Lab 04 – Multi-Table Queries and Views

Objective:

The purpose of this lab is to introduce students to querying data from multiple tables. Relationships are used in relational databases to reduce redundant and repetitive data, but it is necessary to reconnect these tables when extracting data and obtaining information. Student will be able to:

- produce query results containing data from multiple tables using ANSI-92 joins and demonstrate their knowledge of inner, outer and full joins.
- To actively troubleshoot queries to handle potentially ambiguous fields across multiple tables through the use of aliases
- Students learn to create and modify views.

Submission:

Your submission will be a single WORD file with the solutions provided.

Your submission needs to follow the same question order and clearly indicate the answers to each question. Make sure every SQL statement terminates with a semicolon.

ALL questions must be answered using ANSI-92 JOINs unless otherwise stated. ANSI-89 are obsolete and should not be used in new query derivations. We only teach them in case you see them in the workplace, that you know what they are and how they work.

Tasks:

Select data from multiple tables

1. Create a query that shows retail customers first name and last name along with their sales rep employee number and their first name, last name, city, phone numberand postal code for all retail customers who live in Singapore.
    a. Answer this question using an ANSI-89 Join
    b. Answer this question using an ANSI-92 Join

```
SELECT retailcustomers.contactfirstname AS CUST_FNAME ,
retailcustomers.contactlastname AS CUST_LNAME,
retailcustomers.salesrepemployeenumber AS SALESREP_NUMBER,
retailemployees.firstname AS SALESREP_FNAME,retailemployees.lastname AS
SALESREP_LNAME ,
retailcustomers.city AS CUST_CITY,retailcustomers.phone AS
CUST_PHONE,retailcustomers.postalcode AS CUST_POSTCODE
FROM retailemployees , retailcustomers
WHERE retailemployees.employeenumber = retailcustomers.salesrepemployeenumber
AND UPPER(retailcustomers.country) = 'SINGAPORE' ;
-- 2 rows selected.
```

Query Result

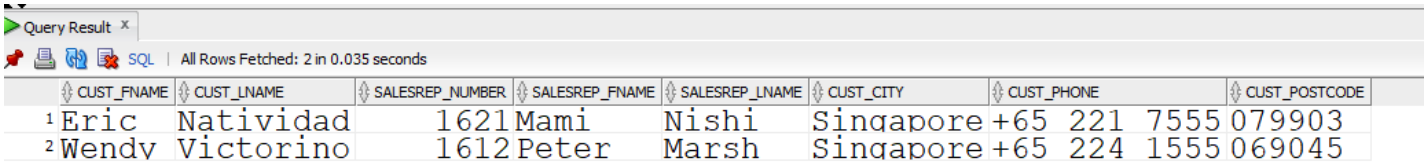
SQL | All Rows Fetched: 2 in 0.035 seconds

	CUST_FNAME	CUST_LNAME	SALESREP_NUMBER	SALESREP_FNAME	SALESREP_LNAME	CUST_CITY	CUST_PHONE	CUST_POSTCODE
1	Eric	Natividad	1621	Mami	Nishi	Singapore	+65 221 7555	079903
2	Wendy	Victorino	1612	Peter	Marsh	Singapore	+65 224 1555	069045

- Answer this question using an ANSI-92 Join

```
SELECT retailcustomers.contactfirstname AS CUST_FNAME ,
retailcustomers.contactlastname AS CUST_LNAME,
retailcustomers.salesrepemployeenumber AS SALESREP_NUMBER,
```

```
retailemployees.firstname AS SALESREP_FNAME,retailemployees.lastname AS
SALESREP_LNAME ,
retailcustomers.city AS CUST_CITY,retailcustomers.phone AS
CUST_PHONE,retailcustomers.postalcode AS CUST_POSTCODE
FROM retailemployees INNER JOIN  retailcustomers
 ON retailemployees.employeenumber =retailcustomers.salesrepemployeenumber
WHERE UPPER(retailcustomers.country) = 'SINGAPORE' ;
--2 rows selected.
```

