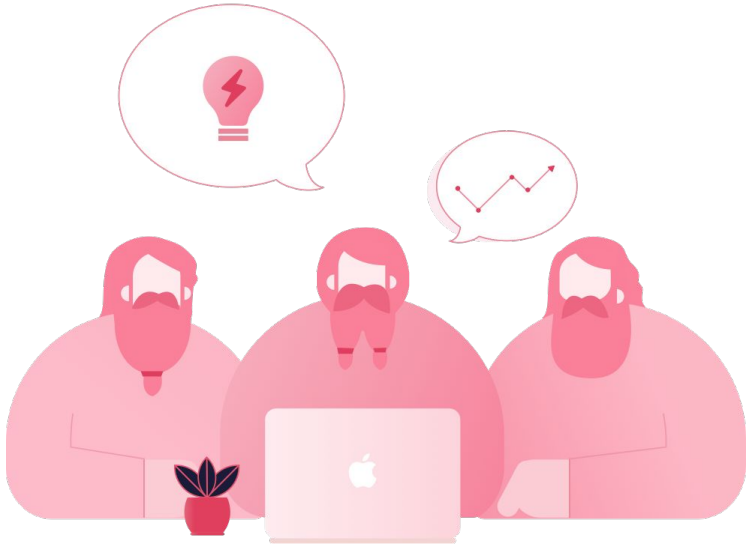
DWARVES FOUNDATION

Evolutionary Database Design

Evolving with the system

Presenter

Cuong Mai (Back-end Engineer)

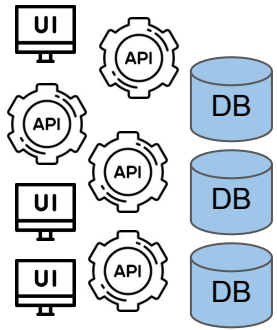


Evolutionary Database Design

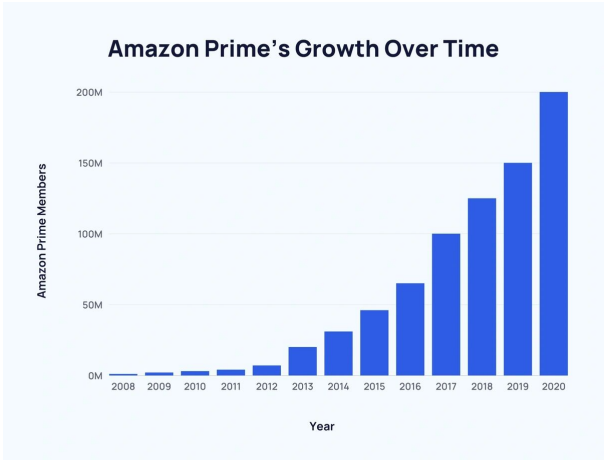
Evolutionary Database Design

A database and the system are like two intertwined vines — as the system scales to meet growing demands, the database must evolve alongside it to maintain performance and integrity

2015



2023



DWARVES FOUNDATION

Evolutionary Database Design

With exponential database growth, the management system no longer bring any context to the schema or data.

As result, practices that involve combination of humans, and systems are introduced to maintain and expand the knowledge of the database

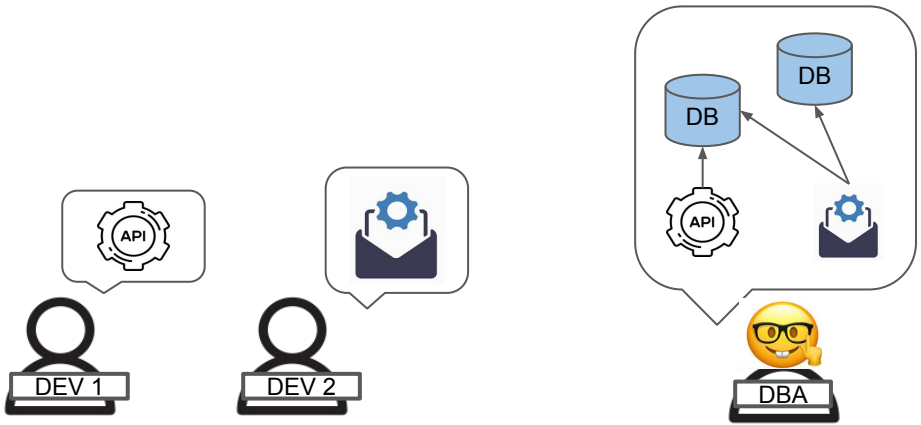
Disclaimer

This presentation is about how to manage the Database changes, knowledge and will not go into the selection of Database management system, or the design decision for database schema.

