

Database Management Systems

Jake Russell

1602835

Table of Contents

Abstract.....	3
Performance Tuning.....	3
Distributed Database Solution	3
Region	3
City	3
Gym	3
Membership Type	3
Gym Membership Price	3
Customer.....	3
MembershipPurchase	4

Abstract

The following report includes ideas for performance tuning using indexes of specific attributes, as well as fragmentation schema suggestions for each table for a distributed database.

Performance Tuning

The author believes the most important/frequently executed queries in the database are:

- Queries that search for customers based on the name
- Queries that search for Purchases based on the purchase date
- Queries that search for Gyms based on the name

The author's logic behind these are as follows:

- Customer's will be one of most checked data within the database specially to update customer's data or to check demographics, the author assumes that the way most users will search the customer table is by using customer's names.
- The author assumes a business would want to regularly check purchases and the dates they were enacted, for uses such as checking how many purchases were done in specific months to see what months are more popular in terms of new memberships.
- Using the same logic as the customer's table, users would like to update gym information and finding the gym will usually be done by searching their name

Distributed Database Solution

Region

The author proposes no fragmentation schema for this table due to it only having one other attribute other than the primary key. No reconstruction necessary.

City

The author proposes horizontal fragmentation for this table with the fragmentation based on region code as cities can't be in two regions. Uses the UNION command to reconstruct.

Gym

The author proposes horizontal fragmentation for this table with the fragmentation based on City Code as a specific gym is unable to be in two cities. Uses the UNION command to reconstruct.

Membership Type

The author proposes horizontal fragmentation for this table with the fragmentation based on cancel any time as a package cannot be both able to cancel at any time and unable to cancel at any time. Uses the UNION command to reconstruct

Gym Membership Price

The author proposes derived fragmentation for this table with the fragmentation based on the parent table Membership type. They would be fragmented based on the cancel any time clause. Uses the UNION command to reconstruct.

Customer

The author proposes vertical fragmentation for this table with the columns of GymID, FirstName, LastName, DateOfBirth, EmailAddress, Passcode, HomeAddress, HomeCityCode, Postcode, ContactPhoneNo, Medical Summary, and JoinDate in one fragmentation. And with the columns of

BankAccountNo and SortCode being in the other. The author believes separating the customer's personal details and bank details would be an effective way to fragment. Uses the JOIN command to reconstruct.

MembershipPurchase

The author proposes vertical fragmentation for this table with the columns of CustomerID, MembershipTypeID, StartDate, EndDate, and Status being in one fragmentation. And with the columns of PurchaseDate, JoiningFeePrice, OneTimeFeePrice, MonthlyFeePrice, MonthlyPaymentDay, LastPaymentDate being in the other. The author believes separating the Membership purchase by the payment details and the customer details would be an effective way to fragment. Uses the JOIN command to reconstruct.