Do not turn this page until you have received the signal to start.
(In the meantime, please fill out the identification section above,
and read the instructions below.)

This test consists of 4 questions on 8 pages (including this one). When
you receive the signal to start, please make sure that your copy of the
quiz is complete.

If you use any space for rough work or need to scratch out an answer,
circle the part that you want us to mark.

Good Luck!
Question 1.  [11 marks]

Recall this schema, which we have used many times in class. Here we are adding one more relation called Program. It records the subject POSets that students are enrolled in. (“POSt” is short for “program of study”, by the way.)

Relations

- Student(sID, surName, firstName, campus, email, cgpa)
- Course(dept, cNum, name, breadth)
- Offering(oID, dept, cNum, term, instructor)
- Took(sID, oID, grade)
- Program(sID, POSt)

Integrity constraints

- Offering[dept, cNum] ⊆ Course[dept, cNum]
- Took[sID] ⊆ Student[sID]
- Took[oID] ⊆ Offering[oID]
- Program[sID] ⊆ Student[sID]

Part (a)  [7 marks]

Write a query to find the cNum of every CSC course that has been taught in the term whose value is 20129, and in every term after 20129. (You may use comparison operators such as < to compare terms.) Write your query using only the basic operators Π, σ, ⋈, ×, ⋃, ⋂, −, ρ, and assignment.
Part (b)  [4 marks]

Consider the following query:

\[ \text{Temp1}(\text{instructor}, oID, \text{term}) := \Pi_{\text{instructor}, oID, \text{term}} \sigma_{\text{dept} = 'CSC'} \text{Offering} \]

\[ \text{Temp2}(\text{instructor}) := \Pi_{\text{O1.instructor}} \sigma_{\text{O1.instructor} = \text{O2.instructor} \land \text{O1.oID} \neq \text{O2.oID} \land \text{O1.\text{term}} = \text{O2.\text{term}}} (\rho_{\text{O1.Temp1 \times \text{O2.Temp1}}}) \]

\[ \text{Answer}(\text{instructor}) := (\Pi_{\text{instructor}} \text{Temp1}) - \text{Temp2} \]

1. Below is an instance of the relation that is relevant to this query, \text{Offering}. Add the fewest possible rows to \text{Offering} so that professors Able and Bland will not appear in the result of the query, but professors Cranky and Devlish will.

\begin{center}
\begin{tabular}{|c|c|c|c|c|}
\hline
\text{oID} & \text{dept} & \text{cNum} & \text{term} & \text{instructor} \\
\hline
\text{O3} & \text{CSC} & 324 & 1 & \text{Able} \\
\text{O6} & \text{CSC} & 324 & 4 & \text{Able} \\
\text{O1} & \text{CSC} & 443 & 5 & \text{Devlish} \\
\text{O7} & \text{CSC} & 148 & 2 & \text{Bland} \\
\text{O8} & \text{ECE} & 345 & 2 & \text{Bland} \\
\text{O9} & \text{ANT} & 101 & 6 & \text{Cranky} \\
\hline
\end{tabular}
\end{center}

2. What does this query compute? Do not describe the steps it takes, only what is in the result, and make your answer general to any instance of the schema.

\text{All instructors who ...}
Question 2.  [6 marks]

Part (a)  [2 marks]

Does our schema enforce the following constraint:

Every course has been offered.

Circle one answer. If the statement is enforced, say what part of the schema enforces it. If it is not enforced, write an integrity constraint that would enforce it, using the form $R = \emptyset$.

Enforced  This part of the schema enforces it:

Not enforced  This new integrity constraint would enforce it:

Part (b)  [4 marks]

Consider this schema:

$A$(alpha, beta, gamma)  $B[delta] \subseteq A[beta]$

$B$(delta, epsilon)  $B[epsilon] \subseteq C[iota]$

$C$(iota, kappa)

Suppose relation $B$ has 50 tuples. How many tuples could $A$ have? Circle all answers that do not violate the schema.

0  1  23  50  128

Suppose relation $B$ has 50 tuples. How many tuples could $C$ have? Circle all answers that do not violate the schema.

0  1  23  50  128
Question 3. [5 MARKS]

The question refers to the schema from Question 1. Write a query in SQL to find the terms in which at least twenty-five grades greater than 90 were given. Note that one student can be given multiple grades greater than 90 in a single term. Report only the terms.
Question 4.  [8 marks]

Part (a)  [3 marks]
Consider the same schema from the Question 1. Suppose we wrote the query

```
SELECT ________________________
FROM Offering, Course
WHERE Offering.dept = Course.dept AND Offering.cNum = Course.cNum
GROUP BY Offering.dept;
```

Which of the following could go in the `SELECT` clause? Circle all that apply.

```
count(term)   Offering.cNum   Offering.dept   avg(cNum)
Course.dept   count(Course.cNum)   dept
```

Part (b)  [3 marks]

We discussed in lecture how the SQL subquery operators could possibly be implemented using other SQL operations. Suppose we have two tables `R(a,b)` and `S(b,c)`.

Consider the following two queries:

```
-- Query 1
SELECT a
FROM R
WHERE b IN (SELECT b FROM S);

-- Query 2
SELECT R.a
FROM R, S
WHERE R.b = S.b;
```

On the next page, give a database instance where these two queries produce different results, and the results of the two queries. Clearly label which result belongs to which query!
Instance of the database where the two queries produce different results:

Result of Query 1 on this instance: Result of Query 2 on this instance:

**Part (c) [2 MARKS]**

Fix Query 2 so that it produces the same results as Query 1 for any valid dataset. Your answer must still include a Cartesian product with R. You may not use the keywords ANY, ALL, IN, EXISTS in your solution. **Hint**: create a view.
This page is left (mainly) blank for things that don’t fit elsewhere.

# 1: _____/11
# 2: _____/ 6
# 3: _____/ 5
# 4: _____/ 8

TOTAL: _____/30