

УДК 621.396.218

ББК 32.884.1

Т 19

Тарасов С. В. СУБД для программиста. Базы данных изнутри. — М.: СОЛОН-Пресс, 2015. — 320 с.: ил.

ISBN 978-2-7466-7383-0

EAN 9782746673830

Книга охватывает различные этапы разработки и сопутствующие им ситуации из практики программистов приложений, работающих с системами управления базами данных. Даются рекомендации по выбору решений как в проектировании (архитектуре), так и в программировании автоматизированных информационных систем уровня предприятия. Приводятся примеры для различных СУБД и моделей: Microsoft SQL Server, PostgreSQL, Firebird, Oracle, XML, NoSQL.

Для программистов, студентов и других специалистов в области информационных технологий, а также всех интересующихся темой разработки приложений баз данных.

Сайт журнала «Ремонт & Сервис»: www.remserv.ru
Сайт издательства «СОЛОН-Пресс»: www.solon-press.ru

По вопросам приобретения книг обращаться:

ООО «ПЛАНЕТА АЛЬЯНС»

Тел: (499) 782-38-89,

www.aliants-kniga.ru

ISBN 978-2-7466-7383-0

EAN 9782746673830

© Тарасов С. В., 2015

© «СОЛОН-Пресс», 2015

Содержание

Введение	7
Основные понятия	9
База данных и СУБД.....	9
Типы приложений: транзакционная и аналитическая обработка.....	11
Клиент-серверные и встроенные СУБД.....	14
Сноска. Firebird 2.5: состояние	19
Основные модели данных: иерархическая, сетевая, реляционная.....	22
Иерархическая модель.....	22
Сетевая модель	28
Реляционная модель.....	33
Другие подходы и модели данных.....	37
Модель «Сущность-атрибут-значение» (EAV).....	37
Неполно структурированные модели данных	46
Документ-ориентированная модель и NoSQL	48
Многомерные модели данных	53
О применимости NoSQL	56
Множественная и навигационная обработка, менеджеры записей.....	61
Объектная модель и объектно-реляционная проекция.....	65
SQL как универсальный входной язык	75
Проектирование	78
Терминология уровней	78
Первичные и прочие ключи	83
Внешние ключи и связи.....	87
Нормализация и денормализация	89
1НФ.....	90
2НФ.....	91
3НФ.....	92
Деморализуем... то есть денормализуем: «звезда» и «снежинка».....	93
Типовая архитектура данных аналитических приложений	98
Переносимость между СУБД.....	100
Абстрагирование от СУБД.....	101
Абстрагирование от входного языка СУБД.....	102
Использование подмножества входного языка	104
Типовые структуры.....	104

Моделирование связей разных типов.....	105
Хронологические данные.....	109
Иерархические данные и деревья в SQL.....	115
Интернационализация/локализация данных и проброс контекста.....	130
Метаданные.....	138
Реестр объектов и аудит.....	143
Безопасность и доступ к данным.....	145
Проектирование физического хранения.....	151
Физическая организация памяти.....	152
Оперативная и долговременная память.....	155
Дисковые массивы.....	157
Оперативная память.....	160
Индексация данных.....	161
Секционирование данных.....	163
Неполно структурированные данные и высокая нагрузка.....	165
Относительность понятия высокой нагрузки.....	165
Особенности использования PCYБД и HCMД (NoSQL).....	168
Нужно ли моделировать?.....	172
Моделирование против ручного кодирования: пример.....	174
Большие данные как состояние отрасли.....	181
Программирование с испытаниями.....	187
Типы соединений в SQL на примерах.....	187
Исходники и синхронизация структур.....	190
Некоторые особенности программирования.....	200
Параметризация запросов и SQL-инъекции.....	200
Сравнение с неопределёнными (пустыми) значениями.....	203
Работа со строками.....	204
Работа с датами.....	207
Генерация идентификаторов записей.....	209
Транзакции, изоляция и блокировки.....	214
Уровни SQL-92.....	215
Блокировки.....	219
Взаимные блокировки процессов (deadlock).....	222
Версии данных.....	225
Проявления эффектов изоляции.....	227

Толстые транзакции	232
Загрузка данных	233
Пакетная загрузка.....	234
Вставка в толстой транзакции	240
PCyБД и неполно структурированные данные.....	241
Поддержка XML.....	242
Поддержка JSON	250
Выводы.....	253
Постраничные выборки.....	254
Обзор способов постраничной выборки.....	256
Тестирование способов постраничной выборки.....	260
Выводы.....	271
SQL и модульное тестирование	271
Место модульного тестирования в системе испытаний.....	271
Особенности разработки на процедурных расширениях SQL	273
Пример задачи для модульного теста.....	273
Создаём специализированный макроязык.....	276
Остановиться и оглянуться	283
Производительность SQL-запросов	284
Общие рекомендации.....	284
Анализ плана выполнения запроса	286
Поиск узких мест	291
Основы нагрузочного тестирования.....	297
Инструменты и методы	297
Учёт степени параллелизма	301
SQL Server и MongoDB на простом тесте.....	304
Тест вставки записей	304
Запросы и хронометраж	308
Выводы.....	315
Тестовые и демонстрационные базы данных.....	315
Заключение	317
Литература	318

