WELCOME

CSCA20 Week 10

Introduction to

Databases

If You Want to Learn More

CSCB20 ← Easy course like CSCA20

Introduction to Databases
(For Non-Computer Science Students)

Difficult course, but you learn a lot more → **CSCC43**

Introduction to Databases
(For Computer Science Students)

Why talk about databases?

Databases are one of the most important topics in computer sciences!

Almost all organizations, whether private or public, use databases in one way or another

You use databases every single day without even realizing it!

When Do You Use A Database?

Every time you search for information on the internet!

Every time you store information online

What is a Database?

A database is just a well-structured collection of data.

Data should be easily stored and retrieved

Often data is stored in the form of tables where the headers are properties, and each row represents an entry

What is a Database?

The format of which data is stored in a database is called its schema.

Uploads:

Attributes

Entries

image_name	uploader	image_size	search_tags
"img_1134.png"	"mrBubbles123"	30	["cats", "weekend", "12"]
"img_6126.jpg"	"hanna_mclean"	13	["beach", "sun", "trip"]
"dsc_2342.tiff"	"ms_skittles"	45	["baby", "weekend", "cute"]
"img_4911.jpeg"	"space_invader"	35	["game", "fortnite", "boy"]
…	…	…	…

What is a Database?

Does this sound familiar?
It should! This is how CSV files are formatted!

Uploads:

image_name	uploader	image_size	search_tags
“img_1134.png”	“mrBubbles123”	30	[“cats”, “weekend”, “12”]
“img_6126.jpg”	“hanna_mclean”	13	[“beach”, “sun”, “trip”]
“dsc_2342.tiff”	“ms_skittles”	45	[“baby”, “weekend”, “cute”]
“img_4911.jpeg”	“space_invader”	35	[“game”, “fortnite”, “boy”]
…	…	…	…

Database Example

Suppose that this is the schema that an image search platform uses to store data.

`Uploads:`

image_name	uploader	image_size	search_tags
“img_1134.png”	“mrBubbles123”	30	[“cats”, “weekend”, “12”]
“img_6126.jpg”	“hanna_mclean”	13	[“beach”, “sun”, “trip”]
“dsc_2342.tiff”	“ms_skittles”	45	[“baby”, “weekend”, “cute”]
“img_4911.jpeg”	“space_invader”	35	[“game”, “fortnite”, “boy”]
…	…	…	…

Database Example

How do we search for names of images that contain the tag "weekend"?

Uploads:

Attributes

Entries

image_name	uploader	image_size	search_tags
"img_1134.png"	"mrBubbles123"	30	["cats", "weekend", "12"]
"img_6126.jpg"	"hanna_mclean"	13	["beach", "sun", "trip"]
"dsc_2342.tiff"	"ms_skittles"	45	["baby", "weekend", "cute"]
"img_4911.jpeg"	"space_invader"	35	["game", "fortnite", "boy"]
…	…	…	…

Our query will be something along the lines of:

```
SELECT image_name
FROM uploads
WHERE search_tags CONTAINS “weekend”
```

Uploads:

image_name	uploader	image_size	search_tags
“img_1134.png”	“mrBubbles123”	30	[“cats”, “weekend”, “12”]
“img_6126.jpg”	“hanna_mclean”	13	[“beach”, “sun”, “trip”]
“dsc_2342.tiff”	“ms_skittles”	45	[“baby”, “weekend”, “cute”]
“img_4911.jpeg”	“space_invader”	35	[“game”, “fortnite”, “boy”]
…	…	…	…

Our query will be something along the lines of:

```
SELECT image_name
FROM uploads
WHERE search_tags CONTAINS “weekend”
```

Uploads:

image_name	uploader	image_size	search_tags
“img_1134.png”	“mrBubbles123”	30	[“cats”, “weekend”, “12”]
“img_6126.jpg”	“hanna_mclean”	13	[“beach”, “sun”, “trip”]
“dsc_2342.tiff”	“ms_skittles”	45	[“baby”, “weekend”, “cute”]
“img_4911.jpeg”	“space_invader”	35	[“game”, “fortnite”, “boy”]
…	…	…	…

Our query will be something along the lines of:

```
SELECT image_name
FROM uploads
WHERE search_tags CONTAINS “weekend”
```

Uploads:

image_name	uploader	image_size	search_tags
“img_1134.png”	“mrBubbles123”	30	[“cats”, “weekend”, “12”]
“img_6126.jpg”	“hanna_mclean”	13	[“beach”, “sun”, “trip”]
“dsc_2342.tiff”	“ms_skittles”	45	[“baby”, “weekend”, “cute”]
“img_4911.jpeg”	“space_invader”	35	[“game”, “fortnite”, “boy”]
…	…	…	…

Our query will be something along the lines of:

```
SELECT image_name
FROM uploads
WHERE search_tags CONTAINS “weekend”
```

Uploads:

image_name	uploader	image_size	search_tags
“img_1134.png”	“mrBubbles123”	30	[“cats”, “weekend”, “12”]
“img_6126.jpg”	“hanna_mclean”	13	[“beach”, “sun”, “trip”]
“dsc_2342.tiff”	“ms_skittles”	45	[“baby”, “weekend”, “cute”]
“img_4911.jpeg”	“space_invader”	35	[“game”, “fortnite”, “boy”]
…	…	…	…

Alternatively, you can think of it this way:

```
SELECT image_name
FROM uploads
WHERE search_tags CONTAINS “weekend”
```

Uploads:

image_name	uploader	image_size	search_tags
“img_1134.png”	“mrBubbles123”	30	[“cats”, “weekend”, “12”]
“img_6126.jpg”	“hanna_mclean”	13	[“beach”, “sun”, “trip”]
“dsc_2342.tiff”	“ms_skittles”	45	[“baby”, “weekend”, “cute”]
“img_4911.jpeg”	“space_invader”	35	[“game”, “fortnite”, “boy”]
…	…	…	…

