

Федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский политехнический университет
Петра Великого»
Институт компьютерных наук и технологий
Кафедра Компьютерных систем и программных технологий

Отчёт по лабораторной №3

Дисциплина: Базы данных

Тема: SQL-DDL

Выполнил студент гр. 43501/4

(подпись) В.С. Филиппов

Руководитель

(подпись) А.В. Мяснов

“ ” _____ 2015 г.

Санкт -Петербург

2015

Цель работы

Познакомить студентов с основами проектирования схемы БД, языком описания сущностей и ограничений БД SQL-DDL.

Программа работы ¶

1. Самостоятельное изучение SQL-DDL
2. Создание скрипта БД в соответствии с согласованной схемой (должны присутствовать первичные и внешние ключи, ограничения на диапазоны значений). Продемонстрировать скрипт преподавателю.
3. Создайте скрипт, заполняющий все таблицы БД данными
4. Выполнение SQL-запросов, изменяющих схему созданной БД **по заданию преподавателя**. Продемонстрировать их работу преподавателю.
5. Изучите основные возможности IBExpert. Получите ER-диаграмму созданной БД с помощью **Database Designer**.
6. Автоматически сгенерируйте данные при помощи IBExpert (для трех или большего числа таблиц, не менее 100000 записей в каждой из выбранных таблиц)

Выполнение работы

1. Скрипт создания

Домены используются как типы перечислений. Генераторы используются как `autoincrement` в `mysql`.

```
SET NAMES CYRL;
create database 'E:\Program Files\Firebird\db\sport_IS.fdb'
user 'SYSDBA' password 'masterkey'
DEFAULT CHARACTER SET CYRL;

connect 'E:\Program Files\Firebird\db\sport_IS.fdb'
user 'SYSDBA' password 'masterkey';
commit;

CREATE DOMAIN pos_type
AS varchar(10) CHECK (value IS NULL or VALUE IN
('Goalkeeper', 'Defender', 'Midfielder', 'Forward', 'Coach'));

CREATE DOMAIN league_type
AS varchar(10) CHECK (value IS NULL or VALUE IN
('World', 'Continent', 'Country', 'City'));

CREATE TABLE players
(player_id INTEGER NOT NULL PRIMARY KEY,
name VARCHAR(16) NOT NULL,
birthdate DATE,
pos pos_type
```

```

    );
commit;

CREATE TABLE clubs
    (club_id INTEGER NOT NULL PRIMARY KEY,
     name VARCHAR(16) NOT NULL,
     birthdate DATE,
     nickname VARCHAR(17),
     country VARCHAR(16)
    );
commit;

CREATE TABLE leagues
    (league_id INTEGER NOT NULL PRIMARY KEY,
     name VARCHAR(16) NOT NULL,
     kind league_type
    );
commit;

CREATE TABLE seasons
    (season_id INTEGER NOT NULL PRIMARY KEY,
     league_id INTEGER REFERENCES leagues(league_id),
     period VARCHAR(9)
    );
commit;

CREATE TABLE matches
    (match_id INTEGER NOT NULL PRIMARY KEY,
     match_date DATE,
     season_id INTEGER REFERENCES seasons(season_id),
     first_club_id INTEGER REFERENCES clubs(club_id),
     second_club_id INTEGER REFERENCES clubs(club_id),
     first_club_goals SMALLINT,
     second_club_goals SMALLINT
    );
commit;

CREATE TABLE goals
    (goal_id INTEGER NOT NULL PRIMARY KEY,
     player_id INTEGER REFERENCES players(player_id),
     match_id INTEGER REFERENCES matches(match_id),
     goal_minute SMALLINT,
     autogoal CHAR(1)
    );
commit;

CREATE TABLE standings
    (season_id INTEGER REFERENCES seasons(season_id),
     club_id INTEGER REFERENCES clubs(club_id),
     victories SMALLINT,
     draws SMALLINT,
     losses SMALLINT,
     goals SMALLINT,
     missed_goals SMALLINT,
     diff_goals SMALLINT,
     points SMALLINT
    );
commit;

CREATE TABLE club_players_list_history
    (club_id INTEGER REFERENCES clubs(club_id),
     player_id INTEGER REFERENCES players(player_id),
     date_begin DATE NOT NULL,
     date_end DATE

