

Bev's Assignment

by Emily Rose Stockdale

FILE	16619_EMILY_ROSE_STOCKDALE_BEVS_ASSIGNMENT_349706_1062008573 .DOCX (170.39K)		
TIME SUBMITTED	11-MAY-2021 11:54PM (UTC+0100)	WORD COUNT	1653
SUBMISSION ID	152784186	CHARACTER COUNT	9029

Database and Development Case Study

Databases and Application Development

ACCA5026

Beverley Holland

Emily Stockdale

1906962

Abstract

This report outlines explains the performance tuning and distributed database solution for a database created for a company called 'Shape Up Gym'. Discussed within this report is data based on secondary index choices and reasoning for why these secondary index's were added to the database. In addition to this fragmentation schema's will be created and examples will be shown based on the database created.

Table of Content

Performance Running

Data queries

Secondary Index's

Distributed Database Solution

Fragmentation Schema

Horizontal fragmentation

Vertical fragmentation

Mixed fragmentation

Derived fragmentation

Global tables

Appendices

Performance Tuning

Data Queries

Database Queries are database transactions used to query a database for information, every database holds information that is important or frequently searched. Typically, a database would allow for a user to search for a client using the ClientID however a user may want to search using client name or other details. To improve performance of data queries secondary indexes can be added, this will then allow a user to 'search by name'.

Shape up gym requested that the most important transactions within their database be investigated in order to improve query performance. The contractor began by identifying the most important or frequently executed queries, by identifying these elements appropriate secondary indexes can be added.

It was identified that frequently searched queries within the Shapeup Gym database are:

- Client Details
- Employee Information
- Client Membership Details
- Class Information

Reasoning for Secondary Index Choice

Adding secondary index's allows users to access specific data rapidly and with ease. The first Secondary index used in this case is client name including both first and last name. This will allow users to access client records with ease rather than using a ClientID which may not be easily accessible, when using a large database with multiple clients each client will have a unique ClientID. Adding a secondary index for both first name and last name will allow users to easily access clients' records using their names. This means that when the finance team are accessing client records there is a clear and accurate display of clients. This will indicate the funding that has been acquired from the gym within the time frame. In addition this will provide the finance team with the relevant information as they will need to know and understand the amount of clients who hold memberships within the organisation however they do not need to know or understand specific client details such as age for example.

Employee information is held within the Shape up gym database, each employee will be qualified to a different standard and level this will determine their job role, salary and duties within the gym. The employee qualification will also determine the type of class that will be run by each individual. By adding a secondary index for employee qualification allows managerial departments and HR to allow rapid access to this information. Creating an index for the employee qualification as stated above determines the employee's salary and job role, this will allow the end user to evaluate the annual cost of each staff member.

Staff and the finance department at Shape Up Gyms will need frequent access to the client membership type. The finance department will need frequent access to complete and process the annual income of each gym. Additionally, staff at Shape Up Gyms will need to access the membership type regularly to ensure that clients do not have any outstanding fees. By adding a secondary index for the membership type it allows employees to outline and identify how many members they have, what membership type they have signed up to and if the gym has reached membership capacity. The secondary index used for this database is the membership name and end date. Using these attributes will allow users to search both when the membership ends/ended and what type of membership is active or inactive. With a growing client base the information held within the database will increase across all locations, ensuring appropriate and rapid access to client membership type is crucial. This will support with gym access and financial records.

Shape Up Gyms runs regular classes across each location with an allocated trainer/employee. All employees at Shape up Gyms across all locations are personal trainers each employee runs a class of their own as well as personal sessions. Enabling access to the employee data within the class table allows staff to identify where staff members may be. This allows for 1-2-1 sessions to be booked appropriately this also identifies where staff members will be at any one time based on their classes. Shape Up Gym may charge per each class, adding a

secondary index will allow the finance team access to each class and each payment received for each individual class held. The employee linked to each class depending on qualification will charge a unique price per class meaning that not all classes will be at the same cost. Adding a secondary index for each class and employee ensures that the end user can access this information with ease. With an increase of locations there will be an influx of data across the database meaning that a simple search will take longer as the data increases, by adding this as a secondary index it allows the end user to access the relevant data in a timely manner.

