CoGrammar

Week 16 – Tutorial Class

SKILLS FOR LIFE

SKILLS BOOTCAMPS

Department for Education

Software Engineering Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(FBV: Mutual Respect.)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: **Open Class Questions**

CoGrammar

Software Engineering Lecture Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**

- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**

- We would love your **feedback** on lectures: **Feedback on Lectures**

Progression Criteria

✓ **Criterion 1: Initial Requirements**

- Complete 15 hours of Guided Learning Hours and the first four tasks within two weeks.

✓ **Criterion 2: Mid-Course Progress**

- Software Engineering: Finish 14 tasks by week 8.
- Data Science: Finish 13 tasks by week 8.

✓ **Criterion 3: Post-Course Progress**

- Complete all mandatory tasks by 24th March 2024.
- Record an Invitation to Interview within 4 weeks of course completion, or by 30th March 2024.
- Achieve 112 GLH by 24th March 2024.

✓ **Criterion 4: Employability**

- Record a Final Job Outcome within 12 weeks of graduation, or by 23rd September 2024.

Lecture Objectives

1. **Recall the fundamentals of SQL.**
2. **Apply knowledge of SQL.**

Create Table Example

- Table names use the snake_case convention with plural nouns.
- Columns name use the snake_case convention with singular nouns.

```
CREATE TABLE employees (
        employee_id int NOT NULL,
        last_name varchar(255) NOT NULL,
        first_name varchar(255),
        address varchar(255),
        phone_number varchar(255),
        );
```

CoGrammar

Inserting Values Example

1. Specify both the column names and the values to be inserted.

```
INSERT INTO employees (employee_id, last_name, first_name, address,
        phone_number)
        VALUES (1234, 'Smith', 'John', '25 Oak Rd', '0837856767');
```

2. Specify the values only.

```
INSERT INTO employees
        VALUES (1, 'Smith', 'John', '25 Oak Rd', '0837856767');
```

CoGrammar

Retrieving Data Example

- To select all the columns in a table:

```
SELECT * FROM employees;
```

- To select specific columns from a table:

```
SELECT first_name, last_name
        FROM employees;
```

CoGrammar

Ordering Data

- You can use the ORDER BY command to sort the results returned in ascending or descending order. The ORDER BY command sorts the records in ascending order by default.
- Ordering records in ascending order add ASC or leave out for default ordering.

```
SELECT * FROM employees
      ORDER BY last_name ASC, first_name ASC;
```

- You need to use the DESC keyword to sort the records in descending order.

```
SELECT * FROM employees
      ORDER BY last_name DESC, first_name DESC;
```

CoGrammar

Using WHERE, IN and BETWEEN keywords

```
SELECT * FROM employees
      WHERE first_name = 'John';
```

```
SELECT * FROM employees
      WHERE city IN (“New York”, “London”);
```

```
SELECT * FROM students
      WHERE grade BETWEEN 60 AND 80;
```

CoGrammar

Modifying Values Example

customer_id	first_name	last_name	address	city
1	Maria	Anderson	23 York Street	New York
2	Jackson	Peters	124 River Road	Berlin
3	Thomas	Hardy	455 Hanover Square	London
4	Kelly	Martins	55 Loop Street	Cape Town

```
UPDATE customers
        SET address = '78 Oak St', city =  'Los Angeles'
        WHERE customer_id = 1;
```

Removing Rows

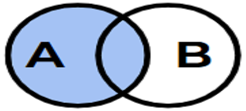
- Removing a row is a simple process. All you need to do is select the right table and row that you want to remove.
- The DELETE statement is used to remove existing rows from a table.

```
DELETE FROM customers
        WHERE customer_id = 4;
```

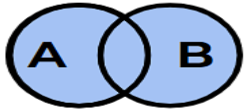
CoGrammar

Accessing Multiple Tables

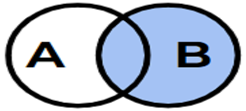
- **LEFT JOIN** - All values in A, and matching values in B
- **INNER JOIN** - Records match in both tables
- **FULL OUTER JOIN** - All values in both tables



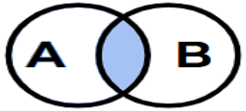
```
SELECT *
FROM A
LEFT JOIN B
ON A.id = B.id
```



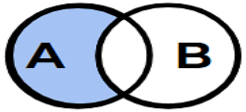
```
SELECT *
FROM A
FULL OUTER JOIN B
ON A.id = B.id
```



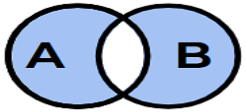
```
SELECT *
FROM A
RIGHT JOIN B
ON A.id = B.id
```



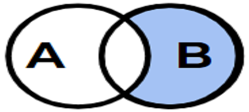
```
SELECT *
FROM A
INNER JOIN B
ON A.id = B.id
```



```
SELECT *
FROM A
LEFT JOIN B
ON A.id = B.id
WHERE B.id IS NULL
```



```
SELECT *
FROM A
FULL OUTER JOIN B
ON A.id = B.id
WHERE A.id IS NULL
OR B.id IS NULL
```



```
SELECT *
FROM A
RIGHT JOIN B
ON A.id = B.id
WHERE A.id IS NULL
```

Removing Tables

- The DROP TABLE statement is used to remove every trace of a table in a database.
- Removing Table Example:

```
DROP TABLE customers;
```

SQLite Syntax

```
import sqlite3

db = sqlite3.connect('data/student_db')
cursor = db.cursor()

cursor.execute('''
    CREATE TABLE student(id INTEGER PRIMARY KEY, name TEXT,
                                  grade INTEGER)
''')

db.commit()
```

CoGrammar

SQLite Syntax ...

```
cursor.execute('''INSERT INTO student(name, grade)
               VALUES(?,?)''', (name1,grade1))
db.commit()

students_ = [(name1,grade1),(name2,grade2),(name3,grade3)]
cursor.executemany(''' INSERT INTO student(name, grade) VALUES(?,?)''',
students_)
db.commit()
```

CoGrammar

Thank you for joining

CoGrammar