marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

Benutzerverwaltung - Modul Data-Tier

CAS Enterprise Application Development mit Java EE

Abstrakt	Dieses Dokument dient als Bericht zur Realisierung einer Test-Applikation nach dem Java Persistence API – Ansatzes im Modul Data-Tier.
Autor	Marc Bouquet
Version	siehe Änderungskontrolle
Status	Freigegeben
Ausgabedatum	18.06.2007
Gültigkeitsbereich	SWS
Gültig ab	21.05.2007
Gültig bis	18.06.2007
Dokumentname	eadj-jpa-bericht.doc
Ablage	C:\work\eadj-jpa-bericht.doc

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

Änderungskontrolle

Änderung

Version	Datum	Ausführende Stelle	Bemerkung / Art der Änderung
0.1	24.05.2007	Marc Bouquet	Erstellung des Dokumentes

Prüfung

Version	Prüfdatum	Prüfende Stelle(n)	Bemerkung
1.0	18.06.2007	Marc Bouquet	

Freigabe

Version	Freigabedatum	Freigebende Stelle(n)	Bemerkung
1.0	18.06.2007	Marc Bouquet	

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

Inhaltsverzeichnis

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

Abbildungsverzeichnis

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

Tabellenverzeichnis

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

1. Einleitung

1.1. Ziel und Zweck

Dieses Dokument wurde im Rahmen eines Projektes im Modul „Data-Tier“ am Lehrgang „CAS Enterprise Application Development mit Java EE“ an der Software-Schule Schweiz erstellt.

In diesem Projekt wurde gemäss der Aufgabenstellung (siehe Punkt 2ff) der Java Persistence API – Ansatz an einem selbst gewählten Thema angewendet. In diesem Dokument wird das gewählte Thema vorgestellt sowie den Lösungsansatz mit dem JPA erläutert.

1.2. Verwendung

Dieses Dokument ist ein Teil der Projektarbeit die zur Bewertung des Moduls „Data-Tier“ beiträgt.

1.3. Einschränkungen und Voraussetzungen

Es wird erwartet, dass der Leser über Grundkenntnisse von XML, SQL, Datenbanken und Java verfügt.

1.4. Lernziel des Moduls

Das Lernziel dieses Modul ist vertiefte Kenntnisse über die Aufgaben und Dienste des „Data-Tier“ in unternehmensweiten Applikationen zu erlangen.

Mein persönliches Ziel ist den Java Persistence API – Ansatz zu verstehen, damit ich diesen gegenüber anderen Techniken/Varianten bewerten und vergleichen kann.

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

2. Aufgabenstellung

Bearbeiten Sie anhand eines Ausschnittes aus dem Datenmodell eines Projektes ihrer Firma, welcher über mittlere Komplexität verfügt, folgende Aufgaben:

- Erstellen von Entity-Klassen mit:
  a. OneToMany Beziehung / ManyToManyBeziehung
  b. OneToOne Beziehung
  c. Vererbungsbeziehung
  d. Feature aus dem Spezialitätenkapitel IV (Transaktionen, Serialisierbarkeit, Geschwindigkeitsverbesserung, etc.)
- Testprogramm erstellen für das Einfügen, Ändern, Lesen und Löschen von Daten.
- Implementieren Sie das JPA Optimistic Locking
  - Identifizieren in Ihrer Programm die Entities, auf welche nebenläufig zugegriffen wird. Legen Sie Ihre Entscheidungen dar (weshalb, weshalb nicht)
  - Legen Sie das Design Ihres Wiederaufsetzens der Transaktion dar
- Testen Sie das Laden, Lesen, Ändern, Löschen der Daten
  - Benchmark: Hibernate Einstellung für REPEATABLE READ; Zeitmessung für einen von Ihnen zu bestimmenden Testdatensätzen
  - Verwendung Ihres Programms im Read Committed Modus; Zeitmessung mit dem gleichen Testdatensätzen
- ZusatzProzente: Testen Sie Ihr Programm ohne Ihr Sperren; Zeitmessung mit den gleichen Datensätzen. Haben Sie nach dem Durchlauf eine andere Menge von Daten in den DB Tabellen oder sogar andere Ergebnisse?
- Einbau von Queries in das Testprogramm.

2.1. Organisation

Gruppenarbeit mit 1-2 Teilnehmern. Meldungen der Gruppen und des Themas bis 25.05.07.
Abgabe des Source Codes und eines schriftlichen Berichtes, der das gesamte dokumentiert: was genau implementiert wurde, bzw. die Begründung weshalb es so implementiert wurde (in Fällen mit mehreren Möglichkeiten). Dokumentation der Korrektheit des Programms im Parallelbetrieb ist! Beschreiben Sie die Schwierigkeiten in der Realisierung des ORMs Ihres (sinnvollen) Ausschnitts und bewerten Sie die Praktikabilität des JPA-Ansatzes im Gegensatz zur vorhandenen Firmenlösung.

2.2. Bewertung der Aufgabe

Das Modul gilt als erfüllt, wenn 80% der Aufgabe erfüllt ist. Das ist:

- Korrektes Programm
- Testprogramm, das die Korrektheit des Programms darstellt
- Schriftlicher Bericht (lesbar, konsistent)

Die restlichen 20% werden für die Aufmachung, Lesbarkeit, Darstellung des Problems und der Lösung bzw. Eigenständigkeit und Ideenreichtum vergeben.

2.3. Abgabe der Aufgabe

Source und Bericht elektronisch an christian.kanele@bfh.ch.
Abgabetermin: 18. Juni 2007 um 24:00

3. Benutzerverwaltung – eine Erweiterung von Swissreg

3.1. Was ist Swissreg

Als E-Government-Pionier bietet das Institut für Geistiges Eigentum mit Swissreg seit 2OO1 allen Interessierten gratis ein Online-Register für Erstinformationen an. Im Juli 2OO6 konnte nun die vollständig überarbeitete Version aufgeschaltet werden. Mittelfristig will das Institut Swissreg so ausbauen, dass es die Papierpublikationen «mod.dép.», «+pat+» und «SHAB» ablösen kann.

Die kostenlose und mehrsprachige Datenbank Swissreg liefert Erstinformationen zu geschützten Titeln aus den Schutzrechtsbereichen Patente, Marken und Designs und wird heute von Juristen, Unternehmern, Medienschaffenden, Entwicklern und Kreativen als Arbeitsinstrument rege genutzt. Nach fünf Jahren ist Swissreg komplett überarbeitet worden: Die neue Version bietet eine übersichtlichere Navigation, zahlreiche zusätzliche Such- und Servicefunktionen, ein benutzerfreundlicheres Design und eine wesentlich schnellere Datenaufbereitung.

Mehr Service
Neu stehen für alle Schutzrechtsbereiche drei Suchmasken zur Auswahl: Neben der einfachen Registersuche und der erweiterten Suche wurde mit der «Publikationssuche» eine Möglichkeit geschaffen, schnell und gezielt nach formellen Publikationen zu suchen. Ein weiterer Vorteil: Die Detailansicht der Titel entspricht dem Registerauszug und zeigt die gesamte Historie an. Weitere Nachforschungen beim Institut oder die Bestellung eines Registerauszugs werden somit in vielen Fällen überflüssig – rechtlich verbindlich sind allerdings zurzeit noch ausschliesslich vom Institut ausgestellte Registerauszüge und die offiziellen Publikationsorgane shab.ch (Marken), +pat+ (Patente) und mod.dép. (Designs). Der detaillierte Hilfetext bietet Unterstützung beim Recherchieren und zahlreiche nützliche Hinweise. Ebenfalls verbessert worden ist die Qualität der Abbildungen: Vom Thumbnail in der Trefferliste über die druckoptimierte Darstellung in der Detailansicht bis zur maximalen Bildqualität kann der Benutzer die gewünschte Qualität wählen.
Swissreg ist ein ideales Instrument für Erstinformationen, aber kein professionelles Suchwerkzeug. Für vollständige und verlässliche Entscheidungsgrundlagen (beispielsweise zum Stand der Technik) empfehlen sich deshalb professionell erstellte Recherchen des Instituts oder privater Anbieter.

Marken
Der Markenbereich verfügt über 200 000 Schutztitel: Neben allen eingetragenen Marken sind neu auch hängige Gesuche (in der Regel innerhalb von 2 Arbeitstagen nach Gesuchseingang) sowie gelöschte Gesuche und gelöschte Marken (alle diejenigen, welche ab Juli 2006 gelöscht worden sind) enthalten. Die Eintragungen sowie Änderungen an Gesuchen und eingetragenen Marken werden in der Datenbank sofort sichtbar, also noch vor der formellen Publikation. Swissreg 2.0 erlaubt eine gezielte Suche: zum Beispiel auch nach Markentypen (Wortmarke, Marke mit Bildelementen, Farbmarke, akustische Marke usw.) und nach verbalen Elementen aller Markentypen (bisher nur bei Wortmarken möglich). Zu eingetragenen Marken werden in der Detailansicht neu auch Informationen über den Stand allfälliger Widerspruchsverfahren angezeigt, was eine Abschätzung des Beginns der fünfjährigen Karenzfrist erlaubt.

Patente
Die Patent-Suche ermöglicht die Durchführung von Online-Recherchen zu erteil- ten Patenten, die Wirkung für die Schweiz und Liechtenstein haben und in den Registern des Instituts eingetragen sind: Darunter fallen alle in Kraft stehende und gelöschte schweizerische Patente (CH-Patente) ab Anmeldedatum Januar 1978 (teilweise sogar ältere Patente) und Europäische Patente (EP-Patente) ab der Nummer 00 000 001.

Mit der Patentnummer oder der Patentgesuchsnummer kann in der «Einfachen Suche» nach dem aktuellen Stand oder auch die Historie zu einem bestimmten Patent gesucht werden: Um beispielsweise den Stand der Jahresgebührenzahlungen zu erfahren, genügt ein Klick auf die Patentnummer in der Trefferliste. Dadurch gelangt man zur Detailansicht mit allen registerpflichtigen Daten. In der «Erweiterten Suche» lassen sich auch mit anderen Suchkriterien, z.B. dem Inhaber- oder Erfindernamen oder dem Titel, Informationen finden. Sollte die Trefferliste zu lang sein, kann die Suche weiter eingeschränkt werden. Auch wenn ein Kunde nur über sehr spezifische Daten verfügt, beispielsweise über Erteilungs- oder Löschungsdaten, bietet die «Erweiterte Suche» viele Möglichkeiten, ein Patent und dessen Stand ausfindig zu machen. Bei jedem Patent in der Trefferliste und in der Detailansicht gibt es einen Link auf die Esp@cenet-Datenbank des Europäischen Patentamtes. Dort finden sich die entsprechenden Patentschriften mit den technischen Angaben der Erfindungen sowie zusätzliche Informationen, die über das Schutzgebiet der Schweiz und Liechtensteins hinausgehen und nicht vom Institut verwaltet werden.

