



ВВЕДЕНИЕ В ПОСТ-РЕЛЯЦИОННЫЕ СИСТЕМЫ БАЗ ДАННЫХ

Выбор точки зрения – это первичный акт культуры.

Х. Ортега-и-Гассет

Пост-реляционные системы баз данных

Содержание

2

- Мотивация пост-реляционных систем баз данных
- Классификация пост-реляционных систем баз данных
- Основные положения Манифеста систем объектно-ориентированных баз данных

Пост-реляционные системы баз данных © М.Л. Цымблер

Мотивация пост-реляционных систем баз данных

3

- Проблема "несоответствия импеданса" (*impedance mismatch*) – кардинальные различия между прикладной программой и данными, хранимыми в реляционной базе данных.
 - Отсутствие неатомарных типов данных.
 - ИНФ запрещает «вложенные» таблицы.
 - Отсутствие пользовательских типов данных.
 - Нельзя определить новый тип данных и его операции на базе имеющихся типов и операций.
 - Отсутствие типов данных со сложной структурой
 - Структуры (записи), коллекции (множество, массив, список и др.), данные САПР и ГИС, документ-данные.
 - Отсутствие поддержки концепций ООП (объект, инкапсуляция, наследование, полиморфизм).

Пост-реляционные системы баз данных © М.Л. Цымблер

Пост-реляционные подходы

4

- Модель ОО данных.
 - 1-й манифест*, стандарт ODMG.
- Модель ОР данных.
 - 2-й манифест.
- (Истинно) реляционная модель данных.
 - 3-й манифест.
- Модель слабоструктурированных данных.

* Манифест – воззвание, декларация, содержащее программу и принципы деятельности.

Пост-реляционные системы баз данных © М.Л. Цымблер

Манифесты систем баз данных

5

- **1989, 1-й манифест**
 - *Atkinson M.P., et al.* The Object-Oriented Database System Manifesto // Deductive and Object-Oriented Databases, Proc. of the 1st Int. Conf. on Deductive and Object-Oriented Databases (DOOD'89), Kyoto, Japan, 4-6 December, 1989. North-Holland: Elsevier. -1990. -P. 223-240.
 - *Аткинсон М. и др.* Манифест систем объектно-ориентированных баз данных // СУБД. -1995. -№ 4. -С. 142-155.
- **1990, 2-й манифест**
 - *Stonebraker M., et al.* Third-Generation Database System Manifesto // ACM SIGMOD Record. -1990. -Vol. 19, No. 3. -P. 31-44.
 - *Стоунбрейкер М. и др.* Системы баз данных третьего поколения: Манифест // СУБД. -1996. -№ 2.
- **1995, 3-й манифест**
 - *Darwen H., Date C.J.* The Third Manifesto // ACM SIGMOD Record. -1995. -Vol. 24, № 1. -P. 39-49.
 - *Дарвин Х., Дейт К.* Третий манифест // СУБД. -1996. -№ 1. -С. 110-123.

Пост-реляционные системы баз данных © М.Л. Цымблер

Манифесты систем баз данных

6

- **1989, 1-й манифест**
 - Попытка дать научное определение системы ОО баз данных.
 - Новые системы необходимо строить на базе новой ООМД (имеющуюся РМД нужно отбросить).
- **1990, 2-й манифест**
 - Инженерно-публицистический документ – ответ сообщества SQL-разработчиков авторам 1-го манифеста.
 - Новые системы необходимо строить на базе имеющейся SQL-модели.
- **1995, 3-й манифест**
 - Строго научное определение новых систем баз данных.
 - Новые системы необходимо строить на базе истинно реляционной модели (имеющиеся реляционную и SQL модели нужно отбросить).

Пост-реляционные системы баз данных © М.Л. Цымблер

Первый манифест

7

- Характерные черты тогдашнего состояния дел в области ООСБД:
 - отсутствие общепринятой модели данных
 - отсутствие формального базиса
 - активная экспериментаторская деятельность.
- Описание основных свойств и характеристик, которыми должна обладать система, претендующая на то, чтобы быть квалифицированной как ООСБД.
 - Обязательные – которыми система должна обладать безусловно.
 - Необязательные – которые могут быть добавлены к системе для ее улучшения, но обязательными не являются.
 - Открытые – которые проектировщик может реализовать по собственному усмотрению.

