Informationssysteme II
Übungsblatt 3

Christine Pries, Mareike Wagner, Felix Oppermann (Gruppe 18)

19. Mai 2005

Aufgabe 1 – ORDBMS – ORACLE 9i (40 Punkte)

a)
```
CREATE OR REPLACE TYPE Lied AS OBJECT
(
  titel              VARCHAR2(50),
  laenge             NUMBER
)
/

CREATE OR REPLACE TYPE Lied_table AS TABLE OF Lied
/

CREATE OR REPLACE TYPE Album AS OBJECT
(
  titel              VARCHAR2(50),
  erscheinungsjahr   NUMBER,
  lieder             Lied_table
)
/

CREATE OR REPLACE TYPE Album_table AS TABLE OF Album
/

CREATE OR REPLACE TYPE Interpret AS OBJECT
(
  name               VARCHAR2(50),
  alben              Album_table
)
/

CREATE TABLE interpreten OF Interpret
NESTED TABLE alben STORE AS alben
(
  NESTED TABLE lieder STORE AS lieder
);
```

Ausgabe:

```
Type created.

Type created.

Type created.

Type created.

Type created.

Table created.
```

b)
```
INSERT INTO interpreten VALUES
(
  Interpret
  (
    'Wir sind Helden',
    Album_table
    (
      Album
      (
```

```
        'Von hier an blind',
        2005,
        Lied_table
        (
          Lied
          (
            'Echolot',
            2
          )
        )
      )
    )
  )
);

INSERT INTO interpreten VALUES
(
  Interpret
  (
    'AC/DC',
    Album_table
    (
      Album
      (
        'Family jewels',
        2005,
        Lied_table
        (
          Lied
          (
            'Jailbreak',
            3
          ),
          Lied
          (
            'T.N.T.',
            2
          )
        )
      )
    )
  )
);
```

Ausgabe:

```
1 row created.

1 row created.
```

c)
```
SELECT DISTINCT(interpreten.name)
  FROM interpreten, TABLE(interpreten.alben) alben,
       TABLE(alben.lieder) lieder
  WHERE lieder.laenge = 2;
```

Ausgabe:

NAME
AC/DC
Wir sind Helden

Aufgabe 2 – OODBMS – OBJEKTSTORE (60 Punkte)

1.
```java
package objectstore;

import java.util.Enumeration;
import java.util.Iterator;
import java.util.Set;
import java.util.NoSuchElementException;

import com.odi.*;
import com.odi.util.query.FreeVariableBindings;
import com.odi.util.query.FreeVariables;
import com.odi.util.query.Query;

public class Anfrage1 {
    public static void main(String[] args) {

        Session session = Session.create(null, null);
        session.join ();

        Database db = Database.open("uni.odb", ObjectStore.READONLY);
        Transaction tr = Transaction.begin(ObjectStore.READONLY);
        Uni uni = (Uni) db.getRoot("uni");

        System.out.println("Alle Vorlesungen von \"Chaya Mainz\":");

        Query query = new Query(Dozent.class, "nachname == \"Mainz\""
                                    + " && vorname == \"Chaya\"");

        Dozent dozent = null;
        try {
            dozent = (Dozent) query.pick(uni.dozenten.values());
        } catch (NoSuchElementException ex) {
            // Kein Element gefunden. Nichts weiter zu tun.
            return;
        }

        Iterator j = dozent.vorlesungen.iterator();
        while (j.hasNext()) {
            Vorlesung vorlesung = (Vorlesung) j.next();
            System.out.println (vorlesung.name);
        }

        tr.commit();
        db.close();
    }
}
```

2.
```java
package objectstore;

import java.util.Enumeration;
import java.util.Iterator;
import java.util.Set;

import com.odi.*;
import com.odi.util.query.FreeVariableBindings;
import com.odi.util.query.FreeVariables;
import com.odi.util.query.Query;

public class Anfrage2 {
    public static void main(String[] args) {

        Session session = Session.create(null, null);
        session.join ();

        Database db = Database.open("uni.odb", ObjectStore.READONLY);
```

```
        Transaction tr = Transaction.begin(ObjectStore.READONLY);
        Uni uni = (Uni) db.getRoot("uni");

        System.out.println("Die Nachnamen aller Tutoren der beiden "
                           + "\"Informationssysteme\"-Veranstaltungen:");

        Query query = new Query(Vorlesung.class, "name ~~ \"*Informationssysteme*\"");

        Set result = query.select(uni.vorlesungen);

        Iterator i = result.iterator();
        while (i.hasNext()) {
            Vorlesung vorlesung = (Vorlesung) i.next();

            Iterator j = vorlesung.tutorien.iterator();
            while (j.hasNext()) {
                Person tutor = ((Tutorium) j.next()).tutor;
                System.out.println(tutor.nachname);
            }
        }

        tr.commit();
        db.close();
    }
}
```

