CIS 2109 HW 8

11/12/2018

Scott O’Hara

5-35)

1. List of users sorted by zip code:

I would suggest using the UserZip field to create a secondary key index since the UserZip would, hopefully, return more than one user.

1. Access to a specific client with the client’s contact person’s name, e-mail address, and phone number:

Creating a join index using the ClientContactID column in the ADVERTISERCLIENT table and the ContactID column of the CONTACT table would be beneficial.

1. List of users within a specific age range sorted by their category and within the category by zip code:

Making a secondary key index using UserCategoryID from the USER relation would be useful, and a secondary key index inside the previous index using UserZip to further organize the data.

1. Access to a specific user based on their e-mail address.

Making a primary key index using the UserEmail column in the USER relation would be useful, since each UserEmail must be unique to exist.

5-42)

1. Identify the foreign key:

* TeamManager in TEAM relation.
* TeamLeague in TEAM relation

1. Types of indexes:

* Since reporting players by team is a typical operation, using the TeamID to create a secondary key index would be useful.
* Since reporting players by team and specialty is a typical operation, using a join index on column TeamID from the TEAM relation along with PlayerID and PlayerSpecialtyCode from the PLAYER relation would save time.
* Creating a secondary key index using Salary from the SPECIALTY relation, since reporting players by salary is a typical operation.
* Since reporting teams and their players by city is typical, using city to create a secondary key index would be helpful.

5-47)



A close up of text on a white background

Description automatically generated

CUSTOMER (Customer ID, CustName, CustAddress)

CARD ACCOUNT (Account ID, ExpDate, Card Type)

DEBIT CARD (Card Type, Bank No.)

CREDIT CARD (Card Type, Cur Bal)

MERCHANT (Merch ID, Merch addr)

CHARGES (Account ID, Merch ID, Charge Date, Charge Time, Amount)

1. Horizontally partitioning the CARD ACCOUNT entity into DEBIT CARD’s and CREDIT CARD’s may help to optimize input-output operations.