Введение

Разработка приложений баз данных распространена не только в «корпоративном секторе» автоматизации производственных процессов предприятий и их отделов. Из классического определения «программы — это алгоритмы плюс данные» следует, что сколь верёвочке не виться, пройдя через цепочку служб, запрос в итоге обрабатывается системой управления базами данных (СУБД) или её неполнофункциональным аналогом. Активно развивающийся рынок мобильных устройств широко использует встраиваемые (embedded) СУБД, ранее применявшиеся в основном для управления оборудованием.

Если программист сознательно не ограничивает себя разработкой служб и человеко-машинных интерфейсов, взаимодействующих исключительно с другими службами, то вскоре возникает необходимость непосредственной работы с какой-либо СУБД, вероятнее всего реляционной.

Как правило, обладателям профильного образования преподаватели читали соответствующий теоретический курс в вузе [2], сопровождаемый практическими занятиями, пересекающийся по тематике с другими предметами. И, тем не менее, приступая к производственным задачам, вчерашний студент быстро ощутит отличие академических подходов от открывающегося взгляду пейзажа строек в недрах корпоративного софтостроения или в глубоком тылу веб-служб.

Тяжелее пришедшим в программирование из других областей деятельности. Окунувшись в реальность без багажа теории, соответствующего, как говорят американцы, бэкграунда, трудно сформировать в голове целостную картину, охватывающую важные детали происходящего, пропуская несущественные. Не хватает ни времени, ни мотивации читать достаточно скучные, толстые монографии вроде многократно переизданного Дейта [1], когда надо решать текущие задачи. Программисту становится не до вопросов философии кунг-фу, освоить бы побыстрее основные удары и блоки, чтобы получать поменьше оплеух от брыкающейся техники, исполненных значимости системных администраторов и недовольного начальства. Хорошо, если удастся выкроить часок-другой и потренироваться в сиквел¹ на примерах из книжки Грабера [3]...

¹ Сиквелом на жаргоне программистов называется язык SQL (Structured Query Language). Первоначальная его версия так и называлась SEQUEL. Впрочем, если вы будете говорить просто «эс-ку-эль», это не повлияет на ваш профессионализм.

Справиться с растущим потоком информации, не пропуская её через фильтры теоретического багажа, становится всё труднее. Например, последние годы активно пропагандируют NoSQL, может быть там все будет проще, не надо задумываться о нормальных формах и тренировать мозги на непривычную множественную обработку? Или может быть лучше работать через проекцию таблиц на объекты и не использовать прямой доступ к базе данных?

Книга «Софтстроение изнутри» [13] посвятила немалое количество сюжетов теме «**как это не надо делать**» и прогрессирующей в среде программистов некомпетентности в области баз данных, приводящей к катастрофическим для проектов последствиям на более поздних стадиях. В отличие от предтечи, настоящее издание будет носить ровно противоположный характер, следуя принципу «**как это лучше сделать**». Опираясь на опыт работы в продуктивном софтстроении и в технической экспертизе СУБД-решений, автор постарается в рамках повествования помочь вам не утонуть в информационном потоке и разобраться в часто возникающих на практике проблемах, не отрывая их от теории.

Почему не блог, а книга? Действительно, в Сети можно найти немало интересных статей. Однако, во-первых, чтобы найти нужную информацию по правильно заданным ключевым словам, а, во-вторых, оценить достоверность найденной публикации, нужно уже иметь определённый уровень компетенции, который кроме как чтением книг и практикой не поднять. Поэтому рекомендую и начинающим, и программистам с небольшим (2-3 года) опытом не увлекаться малоосмысленным копированием кода со страниц в Сети и прочим натягиванием глобуса на Меркаторову проекцию.

Цель книги не в том, чтобы заменить чтение упомянутых выше монографий или других книг из прилагаемого списка литературы, но подготовить и подвести к нему осознанно, исходя из нужд решения практических задач. Попытаться выстроить мост между бескомпромиссным академическим гранитом и производственными зыбучими песками, бесследно засасывающих неосмотрительных путников своими половинчатыми решениями и постоянной текучкой.

Все-таки, нет ничего практичнее, чем хорошая теория.