Intro to SQL

SQL

You will be able to...

- Explain primary and foreign keys
- Use `SELECT`, `FROM`, `WHERE` and `JOIN`
- Explain the differences between `INNER`, `OUTER`, `LEFT` and `RIGHT` joins
- Alias tables (`AS`)
- Use `ORDER BY` and `COUNT`

Example Database

students

id	name	age	gender	address
1	Nick D.	20	M	2
2	Andy D.	28	M	2
3	Beth M.	23	F	1
4	Lisa N.	20	F	4

addresses

id	street	zip	city	state
1	423 Main St.	60647	Chicago	IL
2	13 Main St	60655	Barrington	IL
3	15 Main St	60651	Elsewhere	IL
4	14 Main St	60650	Chicago	IL

Objective: Find all 20-year-old students

students

id	name	age	gender	address
1	Nick D.	20	M	2
2	Andy D.	28	M	2
3	Beth M.	23	F	I
4	Lisa N.	20	F	4

20 Year Old Students
(results)

id	name	age
1	Nick D.	20
4	Lisa N.	20

```
SELECT id, name, age
FROM students
WHERE age = 20;
```

Some Common SQL Keywords

Keyword	Action
SELECT	Which COLUMNS to include in output table (shrinks the result horizontally!)
FROM	Which TABLE to pull data from
JOIN	Another TABLE to glue / concatenate to the output
ON	What COLUMNS must match when joining two tables
WHERE	Which ROWS to include in the output table (shrinks the result vertically!)

students

id	name	age	gender	address
1	Nick D.	20	M	2
2	Andy D.	28	M	2
3	Beth M.	23	F	1
4	Lisa N.	20	F	4

addresses

id	street	zip	city	state
1	423 Main St.	60647	Chicago	IL
2	13 Main St	60655	Barrington	IL
3	15 Main St	60651	Elsewhere	IL
4	14 Main St	60650	Chicago	IL

Objective: Show all students and their addresses

```sql
SELECT students.id, name, street, zip, city
FROM students
JOIN addresses
  ON students.address = addresses.id;
```

Students with Addresses

(results)

student.id	name	street	zip	city
1	Nick D.	13 Main St.	60655	Barrington
2	Andy D.	13 Main St.	60655	Barrington
3	Beth M.	423 Main St.	60647	Chicago
4	Lisa N.	14 Main St.	60650	Chicago

students

id	name	age	gender	address
1	Nick D.	20	M	2
2	Andy D.	28	M	2
3	Beth M.	23	F	1
4	Lisa N.	20	F	4

addresses

id	street	zip	city	state
1	423 Main St.	60647	Chicago	IL
2	13 Main St	60655	Barrington	IL
3	15 Main St	60651	Elsewhere	IL
4	14 Main St	60650	Chicago	IL

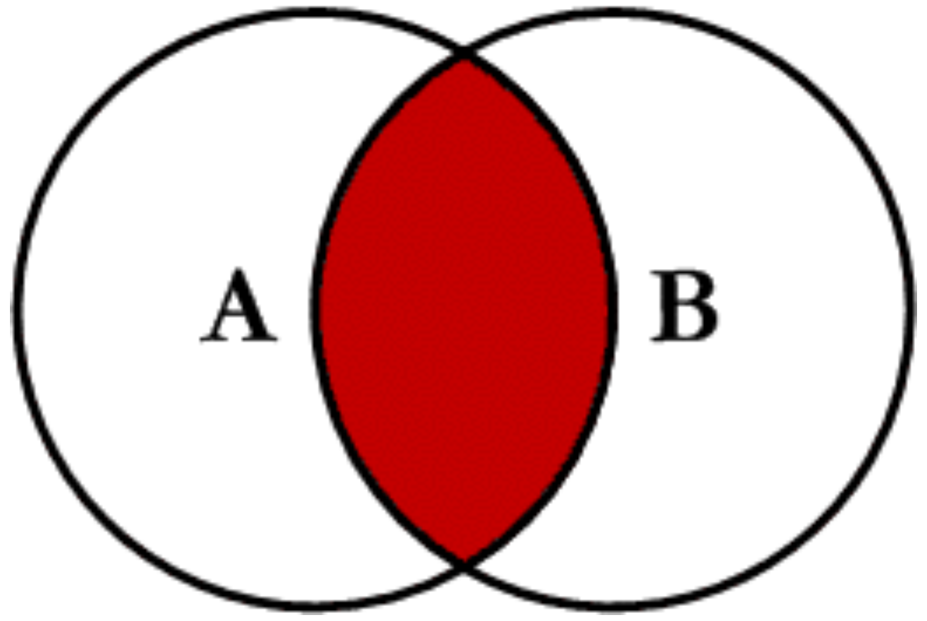
Objective: Show all students and their addresses that live in Chicago

```
SELECT students.id, name, street, zip, city
FROM students
JOIN addresses
  ON students.address = addresses.id
WHERE addresses.city = "Chicago”;
```

