JOINS

Joins

- Relational databases create relationships between any two tables
- Consider this:
- We want to store information about product and our supplier
- We could create a table that stores a product name, a qty and a supplier

```
CREATE TABLE product(
id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
prod_name VARCHAR(30) NOT NULL,
qty SMALLINT NOT NULL,
supplier VARCHAR(10)
)ENGINE=INNODB;
```

Joins

- But what if we wanted to store more than that – what if we wanted to store all the contact information for a supplier as well?

```
CREATE TABLE product(
id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
prod_name VARCHAR(30) NOT NULL,
qty SMALLINT NOT NULL,
supplier VARCHAR(10),
contactName VARCHAR(50),
contactPhone CHAR(10),
contactEmail VARCHAR(30),
contactPosition VARCHAR(15)
)ENGINE=INNODB;
```

Joins

- We start inserting information into our new table:

```
INSERT INTO product (prod_name,qty, supplier, contactName, contactPhone, contactEmail,
contactPosition)
VALUES
('Chicken', 2, 'Farm Chicken Supplier', 'John Doe', '555-6656','jdoe@email.com','Buyer'),
('Turkey',14,'Farm Chicken Supplier', 'John Doe','555-6666','jdoe@email.com','Buyer'),
('Beef',22, 'Cow Farms','Mike Smith', '666-9656', 'msmith@email.com', 'manager');
```

- As this information grows it becomes more and more cumbersome….
  - Repeating information – waste of storage space
  - If the vendor changes – multiple records need to be updated
  - When repeated data is used, the higher the chance of errors

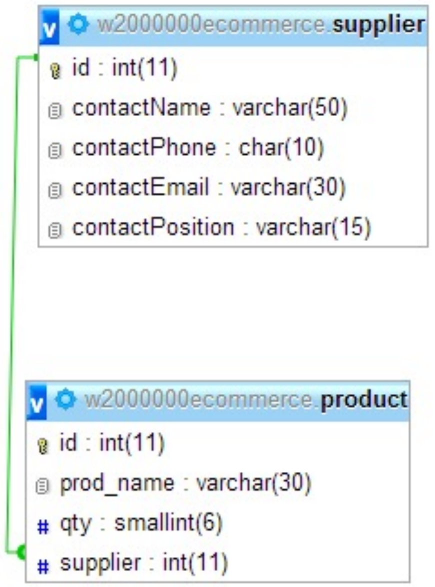
Joins

- Relational databases help with these issues

```
CREATE TABLE supplier(
id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
supplierName VARCHAR(30),
contactName VARCHAR(50),
contactPhone CHAR(10),
contactEmail VARCHAR(30),
contactPosition VARCHAR(15)
);

CREATE TABLE product(
id INTEGER NOT NULL AUTO_INCREMENT
PRIMARY KEY,
prod_name VARCHAR(30) NOT NULL,
qty SMALLINT NOT NULL,
supplier INTEGER,
FOREIGN KEY(supplier)
REFERENCES supplier(id)
);
```

Joins

- Relational Database



w2000000ecommerce.supplier
id : int(11)
contactName : varchar(50)
contactPhone : char(10)
contactEmail : varchar(30)
contactPosition : varchar(15)

w2000000ecommerce.product
id : int(11)
prod_name : varchar(30)
qty : smallint(6)
supplier : int(11)

Joins

- Add information to the Supplier table

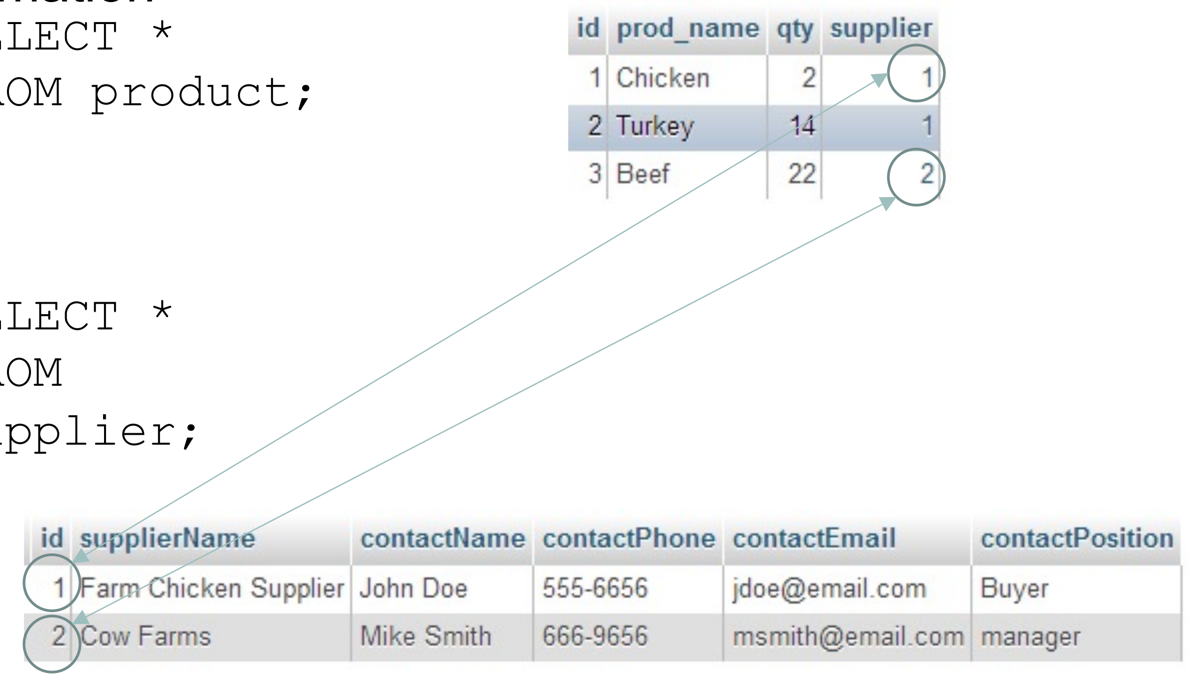
```
INSERT INTO supplier
(supplierName,contactName, contactPhone, contactEmail, contactPosition)
VALUES
('Farm Chicken Supplier', 'John Doe', '555-6656','jdoe@email.com','Buyer'),
('Cow Farms','Mike Smith', '666-9656', 'msmith@email.com', 'manager');


INSERT INTO product
(prod_name, qty, supplier)
VALUES
('Chicken', 2, 1),
('Turkey',14,1),
('Beef',22,2);
```

