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 this week
- Programming style

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	ma	nufacturer	_	_	cna	ame	CO	untry	
	Gizmo	gadget	GizmoWorks				Gizmo	GizmoWorks		JSA	
(Camera	Photo	Hitachi			-	Ca	Canon		Japan	
C	DneClick	Photo	Hitachi				Hit	achi	Ja	apan	
	pname	category	/	manufactu	rer	CNa	ame	cour	ntry		
	Gizmo	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

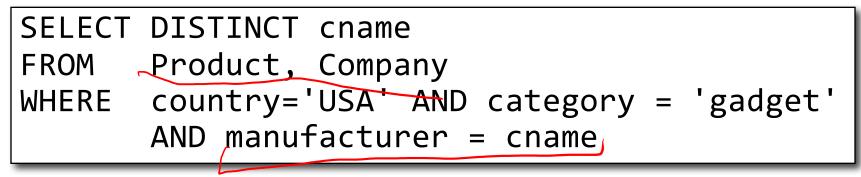
cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

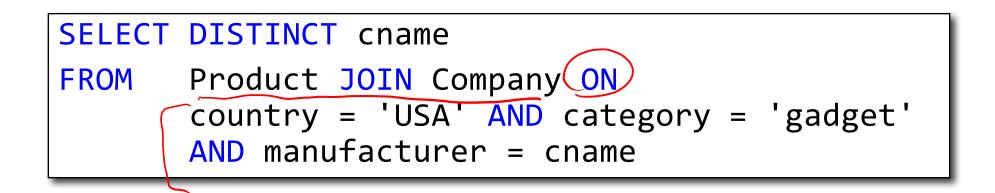
SELECT DISTINCT cname
FROM Product, Company
WHERE country='USA' AND category = 'gadget'
AND manufacturer = cname

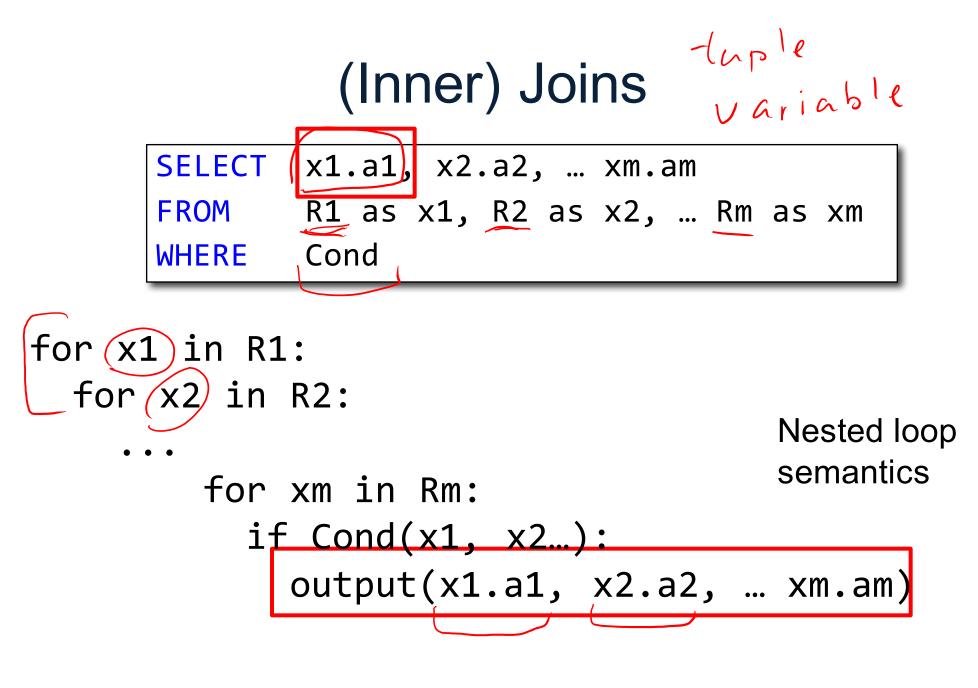
Product

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

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan







CSE 344 - Fall 2016

Playtime!

