

For this assignment, write queries using SQL to acquire data about customers, vendors, products, and employees in a fictitious sales database. These queries will cover many of the core aspects of writing SQL to produce data for reporting and analyzing information. There may be multiple ways to produce the same results, but ensure you are returning the requested fields.

Using the Sales Orders database, complete the queries below.

**1. Show all the information on our customers.**

- a) Query: `SELECT * FROM Customers;`
- b) Columns: 9
- c) Expected Row Count: 28
- d) Screenshot:

CustomerID	CustFirstName	CustLastName	CustStreetAddress	CustCity	CustState	CustZipCode	CustAreaCode	CustPhoneNumber
1002	Vernon	Thompson	122 Spring River Drive	Juvall	WA	98019	420	555-2001
1003	Gary	Hallmark	Route 2, Box 203B	Auburn	WA	98002	253	555-2676
1004	Robert	Brown	672 Lamont Ave	Houston	TX	77201	713	555-2491
1005	Dean	McCrae	4110 Old Redmond Rd.	Redmond	WA	98052	425	555-2506
1006	John	Viescas	15127 NE 24th, #883	Redmond	WA	98052	425	555-2511
1007	Mariya	Sergienko	901 Pine Avenue	Portland	OR	97208	503	555-2526
1008	Neil	Patterson	233 West Valley Hwy	San Diego	CA	92199	619	555-2541
1009	Andrew	Cencini	507 - 20th Ave. E. Apt...	Seattle	WA	98105	206	555-2601
1010	Angel	Kennedy	667 Red River Road	Austin	TX	78710	512	555-2571
1011	Alaina	Hallmark	Route 2, Box 203B	Woodin...	WA	98072	425	555-2631
1012	Liz	Keyser	13920 S.E. 40th Street	Bellevue	WA	98006	425	555-2556
1013	Rachel	Patterson	2114 Longview Lane	San Diego	CA	92199	619	555-2546
1014	Sam	Abolrous	611 Alpine Drive	Palm Sp...	CA	92263	760	555-2611
1015	Darren	Gehring	2601 Seaview Lane	Chico	CA	95926	530	555-2616

Customers 1

Action Output: c

Time	Action	Response
51... 21:27:15	SELECT * FROM Customers LIMIT 0, 1000	28 row(s) returned

**2. Show a list of states, in reverse alphabetical order, where our vendors are located, and include the names of the vendor.**

- a) Query: `SELECT * FROM Vendors ORDER BY VendState DESC;`
- b) Columns: 9
- c) Expected Row Count: 11
- d) Screenshot:

VendorID	VendName	VendStreetAddress	VendCity	VendState	VendZipCode	VendPhoneNumber	VendFaxNumber	VendWebPage
1	Shinoman, Incorporated	3042 19th Avenue South	Bellevue	WA	98001	(425) 888-1234	(425) 888-1235	#http://www.shinoman.com
3	Nikoma of America	88 Old North Road Ave	Ballard	WA	91324	(206) 666-1234	(314) 666-1235	#http://www.nikomabikes.c
5	Kona, Incorporated	PO Box 10429	Redmond	WA	98052	(425) 333-1234	(425) 333-1235	#http://www.konabikes.con
9	Lone Star Bike Supply	7402 Kingman Drive	El Paso	TX	79915	(915) 666-9876	(915) 666-9877	null
10	Armadillo Brand	12330 Side Road Lane	Dallas	TX	75137	(214) 444-9876	(214) 444-9877	#http://www.DilloBikes.con
4	ProFormance	29 N. Quail St.	Albany	NY	12012	(518) 444-1234	(518) 444-1235	#http://www.ProFormBikes
7	Dog Ear	575 Madison Ave.	New York	NY	10003	(212) 888-9876	(212) 888-9877	null
2	Viscount	1911 Commerce Way	St. Louis	MO	63127	(314) 777-1234	(314) 777-1235	#http://www.viscountbikes.
11	Astro Paper Products	5639 N. Riverside	Chicago	IL	60637	(312) 555-9876	(312) 555-9875	null
8	Sun Sports Suppliers	PO Box 8082	Santa Monica	CA	91003	(310) 777-9876	(310) 777-9877	null
6	Big Sky Mountain Bikes	Glacier Bay South	Anchorage	AK	99209	(907) 222-1234	(907) 222-1235	null

Vendors 4

Action Output: c

Time	Action	Response	Duration / Fe
51... 21:33:16	SELECT * FROM Vendors ORDER BY VendState DESC LIMIT 0, 1000	11 row(s) returned	0.00063 sec

**3. What if we adjusted the retail price of each product by increasing it 7 percent?**

a) Query:

```
UPDATE Products SET RetailPrice = RetailPrice + (RetailPrice * 1.7);
SELECT * FROM Products;
```

b) Columns: 6

c) Expected Row Count: 40

d) Screenshot:

The screenshot shows a database interface with two main sections. The top section displays the 'Products' table with 14 rows of data. The columns are: ProductNumber, ProductName, ProductDescription, RetailPrice, QuantityOnHand, and CategoryID. The bottom section shows the 'Action Output' pane with a log of a single UPDATE query executed at 21:40:20. The log entry is: '5150 21:40:20 UPDATE Products SET RetailPrice = RetailPrice + (RetailPrice \* 1.7)'. It also indicates '40 row(s) affected, 4 warning(s): 1265 Data truncated...' and a duration of '0.0066 sec'.

