**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Институт информационных технологий и управления**

**Кафедра компьютерных систем и программных технологий**

**ОТЧЕТ**

**по лабораторной работе №2**

**«Разработка структур и нормализация БД»**

**Информационное обеспечение систем управления**

Студент гр. 43501/1 Васильев А.М.

Преподаватель Мяснов А.В.

**Санкт-Петербург**

**2014**

# 1.Цель работы

Ознакомиться с основами проектирования схемы БД, способами нормализации отношений в БД.

## Программа работы:

1. Представить SQL-схему БД, соответствующую заданию (должно получиться не менее 7 таблиц)
2. Привести схему БД к [3НФ](http://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D1%82%D1%8C%D1%8F_%D0%BD%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0)
3. Обосновать соответствие схемы [3НФ](http://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D1%82%D1%8C%D1%8F_%D0%BD%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0).

## Индивидуальное задание:

БД хранит информацию об абонентах мобильного оператора, услугах (с фиксированной и посекундной тарификацией), использованию абонентами услуг, счетах, списаниях и зачислениях на счета. ИС дает возможность управлять различными акциями и начислением бонусов.

# 2. Разработка схемы БД

По заданию была разработана схема:

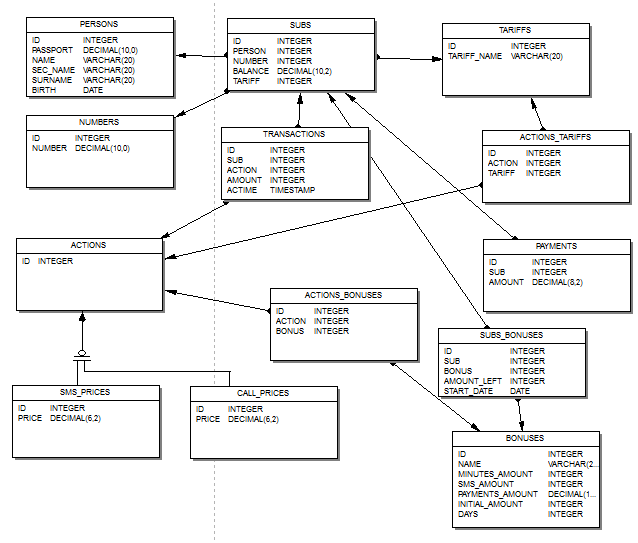


Рис 1. Схема БД.

В данной схеме присутствуют таблицы:

* **Persons –** личные данные клиентов
* **Numbers –** все телефонные номера клиентов
* **Subs** – договора абонентов, ссылается на личные данные абонента, номер телефона и тариф, хранит баланс.
* **Tariffs** – названия тарифов
* **Payments** – платежи абонентов, ссылается на договор абонента
* **Transactions** – оказанные услуги, ссылается на абонента, услугу, указано количество минут или sms (длинные сообщения тарифицируются как несколько)
* **Actions** – услуги (sms или звонки)
* **Sms\_prices** – стоимость sms, наследуется от actions.
* **Call\_prices** – стоимость звонка, наследуется от actions.
* **Actions\_tariffs** – услуга, оказанная на условиях тарифа, ссылается на соотв. таблицы.
* **Actions\_bonuses** – услуга, оказанная на условиях бонуса, ссылается на соотв. таблицы.
* **Bonuses** – таблица названий и условий бонусов, содержит либо количество минут, либо количество sms, либо объем платежей, которые нужно совершить за месяц, чтобы получить бонус. Также содержит либо количество услуг для бонусов с лимитом, либо время действия в сутках для ограниченных по времени бонусов.
* **Subs\_bonuses** – таблица бонусов абонентов, ссылается на абонента и бонус и содержит сведения об остатках действия бонуса (дата окончания или количество).

Всего 13 таблиц.

