

Выводы:

1. При увеличении количества прогонов величина доверительного интервала постепенно уменьшается и стремится к нулю.
2. При оценке чувствительности имитационной модели выявили, что отклик чувствителен к изменениям переменной на интервале ее варьирования, т.к. коэффициент корреляции R^2 между значениями коэффициента загрузки робота и временного интервала перевозки им деталей составил 0,9337, что очень близко к единице.
3. В результате постановки имитационного эксперимента по сравнению альтернатив можно сформулировать выводы о наилучшем использовании ресурсов автоматизированного участка обработки деталей с учетом интересов производства. Анализ полученной математической модели позволит определить параметры оборудования для обработки деталей с целью наименьшего износа оборудования.

Литература

1. Лоу, А.М. Имитационное моделирование / А.М. Лоу, В.Д. Кельтон. СПб.: Издательский дом ПИТЕР, 2004. - 848 с.

УДК: 004.652.5

ОБЪЕКТНО-РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ**Лысюк А.Н***УО «Брестский государственный технический университет», г. Брест*

Разработка любой информационной системы основана на построении модели данных. Базовыми понятиями в модели данных являются объекты и отношения между ними. Спроектированная модель ложится в основу базы данных (включается в концептуальный и внешний уровни ее архитектуры). Качество проектирования модели данных непосредственно влияет на качество работы с конкретной базой данных.

В настоящее время широко распространена реляционная модель данных. Основной структурой данных в реляционной модели является отношение, как подмножество декартова произведения, представимое в виде таблицы. Каждая строка этой таблицы представляет запись (кортеж), а каждый столбец соответствует атрибуту отношения из заданного домена. В таблице не может быть двух одинаковых записей. Таким образом, реляционная модель данных представляется множеством таблиц-отношений.

Иерархические связи между отношениями поддерживаются неявным образом. В каждой связи одно отношение - основное, а другое – подчиненное. Таким образом, один кортеж основного отношения может быть связан с несколькими кортежами подчиненного отношения. Для поддержки этих связей оба отношения должны содержать набор атрибутов, по которым они связаны. В основном отношении это первичный ключ (primary key), в подчиненном отношении это внешний ключ (foreign key). Над множеством отношений определен ряд операций. В алгебре Кодда определены 8 операций, которые делятся на две группы: теоретико-множественные и специализированные операции.

Основными достоинствами реляционной модели данных является: полная независимость данных, строгий математический аппарат. Недостатками этой модели являются трудность понимания структуры данных, низкая скорость доступа, большой объем внешней памяти. Кроме того, не всегда предметную область можно представить в виде совокупности таблиц.

Рассмотрим следующий пример. При реализации, средствами языка С++, информационной системы, которая должна выполнять расчет численности профессорско-преподавательского состава кафедр университета, возникла необходимость в представлении объекта «Академическая группа». Для данной сущности был определен набор существенных в рамках рассматриваемой модели атрибутов. Кроме того, при проведении лабораторных занятий группа делится на подгруппы, а механизм деления зависит от нескольких факторов. Целесообразно реализовать деление на подгруппы в виде отдельной функции-метода.

Способ реализации рассматриваемой функции заключается в следующем: создается класс Group, в этом классе инкапсулируются атрибуты, а также функции доступа к данным атрибутам (интерфейс). Затем в этот класс добавляется метод деления группы на подгруппы (DivideBySubGroup). Схема реализации показана на рисунке 1.

