

Bericht

Labor Datenbanken Gruppe 13

Andrey Nikolaev
Anna Ostrovskaya



Hochschule Offenburg,
Wintersemester 2013/14

Inhaltsverzeichnis

1.Aufgabenbeschreibung.....	3
2.Designentscheidung.....	4
2.1.Datenbankdesign.....	4
2.1.1.Zusammenwirken mit Oracle database.....	5
2.1.2.Zusammenwirken mit MySQL database.....	5
2.2.Anwendungsdesign.....	6
2.2.1.Entwicklungswerkzeug.....	8
2.2.2.Testen.....	8
3.Kurzbeschreibung der Anwendung.....	9
4.Abschluss.....	11
5.Anhang.....	12
5.1.SQL-Skript für Oracle.....	12
5.2.SQL-Skript für MySQL.....	12
5.3.Quelltext einiger Java Klassen.....	12
6.Quelle.....	13
6.1.Literatur.....	13
6.2.Andere Quelle.....	13

1. Aufgabenbeschreibung

Die Aufgabe bestand aus zwei Teilen. Der erste Teil ist die Implementierung einer eigenen relationalen Datenbank. Der zweite Teil ist die Realisierung einer webbasierten Applikation, basierend auf JSP-Technologie, um unsere Datenbank zu verwalten.

Bei der Erstellung unserer Anwendung haben wir ein Zugangskontrollsystem für die verschiedenen Objekte eines Unternehmens, sowie ein Zeiterfassungssystem implementiert.

Im allgemeinen Fall kann man als Zugriffsobjekte entweder immaterielle Objekte verstehen, wie Dateien, Konten, SQL-Tabellen oder materielle Objekten, wie Autos, Räume usw. verstehen. Dafür muss man das Zusammenwirken zwischen den verschiedenen Anwendungen und Geräten, wie Fingerabdruckscannern oder Schlüsselkartenlesern, betrachten. In unserer Arbeit haben wir diesen Aspekt nicht betrachtet. Zugriffssubjekte können Benutzer, Kunden, Angestellte oder Rechner sein.

Als Vorbild haben wir ein Bergwerk genommen. Die Zugriffsobjekte hier sind Schächte, die Zugriffssubjekte sind Angestellte des Bergwerks.

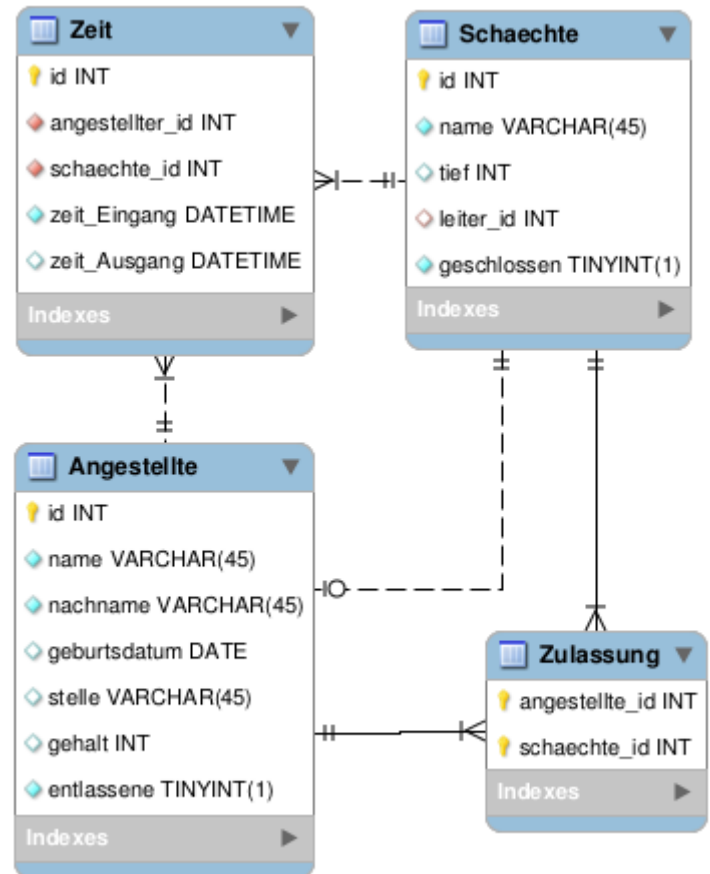
2. Designentscheidung

2.1. Datenbankdesign

Unsere Datenbank enthält vier Tabellen: Angestellte, Schächte, Zeit und Zulassung.