Пост-реляционные системы баз данных © М.Л. Цымблер

Обязательные свойства ООСБД

8

- Сложные объекты
- Идентифицируемость (*identity*) объектов
- Инкапсуляция, типы или классы
- Наследование
- Перегрузка методов совместно с поздним связыванием
- Расширяемость
- Вычислительная полнота
- Стабильность (персистентность)
- Управление вторичной памятью
- Параллелизм
- Восстанавливаемость
- Средства обеспечения незапланированных (*ad hoc*) запросов

Пост-реляционные системы баз данных © М.Л. Цымблер

Сложные объекты

9

- Сложные объекты строятся из более простых при помощи конструкторов
 - Простейшие объекты: целые числа, символы, символьные строки произвольной длины, булевские переменные и числа с плавающей точкой.
 - Конструкторы сложных объектов: конструкторы множеств, кортежей, списков, кортежей (минимально), а также мультимножеств, массивов и др.
 - Конструкторы объектов должны быть ортогональными, т.е. любой конструктор должен быть применим к любому объекту.
- Для поддержки сложных объектов необходимы соответствующие операторы (операции со сложным объектом должны транзитивно распространяться на все его компоненты).

Пост-реляционные системы баз данных © М.Л. Цымблер

Идентифицируемость объектов

10

- Объект существует независимо от его значения. Два объекта могут быть идентичны (они представляют собой один и тот же объект) или они могут быть равны (они имеют одно и то же значение).
- В императивных языках программирования каждый программный объект идентифицируем и может быть изменен. Это обеспечивается за счет наличия имени переменной или соответствующего физического адреса памяти. Данная концепция нова для чисто реляционных систем, где идентифицируемость кортежей отношения основывается на значениях.

Пост-реляционные системы баз данных © М.Л. Цымблер

Инкапсуляция объектов

11

- Объект делится на интерфейсную и реализационную части.
 - Интерфейсная часть специфицирует набор допустимых над объектом операций.
 - Реализационная часть состоит из части данных (представление или состояние объекта) и процедурной части (описание реализации каждой операции на некотором языке программирования).
- *Инкапсуляция* определяет видимыми операции, а данные и реализация операций – скрытыми в объектах.

Пост-реляционные системы баз данных © М.Л. Цымблер

Объектные типы

12

- *Объектный тип* \cong АД
- Тип делится на две части – интерфейс и реализация. Пользователям типа видна только интерфейсная часть, а реализация объекта видна только проектировщику типа.
- Интерфейс состоит из списка операций вместе с их сигнатурами (набором типов входных параметров и типом результата).
- Реализация типа состоит из части данных и части операций. Часть данных описывает внутреннюю структуру данных объекта. Часть операций состоит из процедур, которые реализуют операции интерфейсной части.

Пост-реляционные системы баз данных © М.Л. Цымблер

Классы

13

- *Класс* \cong Объектный тип. Спецификация класса совпадает со спецификацией типа, но является более динамическим понятием.
 - *Фабрика объектов* используется для создания новых объектов посредством выполнения операции *new* данного класса операции или клонирования некоторого объекта-прототипа, являющегося представителем данного класса.
 - *Склад объектов* – это класс и набор объектов, которые являются экземплярами этого класса. Пользователь может манипулировать складом, применяя операции ко всем экземплярам класса.

Пост-реляционные системы баз данных © М.Л. Цымблер

Иерархии классов или типов

14

- *Наследование через подстановку*
 - Тип $T_{\text{потомок}}$ наследует от типа $T_{\text{предок}}$, если над объектами типа $T_{\text{потомок}}$ можно выполнить больше операций, чем над объектами типа $T_{\text{предок}}$ (вместо объекта типа $T_{\text{предок}}$ всегда можно подставить объект типа $T_{\text{потомок}}$).
- *Наследование путем включения*
 - Тип $T_{\text{потомок}}$ наследует от типа $T_{\text{предок}}$, если каждый объект типа $T_{\text{потомок}}$ является также объектом типа $T_{\text{предок}}$.
- *Наследование через ограничение*
 - Тип $T_{\text{потомок}}$ наследует от типа $T_{\text{предок}}$, если к нему относятся все объекты типа $T_{\text{предок}}$ которые удовлетворяют заданному ограничению.
- *Наследование через специализацию*
 - Тип $T_{\text{потомок}}$ наследует от типа $T_{\text{предок}}$ если объекты типа $T_{\text{потомок}}$ являются объектами типа $T_{\text{предок}}$ но при этом содержат более конкретные данные.