Table of contents

1. Knowledge sharing

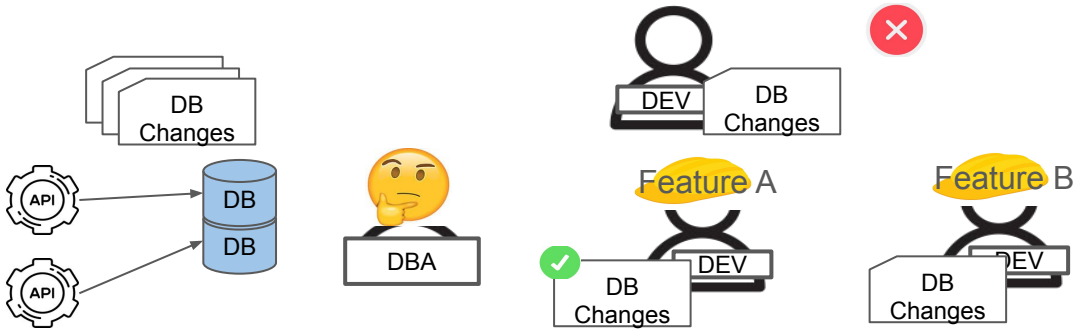
A person called DBA(Database Administrator) maintains and transfers the database context to be used effectively for the system development.



DWARVES FOUNDATION

1. Knowledge sharing

When saw/asked for changes to the databases. The DBA consults the changes with regard to the upstream/downstream services and provide alternative solutions if needed

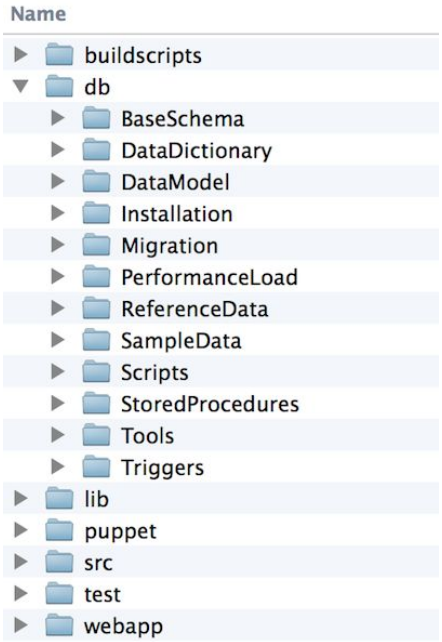


1. Knowledge sharing

After a change is approved. The DBA need to document the change, create migrations for it and store somewhere that is easy to access, and lookup like a centralized repository

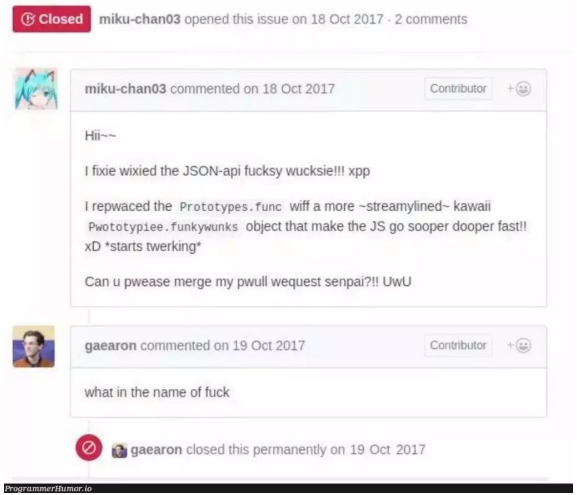
2. Repository structure

A Database Repository stores DB artifacts(run scripts, credentials, configuration), migrations, data, and documentation.



