

1. Show all the information on our customers.

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane shows a tree view with 'salesorders' selected. The main pane shows the 'SQL File 4*' window with the query `SELECT * from customers;`. Below the query, the 'Result Grid' shows a table with 12 rows of customer data. The 'Output' pane at the bottom shows a message: '28 row(s) returned'.

CustomerID	CustFirstName	CustLastName	CustStreetAddress	CustCity	CustState	CustZipCode	CustAreaCode	CustPhoneNumber
1001	Suzanne	Viescas	15127 NE 24th, #383	Redmond	WA	98052	425	555-2686
1002	William	Thompson	122 Spring River Drive	Duvall	WA	98019	425	555-2681
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, #383	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. Ap...	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	Woodinville	WA	98072	425	555-2631
1012	Liz	Keyser	13920 S.E. 40th Street	Bellevue	WA	98006	425	555-2556

Output: 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.

The screenshot displays the SQL Enterprise Manager interface. On the left, the 'SCHEMAS' tree shows the 'vendors' table with columns including 'VendState'. A blue arrow points to 'VendState'. Below the tree, the 'Column: VendState' details are shown: 'Collation: utf8_general_ci' and 'Definition: VendState varchar(2)'. The main window shows a SQL query in 'SQL File 4' with the following code:

```
1 SELECT * from customers;
2
3 -- 2. Show a list of states, in reverse alphabetical order,
4 -- where our vendors are located, and include the names of the vendor.
5
6 • SELECT VendState, VendName FROM vendors
7 ORDER BY VendState DESC;
8
```

A blue arrow points to the query text. Below the query, the 'Result Grid' shows the results of the query:

VendState	VendName
WA	Shinoman, Incorporated
WA	Nikoma of America
WA	Kona, Incorporated
TX	Lone Star Bike Supply
TX	Armadillo Brand
NY	ProFormance
NY	Dog Ear
MO	Viscount
IL	Astro Paper Products
CA	Sun Sports Suppliers
AK	Big Sky Mountain Bikes

At the bottom, the 'Output' window shows the execution log:

#	Time	Action	Message
1	11:47:09	SELECT * from customers	28 row(s) returned
2	11:50:30	SELECT * from customers	28 row(s) returned
3	11:50:30	SELECT VendState, VendName FROM vendors ORDER BY VendState DESC	11 row(s) returned

A blue arrow points to the third row in the output log.

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

The screenshot shows the MySQL Workbench interface with a SQL query executed. The query is as follows:

```
-- Question 3. What if we adjusted the retail price of each product by increasing it 7 percent?  
  
UPDATE products  
SET RetailPrice = RetailPrice + (RetailPrice * .07);  
  
SELECT * FROM products;
```

The **Result Grid** displays the following data:

ProductNumber	ProductName	ProductDescription	RetailPrice	QuantityOnHand	CategoryID
1	Trek 9000 Mountain Bike	HULL	1470.05	6	2
2	Eagle FS-3 Mountain Bike	HULL	2205.08	8	2
3	Dog Ear Cyclecomputer	HULL	91.88	20	1
4	Victoria Pro All Weather Tires	HULL	67.92	20	4
5	Dog Ear Helmet Mount Mirrors	HULL	9.13	12	1
6	Viscount Mountain Bike	HULL	777.90	5	2
7	Viscount C-500 Wireless Bike Computer	HULL	60.03	30	1
8	Kryptonite Advanced 2000 U-Lock	HULL	61.26	20	1
9	Nikoma Lok-Tight U-Lock	HULL	40.42	12	1
10	Viscount Microshell Helmet	HULL	44.11	20	1
11	GT RTS-2 Mountain Bike	HULL	2021.33	5	2
12	Shinoman 105 SC Brakes	HULL	28.79	16	4

The **Table: products** schema is shown in the bottom left:

Columns:

- ProductNumber: int AI PK
- ProductName: varchar(50)
- ProductDescription: varchar(100)
- RetailPrice: decimal(15,2)
- QuantityOnHand: smallint
- CategoryID: int

The **Output** pane shows the following actions:

#	Time	Action	Message
1	12:15:14	UPDATE products SET RetailPrice = RetailPrice + (RetailPrice * .07)	40 row(s) affected; 99 warning(s): 1265 Data truncated for column 'RetailPrice' at row 1 1265 Data trunc...
2	12:15:14	SELECT * FROM products	40 row(s) returned

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

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Schemas' pane shows the 'salesorders' schema with the 'orders' table selected. The 'Columns' pane for 'orders' lists 'OrderNumber', 'OrderDate', 'ShipDate', 'CustomerID', 'EmployeeID', and 'OrderTotal'. The 'Definition' pane shows 'OrderDate' as a 'date' type. The main window shows a SQL query in 'SQL File 7':

```
-- Question 4. 4. Show a list of orders made by each customer in ascending date order.

SELECT * FROM orders
ORDER BY OrderDate;
```

The 'Result Grid' shows the following data:

	OrderNumber	OrderDate	ShipDate	CustomerID	EmployeeID	OrderTotal
▶	1	2012-09-01	2012-09-04	1018	707	12751.85
	2	2012-09-01	2012-09-03	1001	703	816.00
	3	2012-09-01	2012-09-04	1002	707	11912.45
	4	2012-09-01	2012-09-03	1009	703	6601.73
	5	2012-09-01	2012-09-01	1024	708	5544.75
	6	2012-09-01	2012-09-05	1014	702	9820.29
	7	2012-09-01	2012-09-04	1001	708	467.85
	8	2012-09-01	2012-09-01	1003	703	1492.60
	9	2012-09-01	2012-09-04	1007	708	69.00
	10	2012-09-01	2012-09-04	1012	701	2607.00
	11	2012-09-02	2012-09-04	1020	706	11070.65
	12	2012-09-02	2012-09-05	1024	706	72.00
	13	2012-09-02	2012-09-02	1024	704	7545.00

The 'Output' pane shows the execution log:

#	Time	Action	Message
⚠ 1	12:15:14	UPDATE products SET RetailPrice = RetailPrice + (RetailPrice * .07)	40 row(s) affected, 39 warning(s): 1265 Da
✓ 2	12:15:14	SELECT * FROM products	40 row(s) returned
✓ 3	12:29:27	SELECT * FROM orders	944 row(s) returned
✓ 4	12:30:13	SELECT * FROM orders ORDER BY OrderDate	944 row(s) returned

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

The screenshot displays the SQL Enterprise Manager interface. On the left, the 'SCHEMAS' pane shows a tree view of databases including 'salesorders', with 'vendors' selected under 'Tables'. The 'Columns' pane for 'vendors' lists 'VendorID', 'VendName', 'VendStreetAddress', 'VendCity', 'VendState', and 'VendZipCode'. The 'Information' pane shows details for the 'VendCity' column: 'Column: VendCity', 'Collation: utf8_general_ci', and 'Definition: VendCity varchar(30)'. The main window shows a SQL query in 'SQL File 7*':

