



Einführung in Datenbanksysteme

Projektbeschreibung

Agnès Voisard, Daniel Kreßner

Ausgabe: 26. Mai 2016

1. Anwendungsbeschreibung

Es soll eine Datenbank für Ergebnisse von Hunderennen erstellt werden. Die Datenbank speichert Hunde, Verwandtschaften, Rennen, Ergebnisse und weitere Informationen.

Eine Hunderennveranstaltung wird immer ein Jahr im Voraus durch den betreffenden Hunde(dach)verband offiziell angekündigt. Während eines Hunderennens finden eine Vielzahl von Läufen statt. Ein gewerteter Lauf kommt immer dann zu Stande, wenn mindestens 3 Hunde einer Rasse zum entsprechenden Hunderennen gemeldet wurden. In einem Lauf treten mindestens 3 und maximal 6 Hunde gegeneinander an. Ein Hund hat immer einen Besitzer, aber nicht notwendiger Weise einen Züchter. Ein Hund kann im Alter von 15 bis 96 Monaten an Rennveranstaltungen teilnehmen, danach scheidet er aus dem aktiven Sport aus und darf nicht mehr an Rennveranstaltungen teilnehmen. Hunde die jünger als 24 Monate und älter als 72 Monate sind müssen vor jedem Rennen gesondert durch einen Tierarzt untersucht werden.

Bei den Rennergebnissen sind die Einzelergebnisse der jeweiligen Hunde, aber auch die Ergebnisse des Zwingers von Interesse. Der Zwingername ist Bestandteil des Namens eines Rassehundes, z. B. der Hundename „Akinlana Olabisi“ setzt sich zusammen aus dem Hundenaamen „Akinlana“ dem Zwingernamen „Olabisi“, es sind aber auch Kombinationen wie „BelleVite’s Bahira“, wobei „BelleVite“ der Zwingername und „Bahira“ der Hundename ist, möglich. Es können auch Hunde ohne Papiere (keine Rassehunde) an Rennveranstaltungen teilnehmen. Diese Hunde haben nur einen Rufnamen und keinen Zwingernamen (da kein Züchter). Der Anfangsbuchstabe des Rufnamens der Hunde eines Zwingers hängt von der Anzahl der Würfe des Zwingers ab. Der Rufname der Hunde des ersten Wurfes des Zwingers beginnt mit dem Buchstaben A, der zweite Wurf mit B usw.

Für jedes erfolgreiche Rennen (Hund läuft durchs Ziel) erhält ein Hund Punkte. Der erste Hund erhält Punkte in Höhe der Anzahl der Teilnehmer +1, der zweite Hund erhält Punkte in Höhe der Anzahl der Teilnehmer -1, der dritte Hund erhält einen Punkt weniger als der zweite Hund so weiter

Die zu entwickelnde Anwendung stellt unter anderem folgende Informationen zu den Hunden bereit:

- Rang des Hundes in der Jahresrangliste
- Hundename
- Geschlecht
- Elterntiere (Rüde + Hündin)
- Anzahl der Rennen
- Kumulierte Punktzahl in den Rennen
- Durchschnittliche Renndistanz

Zusätzliche Daten ergeben sich aus den zur Verfügung gestellten Rohdaten.

Weiterhin wird die Datenbank für eine OLAP/Data-Mining-Anwendung zur Platzierungsprognose genutzt. Details hierzu finden Sie im Abschnitt 3.7 „Data-Mining“.

Die folgenden Anfragen sollen durch die Anwendung beantwortet werden können (exemplarisch):

- Geben Sie die 20 schnellsten Hunde aus.
- Geben Sie die 5 erfolgreichsten Zwinger aus.
- In welchem Jahr hatte ein Hund das erste Rennen?
- Welche Eltern haben die schnellsten Nachkommen?
- Gibt es Rüden, die besonders viele Nachkommen haben, aber selbst nicht besonders schnell sind?
- Gibt es Hündinnen, die nur Nachkommen haben, die in den Top 10 sind?
- Welche Hunde sind besonders langsam?
- Welche 10 Hunde haben die höchste durchschnittliche Punktzahl pro Rennen?
- Welche 10 Hunde haben die höchste durchschnittliche Punktzahl pro gelaufenem Meter?

2. Technologien

Die Tutoren könnten Hilfestellung zu folgenden Technologien geben: Java, JDBC und PostgreSQL. Nach Absprache können auch andere Technologien verwendet werden. Die Verwendung von Frameworks zum OR-Mapping wie Hibernate oder Oberflächen wie AWT/Swing sind ausgeschlossen. Die Benutzerschnittstelle ist als Webseite mit JSPs umzusetzen.

3. Teilaufgaben

3.1. Voraussetzung

- Installation von PostgreSQL als Server
- Installation des Java JDK
- Installation einer Entwicklungsumgebung (z. B. IntelliJ)

3.2. Anlegen der Datenbank

Legen Sie eine Datenbank mit dem Namen „dograce“ an.

