Factory

- Simple
- Factory Method
- Abstract Factory.

```
UserService {

    Database db = ____________;

    createUser() {
        if ( db instanceof mysql)
            Query q = new SQLQuery();
        else if ( db instanceof mongodb)
            Query q = new NoSQLQuery();
    }
```

mysql
mongodb
pgsql

Query
- SQLQuery
- NoSQLQuery

SRP , OCP

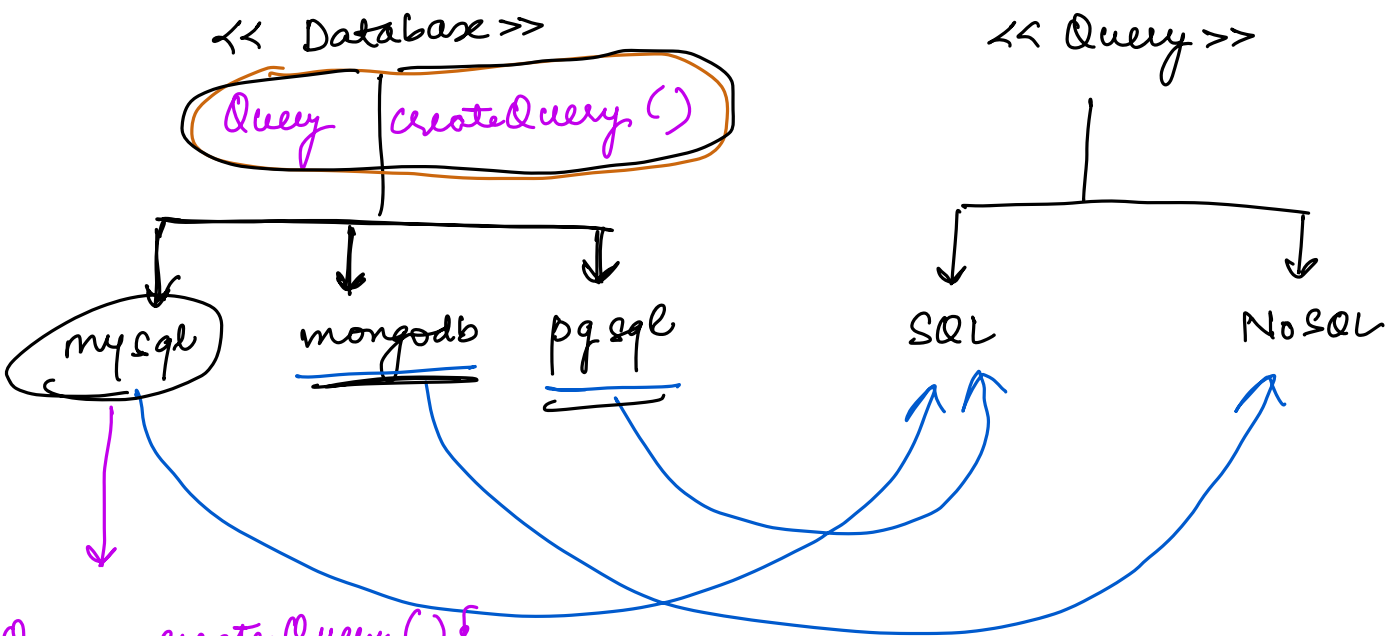
```
QueryCreator {
    static
    Query createQueryByDBType ( Type) {
        if ..
        else ..
    }
```

Factory

createUser() {

_____ = QueryCreator.createQueryByDBtype(db.getType());

}



Query createQuery() {

return new SQLQuery();

}

Database db = _____;

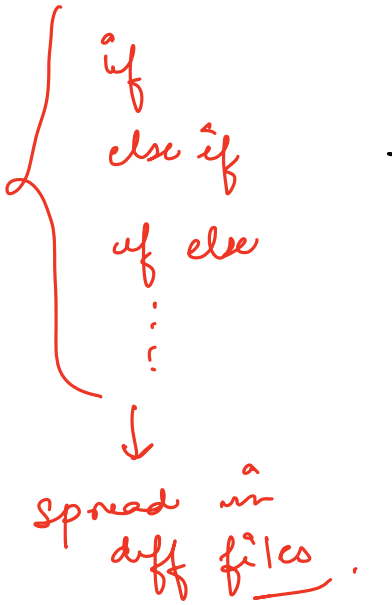
createUser() {

Query q = db.createQuery();

