

# Practical MySQL Tables

Class 35

## Creating a Table

- in a previous class, I showed creating a simple table from the command line

```
create table pet
(
  id int unsigned not null auto_increment,
  name varchar(255),
  breed varchar(255),
  sex enum('F', 'M', 'U'),
  birth date
);
```

- this works, but is too simplistic
- real-world code needs some enhancements

# Charset

- a character set (charset) is a set of symbols and encodings
- suppose that we have an alphabet with four letters: A, B, a, and b
- we give each letter a number:  $A = 0$ ,  $B = 1$ ,  $a = 2$ , and  $b = 3$
- A is a symbol, 0 is the **encoding** for A
- the combination of all four letters and their encodings is a **character set**
- you are most familiar with the ASCII character set:  $A = 65$ ,  $a = 97$ , etc.

# Collation

- a collation is a set of rules for comparing characters (and strings made of characters) in a character set
- what is the value of "A" < "B"?
- the simplest way is to directly use the encodings
- since the encoding for character A is numerically less than the encoding for character B, we say the string "A" is less than the string "B"
- this simplest of all possible collations is called the binary collation

## More Complex Collations

- but what if we want to say that the lowercase and uppercase letters are equivalent, so we get case-insensitive comparisons?
- then we need at least two rules
  1. treat the lowercase letters a and b as equivalent to A and B
  2. then compare the encodings
- this a case-insensitive collation

## More Complex Collations

- in real life, character sets have many characters along with many special symbols and punctuation marks
- for example, a German collation may need Ö and Æ to be considered the same letter
- every defined charset has a default collation, and most have several available collations
- trying to compare a string using one collation to a string using a different collation in the same charset doesn't work, much less strings from two different charsets

## Charset and Collation

- you can specify charsets and collations per-column in a single table, but that's typically bad design
- typically an entire database has a single charset and collation
- sometimes the database's default charset and collation are set by the DB admin, and you need different ones for your tables
- your databases on borax have the default charset utf8mb4 (full 4-byte utf8) and the default collation utf8mb4\_0900\_ai\_ci (case-insensitive)
- this matches the HTML charset we have been using all along:  
`<meta charset="utf-8" />`  
and the connect command in PHP:

```
$db = new PDO(
    "mysql:host=$db_host;dbname=world;charset=utf8mb4",
    etc.
```

# Full Create Table

the contents of create\_tables.sql:

```
use jbeck;
```

```
drop table if exists pet;
```

```
create table pet
```

```
(  
  id int unsigned not null auto_increment,  
  name varchar(255) not null,  
  breed varchar(255) not null default '',  
  sex enum('F', 'M', 'U') not null default 'U',  
  birth date  
) engine=InnoDB default charset=utf8mb4 collate=utf8mb4_0900_ai_ci;
```



# Foreign Keys

- earlier we saw commands for a foreign key: a field that references a primary key in another table
- the other table must exist before you can create a table that references its primary key in a foreign key field
- when inserting data, the referenced foreign key value must exist before a row that references it can be inserted

## Populating the Tables

- now that the table has been created, you need to put data in it
- you have a table for words
- you have a file with words and parts of speech
- mysql has a command for loading a table with contents of a file
- but it is so difficult to use it's hopeless
- instead we write a program to load the data

# A Script

to reproducibly create a clean database:

- drop tables (if they exist) that have foreign keys into another table
- drop tables (if they exist) with primary keys
- create tables with primary keys
- populate tables with primary keys
- create tables that have foreign keys into another table
- populate the foreign-key containing tables