Fluxx Data Structure Simplified

The information here is mostly metaphorical examples of real life to help explain complex database concepts in regards to Fluxx data model to serve a more general audience.

Tables

Imagine the database being made out of different spreadsheets which we call Tables. Every row on that table is representing one record of a type represented by that table. For example, if we are adding a Testing User (First Name: Testing, Last Name: User) as a contact to our database, an example of what that looks like can be seen below.

Table would be People (Users). It would look like this:

Person ID	Last Name	First Name	Prefix	Email	Salutation	Suffix	User Title	Work Phone	Mobile Phone	Primary Organization	Primary Title	Personal Street Address	Personal Street Address2	Personal City	State	Country	Postal Code	Date Created	Date Updated	Highest Degree Year	UW Department	Primary Affiliation	Highest Degree	Prefix Dynamic	Track	Co-Investigator / Collaborator / Academic Partner Affiliation	Department	Role Name	Citizenship Status	SMPH Division	Primary Affiliation Other	office_telephone	office_fax	office_extension	employee_id	department_division	degree	academic_partner_affiliation_other	Job Title	Individual Vendor Number	Middle Name	Bulk Update Test Field	Alias (or AKA)	Biographical Statement
3055	User	Testing		example email@wisc.edu				(608) 123-9874		One Test Organization	Chief Executive Officer							9/20/2021	2/11/2022																									

Person ID is a unique number that has no particular meaning other than uniquely identifying the record in that table/spreadsheet in our database.

In the Fluxx database, the column names are not the same as the labels we see in our views; column names are technically the **"back-end names**" of these fields (e.g. office_telephone and not "Office Telephone Number"). Everywhere else, we will see the human readable labels.

Links

Let's start this section with a story:

When we invite a guest to our house, we prepare for them. We make sure that we make space for them to come and we expect them to come. The other way around is not the case; they do not expect us to go to their house. When they show up to our house, we know what their name is and other information related to them. So, if anyone asks us who is your guest, we are able to tell them the name of our guest and their relationship to us and also their food allergies, because we were expecting them and we were cooking for them.

Imagine at the same time some birds flying and sitting on our roof. We don't expect them and we don't even know they exist. So, if anyone asks us about those birds, we would probably say "what birds?!". The birds however, know about our house. They were expecting to sit on a roof and that's what they did. There is only one way relationship between our house and the birds. Our house is going to be our house with or without the birds. But the birds need a roof to sit on.

The same idea is happening in our database. As we mentioned in Tables section, we can imagine tables as spreadsheets. So, like spreadsheets, a Request spreadsheet would not know anything about Organization spreadsheets unless we connect them together somehow. In databases, we can ***Link*** th e spreadsheets that have relationship together. For example, a Request would like to know which Organization it belongs to. So, there is an empty seat in Requests spreadsheets that refers to the related Organization:

Organizations spreadsheet:

Contig ID (AKA Org ID)	Name	Acronym	Legal Name	Street Address	Street Address 2	City	State	Country	Postal Code	Phone	Email	Website	Date Created	Date Updated	Tax Class	Tax ID	tax_registration_date	Organization Total Budget	number_of_paid_staff	number_of_full_time_equivalent	Organization Type	DBA Test	organization_category_other
1665	**One Test Organization**			1 Test Road	Apt 123	Madison	Wisconsin	United States	53711	(608) 123-4567	testing@wisc.edu		9/20/2021	2/4/2022		XX-XXXXXXX							