Students with Addresses
(results)

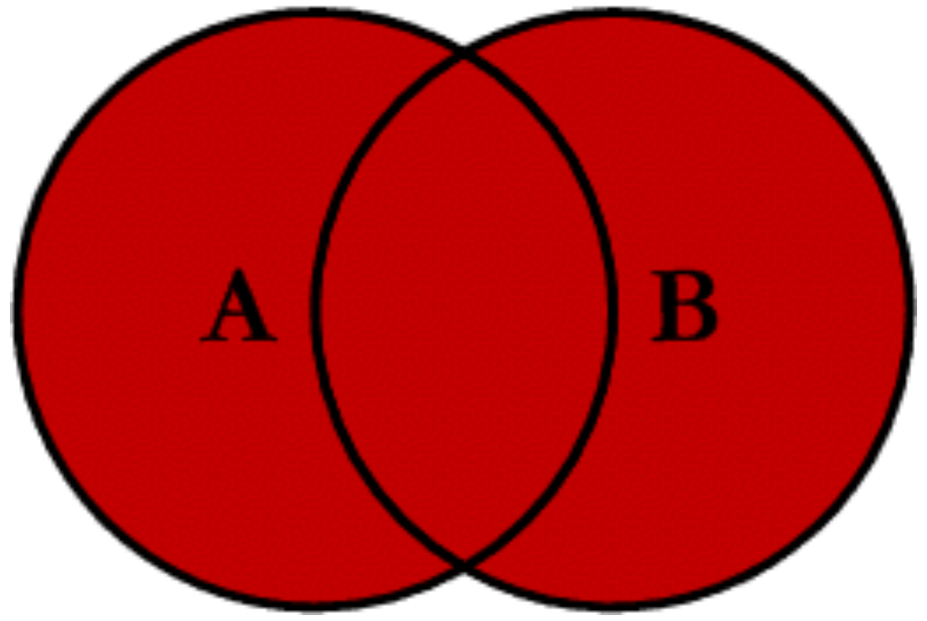
student.id	name	street	zip	city
3	Beth M.	423 Main St.	60647	Chicago
4	Lisa N.	14 Main St.	60650	Chicago

Inner Join



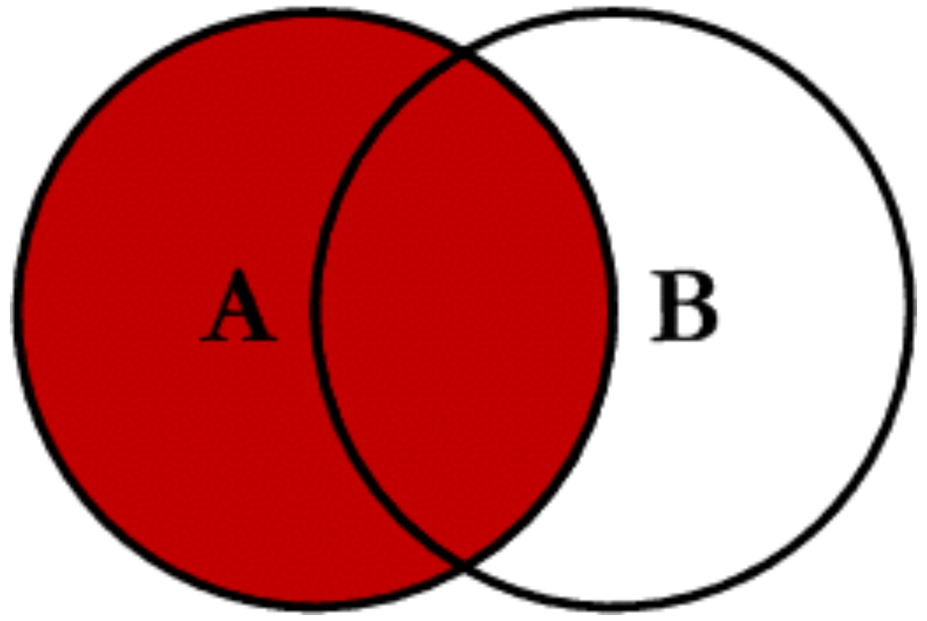
```
SELECT *
FROM A
INNER JOIN B
ON A.Key = B.Key;
```

