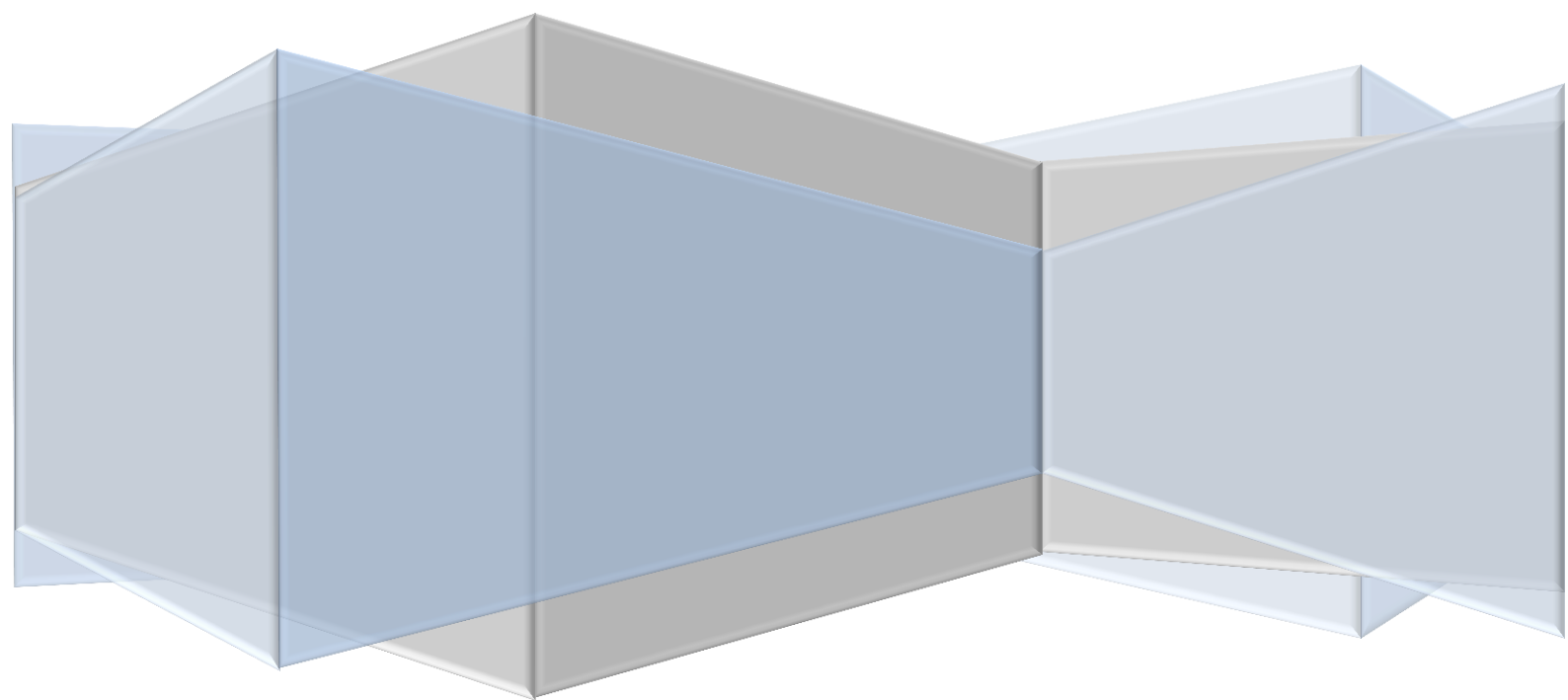


TGM Wien

Segelverein

INSY 2014/2015 | Stand: 18.03.2015

Michael Weinberger 4AHIT



Inhalt

Aufgabenstellung.....	2
Beschreibung auf Moodle	2
Beschreibung	2
ER-Diagramm	2
Relationenmodell	3
SQL-Abfragen.....	4
Java & JDBC	4
Abgabe.....	5
Durchführung	6
Erster Ansatz.....	Fehler! Textmarke nicht definiert.
Konkrete Idee	Fehler! Textmarke nicht definiert.
Umsetzung.....	Fehler! Textmarke nicht definiert.
Detaillierte Arbeitsaufteilung (Aufwandsabschätzung, Endzeitaufteilung)	7
Aufgabentrennung	Fehler! Textmarke nicht definiert.
Aufwandabschätzung	7
Endzeitaufteilung	7
Fazit	7
Arbeitsdurchführung	8
Resultate.....	8
Quellenangaben:	9

Aufgabenstellung

Beschreibung auf Moodle

Sie erhalten den Auftrag für den Europäischen Dachverband der Segelvereine eine Datenbank zu implementieren.

Beschreibung

Für Segler und Trainer sind Name (NAME) und Geburtsdatum (GEBURTSDATUM) bekannt. Sie werden beide identifiziert durch eine eindeutige Nummer (KEY). Mindestens zwei Segler, maximal jedoch vier Segler bilden eine Mannschaft. Für jede Mannschaft wird ein eindeutiger Name (NAME) und eine Altersklasse (AKLASSE) gespeichert. Jede Mannschaft wird genau von einem Trainer betreut. Ein Trainer kann jedoch mehrere Mannschaften betreuen.