Пост-реляционные системы баз данных © М.Л. Цымблер

Перекрытие, перегрузка и позднее связывание

15

- *Перегрузка* – разделение имени операции в иерархии типов (классов).
- *Перекрытие* – переопределение реализации операции для каждого типа в соответствии с его особенностями.
- *Позднее связывание* – трансляция имен операций в адреса программ во время выполнения (а не компиляции) программы.

Пост-реляционные системы баз данных © М.Л. Цымблер

Вычислительная полнота

16

- Любую вычислимую функцию можно выразить с помощью языка манипулирования данными системы баз данных.

Пост-реляционные системы баз данных © М.Л. Цымблер

Расширяемость

17

- Должны иметься средства для определения новых типов и не должно быть различий в использовании системных и определенных пользователем типов.

Пост-реляционные системы баз данных © М.Л. Цымблер

Стабильность (персистентность)

18

- Стабильность – возможность программиста обеспечить сохранность данных после завершения выполнения процесса с целью последующего использования в другом процессе.
- Свойство стабильности должно быть ортогональным к другим свойствам, т.е. для каждого объекта вне зависимости от его типа должна иметься возможность сделать его стабильным (без какой-либо явной переделки объекта).
- Стабильность должна быть неявной: пользователь не должен явно перемещать или копировать данные, чтобы сделать их стабильными.

Пост-реляционные системы баз данных © М.Л. Цымблер

Управление вторичной памятью

19

- Как и в классических СУБД – управление индексами, кластеризация данных, буферизация данных, выбор пути доступа и оптимизация запросов.
- Данные механизмы должны быть невидимыми для пользователя. Прикладной программист не должен писать программы для сопровождения индексов, распределять память на диске или перемещать данные между диском и основной памятью.

Пост-реляционные системы баз данных © М.Л. Цымблер

Параллелизм и восстановление

20

- Как и в классических СУБД – должно обеспечиваться гармоническое сосуществование пользователей, одновременно работающих с базой данных.
 - Должно поддерживаться стандартное понятие атомарности последовательности операций и управляемого совместного доступа. Должна обеспечиваться сериализуемость операций.
- Как и в классических СУБД – в случае аппаратных (процессор или диски) или программных сбоев система должна возвращаться к некоторому согласованному состоянию данных.

Пост-реляционные системы баз данных © М.Л. Цымблер

Средства обеспечения *ad hoc* запросов

21

- Незапланированные (*ad hoc*) запросы – запросы, не входящие в интерфейс типа (класса).
- Должен поддерживаться декларативный высокоуровневый язык запросов с возможностью оптимизации запросов.

Пост-реляционные системы баз данных © М.Л. Цымблер

Другие свойства ООСБД

22

- Не достигнуто согласие по поводу обязательности следующих свойств:
 - Определение представлений и порождаемых данных.
 - Утилиты администрирования баз данных.
 - Ограничения целостности.
 - Утилиты, поддерживающие эволюцию схемы.

Пост-реляционные системы баз данных © М.Л. Цымблер

Необязательные свойства ООСБД

23

- Множественное наследование
- Проверка типов данных во время компиляции
- Распределенность базы данных
- Версионирование базы данных

Пост-реляционные системы баз данных © М.Л. Цымблер

Открытые свойства ООСБД

24

- Парадигма программирования
- Система типов данных

Пост-реляционные системы баз данных © М.Л. Цымблер

Заключение

25

- Проблема "несоответствия импеданса" – основная причина появления пост-реляционных систем.
- Пост-реляционные подходы
 1. Модель ОО данных
 2. Модель ОР данных
 3. (Истинно) реляционная модель данных
 4. Модель слабоструктурированных данных.
- Базовые положения подходов 1.-3. определены в соответствующих манифестах.
- Рассмотрены положения 1-го манифеста.

Пост-реляционные системы баз данных © М.Л. Цымблер