3.
```
package objectstore;

import java.util.Enumeration;
import java.util.Iterator;
import java.util.Set;

import com.odi.*;
import com.odi.util.query.FreeVariableBindings;
import com.odi.util.query.FreeVariables;
import com.odi.util.query.Query;

public class Anfrage3 {
    public static void main(String[] args) {

        Session session = Session.create(null, null);
        session.join ();

        Database db = Database.open("uni.odb", ObjectStore.READONLY);
        Transaction tr = Transaction.begin(ObjectStore.READONLY);
        Uni uni = (Uni) db.getRoot("uni");

        System.out.println("Alle Dozenten der Abteilung Betriebssysteme:");

        Query query = new Query(Dozent.class, "abteilung.name == \"Betriebssysteme\"");

        Set result = query.select(uni.dozenten.values());

        Iterator i = result.iterator();
        while (i.hasNext()) {
            Dozent dozent = (Dozent) i.next();

            System.out.println(dozent.vorname + " " + dozent.nachname);
        }

        tr.commit();
        db.close();
    }
}
```

4.
```
package objectstore;
```

```
import java.util.Enumeration;
import java.util.Iterator;
import java.util.Set;

import com.odi.*;
import com.odi.util.query.FreeVariableBindings;
import com.odi.util.query.FreeVariables;
import com.odi.util.query.Query;

public class Anfrage4 {
    public static void main(String[] args) {

        Session session = Session.create(null, null);
        session.join ();

        Database db = Database.open("uni.odb", ObjectStore.READONLY);
        Transaction tr = Transaction.begin(ObjectStore.READONLY);
        Uni uni = (Uni) db.getRoot("uni");

        System.out.println("Alle Vorlesungen, die von einem Dozenten aus "
                           + "einer anderen Abteilung gehalten werden:");

            Iterator i = uni.dozenten.values().iterator();
            while (i.hasNext()) {
                Dozent dozent = (Dozent) i.next();

            Iterator j = dozent.vorlesungen.iterator();
                while (j.hasNext()) {
                Vorlesung vorlesung = (Vorlesung) j.next();

                if (vorlesung.abteilung != dozent.abteilung) {
                        System.out.println(vorlesung.name);
                }
                }

        }

        tr.commit();
        db.close();
    }
}
```

5.
```
package objectstore;

import java.util.Enumeration;
import java.util.Iterator;
import java.util.Set;
import java.util.HashSet;

import com.odi.*;
import com.odi.util.query.FreeVariableBindings;
import com.odi.util.query.FreeVariables;
import com.odi.util.query.Query;

public class Anfrage5 {
    public static void main(String[] args) {

        Session session = Session.create(null, null);
        session.join ();

        Database db = Database.open("uni.odb", ObjectStore.READONLY);
        Transaction tr = Transaction.begin(ObjectStore.READONLY);
        Uni uni = (Uni) db.getRoot("uni");

        System.out.println("Die Namen aller Tutoren:");
```

```
        // Um doppelte Einträge auszusortieren
        Set tutoren = new HashSet();

        Iterator i =  uni.vorlesungen.iterator();
        while (i.hasNext()) {

            Iterator j = ((Vorlesung) i.next()).tutorien.iterator();
            while (j.hasNext()) {
                Person tutor = (Person) ((Tutorium) j.next()).tutor;
                tutoren.add(tutor);
            }
        }

        Iterator k = tutoren.iterator();
        while (k.hasNext()) {
            Person tutor = (Person) k.next();
            System.out.println (tutor.vorname + " " + tutor.nachname);
        }

        tr.commit();
        db.close();
    }
}
```

6.
```
package objectstore;

import java.util.Enumeration;
import java.util.Iterator;
import java.util.Set;

import com.odi.*;
import com.odi.util.query.FreeVariableBindings;
import com.odi.util.query.FreeVariables;
import com.odi.util.query.Query;

public class Anfrage6 {
    public static void main(String[] args) {

        Session session = Session.create(null, null);
        session.join ();

        Database db = Database.open("uni.odb", ObjectStore.READONLY);
        Transaction tr = Transaction.begin(ObjectStore.READONLY);
        Uni uni = (Uni) db.getRoot("uni");

        System.out.println("Alle studentischen Tutoren, "
                           + "die für William Gouchie arbeiten:");

        Query query = new Query(Dozent.class, "nachname == \"Gouchie\""
                                + " && vorname == \"Willian\"");

        Dozent dozent = null;
        try {
            dozent = (Dozent) query.pick(uni.dozenten.values());
        } catch (NoSuchElementException ex) {
            // Kein Element gefunden. Nichts weiter zu tun.
            return;
        }

        Iterator i = dozent.vorlesungen.iterator();
        while (i.hasNext()) {
            Vorlesung vorlesung = (Vorlesung) i.next();

            Iterator j = vorlesung.tutorien.iterator();
            while (j.hasNext()) {
```

```
                    Person tutor = (Person) ((Tutorium) j.next()).tutor;

                    if (tutor instanceof Student) {
                        System.out.println (tutor.vorname + " " + tutor.nachname);
                    }
                }
            }

            tr.commit();
            db.close();
        }
    }
```

