



(ПИТ 2017) труды Международной научно-технической конференции. 2017. С. 753-755.

3. Гаранин, М.А. Распознавание проводов контактной сети по измерениям бесконтактных лазерных сканеров/ Гаранин М.А., Бахрах А.Г., Митрофанов С.А. //Актуальные проблемы проектирования и эксплуатации контактных подвесок и токоприемников электрического транспорта – Омск : ОмГУПС, 2011. С. 189-196.

4. Гаранин, М.А. Мобильный контрольно-вычислительный комплекс для диагностики контактной сети / А.Г. Бахрах, М.А. Гаранин, С.А. Митрофанов // Вестник транспорта Поволжья – Самара : СамГУПС, 2010. № 4. С. 49-52.

5. Гаранин, М.А. Устройство для контроля геометрии контактной сети/ М.А. Гаранин, А.Н. Митрофанов , С.А. Митрофанов// Вестник транспорта Поволжья – Самара: СамГУПС, 2016. № 1 (55). С. 24-29.

Е.И. Чигарина, К.С. Заикин

## СРАВНЕНИЕ ЭТАПОВ ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННЫХ И ОБЪЕКТНЫХ БАЗ ДАННЫХ

(Самарский национальный исследовательский университет  
имени академика С.П. Королева)

В системах баз данных проектирование включает такие основные этапы как концептуальное, логическое и физическое проектирование. Этап концептуального проектирования реализуется чаще всего с использованием модели «сущность-связь». На логическом уровне представления данных используются такие модели, как иерархические, сетевые, реляционные, объектно-реляционные и объектные [1].

В настоящее время наиболее часто используются реляционные базы данных. Однако все большее число разработчиков пользовательских приложений, использующих РСУБД, выражают неудовлетворение табличной структурой реляционной модели, приводящей к уменьшению скорости выполнения операций по манипулированию данными. Особенно это проявляется при проектировании систем, в которых хранятся сложные неструктурированные данные.

В объектных базах данных используется такая структура данных, как объект. Как и в ООП, при работе с объектными базами данных, используются такие понятия, как класс, объект, свойство, метод, наследование, полиморфизм, инкапсуляция. Объектные базы данных имеют преимущество в хранении больших объемов данных, за счет физической организации файлов. Однако для объектных баз данных не существует четких правил перехода от концептуального к логическому уровню. Сложностью разработки стандартизированного подхода является то, что реализация каждой объектной базы данных уникальна и не существует общих правила физического представления таких данных, в



отличие от реляционных баз данных использующих понятие «плоского» файла для физического хранения.

На этапе концептуального проектирования описывается семантическая модель данных. На этапе логического проектирования данные представляются в виде структур, в частности в виде табличной структуры (для реляционных баз данных) и объектной структуры (для объектных баз данных).

Для перехода от концептуальной модели к логической в реляционных базах данных используются правила Джексона (рисунок 1). Реляционная база данных представляет собой совокупность взаимосвязанных отношений. Отношение является множеством упорядоченных наборов данных, которые можно представить в виде двумерной таблицы.

В данной работе сформулированы правила перехода от концептуальной модели к логической для объектных баз данных (рисунок 2).

В объектной модели данных предметная область разбивается на классы, которые после компиляции представляют собой объекты. Объект – это структура данных, содержащая описание свойств, включая поля и методы, части предметной области.

Сравнивая полученные правила перехода для объектных баз данных и правила перехода Джексона для реляционных баз данных, видно, что для хранения объектных баз данных требуется меньшее количество объектов по сравнению с количеством отношений в реляционных базах данных, за счет особенностей организации объектов.

На этапе физического проектирования осуществляется выбор конкретного средства реализации базы данных. Вне зависимости от конкретной РСУБД все файлы базы данных реализуют модель «плоского» файла. Для объектных баз данных каждая ОСУБД имеет свой способ физического хранения данных.

