



Today's agenda

↳ factory design pattern

↳ factory method design pattern

↳ Abstract factory design pattern.



AlgoPrep



SO

<< Database >>



↳ To generate query based on type of database:

Database db = input();

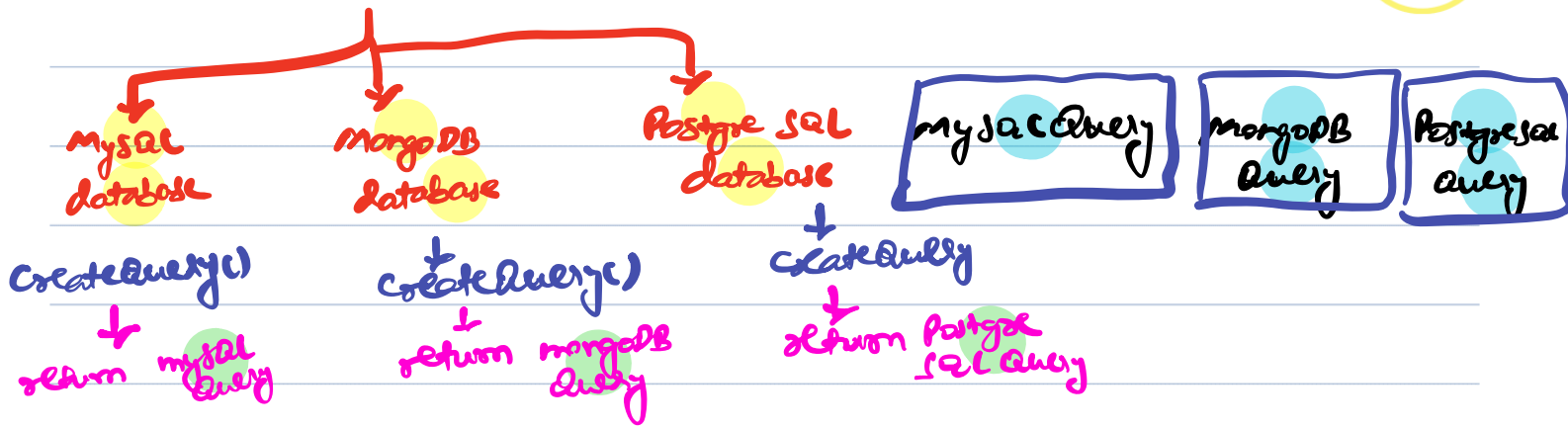
SRP xx
OCP xx

```
if (db instanceof mysql) {  
    createQuery for mysql  
}  
else if (db instanceof mongoDB) {  
    createQuery of mongoDB  
}  
:  
:
```

S1



<< Database >>



mongodb database

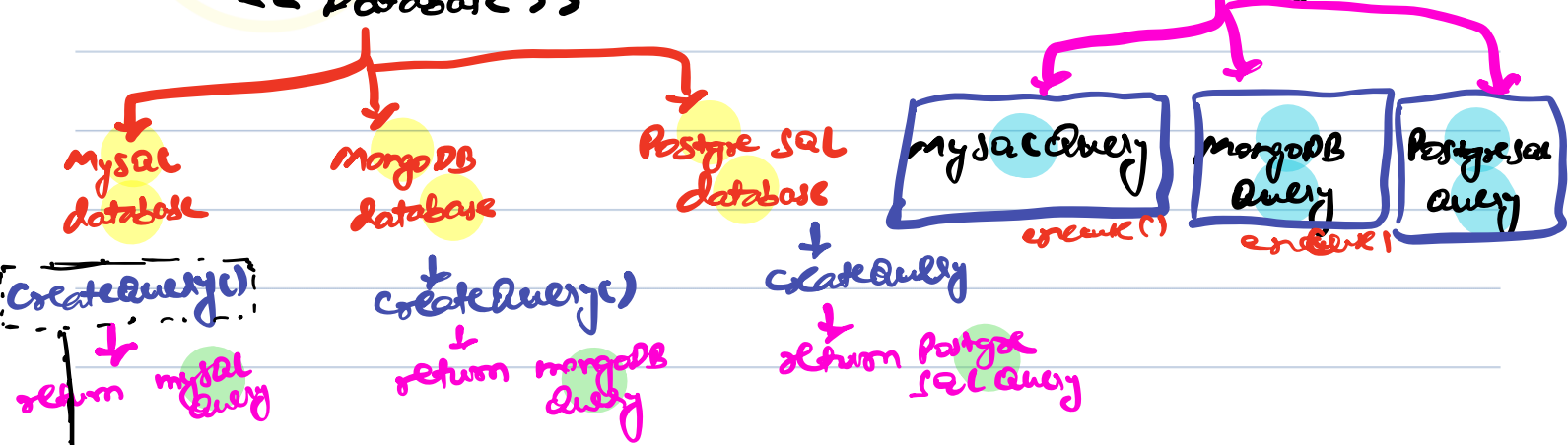
Database db = input();

?? ?? = db.createQuery();

S3

<< Database >>

<< Query >>



mongodb database

Database db = input();

Query q = db.createQuery();

q.execute();



→ Factory method: A method in an interface/abstract class whose purpose is to return the object of a certain class.