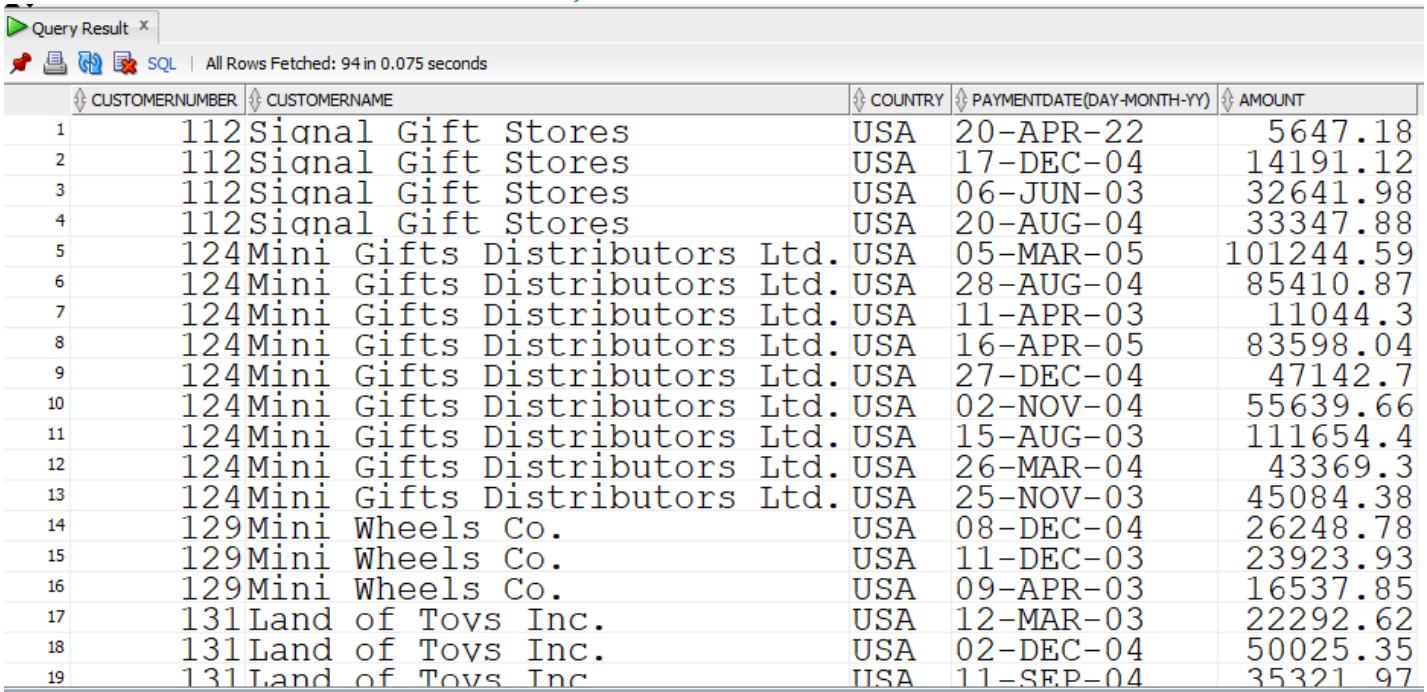


Query Result | SQL | All Rows Fetched: 2 in 0.035 seconds

	CUST_FNAME	CUST_LNAME	SALESREP_NUMBER	SALESREP_FNAME	SALESREP_LNAME	CUST_CITY	CUST_PHONE	CUST_POSTCODE
1	Eric	Natividad	1621	Mami	Nishi	Singapore	+65 221 7555	079903
2	Wendy	Victorino	1612	Peter	Marsh	Singapore	+65 224 1555	069045

2. Create a query that displays all retail payments made by retail customers from USA.
    a. Sort the output by Customer Number.
    b. Only display the Customer Number, Customer Name,Country, Payment Date and Amount.
    c. Make sure the date is displayed clearly to know what date it is. (i.e. what date is02-04-19??? – Feb 4, 2019, April 2, 2019, April 19, 2002, ….)

```
--Default day-month-year selection

SELECT customernumber, customername, country, paymentdate AS
"PAYMENTDATE(DAY-MONTH-YY)",amount
FROM retailcustomers JOIN retailpayments USING (customernumber)
WHERE UPPER(retailcustomers.country) LIKE 'USA'
ORDER BY customernumber;
```



