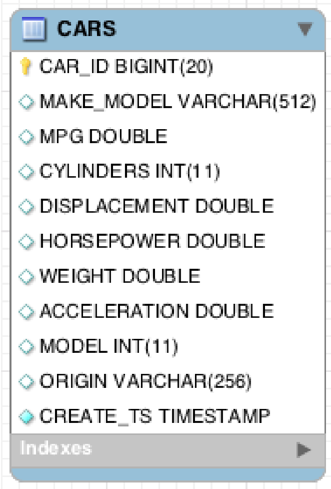
```
USE IT102;

CREATE TABLE CARS (
    CAR_ID          bigint auto_increment primary key,
    MAKE_MODEL      varchar(512),
    MPG             double,
    CYLINDERS       int,
    DISPLACEMENT    double,
    HORSEPOWER      double,
    WEIGHT          double,
    ACCELERATION    double,
    MODEL           int,
    ORIGIN          varchar(256),
    CREATE_TS       timestamp default current_timestamp
);
SET SQL_SAFE_UPDATES = 0;
DELETE FROM CARS;

LOAD DATA LOCAL INFILE '/Users/serge/Downloads/cars.csv'
INTO TABLE IT102.CARS
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\n'
(MAKE_MODEL, MPG, CYLINDERS, DISPLACEMENT, HORSEPOWER,
WEIGHT, ACCELERATION, MODEL, ORIGIN);

SELECT * FROM CARS;
```

Homework

- Open MySQL Workbench and connect to your database.
- Write a DDL statement to create a CARS table from the ER Diagram on your right.
    - CAR_ID is AUTO_INCREMENT PRIMARY KEY
    - CREATE_TS has DEFAULT CURRENT_TIMESTAMP
    - Tutorial: http://www.tutorialspoint.com/mysql/mysql-create-tables.htm
    - Reference: http://dev.mysql.com/doc/refman/5.1/en/create-table.html
- Enter a couple of records using SQL statements, for example:

```
INSERT INTO CARS (MAKE_MODEL, MPG, CYLINDERS, DISPLACEMENT, MODEL)
VALUES ('Volkswagen Jetta', 29, 4, 1600, 2016);
```

- Run a SELECT * FROM CARS statement to see if the data loaded successfully.



CARS

- CAR_ID BIGINT(20)
- MAKE_MODEL VARCHAR(512)
- MPG DOUBLE
- CYLINDERS INT(11)
- DISPLACEMENT DOUBLE
- HORSEPOWER DOUBLE
- WEIGHT DOUBLE
- ACCELERATION DOUBLE
- MODEL INT(11)
- ORIGIN VARCHAR(256)
- CREATE_TS TIMESTAMP