* what if database interface has many factory method?

<< Database >>

createUser();

createConnection();

generateurl();

seturl();

setpassword();

factory method

common method

responsibility of database interface

↳ Has methods needed to perform db operation.

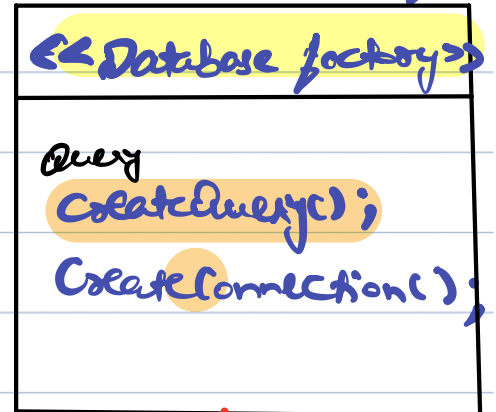
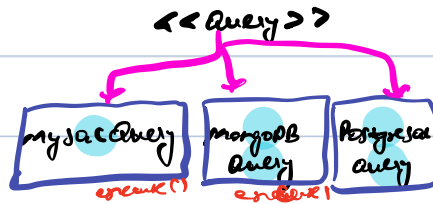
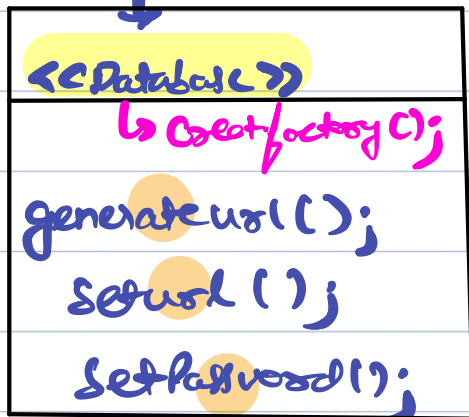
↳ create objects of diff class.

↓
SRP is violated



Segregate the responsibility

abstract factory



Database db = input();

Query q = db.createQuery();

Database db = input();

DatabaseFactory dbf;

if (db instanceof MySQL) {
dbf = new MySQLFactory();
}

else if (db instanceof MongoDB) {
dbf = new MongoDBFactory();
}

...

Query q = dbf.createQuery();

final soln

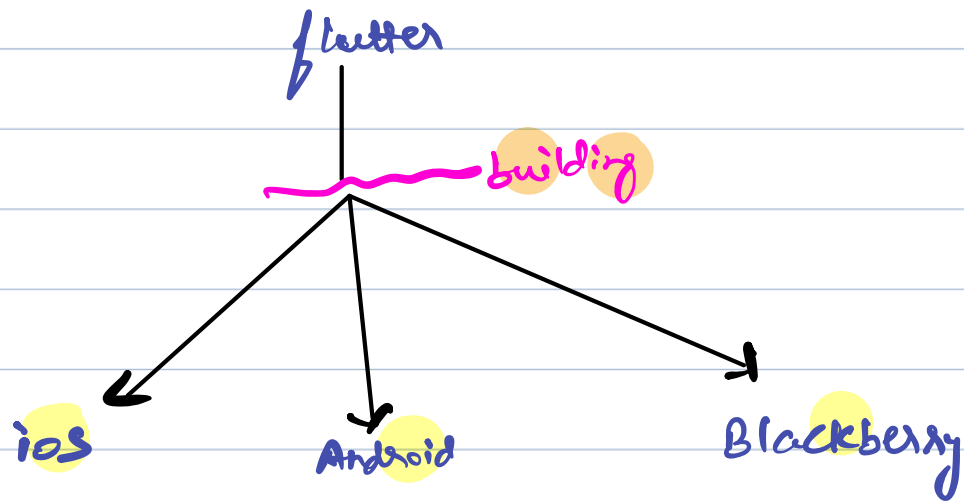
Database db = input();

DatabaseFactory dbf = db.createFactory();

Query q = dbf.createQuery();

Create one factory method in database which returns the database factory instance.

Break till 10:20 PM



AlgoPrep