

Factory
↓

Simple factory

{ factory method
Abstract Factory }

UserService {

Abstract / Interface
class

concrete class

Database db = new mysql() ;

createUser() {

if (db is instance of mysql)

Query x = new SQLQuery();

else if (db is instance of mongodb)
= new NoSQLQuery();

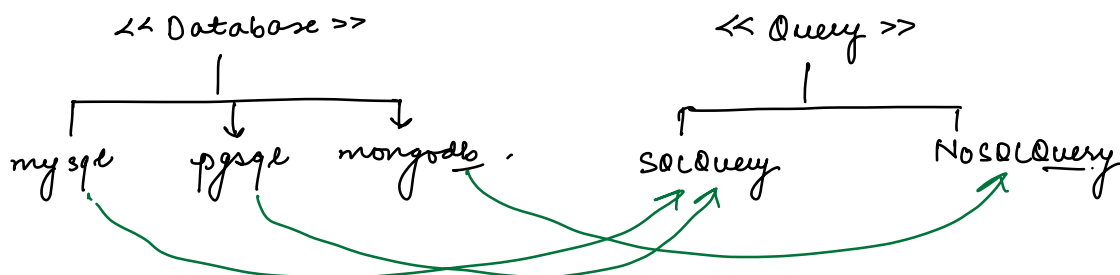
⋮

}

}

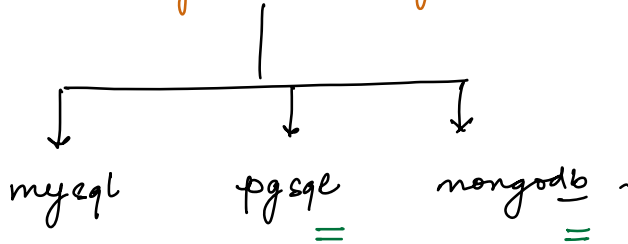
↓ OCP X

SRP X



Database

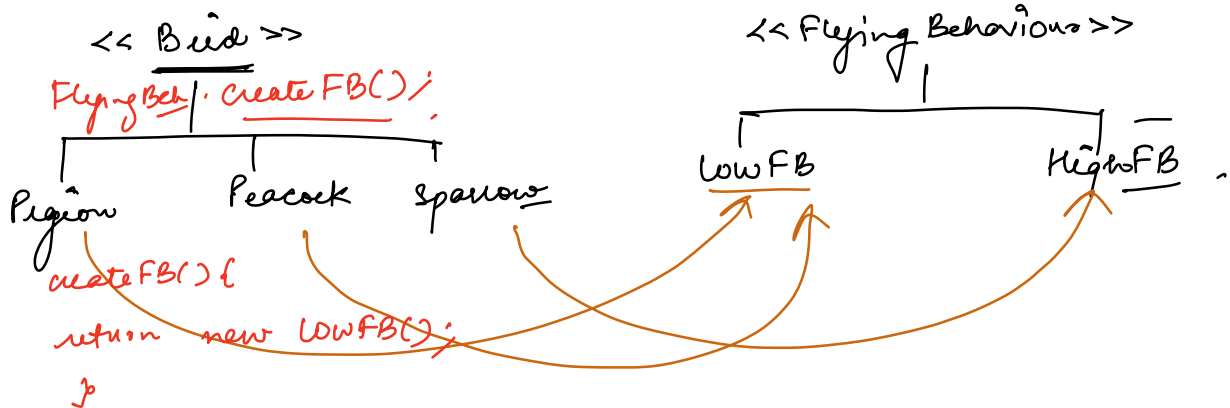
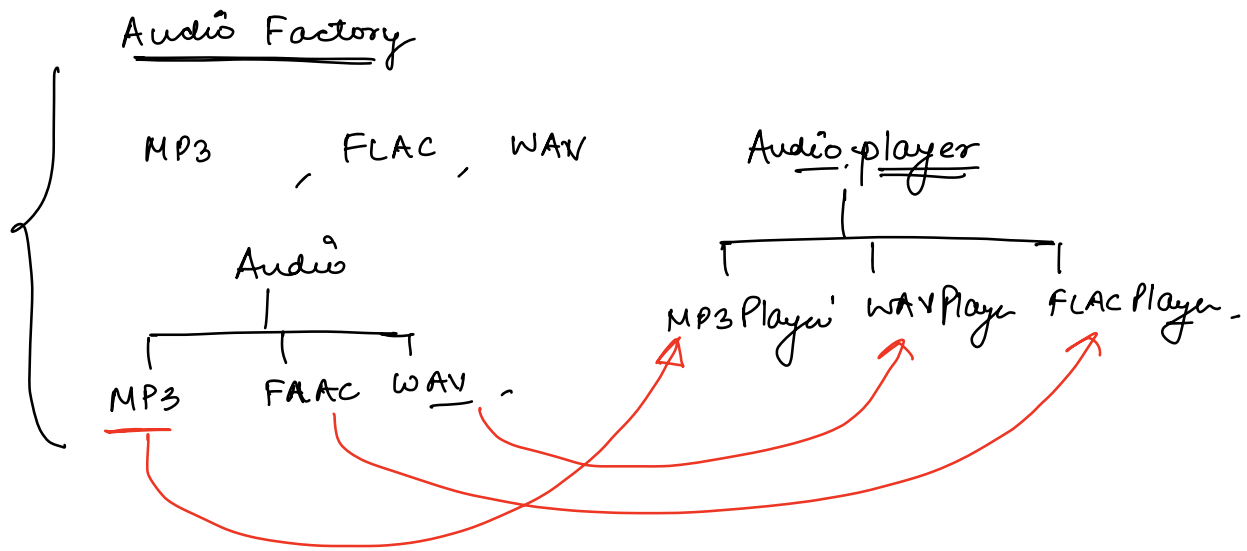
Query createQuery();



```
Query createQuery(){
    return new SQLQuery();
}
```

