

# Relational Databases

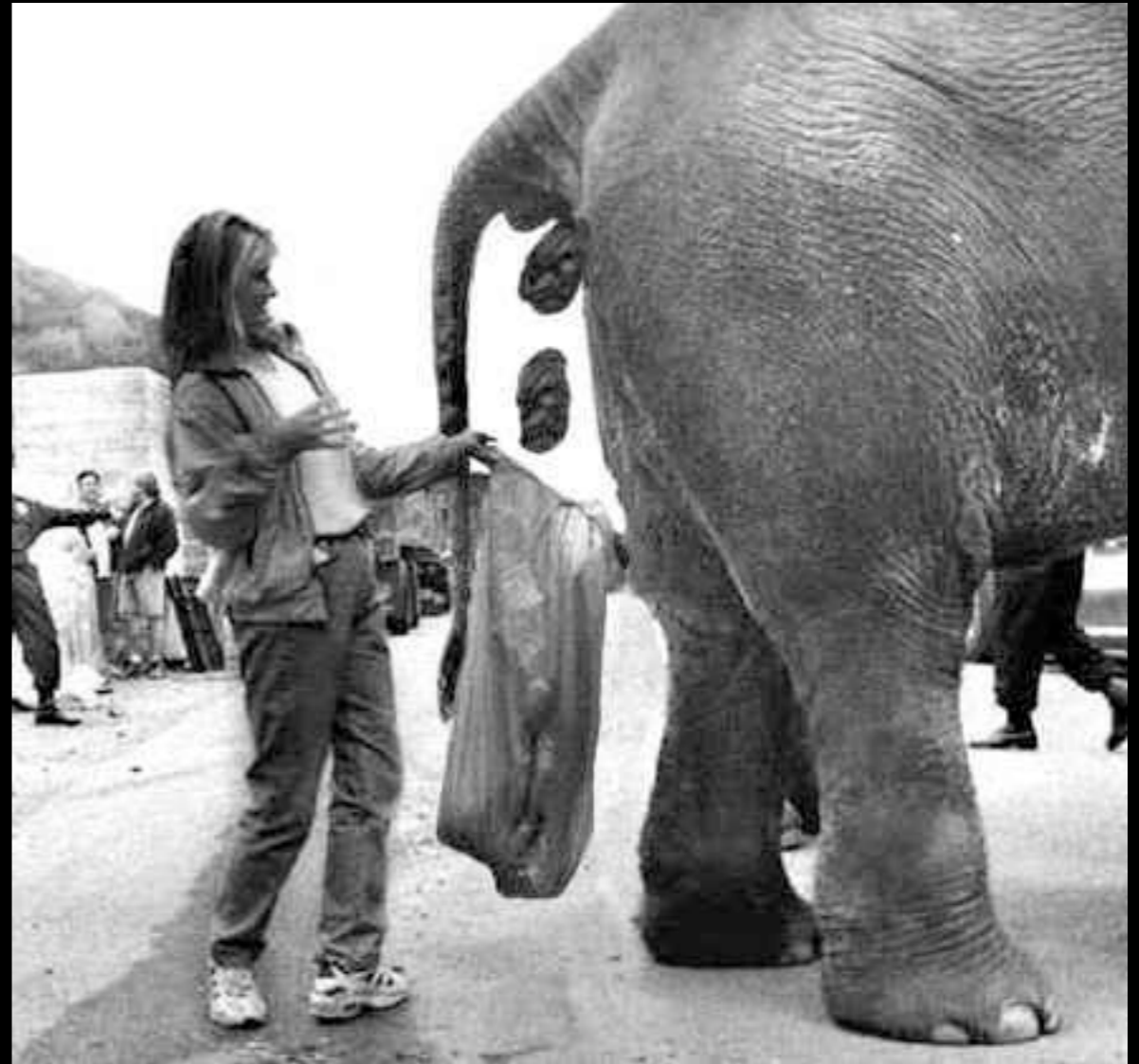
Multimedia Design 2  
Week 11

# Relational Databases

1. CRUD
2. SQL concepts
3. CRUD in SQL
4. Table joins
5. Tweeter

# 1. CRUD

- Create
- Read
- Update
- Delete



# Create

- Instantiates a new data object
- You specify values for the object
- Some values are assigned by default

