

## Exam 2

● Graded

Student

AKSHAJ KAMMARI

Total Points

23.5 / 60 pts

Question 1

1

5 / 5 pts

+ 0 pts Incorrect

💬 + 5 pts Point adjustment

Question 2

2

📌 0 / 5 pts

✓ + 0 pts Incorrect

💬 P, D => R => O

Question 3

3

📌 0.5 / 5 pts

+ 0 pts -

💬 + 0.5 pts Incorrect.

Question 4

4

📌 2 / 5 pts

+ 0 pts -

💬 + 2 pts Incorrect.

Question 5

5

📌 3 / 5 pts

+ 0 pts -

💬 + 3 pts use like M%

Question 6

6

📌 2 / 5 pts

+ 0 pts -

💬 + 2 pts SQL query is wrong.

Question 7

7

2 / 5 pts

+ 0 pts -

🗨 + 2 pts cannot use average this way.

Question 8

8

2 / 5 pts

+ 1 pt Correct

+ 0 pts Click here to replace this description.

+ 0.5 pts Click here to replace this description.

✓ + 2 pts Click here to replace this description.

+ 3 pts Click here to replace this description.

+ 4 pts Click here to replace this description.

+ 5 pts Click here to replace this description.

Question 9

9

2 / 5 pts

✓ + 2 pts Correct

+ 0 pts Click here to replace this description.

+ 3 pts Click here to replace this description.

+ 4 pts Click here to replace this description.

+ 5 pts Click here to replace this description.

+ 1 pt Click here to replace this description.

Question 10

10

1 / 5 pts

+ 0 pts Correct

✓ + 1 pt Click here to replace this description.

+ 2 pts Click here to replace this description.

+ 3 pts Click here to replace this description.

+ 4 pts Click here to replace this description.

+ 5 pts Click here to replace this description.

Question 11

11

2 / 5 pts

+ 0 pts Correct

+ 1 pt Click here to replace this description.

✓ + 2 pts Click here to replace this description.

+ 3 pts Click here to replace this description.

+ 4 pts Click here to replace this description.

+ 5 pts Click here to replace this description.

Question 12

12

2 / 5 pts

+ 0 pts Correct

+ 1 pt Click here to replace this description.

✓ + 2 pts Click here to replace this description.

+ 3 pts Click here to replace this description.

+ 4 pts Click here to replace this description.

+ 5 pts Click here to replace this description.

Name: Akshaj KamnariNetID: ak1990**A. MULTIPLE CHOICE (2 points each)**

Mark on the answer sheet provided the option that best answers each question. Make sure to completely fill each bubble. Pencil or pen is OK.

1. SQL does not include
  - A) A query language
  - B) A schema definition language
  - ☒ C) A programming language
  - D) A data manipulation language
2. `SELECT R.a, R.b from R join S using (c)` assumes that
  - A) c is a field of R but not of S
  - ☒ B) c is a field of R and S
  - C) c is a field of S but not R
  - D) c is not a common field
3. Which of the following SQL instructions might have duplicates
  - A) UNION
  - B) INTERSECT
  - ☒ C) JOIN
  - D) EXCEPT
4. `Select a,b from R union Select c,d from S` produces a table with
  - ☒ A) two columns
  - B) three columns
  - C) no columns
  - D) four columns
5. A condition on `count(*)` can be included in a SQL query after `GROUP BY` using
  - ☒ A) HAVING
  - B) WHERE
  - C) CASE
  - D) IF
6. Result tables from SQL queries
  - ☒ A) can have duplicates
  - B) cannot have duplicates
  - C) are always sorted by id
  - D) always have a key
7. A foreign key must reference
  - A) any field combination of another table
  - ☒ B) all the primary key fields of another table
  - C) some of the primary key fields of another table
  - D) just one field of another table, even if it is not the complete primary key
8. `SELECT * FROM A,B;` computes
  - A)  $A \cup B$
  - B)  $A - B$
  - ☒ C)  $A \times B$
  - D)  $A \cap B$
9. Result tables in SQL are
  - A) Sets
  - B) Relations
  - ☒ C) Lists
  - D) Queries
10. SQL stands for
  - A) Sequel
  - ☒ B) Structured Query Language
  - C) Relational Database System
  - D) Simple Query Logic

B. Problems (5 points each)

Write Only in the space provided for each question.