Joins

- Retrieving Information

```
SELECT *
FROM product;
```

```
SELECT *
FROM
supplier;
```



id	prod_name	qty	supplier
1	Chicken	2	1
2	Turkey	14	1
3	Beef	22	2

id	supplierName	contactName	contactPhone	contactEmail	contactPosition
1	Farm Chicken Supplier	John Doe	555-6656	jdoe@email.com	Buyer
2	Cow Farms	Mike Smith	666-9656	msmith@email.com	manager

Joins

- You can use this relationship to query information from more than one table
- A JOIN relates or associates two tables producing a single table
- There are several types of JOINS:
  - INNER JOIN
  - OUTER JOIN
    - LEFT OUTER JOIN
    - RIGHT OUTER JOIN

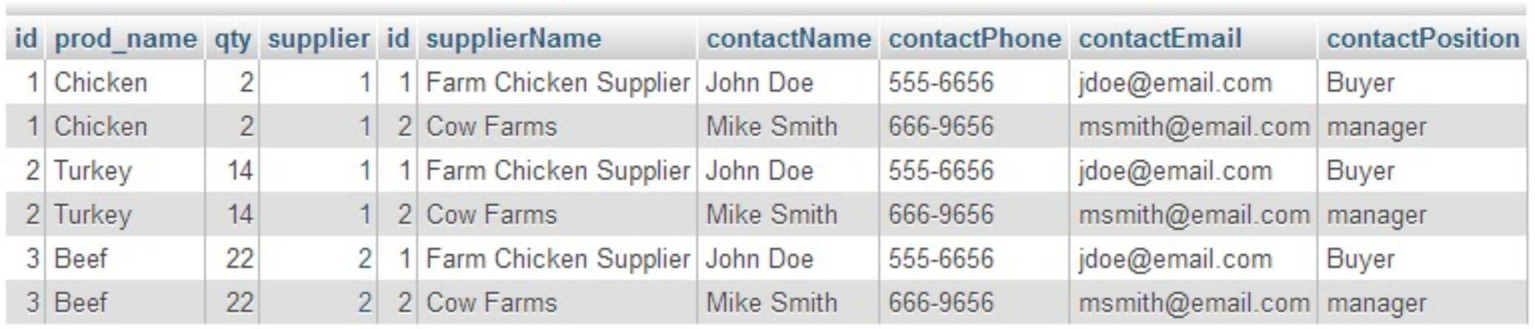
Characteristics of a Join

- Tables are joined, row by row and side by side, by satisfying whatever JOIN conditions are imposed
- Non-matching rows are included or excluded based on the ***join type***
- A typical join condition specifies a **foreign key** in one table and the associated **primary key** in another
- If a join's connecting columns contain nulls the nulls will never join (remember a null is unknown – so an unknown cannot equal another unknown)

Joining the two tables

- Get all the results from the two table joined on the ids

```
SELECT *
FROM product, supplier
```



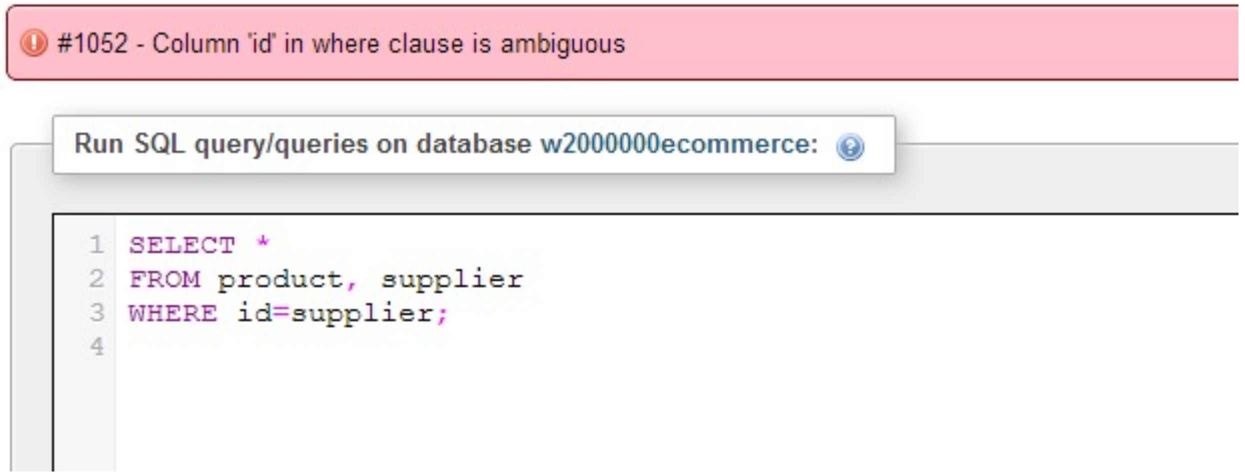
id	prod_name	qty	supplier	id	supplierName	contactName	contactPhone	contactEmail	contactPosition
1	Chicken	2	1	1	Farm Chicken Supplier	John Doe	555-6656	jdoe@email.com	Buyer
1	Chicken	2	1	2	Cow Farms	Mike Smith	666-9656	msmith@email.com	manager
2	Turkey	14	1	1	Farm Chicken Supplier	John Doe	555-6656	jdoe@email.com	Buyer
2	Turkey	14	1	2	Cow Farms	Mike Smith	666-9656	msmith@email.com	manager
3	Beef	22	2	1	Farm Chicken Supplier	John Doe	555-6656	jdoe@email.com	Buyer
3	Beef	22	2	2	Cow Farms	Mike Smith	666-9656	msmith@email.com	manager

This result set is called a Cartesian Product – the number of rows retrieved will be the number of rows in the first table (product) multiplied by the number of rows in the second table (supplier)

Joining the two tables

- The WHERE clause is required to tell the database how to associate the information
- Acts as a filter

```
SELECT *
FROM product, supplier
WHERE id=supplier;
```