3.3. Modellierung

Entwerfen Sie auf Grundlage der Anwendungsbeschreibung und den Daten ein Datenbank-schema in, der aus der Vorlesung bekannten umgekehrten Chen-Notation mit (min, max) Erweiterung.

3.4. Übersetzen ins relationale Modell und SQL

Schreiben Sie die entsprechenden SQL-Befehle zur Erstellung der Tabellen. Achten Sie auf eine gute Wahl von Attributeigenschaften wie NOT NULL, UNIQUE und Schlüsseln.

3.5. Datenimport

- Importieren Sie die notwendigen Daten.
- Definieren Sie Transformationsprozesse zur Umwandlung der importierten Daten in das erstellte Relationenmodell: Definieren Sie einen Transformationsprozess ausschließlich unter Verwendung von SQL Befehlen.
- Definieren Sie einen weiteren Transformationsprozess, in welchem Sie alle Umformungen in Java realisieren und nur durch SELECT und INSERT Statements auf die Datenbank zugreifen (nicht in Kombination).
- Die Daten werden als CSV zur Verfügung gestellt.
- Sehen Sie sich die Importfunktionen von PostgreSQL an. So dauert ein Import z. B. sehr lange, wenn einzelne Statements ausgeführt werden oder Integritätsbedingungen während des Imports geprüft werden. Dies sollte deshalb erst nach dem Import erfolgen.

3.6. Benutzerschnittstelle

Entwerfen Sie eine für die oben beschriebene Anwendung passende Benutzerschnittstelle und setzen Sie diese um. Achten Sie auf die Austauschbarkeit von Backend und Frontend. Für den ersten Schritt genügt eine Konsolenanwendung, welche später durch eine Webseite ersetzt werden soll.

3.7. Data-Mining

Es soll eine OLAP/Data-Mining-Anwendung zur Platzierungsprognose erstellt werden. Hierzu soll zunächst die Leistungskurve eines durchschnittlichen Rüden, bzw. einer durchschnittlichen Hündin eines Zwingers berechnet werden. Im Anschluss werden 6 Hunde zufällig ausgewählt und treten in einem virtuellen Rennen im Jahr 2016 gegeneinander an. Das virtuelle Rennen wird ein Jahr später (2017) wiederholt und der Ausgang wird erneut prognostiziert. Die Prognose wird solange wiederholt, bis der erste Hund aus dem virtuellen Rennen ausscheidet.

Aufgabe: Erstellen Sie eine GUI (Webseite), mit der die beschriebene OLAP/Data-Mining-Anwendung realisiert werden kann.

4. Weiteres

4.1. Dokumentation

Dokumentieren Sie im Verlauf des Projektes alle wichtigen Designentscheidungen und den Quellcode, sodass Sie bei der abschließenden Präsentation nachvollziehen können, wie der Code funktioniert, was an dieser Stelle geschieht und welche Entscheidungen hier eventuell getroffen wurden. Benutzen Sie hierfür für den Java-Code JavaDoc. Um das Projekt erfolgreich abschließen zu können, müssen die Projektdokumentation und der Quelltext abgegeben werden.

4.2. Präsentation

Bewertungsgrundlage sind die Zwischenpräsentationen in Anwesenheit eines Tutors und die Abgabe der Dokumentation und Quelltexte.

4.3. Iterationen

4.3.1. Aufgaben der ersten Projektiteration

- ERM anhand der Projektbeschreibung und Daten erstellen
- Relationales Modell ableiten
- CREATE-Statements zur Erzeugung der Tabellen schreiben
- Tabellen in der Datenbank *dograce* anlegen
- Präsentation der Ergebnisse in Form einer kurzen Präsentation (max 10 min)

4.3.2. Aufgaben der zweiten Projektiteration

- Bereinigung der Daten
- Import der Daten über SQL oder Java
- API (vorerst für Konsolenanwendung)
- Erweiterung der API um Beispielabfragen
- Erweiterung der API um benutzerdefinierte SQL-Abfragen
- Präsentation der Ergebnisse in Form einer kurzen Präsentation (max 10 min)

4.3.3. Aufgaben der dritten Projektiteration

- Entwurf und Umsetzung einer Web-GUI mit Zugriff über API
- Erweiterung der Web-GUI um Data-Mining-Aufgabe
- Präsentation der Ergebnisse in Form einer kurzen Abschlusspräsentation (max 10 min)
- Abgabe der Dokumentation, Präsentationen und des Quelltextes.

Für die Abgabe erstellen Sie ein Zip-Archiv mit den Unterverzeichnissen src und doc mit entsprechenden Inhalten. Benennen Sie das Zip-Archiv wie folgt:

DBS16_Projekt_NameTutor_TUT[_nameStudent]+.zip

Zum Beispiel: DBS16_Projekt_Ajit_TUT_Mueller_Meier.zip