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
- Use ORDER BY and COUNT

Example DB

students

id	name	age	gender	address_id
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

All 20 Year Old Students

students

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

20 Year Old Students

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

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

students

id	name	age	gender	address_id
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

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

Students with Addresses

students.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_id
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

```
SELECT students.id, name, street, zip, city
  FROM students
JOIN addresses
  ON students.address_id = addresses.id
WHERE addresses.city = 'Chicago';
```

Students with Addresses

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

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!)

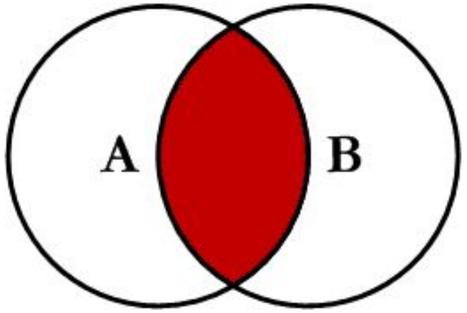
CRUD Operations

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

- INSERT: Insert new rows into a table
- SELECT: Get data from a database
- UPDATE: Update existing rows in a table
- DELETE: Delete rows from a table

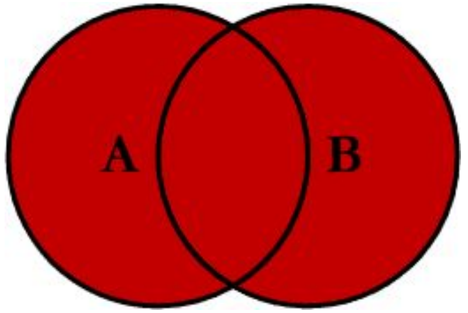
- CREATE / DROP: Make / delete new dbs/tables/views/indexes

Inner Join



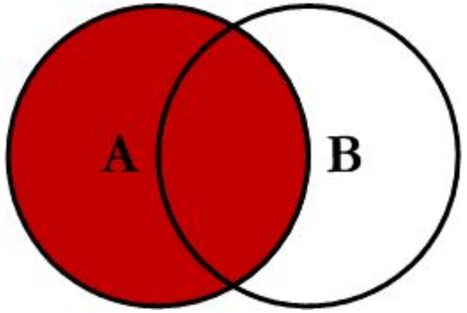

```
SELECT *
FROM A
INNER JOIN B
ON A.Key = B.Key
```

Outer Join



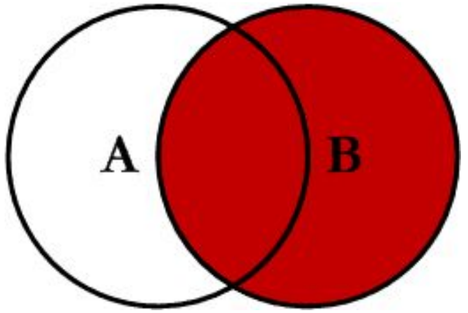

```
SELECT *
FROM A
FULL OUTER JOIN B
ON A.Key = B.Key
```

Left Join




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

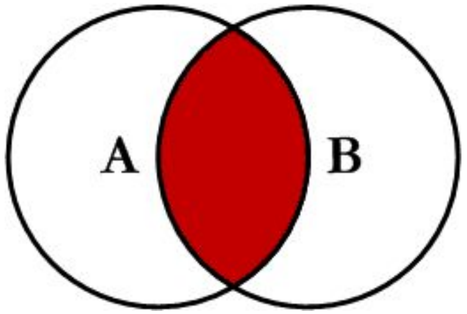
Right Join




```
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

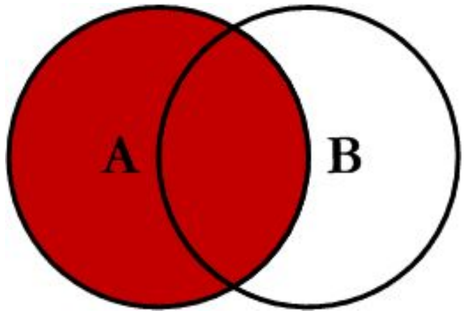
id	name
1	Geordi
2	Janeway
3	Data
4	Spock

PETS

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

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

Left Join




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

OWNERS

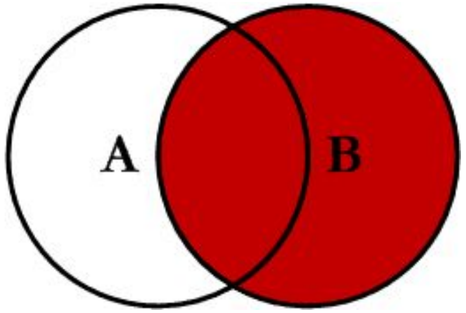
id	name
1	Geordi
2	Janeway
3	Data
4	Spock

PETS

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

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.owner_id = owners.id
```

OWNERS

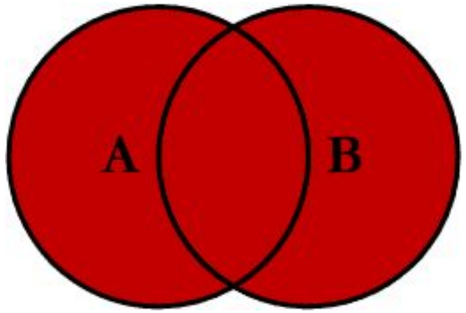
id	name
1	Geordi
2	Janeway
3	Data
4	Spock

PETS

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

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.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

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 enrollments AS e
    ON st.id = e.student_id
  INNER JOIN school AS sc
    ON e.school_id = sc.id;
```

Result

st.ID	st.Name	Age	StudentID	SchoolID	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;
```

Result

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

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

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

Result

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
      INNER JOIN enrollments
        ON students.id = enrollments.student_id
      WHERE students.name = 'Lisa S.'
  ) AS lisa_schools
    ON lisa_schools.school_id = enrollments.school_id
  WHERE Name != 'Lisa S.';
```

Result

id	name	age
1	Bart S.	10

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
- Use ORDER BY and COUNT

WORKSHOP