



10: Entity Command





Automatic Merge Fields

From a Block or Rock Itself

```
1 {{ 'Lava' | Debug }}
```

Lava Debug Info

Below is a listing of available merge fields for this block. Find out more on Lava at rockrms.com/lava.

Page Parameter

Current Person

Current Visitor

Campuses

Pagination

Linked Pages

Items

Item Tag List

Archive Summary

- RockVersion - 1.16.6.6
- CurrentPageUrl - /blog?Tag=TagTemplate
- ArchiveSummaryPageUrl - /blog?Year=YearTemplate&Month=MonthTemplate

Person

Global Attribute



What if...

the record I need isn't in the list?



Lava Debug Info

Below is a listing of available merge fields for this block. Find out more on Lava at rockrms.com/lava.

Page Parameter

Current Person

Current Visitor

Campuses

Pagination

Linked Pages

Items

Item Tag List

Archive Summary

- RockVersion - 1.16.6.6
- CurrentPageUrl - /blog?Tag=TagTemplate
- ArchiveSummaryPageUrl - /blog?Year=YearTemplate&Month=MonthTemplate

Person

Global Attribute



What if...

the record I need isn't already provided?

If you need:

- A person:

```
1  {% assign person = 4 | PersonById %}
```

```
1  {% assign personAliasGuid = Item | Attribute:'Author','RawValue' %}  
2  {% assign person = personAliasGuid | PersonById %}
```





What if...

the record I need isn't already provided?

If you need:

- A group:

```
1  {% assign group = 115 | GroupById %}
```

```
1  {% assign groupGuid = Item | Attribute:'Group','RawValue' %}
2  {% assign group = groupGuid | GroupById %}
```





What if...

the record I need isn't already provided?

If you need:

- Another Entity Type:

There are no other “__Byld” filters





Commands

```
{% person where:'LastName == "Decker"' %}
```

- Familiar syntax
- Specific to Rock
 - (not available in other adaptions of Liquid)
- Custom filtering





Entity Command

```
{% person where:'LastName == "Decker"' %}
```



Entity Command

```
{% person where:'LastName == "Decker"' %}
```



Entity Command

```
{% person where:'LastName == "Decker"' %}
```

```
{{ personItems }}
```



Entity Command

```
{% person where:'LastName == "Decker"' %}
```

```
{{ personItems }}
```



Entity Command

```
{% person where:'LastName == "Decker"' %}
```

```
{{ personItems }}
```

```
{% endperson %}
```



Entity Command

```
1  {% person where:'LastName == "Decker"' %}  
2  
3  {% endperson %}
```

Works for all entity types

By default, this provides a variable named [entity_type]Items, such as personItems.



Entity Command

```
1  {% person where:'LastName == "Decker"' %}  
2    <pre>{{ personItems | ToJSON }}</pre>  
3  {% endperson %}
```



Important to Know:

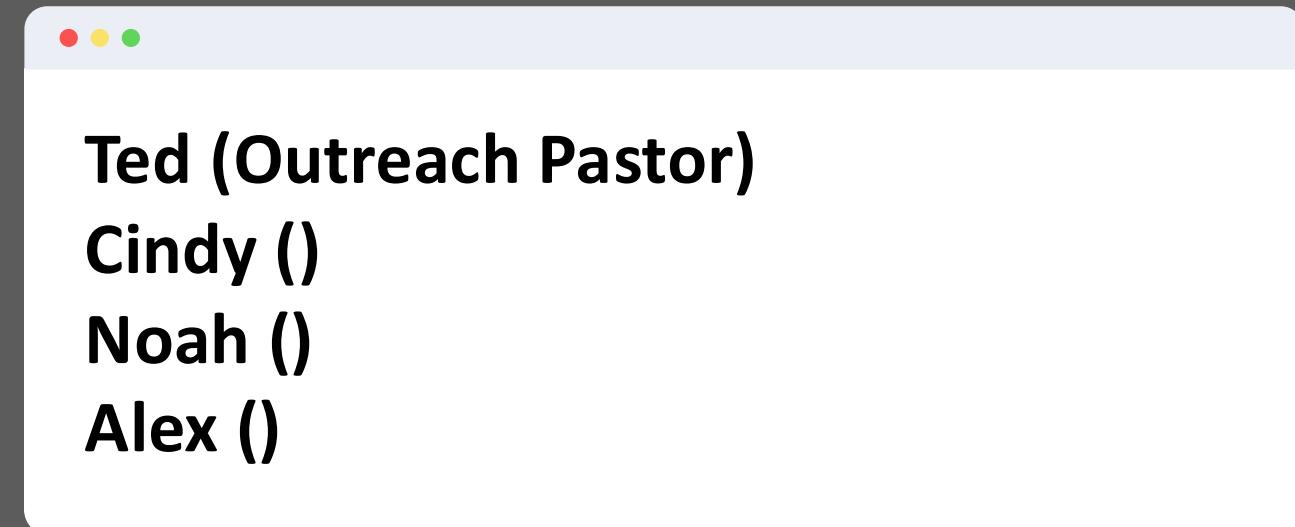
The elements in the array are already full entity records.

```
1  [  
2  {  
3    "Id": 4,  
4    "NickName": "Ted",  
5    "FirstName": "Theodore",  
6    "LastName": "Decker",  
7    "ConnectionStatusValueId": 65,  
8    "PhoneNumbers": [ ... ]  
9  },  
10 {  
11   "Id": 5,  
12   "NickName": "Cindy",  
13   "FirstName": "Cynthia",  
14   "LastName": "Decker",  
15   "ConnectionStatusValueId": 65,  
16   "PhoneNumbers": [ ... ]  
17 },  
18 {  
19   "Id": 6,  
20   "NickName": "Noah",  
21   "FirstName": "Noah",  
22   "LastName": "Decker",  
23   "ConnectionStatusValueId": 146,  
24   "PhoneNumbers": [ ... ]  
25 },  
26 {  
27   "Id": 7,  
28   "NickName": "Alex",  
29   "FirstName": "Alexis",  
30   "LastName": "Decker",  
31   "ConnectionStatusValueId": 146,  
32   "PhoneNumbers": [ ... ]  
33 }  
34 ]
```



Entity Command

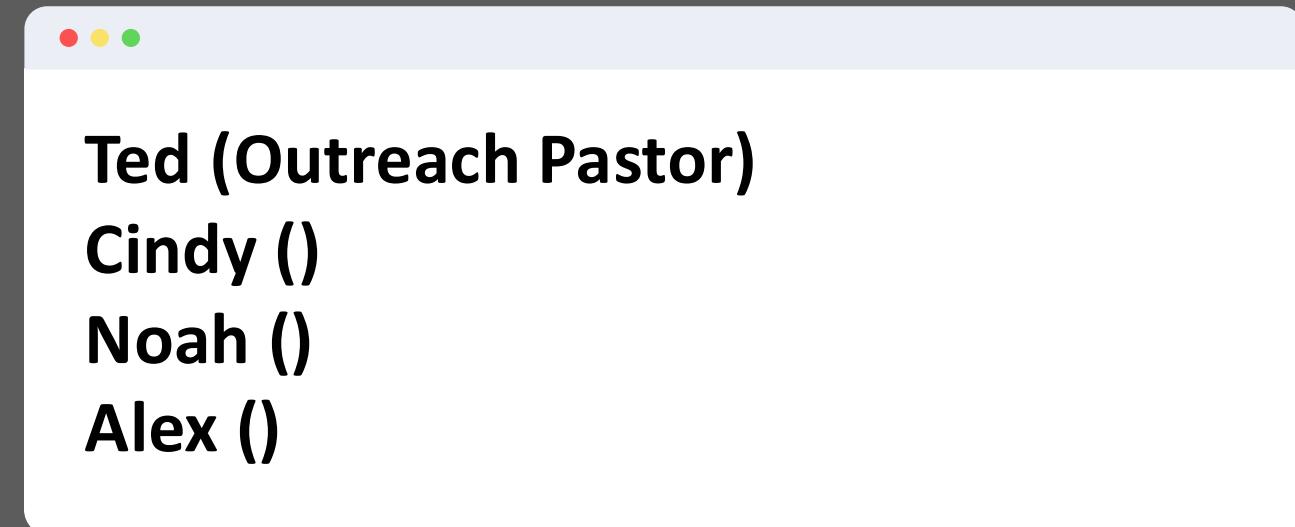
```
1  {% person where:'LastName == "Decker"' %}  
2    {% for individual in personItems %}  
3      <p>{{ individual.NickName }} ({{ individual | Attribute:'Position' }})</p>  
4    {% endfor %}  
5  {% endperson %}
```