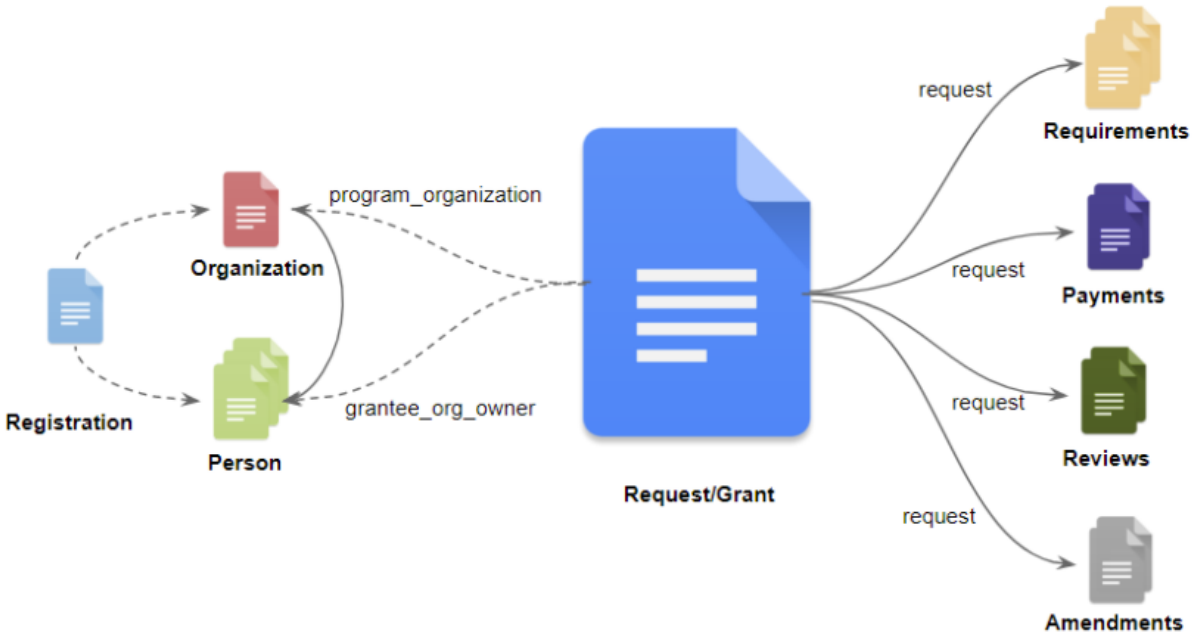
Requests spreadsheet:

Request ID	Grantee Organization	Project Title	Request Type	Status	Grant Start Date	Grant End Date	Amount Requested	Amount Funded	Base Request ID	Document Types	Fiscal Year	Grantee Org. Street Address	Grantee org Street Address2	Grantee Org City	Contig ID	Grantee Org State	Grantee Org Country	Grantee Org Postal Code	Grantee Org URL	Grantee Org Tax ID	Fiscal Organization	Fiscal Street Address	Fiscal Street Address2	Fiscal City	Fiscal State	Fiscal Country	Fiscal Postal Code	Fiscal URL	Fiscal Tax ID	Lead PO /PD	Secondary Lead	Program	Sub Program	Initiatives	Sub Initiatives	Date Request Received	Duration	Funding Source	Final Proposal Date	Final Budget Date	Date Created	Date Last Updated	Primary Contact First Name	Primary Contact Last Name	Primary Contact Email	Program Lead First Name	Program Lead Last Name	Program Lead Contact Email	Sgo Fr Na
R-5069-2022	One Test Organization	Testing December 2021 Strategic Grant Application	December 2021 PERC Strategic Grant	Loi Draft	12/31/2021	12/30/2022	$ 1.00		5069-2022			1 Test Road	Apt 123	Madison	980	Wisconsin	United States	53711		XX-XXXXXXX										WPP Program Officer Name		Partnership Education and Research Committee	PERC Strategic			12/8/2021	12				12/8/2021	1/20/2022	Testing	User	testing@wisc.edu	Program Officer First Name	Program Officer Last Name	Program Officer Email@wisc.edu	Te ti

In other words, there is <u>one</u> empty space for an organization in a certain request. That is why we can only have one organization for a request.

Organization however, is independent from the request. Requests are like birds on its roof. It doesn't even know that they are there (we can retrieve the information when we do a query).

The following chart is showing the main linking that is in place in Fluxx Grantmaking Database:


request
Requirements
program_organization
Organization
request
Payments
Registration
grantee_org_owner
request
Reviews
Person
Request/Grant
request
Amendments