ABSTRACT FACTORY PATTERN

CREATING FAMILIES OF CLASSES

CREATING FAMILIES OF CLASSES

LET'S GO BACK TO OUR EXAMPLE WHERE WE ORIGINALLY USED A DATABASE PROVIDED BY MICROSOFT AND HAD TO MOVE TO AN ORACLE DATABASE AT THE BEHEST OF THE NEW CIO

SUPPOSE WE HAVE MANY PRODUCTION SYSTEMS WHICH DEPEND ON DIFFERENT FLAVORS OF THESE DATABASES

BASED ON THE KIND OF SUPPORT THAT A PRODUCTION SYSTEM NEEDS, IT COULD CHOOSE A PARTICULAR FLAVOR OF DATABASE

EVERY DATABASE SYSTEM BE IT MICROSOFT OR ORACLE WOULD HAVE DIFFERENT IMPLEMENTATIONS OF EACH OF THESE FLAVORS

DATABASE FLAVORS

LITE VERSIONS

THESE MIGHT OFFER LIMITED FUNCTIONALITY, LESS STORAGE, AND MIGHT BE USED FOR INTERNAL SYSTEMS

PROFESSIONAL EDITIONS

THESE WOULD BE FULL-FLEDGED, OFFERING THE MOST POWERFUL FEATURES FOR USER-FACING SYSTEMS

READ-ONLY

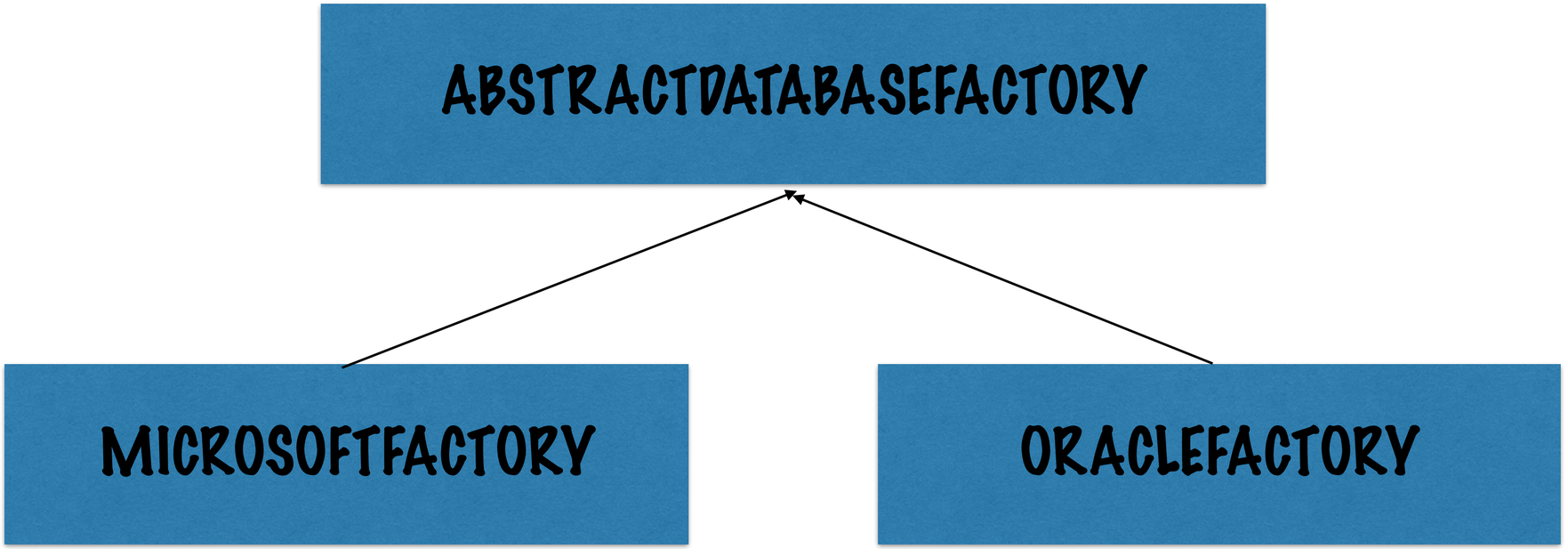
THESE MIGHT BE USED FOR BULK READ OPERATIONS AND OPTIMIZED TO BE REALLY FAST FOR READ OPERATIONS

EACH FLAVOR CAN BE CONSIDERED A FAMILY OF DATABASES IMPLEMENTED BY DIFFERENT COMPANIES

ABSTRACT FACTORY PATTERN

```
public abstract class DatabaseFactory {

    public abstract IDatabase getDatabaseLite();

    public abstract IDatabase getDatabasePro();

    public abstract IReadOnlyDatabase getReadOnlyDatabase();

}
```

BOTH MICROSOFT PRODUCTS AND ORACLE PRODUCTS SHOULD PROVIDE THESE FAMILIES OF DATABASES FOR THE TRANSITION TO BE SUCCESSFUL



WHEN SWITCHING FROM MICROSOFT TO ORACLE PRODUCTS SIMPLY CHANGE THE REAL FACTORY WE USE BASED ON WHAT IS SPECIFIED IN THE CONFIGURATION FILE

ALL OUR DATABASE FLAVORS SHOULD WORK AS IS POST-SWITCHOVER

THE ABSTRACT FACTORY PATTERN IS THUS USEFUL TO CREATE FAMILIES OF PRODUCT WHILE ABSTRACTING AWAY CREATION AND IMPLEMENTATION DETAILS FROM THE USER