Query Result | SQL | All Rows Fetched: 94 in 0.075 seconds

	CUSTOMERNUMBER	CUSTOMERNAME	COUNTRY	PAYMENTDATE(DAY-MONTH-YY)	AMOUNT
1	112	Signal Gift Stores	USA	20-APR-22	5647.18
2	112	Signal Gift Stores	USA	17-DEC-04	14191.12
3	112	Signal Gift Stores	USA	06-JUN-03	32641.98
4	112	Signal Gift Stores	USA	20-AUG-04	33347.88
5	124	Mini Gifts Distributors Ltd.	USA	05-MAR-05	101244.59
6	124	Mini Gifts Distributors Ltd.	USA	28-AUG-04	85410.87
7	124	Mini Gifts Distributors Ltd.	USA	11-APR-03	11044.3
8	124	Mini Gifts Distributors Ltd.	USA	16-APR-05	83598.04
9	124	Mini Gifts Distributors Ltd.	USA	27-DEC-04	47142.7
10	124	Mini Gifts Distributors Ltd.	USA	02-NOV-04	55639.66
11	124	Mini Gifts Distributors Ltd.	USA	15-AUG-03	111654.4
12	124	Mini Gifts Distributors Ltd.	USA	26-MAR-04	43369.3
13	124	Mini Gifts Distributors Ltd.	USA	25-NOV-03	45084.38
14	129	Mini Wheels Co.	USA	08-DEC-04	26248.78
15	129	Mini Wheels Co.	USA	11-DEC-03	23923.93
16	129	Mini Wheels Co.	USA	09-APR-03	16537.85
17	131	Land of Toys Inc.	USA	12-MAR-03	22292.62
18	131	Land of Toys Inc.	USA	02-DEC-04	50025.35
19	131	Land of Toys Inc.	USA	11-SEP-04	35321.97

```
--Display in Mon Day Year format

SELECT customernumber, customername, country, TO_CHAR(paymentdate,'MON
```

```
DD, YYYY') AS "PAYMENTDATE('MON DD, YYYY')",amount
FROM retailcustomers JOIN retailpayments USING (customernumber)
WHERE UPPER(retailcustomers.country) LIKE 'USA'
ORDER BY customernumber;
```

	CUSTOMERNUMBER	CUSTOMERNAME	COUNTRY	PAYMENTDATE('MON DD, YYYY')	AMOUNT
1	112	Signal Gift Stores	USA	APR 20, 2022	5647.18
2	112	Signal Gift Stores	USA	DEC 17, 2004	14191.12
3	112	Signal Gift Stores	USA	JUN 06, 2003	32641.98
4	112	Signal Gift Stores	USA	AUG 20, 2004	33347.88
5	124	Mini Gifts Distributors Ltd.	USA	MAR 05, 2005	101244.59
6	124	Mini Gifts Distributors Ltd.	USA	AUG 28, 2004	85410.87
7	124	Mini Gifts Distributors Ltd.	USA	APR 11, 2003	11044.3
8	124	Mini Gifts Distributors Ltd.	USA	APR 16, 2005	83598.04
9	124	Mini Gifts Distributors Ltd.	USA	DEC 27, 2004	47142.7
10	124	Mini Gifts Distributors Ltd.	USA	NOV 02, 2004	55639.66
11	124	Mini Gifts Distributors Ltd.	USA	AUG 15, 2003	111654.4
12	124	Mini Gifts Distributors Ltd.	USA	MAR 26, 2004	43369.3
13	124	Mini Gifts Distributors Ltd.	USA	NOV 25, 2003	45084.38
14	129	Mini Wheels Co.	USA	DEC 08, 2004	26248.78
15	129	Mini Wheels Co.	USA	DEC 11, 2003	23923.93
16	129	Mini Wheels Co.	USA	APR 09, 2003	16537.85
17	131	Land of Toys Inc.	USA	MAR 12, 2003	22292.62
18	131	Land of Toys Inc.	USA	DEC 02, 2004	50025.35
19	131	Land of Toys Inc.	USA	SEP 11, 2004	35321.97

3. Create a query that shows all Canada customers who have not made a payment. Display only the customer number ,customer name, amount sorted by customer number.

