

Information System Performance and Design Report

by Brandon Philip Gorman

FILE	19441_BRANDON_PHILIP_GORMAN_INFORMATION_SYSTEM_PERFORMAN CE_AND_DESIGN_REPORT_349706_311330129.DOCX (1.54M)		
TIME SUBMITTED	03-MAY-2021 11:26PM (UTC+0100)	WORD COUNT	1960
SUBMISSION ID	151491827	CHARACTER COUNT	10739

Information System Performance and Design Report.

Brandon Philip Gorman AMBCS (1909188)
University of Wales Trinity Saint David
Beverley Holland | Databases and Application Development (ACCA5026)

Table of Contents

Abstract..... 3

Database Performance Tuning 3

Distributed Database Solution..... 5

References..... 10

Appendix 10

Abstract

The following report documents possible solutions to Shape Up Gyms (the client) performance tuning problems as well as provide a suitable distributed database design. The distributed design will highlight examples using the gym management system developed for the client.

Database Performance Tuning

The client has highlighted the requirement to filter through many records to assist with the company's administrative business as usual activities. As the client highlighted in the task brief the membership system will hold vital core demographics on the client's customers. The demographics held on the membership system can assist in the development of targeted advertising, specialist activities in certain areas as well as the ability to analyse the number of memberships sold by region.

Please find below the following index solutions to assist in the development of data retrieval.

Firstly, core demographics and membership validity. As mentioned in the brief the new system was proposed to also assist the finance department to make certain based decisions based on evidence. Having the ability to search through many records in a quicker amount of time brings value to an organisation's productivity. The finance department will need to ensure the clients financial affairs are in order and meets their financial criteria. Having the ability to locate active members with valid memberships will assist the finance department in managing their local and national finance models which will influence future business operations.

Core Demographics:

- First Name
- Last Name
- Membership Valid

First name and last name assists with basic identification without the requirements for a unique identifier. This basic information can give the finance department a standard overview of the individuals who attend the gym. This basic information if required can then be manipulated to produce registers for each gym.

Membership validity. The membership validity gives the client the ability to manage space and allocate additional resources if required. This field will assist all teams in the organisation in managing gym capacity over an annual period. This indexed data if required can then be grouped together to calculate potential overhead in the coming years using various forms of statistical analysis.

Secondly, membership options. The clients brief emphasised the ability of each gym to have local autonomy over membership rates and membership types. Due to the potential

uniqueness of the membership types the probability of the memberships table having a high number of records increases. Indexing the memberships will allow for quick access to each membership type. This will also assist with data reports, requirements gathering as well as trend analysis. – This index will also have a positive impact on the finance department as they will also be able to report sales by membership a lot quicker and propose financial changes to ensure competitiveness on the open market.

Memberships:

- MembershipsID
- LocationID
- StartFee
- Cost

Memberships unique ID. This will ensure if any reports or potential views that are later developed are bringing back unique values and the possibility of duplication is reduced. The unique ID field indexed will also assist if the membership system is interfaced with any related systems, for example, the sessions system that would manage gym training sessions. The memberships ID will relate to the customer and those details will be sent via a stored procedure or SQL job to the relevant table on the session system.

Location ID. Business is never static. The need to ensure quick and reliable access to business data is key to the client's financial security. By indexing the Location ID the finance team and other teams will have the ability to produce estates registers along with analysing membership statistics via location.

Start fee and cost. The start fee and costs can differ across each site. Having the ability to quickly locate a start fee and cost can assist with potential bulk updates. If the client decides to amend costings this can be done via a bulk update and the database index would react quicker and proactively as the index logic understands each row better as opposed to a column within an entity that hasn't been indexed.

Next, membership records. The membership records table records all details in relation to a customer's memberships and activity status. By indexing elements of the memberships table will assist in time management. As well as holding core details in relation to a customer membership, this table is also used as an audit.