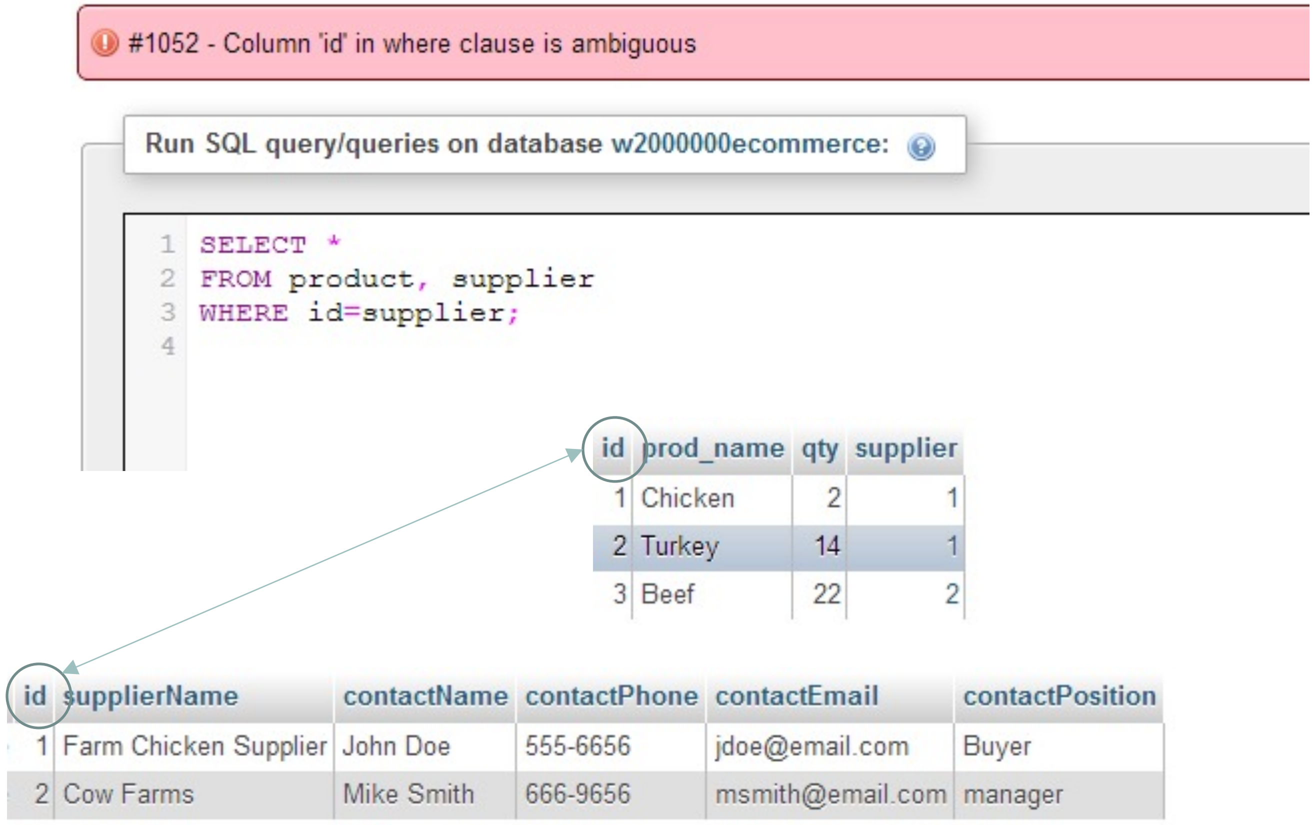


#1052 - Column 'id' in where clause is ambiguous

Run SQL query/queries on database w2000000ecommerce:

```
SELECT *
FROM product, supplier
WHERE id=supplier;
```

INNER JOIN



#1052 - Column 'id' in where clause is ambiguous

Run SQL query/queries on database w2000000ecommerce:

```
SELECT *
FROM product, supplier
WHERE id=supplier;
```

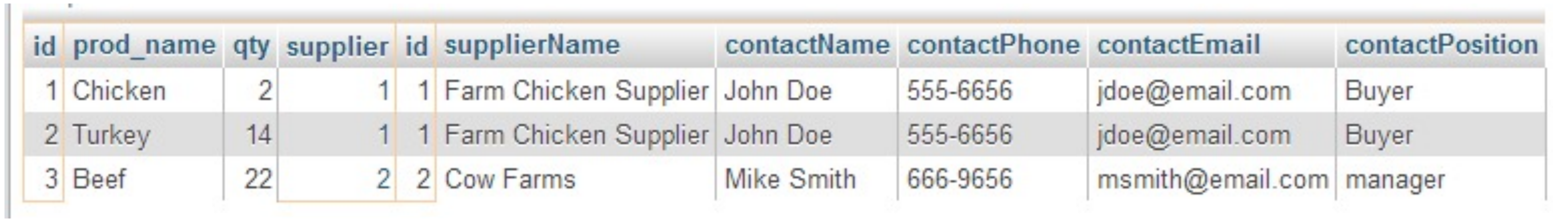
id	prod_name	qty	supplier
1	Chicken	2	1
2	Turkey	14	1
3	Beef	22	2

id	supplierName	contactName	contactPhone	contactEmail	contactPosition
1	Farm Chicken Supplier	John Doe	555-6656	jdoe@email.com	Buyer
2	Cow Farms	Mike Smith	666-9656	msmith@email.com	manager

Qualifying Columns

- Qualifying column names ***is mandatory*** when there is more than one instance

```
SELECT *
FROM product, supplier
WHERE product.supplier=supplier.id;
```



id	prod_name	qty	supplier	id	supplierName	contactName	contactPhone	contactEmail	contactPosition
1	Chicken	2	1	1	Farm Chicken Supplier	John Doe	555-6656	jdoe@email.com	Buyer
2	Turkey	14	1	1	Farm Chicken Supplier	John Doe	555-6656	jdoe@email.com	Buyer
3	Beef	22	2	2	Cow Farms	Mike Smith	666-9656	msmith@email.com	manager

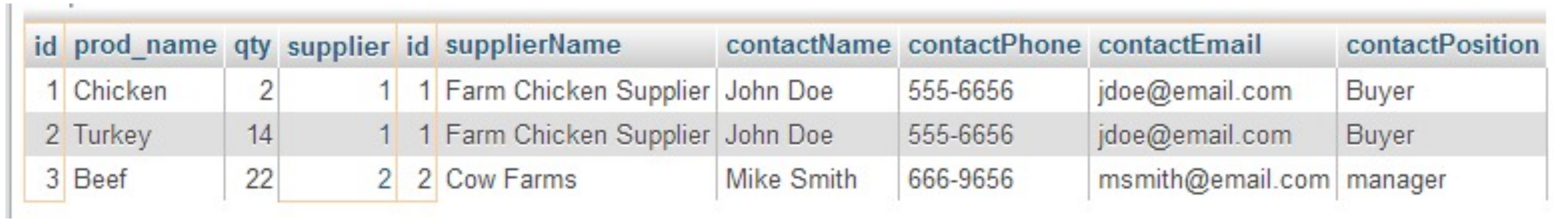
Explicit Declarations

- Use the keyword INNER JOIN and the ON clause
- Only the rows that satisfy the condition in the ON clause are returned

```
SELECT column1, column2, column3
FROM table_a INNER JOIN table_b
  ON column_x=column_y
```