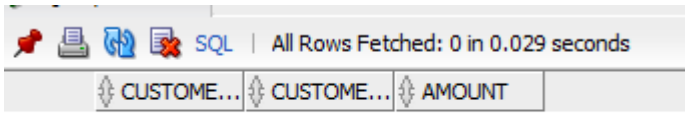
All Canadian customers are

```
SELECT retailcustomers.customernumber, retailcustomers.customername,AMOUNT
FROM retailcustomers LEFT OUTER JOIN retailpayments
ON retailcustomers.customernumber = retailpayments.customernumber
WHERE UPPER(retailcustomers.country) LIKE 'CANADA'
```

	CUSTOMERNUMBER	CUSTOMERNAME	AMOUNT
1	202	Canadian Gift Exchange Network	36527.61
2	202	Canadian Gift Exchange Network	33594.58
3	233	Québec Home Shopping Network	29070.38
4	233	Québec Home Shopping Network	22997.45
5	233	Québec Home Shopping Network	16909.84
6	260	Royal Canadian Collectables, Ltd.	37527.58
7	260	Royal Canadian Collectables, Ltd.	29284.42

All Canadian customers who have not made a payment are

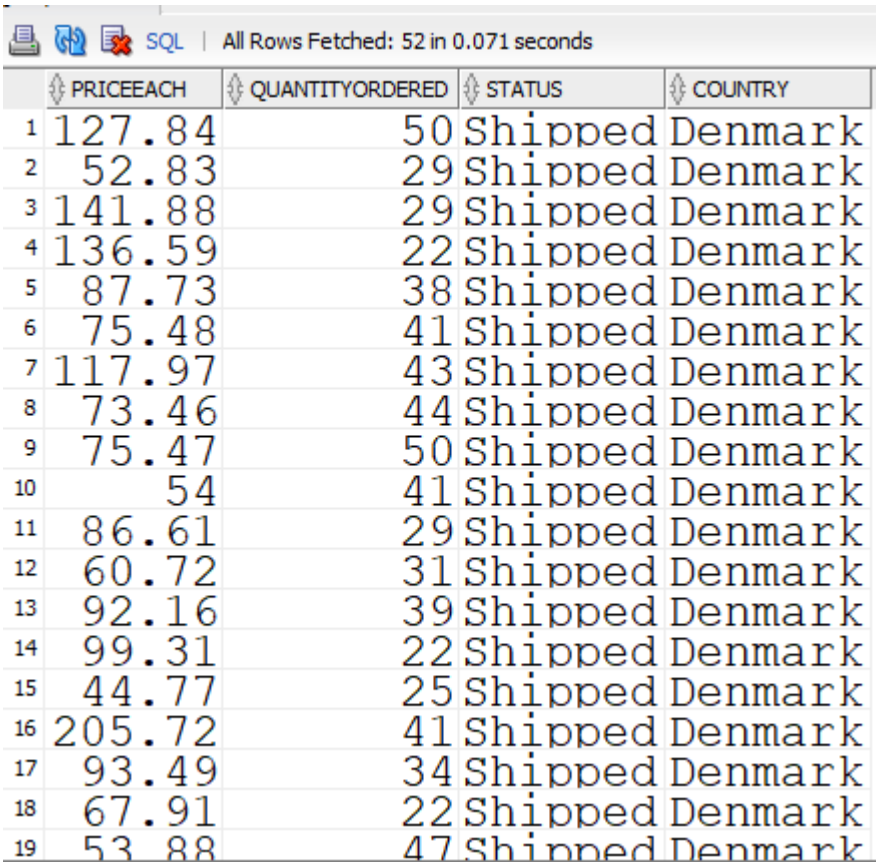
```
SELECT retailcustomers.customernumber, retailcustomers.customername,AMOUNT
FROM retailcustomers LEFT OUTER JOIN retailpayments
ON retailcustomers.customernumber = retailpayments.customernumber
WHERE UPPER(retailcustomers.country) LIKE 'CANADA' AND retailpayments.AMOUNT=0
ORDER BY customernumber;
```



