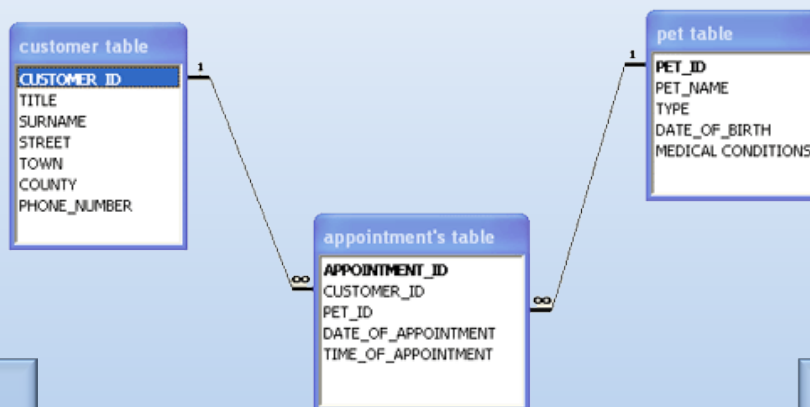




Intermediate: Site Building

# Florida DrupalCamp 2014 Parent/Child Data Modeling



- Absolutely no code
- No custom entities
- No ECK

- User thought UI's
- Tiny site footprint
- Super expand-able

# Florida DrupalCamp 2014

Parent/Child Data Modeling



## Jay Epstein

- Senior Developer with Traktek Partners, LLC.
- Co-Organizer: Broward Drupal Users Group
- Mentor: Drupal Career Starter Program
- 3+ Years working with Drupal
- Over 20 Years in IT



<http://browarddrupal.org>



<http://meetup.com/broward-drupal>

@jeppy64



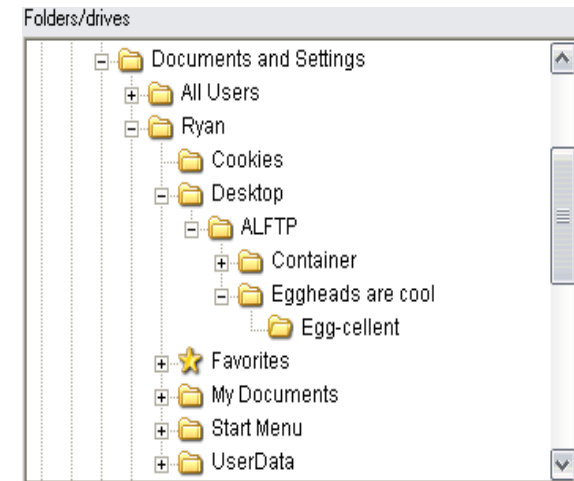
**BROWARD DRUPAL**  
USERS GROUP

Jay-epstein.com



## What you can look forward to today:

- The toolbox ( smaller than you'd expect ! )
- The **top down hierarchy** approach to content types.
- UI's that are **user thought process** driven.
- Visualizing your content in relational terms.
- Views and displaying fields based on relationships.
- Saying goodbye to taxonomy ( Relying on groups of content ).





## The toolbox:

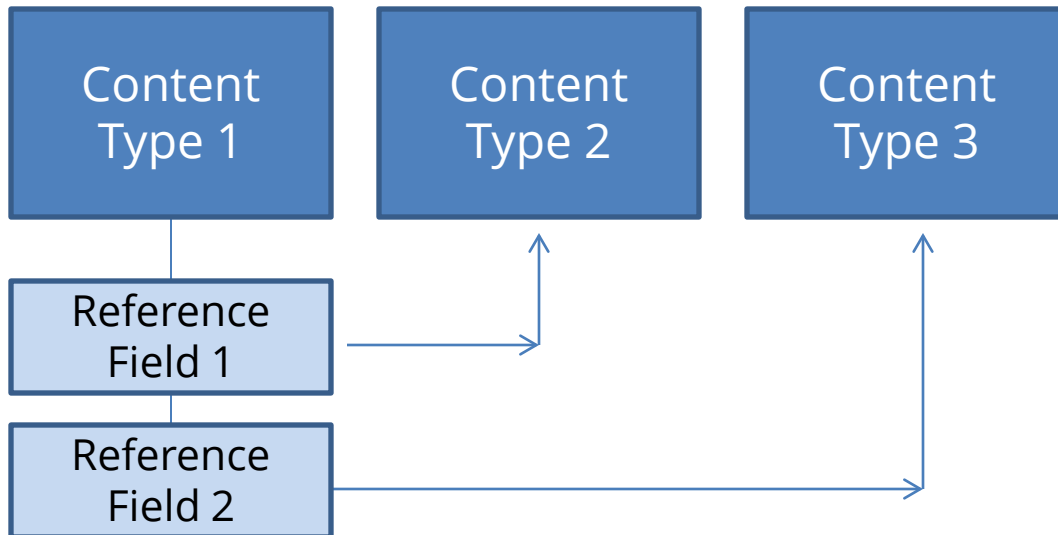
### Modules:

- Entity API: <https://drupal.org/project/entity>
- Entity Reference: <https://drupal.org/project/entityreference>
- Entity Connect: <https://drupal.org/project/entityconnect>
- Auto Entity Label: [https://drupal.org/project/auto\\_entitylabel](https://drupal.org/project/auto_entitylabel)
- Views: <https://drupal.org/project/views>



## The Comparison: The Flat Approach

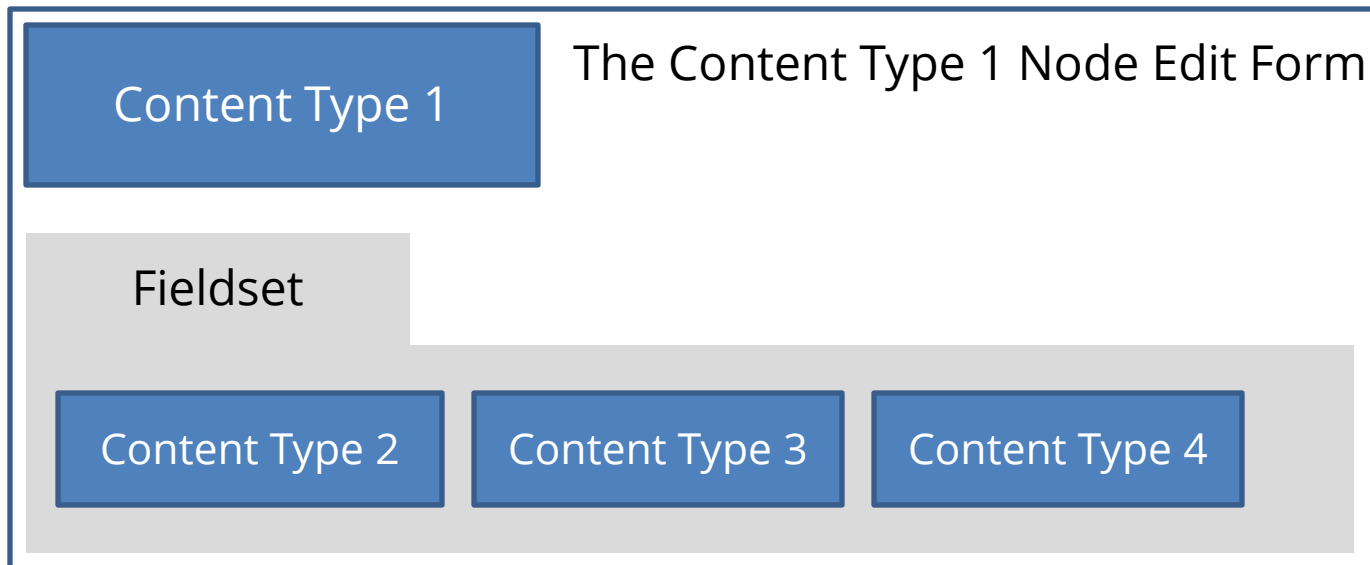
- Content types reside on a **level thought** architecture.
- Reference fields, allow for **data** to tie or group ( usually by taxonomy ).





## The Comparison: The Hierarchical Approach

- Content types reside on a **multi-level visual** architecture.
- Reference fields as entity connect fields, allow for **content types** to tie or group.



# Florida DrupalCamp 2014

## Parent/Child Data Modeling



### Our mission:

Build a new data model for a cruise ship company that sells River Cruises.

*Hundreds of trips.*

*Deep relational data per trip. ( 10 content types deep per cruise... minimum ~ WHOA ! )*

*Bolt on ability for future functions without affecting existing site content... ( gulp, sigh )*

### **Pain Points on Current System**

- Node UI's confusing.
- Non intuitive content list displays.
- Having to "hunt" for content ( with thousands of nodes ).
- Marathon data entry tasks.
- Down time for adding enhancements.

@jeppy64



**BROWARD DRUPAL**  
USERS GROUP

Jay-epstein.com



We need to think differently on how we ask a user to interact with Drupal.

