CS127 Homework #4

Due: October 25th, 2017 2:59 P.M.

Warmup #1

What is the difference between primary key, candidate key and super key in terms of functional dependency?

Warmup #2

Given that the average seek time on a hard disk is 45 ms and rotates at the rate of 12000 rpm with 200 tracks, calculate the time needed to access all the data.

Note: Take average rotational latency into consideration and assume one entire track is read during one disk rotation

Warmup #3

You are designing a database for book sales. Consider the following `Book_Sale` relation:

`Book_Sale(Sale_ID, Bookstore_ID, Bookstore_Name, Location, Author_ID, Author_Name, Book_ID, Book_Title, Date_Published, Price)`

Decompose the above relation into BCNF using following set of functional dependencies, where * represents all columns:

Bookstore_ID → *Bookstore_Name, Location*
Author_ID → *Author_Name*
Book_ID → *Book_Title, Date_Published*
Book_Title → *Price*
Book_Title → *Author_Name*
Sale_ID → *

Warmup #4

Consider the following set of schemas:

```
CREATE TABLE bank (
      branch_ID varchar(3) PRIMARY KEY,
      bname varchar(20),
      no_of_customers int,
      city varchar(10),
      state varchar(20));

CREATE TABLE customer (
      cust_ID varchar(3) PRIMARY KEY,
      cust_name varchar(20),
      acc_type char(1),
      acc_bal int,
      no_of_loans int,
      credit_score int,
      branch_ID varchar(3),
      city varchar(10),
      state varchar(20),
      FOREIGN KEY(branch_ID) REFERENCES bank(branch_ID));
```

Note: `acc_type` is of length one character that is either of the type ‘S’ (Savings) or ‘C’ (Checking)

```
CREATE TABLE loan (
      loan_ID varchar(3) PRIMARY KEY,
      loan_type varchar(15),
      cust_ID varchar(3),
      branch_ID varchar(3),
      amount int,
      FOREIGN KEY(branch_ID) REFERENCES bank(branch_ID),
      FOREIGN KEY(Cust_ID) REFERENCES customer(Cust_ID));
```

Write an assertion that states that a person can apply for a car loan in downtown branch of Providence **only** if he/she has a minimum balance of $1000 in their savings account, has a credit score of 700 or more, and has taken out fewer than 10 loans.

Problem 5 (To Be Graded)

Consider the following sample database for a retailer:

- regions(id, zip_code_served)
- customers(id, status, street, city, zip, region_id, cc_number)
- orders(id, customer_id, total, qty)
- order_items(order_id, item_id, qty)
- items(id, price)

Where:

- *cc_number* is a credit card number, which is used by the customer for payments.
- *region_id* is a region that serves the customer.

Let's assume that the database was set up using where the architect was aware of BCNF and ensured the relationship was in BCNF.

1. List all of the nontrivial functional dependencies that you can infer from these relations.

Now assume that they want to change some of the requirements to give their customers greater flexibility. They decide that they want to allow for multiple credit cards and addresses per customer.

2. Does this requirement change any functional dependencies? Is the schema above still in BCNF? You do not have to prove it, but if it is not then provide a decomposition that is.

Assume the following new requirement: every region serves different zip areas, but a zip area may be served only by a single region.

3. Does this requirement bring any new functional dependencies? Is the schema still in BCNF? You do not have to prove it, but if it is not then provide a decomposition that is.

Problem 6 (Bonus Question)

For the following question, you can research online or talk about content that Stan covers in lecture regarding storage layouts.

Traditional database systems have been broken down into two different families, online transactions processing systems (OLTP) and online analytical processing databases (OLAP). They differ in their workload and storage layout. OLTP systems store data in rows, while OLAP systems store data in column layout. OLTP systems have workloads based on data that is frequently updated, whereas OLAP systems run large-scale analytical queries.

1. For OLAP query engines, come up with three advantages of why storing data in a columnar fashion is better. Be specific in your explanation.
2. For OLTP engines, come up with two reasons why they are stored in a row-based structure.