```
/Library/Java/JavaVirtualMachines/jdk-17.0.1.jdk/Contents/Home/bin/java -XX:TieredStopAtLevel=1 -noverify -
Dspring.output.ansi.enabled=always -Dcom.sun.management.jmxremote -Dspring.jmx.enabled=true -
Dspring.liveBeansView.mbeanDomain -Dspring.application.admin.enabled=true -javaagent:/Applications/IntelliJ
IDEA.app/Contents/lib/idea_rt.jar=50582:/Applications/IntelliJ IDEA.app/Contents/bin -Dfile.encoding=UTF-8 -classpath
/Users/sophearyrin/Documents/Pheary_MIU/DE courses/On-Campus course/CS489/cs489-
appsd/labsolutions/lab6/target/classes:/Users/sophearyrin/.m2/repository/org/springframework/boot/spring-boot-starter-data-
jpa/3.1.4/spring-boot-starter-data-jpa-3.1.4.jar:/Users/sophearyrin/.m2/repository/org/springframework/boot/spring-boot-starter-
aop/3.1.4/spring-boot-starter-aop-3.1.4.jar:/Users/sophearyrin/.m2/repository/org/springframework/spring-aop/6.0.12/spring-aop-
6.0.12.jar:/Users/sophearyrin/.m2/repository/org/aspectj/aspectjweaver/1.9.20/aspectjweaver-
1.9.20.jar:/Users/sophearyrin/.m2/repository/org/springframework/boot/spring-boot-starter-jdbc/3.1.4/spring-boot-starter-jdbc-
3.1.4.jar:/Users/sophearyrin/.m2/repository/com/zaxxer/HikariCP/5.0.1/HikariCP-
5.0.1.jar:/Users/sophearyrin/.m2/repository/org/springframework/spring-jdbc/6.0.12/spring-jdbc-
6.0.12.jar:/Users/sophearyrin/.m2/repository/org/hibernate/orm/hibernate-core/6.2.9.Final/hibernate-core-
6.2.9.Final.jar:/Users/sophearyrin/.m2/repository/jakarta/persistence/jakarta.persistence-api/3.1.0/jakarta.persistence-api-
3.1.0.jar:/Users/sophearyrin/.m2/repository/jakarta/transaction/jakarta.transaction-api/2.0.1/jakarta.transaction-api-
2.0.1.jar:/Users/sophearyrin/.m2/repository/org/jboss/logging/jboss-logging/3.5.3.Final/jboss-logging-
3.5.3.Final.jar:/Users/sophearyrin/.m2/repository/org/hibernate/common/hibernate-commons-annotations/6.0.6.Final/hibernate-
commons-annotations-6.0.6.Final.jar:/Users/sophearyrin/.m2/repository/io/smallrye/jandex/3.0.5/jandex-
3.0.5.jar:/Users/sophearyrin/.m2/repository/com/fasterxml/classmate/1.5.1/classmate-
1.5.1.jar:/Users/sophearyrin/.m2/repository/net/bytebuddy/byte-buddy/1.14.8/byte-buddy-
1.14.8.jar:/Users/sophearyrin/.m2/repository/org/glassfish/jaxb/jaxb-runtime/4.0.3/jaxb-runtime-
4.0.3.jar:/Users/sophearyrin/.m2/repository/org/glassfish/jaxb/jaxb-core/4.0.3/jaxb-core-
4.0.3.jar:/Users/sophearyrin/.m2/repository/org/eclipse/angus/angus-activation/2.0.1/angus-activation-
2.0.1.jar:/Users/sophearyrin/.m2/repository/org/glassfish/jaxb/txw2/4.0.3/txw2-
4.0.3.jar:/Users/sophearyrin/.m2/repository/com/sun/istack/istack-commons-runtime/4.1.2/istack-commons-runtime-
4.1.2.jar:/Users/sophearyrin/.m2/repository/jakarta/inject/jakarta.inject-api/2.0.1/jakarta.inject-api-
2.0.1.jar:/Users/sophearyrin/.m2/repository/org/antlr/antlr4-runtime/4.10.1/antlr4-runtime-
