#### Introduction to Data Management CSE 344

Lecture 4: Joins and Aggregates

#### Announcements

- HW1 is due tonight at 11pm
- WQ2 is out
- HW2 is out
  - Write queries using real-world dataset
  - Due in 1 week
- Sections tomorrow
  - Please go to your assigned sections

# Today

- Inner joins (6.2)
- Outer joins (6.3.8)
- Aggregations (6.4.3 6.4.6)
- Examples, examples, examples...

# Our SQL Toolchest

- Selection
- Projection
- Ordering and distinct
- Inner Join
- Outer Join

Product(pname, price, category, manufacturer)
Company(cname, country)

-- manufacturer is foreign key to Company

SELECT DISTINCT cname
FROM Product, Company
WHERE country='USA' AND category = 'gadget'
AND manufacturer = cname

SELECT DISTINCT cname
FROM Product, Company
WHERE country='USA' AND category = 'gadget'
AND manufacturer = cname

#### Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

SELECT DISTINCT cname
FROM Product, Company
WHERE country='USA' AND category = 'gadget'
AND manufacturer = cname

#### Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

SELECT DISTINCT cname
FROM Product, Company
WHERE country='USA' AND category = 'gadget'
AND manufacturer = cname

#### Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

SELECT DISTINCT cname
FROM Product, Company
WHERE country='USA' AND category = 'gadget'
AND manufacturer = cname

#### Product

	pname	category	manufacturer		_	_	cna	ame	CO	untry		
	Gizmo	gadget	GizmoWorks		GizmoWorks				Gizmo	GizmoWorks		JSA
(	Camera	Photo	Hitachi			-	Ca	Canon		apan		
C	DneClick	Photo	Hitachi				Hit	achi	Ja	apan		
	pname	category	category		rer	CNa	ame	cour	ntry			
	Gizmo	no gadget		GizmoWor	ks	Gizmo	oWorks	US	A	9		

SELECT DISTINCT cname
FROM Product, Company
WHERE country='USA' AND category = 'gadget'
AND manufacturer = cname

#### Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

SELECT DISTINCT cname
FROM Product, Company
WHERE country='USA' AND category = 'gadget'
AND manufacturer = cname

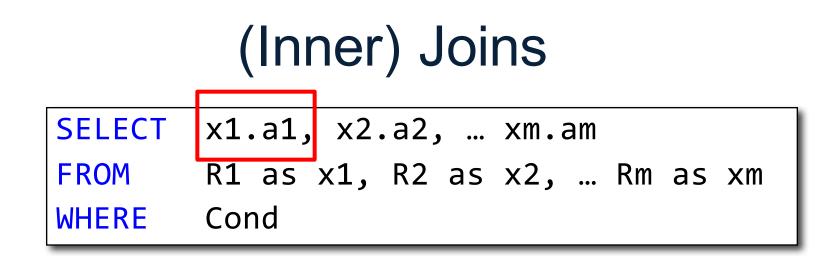
#### Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

SELECT DISTINCT cname
FROM Product, Company
WHERE country='USA' AND category = 'gadget'
AND manufacturer = cname

SELECT	DISTINCT cname
FROM	Product JOIN Company ON country = 'USA' AND category = 'gadget' AND manufacturer = cname



### Self-Joins and Tuple Variables

- Find all companies that manufacture both products in the 'gadgets' and 'photo' category
- Joining Product with Company is insufficient: need to join Product, with Product, and with Company
- When a relation occurs twice in the FROM clause we call it a *self-join* 
  - in that case we must use tuple variables (why?)