→ Die Tabelle „Angestellte“ enthält Information über Mitarbeiter und besteht aus den Attributen:

- id: primary key, Identifizierer Angestellten;
- name: darf nicht null sein;
- nachname: darf nicht null sein;
- geburtsdatum;
- stelle: Benennung der Stelle, die ein Mitarbeiter besetzt;
- gehalt;
- entlassene: boolean Attribut, darf nicht null sein. Falls „true“, gilt der Mitarbeiter als entlassen.



ER-Schema der Datenbank

→ Die Tabelle „Schächte“ enthält Information über Schächte und besteht aus weiteren Attributen:

- id: primary key, Identifizierer des Schachts;
- name: darf nicht null sein;
- tief: gibt die Tiefe an;
- leiter_id: foreign key, ID des Angestellte, der für diesen Schacht verantwortlich ist (1-zu-1-Beziehung mit der Tabelle „Angestellte“);
- geschlossen: boolean Attribut, darf nicht null sein. Falls „true“: Schacht ist geschlossen.

→ Die Tabelle „Zulassung“ enthält Information über die Zugriffsrechte der Schächte. Eigentlich ist diese Tabelle nur

eine n-zu-n-Beziehung zwischen den Tabellen „Angestellte“ und „Schächte“. Sie besteht aus weiteren Attribute:

- `angestellte_id`: primary key, foreign key, ID des Angestellten, der das Zugriffsrecht hat;
- `schaechte_id`: primary key, foreign key, ID des Schachts für den der Angestellte zugelassen ist.

➔ Die Tabellle „Zeit“ enthält Information über Zeiterfassung und besteht aus weiteren Attributen:

- `id`: primary key, Identifizierer des Tupels;
- `angestellte_id`: foreign key, darf nicht null sein. ID des Angestellten, der eingetreten ist (1-zu-n-Beziehung mit der Tabelle „Angestellte“);
- `schaechte_id`: foreign key, darf nicht null sein. ID des Schachts, in den Angestellte eingetreten sind (1-zu-n-Beziehung mit der Tabelle „Schaechte“);
- `zeit_eingang`: darf nicht null sein;
- `zeit_ausgang`.

Für das Anbinden des Datenbankschema verwenden wir das Framework „Apache Cayenne“.

2.1.1. Zusammenwirken mit Oracle database

Für die Anwendung werden die folgenden Objekte benötigt:

Index: `SCHAECHTE_PK`, `ANGESTELLTE_PK`, `ZEIT_PK`, `ZULASSUNG_PK`;

Trigger: `SCHAECHTE_TRG`, `ANGESTELLTE_TRG`, `ZEIT_TRG`;

Sequence: `PK_SCHAECHTE`, `PK_ANGESTELLTE`, `PK_ZEIT`.

Mithilfe dieser Objekte wird die Einzigkeit der primary keys in den Tabellen „Angestellte“, „Schaechte“ und „Zeit“ beobachtet.