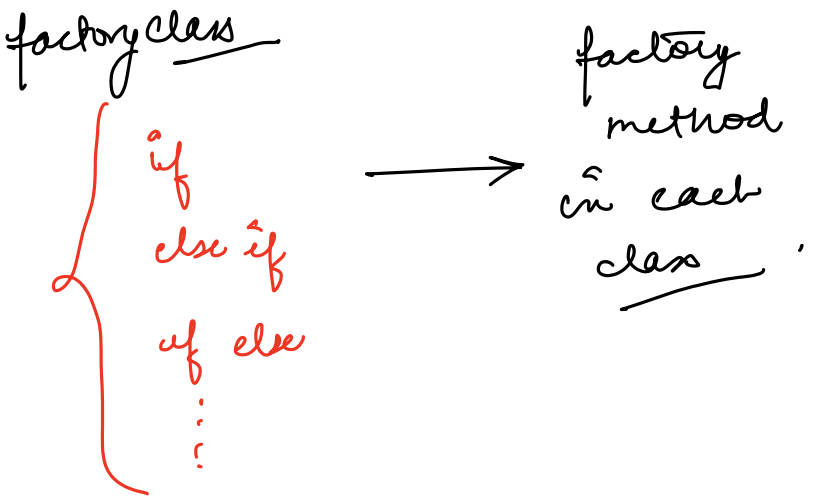
factory method

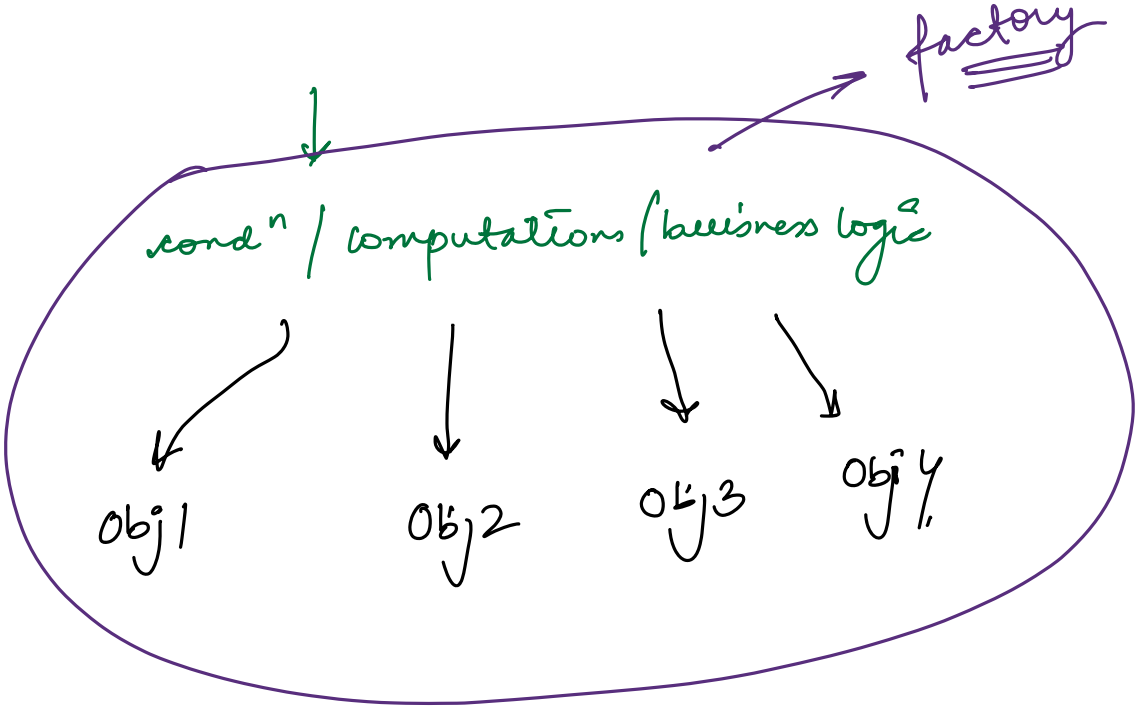
}

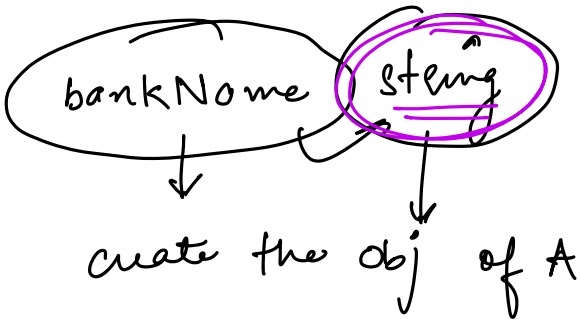
V0

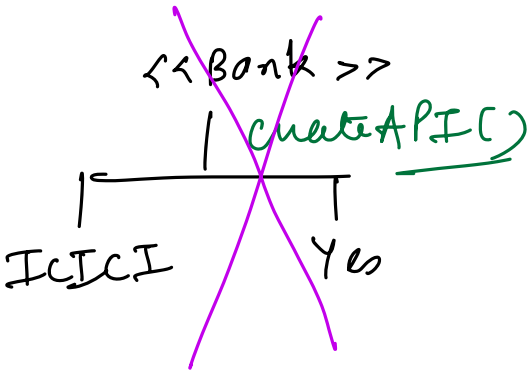

if
else if
if else
spread in diff files.

V1

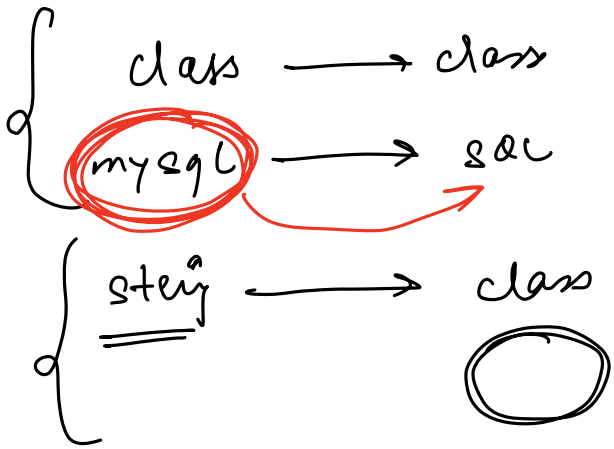

factoryclass
if
else if
if else
factory method in each class.