В данной работе для сравнения этапов физического проектирования рассмотрены СУБД MS SQL Server и Cache. СУБД Cache использует понятие глобалов. Глобал – это постоянный, разреженный, динамический, многомерный массив, содержащий текстовые значения, причем это массив, который автоматически сохраняется на диск [2]. При вставке информации в глобал автоматически происходят сохранение данных на диск, индексация и сортировка. В общем случае глобал представляет собой упорядоченное дерево с возможностью хранения данных в каждом узле. Наибольшее преимущество глобалы имеют при выполнении операций добавления новых узлов, поэтому глобалы часто используются для реализации объектных баз данных, в которых необходимо хранить много однотипной информации. На физическом уровне в реляционных базах данных на каждое отношение создается файл и на каждую связь – индексный файл. В объектных базах данных на каждый объект создается только один глобал. Поэтому для хранения информации в объектных базах данных требуется меньшее количество файлов. Например, для связи «один ко многим» с обязательным классом принадлежности в реляционных базах данных требуется три файла данных и два индексных файла, а в объектных базах данных требуется только два глобала.



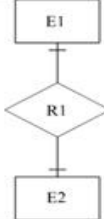

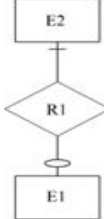

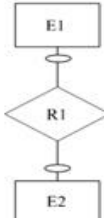

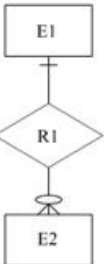

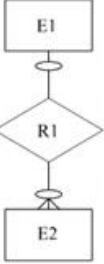

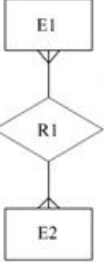

Пример связи	Реляционная модель	
	<p><b>ПРАВИЛО 1.</b> Если степень связи равна 1:1 и класс принадлежности обеих сущностей является обязательным, то требуется только одно отношение. Первичным ключом этого отношения может быть ключ любой из двух сущностей.</p>	
	<p><b>ПРАВИЛО 2.</b> Если степень связи равна 1:1 и класс принадлежности одной сущности является обязательным, а другой - необязательным, то необходимо построение двух отношений. Под каждую сущность необходимо выделение одного отношения, при этом ключ сущности должен служить первичным ключом для соответствующего отношения. Кроме того, ключ сущности, для которого класс принадлежности является необязательным, добавляется в качестве атрибута в отношение, выделенное для сущности с обязательным классом принадлежности.</p>	
	<p><b>ПРАВИЛО 3.</b> Если степень связи равна 1:1 и класс принадлежности ни одной сущности не является обязательным, то необходимо использовать три отношения: по одному для каждой сущности, ключи которых служат в качестве первичных в соответствующих отношениях, и одного для связи. Среди своих атрибутов отношение, выделяемое связи, будет иметь по одному ключу сущности от каждой сущности.</p>	
	<p><b>ПРАВИЛО 4.</b> Если степень связи равна 1:n и класс принадлежности п-связной сущности является обязательным, то достаточным является использование двух отношений, по одному на каждую сущность, при условии, что ключ сущности каждой сущности служит в качестве первичного ключа для соответствующего отношения. Дополнительно ключ 1-связной сущности должен быть добавлен как атрибут в отношение, отводимое п-связной сущности.</p>	
	<p><b>ПРАВИЛО 5.</b> Если степень связи равна 1:n и класс принадлежности п-связной сущности является необязательным, то необходимо формирование трех отношений: по одному для каждой сущности, причем ключ каждой сущности служит первичным ключом соответствующего отношения, и одного отношения для связи. Связь должна иметь среди своих атрибутов ключ сущности от каждой сущности.</p>	
	<p><b>ПРАВИЛО 6.</b> Если степень связи равна m:n, то для хранения данных необходимо три отношения: по одному для каждой сущности, причем ключ каждой сущности используется в качестве первичного ключа соответствующего отношения, и одного отношения для связи. Последнее отношение должно иметь в числе своих атрибутов ключ сущности каждой сущности.</p>	

