Structured Query Language (SQL)

SQL standardization
- A series of standards: ’86, ’89, ’92 (SQL2), ’99 (SQL3), ... 2016
- DBMS products differ in how much they support
- Many implement extra features (extensions)

SQL is a “declarative” command-based language
- In general, this means that you say what you want to happen
- Not how to perform it (the DBMS figures it out)
- commands such as: CREATE, INSERT, SELECT, etc

SQL largely case insensitive
- Various conventions (lowercase, uppercase, camel case, etc.)
- Keywords usually in all uppercase, ids in all lowercase
- Some systems allow for case-sensitive names (at least partially)
CS Database Server Basics

See the db-server info sheet on the course webpage ...

Some prefer to use MySQL workbench

- similar idea, just a GUI “veneer” ... better to learn command line first

Creating Tables in SQL (MySQL/MariaDB)

Basic **create table** syntax:

```sql
CREATE TABLE  tablename (
attribute1 type1,
attribute2 type2,
...
);
```

where attribute; is the name and type; is the data type (e.g., vt_id INT)

Can also add various additional information:

- constraints (e.g., NOT NULL, keys, foreign keys, UNIQUE, etc.)
- default values
- etc.
Basic SQL Data Types (supported by MariaDB)

(1) Integer values:
- INT or INTEGER 4 bytes
- MEDIUMINT 3 bytes (not SQL)
- SMALLINT 2 bytes
- TINYINT 1 byte (not SQL)
- BIGINT 8 bytes (not SQL)
- Also UNSIGNED (e.g., INT UNSIGNED)

Basic SQL Data Types (supported by MariaDB)

(2) Floating-point values:
- FLOAT 4 bytes
- DOUBLE 8 bytes
- FLOAT($M$, $D$) and DOUBLE PRECISION($M$, $D$)
  - values up to $M$ digits of which $D$ may be after the decimal point
  - values (with more precision) are rounded
- Also supports “exact” floating point types (e.g., DECIMAL(8,2))

Basic SQL Data Types (supported by MariaDB)

(3) Boolean and Binary values:
- BOOL or BOOLEAN 0 is false, not 0 is true
- BIT($M$) $M$ binary digits (bits)
Basic SQL Data Types (supported by MariaDB)

(4) String values:
- **CHAR(N)** \(0 \leq N \leq 255\) characters (fixed length)
- **VARCHAR(N)** \(0 \leq N \leq 65,535\) characters (variable length)
- **ENUM(`v1', `v2', ...)** one of given string values
- **BLOB**
  - “binary large object”
  - can store a variable amount of data (variable length)
  - stored as byte strings (no character set)
  - also TINYBLOB, MEDIUMBLOB, LONGBLOB (different max lengths)
- **TEXT** same as blob for character data
- **TINYTEXT** up to 255 characters

(5) Date and Time values:
- **YEAR** YYYY (or `YYYY`) format from 1901 to 2155 (and 0000)
- **DATE** `YYYY-MM-DD` format
- **TIME** multiple formats, e.g., `HH:MM` and `HH:MM:SS`
- **DATETIME** both a date and a time, multiple formats

(6) Others:
- various other string and numeric types
- geometric types
- JSON values
- **SET** types (e.g., `SET(`value1', `value2', ...)`)
Dropping Tables

Removing a table that already exists

DROP TABLE account;

Note: this removes the entire table!

Checking table exists before removing it

DROP TABLE IF EXISTS account;

• only removes table if it has already been created
• good for avoiding errors in a script

Inserting Rows

Basic forms of row insertion ...

INSERT INTO table VALUES (v1, v2, ...);
INSERT INTO table VALUES (v1, v2, ...), (v3, v4, ...), ...;
INSERT INTO table(a1, a2) VALUES (v1, v2);
INSERT INTO table SET a1 = v1, a2 = v2, ...;

For example:

INSERT INTO account VALUES (101, 'Alice', 'Central');

INSERT INTO account VALUES
(102, 'Bob', 'Central'), (103, 'Charlie', 'Shadle');
Table Contents: Basics

To print the contents of the table:

```
SELECT * FROM account;
```

Useful to check that correct insertions made, etc.

Note if table is empty, will print:

```
MariaDB [bowersDB]> SELECT * FROM account;
Empty set (0.001 sec)
```

We’ll go over SELECT in detail later with SQL Queries

Using SQL Script Files

An SQL script file is a text file with SQL commands

- allows for commands to be rerun without having to use shell directly
- often one file for schema, one or more for data
- we'll use script files for all homework assignments

Within MariaDB, use the “source” command to “run” your scripts

- e.g., "MariaDB [bowersDB]> source hw1-schema.sql"
- assumes mysql command run from directory with source file
- can also specify full path to file