

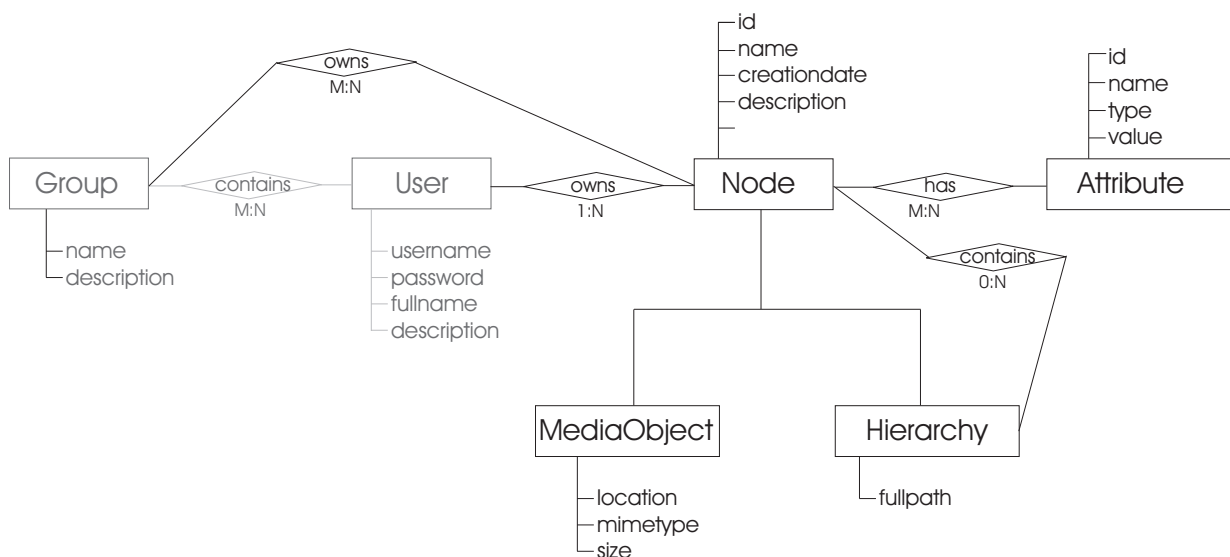
Datenbankpraktikum

Sommersemester 2008

3. Übungsblatt

Aufgabe 1: Medienarchiv

Das folgende EER-Diagramm modelliert die Entitäten und Relationen des Medienarchives.



Das Medienarchiv verwaltet Medienobjekte und deren Metadaten (Daten über Daten). Die Metadaten werden in einer relationalen Datenbank gespeichert, die Binärdaten selber werden im Filesystem abgelegt.

Die gemeinsamen Attribute der Medienobjekte und Hierarchien (*id, name, description, etc.*) werden in der Entität **Node** modelliert, von der die Entitäten **MediaObject** und **Hierarchy** erben. Medienobjekte sind Dateien (Textdokumente, Bilder, Dias, Videos etc.), die also zusätzlich die Dateiattribute *Size, MIME-Type* und *Location* besitzen.

Zum Navigieren werden Medienobjekte in einer hierarchischen Struktur angeordnet (ähnlich einem Dateisystem). Hierarchien enthalten Nodes (andere Hierarchien und Medienobjekte), wobei jedes Node nur in genau einer Hierarchie enthalten ist.

Für die Nodes können dynamisch weitere **Attribute** definiert werden, die der symbolischen Beschreibung dienen. Jedem Attribut ist ein bestimmter Typ zugeordnet, der dessen Datentyp charakterisiert.

Im Medienarchiv gibt es unterschiedliche Benutzer und Gruppen, die mit den Medienobjekten und Hierarchien in Beziehung stehen. Die Benutzer und Gruppen wurden im 2. Aufgabenblatt bereits modelliert. Eine GUI zur Verwaltung der Benutzer und Gruppen wurde im 3. Aufgabenblatt erstellt. Die Zugriffsrechte der Benutzer und Gruppen auf Medienobjekte und Hierarchien sollen wie folgt organisiert sein:

- Schreiben: Jeder Benutzer darf Medienobjekte und Verzeichnisse erstellen. Der Benutzer wird als Autor in die *UserOwnsNode* Tabelle eingetragen.
- Lesen: Medienobjekte/Verzeichnisse sind sichtbar für einen Benutzer, wenn der Benutzer einer Gruppe angehört, die in der Relation *GroupOwnsNode* für das jeweilige Medienobjekt eingetragen ist.
- Löschen: Medienobjekte dürfen nur vom Autor und Benutzern der *Administrator* Gruppe gelöscht werden.
- Erweitern Sie das Interface `common.UserIF` vom letzten Aufgabenblatt um die beiden Methoden `String getPassword()` und `void setPassword(String arg)`, um das Passwort für das User-Objekt zu lesen und zu setzen. Passen Sie Ihre Implementierung der Klassen *User* und *DataManager* an. Benutzer sollen nur Zugriff zum Medienarchiv erhalten, wenn das Passwort korrekt auf einer Login Seite eingegeben wurde.
- Definieren Sie zu dem ER-Diagramm geeignete Relationen (Tabellen). Wählen Sie geeignete Spaltentypen. Welche Fremdschlüsselbeziehungen bestehen zwischen den Tabellen? Geben Sie die **SQL-Create-Statements** an. Speichern Sie diese wie auf dem vorherigen Aufgabenblatt im Pfad `~/dbp/texte/blatt3/`.

Aufgabe 2: Java-Modellierung

Folgende Java-Interfaces repräsentieren die o.a. Entitäten:

```
1 package common;
2
3 public interface NodeIF {
4     public java.util.List getAttributes();
5     public java.util.Date getCreationDate();
6     public String getDescription();
7     public GroupIF getGroup();
8     public int getId();
9     public java.util.Date getLastModified();
10    public String getName();
11    public UserIF getOwner();
12    public HierarchyIF getParent();
13    public boolean isHierarchy();
14    public void setAttributes(java.util.List arg);
15    public void setCreationDate(java.util.Date arg);
16    public void setDescription(String arg);
17    public void setGroup(GroupIF arg);
18    public void setId(int arg);
19    public void setLastModified(java.util.Date arg);
20    public void setName(String arg);
21    public void setOwner(UserIF arg);
22    public void setParent(HierarchyIF arg);
23 }
```

```
1 package common;
2
3 public interface MediaObjectIF extends NodeIF {
4
5     public String getLocation();
6     public String getMimeType();
7     public int getSize();
8     public void setLocation(String arg);
9     public void setMimeType(String arg);
10    public void setSize(int arg);
11 }
```

```

1 package common;
2
3 public interface HierarchyIF extends NodeIF {
4
5     public java.util.List getChildren();
6     public String getFullPath();
7     public void setFullPath(String arg);
8     public void setChildren(java.util.List arg);
9 }
10
11 package common;
12
13 public interface AttributeIF {
14     public int getId();
15     public String getName();
16     public NodeIF getNode();
17     public String getType();
18     public String getValue();
19     public void setId(int newId);
20     public void setName(String arg);
21     public void setNode(NodeIF arg);
22     public void setType(String arg);
23     public void setValue(String arg);
24 }

```

Erstellen Sie analog zum vorherigen Übungszettel die Klassen `Node`, `MediaObject`, `Hierarchy` und `Attribute`, die die Schnittstellen implementieren. Organisieren Sie Ihre Klassen im Package `model`.

Aufgabe 3: Datamanager

Erweitern Sie Ihre Klasse `DataManager` um folgende Funktionalitäten:

- die benötigten Tabellen erzeugen und löschen,
- Mit Hilfe einer Methode `public void insertHierarchy(common.HierarchyIF arg)` eine Hierarchie in die Tabelle `Hierarchy` hinzufügen
- Mit Hilfe der Methode `public void deleteHierarchy(common.HierarchyIF arg)` eine Hierarchie aus der Datenbank löschen.
- Mit Hilfe einer Methode `public void insertMediaObject(common.MediaObjectIF arg)` ein Medienobjekt in die Tabelle `MediaObject` hinzufügen.

- Mit Hilfe der Methode `public void deleteMediaObject(common.MediaObjectIF arg)` ein Medienobjekt aus der Datenbank löschen.
- Mit Hilfe der Methode `public common.NodeIF findNodeByID(int arg)` das Node-Objekt mit dem gegebenen Schlüsselwert aus der Datenbank lesen.
- Mit Hilfe der Methode `public common.NodeIF[] findNodesByParent(NodeIF parent)` die Liste der Nodes lesen, die in der gegebenen Hierarchie enthalten sind.
- Mit Hilfe der Methode `public common.HierarchyIF findHierarchyByFullPath(String path)` die Hierarchie mit dem gegebenen Pfad aus der Datenbank lesen.
- Die Attribute sind abhängige Objekte: die gehören zu einem bestimmten Node. Daher werden sie zusammen mit dem Objekt, zu dem sie gehören, manipuliert. Das bedeutet: wenn eine Hierarchie oder ein Medienobjekt in die Datenbank eingefügt wird, werden die zugehörigen Attribute auch eingefügt. Dasselbe geschieht beim Löschen oder beim Lesen der Nodes aus der Datenbank. Beachten Sie, dass Sie diese Aufgabe mit Transaktionen lösen müssen.

Abgabetermin: Mittwoch, der 07.05.2008