Explicit Declarations

```
SELECT *
FROM product INNER JOIN supplier
ON product.supplier=supplier.id;
```



id	prod_name	qty	supplier	id	supplierName	contactName	contactPhone	contactEmail	contactPosition
1	Chicken	2	1	1	Farm Chicken Supplier	John Doe	555-6656	jdoe@email.com	Buyer
2	Turkey	14	1	1	Farm Chicken Supplier	John Doe	555-6656	jdoe@email.com	Buyer
3	Beef	22	2	2	Cow Farms	Mike Smith	666-9656	msmith@email.com	manager

Qualifying Column Names

- Table Aliases
  - An alternate name assigned to a table in the query (often shorter than the table name)
  - Aliases are temporary and are only valid for the duration of the query

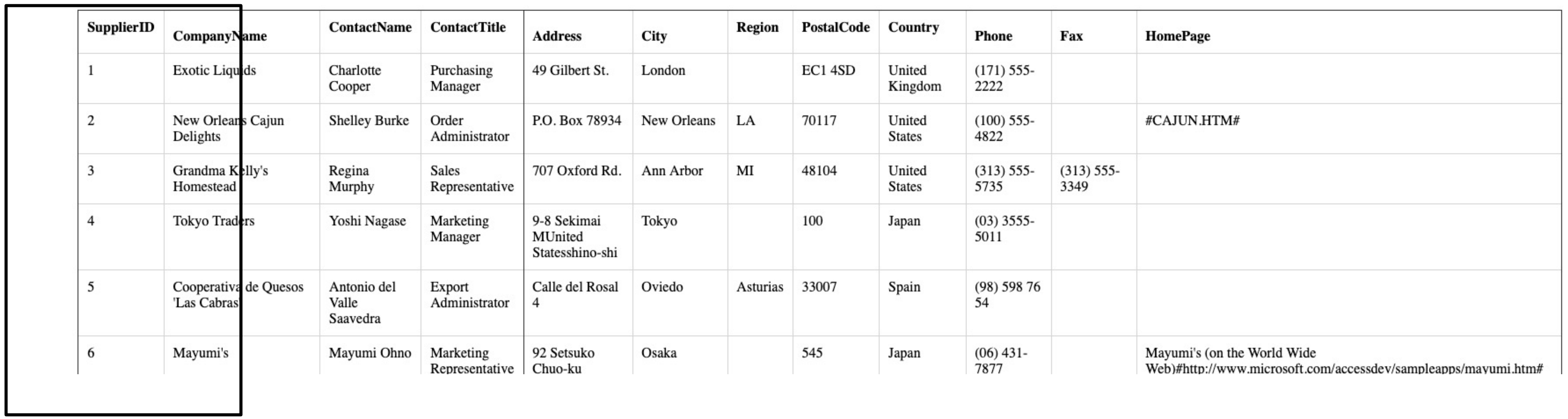
```
SELECT t.title_name, t.type, p.pub_name
FROM titles AS t INNER JOIN publishers AS p
  ON t.pub_id=p.pub_id;
```

Qualifying Column Names

- You can mix qualified and unqualified names
- It is not required – but helps to improve system performance to qualify all columns AND helps makes the query self-documenting (you know what it is doing and where the columns come from)
- Helps to prevent any disruptions to your query caused by additions later (where a column is added to a table that now makes your query require qualification)

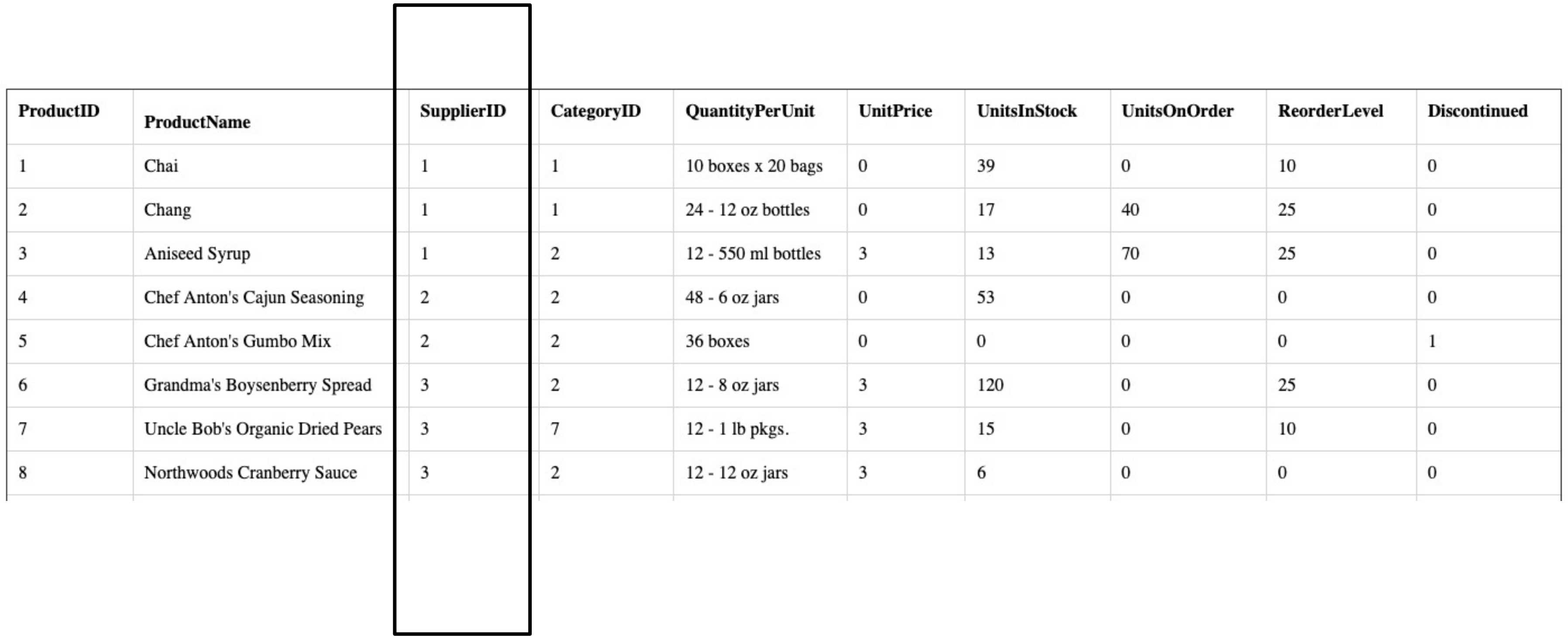
Inner Join

Suppliers table:



SupplierID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone	Fax	HomePage
1	Exotic Liquids	Charlotte Cooper	Purchasing Manager	49 Gilbert St.	London		EC1 4SD	United Kingdom	(171) 555-2222		
2	New Orleans Cajun Delights	Shelley Burke	Order Administrator	P.O. Box 78934	New Orleans	LA	70117	United States	(100) 555-4822		#CAJUN.HTM#
3	Grandma Kelly's Homestead	Regina Murphy	Sales Representative	707 Oxford Rd.	Ann Arbor	MI	48104	United States	(313) 555-5735	(313) 555-3349	
4	Tokyo Traders	Yoshi Nagase	Marketing Manager	9-8 Sekimai MUnited Statesshino-shi	Tokyo		100	Japan	(03) 3555-5011		
5	Cooperativa de Quesos 'Las Cabras'	Antonio del Valle Saavedra	Export Administrator	Calle del Rosal 4	Oviedo	Asturias	33007	Spain	(98) 598 76 54		
6	Mayumi's	Mayumi Ohno	Marketing Representative	92 Setsuko Chuo-ku	Osaka		545	Japan	(06) 431-7877		Mayumi's (on the World Wide Web)#http://www.microsoft.com/accessdev/sampleapps/mayumi.htm#

Products table



ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued
1	Chai	1	1	10 boxes x 20 bags	0	39	0	10	0
2	Chang	1	1	24 - 12 oz bottles	0	17	40	25	0
3	Aniseed Syrup	1	2	12 - 550 ml bottles	3	13	70	25	0
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	0	53	0	0	0
5	Chef Anton's Gumbo Mix	2	2	36 boxes	0	0	0	0	1
6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	3	120	0	25	0
7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	3	15	0	10	0
8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	3	6	0	0	0

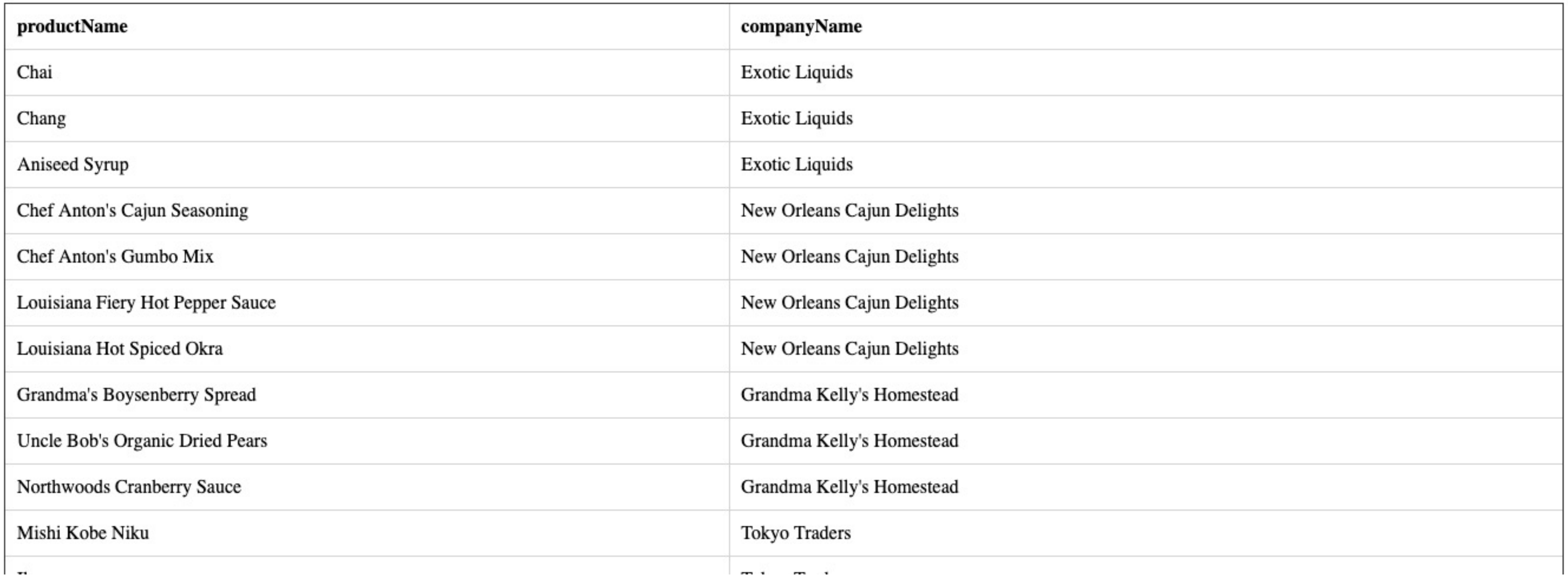
Joining the two tables

```
SELECT p.productName, s.companyName
FROM products AS p INNER JOIN suppliers AS s
ON p.SupplierID = s.SupplierID;
```

OR

```
SELECT products.productName, suppliers.companyName
FROM products INNER JOIN suppliers
ON products.SupplierID = suppliers.SupplierID;
```

