DBAS PROJECT 2 – DATA MIGRATION

PROJECT INTRODUCTION

In this database project, you will take an existing, populated database, re-design parts of it according to the provided requirements, implement those design changes, and then migrate all existing data from the original database to the newly-enhanced database. Part of the data migration will entail some data manipulation. Finally, you will create some test scripts to verify the accuracy of your data migration.

This project is intended to evaluate you on all the core DBAS course concepts, so will require analysis and design of an ERD, implementation of both Data Definition Language and Data Manipulation Language scripts, as well as using queries in a variety of ways to help transform, populate and test the validity of your work.

PROJECT EVALUATION & DEADLINE

Due Date: **Monday, April 18, 4:30 PM**

This project is worth 30% of your overall course mark. There are four deliverables:

1. An ERD design containing the required changes/enhancements to the Northwind database.
2. A DDL script to create all database objects for the enhanced Northwind database.
3. A DML script to migrate and/or populate all tables with the existing data from the original Northwind database.
4. A Test Plan script, used to verify the accuracy of your database creation and all data migration.

Values below show the breakdown of project deliverables that will combine to make the overall mark.

Deliverable	Value
ERD Document	**20%**
DDL script	**20%**
DML script	**40%**
Test Plan script	**20%**

REQUIRED MATERIALS

To complete this project, you will need a copy of the Northwind ERD and database creation scripts, which are available on D2L. All software requirements (Oracle 11g Express, SQL Developer, Visio, etc.) are available on the DBAS virtual machine, which is available on the S: drive or on the NetShare drive on Net172.

SUBMISSION

File deliverables will be uploaded to the appropriate Project 2 dropboxes on D2L. If an individual deliverable includes multiple files, please put all files into a single ZIP file.

Please ensure all uploaded file names includes your **name**, the **deliverable name** and your **class section number**.

Ex: **DoeJohn_Project2_ERD_702.zip**

PROJECT SCENARIO & REQUIREMENTS

A fictional company named *NorthWind Traders* has hired you to make some enhancements to their existing order-tracking database. You will be required to analyze their existing database, design a solution to meet their requirements, create the new database according to the new design, and then migrate all data from their existing database into the new one. They would also like some basic QA scripts to verify the success of the data migration.

ENHANCEMENT #1 – BETTER INVENTORY TRACKING

The existing database has three fields in the Products table that are currently used to track inventory stock (UnitsInStock, UnitsOnOrder, ReorderLevel). However, the company owns three different warehouses, and currently has no way of tracking stock levels for each product in each warehouse. Inventory for any product could be stored at any of the three warehouse locations. Each warehouse has a name and a street address that should be stored in the new database. The warehouses are commonly known by their location within the city: The Dockside warehouse, the Airport warehouse and the Central warehouse. The new scheme should be able to record the same basic data as the existing database (UnitsInStock, UnitsOnOrder, ReorderLevel), but also reference the warehouse in which the products are currently being stored in, or re-ordered from.

ENHANCEMENT #2 – RESOLUTION OF CUSTOMER IDENTIFICATION ISSUES

In the existing database, unique customers are currently identified by a three- or five-character alpha code. In theory, these codes are never supposed to be changed, so a particular company can always be identified by its code. However, in practice, over time some codes have been changed to reflect updates to company names. These changes only occur on occasion, but when they do, the database either disallows the updated codes, or seems to lose track of some of the orders that should belong to those customers. It has caused a lot of confusion and errors in data entry. The company would like to keep the alpha codes, but not have the system so dependent on them.

ENHANCEMENT #3 – CENTRALIZATION OF COUNTRY DATA

The accounting team often uses country-related data from the database when generating reports based on financial, supply chain or human resources data tracked in the system. The current database records this country data in several tables (Customers, Orders, Suppliers, & Employees). However, frequent report errors are being encountered due to problems with the country data. For example, if a country name is misspelled, it shows up as a separate report item, even though it is obviously the same country with a typo in the name.

The accounting team has proposed a centralized list of countries in the database. Each of the tables that currently record Country data should no longer allow user-entered country names, but should instead "look up" the desired country from the new central Country table. During data entry via the website, the country will be selected from a dropdown list instead of being typed manually. The software dev team will implement that change on the website, but the new database design should support a centralized Country table.

ENHANCEMENT #4 – FULL NAMES WITH TITLE

The HR department has requested a database change to help integration with the new HR software program. The new software does not work well when importing separate values for first/last names and the title of courtesy. They propose adding a new field that contains a combined value in the following format: <Courtesy Title> <First Name> <Last Name>. Examples: Mr. John Doe or Dr. Jane Smith. The proposed field will not replace the usage of the existing separate fields; it will simply be a redundant field containing the full value from the three other fields, for use only with the new HR software program.

Deliverable Descriptions & Requirements

Entity Relational Diagram (ERD)

The ERD submitted for this deliverable should meet all the following requirements:

- Include all entities and attributes needed from the original database
- Include all appropriate relationships (including recursive or hierarchical, if applicable).
- Include a primary key for every entity.
- Include a foreign key for every relationship.
- Matching datatypes and sizes for all foreign keys with their associated primary keys.
- Appropriate data-types and sizing for all fields.
- Appropriate decisions on which fields should be mandatory or not.
- Crow's-foot notation and physical data-types displayed.
- Where applicable, other constraints may be indicated in your ERD.

Data Definition Language (DDL) Script(s)

The DDL scripts submitted for this deliverable should meet all the following requirements:

1. An ERD is a blueprint for a database. The data structures created by your scripts should match your ERD **exactly** in every manner, including structure, relationships, table/field names, datatypes and constraints.
2. Create or alter all entities, attributes, constraints and datatypes present in your new ERD.
3. Create or alter all objects in an order that the full script can be run as a whole, with no errors.
4. Create or alter all key constraints (Primary and Foreign Keys), as identified in your ERD.
5. Create or alter all appropriate data integrity constraints (NOT NULL, CHECK, DEFAULT, etc.).
6. Create or alter any sequences required by aspects of your design.
7. Create DROP statements for all applicable database objects.
8. If temporary columns are added for use during the data migration, they should also be removed later.

Data Manipulation Language (DML) Script(s)

The DML scripts submitted for this deliverable should meet all the following requirements:

1. **All** data from the original database should be properly inserted into your newly designed database.
2. In some cases, new data that did not exist in the previous database may need to be added.
3. There may be some data in the original database that does not meet the constraints in your new database. In these case, the data will need to be manipulated or transformed during insertion.
4. All data should be added in a single running of your DML script, so the order in which tables are populated should be appropriately considered.
5. All data used for primary key fields should be unique between all records in a particular table, and should not contain any missing values.
6. All data used for foreign key fields should demonstrate the relationship to related data in the associated parent table, and match in datatype and size.
7. Where applicable, DML scripts should show proper, dynamic of use sequences.
8. All data records added to the database should adhere to all data integrity constraints established in the design.

TEST PLAN SCRIPT

The Test Plan script submitted for this deliverable should meet all the following requirements:

1. Each test case should be commented in the script, to indicate the purpose for the test, and the expected results.
2. Include queries to verify the creation of all required database objects. (Hint: Research the system table named “User_Tables”)
3. Include queries to verify record counts between the original and new databases.
4. Include queries to verify the integrity of all primary to foreign key relationships.
5. Include the creation of a series of test records to ensure new records meet all database constraints and new orders, customers, employees, etc. can be successfully added to the system.
6. Include the removal of all test records mentioned above.