4.10.1.jar:/Users/sophearyrin/.m2/repository/org/springframework/data/spring-data-jpa/3.1.4/spring-data-jpa-
3.1.4.jar:/Users/sophearyrin/.m2/repository/org/springframework/data/spring-data-commons/3.1.4/spring-data-commons-
3.1.4.jar:/Users/sophearyrin/.m2/repository/org/springframework/spring-orm/6.0.12/spring-orm-
6.0.12.jar:/Users/sophearyrin/.m2/repository/org/springframework/spring-context/6.0.12/spring-context-
```

```
6.0.12.jar:/Users/sophearyrin/.m2/repository/org/springframework/spring-tx/6.0.12/spring-tx-
6.0.12.jar:/Users/sophearyrin/.m2/repository/org/springframework/spring-beans/6.0.12/spring-beans-
6.0.12.jar:/Users/sophearyrin/.m2/repository/jakarta/annotation/jakarta.annotation-api/2.1.1/jakarta.annotation-api-
2.1.1.jar:/Users/sophearyrin/.m2/repository/org/slf4j/slf4j-api/2.0.9/slf4j-api-
2.0.9.jar:/Users/sophearyrin/.m2/repository/org/springframework/spring-aspects/6.0.12/spring-aspects-
6.0.12.jar:/Users/sophearyrin/.m2/repository/org/springframework/boot/spring-boot-starter-web/3.1.4/spring-boot-starter-web-
3.1.4.jar:/Users/sophearyrin/.m2/repository/org/springframework/boot/spring-boot-starter/3.1.4/spring-boot-starter-
3.1.4.jar:/Users/sophearyrin/.m2/repository/org/springframework/boot/spring-boot/3.1.4/spring-boot-
3.1.4.jar:/Users/sophearyrin/.m2/repository/org/springframework/boot/spring-boot-autoconfigure/3.1.4/spring-boot-
autoconfigure-3.1.4.jar:/Users/sophearyrin/.m2/repository/org/springframework/boot/spring-boot-starter-logging/3.1.4/spring-
boot-starter-logging-3.1.4.jar:/Users/sophearyrin/.m2/repository/ch/qos/logback/logback-classic/1.4.11/logback-classic-
1.4.11.jar:/Users/sophearyrin/.m2/repository/ch/qos/logback/logback-core/1.4.11/logback-core-
1.4.11.jar:/Users/sophearyrin/.m2/repository/org/apache/logging/log4j/log4j-to-slf4j/2.20.0/log4j-to-slf4j-
2.20.0.jar:/Users/sophearyrin/.m2/repository/org/apache/logging/log4j/log4j-api/2.20.0/log4j-api-
2.20.0.jar:/Users/sophearyrin/.m2/repository/org/slf4j/jul-to-slf4j/2.0.9/jul-to-slf4j-
2.0.9.jar:/Users/sophearyrin/.m2/repository/org/yaml/snakeyaml/1.33/snakeyaml-
1.33.jar:/Users/sophearyrin/.m2/repository/org/springframework/boot/spring-boot-starter-json/3.1.4/spring-boot-starter-json-
3.1.4.jar:/Users/sophearyrin/.m2/repository/com/fasterxml/jackson/core/jackson-databind/2.15.2/jackson-databind-
2.15.2.jar:/Users/sophearyrin/.m2/repository/com/fasterxml/jackson/core/jackson-annotations/2.15.2/jackson-annotations-
2.15.2.jar:/Users/sophearyrin/.m2/repository/com/fasterxml/jackson/core/jackson-core/2.15.2/jackson-core-
2.15.2.jar:/Users/sophearyrin/.m2/repository/com/fasterxml/jackson/datatype/jackson-datatype-jdk8/2.15.2/jackson-datatype-
jdk8-2.15.2.jar:/Users/sophearyrin/.m2/repository/com/fasterxml/jackson/datatype/jackson-datatype-jsr310/2.15.2/jackson-
datatype-jsr310-2.15.2.jar:/Users/sophearyrin/.m2/repository/com/fasterxml/jackson/module/jackson-module-parameter-