Entity Command

```
1  {% person where:'LastName == "Decker"' %}  
2    {% for person in personItems %}  
3      <p>{{ person.NickName }} ({{ person | Attribute:'Position' }})</p>  
4    {% endfor %}  
5  {% endperson %}
```





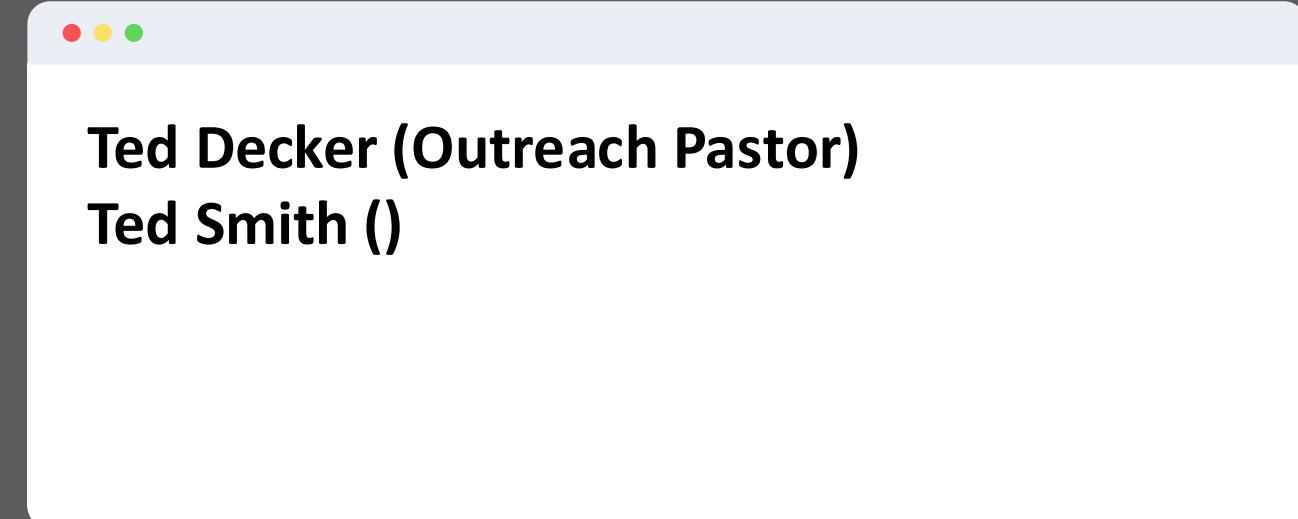
Entity Command

```
1  {% person where:'NickName == "Ted"' %}  
2    {% for person in personItems %}  
3      <p>{{ person.FullName }} ({{ person | Attribute:'Position' }})</p>  
4    {% endfor %}  
5  {% endperson %}
```



Remember:

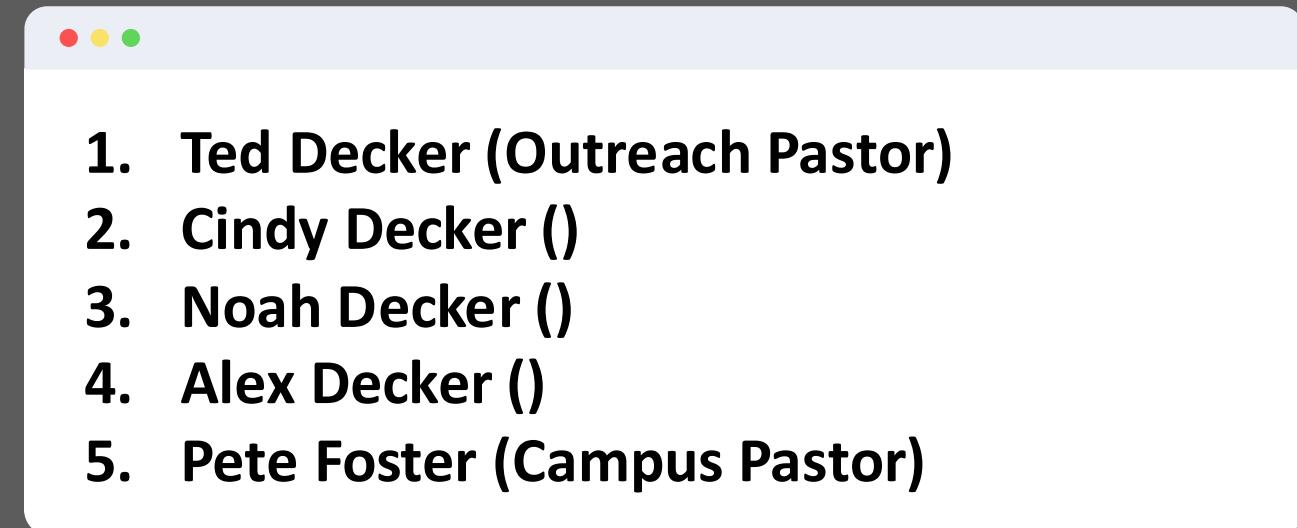
Properties available to Lava can
be found on the Model Map





Entity Command

```
1  {% person where:'LastName == "Decker" || Position *= "Pastor"' %}  
2    <ol>  
3      {% for person in personItems %}  
4          <li>{{ person.FullName }} ({{ person | Attribute:'Position' }})</li>  
5      {% endfor %}  
6    </ol>  
7  {% endperson %}
```





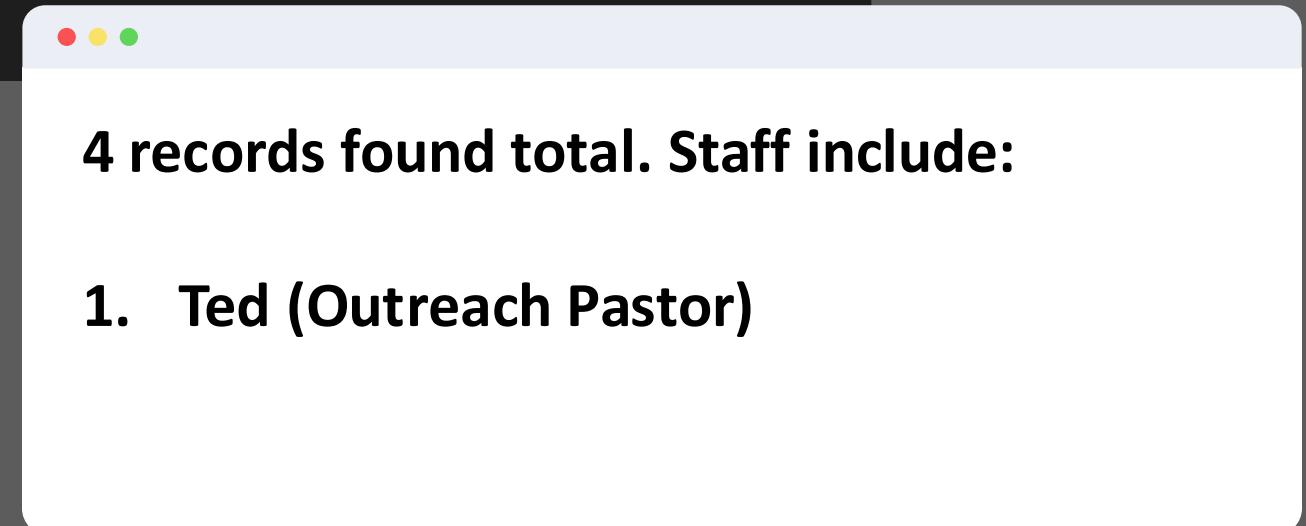
Entity Command Operators

<code>==</code>	Is Equal To	<code>></code>	Is Greater Than
<code>!=</code>	Is Not Equal To	<code>>=</code>	Is Greater Than or Equal To
<code>^=</code>	Starts With	<code><</code>	Is Less Than
<code>*=</code>	Contains	<code><=</code>	Is Less Than or Equal To
<code>*!</code>	Does Not Contain	<code>_=</code>	Is Blank
<code>\$=</code>	Ends With	<code>_!</code>	Is Not Blank
<code>&&</code>	Logical AND operator		
<code> </code>	Logical OR operator		



