
Assignment 7 (60 points)
Due Date: Monday, March 2nd, 11:59 PM

Objectives

This assignment focuses on how to join data contained in various tables using conditions specified in the WHERE clause and using the JOIN keyword in the FROM clause. You should understand how the tables relate to each other as well as how to use the table aliases in the current and future assignments. This assignment reinforces the following objectives:

- Write an equijoin query and inequality join by using the WHERE clause.
- Learn the basic join operation rules.
- Write complex join queries with more than two tables and more than two columns.
- Write outer join and self-join-join queries.

If a sample display is provided, use that to determine the column order and column aliases to use in your result set.

All joins must use the ANSI standard joins in the FROM clause; joins handled in the WHERE clause will lose points for the assignment.

This assignment uses the tables associated with the *bookstore* Database.

Query 1:	Write a SELECT statement that joins the category table to the product table and returns these columns: category_name, product_name, and list_price. Sort the result set by category_name and then by product_name in ascending sequence.
Query 2:	Write a SELECT statement that joins the customer table to the address table and returns these columns: first_name, last_name, line1, city, state, zip_code. Return one row for each address for the customer with an email address of heatheresway@mac.com
Query 3:	Write a SELECT statement that joins the customer table to the address table and returns these columns: first_name, last_name, line1, city, state, and zip_code. Return one row for each customer, but only return addresses that are the shipping address for a customer.
Query 4:	Write a SELECT statement that joins the customer, orders, orderitems, and product tables. This statement should return these columns: last_name, first_name, order_date, product_name, item_price, and quantity. Use aliases for the tables. Sort the final result set by last_name and order_date. Sample Run: +-----------+------------+---------------------+------------------------+------------+----------+ | last_name | first_name | order_date | product_name | item_price | quantity | +-----------+------------+---------------------+------------------------+------------+----------+ | Brown | Christine | 2015-03-30 15:22:31 | Space Force 9 | 13.95 | 1 | | Brown | Christine | 2015-03-30 15:22:31 | Lenovo H50 Desktop | 1199.00 | 2 | | Goldstein | David | 2015-03-31 05:43:11 | HP Pavilion 15t Star | 1299.00 | 1 | | Goldstein | David | 2015-04-03 12:22:31 | From Another Planet | 49.99 | 1 | | Goldstein | David | 2015-04-03 12:22:31 | Dell XPS 8700 | 999.00 | 1 | | Hernandez | Gary | 2015-04-02 11:26:38 | Z Files | 49.99 | 1 |
Query 5:	Write a SELECT statement that returns the product_name and list_price columns from the product table. Return one row for each product that has the same list price as another product. Sort the result set by product_name.

	Hint: Use a self-join to check that the product_id columns aren't equal, but the list_price columns are equal.
Query 6:	Write a SELECT statement that displays the product ID, product name, list price, vendor name, and category name for all products. Sort the results by product ID.
Query 7:	Write a SELECT statement that displays the product ID, product name, and list price for products in the category whose category name value is Computer. Sort the results by product ID.
Query 8:	For each line item of a sales transaction, display the order ID, date of the order, name of the product that was sold, quantity sold, and the amount charged. Sample Run: +----------+---------------------+------------------------+----------+---------+ | order_id | order_date | product_name | quantity | amount | +----------+---------------------+------------------------+----------+---------+ | 1 | 2015-03-28 09:40:28 | Lenovo H50 Desktop | 1 | 1199.00 | | 1 | 2015-03-28 09:40:28 | Z Files | 2 | 99.98 | | 2 | 2015-03-28 11:23:20 | Toshiba Satellite S55T | 1 | 989.99 | | 2 | 2015-03-28 11:23:20 | Modern Science | 1 | 19.95 | | 2 | 2015-03-28 11:23:20 | Dell Inspiron 17 5000 | 1 | 1415.00 | | 3 | 2015-03-29 09:44:58 | HP Envy 750qe Desktop | 1 | 2517.00 |
Query 9:	Determine which orders have not yet shipped and the name of the customer who placed the order. Sort the results by the date on which the order was placed. Display the first name, last name, order ID, order date, and ship date.
Query 10:	Write an SQL statement that displays last name and first name of customers who have purchased a product that costs more than $1300.
Query 11:	Determine which products customer Christine Brown has purchased. Perform the search using the customer name, not the customer ID.
Query 12:	Write an SQL statement that displays last name, first name, and customer ID of customers who have ordered a product that has product id 2234.

Submission Instructions:

- For each of the queries above, submit the query and the result from running the query.
- You will need to label your assignment with your first initial, last name, and the name of the assignment. **Example:** hibrahim_assignment7.sql and hibrahim_assignment7.txt
- Zip the files to upload to Canvas. **Example:** hibrahim_assignment7.zip
- Submit the zipped file containing the script and output TXT via Canvas.
- Remember to include the query number as a comment at each step.
- Read your output TXT file before you submit it.