We need to give users a thought process driven UI.



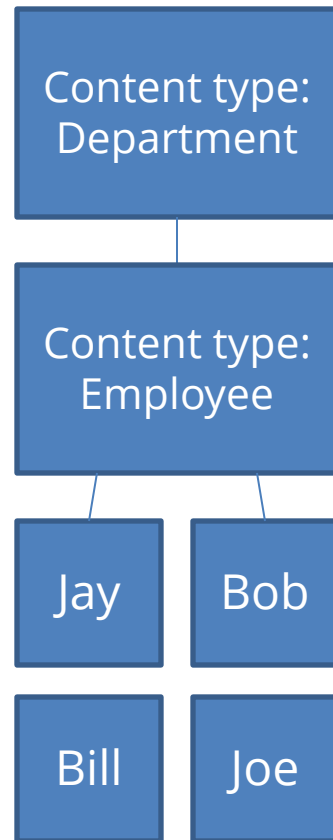


## The Approach: Old School Relational Data Let's call it: Parent/Child Content Types

example: **One** Department can have **Many** employees.

### How To: Relational model building with hierarchy

- Identifying "Parents and their Children".
- Identifying "Siblings".
- Sketching the "family" on paper ( like a genealogy tree )
- Transposing to Drupal through content types and fields.





## Step 1: Content Type Inventory:

Document the content types and short descriptions of why they exist.

### Example:

**Cruise:** The master content type. Container of itself and everything related to a cruise.

**Itinerary:** Describes the length of the trip, destinations, vessel, meals, cabins, upgrades, etc.

**Itinerary Day:** Describes each day of the itinerary in detail.



## Step 2: Identifying Parents & Children:

Document the content relations in a hierarchal fashion.

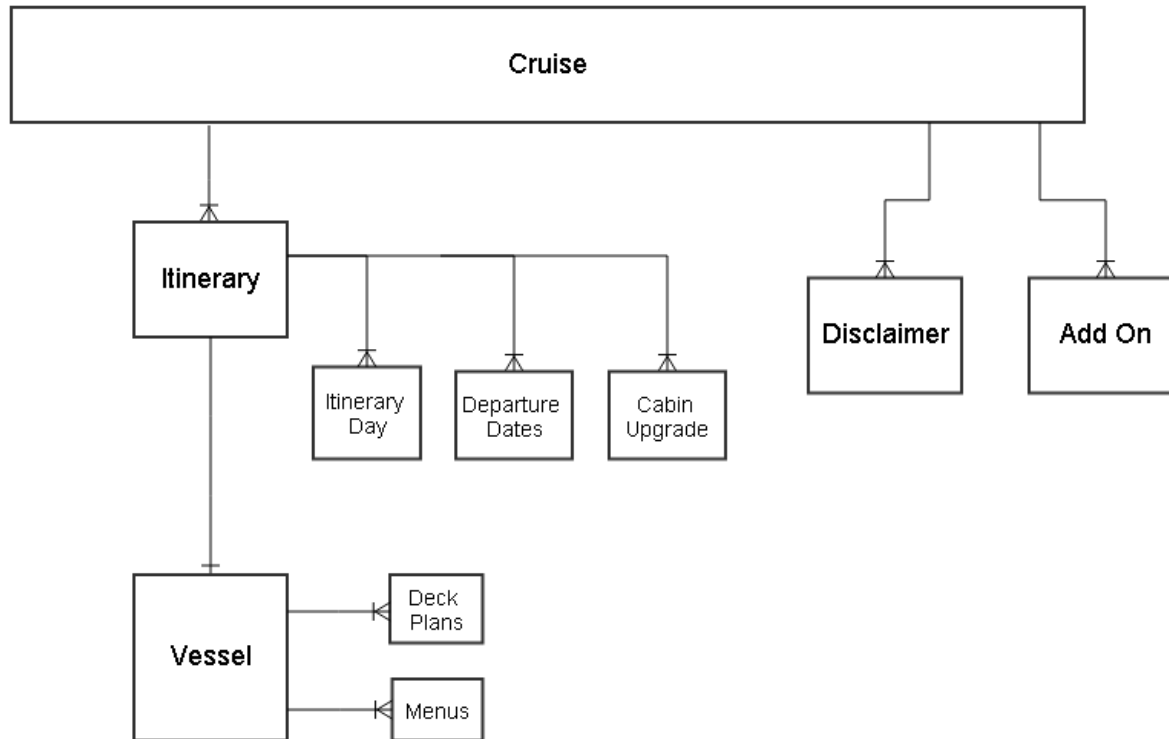
**Cruise:** The master content type. ( AKA The Parent )