Views and Joins

4. Display all the retail orders with quantity ordered, price of each item, who have their order shipped and who live in Denmark

```
SELECT PRICEEACH,QUANTITYORDERED,STATUS,COUNTRY
FROM retailorders INNER JOIN orderdetails
ON retailorders.ordernumber = orderdetails.ordernumber
INNER JOIN retailcustomers ON retailcustomers.customernumber  =
retailorders.customernumber
WHERE UPPER(retailorders.STATUS) LIKE 'SHIPPED' AND UPPER(retailcustomers.country)
LIKE'DENMARK'
ORDER BY retailcustomers.customernumber;
```



SQL | All Rows Fetched: 52 in 0.071 seconds

	PRICEEACH	QUANTITYORDERED	STATUS	COUNTRY
1	127.84	50	Shipped	Denmark
2	52.83	29	Shipped	Denmark
3	141.88	29	Shipped	Denmark
4	136.59	22	Shipped	Denmark
5	87.73	38	Shipped	Denmark
6	75.48	41	Shipped	Denmark
7	117.97	43	Shipped	Denmark
8	73.46	44	Shipped	Denmark
9	75.47	50	Shipped	Denmark
10	54	41	Shipped	Denmark
11	86.61	29	Shipped	Denmark
12	60.72	31	Shipped	Denmark
13	92.16	39	Shipped	Denmark
14	99.31	22	Shipped	Denmark
15	44.77	25	Shipped	Denmark
16	205.72	41	Shipped	Denmark
17	93.49	34	Shipped	Denmark
18	67.91	22	Shipped	Denmark
19	53.88	47	Shipped	Denmark

5. Create a view (***vwProductOrder***) to list all the retail products with the following data:

a) Product code, product name, msrp, buyprice, quantity ordered, and price for each product in every order.

b) Write a statement to view the results of the view just created.

```
CREATE or replace VIEW vwProductOrder AS
SELECT productcode,  productname, msrp, buyprice, quantityordered, priceeach
FROM retailorders JOIN orderdetails USING (ordernumber) JOIN retailproducts USING
(productcode);

SELECT * FROM vwProductOrder
```

SQL | Fetched 1,100 rows in 0.544 seconds

	PRODUCTCODE	PRODUCTNAME	MSRP	BUYPRICE	QUANTITYORDERED	PRICEEACH
1	S18 4721	1957 Corvette Convertible	148.8	69.93	44	147.31
2	S24 1578	1997 BMW R 1100 S	112.7	60.86	48	98.05
3	S24 2000	1960 BSA Gold Star DBD34	76.17	37.32	28	61.7
4	S24 2360	1982 Ducati 900 Monster	69.26	47.1	35	60.95
5	S24 4620	1961 Chevrolet Impala	80.84	32.33	28	68.71
6	S32 2206	1982 Ducati 996 R	40.23	24.14	34	33.39
7	S32 4485	1974 Ducati 350 Mk3 Desmo	102.05	56.13	22	102.05
8	S50 4713	2002 Yamaha YZR M1	81.36	34.17	21	74.85
9	S12 1099	1968 Ford Mustang	194.57	95.34	27	155.66
10	S12 3380	1968 Dodge Charger	117.44	75.16	28	113.92
11	S12 3990	1970 Plymouth Hemi Cuda	79.8	31.92	20	67.03
12	S12 4675	1969 Dodge Charger	115.16	58.73	36	107.1
13	S18 1129	1993 Mazda RX-7	141.54	83.51	44	124.56
14	S18 1589	1965 Aston Martin DB5	124.44	65.96	42	124.44
15	S18 1889	1948 Porsche 356-A Roadster	77	53.9	22	74.69
16	S18 1984	1995 Honda Civic	142.25	93.89	21	129.45
17	S18 2870	1999 Indy 500 Monte Carlo SS	132	56.76	27	130.68
18	S18 3232	1992 Ferrari 360 Spider red	169.34	77.9	45	147.33
19	S18 3278	1969 Dodge Super Bee	80.41	49.05	30	73.17

