CSE 4/560: Project #2 (Due 23:59 11/13/19 EST)

Academic Integrity

The work of this project has to be finished as **INDIVIDUAL WORK**. Copied solutions will be considered violations of academic integrity.

Submission

A zip file has to be submitted through the 'submit_cse460' or 'submit_cse560' script by 11/13/19 23:59 EST. **ONLY** zip extension will be accepted. Zip file naming convection:

- Use *ubit_proj2.zip* (**NO SPACE**) for the filename, e.g., *jsmith_proj2.zip*, where *jsmith* is the ubit.
- After unzipping the submission zip, there should be a folder named as *ubit_proj2*, where *ubit* is your ubit.
- Under the folder *ubit_proj2*, there should be a SQL file. Name your SQL file as ***ubit_proj2.sql***. Use comments to explain your answers.

Problem 1 (9 pts)

You are given the following relational schema (keys underlined):

```
Category(Cname)
Vendor(Vname, Cname, Revenue)
Sell(Vname, Pname, Price)
Product(Pname)
```

Foreign keys:

- `Cname` in `Vendor` references `Category(Cname)`
- `Vname` in `Sell` references `Vendor(Vname)`.
- `Pname` in `Sell` references `Product(Pname)`

Write the following queries in SQL:

- S_1: Find the names of all vendors belonging to the categories with the maximum number of vendors.
- S_2: Find the names of the products sold by vendors whose revenue is greater than 2 million.
- S_3: For every vendor, calculate the average price of products for vendors that are in the same category and have a higher revenue.

Which of the above queries has an equivalent relational algebra formulation? Explain your answer. For the queries that have an equivalent relational algebra formulation provide this formulation.

Problem 2 (6 pts)

You are supposed to represent directed labelled graphs in SQL. A directed labelled graph consists of nodes and directed labelled edges connecting the nodes. We assume the labels are colors.

- Define an appropriate SQL schema.
- Write the following queries in SQL2 or SQL3:
  - Find all pairs of nodes connected by a path consisting of green or yellow edges (SQL3). (The edges in the path may be of different colors, as long as each is green or yellow.)
  - For every node, calculate the number of outgoing edges (SQL2).
  - Find all pairs of nodes not connected by any path (SQL3).
  - A *triangle* is a set of nodes a, b and c such that there are edges (a, b), (b, c) and (c, a) in the graph. Find all the triangles of the same color (SQL2: **extra credit)**.

Problem 3 (2 pts, extra credit)

Assume you have a relation $Manager(\underline{SSN}, ManSSN)$. Write an SQL query that returns `'yes'` if $Manager$ satisfies the foreign key constraint "*ManSSN references Manager(SSN)*" and `'no'` otherwise.You can assume the presence of a nonempty relation R.