

DBS311 Assignment#2

Due Date: August 5, by 11:30pm

Instructions:

Group size: Ideally 3-4 persons in a group. You may work on the assignment alone, but the requirement does not change. Deadline will not be extended due to group issues. Please write full name and id of each group member (no nick name). Only group leader needs to upload in blackboard within deadline. Absolutely no email submission will be accepted.

Part 1: Complete ERD with crow's foot notation based on Part 1 narrative. [5 marks]

Part 2: DDL/DML script to create the database and insert 5-10 records in each table. [10 marks]

Part 3: One stored procedure. [5 marks]

Part 4: A schema-less NoSQL design using sample data from part 2. You are to denote collection name with sample documents. [10 marks]

Submission: zip folder containing (1) a pdf file that will include the ERD and the screenshots of records and table structure (select * from each of your tables, describe each table) and the successful run of stored procedure with code and (2) SQL Scrip file. Zero will be assigned if screenshot of successful run output is missing. (3) Include the collections with sample data in the same pdf file for Part 4.

Part 1 BUSSINES NARRATIVE for “Virtual Songs Ltd.” music store

This case involves a Database creation for a Web-based music store called “Virtual Songs Ltd.” that sells music Discs over the Internet.

Virtual Songs Ltd. wants to keep information about their customers like: customer number, last name, first name, date of birth, street address, city, state (province), zip (postal code), phone and email address.

Each customer may order many orders (or even none if prospective client) and each order will contain: order number, order date and order total. Order must exist for only one customer. Customers will always pay for their order with the credit card, so information about it is stored in the dependent (on Order) listing called “Payment”. It will contain: card number, name on card, expiry date and card type (like Visa, Master, American-Express etc.). Order must be paid with the credit card before on-line transaction is saved and it has to be for at least one Disc. Every payment corresponds to an order and an order may contain multiple lines i.e., Discs.

There is listing of all Discs called “Recording” and it is explained through: recording id, title, performer’s name, selling (or current) price and quantity in stock.

Each Disc belongs to a certain type or category of music like R&B, Jazz, Metal Rock, Alt Rock, Easy Rock, Classic Rock, Hip-Hop etc. and this will be stored in a look-up listing with category id and category description.

Also, each Disc is produced by a certain music label (company) like Sony, Columbia, Virgin etc and it is explained through: label id, label name and its URL site.

You should assume that there may exist a brand new category of music so that no Disc is of that type yet, but for the label you will assume it can not be without any Disc in the catalog.

Each order line will contain information about the actual price and quantity ordered. This price is a historical price of a Disc at the time of being sold to the customer. For example, Disc called “Eyes Open” by “Snow Patrol” may have cost \$14 in September, but in November it may have cost \$16 and for the Christmas sale only \$12. The latest actual price is then equaled to the selling price in the “Recording” listing (this will be done through an Update Trigger in the Application code). So, selling or current price is always the most current price of the Disc.

Music store also wants to track a history of price changes for any Disc, so that dependent listing “Recording History” needs to be created where each Disc can be tracked by price number (like 1, 2, 3 etc.) its old price (like \$14, \$16, \$12) and the start date of each price like (09-SEP-06, 02-NOV-06 and 10-DEC-06).

“Virtual Tunes Ltd.” manages its Disc stock purchase by dealing with label’s sales reps. Each label will have one or more sales reps and they are tracked by: rep number, last name, first name, phone number and email address. Each sales rep has divided its territory in regions, where they keep info only about region id and region name (like US-North, US-West, Canada, Europe, Asia etc.) It is possible that same region is covered with none or more than one sales rep.

Note: Create meaningful names for all attributes and use # symbol for PK and (FK) for FK.
Include ALL attributes for each entity incl. composite PK and every FK.

Part 2 DDL/DML script

Using the Songs ERD, you need to provide 3NF process and SQL scripts with DDL / DML statements that will implement a Physical Solution for that Database Scenario as:

1. 3NF process: define any dependencies and normalize (Hint: Only Customer table needs to be reduced to 3NF).
2. DDL scripts for creation of all Tables from the ERD.
3. Adding all PK, FK, UK and CK Constraints needed for these tables either with CREATE TABLE (UK, CK) or later with ALTER TABLE statements (PK, FK)

4. DML scripts for at least five rows of sample data for each table

Check constraints you must implement:

- *OrderTotal* in table ORDERS must be a positive number
- *CardType* in table PAYMENT must belong to the set of values {VISA, MAST, AMEX}
- *CategoryDesc* must belong to the set of values {'R&B', 'Jazz', 'Metal Rock', 'Alt Rock', 'Easy Rock', 'Classic Rock', 'Hip-Hop'}
- *Qty* in table ORDER_LINE must be non-negative integer number
- *ActPrice* in table ORDER_LINE must be greater than zero
- *Price* in table RECORD_HIST must be positive decimal number not greater than \$999.99
- *You must add not null constraints as you see fit.*

Unique constraints (Not shown in ERD):

- *Email* in table CUSTOMER
- *Email* in table SALES_REP

Part 3 Stored Procedure

Write a procedure named *Recording_Selling_Price_Report* to show the number of recordings in each price category as follows:

If selling price is less than the average selling price then the category is LOW
Else the category is HIGH

The stored procedure must print how many of LOW and HIGH recordings are available in the database.

Part 4 NoSQL Design

You need to identify the collections you need in this part and display sample json data inside them. For example, the Customer collection could contain {"_id":1,"LastName":"Alam", "FirstName":"Tanvir", "DoB":"1/1/2000", "Email":"Tanvir.alam@senecacollege.ca", "phone":"800-800-8080", "Zip":"A1A1A1"} etc. You must use the sample data you used in Part 2.

You do not need to implement using MongoDB. However, you need to research how to handle 1-M relationship in NoSQL to display in your sample data design.