Service Public

Als besondere Dienstleistung stellt das Institut allen Interessierten ab Ende 2006 sämtliche Markendaten von Swissreg kostenlos in elektronischer Form zur Verfügung. Dies ermöglicht beispielsweise Datenveredlern und privaten Rechercheanbietern den Aufbau einer eigenen Datenbank oder kann der Auswertung zu wissenschaftlichen Zwecken dienen. Ganz im Sinn des Service Public wird das Institut Swissreg ausserdem laufend optimieren; mittelfristig soll Swissreg die Papierpublikationen «mod.dép.», «+pat+» und «SHAB» vollständig ablösen.

Quelle: Jahresbericht 2005/2006 des Instituts für Geistiges Eigentum

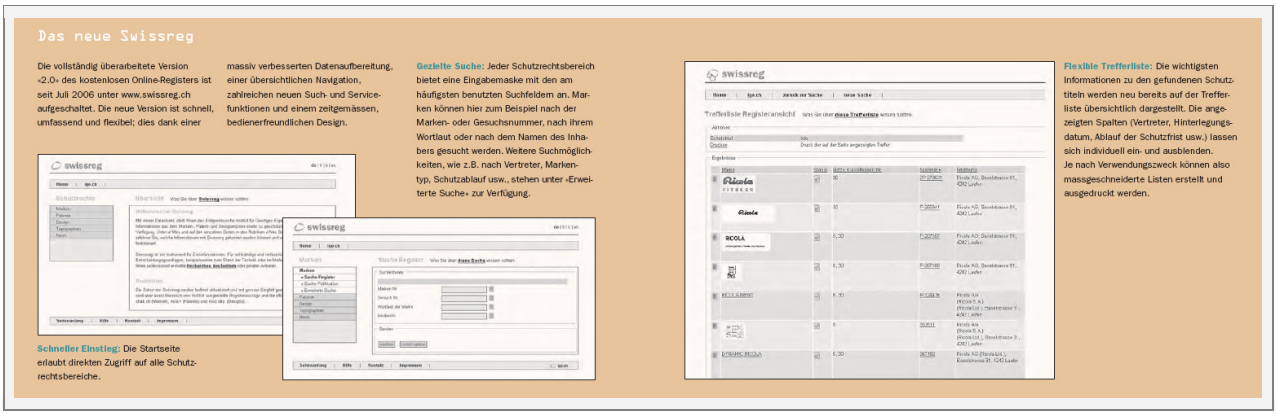

Abbildung 1: Das neue Swissreg

3.2. Projektarbeit

Im Rahmen der Weiterentwicklung von Swissreg 2.0 werden wir in den nächsten Tagen ein weiteres Modul „mypage“ aufschalten. Mit diesem Modul, bieten wir unseren Kunden, die Möglichkeit, sich bei Swissreg 2.0 zu registrieren und die Suchabfragen auf unserem Server zu speichern. Diese Daten können danach als Überwachungen definiert werden. Die definierte Überwachungen des Kunden werden aufgrund des ausgewählten Zyklus (täglich, wöchentlich monatlich), mit einem Batch-Job verarbeitet. Dieser Batch-Job generiert eine PDF- oder XML-Datei, die per Email an den Kunden gesendet wird.

Das Eidgenössische Institut für Geistiges Eigentum wird ab Mitte 2008, alle Eintragungen in das Register nur noch elektronisch publizieren. Heute werden alle Eintragungen in unterschiedlichen Zeitungen veröffentlicht. Sobald die sogenannte ePublikation eingeführt ist, wird das Modul „mypage“ eine zentrale Rolle im Swissreg 2.0 erhalten. Unsere Kunden können sich diese ePublikationen mit einer Überwachung zustellen lassen, quasi die heutigen Zeitschriften in Form eines Emails. Somit wird die heutige Veröffentlichung einer Eintragung bequem ins Büro oder nach Hause gesendet.

Für meine Projektarbeit, habe ich mich entschieden, den Ausschnitt der Benutzerverwaltung nachzubauen. Die Benutzerverwaltung besteht aus zwei Bereichen. Die Authentisierung wurde allgemein gehalten, da dieses Modul in weiteren Web-Applikationen eingesetzt werden soll. Sowie die Verwaltung, die je nach Web-Applikation unterschiedlich sein kann. Das Verwaltungsmodul wird daher in der jeweiligen Web-Applikation implementiert. Das neue Modul wurde nach dem altbekannten Data Access Object – Pattern implementiert und bietet daher den Idealfall für den Vergleich dieser beiden Ansätze.

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

4. Entity-Relationship-Diagram (ERD)

4.1. Beschreibung

Dieses Diagramm zeigt die Tabellen, die für das Modul „Benutzerverwaltung“ benötigt werden. Es wurden zwei Bereiche definiert:

- **Authentisierung (rosa)**
  Für die Tabellen, die für die Authentisierung nötig sind.

- **Verwaltung (gelb)**
  Für die Tabellen, die ausschliesslich dem Module „Verwaltung“ zur Verfügung stehen.

Die Primärschlüssel wurden mit PK und die Fremdschlüssel wurden mit FK gekennzeichnet.

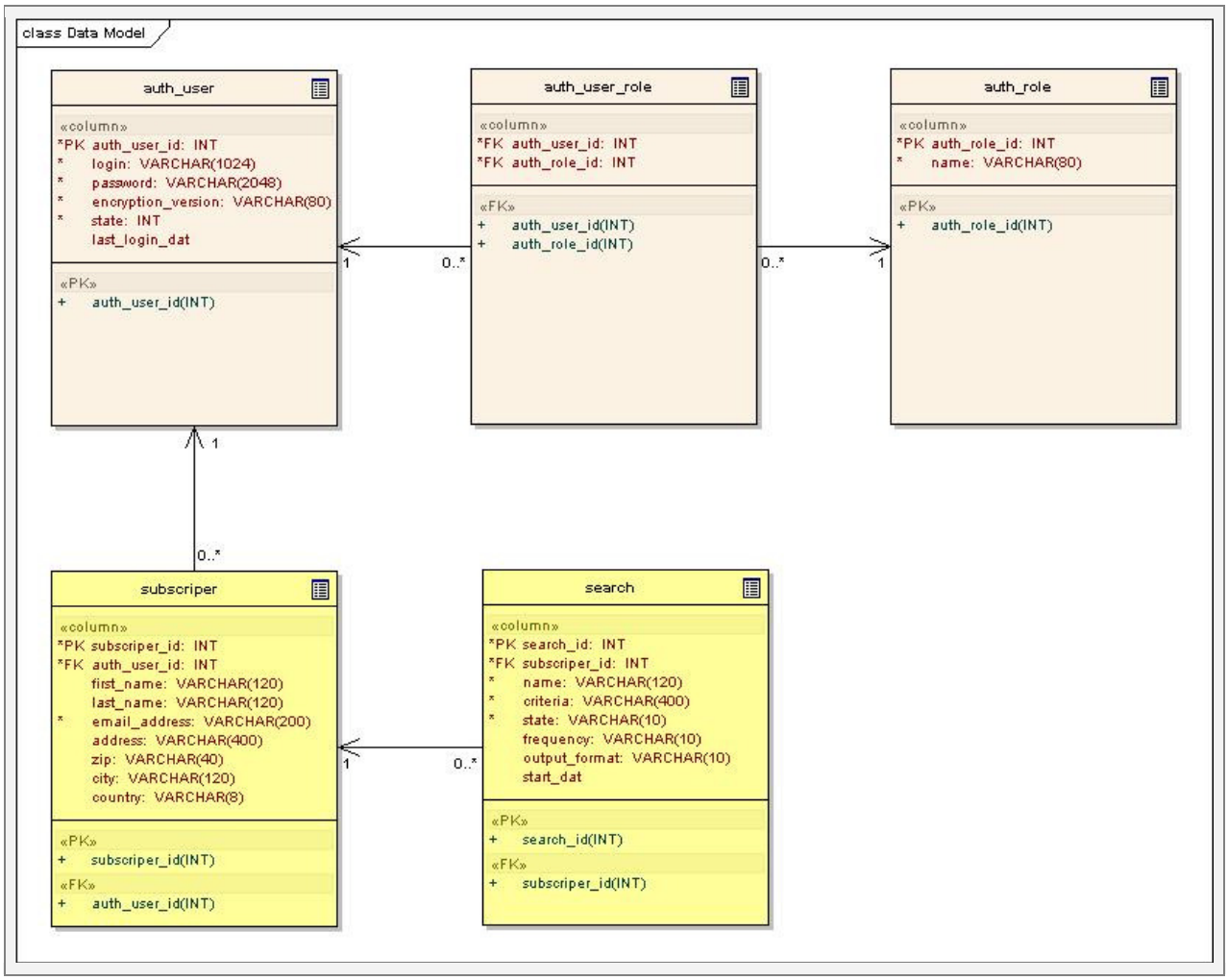

Abbildung 2: ERD der Benutzerverwaltung

4.2. Tabelle AUTH_USER

Diese Tabelle enthält die notwendigen Informationen, um einen Benutzer zu beglaubigen.

Spalte	Datentyp	Null?	Beschreibung
auth_user_id	INT	not null	Der Primärschlüssel.
login	VARCHAR(200)	not null	Der Anmeldename.
password	VARCHAR(2048)	not null	Das verschlüsselte Passwort des Benutzers.
encryption_version	VARCHAR(80)	not null	Die Verschlüsselungsversion.
state	INT	not null	Der Status des Kontos: -1 gesperrt 0 deaktiviert 1 aktiviert
last_login	TIMESTAMP	null	Das Datum der letzten Anmeldung des Benutzers.

Tabelle 1: Beschreibung Tabelle AUTH_USER

4.2.1. Skript für die Tabelle AUTH_USER

```
-- ----------------------------------------------------------------------
-- create table auth_user
-- ----------------------------------------------------------------------
create table auth_user (
  auth_user_id                          int unsigned not null auto_increment,
  login                                 varchar(1024)       not null,
  password                              varchar(2048)       not null,
  encryption_version                    varchar(80)         not null,
  state                                 int unsigned        not null,
  last_login_dat                        timestamp           null,

  primary key  (auth_user_id)
) engine=InnoDB default charset=latin1;
```

Tabelle 2: Erzeugung der Tabelle AUTH_USER

4.3. Tabelle AUTH_ROLE

Diese Tabelle enthält die möglichen Rollen.