Secondary Indexes

```
CREATE INDEX client_FirstName_LastName_idx  
ON client (FirstName,LastName);
```

```
CREATE INDEX Employee_Qualifications_idx  
ON Employee(Qualifications);
```

```
CREATE INDEX Membership_MembershipName_Enddate_idx  
ON Membership (MembershipName,Enddate);
```

```
CREATE INDEX Class_EmployeeID_Type_idx  
ON Class (EmployeeID,Type);
```

Distributed Database Solution

Fragmentation is where a database is essentially broken into smaller elements/pieces the fragmented pieces are then distributed. The implementation of fragmentation will ensure that those who access the database will see only data relevant to their needs. To successfully and properly split the data the distributed design relies on business analysts and other professionals this will support in splitting data highlighting frequent transactions and transaction locations. This is linked to quantitative data; quantitative data emphasises edit needs and requirements where views are more suitable than tables. Views are allowed to be developed when fragmentation has been used, this ensures that the relevant users have access to relevant data when it is needed. When reference tables are updated, the views associated with that table will be updated as views are linked to tables.

Fragmentation Schema

Client (ClientID, FirstName, LastName, Email, Address, Postcode, DOB, PaymentID, MembershipCode)

Membership (MembershipID, MembershipName, Startdate, Enddate, Location)

Location (LocationID, LocationName, Address, Postcode)

Payment(PaymentID, BankName, SortCode,AccountNumber)

Employee(EmployeeID, FirstName, LastName, Email, Address, Postcode, DOB, StartDate, EndDate, Qualifications, JobRole)

Class (ClassID, Type, Time, EmployeeID, Membershipcode)

Class Register (RegisterID, MembershipCode, ClientID, EmployeeID, Time)

Horizontal Fragmentation

The below table has been fragmented horizontally. The horizontal fragmentation of data separates those with similarities for example, all clients linked to the same membership. For example those linked to annual memberships would be fragmented into a table similarly with monthly memberships, daily, and weekly.

The below table is an example of a fragmented horizontal table linked to a monthly membership.

MembershipID	MembershipName	Price
1	MonthlyMembership	18
5	Monthly	18

The tables below represent all other membership types. As my shape gyms expand and more locations become available the amount of data in each table will increase.

MembershipID	MembershipName	Price
2	Annual	30

MembershipID	MembershipName	Price
3	Day Pass	45

MembershipID	MembershipName	Price
4	Week Pass	16

Using UNION each table above table can be brought back together.

Vertical Fragmentation

The below tables are fragmented vertically, vertical fragmentation assigns subsets of attributes to different tables.

Membership Prices

MembershipID	MembershipName	Price
1	MonthlyMembership	18
2	Annual	30
3	Day Pass	45
4	Week Pass	16
5	Monthly	18

Membership Locations

MembershipID	MembershipName	Location
1	MonthlyMembership	1
2	Annual	2
3	Day Pass	3
4	Week Pass	4
5	Monthly	5

Mixed Fragmentation

The table below shows a mixed fragmentation, the tables below are a mix of both horizontal and vertical fragmentation, this is fragmenting a dataset which has already been fragmented. The below tables would support in identifying how many members, on what type of membership are accessing classes.

ClassID	Type	Memberships	MembershipName	Price
1	yoga	NULL	MonthlyMembership	18
2	Body Pump	1	Annual	30
3	Swim	4	Day Pass	45
4	Pilates	3	Week Pass	16
5	Boxersize	5	Monthly	18

The below table using mixed fragmentation would support in identifying which membership is most popular or purchased depending on location.

