

ОБУЧАЮЩАЯ СИСТЕМА ДЛЯ ДВУНАПРАВЛЕННОГО ПРЕОБРАЗОВАНИЯ РЕЛЯЦИОННЫХ И SQL ВЫРАЖЕНИЙ

МИХАЙЛЕНКО Евгений, ТУРЧАК Даниил, ВИЗИР Иван, КУЛЧУ Татьяна
Coordonator: САРАНЧУК Дориан

Технический Университет Молдовы

Аннотация: В работе представлена система «SQLtoRA» осуществляющая преобразование запросов написанных на языке реляционной алгебры в язык SQL и обратно. Система реализована на основе Qt версии 4.8.1 с использованием библиотеки для генерации парсеров Boost Spirit. Цель продукта - упростить процесс изучения реляционной алгебры для начинающих специалистов.

Ключевые слова: реляционная алгебра, реляционная база, SQL, парсер, Boost Spirit.

1. ВВЕДЕНИЕ

Реляционная алгебра представляет собой абстрактный язык запросов в реляционной БД. Она состоит из набора операндов, использующих отношения в качестве аргументов и возвращающих отношения в качестве результата. Язык реляционной алгебры не получил распространения в современных СУБД, из-за этого изучение данного языка лишено мотивации со стороны молодых специалистов.

Программный продукт «SQLtoRA» реализован с целью облегчения изучения языка реляционной алгебры. Главный способ упростить процесс изучения, это преобразование выражений известного многим молодым специалистам языка SQL, в выражения реляционной алгебры и наоборот. Такой подход поможет построить связь между выражениями обоих языков, тем самым специалистам не придётся запоминать новые для них операции запросов.

2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

В начале 70-х годов SQL являлся лишь языком запросов. Он, по сути содержал только оператор SELECT, который позволял формулировать запросы для выборки данных из базы. Современный SQL является язык программирования, применяемый для создания, модификации и управления данными в реляционных базах данных [1].

Реляционная алгебра (РА) – абстрактный язык запросов в реляционной БД. Не получившая распространения РА мало применима на практике и значительно уступает языку SQL. Однако в связи с тем, что РА базируется на теории множеств, её изучение в качестве теоретической базы вполне оправдано. Кроме операции из теории множеств РА обладает специальными реляционными операциями: проекция, селекция, соединение, тэта-соединение, полу-соединение, деление, транзитивное замыкание и расщепление. Любую операцию, исключая дополнение, можно представить запросом на языке SQL.

Реляционная алгебра состоит из набора операндов, использующих отношения в качестве аргументов и возвращающих отношения в качестве результата. Поэтому реляционная алгебра является замкнутой относительно понятия отношения [2]:

$$r = f(r_1, r_2, \dots, r_n)$$

Данное свойство замкнутости реляционной алгебры позволяет подставлять в качестве аргументов в реляционные операторы другие реляционные операторы:

$$r = f(f_1(r_1, r_2, \dots, r_n), f_2(s_1, s_2, \dots, s_m)).$$

Все операции реляционной алгебры делятся на две большие категории:

- теоретико-множественные операции – это операции заимствованные из теории множеств. К ним относятся объединение, пересечение, разность, декартово произведение, дополнение отношений.
- специальные реляционные операции: проекция, селекция, соединение, Θ -соединение, полу-соединение, деление.

С точки зрения количества аргументов операции реляционной алгебры бывают унарные и бинарные.

Бинарные операции в некоторых случаях требуют совместимости отношений. Чтобы отношения были совместимыми они должны иметь совместимые схемы.

Два отношения $r(R)$ и $s(S)$ являются совместимыми, если между схемами R и S существует взаимно однозначная корреспонденция f , таким образом, чтобы домен атрибута из первой схемы, равнялся домену атрибута из второй схемы ($\forall A \in R, \exists B \in S, dom(A)=dom(B)$) и имело место выражение [2]:

$$B=f(A) \text{ and } A=f^{-1}(B).$$

3. ПРИМЕНЕНИЕ НА ПРАКТИКЕ

Ниже представлены несколько примеров использования программного продукта "SQLtoRA". Представленные примеры будут базироваться на структуре БД, состоящая из 3 отношений: (рис.1)

orders(OrderID, ClientID, Volume, Price),
register(ClientID, Account, PersonalID),
users(PersonalID, Name, Age, Sex).