Spalte	Datentyp	Null?	Beschreibung
auth_role_id	INT	not null	Der Primärschlüssel.
name	VARCHAR(80)	not null	Der Name einer Rolle. **Bemerkung:** Bsp: *user, guest, …*

Tabelle 3: Beschreibung Tabelle AUTH_ROLE

4.3.1. Skript für die Tabelle AUTH_ROLE

```
-- ------------------------------------------------------------------------
-- create table auth_role
-- ------------------------------------------------------------------------
create table auth_role (
  auth_role_id                          int unsigned not null auto_increment,
  name                                  varchar(80)  not null,

  primary key  (auth_role_id)
) engine=InnoDB default charset=latin1
```

Tabelle 4: Erzeugung der Tabelle AUTH_ROLE

4.4. Tabelle AUTH_USER_ROLE

Diese Tabelle enthält die Rollen, die einem bestimmten Benutzer zugewiesen werden.

Spalte	Datentyp	Null?	Beschreibung
auth_user_id	INT	not null	Der Benutzer.
auth_role_id	INT	not null	Die Rolle eines Benutzers.

Tabelle 5: Beschreibung Tabelle AUTH_USER_ROLE

4.4.1. Skript für die Tabelle AUTH_USER_ROLE

```
-- ------------------------------------------------------------------------
-- create relationship table auth_user_role
-- note:
--  (1) a auth_role may have * auth_user entries but a auth_user
--      entry belongs to one auth_role.
--  (2) to enforce this constraint, a uniqe index is defined for the
--      auth_user_id row
-- ------------------------------------------------------------------------
create table auth_user_role (
  auth_user_id                          int unsigned not null,
  auth_role_id                          int unsigned not null,

  unique(auth_user_id),
  FOREIGN KEY (auth_user_id) REFERENCES auth_user(auth_user_id) ON DELETE CASCADE,
  FOREIGN KEY (auth_role_id) REFERENCES auth_role(auth_role_id) ON DELETE CASCADE
) engine=InnoDB default charset=latin1;
```

Tabelle 6: Erzeugung der Tabelle AUTH_USER_ROLE

4.5. Tabelle SUBSCRIBER

Diese Tabelle enthält die Informationen eines registrierten Benutzers.

Spalte	Datentyp	Null?	Beschreibung
subscriper_id	INT	not null	Der Primärschlüssel.
auth_user_id	INT	not null	Der Fremdschlüssel des Benutzers, der dem Authentisierung Modul gehört.
first_name	VARCHAR(120)	null	Der Vorname.
last_name	VARCHAR(120)	null	Der Nachname.
email_address	VARCHAR(200)	not null	Die Email-Adresse. Sie darf nicht null sein, da E-Mails gesendet werden. Bsp: Ergebnisse von den ausgeführten Überwachungen). Im Moment ist die E-Mail-Adresse auch der Anmeldename.
address	VARCHAR(400)	null	Die Adresse (Strassenname und Strassennummer).
zip	VARCHAR(40)	null	Die Postleitzahl.
city	VARCHAR(120)	null	Der Name einer Stadt.
country	VARCHAR(8)	null	Das Land als ISO-Code.

Tabelle 7: Beschreibung Tabelle SUBSCRIPER

4.5.1. Skript für die Tabelle SUBSCRIPER

```
-- ------------------------------------------------------------------------
-- create table subscriper
-- ------------------------------------------------------------------------
create table subscriper (
  subscriper_id                         int unsigned not null auto_increment,
  auth_user_id                          int unsigned not null,
  first_name                            varchar(120) null,
  last_name                             varchar(120) null,
  email_address                         varchar(200) not null,
  address                               varchar(400) null,
  zip                                   varchar(40)  null,
  city                                  varchar(120) null,
  country                               varchar(8)   null,

  primary key  (subscriper_id),
  FOREIGN KEY (auth_user_id) REFERENCES auth_user(auth_user_id) ON DELETE CASCADE
) engine=InnoDB default charset=latin1;
```

Tabelle 8: Erzeugung der Tabelle SUBSCRIPER

4.6. Tabelle SEARCH

Diese Tabelle enthält alle Suchen, die von einem Benutzer gespeichert werden.

Spalte	Datentyp	Null?	Beschreibung
search_id	INT	not null	Der Primärschlüssel.
subscriper_id	INT	not null	Der Fremdschlüssel des Benutzers, der die Suche besitzt.
name	VARCHAR(120)	not null	Der Name der Suche.
criteria	VARCHAR(400)	not null	Die Suchkriterien
state	VARCHAR(10)	not null	Der Zustand der Suche. ACTIVE: Die Suche wird automatisch ausgeführt. INACTIV: Die Suche kann manuell ausgeführt werden.
frequency	VARCHAR(10)	null	Die Frequenz der Durchführung dieser Suche. Sie kann sein: TÄGLICH, WÖCHENTLICH oder MONATLICH.
output_format	VARCHAR(10)	null	Das Format der Ausgabe, wenn diese Suche automatisch durchgeführt wird. Die können für die Ausgabeformaten: PDF, XML, EXCEL definiert werden.
start_date	TIMESTAMP	null	Das Start Datum für die automatische Suche.

Tabelle 9: Beschreibung Tabelle SEARCH

4.6.1. Skript für die Tabelle SEARCH

```
-- ------------------------------------------------------------------------
-- create table search
-- ------------------------------------------------------------------------
create table search (
  search_id                          int unsigned not null auto_increment,
  subscriper_id                      int unsigned not null,
  name                               varchar(120) not null,
  criteria                           varchar(400) not null,
  state                              varchar(10)  not null,
  frequency                          varchar(10)  null,
  output_format                      varchar(10)  null,
  start_dat                          timestamp    null,

  primary key  (search_id),
  FOREIGN KEY (subscriper_id) REFERENCES subscriper(subscriper_id) ON DELETE CASCADE
) engine=InnoDB default charset=latin1;
```

Tabelle 10: Erzeugung der Tabelle SEARCH

5. Test-Applikation

5.1. Vorbedingungen

Damit die Test-Applikation gestartet werden kann, muss gemäss dem Punkt 7ff eine MySQL-Datenbank vorhanden sein.

Das Datenbank-Schema und die Tabellen können mit dem mitgelieferten SQL-Skript oder gemäss den Angaben unter dem Punkt 4.2 – 4.6 erstellt werden.

5.2. Ablauf der Test-Applikation – Variante Sequenziell

1. Die Test-Applikation wird mit der main()-Methode der Klasse ch.bouquet.jpa.JpaRunner gestartet. Die Test-Applikation verfügt über zwei unterschiedliche Starter-Methoden. Die Tests können seriell oder parallel gestartet werden.

2. In der main()-Methode wird die Log-Datei initialisiert, damit die Resultate protokolliert werden können.

3. Es werden die Entity-Objekte in die Datenbank geladen. Dafür wird die Klasse ch.bouquet.jpa.data.DataLoader verwendet.

4. Danach werden die Anwendungsfälle [1–9.2] in sequenzieller Reihenfolge verarbeitet.

5. Sind die Verarbeitungen abgeschlossen, werden die Testdaten wieder aus der Datenbank gelöscht.

6. Nach erfolgreichem Löschen der Testdaten wird die Test-Applikation beendet.

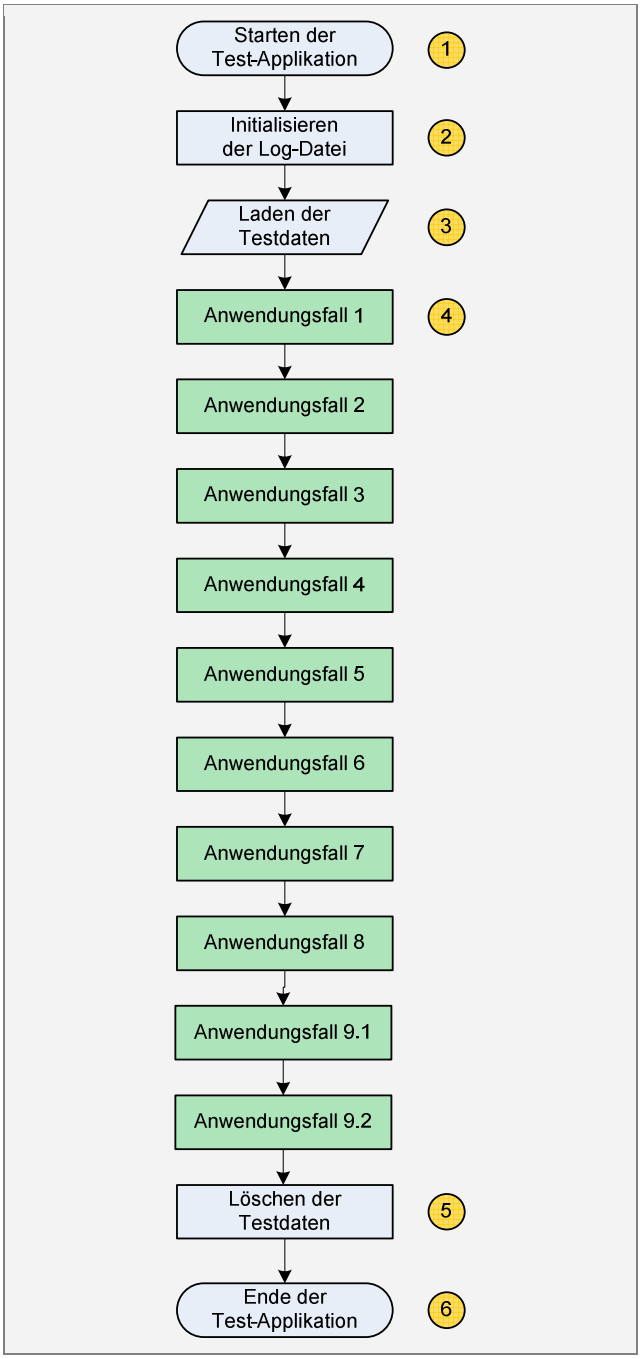

Abbildung 3: Ablaufdiagramm der Test-Applikation im sequenziellen Betrieb

5.3. Ablauf der Test-Applikation – Variante Parallel

1. Die Test-Applikation wird mit der main()-Methode der Klasse ch.bouquet.jpa.JpaRunner gestartet. Die Test-Applikation verfügt über zwei unterschiedliche Starter-Methoden. Die Tests können seriell oder parallel gestartet werden.

