This homework should be done in parts as soon as relevant topics are covered in lectures (e.g. start with RA asap!). If you wait until the last minute, you might be overwhelmed.

Before you start working on the problems, please make sure you have understood the homework policy (see lecture 2 slides, announcements at the end).

For Problems 1, 3, and 5, you will need to use Gradiance. Gradiance is an online testing system that provides immediate feedback to your answers, and allows you to retry a problem multiple times until you get it right. Please read the “Help → Getting Started with Gradiance” section of the course website for instructions on how to get started. There is no need to turn in anything else for these problems; your scores will be tracked automatically. The gradiance problemsets will be posted “after” the corresponding lectures (by 8 pm).

For other problems, you will need to turn in the required files electronically. Please read the “Help → Submitting Non-Gradiance Work” section of the course website for instructions. For Problems 4 and X1, you may prepare your answers electronically or on paper (handwritten). In the latter case, scan or photograph the pages and submit the resulting PDF (preferred) or JPG files. Please name your files informatively, e.g., as n.pdf, where n is the problem number.

Problem 2 should be completed on a virtual machine (VM). However, you can start solving the questions on paper and then test them on the VM. Please read the VM-related section under “Help” on the course website, and follow the instructions therein to get your VM running. Post a question on piazza if you have questions Google Cloud credits, if you opt to run your VM on Google Cloud.

Problem 1 (13 points)
Complete the Gradiance homework titled “Homework 1.1 (Relational Algebra Basics).” Note that some of the online exercises use English names of relational algebra operators instead of symbols.

Problem 2 (32 points)
Consider a database containing information about bars, beers, and bar-goers.

```
Drinker(name, address)
Bar(name, address)
Beer(name, brewer)
Frequents(drinker, bar, times_a_week)
Likes(drinker, beer)
Serves(bar, beer, price)
```
Write the following queries in relational algebra using RA, our homegrown relational algebra interpreter. To set up the sample database called beers, issue this command in your VM shell:

```
/opt/dbcourse/examples/db-beers/setup.sh
```

Then, type “ra beers” to run RA. See “Help → RA: A Relational Algebra Interpreter” on the course website for additional instructions on using RA, including syntax of relational operators.

When you run RA, as soon as you get a working solution, record the query in a plain-text file named 2-queries.txt (use Java/C++-style comments in the file to indicate which problem each query corresponds to). When you are done with all queries, run

```
ra beers 2-queries.txt > 2-answers.txt
```

to generate the final answers. Submit the files 2-queries.txt and 2-answers.txt through the submission website. If you cannot get a query to parse correctly or return the right answer, include your best attempt and explain it in comments, to earn possible partial credit.

(UPDATE: in red – the constants have been fixed)

(a) Find names of beers that Dan likes.
(b) Find names and addresses of bars that serve some beer for more than $3.00 each.
(c) Find names of bars serving some beer of price > $2.50 that Eve does not like.
(d) Find names of pairs of drinkers who frequent some bar the same number of times a week. (Just list the drinker names, not the bar. Don’t list \((\text{drinker}_A, \text{drinker}_A)\). If you list \((\text{drinker}_A, \text{drinker}_A)\) in the answer, don’t list \((\text{drinker}_B, \text{drinker}_A)\) again.)
(e) Find names of all drinkers who frequent Talk of the Town pub and like all beers served there.
(f) For each drinker, find the beers that they like and that are served by at least two bars they frequent.
(g) Find names of all drinkers who frequent only those bars that serve only beers they like.
(h) Find names of all drinkers who frequent only those bars that serve some beers they like.

Problem 3 (15 points)
Complete the Gradiance homework titled “Homework 1.3 (E/R Design).”

Problem 4 (20 points)
Since you are an expert in “art of database design” using E/R diagram, you thought why not use “database design for art”, so have taken up the challenge of designing a database for art galleries that you plan to sell to art galleries as a product. You want your design to be fairly flexible and generic (you never know what the customers would want at the end), and have come up with the following requirements:

- Galleries are likely to keep information about artists, assign them an id, their names, birthplaces (if known), date of birth, date of death (if holds), and style of art (impressionism, post-impressionism, expressionism, contemporary, surrealism, etc).
- For each piece of artwork, they need to store the artist, the year it was made, its title, a unique id, whether it is for sale or for display only, and its quoted price (optional). Note that each piece is created by exactly one artist. For instance, “The Thinker” is a sculpture made by Rodin but “The Starry Night” is a painting by Van Gogh (not that they are for sale though 😊).
- However, there are two types of artwork: “unique-art” (painting or sculpture) and “photograph”. As the name suggests, there is a single copy of unique art, but multiple copies of photographs can be
created on demand. As additional information, for “photograph”-s you store the number of copies that have been sold so far, and what “camera” has been used. For a “unique-art”, you store whether it is still available.

- Pieces of artwork are also classified into “groups” of various kinds, for example, portraits, still lifes, works by Picasso, or works of the 19th century; a given piece may belong to more than one group. Each group is identified by a name (like the above examples) that describes the group.
- Finally, galleries would want to keep information about all customers till date. For each customer, galleries keep that person’s customer-id, name, address, total dollars spent in the gallery (very important - always try to attract customers who buy often!), date of last purchase, and both the artists and groups of art that the customer tends to like (may be multiple or none). In addition, the galleries keep track of which customer had purchased which artwork, and information on the purchase, i.e. the date of purchase and the actual price paid (after negotiation or auctions). Of course, one artwork of “unique-art” type can be purchased by only one customer, “photographs” can be purchased by multiple customers (and one customer may purchase several copies of the same photograph), and both types may not have been purchased by anyone yet.

If you think some aspects of the above description are unclear, feel free to make additional, reasonable assumptions, but state them clearly in your answer. Also, keep in mind that there is no single “correct” design; if you think you are making a non-obvious design decision, please explain it briefly.

(a) Design an E/R diagram for this database. Very briefly explain the intuitive meaning of any entity and relationship sets. Do not forget to indicate keys and multiplicity of relationships, as well as ISA relationships and weak entity sets (if any), using appropriate notation.

(b) Design a relational schema for this database. (You can start by translating the E/R design.) You may ignore attribute types, and you do not need to show any sample data. Indicate all keys and non-trivial functional dependencies in the schema. Check if the schema is in BCNF. If not, decompose the schema into BCNF.

**Problem 5 (20 points)**

Complete the Gradiance homework titled “Homework 1.5 (Relational Design Theory: FD).”

**Extra Credit Problem X1 (5 points)**

Using the schema in Problem 2, write RA queries to answer the following:

- list all drinkers that frequent at least two bars that serve only beers they like (i.e. among the bars that serve only beers they like, the drinker frequents at least two).

**Extra Credit Problem X2 (5 points)**

As discussed in class, the core operators in relational algebra are selection ($\sigma_p$), projection ($\pi_L$), cross product ($\times$), union ($\cup$), and difference ($-$). Prove that the selection operator is necessary; that is, some queries that use this operator cannot be expressed using any combination of the other operators.