Shop.mdb

Orders				
Id	**OrderDate**	**OrderPrice**	**Customer_ID**	**Placed_By**
1	2008/11/12	1000	2224	333
2	2008/10/23	1600	4532	358
3	2008/09/02	700	3435	359
4	2008/09/03	300	2547	311
5	2008/08/30	2000	6764	324
6	2008/10/04	100	4532	333
…	…	…	…	…

Customers			
Id	**Name**	**Phone**	**Suburb**
2224	Schoemaker	32998764	Marsden
4532	Nilsen	38074277	Kingston
3435	Hansen	38050396	Loganlea
2547	Hartley	55460745	Beenleigh
6764	Jensen	55678362	Kuraby
6784	Bird	38028900	Waterford
…	…	…	…

Employees		
Id	**Name**	**Age**
333	Megan	14
358	Andy	21
313	Lisa	19
367	Heidi	20
327	Christian	16
357	Sandra	16
…	…	…

REVISION

1. Show the **relationships** between the tables in this database

2. Write the **WHERE** statement that would join the Orders table with Employees

3. Write the **WHERE** statement that would join the Customers table with Orders table

4. Write the **WHERE** statement that would join the Customers table with Employees

5. Write a query that:

- Displays the Order ID of all purchases by Schoemaker (using a Join):

- Displays the Order ID of all purchases by Nilsen or Hartley:

ORDERING

6. Write a query that:

- Displays all details of orders placed, sorted from earliest to latest dates:

- Lists the Names and Ages of employees, from oldest to youngest:

- Using a subquery, display the ages of employees, from oldest to youngest, who served the Customer with the ID “2224”:

- Using a JOIN, display the ages of employees, from youngest to oldest, who served the Customer with the ID “2224”:

SQL FUNCTIONS

7. Write a query that:

- Finds the average order price and renames the column:

- Find all the names of our important customers (those who spend **more** than the average):

- Finds the average age of our employees::

- Finds the youngest and oldest age of our employees::

- Finds the name of youngest and oldest employees::

- Lets your boss know how many customers we have on the database:

- Lets your boss know how many orders were placed, using the Order ID column:

- Lets your boss know how many days orders were placed:

- Calculates the total cost of all orders:

GROUP BY/ HAVING

8. Write a query that:

- Finds the total purchase costs of each customers' orders:

- Finds the total purchase costs of each customers' orders for each date:

- Finds customers who have spent a total cost less than $1000:

- Mrs Schoemaker phones up, wanting to know if her total order is greater than $1500, how would you write this query using Group By/Having:

Your Task:

It's your turn to develop questions for your friend to solve. You need to:

- Develop 5 -10 problems (these can either be queries they need to describe, or questions they need to develop queries for)
- Make sure these problems include a range of SQL Functions, Group By/Having, Count, Where, Subquery and Joins