2. In der main()-Methode wird die Log-Datei initialisiert, damit die Resultate protokolliert werden können.

3. Es werden die Entity-Objekte in die Datenbank geladen. Dafür wird die Klasse ch.bouquet.jpa.data.DataLoader verwendet.

4. Danach werden die Anwendungsfälle [1 – 8] in sequenzieller Reihenfolge verarbeitet.

5. Nach der seriellen Verarbeitung werden zwei Threads gestartet, die parallel die Anwendungsfälle [9.1] und [9.2] ausführen.

6. Sind die Verarbeitungen abgeschlossen, werden die Testdaten wieder aus der Datenbank gelöscht.

7. Nach erfolgreichem Löschen der Testdaten wird die Test-Applikation beendet.

Abbildung 4: Ablaufdiagramm der Test-Applikation im parallelen Betrieb

5.4. Nachbedingungen

Nachdem die Test-Applikation beendet wurde, wird in der Datenbank kein Eintrag mehr ersichtlich sein. Die Daten werden in der Test-Applikation wieder aus der Datenbank gelöscht. Somit ist der Test fast unendlich durchführbar. Einzig die erzeugten Primärschlüssel werden immer hoch gezählt. Dies kann nur durch das Löschen und erneutes erstellen der Tabellen behoben werden.

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

5.5. Nebenläufigkeit von Entities

In meiner Test-Applikation kann nur eine Nebenläufigkeit auftreten, wenn ein Benutzer sein Benutzername und das Kennwort offenlegt. Damit kann sich ein Benutzer [A] mit den Kenndaten des Benutzer [B] zur gleichen zeit auf dem System anmelden. Dadurch kann der Benutzer [A] die Suchabfragen (Search-Tabelle) oder die Benutzerdaten (Subscriber-Tabelle) des Benutzer [B] verändern.

5.6. Parallelbetrieb der Test-Applikation

Beim starten der Test-Applikation werden die Anwendungsfälle [1 – 8] seriell verarbeitet. Für den Anwendungsfall [9.x] werden nach der seriellen Verarbeitung zwei Threads gestartet. Die Anwendungsfälle sind ab den Punkt 6ff beschrieben. Die Tests zum Parallelbetrieb sind unter den Resultaten des Anwendungsfall [9.x] beschrieben.

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

6. Anwendungsfälle

Die folgenden Anwendungsfälle sind eine Auswahl von vielen Möglichkeiten, die in der Test-Applikation umgesetzt wurden.

6.1. Anwendungsfall 1: Auslesen der aktiven Benutzer

6.1.1. Beschreibung

In diesem Anwendungsfall werden alle aktiven Benutzer ausgelesen. Ein Benutzer ist aktiv, wenn in der AuthUser-Tabelle das Attribut <`state`> auf 1 gesetzt ist.

6.1.2. Query

```
SELECT COUNT(a) FROM AuthUser a WHERE a.state = 1
```

Tabelle 11: Query zum Anwendungsfall 1

6.1.3. Resultat

Das folgende Resultat wird durch die Abfrage in die Log-Datei geschrieben.

```
Es sind zurzeit <3> aktive Benutzer registriert.
```

Tabelle 12: Resultat des Anwendungsfall 1

6.2. Anwendungsfall 2: Auslesen der Benutzer und Rollen

6.2.1. Beschreibung

In diesem Anwendungsfall werden von allen Benutzern der Primärschlüssel, der Login-Name sowie die zugewiesene Authentifizierungsrollen ausgelesen.

6.2.2. Query

```
SELECT a FROM AuthUser AS a INNER JOIN FETCH a.authRoleList
```

Tabelle 13: Query zum Anwendungsfall 2

6.2.3. Resultat

Das folgende Resultat wird durch die Abfrage in die Log-Datei geschrieben.

```
Der Benutzer <1> mit dem Login-Name <admin@bouquet.ch> ist in der Rolle <admin>.
Der Benutzer <2> mit dem Login-Name <manager@bouquet.ch> ist in der Rolle <manager>.
Der Benutzer <3> mit dem Login-Name <user@bouquet.ch> ist in der Rolle <user>.
Der Benutzer <4> mit dem Login-Name <guest@bouquet.ch> ist in der Rolle <guest>.
```

Tabelle 14: Resultat des Anwendungsfall 2

6.3. Anwendungsfall 3: Auslesen aller Registrierungsdaten

6.3.1. Beschreibung

In diesem Anwendungsfall werden von jedem registrierten Benutzer die Registrierungsdaten ausgelesen.

6.3.2. Query

```
SELECT s FORM Subscriber s
```

Tabelle 15: Query zum Anwendungsfall 3

6.3.3. Resultat

Das folgende Resultat wird durch die Abfrage in die Log-Datei geschrieben.

```
--- Registrierungsdaten des Benutzer <1> ---------------------
Marc Bouquet
Kappelenring 49b
CH-3032 Hinterkappelen
admin@bouquet.ch
---------------------------------------------------------------

--- Registrierungsdaten des Benutzer <2> ---------------------
Thomas Iten
Friedlistrasse 14
CH-3006 Bern
manager@bouquet.ch
---------------------------------------------------------------

--- Registrierungsdaten des Benutzer <3> ---------------------
Hans Muster
Musterweg 23
CH-7000 Chur
user@bouquet.ch
---------------------------------------------------------------

--- Registrierungsdaten des Benutzer <4> ---------------------
Greg Miles
14 Hazelwood Lane
UK-Co. Down,BT23 6DG Comber
guest@bouquet.ch
---------------------------------------------------------------
```

Tabelle 16: Resultat des Anwendungsfall 3

6.4. Anwendungsfall 4: Mutieren der Registrierungsdaten

6.4.1. Beschreibung

In diesem Anwendungsfall werden zuerst die Registrierungsdaten des Benutzers mit der Email-Adresse <admin@bouquet.ch> ausgelesen und danach wird die Adresse, Postleitzahl und der Ort des Benutzers mutiert.

6.4.2. Query

```
// Benutzer auslesen
SELECT s FROM Subscriber s WHERE s.emailAddress = 'admin@bouquet.ch'
// Benutzer mutieren auf dem EntitiManager wird der LockModeTyp gesetzt
// em.lock( subscriber, LockModeType.WRITE );
```

Tabelle 17: Query zum Anwendungsfall 4

6.4.3. Resultat

Das folgende Resultat wird durch die Abfrage in die Log-Datei geschrieben.

```
// Benutzer auslesen
--- Registrierungsdaten des Benutzer <1> ---------------------
Marc Bouquet
Kappelenring 49b
CH-3032 Hinterkappelen
admin@bouquet.ch
----------------------------------------------------------------

// Benutzer mutieren
Der Datensatz wurde erfolgreich mutiert ...
// Mutierter Benutzer auslesen
--- Registrierungsdaten des Benutzer <1> ---------------------
Marc Bouquet
Stauffacherstrasse 65
CH-3003 Bern
admin@bouquet.ch
----------------------------------------------------------------
```

Tabelle 18: Resultat des Anwendungsfall 4

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

6.5. Anwendungsfall 5: Auslesen aller täglichen Suchen

6.5.1. Beschreibung

In diesem Anwendungsfall werden die von den Benutzern gespeicherten Suchen analysiert. Es werden von der Search-Tabelle alle Suchabfragen mit der Frequenz <täglich> ausgelesen und als Zusammenfassung ausgegeben.

6.5.2. Query

```
SELECT s FROM Search s WHERE s.frequency='DAILY'
```

Tabelle 19: Query zum Anwendungsfall 5

6.5.3. Resultat

Das folgende Resultat wird durch die Abfrage in die Log-Datei geschrieben.

```
Es wurden <2> Suchabfragen gefunden, die <DAILY> durchgefuehrt werden.

Der Benutzer <1> hat eine Suche gespeichert mit dem Namen <Cola-PDF>.
Diese wird <DAILY> durchgefuehrt. Der aktuelle Status ist <INACTIV>.

Der Benutzer <2> hat eine Suche gespeichert mit dem Namen <Carlsberg.xls>.
Diese wird <DAILY> durchgefuehrt. Der aktuelle Status ist <ACTIV>.
```

Tabelle 20: Resultat des Anwendungsfall 5

6.6. Anwendungsfall 6: Hinzufügen einer wöchentlichen Suche

6.6.1. Beschreibung

In diesem Anwendungsfall wird dem Benutzer mit der Email-Adresse <manager@bouquet.ch> eine wöchentliche Suche hinzugefügt.

6.6.2. Query

```
SELECT s FROM Subscriber s WHERE s.emailAddress = 'manager@bouquet.ch'
```

Tabelle 21: Query zum Anwendungsfall 6

6.6.3. Resultat

Das folgende Resultat wird durch die Abfrage in die Log-Datei geschrieben.

```
Der Benutzer <2> hat eine Suche gespeichert mit dem Namen <Java>.
Diese wird <WEEKLY> durchgefuehrt. Der aktuelle Status ist <INACTIV>.
```

Tabelle 22: Resultat des Anwendungsfall 6

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

6.7. Anwendungsfall 7: Auslesen aller wöchentlichen Suchen

6.7.1. Beschreibung

In diesem Anwendungsfall werden die von den Benutzern gespeicherten Suchen analysiert. Es werden von der Search-Tabelle alle Suchabfragen mit der Frequenz <wöchentlich> ausgelesen und als Zusammenfassung ausgegeben.

6.7.2. Query

```
SELECT s FROM Search s, IN(s.subscriber) sub WHERE s.frequency='WEEKLY'
      ORDER BY sub.subscriberId
```

Tabelle 23: Query zum Anwendungsfall 7

6.7.3. Resultat

Das folgende Resultat wird durch die Abfrage in die Log-Datei geschrieben. Zu beachten ist, dass auch die wöchentliche Suchabfrage aus dem Anwendungsfall 6 ausgegeben wird.

```
Es wurden <3> Suchabfragen gefunden, die <WEEKLY> durchgefuehrt werden.

Der Benutzer <1> hat eine Suche gespeichert mit dem Namen <Pepsi-XML>.
Diese wird <WEEKLY> durchgefuehrt. Der aktuelle Status ist <INACTIV>.

Der Benutzer <2> hat eine Suche gespeichert mit dem Namen <Java>.
Diese wird <WEEKLY> durchgefuehrt. Der aktuelle Status ist <INACTIV>.

Der Benutzer <3> hat eine Suche gespeichert mit dem Namen <emmi pdf>.
Diese wird <WEEKLY> durchgefuehrt. Der aktuelle Status ist <ACTIV>.
```

Tabelle 24: Resultat des Anwendungsfall 7

6.8. Anwendungsfall 8: Löschen eines bestimmten Benutzers

6.8.1. Beschreibung

In diesem Anwendungsfall soll der Benutzer mit der Email-Adresse <user@bouquet.ch> gelöscht werden.

