SQL / Active Record

Devpoint Labs - Dave Jungst / Jake Sorce

Active Record VS SQL

Agent.all	SELECT agents FROM agents
Agent.select(“*”)	SELECT * FROM agents
Agent.select(:first_name)	SELECT first_name FROM agents
Agent.find(1)	SELECT * FROM agents WHERE agents.id = 1 LIMIT 1
Agent.where(first_name: “Steve”)	SELECT * FROM agents WHERE first_name = ‘Steve’
Agent.order(:last_name)	SELECT * FROM agents ORDER BY last_name ASC
Agent.find_by(last_name: “Jones”)	SELECT * FROM agents WHERE last_name = ‘Jones’ LIMIT 1

Using ACTIVE RECORD

- Find all buyers
- Find a buyer with ID 123
- Find the first buyer with the first name “Dave”
- Find all buyers with the first name “Dave”
- Order buyers by last name
- Get first_name, last_name, & email from all buyers

Using SQL

- Find all buyers as YAML
- Retrieve all columns from all buyers
- Find a buyer with an ID of 2
- Find the first buyer with a first_name of “Jake”
- File all buyers with the first_name of “Jake”
- Order buyers by interest_level from most interested to least interested
- Get first_name, last_name, email & interest level from all buyers

BONUS:
Find the first 10 buyers first_name, last_name, email, interest_level
Where the interest level is greater than 5
Ordered by highest interest level to lowest interest level

SQL solution

```
    SELECT first_name, last_name, email, interest_level
      FROM buyers
    WHERE interest_level > 5
ORDER BY interest_level DESC
```

ActiveRecord solution

```
Buyer.select(:first_name, :last_name, :email, :interest_level)
        .where("interest_level > ?", 5)
         .order(interest_level: :desc)
```

Joins

Joins allow for more complex queries by bringing in relational queries

Let's say you wanted specific fields from the Agents table and specific fields from each agent's sellers

You could get each Agent 1 at a time and then loop through that agents sellers but now WAY too many SQL queries are being made. A join will allow you to query multiple tables at once.

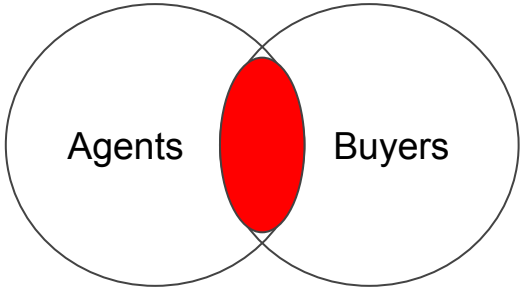
Agent.joins(:sellers) will return each agent and each agent's sellers

INNER JOIN

Inner Join

SELECT agents.last_name, b.last_name FROM agents

INNER JOIN buyers b ON b.agent_id = agents.id



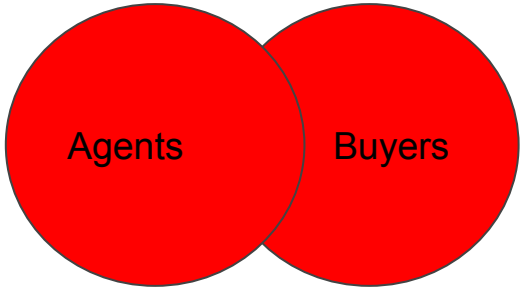
last_name character varying	last_name character varying
Lakin	Stehr
Lakin	Hirthe
Lakin	Haag
Lakin	Lehner
Deckow	O'Kon
Deckow	Bartell

FULL OUTER JOIN

FULL OUTER Join

SELECT agents.last_name, b.last_name FROM agents

FULL OUTER JOIN buyers b ON b.agent_id = agents.id



Data Output | Explain | Messages

	last_name character varying	last_name character varying
1092	Swift	Jones
1093	Crooks	Jones
1094	Dickinson	Jones
1095		Jungst
1096	Spinka	Kassulke
1097	Murray	Kassulke

