

### Project Assignment

Due Monday, October 19, 2020 @ 6:40 pm

Refer to the attached description for the Eggshell Home Builder Case Study. You'll use this case study to design and implement a database schema, develop triggers, procedures and other objects to support their business operations.

The project will be described in 3 parts:

- 1) Design and Development of a Logical model
- 2) Design and Development of a Physical Model
- 3) Design and Development of objects supporting DB

---

### 1. Logical Data Model

Using the information contained in the Eggshell Home Builder Case Study, develop a LOCAL LOGICAL data model. State any assumption necessary to support your design. When developing your logical model, be sure to follow the steps outlined in C&B in Chapter 16 - 18. The output should be a well-developed E-R diagram, a list of normalized schemas, and a well-developed data dictionary.

**You will develop the following 3 items.**

#### 1. Data Dictionary

- a) For documenting entities, follow the format shown in Fig 16.3 on page 425 in CB&S.
- b) For documenting attributes, create a data dictionary following the format shown in Fig 16.4 on page 428. Include the following: entity name, attribute name, attribute description, data type and length, domain of attribute, default value, primary key?, alternate key?, multi-valued?, derived? (if derived, include calculation), nulls allowed?

#### 2. E-R Diagram

Include participation and cardinality. Use the UML modeling notation depicted in CB&S. For attributes, show only the primary key on the E-R diagram. Be sure to specialize/generalize where appropriate. Do not submit a hand-drawn diagram. You may use whatever diagramming tool you wish, Oracle Data Modeler, Powerpoint, Word, Visio, etc.

#### 3. Requirements and assumptions regarding business rules

- a) List all transaction requirements that are associated with the case study. Be sure to list at least 4.
- b) List any assumptions you made or additional business rules that support your logical model that are not explicitly stated in the Eggshell Home Builder Case Study.

## 2. Physical Data Model

Using the logical data model you created, develop a physical database design.

The physical design should include DDL scripts for the creation of all tables and views, as well as enforcement of integrity constraints. For integrity constraints use the appropriate mechanisms to enforce the following:

- a) required data
- b) domain constraints
- c) entity integrity
- d) referential integrity
- e) enterprise constraints

You must enforce rules defined by the case – in addition you may choose to come up with additional ones that are reasonable and do not conflict with what was stated in the case.

Your project will be judged on how well you enforce the business rules & integrity constraints.

Enforcement mechanisms **MUST** include:

- check constraints
- table constraints
- mandatory columns enforced where required
- participation enforced for foreign keys
- default values enforced where appropriate
- look-up tables
- database triggers

You must also use at least 2 views and sequences for the population of surrogate keys, where appropriate.

**You will develop the following item(s).**

1. 1 SQL script creating ALL of your objects (tables, views, sequences, etc...)
2. 1 SQL script to insert data into your tables. You should create sample data for each of the tables on your own. Make sure you insert enough rows in each of the tables as is appropriate. By this, I mean, insert enough rows so that I can test different cases. A good rule of thumb would be 10 rows per table though this might vary according to the type of data in each table.
3. Code for any database triggers, procedures, functions, and/or packages.

## 3. Database Objects

In addition to the logical and physical model you created, you must add database objects that will support your business rules.

You should deliver the following objects:

- Database level authentication – I will log into your team schema to test functionality of your database.
- A minimum of 2 roles with different level of permissions to view/modify data
- A minimum of 1 database job, which will schedule a regular operation to happen periodically. Frequency of activity should be set as an input parameter, so that I would be able to verify that it is working appropriately.
- Alternate Indexes – group identified a minimum of 1 alternate index and explanation why they might improve the efficiency of a query
- A minimum of 1 instance of de-normalization practice, and explanation why it improves efficiency of system
- Reports using SQL\*Plus or Graphical Report Tool (Oracle Application Express). A minimum of 2 (two) reports are to be created the meet a business need for Eggshell Home Builders. Each report must join at least 2 tables.

Additionally, you should prepare Data Warehouse Migration Guide, which would identify a minimum of 1 FACT and at least 3 dimensions that could be potentially migrated to a Data Warehouse.