6. Using the ***vwProductOrder*** view, display the product order information with product name, buyprice ,order line number and whose buy price is in the range from $30 to $40 and whose product code starts with ‘s32’. Sort the output based on product name and then buy price. (Hint: orderLineNumber is not in the view then how can you get in this query?)

```
SELECT
vwProductOrder.productcode, vwProductOrder.productname,
vwProductOrder.buyprice,orderdetails.orderlinenumber
FROM vwProductOrder JOIN orderdetails
ON vwProductOrder.productcode=orderdetails.productcode
WHERE vwProductOrder.buyprice between 30 and 40 and
lower(vwProductOrder.productcode) like 's32%'
ORDER BY vwProductOrder.productname;
```

	PRODUCTCODE	PRODUCTNAME	BUYPRICE	ORDERLINENUMBER
1	S32 4289	1928 Ford Phaeton Deluxe	33.02	7
2	S32 4289	1928 Ford Phaeton Deluxe	33.02	7
3	S32 4289	1928 Ford Phaeton Deluxe	33.02	7
4	S32 4289	1928 Ford Phaeton Deluxe	33.02	7
5	S32 4289	1928 Ford Phaeton Deluxe	33.02	7
6	S32 4289	1928 Ford Phaeton Deluxe	33.02	7
7	S32 4289	1928 Ford Phaeton Deluxe	33.02	7
8	S32 4289	1928 Ford Phaeton Deluxe	33.02	7
9	S32 4289	1928 Ford Phaeton Deluxe	33.02	7
10	S32 4289	1928 Ford Phaeton Deluxe	33.02	7
11	S32 4289	1928 Ford Phaeton Deluxe	33.02	7
12	S32 4289	1928 Ford Phaeton Deluxe	33.02	7
13	S32 4289	1928 Ford Phaeton Deluxe	33.02	7

7. Create a query that displays the retail customer order information with customer number, first name, last name, phone, and credit limits for all retail customers who do not have any orders.

```
SELECT customernumber, contactfirstname, contactlastname, phone, creditlimit
FROM retailcustomers LEFT OUTER JOIN retailorders USING (customernumber)
WHERE ordernumber IS NULL;
```

SQL | All Rows Fetched: 24 in 0.137 seconds

	CUSTOMERNUMBER	CONTACTFIRSTNAME	CONTACTLASTNAME	PHONE	CREDITLIMIT
1	293	Ed	Harrison	+41 26 425 50 01	0
2	335	Philip	Cramer	0555-09555	0
3	125	Zbyszek	Piestrzeniewicz	(26) 642-7555	0
4	480	Alexander	Semenov	+7 812 293 0521	0
5	303	Bradley	Schuyler	+31 20 491 9555	0
6	409	Rita	Müller	0711-555361	0
7	369	Lino	Rodriguez	(1) 354-2555	0
8	459	Sven	Ottlieb	0241-039123	0
9	169	Isabel	de Castro	(1) 356-5555	0
10	273	Peter	Franken	089-0877555	0
11	168	Keith	Franco	2035557845	0
12	443	Alexander	Feuer	0342-555176	0
13	348	Patricia	McKenna	2967 555	0
14	361	Karin	Josephs	0251-555259	0
15	465	Carmen	Anton	+34 913 728555	0

8. Create a view (***vwEmployeeManager***) to display the information of all retail employees first name and last name and their managers first name and managers last name if there is any manager that the employee reports to. Include all employees, including those who do not report to anyone.

