

# CCT395, Week 3

## SQL Queries Using Multiple Tables

Yuri Takhteyev  
University of Toronto  
September 22, 2010



This presentation is licensed under Creative Commons Attribution License, v. 3.0. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/>. This presentation incorporates images from the Crystal Clear icon collection by Everaldo Coelho, available under LGPL from <http://everaldo.com/crystal/>.

The Books Are In

# Assignment 1 Is Out

- Available on the course website
- Due at noon on October 6
  - 3% late penalty per day (full or part)
  - not accepted if more than 7 days late
  - late assignments may be graded with a substantial delay

# Basic Requirements

- **Printed** → If there is a good reason why you cannot come to class, submit the assignment by email as **PDF** or **plain text** and then bring a paper copy with a documented explanation of why you could not submit the paper copy on time.
- **Stapled**
- **Your name in top-right corner**
- **Normal text in 12 point serif font**
- **SQL code in a monospaced font**

# The Database Server

- **Appropriate Use Policy**

<http://www.provost.utoronto.ca/policy/use.htm>

- Respect for others
  - Sharing the resources
  - Intellectual property
- **A temporary resource**
    - Will be deleted when the course ends

# Please Test It **Now**

No extensions for  
failure to connect

# Projection



name	owner	species	sex	birth
Fluffy	Harold	cat	f	1993-02-04
Bluffy	Harold	dog	f	1989-05-13
Chirpy	Gwen	bird	f	1998-09-11

# Projection

name	species	sex
Fluffy	cat	f
Bluffy	dog	f
Chirpy	bird	f



# Selection

("Restriction" in Harrington)



name	owner	species	sex	birth
Fluffy	Harold	cat	f	1993-02-04
Bluffy	Harold	dog	f	1989-05-13
Chirpy	Gwen	bird	f	1998-09-11

projection

```
select name, species  
from pet
```

```
where weight < 1;
```

selection

# Also:

- Ordering the results
- Limit
- Aggregation (count, sum, avg)
- Grouping

# Multiple Tables

pet

name	owner	species
Fluffy	Harold	cat
Bluffy	Harold	dog
Chirpy	Gwen	bird

species

name	food
dog	dog food
bird	seeds
cat	cat food

# Join

name	owner	species	food
Fluffy	Harold	cat	cat food
Bluffy	Harold	dog	dog food
Chirpy	Gwen	bird	seeds

# Cartesian Product

$$\left\{ \begin{array}{l} \text{Fluffy} \\ \text{Buffy} \\ \text{Chirpy} \end{array} \right\} \times \left\{ \begin{array}{l} \text{dog} \\ \text{cat} \\ \text{bird} \end{array} \right\} = \left\{ \begin{array}{l} (\text{Fluffy}, \text{dog}) \\ (\text{Fluffy}, \text{cat}) \\ (\text{Fluffy}, \text{bird}) \\ (\text{Buffy}, \text{dog}) \\ (\text{Buffy}, \text{cat}) \\ (\text{Buffy}, \text{bird}) \\ (\text{Chirpy}, \text{dog}) \\ (\text{Chirpy}, \text{cat}) \\ (\text{Chirpy}, \text{bird}) \end{array} \right\}$$

# Product of Tables

pet

name	owner	species
Fluffy	Harold	cat
Bluffy	Harold	dog
Chirpy	Gwen	bird

×

species

name	food
dog	dog food
bird	seeds
cat	cat food

name	owner	species	species	food
Fluffy	Harold	cat	cat	cat food
Bluffy	Harold	dog	cat	cat food
Chirpy	Gwen	bird	cat	cat food
Fluffy	Harold	cat	dog	dog food
Bluffy	Harold	dog	dog	dog food
Chirpy	Gwen	bird	dog	dog food
Fluffy	Harold	cat	bird	seeds
Bluffy	Harold	dog	bird	seeds
Chirpy	Gwen	bird	bird	seeds



name	owner	species	species	food
Fluffy	Harold	cat	cat	cat food
Bluffy	Harold	dog	cat	cat food
Chirpy	Gwen	bird	cat	cat food
Fluffy	Harold	cat	dog	dog food
Bluffy	Harold	dog	dog	dog food
Chirpy	Gwen	bird	dog	dog food
Fluffy	Harold	cat	bird	seeds
Bluffy	Harold	dog	bird	seeds
Chirpy	Gwen	bird	bird	seeds

name	owner	species	species	food
Fluffy	Harold	cat	cat	cat food
Bluffy	Harold	dog	cat	cat food
Chirpy	Gwen	bird	cat	cat food
Fluffy	Harold	cat	dog	dog food
Bluffy	Harold	dog	dog	dog food
Chirpy	Gwen	bird	dog	dog food
Fluffy	Harold	cat	bird	seeds
Bluffy	Harold	dog	bird	seeds
Chirpy	Gwen	bird	bird	seeds

cartesian product + selection  
=  
relational join

selection based on equality

↓  
"equi-join"

