Introduction to MySQL
Outline

- How to Manage your data?
- Introduction to MySQL
- Connecting and Disconnecting
- Entering Basic Queries
- Creating and Using a Database
How to Manage your data?

- Access vs. Excel

- In both Access and Excel, you can:
  - Import data from external sources
  - Sort and filter your data.
  - Do calculations to derive the information you want.

- Excel is spreadsheet software, that stores data in columns and rows.
- Access stores data in tables similar to worksheets in Excel

How to Manage your data?

- However, table in Access are designed for complex querying in relation to data stored in other tables.

both programs work well for managing data

How to Manage your data?

- The key to your decision:
  
  _Is the data relational or not?_

- **Flat** or **non-relational** data:
  - data in each column is directly related and need only reside in a single, flat table.

- **Relational** data:
  - Data needs to be stored in more than one table
  - Each table is a description of a type of data
  - For example: customer information in one table and their orders in another table

How to Manage your data?

Use Access when you:

- Require a relational database (multiple tables) to store your data.
- May need to add more tables in the future to an originally flat or non-relational data set.
- Want to run complex queries.

Use Excel when you:

- Require a flat or non-relational view of your data
- Want to run primarily calculations and statistical comparisons on your data
- Know your dataset is manageable in size

MySQL

- MySQL is a very popular, open source database (since 2008).
- MySQL is a *relational* database.
- Officially pronounced “my S-Q-L” (also my sequel).
- MySQL runs as a server and provides multi-user access to a number of databases.
- Handles very large databases; very fast performance.
- Why are we using MySQL?
  - Free (much cheaper than Oracle!)
  - Everyone can install MySQL locally.
When to use MySQL

- MySQL is a much more stable and faster database than Microsoft Access
- Microsoft Access has **2GB** size limitations
- MySQL is free!
Installing MySQL

- MySQL Community Server 5.6.11
  The software can be downloaded from [http://www.mysql.com/downloads/mysql/](http://www.mysql.com/downloads/mysql/)

- Mysql-connector-net-6.2
  The software can be downloaded from [http://www.mysql.com/downloads/connector/net/6.2.html#downloads](http://www.mysql.com/downloads/connector/net/6.2.html#downloads)
User Interface

- MySQL Workbench
  - Database Design & Modeling
  - Database Administration
  - Database Migration

The software can be downloaded from:

http://dev.mysql.com/downloads/tools/workbench/
Download MySQL Community Server

MySQL Community Edition is a freely downloadable version of the world’s most popular open source database that is for enthusiasts.

MySQL Cluster Community Edition is available as a separate download. The reason for this change is so that MySQL C sources of MySQL Cluster Carrier Grade Edition.

Important Platform Support Updates

Generally Available (GA) Releases

MySQL Community Server 5.6.10

Select Platform:

Select Platform... Select

Recommended Download:

MySQL Installer 5.6 for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the server-only MSI packages.

Installation Progress

The following products will be installed or updated.

<table>
<thead>
<tr>
<th>Product</th>
<th>Status</th>
<th>Progress</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL Server 5.6.10</td>
<td></td>
<td>Install</td>
<td>success</td>
</tr>
<tr>
<td>MySQL Workbench CE 5.2.47</td>
<td></td>
<td>Install</td>
<td>success</td>
</tr>
<tr>
<td>MySQL Notifier 1.0.3</td>
<td></td>
<td>Install</td>
<td>success</td>
</tr>
<tr>
<td>MySQL For Excel 1.1.0</td>
<td></td>
<td>Install</td>
<td>success</td>
</tr>
<tr>
<td>Connector/ODBC 5.2.4</td>
<td></td>
<td>Install</td>
<td>success</td>
</tr>
<tr>
<td>Connector/C++ 1.1.2</td>
<td></td>
<td>Install</td>
<td>success</td>
</tr>
<tr>
<td>Connector/J 5.1.23</td>
<td></td>
<td>Install</td>
<td>success</td>
</tr>
<tr>
<td>Connector/.NET 6.6.5</td>
<td></td>
<td>Install</td>
<td>success</td>
</tr>
<tr>
<td>MySQL Documentation 5.6.10</td>
<td></td>
<td>Install</td>
<td>success</td>
</tr>
<tr>
<td>Samples and Examples 5.6.10</td>
<td></td>
<td>Install</td>
<td>success</td>
</tr>
</tbody>
</table>

MySQL Server Configuration

Root Account Password

Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password: *********
Repeat Password: *********
Password Strength: Medium

MySQL User Accounts

Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

<table>
<thead>
<tr>
<th>MySQL Username</th>
<th>Host</th>
<th>User Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>jon</td>
<td>%</td>
<td>Backup Admin</td>
</tr>
</tbody>
</table>

Connecting to MySQL
Connecting to MySQL

[Image of MySQL Workbench interface with options for SQL Development, Data Modeling, and Server Administration.]