Скрипт для создания БД:

SET NAMES WIN1251;

create database 'C:/db/nanofon.fdb' user 'SYSDBA' password 'masterkey' DEFAULT CHARACTER SET WIN1251;

connect 'C:/db/nanofon.fdb' user 'SYSDBA' password 'masterkey';

create table persons (

id int primary key,

passport decimal(10,0) not null unique check (passport >= 1000000000),

name varchar(20) not null,

sec\_name varchar(20),

surname varchar(20) not null,

birth date not null);

create table tariffs (

id int primary key,

tariff\_name varchar(20));

create table numbers (

id int primary key,

number decimal(10,0) not null unique check (number >= 1000000000));

create table subs (

id int primary key,

person int references persons(id) not null,

number int references numbers(id) not null,

balance decimal(10,2) not null,

tariff int references tariffs(id) not null);

create table payments (

id int primary key,

sub int references subs(id) not null,

amount decimal(8,2) not null);

create table sms\_prices (

id int primary key,

price decimal(6,2) not null);

create table call\_prices (

id int primary key,

price decimal(6,2) not null);

create table actions (

id int primary key);

create table transactions (

id int primary key,

sub int references subs(id) not null,

action int references actions(id) not null,

amount int not null,

actime timestamp not null);

create table bonuses (

id int primary key,

name varchar (20) not null unique,

minutes\_amount int,

sms\_amount int,

payments\_amount decimal(10,2),

initial\_amount int,

days int);

create table actions\_bonuses(

id int primary key,

action int references actions(id) not null,

bonus int references bonuses(id) not null);

create table actions\_tariffs(

id int primary key,

action int references actions(id) not null,

tariff int references tariffs(id) not null);

create table subs\_bonuses(

id int primary key,

sub int references subs(id) not null,

bonus int references bonuses(id) not null,

amount\_left int,

start\_date date);

commit;

## Нормализация

Условие приведения БД к 1 НФ – отсутствие неатомарных атрибутов. В данной БД их нет.

Условие приведения БД ко2 НФ – 1 НФ и требование, чтобы все неключевые атрибуты функционально полно зависели от ключа.. В данной БД это требование выполняется, так как отсутствуют таблицы, в первичный ключ составной и есть неполная функциональная зависимость от него. Если, например, создать отношение ТРАНЗАКЦИЯ (АБОНЕНТ, УСЛУГА, СТОИМОСТЬ) с первичным ключом (АБОНЕНТ+УСЛУГА), то условие нормализации будет нарушено.

Условие приведения к 3 НФ – 2 НФ и требование, чтобы каждый неключевой атрибут нетранзитивно зависел от ключа. В данной базе нет таблиц, в которых были бы зависимости одного неключевого атрибута от другого. Если бы существовало отношение ТРАНЗАКЦИЯ (УСЛУГА, ДОГОВОР, НОМЕР), где номер абонента зависит от договора, условие было бы нарушено.

# 3. Вывод

В ходе выполнения работы были изучены такие вопросы, как синтез базы данных по тематике, указанной в формулировке задания и нормализация полученной БД.Третья нормальная форма позволяет избавиться от большинства основных аномалий в БД, таких как избыточность – когда, одинаковые элементы информации повторяются многократно в нескольких кортежах, аномалии изменения - один и тот же фрагмент данных изменяется в одном кортеже, и остается неизменным в другом и аномалии удаления - если множество значений становится пустым, это может косвенным образом привести к потере другой информации. Приведение к последующим нормальным формам, в рамках нашей задачи, уже не дадут таких значительных улучшений.

Нормализация предназначена для приведения структуры БД к виду, обеспечивающему минимальную логическую избыточность, и не имеет целью уменьшение или увеличение производительности работы или же уменьшение или увеличение физического объёма базы данных. Конечной целью нормализации является уменьшение потенциальной противоречивости хранимой в базе данных информации.

При этом для ускорения работы запросов к БД иногда требуется проводить денормализацию, то есть сознательно нарушать требования нормальных форм, в тех случаях, когда потенциальная выгода перевешивает необходимость дополнительно следить за аномалиями в БД.