# SQL-92 Inner Join

(aka "ANSI Join")

```
select ... from <table1>  
join <table2> on <conditions>;
```

For instance:

```
select pet.name, species.food  
from pet join species  
on pet.species = species.name;
```

# SQL-92 Inner Join

```
select ... from <table1>  
join <table2> on <conditions>;
```

For instance:

```
select pet.name, species.food  
from pet join species  
on pet.species = species.name;
```

```

+-----+-----+
| name   | food   |
+-----+-----+
| Fluffy | cat food |
| Claws  | cat food |
| Buffy  | dog food |
| Fang   | dog food |
| Bowser | dog food |
| Chirpy | seeds   |
| Whistler | seeds  |
| Slim   | mice   |
+-----+-----+
8 rows in set (0.00 sec)

```

```

+-----+-----+
| name   | food   |
+-----+-----+
| Fluffy | cat food |
| Fluffy | dog food |
| Fluffy | seeds   |
| Fluffy | mice   |
| Claws  | cat food |
| Claws  | dog food |
| Claws  | seeds   |
| Claws  | mice   |
| Buffy  | cat food |
...
| Slim   | cat food |
| Slim   | dog food |
| Slim   | seeds   |
| Slim   | mice   |
| Puffball | cat food |
| Puffball | dog food |
| Puffball | seeds   |
| Puffball | mice   |
+-----+-----+
36 rows in set (0.00 sec)

```

without the "on" clause →  
 (depends on the db)

# Table Aliases

```
select pet.name, species.food  
from pet join species  
on pet.species = species.name;
```



```
select p.name, s.food  
from pet as p join species as s  
on p.species = s.name;
```

# Self-Join

```
select ... from <table> <alias1>  
join <table> <alias1>  
on <conditions>;
```

For instance:

```
select p1.name, p2.name  
from pet as p1 join pet as p2  
on p1.species = p2.species  
where p1.name < p2.name;
```



```
+-----+-----+
| name   | name   |
+-----+-----+
| Claws  | Fluffy |
| Bowser | Buffy  |
| Buffy  | Fang   |
| Bowser | Fang   |
| Chirpy | Whistler |
+-----+-----+
5 rows in set (0.00 sec)
```

# More Tables

```
select ... from <table1>  
join <table2> on <condition1>  
join <table3> on <condition2>;
```

For instance:

```
select owner.name, food from pet  
join species  
  on pet.species = species.name  
join owner  
  on pet.owner = owner.name;
```

```
+-----+-----+
| name   | food   |
+-----+-----+
| Harold | cat food |
| Gwen   | cat food |
| Harold | dog food |
| Diane  | dog food |
| Gwen   | seeds   |
| Gwen   | seeds   |
+-----+-----+
6 rows in set (0.00 sec)
```

# Inner vs. Outer

## **Inner Join:**

only pairs that satisfy the condition

## **Outer Joins:**

includes non-matched rows from one of the tables, or both  
-> "left", "right", "full"

# Why Do Outer Joins?

pet

name	owner
Fluffy	Harold
Bluffy	Harold
Chirpy	Gwen
Fang	Benny

owner

name	telephone
Gwen	16472939823
Harold	14092938489
Diane	552122347849

# An Inner Join

pet join owner on pet.owner=owner.name

name	owner	telephone
Fluffy	Harold	14092938489
Bluffy	Harold	14092938489
Chirpy	Gwen	552122347849

What happened to Fang?