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

SELECT * FROM vendors
WHERE VendCity IN ('Albany', 'Anchorage', 'Dallas');
```

Below the query, the 'Result Grid' shows the results of the query:

	VendorID	VendName	VendStreetAddress	VendCity	VendState	VendZipCode	VendPhoneNumber	VendFaxNumber	VendWebPage	VendEmailAddress
▶	4	ProFormance	29 N. Quail St.	Albany	NY	12012	(518) 444-1234	(518) 444-1235	#http://www.ProFormBikes.com/#	Sales@ProFormBikes.com
	6	Big Sky Mountain Bikes	Glacier Bay South	Anchorage	AK	99209	(907) 222-1234	(907) 222-1235	NULL	NULL
	10	Armadillo Brand	12330 Side Road Lane	Dallas	TX	75137	(214) 444-9876	(214) 444-9877	#http://www.DilloBikes.com/#	BikeProducts@DilloBikes.com
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

At the bottom, the 'Output' pane shows the execution log:

#	Time	Action	Message
✓ 3	12:29:27	SELECT * FROM orders	944 row(s) returned
✓ 4	12:30:13	SELECT * FROM orders ORDER BY OrderDate	944 row(s) returned
✓ 5	16:19:34	SELECT * FROM vendors WHERE VendCity IN ('Albany', 'Anchorage', 'Dallas')	3 row(s) returned

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

The screenshot displays the SQL Enterprise Manager interface. On the left, the 'SCHEMAS' pane shows a tree view of the database structure, including tables like 'products' and columns like 'ProductNumber', 'ProductName', 'ProductDescription', 'RetailPrice', 'QuantityOnHand', and 'CategoryID'. The 'products' table is selected, and its columns are listed below it.

The main pane shows a SQL query in a file named 'SQL File 7'. The query is as follows:

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

SELECT * FROM products
WHERE QuantityOnHand >= 30
ORDER BY ProductName;
```

The 'Result Grid' shows the results of the query. The columns are ProductNumber, ProductName, ProductDescription, RetailPrice, QuantityOnHand, and CategoryID. The results are as follows:

ProductNumber	ProductName	ProductDescription	RetailPrice	QuantityOnHand	CategoryID
37	AeroFlo ATB Wheels	NULL	231.54	40	4
20	Dog Ear Monster Grip Gloves	NULL	18.37	30	1
25	King Cobra Helmet	NULL	170.28	30	1
16	ProFormance ATB All-Terrain Pedal	NULL	34.30	40	4
15	ProFormance Toe-Klips 2G	NULL	6.11	40	4
17	Shinoman Deluxe TX-30 Pedal	NULL	55.13	60	4
23	Ultra-Pro Rain Jacket	NULL	104.13	30	3
7	Viscount C-500 Wireless Bike Computer	NULL	60.03	30	1
33	Wonder Wool Cycle Socks	NULL	23.27	30	3
NULL	NULL	NULL	NULL	NULL	NULL

The 'Output' pane shows the execution of the query. The first three rows show the execution of the query, and the last row shows the results of the query.

#	Time	Action	Message
4	12:30:13	SELECT * FROM orders ORDER BY OrderDate	944 row(s) returned
5	16:19:34	SELECT * FROM vendors WHERE VendCity IN ('Albany', 'Anchorage', 'Dallas')	3 row(s) returned
6	16:27:34	SELECT * FROM products WHERE QuantityOnHand >= 30 ORDER BY ProductName	9 row(s) returned

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

SQL File 7

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

1
2
3 • SELECT * FROM vendors
4 WHERE VendEmailAddress IS NULL;
5
6
```

Result Grid

	VendorID	VendName	VendStreetAddress	VendCity	VendState	VendZipCode	VendPhoneNumber	VendFaxNumber	VendWebPage	VendEmailAddress
▶	6	Big Sky Mountain Bikes	Glacier Bay South	Anchorage	AK	99209	(907) 222-1234	(907) 222-1235	NULL	NULL
	7	Dog Ear	575 Madison Ave.	New York	NY	10003	(212) 888-9876	(212) 888-9877	NULL	NULL
	8	Sun Sports Suppliers	PO Box 8082	Santa Monica	CA	91003	(310) 777-9876	(310) 777-9877	NULL	NULL
	9	Lone Star Bike Supply	7402 Kingman Drive	El Paso	TX	79915	(915) 666-9876	(915) 666-9877	NULL	NULL
	11	Astro Paper Products	5639 N. Riverside	Chicago	IL	60637	(312) 555-9876	(312) 555-9875	NULL	NULL
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Column: VendEmailAddress
Collation: utf8_general_ci
Definition: VendEmailAddress varchar(50)

vendors 8

Output

#	Time	Action	Message
✓ 5	16:19:34	SELECT * FROM vendors WHERE VendCity IN ('Albany', 'Anchorage', 'Dallas')	3 row(s) returned
✓ 6	16:27:34	SELECT * FROM products WHERE QuantityOnHand >= 30 ORDER BY ProductName	9 row(s) returned
✓ 7	16:31:48	SELECT * FROM vendors WHERE VendEmailAddress IS NULL	5 row(s) returned

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

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane shows a tree view of the 'salesorders' database, with the 'orders' table and its 'ShipDate' column selected. The 'Column: ShipDate' definition is shown as 'date'. The main pane shows a SQL query in 'SQL File 7*' with the following text:

```
-- Question 8. List employees and the dates their orders shipped sorted by order date.

SELECT employees.EmpFirstName, employees.EmpLastName, orders.ShipDate FROM employees
INNER JOIN orders ON employees.EmployeeID = orders.EmployeeID
ORDER BY ShipDate;
```

Below the query, the 'Result Grid' shows the results of the query, sorted by 'ShipDate'.

EmpFirstName	EmpLastName	ShipDate
Susan	McLain	2012-09-03
Ann	Patterson	2012-09-04
Kirk	DeGrasse	2012-09-04
David	Viescas	2012-09-04
Kathryn	Patterson	2012-09-04
Kathryn	Patterson	2012-09-04
Susan	McLain	2012-09-04
Susan	McLain	2012-09-04
Susan	McLain	2012-09-04
Mary	Thompson	2012-09-05
Matt	Berg	2012-09-05
Kirk	DeGrasse	2012-09-05
David	Viescas	2012-09-05

At the bottom, the 'Output' pane shows the 'Action Output' for the query, indicating that 944 row(s) were returned.

#	Time	Action	Message
6	16:27:34	SELECT * FROM products WHERE QuantityOnHand >= 30 ORDER BY ProductName	9 row(s) returned
7	16:31:48	SELECT * FROM vendors WHERE VendEmailAddress IS NULL	5 row(s) returned
8	16:39:03	SELECT employees.EmpFirstName, employees.EmpLastName, orders.ShipDate FROM employees I...	944 row(s) returned

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

SQL File 7*

1 -- Question 8. List employees and the dates their orders shipped sorted by order date.
2
3 • SELECT employees.EmpFirstName, employees.EmpLastName, orders.ShipDate FROM employees
4 INNER JOIN orders ON employees.EmployeeID = orders.EmployeeID
5 ORDER BY ShipDate;
6
7

Result Grid

	EmpFirstName	EmpLastName	ShipDate
	Susan	McLain	2012-09-03
	Ann	Patterson	2012-09-04
	Kirk	DeGrasse	2012-09-04
	David	Viescas	2012-09-04
	Kathryn	Patterson	2012-09-04
	Kathryn	Patterson	2012-09-04
	Susan	McLain	2012-09-04
	Susan	McLain	2012-09-04
	Susan	McLain	2012-09-04
	Mary	Thompson	2012-09-05
	Matt	Berg	2012-09-05
	Kirk	DeGrasse	2012-09-05
	David	Viescas	2012-09-05

Result 9

Output

Action Output

#	Time	Action	Message
✓ 6	16:27:34	SELECT * FROM products WHERE QuantityOnHand >= 30 ORDER BY ProductName	9 row(s) returned
✓ 7	16:31:48	SELECT * FROM vendors WHERE VendEmailAddress IS NULL	5 row(s) returned
✓ 8	16:39:03	SELECT employees.EmpFirstName, employees.EmpLastName, orders.ShipDate FROM employees I...	944 row(s) returned

Column: ShipDate
Definition: ShipDate date

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

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'employees' and 'vendors'. The 'vendors' table is selected, and its schema is shown in the 'Table: vendors' section below. The main pane displays a SQL query in 'SQL File 7*' and its results in the 'Result Grid'.

Table: vendors

Columns:

- VendorID** int AI PK
- VendName varchar(25)
- VendStreetAddress varchar(50)
- VendCity varchar(30)
- VendState varchar(2)
- VendZipCode** varchar(10)
- VendPhoneNumber varchar(15)
- VendFaxNumber varchar(15)
- VendWebPage text
- VendEmailAddress varchar(50)

SQL Query:

```
-- Question 10. Show employees who live in the same city and state as our vendors.