2.1.2. Zusammenwirken mit MySQL database

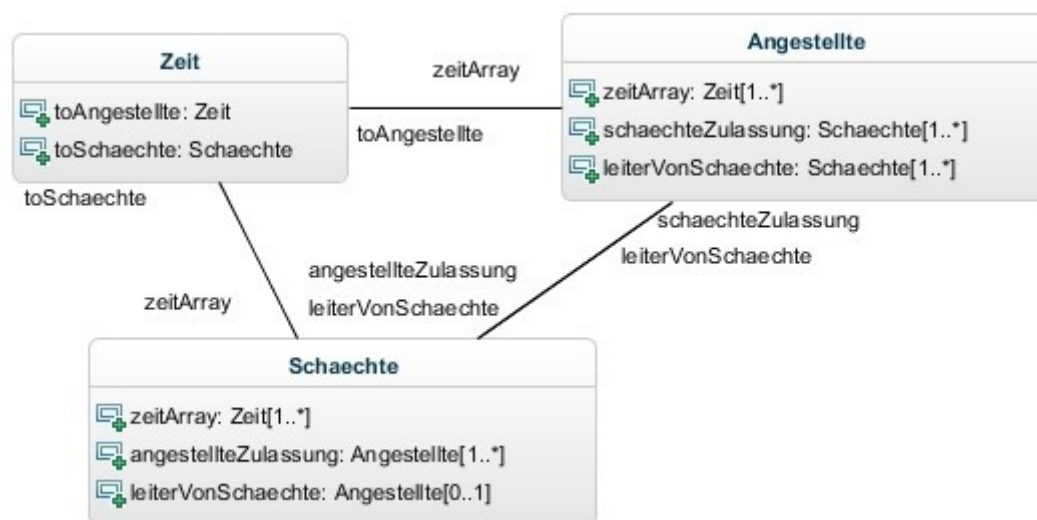
Für die die Anwendung ist es nötig eine Tabelle „`AUTO_PK_SUPPORT`“ zu erstellen. Diese Tabelle enthält den primary key „next value“ für die automatische Generierung der „id“-Spalte in den Tabellen „Angestellte“, „Schaechte“ und „Zeit“.

2.2. Anwendungsdesign

Wir verwenden für das Abbilden des Datenbankschemas auf Java-Klassen das ORM-Framework „Apache Cayenne“.

Die Datei „cayenne-dblab.xml“ enthält die Einstellungen, die für das Verbinden zur Datenbank benötigt werden. Die Datei „dbl原因ab.map.xml“ enthält die Beschreibung aller Entitäten und Relationen der Datenbank. Diese zwei Dateien wurden mithilfe des Programms „Cayenne Modeler“ generiert.

Im Package „de.dblab.domain“ liegen die Klassen `Angestellte.class`, `Schaechte.class` und `Zeit.class`. Das sind die Java-Klassen, mithilfe derer wir die entsprechenden Datenbankobjekte verbinden.