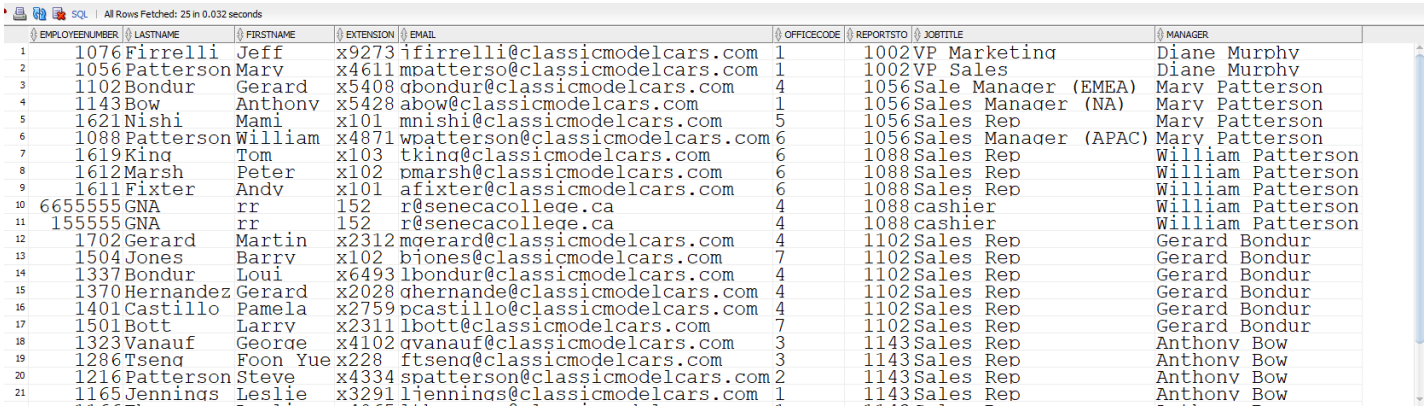
```
CREATE or REPLACE VIEW vwEmployeeManager AS
SELECT e.employeenumber,e.lastname,e.firstname,e.extension,e.email,e.officecode,
e.reportsto,e.jobtitle, m.firstname ||' ' || m.lastname AS manager
FROM retailemployees e FULL JOIN retailemployees m ON m.employeenumber = e.reportsto

SELECT * FROM vwEmployeeManager ORDER BY REPORTSTO;
```

SQL | All Rows Fetched: 45 in 0.031 seconds

	EMPLOYEENUMBER	LASTNAME	FIRSTNAME	EXTENSION	EMAIL	OFFICECODE	REPORTSTO	JOBTITLE	MANAGER
1	1056	Patterson	Mary	x4611	mpatterso@classicmodelcars.com	1	1002	VP Sales	Diane Murphy
2	1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com	1	1002	VP Marketing	Diane Murphy
3	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)	Mary Patterson
4	1102	Bondur	Gerard	x5408	gbondur@classicmodelcars.com	4	1056	Sale Manager (EMEA)	Mary Patterson
5	1088	Patterson	William	x4871	wpatterson@classicmodelcars.com	6	1056	Sales Manager (APAC)	Mary Patterson
6	1621	Nishi	Mami	x101	mnishi@classicmodelcars.com	5	1056	Sales Rep	Mary Patterson
7	6655555	GNA	rr	152	r@senecacollege.ca	4	1088	cashier	William Patterson
8	1619	King	Tom	x103	tking@classicmodelcars.com	6	1088	Sales Rep	William Patterson
9	1612	Marsh	Peter	x102	pmarsh@classicmodelcars.com	6	1088	Sales Rep	William Patterson
10	1611	Fixter	Andy	x101	afixter@classicmodelcars.com	6	1088	Sales Rep	William Patterson
11	155555	GNA	rr	152	r@senecacollege.ca	4	1088	cashier	William Patterson
12	1370	Hernandez	Gerard	x2028	ghernande@classicmodelcars.com	4	1102	Sales Rep	Gerard Bondur
13	1337	Bondur	Loui	x6493	lbondur@classicmodelcars.com	4	1102	Sales Rep	Gerard Bondur
14	1401	Castillo	Pamela	x2759	pcastillo@classicmodelcars.com	4	1102	Sales Rep	Gerard Bondur
15	1504	Jones	Barry	x102	bjones@classicmodelcars.com	7	1102	Sales Rep	Gerard Bondur
16	1501	Bott	Larry	x2311	lbott@classicmodelcars.com	7	1102	Sales Rep	Gerard Bondur
17	1702	Gerard	Martin	x2312	mgerard@classicmodelcars.com	4	1102	Sales Rep	Gerard Bondur
18	1188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep	Anthony Bow
19	1166	Thompson	Leslie	x4065	lthompson@classicmodelcars.com	1	1143	Sales Rep	Anthony Bow
20	1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep	Anthony Bow
21	1323	Vanauf	George	x4102	gvanauf@classicmodelcars.com	3	1143	Sales Rep	Anthony Bow