names/2.15.2/jackson-module-parameter-names-2.15.2.jar:/Users/sophearyrin/.m2/repository/org/springframework/boot/spring-
boot-starter-tomcat/3.1.4/spring-boot-starter-tomcat-
3.1.4.jar:/Users/sophearyrin/.m2/repository/org/apache/tomcat/embed/tomcat-embed-core/10.1.13/tomcat-embed-core-
10.1.13.jar:/Users/sophearyrin/.m2/repository/org/apache/tomcat/embed/tomcat-embed-el/10.1.13/tomcat-embed-el-
10.1.13.jar:/Users/sophearyrin/.m2/repository/org/apache/tomcat/embed/tomcat-embed-websocket/10.1.13/tomcat-embed-
websocket-10.1.13.jar:/Users/sophearyrin/.m2/repository/org/springframework/spring-web/6.0.12/spring-web-
6.0.12.jar:/Users/sophearyrin/.m2/repository/io/micrometer/micrometer-observation/1.11.4/micrometer-observation-
1.11.4.jar:/Users/sophearyrin/.m2/repository/io/micrometer/micrometer-commons/1.11.4/micrometer-commons-
```

```
1.11.4.jar:/Users/sophearyrin/.m2/repository/org/springframework/spring-webmvc/6.0.12/spring-webmvc-
6.0.12.jar:/Users/sophearyrin/.m2/repository/org/springframework/spring-expression/6.0.12/spring-expression-
6.0.12.jar:/Users/sophearyrin/.m2/repository/com/mysql/mysql-connector-j/8.0.33/mysql-connector-j-
8.0.33.jar:/Users/sophearyrin/.m2/repository/org/projectlombok/lombok/1.18.30/lombok-
1.18.30.jar:/Users/sophearyrin/.m2/repository/jakarta/xml/bind/jakarta.xml.bind-api/4.0.1/jakarta.xml.bind-api-
4.0.1.jar:/Users/sophearyrin/.m2/repository/jakarta/activation/jakarta.activation-api/2.1.2/jakarta.activation-api-
2.1.2.jar:/Users/sophearyrin/.m2/repository/org/springframework/spring-core/6.0.12/spring-core-
6.0.12.jar:/Users/sophearyrin/.m2/repository/org/springframework/spring-jcl/6.0.12/spring-jcl-6.0.12.jar
miu.edu.cs489.ads_springapplication.AdsSpringApplication
Java HotSpot(TM) 64-Bit Server VM warning: Options -Xverify:none and -noverify were deprecated in JDK 13 and will likely be
removed in a future release.

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.1.4)

2023-10-16T23:17:16.639-05:00  INFO 49411 --- [           main] m.e.c.a.AdsSpringApplication             : Starting AdsSpringApplication
using Java 17.0.1 with PID 49411 (/Users/sophearyrin/Documents/Pheary_MIU/DE courses/On-Campus course/CS489/cs489-
appsd/labsolutions/lab6/target/classes started by sophearyrin in /Users/sophearyrin/Documents/Pheary_MIU/DE courses/On-
Campus course/CS489/cs489-appsd)
2023-10-16T23:17:16.644-05:00  INFO 49411 --- [           main] m.e.c.a.AdsSpringApplication             : No active profile set, falling back
to 1 default profile: "default"
2023-10-16T23:17:17.427-05:00  INFO 49411 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data
JPA repositories in DEFAULT mode.
2023-10-16T23:17:17.484-05:00  INFO 49411 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data
repository scanning in 49 ms. Found 5 JPA repository interfaces.
