

Introduction to SQL

(emphasis on MySQL)

Lukas Hoffmann, 11/20/17



Preliminaries

- Lecture shamelessly plagiarized from...
 - <http://www.math.ucla.edu/~virtanen/40a.1.15s/schedule/lec21.pdf>
 - [miageprojet2.unice.fr/@api/deki/files/1830/=Intro to MySQL.ppt](http://miageprojet2.unice.fr/@api/deki/files/1830/=Intro_to_MySQL.ppt)
 - With minor edits
- Lukas wrote last slide: very useful links to explain “JOIN”.
- **Murach’s MySQL** textbook & **MySQL Workbench** good for learning
<https://www.murach.com/shop/murach-s-mysql-detail>

What is a database?

A database is a way of storing data.

Databases are specifically designed to be very efficient in storing and retrieving data.

The most common way of structuring data in a database is relational database system.

Relational databases

- A relational database is a structured collection of tables.
- Each table consists of rows called records.
- Columns of the table have keys called fields. Each cell contains data.
- Most tables have a special column that identifies the rows of the table. The values in this column are called **primary keys**.

Relational database table example

Fields

Primary key

	IP_address	num_visits	timestamp
Record 1	123.25.67	7	1291 193
Record 2	98.104.22	3	13056782

A more complex relational database

Vette_id	Body_style	Miles	Year	State	Equip
1	coupe	18.0	1997	Arkansas	Automatic,leather , CD
2	hatchback	58.0	1996	Connecticut	Automatic,leather , CD
3	convertible	13.5	2001	Alabama	Automatic,leather
4	hatchback	19.5	1995	Alaska	4-speed,leather
5	hatchback	25.0	1991	California	Automatic,leather
6	hardtop	15.0	2000	Alaska	4-speed
7	coupe	55.0	1979	Georgia	6-speed,leather
8	convertible	17.0	1999	California	6-speed,leather,CD
9	hardtop	17.0	2000	California	6-speed,leather,CD
10	hatchback	50.0	1995	Connecticut	Automatic,CD

Corvettes table

Example continued

Simplification: Move some data to a new table. To accomplish this we create a separate equipment table.

Equip_id	Equip
1	Automatic
2	4-speed
3	5-speed
4	6-speed
5	CD
6	Leather

Equipment table

Example continued.

Next we need a way to indicate which cars have what equipment. To accomplish this we create a cross reference table. To emphasize that this table ties the Corvettes table to equipment table we call it `Corvettes_Equipment`.

Vette_id	Equip
1	1
1	5
1	6
2	1
2	5
2	6
3	1
3	6
4	2
4	6

Example continued

A further simplification is to code the state names with a number.

State_ID	State
1	Alabama
2	Alaska
3	Arizona
4	Arkansas
5	California
6	Colorado
7	Connecticut
8	Delaware
9	Florida
10	Georgia

Example continued

What we have accomplished:

Primary key

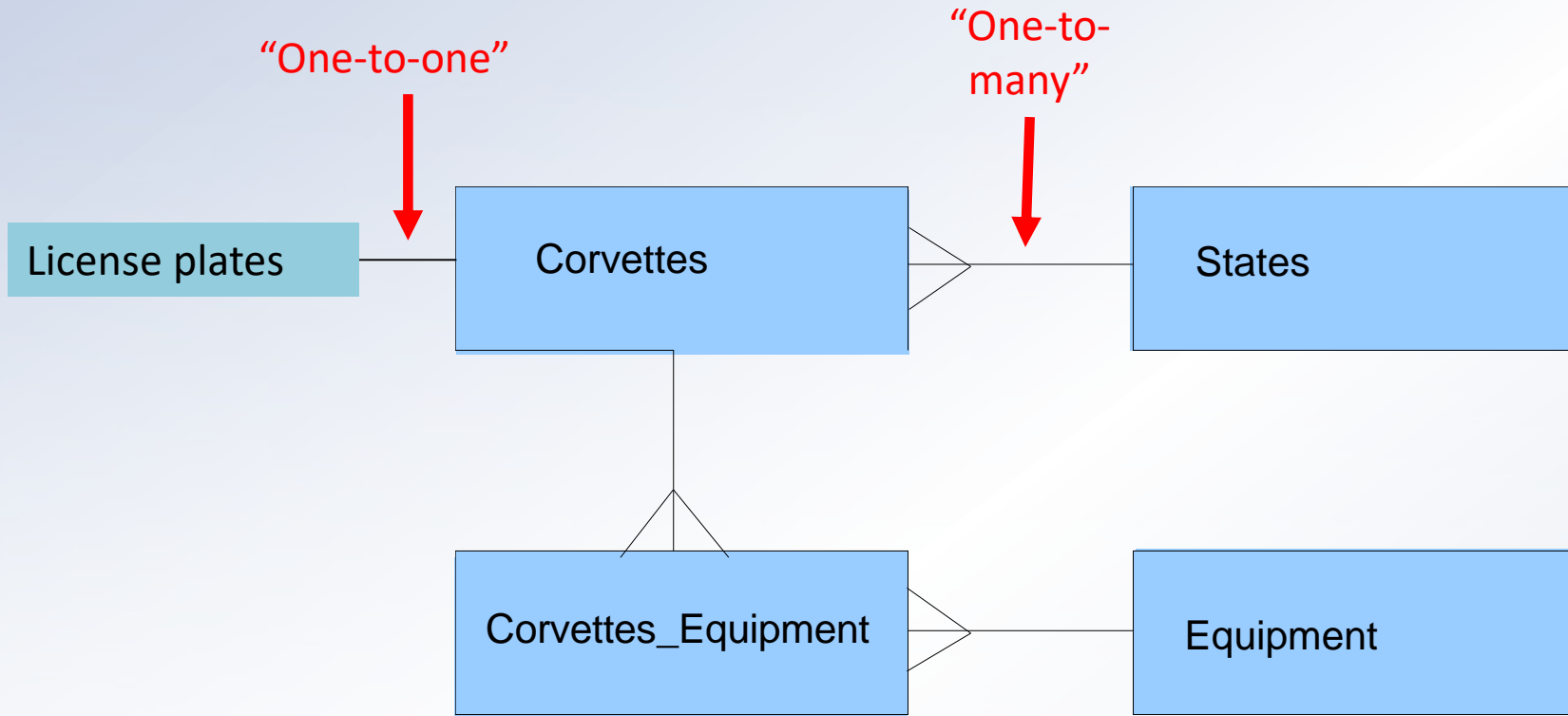


A diagram illustrating a primary key relationship. A red arrow labeled "Primary key" points to the first cell of the first row (Vette_id: 1). Another red arrow labeled "Foreign key" points to the last cell of the first row (State: 4). The table below shows the data for 10 vehicles, with Vette_id as the primary key and State as the foreign key.

Vette_id	Body_style	Miles	Year	State
1	coupe	18.0	1997	4
2	hatchback	58.0	1996	7
3	convertible	13.5	2001	1
4	hatchback	19.5	1995	2
5	hatchback	25.0	1991	5
6	hardtop	15.0	2000	2
7	coupe	55.0	1979	10
8	convertible	17.0	1999	5
9	hardtop	17.0	2000	5
10	hatchback	50.0	1995	7

Foreign key

Example continued



Had we not used a cross reference table the relationship between Corvettes and Equipment would have been **many to many**.

What is SQL?

SQL stands for **S**tructured **Q**uery **L**anguage.

It is a standard language developed for accessing and modifying relational databases.

SQL in turn is used by a database management system.
Some common database management systems are:

- MySQL
- SQLite
- PostgreSQL (*Post Ingres SQL*) more functionality →
- Oracle
- Microsoft SQL Server



Using a Database

- To get started on your own database, first check which databases currently exist.
- Use the SHOW statement to find out which databases currently exist on the server:

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| mysql    |  
| test     |  
+-----+  
2 rows in set (0.01 sec)
```

Using a Database

- To create a new database, issue the “create database” command:
 - `mysql> create database webdb;`
- To select a database, issue the “use” command:
 - `mysql> use webdb;`

Creating a Table

- Once you have selected a database, you can view all database tables:

```
mysql> show tables;
```

```
Empty set (0.02 sec)
```

- An empty set indicates that I have not created any tables yet.

Creating a Table

- Let's create a table for storing pets.

- Table: pets

- name: VARCHAR(20)
- owner: VARCHAR(20)
- species: VARCHAR(20)
- sex: CHAR(1)
- birth: DATE
- date: DATE

Other MySQL datatypes

Integer

Float

Time

Large object (binary/char)

Enum ('yes' or 'no')

Set ('mushrooms, anchovies')

Creating a Table

- To create a table, use the CREATE TABLE command:

```
mysql> CREATE TABLE pet (  
    -> name VARCHAR(20),  
    -> owner VARCHAR(20),  
    -> species VARCHAR(20),  
    -> sex CHAR(1),  
    -> birth DATE, death DATE);  
Query OK, 0 rows affected (0.04 sec)
```

Showing Tables

- To verify that the table has been created:

```
mysql> show tables;
```

```
+-----+
```

```
| Tables_in_test |
```

```
+-----+
```

```
| pet            |
```

```
+-----+
```

```
1 row in set (0.01 sec)
```

Describing Tables

- To view a table structure, use the DESCRIBE command:

```
mysql> describe pet;
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	YES		NULL	
owner	varchar(20)	YES		NULL	
species	varchar(20)	YES		NULL	
sex	char(1)	YES		NULL	
birth	date	YES		NULL	
death	date	YES		NULL	

```
6 rows in set (0.02 sec)
```

Deleting a Table

- To delete an entire table, use the DROP TABLE command:

```
mysql> drop table pet;
```

```
Query OK, 0 rows affected (0.02 sec)
```

Loading Data

- Use the INSERT statement to enter data into a table.
- For example:

```
INSERT INTO pet VALUES ('Fluffy', 'Harold', 'cat', 'f',  
    '1999-02-04', NULL);
```

- The next slide shows a full set of sample data.