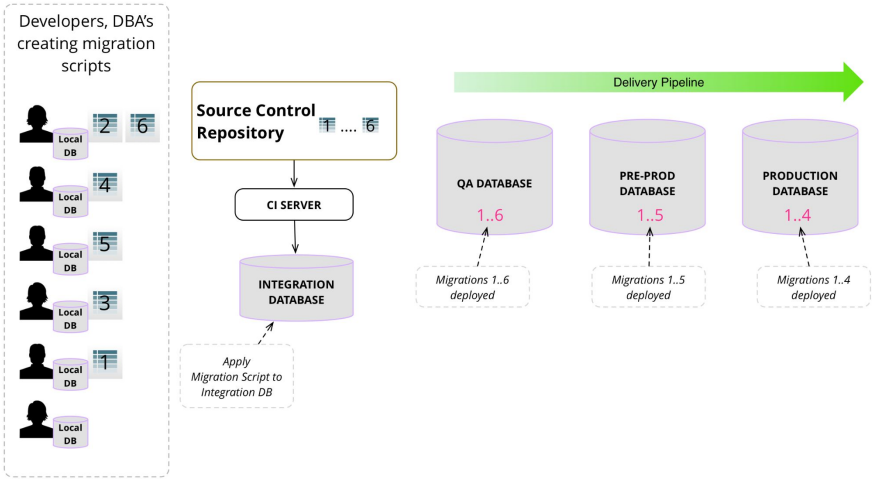
2. Repository structure

A Version Control System is used to manage changes, context and history of the DB Repository, where changemakers create PR with explanation of the DB change and provide relevant context to the reviewers.



2. Repository structure

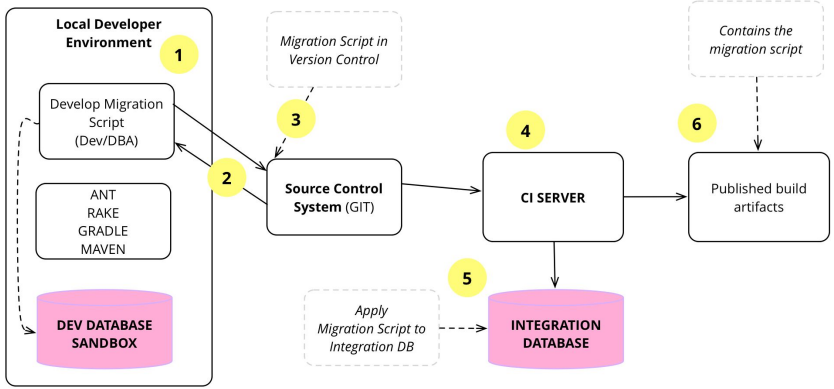
DB Repository changes have versioning for roll back and able to deploy specific version for DEV, QA, Prod environments



DWARVES FOUNDATION

3. Continuously integrate changes

When a DB instance checked out from the Integration DB to use for development, It's required to use Continuous Integration to verify changes to the Integration DB, and to be notified about schema/data conflict.



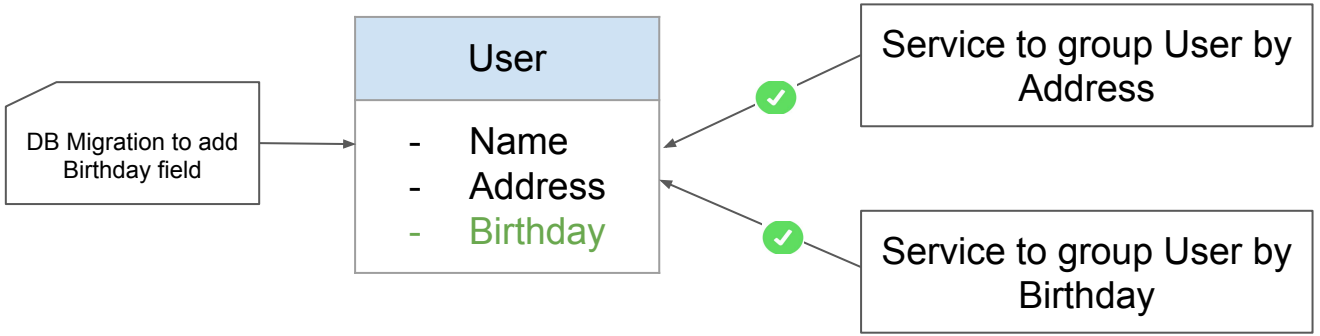
4. Refactoring

Every DB changes involve three steps:

- Changing the database schema
- Migrating the data in database
- Changing the database access code