```

```
2023-10-16T23:17:17.886-05:00  INFO 49411 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with
port(s): 8080 (http)
2023-10-16T23:17:17.895-05:00  INFO 49411 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2023-10-16T23:17:17.896-05:00  INFO 49411 --- [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine:
[Apache Tomcat/10.1.13]
2023-10-16T23:17:17.961-05:00  INFO 49411 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded
WebApplicationContext
2023-10-16T23:17:17.961-05:00  INFO 49411 --- [           main] w.s.c.ServletWebServerApplicationContext : Root
WebApplicationContext: initialization completed in 1235 ms
2023-10-16T23:17:18.079-05:00  INFO 49411 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing
PersistenceUnitInfo [name: default]
2023-10-16T23:17:18.117-05:00  INFO 49411 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core
version 6.2.9.Final
2023-10-16T23:17:18.119-05:00  INFO 49411 --- [           main] org.hibernate.cfg.Environment            : HHH000406: Using bytecode
reflection optimizer
2023-10-16T23:17:18.229-05:00  INFO 49411 --- [           main] o.h.b.i.BytecodeProviderInitiator        : HHH000021: Bytecode provider
name : bytebuddy
2023-10-16T23:17:18.311-05:00  INFO 49411 --- [           main] o.s.o.j.p.SpringPersistenceUnitInfo      : No LoadTimeWeaver setup:
ignoring JPA class transformer
2023-10-16T23:17:18.320-05:00  INFO 49411 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2023-10-16T23:17:18.547-05:00  INFO 49411 --- [           main] com.zaxxer.hikari.pool.HikariPool        : HikariPool-1 - Added
connection com.mysql.cj.jdbc.ConnectionImpl@18cf5c52
2023-10-16T23:17:18.549-05:00  INFO 49411 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start
completed.
2023-10-16T23:17:18.744-05:00  INFO 49411 --- [           main] o.h.b.i.BytecodeProviderInitiator        : HHH000021: Bytecode provider
name : bytebuddy
2023-10-16T23:17:19.222-05:00  INFO 49411 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatform
implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: 
    alter table appointments 
       drop 
```

```
        foreign key FKqsnula0nj86s67otibu2j2hsv
Hibernate: 
    alter table appointments 
        drop 
        foreign key FK8exap5wmg8kmb1g1rx3by21yt
Hibernate: 
    alter table appointments 
        drop 
        foreign key FK68v0hdkvl8qnssll24tmugia6
Hibernate: 
    alter table patients 
        drop 
        foreign key FKjc8017x8ae0rqi11m8jmny646
Hibernate: 
    alter table surgeries 
        drop 
        foreign key FK51c727fvh92teps6wulnss1hs
Hibernate: 
    drop table if exists addresses
Hibernate: 
    drop table if exists appointments
Hibernate: 
    drop table if exists dentists
Hibernate: 
    drop table if exists patients
Hibernate: 
    drop table if exists surgeries
Hibernate: 
    create table addresses (
        address_id integer not null auto_increment,
        zip_code varchar(5),
        city varchar(20),
        state varchar(20),
        street varchar(20),
        primary key (address_id)
    ) engine=InnoDB
Hibernate:
    create table appointments (
        appointment_id integer not null auto_increment,
        date date,
        dentist_id integer,
        patient_id integer,
        surgery_id integer,
        primary key (appointment_id)
    ) engine=InnoDB
Hibernate:
    create table dentists (
        dentist_id integer not null auto_increment,
        dob date,
        email varchar(255),
        first_name varchar(255),
        last_name varchar(255),
        phone_number varchar(255),
        specialization varchar(255),
        primary key (dentist_id)
    ) engine=InnoDB
Hibernate:
    create table patients (
        address_id integer,
        dob date,
        patient_id integer not null auto_increment,
        email varchar(255),
        fist_name varchar(255),
        last_name varchar(255),
        phone_number varchar(255),
        primary key (patient_id)
    ) engine=InnoDB
Hibernate: 
    create table surgeries (
        address_id integer,
        surgery_id integer not null auto_increment,
        name varchar(255),
        phone_number varchar(255),
        primary key (surgery_id)
    ) engine=InnoDB
Hibernate: 
    alter table patients 
       add constraint UK_pveescs3fe3p1eaabryivpydo unique (address_id)
Hibernate: 
    alter table surgeries 
