

**МЕТОД АГРЕГИРОВАНИЯ КЛАССА ПРИ ЛОГИЧЕСКОМ
ПРОЕКТИРОВАНИИ ПРЕДМЕТНОЙ ОБЛАСТИ В
ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ СУБД**

05.25.05

Шведенко В.Н., д.т.н., профессор, научный руководитель проекта «Cobra++»,
ЗАО «РЕГУЛ», г. Санкт-Петербург

Волков А.А., соискатель учёной степени кандидата технических наук,
инженер-программист, ЗАО «РЕГУЛ», г. Санкт-Петербург

В статье рассматривается метод «Коллинеарных связей» для поддержания связей между экземплярами множества классов в объектно-ориентированной СУБД, который позволяет решить проблему извлечения объектов из нескольких классов по общему условию. Предложена логическая схема организации метаданных для формирования коллинеарных связей в многомерной модели данных.

Ключевые слова: многомерная модель данных, объектно-ориентированная СУБД, коллинеарная связь, агрегирование, связи между объектами, класс.

UDC 004.652

**THE METHOD OF AGGREGATION OF THE CLASS
WHEN THE LOGICAL DESIGN OF THE DOMAIN
IN AN OBJECT-ORIENTED DBMS**

05.25.05

Shvedenko V.N., Doctor of Technical Sciences, Professor, supervisor of the
project «Cobra ++», company «Regul», St. Peterburg

Volkov A.A., applicant of scientific degree of candidate of technical sciences,
software engineer, company «Regul», St. Peterburg

The article proposes a method of «Collinear connections» to maintain the links between instances of the set of classes in object-oriented database that allows you to solve the problem of retrieving objects from multiple classes to the general condition. We propose a logical scheme of metadata for the formation of collinear relationships in the multidimensional data model.

Keywords: multidimensional data model, object-oriented DBMS, collinear relationship, aggregation, communication between objects, class.

На этапе проектирования информационной системы, как правило, появляются сущности, описывающие разные процессы в пределах одной пред-

метной области, но имеющие одинаковые фрагменты в своих структурах. На их основе формируются коллекции данных, которые создают или извлекают из хранилища данных по определенным правилам и объединяют по различным условиям. Для традиционного реляционного подхода решать такую задачу можно путём реализации сложной структуры, приводя её к нормальной форме, либо программированием хранимых процедур и обработки данных, полученных несколькими запросами. Оба подхода связаны с большими затратами времени на разработку прикладного программного обеспечения. Для снижения трудоемкости разработки информационных систем предлагаются объектные технологии.

Объектно-ориентированные технологии, по сравнению с реляционными системами, позволяют наглядно представлять сложные структуры объектов, что даёт возможность преодолеть ограничения реляционной модели, связанной с атомарностью атрибутов отношения.

В классической постановке объектно-ориентированного подхода предметная область определяется в виде совокупности объектов, характеризующихся набором атрибутов и методов. Множество объектов с одним и тем же набором атрибутов и методов образует класс. Иерархия классов имеет неразделяемую структуру, а объекты классов неразделяемую память. При абстрагировании предметной области, объекты одного класса могут являться атрибутами объектов других различных между собой классов и находиться на разных уровнях иерархии. Поэтому выборку данных из объектов различных классов по заданному атрибуту, сделать затруднительно.

Проведён анализ методов организации связей между объектами, а также методов извлечения данных в объектно-ориентированных СУБД, на основании которого сделан вывод, что извлечение данных из объектов различных классов возможно либо явным образом, т.е. сопоставлением двух объектов разных классов на основе совпадения значений соответствующих пар атрибутов объектов, либо не явным образом, т.е. созданием специальных классов

для поддержания неявных ассоциативных соединений, порождаемых иерархической вложенностью объектов [1, 2].

Первый подход является аналогом реляционных соединений и имеет серьёзные ограничения, связанные с не атомарностью атрибутов объекта и несовпадением доменов атрибутов. Второй подход приводит к избыточности метаданных и созданию ряда сложных классов и методов агрегирования объектов с поддержкой целостности данных.

Для решения рассмотренных выше проблем необходима разработка новых методов и программных средств, реализующих внешние связи между объектами в многомерной модели данных.