Entity Command

```
1  {% person where:'LastName == "Decker"' %}  
2      <p>{{ personItems | Size }} records found total. Staff include:</p>  
3      <ol>  
4          {% for person in personItems %}  
5              {% assign position = Person | Attribute:'Position' %}  
6              {% assign positionLength = position | Size %}  
7              {% if positionLength > 0 %}  
8                  <li>{{ person.FullName }} ({{ position }})</li>  
9              {% endif %}  
10         {% endfor %}  
11     </ol>  
12  {% endperson %}
```



The screenshot shows a window with a dark header bar featuring the red, yellow, and green Mac OS X window control buttons. The main content area is white and displays the following text:

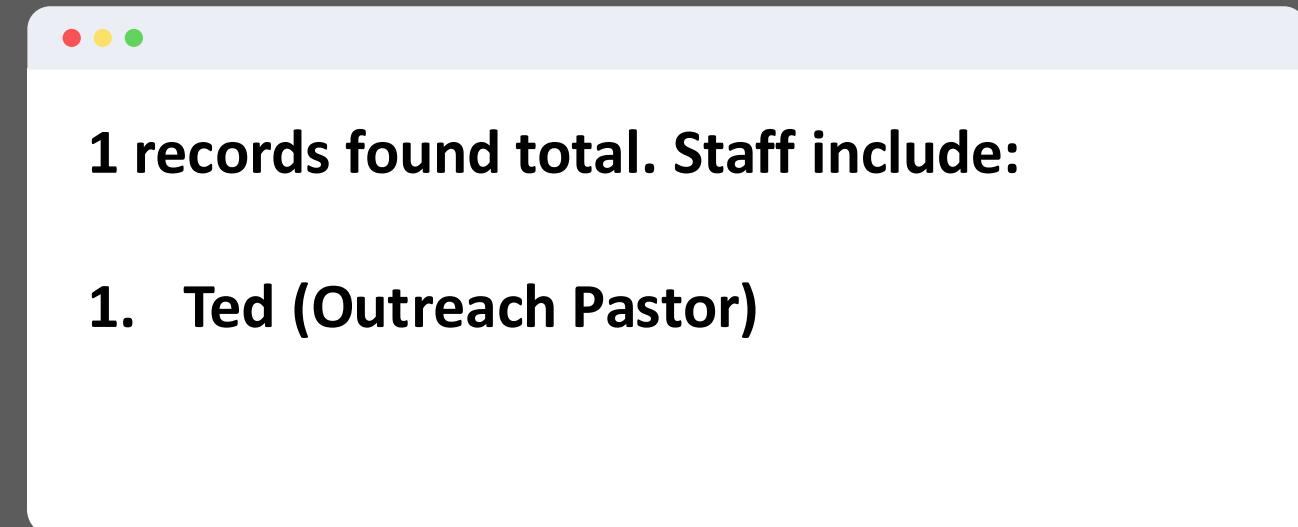
4 records found total. Staff include:

1. Ted (Outreach Pastor)



Entity Command

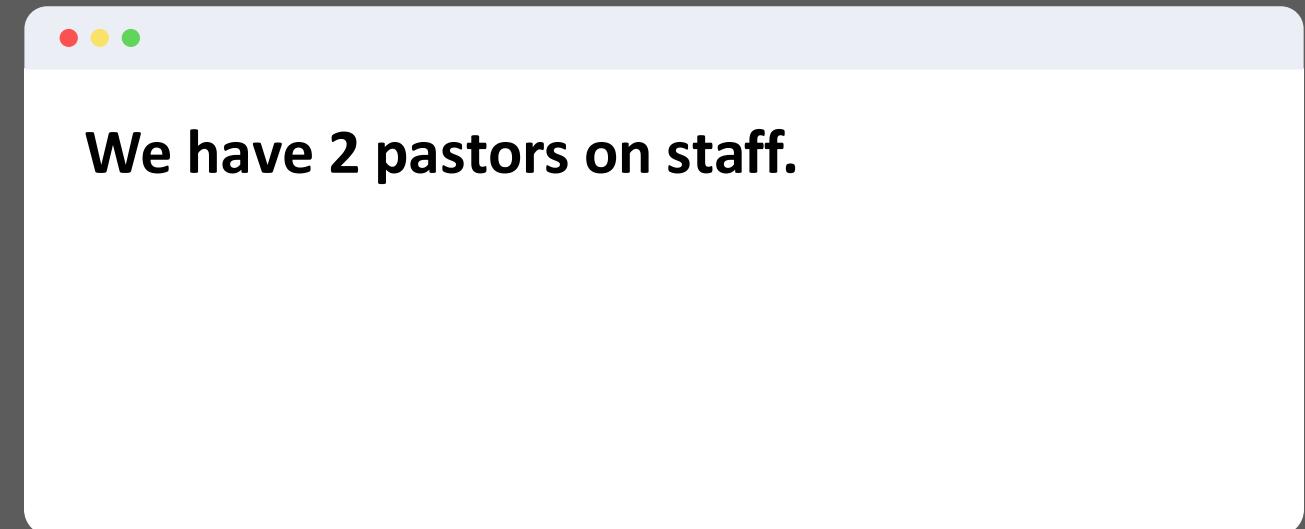
Property	Attribute
1 {% person where:'LastName == "Decker" && Position != "1"' %}	
2 <p>{{ personItems Size }} records found total. Staff Include:</p>	
3 	
4 {% for person in personItems %}	
5 {{ person.FullName }} ({{ person Attribute:'Position' }})	
6 {% endfor %}	
7 	
8 {% endperson %}	





Count

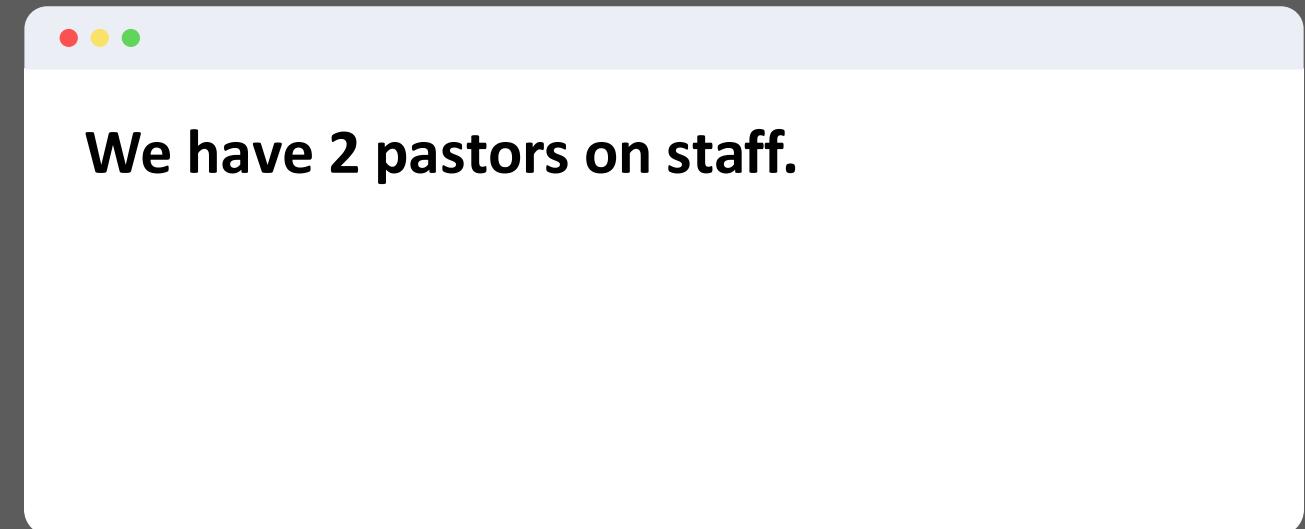
```
1  {% person where:'Position _! "1"' count:'true' %}  
2  We have {{ count }} pastors on staff.  
3  {% endperson %}
```





