Assignment 1

Start date – available the week of Sept 16 and due by Sunday, Oct 6

**Submission instructions**

Full set of scripts and jpgs are to be zipped and uploaded to Brightspace (in your specific section) by due date (Oct 6th). You have three attempts to submit, the last upload will be evaluated. If you upload a revised version, submit *a new zip file.*

Included in your submitted zip file:

* ERD diagram of the Inventory database in .jpeg file or .pdf format.
* All DML and DDL scripts – use tablename\_ddl or tablename\_dml as naming standard (for example, if you are making changes to the COUNTRY table filename would be COUNTRY\_DDL, if you are changing the data inside the COUNTRY table then filename would be COUNTRY\_DML)
* Submit both .doc(x) and .sql files.
* Separate DDL and DML statements in two .sql files.
* Create a separate DDL script for your ‘create view’ statements
* Ensure commands in both files can be executed sequentially and independently.

Refer to the rubric to verify that you have met the requirements.

**Objective**

* Use PG Modeler to update your database design
* Extend, enhance and populate existing database.
* Enforce constraints to maintain referential integrity.
* Write queries using JOIN, subquery.
* Create VIEWs.

**Business Requirement**

The trading company sells its products to customers in other countries, it wants to identify its customers by country. The company also purchases products from different countries and wants to keep track of country of origin for each of the products.

**Creating a new set of database objects and physical ERD**

* You are required to modify the two scripts (DDL and DML), run and test them.
* The scripts will create the database and the tables.
* Using PGModeler to import these new tables.
* Rearrange the entities in the ER Diagram if required*, marks are awarded for format and presentation.*

**Country Table (COUNTRY\_T)**

* Create a country table, call it COUNTRY\_T with the following attributes
  + Cntry\_Code, character 3, Primary Key
  + Cntry\_Name, variable length 30 characters
  + Cntry\_Population, integer, mandatory
* Add a primary key constraint, use the same naming convention for constraints as in the Inventory database, begin the name with PK\_.

*Note:* Do not copy the statements shown below directly to the DDL file. Think how these specifications can be translated to SQL statements.

Think how the requirement for mandatory data is implemented in SQL. What is the key word for variable length 30.

**City Table (CITY\_T)**

* Create a city table, call it CITY\_T with the following attributes.
  + City\_ID, INTEGER, Primary Key
  + City\_Name, variable length 30 characters
  + Cntry\_Code, fixed length character, 3 City\_Population, integer
* Add a foreign key constraint on the Cntry\_Code attribute. Reference it to Cntry\_Code in the COUNTRY\_T table.

**ALTER Customer\_T table**

* Add a column to the Customer\_T table.
* Column name should be Cust\_Country, use an appropriate datatype to accommodate the Country Code (Cntry\_Code) from the Country\_T table.
* Add a foreign key constraint to the Customer\_T table, Cust\_Country references Cntry\_Code in the Country\_T table. Use the same naming conventions for this constraint.

**ALTER Product\_T table**

* Add a column to the Product\_T table, call it Cntry\_Origin. This attribute indicates the country of origin of the product.
* Add a foreign key constraint to the Product\_T table to reference CntryCode in the Country\_T table.

**Populate City and Country table**

* Add about 10 countries and 20 cities.
* You may use the existing data from the world database.
* Add at least the following six countries in the COUNTRY\_T table.
* The sample code should work in the table you create.
* All INSERT statements you write should have the names of attributes in it, as shown in the sample below.

**Modify all INSERT statements in the Inventory-DML.sql file**

INSERT INTO COUNTRY\_T( Cntry\_Code, Cntry\_Name, Cntry\_Population ) VALUES( ’RUS’, ’Russian Federation’, 144192450 );

INSERT INTO COUNTRY\_T( Cntry\_Code, Cntry\_Name, Cntry\_Population ) VALUES( ’MEX’, ’Mexico’, 119530753 );

INSERT INTO COUNTRY\_T( Cntry\_Code, Cntry\_Name, Cntry\_Population ) VALUES( ’DZA’, ’Algeria’, 40400000 );

INSERT INTO COUNTRY\_T( Cntry\_Code, Cntry\_Name, Cntry\_Population ) VALUES( ’CHN’, ’China’, 1376049000 );

INSERT INTO COUNTRY\_T( Cntry\_Code, Cntry\_Name, Cntry\_Population ) VALUES( ’CHL’, ’Chile’, 18006407 );

INSERT INTO COUNTRY\_T( Cntry\_Code, Cntry\_Name, Cntry\_Population ) VALUES( ’CAN’, ’Canada’, 36155487 );

**Populate Customer and Product tables**

* Modify the insert statements from Lab 04 on page 114 to accommodate the new fields.
* Add at least one customer from Canada.
* Add at least one product from Russian Federation and one product from Chile.
* Add City\_T and Country\_T to the DROP TABLE list
* You should be able to run the DDL file multiple times. Country\_T table is the last one to be dropped. City\_T table should be dropped before the Country\_T table.
* Change the order in the DROP TABLE list, observe the error message, determine the cause of the message.

*Aside: Tables need to be created in a certain order, foreign key constraints determine this. If the order of creation is changed, use ALTER TABLE to add constraints. Your database should have six tables, four tables from the Inventory Database and two tables from the world database.*

**Create Views**

* Write two views of your choice.
* Use the naming convention as 〈ViewName〉\_V.
* CREATE VIEW is a DDL statement, it should be in the file which has DDL statements.   
  Use DROP VIEW IF EXISTS in the DDL file before the DROP TABLE statements. Tables cannot be dropped if there are views that depend on them.

**DELETE Statements**

* After you have populated the Customer\_T and Product\_T tables, run the following DELETE statements.
* Indicate the error number and error message. Explain in your own words the reason for failure.
* DELETE FROM Customer\_T WHERE Cust\_Country = 'CHL';
* DELETE FROM Product\_T WHERE Cntry\_Origin = 'RUS';

**JOINS**

* Write two joins. You may use RIGHT JOIN and/or LEFT JOIN
* Write a SQL statement to list countries that do not have any Customers.
* Write an SQL statement to list countries from which no products are bought.

**Mandatory Data**

Your DML file should contain the following statements. Place them in the appropriate sections of your code.

-- mandatory data for customer table

INSERT INTO Customer\_T( Cust\_ID, Cust\_FName, Cust\_LName, Cust\_Phone, Cust\_Address,

Cust\_City, Cust\_Prov, Cust\_PostCode, Cust\_Country, Cust\_Balance )

VALUES( ’C097’, ’Aze’, ’Balai’, ’7-731-707-7243’, ’Ulitsa Aleutskaya’,

’Valdivostok’, ’VL’,’VL7SK4’, ’RUS’, 0 );

-- mandatory data for product table

INSERT INTO Product\_T( Prod\_Code, Prod\_Description, Prod\_InDate, Prod\_QOH, Prod\_Min,

Cntry\_Origin, Prod\_Price, Prod\_Discount )

VALUES( ’P2119’, ’Organic Chard’, ’2018-01-14’, 240, 100, ’RUS’, 15.00, 5 );

-- mandatory data for product table

INSERT INTO Product\_T( Prod\_Code, Prod\_Description, Prod\_InDate, Prod\_QOH, Prod\_Min, Cntry\_Origin, Prod\_Price, Prod\_Discount )

VALUES( ’P2020’, ’Organic Collard Green’, ’2018-01-14’, 140, 110, ’RUS’, 11.00, 5 );

-- mandatory data for invoice table

INSERT INTO Invoice\_T( Invoice\_Number, Cust\_Id, Invoice\_Date )

VALUES( ’I88001’, ’C097’, ’2018-01-15’ );

-- mandatory data for Invoice\_Line Table

INSERT INTO Invoice\_Line\_T( Invoice\_Number, Invoice\_Line, Prod\_Code,

Line\_Units, Line\_Price )

VALUES( ’I88001’, 1, ’P2119’, 3, 15.00 );

INSERT INTO Invoice\_Line\_T( Invoice\_Number, Invoice\_Line, Prod\_Code,

Line\_Units, Line\_Price )

VALUES( ’I88001’, 2, ’P2020’, 3, 11.00 );

*Tips & Hints*

*Object Names cannot be duplicated*

*Ensure the constraint names for foreign keys are unique.*

*Duplicate Values in Prime Keys are Not Permitted*

*Do not run the DML without first running the DDL.*

*After adding insert statements to the DML file you will need to first run the DDL and then run the entire DML file.*

*You can execute a single DML statement, if the prime key has not been entered before.*

*Create prime key constraints before creating foreign key constraints*

*Foreign key constraints cannot be created unless a prime key is defined in the parent table.*

Suggested Schedule

Week 4 : Identify entities, relationships and attributes on paper. Prepare draft ERD on paper. Create Tables, write constraints, reverse engineer.

Week 5 : Write queries and views. Test database, test queries.

Week 6 : Refine and Submit.

Rubric - Assignment 01 Fall 2019

|  |  |  |  |
| --- | --- | --- | --- |
| No | Requirement | Maximum | Earned |
| 1 | Formatting & Presentation – proper ddl and dml files and data model (ERD) showing changes | 3 |  |
| 2 | Country\_T and City\_T created. | 3 |  |
| 3 | Primary & Foreign Keys defined. | 3 |  |
| 4 | Delete statements tested and before/after screenshots provided (counts will suffice) | 3 |  |
| 5 | INSERT and SELECT statements tested and functional | 3 |  |
| 6 | VIEW’s Created. | 3 |  |
| 7 | JOIN statements tested and functional. | 3 |  |
| 8 | Demonstration given. | 3 |  |
|  | (Maximum possible marks 24) Total | 24 |  |