Product(pname, price, category, manufacturer)
Company(cname, country)
-- manufacturer is foreign key to Company

Retrieve all Japanese companies that manufacture products in both 'gadget' and 'photography' categories

Playtime!

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

-- manufacturer is foreign key to Company

Retrieve all Japanese companies that manufacture products in both 'gadget' and 'photography' categories

```
SELECT DISTINCT cname
FROM Product P1, Product P2, Company
WHERE country = 'Japan' AND P1.category = 'gadget'
AND P2.category = 'photography'
AND P1.manufacturer = cname
AND P2.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
ľ	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;

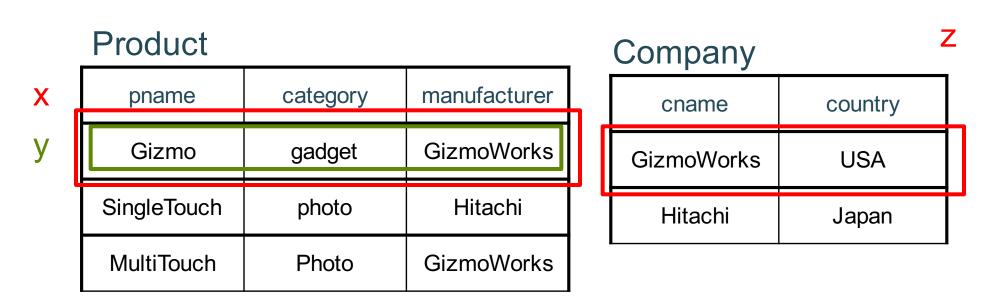
Product

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

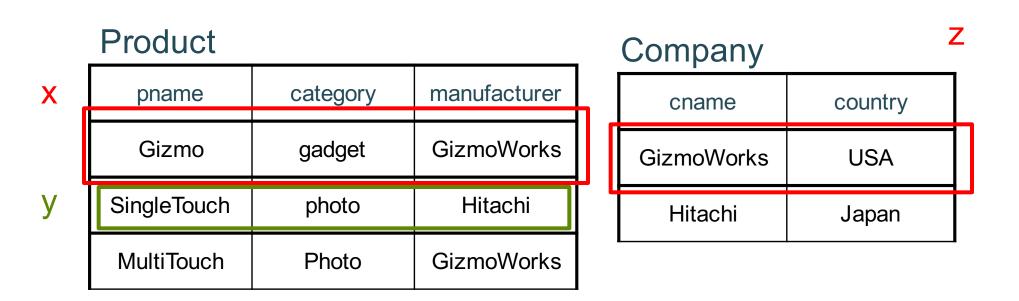
AND y.manufacturer = cname;



SELECT DISTINCT z.cname

FROM Product x, Product y, Company z

