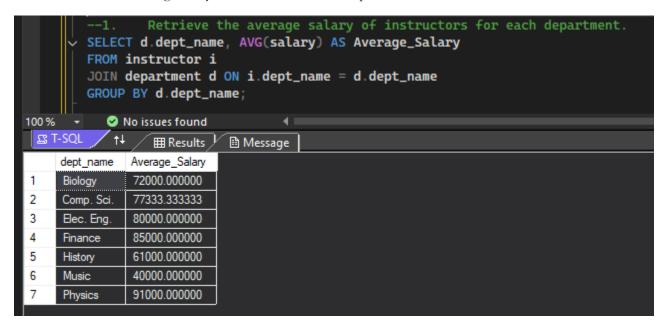
COSC 4301 – Database Theory and Practice Lab 04

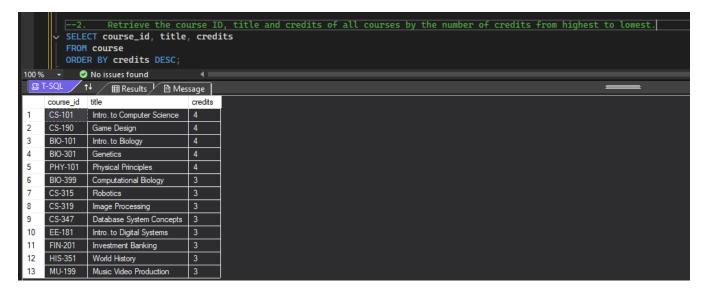
Construct the university database using DDL.sql and insert the data using smallRelationsInsertFile.sql.

Develop the following queries:

1. Retrieve the average salary of instructors for each department.



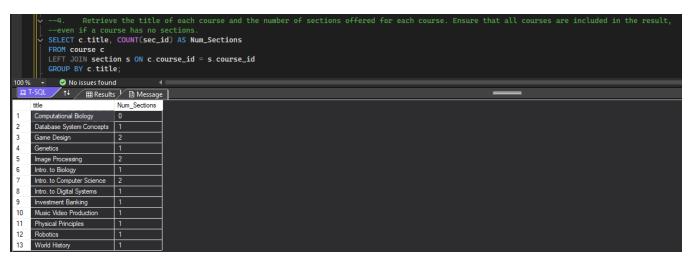
2. Retrieve the course ID, title and credits of all courses by the number of credits from highest to lowest.



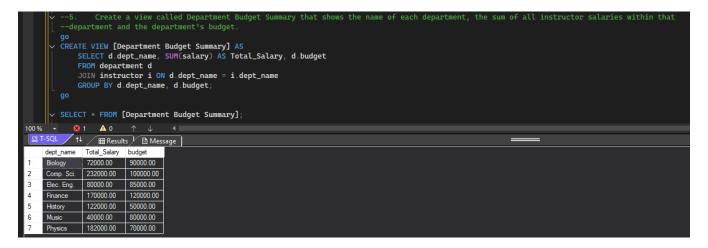
3. Retrieve the name of each department and the number of instructors in each department. Ensure that *all* departments are included in the result, even if a department has no instructors.



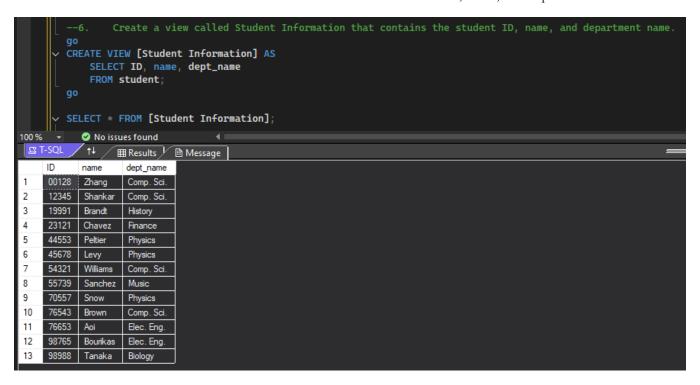
4. Retrieve the title of each course and the number of sections offered for each course. Ensure that *all* courses are included in the result, even if a course has no sections.



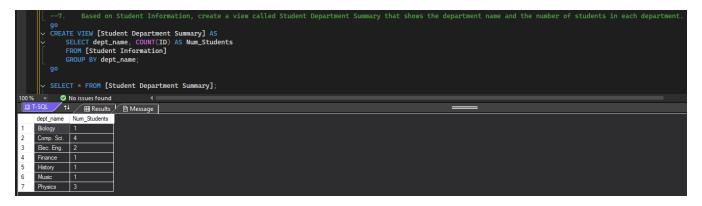
5. Create a view called Department Budget Summary that shows the name of each department, the sum of all instructor salaries within that department and the department's budget.



6. Create a view called Student Information that contains the student ID, name, and department name.



7. Based on Student Information, create a view called Student Department Summary that shows the department name and the number of students in each department.



8. Insert values into Student Information and then display the new tuple within the student table.

```
--8. Insert values into Student Information and then display the new tuple within the student table.

INSERT INTO [Student Information] VALUES (11111, 'Williams-Wimmer', 'Comp. Sci.');

SELECT *
FROM student
WHERE ID = 11111;

100 % • No issues found

ID name dept_name tot_cred

1 11111 Williams-Wimmer Comp. Sci. NULL
```

Code:

```
--1. Retrieve the average salary of instructors for each department.
SELECT d.dept_name, AVG(salary) AS Average_Salary
FROM instructor i
JOIN department d ON i.dept_name = d.dept_name
GROUP BY d.dept_name;
      Retrieve the course ID, title and credits of all courses by the number of credits
from highest to lowest.
SELECT course_id, title, credits
FROM course
ORDER BY credits DESC;
      Retrieve the name of each department and the number of instructors in each
department. Ensure that all departments are included in the result,
--even if a department has no instructors.
SELECT d.dept_name, COUNT(ID) AS Num_Instructors
FROM instructor i
RIGHT JOIN department d ON d.dept_name = i.dept_name
GROUP BY d.dept_name;
      Retrieve the title of each course and the number of sections offered for each
course. Ensure that all courses are included in the result,
--even if a course has no sections.
SELECT c.title, COUNT(sec_id) AS Num_Sections
FROM course c
LEFT JOIN section s ON c.course_id = s.course_id
GROUP BY c.title;
--5. Create a view called Department Budget Summary that shows the name of each
department, the sum of all instructor salaries within that
--department and the department's budget.
go
CREATE VIEW [Department Budget Summary] AS
      SELECT d.dept_name, SUM(salary) AS Total_Salary, d.budget
      FROM department d
      JOIN instructor i ON d.dept_name = i.dept_name
      GROUP BY d.dept_name, d.budget;
go
SELECT * FROM [Department Budget Summary];
      Create a view called Student Information that contains the student ID, name, and
department name.
CREATE VIEW [Student Information] AS
      SELECT ID, name, dept_name
      FROM student;
go
SELECT * FROM [Student Information];
      Based on Student Information, create a view called Student Department Summary that
shows the department name and the number of students in each department.
CREATE VIEW [Student Department Summary] AS
      SELECT dept_name, COUNT(ID) AS Num_Students
      FROM [Student Information]
      GROUP BY dept_name;
go
```

```
SELECT * FROM [Student Department Summary];

--8. Insert values into Student Information and then display the new tuple within the student table.
INSERT INTO [Student Information] VALUES (11111, 'Williams-Wimmer', 'Comp. Sci.');
SELECT *
FROM student
WHERE ID = 11111;
```