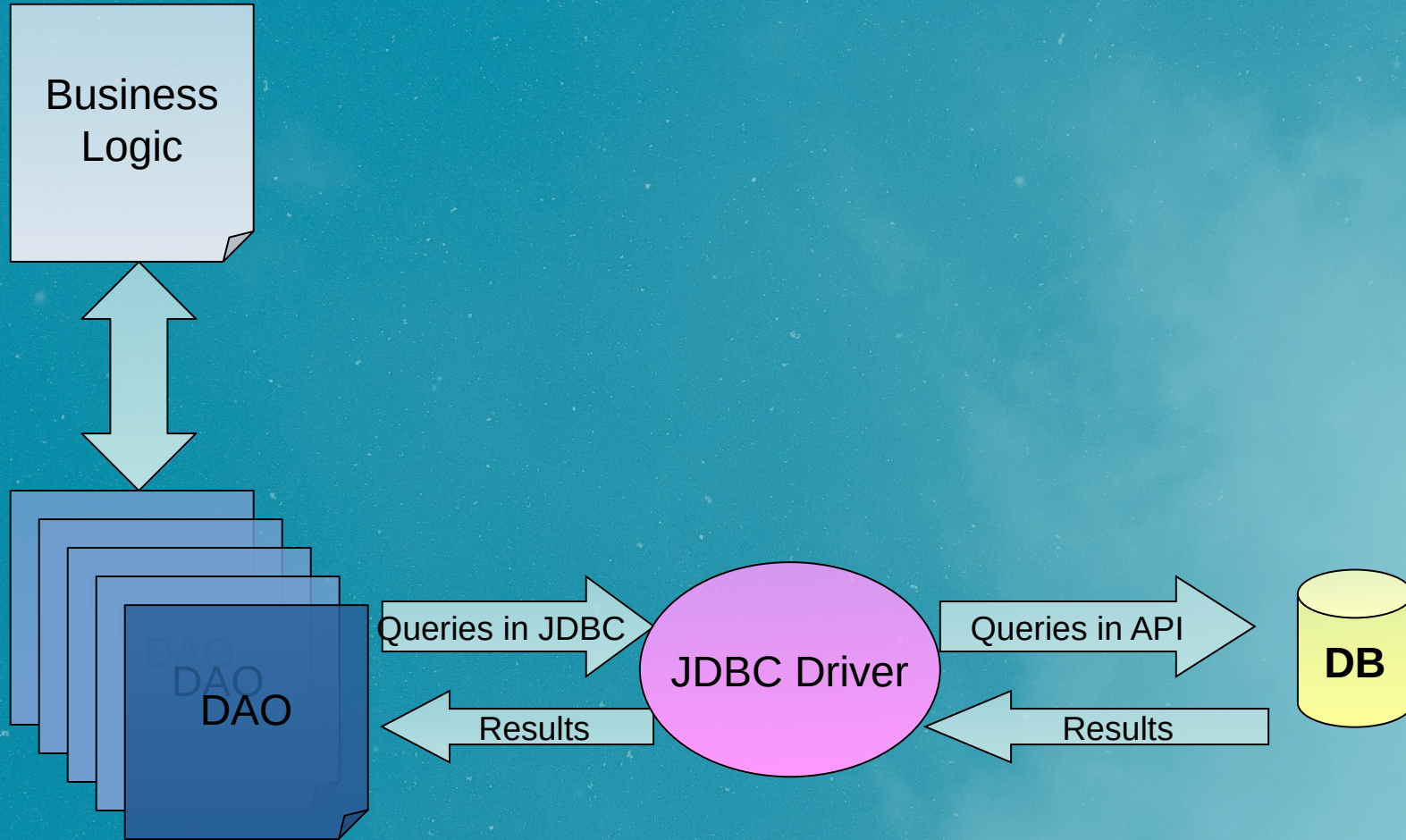


Data Access Object (DAO)

What?

- The Data Access Object (DAO) pattern is a structural pattern that allows us to isolate the application/business layer from the persistence layer (usually a relational database but could be any other persistence mechanism) using an abstract API.
- The API hides from the application all the complexity of performing CRUD operations in the underlying storage mechanism.