Count

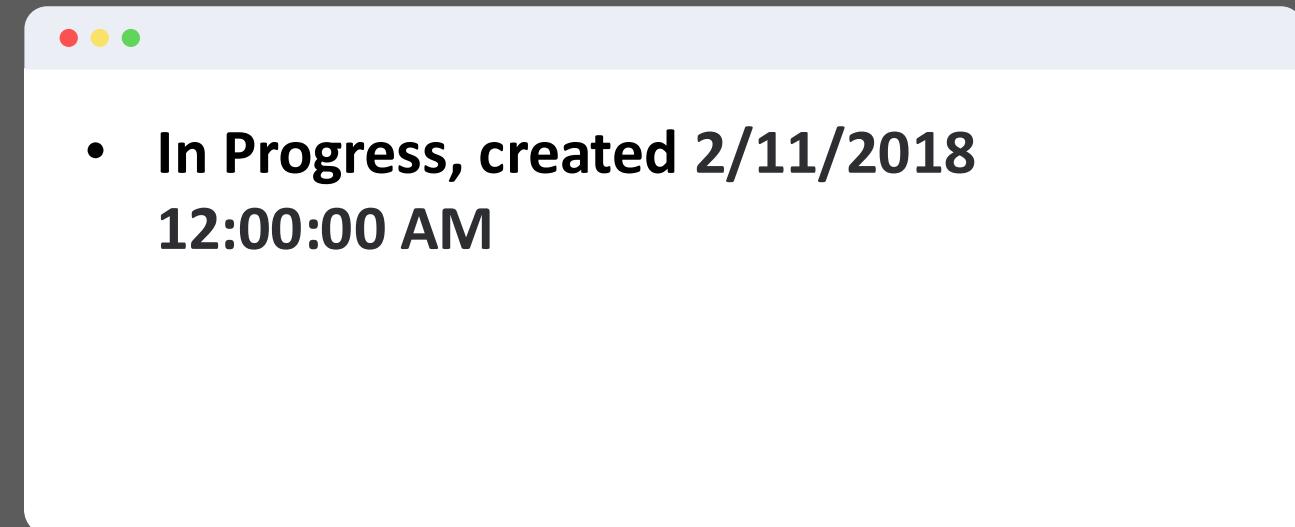
```
1  {% person where:'Position _! "1"' count:'true' %}  
2  We have {{ 'pastor' | ToQuantity:count }} on staff.  
3  {% endperson %}
```





Entities

```
1  {% connectionrequest where:'ConnectionOpportunityId == "3" && ConnectionStateId == "0"' %}  
2    <ul>  
3      {% for request in connectionrequestItems %}  
4        <li>{{ request.ConnectionStatus.Name }}, created {{ request.CreatedDateTime }}</li>  
5      {% endfor %}  
6    </ul>  
7  {% endconnectionrequest %}
```





Model Map

Assemble your Lava templates

The screenshot shows the Lava Model Map interface. On the left is a vertical navigation bar with icons for Home, Power Tools, and Model Map. The main area is titled "Model Map" and shows a "Home > Power Tools > Model Map". Below this is a "Model Map" title and a 3x5 grid of categories:

Category	Icon	Description
Check -in	Check-in icon	Check -in
Engagement	Gears icon	Engagement
Meta	Grid icon	Meta
CMS	Code icon	CMS
Event	Phone icon	Event
Prayer	Cloud icon	Prayer
Communication	Speech bubble icon	Communication
Finance	Dollar sign icon	Finance
Reporting	Document icon	Reporting
Core	Gear icon	Core
Group	People icon	Group
Web Farm	Server icon	Web Farm
LMS	User icon	LMS
Workflow	Gears icon	Workflow

A red arrow points to the "Engagement" category in the second row.



Model Map

Assemble your Lava templates

The screenshot shows the 'Model Map' interface. On the left is a vertical sidebar with icons for Home, Power Tools, Model Map, and Profile. The main area is titled 'Model Map' and shows a grid of 15 icons representing different models: Check-in, CMS, Communication, Core, CRM, Engagement, Event, Finance, Group, LMS, Meta, Prayer, Reporting, Web Farm, and Workflow. A red arrow points from the bottom of the sidebar towards the right-hand details panel.

The screenshot shows the 'AchievementAttempt' model details page. At the top is an orange header with a logo, search bar, and user profile. Below is a navigation bar with icons for Meta, Prayer, Reporting, Web Farm, and Workflow. The main content is divided into two sections: 'Engagement Models' (list of related models) and 'Model Details' (description and properties).
Engagement Models:
Achievement Attempt, Achievement Type, Achievement Type Prerequisite, Connection Activity Type, Connection Opportunity, Connection Opportunity Campus, Connection Opportunity Connector, Group, Connection Opportunity Group, Connection Opportunity Group Config, Connection Request, Connection Request Activity, Connection Request Workflow, Connection Status, Connection Status Automation, Connection Type, Connection Workflow, Step, Step Program, Step Program Completion.
Model Details:
Filter Options, Show: Methods
AchievementAttempt
Represents an Achievement Attempts in Rock.
Properties:
AchievementAttemptEndDateTime Gets or sets the achievement attempt end date time.
AchievementAttemptStartTime* Gets or sets the achievement attempt start date time.
AchievementType* Gets or sets the AchievementType of this attempt.
AchievementTypeId* Gets or sets the Id of the AchievementType to which this attempt belongs. This property is required.
AchieverEntityId* Gets or sets the achiever entity identifier. The



Model Map

Assemble your Lava templates

{{ item

The screenshot shows the 'Engagement Models' section of the Model Map interface. On the left, there's a sidebar with icons for Meta, Prayer, Reporting, Web Farm, and Workflow. The main area has tabs for 'Engagement Models', 'Model Details', and 'Filter Options'. Under 'Engagement Models', a list includes Achievement Attempt, Achievement Type, Achievement Type Prerequisite, Connection Activity Type, Connection Opportunity, Connection Opportunity Campus, Connection Opportunity Connector Group, Connection Opportunity Group, Connection Opportunity Group Config, Connection Request, Connection Request Activity, Connection Request Workflow, Connection Status, Connection Status Automation, Connection Type, Connection Workflow, Step, Step Program, Step Program Completion, and Step Status. The 'Model Details' tab shows 'AchievementAttempt' with properties like 'AchievementAttemptEndDateTime', 'AchievementAttemptStartTime', 'AchievementType', 'AchievementTypeId', and 'AchieverEntityId'.

The screenshot shows two side-by-side views of the Model Map interface. The left view shows the 'Engagement Models' section with the same list of entities as the first screenshot. The right view shows the 'Model Details' section for 'ConnectionRequest'. It includes a 'Properties' table with columns for name, description, and type. Properties listed include 'AdditionalLavaFields', 'AssignedGroup', 'AssignedGroupId', 'AssignedGroupMemberAttributeValues', 'AssignedGroupMemberRoleId', 'AssignedGroupMemberStatus', and 'Attributes'.



Model Map

Assemble your Lava templates

`{{ item.ConnectionStatus}}`

The screenshot shows the Lava Model Map interface. On the left is a navigation sidebar with icons for Home, Meta, Prayer, Reporting, Web Farm, and Workflow. The main area displays the 'Engagement Models' section under 'Achievement Attempt'. It includes a 'Model Details' panel with a 'Filter Options' dropdown and a 'Show Methods' checkbox. Below this is a detailed description of the 'AchievementAttempt' class, which represents an Achievement Attempts in Rock. It lists properties: 'AchievementAttemptEndDateTime' (Gets or sets the achievement attempt end date time), 'AchievementAttemptStartTime' (Gets or sets the achievement attempt start date time), 'AchievementType' (Gets or sets the AchievementType of this attempt), 'AchievementTypeId' (Gets or sets the Id of the AchievementType to which this attempt belongs. This property is required), and 'AchieverEntityId' (Gets or sets the achiever entity identifier. The).

The screenshot shows the Lava Model Map interface. On the right is a list of properties for the 'ConnectionState' class. A red arrow points from the 'AchievementAttempt' model details in the previous screenshot to the 'ConnectionStatus' property. Another red arrow points from the 'AchievementAttempt' model details to the 'CreatedDateTime' property.

<code> ConnectionState *</code>	Gets or sets the state of the connection. This is a hard coded list of values defined in the code as an enumeration.
<code> ConnectionStatus</code>	Gets or sets the <code>ConnectionStatus</code> .
<code> ConnectionStatusId *</code>	Gets or sets the <code>ConnectionStatus</code> identifier.
<code> ConnectionTypeId *</code>	Gets or sets the <code>ConnectionType</code> identifier.
<code> ConnectorPersonAlias</code>	Gets or sets the connector <code>PersonAlias</code> .
<code> ConnectorPersonAliasId</code>	Gets or sets the connector <code>PersonAlias</code> identifier.
<code> ContextKey</code>	
<code> CreatedByPersonAlias</code>	
<code> CreatedByPersonAliasId</code>	
<code> CreatedByPersonId</code>	
<code> CreatedByPersonName</code>	
<code> CreatedDateKey</code>	Gets the created date key.
<code> CreatedDateTime</code>	



Model Map

Assemble your Lava templates

The screenshot shows the Model Map application interface. On the left, there's a sidebar with various icons and a main content area divided into two sections: 'Engagement Models' and 'Model Details'.

