CS3550: Database Management Systems - I
Assignment Report

Pandrangi Aditya Sriram
EE22BTECH11039

Question 1a

Query:

```
SELECT instructor.name, instructor.id, COUNT(DISTINCT
teaches.course_id) as no_of_courses
FROM instructor, teaches
WHERE instructor.id = teaches.id
GROUP BY instructor.id
ORDER BY no_of_courses DESC
LIMIT 3;
```

Output:

```
   name    |  id   | no_of_courses
-----------+-------+---------------
 DAgostino | 22591 |            11
 Mingoz    | 6569  |             8
 Dale      | 99052 |             8
(3 rows)
```

Question 1b

Query:

```
SELECT instructor.name, instructor.id, COUNT(DISTINCT
teaches.course_id) as no_of_courses
FROM instructor, teaches
WHERE instructor.id = teaches.id and instructor.dept_name =
'Statistics'
GROUP BY instructor.id
ORDER BY no_of_courses DESC
LIMIT 3;
```

Output:

```
   name    |  id   | no_of_courses
-----------+-------+---------------
 Atanassov | 28400 |             2
 Choll     | 90643 |             1
(2 rows)
```

Question 2

Query:

```
SELECT
    instructor.name,
    instructor.dept_name,
    teaches.course_id,
    course.title AS course_title,
    (
        SELECT(COUNT(takes.course_id))
        FROM takes
        WHERE takes.course_ID = teaches.course_id AND takes.year =
teaches.year AND takes.semester = teaches.semester AND
takes.sec_id = section.sec_id
    ) AS total_enrollment, teaches.semester, teaches.year,
section.sec_id
FROM
    instructor,
    teaches,
    course,
    section
WHERE
    salary = (SELECT MAX(salary) FROM instructor) AND teaches.id =
instructor.id AND course.course_id = teaches.course_id AND
course.course_id = section.course_id
ORDER BY
    course.course_id ASC, teaches.semester ASC, teaches.year ASC;
```

Output:

```
  name   | dept_name | course_id |      course_title       | total_enrollment | semester | year | sec_id
---------+-----------+-----------+-------------------------+------------------+----------+------+--------
 Wieland | Pol. Sci. | 545       | International Practicum |              306 | Fall     | 2001 | 1
 Wieland | Pol. Sci. | 581       | Calculus                |              313 | Spring   | 2005 | 1
 Wieland | Pol. Sci. | 591       | Shakespeare             |              291 | Spring   | 2005 | 1
(3 rows)
```

Question 3

Query:

```
SELECT
    course.course_id,
    course.title AS course_title,
    course.dept_name AS course_dept_name,
    instructor.name AS instructor_name,
    section.sec_id,
    section.semester,
    section.year,
    section.time_slot_id,
    COUNT(takes.id) AS no_of_students
FROM
    course
    NATURAL JOIN section
    NATURAL JOIN teaches
    JOIN instructor ON instructor.id = teaches.id
    JOIN takes ON takes.course_id = section.course_id AND
takes.sec_id = section.sec_id AND takes.semester =
section.semester AND takes.year = section.year
WHERE
    course.course_id = '362'
GROUP BY
    course.course_id, course.title, course.dept_name,
instructor.name, section.sec_id, section.semester, section.year,
section.time_slot_id
ORDER BY section.year DESC;
```

Output:

```
 course_id |   course_title   | course_dept_name | instructor_name | sec_id | semester | year | time_slot_id | no_of_students
-----------+------------------+------------------+-----------------+--------+----------+------+--------------+----------------
 362       | Embedded Systems | Finance          | Mingoz          | 3      | Spring   | 2008 | L            |            322
 362       | Embedded Systems | Finance          | Mingoz          | 2      | Fall     | 2006 | A            |            320
 362       | Embedded Systems | Finance          | Mingoz          | 1      | Fall     | 2005 | I            |            338
(3 rows)
```

Question 4

Query:

```
SELECT
    COUNT(*) AS out_of_department_registrations
FROM
    takes
    NATURAL JOIN course
    JOIN student ON student.id = takes.id
WHERE
    course.course_id = '319'
    AND takes.year = 2003
    AND student.dept_name <> course.dept_name;
```

Output:

```
 out_of_department_registrations
--------------------------------
                            303
(1 row)
```

Question 5

Query:

```
SELECT
    takes.id AS student_id,
    student.name,
    student.dept_name,
    SUM(credits) AS total_credits
FROM
    takes
    NATURAL JOIN course
    JOIN student ON takes.id = student.id
GROUP BY
    takes.id, student.name, student.dept_name
ORDER BY
    total_credits DESC,
    student.name
LIMIT 3;
```

