

csc343 winter 2020
assignment #3: database design
due March 30th, 4 p.m.

goals

This assignment aims to help you learn to:

- informally design a database to represent a real-world scenario
- express your design as an SQL schema
- reason about functional dependencies and use normal forms

design context

You are helping design a database back-end for Wet World's dive booking app. Qualified divers will use the app to book dives with a dive monitor and dive site.

Divers must be at least 16 when a dive occurs and have an open-water certification from one of: NAUI, CMAS, or PADI. One diver, the "lead," books a dive, possibly to include other qualified divers, up to the limit the monitor is willing to take, and taking into account the number the dive site has capacity for at that date and time.

Each monitor has their own price list, according to whether a dive will be morning (9:30 to 11 a.m.), afternoon (12:30 to 2 p.m.) or night (8:30 to 10 p.m.), and whether the dive will be open water, cave diving, or deeper than 30 meters. The price charged by a monitor includes any fees a dive site charges per diver, and a monitor may have booking privileges with one or more dive sites. Monitors also have a maximum group size for each category: open water, cave, or deeper than 30 meters, and they set the maximum to 0 if they do not supervise that category of dive.

All dive sites provide 12 litre aluminum tanks, weight belts and bouyancy compensation vests as part of their fee. Some also provide mask, regulator, fins, and wrist-mounted dive computer, each for additional fees. In addition some sites provide free services from among: video of dives, snacks, hot showers, towel service. Each site lists a maximum number of divers, combining one or more bookings, allowed on site at any daylight hour, and lists smaller maxima for night, cave, or deeper than 30 meter dives. Each site lists one or more dive types they provide from: open water, cave dive, or beyond 30 meters.

A lead diver may book as many dives as they can find monitors and dive sites for, but only one dive at a given date and time. No monitor is allowed to book more than two dives per 24 hour period, in order to keep their blood nitrogen at safe levels. The lead diver provides credit card information, email addresses and dive certification for all divers, and credit card information to secure the dive(s).

All divers may, optionally, rate a site on a scale of 0 (very poor) to 5 (excellent), and lead divers may rate monitors on the same scale.

schema

Define a schema in DDL, stored in file `schema.ddl`. Your goals, in priority order, are:

1. Allow as few redundancies as possible.
2. Allow as few NULL or DEFAULT values as possible.
3. Without using assertions or triggers, enforce as many constraints from the database description as you can.

Wherever goals are in conflict, implement the higher-priority goal. Also you should:

- Define a primary key for each table, and in addition use UNIQUE and define foreign keys where appropriate.

You will be repeatedly re-importing `schema.ddl` as you develop and test it, so it should begin with the following:

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

At the top of `schema.ddl` use a comment to document which constraints could not be enforced, and which you chose not to enforce, and why. We will also provide a `data.txt` file with a plain text description of a database.

some queries

After you have finished `schema.ddl` define a new file `data.sql` to populate an instance of your database with sample data. We will not be autotesting this, so you get to decide on table structure and attribute names as you like. Write queries for the following:

1. For each dive category from open water, cave, or beyond 30 meters, list the number of dive sites that provide that dive type and have at least one monitor with booking privileges with them who will supervise groups for that type of dive.
2. Find monitors whose average rating is higher than that of all dives sites that the monitor uses. Report each of these monitor's average booking fee and email.
3. Find the average fee charged per dive (including extra charges) for dive sites that are more than half full on average, and for those that are half full or less on average. Consider both weekdays and weekends for which there is booking information. Capacity includes all divers, including monitors, at a site at a morning, afternoon, or night dive opportunity.
4. For each dive site report the highest, lowest, and average fee charged per dive.

Queries are stored in files `q1.sql`, `q2.sql`, `q3.sql`, and `q4.sql`.

dependencies, decompositions, normal forms

In file `dependencies.pdf` report the following:

1. Relation R has attributes $IJKLMNOP$ and functional dependencies:

$$S_P = \{M \rightarrow IJL, J \rightarrow LI, JN \rightarrow KM, M \rightarrow J, KLN \rightarrow M, K \rightarrow IJL, IJ \rightarrow K\}$$

- (a) Find a minimal basis for R . List the attributes in each LHS in alphabetical order, each RHS in alphabetical order, and your entire set FDs in alphabetical order.
- (b) Find all keys for R .
- (c) Uses 3NF to find a lossless, dependency-preserving decomposition of R . Be sure to combine FDs with the same LHS into the same into a single relation, and omit relations that are subsets of other relations.
- (d) Does your schema allow redundancy? Explain why, or why not.

Show all your intermediate steps, and explain any steps you may legitimately omit.

2. Relation T contains attributes $CDEFGHIJ$ and functional dependencies:

$$S_T = \{C \rightarrow EH, DEI \rightarrow F, F \rightarrow D, EH \rightarrow CJ, J \rightarrow FGI\}$$

- (a) Which of the FDs in S_T violate BCNF?
- (b) Decompose T into a collection of relations that are each in BCNF, using the BCNF decomposition algorithm. Your decomposition should be lossless and redundancy-preventing. Show your intermediate steps, explaining any steps that you may legitimately omit. Project S_T onto your final set of relations, where every LHS and RHS is in alphabetical order, and each list of projected FDs itself should be in alphabetical order.

submissions

Submit the following files on [MarkUs](#) well before 4:00 on March 27:

- `schema.ddl`
- `data.sql`
- `q1.sql`, `q2.sql`, `q3.sql`, and `q4.sql`
- `dependencies.pdf`

One submission per group, whether a group is one or two people. Declare your 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. We will not have much patience with problems forming groups at the last minute, since this indicates lack of seriousness in group work.

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

marking

We assess `schema.ddl` and `data.sql` according to how well they represent the Wet World domain and the aims listed above. We will mark `q1.sql`, `q2.sql`, `q3.sql`, and `q4.sql` for correctness. In addition, we will mark the following style considerations:

- clear and informative table, view and attribute names
- comments for every view or table describing the rows of the table according to **what** the data means, rather than **how** to obtain it
- 80 character limit per line
- view and table names start with a capital, CamelCase if they use multiple words
- SQL keywords should either consistently lower-case or consistently all-caps
- indentation to improve readability