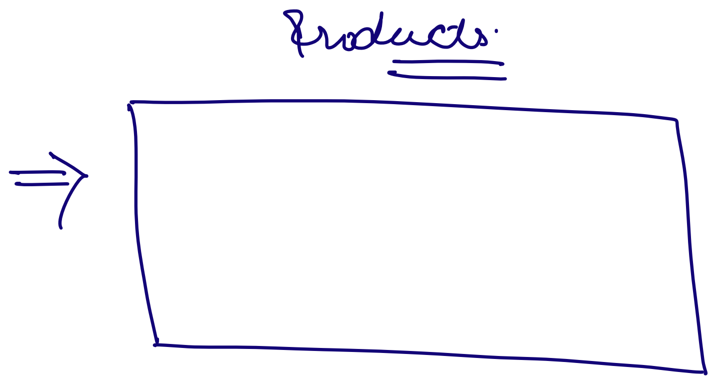
⇒ Schema **Versioning** & **Migration**.

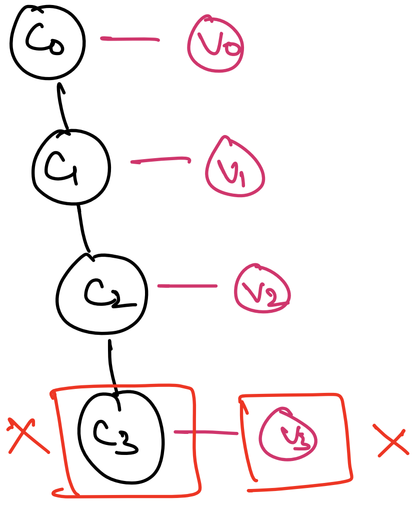
```
@Entity
Class Product {
    id
    title
    description
    category
    ___
    ___
    ___
    ___
}
```

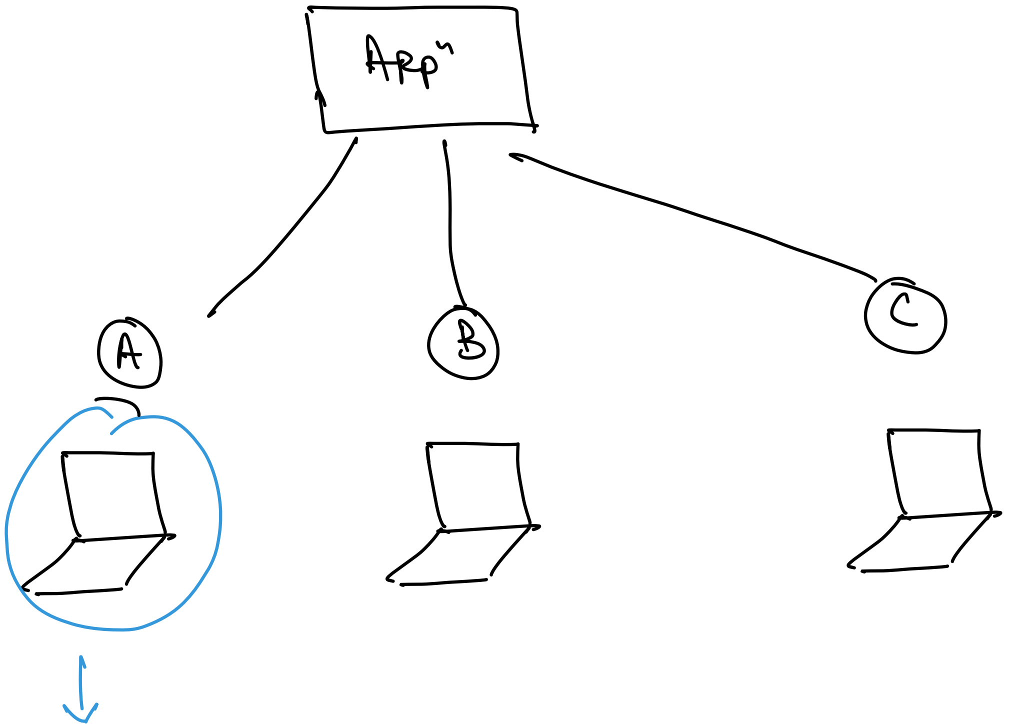


⇒ How the tables are getting are getting created by ORM, How ORM is writing queries for different usecases, We are not maintaining these queries as of now.

1. What if, we want to have some control over how the tables are getting created internally?

2. Just like we are maintaining the code history, we should also maintain our DB version history.





Did some code & DB changes.

