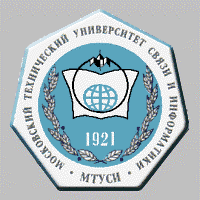
Министерство цифрового развития, связи и массовых коммуникаций

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное

учреждение высшего профессионального образования

Московский технический университет связи и информатики (МТУСИ)



Кафедра бизнес информатики

**Курсовая работа**

**по дисциплине:**

**«Базы данных»**

**На тему:**

**«Предметная область ИС: БД для анализа тарифных планов сотовых компаний.»**

**Вариант 14**

Выполнил:

студент гр. БЭИ1802

Пальчун Д.А.

Принял:

Фролова Е.А

**Москва, 2021**

**Содержание**

[**Введение**](#_pjk19kebra88) **3**

[**1. Концептуальное проектирование**](#_w1rmij27ubj7) **4**

[1.1. Анализ предметной области](#_jvw04hkbgf5w) 4

[1.2. Выявление сущностей](#_lnuemjx5hzpu) 5

[1.3. Анализ информационных задач и круга пользователей системы](#_ud7yrh9hiwq2) 6

[1.4. Выбор СУБД и других программных средств](#_gondrmt2w9vl) 7

[**2. Логическое проектирование**](#_jiw47xkppwm7) **8**

[2.1. Преобразование концептуальной модели в схему базы данных](#_kfox3dekfbj7) 8

[2.2. Описание групп пользователей и прав доступа](#_9ghjfyuxblu8) 12

[**3. Физическое проектирование**](#_pa8qx2tthxrz) **13**

[**4. Демонстрация приложения**](#_ri4q26wkuz2z) **16**

[**Заключение**](#_5y5n1f8y3ddn) **21**

[**Список использованной литературы**](#_cylgiyppvpo2) **22**

[**Приложение**](#_wf1iti3dpy3x) **23**

# **Введение**

В современном мире большую популярность имеют сервисы контент-провайдеров и контент агрегаторов. Они присутствуют в самых разных сферах, от магазинов еды, до путешествий по миру. Их использование сильно облегчает поиск нужного товара, это очень удобно, ведь не нужно никуда ходить, при этом сохраняется много времени.

Если человек задается целью выяснить, какой тарифный план ему подходит больше всего, то перед ним встает долгая и нудная работа по поиску данных тарифов на сайтах компаний, которые их предоставляют. Если брать одного человека, то в данной задаче нет ничего страшного, но такой вопрос возникает у многих людей по всему миру, следовательно, создание сайта-агрегатора, в котором представлены тарифные планы, которые можно сравнить, является полезным действием, которая в перспективе также может приносить прибыль. Но прежде чем приступать к созданию сайта, необходимо продумать то, как он будет функционировать изнутри.

Таким образом, целью данной работы является создание базы данных для анализа тарифных планов сотовых компаний

# 

# **1. Концептуальное проектирование**

Самый часто используемый способ проектирования баз данных – трёхуровневое проектирование. В него входят три уровня: концептуальный, логический и физический.

## **1.1. Анализ предметной области**

Сайт-агрегатор тарифных планов представляет собой структуру, собирающую информацию по тарифам и операторам в одном месте. Такая структура может представлять собой сайт, являющийся либо информационным ресурсом, на котором будет находиться только лишь информация о тарифах, либо выступать в роли посредника, позволяя клиентам заключать договоры с операторами перейдя на их тариф. В такой базе данных основой является список тарифов, открыв который, можно будет провести их сравнительный анализ.

Если рассматривать второй вариант, то в таком случае, для того, чтобы перейти на тариф через посреднический сайт, придется зарегистрироваться на этом сайте, и при переходе на данный тариф через этот сайт, информация об этом действии будет передаваться непосредственно оператору, либо договор можно заключать через определённые окна, которые хоть и находятся на сайте-агрегаторе, но вся информация, которая в них вносятся, передается непосредственно оператору.

В первом случае, кроме хранение данных о тарифах, необходимо также продумать и расположение в базе данных информации о зарегистрированных клиентах. Вести учёт их и их действий.

Информация о тарифах может добываться несколькими путями, одним из которых является ручной ввод необходимых данных, например, администратор сайта-агрегатора может последовательно заходить на сайты ты операторов, тарифы которых необходимо добавить в базу данных, либо настроить сайт агрегатор таким образом, чтобы он автоматически заходил на нужные сайты и получал информацию, либо договориться с операторами о том, чтобы они присылали необходимую информацию.

## **1.2. Выявление сущностей**

Самые используемые методологии инфологического проектирования: предметный подход, функциональный подход и проектирование с использованием метода «сущность-связь». Мы будем использовать метод «сущность-связь».

Сущность - это класс однотипных объектов, информация о которых должна быть учтена в модели.

Связь - это некоторая ассоциация между двумя сущностями.

В данной предметной области можно выделить следующие предметные сущности и их атрибуты:

* Тариф
  + Название тарифа
  + Цена
  + Количество звонков
  + Количество SMS
  + Рейтинг тарифа
  + Регион
  + Роуминг
  + Сотовый оператор
  + Количество абонентов
  + Кэшбэк
  + Описание
* Регион
  + Определенный регион
* Сотовый оператор
  + Название оператора
  + Рейтинг оператора
* Клиент
  + Обозначение
  + Паспорт
* Комментарий
  + Клиент
  + Тариф
  + Содержание
* История всех тарифов
  + Клиент
  + Тариф
  + Дата окончания пользования тарифом
* Информация о тарифах пользователя
  + Клиент
  + Тариф
  + Количество оставшихся минут
  + Количество оставшихся SMS

С помощью выявленных сущностей построим концептуальную модель, содержащую информацию, определяющую смысловую структуру рассматриваемой области.

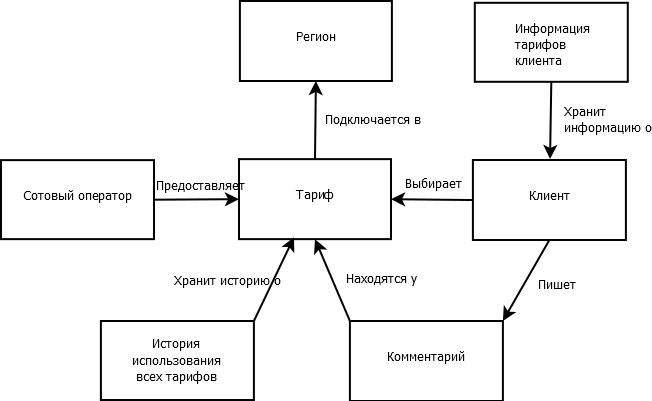


Рис. 1 Концептуальная модель

## **1.3. Анализ информационных задач и круга пользователей системы**

Созданием данного сайта-агрегатора занимался один программист, так как в дополнительных руках нет необходимости, он и будет единственным администратором, который будет полностью управлять базой данных и пользоваться приложением.

В данной работе были поставлены задачи по автоматизации таких процессов, как внесение, удаление и изменение данных о:

* Тарифах;
* Регионах;
* Операторах;
* Клиентах;
* Истории тарифов;
* Текущих тарифах пользователей;
* Комментариях.