Engagement Models:

- Achievement Attempt
- Achievement Type
- Achievement Type Prerequisite
- Connection Activity Type
- Connection Opportunity
- Connection Opportunity Campus
- Connection Opportunity Connector
- Group
- Connection Opportunity Group
- Connection Opportunity Group Config
- Connection Request
- Connection Request Activity
- Connection Request Workflow
- Connection Status
- Connection Status Automation
- Connection Type
- Connection Workflow
- Step
- Step Program
- Step Program Completion

Model Details:

AchievementAttempt

Represents an Achievement Attempts in Rock.

Properties

- AchievementAttemptEndDateTime** Gets or sets the achievement attempt end date time.
- AchievementAttemptStartTime** Gets or sets the achievement attempt start date time.
- AchievementType** Gets or sets the [AchievementType](#) of this attempt.
- AchievementTypeId** Gets or sets the Id of the [AchievementType](#) to which this attempt belongs. This property is required.
- AchieverEntityId** Gets or sets the achiever entity identifier. The

A red arrow points from the bottom of the sidebar towards the 'Engagement Models' list.



Model Map

Assemble your Lava templates

`{{ item.ConnectionStatus }}`

The screenshot shows the 'Engagement Models' section of the application. On the left is a sidebar with icons for Meta, Prayer, Reporting, Web Farm, and Workflow. The main area displays the 'Model Details' for 'AchievementAttempt'. It includes a 'Properties' table with columns for name, description, and type. Properties listed include 'AchievementAttemptEndDateTime', 'AchievementAttemptStartTime', 'AchievementType', 'AchievementTypeId', and 'AchieverEntityId'.

The screenshot shows the 'Model Details' section for 'ConnectionStatus'. It includes a 'Properties' table with columns for name, description, and type. Properties listed include 'AdditionalLavaFields', 'Attributes', 'AttributeValueDefaults', 'AttributeValues', 'AutoInactivateState', 'AvailableKeys', and 'ConnectionStatusAutomations'.



Model Map

Assemble your Lava templates

`{{ item.ConnectionStatus.Name }}`

The screenshot shows the Engagement Models section of the application. On the left is a sidebar with icons for Meta, Prayer, Reporting, Web Farm, and Workflow. The main area displays the 'Engagement Models' list, which includes: Achievement Attempt, Achievement Type, Achievement Type Prerequisite, Connection Activity Type, Connection Opportunity, Connection Opportunity Campus, Connection Opportunity Connector Group, Connection Opportunity Group, Connection Opportunity Group Config, Connection Request, Connection Request Activity, Connection Request Workflow, Connection Status, Connection Status Automation, Connection Type, Connection Workflow, Step, Step Program, and Step Program Completion. On the right, the 'Model Details' panel is open for 'AchievementAttempt'. It shows the class name 'AchievementAttempt' and its description: 'Represents an Achievement Attempts in Rock.' Below this are sections for 'Properties' and 'Methods'. The 'Properties' section lists: 'AchievementAttemptEndDateTime' (Gets or sets the achievement attempt end date time), 'AchievementAttemptStartTime' (Gets or sets the achievement attempt start date time), 'AchievementType' (Gets or sets the AchievementType of this attempt), 'AchievementTypeId' (Gets or sets the Id of the AchievementType to which this attempt belongs. This property is required), and 'AchieverEntityId' (Gets or sets the achiever entity identifier. The). The 'Methods' section is currently collapsed.

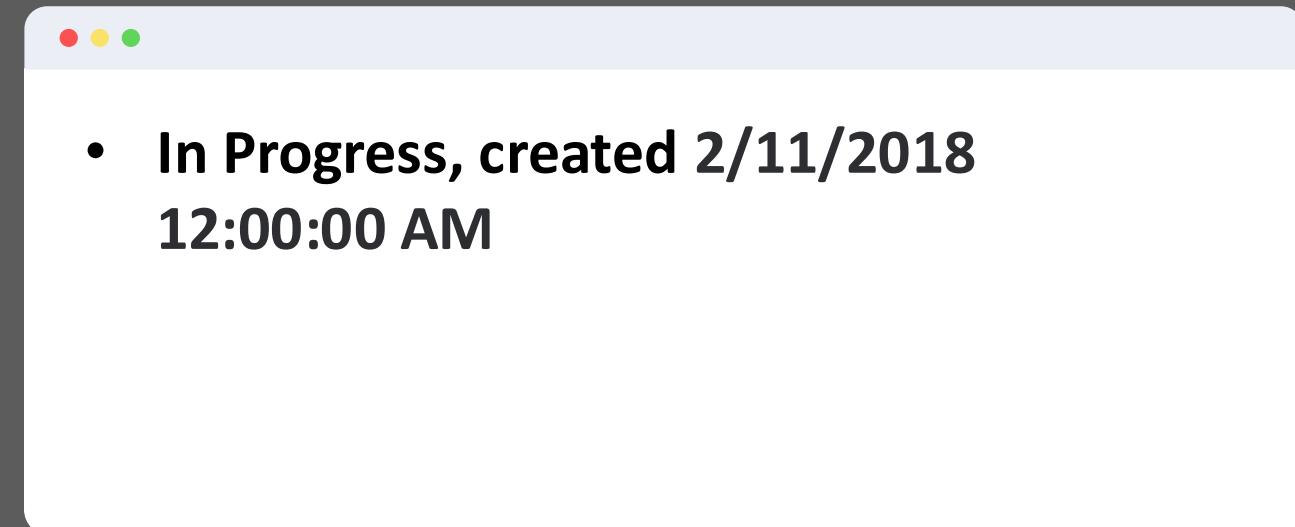
The screenshot shows the Model Map interface. On the left is a large black vertical bar. To its right is a list of properties for the 'AchievementAttempt' class. A red arrow points from the 'Name' property in the list to the 'Name' property in the 'Properties' section of the Engagement Models screenshot. The properties listed are:

- IsValid
- Item
- ModifiedAuditValuesAlreadyUpdated
- ModifiedByPersonAlias
- ModifiedByPersonAliasId
- ModifiedByPersonId
- ModifiedByPersonName
- ModifiedDate
- Name * (highlighted by a red arrow)
- Order
- ParentAuthority
- ParentAuthorityPre
- SupportedActions
- TypeId
- TypeName



Entities

```
1  {% connectionrequest where:'ConnectionOpportunityId == "3" && ConnectionStateId == "0"' %}  
2    <ul>  
3      {% for request in connectionrequestItems %}  
4        <li>{{ request.ConnectionStatus.Name }}, created {{ request.CreatedDateTime }}</li>  
5      {% endfor %}  
6    </ul>  
7  {% endconnectionrequest %}
```





Entities

```
1  {% group where:'GroupTypeId == "10"' %}  
2    <ul>  
3      {% for g in groupItems %}  
4          <li>{{ g.Name }}</li>  
5      {% endfor %}  
6    </ul>  
7  {% endgroup %}
```

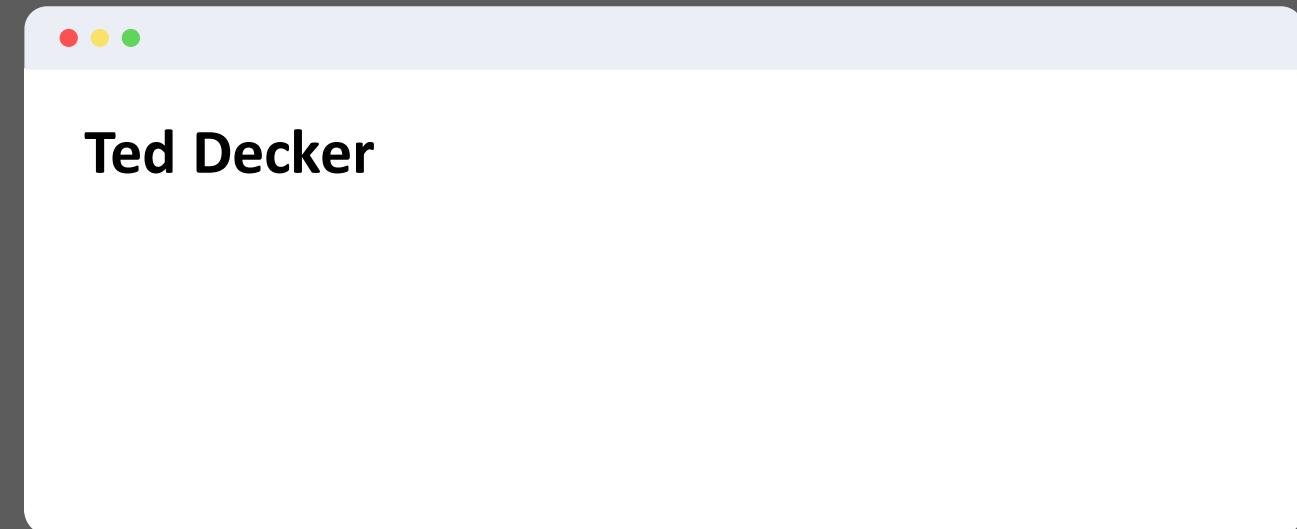
The screenshot shows a Mac OS X style application window with a title bar containing three colored dots (red, yellow, green). The main content area displays a bulleted list of five family names, each in a bold, black, sans-serif font. The list items are:

- **Decker Family**
- **Jones Family**
- **Simmons Family**
- **Jackson Family**
- **Lowe Family**



Id

```
1  {% person id:'5' %}  
2      {% for person in personItems %}  
3          {{ person.FullName }}  
4      {% endfor %}  
5  {% endperson %}
```





Id

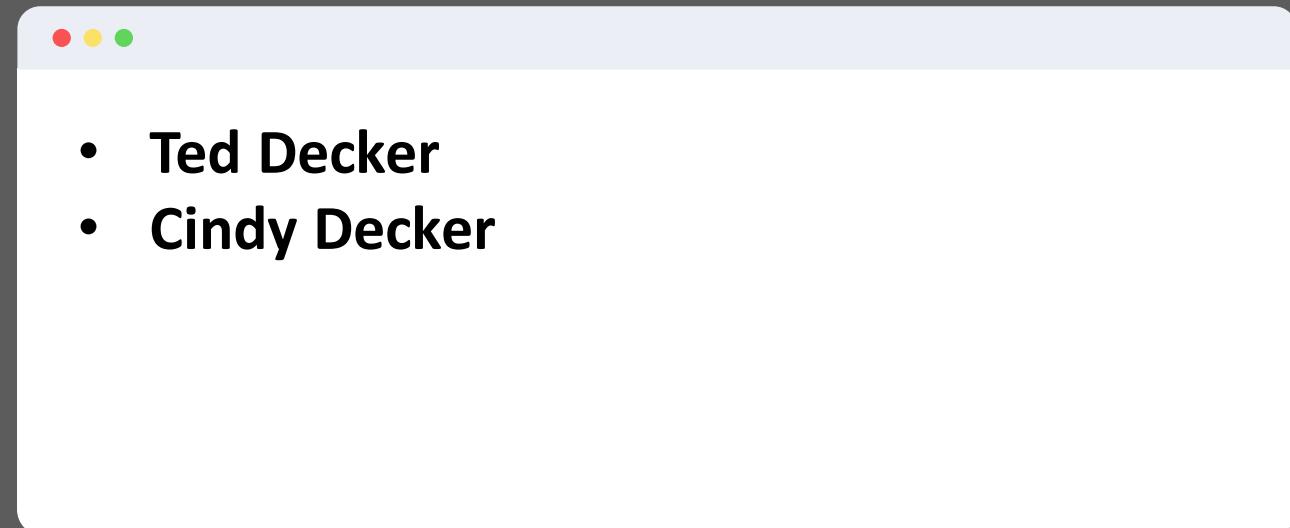
```
1  {% person id:'5' %}  
2    {{ person.FullName }}  
3  {% endperson %}
```





Id

```
1  {% person ids:'5,6' %}  
2    <ul>  
3      {% for p in personItems %}  
4          <li>{{ p.FullName }}</li>  
5      {% endfor %}  
6    </ul>  
7  {% endperson %}
```





Dataview

```
1  {% person dataview:'1' %}  
2    <ul>  
3      {% for p in personItems %}  
4          <li>{{ p.FullName }}</li>  
5      {% endfor %}  
6    </ul>  
7  {% endperson %}
```





Dataview

```
1  {% person dataview:'1' where:'Gender == "Female"' %}  
2    <ul>  
3      {% for p in personItems %}  
4        <li>{{ p.FullName }}</li>  
5      {% endfor %}  
6    </ul>  
7  {% endperson %}
```

The screenshot shows a Mac OS X application window with a white background and a title bar featuring the standard red, yellow, and green buttons. Inside the window, there is a single bullet-pointed list of five names, all of which are bolded:

- **Cindy Decker**
- **Sarah Simmons**
- **Mariah Jackson**
- **Alisha Marble**
- **Pamela Foster**



Sort

```
1  {% person dataview:'1' where:'Gender == "Female"' sort:'LastName, FirstName desc' %}  
2    <ul>  
3      {% for p in personItems %}  
4        <li>{{ p.FullName }}</li>  
5      {% endfor %}  
6    </ul>  
7  {% endperson %}
```





Offset and Limit

```
1  {% person dataview:'1' where:'Gender == "Female"' sort:'LastName, FirstName desc' offset:'5' limit:'5' %}  
2    <ul>  
3      {% for p in personItems %}  
4        <li>{{ p.FullName }}</li>  
5      {% endfor %}  
6    </ul>  
7  {% endperson %}
```





Offset and Limit

```
1  {% person
2      dataview:'1'
3      where:'Gender == "Female"'
4      sort:'LastName, FirstName desc'
5      offset:'5'
6      limit:'5' %}
7  <ul>
8      {% for p in personItems %}
9          <li>{{ p.FullName }}</li>
10     {% endfor %}
11 </ul>
12 {% endperson %}
```



The screenshot shows a window with a title bar containing three colored dots (red, yellow, green). The main content area displays a bulleted list of five names:

- **Jenny Michaels**
- **Becky Peterson**
- **Sarah Simmons**
- **Corrie Stone**
- **Nancy Sweeny**



Security

```
1  {% group where:'GroupTypeId == "25"' %}  
2    <ul>  
3      {% for g in groupItems %}  
4          <li>{{ g.Name }}</li>  
5      {% endfor %}  
6    </ul>  
7  {% endperson %}
```





Security

```
1  {% group where:'GroupTypeId == "25"' securityenabled:'false' %}  
2    <ul>  
3      {% for g in groupItems %}  
4        <li>{{ g.Name }}</li>  
5      {% endfor %}  
6    </ul>  
7  {% endperson %}
```





Iterator

```
1  {% group where:'GroupTypeId == "25"' securityenabled:'false' iterator: Groups %}
2  <ul>
3      {% for group in Groups %}
4          <li>{{ group.Name }}</li>
5      {% endfor %}
6  </ul>
7  {% endperson %}
```





Power Tools

For Speed and Cross-Entity Filtering





Expression

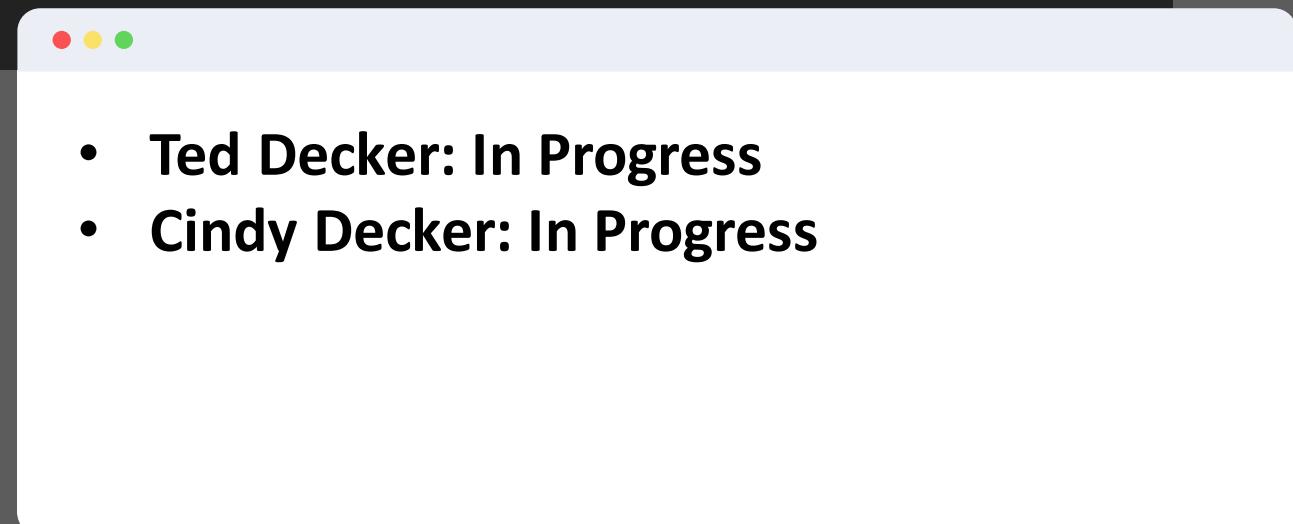
```
1  {% step where:'StepTypeId == 2'
2      expression:'PersonAlias.Person.AgeClassification == 1'
3      securityenabled:'false' %}

4

5  {% for step in stepItems %}
6      <ul>
7          <li>{{ step.PersonAlias.Person.FullName }}: {{ step.StepStatus }}</li>
8      </ul>
9  {% endfor %}

10

11  {% endstep %}
```