SELECT * FROM employees INNER JOIN vendors ON
(employee.EmpCity = vendors.VendCity AND employees.EmpState = vendors.VendState);
```

Result Grid:

	EmployeeID	EmpFirstName	EmpLastName	EmpStreetAddress	EmpCity	EmpState	EmpZipCode	EmpAreaCode	EmpPhoneNumber	VendorID	VendName
▶	708	Susan	McLain	511 Lenora Ave	Bellevue	WA	98006	425	555-2301	1	Shinoman, Incorporated
	706	David	Viescas	16679 NE 42nd Court	Redmond	WA	98052	425	555-2661	5	Kona, Incorporated

Output:

Result 11 x

Output

Action Output

#	Time	Action	Message
✓ 8	16:39:03	SELECT employees.EmpFirstName, employees.EmpLastName, orders.ShipDate FROM employees I...	944 row(s) returned
✓ 9	16:45:10	SELECT vendors.VendName, vendors.VendState, products.ProductName, product_vendors.Whole...	12 row(s) returned
✓ 10	16:49:35	SELECT * FROM employees INNER JOIN vendors ON (employees.EmpCity = vendors.VendCity AN...	2 row(s) returned

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

The screenshot shows the SQL Developer interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'customers' and 'employees' under the 'salesorders' schema. The 'Columns' pane for the 'employees' table is visible, listing columns such as 'EmployeeID', 'EmpFirstName', 'EmpLastName', 'EmpStreetAddress', 'EmpCity', 'EmpState', 'EmpZipCode', 'EmpAreaCode', and 'EmpPhoneNumber'. The main editor window shows a SQL query for 'Question 11'.

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

SELECT * FROM customers
LEFT JOIN employees ON customers.CustState = employees.EmpState
WHERE EmployeeID IS NULL;
```

Below the query editor, the 'Result Grid' displays the query results. The grid has 12 columns: CustomerID, CustFirstName, CustLastName, CustStreetAddress, CustCity, CustState, CustZipCode, CustAreaCode, CustPhoneNumber, EmployeeID, and EmpFir. The results show 12 rows of customer data, all with NULL values in the EmployeeID and EmpFir columns, indicating no sales rep is assigned to these customers.

CustomerID	CustFirstName	CustLastName	CustStreetAddress	CustCity	CustState	CustZipCode	CustAreaCode	CustPhoneNumber	EmployeeID	EmpFir
1007	Mariya	Sergienko	901 Pine Avenue	Portland	OR	97208	503	555-2526	NULL	NULL
1008	Neil	Patterson	233 West Valley Hwy	San Diego	CA	92199	619	555-2541	NULL	NULL
1013	Rachel	Patterson	2114 Longview Lane	San Diego	CA	92199	619	555-2546	NULL	NULL
1014	Sam	Abolrous	611 Alpine Drive	Palm Springs	CA	92263	760	555-2611	NULL	NULL
1015	Darren	Gehring	2601 Seaview Lane	Chico	CA	95926	530	555-2616	NULL	NULL
1016	Jim	Wilson	101 NE 88th	Salem	OR	97301	503	555-2636	NULL	NULL
1017	Manuela	Seidel	66 Spring Valley Drive	Medford	OR	97501	541	555-2641	NULL	NULL
1018	David	Smith	311 20th Ave. N.E.	Fremont	CA	94538	510	555-2646	NULL	NULL
1019	Zachary	Ehrlich	12330 Kingman Drive	Glendale	CA	91209	818	555-2721	NULL	NULL
1022	Caleb	Viescas	4501 Wetland Road	Long Beach	CA	90809	562	555-0037	NULL	NULL
1027	Luke	Patterson	877 145th Ave SE	Portland	OR	97208	503	555-2316	NULL	NULL

