ER Diagram



**Table DDL**

-- divisions

CREATE TABLE division(

divisionID int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

divisionName varchar(30) NOT NULL,

amountBest int NOT NULL DEFAULT 2,

conference char(1) CHECK (conference IN ('w','e')));

-- position

CREATE TABLE `position`(

positionID int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`position` VARCHAR(30) NOT NULL

);

-- teams

CREATE TABLE team(

teamID int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name varchar(50) NOT NULL,

city varchar(30) NOT NULL,

arena varchar(50),

generalManager varchar(30),

coach varchar(30),

yearFounded int,

website varchar(100),

divisionID int,

FOREIGN KEY (divisionID) REFERENCES division(divisionID));

-- players

CREATE TABLE player(

playerID int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

firstName varchar(30),

lastName varchar(30),

birthDate date,

height float,

weight float,

country varchar(30),

shoots char(1),

CHECK (shoots IN ('l','r','b')),

positionID int NOT NULL,

FOREIGN KEY (positionID) REFERENCES `position`(positionID));

-- contracts

CREATE TABLE contracts

(teamID int NOT NULL,

playerID int NOT NULL,

dateStart date,

dateEnd date,

salary decimal(10,3),

number int,

FOREIGN KEY (teamID) REFERENCES team(teamID),

FOREIGN KEY (playerID) REFERENCES player(playerID)

);

CREATE TABLE typeGame (

typeGameID int AUTO\_INCREMENT PRIMARY KEY,

typeName varchar(30)

);

CREATE TABLE game

(

gameID int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

teamHome int NOT NULL,

teamGuest int NOT NULL,

dateGame datetime,

place varchar(30),

goal1 int DEFAULT 0,

goal2 int DEFAULT 0,

typeGameID int,

FOREIGN KEY (teamHome) REFERENCES team(teamID),

FOREIGN KEY (teamGuest) REFERENCES team(teamID),

FOREIGN KEY (typeGameID) REFERENCES typeGame(typeGameID)

);

CREATE TABLE goals

(gameID int NOT NULL,

numberPlayer int,

timeGoal time,

isHomePlayer bool,

isGoodGoal bool,

isPenalty bool,

FOREIGN KEY (gameID) REFERENCES game(gameID)

);

CREATE TABLE onGame

(gameID int NOT NULL,

numberPlayer int,

timeStart time,

timeEnd time,

isHomePlayer bool,

FOREIGN KEY (gameID) REFERENCES game(gameID)

);

CREATE TABLE fouls

(gameID int NOT NULL,

numberPlayer int,

timeStart time,

timeEnd time,

isHomePlayer bool,

`comment` varchar(50),

FOREIGN KEY (gameID) REFERENCES game(gameID)

);

**View DDL**

CREATE VIEW win AS

SELECT teamHome,COUNT(gameID)\*2

FROM game, team

WHERE ((goal1>goal2) and (teamHome=teamID)) or

((goal1<goal2) and (teamGuest=teamID))

GROUP BY teamHome;

CREATE VIEW POINTS AS

SELECT teamID,SUM(winPoints(gameID, IFF(teamHome=teamID,1,0)))

FROM game, team

WHERE (teamHome=teamID) or (teamGuest=teamID)

GROUP BY teamID;

**Function DDL**

--Function DDL (minimum 1): function which define by game win points

DELIMITER //

CREATE FUNCTION winPoints(idGame int, forHome bool)

RETURNS int

LANGUAGE SQL

BEGIN

DECLARE baseDifference int;

DECLARE points int;

SELECT SUM(IFF((isHomePlayer-1)\*(isGoodGoal-1)=0,1,-1))INTO baseDifference

FROM goals

WHERE timeGoal<='1:00:00';

IF baseDifference=0 THEN

IF forHome=1 THEN

SELECT IFF(goal1>goal2,1,0) INTO points

FROM goals;

ELSE

SELECT IFF(goal1>goal2,0,1) INTO points

FROM goals;

END IF;

ELSE

IF forHome=1 THEN

SELECT IFF(goal1>goal2,1,0) INTO points

FROM goals;

ELSE

SELECT IFF(goal1>goal2,0,1) INTO points

FROM goals;

END IF;

END IF;

RETURN points;

END;//

DELIMITER ;

**Procedures DDL**

-- procedure which change players on game: find, is exist this number on game, set for him endTime and add record for new player:

DELIMITER //

CREATE PROCEDURE changePlayer(idGame int, isHome bool, numberNew int, numberOld int,timeChange time)

LANGUAGE SQL

BEGIN

DECLARE countNumberOnGame int;

DECLARE points int;

SELECT COUNT(gameID) INTO countNumberOnGame

FROM onGame

WHERE (timeStart<=timeChange) AND ((timeChange<=timeEnd) OR (timeEnd is NULL)) AND (isHome=isHomePlayer) AND numberOld=numberPlayer;

IF countNumberOnGame=1 THEN

UPDATE onGame SET timeEnd=timeChange WHERE (gameID=idGame) AND (numberOld=numberPlayer)AND ( isHome=isHomePlayer);

INSERT INTO onGame VALUES(idGame,numberNew,timeChange,NULL,isHome);

END IF;

END;//

DELIMITER ;

**Trigger DDL**

--Trigger DDL (minimum 1): if we add one goal in the table goals, update table game

DELIMITER //

CREATE TRIGGER addGoal

BEFORE INSERT ON goals FOR EACH ROW

BEGIN

IF ((New.isHomePlayer=1) AND (NEW.isGoodGoal=1)) OR ((New.isHomePlayer=0) AND (NEW.isGoodGoal=0)) THEN

UPDATE game SET goal1=goal1+1 WHERE gameID=New.gameID;

ELSE

UPDATE game SET goal2=goal2+1 WHERE gameID=New.gameID;

END IF;

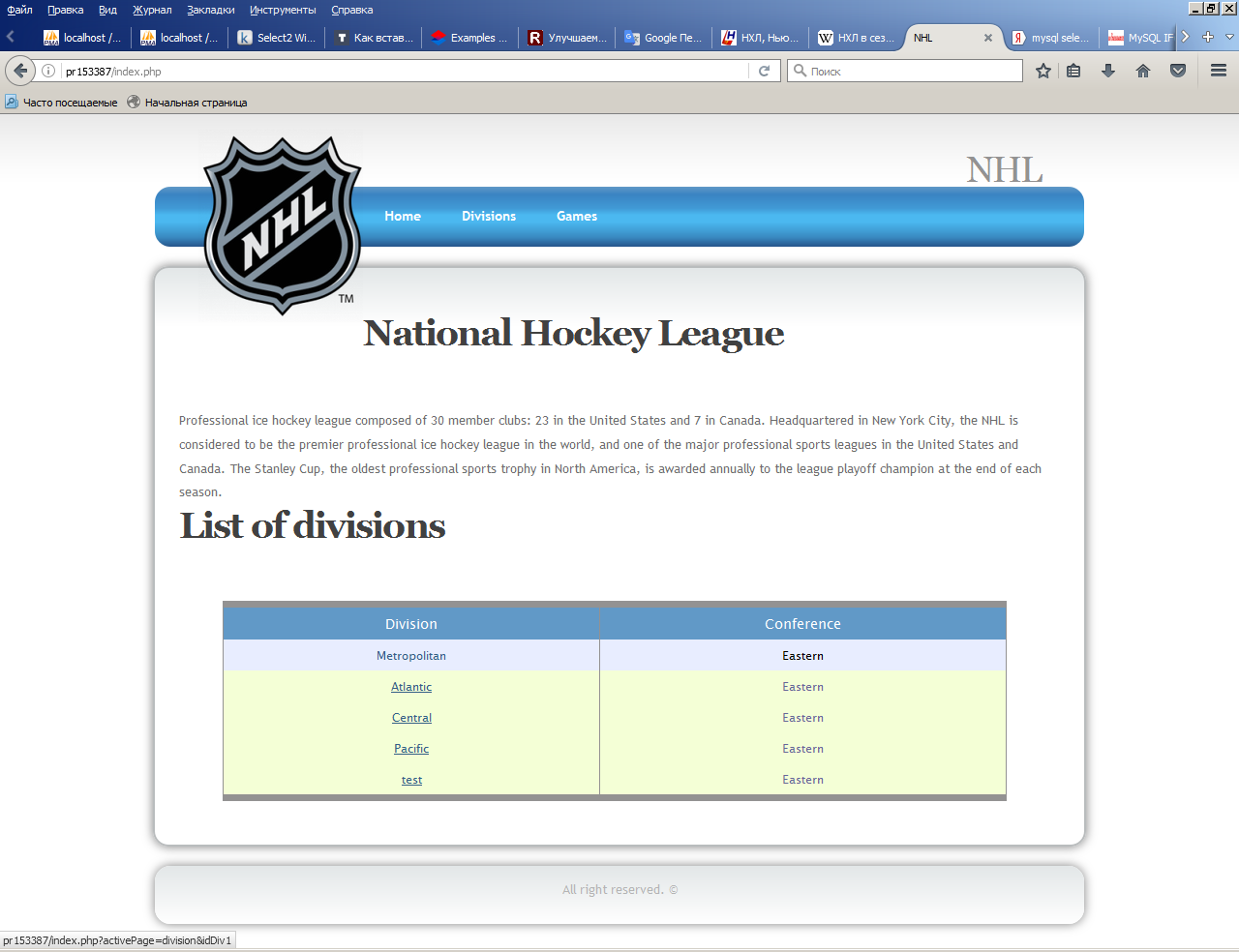
END;