- **SQL Development**: Connect to existing databases and run SQL queries, SQL scripts, edit data, and manage database objects.
- **Data Modeling**: Create and manage models, forward & reverse engineer, compare and synchronize schemas, report.
- **Server Administration**: Configure your database server, setup user accounts, browse status variables and server logs.

[Password dialog box showing service, user, and password fields.]
Connecting to MySQL
Basic Queries

- Note that most MySQL commands end with a semicolon (;)
- MySQL returns the total number of rows found, and the total time to execute the query.

```sql
SELECT VERSION(), CURRENT_DATE;
```
Basic Queries

```
1. SELECT VERSION(), CURRENT_DATE;
```

![SQL Query Result](image)
Basic Queries

- Keywords may be entered in any letter-case.
- The following queries are equivalent:
  
  ✓ SELECT VERSION(), CURRENT_DATE;
  ✓ select version(), current_date;
  ✓ SeLeCt vErSiOn(), current_DATE;
  ✓ SeLeCrTy vErSiOn(), current_DATE;
Basic Queries

```
1. SELECT cos(pi()/3);
2. cos(pi()/3)
   0.5000000000000001
```
Creating a Database

[Diagram of MySQL Workbench with steps to create a new schema in the connected server]

[Diagram showing the process of creating a schema including naming and specifying the default collection]
Creating a Database
Creating a Table

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>PK</th>
<th>NN</th>
<th>UQ</th>
<th>BIN</th>
<th>UN</th>
<th>ZF</th>
<th>AI</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>INT(11)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>VARCHAR(45)</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CountyCode</td>
<td>VARCHAR(45)</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Population</td>
<td>INT(11)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data Type: INT(11)

Apply
Creating a Table

```
CREATE TABLE `AWMA_Workshop`.`city` (ni
`ID` INT(11) NOT NULL ,
`Name` VARCHAR(45) NOT NULL ,
`CountyCode` VARCHAR(45) NOT NULL ,
`Population` INT(11) NOT NULL ,
PRIMARY KEY (`ID`));
```
Basic Queries

1. SELECT * FROM AWMA_Workshop.city;
2. 

![MySQL Workbench interface showing a SQL query and result set.](image)
Basic Queries

