

CatalogID	SKU	SKU_Description	Department	CatalogPage	DateOnWebSite
20130001	100100	Std. Scuba Tank, Yellow	Water Sports	23	2013-01-01
20130002	100500	Std. Scuba Tank, Light Green	Water Sports	NULL	2013-07-01
20130003	100600	Std. Scuba Tank, Dark Green	Water Sports	NULL	2013-07-01
20130004	101100	Dive Mask, Small Clear	Water Sports	24	2013-01-01
20130005	101200	Dive Mask, Med Clear	Water Sports	24	2013-01-01
20130006	201000	Half-dome Tent	Camping	45	2013-01-01
20130007	202000	Half-dome Tent Vestibule	Camping	47	2013-01-01
20130008	301000	Light Fly Climbing Harness	Climbing	76	2013-01-01
20130009	302000	Locking Carabiner, Oval	Climbing	78	2013-01-01

FIGURE 2-40

Cape Codd Database
CATALOG_SKU_2013
Table Data

If you are using the Microsoft Access 2013 Cape_Codd.accdb database, simply copy it to an appropriate location in your Documents folder. Otherwise, you will need to use the discussion and instructions necessary for setting up the Cape_Codd database in the DBMS product you are using:

- For Microsoft SQL Server 2014, see online Chapter 10A.
- For Oracle Database 12c or Oracle Express Edition 11g Release 2, see online Chapter 10B.
- For MySQL 5.6 Community Server, see online Chapter 10C.

Once you have setup your Cape_Codd database, create an SQL script named Cape-Codd-CHO2-RQs.sql, and use it to record and store SQL statements that answer each of the following questions (if the question requires a written answer, use an SQL comment to record your answer):

- 2.16** There is an intentional flaw in the design of the INVENTORY table used in these exercises. This flaw was purposely included in the INVENTORY tables so you can answer some of the following questions using only that table. Compare the SKU and INVENTORY tables, and determine what design flaw is included in INVENTORY. Specifically, why did we include it?

Use only the INVENTORY table to answer Review Questions 2.17 through 2.39:

2.17 Write an SQL statement to display SKU and SKU_Description.

2.18 Write an SQL statement to display SKU_Description and SKU.

2.19 Write an SQL statement to display WarehouseID.

2.20 Write an SQL statement to display unique WarehouseIDs.

2.21 Write an SQL statement to display all of the columns without using the SQL asterisk (*) wildcard character.

2.22 Write an SQL statement to display all of the columns using the SQL asterisk (*) wildcard character.

2.23 Write an SQL statement to display all data on products having a QuantityOnHand greater than 0.

SAVE EACH
QUESTION

Q9-99

(see pg 61)

2.24 Write an SQL statement to display the SKU and SKU_Description for products having QuantityOnHand equal to 0.

2.25 Write an SQL statement to display the SKU, SKU_Description, and WarehouseID for products that have a QuantityOnHand equal to 0. Sort the results in ascending order by WarehouseID.

2.26 Write an SQL statement to display the SKU, SKU_Description, and WarehouseID for products that have a QuantityOnHand greater than 0. Sort the results in descending order by WarehouseID and in ascending order by SKU.

2.27 Write an SQL statement to display SKU, SKU_Description, and WarehouseID for all products that have a QuantityOnHand equal to 0 and a QuantityOnOrder greater than 0. Sort the results in descending order by WarehouseID and in ascending order by SKU.

2.28 Write an SQL statement to display SKU, SKU_Description, and WarehouseID for all products that have a QuantityOnHand equal to 0 or a QuantityOnOrder equal to 0. Sort the results in descending order by WarehouseID and in ascending order by SKU.

2.29 Write an SQL statement to display the SKU, SKU_Description, WarehouseID, and QuantityOnHand for all products having a QuantityOnHand greater than 1 and less than 10. Do not use the BETWEEN keyword.

2.30 Write an SQL statement to display the SKU, SKU_Description, WarehouseID, and QuantityOnHand for all products having a QuantityOnHand greater than 1 and less than 10. Use the BETWEEN keyword.

2.31 Write an SQL statement to show a unique SKU and SKU_Description for all products having an SKU description starting with 'Half-Dome'.

2.32 Write an SQL statement to show a unique SKU and SKU_Description for all products having a description that includes the word 'Climb'.

2.33 Write an SQL statement to show a unique SKU and SKU_Description for all products having a 'd' in the third position from the left in SKU_Description.

2.34 Write an SQL statement that uses all of the SQL built-in functions on the QuantityOnHand column. Include meaningful column names in the result.

2.35 Explain the difference between the SQL built-in functions COUNT and SUM.

2.36 Write an SQL statement to display the WarehouseID and the sum of QuantityOnHand, grouped by WarehouseID. Name the sum TotalItemsOnHand and display the results in descending order of TotalItemsOnHand.

2.37 Write an SQL statement to display the WarehouseID and the sum of QuantityOnHand, grouped by WarehouseID. Omit all SKU items that have 3 or more items on hand from the sum, and name the sum TotalItemsOnHandLT3 and display the results in descending order of TotalItemsOnHandLT3.

2.38 Write an SQL statement to display the WarehouseID and the sum of QuantityOnHand grouped by WarehouseID. Omit all SKU items that have 3 or more items on hand from the sum, and name the sum TotalItemsOnHandLT3. Show the WarehouseID only for warehouses having fewer than 2 SKUs in their TotalItemsOnHandLT3. Display the results in descending order of TotalItemsOnHandLT3.

2.39 In your answer to Review Question 2.38, was the WHERE clause or the HAVING clause applied first? Why?

Use both the INVENTORY and WAREHOUSE tables to answer Review Questions 2.40 through 2.52:

2.40 Write an SQL statement to display the SKU, SKU_Description, WarehouseID, WarehouseCity, and WarehouseState for all items stored in the Atlanta, Bangor, or Chicago warehouse. Do not use the IN keyword.

ADD MORE COLUMNS
TO DISPLAY TO
CHECK WORK

SQL 72-74