       add constraint UK_rrc99l4xsueifgrsbed241ehl unique (address_id)
Hibernate: 
    alter table appointments 
       add constraint FKqsnula0nj86s67otibu2j2hsv 
       foreign key (dentist_id) 
       references dentists (dentist_id)
Hibernate: 
    alter table appointments 
       add constraint FK8exap5wmg8kmb1g1rx3by21yt 
       foreign key (patient_id) 
       references patients (patient_id)
Hibernate: 
    alter table appointments 
```

```
        add constraint FK68v0hdkvl8qnssll24tmugia6
        foreign key (surgery_id)
        references surgeries (surgery_id)
Hibernate:
    alter table patients
        add constraint FKjc8017x8ae0rqi11m8jmny646
        foreign key (address_id)
        references addresses (address_id)
Hibernate:
    alter table surgeries
        add constraint FK51c727fvh92teps6wulnss1hs
        foreign key (address_id)
        references addresses (address_id)
2023-10-16T23:17:19.565-05:00  INFO 49411 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA
EntityManagerFactory for persistence unit 'default'
2023-10-16T23:17:19.855-05:00  WARN 49411 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-
view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure
spring.jpa.open-in-view to disable this warning
2023-10-16T23:17:20.108-05:00  INFO 49411 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on
port(s): 8080 (http) with context path ''
2023-10-16T23:17:20.115-05:00  INFO 49411 --- [           main] m.e.c.a.AdsSpringApplication             : Started AdsSpringApplication in
3.842 seconds (process running for 4.452)
Hibernate:
    insert
    into
        addresses
        (city,state,street,zip_code)
    values
        (?,?,?,?)
Hibernate:
    select
```

```
        a1_0.address_id,
        a1_0.city,
        a1_0.state,
        a1_0.street,
        a1_0.zip_code
    from
        addresses a1_0
Hibernate:
    select
        p1_0.patient_id,
        a1_0.address_id,
        a1_0.city,
        a1_0.state,
        a1_0.street,
        s1_0.surgery_id,
        s1_0.name,
        s1_0.phone_number,
        a1_0.zip_code,
        p1_0.dob,
        p1_0.email,
        p1_0.fist_name,
        p1_0.last_name,
        p1_0.phone_number
    from
        patients p1_0
    left join
        addresses a1_0
            on a1_0.address_id=p1_0.address_id
    left join
        surgeries s1_0
            on a1_0.address_id=s1_0.address_id
```

```
   where
      p1_0.address_id=?
Hibernate: 
   select
      s1_0.surgery_id,
      a1_0.address_id,
      a1_0.city,
      p1_0.patient_id,
      p1_0.dob,
      p1_0.email,
      p1_0.fist_name,
      p1_0.last_name,
      p1_0.phone_number,
      a1_0.state,
      a1_0.street,
      a1_0.zip_code,
      s1_0.name,
      s1_0.phone_number
   from
      surgeries s1_0
   left join
      addresses a1_0
         on a1_0.address_id=s1_0.address_id
   left join
      patients p1_0
         on a1_0.address_id=p1_0.address_id
   where
      s1_0.address_id=?
[Address{addressId=1, street='1000N', city='FairField', state='IA', zipCode='78660'}]
Hibernate: 
   select
```

```
        a1_0.address_id,
        a1_0.city,
        a1_0.state,
        a1_0.street,
        a1_0.zip_code
    from
        addresses a1_0
Hibernate:
    select
        p1_0.patient_id,
        a1_0.address_id,
        a1_0.city,
        a1_0.state,
        a1_0.street,
        s1_0.surgery_id,
        s1_0.name,
        s1_0.phone_number,
        a1_0.zip_code,
        p1_0.dob,
        p1_0.email,
        p1_0.fist_name,
        p1_0.last_name,
        p1_0.phone_number
    from
        patients p1_0
    left join
        addresses a1_0
            on a1_0.address_id=p1_0.address_id
    left join
        surgeries s1_0
            on a1_0.address_id=s1_0.address_id
    where
        p1_0.address_id=?
Hibernate: 
    select
        s1_0.surgery_id,
        a1_0.address_id,
        a1_0.city,
        p1_0.patient_id,
        p1_0.dob,
        p1_0.email,
        p1_0.fist_name,
        p1_0.last_name,
        p1_0.phone_number,
        a1_0.state,
        a1_0.street,
        a1_0.zip_code,
        s1_0.name,
        s1_0.phone_number 
    from
        surgeries s1_0 
    left join
        addresses a1_0 
            on a1_0.address_id=s1_0.address_id 
    left join
        patients p1_0 
            on a1_0.address_id=p1_0.address_id 
    where
        s1_0.address_id=?
