



Projektbeschreibung

Projekttitle: Fußball WM DB

Einführung in Datenbanksysteme
Datenbanken für die Bioinformatik

Heinz Schweppe, Katharina Hahn

Überblick

Das diesjährige Projekt im Rahmen der Datenbankeinführungsveranstaltung legt den Fokus auf relationale Datenmodellierung. Die Erfahrung aus den vergangenen Semestern hat gezeigt, dass die Verwendung von Webframeworks (Struts, JSF, etc.) eine zu flache Lernkurve aufweist. Die Einarbeitung in ein solches Framework nimmt zu viel Zeit ein, die dann bei der Behandlung der in Bezug auf diese Veranstaltung relevanten Themen fehlt. Um diesen Themen (siehe auch Lernziele) mehr Raum zu geben, haben wir uns entschieden, auf eine graphische (Web-) Oberfläche der Anwendung als Anforderung zu verzichten. Sie können natürlich dennoch eine Benutzeroberfläche erstellen, wenn Sie das möchten.

Lernziele des Datenbankprojekts sind die Modellierung eines Datenbankmodells aus komplexen, vorgegebenen Anforderungen (die Raum zur Interpretation lassen), die Umsetzung in ein konkretes Datenbankschema, sowie die Umsetzung und Handhabung von server-seitiger Applikationslogik (Stored Procedures).

Einführung

Dieses Jahr findet vom 11. Juni bis 11. Juli die Endrunde der Fußball Weltmeisterschaft der Männer in Südafrika statt. Dort spielen insgesamt 32 Mannschaften um den Titel. In der Gruppenphase des Turniers spielen die Teams in 8 Gruppen jeweils gegeneinander. Die besten beiden einer Gruppe qualifizieren sich für die 2. Phase des Turniers, in der dann die ersten 4 Plätze im einfachen K.O.-Modus ausgespielt werden. Die Endrunde der Fußball Weltmeisterschaft gehört neben den Olympischen Spielen zu den bedeutendsten Sportereignissen der Welt. In diesem Projekt wird ein System entwickelt, das die Modalitäten von Fußball-WM-Turnieren abbildet. Die funktionalen Ziele des Systems sind im folgenden knapp formuliert:

- „Fußball WM DB“ basiert auf dem etablierten Modus für WM Endrunden mit Gruppen- und K.O. Phase. An einem Turnier nehmen 32 Mannschaften teil.
- Die „Fußball WM DB“ dient zur Datenerfassung und -speicherung vor einem solchen Turnier. Bevor das Turnier stattfindet, sollen bereits die Spielpläne eines Turniers abgebildet werden können.
- Zusätzlich werden Daten während des Turniers, die den direkten Turnierablauf betreffen (u.a., Spiele, Torerfolge, Gewinner, etc.) abgebildet, so dass aktuelle Informationen während des Turniers eingefügt und eingesehen werden können.

Lesen Sie die Spezifikation aufmerksam. Es werden einige vereinfachende Annahmen gemacht, um den Aufwand des Projekts zu reduzieren.

Projektaufgaben

Da bekannt ist, dass das Datenbankprojekt stets in die Klausurvorbereitungszeit fällt, wird es bereits vor dem Datenbank-Klausurtermin veröffentlicht und die Abgabe auf die Woche nach Vorlesungsende (**31. Juli**) verschoben. Damit steht Ihnen ausreichend Zeit für eine vernünftige Bearbeitung des Projekts zur Verfügung. **Verspätete Abgaben können nicht gewertet werden!** Bearbeiten Sie das Projekt stets in **3er Gruppen**, bearbeiten Sie die Aufgaben **nicht alleine**, suchen Sie ggf. in den Tutorien oder im Forum (s.u.) Projektpartner. Gruppen mit mehr als 3 Teilnehmern sind nicht zulässig. Bearbeiten Sie folgende Aufgaben:

1. Studieren Sie die funktionalen und nicht-funktionalen Anforderungen, die sich aus der unten stehenden Spezifikation ergeben. Es ist der Normalfall, dass das Ergebnis einer Anforderungsanalyse noch Interpretationsmöglichkeiten offen lässt. Diskutieren Sie innerhalb Ihrer Projektgruppe die für Sie unklaren Anforderungen. Nutzen Sie insbesondere auch das eingerichtete Forum unter <http://foren.spline.inf.fu-berlin.de/viewforum.php?f=420> um etwaige Unklarheiten bezüglich der Anforderungen zu klären. Natürlich können Unklarheiten auch in den Tutorien diskutiert werden. Dokumentieren Sie ihre Entscheidungen (ggf. mit Begründung) in Ihren Unterlagen.
2. Entwerfen Sie ein Datenbankschema, das alle beschriebenen Anforderungen abbildet. Modellieren Sie zunächst auf ER-Ebene (ggf. in UML Notation). Überführen Sie dann Ihr ER-Modell in ein relationales Modell. Nutzen Sie sowohl für die Modellierung auf ER-Ebene, als auch für die Darstellung des relationalen Modells geeignete Werkzeuge. Für die Modellierung auf ER-Ebene können Sie bspw. MS Visio (Windows) oder Dia (Linux) verwenden. Für die Darstellung des relationalen Modells bieten sich der DBDesigner (Windows/Linux) oder die MySQLWorkbench (Windows/Linux) an. **Als Abgabe wird keine Handzeichnung der Modelle akzeptiert!** Die Abgabe des Modells erfolgt sowohl elektronisch als auch als Ausdruck der Dokumente. Abzugeben ist sowohl Ihr ER-Modell, als auch Ihr relationales Modell.

Wichtiger Hinweis: Dokumentieren Sie alle *wichtigen* Designentscheidungen bzgl. des Datenbankschemas und des Objektmodells detailliert in *zwei* separaten Dokumenten. Gibt es z.B. mehrere sinnvolle Alternativen eine Gegebenheit abzubilden, so dokumentieren Sie die Vor- und Nachteile der Alternativen. Begründen Sie warum Sie sich für eine bestimmte Alternative entschieden haben. (Z.B.: Warum wurde X als eigene Entität abgebildet und nicht als Attribut von Y?)

3. Erstellen Sie in dem Datenbanksystem Ihrer Wahl (*Oracle oder Postgres, andere sind nicht zulässig!*) das Ihrem Modell entsprechende Datenbankschema. Nutzen Sie dazu ggf. die Funktionalität Ihres Modellierungswerkzeugs.
4. Entwerfen und Implementieren Sie ein Objektmodell in Java, das die geforderten Anforderungen erfüllt und natürlich zu Ihrem Datenbankentwurf passt. Dokumentieren Sie ihr Objektmodell in UML.

Es steht Ihnen frei, alternativ das Hibernate Framework als OR-Mapper (ORM) und die entsprechenden Codegenerierungswerkzeuge zu verwenden. Bitte beachten Sie, dass für Hibernate Ihnen seitens der Tutoren kein Support zur Verfügung steht. Dokumentieren Sie

in diesem Fall in einem separaten Dokument detailliert ihre Vorgehensweise (u.a., welche Art des Mappings), sowie das Mapping zwischen relationalem Modell und Objektmodell. Dokumentieren Sie in jedem Fall Ihr Datenbankschema grafisch. Geben Sie ggf. an, an welchen Stellen Sie nach Anwendung der Generierungswerkzeuge manuell nachgebessert haben.

5. Implementieren Sie den für die UseCases nötigen Datenbankzugriff in separaten DataAccessObjects (DAO, siehe <http://java.sun.com/blueprints/patterns/DAO.html>) in einem DAOLayer. (Bei Verwendung eines ORM, wird in der DAO in der Regel kein JDBC-SQL geschrieben, sondern es werden die Möglichkeiten des ORM genutzt. Beachten Sie, dass die Hibernate-Werkzeuge einen Code-Generator für DAOs bereitstellen.)
6. Implementieren Sie die Funktionalität der geforderten Anwendungsfälle (siehe unten). Eine mögliche Herangehensweise ist, die Anwendungsfälle als *ServiceLayer* zu kapseln (siehe <http://www.martinfowler.com/eaCatalog/serviceLayer.html>). In diesem Fall existiert für den fiktiven Anwendungsfall „Erstellen“, „Löschen“ und „Modifizieren“ einer Spielerin/eines Spielers¹ (Player) eine Klasse *PlayerServices* im *ServiceLayer*, und als DAO eine Klasse *PlayerDAO*. *PlayerServices* hat als Methoden beispielsweise *createSpieler(String lastname, String firstname, ...)*, die die *PlayerDAO* nutzt, um die Spieler zu persistieren. Es sind selbstverständlich alternative Herangehensweisen zu dem beschriebenen *ServiceLayer* möglich.
7. Implementieren Sie eine (oder mehrere) Klasse(n), die die geforderten Anwendungsfälle testweise aufruft. Es ist Ihnen freigestellt, ob Sie dazu existierende Frameworks (z.B. JUnit) oder selbstgeschriebene Testfälle verwenden.
8. Um Ihre Anwendung sinnvoll testen zu können, müssen Sie sich passende Testdaten generieren. Auf Grund unterschiedlicher Modellierungsmöglichkeiten, ist es nicht möglich, einen kompletten Testdatensatz für das gesamte Modell vorzugeben. Es werden auf der Projektwebseite im Laufe des Projekts Testdaten für einige Objekte, sowie Generatoren für einige Objekte zur Verfügung gestellt. Modifizieren Sie diese ggf., um sich für ihr Modell passende Testdaten generieren zu lassen. Ergänzen Sie fehlende Daten. Laden Sie die Testdaten in Ihre Datenbank. Nutzen Sie dazu z.B. die Bulk-Load Fähigkeiten es gewählten Datenbanksystems (SQL-Loader unter Oracle bzw. COPY in Postgres).

Projektabgabe

Abgabe: Abzugeben sind die im folgenden aufgeführten Dateien und Dokumente. Fassen Sie *alle* Dateien in einem ZIP-Archiv zusammen. Benennen Sie die ZIP-Datei nach dem Schema <VornameTutor>_<NachnamenGruppenmitglieder>, z.B. Konrad_GötheSchillerLessing.zip. Folgende Dateien *müssen* in dem Archiv enthalten sein:

- Eine Textdatei mit Namen <namen.txt>, die eine Liste Ihrer Namen, Vornamen und Matrikelnummern enthält.
- Ihr ER-Modell und das relationale Modell als PDF oder PNG/JPG. (Wenn Sie mit Hibernate die Vorgehensweise gewählt haben, zuerst ein Objektmodell zu erstellen und sich daraus das relationale Modell generieren haben lassen, müssen Sie kein ER-

1 Zu Gunsten der Lesbarkeit wird im Folgenden auf die explizite Nennung der weiblichen Form verzichtet.

Modell abgeben. In diesem Fall ist trotzdem eine grafische Dokumentation des Datenbankschemas abzugeben. Vermerken Sie in diesem Fall Ihre Vorgehensweise in den entsprechenden Dateien.)

- Ihr Objektmodell als PDF oder PNG/JPG.
- Zwei separate Dokumente mit Erläuterungen Ihrer Designentscheidungen bzgl. Daten und Objektmodell als PDF.
- Ein SQL-Skript, das Ihr Datenbankschema in der von Ihnen gewählten Datenbank (Oracle, Postgres) erzeugt. Das Skript soll zunächst alle Ihre Tabellen löschen und dann neu erstellen. Nach Anlegen der Tabellen, sollen Testdaten in die entsprechenden Tabellen eingefügt werden, z.B. mittels INSERT. Der Name des Skripts muss *<create.sql>* lauten.
- Eine *.jar-Datei, die Ihre vollständige Implementierung enthält. Achten Sie darauf, dass ihre *.jar-Datei den Java Quelltext enthält!

