SPRING DATA ACCESS

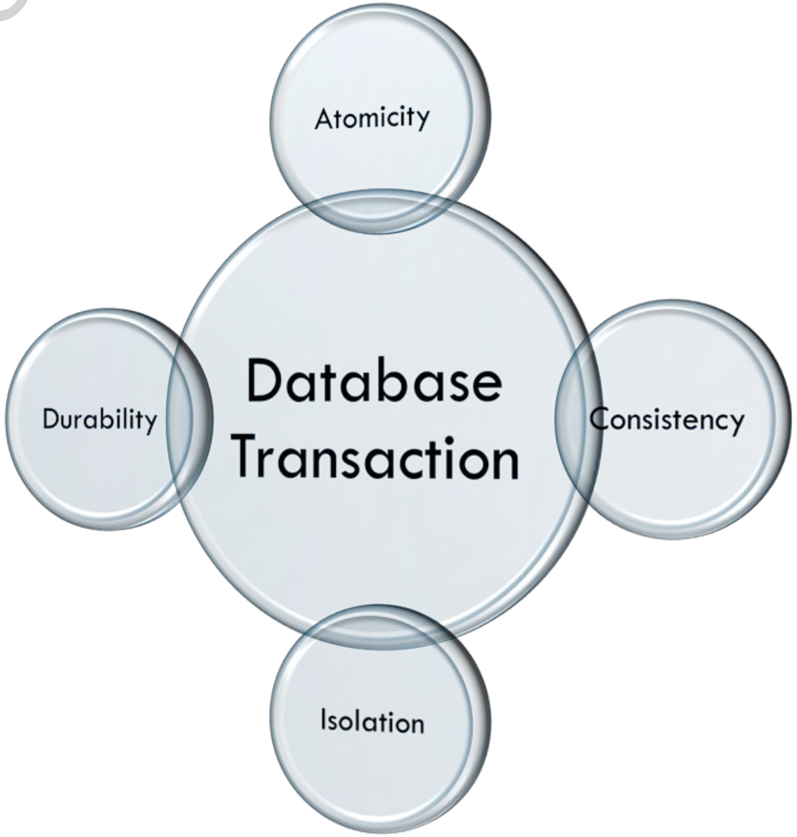
Session 3

OBJECTIVES

- Explain data access with Spring
- Describe Java Database Connectivity (JDBC) and Object-Relational Mapping (ORM) with MySQL
- Describe how to use ORM programming techniques

ACCESSING DATA WITH SPRING

- To maintain data integrity and consistency for RDBMS-oriented applications, transaction management is essential.
- Spring Framework provides transaction capabilities to Plain Old Java Objects (POJOs), thereby, serving as an alternate to Enterprise JavaBeans (EJB) transactions.



Database Transaction Properties

DAO SUPPORT

- A Data Access Object (DAO) is an object, which performs some specific data operations without revealing the details of the database to the user.
- Spring provides a DAO framework which contains interfaces, utility classes, and abstract classes.

Class	Description
JdbcDaoSupport	It is a super class for JDBC data access objects and requires a DataSource instance to be provided.
HibernateDaoSupport	It is a super class for Hibernate data access objects and requires a SessionFactory instance to be provided.
JdoDaoSupport	It is a super class for JDO data access objects and requires a PersistenceManagerFactory instance to be provided.
JpaDaoSupport	It is a super class for JPA data access objects and requires an EntityManagerFactory instance to be provided.

Abstract Classes for DAO Support in Spring

DAO DESIGN PATTERN

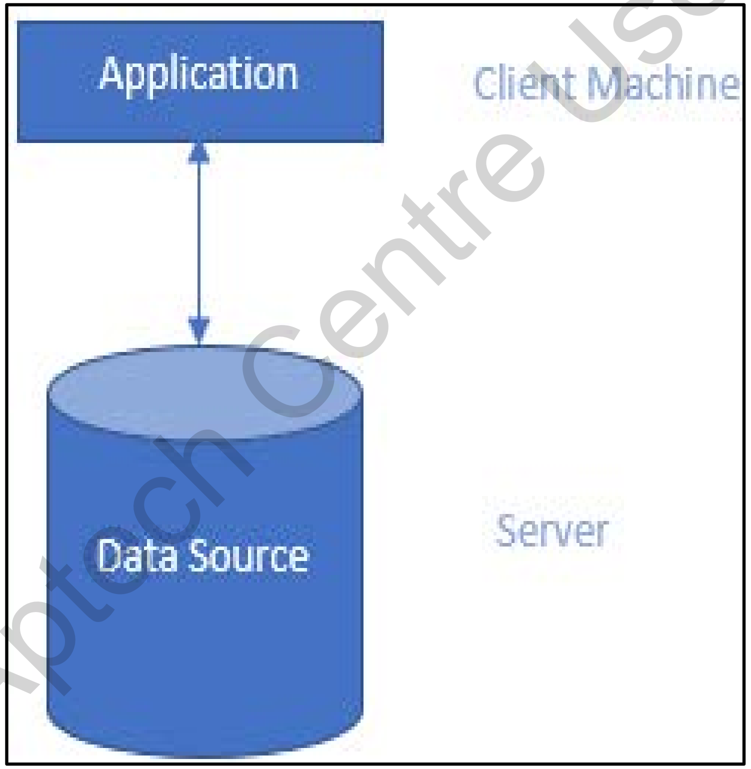
- DAO design pattern is built on encapsulation and abstraction principles.
- Protects the Spring application from any change made in the database or data access technology

