CS3550 DBMS 1 Assignment

Himanshu Jindal
Roll Number: MA21BTECH11007

November 21, 2023

Question 1 (a)

Find the top-3 instructors who have have taught most number of distinct courses from across all departments.

Listing 1: SQL Query

```
SELECT instructor.id, instructor.name, instructor. dept_name,
count(DISTINCT course_id) AS number_of_courses
FROM instructor
LEFT JOIN teaches
ON teaches.id = instructor.id
GROUP BY instructor.id
ORDER BY count(DISTINCT course_id) DESC
LIMIT 3;
```

Table Output

id	*name*	*dept_name*	*number_of_courses*
22591	DAgostino	Psychology	11
6569	Mingoz	Finance	8
99052	Dale	Cybernetics	8

Table 1: Top-3 instructors with the most distinct courses across all departments.

Question 1 (b)

Find the top-3 instructors who have have taught most number of distinct courses from the Statistics department.

Listing 2: SQL Query

```
SELECT instructor.id, instructor.name, instructor. dept_name,
count(DISTINCT course_id) AS number_of_courses
```

```
FROM instructor
LEFT JOIN teaches
ON teaches.id = instructor.id
WHERE instructor.dept_name = 'Statistics'
GROUP BY instructor.id, instructor.name
ORDER BY count(DISTINCT course_id) DESC
LIMIT 3;
```

Table Output

id	*name*	*dept_name*	*number_of_courses*
28400	Atanassov	Statistics	2
90643	Choll	Statistics	1
37687	Arias	Statistics	0

Table 2: Top-3 instructors with the most distinct courses in the Statistics department.

Question 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.

Listing 3: SQL Query

```
SELECT instructor.dept_name, course.course_id AS course_identifier,
course.title AS course_title, teaches.sec_id, teaches.semester,
teaches.year, instructor.salary, count(*) AS total_enrollment
FROM instructor
INNER JOIN teaches
ON instructor.id = teaches.id
INNER JOIN course
ON teaches.course_id = course.course_id
INNER JOIN takes
ON takes.course_id = course.course_id
WHERE instructor.id IN (
    SELECT id from instructor
    ORDER BY salary DESC
    LIMIT 1
)
GROUP BY instructor.dept_name, course.course_id,
course.title, teaches.sec_id,
teaches.semester, teaches.year, instructor.salary
ORDER BY course.course_id, teaches.year, teaches.semester;
```

Table Output

Table 3: Teaching record of the instructor with the highest salary.

dept_name	*course_identifier*	*course_title*	*sec_id*	*semester*	*year*	*salary*	*total_enrollment*
Pol. Sci.	545	International Practicum	1	Fall	2001	124651.41	306
Pol. Sci.	581	Calculus	1	Spring	2005	124651.41	313
Pol. Sci.	591	Shakespeare	1	Spring	2005	124651.41	291

Question 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.

Listing 4: SQL Query

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

Table Output

course_id	*course_title*	*dept_name*	*instructor_name*	*numer_of_students*	*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

Table 4: History of the course with course_id = 362.

Question 4

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

Listing 5: SQL Query

```
WITH course_dept AS (
    SELECT DISTINCT dept_name
    FROM course
    WHERE course_id = '319'
)
, student_in_course AS (
    SELECT id
    FROM takes
    WHERE course_id = '319' AND year = 2003
)

SELECT count(id) AS number_of_out_of_department_students
FROM student
WHERE id IN (SELECT id FROM student_in_course)
AND dept_name NOT IN (SELECT dept_name FROM course_dept);
```

Table Output

number_of_out_of_department_students
303

Table 5: Count of out-of-department student registration for course_id 319 in 2003.

Question 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.)

Listing 6: SQL Query

```
SELECT student.id, student.name,student.dept_name,
sum(course.credits) AS total_credits
FROM takes
INNER JOIN course
ON course.course_id=takes.course_id
INNER JOIN student
ON takes.id = student.id
GROUP BY takes.id,student.id
ORDER BY sum(course.credits) DESC , student.name ASC
```

```
LIMIT 3;
```

Table Output

id	*name*	*dept_name*	*total_credits*
12078	Knutson	Languages	93
90448	Godfrey	English	90
44551	Nguyen	Astronomy	90

Table 6: Top-3 students with the highest number of course credits.

Question 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.

Listing 7: SQL Query

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

Table Output

Table 7: Courses not offered during 2003 and 2004.

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

Table 7: Courses not offered during 2003 and 2004 (continued).

course_id	*title*
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

Table 7: Courses not offered during 2003 and 2004 (continued).

course_id	*title*
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

Table 7: Courses not offered during 2003 and 2004 (continued).

course_id	*title*
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

Table 7: Courses not offered during 2003 and 2004 (continued).

course_id	*title*
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

Question 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.]

Listing 8: SQL Query

```
WITH first_offered AS (SELECT course_id, MIN(year) AS year
FROM teaches
GROUP BY course_id)
 ,maxi AS (SELECT Max(year) AS year
FROM first_offered)
 ,recent_course_ids AS (SELECT first_offered.course_id
AS course_id, first_offered.year
FROM first_offered, maxi
WHERE first_offered.year = maxi.year
ORDER BY first_offered.course_id)

SELECT course.course_id, course.title, instructor.name
AS instructor_name, recent_course_ids.year
FROM instructor
INNER JOIN teaches
ON teaches.id=instructor.id
INNER JOIN recent_course_ids
ON teaches.course_id = recent_course_ids.course_id
INNER JOIN course
ON course.course_id = teaches.course_id
ORDER BY course.course_id;
```

Table 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

Table 8: Courses offered for the first time.

Question 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.

Listing 9: SQL Query

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

Table Output

course_id	*title*
353	Operating Systems
362	Embedded Systems
451	Database System Concepts
457	Systems Software
604	UNIX System Programming
791	Operating Systems
793	Decision Support Systems
857	UNIX System Programming
893	Systems Software
947	Real-Time Database Systems

Table 9: Courses with title length ¿ 15 and 'sys' as a substring.

Question 9

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

Listing 10: SQL Query

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

Table Output

dept_name	*avg_salary*
Physics	114576.90

Table 10: Department offering the highest average salary to instructors.

Question 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.

Listing 11: SQL Query

```
(SELECT instructor.id, instructor.name, instructor.dept_name
FROM instructor
WHERE instructor.id IN(SELECT id
FROM teaches
WHERE year = 2003
GROUP BY id
Having count(course_id)=1))
UNION
(SELECT instructor.id, instructor.name, instructor.dept_name
FROM instructor
WHERE instructor.id NOT IN(SELECT DISTINCT id
FROM teaches
WHERE year = 2003))
ORDER BY id;
```

Table 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)

Table 11: Instructors who taught at most once in 2003.