

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по научно-исследовательской работе
Тема: Программная реализация модели базы данных TransRelational

Студентка гр. 4303

Полушина Ю.И.

Преподаватель

Фомичёва Т.Г.

Санкт-Петербург

2019

**ЗАДАНИЕ
НА НИР**

Студентка Полушина Ю.И.

Группа 4303

Тема работы: Программная реализация модели базы данных TransRelational

Дата сдачи реферата: 25.12.2019

Дата защиты реферата: 25.12.2019

Студентка

Полушина Ю.И.

Преподаватель

Фомичёва Т.Г.

АННОТАЦИЯ

В работе рассматривается техническое описание модели TransRelational, которая может быть использована в качестве средства реализации реляционной модели. Описываются уровни представления данных, взаимодействия между ними и преобразования, с помощью которых логический уровень отображается на физический, и наоборот. На основе анализа недостатков существующих систем выявляются преимущества модели TransRelational, проблемы, которые она может решить, например, может обеспечить независимость от физических данных, а также области применения описанной модели. В качестве заключения описывается алгоритм преобразования стандартной реляционной модели в модель TransRelational.

SUMMARY

The paper introduces the technical description of the TransRelational model, which can be used for implementing the relational model. The three-level architecture of databases and the transformations between levels, which reflect conceptual level to the physical level and vice versa, were described. Based on the analysis of the shortcomings of existing systems, the advantages of the TransRelational model are revealed. Also, TR model can improve existing solutions, for example, providing independence from physical data. As conclusion, the algorithm of transformation relational model to TransRelational model is described.

СОДЕРЖАНИЕ

	Введение	5
1.	ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ	6
1.1.	Уровни представления данных	6
1.2.	Модель TransRelational	6
1.3.	Преимущества TR модели	8
2.	ОПИСАНИЕ МЕТОДА РЕШЕНИЯ	9
2.1.	Алгоритм преобразования в модель TR	9
2.2.	Описание используемых технологий	11
	Заключение	12
	Список использованных источников	13

ВВЕДЕНИЕ

Реляционные базы данных используются повсеместно очень давно. Они стали популярными благодаря успешным реализациям реляционных моделей в системах управления, оказавшимся удобными для работы с данными.

Модель TransRelational(TR) - это подход к реализации реляционной модели. Модель TR разработана Стивеном Тарином на основе его запатентованной технологии, получившей название преобразования Тарена. Предполагается, что модель TR предлагает улучшенную физическую независимость данных и производительность по сравнению с различными популярными стратегиями для реализации реляционных моделей.

В настоящее время не существует СУБД основанной на данной модели.

Цель данной работы состоит в реализации инструмента, позволяющего осуществить преобразование стандартной реляционной модели в модель TR, а также предоставляющего возможность работы с таблицами описанной модели.

Для достижения цели необходимо решить следующие **задачи**:

- Провести обзор предметной области.
- Рассмотреть способы преобразования между уровнями данных в существующих СУБД и провести сравнительный анализ.
- Выявить и сформулировать требования к разрабатываемому инструменту.
- Описать алгоритм преобразования из реляционной модели в TR модель.
- Описать операции реляционной алгебры, необходимые для базовой работы с БД.
- Реализовать инструмент, удовлетворяющий всем указанным критериям.
- Провести тестирование разработанного инструмента.

Объектом исследования является модель TransRelational.

Предметом исследования является инструмент, позволяющий работать с TR моделью.

1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Уровни представления данных

Современные подходы к созданию БД предполагают их трёхуровневую организацию.

На самом верхнем уровне может быть множество моделей. Этот уровень определяет точку зрения на БД отдельных пользователей приложений. Каждое приложение видит и обрабатывает только те данные, которые необходимы именно ему.

На концептуальном уровне БД представлена в наиболее общем виде, который объединяет все внешние представления предметной области. При разработке концептуальной модели усилия направлены на структуризацию данных и выявление взаимосвязей, без рассмотрения особенностей реализации и эффективности разработки.

Внутренний (физический) уровень – это данные, расположенные на внешних носителях информации. Внутренняя модель определяет размещение данных, методы доступа, технику индексирования.

Трёхуровневая организация БД позволяет обеспечить логическую и физическую независимость при работе с данными. Логическая независимость предполагает возможность изменения одного приложения, без корректировки других приложений, работающих с этой же БД.