# Read

- You *query* for data objects
- You get a *set* of results
- A set can have zero or more objects

# Update

- You *query* for a set of objects and assign new values for them
- The new values overwrite existing ones
- Values you didn't specify are unchanged

# Delete

- You *query* for a set of objects that are deleted

# 2. SQL concepts

- Columns & rows
- Tables
- Databases



# Columns & rows

| Performer         | Role                      | Position   | Rank                    |
|-------------------|---------------------------|--|-------------------------|
| William Shatner   | James T. Kirk             | Commanding officer                                     | Captain                 |
| Leonard Nimoy     | Spock                     | First officer<br>Science officer                       | Lt. Cmdr /<br>Commander |
| DeForest Kelley   | Leonard "Bones" McCoy     | Chief medical officer                                  | Lieutenant Commander    |
| James Doohan      | Montgomery "Scotty" Scott | Chief engineer<br>Second officer                       | Lieutenant Commander    |
| Nichelle Nichols  | Nyota Uhura               | Communications officer                                 | Lieutenant              |
| George Takei      | Hikaru Sulu               | Helmsman<br>Weapons<br>Staff physicist (pilot episode) | Lieutenant              |
| Walter Koenig     | Pavel Chekov              | Navigator  | Ensign                  |
| Grace Lee Whitney | Janice Rand               | Captain's yeoman                                       | Yeoman                  |
| Majel Barrett     | Christine Chapel          | Head nurse   | Lieutenant              |

# Columns & rows

One row

| Performer         | Role                      | Position   | Rank                    |
|-------------------|---------------------------|--|-------------------------|
| William Shatner   | James T. Kirk             | Commanding officer                                     | Captain                 |
| Leonard Nimoy     | Spock                     | First officer<br>Science officer                       | Lt. Cmdr /<br>Commander |
| DeForest Kelley   | Leonard "Bones" McCoy     | Chief medical officer                                  | Lieutenant Commander    |
| James Doohan      | Montgomery "Scotty" Scott | Chief engineer<br>Second officer                       | Lieutenant Commander    |
| Nichelle Nichols  | Nyota Uhura               | Communications officer                                 | Lieutenant              |
| George Takei      | Hikaru Sulu               | Helmsman<br>Weapons<br>Staff physicist (pilot episode) | Lieutenant              |
| Walter Koenig     | Pavel Chekov              | Navigator  | Ensign                  |
| Grace Lee Whitney | Janice Rand               | Captain's yeoman                                       | Yeoman                  |
| Majel Barrett     | Christine Chapel          | Head nurse   | Lieutenant              |

# Columns & rows

| Performer         |                           | Position   | Rank                    |
|-------------------|---------------------------|--|-------------------------|
| William Shatner   | Jarvis                    | standing officer                                       | Captain                 |
| Leonard Nimoy     | Spock                     | First officer<br>Science officer                       | Lt. Cmdr /<br>Commander |
| DeForest Kelley   | Leonard "Bones" McCoy     | Chief medical officer                                  | Lieutenant Commander    |
| James Doohan      | Montgomery "Scotty" Scott | Chief engineer<br>Second officer                       | Lieutenant Commander    |
| Nichelle Nichols  | Nyota Uhura               | Communications officer                                 | Lieutenant              |
| George Takei      | Hikaru Sulu               | Helmsman<br>Weapons<br>Staff physicist (pilot episode) | Lieutenant              |
| Walter Koenig     | Pavel Chekov              | Navigator  | Ensign                  |
| Grace Lee Whitney | Janice Rand               | Captain's yeoman                                       | Yeoman                  |
| Majel Barrett     | Christine Chapel          | Head nurse   | Lieutenant              |

