



**FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI**

Fotbalová liga

Semestrální práce KIV/DB2

Vypracoval: Libor Váchal
Osobní číslo: A14N0092P
Email: vachall@students.kiv.zcu.cz
Datum: 9.5.2015

Zadání

Cílem této práce je navrhnout a vytvořit informační systém pro evidenci fotbalové ligy. Systém nabídne možnost evidovat fotbalové zápasy a jejich průběh. Po vstřelení gólu budou automaticky aktualizovány statistiky hráčů a také průběžné skóre zápasu. Po skončení zápasu bude zaktualizována tabulka pořadí týmů ligy.

Fotbalovou ligu hraje několik týmů. Každé dva týmy se v jednom ročníku utkají dvakrát, jednou na domácím hřišti a jednou na hřišti hostujícího týmu. Za vítězství si každý tým připsá 3 body, za remízu 1 bod a za prohru žádný. Každý zápas se hraje na dva poločasy, jeden poločas trvá 45 minut. V každém týmu hraje několik hráčů, kteří zastávají různé posty – brankář, obránce a útočník. Na začátku každého zápasu nastoupí za každý tým 11 hráčů – 1 brankář a 10 hráčů do pole (obránci a útočníci). Během zápasu mohou padat góly (do brány soupeře i do vlastní), které střílí hráči na hřišti. Po skončení zápasu se přidělí body a zaktualizuje se pořadí a skóre v tabulce týmů. Týmy budou seřazeny podle těchto kritérií:

1. počet dosažených bodů (více bodů lepší umístění),
2. při rovnosti bodů rozhoduje větší rozdíl mezi vstřelenými a obdrženými brankami,
3. při rovnosti bodů a stejném rozdílu branek rozhoduje větší počet vstřelených gólů,
4. jinak rozhoduje abecední uspořádání podle názvu týmu (dodatečné pravidlo).

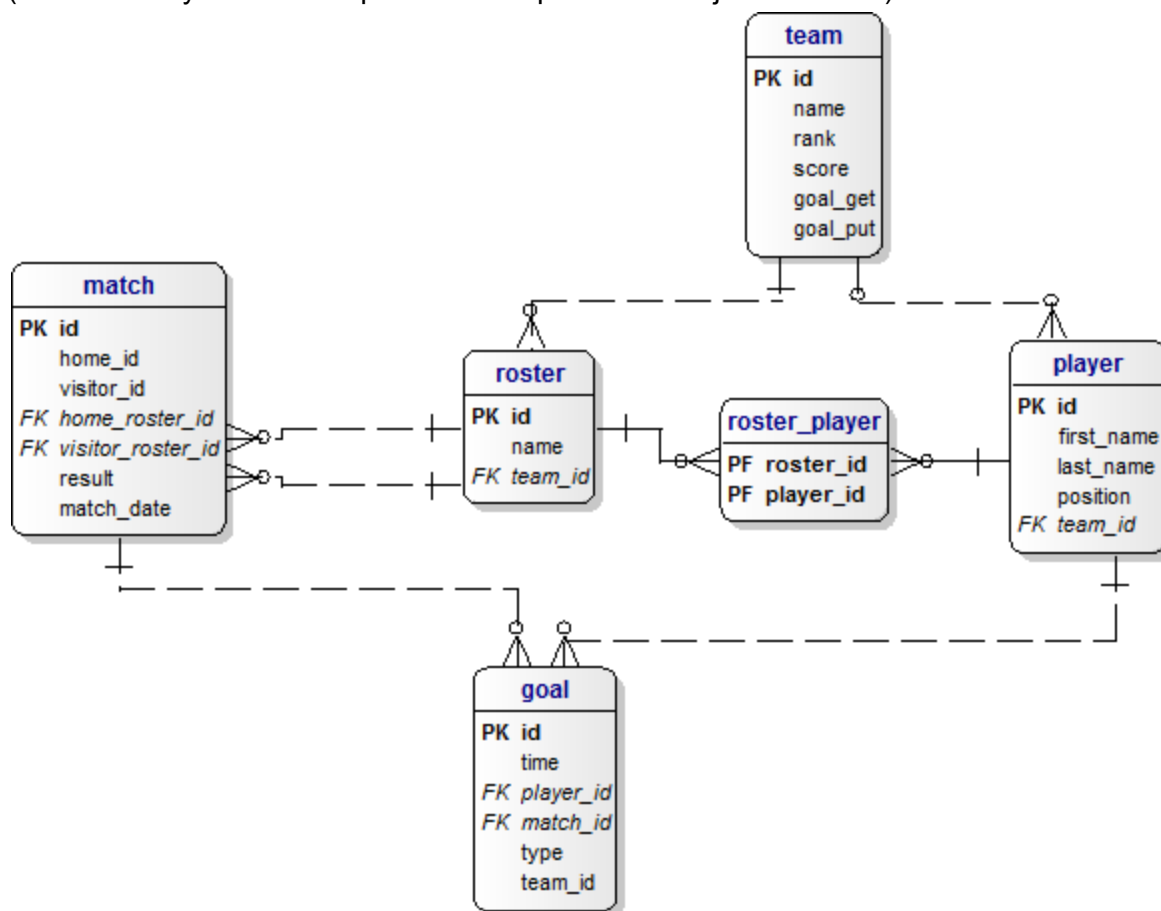
Navrhněte a realizujte datový model splňující 3NF (normální formu) pro evidenci fotbalové ligy. Dále realizujte potřebné uložené procedury, funkce a trigger, které budou průběžně aktualizovat skóre běžícího zápasu po vstřelení branky a také aktualizovat pořadí týmů v lize po skončení zápasu. Také pohlíďte, aby branku mohl vstřelit jen hráč, který je zrovna na hřišti a také situaci, aby za jeden tým mohlo být na hřišti maximálně 11 hráčů. Systém též nabídne celou řadu výpisů/reportů, které mohou být filtrovány podle zadaných kritérií.

V databázi budou evidována minimálně tato data:

- **tým** – skupina hráčů, kteří hrají spolu a snaží se vyhrát ligu. Každý tým bude obsahovat informaci o průběžném skóre odehraných zápasů, počtu získaných bodů a aktuálním pořadí v tabulce.
- **hráč** – základní údaje o hráči, který hraje za daný tým. Pro jednoduchost budeme předpokládat, že složení týmu se v čase nemění. U každého hráče bude uvedeno, na jakém postu hraje a kolik celkem nastřílel gólů.
- **zápas** – základní informace o tom, kdy a kde se dva týmy utkaly a jak to dopadlo.
- **soupiska** – seznam 11 hráčů za jeden tým, který hraje konkrétní zápas.
- **branka** – základní informace o tom, ve kterém utkání a v jakém čase padl gól. Chceme znát jeho autora a také typ gólu (branka v utkání, pokutový kop, vlastní branka, ...)

Datová analýza

Datový model obsahuje celkem 6 tabulek, 5 datových a 1 pomocnou, řešící M:N vazbu. Datové tabulky vychází přímo ze zadání, pomocná tabulka řeší vztah mezi rozpisky a hráče (hráč může být na více rozpiskách a rozpiska obsahuje více hráčů).



Match

id - identifikátor, **home_id** - id domácího týmu, **visitor_id** - id hostujícího týmu, **home_roster_id** - id rozpisky domácího týmu, **visitor_roster_id** - id rozpisky hostujícího týmu, **result** - výsledek zápasu (0, 1, 2, null), **match_date** - datum zápasu

Team

id - identifikátor, **name** - název týmu, **rank** - pořadí v tabulce, **score** - počet bodů, **goal_get** - počet inkasovaných gólů, **goal_put** - počet vstřelených gólů

Goal

id - identifikátor, **time** - minuta ve které gól padl, **player_id** - hráč, který dal gól, **team_id** - id týmu, který dal gól, **match_id** - id zápasu ve kterém padl gól, **type** - typ (standartní, roh, penalta, hlavička)

Player

id - identifikátor, **first_name** - jméno, **last_name** - příjmení, **position** - hráčův post (u, o, z, b), **team_id** - id týmu za který hráč kope

Roster

id - identifikátor, **name** - název rozpisky, **team_id** - id týmu, ke kterému rozpiska patří

Roster_player

roster_id - id rozpisky, **player_id** - id hráče

Funkční analýza

Liga

1. Zobrazení probíhajících a ukončených zápasů
 - a. `SELECT * FROM TEAM`
 - b. `SELECT * FROM MATCH`
2. Vytvoření zápasu
 - a. `INSERT INTO MATCH VALUES ('', ".$homeId.", ".$visitorId.", ".$homeRosterId.", ".$visitorRosterId.", '', ' ".$date."')`
 - b. trigger `TRG_CREATE_MATCH` - oveří jestli spolu týmy už 2x hráli a zamezí vytvoření zápasu týmu sám se sebou
3. Informace o ukončených zápasech
 - a. `"SELECT COUNT(*) AS GOAL_COUNT FROM GOAL G, MATCH M WHERE G.MATCH_ID = M.ID AND G.TEAM_ID = ".$teamId." AND G.MATCH_ID = " . $matchId`
4. Možnost upravovat běžící zápas
 - a. `INSERT INTO GOAL VALUES ('', ".$time.", ".$playerId.", ".$matchId.", ' ".$type."', ".$teamId."')`
 - b. procedura `CLOSE_MATCH(MATCH_ID)` - podle počtu vstřelených gólů doplní výsledek(0, 1, 2) do tabulky `MATCH`
 - c. trigger `TRG_CLOSE_MATCH` - pozměně výsledku v tabulce `MATCH` upraví v tabulce `TEAM` skóre zúčastněných týmů a přepočítá pořadí týmů
 - d. trigger `TRG_PUT_GOAL` - po vložení do tabulky `GOAL`, přičte vstřelený a inkasovaný gól zúčastněným týmům v tabulce `TEAM`
 - e. trigger `TRG_OWN_GOAL` - před vložení do tabulky `GOAL`, pokud jde o vlastní gól, upraví jeho typ
 - f. trigger `TRG_PLAYER_IN_MATCH` - před vložení do tabulky `GOAL`, zjistí zda skórující hráč je na soupisce zápasu

Statistiky

1. Hráči podle vstřelených gólů

- a. `SELECT P.ID, P.FIRST_NAME, P.LAST_NAME, COUNT(G.ID) AS GOAL_COUNT FROM
PLAYER P, GOAL G WHERE G.PLAYER_ID = P.ID AND G.TEAM_ID = P.TEAM_ID GROUP
BY P.FIRST_NAME, P.LAST_NAME, P.ID`
- b. `SELECT P.ID, P.FIRST_NAME, P.LAST_NAME, COUNT(G.ID) AS OWN_GOAL_COUNT
FROM PLAYER P, GOAL G WHERE G.PLAYER_ID = P.ID AND G.TEAM_ID != P.TEAM_ID
GROUP BY P.FIRST_NAME, P.LAST_NAME, P.ID`

2. Hráči podle pozice na hřišti

- a. `SELECT P.ID, P.FIRST_NAME, P.LAST_NAME, P.POSITION FROM PLAYER P`

3. Hráči podle zápasů doma a venku

- a. `SELECT P.ID, P.FIRST_NAME, P.LAST_NAME, COUNT(RP.PLAYER_ID) AS MATCHES
FROM ROSTER_PLAYER RP, MATCH M, PLAYER P WHERE P.ID = RP.PLAYER_ID(+) AND
(RP.ROSTER_ID = M.HOME_ROSTER_ID OR RP.ROSTER_ID = M.VISITOR_ROSTER_ID)
GROUP BY P.FIRST_NAME, P.LAST_NAME, P.ID ORDER BY P.ID`

4. Žebříček týmů

- a. `SELECT * FROM TEAM`

5. Zápasů týmů doma a venku

- a. `SELECT T.ID, T.NAME, COUNT(M.ID) AS HOME_MATCH_COUNT FROM TEAM T, MATCH M
WHERE T.ID = M.HOME_ID(+) GROUP BY (T.ID, T.NAME) ORDER BY T.ID`
- b. `SELECT T.ID, T.NAME, COUNT(M.ID) AS VISITOR_MATCH_COUNT FROM TEAM T,
MATCH M WHERE T.ID = M.VISITOR_ID(+) GROUP BY (T.ID, T.NAME) ORDER BY
T.ID`

6. Týmy podle výsledku zápasu

- a. `SELECT T.ID, T.NAME, M.MATCH_DATE, M.RESULT FROM TEAM T, MATCH M WHERE
(T.ID = M.HOME_ID) AND M.RESULT IS NOT NULL`
- b. `SELECT T.ID, T.NAME, M.MATCH_DATE, M.RESULT FROM TEAM T, MATCH M WHERE
(T.ID = M.VISITOR_ID) AND M.RESULT IS NOT NULL`

Databáze dále obsahuje pomocné triggery pro automatickou inkrementaci ID tabulek TRG_nazevtabulky_ID. Poslední trigger TRG_MAX_PLAYERS - před vložením do tabulky ROSTER_PLAYER ověří zda rozpiska obsahuje méně než 11 hráčů.

Realizace

Databáze - Oracle jazyk PL/SQL

Aplikace

- server - Apache, PHP
- klient - Html, Css, Bootstrap, AngularJs, Highcharts

Uživatelský manuál

Interface aplikace funguje jako obyčejná webová stránka. Po otevření příslušné stránky se kliknutím na zobrazené zápasy dostaneme k bližším informacím o zápase (góly, hráči, atd.). Pokud není zápas ukončen můžeme přidávat góly, případně ukončit zápas. Přidání nového zápasu probíhá kliknutím na ikonu + . V pravém horním rohu je odkaz na žebříčky a jiné statistické informace o zápase, gólech a hráčích.

Závěr

Semestrální práce mi poskytla praktickou zkušenost s vytvářením asynchroní webové aplikace a práci s jazykem PL/SQL. Hluběji jsem se seznámil s databází Oracle, frontendovým js frameworkem AngularJs, knihovnou Highcharts a PHP knihovnou OCI. Aplikace jako taková je celkem snadno rozšiřitelná ovšem velmi těžko znovupoužitelná.

Zdroje

courseware.zcu.cz

<http://php.net/>

<http://www.highcharts.com/>

<http://stackoverflow.com/>