Given the following relations:

- registered(pnum:integer, hospital:string)
- operation(underlinepnum:integer, hospital:string, when:date-time, op\_room:string, doc:integer)
- doctor(doc:integer, dname:string, dept:string)
- patient(pnum:integer, pname:string, illness:string, age:integer)

Provide SQL instructions for each of the following questions. You cannot use instructions not covered in class such as NVL.

1. Create the database schema including primary/foreign key constraints. If you need assumptions, write them down.

```
create table registered ( pnum: int,  
                        hospital: varchar(50),  
                        primary key (pnum, hospital);  
                        foreign key (pnum) references patient);  
  
create table operation ( pnum: int,  
                        hospital: varchar(50),  
                        when: datetime,  
                        op_room: varchar(30),  
                        doc: int,  
                        primary key (pnum, when, doc),  
                        foreign key (pnum) references registered,  
                        foreign key (hospital) references registered,  
                        foreign key (doc) references doctor);  
  
create table doctor ( doc: int,  
                     dname: varchar(50),  
                     dept: varchar(20),  
                     primary key (doc));  
  
create table patient ( pnum: int,  
                      pname: varchar(50),  
                      illness: varchar(100),  
                      age: int,  
                      primary key (pnum));
```

2. If populating the tables with outside data, what is the correct order to fill the tables?

From left to right, primary keys, attributes, and then foreign keys.

3. Use set operations to determine the names of doctors who operated in a hospital that has patients registered for covid and cancer.

```
SELECT doc FROM operation
WHERE patient.illness = 'covid' and patient.illness = 'cancer';
```

4. Find the names of patients older than 'Besiana' (assume that there is only one patient called 'Besiana').

```
SELECT age FROM patient WHERE pname = 'Besiana';
SELECT * FROM patient WHERE p.age > age;
```

Group by pname;

5. Find the names of all cancer patients whose name starts with 'M'.

```
SELECT * p.pname  
FROM patient  
WHERE illness = 'cancer' and pname [0] = 'M',  
Group by pname;
```

6. Names of all patients operated for appendicitis in 'Princeton-Plainsboro' hospital.

```
SELECT * pname referencing pnum  
FROM operation  
WHERE hospital = 'Princeton-Plainsboro' and p-illness = 'appendicitis'  
FROM patient,  
Group by pname;
```

7. Find the name of all patients younger than average.

```
SELECT * pname FROM patient  
WHERE age < avg(age)  
Group by pname;
```

8. Find the hospital(s) with the maximum number of registered patients.

```
SELECT * hospital FROM registered  
WHERE pnum = max;
```



9. For each doctor (name) determine the names the oldest patients among those operated by the doctor.

```
SELECT dname referencing doctor.  
FROM operation  
WHERE pname referencing age > * p.age  
FROM patient  
Group by dname;
```

10. Use outer join (and any other necessary instructions) to determine the number of hospitals in which each doctor operates. Your result must include all doctors, even if they did not perform any operations.

```
SELECT * FROM doctor  
OUTER JOIN doc, hospital
```

11. Produce a table containing the number of operations performed in each hospital by a doctor that works in the oncology, ENT, OB-GYN, Neurology, or Cardiology department.

Hospital	Oncology	ENT	OB-GYN	Neurology	Cardiology

For example, if one of the tuples produced by your query is:

Princeton Plainsboro	22	3	138	15	239
----------------------	----	---	-----	----	-----

It means that there were 22 operations performed at Princeton Plainsboro Hospital by Oncology department doctors, 3 by ENT doctors, etc.

```
SELECT hospital:
sum(if(hospital name = 'princeton plainsboro', 22, 3, 138, 15,
cardiology = 239))
```

```
FROM hospital
GROUP BY hospital;
```

12. USE CORRELATED subqueries to find the names of doctors who did not perform any operations (no credit if you did not use correlated subqueries).

```
SELECT dname  
FROM doctor  
WHERE operations = 0
```

```
FROM doctor,  
GROUP BY dname;
```