Membership Records:

- MembershiprecordID

Membership Records ID. As this is a unique ID indexing this will ensure results return quicker. This will also alleviate pressure from the database. By indexing the MembershiprecordID obtaining additional data from the table will be easier for the user.

Finally, views. Due to the use of indexing, the use of views will be an appropriate step forward once additional business requirements have been proposed. For example, an invoice view showing the customers memberships along with cost and any fees etc.

Distributed Database Solution

The client has requested a proposed distributed database design that will have the ability to share key data across the client's information architecture. This distributed database solution is to work alongside the database schema implemented.

Fragmentation Schema

Fragmentation in its simplest form is where a table can be broken up (fragmented) into smaller pieces and distributed across a services information architecture. In addition to fragmentation the data or sub-fragment is then allocated via a particular identifier in the client's case the Location. By implementing fragmentation and ensuring the correct allocation the systems stakeholders will view data that is relevant to them.

Firstly, fragmentation design. Database fragmentation is a user requirement-based solution. This means without clear requirements to develop; such a solution would be difficult to produce. The client highlighted in the brief that the finance team will be based in Swansea. This suggests data must be fragmented and allocated to Swansea, using the unique IDs implemented in the membership management system.

Secondly, transactions relating to quantitative data. Distributed design relies on business analysts along with other business professionals to highlight common transactions and transaction locations. – This will assist in splitting the data appropriately.

Next, Qualitative data. This focuses more the database entities, such as columns and rows as well as additional clause criteria such as WHERE and IN. Qualitative data also highlights add / edit requirements, emphasising where views are more suitable as opposed to tables.

To continue, fragmentation allows views to be developed to ensure users have access to the data they need when they need it. Views are based off tables, meaning once a reference table has been updated the associated views will also be updated.

Fifthly, fragmentation allows for efficient data storage, meaning only data required will be stored locally. This will reduce costs on infrastructure.

Finally, fragments allow transactions to flow without a direct reliance on its reference data.

Horizontal Fragments

The following tables are to be broken up horizontally. The following table fragments also follows the fragment correctness rules as each row in the initial relation appear in one of the proposed fragments. This data can be reconstructed with the use of a union.

Location, Memberships, Gender and Pay Type.

A written example of this has been included below:

- Location table.
 - o L1 type = 'North Wales'
 - o L2 type = 'East Wales'
 - o L3 type = 'South Wales'
 - o L4 type = 'West Wales'

By fragmenting the table by region, hypothetically, those stakeholders will have the ability to focus certain tasks concerning a particular region in Wales.

Please find further table fragments below. The below tables as stated above can be brought back together with the use of a UNION.

locationID	regionID	name	Addr1	Addr2	Addr3	Postcode
L00001	R00004	ShapeUpGyms: UWTSD	1	Kings Road	Swansea	SA1 8PH
locationID	regionID	name	Addr1	Addr2	Addr3	Postcode
L00002	R00001	ShapeUpGyms: Rhyl	Rhuddlan Road	Bodelwyddan	Rhyl	LL18 5UJ
locationID	regionID	name	Addr1	Addr2	Addr3	Postcode
L00003	R00003	ShapeUpGyms: Cowbridge	Cowbridge Road East	Cardiff		CF11 9XB
locationID	regionID	name	Addr1	Addr2	Addr3	Postcode
L00004	R00002	ShapeUpGyms: Brecon	Cerrigcochion Roadt	Brecon		LD3 7NS

membershipsID	name	description	locationID	validfor	signupfee	cost
M00001	Annual Plan	Full access 12	L00001	365	15.00	155.00
M00005	Monthly Plan	Full access 1 m	L00001	31	5.00	65.00

membershipsID	name	description	locationID	validfor	signupfee	cost
M00006	Monthly Plan	Full access 1 m	L00002	31	0.00	55.00
M00002	Annual Plan	Full access 12	L00002	365	0.00	190.00