## **1.4. Выбор СУБД и других программных средств**

Анализировать тарифные планы многих операторов, собирая информацию в интернете, это довольно трудная задача, а хранение данной информации ещё труднее. Основываясь на многочисленных тестах, которые проводят люди, имеющие опыт в данной сфере, и публикующие данные, полученные в ходе эксперимента, в открытый доступ, можно прийти к выводу, что самым подходящим вариантом для обработки подобных массивов данных является oracle.

Но и она имеет недостатки, такие как неадаптированный для пользователя стандартный интерфейс, поэтому для более удобной работы с ней существует специальная среда разработки от самой oracle - apex.

Операционная система windows, а браузер Chrome, потому что они используются чаще всего.

# 

# **2. Логическое проектирование**

## **2.1. Преобразование концептуальной модели в схему базы данных**

Одним из этапов логического проектирования БД является этап проверки схемы БД на отсутствие аномалий преобразований данных. Чтобы решить эту проблему преобразований данных при проектировании реляционной БД проводится нормализация отношений.

В процессе преобразования концептуальной модели в логическую схему каждый объект предметной области предстает в виде взаимосвязанных таблиц, а также выполняется сопоставление связей.

**Без нормализации**:

**R0** (Название тарифа, Цена, Количество звонков, Количество SMS, Рейтинг тарифа, Количество абонентов, Кэшбэк, Описание, Определенный регион, Наличие роуминга, Название оператора, Рейтинг оператора, Обозначение клиента, Паспорт, Содержание комментария, Дата окончания пользования тарифом, Количество оставшихся минут, Количество оставшихся SMS)

**1-я нормальная форма реляционного отношения**:

Отсутствие множественности атрибутов, все строки не повторяются и атрибуты простые:

**R1** [Тариф] (Название тарифа, Цена, Количество звонков, Количество SMS, Рейтинг тарифа, Количество абонентов, Кэшбэк, Описание)

**R1\_1** [Регион] (Определенный регион)

**R1\_2** [Роуминг] (Наличие роуминга)

**R1\_3** [Оператор] (Название оператора, Рейтинг оператора)

**R2** [Клиент] (Обозначение, Паспорт)

**R3** [Комментарий] (Содержание комментария)

**R4** [История всех тарифов] (Дата окончания пользования тарифом)

**R5** [Информация о тарифах пользователя] (Количество оставшихся минут, Количество оставшихся SMS)

**2-я нормальная форма реляционного отношения**:

1-я нф + у таблицы есть ключ и все неключевые столбцы зависят от всего ключа, а не его части (если он составной) или описываю его:

**R1** [Тариф] (Тариф (PK), Название тарифа, Цена, Количество звонков, Количество SMS, Рейтинг тарифа, Регион (FK), Роуминг (FK), Оператор (FK), Количество абонентов, Кэшбэк, Описание)

**R1\_1** [Регион] (Регион (PK), Определенный регион)

**R1\_2** [Роуминг] (Роуминг (PK), Наличие роуминга)

**R1\_3** [Оператор] (Оператор (PK), Название оператора, Рейтинг оператора)

**R2** [Клиент] (Клиент (PK), Обозначение, Паспорт)

**R3** [Комментарий] (Комментарий (PK), Тариф (FK), Клиент (FK), Содержание комментария)

**R4** [История всех тарифов] (История (PK), Клиент (FK), Тариф (FK), Дата окончания пользования тарифом)

**R5** [Информация о тарифах пользователя] (Информация (PK), Клиент (FK), Тариф (FK), Количество оставшихся минут, Количество оставшихся SMS)

**3-я нормальная форма реляционного отношения**:

2-я нф + неключевые атрибуты не зависят друг от друга:

**R1** [Тариф] (Тариф (PK), Название тарифа, Цена, Количество звонков, Количество SMS, Рейтинг тарифа, Регион (FK), Роуминг (FK), Оператор (FK), Количество абонентов, Кэшбэк, Описание)

**R1\_1** [Регион] (Регион (PK), Определенный регион)

**R1\_2** [Роуминг] (Роуминг (PK), Наличие роуминга)

**R1\_3** [Оператор] (Оператор (PK), Название оператора, Рейтинг оператора)

**R2** [Клиент] (Клиент (PK), Обозначение, Паспорт)

**R3** [Комментарий] (Комментарий (PK), Тариф (FK), Клиент (FK), Содержание комментария)

**R4** [История всех тарифов] (История (PK), Клиент (FK), Тариф (FK), Дата окончания пользования тарифом)

**R5** [Информация о тарифах пользователя] (Информация (PK), Клиент (FK), Тариф (FK), Количество оставшихся минут, Количество оставшихся SMS)

**4-я нормальная форма реляционного отношения**:

3-я нф + отсутствие нетривиальных многозначных зависимостей - наличие ситуации, где в таблице, с тремя ключевыми полями А В С, для одного значения А существуют множества значений В и С.

**R1** - каждый конкретный тариф может иметь только одну совокупность атрибутов, а следовательно, утверждение, где для одного значения А существуют множества значений В и С, неверно.

**R3**, **R4**, **R5** - здесь, первичные ключи представлены в качестве порядкового номера и счётчиков, а следовательно, они указывают на объекты с фиксированными атрибутами, которые не могут меняться для каждого из номеров.

Словарь данных. Здесь указано название полей в базе данных, их тип данных и размер, также, указано поле первичного и внешнего ключа.

Таблица 1. region

| regid | NUMBER(3) | PK |
| --- | --- | --- |
| reg | VARCHAR(30) |  |

Таблица 2. roaming

| roaid | NUMBER(3) | PK |
| --- | --- | --- |
| roa | VARCHAR(30) |  |

Таблица 3. oper

| operid | NUMBER(3) | PK |
| --- | --- | --- |
| nameOper | VARCHAR(30) |  |
| score | NUMBER(3) |  |

Таблица 4. tarif

| tarid | NUMBER(3) | PK |
| --- | --- | --- |
| tar | VARCHAR(30) |  |
| price | NUMBER(4) |  |
| calls | NUMBER(4) |  |
| sms | NUMBER(4) |  |
| score | NUMBER(3) |  |
| regid | NUMBER(3) | FK |
| roaid | NUMBER(3) | FK |
| operid | NUMBER(3) | FK |
| subs | NUMBER(10) |  |
| cashb | NUMBER(4) |  |
| descr | VARCHAR(30) |  |

Таблица 5. client

| clientid | NUMBER(3) | PK |
| --- | --- | --- |
| name | VARCHAR(30) |  |
| passport | NUMBER(10) |  |

Таблица 6. comments

| comid | NUMBER(3) | PK |
| --- | --- | --- |
| tarid | NUMBER(3) | FK |
| clientid | NUMBER(3) | FK |
| content | VARCHAR(30) |  |

Таблица 7. histAllTar

| histid | NUMBER(3) | PK |
| --- | --- | --- |
| tarid | NUMBER(3) | FK |
| clientid | NUMBER(3) | FK |
| dateov | DATE |  |

Таблица 8. infTarUser

| infid | NUMBER(3) | PK |
| --- | --- | --- |
| tarid | NUMBER(3) | FK |
| clientid | NUMBER(3) | FK |
| amCall | NUMBER(4) |  |
| amSms | NUMBER(4) |  |

## **2.2. Описание групп пользователей и прав доступа**

Приложением будет пользоваться только тот самый единственный создатель сайта-агрегатора, потому что информация о тарифах будет собираться автоматически с сайтов сотовых операторов, а информация о клиентах будет также в автоматическом порядке собираться с сайта-агрегатора, при этом, администратор, в данном случае будет следить за правильностью информации и, если это необходимо, дополнять или редактировать её. К тому же, простое предоставление доступа к базе данным другим пользователям в данном случае не является безопасным решением.