Data Access Object Interface	Defines the standard operations to be performed on a model object.
Data Access Object Concrete Class	Implements the DAO interface and is responsible to get data from a data source (database/XML or any other storage mechanism).
Model Object or Value Object	Refers to a simple POJO that contains get/set methods to store data, which is retrieved using DAO class.

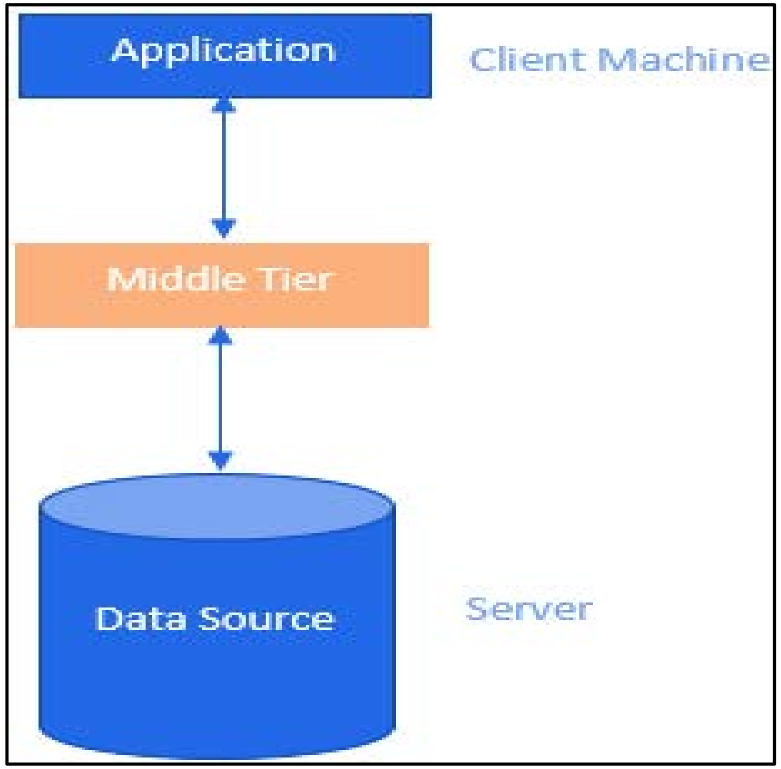
Members of DAO Pattern

JDBC

- JDBC is a data-access technology.
- JDBC performs data access in two- as well as three-tier architectures.