Рисунок 1 – Правила Джексона для реляционной модели данных



Пример связи		Объектная модель
	<p><b>ПРАВИЛО 1.</b></p> <p>Если степень связи равна 1:1 и класс принадлежности обеих сущностей является обязательным, то создается два объекта, при этом один из объектов содержит поле типа «объект второго объекта», который содержит ссылку на объект второго объекта.</p>	
	<p><b>ПРАВИЛО 2.</b></p> <p>Если степень связи равна 1:1 и класс принадлежности одной сущности является обязательным, а другой – необязательным, то создается два объекта, при этом один из объектов содержит поле типа «объект второго объекта», который содержит необязательную ссылку на объект второго объекта.</p>	
	<p><b>ПРАВИЛО 3.</b></p> <p>Если степень связи равна 1:1 и класс принадлежности ни одной сущности не является обязательным, то в объектно-реляционной модели создается два объекта, каждый из которых содержит поле типа «объект другого объекта», который содержит необязательную ссылку на объект другого объекта.</p>	
	<p><b>ПРАВИЛО 4.</b></p> <p>Если степень связи равна 1:n и класс принадлежности n-связной сущности является обязательным, то в объектно-реляционной модели создается два объекта, при этом между объектами устанавливается специальное отношение связи Relationship, которое хранит ID первого объекта в n-связной сущности.</p>	
	<p><b>ПРАВИЛО 5.</b></p> <p>Если степень связи равна 1:n и класс принадлежности n-связной сущности является необязательным, то создается два объекта, при этом между объектами устанавливается специальное отношение связи Relationship и объект n-связной сущности имеет обязательное поле ID первого объекта.</p>	
	<p><b>ПРАВИЛО 6.</b></p> <p>Если степень связи равна 1:n, создается три объекта: два объекта сущности и один объект на связь, при этом между объектами сущности и объектом связи устанавливается отношение связи Relationship.</p>	

Рисунок 2 – Правила перехода от концептуальной модели к логической в объектных базах данных



В рамках данной работы, учитывая особенности логического проектирования и физического хранения баз данных, планируется рассмотреть особенности реализации ограничений целостности данных и манипулирования данными в реляционных и объектных базах данных, реализуя различные по сложности запросы с оценкой времени их выполнения.

### Литература

1. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. Третье издание. [Текст] / Т. Каннолли, К. Бегг. – Москва, Санкт-Петербург: изд-во Вильямс, 2003. – 1440 с.
2. InterSystems. Caché: Современная система управления базами данных [Электронный ресурс]. – <http://www.intersystems.com/ru/our-products/cache/cache-overview/>

Е.И. Чигарина, Ю.С. Чуркина

## МЕТОДИКА СРАВНИТЕЛЬНОГО АНАЛИЗА СРЕДСТВ ГЕНЕРАЦИИ ОТЧЁТОВ В РЕЛЯЦИОННЫХ БАЗАХ ДАННЫХ

(Самарский национальный исследовательский университет  
имени академика С.П. Королева)

Генераторы отчётов представляют собой специализированные средства, позволяющие представить информацию, хранящуюся в информационной системе, в виде документа или отчёта, который можно распечатать или сохранить в различных электронных форматах [1].

В настоящее время на рынке представлено большое количество систем генерации отчётов, и пользователю приходится тратить значительное время для того, чтобы понять какое из средств дает наибольшую функциональность и гибкость, соответствующую сложности и требованиям отчёта. К тому же непонятно как именно выбирать и что сравнивать. Можно, например, сравнить по размеру, который добавится к приложению при установке средства генерации. Но есть такие генераторы отчётов, которые к приложению ничего не добавляют, а представлены отдельным dll файлом, размеры которого гораздо больше самого приложения. Или же к исполняемому файлу добавляется немного, зато файлы сформированных отчётов в несколько раз больше самого приложения. Может проводить сравнение по удобству использования? Но в чем его измерить?

В связи с этим главной задачей данной работы является проведение сравнительного анализа использования средств генерации отчётов в реляционных базах данных. В работе предлагается следующая методика сравнения:

1. Выбор универсальных средств создания отчётов, обладающих максимальными функциональными возможностями для формирования и генерации отчётов. Среди функциональных возможностей были выделены следующие:  
– допустимые виды отчётов;