

Module

5

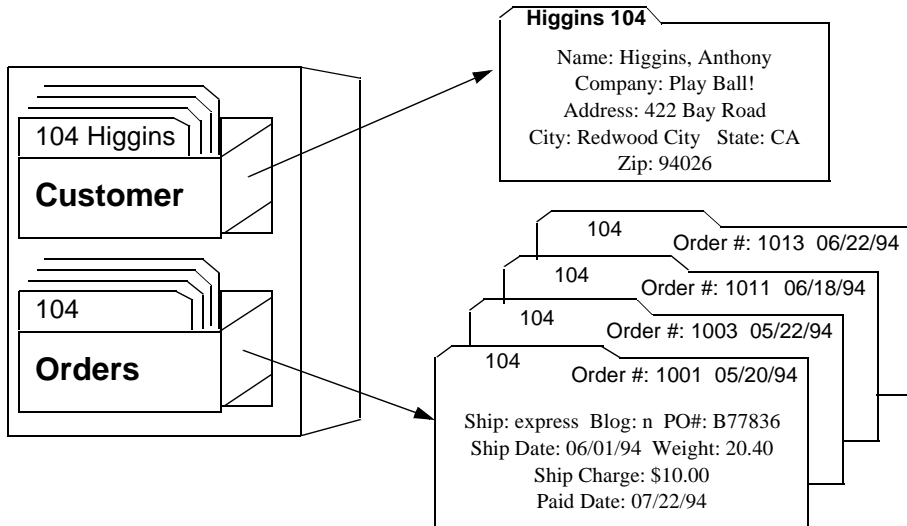
Joins and Subqueries

Objectives

At the end of this module, you will be able to:

- Write a `SELECT` statement to retrieve information from two tables
- Use an `OUTER JOIN` in a query
- Substitute a subquery for an expression in an `SQL` statement

Information from Two Tables



Two Tables in the SELECT

```
SELECT customer.customer_num, company, phone,  
       order_num, order_date, backlog  
FROM customer, orders  
WHERE customer.customer_num =  
       orders.customer_num  
ORDER BY customer_num
```

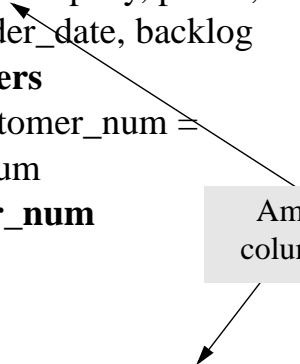
Two Tables in the SELECT: Result

...	
customer_num	104
company	Play Ball!
phone	415-368-1100
order_num	1001
order_date	05/20/1994
backlog	n
customer_num	104
company	Play Ball!
phone	415-368-1100
order_num	1003
order_date	05/22/1994
backlog	n
...	

A Common Error when Joining

```
SELECT customer_num, company, phone,  
       order_num, order_date, backlog  
FROM customer, orders  
WHERE customer.customer_num =  
       orders.customer_num  
ORDER BY customer_num
```

Ambiguous
column name



324: Ambiguous column (customer_num).

A Cartesian Product

```
SELECT customer_num, state, sname  
FROM customer, state
```

customer_num	state	sname
101	CA	Alaska
101	CA	Hawaii
101	CA	California
101	CA	Oregon
101	CA	Washington
...
102	CA	Alaska
102	CA	Hawaii
102	CA	California
102	CA	Oregon
...

Another Join Example

```
SELECT *  
FROM stock, manufact  
WHERE stock.manu_code = manufact.manu_code
```

stock_num	1
manu_code	SMT
description	baseball gloves
unit_price	\$450.00
unit	case
unit_descr	10 gloves/case
manu_code	SMT
manu_name	Smith
lead_time	3
...	...

Natural Joins

```
SELECT stock.*, manu_name, lead_time
FROM stock, manufact
WHERE stock.manu_code = manufact.manu_code
```

stock_num	1
manu_code	SMT
description	baseball gloves
unit_price	\$450.00
unit	case
unit_descr	10 gloves/case
manu_name	Smith
lead_time	3
...	...