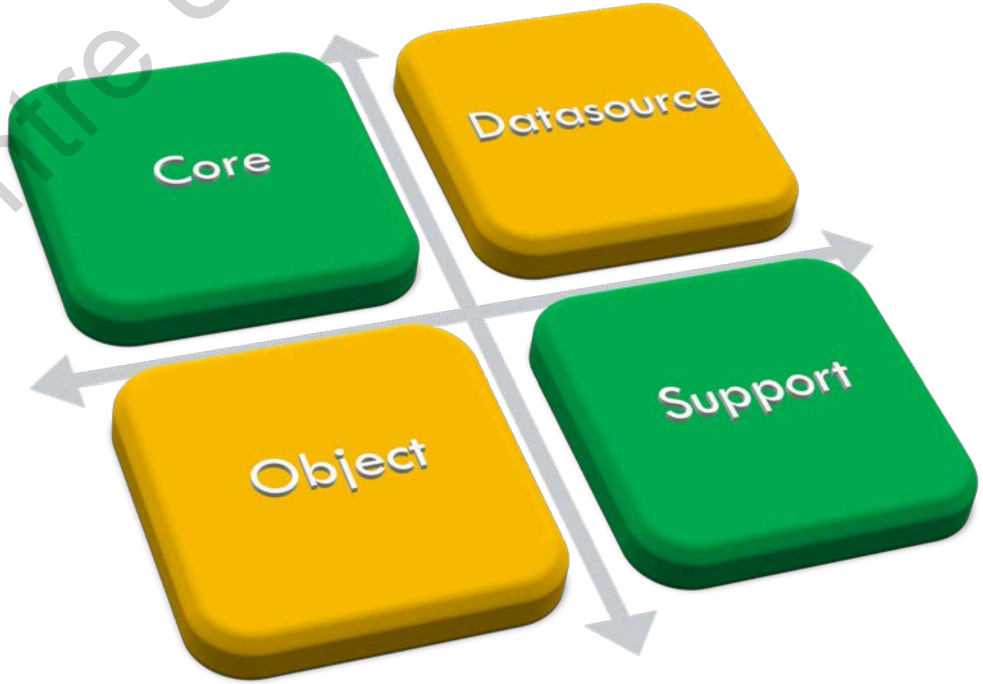
2-Tier Architecture



3-Tier Architecture

SPRING JDBC PACKAGES

- Spring Framework provides several in-built classes for JDBC that developers can extend.
- These classes belong to a specific Spring JDBC package.
- The most common class is the JdbcTemplate class.



Spring JDBC Packages

USING JDBCTEMPLATE CLASS

- Using this class in an application eliminates the need to write code for operational activities.
- This class helps catch exceptions in applications and map them to the corresponding exception hierarchy categories defined in the *org.springframework.dao* package.

Method	Description
public int update(String query)	Helps to insert, update, and delete records.
public int update(String query, Object... args)	Helps to insert, update, and delete records using PreparedStatement using given arguments.
public void execute(String query)	Helps to execute DDL query.
public T execute(String sql, PreparedStatementCallback action)	Helps to execute the query by using PreparedStatement callback.
public T query (String sql, ResultSetExtractor rse)	Helps to fetch records using ResultSetExtractor.
public List query(String sql, RowMapper rse)	Helps to fetch records using RowMapper.