Context

- You have to implement a project that requires a relational database in the backend
- You have to use Data Access Object (DAO) design pattern for database programming
- Access to data varies depending on the source of the data.
- Access to persistent storage, such as to a database, varies greatly depending on the type of storage (relational databases, object-oriented databases, flat files, and so forth) and the vendor implementation

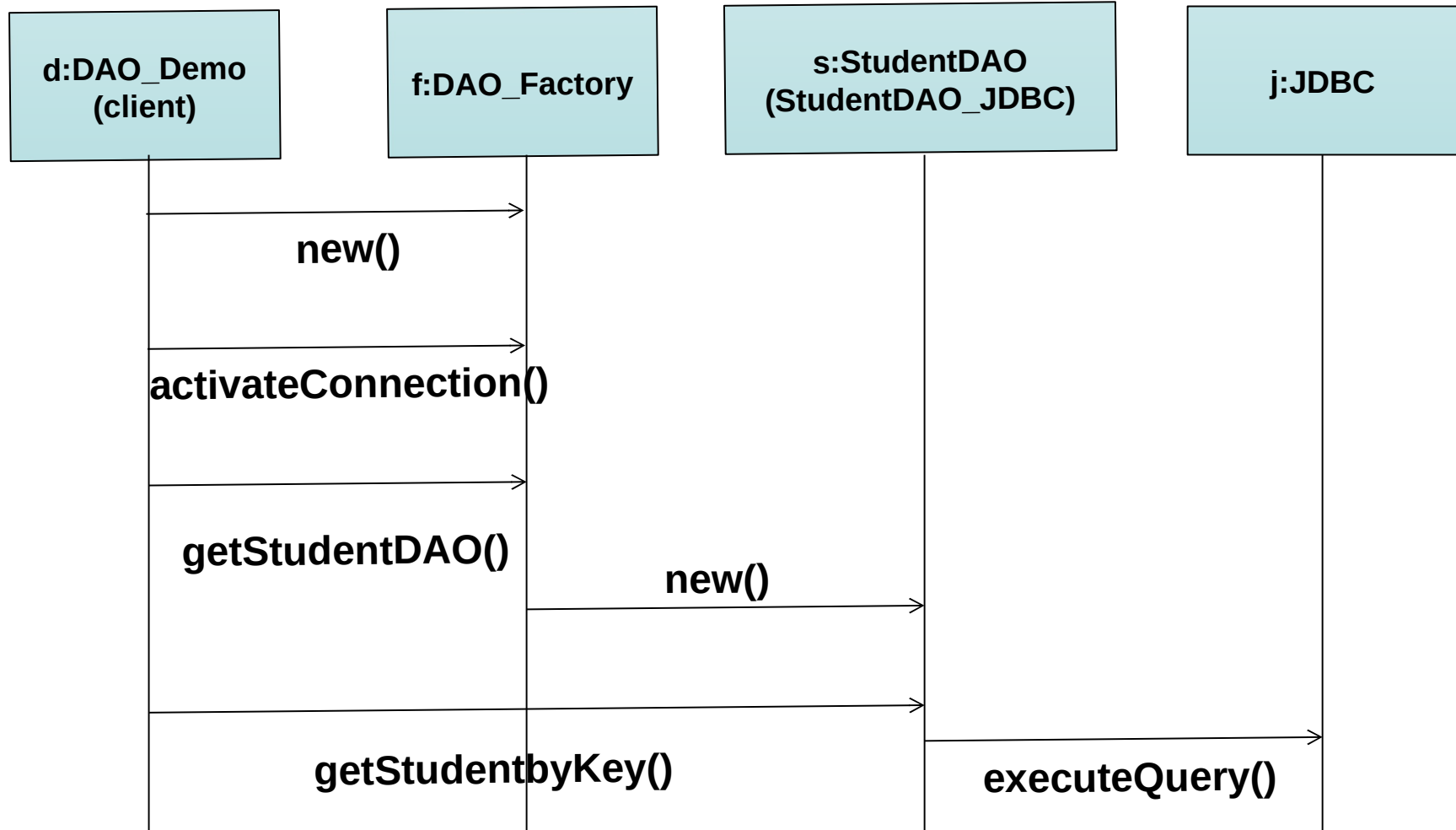
Summary

- Use DAO Factory to create DAO objects
- For each data class, create:
 - A bean class (“Student”)
 - A DAO interface (“StudentDAO”)
 - A DAO implementation (“StudentDAO_JDBC”)
- Transactions can be committed / rolled back only if they all SHARE the same connection

DAO Factory

- Isolate connection and DAO management to a DAO Factory class
- Two responsibilities for DAO Factory:
 - Keep track of when a new connection should be created and when it should be reused
 - Provide link to DAOs paired with appropriate connection object

Sequence Diagram





Steps

- Create a database with name daoproject
- Create a table student with fields rollno int type primary key and name varchar(20)
- Use the 5 java files in LMS
- Task : Write the code for update and delete
- Submit : Zip file containing all 5 files implementing the update and delete functionality.
- Deadline : 5 April 2023, 11:59 PM