# Columns & rows

A set of results

| Performer         | Role                      | Position   | Rank                    |
|-------------------|---------------------------|--|-------------------------|
| William Shatner   | James T. Kirk             | Commanding officer                                     | Captain                 |
| Leonard Nimoy     | Spock                     | First officer<br>Science officer                       | Lt. Cmdr /<br>Commander |
| DeForest Kelley   | Leonard "Bones" McCoy     | Chief medical officer                                  | Lieutenant Commander    |
| James Doohan      | Montgomery "Scotty" Scott | Chief engineer<br>Second officer                       | Lieutenant Commander    |
| Nichelle Nichols  | Nyota Uhura               | Communications officer                                 | Lieutenant              |
| George Takei      | Hikaru Sulu               | Helmsman<br>Weapons<br>Staff physicist (pilot episode) | Lieutenant              |
| Walter Koenig     | Pavel Chekov              | Navigator  | Ensign                  |
| Grace Lee Whitney | Janice Rand               | Captain's yeoman                                       | Yeoman                  |
| Majel Barrett     | Christine Chapel          | Head nurse   | Lieutenant              |

# Tables

- A table is a set of columns and rows
- Each column has a *type*
  - Integer
  - String
  - Date
  - etc.

# Databases

- A database is a set of tables
- Each table must have a *unique name*
- Each table also has a unique set of columns and rows

# 3. CRUD in SQL

- **INSERT** is the *create* command
- **SELECT** is the *read* command
- **UPDATE** is the *update* command
- **DELETE** is the *delete* command

# Using INSERT

```
INSERT INTO table  
(column, column, ...)  
VALUES (value, value, ...);
```

# Using INSERT

Which **table** to create a new row in

```
INSERT INTO actor  
(column, column, ...)  
VALUES (value, value, ...);
```

# Using INSERT

```
INSERT INTO actor  
(performer, role, position, rank)  
VALUES (value, value, ...);
```

Which **columns** to  
initialize the values of

# Using INSERT

```
INSERT INTO actor
(performer, role, position, rank)
VALUES (
    'William Shatner',
    'James T. Kirk',
    'Commanding officer',
    'Captain'
);
```

Which values to assign

# Using INSERT

```
INSERT INTO actor
(performer, role, position, rank)
VALUES (
    'William Shatner',
    'James T. Kirk',
    'Commanding officer',
    'Captain'
);
```

Initialize the  
performer value

# Using INSERT

```
INSERT INTO actor
(performer, role, position, rank)
VALUES (
    'William Shatner',
    'James T. Kirk',
    'Commanding officer',
    'Captain'
);
```

Initialize the role value

# Using INSERT

```
INSERT INTO actor
(performer, role, position, rank)
VALUES (
    'William Shatner',
    'James T. Kirk',
    'Commanding officer',
    'Captain'
);
```

Initialize the position  
value

# Using INSERT

```
INSERT INTO actor
(performer, role, position, rank)
VALUES (
    'William Shatner',
    'James T. Kirk',
    'Commanding officer',
    'Captain'
);
```

Initialize the rank value

# Using SELECT

```
SELECT column, column, ...  
FROM table  
WHERE column = value;
```

# Using SELECT

```
SELECT *  
FROM table  
WHERE column = value;
```

**What** you want to select  
(\* selects all columns)

# Using SELECT

```
SELECT *  
FROM actor  
WHERE column = value;
```

**Where** you want to select  
from (a table name)

# Using SELECT

```
SELECT *  
FROM actor  
WHERE role = "Spock";
```

Get a specific row

# Using SELECT

```
SELECT *  
FROM actor  
WHERE role <> "Spock";
```

Get everything *except* a specific row (<> means **not equal to**)

# Using SELECT

```
SELECT *  
FROM actor  
WHERE rank = "Lieutenant";
```

Get a set of results that  
have **rank** in common

# Using SELECT

```
SELECT *  
FROM actor  
WHERE rank = "Lieutenant"  
ORDER BY performer;
```

Get a set of results,  
sorted by **performer**

# Using UPDATE

```
UPDATE table
SET column = value,
    column = value,
    ...
WHERE column = value;
```

# Using UPDATE

Which **table** to update a row on

```
UPDATE actor
SET column = value,
    column = value,
    ...
WHERE column = value;
```