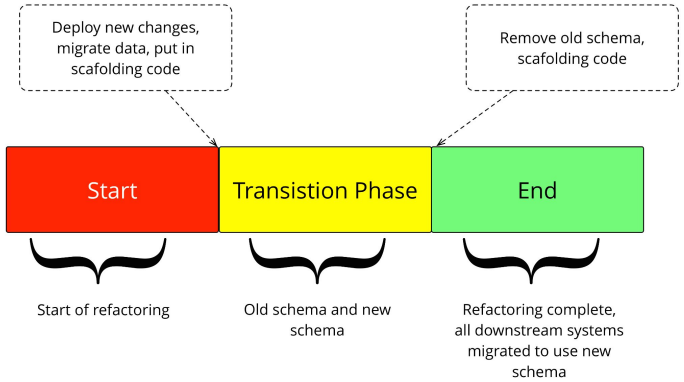
4. Refactoring

Minor DB changes like introducing new columns, tables can be done without changing the DB access layer of integrated services where service can ignore the data if not needed.



4. Refactoring

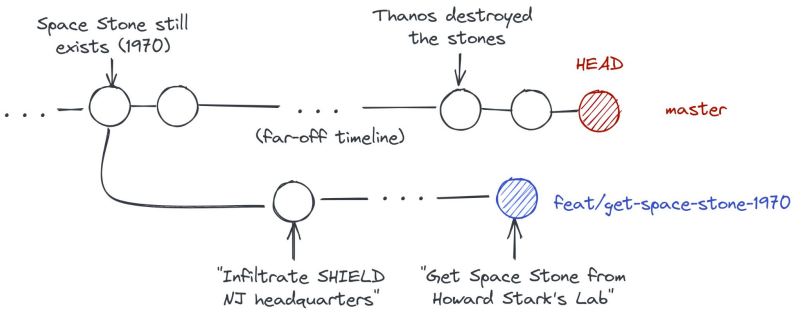
Destructive changes like introducing non-null columns, split tables requires services to change the database access layer to avoid errors, changing application code or introducing transitional phases until all changes are completed.



DWARVES FOUNDATION

4. Refactoring

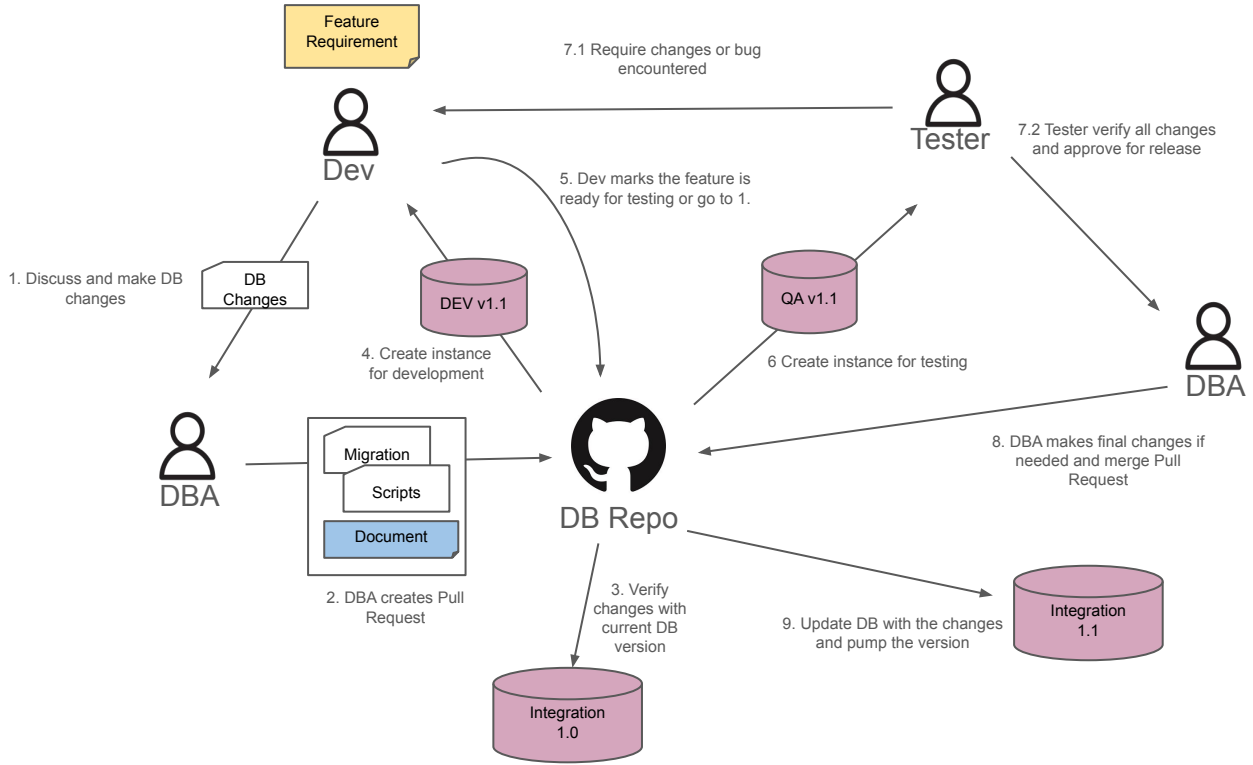
After refactoring, It is idea to release changes to the Integration Database frequently and notify others for breaking changes to minimize conflict resolution efforts.



