# Assignment 6: Data Representations

EPPS 6354: Information Management

# Basic Information

- API → Application Programming Interface which outlines rules in how programs work together
  - Allows users to access data files from certain websites
- REST → Representational State Transfer which determines how API is structured

- JSON → JavaScript Object Notation
- XML → Extensible Markup Language

# 1. Look up websites using JSON and XML data representations.

Analyze the websites in terms of structure and composition. Name the technology/methods used for creating the web database.

- Twitter API uses JSON data representation
  - Data objects are encoded using JSON where a key representing Twitter-related attributes is paired with value(s)
  - Access to Twitter API is possible through a frontend interface which includes an account
  - Twitter Interface is a progressive web app → web database is probably created with HTML, CSS, and JavaScript
- Cisco WebEx API uses XML data representation
  - Data requests are executed with a HTTP request and data responses are available in XML format
  - Structure of XML syntax is layered where XML document has a root element which expands to additional elements
  - Web database is most likely created with HTML

# 2. SQL exercise

- i. Express the following query in SQL using no subqueries and no set operations.

```
select ID
from student
except
select s id
from advisor
where i ID is not null
```

```
SELECT s.ID
FROM student AS s
LEFT JOIN advisor AS a
ON s.ID = a.s_ID
WHERE a.s_ID IS NULL;
```

| ID |
|---|
| 19991 |
| 54321 |
| 55739 |
| 70557 |

# 2. SQL exercise

- ii. Using the university schema, write an SQL query to find the names and IDs of those instructors who teach every course taught in his or her department (i.e., every course that appears in the *course* relation with the instructor's department name). Order result by name.

```sql
SELECT cnt.name, cnt.ID
FROM (
    SELECT COUNT(DISTINCT i.name) AS numProf, c.dept_name, i.name, i.ID
    FROM course AS c
    NATURAL JOIN instructor AS i
    GROUP BY c.dept_name) AS cnt
WHERE numProf = 1
ORDER BY cnt.name;
```

| name | ID |
|---|---|
| Crick | 76766 |
| Kim | 98345 |
| Mozart | 15151 |

# 2. SQL exercise

```
SELECT i.dept_name, i.name
FROM course AS c
JOIN instructor AS i
ON c.dept_name = i.dept_name
```

| dept_name | name |
|---|---|
| Biology | Crick |
| Biology | Crick |
| Biology | Crick |
| Comp. Sci. | Brandt |
| Comp. Sci. | Katz |
| Comp. Sci. | Srinivasan |
| Comp. Sci. | Brandt |
| Comp. Sci. | Katz |
| Comp. Sci. | Srinivasan |
| Comp. Sci. | Brandt |
| Comp. Sci. | Katz |
| Comp. Sci. | Srinivasan |
| Comp. Sci. | Brandt |
| Comp. Sci. | Katz |
| Comp. Sci. | Srinivasan |
| Comp. Sci. | Brandt |
| Comp. Sci. | Katz |
| Comp. Sci. | Srinivasan |
| Elec. Eng. | Kim |
| Finance | Singh |
| Finance | Wu |
| History | Califieri |
| History | El Said |
| Music | Mozart |
| Physics | Einstein |
| Physics | Gold |