membershipsID	name	description	locationID	validfor	signupfee	cost
M00003	Annual Plan	Full access 12	L00003	365	1.99	99.00
M00007	Monthly Plan	Full access 1 m	L00003	31	1.00	52.00
M00010	Pay As You Go	Full access 1 d	L00003	1	0.00	15.00

membershipsID	name	description	locationID	validfor	signupfee	cost
M00008	Monthly Plan	Full access 1 m	L00004	31	0.00	40.00
M00009	Pay As You Go	Full access 1 d	L00004	1	0.00	10.00
M00004	Annual Plan	Full access 12	L00004	365	10.00	87.00

genderID	name
G00001	Male

paytypeID	name
P00001	Debit/Credit Card

genderID	name
G00002	Female

paytypeID	name
P00002	PayPal

genderID	name
G00003	Transgender

paytypeID	name
P00003	Cheque

genderID	name
G00004	Other

Vertical Fragments

The following tables are to be broken up vertically. The following table fragments also follows the fragment correctness rules as each row in the initial relation appear in one of the proposed fragments.

Memberships and Customers.

A written example of this has been included below:

- Memberships table.
 - o M1: membershipsID, name StartFee, Cost (Finance Fragment).
 - o M2: membershipsID, Name , locationID , validfor (Memberships Fragment)

Finance Fragment

membershipsID	name	signupfee	cost
M00001	Annual Plan	15.00	155.00
M00005	Monthly Plan	5.00	65.00
M00006	Monthly Plan	0.00	55.00
M00002	Annual Plan	0.00	190.00
M00003	Annual Plan	1.99	99.00
M00007	Monthly Plan	1.00	52.00
M00010	Pay As You Go	0.00	15.00
M00008	Monthly Plan	0.00	40.00
M00009	Pay As You Go	0.00	10.00
M00004	Annual Plan	10.00	87.00

Memberships Fragment

membershipsID	name	description	locationID	validfor
M00001	Annual Plan	Full access 12 months.	L00001	365
M00005	Monthly Plan	Full access 1 month.	L00001	31
M00006	Monthly Plan	Full access 1 month.	L00002	31
M00002	Annual Plan	Full access 12 months..	L00002	365
M00003	Annual Plan	Full access 12 months.	L00003	365
M00007	Monthly Plan	Full access 1 month.	L00003	31
M00010	Pay As You Go	Full access 1 day.	L00003	1
M00008	Monthly Plan	Full access 1 month.	L00004	31
M00009	Pay As You Go	Full access 1 day.	L00004	1
M00004	Annual Plan	Full access 12 months.	L00004	365

This fragment follows the fragment correctness rules as each row in the initial relation appear in one of the two fragments. This data can be reconstructed with the use of an inner join. The fragments are disjointed, however the membershipsID is the primary key and is an exception.

Customer Management Fragment

customerID	firstname	lastname	genderID	joined	membershipvalidto
C00001	John	Doe	G00001	2020-01-01	2021-06-02
C00002	Jonannah	Johesph	G00002	2021-01-01	2021-06-02
C00003	Stephen	Andrews	G00003	2020-06-01	2021-05-29
C00004	Amanda	Bloggs	G00002	2020-06-01	2022-01-29

Finance Fragment

customerID	genderID	paydetails	membershipvalidto
C00001	G00001	5	2021-06-02
C00002	G00002	6	2021-06-02
C00003	G00003	8	2021-05-29
C00004	G00002	7	2022-01-29

Mixed Fragments

Due to the database schema and its breakdown into types, it is a lot easier to identify key areas for mixed fragmentation, fragmenting an already fragmented data set.

For example, the Vertical Customer Management Fragment can also be broken up horizontally by gender.

customerID	firstname	lastname	genderID	joined	membershipvalidto
C00001	John	Doe	G00001	2020-01-01	2021-06-02

customerID	firstname	lastname	genderID	joined	membershipvalidto
C00002	Jonannah	Johesph	G00002	2021-01-01	2021-06-02
C00004	Amanda	Bloggs	G00002	2020-06-01	2022-01-29

customerID	firstname	lastname	genderID	joined	membershipvalidto
C00003	Stephen	Andrews	G00003	2020-06-01	2021-05-29

This would be useful to identify what memberships a preferred by different genders. The same logical can be used across the system.

