CPS 196.3 Fall 2003
Homework #1
Assigned: Wednesday, September 3
Due: Friday, September 12

Problem 1.

Consider a database containing information about bars, beers, and bar-goers.

Drinker (*name*, *address*)
Bar (*name*, *address*)
Frequents (*drinker*, *bar*, *times_a_week*)
Likes (*drinker*, *beer*)
Serves (*bar*, *beer*, *price*)

Write the following queries in *relational algebra*. You may use expression trees to improve readability.

(a) Find all drinkers who frequent James Joyce Pub.
(b) Find all bars that serve both Amstel and Corona.
(c) Find all bars that serve at least one of the beers Amy likes for no more than $2.50.
(d) For each bar, find all beers served at this bar that are liked by none of the drinkers who frequent that bar.
(e) Find all drinkers who frequent *only* those bars that serve some beers they like.
(f) Find all drinkers who frequent *every* bar that serves some beers they like.

Optional: You can test your queries on a sample database on `rack40`. Follow the instructions at `http://www.cs.duke.edu/courses/fall03/cps196.3/faqs/login.html` to log into `rack40` and set up the environment for DB2. Then, run `/home/dbcourse/examples/db-beers/setup.sh` to setup a database with some sample data. Use the command `ra` to test your relational algebra queries. For instructions on using `ra`, please refer to `http://www.cs.duke.edu/courses/fall03/cps196.3/faqs/ra.html`. You can submit a script of running `ra`, showing all the queries and answers.

Problem 2.

As discussed in class, the core operators in relational algebra are selection (σ_p), projection (π_L), cross product ($\times$), union ($\cup$), and difference ($-$).

(a) Show that the projection operator is necessary; that is, some queries that use the projection operator cannot be expressed using any combination of the other operators.
(b) Show that the selection operator is necessary; that is, some queries that use the selection operator cannot be expressed using any combination of the other operators.

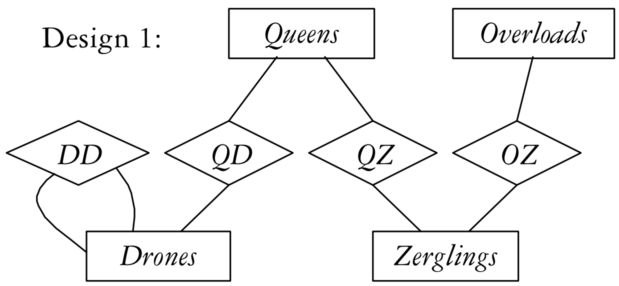
Problem 3.

(a) You are hired by the Terran Space Commission (apologies to Blizzard Entertainment) to design an E/R diagram for a genealogical database of an alien race known as the Zerg. There are four species of Zerg: Queens, Overlords, Zerglings and Drones.

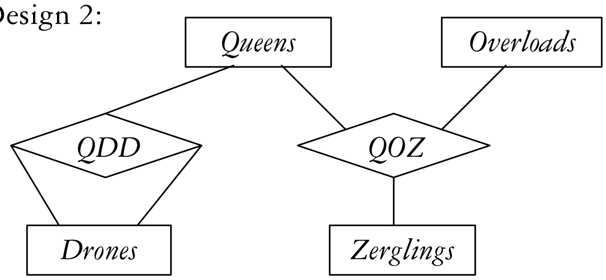
- A Zergling is an offspring of a Queen and an Overlord. A Queen and an Overlord together can produce any number of Zerglings. Each Queen may mate with multiple Overlords, and each Overlord may mate with multiple Queens.
- A Drone is the offspring of a Queen and another Drone. Each Drone may have only one offspring in its lifetime, although each Queen may mate with multiple Drones and thus have multiple Drone offsprings.

One possible design uses four binary parent-offspring relationship sets. Another possible design uses two ternary parent-parent-offspring relationship sets. The (incomplete) E/R diagrams for these two designs are shown below.

Design 1:



Design 2:



i. Complete the E/R diagrams above by adding arrowheads to indicate the multiplicity of relationships. Also put labels on the edges to indicate roles whenever necessary. You may ignore the attributes of the entity sets.

ii. Do both design faithfully capture all known facts about Zerg genealogy stated above? Or are there any assumptions that cannot be encoded exactly by the arrowhead notation?

(b) The Commission asks you to do another design for an alien race known as the Protoss. Protoss uses gene-splicing technology to reproduce. Genes from one or more Protoss are combined into a new gene for the offspring; in other words, each Protoss has one or more parents. A Protoss can have any number of offsprings. Draw the E/R diagram for your design. Again, you may ignore the attributes.

Problem 4.

In this problem you will explore the other side of the dot-com. Your job is to design a database that holds information about startups and venture capital funds that invest in these startups.

- Every venture capital (VC) fund is identified by its name and number, e.g., "Kleiner-Perkins, VII." A VC fund also has a size, which is the amount of money to be invested by the fund, and a closing date, by which the fund must invest the money.
- VC funds invest in two types of startup companies: stealth and private.
- Every startup has a name, an address, and belongs to exactly one industry. In general, there may be several companies with the same name, but within each industry company names are unique.
- Every industry has a unique name, market size, and consists of several sectors. Sectors have names and projected growths. Sector names are unique within an industry but not across industries.
- Stealth companies operate in a "stealth mode" and have a buzz factor. A stealth company is financed by at most one VC fund since it is difficult to remain in stealth mode when several sets of lawyers get involved.
- A private company has a Web site and a CEO. Every private company belongs to exactly one industry sector. For ethical reasons, VC funds invest in at most one company in any given industry sector.
- Finally, stealth companies may be acquisition targets of private companies.

(a) Give an E/R design for this database. Very briefly explain the intuitive meaning of any entity and relationship sets. Do not forget to indicate any keys, multiplicity of relationships, and weak entity sets in their appropriate ways.
(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.

Consider a relation $R(A, B, C, D)$ with FD's $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow B$.

(a) Show that $\{A, B\}$ is a key of R (remember a key has to be minimal).
(b) What are the other keys of R? (Hint: A must be in every key of R; why?)
(c) $D \rightarrow B$ is a BCNF violation. Using this violation, we decompose R into $R_1(B, D)$ and $R_2(A, C, D)$. What are the keys of R_1?
(d) What are the FD's that hold in R_1? Do not list them all; instead, give a set of FD's from which all other FD's in R_1 follow. This set of FD's is called a *basis*. When checking for BCNF violations, it suffices to check just the basis.
(e) Is R_1 in BCNF? Briefly explain why.
(f) What are the keys of R_2? (Hint: There is more than one.)
(g) What are the FD's that hold in R_2? Again, do not list them all; instead, give a basis.
(h) Is R_2 in BCNF? If yes, briefly explain why. Otherwise, decompose further until all decomposed relations are in BCNF, and then show your final results.