---

### **What I'll be grading on:**

- Completeness of your ERD. You should make sure you have all the entities you need and that the relationships make sense and will lead to the appropriate foreign keys and linking tables, if appropriate. Your cardinality and participation should be correct. Be sure to label each relationship (both ways).
- Structure of your tables. Make sure your tables are defined appropriately, all columns having the right data type, length, NULL, NOT NULL. Primary and foreign keys should be defined for tables as appropriate.
- Other database objects. You need to include definitions for other database objects like sequences, views, procedures, triggers, and functions, jobs etc.
- How your solution supports assumptions and business rules stated.

### **What you should hand in – deliverables checklist**

This submission will consist of the following items:

- 1) an ERD

Copy and paste your ERD into an MS-Word file and call it groupnum\_erd.doc where groupnum is equal to your particular group project number. So, if your group number is 1, the name of this file would be G1\_erd.doc.

- 2) a SQL script for creating tables, views, triggers, sequences, and procedures, packages, functions, and populating the tables with data

Table definitions and row insertions. Put your CREATE TABLE statements in a Notepad file that I can run using the SQL> START command (same as the @ sign). The first set of commands in the file should be a list of DROP TABLE commands, just like in the dreamhome\_tables.sql. Also, if you create other database objects, like sequences or views, please include DROP statements for them as well. You should make sure that the tables are dropped in the order that corresponds to the disabling of foreign keys. Another way to drop tables without considering constraints is to drop all of the constraints first. To do it this way, look at the online documentation for the DROP CONSTRAINT command.

After the DROP TABLE commands, include your CREATE TABLE commands. If you have other database objects that you create, include the CREATE statements in this file as well at the end of the file. Finally, include an entire set of INSERT statements to populate the tables with data. Make sure the order of your inserts conforms to the foreign key dependencies.

Also include in the script, the code to create any database triggers, procedures, functions, and packages.

Call this file eggshell.sql. If your group number is G1, the name of this file would be G1\_eggshell.sql.

- 3) Report files (.sql). For each one of these items, you need to include your project group number in the file name
- 4) Explanatory texts

Make these MS-Word documents:

- a) Requirements Document (summarized version of case description)
- b) Data Dictionary (this may also be Excel)
- c) Entity Description Document
- d) Object Description Document - this document should list and describe all objects that you've created.  
Minimum: 2 sequences, 2 views, 2 procedures, 1 function, 1 package, 2 database triggers, 1 scheduled job, 2 roles, 1 instance of de-normalization, 2 alternate indexes
- e) Business Rules / Assumptions - this document should also include a list of any additional files you have sent me and an explanation of why you are sending them. Call this document explain.doc.
- f) Database User Guide, which I will follow to test your application. If your group number is G1, the name of this file would be G1\_user\_guide.doc. This document should contain at least 5 scenarios of using your DB to solve some business issues.  
Examples of test/use cases has been presented at the end of the document.
- g) Data Warehouse Migration Document

## **Presentation Guidelines**

YOU DO NOT NEED TO PRESENT YOUR FINAL PRESENTATION TO THE CLASS BUT YOU SHOULD PREPARE AS IF YOU NEED TO. You need to create a presentation in MS Powerpoint

describing the functionality of the database. It should contain enough detail to describe the functionality of the database, business rules enforced, screen shots, reports output, etc...

Assume that your group will be given 10 minutes for presentation to an executive board level organization (Board of Directors, CEO, CFO, CIO, CISO, etc...)

## **Final Submission**

Please zip up all content into 1 file named **<GroupNumber>\_eggshell.zip**. The final submission is to be completed via the digital dropbox NO later than 6:40 pm, October 19, 2020. **DO NOT BE LATE.** Failure to submit the project by 6:40 pm will result in a 0 for the project.

Each group must also submit ONE Peer Evaluation Form signed by each team member. This must be delivered to class on October 19, 2020 (or before). Failure to submit the form, or absent a team member, will result in a 10% penalty for the project grade.

### **Types of questions to think about when defining business rules:**

- 1) How would the system allow buyers to provide changes to their initial order, how would that be constrained?
- 2) What happens to expired contracts? What if someone wants to break it?
- 3) Are there any operations that need to be performed or reported on regular basis, can they be automated?