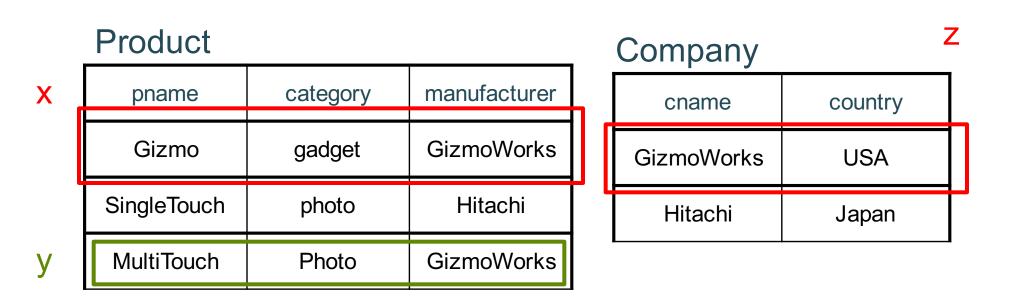
WHERE z.country = 'USA' AND x.category = 'gadget' AND y.category = 'photo' AND x.manufacturer = cname AND y.manufacturer = cname;

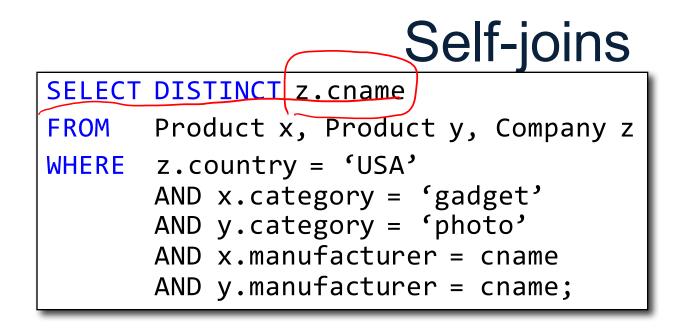


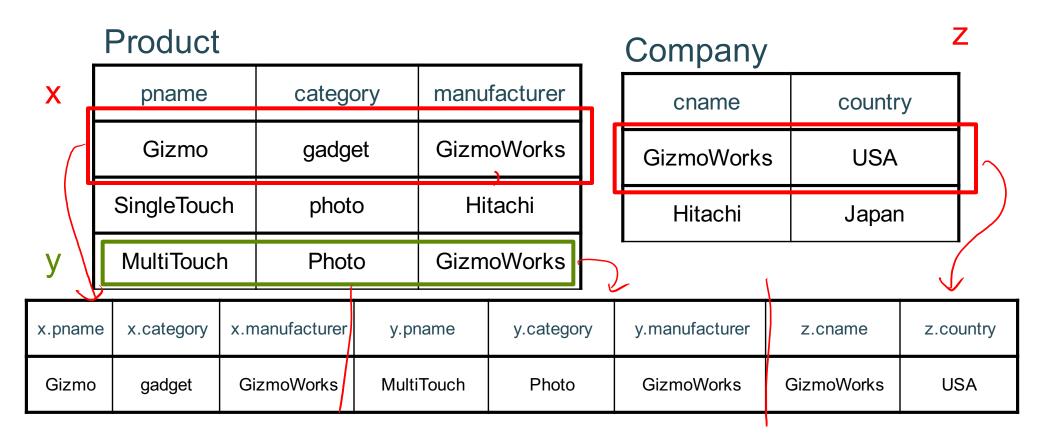
SELECT DISTINCT z.cname

FROM Product x, Product y, Company z

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

	Ρ	roduct							Company	y			Ζ	
Х		pname		catego	tegory		facturer	1	cname		country			
		Gizmo		gadge	et	Gizm	oWorks] [GizmoWorks		USA			