6.8.2. Query

```
SELECT s FROM Subscriber s WHERE s.emailAddress = 'user@bouquet.ch'
```

Tabelle 25: Query zum Anwendungsfall 8

6.8.3. Resultat

Das folgende Resultat wird durch die Abfrage in die Log-Datei geschrieben.

```
Die Daten des Benutzers <4> wurden geloescht!
```

Tabelle 26: Resultat des Anwendungsfall 8

6.9. Anwendungsfall 9: Gleichzeitiges mutieren eines Datensatzes

6.9.1. Beschreibung

In diesem Anwendungsfall werden zwei Benutzer versuchen, denselben Datensatz, gleichzeitig zu mutieren. Der Thread [1] liest den Benutzer mit der Email-Adresse <admin@bouquet.ch> aus und mutiert dessen Adressangaben wie folgt:

- Strasse `<Kappelenring 49b>`
- PLZ `<3032>`
- Ort `<Hinterkappelen>`

Der Thread [2] liest ebenfalls den Benutzer mit der Email-Adresse <admin@bouquet.ch> aus und mutiert dessen Adressangaben wie folgt:

- Name `<Boeser>`
- Vorname `<Haecker>`
- Strasse `<Hackeralle 33>`
- PLZ `<8000>`
- Ort `<Chur>`

Wie sich die Test-Applikation unter den verschiedenen Isolations-Level verhält ist unter dem Punkt 6.9.3ff beschrieben.

6.9.2. Query

```
SELECT s.subscriberId FROM Subscriber s WHERE s.emailAddress = 'admin@bouquet.ch'
```

Tabelle 27: Query zum Anwendungsfall 9

6.9.3. Resultat mit Persistence-Unit «BFH0» (ohne Sperren)

Mit dem Isolations-Level «ohne Sperre» wurden die Daten in die Datenbank geschrieben. Wobei zwischen dem Anwendungsfall [9.1] und [9.2] kein Rollback statt fand, wenn beide Threads gleichzeitig zugriffen. Es ist daher bei den Tests vorgekommen, dass die beiden Updates aus den Anwendungfällen [9.1] und [9.2] vermischt wurden.

```
<persistence-unit name="BFH0" transaction-type="RESOURCE_LOCAL">
  <provider>org.hibernate.ejb.HibernatePersistence</provider>
  <class>ch.bouquet.jpa.domain.AuthRole</class>
  <class>ch.bouquet.jpa.domain.AuthUser</class>
  <class>ch.bouquet.jpa.domain.Subscriber</class>
  <class>ch.bouquet.jpa.domain.Search</class>
  <properties>
    <property name="hibernate.show_sql" value="false" />
    <property name="hibernate.format_sql" value="false" />
    <property name="use_sql_comments" value="false" />
    <property name="hibernate.c3p0.min_size" value="5" />
    <property name="hibernate.c3p0.max_size" value="20" />
    <property name="hibernate.c3p0.timeout" value="300" />
    <property name="hibernate.c3p0.max_statements" value="50" />
    <property name="hibernate.c3p0.idle_test_period" value="3000" />
    <property name="hibernate.connection.url" value="jdbc:mysql://localhost:3306/eadj" />
    <property name="hibernate.connection.driver_class" value="com.mysql.jdbc.Driver" />
    <property name="hibernate.connection.username" value="root" />
    <property name="hibernate.connection.password" value="admin" />
    <property name="show_sql" value="true" />
    <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLInnoDBDialect" />
    <property name="dialect" value="org.hibernate.dialect.MySQLInnoDBDialect" />
  </properties>
</persistence-unit>
```

Tabelle 28: persistence.xml für den Isolations-Level 0

6.9.4. Resultat mit Persistence-Unit «BFH2» (Read Committed)

Mit dem Isolation-Level «Read Committed» wurden beim Update eines Datensatzes die Werte des Anwendungsfall [9.1] und [9.2] vermischt. Es kam auch vor das sich die Test-Applikation blockierte, resp. einfach nicht weiter lief. Den Grund habe ich nicht herausgefunden.

```
<persistence-unit name="BFH2" transaction-type="RESOURCE_LOCAL">
  <provider>org.hibernate.ejb.HibernatePersistence</provider>
  <class>ch.bouquet.jpa.domain.AuthRole</class>
  <class>ch.bouquet.jpa.domain.AuthUser</class>
  <class>ch.bouquet.jpa.domain.Subscriber</class>
  <class>ch.bouquet.jpa.domain.Search</class>
  <properties>
    <property name="hibernate.show_sql" value="false" />
    <property name="hibernate.format_sql" value="false" />
    <property name="use_sql_comments" value="false" />
    <property name="hibernate.c3p0.min_size" value="5" />
    <property name="hibernate.c3p0.max_size" value="20" />
    <property name="hibernate.c3p0.timeout" value="300" />
    <property name="hibernate.c3p0.max_statements" value="50" />
    <property name="hibernate.c3p0.idle_test_period" value="3000" />
    <property name="hibernate.connection.url" value="jdbc:mysql://localhost:3306/eadj" />
    <property name="hibernate.connection.driver_class" value="com.mysql.jdbc.Driver" />
    <property name="hibernate.connection.isolation" value="2" />
    <property name="hibernate.connection.username" value="root" />
    <property name="hibernate.connection.password" value="admin" />
    <property name="show_sql" value="true" />
    <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLInnoDBDialect" />
    <property name="dialect" value="org.hibernate.dialect.MySQLInnoDBDialect" />
  </properties>
</persistence-unit>
```

Tabelle 29: persistence.xml für den Isolations-Level 2

6.9.5. Resultat mit Persistence-Unit «BFH4» (Repeatable Read)

Mit dem Isolation-Level «Repeatable Read» wurde die Datensätze wie erwartet verarbeitet. Bei einem Read-Lock wurde die OptimisticLockException geworfen und nochmals versucht den Datensatz zu mutieren. Dies funktionierte ohne grössere Probleme. Beim Löschen meiner Testdaten konnten aber zum Teil die zu löschenden Daten nicht gelesen werden, obschon diese in der Datenbank vorhanden waren und nicht über den Primärschlüssel gelesen wurden. Wie ich dieses Problem lösen könnte, habe ich in der kurzen Zeit nicht herausgefunden.

```
<persistence-unit name="BFH4" transaction-type="RESOURCE_LOCAL">
  <provider>org.hibernate.ejb.HibernatePersistence</provider>
  <class>ch.bouquet.jpa.domain.AuthRole</class>
  <class>ch.bouquet.jpa.domain.AuthUser</class>
  <class>ch.bouquet.jpa.domain.Subscriber</class>
  <class>ch.bouquet.jpa.domain.Search</class>
  <properties>
    <property name="hibernate.show_sql" value="false" />
    <property name="hibernate.format_sql" value="false" />
    <property name="use_sql_comments" value="false" />
    <property name="hibernate.c3p0.min_size" value="5" />
    <property name="hibernate.c3p0.max_size" value="20" />
    <property name="hibernate.c3p0.timeout" value="300" />
    <property name="hibernate.c3p0.max_statements" value="50" />
    <property name="hibernate.c3p0.idle_test_period" value="3000" />
    <property name="hibernate.connection.url" value="jdbc:mysql://localhost:3306/eadj" />
    <property name="hibernate.connection.driver_class" value="com.mysql.jdbc.Driver" />
    <property name="hibernate.connection.isolation" value="4" />
    <property name="hibernate.connection.username" value="root" />
    <property name="hibernate.connection.password" value="admin" />
    <property name="show_sql" value="true" />
    <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLInnoDBDialect" />
    <property name="dialect" value="org.hibernate.dialect.MySQLInnoDBDialect" />
  </properties>
</persistence-unit>
```

Tabelle 30: persistence.xml für den Isolations-Level 4

6.9.6. Zeitmessung der Testdurchläufe

Die Zeitmessungen sind meines erachten sehr unregelmässig und entsprechen nicht der Theorie. Bei mehr als 50 Durchläufen, wurden nach ca. 2/3 der Verarbeitung, die Zugriffe auf die Datenbank immer langsamer. Dieser Zustand erholte sich aber auf einmal wieder.

Erst bei einem Durchlauf von 500 gibt es grössere Abweichungen bei den verschiedenen Isolations-Levels.

Durchläufe	BFH0	BFH2	BFH4
10 x	13 Sekunden 14 Sekunden	12 Sekunden 14 Sekunden	14 Sekunden 14 Sekunden
50 x	82 Sekunden 53 Sekunden	55 Sekunden 74 Sekunden	56 Sekunden 57 Sekunden
100 x	153 Sekunden 160 Sekunden	159 Sekunden 159 Sekunden	153 Sekunden 164 Sekunden
500 x	843 Sekunden 840 Sekunden	840 Sekunden 838 Sekunden	875 Sekunden 861 Sekunden

Tabelle 31: Zeitmessung der Testdurchläufe

7. Installationsanleitung

7.1. MySQL-Datenbank

Als Datenbank habe ich für mein Projekt eine MySQL-Datenbank (Version 5) verwendet. Damit die Test-Applikation gestartet werden kann, ist es notwendig eine lokale MySQL-Datenbank zu erstellen.

Mit dem vorhandenen SQL-Script (siehe Kapitel 4.2 – 4.6), kann das benötigte Datenbank-Schema erstellt und die Tabellen erzeugt werden.

Das SQL-Script ist auch in der zip-Datei unter « .\eadj_jpa\config\db\setup.sql » vorhanden.

7.2. Test-Applikation

Die Test-Applikation kann mit der Batch-Datei « .\run.bat » gestartet werden.

