# File Permissions and Entity Relationship Diagramming

# File Permissions

# Accessing files

- Sometimes sorting data is a database is overkill
- PHP data is static data (unchangeable outside of the code)
- data files are a good alternative for changing data that is not robust enough for a database

# Reading from a file example!

# Writing to a file example!

# Checking file permissions


**ls** list directory contents

**-l** long format will list file permissions, size, user owner, group owner, etc.

# Examples of permissions

Directory


File

| - | rwx | r-x | r-- |
| --- | --- | --- | --- |
| directory marker<br>there will be a "d" if the file is a directory | user | group | other |

| r | w | x |
| --- | --- | --- |
| read | write | execute |

# Changing the file permissions

```
chmod 666 count.txt
```

**chmod** change file modes

**- - -** three numbers that will determine the levels of file access (done using binary numbers)

**count.txt** the file whose permissions you are changing

# Quick Binary Lesson

# What is a binary number?

Binary numbers are the expression of a number in the base-2 (binary) numeral system.

Uses 0s and 1s only

The placement of the 0s and 1s determine the number

# Binary layout

| 4 | 2 | 1 |
|---|---|---|

# Example of how this works

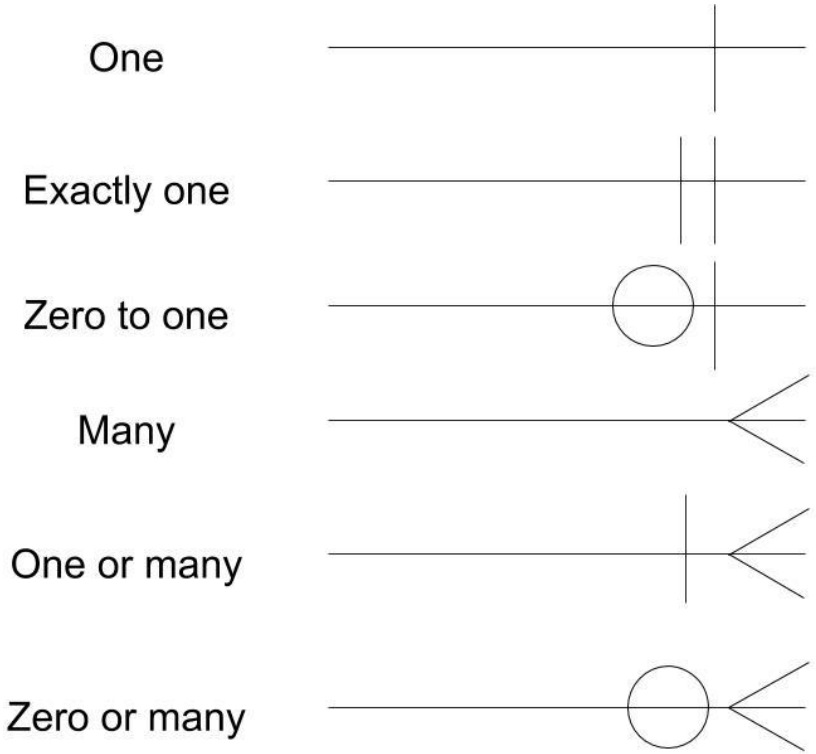
| 4 | 2 | 1 |
|---|---|---|
| r | w | x |
| 1 | 1 | 0 |

$$4 + 2 + 0 = 6$$

# Break time!

# Entity Relationship (ER) Diagramming

# Representing Relationships



# Professors table

| ProfID | Name | Course |
|---|---|---|
| 1 | Allen | Database Management |
| 2 | Gunn | Database Management |
| 3 | Riley | 3-D Printing |
| 4 | Hayden | Information Security |
| 5 | Bias | Understanding Research |
| 6 | Fleischmann | InfoSoc |
| 7 | Galloway | InfoSoc |
| 8 | Bailey | Understanding Research |

# Class table

| ClassID | Name | Professor |
|---|---|---|
| 1 | Database Management | Allen<br>Gunn |
| 2 | 3-D Printing | Riley |
| 3 | Information Security | Hayden |
| 4 | Understanding Research | Bailey<br>Bias |
| 5 | InfoSoc | Fleischmann<br>Galloway |

Neither of those is effective.

# Join tables

- allow you to create logistical relationships between tables
- cross-referencing, associative tables

# Professors table

| ProfID (PK) | Name | Course |
|---|---|---|
| 1 | Allen | Database Management |
| 2 | Gunn | Database Management |
| 3 | Riley | 3-D Printing |
| 4 | Hayden | Information Security |
| 5 | Bias | Understanding Research |
| 6 | Fleischmann | InfoSoc |
| 7 | Galloway | InfoSoc |
| 8 | Bailey | Understanding Research |

# Class table

| ClassID (PK) | Name | Professor |
|---|---|---|
| 1 | Database Management | Allen<br>Gunn |
| 2 | 3-D Printing | Riley |
| 3 | Information Security | Hayden |
| 4 | Understanding Research | Bailey<br>Bias |
| 5 | InfoSoc | Fleischmann<br>Galloway |

# ProfessorClass table

| PC_ID (PK) | ClassID (FK) | ProfID (FK) |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 2 | 3 |
| 4 | 3 | 4 |
| 5 | 4 | 5 |
| 6 | 4 | 8 |
| 7 | 5 | 6 |
| 8 | 5 | 7 |

# Layout


Professors
ProfID (PK)
Name
Prof Class
PCID (PK)
ProfID (FK)
ClassID (FK)
Class
ClassID (PK)
Name

# Create an ER diagram for a basic library information system.

Set of tables and relationships for:

- First and last name of author
- book title
- genre of book

Things to take into consideration:

- a book can have multiple authors
- an author can have multiple books
- books can have multiple genres