Global Relation from the Fragments

A global schema is the original source or reference data used to develop fragmentation schemas. It is then the fragmentation schemas which are then allocated across the information architecture.

It's the information architecture which enables its users to execute commands at several sites locally. In the case of the membership management system, the structure in place is that of a homogeneous Database Management System [1]. One central data repository but broken up into fragments to meet the stakeholder requirements. This form of data management reduces costs and removes data redundancy as well as ensuring data availability at all times.

From a user perspective the local schema they would view would be their local data recourse (LDR), however, in reality the local schema is just a fragment that's a relation to their LDR.

Derived Fragmentation

Derived fragmentation is based on a horizontal fragmentation as a primary data source which has been related and joined to another table [2]. In the case of the Gym Membership System the Customer Management fragment which has a primary fragment based on its membership also has a child fragment of gender. This can be identified by its reliance on the

horizontal fragment which provides scope and context. Removing this scope will result in data redundancy on the global schema, which will affect the access schema as well as other LDRs across the clients' network.

Conclusion

To conclude, the membership management system now has the ability to perform certain query searches with quicker results. This is due to newly introduced indexes. The newly developed indexes will also assist the view and triggers related to the relevant tables to perform faster updates improving useability.

In addition to the new indexes a new view for the finance department based on horizontal fragmentation has been produced ensure the data they require will be available locally. A new fragmentation schema has been proposed for the clients LDRs ensuring the correct data is available at the correct time. The view is synchronous and will change when users make changes to the database, this can be users using fragmented data views or on the global schema.

References

- [1] Single, "What are the Homogeneous and Heterogeneous Distributed DBMS.," Ques10, 1 March 2017. [Online]. Available: <https://www.ques10.com/p/17615/what-are-the-homogeneous-and-heterogeneous-distr-1/>. [Accessed 3 May 2021].
- [2] L. Wiese, "Clustering-based fragmentation and data replication for flexible query answering in distributed databases," *Journal of Cloud Computing*, vol. 3, no. 18, 2014.

Appendix

Appendix One

Data Normalisation Diagram showing database proposal in 3NF.


```
/*
----- STEP FOUR: CREATE DATABASE TRIGGERS. -----
*/

CREATE TRIGGER TRG_UpdateCustomerValid
ON membershiprecord
AFTER INSERT
AS
DECLARE @newdate as date;
DECLARE @customerID as CHAR(6);
BEGIN
    SET NOCOUNT ON;

    set @newdate = (select ActiveTo from inserted);
    set @customerID = (select customerID from inserted);
    UPDATE customer set membershipvalidto = (@newdate)
    WHERE customerID = @customerID;
END
GO
```

10 % 4

Messages

Command completed successfully.

Completion time: 2021-05-03T13:43:16.7781730+00:00

Appendix Four

View developed to assist the finance department based at Swansea.

```
/*
----- STEP FIVE: CREATE DATABASE VIEW. -----
*/

CREATE VIEW Vw_Invoice AS
SELECT DISTINCT InvoiceDetailsID , InvoiceID , a.membershipID , cost , signupfee , b.name , description , ActiveFrom , ActiveTo
FROM Invoicedetails a
CASE
    WHEN Active = 1 THEN 'Active Plan'
    ELSE 'In-Active Plan'
END AS ActiveStatus , e.name [Payment Type] , d.BankName , d.AccountNumber , d.SortCode , d.PayEmail
FROM Invoicedetails a
INNER JOIN memberships b on b.membershipID = a.membershipID
INNER JOIN membershiprecord c on c.membershipID = a.membershipID
INNER JOIN paydetails d on d.customerID = c.customerID
INNER JOIN paytype e on e.paytypeID = d.paytypeID;

/*
----- STEP SIX: CREATE DATABASE INDEX. -----
*/
```

10 % 4

Messages

Command completed successfully.

