Assignment 4 (SQL)

Lab Three-Part 1- Northwoods University Design -

The Northwoods University Student Registration Database

Northwoods University has decided to replace its aging mainframe-based student registration system with a more modern client/server database system.

School officials want students to be able to retrieve course availability information, register for courses, and print transcripts using personal computers located in the student computer labs. In addition, faculty members must be able to retrieve student course lists, drop and add students, and record course grades. Faculty members must also be able to view records for the students they advise.

Security is prime concern, so student and course records must be protected by password access.

The data items for the Northwoods database are:

- Student name, address, telephone number, class (freshman, sophomore, junior, or senior), date of birth, PIN (personal identification number), and advisor
- Course call number (such as COM340), course name, credits, location, duration, maximum enrollment, instructor, and term offered
- Instructor name, office location, telephone number, rank (Professor, Instructor, etc.), and PIN
- Student enrollment and grade information.

The database must be able to allow multiple sections of the same course to be taught by different instructors and on different days and times.

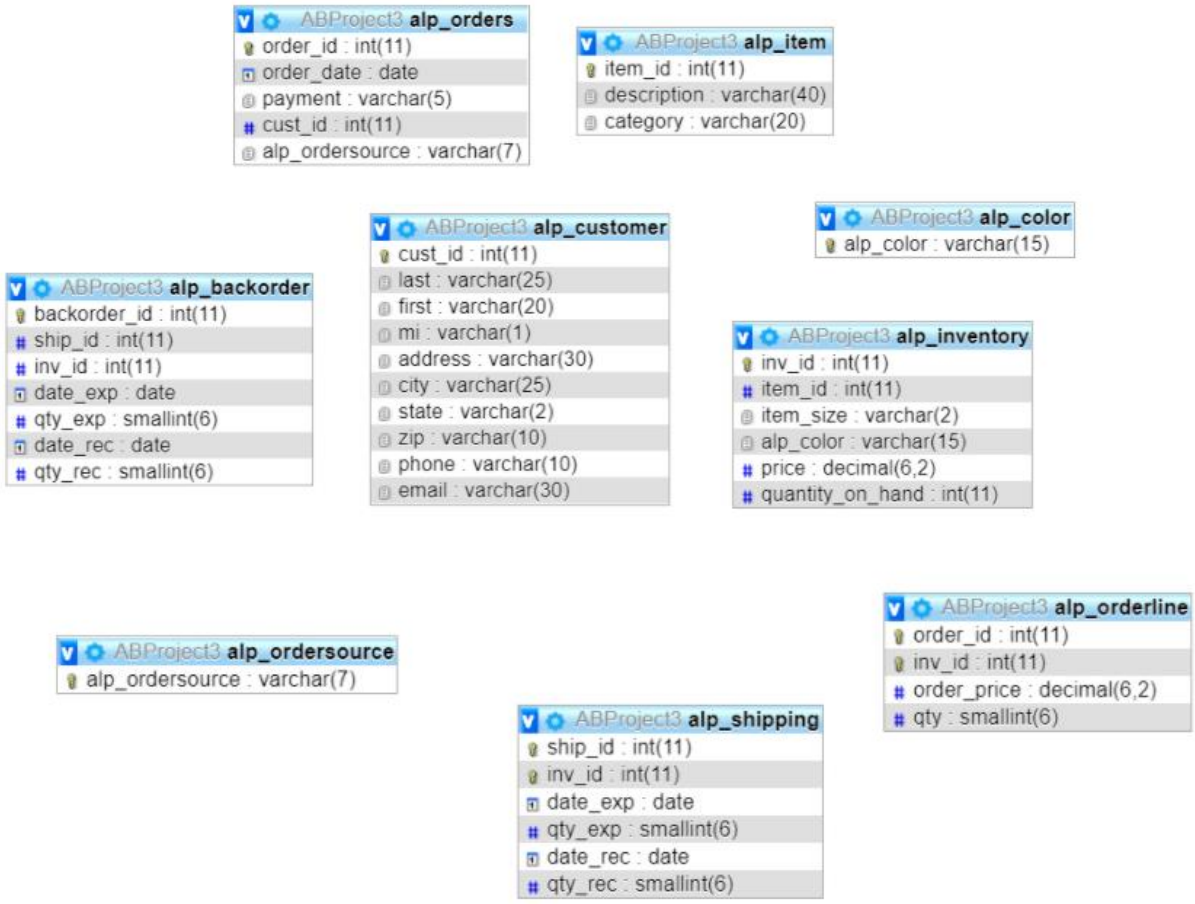
You have been asked to design a relational database to support the required tasks. You will provide three diagrams for this project.

- A Semantic Object Model of the basic information: Student, Course, and Instructor (this will be considered a conceptual diagram).
- An Entity Relationship Diagram (ERD) using the Chen model (this will be considered a more detailed conceptual diagram, showing each table that will exist and how they relate).

The tables will need to be in Third Normal form.