	5	SingleTouc	h	phote	0	Hitac		- 1	Hitachi		Japan			1
у		MultiTouch	n	Phote	0	o Gizmo ^v								
x.pname x.category x.manufacturer y.pna		name	y.categor	у	y.manufacturer		z.cname	Z	z.coun	ıtry				
Gizmo		gadget	Gi	zmoWorks	Multi	Touch	Photo		GizmoWorks	(GizmoWorks		USA	

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

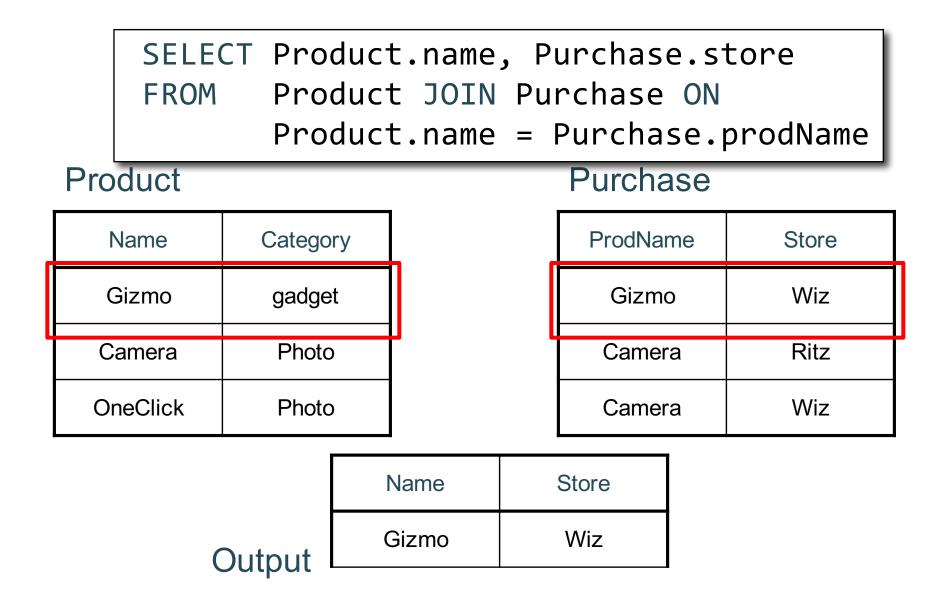
SELECT Product.name, Purchase.store FROM Product JOIN Purchase ON Product.name = Purchase.prodName

