Department of Computer Science

Subject: Database Systems(COSC-2103) Fall-2024 Midterm Time Allowed: 60 Minutes MA

Rollnumber_____ Name_____

Q#1: DBMS provide interface to retrieve, manipulate and store data
Q#2: LIMIT, FROM, WHERE , HAVING, GROUP BY, SELECT ,arrange these keywords according to execution
FROM, WHERE, GROUP BY, HAVING, SELECT, LIMIT
Q#3: Describe the differences in meaning between the tuple and feature
Tuple is callted, feature is called Attribute
Q#4: When would you use the HAVING clause instead of the WHERE clause in a query?
Both are used to filter data. Where is use to filter data from a given table while having is used to apply filter on grouped data.
Q#5: Explain, SELECT name FROM people WHERE name NOT LIKE '%Ade'
Select name from people table that ends at Ade.
Q#6: How UNION is different from UNION ALL union combine data from two fields with duplication, union all combines with duplication
Q#7: Describe, SELECT COUNT (*) AS count_records FROM people; Provides total number of records in a table
Q#8: Raw facts and figures are calle Data
Q#9: SQL is the abbreviation of Structured Query Language
Q#10: Data is stored in table, image, audio or csv format which is structured, unstructured and semi structured. Table structured, image unstructured, audio unstructured, csv semi-structured

---

Question#1: Explain the Query below (5)

```
SELECT name, section rollnumber, MAX(gpa) AS max_gpa
FROM spring_23
GROUP BY section
ORDER BY max_gpa DESC
```

This query makes group according to the section and calculate maximum gpa in each section. The final output will be name, section, rollnumber and max_gpa. It will arrange output such that student having maximum gpa in all section will be on top.

Question#2:

Table 1: students

roll_num	name	city	phone_num
S21BDOCS1M1001	Ali	BWP	03001234567
S21BDOCS1M1002	Hamza	BWP	03011234567
S21BDOCS1M1003	Ali	Multan	03021234567
S21BDOCS1M1004	Farhan	LHR	03031234567
S21BDOCS1M1005	Usman	ISB	03456825117

Table 2: courses

roll_num	teacher_id	course_Code
S21BDOCS1M1001	0156	COSC-2001
S21BDOCS1M1002	1456	COSC-2002
S21BDOCS1M1003	1234	COSC-2003

Consider tables above,

1. Write a Query to show which student is taking course with which teacher for example the output table contain roll_number, name and teacher_id also write output table (2+1)

```
--solution#1

SELECT roll_num,
name , teacher_id
FROM studentss
INNER JOIN courses
Using(roll_num)
```

```
--solution#2
SELECT s.roll_num, s.name, c.teacher_id
FROM studentss AS s
FULL JOIN
courses AS c
USING (roll_num)
WHERE s.roll_num= c.roll_num
```

```
--solution#2
SELECT
   s.roll_num,
   s.name,
   (SELECT teacher_id FROM courses c
WHERE c.roll_num = s.roll_num) AS teacher_id
FROM
   students s
WHERE
   s.roll_num IN (SELECT roll_num FROM
courses);
```

roll_num character varying (20)	name character varying (50)	teacher_id integer
S21BDOCS1M1001	Ali	156
S21BDOCS1M1002	Hamza	1456
S21BDOCS1M1003	Ali	1234

2. Write a Query to Count number of students in student table. (2)

```
SELECT COUNT(roll_num)
FROM students
```

3. Write a Query to Show students' rollnumber and phonenumber who are not registered in an any course. (2)

```
SELECT roll_num, phone_num
FROM studentss AS s
WHERE roll_num NOT
IN(SELECT roll_num FROM
courses)
```

```
SELECT rollnumber, phone_num
FROM studentss
where roll_num IN(
SELECT roll_num
FROM studentss
FULL JOIN
courses
USING(roll_num)
WHERE teacher_id IS NULL
)
```

```
SELECT roll_num, phone_num
FROM studentss
WHERE roll_num IN(
SELECT roll_num
FROM studentss
EXCEPT
SELECT roll_num
FROM courses)
```

4. Write a Query to Count the number of cities from where students are coming for study purpose, show the output. (2+1)

```
SELECT COUNT(DISTINCT city)
FROM students.  Output =4
```

Question#3: List the names of employees whose salaries are below the average salary in their department.

Table 3: employees

e_id	e_name	department	salary
1	Zohaib	HR	50000
2	Farhan	IT	70000
3	Hamza	Sales	45000
4	Ali	IT	80000
5	Usman	HR	55000

1. Subquery Calculate the average salary for the department and show output (1.5+1)

```
SELECT department ,AVG(salary)
FROM employees
GROUP BY department
```

	department character varying (50)	avg numeric
1	Sales	45000.000000000000
2	IT	75000.000000000000
3	HR	52500.000000000000

2. Main Query - Use the result of the subquery to filter employes who salary is below the average in his department and show output. (1.5+1)

```
SELECT e_name
FROM employees as e1
WHERE salary <( SELECT AVG(salary)
FROM employees as e2
WHERE e1.department=e2.department)
```

```
--using self join
SELECT
   e1.e_name,
   e1.department,
   e1.salary
FROM
   employees e1
JOIN
   (SELECT department, AVG(salary) AS avg_salary
    FROM employees
    GROUP BY department) e2
ON
   e1.department = e2.department
WHERE
   e1.salary < e2.avg_salary;
```

	e_name character varying (50)
1	Zohaib
2	Farhan

Helpful queries:

Question #2: students table

```
-- Create the table
CREATE TABLE students (
    roll_num VARCHAR(20) PRIMARY KEY,
    name VARCHAR(50),
    city VARCHAR(50),
    phone_num VARCHAR(15)
);

-- Insert the data into the table
INSERT INTO students (roll_num, name, city, phone_num) VALUES
('S21BDOCS1M1001', 'Ali', 'BWP', '03001234567'),
('S21BDOCS1M1002', 'Hamza', 'BWP', '03011234567'),
('S21BDOCS1M1003', 'Ali', 'Multan', '03021234567'),
('S21BDOCS1M1004', 'Farhan', 'LHR', '03031234567'),
('S21BDOCS1M1005', 'Usman', 'ISB', '03456825117');
```

courses table

```
-- Create the table
CREATE TABLE courses (
    roll_num VARCHAR(20),
    teacher_id INT,
    course_code VARCHAR(20)
);

-- Insert the data into the table
INSERT INTO courses (roll_num, teacher_id, course_code) VALUES
('S21BDOCS1M1001', 0156, 'COSC-2001'),
('S21BDOCS1M1002', 1456, 'COSC-2002'),
('S21BDOCS1M1003', 1234, 'COSC-2003');
```

Question #3:

```
-- Create the table
CREATE TABLE employees (
```

```
    e_id INT PRIMARY KEY,
    e_name VARCHAR(50),
    department VARCHAR(50),
    salary INT
);

-- Insert the data into the table
INSERT INTO employees (e_id, e_name, department, salary) VALUES
(1, 'Zohaib', 'HR', 50000),
(2, 'Farhan', 'IT', 70000),
(3, 'Hamza', 'Sales', 45000),
(4, 'Ali', 'IT', 80000),
(5, 'Usman', 'HR', 55000);

Part 1 solution
SELECT department ,AVG(salary)
FROM employees
GROUP BY department
Part 2 solution
SELECT e_name
FROM employees as e1
WHERE salary <( SELECT AVG(salary)
FROM employees as e2
WHERE e1.department=e2.department)
```