Физическая независимость предполагает возможность переноса хранимой информации с одних носителей на другие при сохранении работоспособности всех приложений, использующих эту БД.

1.2. Модель TransRelational

Метод преобразования Тарена, на который распространяется патент США [1], предназначен для использования в качестве технологии реализации для систем хранения и выборки данных многих типов, включая, например, системы хранилищ данных, инструментальные средства разработки данных, системы

SQL, машины поиска Web, системы, основанные на использовании языка XML и т.д. В отличие от этого, модель TransRelational представляет собой пример применения этой более общей технологии для реализации, в частности, реляционных систем.

Модель TR удобно рассматривать с точки зрения ее использования для решения задачи — обеспечения независимости от физических данных. Обеспечение независимости от данных означает проведение четкого различия между логическим и физическим уровнями системы, а проведение такого четкого различия, в свою очередь, требует наличия средств преобразования между этими двумя уровнями, с помощью которых логический уровень отображается на физический, и наоборот. Но в большинстве современных СУБД применяется такой способ преобразования, который может рассматриваться почти как взаимно-однозначный. В подобной системе все, что хранится на физическом уровне, можно рассматривать в качестве непосредственного отображения того, что пользователь видит на логическом уровне. Одним из следствий из указанного факта является то, что подобные системы в действительности не обеспечивают настолько уж значительную независимость от данных. Еще одним следствием является то, что обязательно приходится располагать данные в памяти только в одной физической последовательности, а в результате этого возникает необходимость в использовании индексов и других избыточных структур для поддержки доступа к данным, расположенным в той последовательности, которая отличается от требуемой. Кроме того, возникает и такое следствие, что для достижения приемлемой производительности требуется сложная оптимизация. Наконец, важным следствием является то, что задача администрирования базы данных становится намного более сложной, чем должна была быть.

1.3. Преимущества TR модели

В отличие от большинства СУБД, в модели TR используются намного более совершенные средства преобразования. Ниже описаны некоторые непосредственные следствия из этого факта.

- Модель TR обеспечивает гораздо большую независимость от данных по сравнению с той, которая достигнута или может быть достигнута в системах с непосредственным отображением.
- В модели TR данные по существу хранятся во многих разных физических последовательностях одновременно, поэтому исключается необходимость в использовании индексов и тому подобных структур.
- Производительность модели TR на несколько порядков выше по сравнению с системами непосредственного отображения.
- Администрирование системы в значительной степени упрощается, поскольку гораздо реже приходится принимать субъективные решения.

2. ОПИСАНИЕ МЕТОДА РЕШЕНИЯ

2.1. Алгоритм преобразования в модель TR

Допустим, что r — запись некоторого файла на файловом уровне. В таком случае справедливо утверждение: хранимая форма записи r состоит из двух логических различных частей — множества значений полей и множества «связующих» данных, позволяющих связать друг с другом эти значения полей, причем для физического хранения каждой из этих частей можно воспользоваться широким спектром возможностей [2].

В системах с непосредственным отображением эти две части хранятся вместе. В таких системах связующая информация задается по принципу физической близости. В отличие от этого, в модели TR эти две части хранятся отдельно — значения полей находятся в таблице значений полей, а связующая информация — в таблице реконструкции записей. А именно такое разделение является основным источником многочисленных преимуществ, которые способна предоставить модель TR.

На рис. 2.1. приведен пример таблицы с данными, ниже приведено преобразование данной таблицы в таблицу значений полей и таблицу реконструкции записей.

Последовательность полей:		1	2	3	4																													
Последовательность записей:		<table><tr><th>#</th><th>Name</th><th>Status</th><th>City</th></tr><tr><td>1</td><td>S4</td><td>Clark</td><td>20</td><td>London</td></tr><tr><td>2</td><td>S5</td><td>Adams</td><td>30</td><td>Athens</td></tr><tr><td>3</td><td>S2</td><td>Jones</td><td>10</td><td>Paris</td></tr><tr><td>4</td><td>S1</td><td>Smith</td><td>20</td><td>London</td></tr><tr><td>5</td><td>S3</td><td>Blake</td><td>30</td><td>Paris</td></tr></table>				#	Name	Status	City	1	S4	Clark	20	London	2	S5	Adams	30	Athens	3	S2	Jones	10	Paris	4	S1	Smith	20	London	5	S3	Blake	30	Paris
	#	Name	Status	City																														
	1	S4	Clark	20	London																													
	2	S5	Adams	30	Athens																													
	3	S2	Jones	10	Paris																													
	4	S1	Smith	20	London																													
5	S3	Blake	30	Paris																														

