csc343 winter 2021
assignment #3: database (re)design
due April 2nd, 4 p.m.

goals

This assignment aims to help you learn to:

- design a good schema
- understand violations of functional dependencies
- create a minimal basis for a set of functional dependencies
- project a set of functional dependencies onto a set of attributes
- find all the keys for a set of functional dependencies
- re-factor relation(s) into BCNF
- re-factor relations into 3NF

Your assignment must be typed to produce a PDF document **a3.pdf**, and a plain text document **reservation.ddl** (hand-written submissions are not acceptable). You may work on the assignment in groups of 1 or 2, and submit a single assignment for the entire group on MarkUs. You must establish your group well before the due date by submitting an incomplete, or even empty, submission (of course, we only grade the last submission before the due date/time).

exercises

1. Relation *Reservation* is meant to keep track of which skipper reserves which craft on which date, but its design has some redundancy:

   $$Reservation(sID, age, length, sName, day, cName, rating, cID)$$

   ... where sID identifies the skipper, $sName$ is the skipper's name, whereas $rating$ and age record the skipper's skill (a number between 0 and 5, inclusive) and age (a number greater than 0). The reserved craft is identified by cID, its name is $cName$, $length$ is in feet, and the date and time the craft is reserved for by the skipper is day. The following dependencies hold:[1]

   $$S = \{sID \rightarrow sName, rating, age;\ \ cID \rightarrow cName, length\}$$

[1] Since multiple attributes may be separated by commas, we use semicolons to separate FDs.

(a) Give one example of a redundancy that relation $Reservation$, combined with FDs S, allow.

(b) Design a schema in DDL called `reservation.ddl` that represents the same information as $Reservation$, using exactly the same attribute names, but has the following goals, in descending order of importance:

i. has as few redundancies as possible;

ii. allows as few NULL or DEFAULT values as possible;

iii. enforces as many constraints from the description above as possible, without using triggers or assertions.

Your schema should import into psql without error using the command:

```
\i reservation.ddl
```

While you are developing your schema you may want to ensure that your previous version is removed before you read in a new one:

```
drop schema if exists reservation cascade;
create schema reservation;
set search_path to reservation;
```

Use comments at the beginning of `reservation.ddl` to explain which constraints were not enforced (if any) and which redundancies are still allowed (if any). As the designer you have freedom to choose datatypes for the various attributes.

2. Relation F has attributes $KLMNOPQRS$ and functional dependencies G:

$$G = \{KOQ \rightarrow PS, L \rightarrow KN, KQ \rightarrow RS\}$$

(a) Which FDs in G violate BCNF? List them.

(b) Use the BCNF decomposition method to derive a redundancy-preventing, lossless, decomposition of F into a new schema consisting of relations that are in BCNF. Be sure to project the FDs from G onto the relations in your final schema. There may be more than one correct answer possible, since there are choices possible at steps in the decomposition. List your final relations alphabetically, and order the attributes within each relation alphabetically (this avoids combinatorial explosion of the number of alternatives we have to check).

(c) Does your final schema preserve dependencies? Explain why you answer yes or no.

(d) BCNF guarantees a lossless join. However demonstrate this to a possibly-skeptical observer using the Chase Test.

Show us the steps in your work. This allows us to assign part marks, if needed, and to be sure that you have not taken any unwarranted short cuts.

3. Relation R has attributes $ABCDEFGH$ and functional dependencies S:

$$S = \{ACDE \rightarrow B, B \rightarrow CF, CD \rightarrow AF, BCF \rightarrow AD, ABF \rightarrow H\}$$

(a) Find a minimal basis for S. Your final answer must put the FDs in ascending alphabetical order, and the attributes within the LHS and RHS of each FD into alphabetical order.

(b) Find all the keys for R using your solution for a minimal basis.

(c) Use the 3NF synthesis algorithm to find a lossless, dependency-preserving decomposition of relation R into a new schema consisting of relations that are in 3NF. Your final answer should combine FDs with the same LHS to create a single relation. If your schema has a relation that is a subset of another, keep only the larger relation.

(d) Does your solution allow redundancy? Explain how (with an example), or why not.

Show us the steps in your work. This allows us to assign part marks, if needed, and to be sure that you have not taken any unwarranted short cuts.

submissions

Submit **a3.pdf** and **reservation.ddl** on MarkUs. One submission per group, whether a group is one or two people. You declare a group by submitting an empty, or partial, file, and this should be done **well before** the due date. You may always replace such a file with a better version, until the due date.

Double check that you have submitted the correct version of your file by downloading it from MarkUs.