**Hands on Exercise 2 - Write queries on stock table using Query**

**CODE:**

**Stock Class: -**

package com.cognizant.ex2handson2.model;  
import jakarta.persistence.\*;  
import java.math.BigDecimal;  
import java.time.LocalDate;  
  
@Entity  
@Table(name = "stock")  
public class Stock {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name = "st\_id")  
 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;  
  
 public Stock() {  
 }  
  
 public Stock(int id, String code, LocalDate date, BigDecimal open, BigDecimal close, BigDecimal volume) {  
 this.id = id;  
 this.code = code;  
 this.date = date;  
 this.open = open;  
 this.close = close;  
 this.volume = volume;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getCode() {  
 return code;  
 }  
  
 public void setCode(String code) {  
 this.code = code;  
 }  
  
 public LocalDate getDate() {  
 return date;  
 }  
  
 public void setDate(LocalDate date) {  
 this.date = date;  
 }  
  
 public BigDecimal getOpen() {  
 return open;  
 }  
  
 public void setOpen(BigDecimal open) {  
 this.open = open;  
 }  
  
 public BigDecimal getClose() {  
 return close;  
 }  
  
 public void setClose(BigDecimal close) {  
 this.close = close;  
 }  
  
 public BigDecimal getVolume() {  
 return volume;  
 }  
  
 public void setVolume(BigDecimal volume) {  
 this.volume = volume;  
 }  
}

**StockRepository Interface: -**

package com.cognizant.ex2handson2.repository;  
  
import com.cognizant.ex2handson2.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 price);  
 List<Stock> findTop3ByOrderByVolumeDesc();  
 List<Stock> findTop3ByCodeOrderByCloseAsc(String code);  
}

**StockService Class: -**

package com.cognizant.ex2handson2.service;  
import com.cognizant.ex2handson2.model.Stock;  
import com.cognizant.ex2handson2.repository.StockRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.math.BigDecimal;  
import java.time.LocalDate;  
import java.util.List;  
  
@Service  
public class StockService {  
  
 @Autowired  
 private StockRepository stockRepository;  
  
 public List<Stock> getFacebookStocksInSeptember2019() {  
 return stockRepository.findByCodeAndDateBetween(  
 "FB",  
 LocalDate.*of*(2019, 9, 1),  
 LocalDate.*of*(2019, 9, 30));  
 }  
  
 public List<Stock> getGoogleStocksAbove1250() {  
 return stockRepository.findByCodeAndCloseGreaterThan("GOOGL", new BigDecimal("1250"));  
 }  
  
 public List<Stock> getTop3ByHighestVolume() {  
 return stockRepository.findTop3ByOrderByVolumeDesc();  
 }  
  
 public List<Stock> getLowestNetflixStocks() {  
 return stockRepository.findTop3ByCodeOrderByCloseAsc("NFLX");  
 }  
}

**Main Class: -**

package com.cognizant.ex2handson2;  
  
import com.cognizant.ex2handson2.model.Stock;  
import com.cognizant.ex2handson2.service.StockService;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.ApplicationContext;  
  
@SpringBootApplication  
public class Ex2handson2Application {  
  
 public static void main(String[] args) {  
 ApplicationContext context =SpringApplication.*run*(Ex2handson2Application.class, args);  
 StockService service = context.getBean(StockService.class);  
 System.*out*.println("Facebook stocks in September 2019:");  
 service.getFacebookStocksInSeptember2019()  
 .forEach(Ex2handson2Application::*printStock*);  
 System.*out*.println("\nGoogle stocks > 1250:");  
 service.getGoogleStocksAbove1250()  
 .forEach(Ex2handson2Application::*printStock*);  
  
 System.*out*.println("\nTop 3 Highest Volume Days:");  
 service.getTop3ByHighestVolume()  
 .forEach(Ex2handson2Application::*printStock*);  
  
 System.*out*.println("\nLowest Netflix Stock Days:");  
 service.getLowestNetflixStocks()  
 .forEach(Ex2handson2Application::*printStock*);  
  
 }  
 public static void printStock(Stock stock) {  
 System.*out*.printf("%s | %s | %.2f | %.2f | %.0f\n",  
 stock.getCode(), stock.getDate(), stock.getOpen(),  
 stock.getClose(), stock.getVolume());  
 }  
  
}

**OUTPUT:**