# Using UPDATE

Which **columns** to update  
and their new **values**

```
UPDATE actor  
SET position = 'Helmsman, Weapons'  
WHERE column = value;
```

# Using UPDATE

```
UPDATE actor  
SET position = 'Helmsman, Weapons'  
WHERE role LIKE '%sulu%';
```

Search for values that match a specific **pattern** (% is a wildcard)

# Using DELETE

```
DELETE FROM table  
WHERE column = value;
```

# Using DELETE

Which table to **delete** from

```
DELETE FROM actor  
WHERE column = value;
```

# Using DELETE

```
DELETE FROM actor;
```

Leaving out the WHERE  
clause deletes **everything**

# Escaping strings

- In SQL, like many languages, strings begin and end with ' '
- Things can go screwy when you try to store or retrieve values like '0'Neil'
- Use `mysql_real_escape_string()`

# Escaping strings

```
<?php

$content = $_POST['content'];
$content = mysql_real_escape_string($content);
$now = date('Y-m-d H:i:s');

$sql = "
    INSERT INTO mytable
    (content, posted)
    VALUES ('$content', '$now')
";

?>
```

# 4. Table joins

- Probably the most important feature of relational databases
- Allows you to associate data between tables
- We'll look at a simple two table join

# Primary keys

- Set aside one column for identifying each row
- Each primary key value must be unique
- Common practice is to name this column `id`

# External keys

- Whenever you store the `id` of a row in another table, it creates a **relation** between the two tables
- This value, referred from the other table, is called an **external key**

# Table joins

```
SELECT table.column,  
       table.column,  
       ...  
FROM table, table  
WHERE table.column = table.column;
```

# Problem

- Each actor in Star Trek has one or more position
- Sulu is assigned positions Helmsman, Weapons *and* Staff physicist
- We shouldn't put all those values in one column

# Table joins

Which **columns** to select

```
SELECT actor.role,  
       position.title  
FROM table, table  
WHERE table.column = table.column;
```

# Table joins

Which **tables** to join from

```
SELECT actor.role,  
       position.title  
FROM actor, position  
WHERE table.column = table.column;
```

# Table joins

```
SELECT actor.role,  
       position.title  
FROM actor, position  
WHERE position.actor_id = actor.id;
```

How the tables are **related**

# 5. Tweeter

- Connect to the database
- Initialize a table
- Select a set of rows
- **Insert a row**

# database.php

```
<?php
```

```
// Store configuration separately
```

```
include 'config.php';
```

```
// Connect to the server
```

```
$dbh = mysql_connect($server, $user, $password);
```

```
mysql_select_db($dbname);
```

```
?>
```

# config.php

```
<?php
```

```
$server = 'db.ccnymmd2.com';
```

```
$user = 'your username';
```

```
$password = 'your username';
```

```
$dbname = 'your username';
```

```
?>
```

# initialize.php

```
<?php
```

```
include 'database.php';
```

```
$sql = file_get_contents('initialize.sql');  
mysql_query($sql);
```

```
?>
```

# initialize.sql

```
CREATE TABLE tweeter (  
    id INTEGER AUTO_INCREMENT PRIMARY KEY,  
    content VARCHAR(140),  
    posted DATETIME  
);
```

# index.php [1]

```
<?php

include 'database.php';
include 'form.php';

$sql = "
    SELECT *
    FROM tweeter
    ORDER BY posted DESC
";
$twits = mysql_query($sql);

...
```

# index.php [2]

```
while ($twit = mysql_fetch_object($twits)) {  
    echo "  
        <p>  
            $twit->content<br />  
            Posted $twit->posted  
        </p>  
    ";  
}  
?>
```

# form.php

```
<form action="post.php" method="post">  
  <textarea name="content" rows="4" cols="25">  
  </textarea><br />  
  <input type="submit" value="Post a twit" />  
</form>
```

# post.php

```
<?php
```

```
echo "Implement me!";
```

```
?>
```