Следовательно, разграничение на права доступа и роли не является необходимым.

# 

# **3. Физическое проектирование**



Рис. 2. Физическая модель

В соответствии со схемой физической модели были созданы скрипты для создания объектов БД: таблиц, первичных и внешних ключей. Скрипты создания триггеров в “Приложение”.

Необходимо рассчитать объём БД, пользуясь формулами:

1. Рассчитывается длина записи в таблицы. Длина 1 записи = длина поля 1 + длина поля 2 + …
2. Рассчитывается длина индекса. Длина индекса=длина адресного указателя + длина ключевого поля (где длина адресного указателя = 4 байта)
3. Рассчитывается объем таблицы. Объем таблицы = (Длина 1 записи + Длина индекса) \* кол-во предполагаемых строк в таблице
4. Рассчитывается объем всей БД. Объем БД= Объем таблицы 1 + Объем таблицы 2 + Объем таблицы 3 + …

Длины полей приведены в таб 1. Тип данных date занимает 7 байт. Первичный ключ порождает индексную таблицу - 2 столбца в RAM: адрес на указатель - 4 байта и столбец первичного ключа 4. При расчёте длины записи в таблице и длины индекса мы дважды просчитываем PK, так как он хранится в двух экземплярах в БД: в адресной таблице, и в таблице, которая создавалась нами целенаправленно. В качестве примера рассчитаем вес таблицы tarif:

Длина записи = 30+2+2+2+2+2+2+2+5+2+30=81 байт

Длина индекса = 4+2=6 байт

Предполагаемое число записей = 100 штук

Объём таблицы = (81+6) \* 100 = 8700 байт

Аналогично со всеми остальными:

| Имя таблицы | Длина записи  (байт) | Предполагаемое число записей  (кол) | Объёмы таблицы (байт) |
| --- | --- | --- | --- |
| region | 36 | 90 | 3240 |
| roaming | 36 | 5 | 180 |
| oper | 12 | 15 | 180 |
| tarif | 87 | 100 | 8700 |
| client | 46 | 100000 | 4600000 |
| comment | 42 | 10000 | 420000 |
| histAllTar | 19 | 1000000 | 19000000 |
| infTarUser | 20 | 100000 | 2000000 |