ProductNumber	ProductName	ProductDescription	RetailPrice	QuantityOnHand	CategoryID
1	Trek 9000 Mountain Bike	HULL	3240.00	6	2
2	Eagle FS-3 Mountain Bike	HULL	4860.00	8	2
3	Dog Ear Cyclecomputer	HULL	202.50	20	1
4	Victoria Pro All Weather Tires	HULL	148.37	20	4
5	Dog Ear Helmet Mount Mirrors	HULL	20.12	12	1
6	Viscount Mountain Bike	HULL	1714.50	5	2
7	Viscount C-500 Wireless Bike Co...	HULL	132.30	30	1
8	Kryptonite Advanced 2000 U-Lock	HULL	135.00	20	1
9	Nikoma Lok-Tight U-Lock	HULL	89.10	12	1
10	Viscount Microshell Helmet	HULL	97.20	20	1
11	GT RTS-2 Mountain Bike	HULL	4455.00	5	2
12	Shinoman 105 SC Brakes	HULL	63.45	16	4
13	Shinoman Dura-Ace Headset	HULL	182.25	20	4
14	Eagle SA-120 Clipless Pedals	HULL	377.87	20	4

Action Output

Time	Action	Response	Duration / Fetch
5150 21:40:20	UPDATE Products SET RetailPrice = RetailPrice + (RetailPrice * 1.7)	40 row(s) affected, 4 warning(s): 1265 Data truncated...	0.0066 sec

**4. Show a list of orders made by each customer in ascending date order.**

a) Query:

```
SELECT Customers.CustomerID, Customers.CustFirstName,
Orders.OrderNumber, Orders.OrderDate FROM Customers, Orders
ORDER BY OrderDate ASC;
```

b) Columns: 4

c) Expected Row Count: 1000

d) Screenshot:

The screenshot shows a database interface with two main sections. The top section displays the results of a query with 16 rows of data. The columns are: CustomerID, CustFirstName, OrderNumber, and OrderDate. The bottom section shows the 'Action Output' pane with a log of a single SELECT query executed at 22:02:51. The log entry is: '5162 22:02:51 SELECT Customers.CustomerID, Customers.CustFirstNa...'. The 'Result 16' button is visible at the bottom of the result table.

CustomerID	CustFirstName	OrderNumber	OrderDate
1022	Caleb	36	2012-09-06
1021	Estella	36	2012-09-06
1020	Joyce	36	2012-09-06
1019	Zachary	36	2012-09-06
1018	David	36	2012-09-06
1017	Manuela	36	2012-09-06
1016	Jim	36	2012-09-06
1015	Darren	36	2012-09-06
1014	Sam	36	2012-09-06
1013	Rachel	36	2012-09-06
1012	Liz	36	2012-09-06
1011	Alaina	36	2012-09-06
1010	Angel	36	2012-09-06
1009	Andrew	36	2012-09-06

Action Output

Time	Action
5162 22:02:51	SELECT Customers.CustomerID, Customers.CustFirstNa...

5. Give the names of all vendors based in Albany, Anchorage, and Dallas.

a) Query:

```
SELECT * FROM Vendors WHERE VendCity = 'Albany' OR VendCity = 'Anchorage' OR VendCity = 'Dallas';
```

b) Columns: 9

c) Expected Row Count: 3

d) Screenshot:

6. Show an alphabetized list of products with a quantity on hand greater than or equal to 30.

a) Query:

```
SELECT * FROM Products WHERE QuantityOnHand >= 30 ORDER BY Productname ASC;
```

b) Columns: 6

c) Expected Row Count: 9

d) Screenshot:

**7. What vendors do we work with that don't have an email address?**

a) Query:

```
SELECT VendName, VendEMailAddress FROM Vendors WHERE VendEMailAddress IS NULL;
```

b) Columns: 2

c) Expected Row Count: 5

d) Screenshot:

VendName	VendEMailAddress
Big Sky Mountain Bikes	NULL
Dog Ear	NULL
Sun Sports Suppliers	NULL
Lone Star Bike Supply	NULL
Astro Paper Products	NULL

**8. List employees and the dates their orders shipped sorted by order date.**

a) Query:

```
SELECT Employees.EmployeeID, Employees.EmpFirstName,
Employees.EmpLastName, Orders.OrderNumber, Orders.OrderDate FROM
Employees, Orders ORDER BY OrderDate ASC ;
```

b) Columns: 5

c) Expected Row Count: 1000

d) Screenshot:

EmployeeID	EmpFirstName	EmpLastName	OrderNumber	OrderDate
708	Susan	McLain	1	2012-09-01
707	Kathryn	Patterson	1	2012-09-01
706	David	Viescas	1	2012-09-01
705	Kirk	DeGrasse	1	2012-09-01
704	Carol	Viescas	1	2012-09-01
703	Matt	Berg	1	2012-09-01
702	Mary	Thompson	1	2012-09-01
701	Ann	Patterson	1	2012-09-01
709	Daffy	Dumbwit	2	2012-09-01
708	Susan	McLain	2	2012-09-01
707	Kathryn	Patterson	2	2012-09-01
706	David	Viescas	2	2012-09-01
705	Kirk	DeGrasse	2	2012-09-01
704	Carol	Viescas	2	2012-09-01
703	Matt	Berg	2	2012-09-01

**9. Show the vendors and products they supply to us for products over \$75 for vendors in Texas.**

a) Query:

```
SELECT Vendors.VendorID, Vendors.VendorName, Vendors.VendState,  
Product_Vendors.WholesalePrice FROM Vendors, Product_Vendors  
WHERE VendState = 'TX' AND WholesalePrice >= 75;
```

b) Columns: 4

c) Expected Row Count: 54

d) Screenshot:

VendorID	VendorName	VendState	WholesalePrice
9	Lone Star Bike Supply	TX	804.00
10	Armadillo Brand	TX	854.22
9	Lone Star Bike Supply	TX	854.22
10	Armadillo Brand	TX	1269.00
9	Lone Star Bike Supply	TX	1269.00
10	Armadillo Brand	TX	1477.81
9	Lone Star Bike Supply	TX	1477.81
10	Armadillo Brand	TX	403.22
9	Lone Star Bike Supply	TX	403.22
10	Armadillo Brand	TX	448.73
9	Lone Star Bike Supply	TX	448.73
10	Armadillo Brand	TX	1076.62
9	Lone Star Bike Supply	TX	1076.62
10	Armadillo Brand	TX	1178.65
10	Lone Star Bike Supply	TX	1178.65

**10. Show employees who live in the same city and state as our vendors.**

a) Query:

b) Columns:

c) Expected Row Count:

d) Screenshot:

**11. Display customers who have no sales rep (employees) in the same state.**

a) Query:

b) Columns:

c) Expected Row Count:

d) Screenshot:

**12. What is the average quoted price of a helmet?**

a) Query:

```
SELECT AVG(RetailPrice) FROM Products WHERE ProductName LIKE  
'%Helmet%';
```

b) Columns: 1

c) Expected Row Count: 1

d) Screenshot:

AVG(RetailPrice)
207.530000

13.What was the date of the earliest ship date?

a) Query: SELECT \* FROM Orders ORDER BY ShipDate ASC LIMIT 1;

b) Columns: 6

c) Expected Row Count: 1

d) Screenshot:

OrderNumber	OrderDate	ShipDate	CustomerID	EmployeeID	OrderTotal
5	2012-09-01	2012-09-01	1024	708	5544.75

14.What is the total amount (in dollars) of orders from the state of Oregon?

a) Query:

```
SELECT SUM(QuotedPrice) FROM Order_Details WHERE OrderNumber = ANY  
(SELECT OrderNumber FROM Orders  
WHERE CustomerID =  
ANY (SELECT CustomerID FROM Customers WHERE CustState = 'OR'));
```

b) Columns: 1

c) Expected Row Count: 1

d) Screenshot:

SUM(QuotedPrice)
222492.58

**15. Show each employee, the employee's total sales (in dollars), the employee's total sales item quantity, and the average item sales price ordered by the employee's average item sales price highest to lowest.**

a) Query:

```
SELECT DISTINCT Employees.EmployeeID, Employees.EmpFirstName,
t.orderNumber, t.sales, t.average, t.summedQuantity FROM (SELECT
SUM(QuantityOrdered) AS summedQuantity, ROUND(AVG(ProductNumber)) as
average , orderNumber FROM order_details GROUP BY OrderNumber) t
INNER JOIN Employees ON(SELECT EmployeeID FROM Orders WHERE
OrderNumber = t.OrderNumber) = Employees.EmployeeID ORDER BY t.sales
DESC;
```

b) Columns: 6

c) Expected Row Count: 933

d) Screenshot:

EmployeeID	EmpFirstName	orderNumber	sales	average	summedQuantity
701	Ann	550	3893.99	19	29
702	Mary	752	3889.75	19	21
702	Mary	940	3885.19	19	31
703	Matt	434	3884.29	19	24
707	Kathryn	864	3883.30	19	25
707	Kathryn	615	3880.00	19	16
705	Kirk	797	3879.16	19	25
703	Matt	538	3878.61	21	26
703	Matt	636	3877.51	19	26
703	Matt	865	3877.25	19	26
701	Ann	373	3876.25	19	24
705	Kirk	242	3875.59	19	26
703	Matt	414	3873.93	21	34
706	David	363	3870.70	19	21
707	Kathryn	810	3868.11	17	21