SQL DDL

Data Definition Language

Structured Query Language

- Practical implementation of the relational model
- Originally SEQUEL (Structured English QEUry Language) at IBM research
- SQL became standard in 1986
- Supported by all major RDBMS vendors, with minor (and sometimes major) differences

SQL's big advantage: if you stick to ANSI SQL, your database code is portable between RDBMS systems.

SQL Relational Model

- Relations are tables
- Tuples are rows
- Attributes are columns

For the most part these terms are interchangeable.

- Important difference: tables allow duplicate rows

Schemas and Catalogs

A schema (database in the relational model) is a collection of related tables and constructs. A schema has:

- a schema name
- an authorization identifier (user who owns the schema)

In MySQL `create schema` is a synonym for `create database`.
A catalog is a named collection of schemas. MySQL includes a `table_catalog` column in its `information_schema.tables` table for compatibility with the SQL standard, but does not use catalogs.

CREATE TABLE

The `CREATE TABLE` command creates a base table (`CREATE VIEW` creates a virtual or derived table):
General form:

```
CREATE TABLE <table_name> (
 <column_name> <column_type> <column_constraints>...,
 [... ,]
 <table_constraints>,
 [...]
);
```

CREATE TABLE Example

```
CREATE TABLE pub (
  pub_id INT PRIMARY KEY,
  title VARCHAR(16) NOT NULL,
  book_id INT NOT NULL REFERENCES book(book_id)
);
```

By convention, SQL keywords are in ALL CAPS in instructional examples but not when typing.

Note: see pubs-schema.sql and pubs-data.sql for examples of SQL database creation and population commands.

Column Types

Each column, or attribute, is given a data type (domain in the relational model). MySQL has

- Numeric data types,
- String data types, and
- Temporal data types.

Get comprehensive doucmentation at http://dev.mysql.com/doc/refman/5.7/en/data-types.html. We'll cover the most commonly used data types.

Numeric Data Types

- `INT`
- `FLOAT` or `DOUBLE` - IEEE floating point number. Use `DOUBLE` to avoid problems, since MySQL does double-precision calculations.
  - `DOUBLE(5,2)` means a number width of 5 with exactly 2 decimal places
- `DECIMAL` "exact" fixed point decimal. Use for monetary values.
  - `DECIMAL(5,2)` means a number width of 5 with exactly 2 decimal places

String Data Types

- `CHAR` - strings stored with fixed length
- `VARCHAR(M)` - strings stored with variable length, up to `M` characters.
- `TEXT` - large strings
- `ENUM('value1, 'value2', ..., 'valueN')` - enumerated type with `N` possible values whose elements are strings

Note: MySQL allows use of single or double quotes in string literals, but the SQL standard specifies single quotes.

Temporal Data Types

- `DATE` - 'YYY-MM-DD'
- `DATETIME` - 'YYYY-MM-DD HH:MM:SS' - stored in "local time"
- `TIMESTAMP` - 'YYYY-MM-DD HH:MM:SS' - converted to UTC based on client's time zone, converted to local time based on client's time zone
- `TIME` - 'HH:MM:SS' – be sure to include the colons if you abbreviate

See the MySQL reference manual section on date and time types.

Constraints

- Attribute (a.k.a. column) constraints
- Key (a.k.a. unique)
- Primary key
- Foreign key

We'll also learn named constraints, assertions and triggers in Advanced SQL.

Key and Primary Key Constraints

Key:

```
name CHAR(10) UNIQUE,
```

Primary key:

```
pub_id INT PRIMARY KEY,
```

A primary key is implicitly `UNIQUE`

Foreign Key Constratins

```
book_id INT NOT NULL REFERENCES book(book_id)
```

Notice also that we don't allow `book_id` to be `NULL`. So `pub` totally participates in its relationship with `book`.

CHECK Constraints

```
CREATE TABLE bartender (
  id INT PRIMARY KEY,
  name VARCHAR(10) NOT NULL,
  age INT CHECK (age > 20)
);
```

Note: MySQL does not enforce CHECK constraints. We'll learn about triggers in Advanced SQL.

SQL Scripts

Common practice to create scrtipts for creation of a database and insertion of initial data.

dorms-schema.sql:

```
create database dorms;
use dorms;

drop table if exists dorm;
create table dorm (
    dorm_id integer primary key autoincrement,
    name text,
    spaces integer
);
...
```

dorms-data.sql:

```
insert into dorm values(1, 'Armstrong', 124);
...
insert into student values (1, 'Alice', 3.6, 1);
...
```

MySQL Batch Mode

Two ways to run an SQL script:

1. From OS shell:

```
$ mysql -u root < dorms-schema.sql
```

1. From MySQL shell:

```
mysql> source dorms-schema.sql
Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```