LocationID	LocationName	MembershipName	Price
1	Shape Up Gym Cardiff	MonthlyMembership	18
2	Shape Up Gym Swansea	Annual	30
3	Shape Up Gym London	Day Pass	45
4	Shape Up Gym Wrexham	Week Pass	16
5	Shape Up Gym Birmingham	Monthly	18

Derived Fragmentation

Derived fragmentation is a horizontal fragmented table the information is based on data sources related and joined with other tables [1]. Tables are linked together through a primary key and a foreign key and must establish an 'owner-member' relation. Derived fragmentation is used when a horizontal fragmentation does not rely on or is not dependant on another owner table.

The Shape Up Gym database management system has a primary fragment based on memberships and child fragment of price.

M_i : Memberships $A_i \quad 1 \leq i \leq 5$

P_i : Price $\triangleright P_i \quad 1 \leq i \leq 5$

Reconstruction: $P_1 \cup P_2 \cup P_3 \cup P_4 \cup P_5 \cup P_6 \cup P_7 \cup P_8$.

Reconstruction: $O_1 \cup O_2 \cup O_3 \cup O_4 \cup O_5 \cup O_6 \cup O_7 \cup O_8$.

Global relations from the fragments

schema is the data used to develop fragments; these fragments are then distributed across the information architecture [1]. The global schema is essentially the central repository for the data this is then broken up into fragments providing the end user with the data they require [2] ensuring data availability. This also reduces costs.

The Shape Up Gym database is a type of database called 'a homogeneous database management system' as it's accessed through a single interface. Homogeneous databases are created independently for unique purposes [4]. Each element is linked through tables, indexes and schema's which ensures that user requests can be processed successfully [5].

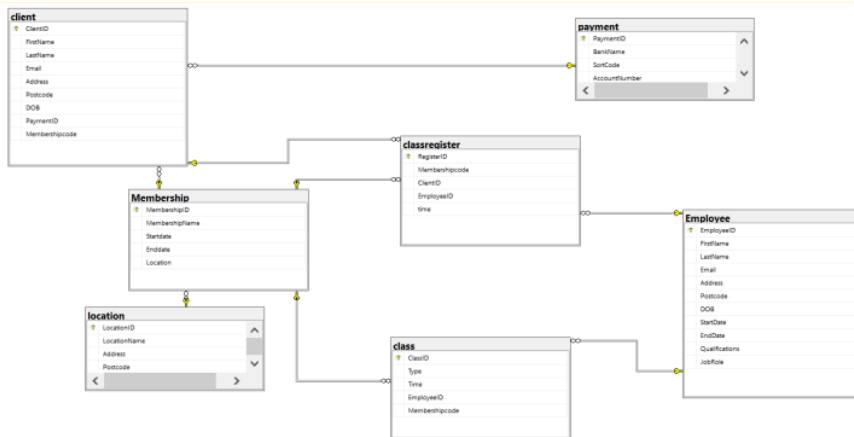
Conclusion

To conclude, the above report has broken down the database performing tuning, in doing so this report identifies which of the queries within the created database will be most frequently ran. By identifying the frequently searched queries it has allowed the author to identify appropriate secondary indexes while providing reasoning as to why each index has been selected and how this will improve the performance of the database and the overall user experience. The author has included evidence of the indexes chosen and used within the database. Following this the distributed database has then been discussed, breaking down each schema in depth while using table evidence to support each element. The fragments discussed have also been supported with evidence created using tables. Overall this report has outlined the design and usability of the proposed database.

Appendices

Appendix 1

ERD



Bev's Assignment

GRADEMARK REPORT

FINAL GRADE

61 / 100

GENERAL COMMENTS

Instructor

You demonstrate a sound understanding of SQL and you have followed your design well and thought about the constraint and relationships. However, if you had tested more I feel you may have changed the relationship directions. For example one payment has many clients? One member has many clients – these table should act as look up table for membership type. Also maybe link the employee to the location.

A suitable layout to your report but try to use the TOC in word to add page numbers to your section headings. Also try to number you section headings.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9