9. Modify the ***vwEmployeeManager*** view so the view returns only employee information for employees who have a manager. Do not DROP and recreate the view – modify it. (Google is your friend).

```
CREATE OR REPLACE VIEW vwEmployeeManager AS
SELECT
e.employeenumber,e.lastname,e.firstname,e.extension,e.email,e.officecode,
e.reportsto,e.jobtitle, m.firstname ||' ' || m.lastname AS manager
FROM retailemployees e FULL JOIN retailemployees m ON m.employeenumber
= e.reportsto
WHERE e.reportsto IS NOT NULL;
```



SQL | All Rows Fetched: 25 in 0.032 seconds

	EMPLOYEENUMBER	LASTNAME	FIRSTNAME	EXTENSION	EMAIL	OFFICECODE	REPORTSTO	JOBTITLE	MANAGER
1	1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com	1	1002	VP Marketing	Diane Murphy
2	1056	Patterson	Mary	x4611	mpatterso@classicmodelcars.com	1	1002	VP Sales	Diane Murphy
3	1102	Bondur	Gerard	x5408	gbondur@classicmodelcars.com	4	1056	Sale Manager (EMEA)	Mary Patterson
4	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)	Mary Patterson
5	1621	Nishi	Mami	x101	mnishi@classicmodelcars.com	5	1056	Sales Rep	Mary Patterson
6	1088	Patterson	William	x4871	wpatterson@classicmodelcars.com	6	1056	Sales Manager (APAC)	Mary Patterson
7	1619	King	Tom	x103	tking@classicmodelcars.com	6	1088	Sales Rep	William Patterson
8	1612	Marsh	Peter	x102	pmarsh@classicmodelcars.com	6	1088	Sales Rep	William Patterson
9	1611	Fixter	Andy	x101	afixter@classicmodelcars.com	6	1088	Sales Rep	William Patterson
10	6655555	GNA	rr	152	r@senecacollege.ca	4	1088	cashier	William Patterson
11	155555	GNA	rr	152	r@senecacollege.ca	4	1088	cashier	William Patterson
12	1702	Gerard	Martin	x2312	mgerard@classicmodelcars.com	4	1102	Sales Rep	Gerard Bondur
13	1504	Jones	Barry	x102	bjones@classicmodelcars.com	7	1102	Sales Rep	Gerard Bondur
14	1337	Bondur	Loui	x6493	lbondur@classicmodelcars.com	4	1102	Sales Rep	Gerard Bondur
15	1370	Hernandez	Gerard	x2028	ghernande@classicmodelcars.com	4	1102	Sales Rep	Gerard Bondur
16	1401	Castillo	Pamela	x2759	pcastillo@classicmodelcars.com	4	1102	Sales Rep	Gerard Bondur
17	1501	Bott	Larry	x2311	lbott@classicmodelcars.com	7	1102	Sales Rep	Gerard Bondur
18	1323	Vanauf	George	x4102	gvanauf@classicmodelcars.com	3	1143	Sales Rep	Anthony Bow
19	1286	Tseng	Foon Yue	x228	ftseng@classicmodelcars.com	3	1143	Sales Rep	Anthony Bow
20	1216	Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales Rep	Anthony Bow
21	1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep	Anthony Bow

10. Drop both ***vwProductOrder*** and ***vwEmployeeManager*** views.

```
DROP VIEW vwProductOrder;
DROP VIEW vwEmployeeManager;
```

```
View VWPRODUCTORDER dropped.


View VWEMPLOYEEMANAGER dropped.
```