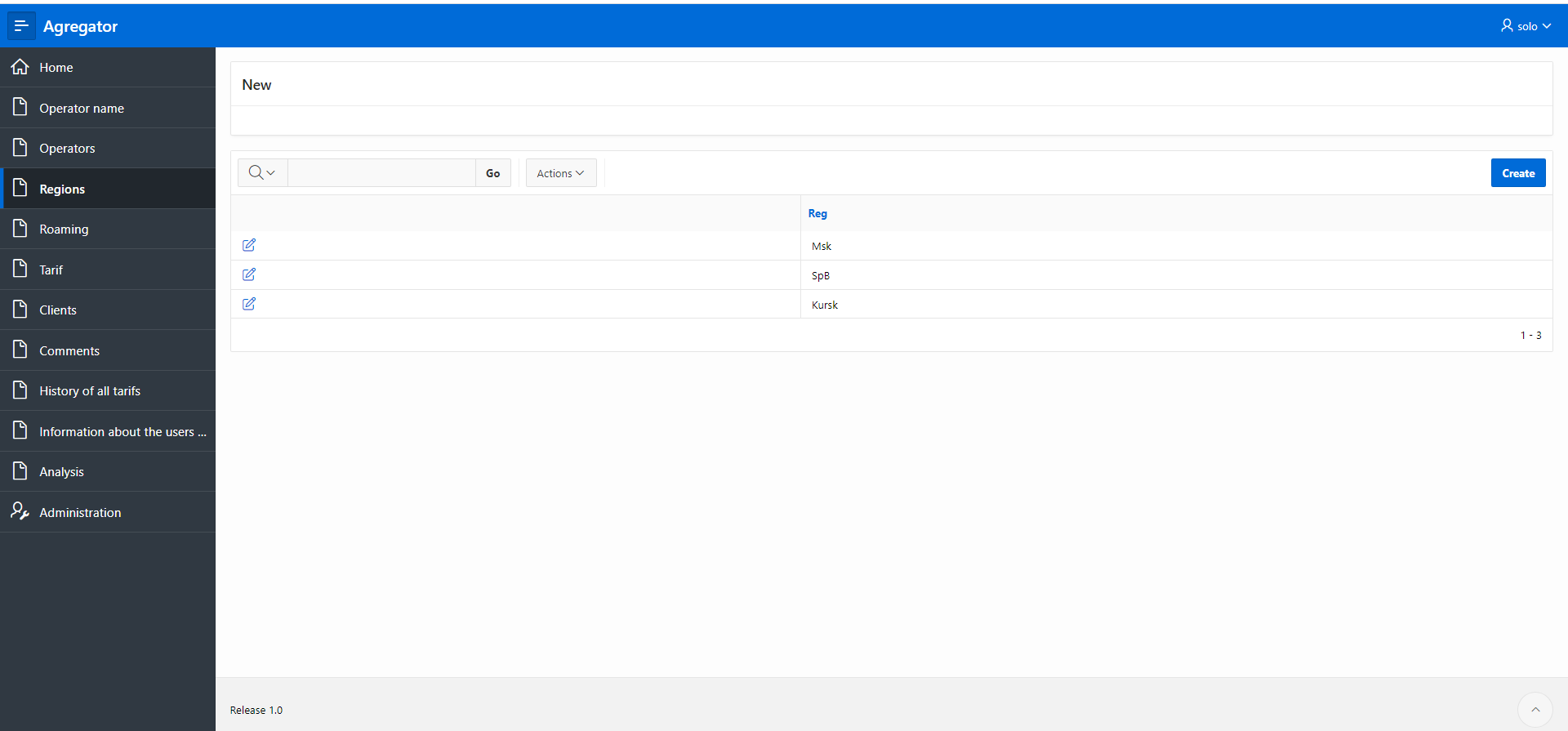
Таб. 1 Расчёт объёма БД

Объем БД: 26034540 байт = 26 Мбайт

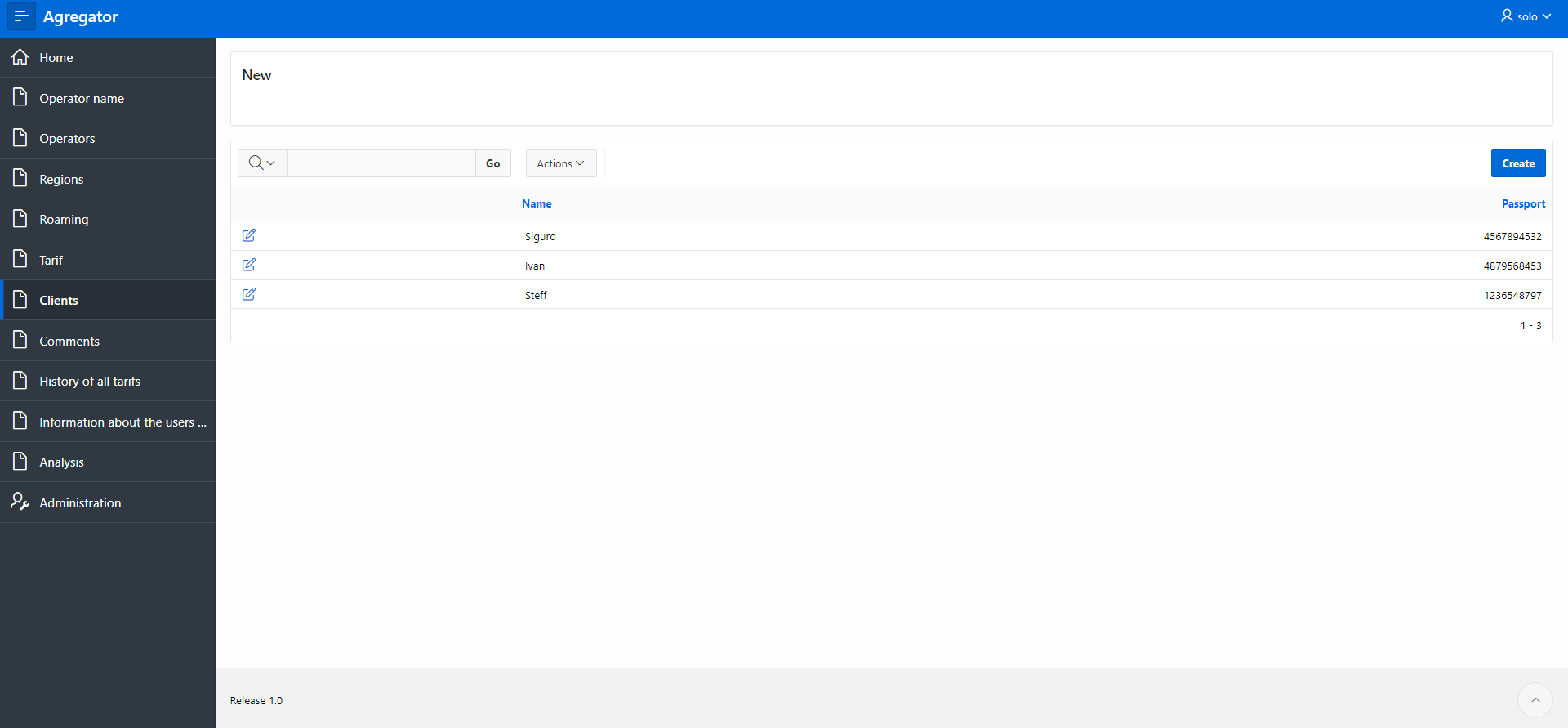
# 

# **4. Демонстрация приложения**

Чтобы начать работу, сперва необходимо, чтобы в таблицах была информация. Пример заполнения таблицы регионы:

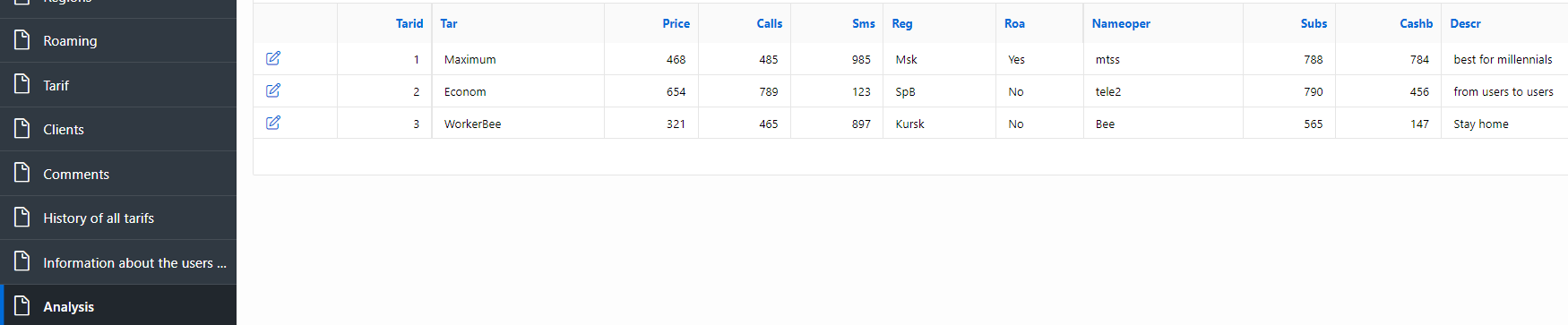


Пример заполнения таблицы клиенты:

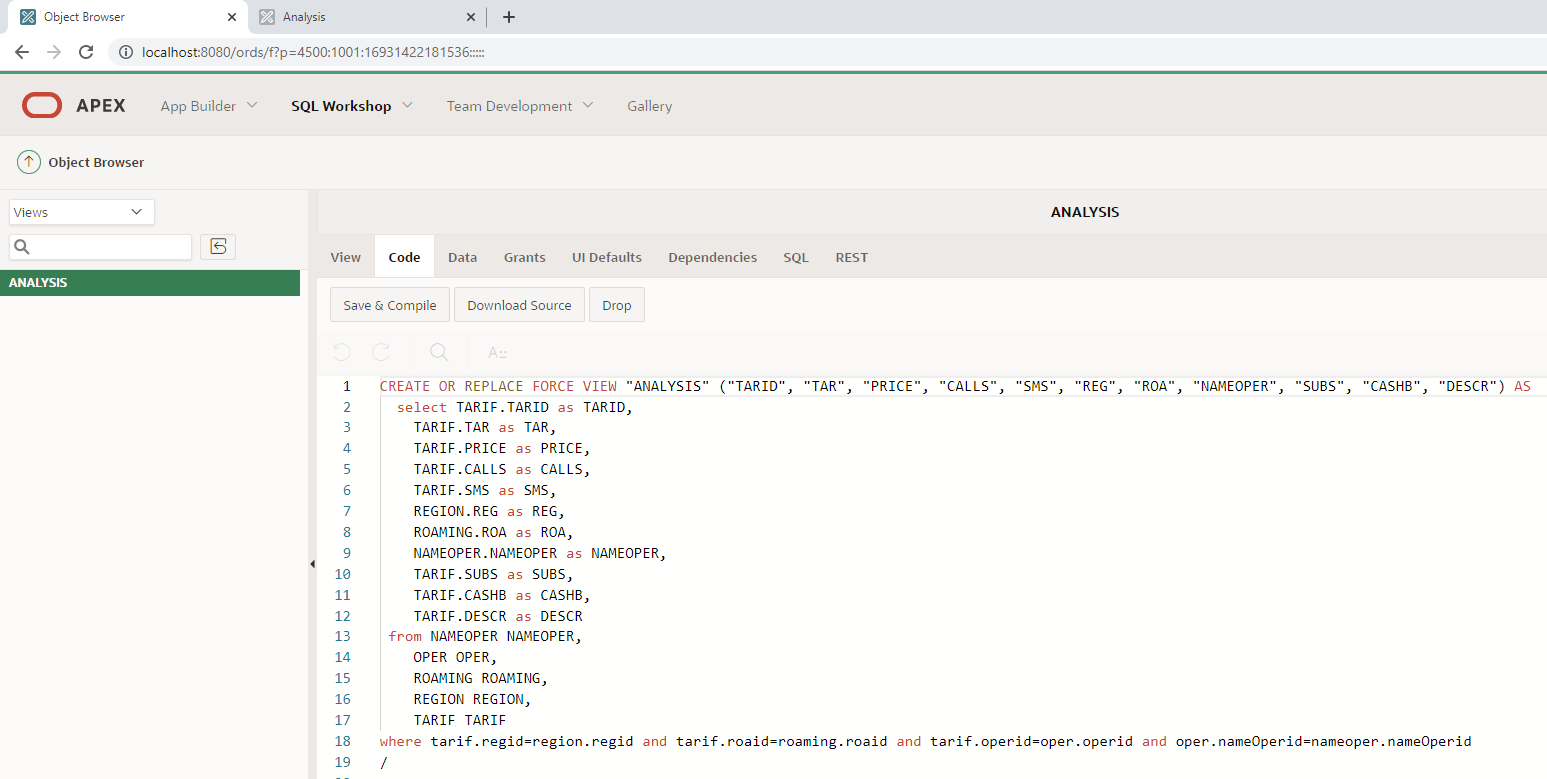


Остальные таблицы заполняются по аналогии.

Самой главной вещью приложения является представление, которое непосредственно отображает нам имеющиеся тарифы, их название, цену, регион, где они работают, и каким оператором в нем представлены:

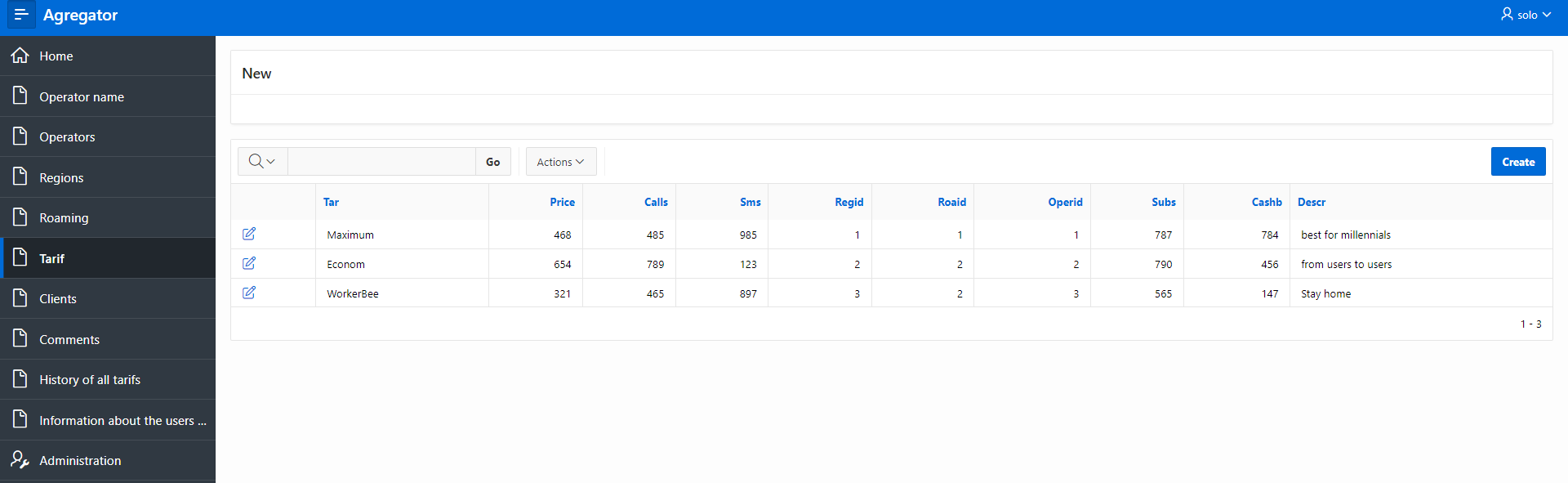


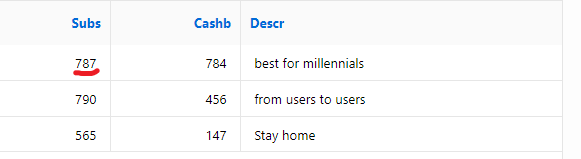
Данное представление реализуется следующим образом:



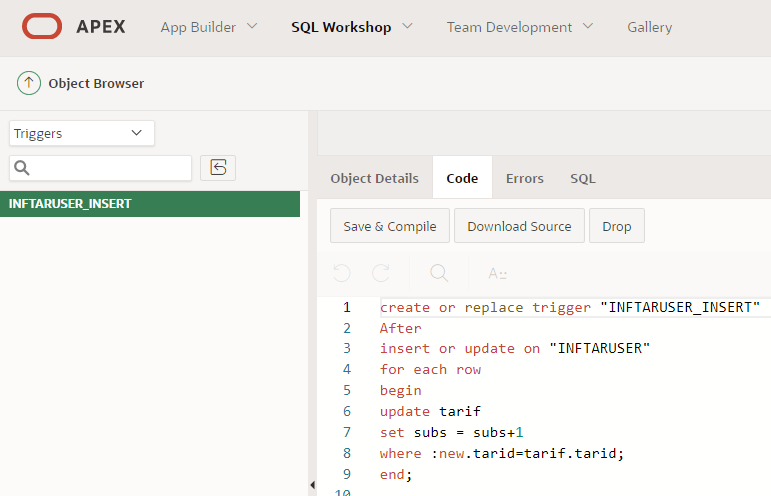
Также, помимо представления, существуют различные триггеры, например:

Имеется таблица с тарифами, где у каждого тарифа есть колонка, с количеством людей, которые им пользуются.