factory
cond^n / computations / business logic
Obj1
Obj2
Obj3
Obj4

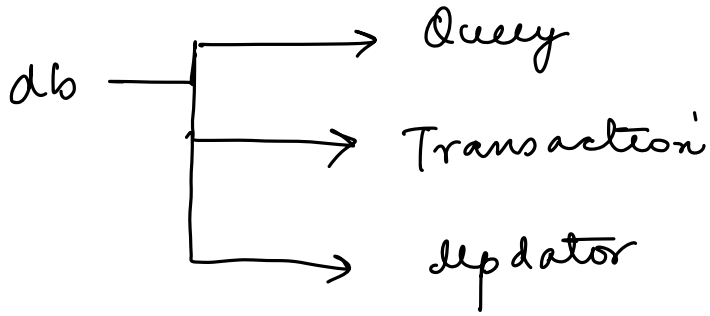


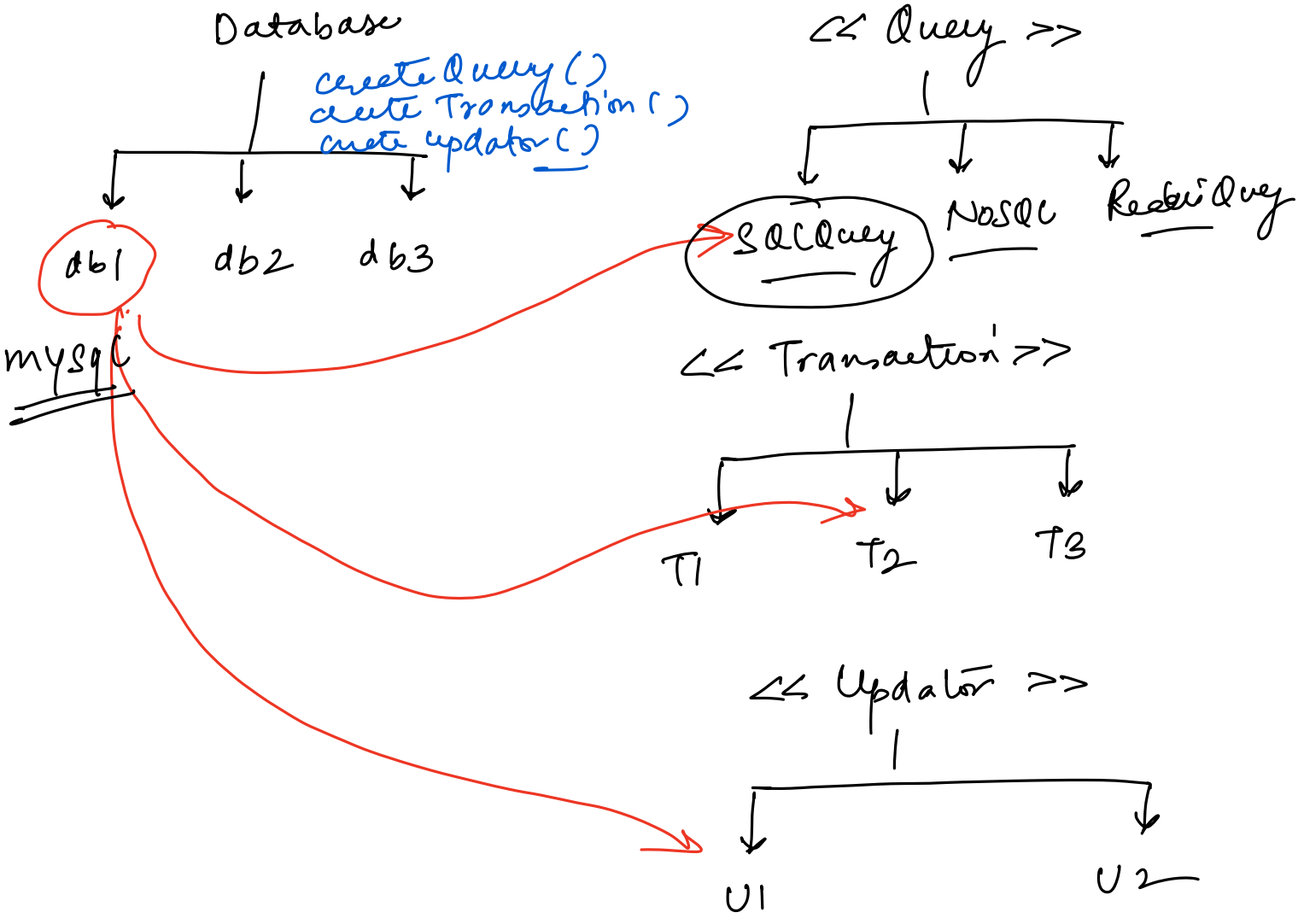


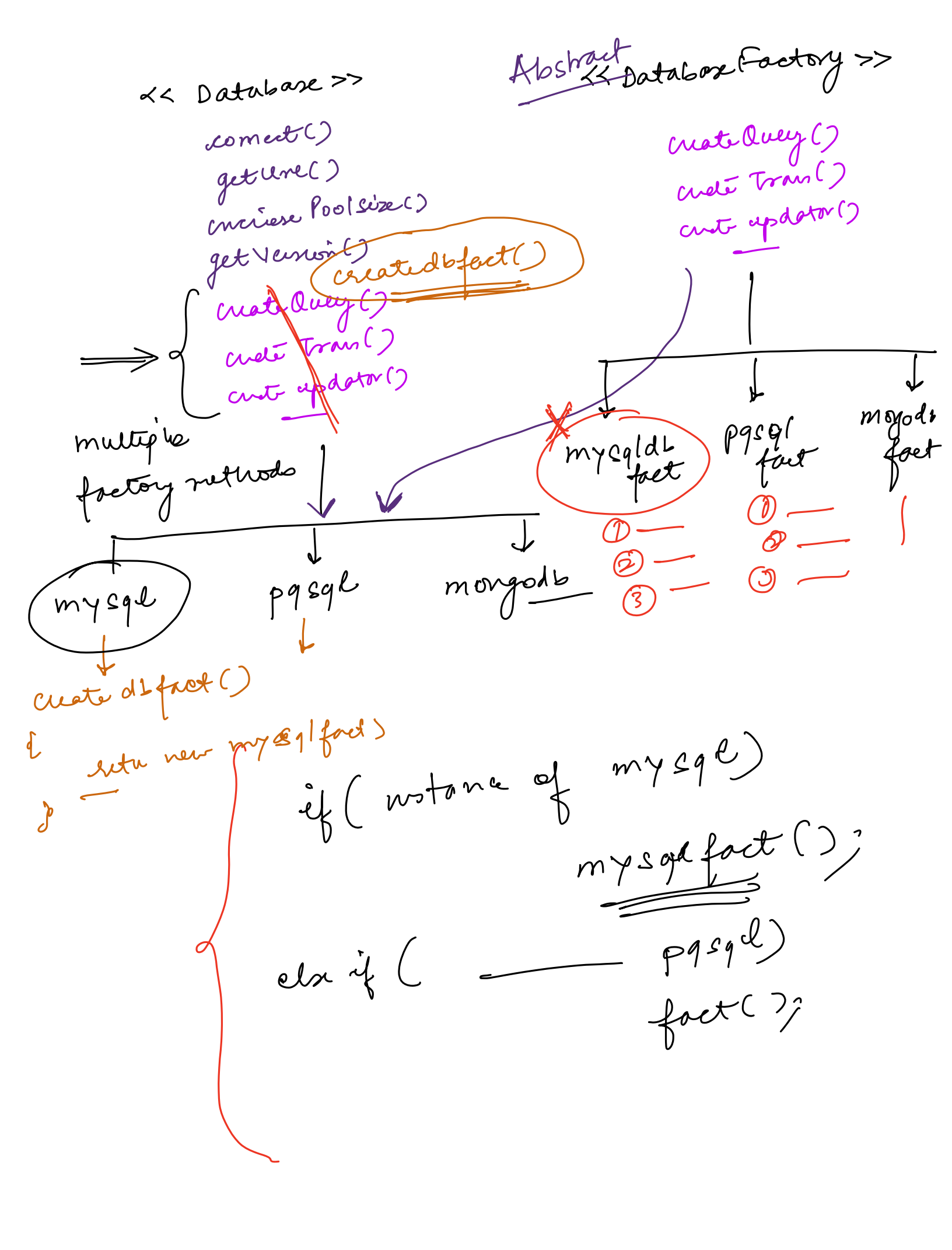

```
if ( bankName.equals(ICICI))
    ICICIBankAPI();
```



: simple factory


db
Query
Transaction
Updator


Database
createQuery()
createTransaction()
createUpdator()
db1
db2
db3
mysql
<< Query >>
SQLQuery
NoSQL
RedisQuery
<< Transaction >>
T1
T2
T3
<< Updator >>
U1
U2


<< Database >>
connect()
getUrl()
increasePoolSize()
getVersion()
createDbFact()
createQuery()
createTran()
createUpdator()
multiple factory methods
mysql
pgsql
mongodb
createDbFact()
{
return new mysqlFact()
}
Abstract << DatabaseFactory >>
createQuery()
createTran()
createUpdator()
mysqlDbFact
pgsql fact
mongodb fact
1
2
3
1
2
3
if(instance of mysql)
mysqlFact();
else if(——— pgsql)
Fact();