Для отражения сути предлагаемого метода, реализующего связь между объектами различных классов, вводится понятие коллинеарной связи. Коллинеарная связь – вид отношения агрегации, при котором экземпляры, класса, являющегося частью некоторого множества других классов, поддерживают связи со всеми объектами каждого класса-целого.

Коллинеарная форма – класс или несколько классов находятся в коллинеарной форме, если их класс-часть, принадлежащие общему классу, имеют коллинеарные связи (см. рис. 1).

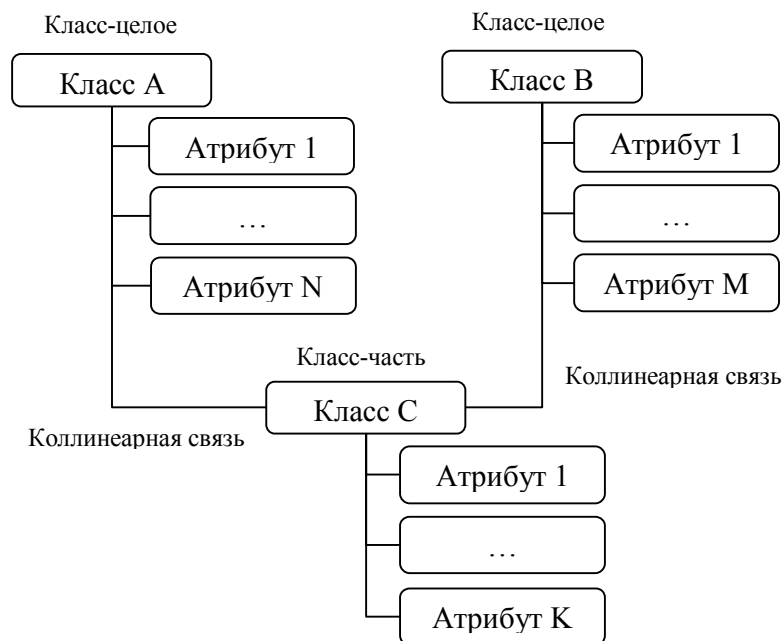


Рис. 1. Классы в коллинеарной форме

Организация метаданных с применением коллинеарных связей предусматривает образование внешних связей между состояниями объектов различных классов, не имеющих явных связей между собой.

Такая организация хранения данных даёт возможность делать информационные срезы по данным объектов различных классов, выполнять анализ данных всей системы по заданному критерию.

Коллинеарная связь представляет собой сетевую структуру, где каждое состояние объекта с коллинеарным поведением может иметь множество связей с множеством состояний объектов различных классов (см. рис. 2).