SELECT DISTINCT z.cname
FROM Product x, Product y, Company z
WHERE z.country = 'USA'

AND x.category = 'gadget' AND y.category = 'photo'

AND x.manufacturer = cname

AND y.manufacturer = cname;

#### Product

pname	category	manufacturer	
Gizmo	gadget	GizmoWorks	
SingleTouch	photo	Hitachi	
MultiTouch	Photo	GizmoWorks	

cname	country
GizmoWorks	USA
Hitachi	Japan

SELECT DISTINCT z.cname

**FROM** Product x, Product y, Company z

WHERE z.country = 'USA' AND x.category = 'gadget' AND y.category = 'photo' AND x.manufacturer = cname AND y.manufacturer = cname;

#### Product

X

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
SingleTouch	photo	Hitachi
MultiTouch	Photo	GizmoWorks

cname	country
GizmoWorks	USA
Hitachi	Japan

SELECT DISTINCT z.cname
FROM Product x, Product y, Company z
WHERE z.country = 'USA'
AND x.category = 'gadget'
AND y.category = 'photo'

AND x.manufacturer = cname

AND y.manufacturer = cname;

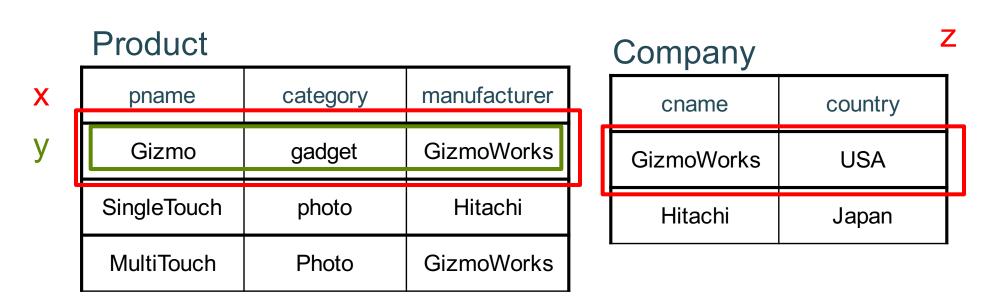
#### Product

x .	pname	category	manufacturer	
у	Gizmo	gadget	GizmoWorks	
	SingleTouch	photo	Hitachi	
	MultiTouch	Photo	GizmoWorks	

cname	country
GizmoWorks	USA
Hitachi	Japan

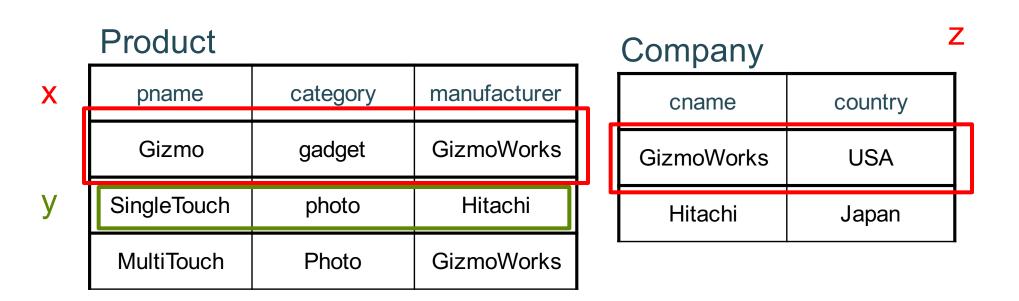
SELECT DISTINCT z.cname
FROM Product x, Product y, Company z
WHERE z.country = 'USA'
AND x.category = 'gadget'
AND y.category = 'photo'
AND x.manufacturer = cname

AND y.manufacturer = cname;