Query Result

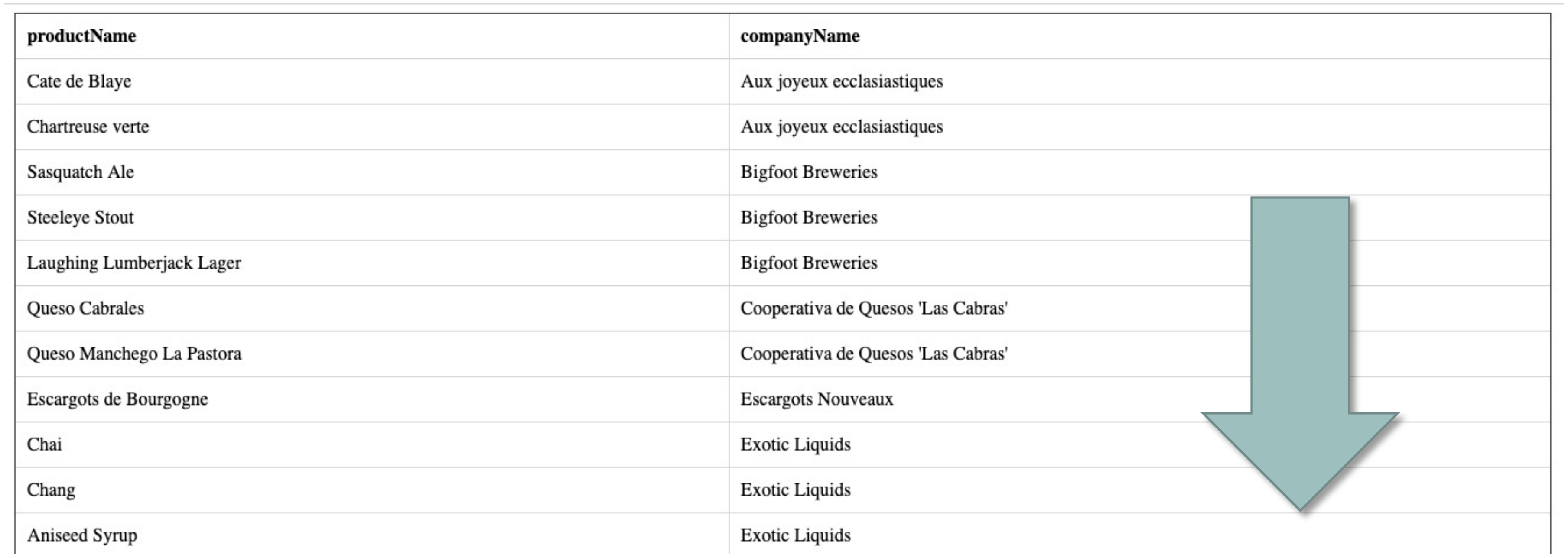


productName	companyName
Chai	Exotic Liquids
Chang	Exotic Liquids
Aniseed Syrup	Exotic Liquids
Chef Anton's Cajun Seasoning	New Orleans Cajun Delights
Chef Anton's Gumbo Mix	New Orleans Cajun Delights
Louisiana Fiery Hot Pepper Sauce	New Orleans Cajun Delights
Louisiana Hot Spiced Okra	New Orleans Cajun Delights
Grandma's Boysenberry Spread	Grandma Kelly's Homestead
Uncle Bob's Organic Dried Pears	Grandma Kelly's Homestead
Northwoods Cranberry Sauce	Grandma Kelly's Homestead
Mishi Kobe Niku	Tokyo Traders
[illegible]	[illegible]

Organize by supplier

```
SELECT products.productName, suppliers.companyName
FROM suppliers INNER JOIN products
ON products.SupplierID = suppliers.SupplierID
ORDER BY suppliers.companyName;
```

Query Result

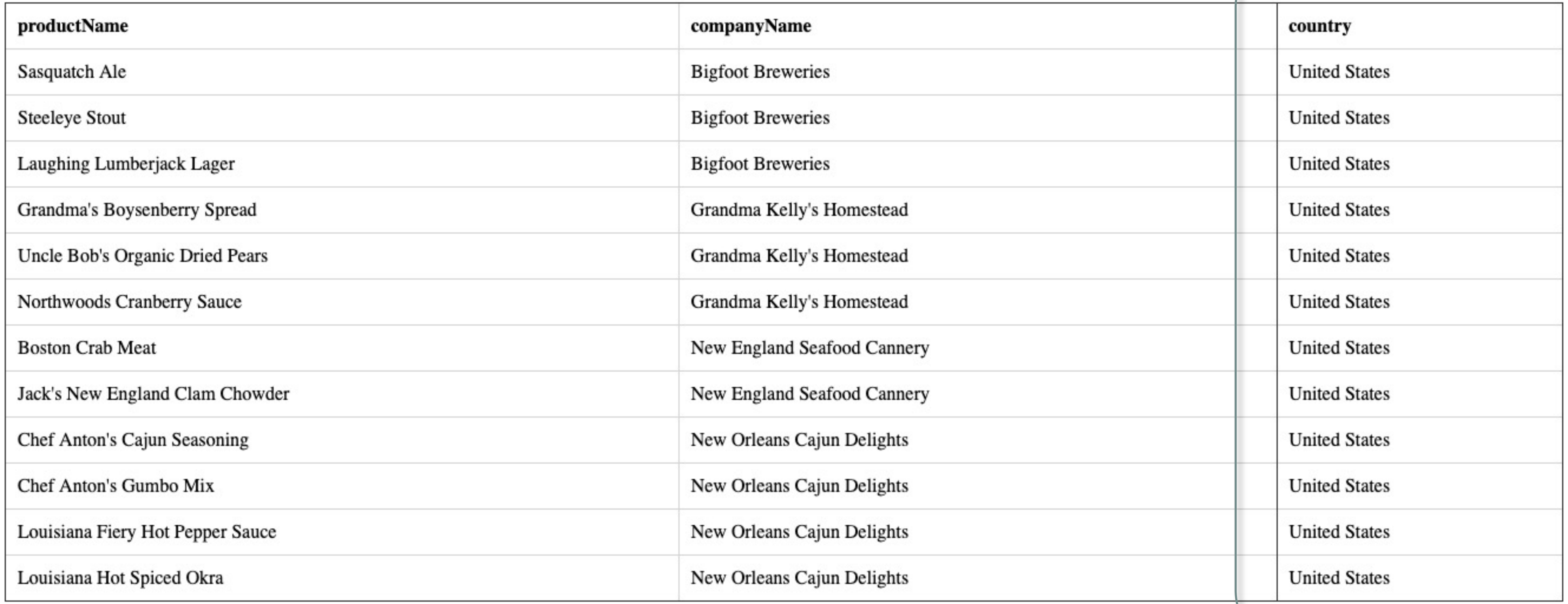


productName	companyName
Cate de Blaye	Aux joyeux ecclasiastiques
Chartreuse verte	Aux joyeux ecclasiastiques
Sasquatch Ale	Bigfoot Breweries
Steeleye Stout	Bigfoot Breweries
Laughing Lumberjack Lager	Bigfoot Breweries
Queso Cabrales	Cooperativa de Quesos 'Las Cabras'
Queso Manchego La Pastora	Cooperativa de Quesos 'Las Cabras'
Escargots de Bourgogne	Escargots Nouveaux
Chai	Exotic Liquids
Chang	Exotic Liquids
Aniseed Syrup	Exotic Liquids