```

```

    );
commit;

CREATE TABLE current_club_players_list
    (club_id INTEGER REFERENCES clubs(club_id) ,
     player_id INTEGER REFERENCES players(player_id)
    );
commit;

CREATE GENERATOR gen_player_id;
SET GENERATOR gen_player_id TO 0;
set term !! ;
CREATE TRIGGER PLAYERS_BI FOR players
ACTIVE BEFORE INSERT POSITION 0
AS
BEGIN
NEW.player_id = GEN_ID(gen_player_id, 1);
END!!
set term ; !!

CREATE GENERATOR gen_club_id;
SET GENERATOR gen_club_id TO 0;
set term !! ;
CREATE TRIGGER CLUBS_BI FOR clubs
ACTIVE BEFORE INSERT POSITION 0
AS
BEGIN
NEW.club_id = GEN_ID(gen_club_id, 1);
END!!
set term ; !!

CREATE GENERATOR gen_league_id;
SET GENERATOR gen_league_id TO 0;
set term !! ;
CREATE TRIGGER LEAGUES_BI FOR leagues
ACTIVE BEFORE INSERT POSITION 0
AS
BEGIN
NEW.league_id = GEN_ID(gen_league_id, 1);
END!!
set term ; !!

CREATE GENERATOR gen_season_id;
SET GENERATOR gen_season_id TO 0;
set term !! ;
CREATE TRIGGER SEASONS_BI FOR seasons
ACTIVE BEFORE INSERT POSITION 0
AS
BEGIN
NEW.season_id = GEN_ID(gen_season_id, 1);
END!!
set term ; !!

CREATE GENERATOR gen_match_id;
SET GENERATOR gen_match_id TO 0;
set term !! ;
CREATE TRIGGER MATCHES_BI FOR matches
ACTIVE BEFORE INSERT POSITION 0
AS
BEGIN
NEW.match_id = GEN_ID(gen_match_id, 1);
END!!
set term ; !!

```

```

CREATE GENERATOR gen_goal_id;
SET GENERATOR gen_goal_id TO 0;
set term !! ;
CREATE TRIGGER GOALS_BI FOR goals
ACTIVE BEFORE INSERT POSITION 0
AS
BEGIN
NEW.goal_id = GEN_ID(gen_goal_id, 1);
END!!
set term ; !!

```

2. Скрипт заполнения

Добавляем игроков Джюбу и Кержакова.

Команды Зенит и Спартак.

Лигу РФПЛ

Сезон 2015/2016

Матч Зенит – Спартак со счётом 2:2

Гол Джюбы и Кержакова

В турнирной таблице командам присваивается по 1 очку за ничью.

В список игроков Зенита добавляются игроки.

```

INSERT INTO players (name, birthdate, pos)
VALUES ('Dzuba', '22-AUG-1988', 'Forward');
INSERT INTO players (name, birthdate, pos)
VALUES ('Kerjakov', '10-JUN-1982', 'Forward');

INSERT INTO clubs (name, birthdate, nickname, country)
VALUES ('Zenit', '25-MAY-1925', 'Blue-White-Blue', 'Russia');
INSERT INTO clubs (name, birthdate, nickname, country)
VALUES ('Spartak', '18-APR-1922', 'White-Red', 'Russia');

INSERT INTO leagues (name, kind)
VALUES ('RFPL', 'Country');

/*lid = (select league_id from leagues where name = 'ДЮ');*/
INSERT INTO seasons (league_id, period)
VALUES (1/*lid*/, '2015/2016');

INSERT INTO matches (match_date, season_id, first_club_id, second_club_id,
first_club_goals, second_club_goals)
VALUES('26-SEP-2015',1,2,1,2,2);

INSERT INTO goals (match_id, player_id, goal_minute, autogoal)
VALUES(1,1,22,0);
INSERT INTO goals (match_id, player_id, goal_minute, autogoal)
VALUES(1,2,41,0);

INSERT INTO standings (season_id, club_id, victories, draws, losses, goals,
missed_goals, diff_goals, points)
VALUES(1,1,0,1,0,2,2,0,1);

```

```

INSERT INTO standings (season_id, club_id, victories, draws, losses, goals,
missed_goals, diff_goals, points)
VALUES (1,2,0,1,0,2,2,0,1);

INSERT INTO club_players_list_history (club_id, player_id, date_begin,
date_end)
VALUES (1, 1, '20-JUL-2015', NULL);
INSERT INTO club_players_list_history (club_id, player_id, date_begin,
date_end)
VALUES (1, 2, '13-MAY-2005', NULL);

INSERT INTO current_club_players_list (club_id, player_id)
VALUES (1, 1);
INSERT INTO current_club_players_list (club_id, player_id)
VALUES (1, 2);

```

3. Скрипт изменения по заданию:

Модифицировать схему БД для удовлетворения следующим требованиям:

1. Добавить учет прочей статистики матчей: желтые карточки, удаления, удары по воротам, владение мячом и пр.
2. Реализовать таблицу текущего положения команд в турнире.

Таблица текущего положения команд в турнирной таблице уже есть, она называется standings.

В связи с учётом статистики, создадим таблицу events, которая «поглотит» таблицу goals. А для учёта статистики, которая относится к матчу в целом добавим в таблицу matches соответствующие поля.

```

CREATE DOMAIN event_type
AS varchar(11) CHECK (value IS NULL or VALUE IN
('GOAL', 'YELLOW', 'RED', 'INJURY', 'REPLACEMENT'));
/* травма      замена */

CREATE TABLE events
(player_id INTEGER REFERENCES players(player_id),
match_id INTEGER REFERENCES matches(match_id),
event_minute SMALLINT,
event_event_type,
ag_io CHAR(1)
);
/* ag_io - GOAL: 1-autogoal, 0-goal; YELLOW,RED: NULL; INJURY: 1-нанёс
травму, 0-получил травму; REPLACEMENT: 1-зашёл на поле, 0-вышел */

insert into events (player_id, match_id, event_minute, ag_io, event)
select player_id, match_id, goal_minute, autogoal, 'GOAL' from goals;

DROP trigger GOALS_BI;
DROP generator gen_goal_id;
commit;
DROP table goals;
commit;

ALTER TABLE matches
ADD first_club_percent SMALLINT; /*владение мячом*/

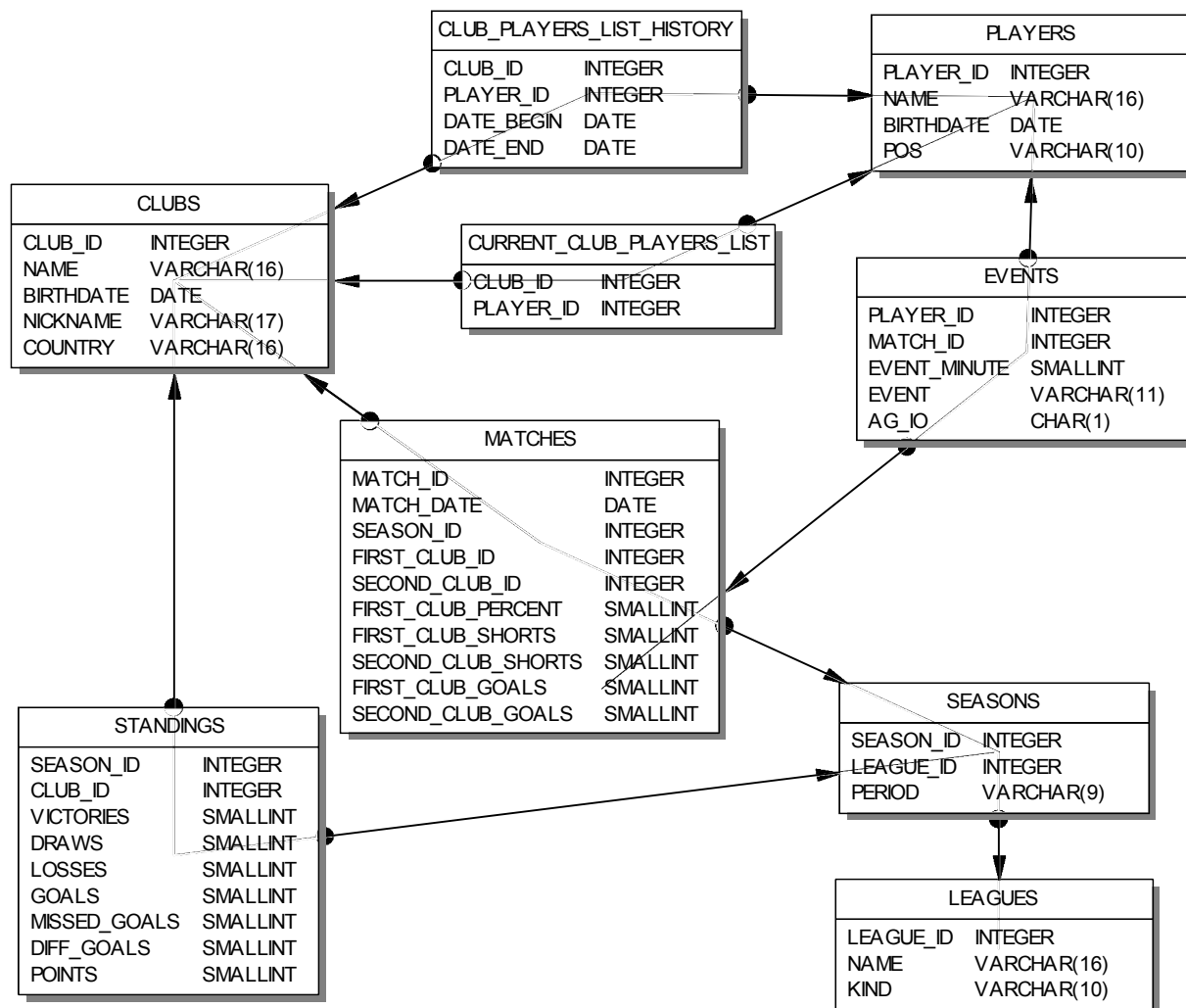
```

```

ALTER TABLE matches
  ADD first_club_shorts SMALLINT; /*удары*/
ALTER TABLE matches
  ADD second_club_shorts SMALLINT;