1. `INSERT INTO AWMA_Workshop.city VALUES (1, "New York", "USA", 8008278);`

2. `SELECT * FROM AWMA_Workshop.city;`
## Loading Data from a TextFile

```plaintext
<table>
<thead>
<tr>
<th></th>
<th>City</th>
<th>Country</th>
<th>Province</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Kabul</td>
<td>AFG</td>
<td>Kabul</td>
<td>1780000</td>
</tr>
<tr>
<td>2</td>
<td>Qandahar</td>
<td>AFG</td>
<td>Qandahar</td>
<td>237500</td>
</tr>
<tr>
<td>3</td>
<td>Herat</td>
<td>AFG</td>
<td>Herat</td>
<td>186800</td>
</tr>
<tr>
<td>4</td>
<td>Mazar-e-Sharif</td>
<td>AFG</td>
<td>Balkh</td>
<td>127800</td>
</tr>
<tr>
<td>5</td>
<td>Amsterdam</td>
<td>NLD</td>
<td>Noord-Holland</td>
<td>731200</td>
</tr>
<tr>
<td>6</td>
<td>Rotterdam</td>
<td>NLD</td>
<td>Zuid-Holland</td>
<td>593321</td>
</tr>
<tr>
<td>7</td>
<td>Haag</td>
<td>NLD</td>
<td>Zuid-Holland</td>
<td>440900</td>
</tr>
<tr>
<td>8</td>
<td>Utrecht</td>
<td>NLD</td>
<td>Utrecht</td>
<td>234323</td>
</tr>
<tr>
<td>9</td>
<td>Eindhoven</td>
<td>NLD</td>
<td>Noord-Brabant</td>
<td>201843</td>
</tr>
<tr>
<td>10</td>
<td>Tilburg</td>
<td>NLD</td>
<td>Noord-Brabant</td>
<td>193238</td>
</tr>
<tr>
<td>11</td>
<td>Groningen</td>
<td>NLD</td>
<td>Groningen</td>
<td>172701</td>
</tr>
<tr>
<td>12</td>
<td>Breda</td>
<td>NLD</td>
<td>Noord-Brabant</td>
<td>160398</td>
</tr>
<tr>
<td>13</td>
<td>Apeldoorn</td>
<td>NLD</td>
<td>Gelderland</td>
<td>153491</td>
</tr>
<tr>
<td>14</td>
<td>Nijmegen</td>
<td>NLD</td>
<td>Gelderland</td>
<td>152463</td>
</tr>
<tr>
<td>15</td>
<td>Enschede</td>
<td>NLD</td>
<td>Overijssel</td>
<td>149544</td>
</tr>
<tr>
<td>16</td>
<td>Haarlem</td>
<td>NLD</td>
<td>Noord-Holland</td>
<td>148772</td>
</tr>
<tr>
<td>17</td>
<td>Almere</td>
<td>NLD</td>
<td>Flevoland</td>
<td>142465</td>
</tr>
<tr>
<td>18</td>
<td>Arnhem</td>
<td>NLD</td>
<td>Gelderland</td>
<td>138020</td>
</tr>
<tr>
<td>19</td>
<td>Zaanstad</td>
<td>NLD</td>
<td>Noord-Holland</td>
<td>135621</td>
</tr>
<tr>
<td>20</td>
<td>Ã´s-Hertogenbosch</td>
<td>NLD</td>
<td>Noord-Brabant</td>
<td>129170</td>
</tr>
<tr>
<td>21</td>
<td>Amersfoort</td>
<td>NLD</td>
<td>Utrecht</td>
<td>126270</td>
</tr>
<tr>
<td>22</td>
<td>Maastricht</td>
<td>NLD</td>
<td>Limburg</td>
<td>122087</td>
</tr>
<tr>
<td>23</td>
<td>Dordrecht</td>
<td>NLD</td>
<td>Zuid-Holland</td>
<td>119811</td>
</tr>
</tbody>
</table>
Loading Data from a TextFile

```sql
1. LOAD DATA LOCAL INFILE 'c:/temp/city.txt'
2. INTO TABLE city
3. FIELDS TERMINATED BY ",";
4. LINES TERMINATED BY "\n";
```

```sql
1. SELECT * FROM awma_workshop.city;
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>CountyCode</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Kabul</td>
<td>AFG</td>
<td>1780000</td>
</tr>
<tr>
<td>2</td>
<td>Qandahar</td>
<td>AFG</td>
<td>237500</td>
</tr>
<tr>
<td>3</td>
<td>Herat</td>
<td>AFG</td>
<td>186800</td>
</tr>
<tr>
<td>4</td>
<td>Mazar-e-Sharf</td>
<td>AFG</td>
<td>127800</td>
</tr>
<tr>
<td>5</td>
<td>Amsterdam</td>
<td>NLD</td>
<td>731200</td>
</tr>
<tr>
<td>6</td>
<td>Rotterdam</td>
<td>NLD</td>
<td>593321</td>
</tr>
<tr>
<td>7</td>
<td>Haag</td>
<td>NLD</td>
<td>440900</td>
</tr>
<tr>
<td>8</td>
<td>Utrecht</td>
<td>NLD</td>
<td>234323</td>
</tr>
<tr>
<td>9</td>
<td>Eindhoven</td>
<td>NLD</td>
<td>201843</td>
</tr>
<tr>
<td>10</td>
<td>Tilburg</td>
<td>NLD</td>
<td>193238</td>
</tr>
</tbody>
</table>
Sending data through MS Access
Sending data through MS Access
Dropping a Table

DROP TABLE AWMA_WORKSHOP.city;
WHERE Clause

```
SELECT Name FROM awma_workshop.city WHERE Population > 8000000;
```
WHERE Clause

```
SELECT Name, population FROM awma_workshop.city WHERE Population > 800000
```
Counting Rows Example

```
1. SELECT count(Name) from awma_workshop.city
2. where city.population > 2000000
```
"LIKE"

```
1. SELECT * from awma_workshop.city
2. WHERE city.Name LIKE "SA%"
```
Counting Rows Example