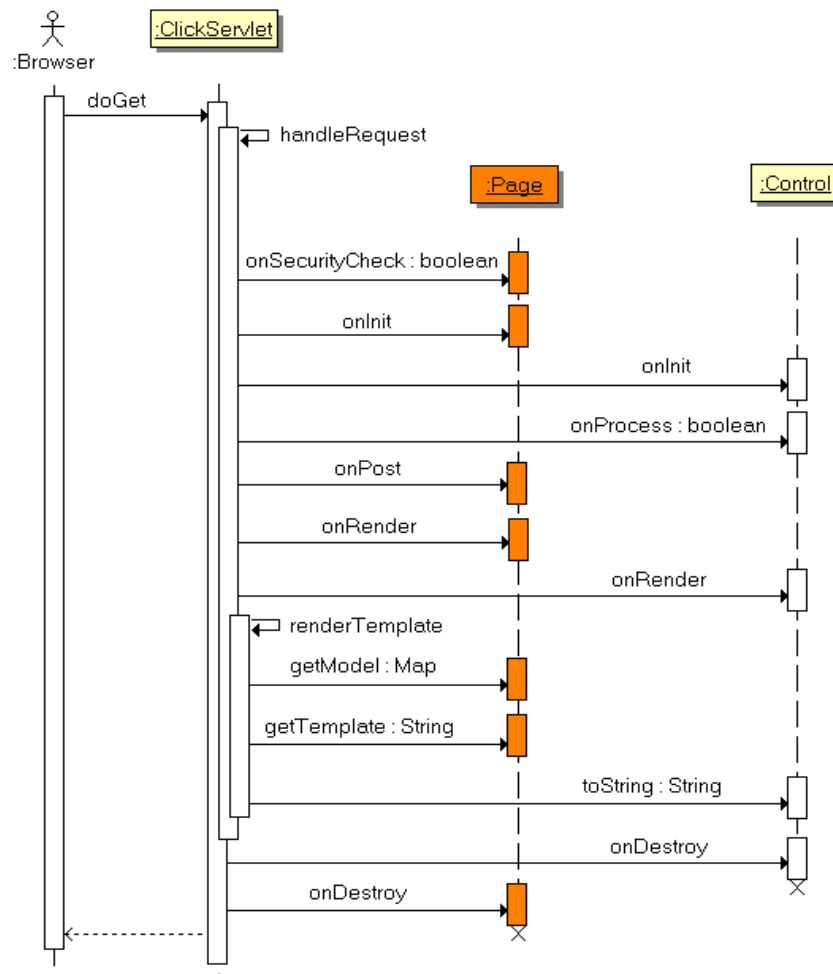
Objekt-Entity Klassendiagramm

Außerdem benutzen wir in unserem Projekt das Framework „Apache Cklick“, mithilfe deren wir das Web-Interface entwickeln. Im Package „de.dblab.page“ liegen die Klassen „`HomePage.class`“, „`TemplatePage.class`“ und die Packages „angestellte“ und „schaechte“. Das sind die Java-Objekte, mithilfe derer wir die entsprechenden Webseiten erstellen. Die Entsprechung zwischen den Java-Klassen und HTML-Seiten wird in der Datei „web/WEB-INF/click.xml“ beschrieben. Die Klasse „`DataBaseService.class`“ enthält die Methoden, die die Datenbankinhalt mit dem Webinterface verbinden.

Im Package „angestellte“ liegt die „Klasse `AngestelltePage.class`“, die den HTML-Code für die Seite „page/angestellte/AngestelltePage.htm“ generiert. Die Klassen „`AngestellteNewForm.class`“ und „`AngestellteSearchForm.class`“ generieren den HTML-Code für die Formulare in der oberen linken

und der oberen rechten Ecke der Seite, die Suchen unter „Angestellte“ und die Erstellung eines neuen Angestellten implementiert. Die Klasse „AngestellterForm.class“ generiert den HTML-Code für das Formular, das die Details des jeweiligen Angestellten zeigt. Dieses Form kann man durch einen Klick auf das Symbol „View“ sehen. Die Klasse „AngestellteEditPage.class“, generiert den HTML-Code für die Seite „page/schaechte/edit-schaechte.htm“. Auf dieser Seite kann man Tupel aus der Tabelle „Angestellte“ ändern.

Das Package „schaechte“ hat eine ähnliche Struktur.



Die POST-Anfrage Seite Ausführungssequenz

Die Datei „web/script/table-ajax.js“ enthält den Ajax-Code, der für das Anzeigen der Formulare und der Sortierung der Tabellen genutzt wird.

2.2.1. Entwicklungswerkzeug

Unsere Anwendung haben wir mit „NetBeans 7.4“ geschrieben. Außerdem wurden „MySQL Workbench 6.0“, „SQL Developer Version 4.0.0“ und „Cayenne Modeler Version 3.2M1“ benutzt.

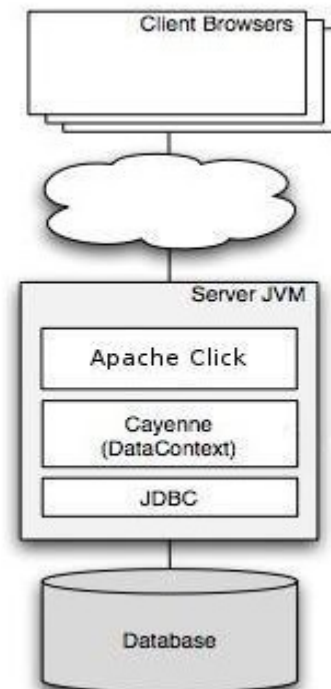
2.2.2. Testen

In unserem System werden der Anwendungsserver „Apache Tomcat“ und entweder Oracle oder MySQL-Datenbanken benutzt. Um Datenbank zu wechseln, genügt es die Datei „cayenne-dblab.xml“ zu wechseln.

Die Server unserer Anwendung wurde mit „Apache Tomcat 7.0“, „Oracle database Express Edition 11.2“ und „MySQL Version 5.5“ getestet.