Zusätzlich sind folgende Dokumente in *ausgedruckter* Form abzugeben:

- Grafische Darstellung des relationalen und des ER-Modells
- Grafische Darstellung des Objektmodells
- Dokumentationen des ER- und des Objektmodells

Projektbeschreibung

Projektziel ist die Modellierung und Implementierung eines Systems zur Planung und Verwaltung von Fußball-WM-Turnieren.

Akteure

Folgende Akteure sind für die Verwendung des Systems einzuplanen.

Ergebnisverwalter ist eine natürlicher Person, die für die Ergebnispflege einzelner oder mehrerer Gruppen eines Turniers zuständig ist. D.h. ein Ergebnisverwalter pflegt die Informationen zu den jeweiligen Spielen (z.B., Einsatzzeiten, Torschützen, etc.). Eine Gruppe kann auch von mehreren Ergebnisverwaltern verwaltet werden.

Turnierverwalter ist eine natürliche Person, die ein gesamtes Turnier anlegen und verwalten kann. Ein Turnierverwalter kann sämtliche seinem Turnier zugeordneten Gruppen, Spiele und Mannschaften verwalten. Er hat somit ebenfalls alle Rechte, die ein Ergebnisverwalter einer Gruppe seines Turniers hat.

Administrator: Ein Administrator ist eine natürliche Person, die Rechte über das gesamte System verfügt.

Externe Personen können lediglich Spielpläne, Paarungen, Gruppen, Ergebnisse, Mannschafts- und Spieler-Informationen betrachten. Sie haben keine Editiermöglichkeiten.

Funktionale Anforderungen: Beschreibung der Objekte und verbundener Anwendungsfälle

Spezifikation Turnier

Ein Turnier ist eine abgeschlossene Einheit, in der 32 Mannschaften zwischen einem bestimmten Anfangs- und Endzeitpunkt um den WM-Titel spielen. Der Anfangszeitpunkt ergibt sich aus dem Datum des ersten Spiels, der Endzeitpunkt aus dem Datum, an dem das Finale stattfindet. Ein Turnier besteht aus einer Gruppenphase mit 8 Gruppen jeweils à 4 Mannschaften und einer K.O.-Runde. Die K.O.-Runde besteht aus Achtel-, Viertel-, Halbfinalspielen, dem Finale sowie dem Spiel um den dritten Platz. Damit hat jedes Turnier nach Ablauf eine Siegermannschaft, eine zweit- und eine drittplatzierte Mannschaft. Jedes Turnier hat einen Turnierplan, in dem die Ansetzung der Partien (wer, wann, wo) festgehalten wird. Vor Beginn des Turniers bzw. vor Ablauf der Gruppenphase eines Turniers kann es vorkommen, dass die Mannschaften eines Spiels noch nicht feststehen. In diesem Fall sollen sinnvolle Platzhalter vorhanden sein (z.B., „Sieger Gruppe A“, „Verlierer Spiel 42“, etc...). Die Spiele eines Turniers werden in verschiedenen Stadien eines Landes ausgetragen.

Hinweis: Diskutieren Sie, ob Informationen zu unterschiedlichen Turnieren im gleichen Schema abgebildet werden sollen oder ob Sie z.B. für jedes Turnier ein eigenes Schema erstellen.

Spezifikation Gruppe

Eine Gruppe ist jeweils eindeutig einem Turnier zugeordnet. Eine Gruppe besteht aus 4 Mannschaften. In der Gruppenphase spielen die Teams einer Gruppe jeweils gegeneinander. Die Besetzung der Partien der K.O.-Runde ergibt sich aus den Platzierungen der Teams innerhalb der einzelnen Gruppen. Jede Gruppe wird durch folgende Informationen beschrieben:

1. Gruppenkennung (üblich ist eine Kennung durch einzelne Buchstaben, z.B. Gruppe A)
2. Mannschaften der Gruppe
3. Spiele innerhalb der Gruppe
4. Ergebnistabelle der Gruppe (Anzahl Siege, Unentschieden, Niederlagen, Punkte, Anzahl geschossene und Gegentore aller Mannschaften der Gruppe)

Spezifikation Mannschaft (Team)

Eine Mannschaft (oder auch Team), die an einem Turnier teilnimmt, besteht aus maximal 23 Spielern und dem Betreuerstab. Ein Team spielt für ein Land. (Für ein Land gibt es maximal eine Mannschaft.) Eine Mannschaft wird vor Beginn eines Turniers zufällig einer Gruppe des Turniers zugelost. Daraus ergibt sich die Zuordnung einer Mannschaft zu den Spielen der Vorrunde. Abhängig von ihrem Abschneiden wird eine Mannschaft den entsprechenden Partien der Hauptrunde zugeordnet. Jeder Mannschaft wird durch folgende Angaben beschrieben:

1. Name des Teams (entspricht dem Ländername)
2. Bild der Landesflagge (optional)
3. Zugehörige Spieler
4. Betreuerstab

- a) Trainer
- b) Assistenztrainer
- c) Mannschaftsarzt

Spezifikation Spieler

Ein Spieler spielt für eine Nation, d.h. für eine Mannschaft. Hat ein Spieler mehrere Staatszugehörigkeiten, kann er trotzdem für maximal eine Mannschaft antreten. Hat er sich einmal für ein Team entschieden, so ist dies endgültig und er kann zu einem späteren Zeitpunkt (im gleichen oder in verschiedenen Turnieren) *nicht mehr* für ein anderes Land antreten. Zu einem Spieler werden folgende Informationen hinterlegt bzw. während eines Turniers laufend ergänzt:

1. Name und Vorname des Spielers
2. Rufname
3. Geburtsdatum
4. Größe
5. Gewicht
6. Verein (optional)
7. Einsatzzeiten in Spielen
8. Torerfolge
9. Eigentore

Beachten Sie, dass es Spieler gibt, die keinen expliziten Rufnamen haben, umgekehrt aber auch Spieler, die nur einen Rufnamen führen (vgl. CACAU, für den kein Vor- und Nachname in der offiziellen Kaderliste des DFB verzeichnet ist).

Spezifikation Person (Betreuer + Akteure)

Zusätzliche Personeninformationen, die verwaltet werden müssen, sind die Akteure (Administrator, Turnierverwalter, Ergebnisverwalter) und die Personen des Betreuerstabs einer Mannschaft. Zu jeder solcher Person gehören die Attribute Name, Vorname, ihre Rolle (z.B. Mannschaftsarzt, Ergebnisverwalter, usw.), bei Akteuren zusätzlich ihre Email-Adresse.

Spezifikation Spiel

Ein Spiel hat eine eindeutige ID und findet entweder in der Gruppenphase oder in der K.O.-Runde eines Turniers statt. Ein Spiel in der Vorrunde besteht aus 2 Halbzeiten à 45 Minuten und hat entweder einen Sieger oder die beiden Mannschaften spielen unentschieden. Für einen Sieg in der Gruppenphase erhält ein Team 3 Punkte, bei einem Unentschieden gibt es je einen Punkt für beide Teams. In der K.O.-Runde hat jedes Spiel einen Sieger. Steht es nach 90 Minuten unentschieden, so gibt es 2 mal 15 Minuten Verlängerung und ggf. Elfmeterschießen. Folgende Informationen werden zu einem Spiel abgelegt:

1. Die beiden Teams, die gegeneinander spielen.
2. Ein Stadion, in dem das Spiel stattfindet.
3. Einen Zeitpunkt (Datum und Uhrzeit des Anpiffs), an dem das Spiel stattfindet.
4. Eine Zuschauerzahl, die das Spiel live im Stadion verfolgt.
5. Gefallene Tore
6. Einsätze der Spieler

Spezifikation Tor

Ein Tor wird von einem Spieler in einem Spiel geschossen. Ein Tor wird durch folgende Informationen beschrieben:

1. Das Spiel, in dem es gefallen ist.
2. Die Minute, in der es gefallen ist.
3. Der Spieler, der es geschossen hat (Torschütze).
4. Das Team, für das es gefallen ist.
5. Handelt es sich bei dem Tor um ein Eigentor?
6. Ist das Tor nach einem Strafstoß gefallen?

Spezifikation Einsatz

Spielt ein Spieler für seine Mannschaft, so wird in der Datenbank abgelegt, in welchem Spiel er von welcher bis zu welcher Minute gespielt hat. Beachten Sie, dass ein Spiel ggf. mit Nachspielzeit die Spielzeit von 90 (bzw. 120) Minuten überschreiten kann. Verhält sich ein Spieler unsportlich (z.B. indem er andere Spieler foult) erhält er eine Verwarnung (sog. „gelbe Karte“). Erhält er in einem Spiel eine weitere gelbe Karte, so bekommt er automatisch einen Platzverweis („rote Karte“). Bei grob unsportlichem Verhalten, kann er auch direkt eine rote Karte bekommen, ohne vorher „gelb gesehen“ zu haben. In dem System werden die Informationen, wann ein Spieler eine gelbe bzw. rote Karte in welchem Spiel bekommen hat, gespeichert.

Spezifikation Stadion

Jedes Turnier wird in bestimmten Stadien ausgetragen. Diese werden dem Turnier vor Beginn zugeordnet. Zu einem Stadion werden folgende Informationen abgebildet:

1. Name
2. Optional der Verein, dem es gehört
3. Kapazität (die maximale Anzahl an Zuschauern)
4. Stadt
5. Koordinaten als *latitude* und *longitude*

Nichtfunktionale Anforderungen:

Sicherheit

Das System nutzt Passwort-Authentifizierung zur Anmeldung aller Akteure (außer Externen). Ein Nutzer ist über seine Email-Adresse eindeutig gekennzeichnet. Das heißt, eine Kombination aus Email-Adresse und Passwort genügt zur Authentifizierung. Passwörter werden verschlüsselt gespeichert. Berechtigungen der einzelnen Akteure werden durch ein Rollenkonzept unterstützt. Berechtigungen (z.B. Lesen, Ändern, ...) können direkt einem Nutzer erteilt oder einer Rolle zugewiesen werden. Ein Nutzer kann beliebig vielen Rollen angehören. Eine Berechtigung bezieht sich immer auf ein konkretes Objekt des Systems. Beispielsweise kann die Berechtigung „Editieren von Spielen“ bezogen auf die „Spiele der

Gruppe A“ der FIFA Fußball WM 2010 an die Rolle „Turnierverwalter WM 2010“ oder „Gruppenverwalter Gruppe A WM 2010“ vergeben werden.

Beachten Sie, dass Sie dazu die vom System verwalteten Objekte zentral erfassen müssen, um explizite Berechtigungen vergeben zu können. Berechtigungen können zunächst nur für folgende Objekte vergeben werden: Turniere, Gruppen und Spiele. Die Berechtigungen sind dabei wie folgt hierarchisch aufgebaut: Wenn eine Rolle für ein Turnier eine Berechtigung erhält, so hat sie diese ebenfalls automatisch für alle Gruppen und Spiele des Turniers. Gleiches gilt für die Berechtigung für eine Gruppe: Diese beinhaltet die Berechtigung für alle Spiele dieser Gruppe.

Beachten Sie, dass es sich bei den Berechtigungs-Rollen und den Akteuren des Systems (bspw. Turnierverwalter) um orthogonale Konzepte handelt.

Anwendungsfälle

Bei der Implementierung der Anwendungsfälle können Sie die Überprüfung der Berechtigungen vernachlässigen. Die Logik der folgenden Anwendungsfälle können Sie wahlweise als Stored Procedure in PL/SQL bzw. PL/pgSQL implementieren, die aus ihrer Anwendung heraus aufgerufen werden, oder als Java-Methoden, die die entsprechenden Datenbankzugriffe des *DAO Layers* verwenden. Wenn Sie sich dafür entscheiden, die Anwendungsfälle als Stored Procedures umzusetzen, implementieren Sie den Aufruf Ihrer Prozeduren in den spezifizierten Methoden in geeigneter Weise. Erweitern Sie ggf. die Signaturen der genannten Methoden um benötigte Informationen (z.B. Turnier).

