Stock.java:

package com.cognizant.springlearn.model;

import jakarta.persistence.\*;

import java.math.BigDecimal;

import java.time.LocalDate;

@Entity

@Table(name = "stock")

public class Stock {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column(name = "st\_code")

private String code;

@Column(name = "st\_date")

private LocalDate date;

@Column(name = "st\_open")

private BigDecimal open;

@Column(name = "st\_close")

private BigDecimal close;

@Column(name = "st\_volume")

private BigDecimal volume;

// Getters and Setters

}

StockRepository.java

package com.cognizant.springlearn.repository;

import com.cognizant.springlearn.model.Stock;

import org.springframework.data.jpa.repository.JpaRepository;

import java.math.BigDecimal;

import java.time.LocalDate;

import java.util.List;

public interface StockRepository extends JpaRepository<Stock, Integer> {

List<Stock> findByCodeAndDateBetween(String code, LocalDate startDate, LocalDate endDate);

List<Stock> findByCodeAndCloseGreaterThan(String code, BigDecimal close);

List<Stock> findTop3ByOrderByVolumeDesc();

List<Stock> findTop3ByCodeOrderByCloseAsc(String code);

}

OrmLearnApplication.java:

@Autowired

private StockRepository stockRepository;

private void testStockQueries() {

List<Stock> fbSept2019 = stockRepository.findByCodeAndDateBetween(

"FB", LocalDate.of(2019, 9, 1), LocalDate.of(2019, 9, 30));

fbSept2019.forEach(System.out::println);

List<Stock> googleAbove1250 = stockRepository.findByCodeAndCloseGreaterThan(

"GOOGL", new BigDecimal("1250"));

googleAbove1250.forEach(System.out::println);

List<Stock> top3Volume = stockRepository.findTop3ByOrderByVolumeDesc();

top3Volume.forEach(System.out::println);

List<Stock> lowestNetflix = stockRepository.findTop3ByCodeOrderByCloseAsc("NFLX");

lowestNetflix.forEach(System.out::println);

}

@Override

public void run(String... args) throws Exception {

testStockQueries();

}