//

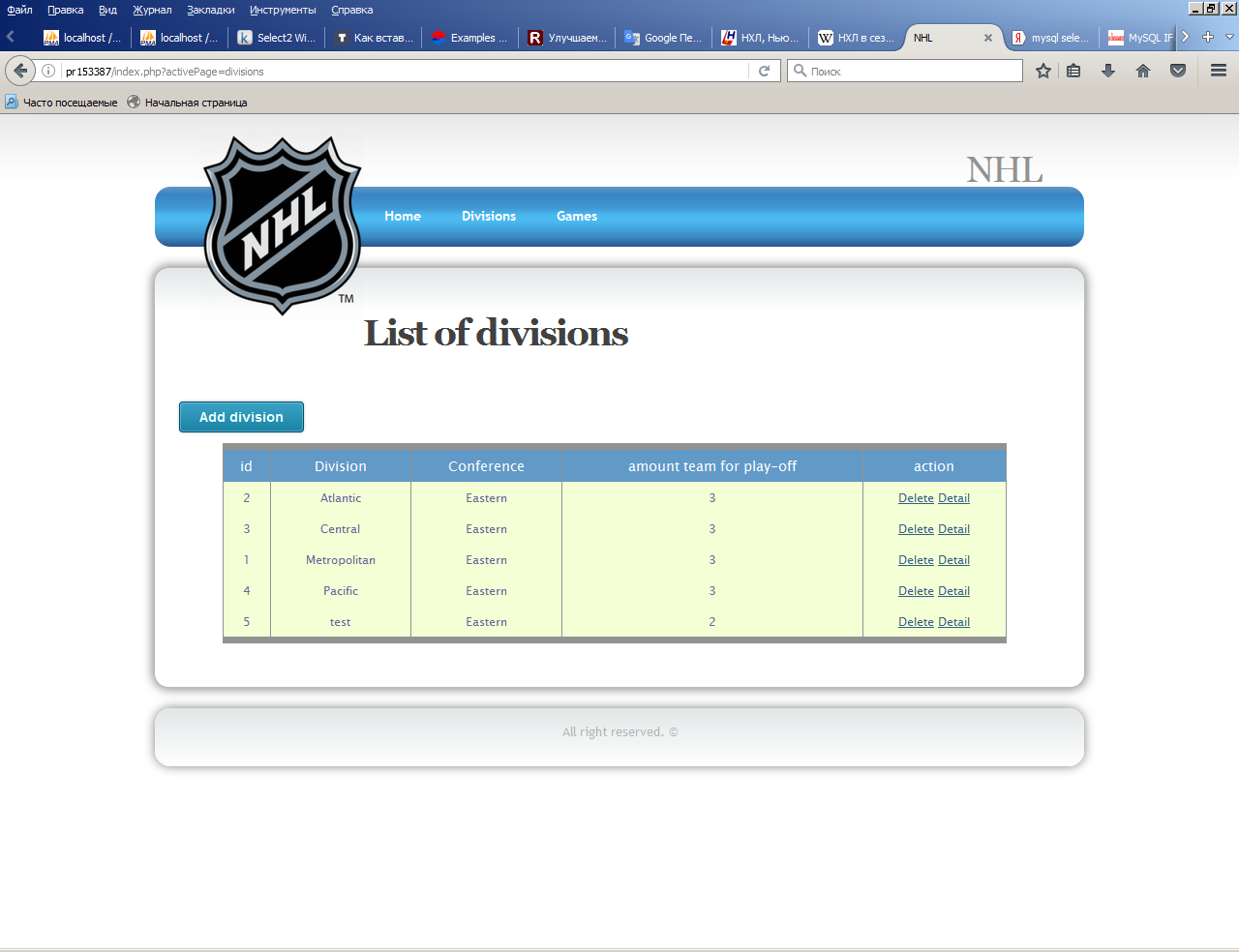
DELIMITER ;