7.
```
    package objectstore;

    import java.util.Enumeration;
    import java.util.Iterator;
    import java.util.Set;

    import com.odi.*;
    import com.odi.util.query.FreeVariableBindings;
    import com.odi.util.query.FreeVariables;
    import com.odi.util.query.Query;

    public class Anfrage7 {
        public static void main(String[] args) {

            Session session = Session.create(null, null);
            session.join ();

            Database db = Database.open("uni.odb", ObjectStore.READONLY);
            Transaction tr = Transaction.begin(ObjectStore.READONLY);
            Uni uni = (Uni) db.getRoot("uni");

            System.out.println("Alle Veranstaltungen, in denen auch der Dozent "
                               + "Tutor ist:");

                Iterator i = uni.dozenten.values().iterator();
                while (i.hasNext()) {
                    Dozent dozent = (Dozent) i.next();

                Iterator j = dozent.vorlesungen.iterator();
                    while (j.hasNext()) {
                    Vorlesung vorlesung = (Vorlesung) j.next();

                    Iterator k = vorlesung.tutorien.iterator();
                        while (k.hasNext()) {
                        Person tutor = ((Tutorium) k.next()).tutor;

                        if (tutor == dozent) {
                                  System.out.println(vorlesung.name);
                                        break;
                                  }
                         }
                    }
            }

            tr.commit();
            db.close();
        }
    }
```

8.
```
    package objectstore;

    import java.util.Enumeration;
    import java.util.Iterator;
```

```
import java.util.Set;

import com.odi.*;
import com.odi.util.query.FreeVariableBindings;
import com.odi.util.query.FreeVariables;
import com.odi.util.query.Query;

public class Anfrage8 {
    public static void main(String[] args) {

        Session session = Session.create(null, null);
        session.join ();

        Database db = Database.open("uni.odb", ObjectStore.READONLY);
        Transaction tr = Transaction.begin(ObjectStore.READONLY);
        Uni uni = (Uni) db.getRoot("uni");

        System.out.println("Alle Studenten, die keine Veranstaltungen besuchen:");

        Query query = new Query(Student.class, "belegungen.isEmpty()");

        Set studenten = query.select(uni.studenten.values());

        Iterator i = studenten.iterator();
        while (i.hasNext()) {
            Student student = (Student) i.next();
            System.out.println (student.vorname + " " + student.nachname);
        }

        tr.commit();
        db.close();
    }
}
```

9.

```
package objectstore;

import java.util.Enumeration;
import java.util.Iterator;
import java.util.Set;

import com.odi.*;
import com.odi.util.query.FreeVariableBindings;
import com.odi.util.query.FreeVariables;
import com.odi.util.query.Query;

public class Anfrage9 {
    public static void main(String[] args) {

        Session session = Session.create(null, null);
        session.join ();

        Database db = Database.open("uni.odb", ObjectStore.READONLY);
        Transaction tr = Transaction.begin(ObjectStore.READONLY);
        Uni uni = (Uni) db.getRoot("uni");

        System.out.println("Alle Studenten, die die Veranstaltung "
                           + "\"Kryptographie\" besuchen:");

            Iterator i = uni.studenten.values().iterator();
            while (i.hasNext()) {
            Student student = (Student) i.next();

            Iterator j = student.belegungen.iterator();
               while (j.hasNext()) {
                Belegung belegung = (Belegung) j.next();
```

```
                if (belegung.vorlesung.name.equals("Kryptographie")) {
                        System.out.println(student.vorname + " "
                                + student.nachname);
                    break;
                }
                  }
            }

            tr.commit();
            db.close();
        }
    }
```

10.
```
    package objectstore;

    import java.util.Enumeration;
    import java.util.Iterator;
    import java.util.Set;

    import com.odi.*;
    import com.odi.util.query.FreeVariableBindings;
    import com.odi.util.query.FreeVariables;
    import com.odi.util.query.Query;

    public class Anfrage10 {
        public static void main(String[] args) {

            Session session = Session.create(null, null);
            session.join ();

            Database db = Database.open("uni.odb", ObjectStore.READONLY);
            Transaction tr = Transaction.begin(ObjectStore.READONLY);
            Uni uni = (Uni) db.getRoot("uni");

            System.out.println("Den Namen des Dozenten der Vorlesung "
                               + "\"Betriebsysteme I\":");

                Iterator i = uni.dozenten.values().iterator();
                while (i.hasNext()) {
                Dozent dozent = (Dozent) i.next();

                Iterator j = dozent.vorlesungen.iterator();
                   while (j.hasNext()) {
                    Vorlesung vorlesung = (Vorlesung) j.next();

                    if (vorlesung.name.equals("Betriebssysteme I")) {
                            System.out.println(dozent.vorname + " "
                                    + dozent.nachname);
                        break;
                    }
                      }
            }

            tr.commit();
            db.close();
        }
    }
```