Multiple Table Joins

```
SELECT customer.company, orders.order_num, item_num
FROM customer, orders, items
WHERE customer.customer_num = orders.customer_num AND
      orders.order_num = items.order_num AND
      customer.customer_num = 104
```

company	order_num	item_num
Play Ball!	1001	1
Play Ball!	1003	1
Play Ball!	1003	2
Play Ball!	1003	3
Play Ball!	1011	1
Play Ball!	1013	1
Play Ball!	1013	2
Play Ball!	1013	3
Play Ball!	1013	4

Aliases

```
SELECT c.customer_num, o.order_num, i.item_num  
FROM customer c, orders o, items i  
WHERE c.customer_num = o.customer_num AND  
      o.order_num = i.order_num AND  
      c.customer_num = 104
```

Outer Joins

OUR CUSTOMER table has :

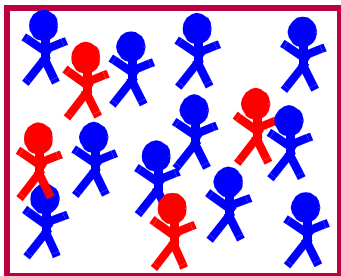


Customers that have placed orders



Customers that have NOT placed orders!

CUSTOMER



■ Join



■ Outer Join



Simple Join vs. Simple Outer Join

```
SELECT customer.customer_num, lname, order_num  
FROM customer, orders  
WHERE customer.customer_num =  
      orders.customer_num
```

Simple Join

```
SELECT customer.customer_num, lname, order_num  
FROM customer, OUTER orders  
WHERE customer.customer_num =  
      orders.customer_num
```

Outer Join

Comparing Results

Result of a Simple Join

customer_num	lname	order_num
104	Higgins	1001
101	Pauli	1002
104	Higgins	1003
106	Watson	1004
116	Parmelee	1005
112	Lawson	1006
117	Sipes	1007
110	Jaeger	1008
...
126	Neelie	1022
127	Satifer	1023

Result of a Outer Join

customer_num	lname	order_num
101	Pauli	1002
102	Sadler	
103	Currie	
104	Higgins	1001
104	Higgins	1003
104	Higgins	1011
104	Higgins	1013
105	Vector	
106	Watson	1004
106	Watson	1014
107	Ream	
108	Quinn	
...
127	Satifer	1023
128	Lessor	

Nested SELECT Statements

- Compare an expression to the result of another SELECT statement
- Determine whether an expression is included in the results of another SELECT statement

Nested SELECT Statement Example

```
SELECT * ← Outer SELECT
  FROM customer
 WHERE city =
  (SELECT city ← Inner SELECT
   FROM customer
   WHERE lname = 'Higgins')
```

Values Returned by Subqueries

■ A single value

```
SELECT *  
  FROM manufact  
 WHERE lead_time >  
        (SELECT lead_time  
          FROM manufact  
          WHERE manu_code = "ANZ");
```

■ A list of values

```
SELECT * FROM items  
  WHERE manu_code IN  
 (SELECT manu_code FROM manufact  
  WHERE lead_time > "1");
```


NOT IN

```
SELECT fname, lname, company  
FROM customer  
WHERE customer_num NOT IN  
  (SELECT customer_num  
   FROM orders)
```



Lab Exercise

Lab Exercise

1. Sam needs a report of all the customers who have placed orders. He'd like the customer number, company name, and order number.
2. Now he'd like the list to include order date, and the list sorted by that date in descending order.
3. It's time to print out the new price lists. You need the manufacturer name, stock number, description, units, and unit price.
4. Shipping has a problem. You need a list of the items on order #1004, including a description of each item.
5. Include the manufacturer and lead time for each item on order #1004.
6. Customer Service sent Sam a list of our customers and the total number of phone calls we've received from the customer. Sam thought he would use that same list for his direct mail campaign, but some of his customers don't seem to be on it. Why?
7. The new data entry person made some mistakes when she was entering orders, and some of the manufacturer's codes may not be valid. Can you check to make sure that all the codes are correct?

Challenge Exercise

You're thinking about grouping your customers into regions. Region 1 would be California, Region 2 would be Arizona, and all the rest would be Region 3. You'd like to see what the dollar amount of all the orders for those regions would be, to help you figure out the sales staffing. You'd like the total per region.