**Pages**

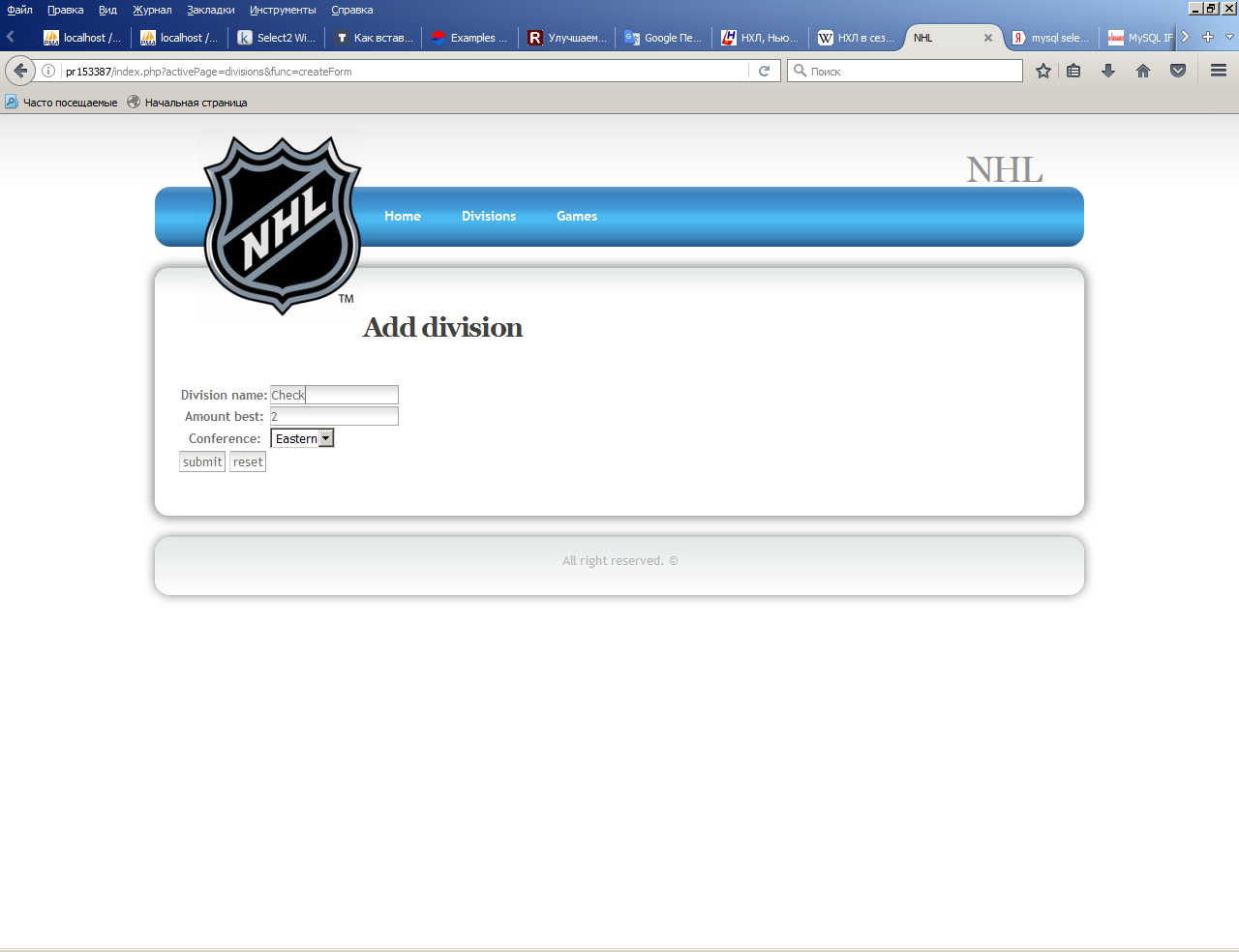
First page



Pages with list of division with main controls element:



Add division:



Update division:

