USE BIKE;

--Which component that had no sales (installations) in 2004 has the highest inventory value (cost basis)?

SELECT Top 1 Manufacturer.ManufacturerName, Component.ProductNumber, Component.Category, Component.ComponentID, SUM(component.ListPrice \* component.QuantityOnHand) AS [value] , BikeParts.DateInstalled

FROM Manufacturer INNER JOIN Component ON Manufacturer.ManufacturerID = Component.ManufacturerID

INNER JOIN BikeParts ON Component.ComponentID = BikeParts.ComponentID

WHERE DateInstalled NOT LIKE '2004'

group by Manufacturer.ManufacturerName, Component.ProductNumber, Component.Category, Component.ComponentID, BikeParts.DateInstalled

ORDER BY SUM(Component.ListPrice \* Component.QuantityOnHand) desc;

--Create a vendor contacts list of all manufacturers and retail stores in California.Include only the columns for VendorName and Phone.

--The retail stores should only include stores that participated in the sale of at least one bicycle in 2004

SELECT RetailStore.StoreName, RetailStore.Phone

FROM RetailStore inner join City on RetailStore.CityID = City.CityID inner join Bicycle on RetailStore.StoreID = Bicycle.StoreID

WHERE YEAR(OrderDate) = 2004 AND City.State = 'CA'

GROUP BY StoreName, Phone;

--List all of the employees who report to Venetiaan.

SELECT (SELECT LASTNAME

FROM Employee

WHERE EmployeeID = (SELECT EmployeeID

FROM Employee

WHERE LastName = 'VENETIAAN'))AS [MANAGER LAST NAME], EMPLOYEEID, FIRSTNAME, LASTNAME, TITLE

FROM Employee

WHERE CurrentManager = (SELECT EmployeeID;

--SELECT LASTNAME, EMPLOYEEID, LASTNAME, FIRSTNAME, TITLE

--FROM EMPLOYEE INNER JOIN

USE BIKE;

select two.LastName, Employee.EmployeeID, Employee.LastName, Employee.FirstName, Employee.Title

from Employee inner join employee two on Employee.CurrentManager = two.EmployeeID

where two.LastName = 'Venetiaan';

SELECT (SELECT LASTNAME

FROM EMPLOYEE

WHERE EMPLOYEEID = (SELECT EMPLOYEEID FROM EMPLOYEE

WHERE LASTNAME = 'VENETIAAN')) AS [MANAGER NAME], EMPLOYEEID,

FIRSTNAME, LASTNAME, TITLE

FROM EMPLOYEE

WHERE CURRENTMANAGER = (SELECT EMPLOYEEID

FROM EMPLOYEE

WHERE LASTNAME = 'VENETIAAN'

);

Use BIKE;

SELECT YEAR(ORDERDATE), AVG(DATEDIFF(DAY, ORDERDATE, SHIPDATE))

FROM BICYCLE

GROUP BY YEAR(ORDERDATE)

HAVING AVG(DATEDIFF(DAY, ORDERDATE, SHIPDATE)) > (

SELECT AVG(DATEDIFF(DAY, ORDERDATE, SHIPDATE))

FROM BICYCLE)

Use BIKE;

SELECT TOP 1 M.MANUFACTURERNAME, C.PRODUCTNUMBER, C.CATEGORY, SUM(C.LISTPRICE \* C.QUANTITYONHAND) AS AMT, BP.DATEINSTALLED

FROM MANUFACTURER M INNER JOIN COMPONENT C ON C.MANUFACTURERID = M.MANUFACTURERID

INNER JOIN BIKEPARTS BP ON C.COMPONENTID = BP.COMPONENTID

WHERE YEAR(BP.DATEINSTALLED) NOT LIKE 2004

GROUP BY C.PRODUCTNUMBER, C.CATEGORY, M.MANUFACTURERNAME, BP.DATEINSTALLED

ORDER BY SUM(C.LISTPRICE \* C.QUANTITYONHAND);

CREATE VIEW V24 AS

SELECT DISTINCT BikeParts.ComponentID

FROM BikeParts

WHERE Year(BikeParts.DateInstalled) = 2004

SELECT Manufacturer.ManufacturerName, Component.ProductNumber, Component.Category,

[EstimatedCost]\*[QuantityOnHand] AS [Value], V.ComponentID

FROM Manufacturer INNER JOIN (Component LEFT JOIN V24 V ON Component.ComponentID = V.ComponentID)

ON Manufacturer.ManufacturerID = Component.ManufacturerID

WHERE V.ComponentID Is Null

ORDER BY [EstimatedCost]\*[QuantityOnHand] DESC;

SELECT DISTINCT RetailStore.StoreName AS [Store Name Or Manufacturer Name], RetailStore.Phone

FROM City RIGHT JOIN (RetailStore INNER JOIN Bicycle ON RetailStore.StoreID = Bicycle.StoreID) ON City.CityID = RetailStore.CityID

WHERE City.State = 'ca' AND

Year(Bicycle.OrderDate) = 2004

UNION

SELECT Manufacturer.ManufacturerName AS [Store Name or Manufacturer Name], Manufacturer.Phone

FROM Manufacturer;

SELECT S.STORENAME, S.PHONE

FROM RetailStoRE S INNER JOIN BICYCLE B ON B.STOREID = S.STOREID

INNER JOIN CITY C ON C.CITYID = S.CityID

WHERE YEAR(B.ORDERDATE) = 2004 AND (C.STATE = 'CA' OR B.SALESTATE = 'CA')

GROUP BY S.StoreName, S.;

SELECT Employee\_1.LastName, Employee.EmployeeID, Employee.LastName, Employee.FirstName, Employee.Title

FROM Employee AS Employee\_1 INNER JOIN Employee ON Employee\_1.EmployeeID = Employee.CurrentManager

WHERE Employee\_1.LastName='Venetiaan'

Select (

Select LastName

From EMPLOYEE

Where EmployeeID = (

Select EmployeeID

From EMPLOYEE

Where LastName = 'Venetiaan'))

As LastName, EmployeeID, LastName, FirstName, Title

From EMPLOYEE

Where CurrentManager = (

Select EmployeeID

From EMPLOYEE

Where LastName = 'Venetiaan')

Select Year(OrderDate) As 'Year', Avg (DateDiff(Day, OrderDate, ShipDate)) As 'Build Time'

From BICYCLE

Group By Year(OrderDate)

Having Avg(DateDiff(Day, OrderDate, ShipDate)) >

(Select Avg(DateDiff(Day, OrderDate, ShipDate)) As 'Average'

From BICYCLE)

Order By Year(OrderDate)

SELECT DATEPART(YEAR, OrderDate) AS [Year], Avg(1.0\* DATEDIFF(DAY, OrderDate, ShipDate)) AS BuildTime

FROM Bicycle

GROUP BY DATEPART(YEAR, OrderDate)

HAVING Avg(1.0\* DATEDIFF(DAY, OrderDate, ShipDate))>

(SELECT Avg(1.0\* DATEDIFF(DAY, OrderDate, ShipDate))

FROM Bicycle)

ORDER BY Avg(1.0\* DATEDIFF(DAY, OrderDate, ShipDate)) DESC;

SELECT TOP 1 M.MANUFACTURERNAME, C.PRODUCTNUMBER, C.CATEGORY, SUM(C.LISTPRICE \* C.QUANTITYONHAND) AS AMT, BP.DATEINSTALLED

FROM MANUFACTURER M INNER JOIN COMPONENT C ON C.MANUFACTURERID = M.MANUFACTURERID

INNER JOIN BIKEPARTS BP ON C.COMPONENTID = BP.COMPONENTID

WHERE YEAR(BP.DATEINSTALLED) NOT LIKE 2004

GROUP BY C.PRODUCTNUMBER, C.CATEGORY, M.MANUFACTURERNAME, BP.DATEINSTALLED

ORDER BY SUM(C.LISTPRICE \* C.QUANTITYONHAND) ;

CREATE VIEW V24 AS

SELECT DISTINCT BikeParts.ComponentID

FROM BikeParts

WHERE Year(BikeParts.DateInstalled) = 2004

SELECT Manufacturer.ManufacturerName, Component.ProductNumber, Component.Category,

[EstimatedCost]\*[QuantityOnHand] AS [Value], V.ComponentID

FROM Manufacturer INNER JOIN (Component LEFT JOIN V24 V ON Component.ComponentID = V.ComponentID)

ON Manufacturer.ManufacturerID = Component.ManufacturerID

WHERE V.ComponentID Is Null

ORDER BY [EstimatedCost]\*[QuantityOnHand] DESC;

Database Code Snippet

SELECT \*

FROM KIMTAYPET..SALES\_REP;

CREATE TABLE SALES\_REP

(

REP\_ID CHAR(2) PRIMARY KEY NOT NULL ,

FIRST\_NAME VARCHAR(20),

LAST\_NAME VARCHAR(20),

ADDRESS VARCHAR(15),

CITY VARCHAR(15),

STATE CHAR(2),

POSTAL CHAR(5),

CELL\_PHONE CHAR(12),

COMMISSION DECIMAL(7,2),

RATE DECIMAL(3,2)

);

DROP TABLE SALES\_REP

SELECT \*

FROM SALES\_REP;

INSERT INTO SALES\_REP VALUES

('05', 'Susan', 'Garcia', '42 Mountain Ln', 'Cody', 'WY', '82414', '307-824-1245', 12743.16, 0.04);

INSERT INTO SALES\_REP VALUES

('10', 'Richard', 'Miller', '87 Pikes Dr', 'Ralston', 'WY', '82440', '307-406-4321', 20872.11, 0.06);

INSERT INTO SALES\_REP VALUES

('15', 'Donna', 'Smithr', '312 Oak Rd', 'Powell', 'WY', '82440', '307-982-8401', 14192.92, 0.04);

INSERT INTO SALES\_REP VALUES

('20', 'Donna', 'Jackson', '19 Lookout Dr', 'Elk Butte', 'WY', '82433', '307-883-9481', 0.00, 0.04);

INSERT INTO SALES\_REP

(

REP\_ID, FIRST\_NAME, LAST\_NAME, COMMISSION, RATE)

VALUES ('25', 'RACHEL', 'HALLS', 0,0.04);

UPDATE SALES\_REP

SET ADDRESS = '136 MAIN AVE', CITY = 'CHEYENNE', STATE ='WY', POSTAL= '82001'

WHERE REP\_ID = '25';

CREATE TABLE CUSTOMER

(

CUST\_ID CHAR(3) PRIMARY KEY,

REP\_ID CHAR(2) FOREIGN KEY REFERENCES SALES\_REP(REP\_ID)

);

SELECT \* FROM CUSTOMER;

ALTER TABLE CUSTOMER

ADD

FIRST\_NAME VARCHAR(20),

LAST\_NAME VARCHAR(20);