# Final Exam CSE 132B Winter 2015

The exam is due back Tuesday March 17, at noon, by sliding a hardcopy under Alin's office door (CSE 3238).
This is an individual exam, no collaboration is allowed. Good luck!

Your name: _____ Student ID#: _____

| P1 | P2 | P3 | P4 | P5 | Total |
|---|---|---|---|---|---|
| | | | | | |

**Problem 1** *(30 pts).* Consider the following schema:

STUDENT(Id, StName, Age)
TRANSCRIPT(StudId, CrsCode, Semester, Grade)
TEACHING(ProfId, CrsCode, Semester)
PROFESSOR(Id, ProfName, Dept, Sal)

1. *(10pts)* Find the professors whose salaries are at least 10% higher than the average salary of all professors *in their departments.*

Text

2. *(15pts) Find the Ids of all students who have taken a course from* each *professor in the* CS *Department. Do not use count, or any other aggregation.*

3. *(5pts)* Define the above query as an SQL view and then use this view to answer the following query: *For each student who has taken a course from every professor in the* CS *Department, show the number of courses taken, provided that this number is more than 10.*

**Problem 2** *(20 pts).* Suppose we have a database for an investment firm, consisting of the following attributes: $B$ (broker), $O$ (office of a broker), $I$ (investor), $S$ (stock), $Q$ (quantity of stoclk owned by an investor), and $D$ (dividend paid by a stock), with the following functional dependencies:

$$S \to D, I \to B, IS \to Q, B \to O.$$

.

1. Find a minimal key for the relation scheme $R = BOSQID$ and prove it is a key.

2. How many minimal keys does relation scheme $R$ have? Prove your answer.

3. Suppose we decomposed relation $R$ into the two relations $R_1 = ISQD$ and $R_2 = IBO$. Is this a BCNF decomposition? Is it a 3NF decomposition? Justify your answers.

**Problem 3** *(15 pts).* Given a view definition, we call *base tables* the tables mentioned in the FROM clause of the query defining the view (the query following the CREATE VIEW V AS command). We say that a view is *updatable* if any update $U_V$ on the view can be translated to updates $U_B$ of the base tables such that

- we obtain the same result if we (a) first update the base tables applying $U_B$ and then run the view query, or (b) first run the view query on the original base tables and then update the result according to $U_V$.
- Moreover, there is a unique base table update $U_B$ with this effect.

A sufficient set of conditions to ensure view updatability is the following:

1. Exactly one table can be mentioned in the FROM clause (and only once). The FROM clause cannot have nested subqueries.
2. Aggregates, GROUP BY, or HAVING clauses and set operations, such as UNION and EXCEPT, are not allowed.
3. Nested subqueries in the WHERE clause of the view cannot refer to the table used in the FROM clause of the view definition. Moreover, a nested subquery cannot refer to this table, either explicitly, in the FROM clause, or implicitly, through a tuple variable defined in the outer query.
4. No expressions and no DISTINCT keyword in the SELECT clause are allowed (i.e. you may only use column names of the unique table, and constants).

Provide a base schema and construct a non-updatable view that violates condition 3 but satisfies conditions 1, 2 and 4 above, such that there is an update to this view that has two different translations into updates on the underlying base schema. Show the update and the two translations.

**Problem 4** *(15 pts).* Create a view V of TRANSCRIPT containing only rows corresponding to classes taught by John Doe. John Doe wants to update the student grades he has assigned, using the view V.

1. Does V satisfy the sufficient set of conditions from Problem 3?
2. Is V updatable according to the definition from Problem 3?
3. Capture the modifications to V (only insertions and deletions) using INSTEAD-OF triggers, and translate them to corresponding modifications of the base tables.

**Problem 5** *(20 pts).* This problem refers to the setting of Problem 4.

1. Materialize the view of Problem 4 into a table M (using a CREATE TABLE M and an INSERT INTO M command).

2. Identify *all* triggers needed to keep M up to date (list the events these triggers must fire on; for update events, also specify the updated attribute).

3. Among the list of triggers above, write a trigger reacting to the update of a student's grade. Avoid blindly recomputing the entire view M, instead updating it incrementally as discussed in class. Use standard SQL syntax (as in the Ullman& Widom book chapter).