More data...

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	
Claws	Gwen	cat	m	1994-03-17	
Buffy	Harold	dog	f	1989-05-13	
Fang	Benny	dog	m	1990-08-27	
Bowser	Diane	dog	m	1998-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	
Whistler	Gwen	bird		1997-12-09	
Slim	Benny	snake	m	1996-04-29	

Loading Sample Data

- You could create a text file `pet.txt' containing one record per line.
- Values must be separated by tabs, and given in the order in which the columns were listed in the CREATE TABLE statement.
- Then load the data via the LOAD DATA Command.

Sample Data File

Fluffy	Harold	cat	f	1993-02-04	\N
Claws	Gwen	cat	m	1994-03-17	\N
Buffy	Harold	dog	f	1989-05-13	\N
Fang	Benny	dog	m	1990-08-27	\N
Bowser	Diane	dog	m	1979-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	\N
Whistler	Gwen	bird	\N	1997-12-09	\N
Slim	Benny	snake	m	1996-04-29	\N

To Load pet.txt:

```
mysql> LOAD DATA LOCAL INFILE "pet.txt" INTO TABLE pet;
```


For each of the examples,
assume the following set of data.

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	
Claws	Gwen	cat	m	1994-03-17	
Buffy	Harold	dog	f	1989-05-13	
Fang	Benny	dog	m	1990-08-27	
Bowser	Diane	dog	m	1998-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	
Whistler	Gwen	bird		1997-12-09	
Slim	Benny	snake	m	1996-04-29	

SQL Select

- The SELECT statement is used to pull information from a table.
- The general format is:

```
SELECT what_to_select  
FROM which_table  
WHERE conditions_to_satisfy
```

Selecting All Data

- The simplest form of SELECT retrieves everything from a table

```
mysql> select * from pet;
```

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1999-02-04	NULL
Claws	Gwen	cat	f	1994-03-17	NULL
Buffy	Harold	dog	f	1989-05-13	NULL
Fang	Benny	dog	m	1999-08-27	NULL
Bowser	Diane	dog	m	1998-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	NULL
Whistler	Gwen	bird		1997-12-09	NULL
Slim	Benny	snake	m	1996-04-29	NULL

```
8 rows in set (0.00 sec)
```

Selecting Particular Rows

- You can select only particular rows from your table.
- For example, if you want to verify the change that you made to Bowser's birth date, select Bowser's record like this:

```
mysql> SELECT * FROM pet WHERE name = "Bowser";
+-----+-----+-----+-----+-----+-----+
| name   | owner | species | sex  | birth       | death       |
+-----+-----+-----+-----+-----+-----+
| Bowser | Diane | dog      | m    | 1998-08-31  | 1995-07-29  |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Selecting Particular Rows

- To find all animals born after 1998

```
SELECT * FROM pet WHERE birth >= "1998-1-1";
```

- To find all female dogs, use a logical AND

```
SELECT * FROM pet WHERE species = "dog" AND sex = "f";
```

- To find all snakes or birds, use a logical OR

```
SELECT * FROM pet WHERE species = "snake"  
OR species = "bird";
```

Selecting Particular Columns

- If you don't want to see entire rows from your table, just name the columns in which you are interested, separated by commas.
- For example, if you want to know when your pets were born, select the name and birth columns.
- (see example next slide.)

Selecting Particular Columns

```
mysql> select name, birth from pet;
```

name	birth
Fluffy	1999-02-04
Claws	1994-03-17
Buffy	1989-05-13
Fang	1999-08-27
Bowser	1998-08-31
Chirpy	1998-09-11
Whistler	1997-12-09
Slim	1996-04-29

8 rows in set (0.01 sec)

UPDATE

Modify an existing record in the database

Syntax:

```
UPDATE tablename SET fieldname = value WHERE  
condition;
```

Example:

```
UPDATE students SET GPA ='3.8' WHERE person =  
'Joe';
```


DELETE

Removes an existing record in a database

Syntax:

```
DELETE FROM tablename WHERE condition;
```

Example:

```
DELETE FROM students WHERE name="Fry";
```

Summary

- SQL provides a structured language for querying/updating multiple databases.
- The more you know SQL, the better.
- The most important part of SQL is learning to retrieve data.
 - selecting rows, columns, boolean operators, pattern matching, etc.

Extra credit: Look up “JOIN”

Select data from multiple tables, usually with primary/foreign key relationships.

JOIN was trickiest for me to understand.

“A JOIN is really a Cartesian product with a filter.” – understand this sentence & understand how JOINS work.

Don't trust Venn diagrams of JOINS. Better explanations:

<https://blog.jooq.org/2016/07/05/say-no-to-venn-diagrams-when-explaining-joins/>

<http://datavirtualizer.com/sql-joins-visualized-in-a-surprising-way/>