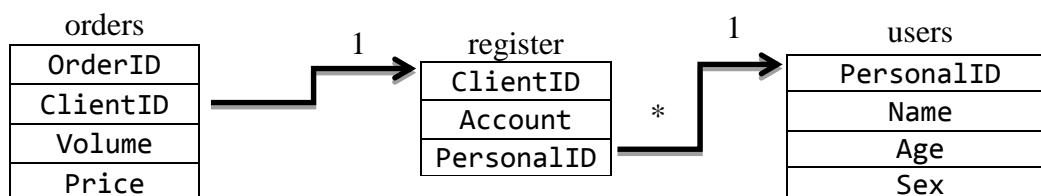


Рисунок 1 – Диаграмма БД

Пример №1

Пусть требуется вывести имена, номера счетов и возраст лиц, не достигших 25 лет. Запрос на языке реляционной алгебры:

$$\pi_{\text{Name, Account, Age}}(\sigma_{\text{Age} < 25}(\text{register} \bowtie \text{users})).$$

3 2 1

Порядок выполнения операций: 1 – естественное соединение, 2 – селекция, 3 – проекция.

Запрос на языке SQL:

```
select Name, Account, Age from
  (select * from register natural join users
   where Age < 25);
```

Пример №2

Пусть требуется получить список всех заказов клиента со счетом номер 7585351. Запрос на языке реляционной алгебры:

$$\pi_{\text{OrderID, Price}}(\sigma_{\text{account}=7585351}(\text{register}) \bowtie \text{orders});$$

3 1 2

Порядок выполнения операций: 1 – селекция, 2 – естественное соединение, 3 – проекция.

Запрос на языке SQL:

```
select OrderID, Price from
  (select * from register where Account = 7585351) natural join orders;
```

Пример №3

Осуществим обратную конвертацию (из SQL в РА).

Пусть требуется вывести информацию о заказах количеством свыше 500 единиц или заказы с ценой свыше 1500, исключая заказы клиента с ID=5.

Запрос на языке SQL:

```
select * from orders where Volume > 500
```

```

union
select * from orders where Price > 1500
except
select * from orders where ClientID = 5;

```

Запрос на языке реляционной алгебры:

$$\sigma_{Volume>500}(orders) \cup \sigma_{Price>1500}(orders) \setminus \sigma_{ClientID=5}(orders);$$

Этот же запрос можно сформулировать иначе:

```

select * from orders where Volume>500 or Price>1500 and ClientID <> 5;

```

Тогда запрос на языке реляционной алгебры будет выглядеть следующим образом:

$$\sigma_{Volume>500 \vee Price>1500 \wedge ClientID \neq 5}(orders);$$

Современные СУБД способны обнаруживать избыточность запросов и самостоятельно оптимизировать их [3].

4. ОПИСАНИЕ СИСТЕМЫ

Приложение «SQLtoRA» осуществляет преобразование запросов написанных на языке реляционной алгебры в язык SQL и обратно. Программа была реализована на основе Qt версии 4.8.1 [4,5] с использованием библиотеки для генерации парсеров Boost Spirit [6].

Реализованный программный продукт представляет собой оконное приложение с простым и удобным интерфейсом. После запуска появляется главное окно, в котором возможно открыть одно или более окна для редактирования, конвертирования и запуска запросов на языке SQL или реляционной алгебры. По умолчанию открыта одна пустая вкладка с приглашением ввести запрос.

Главное окно программы (рис.2) является простым окном с меню 1. Ниже приведены его пункты и детальное описание каждого.

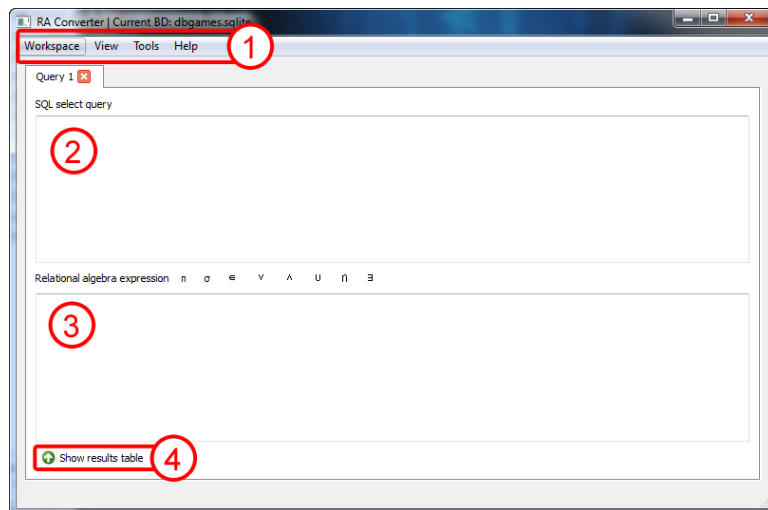


Рисунок 2 — Главное окно программы

Поле 2 служит для ввода и редактирования выражений на языке SQL, которые будут преобразовываться в выражения реляционной алгебры в поле 3. Последнее может само использоваться для ввода выражений реляционной алгебры, которые будут преобразованы в SQL и выведены в верхнем поле. Также под вторым полем имеется кнопка, расширяющая интерфейс вкладки 4. В дополнительном поле отображается результат запроса, выполненного во временной БД.