Outer Join



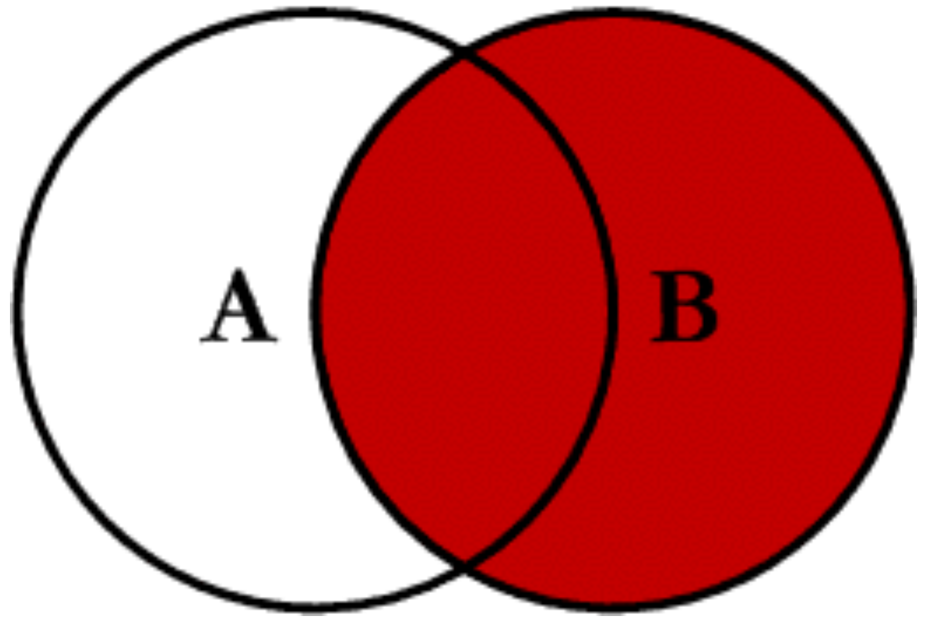
```
SELECT *
FROM A
FULL OUTER JOIN B
ON A.Key = B.Key;
```

Left Join



```
SELECT *
FROM A
LEFT JOIN B
ON A.Key = B.Key;
```

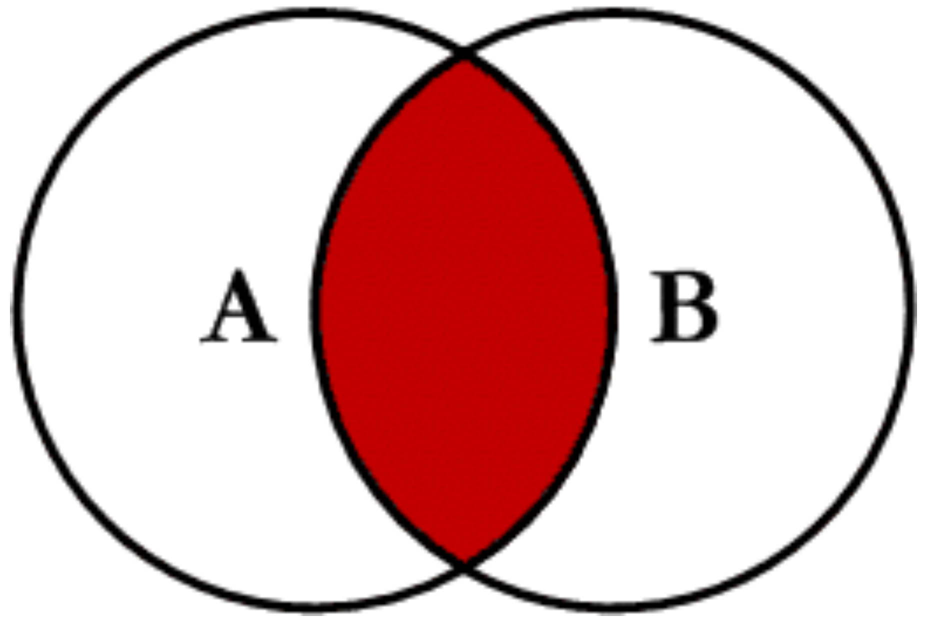
Right Join



```
SELECT *
FROM A
RIGHT JOIN B
ON A.Key = B.Key;
```