Solutions



Solution

Solution

1.

```
SELECT customer.customer_num, company, order_num
FROM customer, orders
WHERE customer.customer_num = orders.customer_num;
```

Results:

customer_num	company	order_num
104	Play Ball!	1001
101	All Sports Supplies	1002
104	Play Ball!	1003
106	Watson & Son	1004
...		
123	Bay Sports	1020
124	Putnum's Putters	1021
126	Neelie's Discount Sp	1022
127	Big Blue Bike Shop	1023

2.

```
SELECT customer.customer_num, company, order_num,
       order_date
FROM customer, orders
WHERE customer.customer_num = orders.customer_num
ORDER BY order_date DESC;
```

Results:

customer_num	company	order_num	order_date
127	Big Blue Bike Shop	1023	07/24/1994
126	Neelie's Discount Sp	1022	07/24/1994
124	Putnum's Putters	1021	07/23/1994
123	Bay Sports	1020	07/11/1994
122	The Sporting Life	1019	07/11/1994
...			
106	Watson & Son	1004	05/22/1994
104	Play Ball!	1003	05/22/1994
101	All Sports Supplies	1002	05/21/1994
104	Play Ball!	1001	05/20/1994



Solution

Solution

3.

```
SELECT manu_name, stock_num, description, unit, unit_price
FROM manufact, stock
WHERE manufact.manu_code = stock.manu_code;
```

Results:

manu_name	stock_num	description	unit	unit_price
Smith	1	baseball gloves	case	\$450.00
Smith	5	tennis racquet	each	\$25.00
Smith	6	tennis ball	case	\$36.00
Anza	5	tennis racquet	each	\$19.80
Anza	6	tennis ball	case	\$48.00
Anza	8	volleyball	case	\$840.00
Anza	9	volleyball net	each	\$20.00
Anza	110	helmet	case	\$244.00
Anza	201	golf shoes	each	\$75.00
Anza	205	3 golf balls	case	\$312.00
Anza	301	running shoes	each	\$95.00
Anza	304	watch	box	\$170.00
Anza	310	kick board	case	\$84.00
Anza	313	swim cap	box	\$60.00
...				

4.

```
SELECT items.*, description
FROM items, stock
WHERE items.stock_num = stock.stock_num
AND items.manu_code = stock.manu_code
AND items.order_num = 1004;
```

Results:

item_num	order_num	stock_num	manu_code	quantity	total_price	description
1	1004	1	HRO	1	\$250.00	baseball gloves
2	1004	2	HRO	1	\$126.00	baseball
3	1004	3	HSK	1	\$240.00	baseball bat
4	1004	1	HSK	1	\$800.00	baseball gloves



Solution

Solution

5.

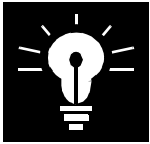
```
SELECT items.*, description, manu_name, lead_time
FROM items, stock, manufact
WHERE items.stock_num = stock.stock_num
AND items.manu_code = stock.manu_code
AND stock.manu_code = manufact.manu_code
AND items.order_num = 1004;
```

Results:

```
item_num    1
order_num   1004
stock_num    1
manu_code   HRO
quantity     1
total_price $250.00
description  baseball gloves
manu_name   Hero
lead_time    4
```

```
item_num    2
order_num   1004
stock_num    2
manu_code   HRO
quantity     1
total_price $126.00
description  baseball
manu_name   Hero
lead_time    4
```

```
item_num    3
order_num   1004
stock_num    3
manu_code   HSK
quantity     1
total_price $240.00
description  baseball bat
manu_name   Husky
lead_time    5
```



Solution

Solution

```

item_num    4
order_num   1004
stock_num    1
manu_code   HSK
quantity    1
total_price $800.00
description baseball gloves
manu_name   Husky
lead_time    5

```

6. The SQL statement didn't use an outer join to retrieve all the customers, regardless of whether they had made calls.

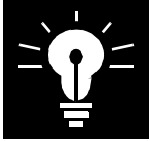
7.

```

SELECT order_num FROM items
WHERE manu_code NOT IN
      (SELECT manu_code FROM manufact);

```

There are no rows returned, so all of the orders have the correct manufacturer's code.



Solution

Solution

Challenge Exercise:

```
SELECT
  CASE
    WHEN state = "CA"
      THEN "Region 1"
    WHEN state = "AZ"
      THEN "Region 2"
    ELSE "Region 3"
  END,
  SUM(total_price)
FROM customer, orders, items
WHERE customer.customer_num = orders.customer_num
  AND orders.order_num = items.order_num
GROUP BY 1;
```

Results:

(expression)	(sum)
Region 1	\$11177.80
Region 2	\$584.00
Region 3	\$6392.97