

## Revision

1. Write a query that will display all the products which have a model associated with them. Your query should have two columns named 'Product Name' and 'Product Model'. These two columns should be retrieved from the Production.Product and Production.ProductModel tables.

Sample output:

	Product Name	Product Model
1	HL Road Frame - Black, 58	HL Road Frame
2	HL Road Frame - Red, 58	HL Road Frame
3	Sport-100 Helmet, Red	Sport-100
4	Sport-100 Helmet, Black	Sport-100
5	Mountain Bike Socks, M	Mountain Bike Socks
6	Mountain Bike Socks, L	Mountain Bike Socks
...		

Table to be used: Production.Product, Production.ProductModel

Number of rows: 295

2. In relation to the previous question, choose the statement/s that are correct:
  - a. The query required the use of an Outer Join to return the mentioned results
  - b. A join is required as data is obtained from different tables
  - c. The implemented join returns all the rows even if the condition is not met
  - d. The query used an Inner Join to retrieve only those rows which have a Product Model
  - e. None of the above

3. Write a query that will display four columns named 'Product Number', 'Product Name', 'Product Category' and 'Product Model'. The first two columns should be obtained from the Production.Product table, the third column should be obtained from the Production.ProductCategory table and the last column should be obtained from the Production.ProductModel table. You are to ensure that the following tasks are also performed:
- The chosen join category should return only those products which are assigned a category and a model. In simple words, the chosen join category should ensure that only the products with a valid category and model are to be displayed.
  - You are to sort your result using the first column in descending order.
  - No WHERE clause should be used in this query

Sample output:

	Product Number	Product Name	Product Category	Product Model
1	WB-H098	Water Bottle - 30 oz.	Bottles and Cages	Water Bottle
2	VE-C304-S	Classic Vest, S	Vests	Classic Vest
3	VE-C304-M	Classic Vest, M	Vests	Classic Vest
4	VE-C304-L	Classic Vest, L	Vests	Classic Vest
5	TT-T092	Touring Tire Tube	Tires and Tubes	Touring Tire Tube
6	TT-R982	Road Tire Tube	Tires and Tubes	Road Tire Tube

Table to be used: Production.Product, Production.ProductCategory,  
Production.ProductModel  
Number of rows: 295

4. Choose the statement/s that are true in relation to the previous question:
- The query makes use of 1 single Inner Join
  - Multiple Join operations are not required to obtain data from 3 or more tables
  - Rows which do not satisfy any of the JOIN condition/s will be returned
  - All of the above
  - None of the above

5. Write a query that will display two columns 'Product Name' and 'Product Model' from the Production.Product and Production.ProductModel tables respectively. You are to make sure that your query:
- Displays all the product models irrespective of whether they have a product associated with them or not.
  - Sorts your result according to the Product Name in ascending order

Sample output:

	Product Name	Product Model
1	NULL	ML Road Seat/Saddle 1
2	NULL	ML Mountain Seat/Saddle 1
3	NULL	Road-350
⋮		
9	NULL	LL Road Seat/Saddle 2
10	All-Purpose Bike Stand	All-Purpose Bike Stand
11	AWC Logo Cap	Cycling Cap
...		

Table to be used: Production.Product, Production.ProductModel

Number of rows: 304

6. Select all the statements that are relevant to the previous question:
- The query requires the use of an inner join as all models should be returned irrespective of products being associated to them.
  - An outer join is required so that all the models irrespective of whether products are associated with them are returned
  - This query could have been implemented using an inner join or a cross join.
  - All of the above
  - None of the above

7. You are to use set operations to display all the different (ProductCategoryID, Name) within the Production.ProductCategory and the Production.oldProductCategory tables. It is important that the ProductCategoryID and Name combinations are not listed more than once in the output

Sample output:

	ProductcategoryID	Name
1	1	Mountain Bikes
2	2	Road Bikes
3	3	Touring Bikes
4	4	Handlebars
5	5	Bottom Brackets
...		

Table to be used: Production.ProductCategory, Production.OLDProductCategory  
Number of rows: 37

8. Modify the previous query, such that this time round all the Product Categories (including duplicates) are displayed. Once again you are to display the ProductCategoryID and Name for each row.

Sample Output:

	ProductcategoryID	Name
1	18	Bib-Shorts
2	26	Bike Racks
3	27	Bike Stands
4	28	Bottles and Cages
...		

Table to be used: Production.ProductCategory, Production.OLDProductCategory  
Number of rows: 69

9. You are to use set operations to return all the new categories (ProductCategoryID, Name) that are included in the Production.ProductCategory table but are not found in the Production.OLDProductCategory table.

Sample Output:

	ProductCategoryID	Name
1	1	Mountain Bikes
2	12	Mountain Frames
3	2	Road Bikes
4	3	Touring Bikes
5	16	Touring Frames

Table to be used: Production.ProductCategory, Production.OLDProductCategory  
Number of rows: 5

10. Write a query that will display the ProductNumber and Name from the Production.Product table of all those products, whose Product Category name is 'Mountain Bikes'. Your query should handle the following requirements as well:
- You are to make use of sub-queries
  - You are to name the two columns 'Product #' and 'Product Name' respectively.
  - You are to sort your results via the Product Number in ascending order

Sample output:

	Product #	Product Name
1	BK-M18B-40	Mountain-500 Black, 40
2	BK-M18B-42	Mountain-500 Black, 42
3	BK-M18B-44	Mountain-500 Black, 44
4	BK-M18B-48	Mountain-500 Black, 48
5	BK-M18B-52	Mountain-500 Black, 52
...		

Table to be used: Production.Product, Production.ProductCategory  
Number of rows: 32

11. Modify the previous question such that the query now returns all the products in the Production.Product table which fall under a category which contains the word 'Bikes' in its name.

Sample output:

	Product #	Product Name
1	BK-M18B-40	Mountain-500 Black, 40
2	BK-M18B-42	Mountain-500 Black, 42
3	BK-M18B-44	Mountain-500 Black, 44
4	BK-M18B-48	Mountain-500 Black, 48
5	BK-M18B-52	Mountain-500 Black, 52

• • •

Table to be used: Production.Product, Production.ProductCategory

12. Write a query that will create a view named ProductCategoryVU. You are to make sure that the view follows the below guidelines:
- This view should have 3 columns, which are to be named 'Product Category Id', 'Category Name' and 'Last Modified'.
  - Users who make use of this view should have all the data in the ProductCategoryID, Name and ModifiedDate columns from the Production.ProductCategory table shown.

Table to be used: Production.ProductCategory

13. Write a query that will return all the rows and columns which are accessible through the view which was created in the previous question. Sample output:

	Product Category Id	Category Name	Last Modified
1	1	Mountain Bikes	2008-04-30 00:00:00.000
2	2	Road Bikes	2008-04-30 00:00:00.000
3	3	Touring Bikes	2008-04-30 00:00:00.000
4	4	Handlebars	2008-04-30 00:00:00.000

• •

Number of rows: 37

14. Write a query that will create a view named ProductVU. You are to make sure that the view follows the below guidelines:
- a. The column names of the view should be the same of those in the original table
  - b. The columns which are to be included in the view should be ProductID, ProductNumber, Name, Color, SellStartDate and ModifiedDate which are to be obtained from the Production.Product
  - c. The view should return only those products which have a Black, Silver or Red color

Table to be used: Production.Product

15. You are to modify the code in the previous question (without removing the view) such that:
- a. The names view's column names are changed to 'ID', 'Product #', 'Product Name', 'Colour', 'Selling Started On' and 'Modified Date'
  - b. Any DML operations performed via the view stay in the domain of the view
16. You are to write a query that will remove the created views named ProductCategoryVU and the ProductVU

\*\*\*