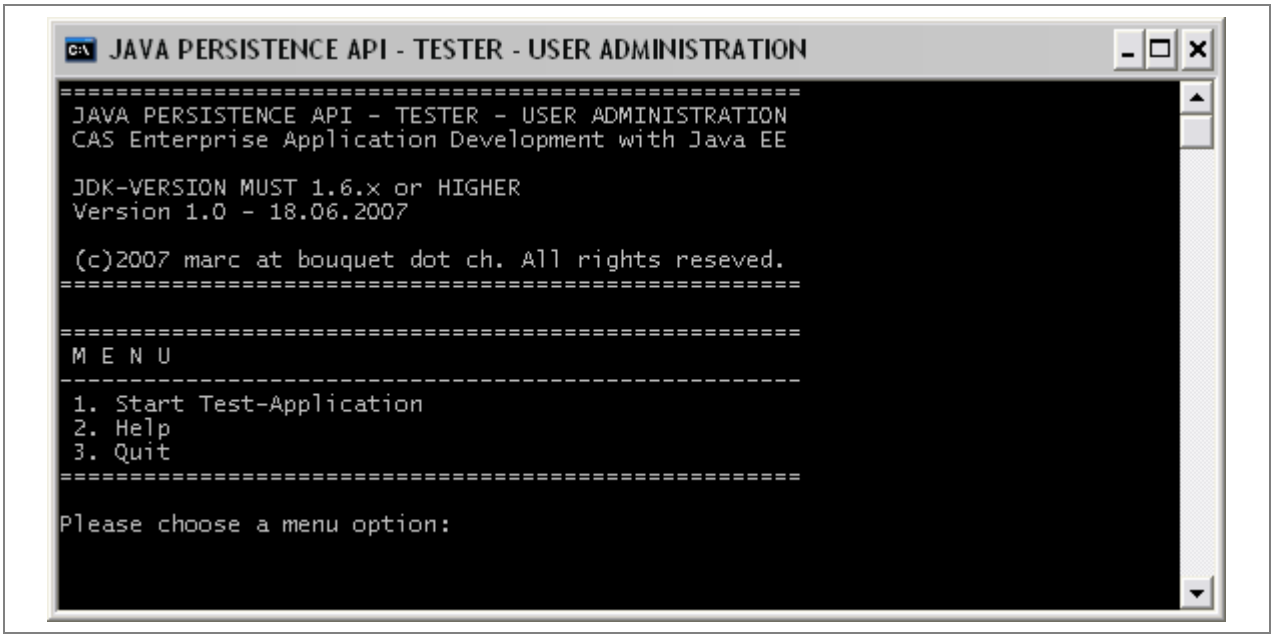

Abbildung 5: run.bat

Zum starten der Test-Applikation müssen vier Parameter übergeben werden.

1. Installationsverzeichnis (Root-Verzeichnis wo die Test-Applikation entpackt wurde)
2. Isolationslevel
3. Anzahl Durchläufe
4. Parallel oder Sequenzielle Ablauf (mit Threads oder ohne Threads)

Abbildung 6: Parameter für run.bat

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

Nach dem die Test-Applikation ihre Verarbeitung beendet hat, werden die Logs in die Log-Datei unter dem folgenden Verzeichnis geschrieben « c:\data\log\eadj-test.log ». Dies können in der Datei « .\eadj_jpa\config\log4j.properies » angepasst werden.

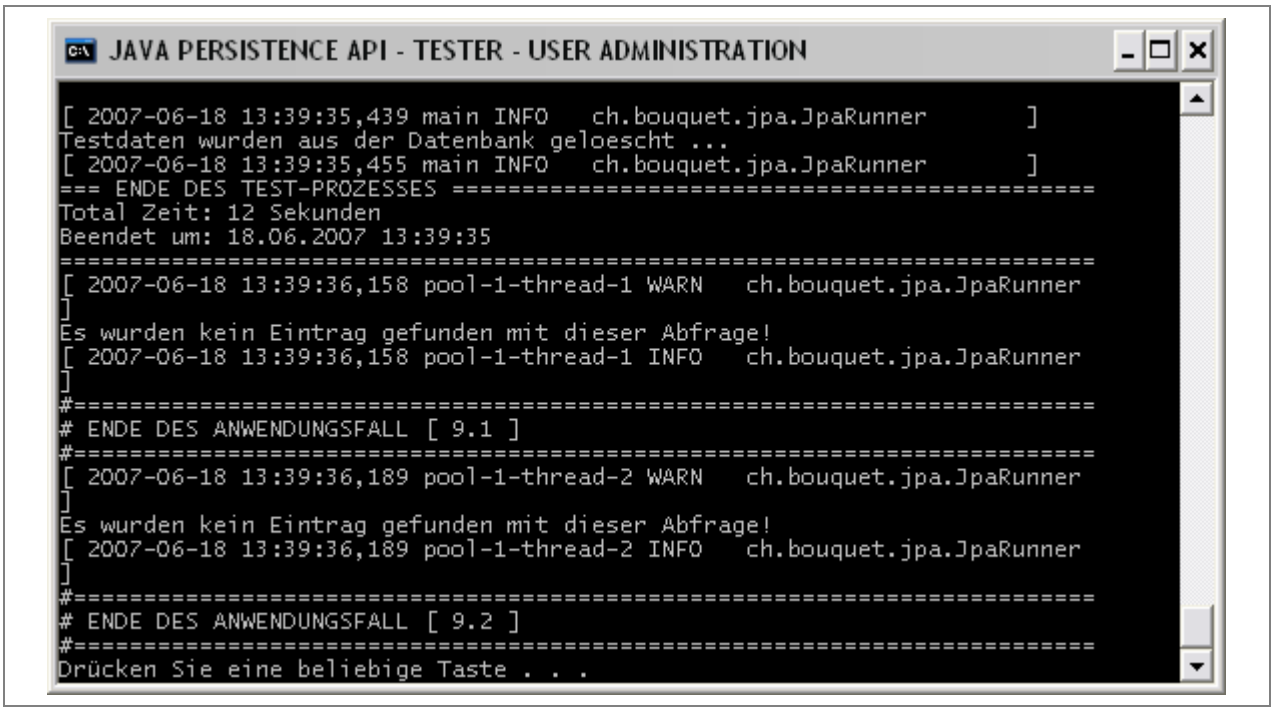

Abbildung 7: Test wurde mit dem run.bat verarbeitet

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

8. Resumée

Da ich mich in den letzten zwei Jahren ausschliesslich mit dem Webframework Java Server Faces beschäftigte, war ich in diesem Gebiet nicht mehr auf dem neusten Stand.

8.1. Java Persistence API

Die Java Persistence API war für mich bis vor diesem Kurs unbekannt. Mit dieser Projektarbeit konnte ich meine ersten eindrücke und Erfahrungen damit erlernen.

8.2. Schwierigkeit mit den Object-Relational-Mapping

Die vier Beziehungstypen werden nach ein paar Implementationen, sicher immer einfacher zu verstehen sein. Am meisten Probleme hatte ich mit den Isolations-Levels resp. mit dem Testen der verschiedenen Speer-Strategien.

8.3. Vererbungsbeziehung

In meinem ausgewählten Model, gibt es keine Vererbungsbeziehungen. Daher wurde dies in der Test-Applikation nicht berücksichtigt.

8.4. Vergleich Firmenlösung vs. Java Persistence API – Ansatz

8.4.1. Vorteile

- Weniger SQL Code, weniger Java Code.
  → **einfachere Wartung**

- Gefahr dass Connections, Statements, etc. durch den Entwickler nicht geschlossen werden ist eliminiert. Mit dem Data Access Object – Pattern muss dies „von Hand“ programmiert werden und so muss jeder Entwickler das Fehlerhandlung selber implementieren.
  → **Qualität, Robustheit**

- Einsatz von Query Language möglich, ist einfacher als SQL.
  → **Änderbarkeit, Wartbarkeit**

- Datenbank Unabhängig, sprich Code ist portierbar. Ist zwar bei SQL auch der Fall, aber die Gefahr dass Datenbank spezifischer Code eingebaut wird ist grösser, wenn man die SQL-Statments selber schreibt.
  → **Übertragbarkeit**

- Effiziente implementierung.
  → **Effizienz**

- Java Persistence API kann auf dem Server als auch Stand Alone (z.B. für Unit Tests) ausgeführt werden, da auf dem Server die Ressourcen „eingespritzt werden“ und für die Unit Tests dies einfach im Setup gemacht werden kann.
  → **testbarer Code**

8.4.2. Nachteile

- Man braucht einen Provider für die Java Persistence API – Klassen
- Ansonsten fast keine, da man auch native Queries machen kann mit der Java Persistence API und somit Datenbank spezifische Abfragen wie bei SQL möglich sind.

8.5. Fazit

Ich habe mit einem relativen grossen Zeitaufwand einen Einblick in den Java Persistence API – Ansatz erhalten. Trotz dem nächtlichen Studium, wurde mein Interesse an der JPA-„Lebenswelt“ geweckt. Ich erhoffe mir mit dem nächsten Kurs noch mehr Erfahrungen zu sammeln und bin auch überzeugt, dass das Java Persistence API in näherer Zukunft in unserer Entwicklungsabteilung im Institut, das bisherige Data Access Object – Pattern ersetzen wird.

9. Glossar und Links

In diesem Kapitel werden Abkürzungen Beschrieben die im Zusammenhang mit dieser Dokumentation stehen oder als Abkürzungen in der Dokumentation vorkommen.

Abkürzung	Beschreibung
Anwendungsfall, UseCase	Ein Anwendungsfall, auch im Deutschen eher unter dem englischen Ausdruck Use Case bekannt, definiert die Interaktionen zwischen Akteuren und dem betrachteten System, die stattfinden, um ein bestimmtes fachliches Ziel (engl. business goal) zu erreichen. Anwendungsfälle beschreiben immer nur genau einen Ablauf oder einen Prozess.
Authentifiziert	Die Authentifizierung (engl. authentication) bezeichnet den Vorgang der Überprüfung der Identität eines Gegenübers (zum Beispiel einer Person oder eines Computerprogramms). Die Authentisierung bezeichnet den Vorgang des Nachweises der eigenen Identität. Im Englischen wird zwischen den beiden Begriffen nicht unterschieden, das Wort authentication steht für beides. Dementsprechend werden auch im Deutschen die Begriffe oft synonym verwendet. Bei einer Identitätsüberprüfung oder Identifizierung gibt es daher immer einen Teilnehmer, der sich authentisiert und einen, der diesen authentifiziert. In einem Computerprogramm werden einer Identität (das heisst einer identifizierten Person) üblicherweise Rechte zugeordnet. Autorisierung bezeichnet den Vorgang, mit dem ein Computerprogramm prüft, ob eine bestimmte Identität ein bestimmtes Recht besitzt (und damit zum Beispiel eine bestimmte Aktion ausführen darf). Die Authentisierung, das heisst das Nachweisen der eigenen Identität, kann auf drei verschiedenen Wegen erfolgen: ▪ Besitz: man hat etwas (Beispiel: Schlüssel, Karte) ▪ Wissen: man weiss etwas (Beispiel: PIN, Passwort) ▪ Körperliches (biometrisches) Merkmal: man ist/kann etwas (Beispiel: Fingerabdruck, Aussehen, siehe auch Biometrische Merkmale) Wenn zwei dieser drei Möglichkeiten kombiniert werden, spricht man von einer 2-Faktor-Authentifizierung. Ein typisches Beispiel dafür ist ein Geldautomat. Man hat etwas, die Bankkarte, zusätzlich muss man aber noch etwas wissen, nämlich die persönliche Identifikations-nummer (PIN).
DAO-Pattern	Data Access Object (DAO, deutsch „Datenzugriffsobjekt“) ist ein Entwurfsmuster, das den Zugriff auf unterschiedliche Arten von Datenquellen (z.B. Datenbanken) so kapselt, dass die angesprochene Datenquelle ausgetauscht werden kann, ohne den aufrufenden Code zu ändern. Dadurch soll die eigentliche Geschäftslogik von technischen Details befreit und flexibler einsetzbar werden. Insbesondere enthält der aufrufende Code keine SQL-Anweisungen und „weiss“ nicht, ob das DAO SQL oder eine andere Technik