http://www.codeproject.com/Articles/33052/Visual-Representation-of-SQL-Joins

Inner Join



```
SELECT pets.name, owners.name
FROM owners
INNER JOIN pets
ON pets.owner_id = owners.id;
```

owners

id	name
1	Geordi
2	Janeway
3	Data
4	Spock

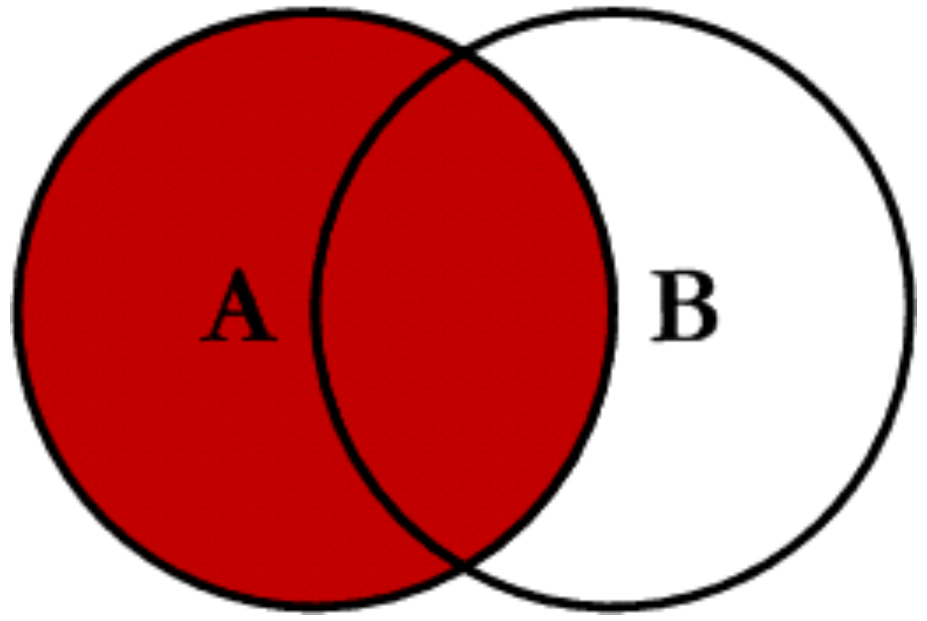
pets

id	owner_id	type	name
1	4	Monkey	Mittens
2	null	Lizard	Carol
3	1	Dog	Rufus
4	3	Cat	Spot

Query Results

pets.name	owners.name
Mittens	Spock
Rufus	Geordi
Spot	Data

Left Join



```
SELECT pets.name, owners.name
FROM owners
LEFT JOIN pets
ON pets.ownerID = owners.ID;
```