- Einsatz eintragen:
Der Einsatz eines Spielers zu einem Spiel wird eingetragen. Implementieren Sie dazu an geeigneter Stelle die Methode *insertPlayerToMatch(Player player, Match match)*; erweitern Sie die Signatur der Methode in geeigneter Weise, z.B. um die Werte der Einsatzzeiten (*von* und *bis*). Die Methode soll eine passende *Exception* werfen, wenn die Mannschaft des Spielers nicht an dem Spiel teilgenommen hat oder ungültige von- bzw. bis-Werte angegeben wurden.
- Tor eintragen:
Es wird ein Torerfolg (oder ein Eigentor) eines Spielers eingetragen. Implementieren Sie dazu an geeigneter Stelle die Methode *insertGoal(Goal goal, Player player, Match match)*. Werfen Sie eine *Exception*, wenn der Spieler nicht an diesem Spiel teilgenommen hat.
- Ergebnis ausgeben:
Das Ergebnis eines Spiels soll in der Form *<Mannschaft 1> – <Mannschaft 2> <x>:<y>*, wobei *x* die Anzahl der Tore von *Mannschaft1* und *y* die Tore von *Mannschaft2* beziffern. Dazu müssen die Torerfolge der beiden Mannschaften in geeigneter Weise aus der Datenbank gelesen werden. Beachten Sie, dass ein Eigentor als Torerfolg für die gegnerische Mannschaft zählt. Implementieren Sie dazu an geeigneter Stelle eine Methode *getResult(Match match)*.
- Gruppentabelle berechnen:
Die Methode gibt für eine gegebene Gruppe die aktuelle Tabelle aus. Beachten Sie, dass dazu nicht alle Spiele der Gruppe bereits gespielt worden sein müssen. Die Mannschaften werden sortiert nach Punkten, dann nach Tordifferenz (Anzahl geschossene Tore – Anzahl Gegentore), dann nach der Anzahl geschossener Toren, dann nach direktem Vergleich (falls vorhanden), dann alphabetisch nach Namen ausgegeben in einer Tabelle, jeweils mit ihren Punkten, Anzahl Spielen,

geschossenen Toren und Gegentoren. Implementieren Sie dazu an geeigneter Stelle die Methode *printStandings(Group group)*.

- Sieger ausgeben:

Die Methode gibt nacheinander nummeriert den Sieger, Zweitplatzierten und Drittplatzierten eines Turniers aus. Implementieren Sie dazu an geeigneter Stelle die Methode *weAreTheChampions(Tournament tournament)*.

Der folgende Anwendungsfall muss in jedem Fall server-seitig, d.h. als Stored Procedure in PL/SQL bzw. PL/pgSQL implementiert werden, der aus Ihrer Anwendung heraus aufgerufen wird. **Die Implementierung der Logik als Java-Methode mit SQL-Statements (ggf. HQL) kann für diesen Fall nicht gewertet werden.**

- Turnier anlegen:

Es soll eine neues Turnier mitsamt Turnierplan angelegt werden. Je nach Anzahl der Mannschaften müssen dazu entsprechende Gruppen angelegt werden. Werden keine Informationen zu den teilnehmenden Mannschaften bei Turniergenerierung mit angegeben, so wird ein Turnierplan mit generischen Bezeichnungen (z.B. „Mannschaft A1“, „Sieger Spiel 27“) gefüllt. Es liegt Ihnen frei, zusätzliche Informationen zu dem Zeitrahmen des Turniers mitanzugeben oder die Spiele an zufälligen Tagen zu generieren. Implementieren Sie dazu an geeigneter Stelle eine Methode *createChampionship(String name)*, die als Eingabe ggf. zusätzlich eine Menge von Mannschaften erhält.

Viel Spaß beim Projekt!