Filter the result

```
SELECT products.productName, suppliers.companyName,
suppliers.country

FROM suppliers INNER JOIN products

ON products.SupplierID = suppliers.SupplierID

WHERE country = 'United States'

ORDER BY suppliers.companyName;
```

Query Results



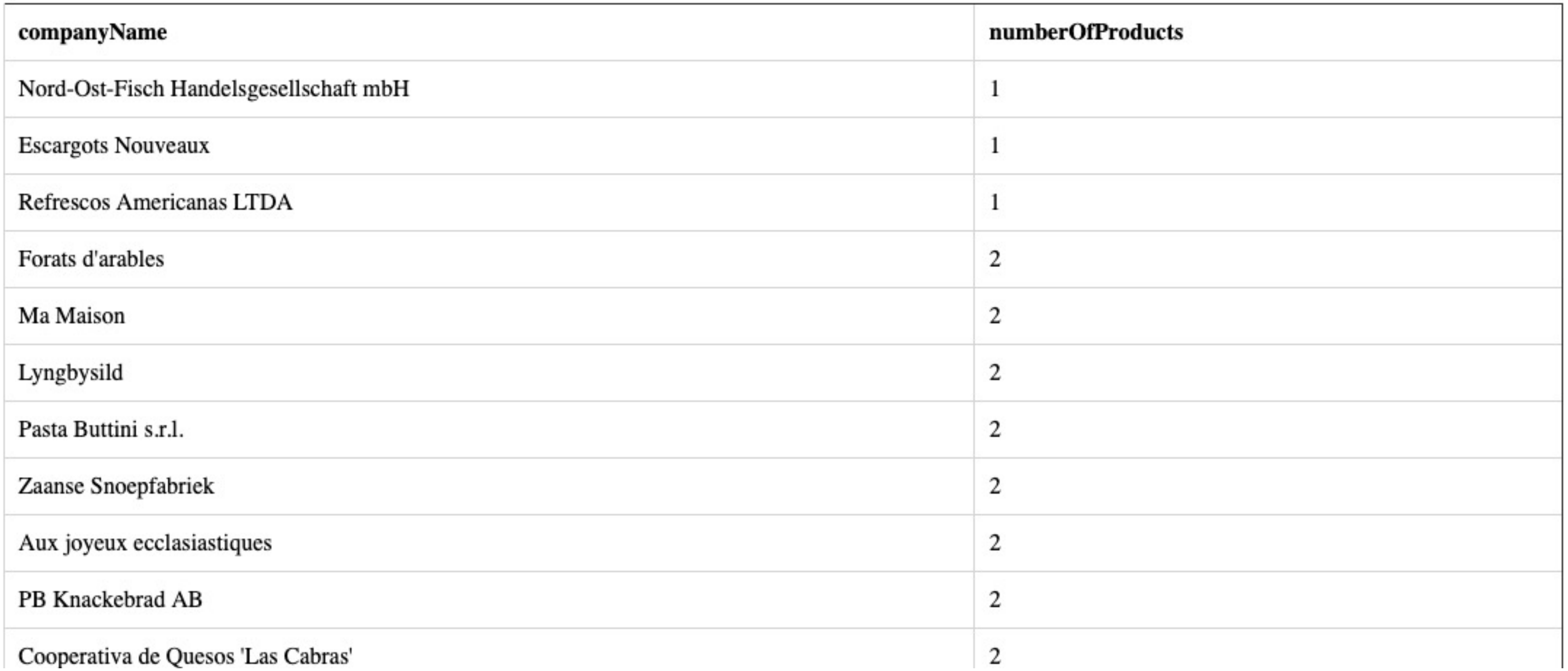
productName	companyName	country
Sasquatch Ale	Bigfoot Breweries	United States
Steeleye Stout	Bigfoot Breweries	United States
Laughing Lumberjack Lager	Bigfoot Breweries	United States
Grandma's Boysenberry Spread	Grandma Kelly's Homestead	United States
Uncle Bob's Organic Dried Pears	Grandma Kelly's Homestead	United States
Northwoods Cranberry Sauce	Grandma Kelly's Homestead	United States
Boston Crab Meat	New England Seafood Cannery	United States
Jack's New England Clam Chowder	New England Seafood Cannery	United States
Chef Anton's Cajun Seasoning	New Orleans Cajun Delights	United States
Chef Anton's Gumbo Mix	New Orleans Cajun Delights	United States
Louisiana Fiery Hot Pepper Sauce	New Orleans Cajun Delights	United States
Louisiana Hot Spiced Okra	New Orleans Cajun Delights	United States

Aggregate

- List the number of products that each supplier provides

```
SELECT s.companyName, COUNT(p.productName) AS numberOfProducts
FROM suppliers AS s INNER JOIN products as p
ON s.SupplierID = p.SupplierID
GROUP BY s.companyName
ORDER BY numberOfProducts;
```