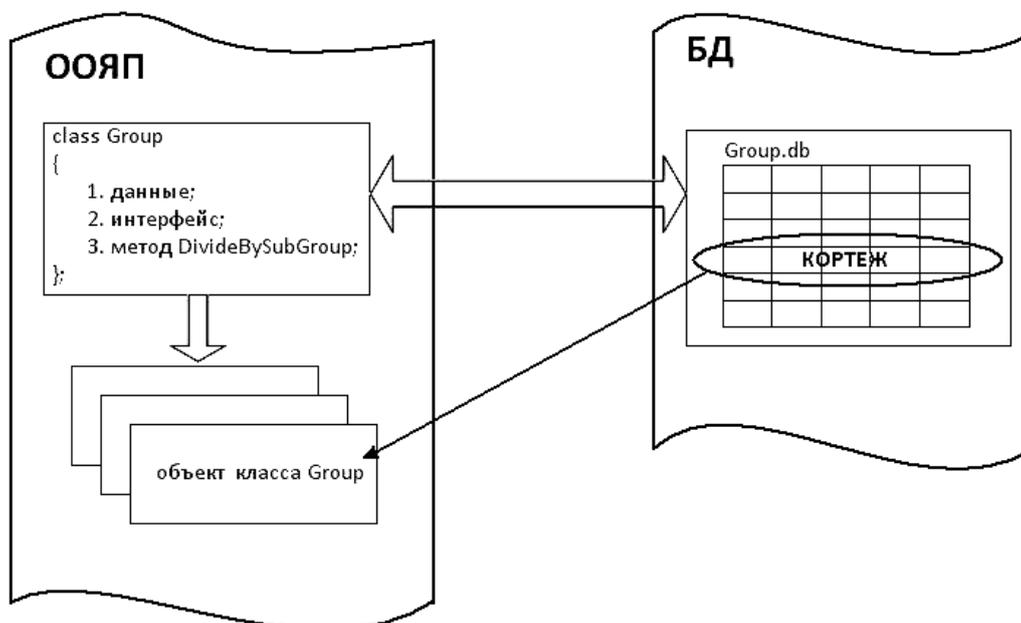


Рисунок 1 – Взаимодействие объектно-ориентированного языка программирования с реляционной базой данных

При таком подходе обнаруживается несоответствие, связанное с тем, что модели данных, используемые в программировании, отличаются от моделей данных СУБД, что влечет за собой необходимость поддерживать соотношения и взаимосоответствия между кортежами базы данных и объектами в программировании. Кроме того, при таком подходе нарушается семантическая целостность данных. В данной ситуации решением является использование объектно-ориентированной модели данных.

Объектно-ориентированные базы данных явились результатом совмещения принципов объектно-ориентированного программирования и принципов управления базами данных. В одной системе объединены понятия инкапсуляции, полиморфизма, наследования из объектно-ориентированного программирования и атомарности, целостности, изоляции из баз данных. В результате мы имеем возможность управлять большими объемами информации при помощи объектно-ориентированного подхода.

Объектно-ориентированная база данных (ООБД) способна хранить объекты в том же виде, в котором они будут доступны для языка программирования. Это обеспечивается за счет того, что объекты в ООБД принадлежат классу, имеющему в своем составе на-

бор атрибутов, выражаемых простыми типами данных или другими классами. К классам применяются правила наследования, несущие в себе преимущества ООП: полиморфизм, переопределение наследованных методов и возможность динамической привязки.

Преимущества:

1. Объекты могут хранить произвольное количество простых типов и других объектов.
2. Объект является наилучшей моделью отображения реального мира, нежели реляционные картежи.
3. Для доступа к данным из систем управления ООБД не обязателен отдельный язык запросов, поскольку доступ происходит непосредственно к объектам.

Недостатки:

1. Изменение схемы данных в результате создания, изменения или удаления таблиц обычно зависит от приложения.
2. В объектно-ориентированном программировании отсутствуют общие средства манипулирования данными, такие как реляционная алгебра или реляционное счисление. Работа с данными ведется с помощью одного из объектно-ориентированных языков программирования общего назначения (Python, Java, C#, Visual Basic .NET, C++, Objective-C).

Очевидно, что оба указанных недостатка связаны с отсутствием развитых средств манипулирования данными. Эта задача решается двумя способами – расширение объектно-ориентированных языков в сторону управления данными (стандарт ODMG) либо добавление объектных свойств в реляционные СУБД (SQL-3, а также так называемых объектно-реляционных СУБД).

Основой предлагаемого подхода является объединение объектного кода с соответствующим отношением реляционной модели, при этом файл базы должен состоять из четырех блоков (рисунок 2).

Заголовок файла
Блок метаданных
Отношение - таблица
Объектный код

Рисунок 2 – Структура файла базы данных

В блоке метаданных необходимо описать реализуемый класс, указать свойства и прототипы методов. Это упростит дальнейшую интеграцию базы данных при разработке приложения. Кроме того, необходимо, чтобы объектный код был платформонезависимым. Лучшим вариантом является использование промежуточного языка платформы .NET.

Развитие данного подхода в дальнейшем позволит устранить недостатки, связанные с использованием только реляционной либо только объектно-ориентированной базы данных.

Литература

1. Кузнецов, С. Д. Основы баз данных / С.Д. Кузнецов – 1-е изд. – М.: Интернет-университет информационных технологий - ИНТУИТ.ру, 2005. – С. 38-58.
2. Когаловский, М.Р. Энциклопедия технологий баз данных / М.Р. Когаловский – М.: Финансы и статистика, 2002. – С. 800.
3. Тяпкин, С.В. Преимущества и недостатки объектно-ориентированных баз данных / С.В. Тяпкин // Журнал научных публикаций аспирантов и докторантов. – 2007.