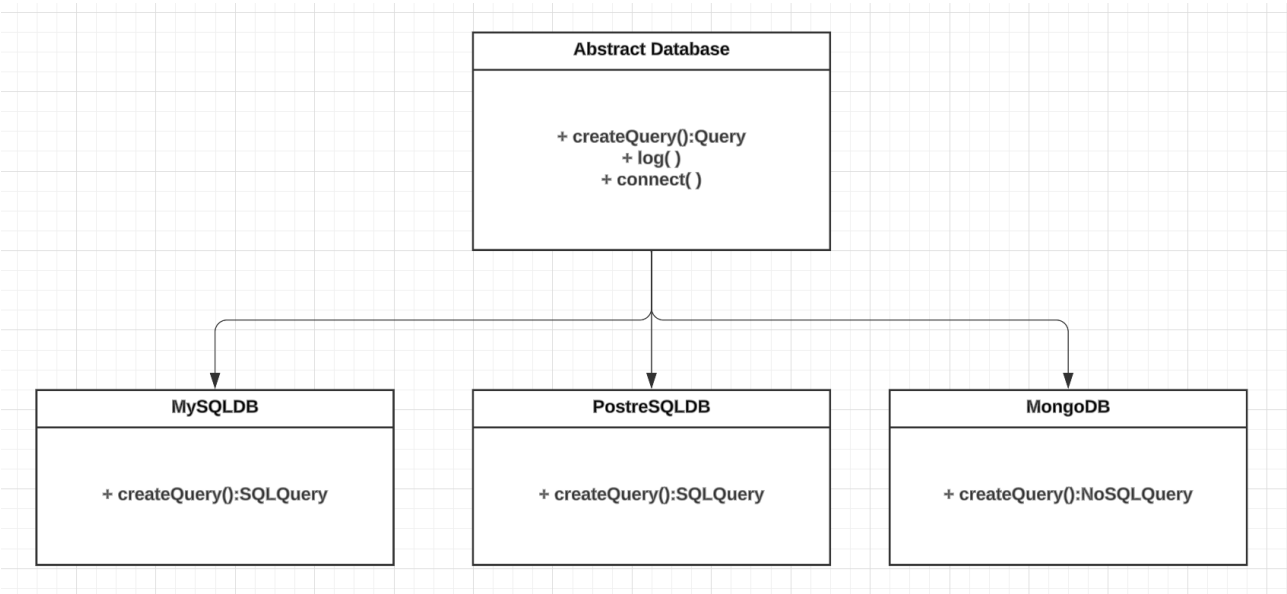
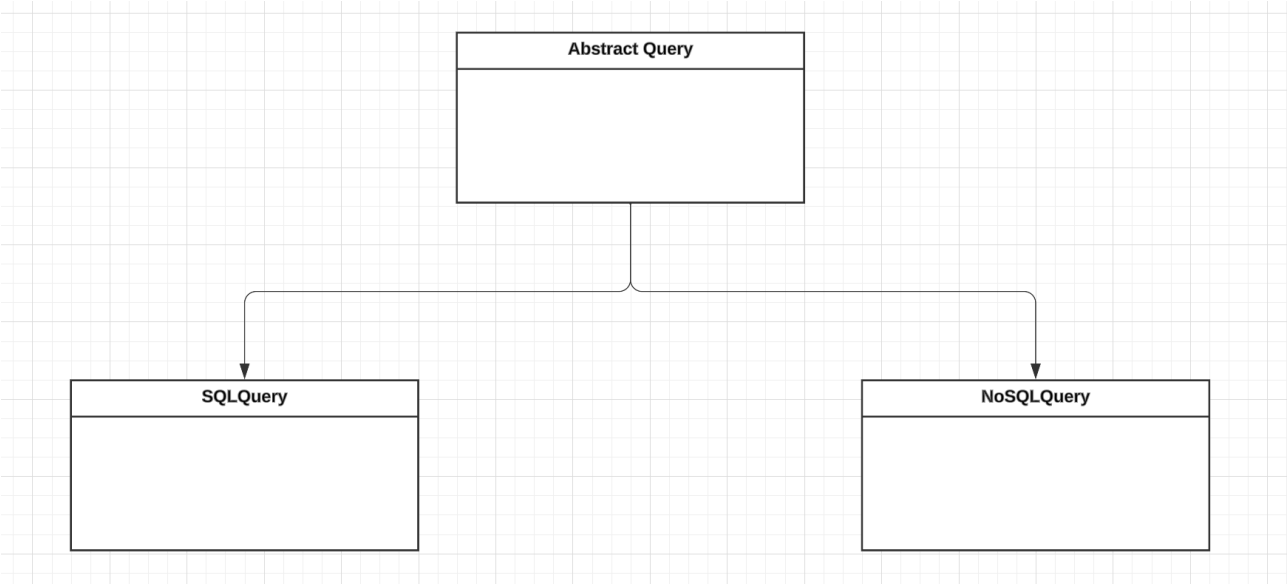
4. Factory Design Pattern

Factory Method

- We have a below use case with abstract Database class with 2 child classes.
- **MySQLDb** & **PostgreSQLDb** createQuery() method will return **SQLQuery** object wherein **MongoDb** createQuery() method returns **NoSQLQuery** object.



- The abstract createQuery method in the parent Database class return the generic Query object, but the child classes of Database returns corresponding Query objects based on the database type. Because of the Inheritance parent class reference can be used to return child class objects.



- The Query method in the parent class is a factory method. The purpose if the createQuery() is to create an object of corresponding Query object based on the type of the database.

Definition of Factory method

- Factory method is a method present in an interface or a parent class whose purpose is to create an object of the another/corresponding object.
- The benefits of using Factory method are,
    - A new Query type can be added into the codebase easily.
    - A new Database type can be added into the codebase easily.
    - Code is more extensible as new features can be added quickly.
- For this pattern to work, the base Database class must work with abstract Query: a base class or an interface that all concrete Queries follow. This way the code within Database remains functional, whichever type of Queries it works with.
- For example, to add a new Database type to the app, you'll only need to create a new Database subclass and override the factory method in it.