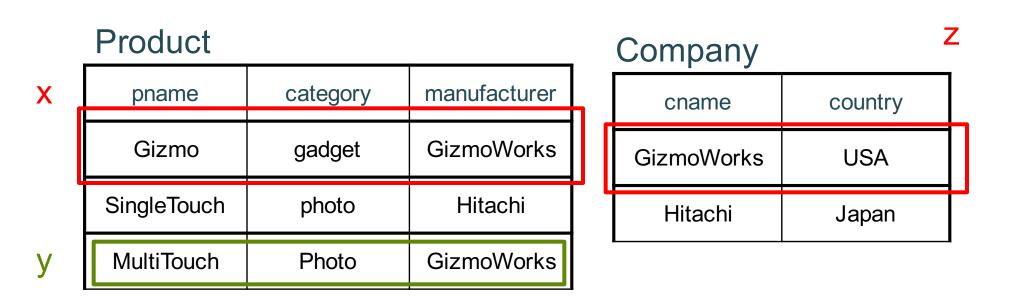
SELECT DISTINCT z.cname

**FROM** Product x, Product y, Company z



SELECT DISTINCT z.cname

**FROM** Product x, Product y, Company z



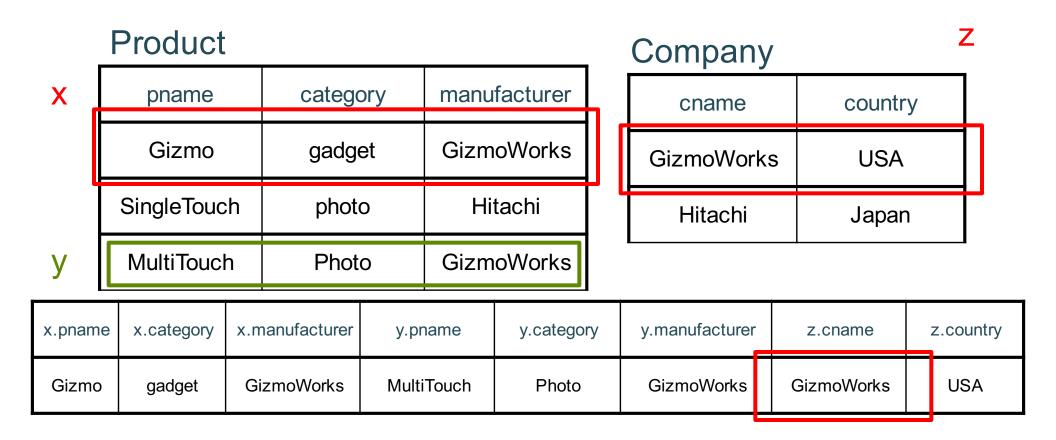
SELECT DISTINCT z.cname

**FROM** Product x, Product y, Company z

	Product							Company		Z	
Х	pname		catego	ory	manu	facturer	7	cname	countr	ſу	
	Gizmo		gadge	et	Gizm	oWorks		GizmoWorks	S USA		
	SingleTouc	h	photo	C	Hi	tachi	- (	Hitachi	Japar	n	-
У	MultiTouc	h	Phote	0	Gizm	oWorks					
x.pnam	e x.category	x.ma	anufacturer	y.pr	name	y.categor	у	y.manufacturer	z.cname	Z.COL	intry
Gizmo	gadget	Giz	zmoWorks	Multi	Touch	Photo		GizmoWorks	GizmoWorks	US	A

SELECT DISTINCT z.cname

**FROM** Product x, Product y, Company z



# Outer joins

Product(<u>name</u>, category)
Purchase(prodName, store)

-- prodName is foreign key

SELECT	Product.name, Purchase.store
FROM	Product, Purchase
WHERE	<pre>Product.name = Purchase.prodName</pre>

We want to include products that are never sold, but some are not listed! Why?

# Outer joins

Product(<u>name</u>, category)
Purchase(prodName, store)

-- prodName is foreign key

SELECT Product.name, Purchase.store
FROM Product LEFT OUTER JOIN Purchase ON
Product.name = Purchase.prodName

