w4111 / hw3

w4111 HW3 Fall 2018

2 commits | 1 branch | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

ivychen Released HW3

README.md	Released HW3
hw3_template.pdf	Released HW3
hw3_template.tex	Released HW3

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/w4111/hw3.git

Open in Desktop | Download ZIP

README.md

Homework 3

Fall 2018 COMS 4111 HW3

- Assigned: 10/18 Thursday
- Due: 11/15 Thursday, 10:00 AM via. Instabase + Gradescope
- Value: 3.75% of your grade

Assignment

HW3 consists of 2 parts.

Submission:

- Submit Part 1 via Instabase.
- Submit Part 2 via Gradescope.

Part 1: SQL Queries

(16 points total)

In Part 1 you will write SQL queries on Instabase with a preloaded database. You are required to complete and submit via Instabase by the token provided. We have setup the environment for you.

Please find and fork HW3 repo under https://www.instabase.com/ewu/w4111-public/fs/Instabase%20Drive/.

Part 2: Normalization

(14 points total)

In Part 2 you will complete the following written task and submit your assignment via Gradescope.

Q2.1: (2 points) You have a relation `R(A,B,C)` and functional dependencies `B → C, C → A`

- What are **all** the non-trivial functional dependencies in the closure that have only one attribute on the right side? The definition of trivial is a functional dependency where the right hand side is included in the left hand side. These are the

functional dependencies that are true via reflexivity.

- What are all the minimal keys of `R` ? (We do not care about super keys.)

Q2.2: (3 points) You have a relation `S(A, B, C, D)` and functional dependencies `AB->D, BD->C, CD->A, and AD->B`

- What are all the non-trivial functional dependencies in the closure that have only one attribute on the right side?
- What are all the minimal keys of `S` ? (We do not care about super keys)

Now we consider a real application. The `iowa` dataset has the following un-normalized schema:

```
CREATE TABLE iowa (
  address char(128),
  bottle_volume_ml integer,
  category integer,
  category_name char(128),
  city char(128),
  county char(128),
  county_number integer,
  date date,
  im_desc char(128),
  invoice_line_no char(128),
  itemno integer,
  name char(128),
  pack integer,
  sale_bottles integer,
  sale_dollars double precision,
  sale_gallons double precision,
  sale_liters double precision,
  state_bottle_cost double precision,
  state_bottle_retail double precision,
  store integer,
  store_location_address char(128),
  store_location_city char(128),
  store_location_zip char(128),
  vendor_name char(128),
  vendor_no integer,
  zipcode text
);
```

Suppose we have the functional dependencies:

- store → address, name, city, zipcode, store_location, county_number, county, store_location_address, store_location_city, store_location_zip
- vendor_no → vendor_name
- category → category_name
- itemno → category, bottle_volume_ml, im_desc, state_bottle_cost, state_bottle_retail
- date, store, vendor_no, itemno, invoice_line_no → pack, sale_bottles, sale_dollars, sale_gallons, sale_liters

Q2.3: (2 points) What are the keys in `iowa` ?

Q2.4: (3 points) Decompose `iowa` into 3NF (Third Normal Form). Write a few sentences to justify why you chose the tables you did.

Q2.5: (1 point) Is your schema free of redundancies and anomalies? Justify your answer in a few sentences.

Q2.6: (1 point) We want to ensure that an order cannot have a individual bottle price more than 50.00 (`state_bottle_retail`). Can you enforce this using functional dependencies? Justify your answer.

Q2.7: (2 points) In class, we discussed that functional dependencies (and constraints in general) cannot be determined just by looking at data in the database. Let's check whether `itemno` determines `vendor_name` .

- How many distinct `vendor_name` values exist for `itemno` number `'3326'` in the `iowa` dataset? Solve this by running a SQL query on the DB instance from Part 1.

- Argue in one or two sentences whether or not `itemno -> vendor_name` should actually be a functional dependency and why given the design of the database.

X. For Giggles (Optional)

(0 points total)

If you are still interested in this dataset, it turns out that you *can* normalize the store data! The state of iowa has released a dataset of all liquor stores as a dataset available at https://data.iowa.gov/Economy/Iowa-Liquor-Stores/ykb6-ywnd

This dataset provides us information about store attributes so we could factor those out of the iowa dataset.