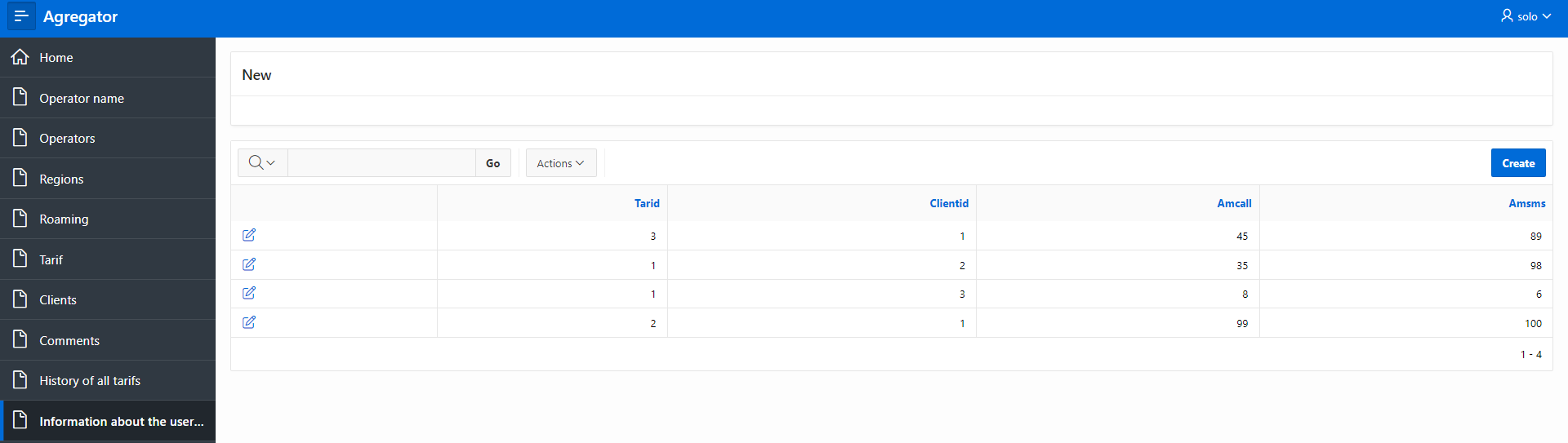




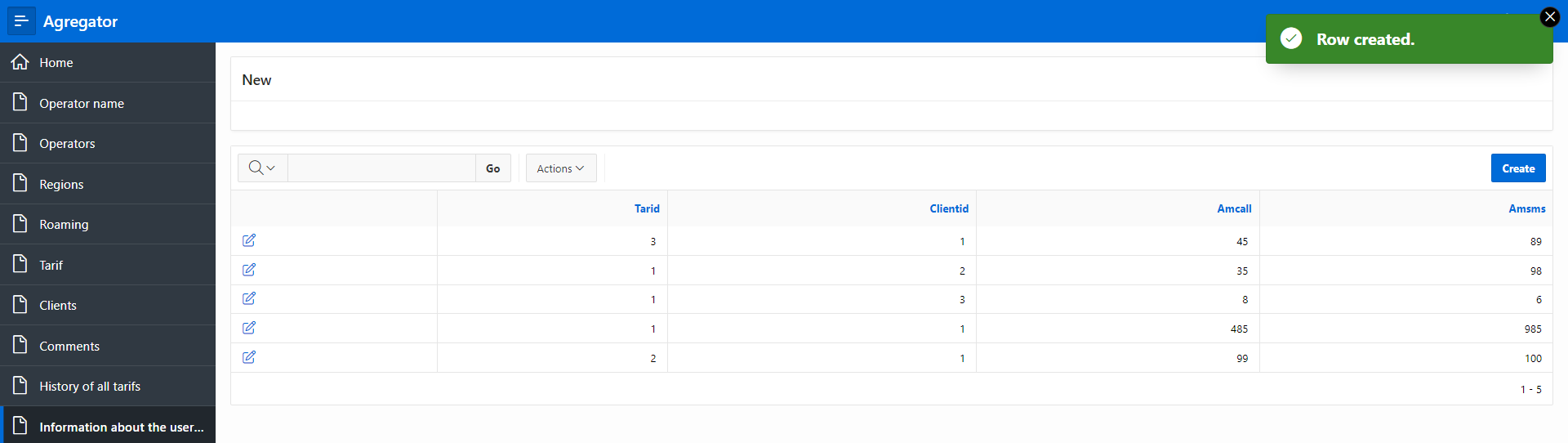
Имеется триггер, которые учитывает присоединение клиента к тарифу:



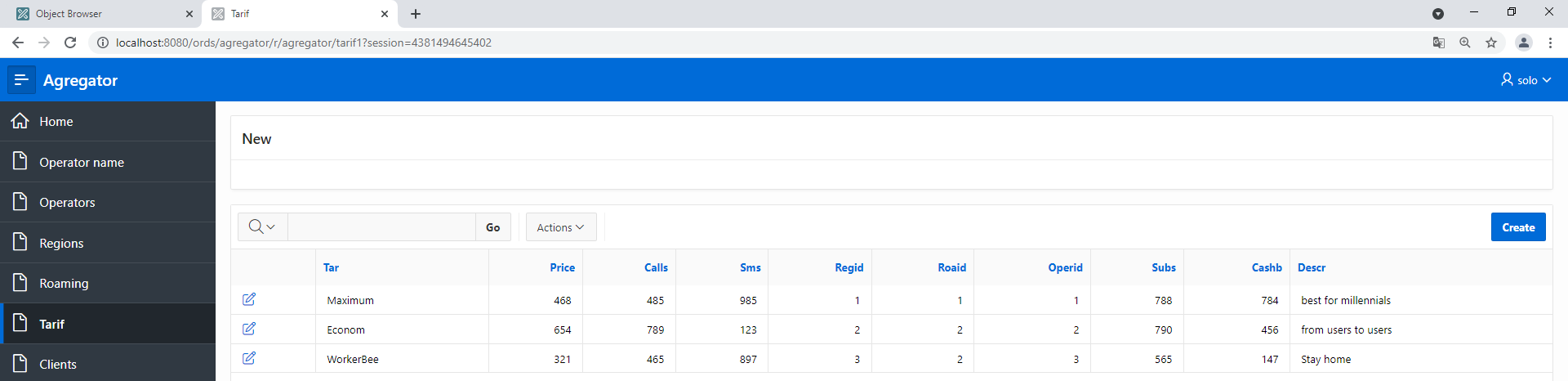
Клиенты и какими тарифами они пользуются:

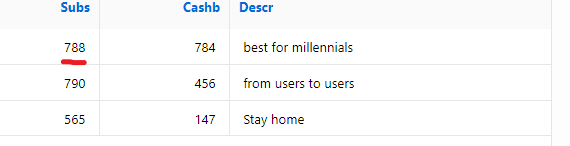


Новый клиент начинается пользоваться первым тарифом:

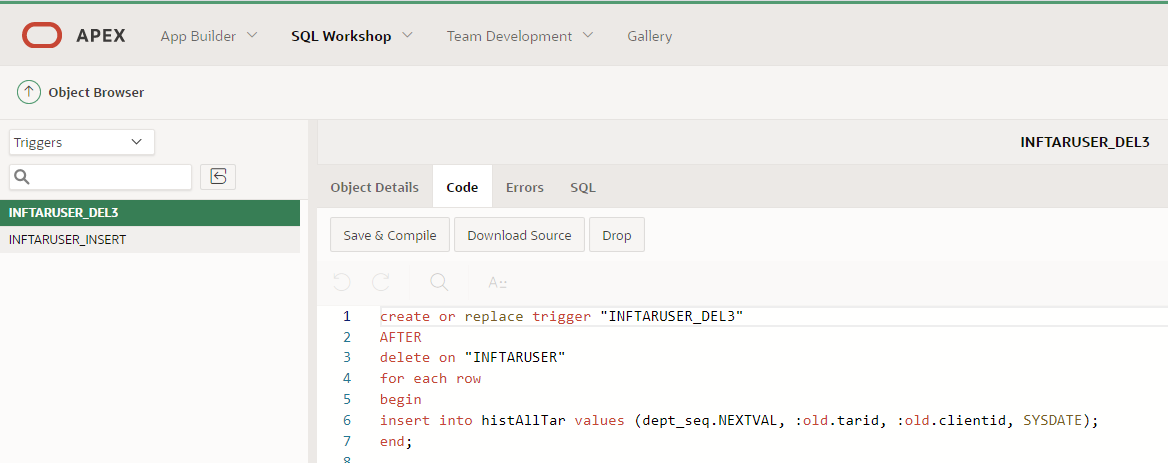


И, следовательно, количество абонентов (subs-subscribers) увеличилось:

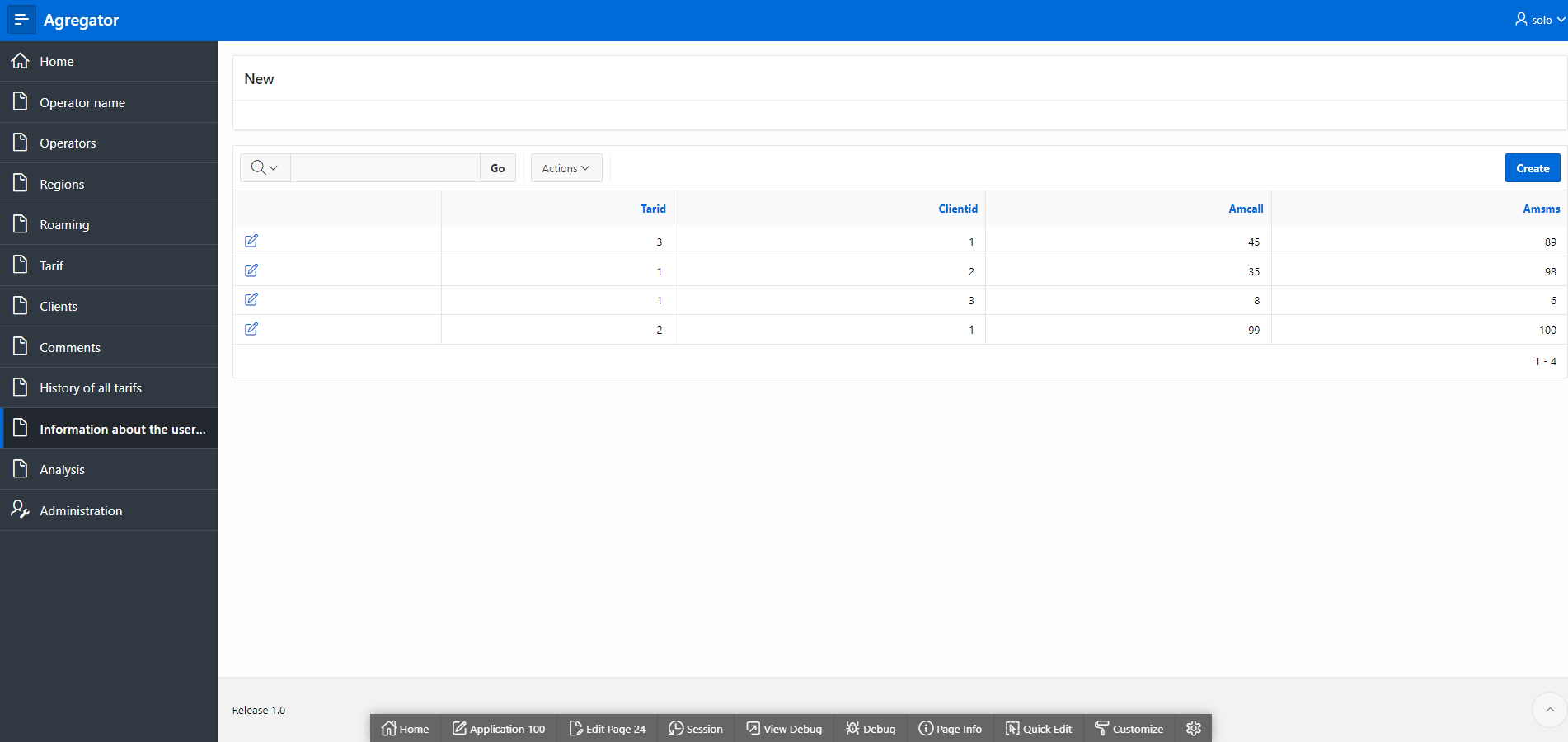




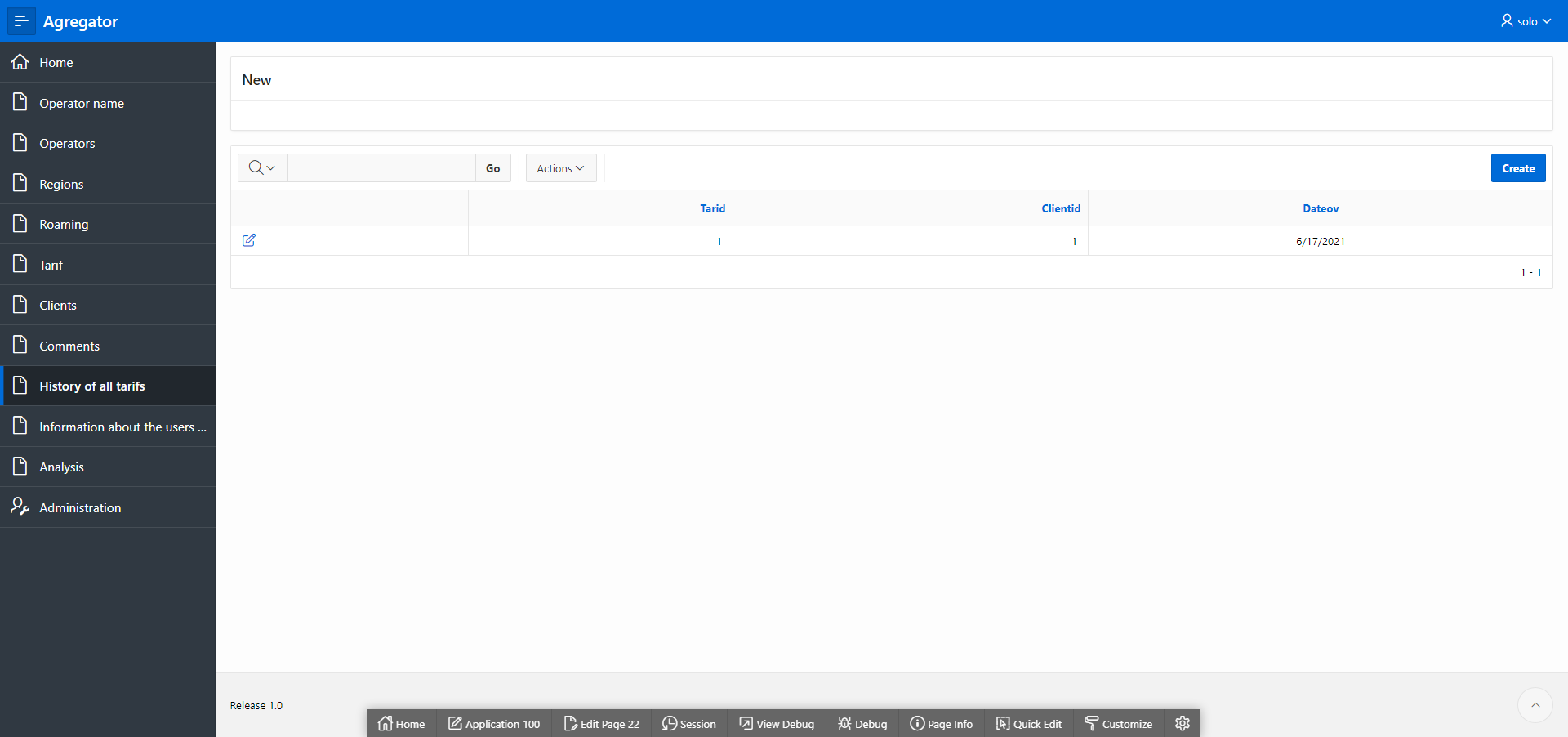
Триггер на удаление :



Он работает следующим образом: если мы, допустим, удалим из “текущих тарифов клиентов”, например четвертую строку, в таком случае, сработает триггер, а таблица примет следующий вид:



Данный триггер добавит пользователя и тариф, который он использовал, в таблицу историй тарифов, которые использовали клиенты:



Таким образом, можно реализовать функцию автоматического слежения за тем, какой клиент, какой тариф прекратил использовать, это облегчит работу при анализе истории.