# SELECT Product.name, Purchase.store FROM Product JOIN Purchase ON Product.name = Purchase.prodName

#### Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

#### Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

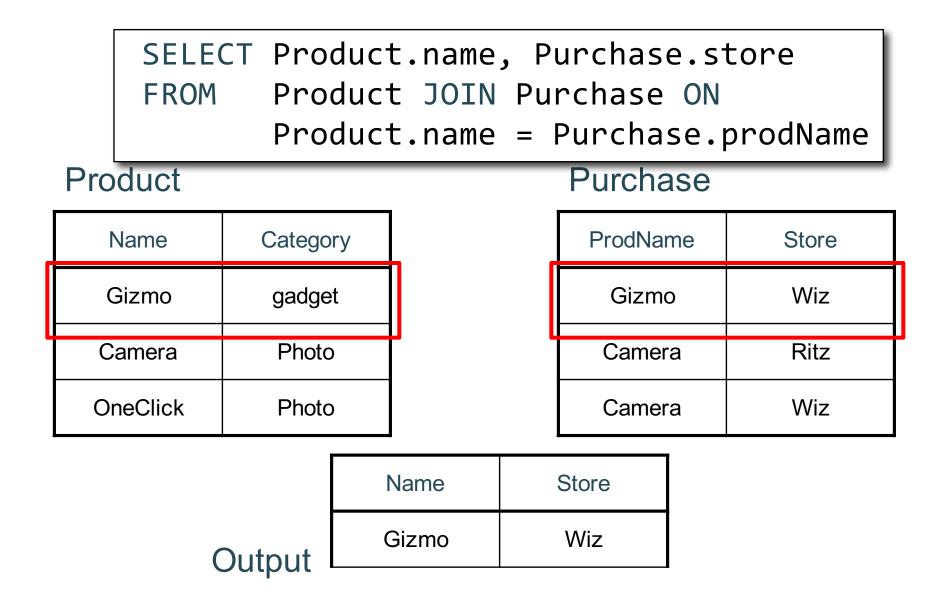
# SELECT Product.name, Purchase.store FROM Product JOIN Purchase ON Product.name = Purchase.prodName

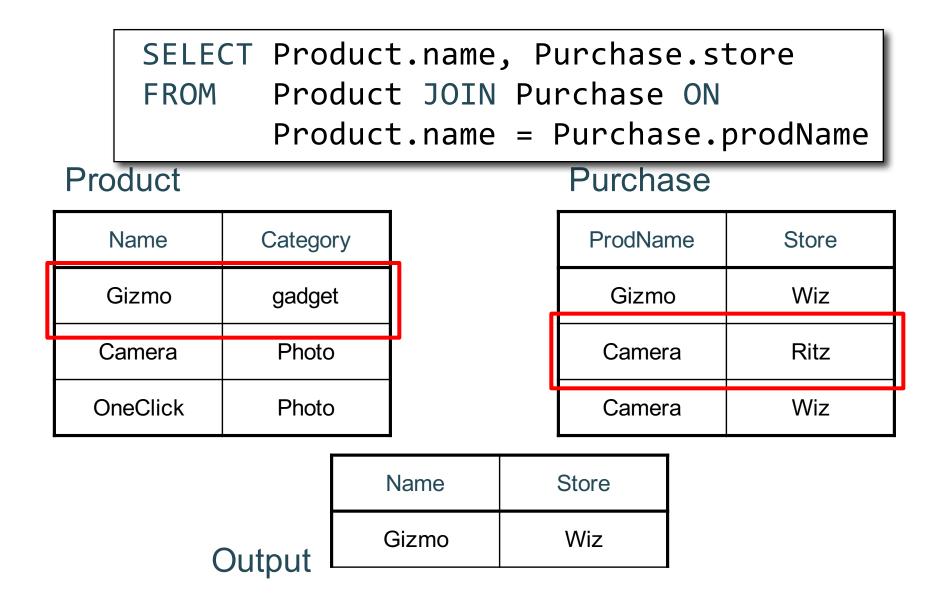
#### Product

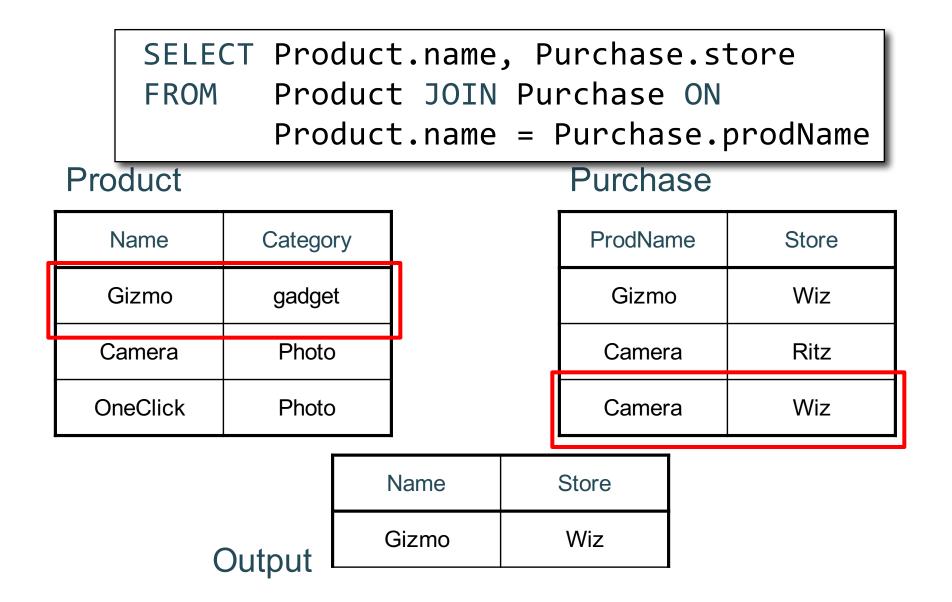
Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