Query Result

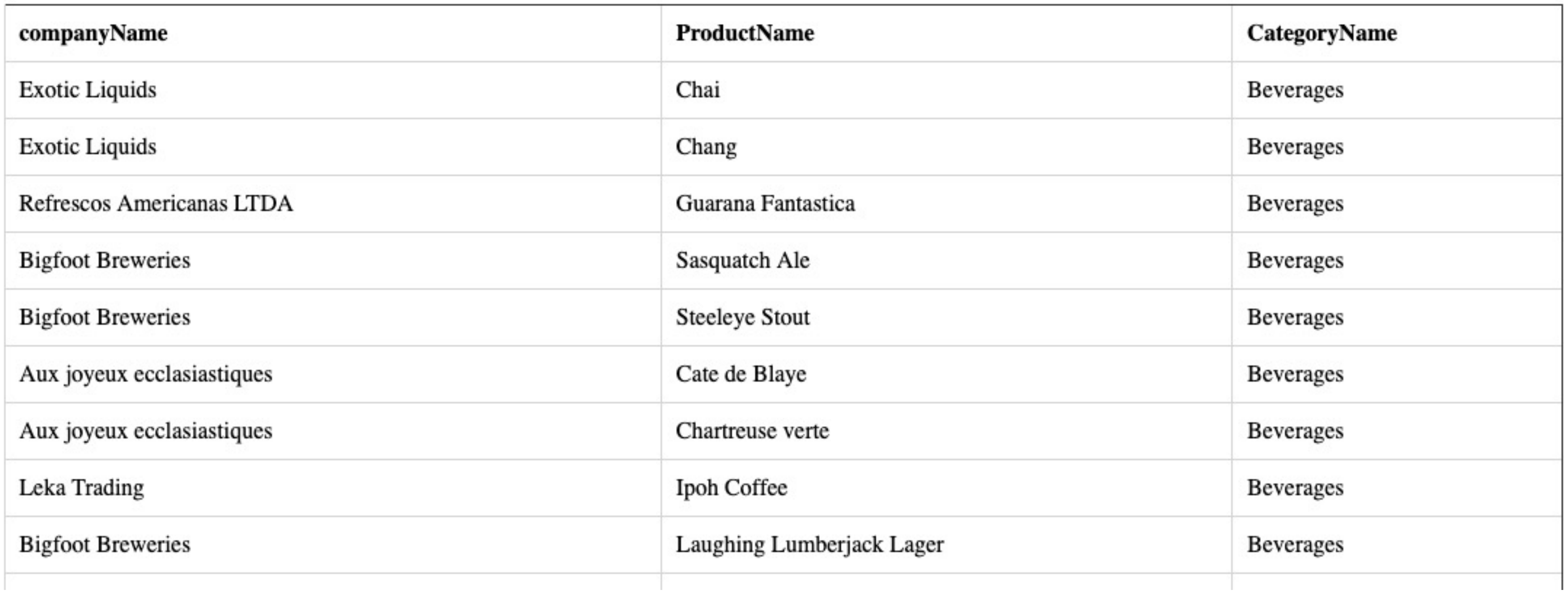


companyName	numberOfProducts
Nord-Ost-Fisch Handelsgesellschaft mbH	1
Escargots Nouveaux	1
Refrescos Americanas LTDA	1
Forats d'arables	2
Ma Maison	2
Lyngbysild	2
Pasta Buttini s.r.l.	2
Zaanse Snoepfabriek	2
Aux joyeux ecclasiastiques	2
PB Knackebrad AB	2
Cooperativa de Quesos 'Las Cabras'	2

JOIN more than one table

```
SELECT s.companyName, p.ProductName, c.CategoryName
FROM suppliers AS s INNER JOIN products as p
ON s.SupplierID = p.SupplierID
INNER JOIN categories as c
ON p.CategoryID = c.CategoryID;
```

Query Result



companyName	ProductName	CategoryName
Exotic Liquids	Chai	Beverages
Exotic Liquids	Chang	Beverages
Refrescos Americanas LTDA	Guarana Fantastica	Beverages
Bigfoot Breweries	Sasquatch Ale	Beverages
Bigfoot Breweries	Steeleye Stout	Beverages
Aux joyeux ecclasiastiques	Cate de Blaye	Beverages
Aux joyeux ecclasiastiques	Chartreuse verte	Beverages
Leka Trading	Ipoh Coffee	Beverages
Bigfoot Breweries	Laughing Lumberjack Lager	Beverages

OUTER JOINS

- INNER JOINS do *NOT* return rows that do not match with a row from the other table
- OUTER JOINS return all rows from at least one of the tables
- Useful for answering questions about missing quantities
  - Authors who have written no books
- Or reports that want all information from one table and matching rows from another

Outer Joins

- The order in which you specify the tables in an outer join are important
- Left outer join includes all the rows from the ***left*** table
  - If a row in the left table has no matching rows in the right, the associated row in the result contains NULLS
- Right outer join includes all the rows from the ***right*** table
- Full outer join returns all rows in the left and right tables

Sample information

```
SELECT au_fname, au_lname, city
FROM authors ;
```

Sarah	Buchman	Bronx
Wendy	Heydemark	Boulder
Hallie	Hull	San Francisco
Klee	Hull	San Francisco
Christian	Kells	New York
	Kellsey	Palo Alto
Paddy	O'Furniture	Sarasota

Sample Information

```
SELECT pub_name, city
FROM publishers
```

Abatis Publishers	New York
Core Dump Books	San Francisco
Schadenfreude Press	Hamburg
Tenterhooks Press	Berkeley

Comparison

- List the authors that live in cities which have a publisher…

```
SELECT a.au_fname, a.au_lname,p.pub_name
FROM authors AS a
INNER JOIN publishers AS p
 ON a.city=p.city;
```

Query Results

au_fname	au_lname	pub_name
Hallie	Hull	Core Dump Books
Klee	Hull	Core Dump Books
Christian	Kells	Abatis Publishers

Left Outer Join

```
SELECT a.au_fname, a.au_lname, p.pub_name
FROM authors AS a  LEFT OUTER JOIN publishers AS p
  ON a.city = p.city;
```

au_fname	au_lname	pub_name
Sarah	Buchman	NULL
Wendy	Heydemark	NULL
Hallie	Hull	Core Dump Books
Klee	Hull	Core Dump Books
Christian	Kells	Abatis Publishers
	Kellsey	NULL
Paddy	O'Furniture	NULL

Right Outer Join

```
SELECT a.au_fname, a.au_lname, p.pub_name
FROM authors AS a RIGHT OUTER JOIN publishers AS p
  ON a.city = p.city
```

au_fname	au_lname	pub_name
Christian	Kells	Abatis Publishers
Hallie	Hull	Core Dump Books
Klee	Hull	Core Dump Books
NULL	NULL	Schadenfreude Press
NULL	NULL	Tenterhooks Press

Outer joins

- Left and right outer joins are equivalent, its just a matter of what table is the outer table

```
SELECT a.au_fname, a.au_lname, p.pub_name
FROM authors AS a RIGHT OUTER JOIN
publishers AS p
  ON a.city = p.city
```

```
SELECT a.au_fname, a.au_lname, p.pub_name
FROM publishers AS p LEFT OUTER JOIN
authors AS a
  ON p.city = a.city
```

au_fname	au_lname	pub_name
Klee	Hull	Core Dump Books
Christian	Kells	Abatis Publishers
NULL	*NULL*	Schadenfreude Press
NULL	*NULL*	Tenterhooks Press

au_fname	au_lname	pub_name
Klee	Hull	Core Dump Books
Christian	Kells	Abatis Publishers
NULL	NULL	Schadenfreude Press
NULL	NULL	Tenterhooks Press