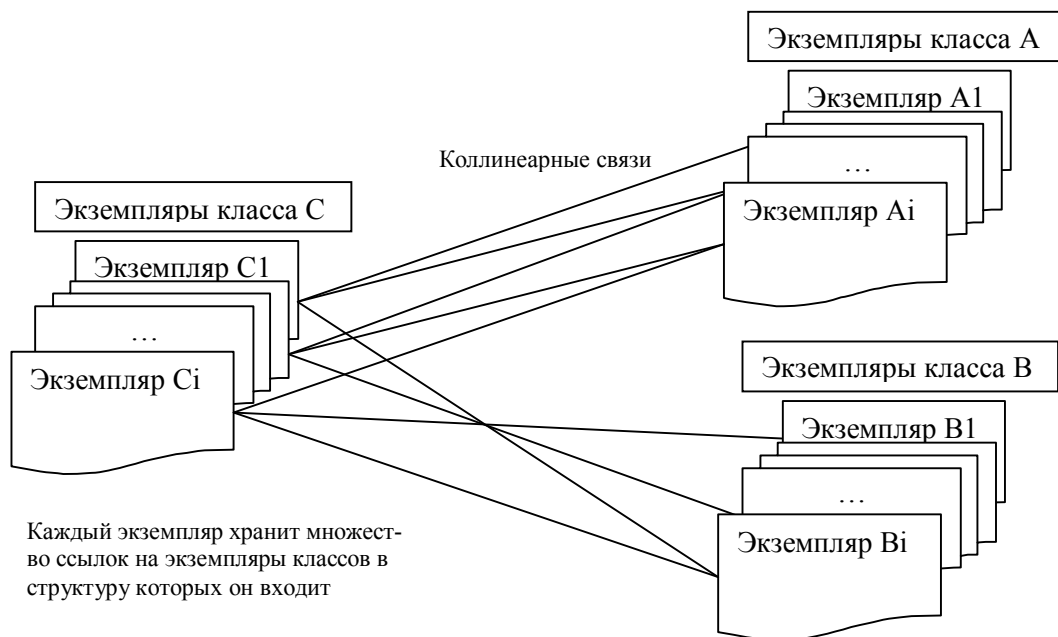


Рис. 2. Объекты с коллинеарными связями

Для реализации предложенного метода был проведён обзор основных способов построения объектно-ориентированной модели с использованием реляционных составляющих. Рассмотрены три разновидности многомерной модели данных MOLAP, ROLAP, HOLAP. Как результат обзора, было принято решение использовать в качестве механизма организации метаданных модель ROLAP, где факты и таблицы с измерениями хранятся в реляционных таблицах, и для хранения агрегатов создаются дополнительные реляционные таблицы [3]. Для построения ROLAP модели метаданных было выбрано табличное пространство СУБД Oracle Database 10g Express Edition. Методы ра-

боты с многомерными данными реализованы в виде набора пакетов хранимых процедур средствами языка PL/SQL. На рисунке 3 представлена общая схема организации прототипа СУБД.

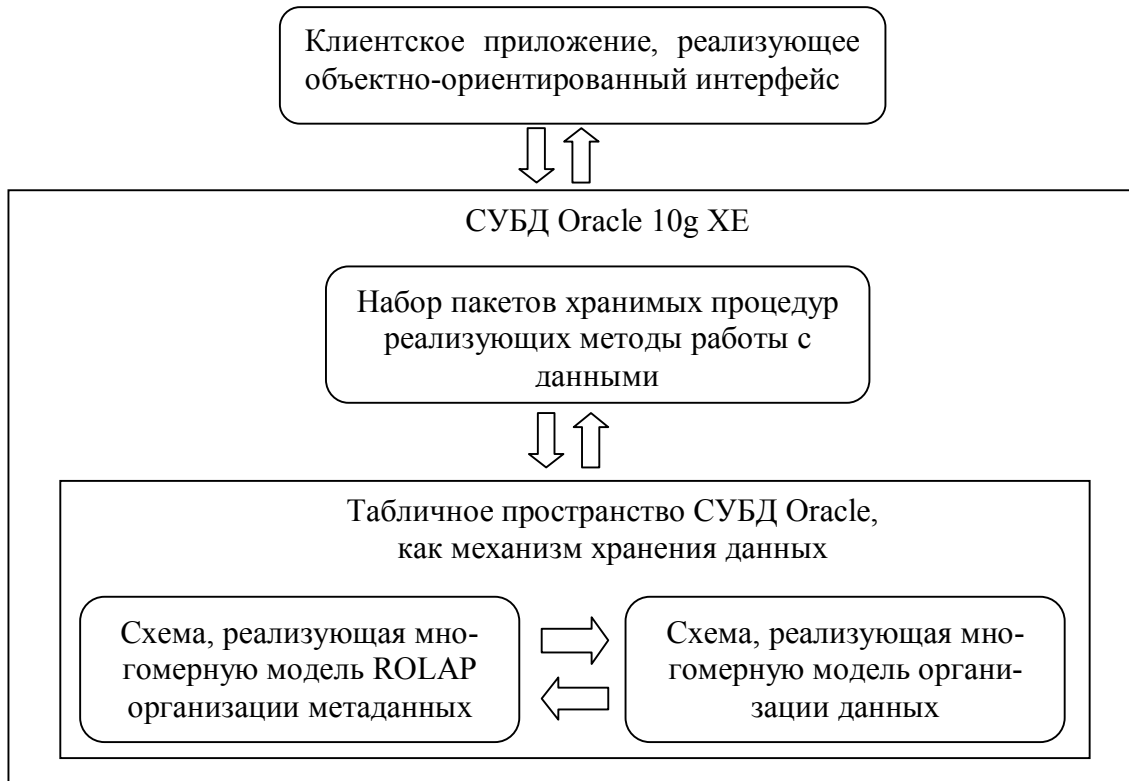


Рис. 3. Общая схема организации СУБД

Для организации многомерной модели данных, с поддержкой коллинеарных связей между объектами, средствами реляционных отношений, разработана схема метаданных (см. рис. 4).