# What We Want

name	owner	telephone
Fluffy	Harold	14092938489
Bluffy	Harold	14092938489
Chirpy	Gwen	552122347849
Fang	Benny	NULL


# Left Outer Join

```
select ... from <table1>  
left outer join <table2>  
on <conditions>;
```

include unmatched  
rows from the **left**  
table

For instance:

```
select pet.name, pet.owner,  
owner.telephone  
from pet left outer join owner  
on pet.owner = owner.name;
```





name	owner	telephone
Fluffy	Harold	14092938489
Claws	Gwen	16472939823
Buffy	Harold	14092938489
Fang	Benny	NULL
Bowser	Diane	552122347849
Chirpy	Gwen	16472939823
Whistler	Gwen	16472939823
Slim	Benny	NULL
Puffball	Diane	552122347849

9 rows in set (0.00 sec)

# Other Outer Joins

## **Right Outer Join:**

unmatched rows from the *right* table

## **Full Outer Join:**

unmatched rows from *both* sides  
(not available in mysql)

# Easier Equi-Joins

pet

name	owner
Fluffy	Harold
Bluffy	Harold
Chirpy	Gwen
Fang	Benny

owner

name	telephone
Gwen	16472939823
Harold	14092938489
Diane	552122347849



# Easier Equi-Joins

pet

pet_name	owner_name
Fluffy	Harold
Bluffy	Harold
Chirpy	Gwen
Fang	Benny

owner

owner_name	telephone
Gwen	16472939823
Harold	14092938489
Diane	552122347849



# Join... Using...

```
select ... from <table1>  
join <table2>  
using (<columns>);
```

For instance:

```
select pet_name, food  
from pet join owner  
on (owner_name);
```

# Yet Easier Equi-Joins

pet

pet_name	owner_name
Fluffy	Harold
Bluffy	Harold
Chirpy	Gwen
Fang	Benny

owner

owner_name	telephone
Gwen	16472939823
Harold	14092938489
Diane	552122347849



# Natural Join

```
select ... from <table1>  
natural join <table2>;
```

For instance:

avoid

```
select pet_name, food  
from pet natural join owner;
```

# Why Avoid It?

implicit selection of columns  
=  
bad idea



# “Traditional” Join

```
select ...  
from <table1>, <table2>  
where <join_conditions>;
```

For instance:

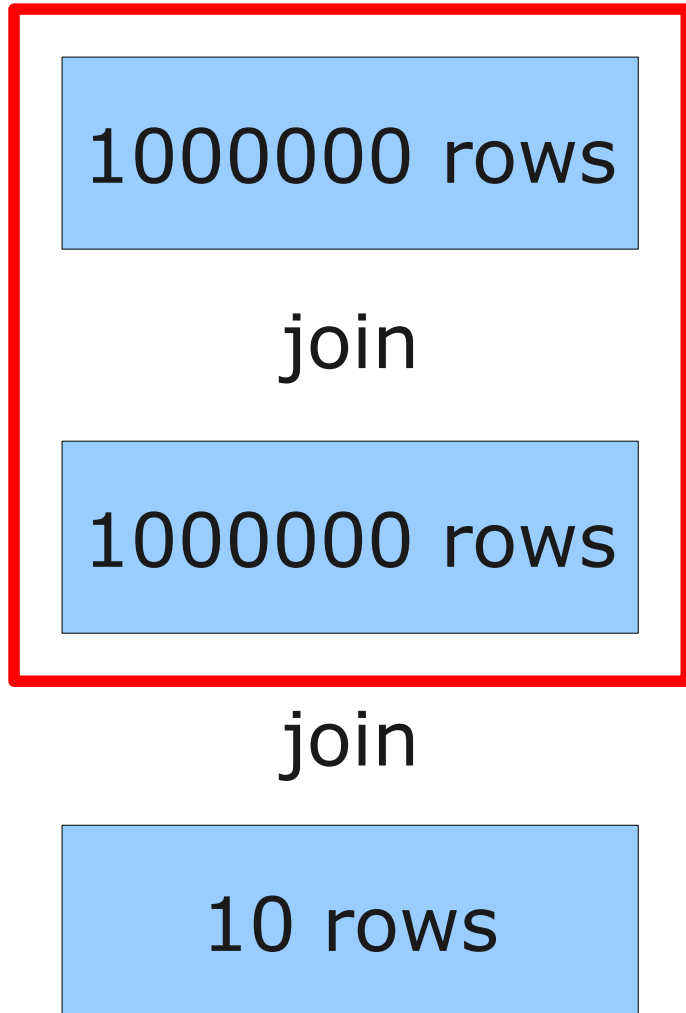
avoid

```
select name, food  
from pet, owner  
where pet.owner=owner.name;
```