Expression

```
1  {% step where:'StepTypeId == 2'  
2      expression:'PersonAlias.Person.AgeClassification == 1'  
3      securityenabled:'false' %}  
4  
5      {% for step in stepItems %}  
6          <ul>  
7              <li>{{ step.PersonAlias.Person.FullName }}: {{ step.StepStatus }}</li>  
8          </ul>  
9      {% endfor %}  
10  
11  {% endstep %}
```





Lazy Loading

```
1  {% step where:'StepTypeId == 2'  
2      expression:'PersonAlias.Person.AgeClassification == 1'  
3      securityenabled:'false'  
4      lazyloadenabled:'false' %}  
5  
6  {% for step in stepItems %}  
7      <ul>  
8          <li>{{ step.Id }} {{ step.PersonAlias.Person.FullName }}: {{ step.StepStatus }}</li>  
9      </ul>  
10     {% endfor %}  
11  
12  {% endstep %}
```





Include

```
1  {% step where:'StepTypeId == 2'  
2      expression:'PersonAlias.Person.AgeClassification == 1'  
3      securityenabled:'false'  
4      lazyloadenabled:'false'  
5      include:'PersonAlias.Person, StepStatus' %}  
6  
7  {% for step in stepItems %}  
8      <ul>  
9          <li>{{ step.PersonAlias.Person.FullName }}: {{ step.StepStatus }}</li>  
10     </ul>  
11  {% endfor %}  
12  
13  {% endstep %}
```





Select

```
1  {% step where:'StepTypeId == 2'  
2      expression:'PersonAlias.Person.AgeClassification == 1'  
3      select:'new( PersonAlias.Person.NickName as NickName,  
4                  PersonAlias.Person.LastName as LastName,  
5                  StepStatus.Name as Status)'  
6      securityenabled:'false' %}  
7  
8  {% for step in stepItems %}  
9      <ul>  
10         <li>{{ step.NickName }} {{ step.LastName }}: {{ step.Status }}</li>  
11     </ul>  
12  {% endfor %}  
13  
14  {% endstep %}
```





Select

```
1  {% step where:'StepTypeId == 2'
2      expression:'PersonAlias.Person.AgeClassification == 1'
3      select:'new( PersonAlias.Person.NickName as NickName,
4                  PersonAlias.Person.LastName as LastName,
5                  StepStatus.Name as Status)'
6      securityenabled:'false' %}
7
8  <pre>{{ stepItems | ToJSON }}</pre>
9
10 {% endstep %}
```

```
1  [
2    {
3      "NickName": "Ted",
4      "LastName": "Decker",
5      "Status": "In Progress"
6    },
7    {
8      "NickName": "Cindy",
9      "LastName": "Decker",
10     "Status": "In Progress"
11   }
12 ]
```



Select

```
1  {% step where:'StepTypeId == 2'
2      expression:'PersonAlias.Person.AgeClassification == 1'
3      select:'new{ PersonAlias.Person.NickName as NickName,
4                  PersonAlias.Person.LastName as LastName,
5                  PersonAlias.Person.PhoneNumbers.Where( NumberTypeValueId == 12 ).Select( new( NumberFormatted as Number ) ) as Phones,
6                  StepStatus.Name as Status}'
7      securityenabled:'false' %}
8
9  {% for step in stepItems %}
10     <ul>
11         <li>{{ step.NickName }} {{ step.LastName}}: {{ step.Phones | First | Property:'Number' }}</li>
12     </ul>
13
14     {% endfor %}
15
16  {% endstep %}
```

The screenshot shows a window with a title bar and three red, yellow, and green buttons. Inside the window, there is a list of two items:

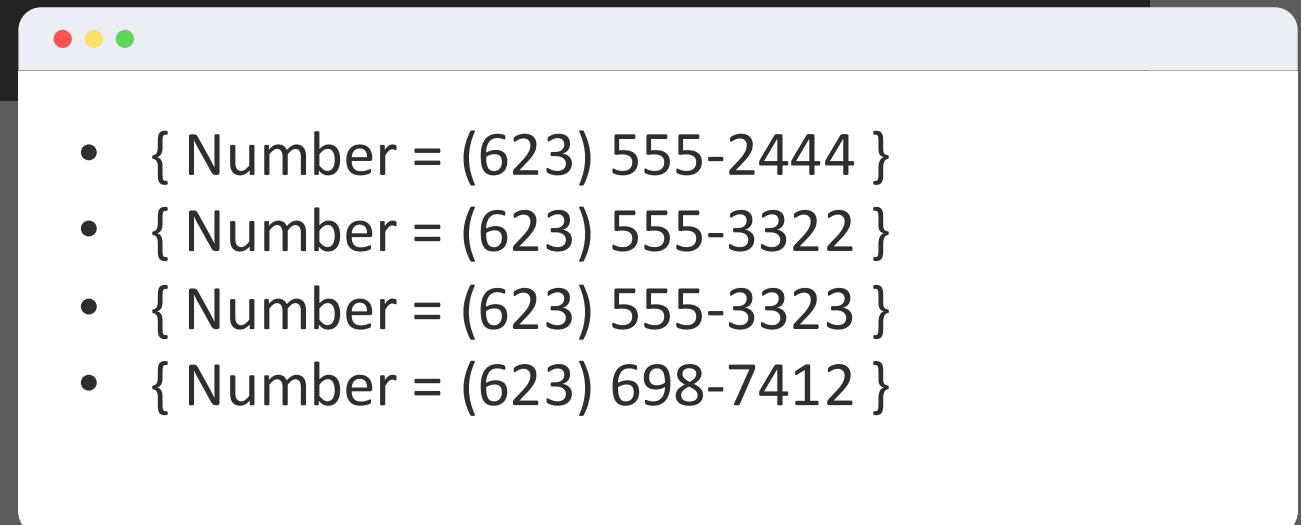
- **Ted Decker: (623) 555-3322**
- **Cindy Decker: (623) 555-3323**



Select Many

```
1  {% step where:'StepTypeId == 2'
2      expression:'PersonAlias.Person.AgeClassification == 1'
3      selectmany:'PersonAlias.Person.PhoneNumbers.Select( new( NumberFormatted as Number )).Distinct()'
4      securityenabled:'false' %}

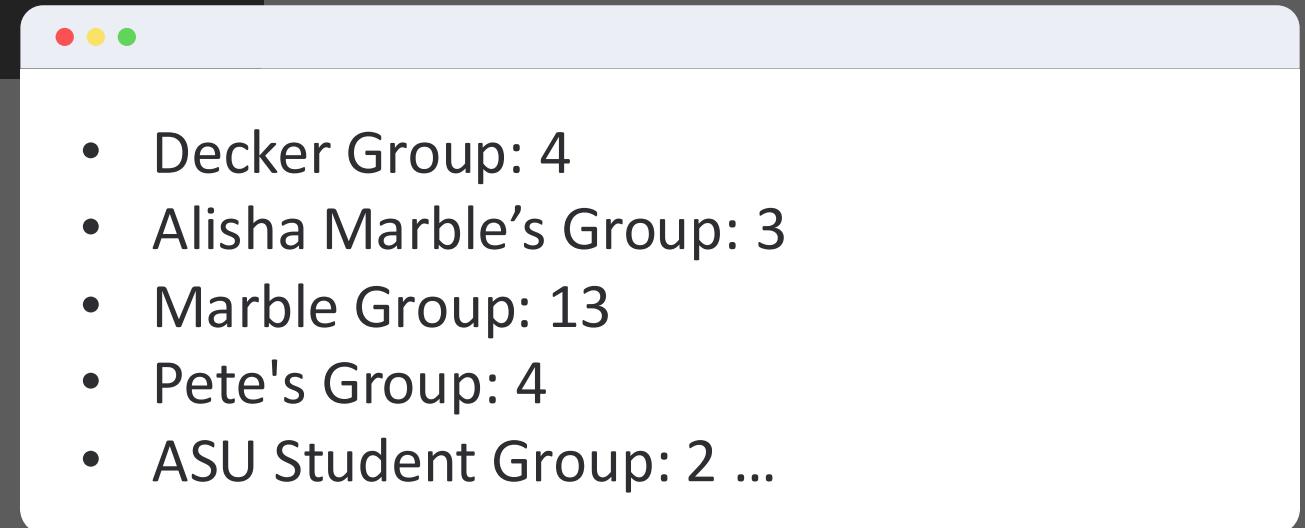
5
6  <ul>
7      {% for step in stepItems %}
8          <li>
9              {{ step }}
10         </li>
11     {% endfor %}
12   </ul>
13
14  {% endstep %}
```