Hibernate: 
    insert 
    into
```

```
        patients
        (address_id,dob,email,fist_name,last_name,phone_number)
    values
        (?,?,?,?,?,?)
Hibernate:
    select
        p1_0.patient_id,
        p1_0.address_id,
        p1_0.dob,
        p1_0.email,
        p1_0.fist_name,
        p1_0.last_name,
        p1_0.phone_number
    from
        patients p1_0
Hibernate:
    select
        a1_0.address_id,
        a1_0.city,
        p1_0.patient_id,
        p1_0.dob,
        p1_0.email,
        p1_0.fist_name,
        p1_0.last_name,
        p1_0.phone_number,
        a1_0.state,
        a1_0.street,
        s1_0.surgery_id,
        s1_0.name,
        s1_0.phone_number,
        a1_0.zip_code
    from
        addresses a1_0
    left join
        patients p1_0
            on a1_0.address_id=p1_0.address_id
    left join
        surgeries s1_0
            on a1_0.address_id=s1_0.address_id
    where
        a1_0.address_id=?
[Patient{patientId=1, fistName='Gillian', lastName='White', phoneNumber='8754321435', email='gillina@gmail.com', dob=1998-01-01}]
Hibernate:
    insert
    into
        dentists
        (dob,email,first_name,last_name,phone_number,specialization)
    values
        (?,?,?,?,?,?)
Hibernate:
    select
        d1_0.dentist_id,
        d1_0.dob,
        d1_0.email,
        d1_0.first_name,
        d1_0.last_name,
        d1_0.phone_number,
        d1_0.specialization
    from
        dentists d1_0
```

```
[Dentist{dentistId=1, firstName='Tony', lastName='Smith', phoneNumber='234567890', email='tony@gmail.com', dob=1997-02-02,
specialization='Head'}]
Hibernate: 
    select
        a1_0.address_id,
        a1_0.city,
        a1_0.state,
        a1_0.street,
        a1_0.zip_code 
    from
        addresses a1_0
Hibernate: 
    select
        p1_0.patient_id,
        a1_0.address_id,
        a1_0.city,
        a1_0.state,
        a1_0.street,
        s1_0.surgery_id,
        s1_0.name,
        s1_0.phone_number,
        a1_0.zip_code,
        p1_0.dob,
        p1_0.email,
        p1_0.fist_name,
        p1_0.last_name,
        p1_0.phone_number 
    from
        patients p1_0 
    left join
        addresses a1_0
```

```
            on a1_0.address_id=p1_0.address_id
    left join
        surgeries s1_0
            on a1_0.address_id=s1_0.address_id
    where
        p1_0.address_id=?
Hibernate:
    select
        s1_0.surgery_id,
        a1_0.address_id,
        a1_0.city,
        p1_0.patient_id,
        p1_0.dob,
        p1_0.email,
        p1_0.fist_name,
        p1_0.last_name,
        p1_0.phone_number,
        a1_0.state,
        a1_0.street,
        a1_0.zip_code,
        s1_0.name,
        s1_0.phone_number
    from
        surgeries s1_0
    left join
        addresses a1_0
            on a1_0.address_id=s1_0.address_id
    left join
        patients p1_0
            on a1_0.address_id=p1_0.address_id
    where
```

```
        s1_0.address_id=?
Hibernate: 
    insert 
    into
        surgeries
        (address_id,name,phone_number) 
    values
        (?,?,?)
Hibernate: 
    select
        s1_0.surgery_id,
        s1_0.address_id,
        s1_0.name,
        s1_0.phone_number 
    from
        surgeries s1_0
Hibernate: 
    select
        a1_0.address_id,
        a1_0.city,
        p1_0.patient_id,
        p1_0.dob,
        p1_0.email,
        p1_0.fist_name,
        p1_0.last_name,
        p1_0.phone_number,
        a1_0.state,
        a1_0.street,
        s1_0.surgery_id,
        s1_0.name,
        s1_0.phone_number,
        a1_0.zip_code
    from
        addresses a1_0
    left join
        patients p1_0
            on a1_0.address_id=p1_0.address_id
    left join
        surgeries s1_0
            on a1_0.address_id=s1_0.address_id
    where
        a1_0.address_id=?