```

4. Построение ER-диаграммы с помощью IB-Expert.



5. Автогенерация тестовых данных

Для таблиц clubs, players и current_club_players_list было автоматически добавлено по 100000 записей.

В таблицу clubs эта операция осуществлялась около 2-х минут. В таблицу players около 1 минуты. В таблицу current_club_players_list около 0,5 минуты.

В таблице players в поле pos данные выбирались из списка домена pos_type. В таблице current_club_players_list были выбраны внешние ключи из других таблиц (clubs и players).

Фрагменты сгенерированных записей приведены ниже.

CLUB_ID	NAME	BIRTHDATE	NICKNAME	COUNTRY
3	qHl>k'rzO/bc{J	23.08.1903	vm	Italy
4	qbP&	28.07.1948	nfwyrtg	England
5	QIl,OGS qx3-;	02.01.1945	bzdpurtu	England
6	i8GuP=*qE	05.01.1843	ogazgc auqlq	Germany
7	fGO4QN	05.06.1909	wcatwpyjxokgnhdd	Spanish
8	O\qc.O	10.04.1829	woopdkvtrknbtx	England
9	3b9UZq	18.04.2012	ssjnxjjtptncfq	Ukraine
10	^VbQ!R6-pc&VJsw1	10.05.1886	cfjjgrtmuavj	Spanish
11	k'@8	02.05.1850	tqapanvwepipzan	France

PLAYER_ID	NAME	BIRTHDATE	POS
3	trgambdnzny	16.09.1700	Defender
4	zlixoslpqfnkwxka	12.01.1725	Coach
5	fxyrjnab	13.11.1753	Defender
6	a	06.01.1773	Defender
7	acpfwfcz	24.06.1700	Defender
8	cxf	27.09.1766	Forward
9	bhaholrqyvumwxv	14.06.1917	Defender
10	xo	05.08.2009	Goalkeeper
11	bxoizgmd	14.06.1717	Forward

CLUB_ID	PLAYER_ID
22 843	58
59 017	88
35 325	79
35 419	48
86 179	71
37 514	130
70 838	44
53 048	40
30 572	2
3 955	126

Вывод

В данной работе для создания базы данных использовался SQL-DDL. Язык описания сущностей SQL-DDL позволяет определять структуры и ограничения целостности баз данных. В ходе работы были определены домены, и взаимосвязанные таблицы; реализована вставка в таблицы и изменения таблиц.

Язык SQL-DLL, удобнее в отличие от обычных языков программирования, для работы с базами данных. Его синтаксис гибкий, и он настроен на работу с определением структур данных. SQL-DLL является стандартным, имеет много реализаций.

По ходу работы SQL-DLL было удобно использовать. Для вставки элементов в таблицу нужно указать таблицу, имена полей, и значения полей.

Для удаления полей из таблицы нужно указать таблицу, и название полей. С точки зрения удобства проще язык не придумать.

Средствами IB-Expert была сгенерирована ER-диаграмма, которая наглядно показывает структуру базы данных. Также выполнена генерация записей в таблицы базы данных.