Рисунок 2.1 – Исходная таблица с данными

На рис. 2.2 показана таблица TR, называемая таблицей значений полей, которая соответствует файлу на рис. 2.1. Стоит обратить внимание на то, что ее строки не соответствуют каким-либо очевидным образом тем записям, которые показаны на рис. 2.1.

Последовательность столбцов:		1	2	3	4
Последовательность строк:		#	Name	Status	City
	1	S1	Adams	10	Athens
	2	S2	Blake	20	London
	3	S3	Clark	20	London
	4	S4	Jones	30	Paris
	5	S5	Smith	30	Paris

Рисунок 2.2 – Таблица значения полей

Для того чтобы иметь возможность реконструировать показанный на рис. 2.1 файл из таблицы значений полей на рис. 2.2, требуется еще одна таблица — таблица реконструкции записей (рис. 2.3). Следует отметить, что в ячейках этой таблицы теперь находятся номера строк.

Последовательность столбцов:		1	2	3	4
Последовательность строк:		#	Name	Status	City
	1	5	4	4	5
	2	4	5	2	4
	3	2	2	3	1
	4	3	1	1	2
	5	1	3	5	3

Рисунок 2.3 – Таблица реконструкций записей

Таблица значений полей — это единственная таблица TR, которая содержит пользовательские данные как таковые. Все остальные таблицы содержат внутреннюю информацию, которая имеет смысл только для модели TR, но непосредственно не касается пользователя. Формируется таблица значений полей с помощью сортировки: каждый столбец таблицы содержит значения из соответствующего поля файла, отсортированные в порядке возрастания. Стоит отметить, что независимо от первоначального расположения записей в файле всегда формируется одна и та же таблица значений полей.

Формирование таблицы реконструкций записей происходит с помощью более сложного алгоритма чем сортировка, который описан ниже:

- Каждый столбец сортируется по возрастанию.
- Для каждого столбца выполняется обратная перестановка. Например, столбец 43512 преобразуется в 45213.

```
fun reversePerm(p : int[], rep : int[])  
    for i = 1 to n  
        rep[p[i]] = i;
```

- Для каждого столбца, поочередно для каждой строки необходимо выполнить следующее: перейти в ячейку [i, 1] таблицы обратных перестановок. Пусть эта ячейка содержит значение r; кроме того, допустим, что следующая ячейка справа [i, 2] содержит значение r1. Перейти в r-ю строку таблицы реконструкций записей и поместить значение r1 в ячейку [r, 1].

2.2. Описание используемых технологий

Инструмент преобразования представляет собой настольное приложение с запуском из командной строки, которое на вход будет получать SQL скрипты.

В качестве используемых технологий разработки предполагается следующий набор:

- Python
- SQL

ЗАКЛЮЧЕНИЕ

В результате проведенной работы был выполнен начальный этап разработки инструмента для позволяющего работать с моделью TransRelational. В частности, был осуществлен анализ предметной области, был сформулирован список основных задач, которые должны быть выполнены для достижения цели. Проведен анализ алгоритма преобразования в TR модель, выбраны инструменты для решения поставленной задачи.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. U.S. Patent and Trademark Office: Value-Instance-Connectivity Computer-Implemented Database. U.S. Patent No. 6,009,432. - December 28, 1999.
2. Дейт, К. Дж. Введение в системы баз данных, 8-е издание/ М.: Издательский дом "Вильямс", 2005. — 1328 с.
3. Date C. J. Go Faster! The TransRelational™ Approach to DBMS Implementation
4. Фомичева Т.Г. Базы данных. Проектирование приложений реляционной БД: конспект лекций. Ч.1 / Т.Г.Фомичева. – СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2008. 82 с.
5. Уровни представления данных – [Электронный ресурс]. URL: <http://economyreview.ru/teoriya-subd/urovni-modeley-dannyh> (Дата обращения: 10.12.2019)