	verwendet um die Daten zu beschaffen.
JPA	Mit EJB 3 wurde im Rahmen des Java Community Process das Java Persistence API (JPA) entworfen und verabschiedet. JPA ist zwar im Rahmen von EJB 3 entstanden, aber durchaus auch separat verwendbar. Die Mission: einen einheitlichen Persistenzmechanismus sowohl für Standard-Java als auch für Enterprise-Java bereitzustellen, der alle anderen Standard-Persistenz-Mechanismen wie JDO (Java Data Objects) oder EJB (Enterprise JavaBeans) ablösen soll.
ORM	Object-Relational-Mapping (O/R-Mapping) bezeichnet die Abbildung von objektorientierten Daten auf relationale Daten und umgekehrt. Trotz der heute starken Verwendung einer objektorientierten Programmierung werden die meisten Datenbestände in relationalen Datenbanken gehalten. Somit ist ein O/R-Mapping nötig, welches die Objekte, deren Attribute und Beziehungen auf die Tabellenstrukturen und Spalten (und umgekehrt) abbildet.

Tabelle 32: Glossar und Links

Weitere Informationen finden Sie auch unter http://de.wikipedia.org.

A Anhang

Start der Test-Applikation

```
[ 2007-06-13 23:52:14,562 main INFO   ch.bouquet.jpa.JpaRunner       ]
#=== START DES TEST-PROZESSES ================================================
# CAS Enterprise Application Development mit Java EE (EADJ) - 2007
# Modul: Data-Tier (JPA)
# Uebung: Benutzerverwaltung (Pruefungsaufgabe)
# Copyright (c) 2007 bouquet dot ch, 3032 Hinterkappelen. All rights reseved.
# Marc Bouquet ( marc at bouquet dot ch, Swiss )
# 1.0 - 18.06.2007
#==============================================================================
# Isolations-Level ist: <4>
# Gestartet um: 13.06.2007 23:52:14
#==============================================================================
```

Laden der Testdaten in die Datenbank

```
[ 2007-06-13 23:52:14,562 main INFO   ch.bouquet.jpa.JpaRunner       ]
--- START LADEN DER DATENBANK ------------------------------------------
[ 2007-06-13 23:52:16,062 main DEBUG  ch.bouquet.jpa.data.DataLoader ]

 --- toString() of ch.bouquet.jpa.domain.Subscriber ----------------------
  subscriber id          <1>
  first name             <Marc>
  last name              <Bouquet>
  email address          <admin@bouquet.ch>
  address                <Kappelenring 49b>
  zip code               <3032>
  city                   <Hinterkappelen>
  country                <CH>
  enitity version        <0>

  auth user element      <1>
 --- toString() of ch.bouquet.jpa.domain.AuthUser ------------------------
  authUserId             <1>
  login                  <admin@bouquet.ch>
  password               <admin>
  encryption version     <SHA-1>
  state                  <1>
  last login date        <null>

  auth role element      <1>
 --- toString() of ch.bouquet.jpa.domain.AuthRole ------------------------
  authRoleId             <1>
  name                   <admin>
 ------------------------------------------------------------------------

[ 2007-06-13 23:52:16,062 main DEBUG  ch.bouquet.jpa.data.DataLoader ]
 --- toString() of ch.bouquet.jpa.domain.Subscriber ----------------------
  subscriber id          <2>
  first name             <Thomas>
  last name              <Iten>
  email address          <manager@bouquet.ch>
  address                <Friedlistrasse 14>
  zip code               <3006>
  city                   <Bern>
  country                <CH>
  enitity version        <0>

  auth user element      <1>
 --- toString() of ch.bouquet.jpa.domain.AuthUser ------------------------
  authUserId             <2>
  login                  <manager@bouquet.ch>
  password               <manager>
  encryption version     <SHA-1>
  state                  <1>
  last login date        <null>

  auth role element      <1>
 --- toString() of ch.bouquet.jpa.domain.AuthRole ------------------------
  authRoleId             <2>
```

```
  name                  <manager>
 ------------------------------------------------------------------------

[ 2007-06-13 23:52:16,093 main DEBUG  ch.bouquet.jpa.data.DataLoader ]
 --- toString() of ch.bouquet.jpa.domain.Subscriber ---------------------
  subscriber id         <3>
  first name            <Hans>
  last name             <Muster>
  email address         <user@bouquet.ch>
  address               <Musterweg 23>
  zip code              <7000>
  city                  <Chur>
  country               <CH>
  enitity version       <0>

  auth user element     <1>
 --- toString() of ch.bouquet.jpa.domain.AuthUser -----------------------
  authUserId            <3>
  login                 <user@bouquet.ch>
  password              <user>
  encryption version    <SHA-1>
  state                 <1>
  last login date       <null>

  auth role element     <1>
 --- toString() of ch.bouquet.jpa.domain.AuthRole -----------------------
  authRoleId            <3>
  name                  <user>
 ------------------------------------------------------------------------

[ 2007-06-13 23:52:16,093 main DEBUG  ch.bouquet.jpa.data.DataLoader ]
 --- toString() of ch.bouquet.jpa.domain.Subscriber ---------------------
  subscriber id         <4>
  first name            <Greg>
  last name             <Miles>
  email address         <guest@bouquet.ch>
  address               <14 Hazelwood Lane>
  zip code              <Co. Down,BT23 6DG>
  city                  <Comber>
  country               <UK>
  enitity version       <0>

  auth user element     <1>
 --- toString() of ch.bouquet.jpa.domain.AuthUser -----------------------
  authUserId            <4>
  login                 <guest@bouquet.ch>
  password              <guest>
  encryption version    <SHA-1>
  state                 <0>
  last login date       <null>

  auth role element     <1>
 --- toString() of ch.bouquet.jpa.domain.AuthRole -----------------------
  authRoleId            <4>
  name                  <guest>
 ------------------------------------------------------------------------

[ 2007-06-13 23:52:16,093 main DEBUG  ch.bouquet.jpa.data.DataLoader ]
 --- toString() of ch.bouquet.jpa.domain.Search -------------------------
  search id     <1>
  subscriber id <1>
  name          <Cola-PDF>
  criteria      <Wortlaut der Marke=Cola>
  state         <INACTIV>
  frequency     <DAILY>
  output format <PDF>
  start date    <null>
 ------------------------------------------------------------------------

[ 2007-06-13 23:52:16,093 main DEBUG  ch.bouquet.jpa.data.DataLoader ]
 --- toString() of ch.bouquet.jpa.domain.Search -------------------------
  search id     <2>
  subscriber id <1>
  name          <Pepsi-XML>
  criteria      <Wortlaut der Marke=Pepsi>
```

```
  state         <INACTIV>
  frequency     <WEEKLY>
  output format <XML>
  start date    <null>
 ------------------------------------------------------------------------

[ 2007-06-13 23:52:16,109 main DEBUG  ch.bouquet.jpa.data.DataLoader ]
 --- toString() of ch.bouquet.jpa.domain.Search -------------------------
  search id     <3>
  subscriber id <2>
  name          <Hein eke>
  criteria      <Wortlaut der Marke=Heineke>
  state         <ACTIV>
  frequency     <MONTHLY>
  output format <XML>
  start date    <2007-06-18 00:00:00.093>
 ------------------------------------------------------------------------

[ 2007-06-13 23:52:16,109 main DEBUG  ch.bouquet.jpa.data.DataLoader ]
 --- toString() of ch.bouquet.jpa.domain.Search -------------------------
  search id     <4>
  subscriber id <2>
  name          <Carlsberg.xls>
  criteria      <Wortlaut der Marke=Carlsberg>
  state         <ACTIV>
  frequency     <DAILY>
  output format <EXCEL>
  start date    <2007-09-02 00:00:00.093>
 ------------------------------------------------------------------------

[ 2007-06-13 23:52:16,109 main DEBUG  ch.bouquet.jpa.data.DataLoader ]
 --- toString() of ch.bouquet.jpa.domain.Search -------------------------
  search id     <5>
  subscriber id <3>
  name          <emmi pdf>
  criteria      <Wortlaut der Marke=Caffe Latte>
  state         <ACTIV>
  frequency     <WEEKLY>
  output format <PDF>
  start date    <2007-11-26 00:00:00.093>
 ------------------------------------------------------------------------

[ 2007-06-13 23:52:16,109 main INFO   ch.bouquet.jpa.JpaRunner       ]
--- ENDE DES LADEN DER DATENBANK ----------------------------------------
```

Verarbeitung des Anwendungsfall 1

```
[ 2007-06-13 23:52:16,218 main INFO   ch.bouquet.jpa.JpaRunner       ]
#========================================================================
# START DES ANWENDUNGSFALL [ 1 ]
#========================================================================

[ 2007-06-13 23:52:16,375 main INFO   ch.bouquet.jpa.JpaRunner       ]
Es sind zurzeit <3> aktive Benutzer registriert.

[ 2007-06-13 23:52:16,375 main INFO   ch.bouquet.jpa.JpaRunner       ]
#========================================================================
# ENDE DES ANWENDUNGSFALL [ 1 ]
#========================================================================
```