Database db = _____;

Query `x = db.createQuery();`
 ↑
 ↓
factory method ^a polymorphism



ENUM

FBType

LOW

HIGH

Bird(b) = _____;

if (b is instance of Pigeon)

 new LowFB();

else if (b ...)

 !

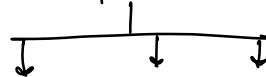
 !

Based on db $\Rightarrow \ll \underline{\text{Query}} \gg$

$\ll \text{Transaction} \gg$



$\ll \text{Updater} \gg$



$\ll \text{Database} \gg$

~~createQuery()~~
~~createTransaction()~~
~~createUpdater()~~

Ⓘ

connect()
setUrl()
increasePoolSize()
getVersion()

Ⓜ

DBFactory createDbFactory()

mysql pgsql mongodb

createDBFactory()

new mysqlfact();
}

if (db == mysql)

new mysqlfact();

⋮

$\ll \text{DatabaseFactory} \gg$

createQuery()
createTransaction()
createUpdater()

mysqlfact pgsqlfact mongofact

==

==

==

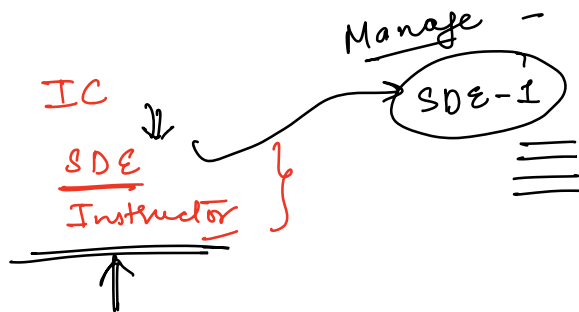
Database db = _____;

DbFactory dbf = db.createDbFact();

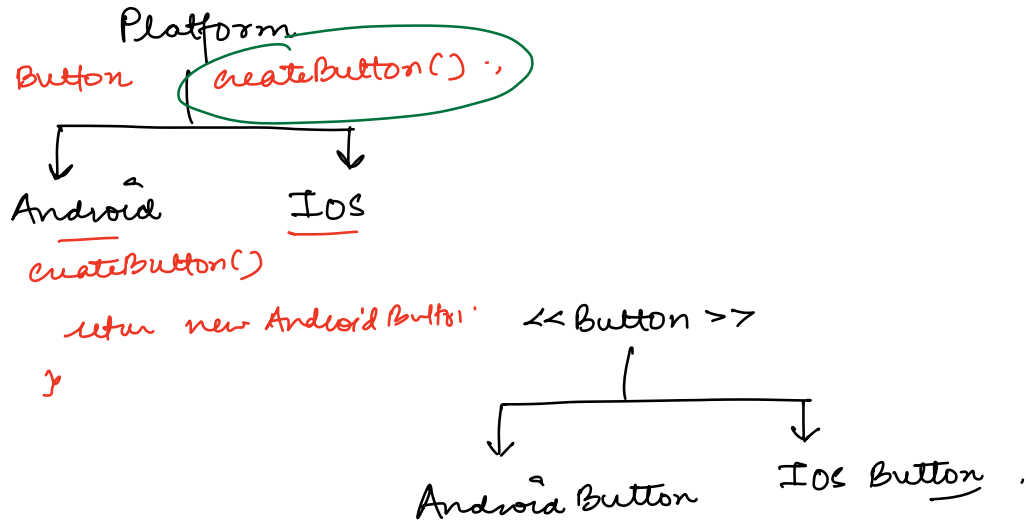
dbf.createQuery();

dbf.createTrans();

dbf.createUpdate();



Flutter / React Native : cross platform UI frameworks.



```
createbutton() {
    if ( platform == IOS )
        new IOSButton();
    else if ( platform == Android )
        new Android Button();
}
```