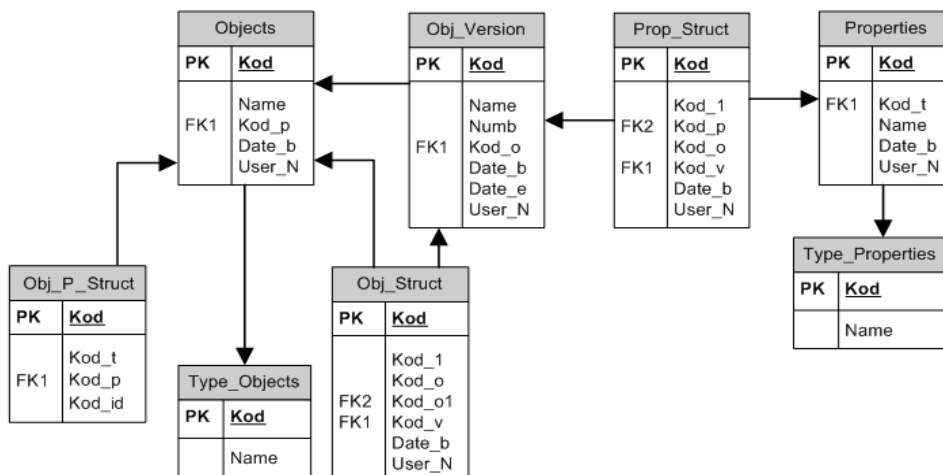


Рис. 4. Схема организации многомерной модели данных средствами реляционных отношений

На основе предложенной схемы был реализован программный модуль системы «Собра ++», реализующий объектно-ориентированный интерфейс взаимодействия с данными, организованными в виде многомерной модели. Окно проектировщика класса представлено на рисунке 5, где выполняется проектирование класса «Освещение», который имеет коллинеарные связи с классами «Работа» и «Помещение».