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

Verarbeitung des Anwendungsfall 2

```
[ 2007-06-13 23:52:16,468 main INFO   ch.bouquet.jpa.JpaRunner       ]
#========================================================================
# START DES ANWENDUNGSFALL [ 2 ]
#========================================================================

[ 2007-06-13 23:52:16,484 main INFO   ch.bouquet.jpa.JpaRunner       ]
Der Benutzer <1> mit dem Login-Name <admin@bouquet.ch> ist in der Rolle <admin>.
Der Benutzer <2> mit dem Login-Name <manager@bouquet.ch> ist in der Rolle <manager>.
Der Benutzer <3> mit dem Login-Name <user@bouquet.ch> ist in der Rolle <user>.
Der Benutzer <4> mit dem Login-Name <guest@bouquet.ch> ist in der Rolle <guest>.

[ 2007-06-13 23:52:16,484 main INFO   ch.bouquet.jpa.JpaRunner       ]
#========================================================================
# ENDE DES ANWENDUNGSFALL [ 2 ]
#========================================================================
```

Verarbeitung des Anwendungsfall 3

```
[ 2007-06-13 23:52:16,640 main INFO   ch.bouquet.jpa.JpaRunner       ]
#========================================================================
# START DES ANWENDUNGSFALL [ 3 ]
#========================================================================

[ 2007-06-13 23:52:16,656 main INFO   ch.bouquet.jpa.JpaRunner       ]
--- Registrierungsdaten des Benutzer <1> --------------------------------
Marc Bouquet
Kappelenring 49b
CH-3032 Hinterkappelen
admin@bouquet.ch
------------------------------------------------------------------------
--- Registrierungsdaten des Benutzer <2> --------------------------------
Thomas Iten
Friedlistrasse 14
CH-3006 Bern
manager@bouquet.ch
------------------------------------------------------------------------
--- Registrierungsdaten des Benutzer <3> --------------------------------
Hans Muster
Musterweg 23
CH-7000 Chur
user@bouquet.ch
------------------------------------------------------------------------
--- Registrierungsdaten des Benutzer <4> --------------------------------
Greg Miles
14 Hazelwood Lane
UK-Co. Down,BT23 6DG Comber
guest@bouquet.ch
------------------------------------------------------------------------

[ 2007-06-13 23:52:16,656 main INFO   ch.bouquet.jpa.JpaRunner       ]
#========================================================================
# ENDE DES ANWENDUNGSFALL [ 3 ]
#========================================================================
```

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

Verarbeitung des Anwendungsfall 4

```
[ 2007-06-13 23:52:16,750 main INFO   ch.bouquet.jpa.JpaRunner       ]
#===============================================================================
# START DES ANWENDUNGSFALL [ 4 ]
#===============================================================================

[ 2007-06-13 23:52:16,750 main INFO   ch.bouquet.jpa.JpaRunner       ]
--- Registrierungsdaten des Benutzer <1> --------------------------------
Marc Bouquet
Kappelenring 49b
CH-3032 Hinterkappelen
admin@bouquet.ch
-------------------------------------------------------------------------

[ 2007-06-13 23:52:16,828 main INFO   ch.bouquet.jpa.JpaRunner       ]
Der Datensatz wurde erfolgreich mutiert ...
--- Registrierungsdaten des Benutzer <1> --------------------------------
Marc Bouquet
Stauffacherstrasse 65
CH-3003 Bern
admin@bouquet.ch
-------------------------------------------------------------------------

[ 2007-06-13 23:52:16,828 main INFO   ch.bouquet.jpa.JpaRunner       ]
#===============================================================================
# ENDE DES ANWENDUNGSFALL [ 4 ]
#===============================================================================
```

Verarbeitung des Anwendungsfall 5

```
[ 2007-06-13 23:52:16,906 main INFO   ch.bouquet.jpa.JpaRunner       ]
#===============================================================================
# START DES ANWENDUNGSFALL [ 5 ]
#===============================================================================

[ 2007-06-13 23:52:16,937 main INFO   ch.bouquet.jpa.JpaRunner       ]
Es wurden <2> Suchabfragen gefunden, die <DAILY> durchgefuehrt werden.

Der Benutzer <1> hat eine Suche gespeichert mit dem Namen <Cola-PDF>.
Diese wird <DAILY> durchgefuehrt. Der aktuelle Status ist <INACTIV>.

Der Benutzer <2> hat eine Suche gespeichert mit dem Namen <Carlsberg.xls>.
Diese wird <DAILY> durchgefuehrt. Der aktuelle Status ist <ACTIV>.

[ 2007-06-13 23:52:16,937 main INFO   ch.bouquet.jpa.JpaRunner       ]
#===============================================================================
# ENDE DES ANWENDUNGSFALL [ 5 ]
#===============================================================================
```

Verarbeitung des Anwendungsfall 6

```
[ 2007-06-13 23:52:17,078 main INFO   ch.bouquet.jpa.JpaRunner       ]
#===============================================================================
# START DES ANWENDUNGSFALL [ 6 ]
#===============================================================================

[ 2007-06-13 23:52:17,109 main INFO   ch.bouquet.jpa.JpaRunner       ]
Der Benutzer <2> hat eine Suche gespeichert mit dem Namen <Java>.
Diese wird <WEEKLY> durchgefuehrt. Der aktuelle Status ist <INACTIV>.

[ 2007-06-13 23:52:17,109 main INFO   ch.bouquet.jpa.JpaRunner       ]
#===============================================================================
# ENDE DES ANWENDUNGSFALL [ 6 ]
#===============================================================================
```

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

Verarbeitung des Anwendungsfall 7

```
[ 2007-06-13 23:52:17,187 main INFO   ch.bouquet.jpa.JpaRunner      ]
#===============================================================================
# START DES ANWENDUNGSFALL [ 7 ]
#===============================================================================

[ 2007-06-13 23:52:17,203 main INFO   ch.bouquet.jpa.JpaRunner      ]
Es wurden <3> Suchabfragen gefunden, die <WEEKLY> durchgefuehrt werden.

Der Benutzer <1> hat eine Suche gespeichert mit dem Namen <Pepsi-XML>.
Diese wird <WEEKLY> durchgefuehrt. Der aktuelle Status ist <INACTIV>.

Der Benutzer <2> hat eine Suche gespeichert mit dem Namen <Java>.
Diese wird <WEEKLY> durchgefuehrt. Der aktuelle Status ist <INACTIV>.

Der Benutzer <3> hat eine Suche gespeichert mit dem Namen <emmi pdf>.
Diese wird <WEEKLY> durchgefuehrt. Der aktuelle Status ist <ACTIV>.

[ 2007-06-13 23:52:17,203 main INFO   ch.bouquet.jpa.JpaRunner       ]
#===============================================================================
# ENDE DES ANWENDUNGSFALL [ 7 ]
#===============================================================================
```

Verarbeitung des Anwendungsfall 8

```
[ 2007-06-13 23:52:17,281 main INFO   ch.bouquet.jpa.JpaRunner       ]
#===============================================================================
# START DES ANWENDUNGSFALL [ 8 ]
#===============================================================================

[ 2007-06-13 23:52:17,281 main INFO   ch.bouquet.jpa.JpaRunner       ]
Die Daten des Benutzers <3> wurden geloescht!

[ 2007-06-13 23:52:17,281 main INFO   ch.bouquet.jpa.JpaRunner       ]
#===============================================================================
# ENDE DES ANWENDUNGSFALL [ 8 ]
#===============================================================================
```

marc bouquet · kappelenring 49b · 3032 hinterkappelen
mobile : +41 79 302 03 84 · email : marc at bouquet dot ch · internet : bouquet dot ch

Verarbeitung des Anwendungsfall 9

```
[ 2007-06-13 23:52:17,562 pool-1-thread-1 INFO   ch.bouquet.jpa.JpaRunner       ]
#=========================================================================
# START DES ANWENDUNGSFALL [ 9.1 ]
#=========================================================================

[ 2007-06-13 23:52:17,578 pool-1-thread-2 INFO   ch.bouquet.jpa.JpaRunner       ]
#=========================================================================
# START DES ANWENDUNGSFALL [ 9.2 ]
#=========================================================================

[ 2007-06-13 23:52:17,578 pool-1-thread-1 INFO   ch.bouquet.jpa.JpaRunner       ]
--- Registrierungsdaten des Benutzer <1> -------------------------------
Marc Bouquet
Stauffacherstrasse 65
CH-3003 Bern
admin@bouquet.ch
-------------------------------------------------------------------------

[ 2007-06-13 23:52:17,625 pool-1-thread-2 INFO   ch.bouquet.jpa.JpaRunner       ]
--- Registrierungsdaten des Benutzer <1> -------------------------------
Marc Bouquet
Stauffacherstrasse 65
CH-3003 Bern
admin@bouquet.ch
-------------------------------------------------------------------------

[ 2007-06-13 23:52:17,625 pool-1-thread-2 INFO   ch.bouquet.jpa.JpaRunner       ]
Der Datensatz wurde erfolgreich mutiert ...
--- Registrierungsdaten des Benutzer <1> -------------------------------
Boeser Haecker
Hackeralle 33
CH-8000 Chur
admin@bouquet.ch
-------------------------------------------------------------------------

[ 2007-06-13 23:52:17,625 pool-1-thread-2 INFO   ch.bouquet.jpa.JpaRunner       ]
#=========================================================================
# ENDE DES ANWENDUNGSFALL [ 9.2 ]
#=========================================================================

[ 2007-06-13 23:52:17,687 pool-1-thread-1 WARN   ch.bouquet.jpa.JpaRunner       ]
Der Datensatz wird zurzeit bearbeitet! Versuchen Sie es spaeter noch einmal ...

[ 2007-06-13 23:52:17,812 pool-1-thread-1 WARN   ch.bouquet.jpa.JpaRunner       ]
Es wurde kein Eintrag gefunden mit dieser Abfrage!

[ 2007-06-13 23:52:17,812 pool-1-thread-1 INFO   ch.bouquet.jpa.JpaRunner       ]
#=========================================================================
# ENDE DES ANWENDUNGSFALL [ 9.1 ]
#=========================================================================

[ 2007-06-13 23:52:17,812 pool-1-thread-1 INFO   ch.bouquet.jpa.JpaRunner       ]
#=========================================================================
# ENDE DES ANWENDUNGSFALL [ 9.1 ]
#=========================================================================
```

Beenden der Test-Applikation

```
[ 2007-06-13 23:52:18,218 main INFO   ch.bouquet.jpa.JpaRunner       ]
Testdaten wurden aus der Datenbank geloescht ...

[ 2007-06-13 23:52:18,218 main INFO   ch.bouquet.jpa.JpaRunner       ]
=== ENDE DES TEST-PROZESSES ============================================
Total Zeit: 3 Sekunden
Beendet um: 13.06.2007 23:52:18
=========================================================================
```

Tabelle 33: Log-Datei eines Parallelen Testlauf mit dem Isolationslevel 4