Output:

```
 student_id |  name   | dept_name | total_credits
------------+---------+-----------+---------------
 12078      | Knutson | Languages |            93
 90448      | Godfrey | English   |            90
 44551      | Nguyen  | Astronomy |            90
(3 rows)
```

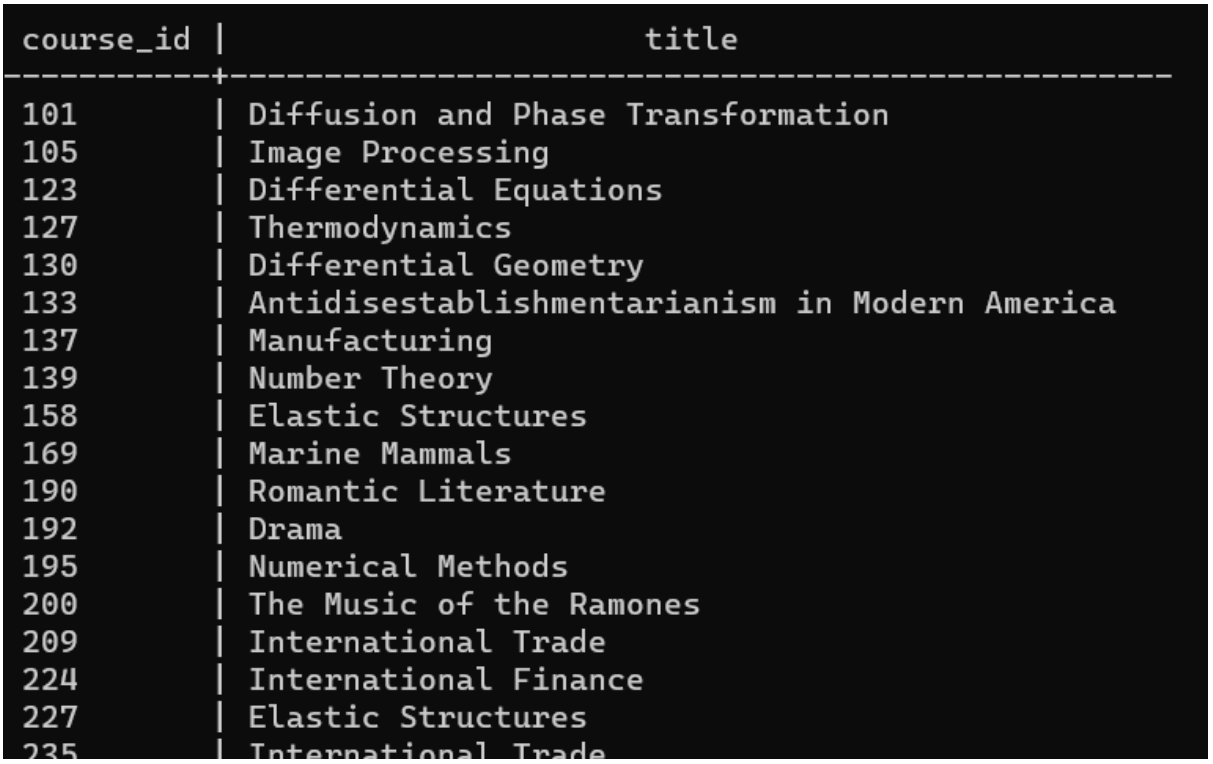
Question 6

Query:

```
SELECT
    course.course_id, course.title
FROM
    course
    NATURAL LEFT OUTER JOIN
        (
            SELECT
                DISTINCT section.course_id
            FROM
                section
            WHERE
                section.year IN (2003, 2004)
        ) AS courses_2003_2004
WHERE
    courses_2003_2004.course_id IS NULL
ORDER BY
    course.course_id ASC;
```

Output:

As the output is very long, text response has been recorded. The output has been shown only partially in the screenshot.