Recap

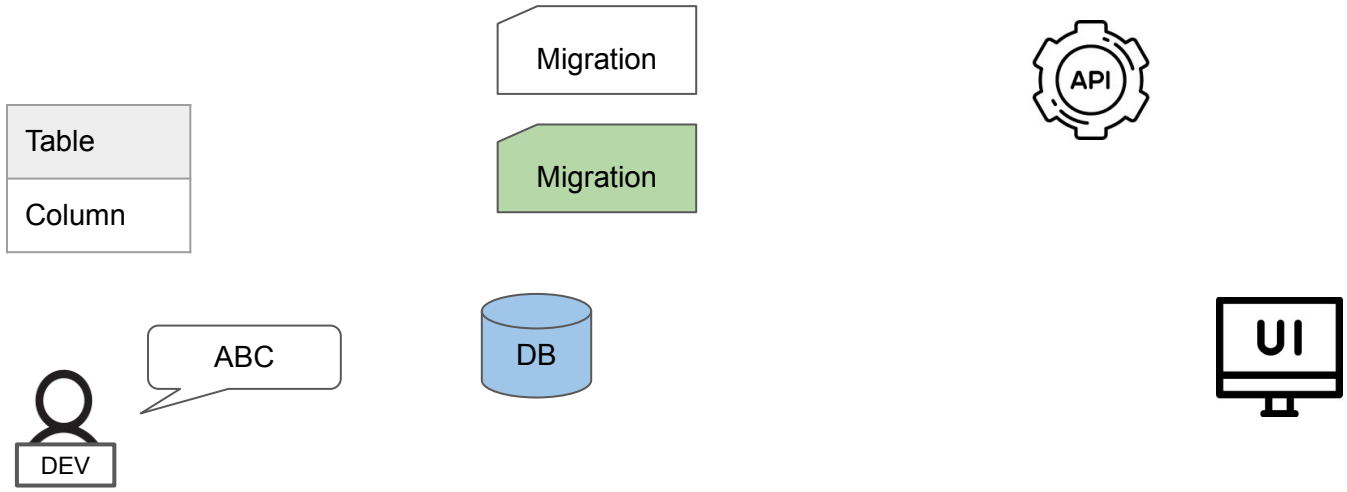
In modern software development, databases must evolve alongside applications, making it essential to maintain a clear understanding of them through knowledge sharing, effective structuring, refactoring, and frequent releases.

Example


Feature Requirement
7.1 Require changes or bug encountered
Tester
7.2 Tester verify all changes and approve for release
Dev
5. Dev marks the feature is ready for testing or go to 1.
1. Discuss and make DB changes
DB Changes
DEV v1.1
QA v1.1
4. Create instance for development
6 Create instance for testing
DBA
DBA
Migration
Scripts
Document
DB Repo
8. DBA makes final changes if needed and merge Pull Request
2. DBA creates Pull Request
3. Verify changes with current DB version
9. Update DB with the changes and pump the version
Integration 1.1
Integration 1.0

Reference

- https://martinfowler.com/articles/evodb.html#DbasCollaborateCloselyWithDevelopers


Migration
Migration
API
Table
Column
ABC
DB
UI
DEV