**Project 3 (SQL - Part2)**

|  |
| --- |
| **Project Three part 3: Advanced Queries**  You will need to write SELECT commands to answer the following questions. Print the SELECT command and your results using PHPMyAdmin. **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)   1. Select the following for each inventory item: Inventory\_ID, Item Description, item\_size, color, and Price 2. Select all orders and the names of the customers who placed the orders. Display Order\_ID, Order\_Date and the Customer Name 3. Display orders that contain the item Men’s Expedition Parka. Display Order\_ID, Order Date 4. List the items ordered for Order Id 6. Display Inventory\_ID, Extended Price and Qty 5. 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) 6. 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.) 7. 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. 8. Display all of the inventory information for inv\_ids that do not have shipping records. 9. Display all of the inventory information and backorder information for inv\_ids that are on backorder 10. Display the customer first and last name and order total for order number 5 11. 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. 12. Display the first name, last name, email address and order total for customers that have placed an order at Alpine Adventures. 13. 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. |