When you have a lot of factory methods

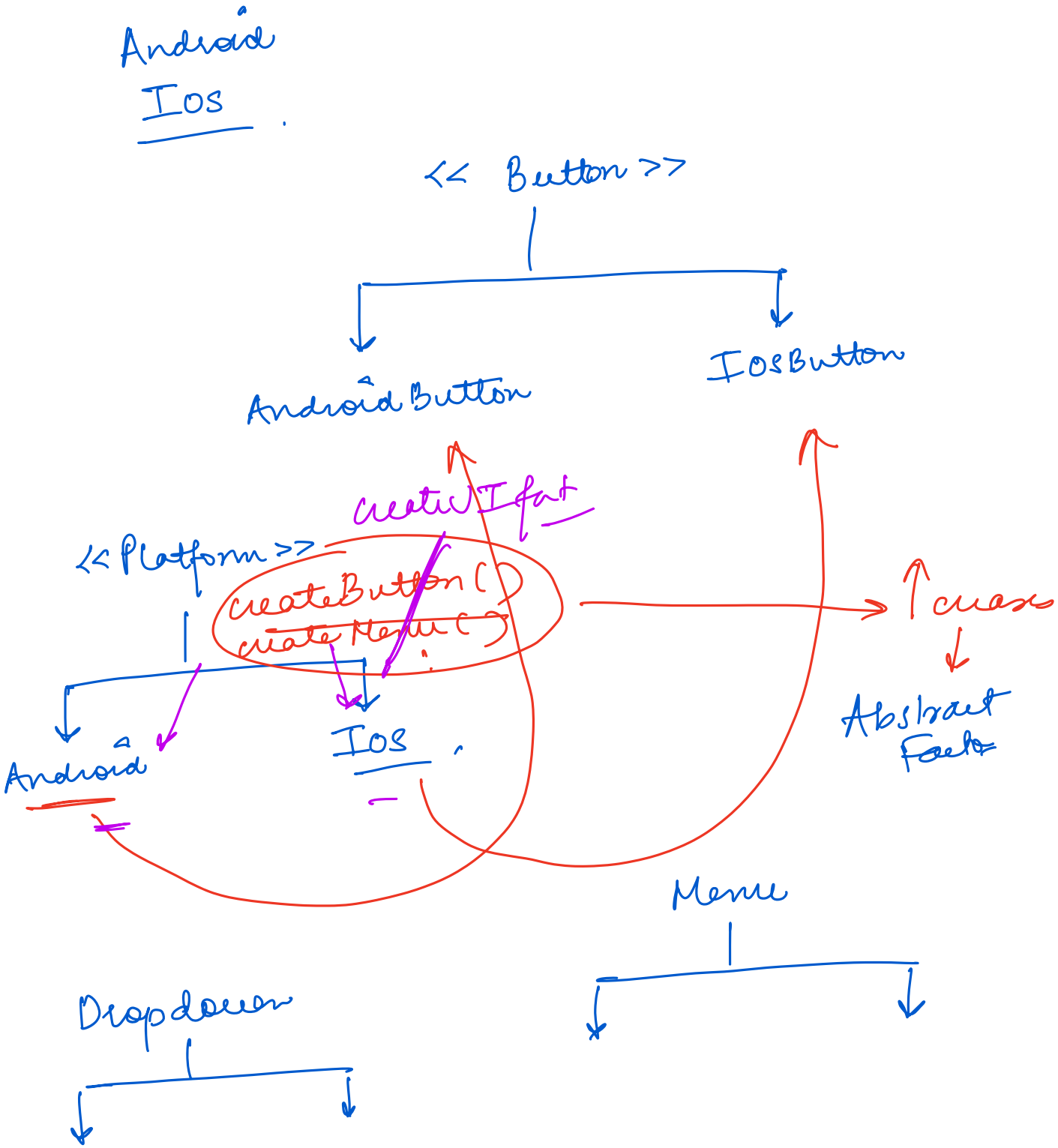
↓

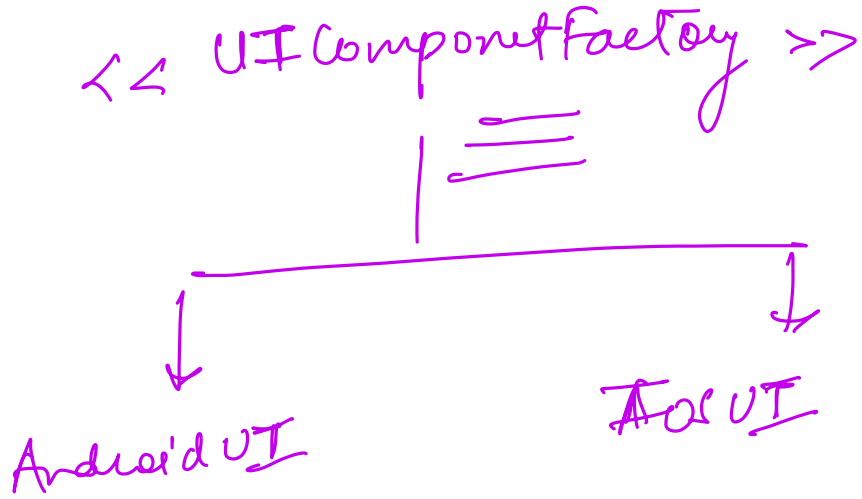
Try to segregate & put it in separate factory structure.

10:33 – Break.

Flutter / React Native :

cross platform UI frameworks.




<< UIComponentFactory >>
Android UI
iOS UI