Completion time: 2021-05-03T13:43:16.7781730+00:00

Appendix Five

Testing check constraints with a transaction.

```

--INCORRECT Postcode length
/*
ERROR MESSAGE: Msg 547, Level 16, State 0, Line 263
The INSERT statement conflicted with the CHECK constraint "CHK_location_postcode". The conflict occurred in database "shapeugyms", table "dbo.location", column "shapeugyms".
*/
INSERT INTO [dbo].[location]
([locationID],[regionID],[name],[Addr1],[Addr2],[Addr3],[Postcode])
VALUES
('L00005','R000002','ShapupGyms: Brecon','Cerrigcochion Roadt','Brecon',' ','LD3');

--Attempt to change membership details.
/*
ERROR MESSAGE: Msg 547, Level 16, State 0, Line 276
The UPDATE statement conflicted with the CHECK constraint "CHK_membershiprecord_ActiveToDate". The conflict occurred in database "shapeugyms", table "dbo.membershiprecord".
*/
UPDATE [dbo].[membershiprecord]
SET
    ActiveFrom = '2021-05-02'
WHERE membershiprecordID = 'M000001';
|
ROLLBACK

```

Messages

Msg 547, Level 16, State 0, Line 261
The INSERT statement conflicted with the CHECK constraint "CHK_customer_customerID". The conflict occurred in database "gym", table "dbo.customer", column 'customerID'.
The statement has been terminated.

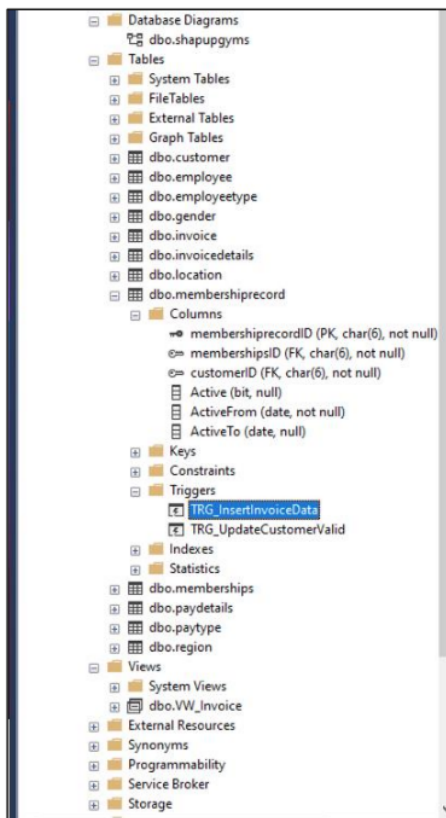
Msg 547, Level 16, State 0, Line 271
The INSERT statement conflicted with the CHECK constraint "CHK_location_postcode". The conflict occurred in database "gym", table "dbo.location", column 'Postcode'.
The statement has been terminated.

Msg 547, Level 16, State 0, Line 276
The UPDATE statement conflicted with the CHECK constraint "CHK_membershiprecord_ActiveToDate". The conflict occurred in database "gym", table "dbo.membershiprecord".
The statement has been terminated.

Completion time: 2021-08-03T14:16:13.6466776+00:00

Appendix Six

Overview of the database via the object explorer.



Information System Performance and Design Report

GRADEMARK REPORT

FINAL GRADE

80/100

GENERAL COMMENTS

Instructor

An excellent report, you demonstrate and in-depth knowledge of database applications through the use of SQL. Good schema design but make sure you do not use too many backward linking references e.g. customer to payment details. Very good overall structure and testing demonstrated to a high standard good demonstration of the use of triggers.

In your report layout try to number your section heading and figures.

Good analysis of the distributed database requirements demonstrated and discussed in detail in your report. Well done.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