Alternatively, you can think of it this way:

```
SELECT image_name
FROM uploads
WHERE search_tags CONTAINS “weekend”
```

Uploads:

image_name	uploader	image_size	search_tags
“img_1134.png”	“mrBubbles123”	30	[“cats”, “weekend”, “12”]
“img_6126.jpg”	“hanna_mclean”	13	[“beach”, “sun”, “trip”]
“dsc_2342.tiff”	“ms_skittles”	45	[“baby”, “weekend”, “cute”]
“img_4911.jpeg”	“space_invader”	35	[“game”, “fortnite”, “boy”]
…	…	…	…

Alternatively, you can think of it this way:

```
SELECT image_name
FROM uploads
WHERE search_tags CONTAINS “weekend”
```

Uploads:

image_name	uploader	image_size	search_tags
“img_1134.png”	“mrBubbles123”	30	[“cats”, “weekend”, “12”]
“dsc_2342.tiff”	“ms_skittles”	45	[“baby”, “weekend”, “cute”]

Alternatively, you can think of it this way:

```
SELECT image_name
FROM uploads
WHERE search_tags CONTAINS “weekend”
```

Uploads:

image_name
“img_1134.png”
“dsc_2342.tiff”

Alternatively, you can think of it this way:

```
SELECT image_name
FROM uploads
WHERE search_tags CONTAINS “weekend”
```

Uploads:

image_name
“img_1134.png”
“dsc_2342.tiff”

$\rightarrow$ (“img_1134.png”, “dsc_2342.tiff”)

Writing A Query

A query defines the parameters for the search that we want to perform on a database.

```
SELECT [some attribute or column]
FROM   [some table]
WHERE  [some condition is true]
```

Depending on the version of SQL that you use, the exact syntax will vary, but the idea is always the same.

Writing A Query

```
FROM [The name of the table we examine]
```

Before we can do a SELECT operation, we must first indicate which table we want SELECT from.

The FROM block will always be run first so that the query has a starting point.

Writing A Query

```
SELECT [some attribute or column]
```

When we select from a database, we want to make sure that the argument is a column or set of columns in our table.

We can also use SELECT * to denote that we want to select ALL the columns.

Writing A Query

```
WHERE [some condition holds true]
```

When we select from a database, we want can include a WHERE block to narrow down our search results to just a certain entries.

The WHERE block is technically optional, but it’s what gives you the actual search functionality.

SQL

SQL is a database management system that can be integrated into various programs and have numerous implementations that work with many programming languages.

In this course, we'll be using pySQLite using the `sqlite3` API (This is the module you have to import)

SQL is not quite like Python:
Python is used to do general computations,
SQL is used manipulate tables in a database.

SQL

SQL is a database management system that can be integrated into various programs and have numerous implementations that work with many programming languages.

In this course, we'll be using pySQLite using the `sqlite3` API (This is the module you have to import)

SQL is not quite like Python:
Python is used to do general computations,
SQL is used manipulate tables in a database.

Working With A Database

The first thing we need to do is import the sqlite3 module.

```
import sqlite3
```

Next, we need to connect to our database
and link to it using a cursor. Now we can do some work.

```
connection = sqlite3.connect(name of database)
cursor = connection.cursor()
```

Once we are done making changes, we need to save.

```
connection.commit()
```

After all changes have been saved, close all connections.

```
cursor.close()
connection.close()
```

Manipulating the Database

The cursor is a link to your database.
in other words, if you want to do something to your database, you must reference it using the cursor.

If you want to think of the database as a Object like a String, List, Dictionary etc, then the cursor is the database object that contains a set of database tools.

`.execute()`

The database cursor's execute method isn't a conventional method like those that you're used to seeing. It doesn't do any one thing...

.execute() does to the database whatever you tell it to do in SQL!

In other words, it's the bridge between your Python code and the SQL that modifies the database.

`.execute()`

SQL queries are always **written and passed** to `.execute()` **as a string.**

The **SQL itself specifies the operation** that `.execute()` performs on your database!

Common Table Tasks

Here are some common tasks that can be done using SQL and the cursor's `.execute()` method:

```
DROP TABLE IF EXISTS table_name
```

If the table already exists, erase it and set it up all over again. This should be used inside your functions before you create any new table.

```
CREATE TABLE table_name(columnName TYPE …)
```

Creates a new table with the given name and columns.

Columns must indicate the names of each column and the type of data that should go into that column. These types are not the same across Python and SQL!

Python Type	SQL Type
`Str`	`TEXT`
`Float`	`REAL`
`Int`	`INTEGER`

Common Table Tasks

```
SELECT columns FROM table WHERE condition
```

SQL queries (Of the format we discussed earlier) Can also be passed into .execute(), indicating that we want to search the database.

```
(INSERT INTO table VALUES (?, ?, …), data)
```

Add an entry (also called a VALUE or row) into the table. This is the query that must be paired with an actual dataset. Each "?" Is a placeholder for an attribute of the actual dataset.

(INSERT INTO table VALUES (?, ?, …), data)

Suppose we have a table called Uploads:

image_name	uploader	image_size
“img_1134.png”	“mrBubbles123”	30
“img_6126.jpg”	“hanna_mclean”	13

We want to add this row:

“dsc_2342.tiff”	“ms_skittles”	45

```
query = “INSERT INTO uploads VALUES (?, ?, ?)”
data = (“dsc_2342.tiff”, “ms_skittles”, 45)
cursor.execute(query, data)
```

And here's a gift to all of you who showed up today...

We're going to be going over **LAB 9** as an in-class demo.

QUESTIONS?

LET ME KNOW!