course_id	title
101	Diffusion and Phase Transformation
105	Image Processing
123	Differential Equations
127	Thermodynamics
130	Differential Geometry
133	Antidisestablishmentarianism in Modern America
137	Manufacturing
139	Number Theory
158	Elastic Structures
169	Marine Mammals
190	Romantic Literature
192	Drama
195	Numerical Methods
200	The Music of the Ramones
209	International Trade
224	International Finance
227	Elastic Structures
235	International Trade

course_id	title
101	Diffusion and Phase Transformation
105	Image Processing

123	Differential Equations
127	Thermodynamics
130	Differential Geometry
133	Antidisestablishmentarianism in Modern America
137	Manufacturing
139	Number Theory
158	Elastic Structures
169	Marine Mammals
190	Romantic Literature
192	Drama
195	Numerical Methods
200	The Music of the Ramones
209	International Trade
224	International Finance
227	Elastic Structures
235	International Trade
236	Design and Analysis of Algorithms
237	Surfing
238	The Music of Donovan
239	The Music of the Ramones
241	Biostatistics
242	Rock and Roll
254	Security
258	Colloid and Surface Chemistry
265	Thermal Physics
267	Hydraulics
270	Music of the 90s
272	Geology
274	Corporate Law
275	Romantic Literature
276	Game Design
278	Greek Tragedy
284	Topology
292	Electron Microscopy
304	Music 2 New for your Instructor
313	International Trade
318	Geology
324	Ponzi Schemes
328	Composition and Literature
334	International Trade
337	Differential Geometry
338	Graph Theory
340	Corporate Law
341	Quantum Mechanics
344	Quantum Mechanics
345	Race Car Driving
348	Compiler Design
349	Networking

352	Compiler Design
353	Operating Systems
359	Game Programming
362	Embedded Systems
366	Computational Biology
371	Milton
376	Cost Accounting
377	Differential Geometry
391	Virology
392	Recursive Function Theory
393	Aerodynamics
394	C Programming
396	C Programming
399	RPG Programming
403	Immunology
407	Industrial Organization
411	Music of the 80s
415	Numerical Methods
416	Data Mining
426	Video Gaming
436	Stream Processing
442	Strength of Materials
443	Journalism
445	Biostatistics
451	Database System Concepts
456	Hebrew
457	Systems Software
458	The Renaissance
461	Physical Chemistry
468	Fractal Geometry
476	International Communication
482	FOCAL Programming
486	Accounting
487	Physical Chemistry
489	Journalism
493	Music of the 50s
494	Automobile Mechanics
496	Aquatic Chemistry
500	Networking
539	International Finance
544	Differential Geometry
545	International Practicum
546	Creative Writing
549	Banking and Finance
558	Environmental Law
559	Martian History
561	The Music of Donovan
577	The Music of Dave Edmunds

580 | The Music of Dave Edmunds
581 | Calculus
582 | Marine Mammals
584 | Computability Theory
586 | Image Processing
591 | Shakespeare
594 | Cognitive Psychology
598 | Number Theory
604 | UNIX System Programmming
608 | Electron Microscopy
612 | Mobile Computing
618 | Thermodynamics
626 | Multimedia Design
628 | Existentialism
630 | Religion
631 | Plasma Physics
634 | Astronomy
647 | Service-Oriented Architectures
656 | Groups and Rings
659 | Geology
663 | Geology
664 | Elastic Structures
666 | Multivariable Calculus
679 | The Beatles
680 | Electricity and Magnetism
681 | Medieval Civilization or Lack Thereof
692 | Cat Herding
694 | Optics
696 | Heat Transfer
702 | Arabic
704 | Marine Mammals
716 | Medieval Civilization or Lack Thereof
730 | Quantum Mechanics
731 | The Music of Donovan
761 | Existentialism
762 | The Monkeys
769 | Logic
770 | European History
774 | Game Programming
780 | Geology
781 | Compiler Design
787 | C Programming
791 | Operating Systems
792 | Image Processing
793 | Decison Support Systems
804 | Introduction to Burglary
805 | Composition and Literature
810 | Mobile Computing

```
 814       | Compiler Design
 818       | Environmental Law
 820       | Assembly Language Programming
 830       | Sensor Networks
 841       | Fractal Geometry
 843       | Environmental Law
 852       | World History
 857       | UNIX System Programmming
 858       | Sailing
 864       | Heat Transfer
 867       | The IBM 360 Architecture
 875       | Bioinformatics
 877       | Composition and Literature
 887       | Latin
 893       | Systems Software
 897       | How to Succeed in Business Without Really Trying
 898       | Petroleum Engineering
 902       | Existentialism
 919       | Computability Theory
 922       | Microeconomics
 927       | Differential Geometry
 947       | Real-Time Database Systems
 949       | Japanese
 958       | Fiction Writing
 959       | Bacteriology
 960       | Tort Law
 962       | Animal Behavior
 963       | Groups and Rings
 966       | Sanitary Engineering
 969       | The Monkeys
 972       | Greek Tragedy
 983       | Virology
 984       | Music of the 50s
 991       | Transaction Processing
 998       | Immunology
(181 rows)
```