Unsere Anwendung wurde unter den Betriebssystemen Linux (Fedora), Windows und Android getestet. Wir haben Seite unter den Browser „Google Chrome“ Version 33, „Mozilla Firefox“ Version 26.0 und „Internet Explore“ Version 10 getestet. Unter „Internet Explore“ Version 6 funktioniert es nicht, weil IE6 einige CSS-Tags nicht beherrscht.

Mit dem Anwendungsserver „Apache Tomcat“ kann das System Plattformunabhängig genutzt werden. Veränderungen der Hauptkonfigurationen des Servers (Betriebssystem, RDBMS) werden für den Benutzer unbemerkt bleiben.



Baum Niveau Web-Applikation

3. Kurzbeschreibung der Anwendung

Unsere Anwendung besteht aus mehreren Seiten, die den gesamten Zyklus der Verarbeitung des Wesens von Datenbank implementieren (Erschaffung, Bearbeiten, Entwerfen von Tupel).

Alle Seiten bestehen aus Titel, Menü und Arbeitsbereich. Der Übergang zwischen den Seiten ist nur im Arbeitsbereich sichtbar, weil alle Seiten die gleiche Templates benutzen (TemplatePage.htm).



Mithilfe der Seite „page/angestellte/AngestelltePage.htm“ kann man:

- Den Inhalt der Tabelle „Angestellte“ ansehen
- Das Ergebnis einer Suche in der Tabelle „Angestellte“ sehen
- Neue Tupel in die Tabelle „Angestellte“ einfügen
- Tupel aus der Tabelle „Angestellte“ ändern oder entfernen.
- Den Inhalt der Tabelle „Zulassung“ für einen Angestellten sehen
- Neue Tupel in der Tabelle „Zulassung“ einfügen
- Den Inhalt der Tabelle „Zeit“ für einen Angestellten sehen

Mithilfe der Seite „page/schaechte/SchaechtePage.htm“ kann man:

- Den Inhalt der Tabelle „Schaechte“ sehen
- Das Resultat einer Suche in der Tabelle „Schaechte“ sehen

- Neue Tupel in die Tabelle „Schaechte“ einfügen
- Tupel aus der Tabelle „Schaechte“ bearbeiten oder löschen
- Den Inhalt der Tabelle „Zulassung“ in Verbindung mit der Tabelle „Zeit“ für einen Schacht sehen. Gezeigt werden nur deren zugelassene Angestellte, die nicht in einem Schacht sind
- Den Inhalt der Tabelle „Zeit“ für einen Schacht sehen
- Neue Tupel in der Tabelle „Zeit“ einfügen

4. Abschluss

Im Laufe des Projekts wurden die relationale Datenbank, die Zugangskontrollsystem und die Zeiterfassungssystem implementiert, projektiert und erstellt. Wir haben eine webbasierten Applikation auf JSP-Basis zur Verwaltung der Datenbank realisiert.

Während des Projekts haben wir:

- Ein Einblick in die Funktionsweise der relationalen Datenbank an den Beispielen „Oracle Database“ und „MySQL“ genommen
- Die Grundfunktionen des ORM-Frameworks „Apache Cayenne“, das ein wichtiger Teil der modernen Programmierung ist, kennengelernt
- Einen Einblick in die Funktionsweise des Webframeworks „Apache Click“, mit dem man Webanwendungen erstellen kann, genommen

Außerdem wurden CSS und Ajax verwendet.

5. Anhang

5.1. SQL-Skript für Oracle

5.2. SQL-Skript für MySQL

5.3. Quelltext einiger Java Klassen

6. Quelle

6.1. Literatur

Dokumentation MySQL: <http://dev.mysql.com/doc/>

Dokumentation Oracle <http://www.oracle.com/>

Dokumentation Apache Cayenne: <http://cayenne.apache.org/docs/3.0/>

Dokumentation Apache Click: <http://click.apache.org/docs/user-guide.html>

Dokumentation Ajax Technologie: <http://api.jquery.com/>

6.2. Andere Quelle

Bilder: <http://findicons.com/>

Beispiele Apache Click: <http://click.avoka.com/click-examples/home.htm>