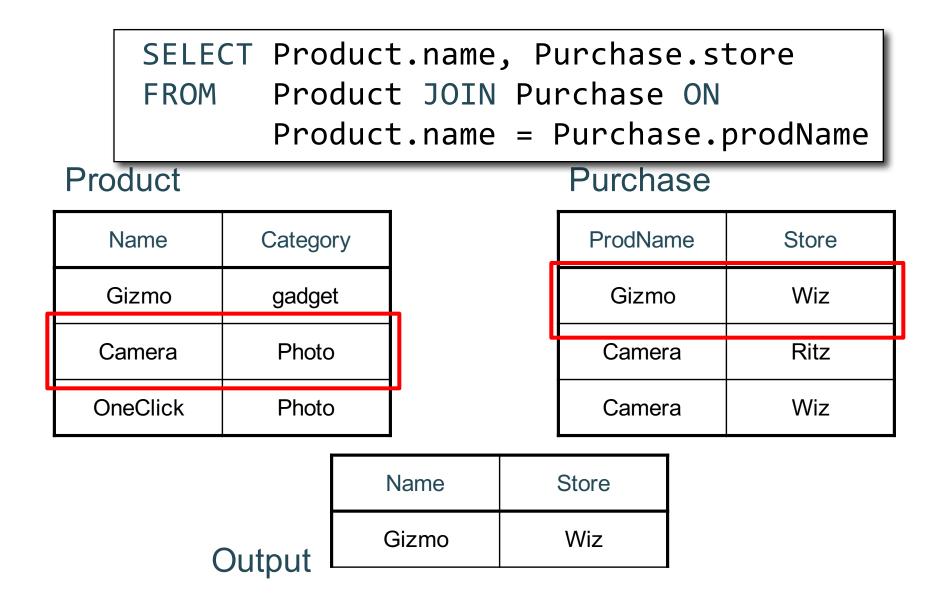
#### Purchase

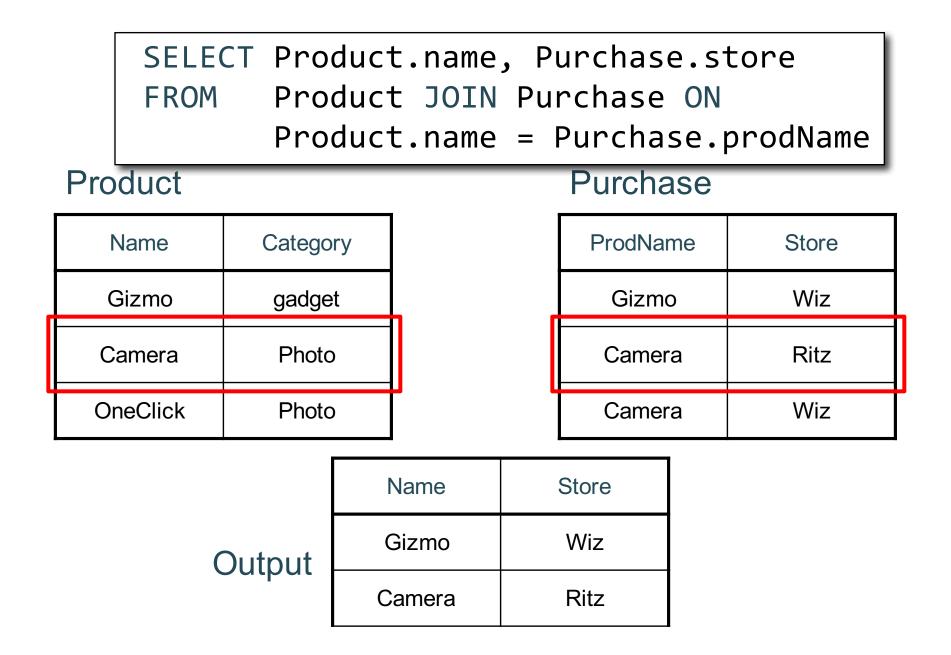
ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