1. SELECT count(*) FROM awma_workshop.city;

2. SELECT count(*) FROM awma_workshop.city WHERE city.Name LIKE "SA%";

Result: 4079

Result: 210
Counting number of cities with population larger than 20,000

```
1. SELECT count(*) FROM awma_workshop.city;
```

```
1. SELECT count(*) FROM awma_workshop.city WHERE Population > 20000;
```
Country Details

1. SELECT * FROM awma_workshop.country;

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Continent</th>
<th>Region</th>
<th>SurfaceArea</th>
<th>IndepYear</th>
<th>Population</th>
<th>LifeExpectancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABW</td>
<td>Aruba</td>
<td>North America</td>
<td>Caribbean</td>
<td>193</td>
<td>NULL</td>
<td>103000</td>
<td>78.4</td>
</tr>
<tr>
<td>AFG</td>
<td>Afghanistan</td>
<td>Asia</td>
<td>Southern and Central Asia</td>
<td>652090</td>
<td>1919</td>
<td>22720000</td>
<td>45.9</td>
</tr>
<tr>
<td>AGO</td>
<td>Angola</td>
<td>Africa</td>
<td>Central Africa</td>
<td>1246700</td>
<td>1975</td>
<td>12878000</td>
<td>38.3</td>
</tr>
<tr>
<td>AIA</td>
<td>Anguilla</td>
<td>North America</td>
<td>Caribbean</td>
<td>96</td>
<td>NULL</td>
<td>8000</td>
<td>76.1</td>
</tr>
<tr>
<td>ALB</td>
<td>Abania</td>
<td>Europe</td>
<td>Southern Europe</td>
<td>28748</td>
<td>1912</td>
<td>34012000</td>
<td>71.6</td>
</tr>
<tr>
<td>AND</td>
<td>Andorra</td>
<td>Europe</td>
<td>Southern Europe</td>
<td>468</td>
<td>1278</td>
<td>78000</td>
<td>83.5</td>
</tr>
<tr>
<td>ANT</td>
<td>Netherlands Antilles</td>
<td>North America</td>
<td>Caribbean</td>
<td>800</td>
<td>NULL</td>
<td>217000</td>
<td>74.7</td>
</tr>
<tr>
<td>ARE</td>
<td>United Arab Emirates</td>
<td>Asia</td>
<td>Middle East</td>
<td>83600</td>
<td>1971</td>
<td>24410000</td>
<td>74.1</td>
</tr>
<tr>
<td>ARG</td>
<td>Argentina</td>
<td>South America</td>
<td>South America</td>
<td>2780400</td>
<td>1816</td>
<td>37032000</td>
<td>75.1</td>
</tr>
<tr>
<td>ARM</td>
<td>Amercia</td>
<td>Asia</td>
<td>Middle East</td>
<td>29800</td>
<td>1991</td>
<td>35200000</td>
<td>66.4</td>
</tr>
<tr>
<td>ASM</td>
<td>American Samoa</td>
<td>Oceania</td>
<td>Polynesia</td>
<td>199</td>
<td>NULL</td>
<td>68000</td>
<td>75.1</td>
</tr>
<tr>
<td>ATA</td>
<td>Antarctica</td>
<td>Antarctica</td>
<td>Antarctica</td>
<td>13120000</td>
<td>NULL</td>
<td>0</td>
<td>NULL</td>
</tr>
<tr>
<td>ATF</td>
<td>French Southern territories</td>
<td>Antarctica</td>
<td>Antarctica</td>
<td>7780</td>
<td>NULL</td>
<td>0</td>
<td>NULL</td>
</tr>
<tr>
<td>ATG</td>
<td>Antiqua and Barbuda</td>
<td>North America</td>
<td>Caribbean</td>
<td>442</td>
<td>1981</td>
<td>68000</td>
<td>70.5</td>
</tr>
</tbody>
</table>
Country Language

1. SELECT * FROM awma_workshop.countrylanguage;
Selecting Distinct **Language**

```
SELECT DISTINCT language FROM awma_workshop.countrylanguage;
```
Joining two table
Joining two table

```
SELECT country.Name, countrylanguage.Language, country.Continent, countrylanguage.Percentage
FROM countrylanguage INNER JOIN country ON countrylanguage.CountryCode = country.Code;
```
Joining two tables

```sql
SELECT country.Name, 
countrylanguage.Language, 
country.Continent, 
countrylanguage.Percentage
FROM countrylanguage INNER JOIN country ON countrylanguage.CountryCode = country.Code;
```
Joining two table using

```
SELECT a.Name, b.language, b.percentage from
city as a inner join countrylanguage as b USING (CountryCode);
```