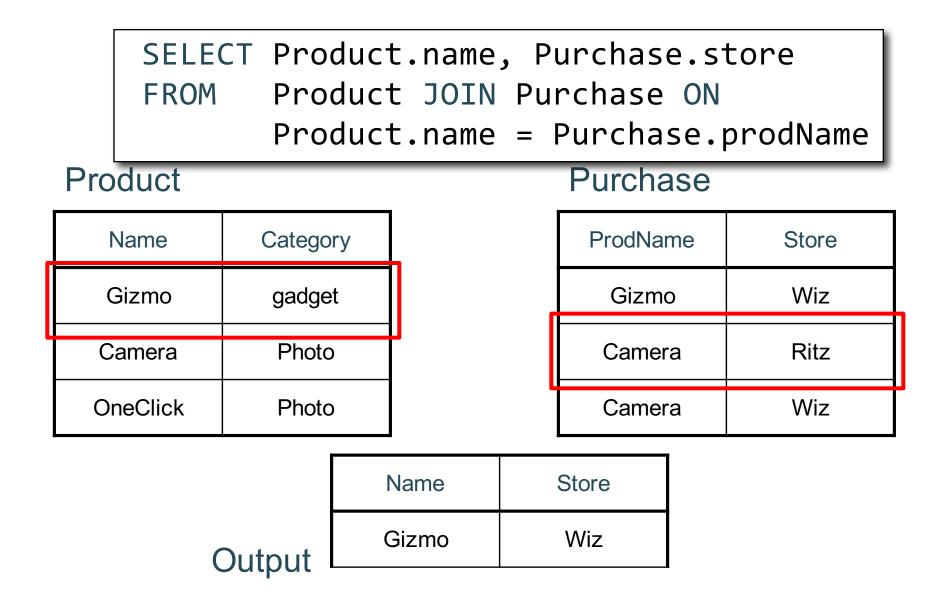
Product

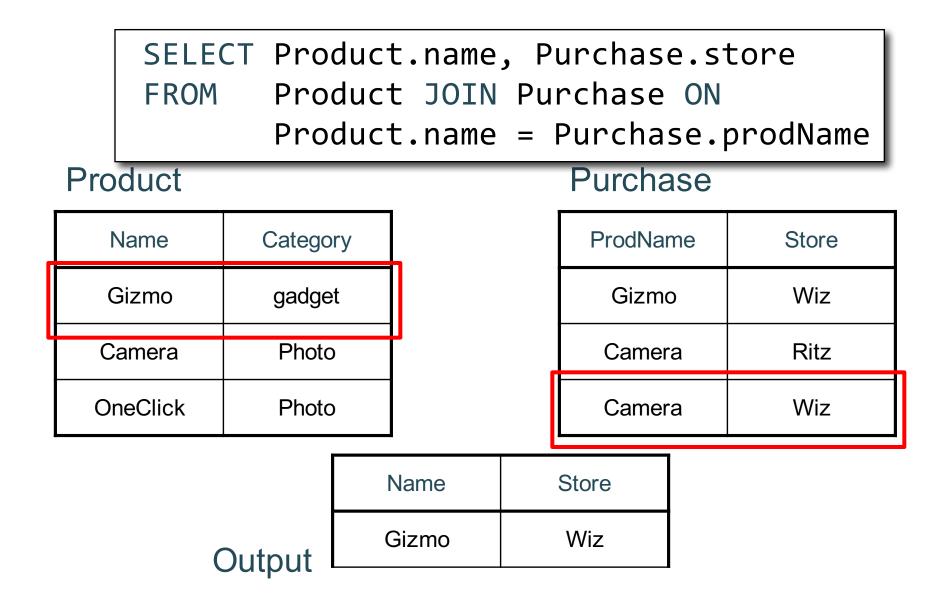
Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

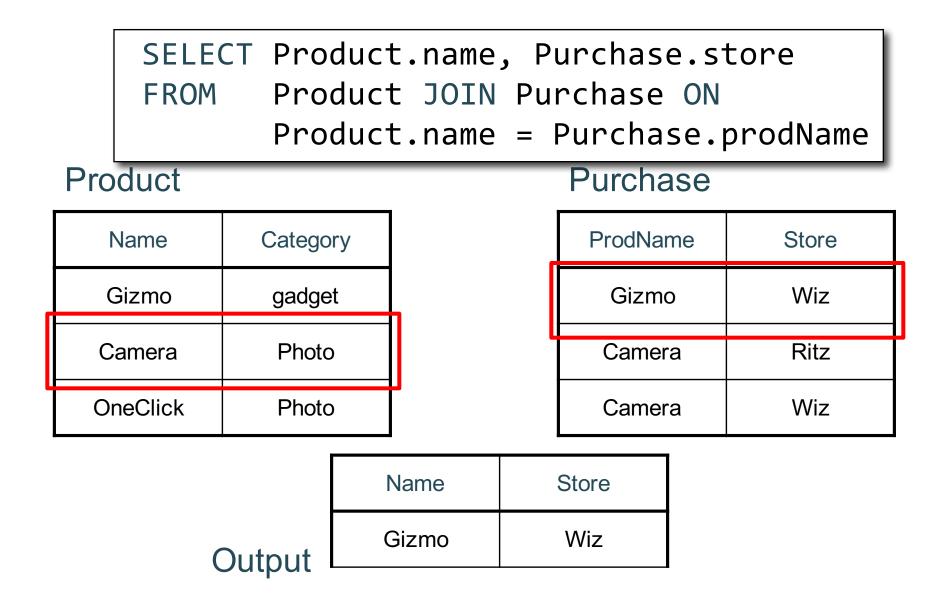
Purchase

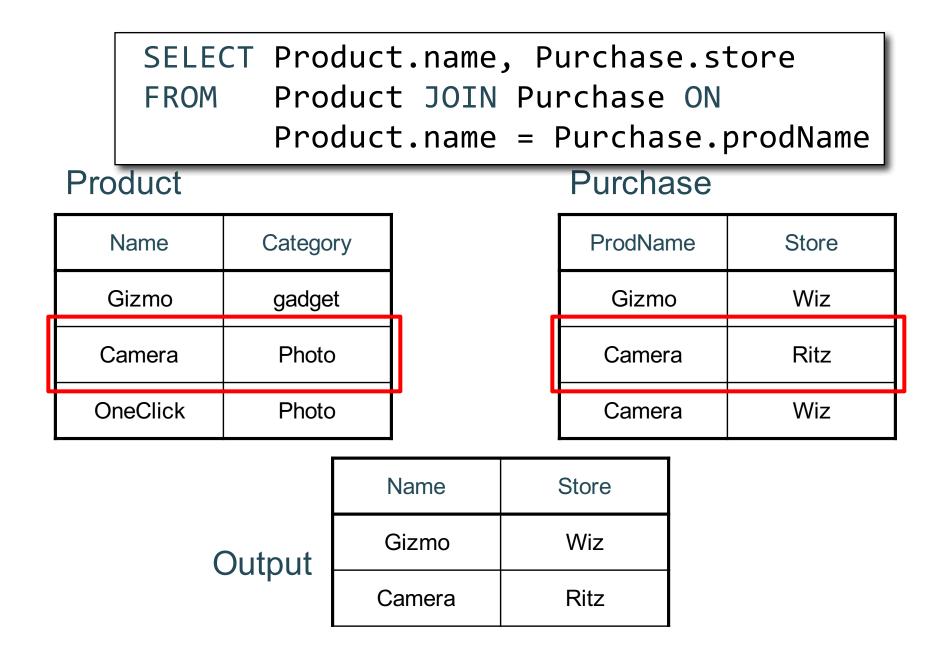
ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

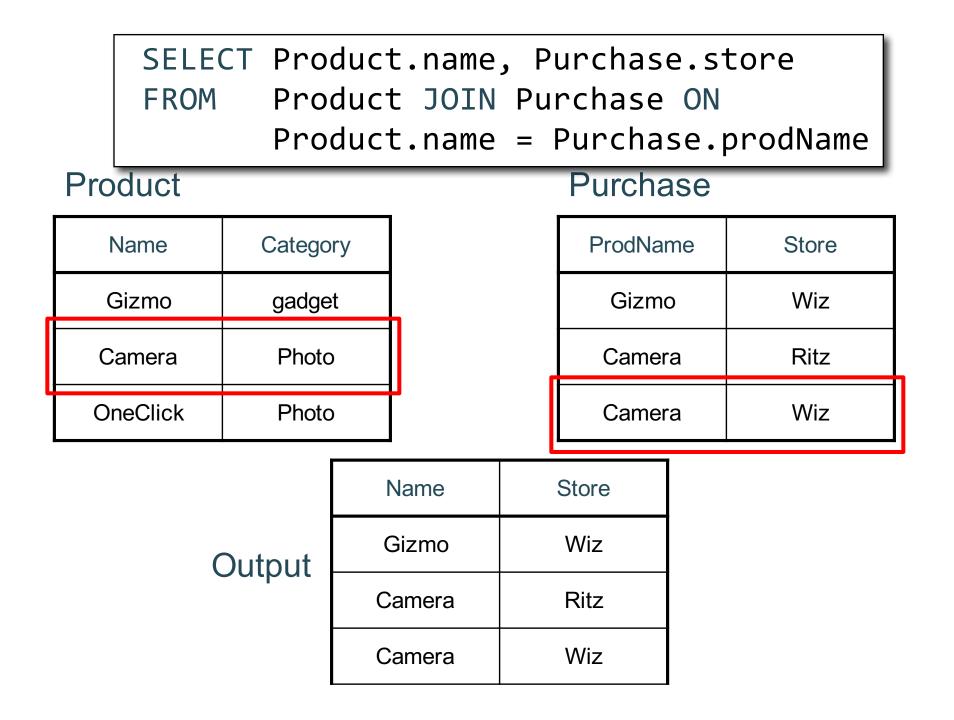


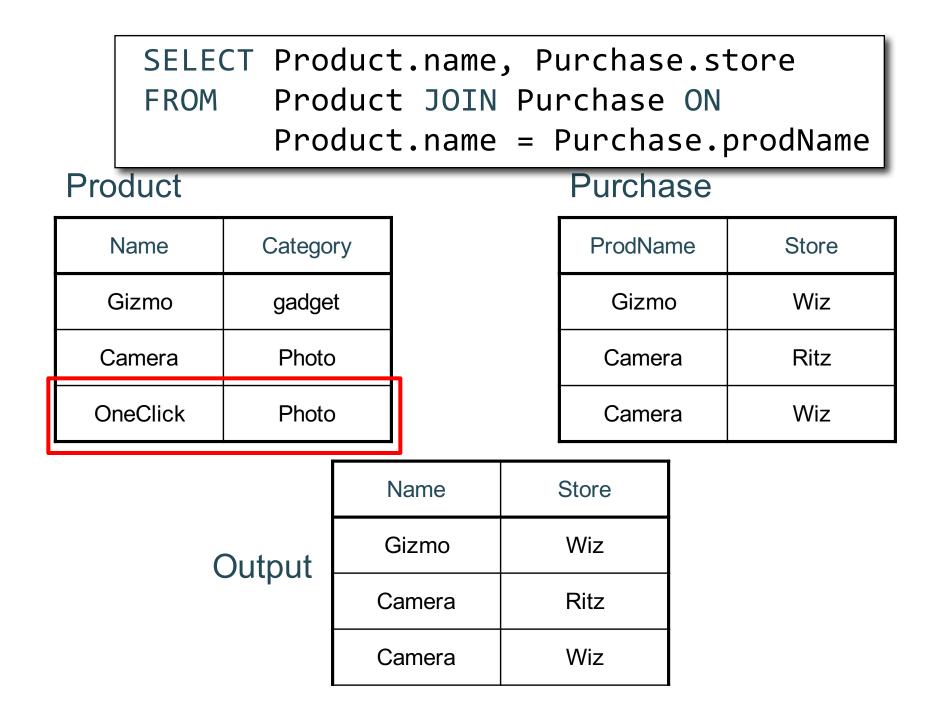


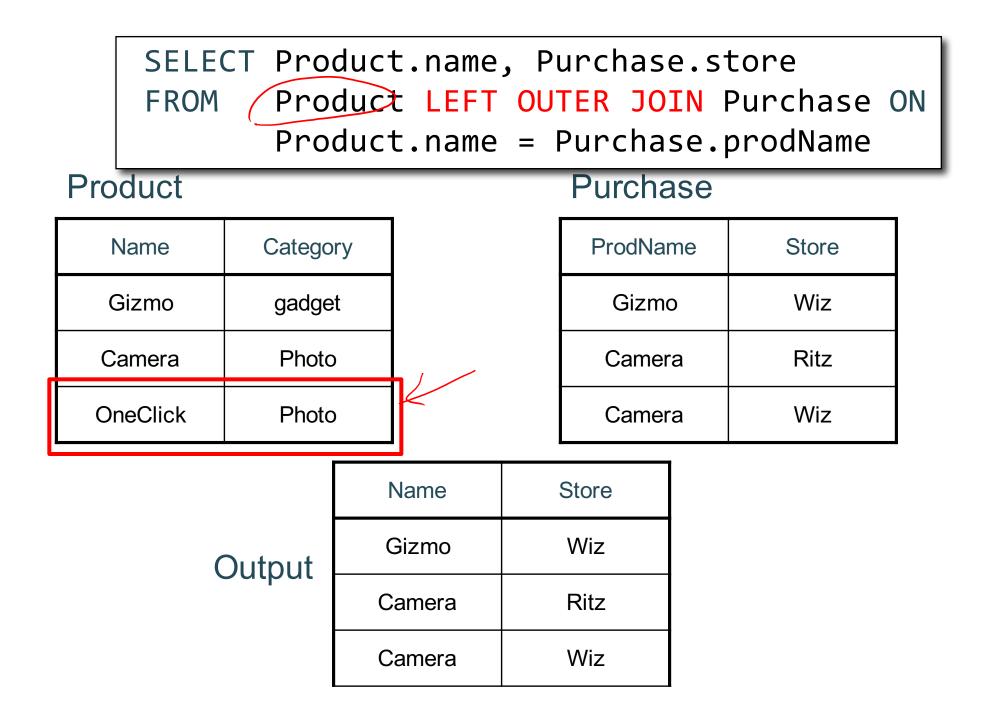


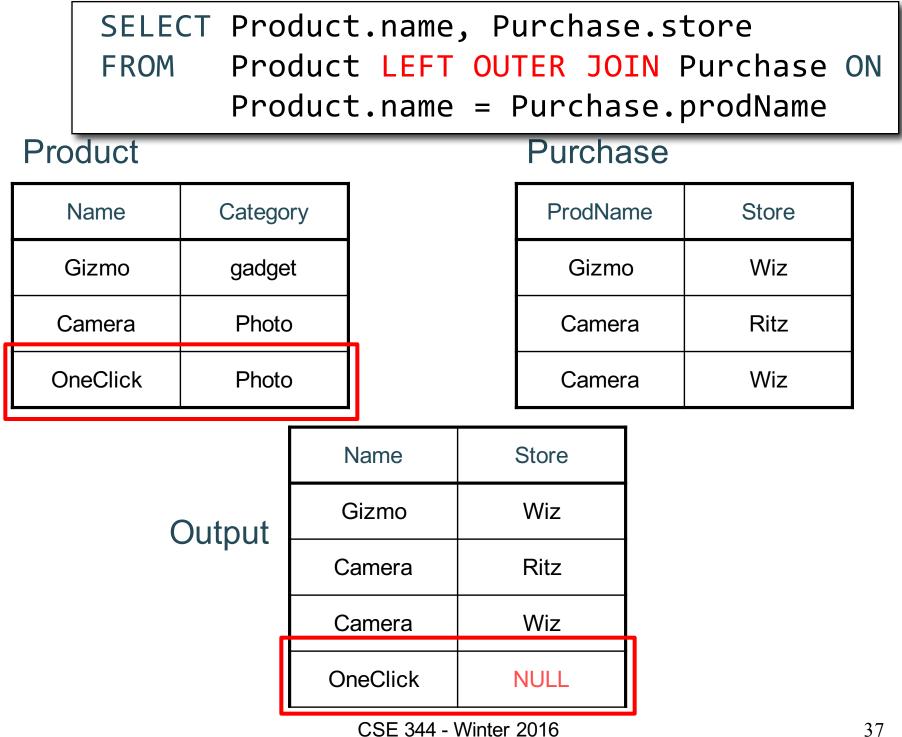