**Itinerary:** ( child of cruise ) ( 1 to many ) (one cruise can have many itineraries )

**Itinerary Day:** ( child of itinerary ) ( 1 to many ) ( one itinerary can have many days )



### Step 3: Create the tree:





## Step 4: It's Drupal Time:

- A well documented data model makes the Drupal part a breeze
- Documenting outside of Drupal allows for a clean, one shot system creation

## One Last Thought:

- Machine names: Ensure matching machine names on content types and fields
- This makes views interaction super simple

# Florida DrupalCamp 2014

## Parent/Child Data Modeling



## Fields

LABEL	MACHINE NAME	FIELD TYPE	WIDGET	OPERATIONS
+ Cruise Build Out	group_cruise_buildout	Horizontal tabs group	htabs classes group-cruise-buildout field-group-htabs	<input type="button" value="edit"/> delete
+ Itinerary	group_cruise_itin	Horizontal tab	htab closed classes group-cruise-itin field-group-htab required_fields yes	<input type="button" value="edit"/> delete
+ Itinerary	field_cruise_intinerary	Entity Reference	Autocomplete	edit delete
+ Reverse Itinerary	group_cruise_revitin	Horizontal tab	htab closed classes group-cruise-revitin field-group-htab required_fields yes	<input type="button" value="edit"/> delete
+ Reverse Itinerary	field_cruise_revitin	Entity Reference	Autocomplete	edit delete
+ CTA	group_cruise_cta	Horizontal tab	htab closed classes group-cruise-cta field-group-htab required_fields yes	<input type="button" value="edit"/> delete
+ CTA	field_cruise_cta	Entity Reference	Select list	edit delete



## Fields

Use sitewide default ▾

Choose the default HTML for this field.

### Display Entity Connect "add" button.

Yes

No

Choose "No" if you want to unload "add" button for the field

### Display Entity Connect "edit" button.

Yes

No

Choose "No" if you want to unload "edit" button for the field



## Views

**TITLE**  
Title: [Day by Day](#)

**FORMAT**  
Format: [Table](#) | [Settings](#)

**FIELDS** [Add](#) ▾

[Content: Day Image](#)  
(Content entity referenced from field\_itin\_day)

[Content: Day Number](#)  
(Content entity referenced from field\_itin\_day)

[Content: Day City \(City\)](#)  
(Content entity referenced from field\_itin\_day)

[Content: Day Description \(Description\)](#)

**FILTER CRITERIA** [Add](#) ▾

[Content: Published \(Yes\)](#)

[Content: Type \(= Cruise\)](#)

**BLOCK SETTINGS**  
Block name: [Itin Day Block](#)

Access: [Permission](#) | [View published content](#)

**HEADER** [Add](#)

**FOOTER** [Add](#)

**PAGER**  
Use pager: [Display all items](#) | [All items](#)

More link: [No](#)





## Views

### BLOCK SETTINGS

Block name: [Itin Day Block](#)

Access: [Permission](#) | [View published content](#)

**HEADER**

[Add](#)

**FOOTER**

[Add](#)

**PAGER**

Use pager: [Display all items](#) | [All items](#)

More link: [No](#)

### ADVANCED

**CONTEXTUAL FILTERS**

[Add](#)

Content: [Nid](#)

**RELATIONSHIPS**

[Add](#)

Entity Reference: [Intinerary](#)

(Content entity referenced from [field\\_cruise\\_intinerary](#))

Entity Reference: [Reverse Itinerary Day](#)

**NO RESULTS BEHAVIOR**

[Add](#)

**EXPOSED FORM**

Exposed form style: [Basic](#) | [Settings](#)

**OTHER**

Machine Name: [block](#)


Comment: [No comment](#)

Use AJAX: [No](#)



## Views

### Add fields

 There is a security update available for your version of Drupal. To ensure the security of your server, you should update immediately! See the [available updates](#) page for more information and to install your missing updates.

For All displays (except overridden) ▼

Search  Filter - All - ▼

- Content: Day City  
Appears in: node:itin\_day.
- Content: Day Description  
Appears in: node:itin\_day.
- Content: Day Image



**Thanks for watching 😊**

Florida DrupalCamp 2014  
**Parent/Child Data Modeling**

Intermediate: Site Building

@jeppy64



**BROWARD DRUPAL  
USERS GROUP**

Jay-epstein.com