⇒ Along with every code push we should also push the schema change versions (if any)

⇒ Schema Migration | Versioning libraries

- Flyway ⇒ More popular.
- Liquibase

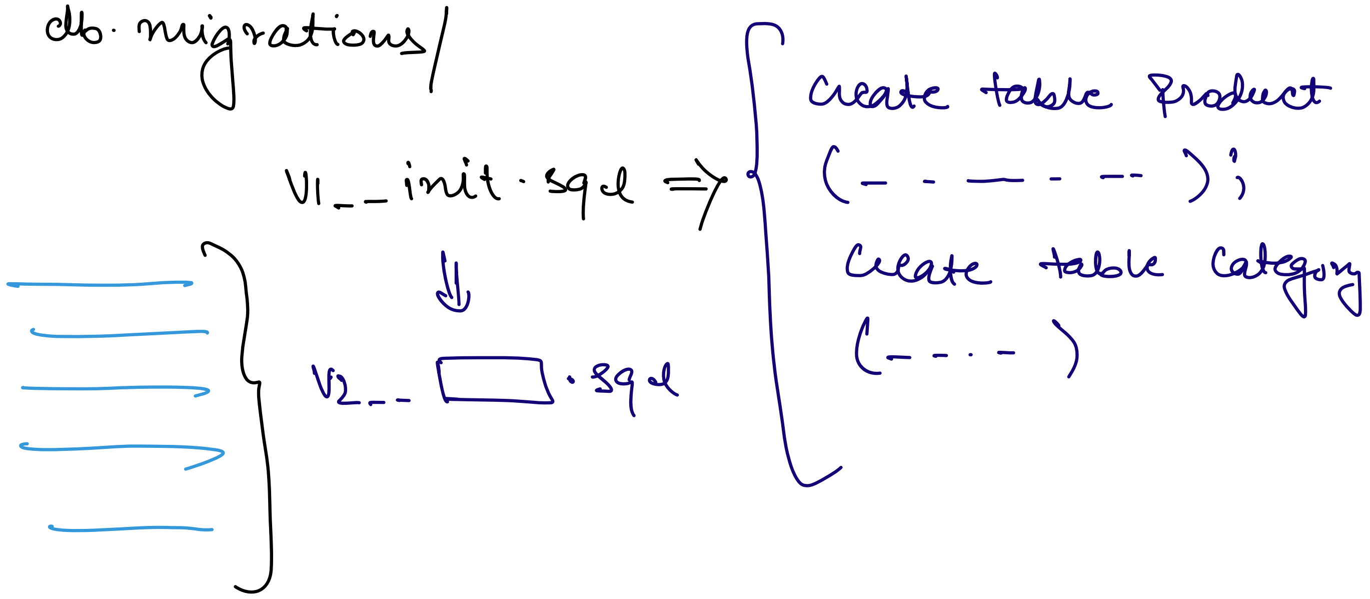
⇒ In our Project, we are going to maintain db.migrations folder.

```
db.migrations /
                V1__init.sql
                V2__addu_user_table.sql
                V3__□.sql
                  ⋮
```

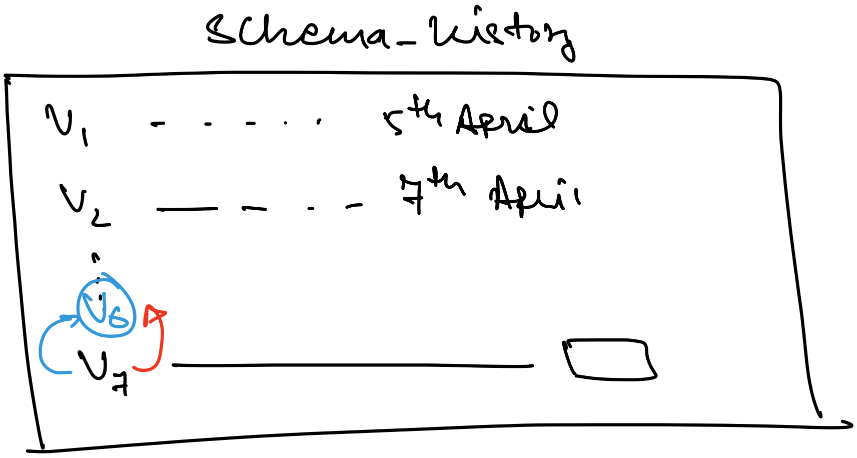
② Whenever we are doing any code change that requires DB changes

i) Create a new DB version SQL file with the SQL queries for these changes.

ii) We are going to maintain all such version files.



⇒ Migration library (flyway) creates an additional table in the DB to track what all the DB versions have been applied now.



schema_history

V1	- - - - . .	5th April
V2	___ _ . . -	7th April
⋮		
V6		
V7	________________	□