Assignment 5: Tour Operator Agency Database

Marcos A. Ortiz Albino

CIS 515 – Strategic Planning for Database Systems

Dr. Osama Murad

Strayer University

May 15, 2016

Assignment 5: Tour Operator Agency Database

**Introduction**

A multinational tour operator agency has gained new business growth in the North American market through the use of social media. Its operation has expanded by 50% within six months, and the agency requires an enhanced data management strategy to sustain their business operations. Their existing data repository for its reservation processing system is limited in business intelligence and reporting functionalities. The tour operator seeks a database management specialist to assist them in leveraging their data sources to enable them to forecast and project tour sales appropriately. The company database has the following characteristics:

* A table that stores all the salespersons
* A table that stores tour customer data and tours sold.
* A tour table that is used as a tour rate sheet which holds the tours offered and the cost per person.

**Data Model Design**

There are many issues with the agency’s database that requires attention. Because of these findings, we need to normalize the company's database to comply with standards and not to have redundant data among other things that normalization covers. Database normalization, or data normalization, is a technique to organize the contents of the tables for transactional databases and data warehouses. Normalization is part of successful database design; without normalization, database systems can be inaccurate, slow, and inefficient, and they might not produce the data you expect. (Poolet, 1999). The removal of duplicate data should be the priority. To perform this normalization needs to be performed by following the data integrity rules. Removing data redundancy helps to prevent deletion, update errors and insertion, as the data is only available in one attribute of one entity in the database. According to the findings, Figure 1 shows a diagram that represents attributes that are necessary for the agency.



*Figure 1. Attributes important to Tour Operator Agency with dependencies.*

**Tour Operator’s Data Structure Using a Normalization Process**

“First normal form (1NF) sets the fundamental rules for database normalization and relates to a single table within a relational database system. Normalization follows three basic steps, each building on the last. The first of these is the first normal form” (Jansen, 2014). 1NF contains following characteristics:

* Each column in the table should not be repeated.
* Each table must be identified with a unique column or columns called the primary key.
* No rows or columns should be duplicated, and intersections contain a null value.

To achieve 1NF, the tables need to be split into three different tables. The tables are the following: Sales Associate, Customer, and Tours. Primary keys were identified for each table as Emp\_ID for Sales Associate, Cust\_ID for Customers and Tour\_ID for Tours table.

“Second normal form (2NF) is the second step in normalizing a database. 2NF builds on the first normal form (1NF)” (Jancen, 2014). The characteristics for 2NF are as follows:

* No Data Redundancy
* Logical Data Dependencies
* All non-prime attributes are functionally dependent on the whole of every candidate key.

The tables gets normalized and set as follows:

* Table: SALES ASSOCIATE:

* Table: CUSTOMERS:

* Table: TOURS:


**Entity and Attributes Naming Convention**

 The main goal of adopting a naming convention for database objects is so that you and others can easily identify the type and purpose of all objects contained in the database (Lively & Sarsany, 2008). Some of the standards that need to be followed are:

* All entity names in Oracle database should be singular and may have spaces. Spaces are replaced with '\_' in Oracle table creation.
* All table names should be plural.
* All entities and tables should be defined a unique.
* Attributes and columns should not be prefixed with a table alias.
* Primary Keys are named after the table or its alias with the suffix of '\_PK'.
* Unique Keys are named after the table or its alias with the suffix of '\_UK'.
* Foreign Keys are suffixed of '\_FK':
* Indexes are created implicitly to support PKs and UKs.
* Trigger names should be made up of the table name, an acronym representing the triggering action and the suffix "\_TRG".

In Tour Operator’s database, the entities and attributes are designed as follow:

* SALES\_ASSOCIATE (**Emp\_ID**, Emp\_FN, Emp\_LN, Cust\_ID, TotalTourSold)
* CUSTOMERS (**Cust\_ID**, Cust\_FN, Cust\_LN, Address, Tour\_ID, NumberOfPersonInTour, TotalAmountPaid, Sold\_date)
* TOURS (**Tour\_ID,** Tour\_Cost\_Per\_Person)

**Entity Relationship Model (ERM)**

****

*Figure 2. Entity Relationship Model (ERM) visually representing relationships between tables.*

**Querying the Database and Triggers**

The following query that can be used on a report to determine how many days the customer’s invoice will require payment if total amount due is within 45 days. A copy of the working code is also provided in Figure 3.

SELECT CUST\_ID, B.TOUR\_ID, SUM(TOTAL\_AMOUNT\_PAID) FROM CUSTOMERS A,TOURS B WHERE A.TOUR\_ID=B.TOUR\_ID AND TRUNC(SYSDATE)-TRUN C(SOLD\_DATE)<=45 GROUP BY CUST\_ID,B.TOUR\_ID;



*Figure 3. Working Code of Previous Query*

 The following trigger will increase the field that holds the total number of tours sold per salesperson by an increment of one (1):

CREATE OR REPLACE TRIGGER TRG\_TOUR AFTER INSERT OR UPDATE OF TOTAL\_TOUR\_SOLD ON SALES\_ASSOCIATE FOR EACH ROW BEGIN UPDATE SALES\_ASSOCIATE SET TOTAL\_TOUR\_SOLD = TOTAL\_TOUR\_SOLD + 1 WHERE EMP\_ID = NEW\_EMP\_ID END;

The following query will produce results that show the quantity of customers each salesperson has sold tours to. A copy of the working code is also included in Figure 4.

SELECT EMP\_ID, EMP\_FIRSTNAME, COUNT (B.CUST\_ID) FROM SALES\_ASSOCIATE A, CUSTOMERS B WHERE B.CUST\_ID=B.CUST\_ID GROUP BY EMP\_ID, EMP\_FIRSTNAME;



*Figure 4. Working Code of Previous Query*

**Database Stored Procedures**

 According to the textbook, stored procedures are a named collection of procedural and SQL statements. They have stored logically on a server in the form of SQL code or another DBMS-specific procedural language. Stored procedures allow the application to encapsulate SQL statements that represent a business transaction and execute them as a single transaction, thus increasing performance and reducing network traffic (Coronel, Morris, & Rob, 2013). In coding terms, they are functions that stored command line SQL statements that are repetitively used by queries to optimize performance. Another benefit of using this stored procedures are the fact that they can be reused or interlink with more advanced queries to produce different functions or selections from the database. The idea behind it is to identify the tasks that will regularly be performed and create or write them in a way that it can be implemented as a stored procedure. By using these procedures, the database will consume fewer resources and provide the output immediately as it is requested.

**References**

Coronel, C., Morris, S., & Rob, P. (2013). *Database Systems Design, Implementation and Management* (10th ed.). Boston, MA, USA: Course Technology, Cengage Learning.

Jancen, C. (2014). *Second Normal Form (2NF)*. Retrieved from TechoPedia: https://www.techopedia.com/definition/21980/second-normal-form-2nf

Jansen, C. (2014). *Database - First Normal Form (1NF)*. Retrieved from TutorialsPoint: http://www.tutorialspoint.com/sql/first-normal-form.htm

Lively, S., & Sarsany, M. (2008). *SQL Server Standards v 1.5*. Retrieved from ISBE: http://www.isbe.net/ILDS/pdf/SQL\_server\_standards.pdf

Poolet, M. A. (1999, February 28). *SQL by Design: Why You Need Database Normalization*. Retrieved from SQL Mag: http://sqlmag.com/database-performance-tuning/sql-design-why-you-need-database-normalization