# 

# **Заключение**

Таким образом была достигнуты цели по автоматизации процесса анализа тарифных планов сотовых компаний, что позволит сэкономить время и нервы не только тому, кто прямо сейчас хочет выбрать себе тариф, но и другим людям, которые также в этом заинтересованы.

В данной работе с помощью oracle и apex была создана структура, позволяющая в более легкой и эффективной форме анализировать тарифы сотовых компаний.

Результатом стала база данных, в которой могут храниться тарифы и прочие сведения о них, а благодаря удобному интерфейсу, можно проводить сравнительный анализ и выбирать для себя то, что больше нравится.

# 

# **Список использованной литературы**

1. Фролова Е.А. Программирование на РТ/ЗОГ. [Электронный ресурс]: Учебно-методическое пособие по дисциплине Базы данных/ — М.: Брис-М, 2016. — 24 с. - Режим доступа: http://lib.mtuci.ru/libdocs/ec 1/dbi/dl/download.php?book\_id=10398 (ЭБС МТУСИ, доступ по сети МТУСИ).
2. Фролова Е.А. Разработка приложения баз данных в среде СУБД. Developer 2000 [Текст] Учебное пособие/ — М.:Брис-М, 2010 - 133 с. – Режим доступа: http://lib.mtuci.ru/libdocs/ec1/dbi/search.php (ЭБС МТУСИ, доступ по сети МТУСИ).
3. Фролова Е.А. Практикум по СУБД Огас!е [Текст] Учебное пособие/ — М.:Брис-М, 2004 - 54 с. — Режим доступа: http://lib.mtuci.ru/libdocs/ec 1/dbi/search.php (ЭБС МТУСИ, доступ по сети МТУСИ).
4. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений/ Под ред. Проф. А.Д. Хомоненко. — CIIб: KOPOHA принт, 2000. <https://studfiles.net/preview/6354063/>
5. <https://betacode.net/10345/oracle-apex-tutorial-for-beginners> (Дата обращения: 12.05.2021)
6. <https://ppt.ru/fls/10466/instruktsiya-po-zapolneniyu.jpg> (Дата обращения: 17.05.2021)
7. <https://diamedclinic.ru/wa-data/public/site/img/%D0%91%D0%BE%D0%BB%D1%8C%D0%BD%D0%BE%D1%87%D0%BD%D1%8B%D0%B9%20%D0%B4%D0%BB%D1%8F%20%D1%81%D0%B0%D0%B9%D1%82%D0%B0%201-2_thumb_730x_prop.jpg> (Дата обращения: 17.05.2021)
8. <https://eopearhiiv.edu.ee/e-kursused/eucip/arendus_vk/243___.html> (Дата обращения: 17.05.2021)

# 

# **Приложение**

**Создание таблиц:**

CREATE TABLE region(

regid number(3) constraint regid\_pk primary key,

reg varchar(30));

CREATE TABLE roaming(

roaid number(3) constraint roaid\_pk primary key,

roa varchar(30));

~~CREATE TABLE nameOper(~~

~~nameOperid number(3) constraint nameOperid\_pk primary key,~~

~~nameOper varchar(30));~~

CREATE TABLE oper(

operid number(3) constraint operid\_pk primary key,

nameOper*~~id number(3) constraint nameOperid\_fk references nameOper (nameOperid)~~* varchar(30),

score number(3),

CONSTRAINT operUniq UNIQUE (nameOper));

CREATE TABLE tarif(

tarid number(3) constraint tarid\_pk primary key,

tar varchar(30),

price number(4),

calls number(4),

sms number(4),

regid number(3) constraint regid\_fk references region (regid),

roaid number(3) constraint roaid\_fk references roaming (roaid),

operid number(3) constraint operid\_fk references oper (operid),

subs number(10),

cashb number(4),

descr varchar(30));

CREATE TABLE client(

clientid number(3) constraint clientid\_pk primary key,

name varchar(30),

passport number(10));

CREATE TABLE comments(

comid number(3) constraint comid\_pk primary key,

tarid number(3) constraint tarid\_fk references tarif (tarid),

clientid number(3) constraint clientid\_fk references client (clientid),

contents varchar(30));

CREATE TABLE histAllTar(

histid number(3) constraint histid\_pk primary key,

tarid number(3) constraint tarid\_fk2 references tarif (tarid),

clientid number(3) constraint clientid\_fk2 references client (clientid),

dateov date,

CONSTRAINT histUniq UNIQUE (tarid, clientid, dateov));

CREATE TABLE infTarUser(

infid number(3) constraint infid\_pk primary key,

tarid number(3) constraint tarid\_fk3 references tarif (tarid),

clientid number(3) constraint clientid\_fk3 references client (clientid),

amCall number(4),

amSms number(4),

CONSTRAINT infUniq UNIQUE (tarid, clientid));

**Создание представления:**

CREATE OR REPLACE FORCE VIEW "ANALYSIS" ("TARID", "TAR", "PRICE", "CALLS", "SMS", "REG", "ROA", "NAMEOPER", "SUBS", "CASHB", "DESCR") AS

select TARIF.TARID as TARID,

TARIF.TAR as TAR,

TARIF.PRICE as PRICE,

TARIF.CALLS as CALLS,

TARIF.SMS as SMS,

REGION.REG as REG,

ROAMING.ROA as ROA,

NAMEOPER.NAMEOPER as NAMEOPER,

TARIF.SUBS as SUBS,

TARIF.CASHB as CASHB,

TARIF.DESCR as DESCR

from NAMEOPER NAMEOPER,

OPER OPER,

ROAMING ROAMING,

REGION REGION,

TARIF TARIF

where tarif.regid=region.regid and tarif.roaid=roaming.roaid and tarif.operid=oper.operid and oper.nameOperid=nameoper.nameOperid

/

**Создание триггеров:**

На ввод данных в таблицу текущие тарифы клиента:

create or replace trigger "INFTARUSER\_INSERT"

After

insert or update on "INFTARUSER"

for each row

begin

update tarif

set subs = subs+1

where :new.tarid=tarif.tarid;

end;

На удаление данных из таблицы текущие тарифы клиента:

create or replace trigger "INFTARUSER\_DEL3"

AFTER

delete on "INFTARUSER"

for each row

begin

insert into histAllTar values (dept\_seq.NEXTVAL, :old.tarid, :old.clientid, SYSDATE);

end;