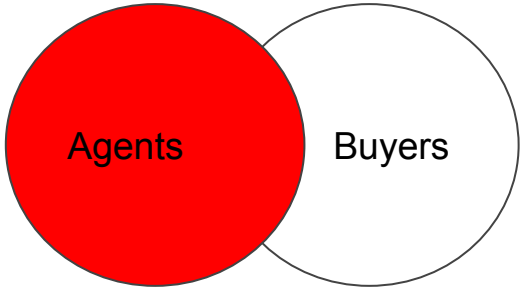
SQL Editor | Graphical Query Builder

LEFT OUTER JOIN

Left Outer Join

SELECT agents.last_name, b.last_name FROM agents

LEFT OUTER JOIN buyers b ON b.agent_id = agents.id



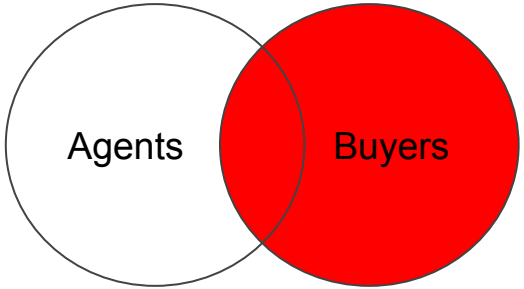
	last_name character varying	last_name character varying
1090	Prunnerstitt	Jones
1091	Luettgen	Jones
1092	Crooks	Jones
1093	Swift	Jones
1094	Dickinson	Jones
1095	Lynch	Kassulke
1096	Jerde	Kassulke

RIGHT OUTER JOIN

Right Outer Join

SELECT agents.last_name, b.last_name FROM agents

RIGHT OUTER JOIN buyers b ON b.agent_id = agents.id



Using Joins

```
SELECT agents.email, s.email, a.street, a.city, a.state, a.zip

FROM agents

INNER JOIN sellers s ON s.agent_id = agents.id

INNER JOIN properties p ON p.seller_id = s.id

INNER JOIN addresses a ON a.property_id = p.id

WHERE p.status = ‘Sold’
```

Same Query in Rails

```
1  class Agent < ActiveRecord::Base
2    has_many :buyers
3    has_many :sellers
4    has_many :properties
5
6    def self.sold_properties
7      select('agents.email, s.email, a.street, a.city, a.state, a.zip')
8      .joins('INNER JOIN sellers s ON s.agent_id = agents.id
9              INNER JOIN properties p ON p.seller_id = s.id
10             INNER JOIN addresses a ON a.property_id = p.id')
11     .where('p.status = ?', 'Sold')
12   end
13 end
```

Agent.sold_properties

Alias

```
1  class Agent < ActiveRecord::Base
2    has_many :buyers
3    has_many :sellers
4    has_many :properties
5
6    def self.sold_properties
7      select('agents.email AS agent_email, s.email AS buyer_email, a.street, a.city, a.state, a.zip')
8      .joins('INNER JOIN sellers s ON s.agent_id = agents.id
9              INNER JOIN properties p ON p.seller_id = s.id
10             INNER JOIN addresses a ON a.property_id = p.id')
11     .where('p.status = ?', 'Sold')
12   end
13
14 end
```

Agent.sold_properties.first.buyer_email

Project

1. clone https://github.com/wdjungst/first_realty
2. bundle exec rake db:create db:migrate populate:db
3. Use pgadmin to solve the queries on next page

Bonus:

1. Create model methods for the queries on the next page
2. Add buttons on the app to fire the query
3. Display results in tables

Queries

1. Find all properties that are not sold
2. Find the address of all properties that are sold
3. Find the agent first_name, last_name, email of the agents with un-intersted buyers (< 5)
4. Find the last property sold
5. Find all buyers first_name, last_name & phone_number for buyers who do not have agents
6. Find the buyers email, agent's email, and the property address for any properties with sales pending
7. Find buyers who were last contacted more than 1 month ago
8. Find homes for sale between 100,000 and 200,000 and get the sellers email and agents info
9. Get all of the agents who have buyers with interest level above 5 ordered by agents last_name and buyers email