[Surgery{surgeryId=1, name='S15', phoneNumber='345678905678'}]
Hibernate:
    select
        s1_0.surgery_id,
        s1_0.address_id,
        s1_0.name,
        s1_0.phone_number
    from
        surgeries s1_0
Hibernate:
    select
        a1_0.address_id,
        a1_0.city,
        p1_0.patient_id,
        p1_0.dob,
        p1_0.email,
        p1_0.fist_name,
        p1_0.last_name,
        p1_0.phone_number,
        a1_0.state,
        a1_0.street,
        s1_0.surgery_id,
        s1_0.name,
        s1_0.phone_number,
        a1_0.zip_code
    from
        addresses a1_0
    left join
        patients p1_0
            on a1_0.address_id=p1_0.address_id
    left join
        surgeries s1_0
            on a1_0.address_id=s1_0.address_id
    where
        a1_0.address_id=?
Hibernate:
    select
        d1_0.dentist_id,
        d1_0.dob,
        d1_0.email,
        d1_0.first_name,
        d1_0.last_name,
        d1_0.phone_number,
        d1_0.specialization
    from
        dentists d1_0
Hibernate:
    select
        p1_0.patient_id,
        p1_0.address_id,
        p1_0.dob,
        p1_0.email,
        p1_0.fist_name,
        p1_0.last_name,
        p1_0.phone_number
    from
        patients p1_0
Hibernate:
    select
        a1_0.address_id,
        a1_0.city,
        p1_0.patient_id,
        p1_0.dob,
        p1_0.email,
        p1_0.fist_name,
        p1_0.last_name,
        p1_0.phone_number,
        a1_0.state,
        a1_0.street,
        s1_0.surgery_id,
        s1_0.name,
        s1_0.phone_number,
        a1_0.zip_code
    from
        addresses a1_0
    left join
        patients p1_0
            on a1_0.address_id=p1_0.address_id
    left join
        surgeries s1_0
            on a1_0.address_id=s1_0.address_id
    where
```

```
        a1_0.address_id=?
Hibernate: 
    insert 
    into
        appointments
        (date,dentist_id,patient_id,surgery_id) 
    values
        (?,?,?,?)
Hibernate: 
    select
        a1_0.appointment_id,
        a1_0.date,
        a1_0.dentist_id,
        a1_0.patient_id,
        a1_0.surgery_id 
    from
        appointments a1_0
Hibernate: 
    select
        d1_0.dentist_id,
        d1_0.dob,
        d1_0.email,
        d1_0.first_name,
        d1_0.last_name,
        d1_0.phone_number,
        d1_0.specialization 
    from
        dentists d1_0 
    where
        d1_0.dentist_id=?
Hibernate: 
    select
        p1_0.patient_id,
        a1_0.address_id,
        a1_0.city,
        a1_0.state,
        a1_0.street,
        s1_0.surgery_id,
        s1_0.name,
        s1_0.phone_number,
        a1_0.zip_code,
        p1_0.dob,
        p1_0.email,
        p1_0.fist_name,
        p1_0.last_name,
        p1_0.phone_number
    from
        patients p1_0
    left join
        addresses a1_0
            on a1_0.address_id=p1_0.address_id
    left join
        surgeries s1_0
            on a1_0.address_id=s1_0.address_id
    where
        p1_0.patient_id=?
[Appointment{appointmentId=1, date=2012-09-13
, surgery=Surgery{surgeryId=1, name='S15', phoneNumber='345678905678'}
, dentist=Dentist{dentistId=1, firstName='Tony', lastName='Smith', phoneNumber='234567890', email='tony@gmail.com',
dob=1997-02-02, specialization='Head'}
, patient=Patient{patientId=1, fistName='Gillian', lastName='White', phoneNumber='8754321435', email='gillina@gmail.com',
dob=1998-01-01}}]
```