Все окна-вкладки, которые открыты в данный момент в редакторе, а также все таблицы текущей базы данных представляют собой рабочее пространство. В программе предусмотрена возможность сохранять отдельные запросы в файлах, а также сохранение всего рабочего пространства.

При вводе выражения реляционной алгебры в нижнее поле можно использовать специальные символы (рис. 3).

Для редактирования баз данных в программу встроен редактор, для его вызова используется команда меню Tools > DB Editor. Редактор представляет собой самостоятельное приложение для создания, просмотра, редактирования и удаления таблиц в базе данных. Более того, редактор представляет удобный режим редактирования *spreadsheet* — то есть отображение всех кортежей отношения и возможность модификаций их на месте и добавления новых кортежей.

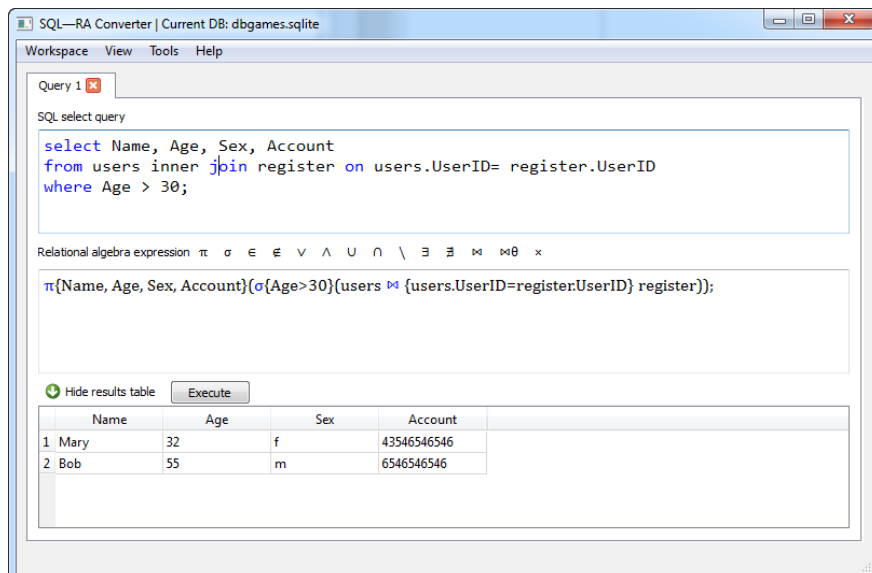


Рисунок 3 — Вкладка редактирования запросов.

Заключение

1. Была реализована система осуществляющая преобразование запросов из языка SQL в выражения реляционной алгебры и обратно.
2. Реализована возможность исполнения запросов реляционной алгебры, путём трансляции их в язык SQL и исполнения их на существующих БД.
3. Возможность транслировать SQL в выражения реляционной алгебры должно способствовать изучению реляционной алгебры у молодых специалистов.
4. Трансляция реляционной алгебры в SQL открывает новые возможности в составление запросов. В некоторых случаях составление запросов на языке реляционной алгебры значительно проще, чем на языке SQL. Благодаря конвертору подобные запросы можно будет составить значительно быстрее.

Библиография

1. Кириллов В.В., Громов Г.Ю. *Введение в реляционные базы данных*. Санкт-Петербург: БХВ-Петербург, 2008, стр. 451.
2. Кузнецов С. Д., *Основы баз данных*. — 2-е изд.: Интернет-Университет Информационных Технологий, 2007, стр. 484.
3. Дейт, Кристофер. *SQL и реляционная теория. Как грамотно писать код на SQL*. : Символ-Плюс, 2010.
4. *Qt-digia*. URL: <http://doc.qt.digia.com>
5. *Qt-project*. URL: <http://qt-project.org/wiki>
6. Саранчук Дориан, Визир Иван, Турчак Даниил, Михайленко Евгений. *Boost.Spirit u разбор SQL Select*. Conferința Științifică a Colaboratorilor, Doctoranzilor și Studenților UTM, 23 noiembrie 2013, UTM, Chișinău, Republica Moldova.