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.

```sql
--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;
```

100 % No issues found

T-SQL | Results | Message

	dept_name	Average_Salary
1	Biology	72000.000000
2	Comp. Sci.	77333.333333
3	Elec. Eng.	80000.000000
4	Finance	85000.000000
5	History	61000.000000
6	Music	40000.000000
7	Physics	91000.000000

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

```sql
--2.    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;
```

100 % No issues found

T-SQL | Results | Message

	course_id	title	credits
1	CS-101	Intro. to Computer Science	4
2	CS-190	Game Design	4
3	BIO-101	Intro. to Biology	4
4	BIO-301	Genetics	4
5	PHY-101	Physical Principles	4
6	BIO-399	Computational Biology	3
7	CS-315	Robotics	3
8	CS-319	Image Processing	3
9	CS-347	Database System Concepts	3
10	EE-181	Intro. to Digital Systems	3
11	FIN-201	Investment Banking	3
12	HIS-351	World History	3
13	MU-199	Music Video Production	3

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.

```sql
--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.
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;
```

100 % No issues found

T-SQL | Results | Message

	dept_name	Num_Instructors
1	Biology	1
2	Comp. Sci.	3
3	Elec. Eng.	1
4	Finance	2
5	History	2
6	Music	1
7	Physics	2

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.

```sql
--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.
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;
```

100 % No issues found

T-SQL | Results | Message

	title	Num_Sections
1	Computational Biology	0
2	Database System Concepts	1
3	Game Design	2
4	Genetics	1
5	Image Processing	2
6	Intro. to Biology	1
7	Intro. to Computer Science	2
8	Intro. to Digital Systems	1
9	Investment Banking	1
10	Music Video Production	1
11	Physical Principles	1
12	Robotics	1
13	World History	1

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.

```sql
--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];
```

100 % 1 0

T-SQL | Results | Message

	dept_name	Total_Salary	budget
1	Biology	72000.00	90000.00
2	Comp. Sci.	232000.00	100000.00
3	Elec. Eng.	80000.00	85000.00
4	Finance	170000.00	120000.00
5	History	122000.00	50000.00
6	Music	40000.00	80000.00
7	Physics	182000.00	70000.00

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

```sql
--6.    Create a view called Student Information that contains the student ID, name, and department name.
go
CREATE VIEW [Student Information] AS
    SELECT ID, name, dept_name
    FROM student;
go

SELECT * FROM [Student Information];
```

100 % No issues found

T-SQL | Results | Message

	ID	name	dept_name
1	00128	Zhang	Comp. Sci.
2	12345	Shankar	Comp. Sci.
3	19991	Brandt	History
4	23121	Chavez	Finance
5	44553	Peltier	Physics
6	45678	Levy	Physics
7	54321	Williams	Comp. Sci.
8	55739	Sanchez	Music
9	70557	Snow	Physics
10	76543	Brown	Comp. Sci.
11	76653	Aoi	Elec. Eng.
12	98765	Bourikas	Elec. Eng.
13	98988	Tanaka	Biology

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.

```sql
--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.
go
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];
```

100 % No issues found

T-SQL | Results | Message

	dept_name	Num_Students
1	Biology	1
2	Comp. Sci.	4
3	Elec. Eng.	2
4	Finance	1
5	History	1
6	Music	1
7	Physics	3

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

```sql
--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

T-SQL | Results | Message

	ID	name	dept_name	tot_cred
1	11111	Williams-Wimmer	Comp. Sci.	NULL

Code:

```sql
--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;

--2.  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;

--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.
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;

--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.
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];

--6.  Create a view called Student Information that contains the student ID, name, and
department name.
go
CREATE VIEW [Student Information] AS
      SELECT ID, name, dept_name
      FROM student;
go

SELECT * FROM [Student Information];

--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.
go
CREATE VIEW [Student Department Summary] AS
      SELECT dept_name, COUNT(ID) AS Num_Students
      FROM [Student Information]
      GROUP BY dept_name;
go
```

```sql
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;
```