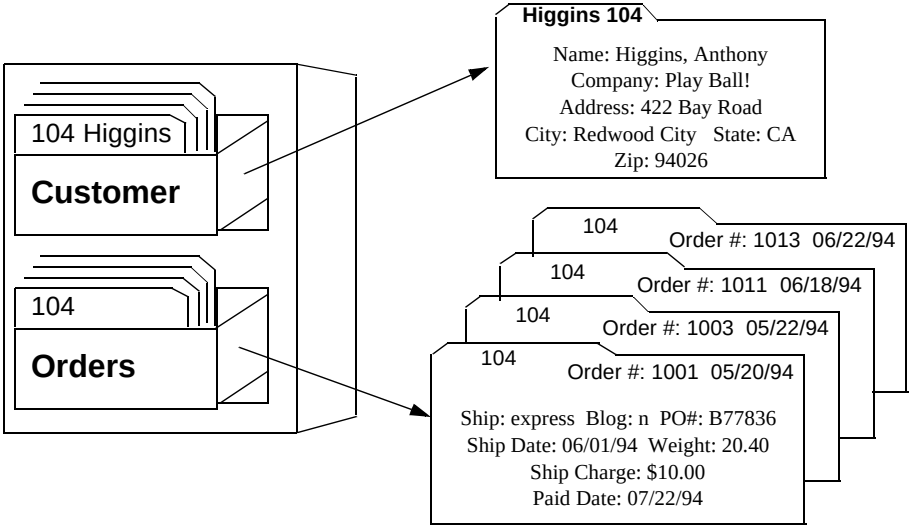
Information from Two Tables



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
```

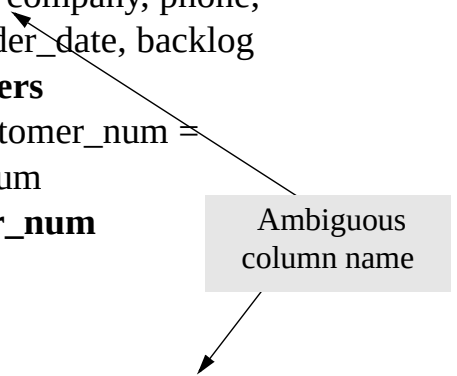
Two Tables in the SELECT: Result

```
...
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
```



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

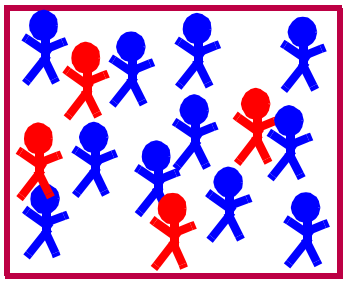
```
SELECT c.customer_num, o.order_num, i.item_num
    FROM customer c, orders o, items i
    WHERE c.customer_num = o.customer_num AND
            o.order_num = i.order_num AND
            c.customer_num = 104
```

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

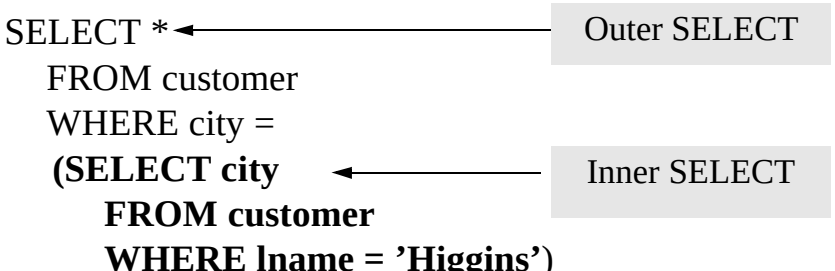
Result of a Outer Join

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



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)
```