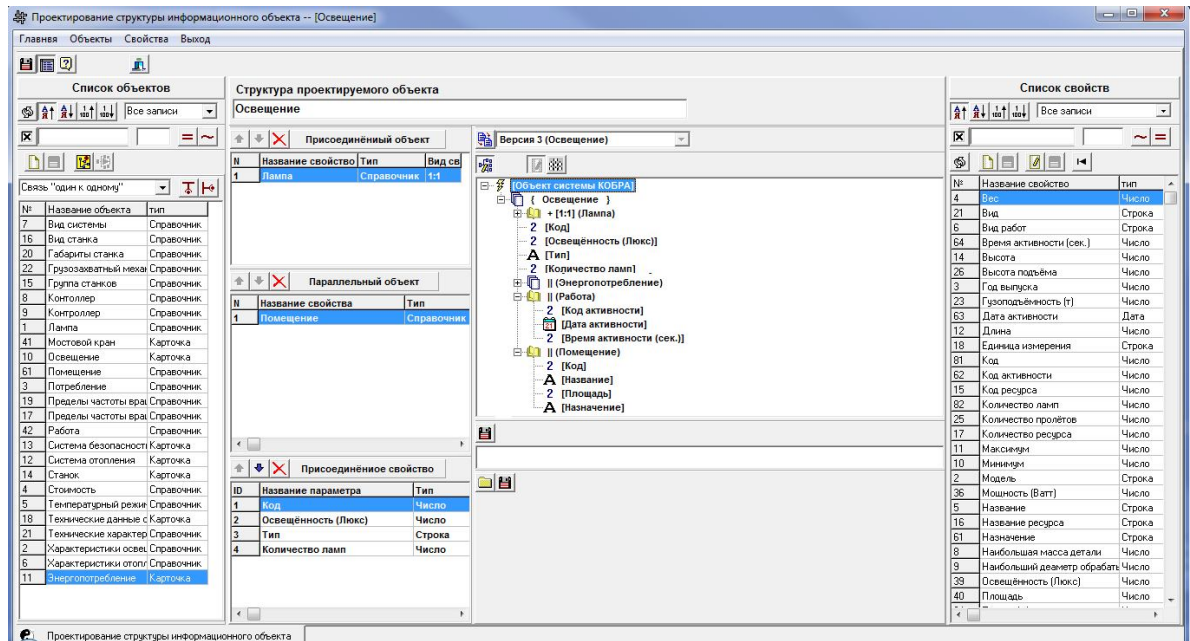


Рис. 5. Окно проектировщика классов системы «Собра++»

Таким образом, класс «Освещение» находится в коллинеарной форме. Также в систему «Собра++» включен модуль выполнения запросов к базе данных, с целью выборки информации по определенным критериям. Он является своеобразным интегрирующим звеном для всех остальных приложений. Позволяет анализировать информацию из хранилища данных, производить поиск нужной информации по определенным критериям, а также является инструментом быстрого доступа к объектам для специалистов предприятия. На рисунке 6 представлен результат выборки данных по объектам класса «Энергопотребление», где результатом является список объектов всех классов, с которыми этот класс имеет коллинеарные связи.

Кобра - Просмотр и фильтрация хранилища данных

Запрос Обработка запроса Обработка результатов запроса Инструменты Помощь

Результат выполнения запроса "Запрос 1" к объекту "Энергопотребление"

№ п/п	Код ресурса	Название ресурса	Потребление	Стоимость	Объект	Код экземпляра
1	302.00	Природный газ	20.00м3/ч	3.05 руб.	Система отопления	76
2	302.00	Природный газ	20.00м3/ч	3.05 руб.	Система отопления	87
3	301.00	Электричество	37.00кВт/ч	3.40 руб.	Система отопления	98
4	302.00	Природный газ	50.00м3/ч	3.05 руб.	Система отопления	102
5	302.00	Электричество	37.00кВт/ч	3.40 руб.	Освещение	67
6	302.00	Электричество	37.00кВт/ч	3.40 руб.	Освещение	45
7	302.00	Электричество	41.00кВт/ч	3.40 руб.	Освещение	44
8	302.00	Электричество	37.00кВт/ч	3.40 руб.	Освещение	41
9	302.00	Электричество	24.00кВт/ч	3.40 руб.	Освещение	39

Время выполнения запроса 0:0:0.49 мсек., количество экземпляров 1 [Всего экземпляров 2]

База данных : XE Пользователь : admin Проект : Начальный проект

Рис. 6. Окно модуля информационных запросов

Выводы:

Предложенный метод организации коллинеарных связей, позволяет решить проблему, связанную с инкапсуляцией данных и ограниченностью взаимодействия объектов разных классов. При агрегировании сложных классов, объекты класса-части инкапсулируют ссылки (коллинеарные связи) на объекты класса-целого, организуют сетевую структуру связей между объектами множества классов системы. При этом, обращение к любому из классов, экземпляры которых инкапсулируют коллинеарные связи, позволяет получить все связанные с ним объекты из некоторого множества классов. Реализация такого метода, средствами реляционной СУБД, демонстрирует возможность не только чисто реляционного или объектного подхода к организации данных, но и существования компромиссного решения, а именно построения многомерной модели средствами реляционных таблиц, отказавшись от требований нормализации, атомарности атрибутов и попытки полного исключения избыточности данных.

Литература

1. Кузнецов С.Д. Объектно-ориентированные базы данных основные концепции, организация и управление: краткий обзор // http://citforum.ncstu.ru/database/articles/art_24.shtml#lit.
2. Sikha Bagui. Achievements and Weaknesses of Object-Oriented Databases // Journal of Object Technology (JOT), vol. 2, no. 4, July-August 2003, p 29-41.
3. Советов Б.Я., Цехановский В.В., Чертовской В.Д. Базы данных. Теория и практика. М.: Высшая школа, 2005.