Jeder Mannschaft sind Boote zugewiesen. Ein Boot kann mehreren Mannschaften zugewiesen sein. Ein Boot wird eindeutig durch eine Nummer (ID) identifiziert. Weiters sind zu jedem Boot ein Name (NAME), die Anzahl der Personen (PERSONEN) und der Tiefgang (TIEFGANG) bekannt. Es gibt Tourenboote und Sportboote. Tourenboote haben zusätzlich eine Bootsklasse (BOOTSKLASSE) und Sportboote haben zusätzlich eine Segelfläche (SEGELFLAECHE) gespeichert. Es ist außerdem bekannt welche Mannschaften mit welchen Sportbooten an welchen Regatten mit welcher Startnummer (STARTNR) teilgenommen haben.

Eine Regatta wird eindeutig identifiziert durch ihren Namen (NAME) und durch das Jahr (JAHR), in dem sie stattgefunden hat. Das Land (LAND) ist außerdem noch bekannt. Jede Regatta besteht aus mindestens drei jedoch maximal fünf Wettfahrten. Wettfahrten werden durch die zugehörige Regatta und das Datum (DATUM) identifiziert, außerdem wird die Länge (LAENGE) der Strecke gespeichert. Mannschaften können bei jeder Wettfahrt Punkte (PUNKTE) erzielen.

ER-Diagramm

Siehe eLearning (<http://bit.ly/1BJkaSx>).

Relationenmodell

Person(key, name, geburtsdatum)

Segler(key: Person.key)

Trainer(key: Person.key)

Boot(id, name, personen, tiefgang)

Tourenboot(id: Boot.id, bootsklasse)

Sportboot(id: Boot.id, segelflaeche)

Mannschaft(name, aklasse, key: Trainer.key)

Regatta(name, jahr, land)

Wettfahrt(name: Regatta.name, jahr: Regatta.jahr, datum, laenge)

bildet(key: Segler.key, name: Mannschaft.name)

zugewiesen(id: Boot.id, name: Mannschaft.name)

nimmt_teil(mname: Mannschaft.name, rname: Regatta.name, rjahr: Regatta.jahr, sportboot: Sportboot.id, startnr)

erzielt(mname: Mannschaft.name, wname: Wettfahrt.name, wjahr: Wettfahrt.jahr, wdatum: Wettfahrt.datum, punkte)

SQL-Abfragen

1. Geben Sie alle Mannschaften aus, die bei der Bodenseeregatta im Jahr 2014 teilgenommen haben. Wenn eine Mannschaft mit einem Boot mit der Segelfläche kleiner als 20 m² teilgenommen hat, soll auch die ID des Bootes ausgegeben werden.
2. Geben Sie den Namen und das Geburtsdatum der jüngsten Trainer aus (können auch mehrere sein).
3. Geben Sie alle Personen geordnet nach Geburtsdatum aus, die sowohl Segler als auch Trainer sind, allerdings in keiner Mannschaft dabei sind.
4. Geben Sie alle Personen geordnet nach Geburtsdatum aus, die entweder Segler oder Trainer sind, jedoch nicht beides und vermerken Sie in einer Spalte, ob es sich um einen Trainer oder einen Segler handelt.
5. Geben Sie die Regatten (Name und Jahr) mit den wenigsten Wettfahrten an und geben Sie auch die Anzahl aus.
6. Geben Sie die Namen jener Trainer aus, die zwei oder mehr Mannschaften betreuen.
7. Welche Altersklasse ist am aktivsten (hat an den meisten Wettfahrten Punkte erzielt)?
8. Um wieviel gehen Tourenboote durchschnittlich tiefer als Sportboote?
9. Geben Sie für alle Mannschaften aus, an wievielen Regatten sie bereits teilgenommen haben und wieviele Punkte sie dort erzielt haben.
10. Welches Land bietet die längste Wettfahrtsstrecke und hat zusätzlich nicht die kürzeste?
11. Wie heißt der Trainer, der die Mannschaft mit den meisten Punkten trainiert hat?
12. Geben Sie für JEDE Mannschaft aus, wieviele Punkte Sie bei der 'Bodenseeregatta' in 'Oesterreich' erzielt haben.
13. Geben Sie die ID und den Namen jener Sportboote aus, die mindestens an zwei Regatten Teil genommen haben, aber keiner Mannschaft zugewiesen sind.
14. Geben Sie die Regatten (Name, Jahr und Land) aus, die über die kürzeste Distanz gehen.

Java & JDBC

Schreiben Sie einen Java Client, der eine JDBC-Verbindung zur Datenbank herstellt und AUTOCOMMIT ausschaltet. Realisieren Sie eine GUI, die einfache CRUD-Befehle auf die Boote des Vereins implementiert (keine explizite SQL-Eingabe). Verwenden Sie dabei auf jeden Fall eine JTable, die auch eine grafische Veränderung der Datensätze erlauben soll.