owners

id	name
1	Geordi
2	Janeway
3	Data
4	Spock

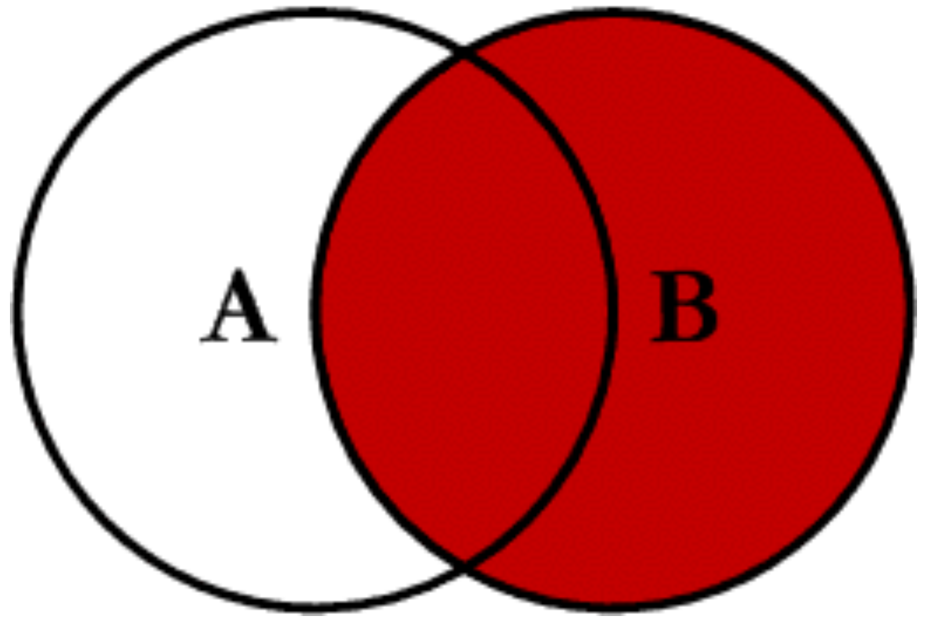
pets

id	owner_id	type	name
1	4	Monkey	Mittens
2	null	Lizard	Carol
3	1	Dog	Rufus
4	3	Cat	Spot

Query Results

pets.name	owners.name
Mittens	Spock
Rufus	Geordi
null	Janeway ←
Spot	Data

Right Join



```
SELECT pets.name, owners.name
FROM owners
RIGHT JOIN pets
ON pets.ownerID = owners.ID;
```

owners

id	name
1	Geordi
2	Janeway
3	Data
4	Spock

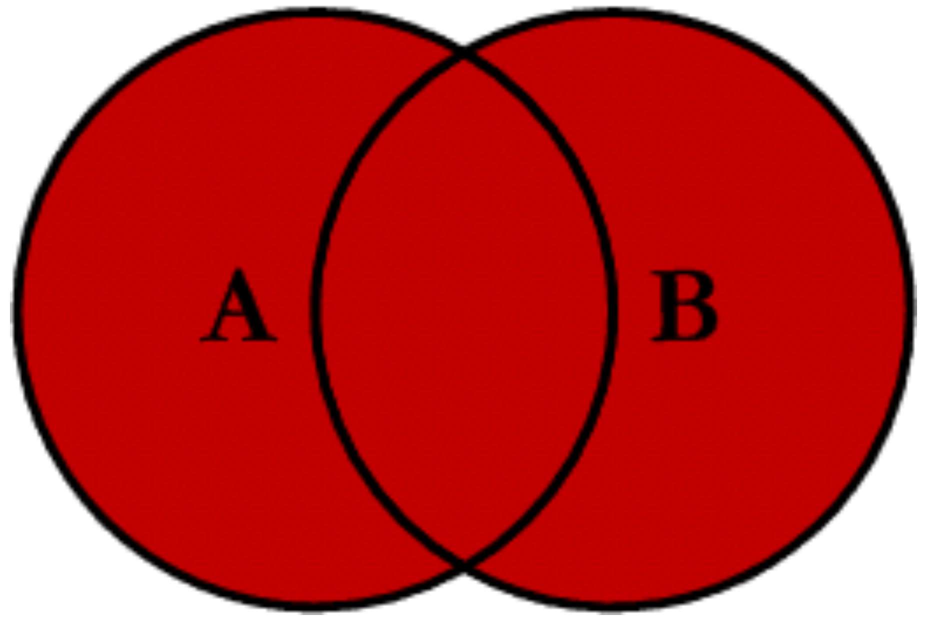
pets

id	owner_id	type	name
1	4	Monkey	Mittens
2	null	Lizard	Carol
3	1	Dog	Rufus
4	3	Cat	Spot