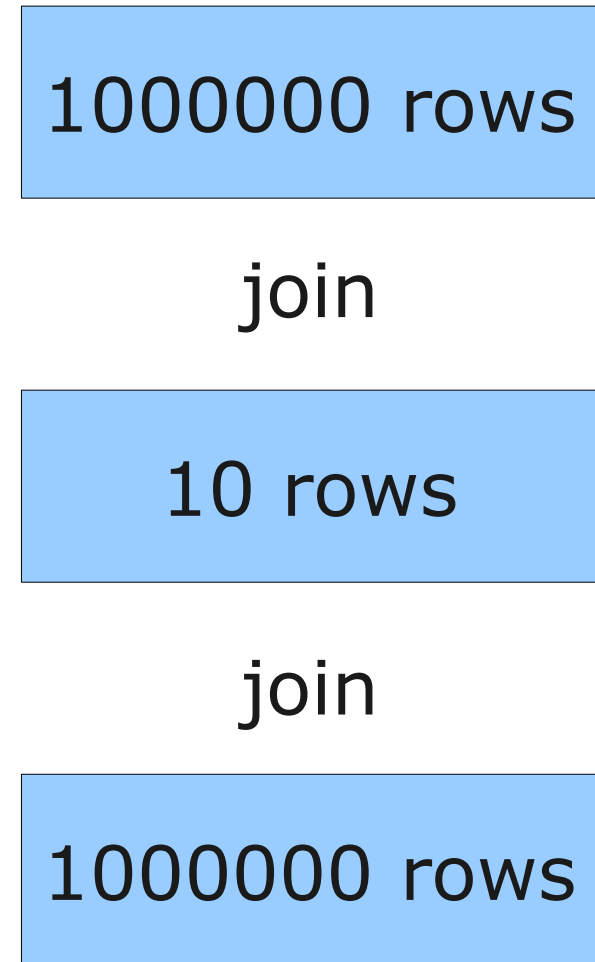
# Use SQL-92 “Join”

- More options
  - e.g., “outer”
- More clear
  - avoids making “where” ambiguous
- Control over the order of joins

# Order of Joins



vs



# Questions about Joins?

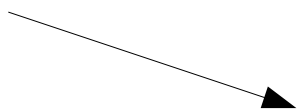
# Subqueries

```
select ... from <some_table>  
where <column> in (<query2>);
```

For instance:


```
select name from pet  
where owner in  
(select name from owner where  
cc_type="visa");
```

```
select name from
owner where
cc_type="visa";
```



name
Gwen
Harold

```
select name
from pet where
owner in (result);
```



name
Fluffy
Claws
Buffy
Chirpy
Whistler

compare with

```
select name from pet
where owner in ("Gwen",
"Harold");
```

# Primary Keys

pet

pet_name	owner_name
Fluffy	Harold
Bluffy	Harold
Chirpy	Gwen
Fang	Benny

owner

owner_name	telephone
Gwen	16472939823
Harold	14092938489
Diane	552122347849

↑  
owner names are  
unique here

foreign key

primary key

pet_name	owner_name
Fluffy	Harold
Bluffy	Harold
Chirpy	Gwen
Fang	Benny

owner_name	telephone
Gwen	16472939823
Harold	14092938489
Diane	552122347849



# Choosing PKs

Usually meaningless IDs are best

# “Harold” is now “Bob”

pet_name	owner_name
Fluffy	Harold
Bluffy	Harold
Chirpy	Gwen
Fang	Benny

owner_name	telephone
Gwen	16472939823
Bob	14092938489
Diane	552122347849

# The Solution

pet_name	owner_id
Fluffy	2
Bluffy	2
Chirpy	1
Fang	???



owner_id	owner_name	telephone
1	Gwen	16472939
2	Harold	14092939
3	Diane	55212234

# The Solution

pet_name	owner_id
Fluffy	2
Bluffy	2
Chirpy	1
Fang	4



owner_id	owner_name	telephone
1	Gwen	16472939
2	Harold	14092939
3	Diane	55212234
4	Benny	NULL

# The Solution

pet_name	owner_id
Fluffy	2
Bluffy	2
Chirpy	1
Fang	4



owner_id	owner_name	telephone
1	Gwen	16472939
2	Bob	14092939
3	Diane	55212234
4	Benny	NULL

Questions?