

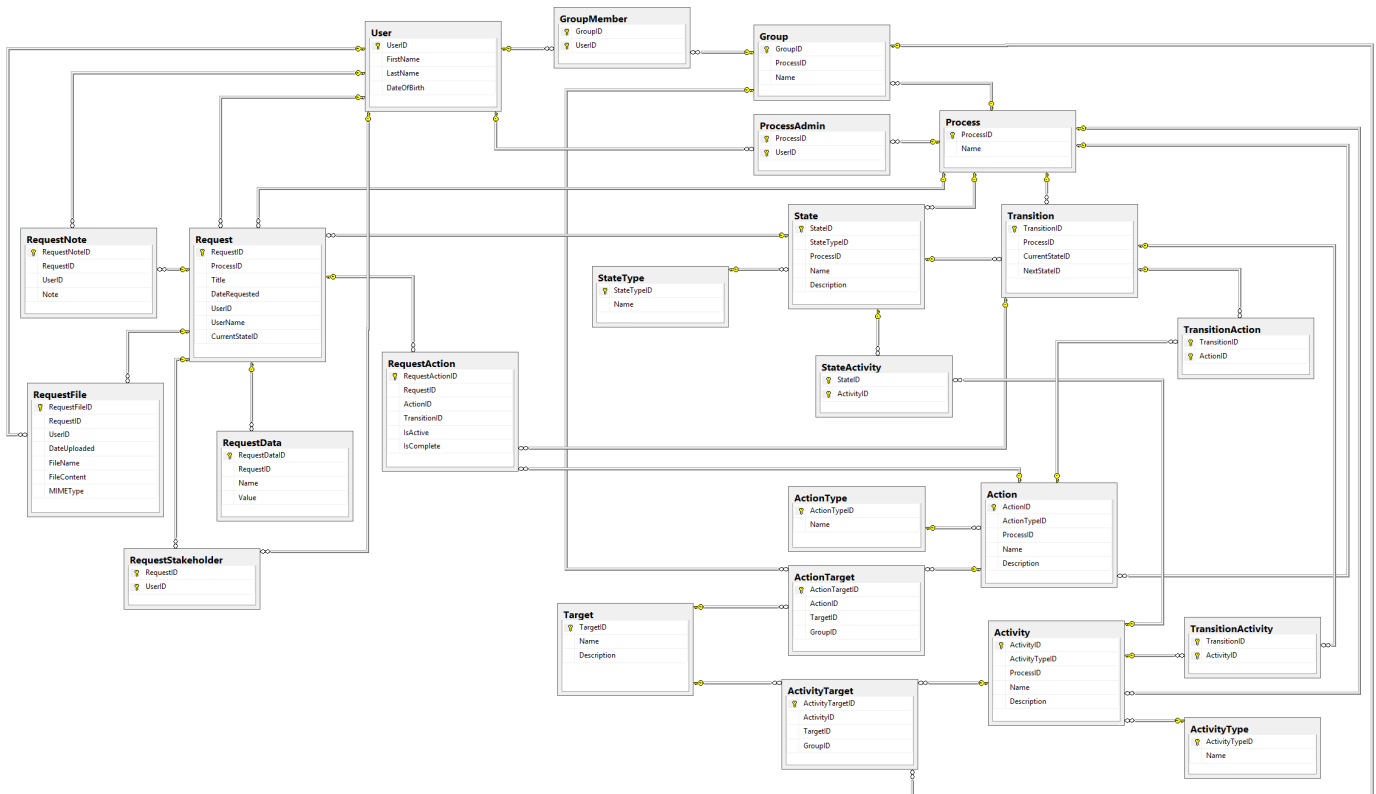
# Exception Not Found

 [exceptionnotfound.net/designing-a-workflow-engine-database-part-8-complete-schema-and-shortcomings](http://exceptionnotfound.net/designing-a-workflow-engine-database-part-8-complete-schema-and-shortcomings)

April 22, 2015

*This is the final part of an eight-part series describing how to design a database for a Workflow Engine. Click here for Part 1*

We've now completed the database design for our Workflow Engine, so let's take a look at the final database diagram (click here for a larger view):



That's a lot of tables! But we're trying to design a generic engine, one that can be used for many disparate processes, and as such the complexity of this design reflects the complexity of the requirements.

## Shortcomings of This Design

There are a few shortcomings of this design:

- Currently, there's no way to say "Stay in this state for X days; after X days, move to another state." We cannot cause Requests to move along in the Process due to time elapsed.
- We don't have a procedure in place to handle a State or Transition ceasing to exist (meaning, even though the Process Admins are supposed to be able to change the Process itself, what happens when they actually do that?)
- The RequestFiles table stores the actual files in the table itself; a more performant way to do this would be to store the files on a filestore somewhere and just keep references to

them in this table.

- We don't currently have a way to designate that a specific individual needs to perform an Action or receive an Activity.

That said, most of these shortcomings are fairly easy to implement; we actually did solve a few of them for our company's workflow engine.

Remember: **this is only a database schema**. The design doesn't account for how this service would actually be implemented in code.

Whew! That was a lot of steps! But what we ended up with was a **generic, extensible, reusable workflow engine database design** that can handle a huge variety of processes.

Thanks for reading! If you found this series helpful (or just need to yell at me for something I did incorrectly) let me know in the comments.

Happy Coding!