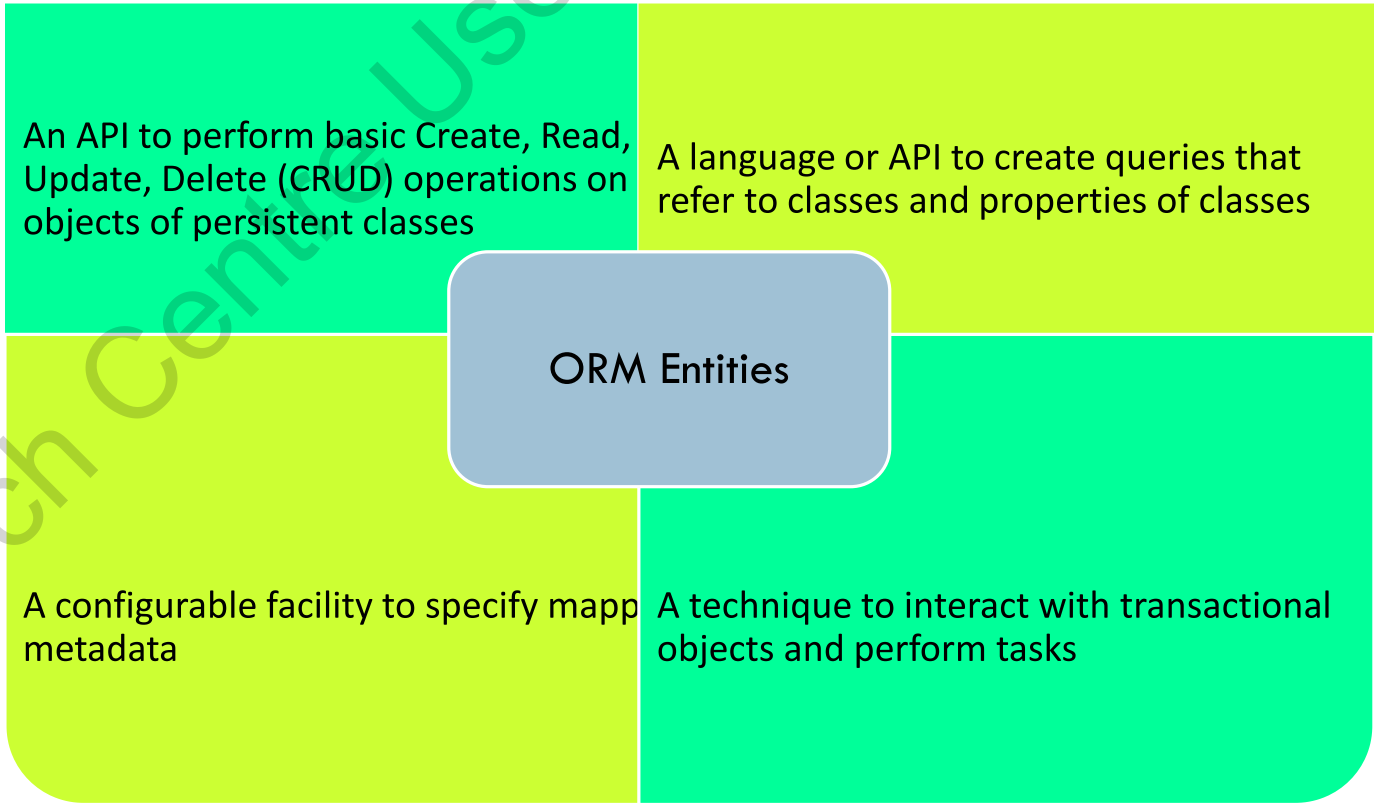
JdbcTemplate Class – Methods

MYSQL DATA ACCESS WITH ORM

- MySQL is an open-source RDBMS that uses Structured Query Language (SQL).
- Spring Framework provides programming support with MySQL database.
- Spring Framework enables the combination of ORM paradigm with MySQL.

INTRODUCING ORM

- ORM is a programming technique that helps developers to convert data between incompatible or conflicting systems in relational databases, XML repositories, other data sources, and OOP languages.



An API to perform basic Create, Read, Update, Delete (CRUD) operations on objects of persistent classes

A language or API to create queries that refer to classes and properties of classes

ORM Entities

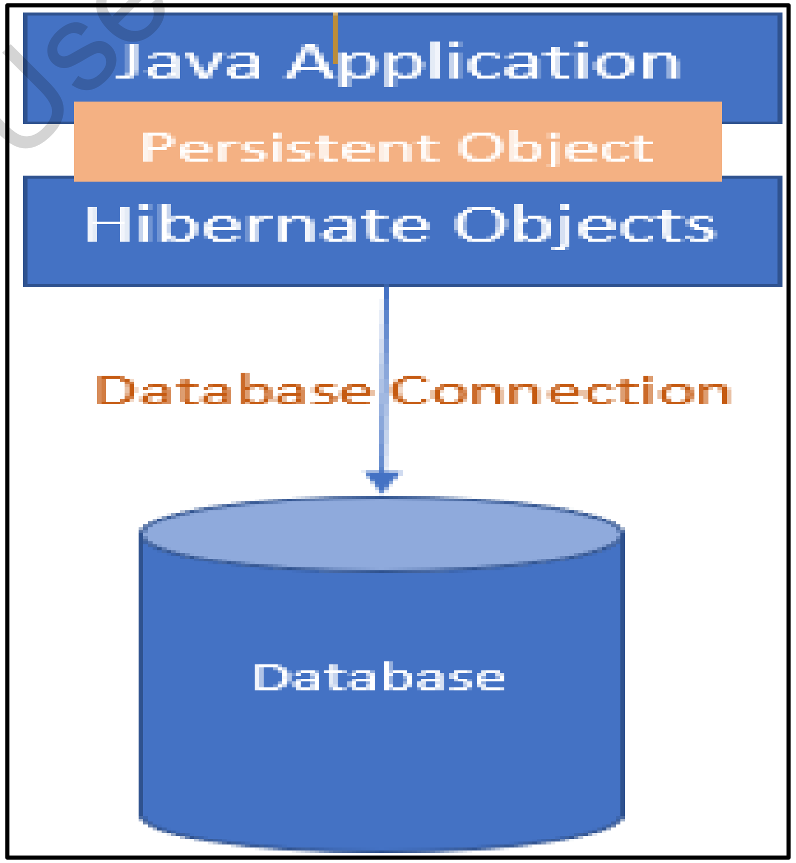
A configurable facility to specify mapp metadata

A technique to interact with transactional objects and perform tasks

ORM Entities

INTRODUCING HIBERNATE

- Hibernate is a powerful and high performance ORM solution created by Gavin King in 2001 for Java applications.
- Hibernate maps Java classes to database tables and Java data types to SQL data types.
- Hibernate replaces direct and persistence database accesses with high-level object handling functions.
- Hibernate does not need any application server to operate.



Hibernate Architecture

SUMMARY

- Spring framework provides transactional management capabilities to Plain Old Java Objects.
- A database transaction must have following four key properties, known as ACID.
- A Data Access Object (DAO) is an object, which performs some specific data operations without revealing the details of the database to the user.
- JDBC is a data-access technology that performs data access in two- as well as three-tier architectures.
- MySQL is an open-source RDBMS that uses Structured Query Language (SQL).
- ORM is an Object-oriented based programming technique that enables you to convert data between incompatible systems in data sources.
- Hibernate is a powerful, high performance ORM solution that does not need any application server support to operate.