Project three Part2 : Simple Queries -

You will need to write SELECT commands to answer the following questions. Print the SELECT command you have the **Alpine Inventory database (see the sqema below:)**



1. Select all the rows from the alp_item table. Display Description and Category
2. Select alp_inventory items that have a price of less than 100 dollars. Display Id, Size, Price and Quantity on Hand.
3. List the alp_inventory items that have a quantity on hand of more than 30. Display Id, Quantity on Hand and Price
4. List the alp_customers in 'Washburn ' and 'Silver Lake'. Display first_name, last_name, mi and city
5. Select the prices that occur in the alp_inventory table. A specific price should only appear once. Display the Price.
6. Select the alp_inventory items that are in stock. Display Id, Price and Quantity on Hand
7. Select the alp_orders placed before November 1, 2007. Display the Order_id and Order_date

8. List the alp_inventory items that are ‘Coral’ or ‘Olive’ and have a Quantity on Hand of less than 105. Display Id and Quantity on Hand
9. List the alp_items that contain the word ‘Fleece’ in the item description. Display Id, Description, and Category
10. List all the alp_inventory items that do not have a size or a color assigned. Display the Id and Price.
11. Determine the number of orders placed on 10 October 2007. Display Number of Orders Hint: Use the COUNT function
12. Determine the extended price for each row in the alp_orderline table. Display Order_id, inv_id, and Extended Price
13. Determine the number of different items on each order. Display Order_ID and Number of Items Hint: Determine the number of different products ordered, not the total quantity ordered. This query requires a GROUP BY clause.
14. Determine the number of orders placed by each customer. Only display the data for customers who have placed more than one order. Display Cust_id and Number of Orders Hint: This query requires a GROUP BY clause and a HAVING clause.
15. Determine the order total for each order that has an order total greater than 100. Display ‘*Order Id*’ and ‘*Order Total’.* Make sure the results are in ascending order total sequence.
16. Determine what is the most expensive price, the least expensive price, and the average price in the alp_inventory table.
17. Now that you know the average price in the inventory table, display all of the information for inventory items whose price is greater than the average price.

Project Three part 3: Advanced Queries

You will need to write SELECT commands to answer the following questions. **You will use the same tables that you used in Project 2 (Alpine Adventures).**

1. List the inventory items that are the same color as inventory item 23. Display the Inventory_ID and color (You must use a subquery in this query. You cannot hard-code the color.)
2. List the inventory items that have an average price greater than the average price of all the inventory items.
   Display the Inventory_ID and the Price (You must use a subquery in this query)
3. Select the following for each inventory item: Inventory_ID, Item Description, item_size, color, and Price
4. Select all orders and the names of the customers who placed the orders. Display Order_ID, Order_Date and the Customer Name
5. Display orders that contain the item Men's Expedition Parka. Display Order_ID, Order Date
6. List the items ordered for Order Id 6. Display Inventory_ID, Extended Price and Qty
7. Display the in-stock quantity for each item. Display Item_ID and Number of items (Order the list by the Item Id in ascending order. This query requires a GROUP BY clause)
8. List the quantity of each inventory item sold. Display Inventory ID and Number Sold. (Order the list by the number of items sold in descending order. This query requires a GROUP BY clause and the SUM function.)
9. Determine the order total (dollar amount) for each order and the customer who placed the order. Display the Order_ID, Customer Name and the Order Total.
10. Display all of the inventory information for inv_ids that do not have shipping records.
11. Display all of the inventory information and backorder information for inv_ids that are on backorder
12. Display the customer first and last name and order total for order number 5
13. Display the inv_id, description, price and color of the least expensive inventory item that we have in the inventory table. Use the join keyword to join the inventory table and item table. You need a subquery to determine the least expensive price then use an outer query to find all inventory items at that price.
14. Display the first name, last name, email address and order total for customers that have placed an order at Alpine Adventures.
15. Modify the above query to display ALL customers, whether they have placed an order or not.

Project three part 4 Procedures and Triggers

In this project you will develop five procedures for Alpine Adventures that will be stored on the server. A Procedure is a block code that accomplishes a task. You will follow the steps that were outlined in the Procedure Tutorial. Be sure to test if the procedure exists so you can drop it if necessary and test that each procedure works correctly.

Procedure 1:

Write a procedure to update the *quantity_on_hand* column in the inventory table. It will accept two arguments (inv_id, qty). It does not return any value. The qty passed into the procedure will be added to the quantity_on_hand in the table.
Name the procedure sp_UpdateInventory

Procedure 2:

Write a procedure that will insert a row into the Orders table. It will accept arguments for each of the columns (except the order_id because that is generated by SQL Server). Name the procedure sp_InsertOrder

Procedure 3:

Write a procedure that will allow a specified color in the inventory table to be changed to a different color (Hint: you will use the UPDATE command). This procedure will be passed two values: the old color and the new color. Test the procedure by changing the color 'Coral' to the color 'Pink'. Name the procedure sp_UpdateColor

Procedure 4

Write a procedure that will allow a user to cancel an order. The procedure will be passed an order_id and the necessary information will be deleted to cancel the order. Name the procedure sp_CancelOrder.

Procedure 5:

Write a procedure that will calculate the total for a specified order_id. The procedure will receive one input parameter (order_id) and return one parameter (order_total). Name the procedure sp_CalcOrderTotal.

Trigger 1:

Create a trigger that will automatically create a shipping record if an inventory item is update with a quantity_on_hand < 5. You can hard code the date_expected or research on the Internet how to get the current date from the system and add 7 days to it for the date expected. Name the trigger: tr_updateInventory.