Als Erweiterung (Bonuspunkte) soll bei der Anzeige der Boote die Möglichkeit der Sortierung und Filterung über ein neues SQL-Kommando bereitgestellt werden. Auch hier soll nicht der Benutzer die SQL-Befehle eingeben, sondern es muss die Funktionalität über entsprechende GUI-Elemente realisiert werden!

Ermöglichen Sie die gleichzeitige Verbindung von mehreren Clients auf die Datenbasis. Implementieren Sie dabei eine transaktionell, gesicherte Erstellung und Änderung von Wettfahrten. Beachten Sie dabei, dass der Punktestand der einzelnen Wettfahrten laufend und von mehreren Clients gleichzeitig aktualisiert werden könnte. Stellen Sie für die Eingabe der Wettfahrt und Mannschaft eine einfache grafische Möglichkeit zur Verfügung.

Abgabe

Die Abgabe ist am 20. März 2015 um 08:00 per elearning zu tätigen. Es wird ein Protokoll (Metaregeln), die SQL-Files (drop.sql, create.sql, start.sql, insert.sql, insert-*.sql, queries.sql) in einem eigenen Verzeichnis. Dies alles wird in einem ausführbaren JAR-Archiv erwartet. Die Abgabe wird mit einem Prüfungsgespräch validiert, wobei auf eine eigenständige Lösung geachtet wird - kopierte Lösungen führen zu einer negativen Benotung! Quellen sollen somit auf den theoretischen Background und auf die Manuals beschränkt sein. Teile von bestehendem JDBC-Code aus dem Internet und vorgefertigte SQL-Abfragen dürfen somit nicht verwendet werden. Im Zweifelsfall ist es notwendig die Lehrkräfte um Freigabe von Quellen zu bitten. Das Beispiel soll für eine Postgresql 9.4 Umgebung implementiert werden.

Die Inserts sollen mindestens 10.000 Einträge enthalten. Es bleibt Ihnen überlassen, ob Sie einen selbstgeschriebenen Generator oder ein externes Tool verwenden möchten. Auf jeden Fall muss Ihre Vorgehensweise gut dokumentiert und nachvollziehbar sein. Die Daten sollen so nahe wie möglich der Wirklichkeit entsprechen, um entsprechende Testfälle und Performancetests auf der Datenbank starten zu können (Person 1..10000 sind somit keine erwünschten Datensätze).

Bei Problemen mit dem Create-Script und den Inserts kann ein Example-Set bei den Lehrenden angefordert werden. Dies muss aber für jeden Kandidaten einzeln geschehen! Anfragen bitte immer per eMail an BEIDE Lehrer.

(c) Markus Pichlmair; adaptiert bei Michael Borko und Erhard List

Durchführung

Da ich diese Aufgabe erst später angefangen habe, da ich im Halbjahreszeugnis positiv bin und noch Zeit war bin ich jetzt zum Abgabetermin nicht vollständig fertig. Ich habe jedoch in wenigen Tagen einen guten Fortschritt erzielen können. Zählt dieses Bsp. als vollwertiges, weshalb ich nun mit Abzügen rechnen muss oder nur als Bonuspunkt?

Zu allererst standen die Queries am Plan. Ich habe eine Testumgebungs-Datenbank mit nur einigen wenigen Werten implementiert, um sofort starten zu können, daher kommt, dass die Abfragen auf die richtige Datenbank erst angepasst werden müssen, jedoch weiterhin in der Funktion richtig sind.

Im SQL-File wurde dokumentiert, woran ich z. T. gescheitert bin oder welche Sonderheiten aufgetreten sind. Am liebsten hatte ich es aber doch, wenn nur „DONE!“ dort stehen konnte.

Die DB wird bereits korrekt erstellt, und derweil mit jeweils 100 Inserts gefüttert.

Zur GUI:

Ist im Grunde genommen der Exporter, in der derzeitigen Version noch angepasst, dass einige Werte in die GUI geschrieben werden. Die Verbindung zu Postgres funktioniert. Mein Plan → Mit ActionListener die Abfragen in die DB schicken und das ResultSet wiederum auf die GUI auszugeben, ging sich jedoch nicht mehr aus, genauso wie die Doku ☺

Auch wenn's dann nicht mehr benotet wird, ich lade den Stand jetzt (Sonntag, 17:50) hoch und mache weiter, Übung ist Übung.

Detaillierte Arbeitsaufteilung (Aufwandsabschätzung, Endzeitaufteilung)

Aufwandabschätzung

Diese Aufgabe ist je nach Fortschritt in 20-30 Arbeitsstunden zu implementieren.

Endzeitaufteilung

Derweil (noch nicht abgeschlossen) sind mehr als 20 Stunden vergangen.

Fazit

Arbeitsdurchführung

Resultate

Quellenangaben: