DBMS-1
Assignment 1:
Report

Waghmare Aditya Abhaykumar
CS22BTECH11061

November 20, 2023

1: Find the top-3 instructors who have have taught most number of distinct courses from

a. Across all departments

Query:

```
SELECT instructor.id AS instructor_id, instructor.name AS
   instructor_name
FROM instructor
INNER JOIN teaches ON teaches.id = instructor.id
GROUP BY instructor.id
ORDER BY COUNT(DISTINCT teaches.course_id) DESC
LIMIT 3;
```

Output:

```
 instructor_id | instructor_name
---------------+-----------------
 22591         | DAgostino
 6569          | Mingoz
 99052         | Dale
(3 rows)
```

Figure 1: Output for 1.a

b. Statistics department

Query:

```
SELECT instructor.id AS instructor_id, instructor.name AS
   instructor_name
FROM instructor
INNER JOIN teaches ON teaches.id = instructor.id
WHERE instructor.dept_name = 'Statistics'
GROUP BY instructor.id
ORDER BY COUNT(DISTINCT teaches.course_id) DESC
LIMIT 3;
```

Output:

```
 instructor_id | instructor_name
---------------+-----------------
 28400         | Atanassov
 90643         | Choll
(2 rows)
```

Figure 2: Output for 1.b

2: Print teaching record of the instructor who has the highest salary, showing the instructor department name, course identifier, course title, section number, semester, year and total enrollment. Sort your result by course_id, year, semester in ascending order.

Query:

```sql
SELECT instructor.dept_name, teaches.course_id, course.title,
   section.sec_id, teaches.semester, teaches.year,
   total_enrollments
FROM instructor
LEFT JOIN teaches ON teaches.id = instructor.id
LEFT JOIN course ON course.course_id = teaches.course_id
LEFT JOIN
    (SELECT course_id AS c_id, COUNT(id) AS total_enrollments
    FROM takes
    GROUP BY course_id)
ON c_id = course.course_id
LEFT JOIN section ON section.course_id = course.course_id
WHERE instructor.id = (SELECT id FROM instructor ORDER BY salary
   DESC LIMIT 1)
ORDER BY teaches.course_id, teaches.year, teaches.semester;
```

Output:

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

Figure 3: Output for 2

3: Print history of the course with course_id = 362. For each offering of the course, print course id, course title, course department name, instructor name, number of registered students, section id, semester, year and timetable slot. Sort your result by year in descending order.

Query:

```sql
SELECT DISTINCT course.course_id, course.title, course.dept_name,
    instructor.name AS instructor_name, registered, teaches.
   sec_id, teaches.semester, teaches.year, section.time_slot_id
FROM course
LEFT JOIN teaches ON course.course_id = teaches.course_id
LEFT JOIN
    (SELECT course_id, year, semester, COUNT(id) AS registered
    FROM takes
    GROUP BY course_id, year, semester)
    AS take
ON (take.course_id = course.course_id and take.semester = teaches
   .semester and take.year = teaches.year)
LEFT JOIN instructor ON instructor.id = teaches.id
LEFT JOIN section ON (course.course_id = section.course_id and
   teaches.sec_id = section.sec_id and teaches.semester = section
   .semester and teaches.year = section.year)
WHERE course.course_id = '362'
ORDER BY teaches.year DESC;
```

Output:

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

Figure 4: Output for 3

4: For the course_id 319 that was offered in 2003, find the count of out of department student registration.

Query:

```sql
SELECT COUNT(takes.id) AS out_of_department_student_registrations
FROM takes
LEFT JOIN student ON student.id = takes.id
LEFT JOIN course ON course.course_id = takes.course_id
WHERE course.course_id = '319' and year = 2003 and student.
   dept_name != course.dept_name;
```

Output:

```
 out_of_department_student_registrations
----------------------------------------
                                    304
(1 row)
```

Figure 5: Output for 4

5: Find top-3 students who have registered for the highest number of course credits. Order by total credits and name. Print student id, name, department and total credits (Compute it from the takes and course tables. Do not use tot_credit in the student table.)

Query:

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

Output:

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

Figure 6: Output for 5

6: Find the distinct set of courses that were not offered during 2003 and 2004. Print the course id and title. Sort your result by course id in ascending order.

Query:

```
SELECT DISTINCT course_id, title
FROM course
WHERE course_id NOT IN
    (SELECT DISTINCT course_id
    FROM teaches
    WHERE year = 2003 OR year = 2004)
ORDER BY course_id;
```

Output:

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

Figure 7: Output for 6_1

```
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
```

Figure 8: Output for 6_2

```
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
```

Figure 9: Output for 6_3

```
 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)
```

Figure 10: Output for 6_4

7: Find the courses that were offered for the first time most recently in terms of year. Print the course id, title, instructor, year. Sort your result by course id in ascending order. [Find the most recent year when a course was offered for the first time. If there are more than one course offered that year for the first time, then print all of them.]

Query:

```sql
SELECT course.course_id AS course__id, course.title, instructor.
   name AS instructor_name, teaches.year
FROM course
INNER JOIN teaches ON teaches.course_id = course.course_id
LEFT JOIN instructor ON instructor.id = teaches.id
WHERE teaches.year IN
        (SELECT MAX(year)
        FROM teaches
        GROUP BY course_id
        HAVING COUNT(DISTINCT year) = 1
        ORDER BY MAX(year) DESC
        LIMIT 1)
    AND course.course_id IN
        (SELECT teaches.course_id
        FROM teaches
        GROUP BY course_id
        HAVING COUNT(DISTINCT teaches.year) = 1)
ORDER BY course.course_id;
```

Output:

course_id	title	instructor_name	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)

Figure 11: Output for 7

8: Find all the courses whose title has more than 15 characters and have a 'sys' as substring in the title. Consider case insensitive matching. 'sys', 'Sys', etc are all fine. Print the course id and title. Sort result by course id.

Query:

```
SELECT course_id, title
FROM course
WHERE LENGTH(title) > 15 AND title ILIKE '%sys%'
ORDER BY course_id;
```

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)

Figure 12: Output for 8

9: Find the department that offers the highest average salary to instructors.

Query:

```
SELECT dept_name, avg(salary) AS avg_salary
FROM instructor
GROUP BY dept_name
ORDER BY avg_salary DESC
LIMIT 1;
```

Output:

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

Figure 13: Output for 9

10: Find all instructors who taught at most once in 2003. (Didn't teach any course in 2003 or taught just one course in 2003). Print instructor id, name and department. Sort your result by instructor id.

Query:

```sql
SELECT instructor.id, instructor.name, instructor.dept_name
FROM instructor
LEFT JOIN
    (SELECT teaches.id AS teaches_id, COUNT(course_id) AS taught
    FROM teaches
    WHERE year = 2003
    GROUP BY teaches.id)
ON teaches_id = instructor.id
WHERE taught = 1 OR taught IS NULL
ORDER BY instructor.id;
```

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)

Figure 14: Output for 10