At the bottom, the 'Output' pane shows the execution log. It includes the following entries:

- 9 16:45:10 SELECT vendors.VendName, vendors.VendState, products.ProductName, product_vendors.Whole... 12 row(s) returned
- 10 16:49:35 SELECT * FROM employees INNER JOIN vendors ON (employees.EmpCity = vendors.VendCity AN... 2 row(s) returned
- 11 16:52:01 SELECT * FROM customers LEFT JOIN employees ON customers.CustState = employees.EmpStat... 11 row(s) returned

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

The screenshot displays the SQL Enterprise Manager interface. On the left, the 'SCHEMAS' pane shows a tree view of databases including 'avengers', 'hr', 'sakila', and 'salesorders'. Under 'salesorders', the 'products' table is selected, showing its columns: ProductNumber, ProductName, ProductDescription, RetailPrice, QuantityOnHand, and CategoryID. The 'Information' pane at the bottom left shows the definition of the 'RetailPrice' column as 'decimal(15,2)'. The main 'SQL File 7' pane contains the following query:

```
-- Question 12. What is the average quoted price of a helmet?

SELECT ROUND(avg(RetailPrice), 0) FROM products
WHERE ProductName LIKE '%Helmet%';
```

Below the query, the 'Result Grid' shows the result of the query:

	ROUND(avg(RetailPrice), 0)
▶	94

The 'Output' pane at the bottom shows the 'Action Output' for the query execution:

#	Time	Action	Message
✓ 13	16:55:05	SELECT ROUND(avg(RetailPrice), 0) FROM products WHERE ProductName LIKE "%Helmet%"	1 row(s) returned
✓ 14	16:55:36	SELECT ROUND(avg(RetailPrice), -1) FROM products WHERE ProductName LIKE "%Helmet%"	1 row(s) returned
✓ 15	16:55:49	SELECT ROUND(avg(RetailPrice), 0) FROM products WHERE ProductName LIKE "%Helmet%"	1 row(s) returned

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

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure. The 'salesorders' schema is expanded, showing tables like 'categories', 'customers', 'employees', 'order_details', 'order_details_archive', and 'orders'. The 'orders' table is selected, and its columns are listed: OrderNumber, OrderDate, ShipDate, CustomerID, EmployeeID, and OrderTotal. The 'Information' pane at the bottom left shows the definition of the 'ShipDate' column as 'date'.

The main query editor window, titled 'SQL File 7*', contains the following SQL query:

```
-- Question 13. What was the date of the earliest ship date?  
  
SELECT ShipDate FROM orders  
ORDER BY ShipDate LIMIT 1;
```

The 'Result Grid' pane shows the result of the query:

ShipDate
2012-09-01

The 'Output' pane at the bottom shows the execution log with three entries:

#	Time	Action	Message
✓ 14	16:55:36	SELECT ROUND(avg(RetailPrice), -1) FROM products WHERE ProductName LIKE "%Helmet%"	1 row(s) returned
✓ 15	16:55:49	SELECT ROUND(avg(RetailPrice), 0) FROM products WHERE ProductName LIKE "%Helmet%"	1 row(s) returned
✓ 16	19:36:43	SELECT ShipDate FROM orders ORDER BY ShipDate LIMIT 1	1 row(s) returned

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

The screenshot shows the SQL Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with the 'customers' table selected under the 'salesorders' schema. The right pane shows a query window with the following SQL code:

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

SELECT round(sum(OrderTotal),0) 'Sum' FROM orders
INNER JOIN customers ON orders.CustomerID = customers.CustomerID
WHERE customers.CustState LIKE 'OR%';
```

Below the query window, the 'Result Grid' shows the following data:

	Sum
	787101

The bottom pane shows the 'Table: customers' structure:

Columns:

Column Name	Data Type
CustomerID	int AI PK
CustFirstName	varchar(25)
CustLastName	varchar(25)
CustStreetAddress	varchar(50)
CustCity	varchar(30)
CustState	varchar(2)
CustZipCode	varchar(10)
CustAreaCode	smallint
CustPhoneNumber	varchar(8)

The bottom right pane shows the 'Output' window with the following results:

#	Time	Action	Message
15	16:55:49	SELECT ROUND(avg(RetailPrice), 0) FROM products WHERE ProductName LIKE "%Helmet%"	1 row(s) returned
16	19:36:43	SELECT ShipDate FROM orders ORDER BY ShipDate LIMIT 1	1 row(s) returned
17	19:40:14	SELECT round(sum(OrderTotal),0) 'Sum' FROM orders INNER JOIN customers ON orders.CustomerID = customers.CustomerID WHERE customers.CustState LIKE 'OR%';	1 row(s) returned

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.

SQL File 7*

Don't Limit

1

-- Question 15. Show each employee, the employee's total sales (in dollars), the employee's total sales item quantity,

2

-- and the average item sales price ordered by the employee's average item sales price highest to lowest.

3

4

• SELECT EmpFirstName, employees.EmployeeID, sum(QuantityOrdered),

5

SUM(OrderTotal), -- this takes the employees info and QuantityOrdered together

6

7

AVG (RetailPrice) FROM employees INNER JOIN orders ON

8

orders.EmployeeID = employees.EmployeeID INNER JOIN order_details ON

9

orders.OrderNumber = order_details.OrderNumber INNER JOIN products ON

10

products.ProductNumber = order_details.ProductNumber GROUP BY EmpFirstName;

11

12

-- then take the average from retail price from products and compared to orders by employee id then take the number of orders /

13

-- products then group by employee name had to look up for some information. very challenging

14

-- Week4 Worksheet Rafael Fernandez FSW140

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	EmpFirstName	EmployeeID	sum(QuantityOrdered)	SUM(OrderTotal)	AVG (RetailPrice)
▶	Ann	701	1559	2727482.60	409.928591
	Mary	702	1862	3370143.46	380.453491
	Matt	703	1488	2498821.88	382.286226
	Carol	704	1734	3174269.73	429.496453
	Kirk	705	1692	3099340.14	397.881470
	David	706	1575	2801948.87	388.464013
	Kathryn	707	2139	4309979.83	426.889562
	Susan	708	1951	3649014.58	431.843603

Result 19

Output

Action Output

#	Time	Action	Message
✓	16 19:36:43	SELECT ShipDate FROM orders ORDER BY ShipDate LIMIT 1	1 row(s) returned
✓	17 19:40:14	SELECT round(sum(OrderTotal),0) 'Sum' FROM orders INNER JOIN customers ON orders.CustomerI...	1 row(s) returned
✓	18 19:59:11	SELECT EmpFirstName, employees.EmployeeID, sum(QuantityOrdered), SUM(OrderTotal), AVG (Re...	8 row(s) returned

Navigation icons

SQL File 7*

Don't Limit

SQL File 7*

Navigation icons

SCHEMAS

Filter objects

order_details

order_details_archive

orders

Columns

OrderNumber

OrderDate

ShipDate

CustomerID

EmployeeID

OrderTotal

Indexes

Foreign Keys

Triggers

orders_archive

product_vendors

products

Columns

ProductNumber

ProductName

ProductDescription

RetailPrice

QuantityOnHand

Administration

Schemas

Information

Table: products

Columns:

ProductNumber

int AI PK

ProductName

varchar(50)

ProductDescription

varchar(100)

RetailPrice

decimal(15,2)

QuantityOnHand

smallint

CategoryID

int