Aggregates

```
1  {% group where:'GroupTypeId == 25'  
2      select:'new( Name, Members.Count() as MemberCount )'  
3      securityenabled:'false' %}  
  
4  
5  {% for group in groupItems %}  
6      <ul>  
7          <li>  
8              {{ group.Name }}: {{ group.MemberCount }}  
9          </li>  
10         </ul>  
11     {% endfor %}  
12  
13  {% endgroup %}
```



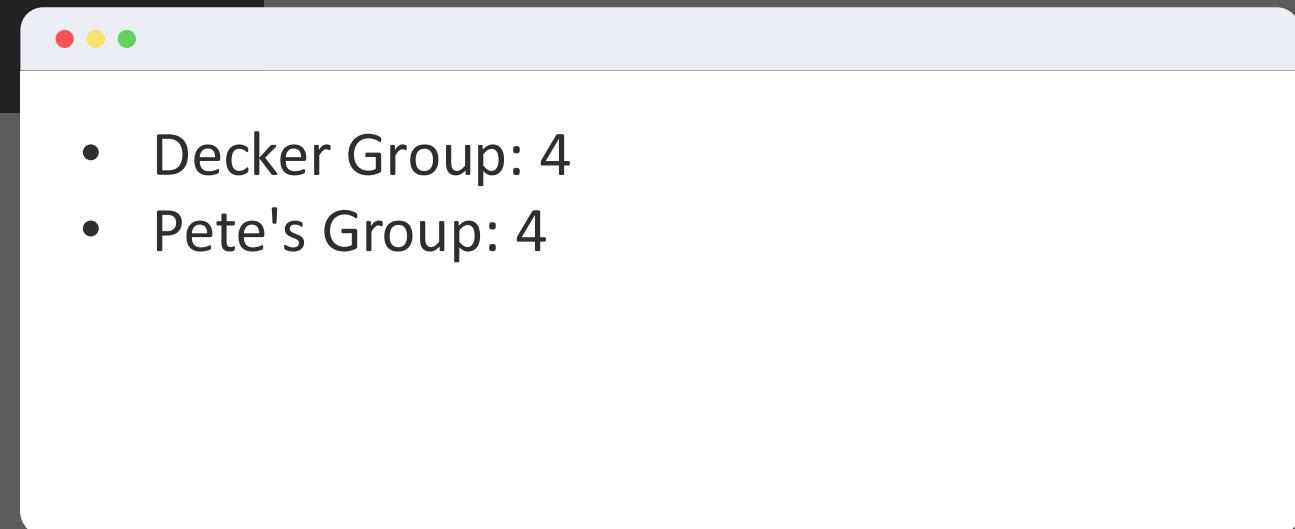
The screenshot shows a Mac OS X application window with a title bar featuring red, yellow, and green buttons. The main content area displays a bulleted list of groups and their member counts:

- Decker Group: 4
- Alisha Marble's Group: 3
- Marble Group: 13
- Pete's Group: 4
- ASU Student Group: 2 ...



Aggregates

```
1  {% group where:'GroupTypeId == 25'  
2      expression:'Members.Count() == 4'  
3      select:'new( Name, Members.Count() as MemberCount )'  
4      securityenabled:'false' %}  
5  
6  {% for group in groupItems %}  
7      <ul>  
8          <li>  
9              {{ group.Name }}: {{ group.MemberCount }}  
10         </li>  
11     </ul>  
12  {% endfor %}  
13  
14  {% endgroup %}
```





Groupby

```
1  {% groupmember groupby:'new( GroupId, Group.Name)'  
2      where:'GroupTypeId == 10'  
3      select:'new( Key as Key, It as GroupMembers)'  
4      securityenabled:'false'  
5      limit:3 %}  
6  
7      {% for family in groupmemberItems %}  
8          {{ family.Key.Name }}  
9          <ul>  
10             {% for member in family.GroupMembers %}  
11                 <li>{{ member.Person.NickName }} {{ member.Person.LastName }}</li>  
12             {% endfor %}  
13         </ul>  
14     {% endfor %}  
15  
16  
17  {% endgroupmember %}
```

The screenshot shows a window with three colored window control buttons (red, yellow, green) at the top right. The main content area displays the word "Admin" in large bold letters. Below it is a bulleted list: "• Admin Admin". Underneath this, the words "Anonymous Family" are displayed. Another bulleted list follows: "• Giver Anonymous". Below this, the words "Presence Family" are shown. A final bulleted list concludes the content: "• Presence Presence".

Admin

- Admin Admin

Anonymous Family

- Giver Anonymous

Presence Family

- Presence Presence



Groupby

```
1  {% groupmember groupby:'new( GroupId, Group.Name )'
2      where:'GroupTypeId == 10'
3      select:'new( Key as Key,
4                  Select( new(Person.Nickname as NickName, Person.LastName as LastName ) ) as GroupMembers )'
5      securityenabled:'false'
6      limit:3 %}
7
8  {% for family in groupmemberItems %}
9    <p>{{ family.Key.Name }}</p>
10   <ul>
11     {% for member in family.GroupMembers %}
12       <li>{{ member.NickName }} {{ member.LastName }}</li>
13     {% endfor %}
14   </ul>
15 {% endfor %}
16
17 {% endgroupmember %}
```

A screenshot of a Mac OS X application window titled "Admin". The window contains a list of family names: "Admin Admin", "Anonymous Family", "Giver Anonymous", "Presence Family", and "Presence Presence". Each item is preceded by a small red dot icon.

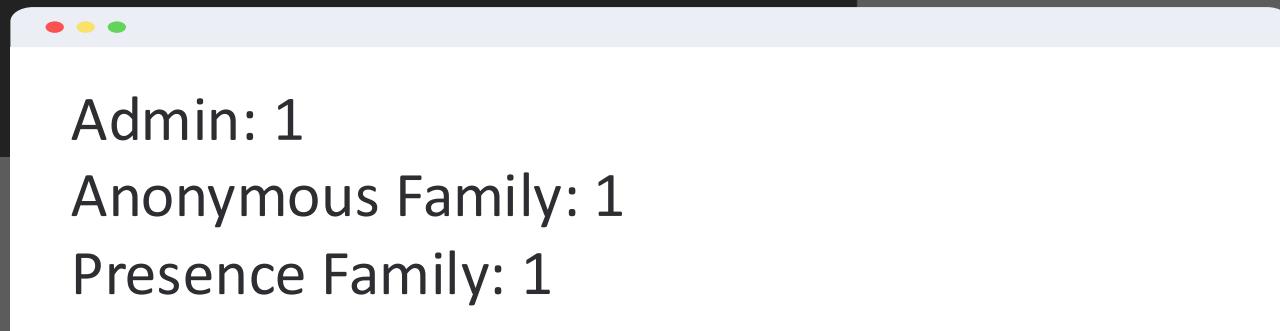
Admin

- Admin Admin
- Anonymous Family
- Giver Anonymous
- Presence Family
- Presence Presence



Groupby with Aggregates

```
1  {% groupmember groupby:'new( GroupId, Group.Name )'
2          where:'GroupTypeId == 10'
3          select:'new( Key as Key, Count() as MemberCount )'
4          securityenabled:'false'
5          limit:3 %}
6
7  {% for family in groupmemberItems %}
8      {{ family.Key.Name }}: {{ family.MemberCount }}<br/>
9
10 {% endfor %}
11
12 {% endgroupmember %}
```



The screenshot shows a Mac OS X application window with three colored status bar buttons (red, yellow, green) at the top. The main area displays the output of the R code. The output consists of three lines of text: "Admin: 1", "Anonymous Family: 1", and "Presence Family: 1".

```
Admin: 1
Anonymous Family: 1
Presence Family: 1
```