Query Results

pets.name	owners.name
Mittens	Spock
Carol	null
Rufus	Geordi
Spot	Data

Outer Join



```
SELECT pets.name, owners.name
FROM owners
FULL OUTER JOIN pets
ON pets.ownerID = owners.ID;
```

owners

id	name
1	Geordi
2	Janeway
3	Data
4	Spock

pets

id	owner_id	type	name
1	4	Monkey	Mittens
2	null	Lizard	Carol
3	1	Dog	Rufus
4	3	Cat	Spot

Query Results

pets.name	owners.name	
Mittens	Spock	
Carol	null	←
Rufus	Geordi	
null	Janeway	←
Spot	Data	

AS

students

id	name	age
1	Bart S.	10
2	Lisa S.	8
3	Jim F.	13
4	Joan B.	15

enrollments

student_id	school_id
1	1
2	1
3	2
4	3

schools

id	name	level
1	Springfield Elementary	E
2	Brook Middle	M
3	Springbrook High	H
4	Springfield University	U

```
SELECT *
FROM students AS st
INNER JOIN enrollment AS e
  ON st.id = e.student_id
INNER JOIN school AS sc
  ON e.school_id = sc.id;
```

Query Results

st.id	st.name	age	student_id	school_id	sc.id	sc.name	level
1	Bart S.	10	1	1	1	Springfield Elementary	E
2	Lisa S.	8	2	1	1	Springfield Elementary	E
3	Jim F.	13	3	2	2	Brook Middle	M
4	Joan B.	15	4	3	3	Springbrook High	H

GROUP BY
+
COUNT

students

id	name	age
1	Bart S.	10
2	Lisa S.	8
3	Jim F.	13
4	Joan B.	15

enrollments

student_id	school_id
1	1
2	1
3	2
4	3

schools

id	name	level
1	Springfield Elementary	E
2	Brook Middle	M
3	Springbrook High	H
4	Springfield University	U

```
SELECT name, COUNT(*)
FROM schools
INNER JOIN enrollments
  ON schools.id = enrollments.school_id
GROUP BY name;
```

Query Results

name	COUNT(*)
Springfield Elementary	2
Brook Middle	1
Springbrook High	1
Springfield University	0

ORDER BY

students

id	name	age
1	Bart S.	10
2	Lisa S.	8
3	Jim F.	13
4	Joan B.	15

enrollments

student_id	school_id
1	1
2	1
3	2
4	3

schools

id	name	level
1	Springfield Elementary	E
2	Brook Middle	M
3	Springbrook High	H
4	Springfield University	U

```sql
SELECT *
FROM students
ORDER BY age DESC;
```

Query Results

id	name	age
4	Joan B.	15
3	Jim F.	13
1	Bart S.	10
2	Lisa S.	8

SUB-QUERIES

students

id	name	age
1	Bart S.	10
2	Lisa S.	8
3	Jim F.	13
4	Joan B.	15

enrollments

student_id	school_id
1	1
2	1
3	2
4	3

schools

id	name	level
1	Springfield Elementary	E
2	Brook Middle	M
3	Springbrook High	H
4	Springfield University	U

```sql
SELECT id, name, age
FROM students
INNER JOIN enrollments
  ON students.id = enrollments.student_id
INNER JOIN (
  SELECT school_id
  FROM students
  WHERE students.name = 'Lisa S.'
  INNER JOIN enrollments
    ON students.id = enrollments.student_id
  ) AS Lisa_Schools
  ON Lisa_Schools.school_id = enrollments.school_id
WHERE name != 'Lisa S.';
```

Sub-Query Results

school_id
1

Query Results

id	name	age
1	Bart S.	10

CRUD Operations

SQL is used to create/read/update/delete (CRUD) data from a database

- `INSERT` – Insert (*create*) new rows into a table
- `SELECT` – Get (*read*) data from a database
- `UPDATE` – *Update* existing rows in a table
- `DELETE` – *Delete* rows from a table

- `CREATE` / `DROP` – Make / Delete new dbs/tables/views/indexes

Solo Exercise: SQL