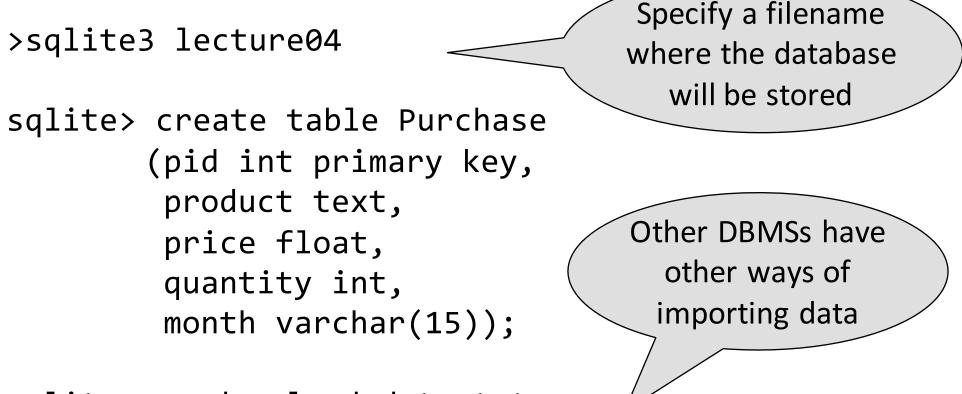
SELECT Product.name, Purchase.store FROM Product FULL OUTER JOIN Purchase O Product.name = Purchase.prodName								
Product	Product Purchase							
Name	Category		ProdName	Store]			
Gizmo	gadget		Gizmo	Wiz				
Camera	Photo		Camera	Ritz				
OneClick	Photo		Camera	Wiz				
	Name	Store	Phone	Foo	1			
	Gizmo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

Loading Data into SQLite



sqlite> -- download data.txt // sqlite> .import lec04-data.txt Purchase

Comment about SQLite

- Cannot load NULL values such that they are actually loaded as null values
- So we need to use two steps:
 - Load null values using some type of special value
 - Update the special values to actual null values

```
update Purchase
  set price = null
  where price = '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

More Examples

SELECTSum(price * quantity)FROMPurchase

SELECT Sum(price * quantity) FROM Purchase WHERE product = 'bagel'

