Post-Products Spring Cleaning

Charlotte Hausman asked an interesting question: once we have adapted to the new product system, what parts of the current database schema will be going away? This page is meant to help us work that out, and one we have agreement, make a ticket to track it, so we can remove those bits once the new stuff is functional.

I think there are some guidelines we can use to come up with the list of what will go away:

- Access to any file goes through a filegroup
- Access to any filegroup is through the science products table or the ancillary products table
- Tying a science product to a project goes through the many to many relationship table

So here are some things I think will go away/need to be changed:

table.column	notes	Objections
execution_blocks.product_code	obsolete	
execution_blocks.filegroup_id	obsolete	
images.file_id	obsolete	
calibrations.project_code	obsolete	
calibrations.filegroup_id	obsolete	
subscans.file_id	VLASS thought it was needed, never been used?	Indexer needs this for VLBA. Only way to link subscans and data descriptions to filegroup_id since execution_block_id is not unique in this case
scans.filegroup_id	obsolete	
scans.filename	never been used?	
image_products	turn into ancillary products, transfer configurations to the relevant image(s)	
images.thumbnail	turn into ancillary products and teach the files table about thumbnail paths	
intents	empty table, never used?	
subscan_intents	empty table, never used?	
execblock_start_stop	empty table, never used?	

This isn't an exhaustive list, feel free to add to it.

Since we'll be linking from a science_product or ancillary_product to a filegroup and filegroups have files, I think we should decide on whether we need or want the recursive filegroup structure. I don't see a real need for it any more, this new stuff replaces what it did and does it better. The requirements for the product system don't definitively say one way or the other.