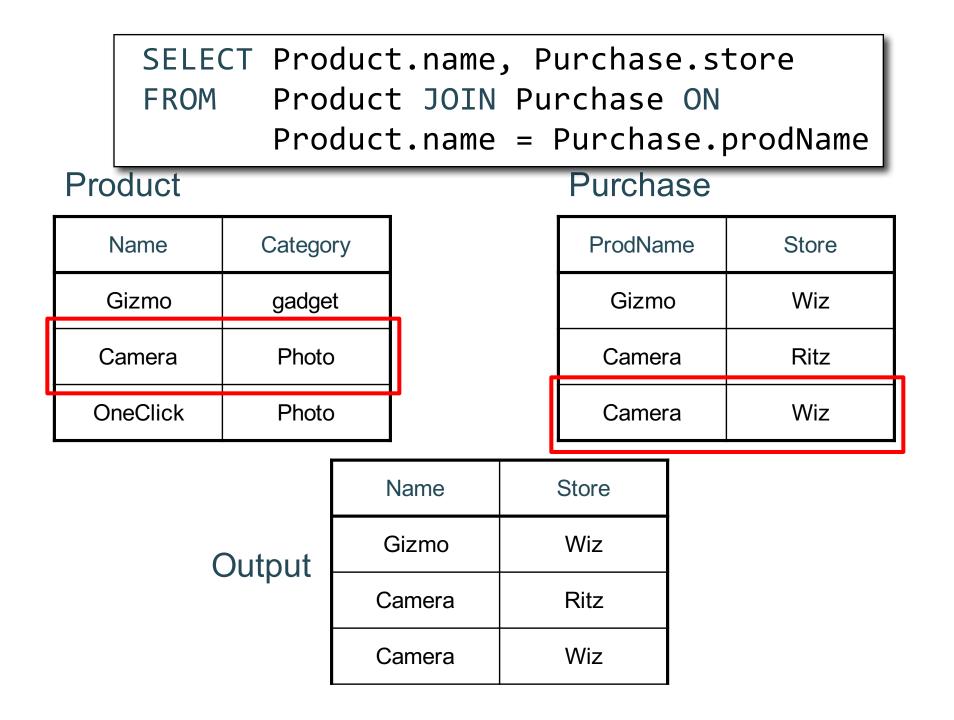


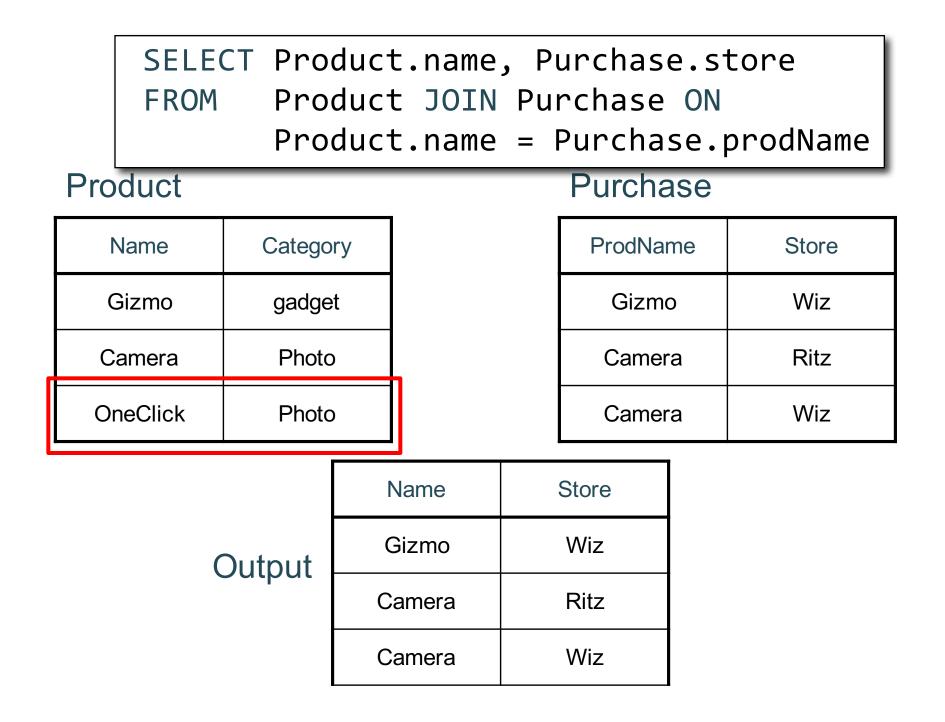


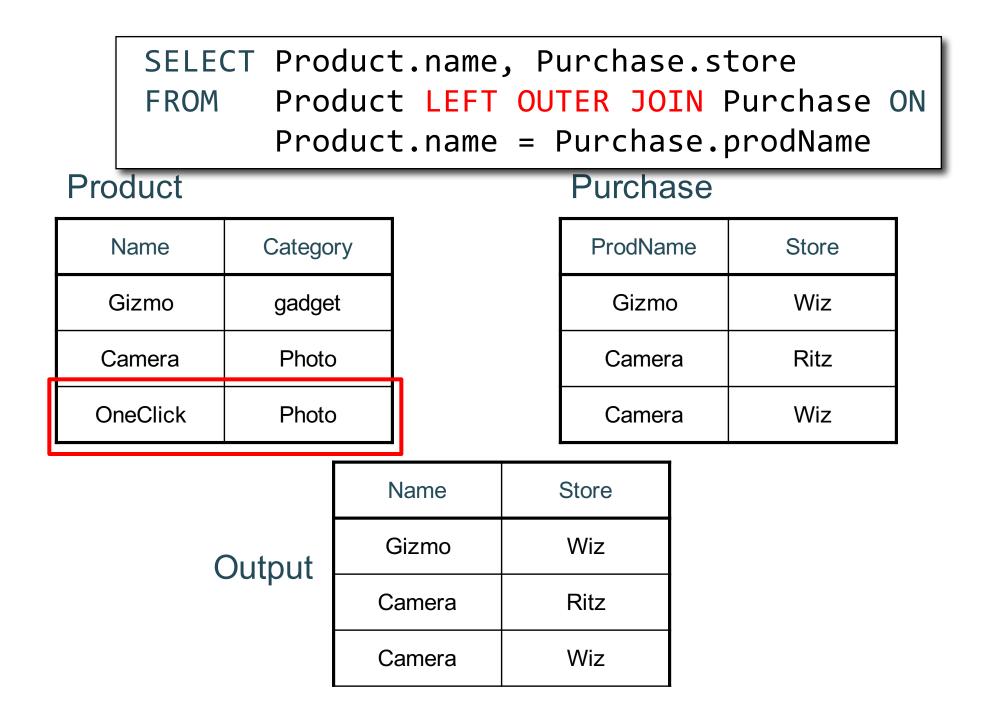


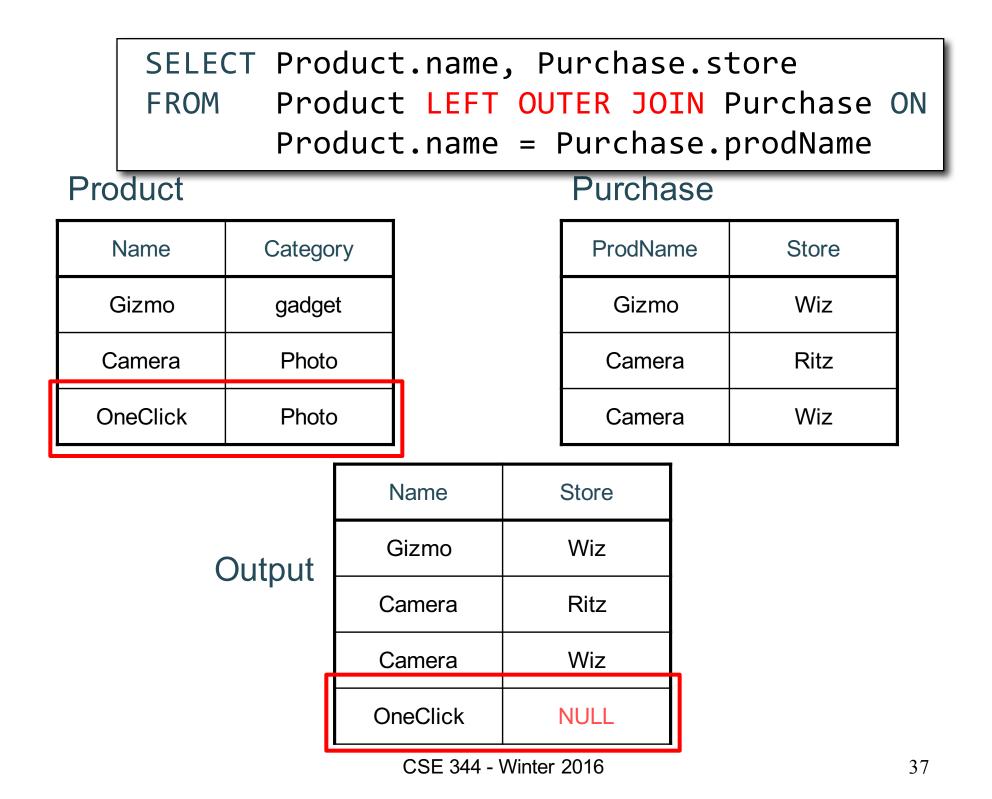












SELECT Product.name, Purchase.store FROM Product FULL OUTER JOIN Purchase O Product.name = Purchase.prodName					
Product Purchase					
Name	Category		ProdName	Store	]
Gizmo	gadget	_	Gizmo	Wiz	
Camera	Photo		Camera	Ritz	
OneClick	Photo		Camera	Wiz	
Output	Name	Store	Phone	Foo	1
	Gizmo	Wiz			-
	Camera	Ritz			
	Camera	Wiz			
	OneClick	NULL			
	NULL	Foo			38

### **Outer Joins**

tableA (LEFT/RIGHT/FULL) OUTER JOIN tableB ON p

- Left outer join:
  - Include tuples from tableA even if no match
- Right outer join:
  - Include tuples from tableB even if no match
- Full outer join:
  - Include tuples from both even if no match
- In all cases:
  - Patch tuples without matches using NULL

# Simple Aggregations

Five basic aggregate operations in SQL

select count(\*) from Purchase
select sum(quantity) from Purchase
select avg(price) from Purchase
select max(quantity) from Purchase
select min(quantity) from Purchase

Except count, all aggregations apply to a single attribute

CSE 344 - Fall 2016

## Aggregates and NULL Values

Null values are not used in aggregates
 insert into Purchase
 values(12, 'gadget', NULL, NULL, 'april')

Let's try the following

select count(\*) from Purchase
select count(quantity) from Purchase

```
select sum(quantity) from Purchase
```

```
select count(*)
from Purchase
where quantity is not null;
```

# **Counting Duplicates**

COUNT applies to duplicates, unless otherwise stated:

SELECT	<pre>count(product)</pre>		
FROM	Purchase		
WHERE	price > 4.99		

same as count(\*) if no nulls

We probably want:

SELECT	<pre>count(DISTINCT product)</pre>
FROM	Purchase
WHERE	price > 4.99