Question 7

Query:

```
WITH firstoffer AS
    (
    SELECT
        section.course_id,
        MIN(year) AS first_time_offered_year
    FROM
        section
    GROUP BY
        section.course_id
    )

SELECT
    course.course_id,
    course.title,
    instructor.name AS instructor,
    firstoffer.first_time_offered_year
FROM
    firstoffer
    JOIN section ON firstoffer.course_id = section.course_id AND
firstoffer.first_time_offered_year = section.year
    JOIN course ON firstoffer.course_id = course.course_id
    JOIN teaches ON section.course_id = teaches.course_id AND
section.sec_id = teaches.sec_id
    JOIN instructor ON teaches.id = instructor.id
WHERE
    firstoffer.first_time_offered_year =
    (
        SELECT
            MAX(first_time_offered_year)
        FROM
            firstoffer
    )
ORDER BY
    course.course_id ASC;
```

Output:

```
 course_id |           title            | instructor | first_time_offered_year
-----------+----------------------------+------------+-------------------------
 270       | Music of the 90s           | Sakurai    |                    2010
 313       | International Trade        | Morris     |                    2010
 415       | Numerical Methods          | Valtchev   |                    2010
 476       | International Communication | Romero     |                    2010
 493       | Music of the 50s           | Mahmoud    |                    2010
 679       | The Beatles                | Luo        |                    2010
 692       | Cat Herding                | Tung       |                    2010
 843       | Environmental Law          | Lembr      |                    2010
(8 rows)
```

Question 8

Query:

```
SELECT
    course_id, title
FROM
    course
WHERE
    title LIKE '_______________%'
    AND LOWER(title) LIKE '%sys%'
ORDER BY
    course_id ASC;
```

Output:

```
 course_id |           title
-----------+----------------------------
 353       | Operating Systems
 362       | Embedded Systems
 451       | Database System Concepts
 457       | Systems Software
 604       | UNIX System Programmming
 791       | Operating Systems
 793       | Decison Support Systems
 857       | UNIX System Programmming
 893       | Systems Software
 947       | Real-Time Database Systems
(10 rows)
```

Question 9

Query:

```
SELECT
    dept_name, AVG(salary) AS average_salary
FROM
    instructor
    NATURAL JOIN department
GROUP BY
    dept_name
ORDER BY
    average_salary DESC
LIMIT 1;
```

Output:

```
 dept_name |   average_salary
-----------+--------------------
 Physics   | 114576.900000000000
(1 row)
```

Question 10

Query:

```
SELECT
    instructor.id, instructor.name, instructor.dept_name
FROM
    instructor
    LEFT OUTER JOIN teaches ON teaches.id = instructor.id AND
teaches.year = 2003
GROUP BY
    instructor.id, instructor.name, instructor.dept_name
HAVING
    COUNT(teaches.course_id) <= 1
ORDER BY
    instructor.id ASC;
```

Output:

id	name	dept_name
14365	Lembr	Accounting
15347	Bawa	Athletics
16807	Yazdi	Athletics
19368	Wieland	Pol. Sci.
25946	Liley	Languages
28097	Kean	English
28400	Atanassov	Statistics
31955	Moreira	Accounting
3199	Gustafsson	Elec. Eng.
3335	Bourrier	Comp. Sci.
34175	Bondi	Comp. Sci.
35579	Soisalon-Soininen	Psychology
36897	Morris	Marketing
37687	Arias	Statistics
4034	Murata	Athletics
41930	Tung	Athletics
4233	Luo	English
42782	Vicentino	Elec. Eng.
43779	Romero	Astronomy
48507	Lent	Mech. Eng.
48570	Sarkar	Pol. Sci.
50330	Shuming	Physics
50885	Konstantinides	Languages
52647	Bancilhon	Pol. Sci.
57180	Hau	Accounting
58558	Dusserre	Marketing
59795	Desyl	Languages
63287	Jaekel	Athletics
63395	McKinnon	Cybernetics
64871	Gutierrez	Statistics
6569	Mingoz	Finance
65931	Pimenta	Cybernetics
72553	Yin	English
73623	Sullivan	Elec. Eng.
74420	Voronina	Physics
74426	Kenje	Marketing
77346	Mahmoud	Geology
78699	Pingr	Statistics
79653	Levine	Elec. Eng.
80759	Queiroz	Biology
81991	Valtchev	Biology
90376	Bietzk	Cybernetics
90643	Choll	Statistics
95030	Arinb	Statistics
95709	Sakurai	English
96895	Mird	Marketing
97302	Bertolino	Mech. Eng.

(47 rows)