



10 сентября 2013 в 13:50

Руководство по проектированию реляционных баз данных (1-3 часть из 15) [перевод]

из песочницы



SQL*, MySQL*

Перевод цикла из 15 статей о проектировании баз данных.

Информация предназначена для новичков.

Помогло мне. Возможно, что поможет еще кому-то восполнить пробелы.

Руководство по проектированию баз данных.

1. Вступление.

Если вы собираетесь создавать собственные базы данных, то неплохо было бы придерживаться правил проектирования баз данных, так как это обеспечит долговременную целостность и простоту обслуживания ваших данных. Данное руководство расскажет вам что представляют из себя базы данных и как спроектировать базу данных, которая подчиняется правилам проектирования реляционных баз данных.

*Базы данных – это программы, которые позволяют сохранять и получать большие объемы связанной информации. Базы данных состоят из **таблиц**, которые содержат **информацию**. Когда вы создаете базу данных необходимо подумать о том, какие **таблицы** вам нужно создать и какие **связи** существуют между информацией в таблицах. Иначе говоря, вам нужно подумать о **проекте** вашей базы данных. **Хороший проект** базы данных, как было сказано ранее, обеспечит целостность данных и простоту их*

Популярное за сутки

Слив данных 180 тысяч пользователей FL.ru

Сайт с нуля на полном стеке БЭМ-технологий. Методология Яндекса

Роутер от оператора? – Нет, спасибо!

12 игр, которые обучают детей программированию

Арабская локализация: окна и рисование

Consulo: Code Coverage, Unity3D и прочие изменения

Fujitsu ETERNUS CD10000: Сeph без забот

Что должен уметь крутой колл-центр по IT-части и какие вообще бывают опции

Мультивендорная корпоративная сеть: мифы и реальность

Генерация текстур планет как в игре Star Control 2

Еще один термостат на Arduino, но с OpenTherm

Удалённое управление для Arduino, проба пера

Ещё один программный UART на ATtiny13

Форматирование Python-кода

Фрактальное пламя — алгоритм построения

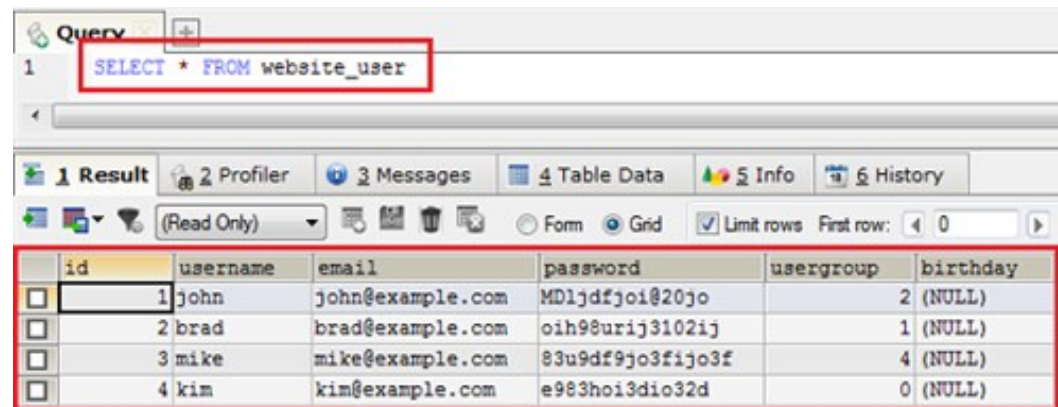
Смайлики в доменных именах

обслуживания.

Структурированный язык запросов (SQL).

База данных создается для хранения в ней информации и получения этой информации при необходимости. Это значит, что мы должны иметь возможность помещать, вставлять (**INSERT**) информацию в базу данных и мы хотим иметь возможность делать выборку информации из базы данных (**SELECT**).

Язык запросов к базам данных был придуман для этих целей и был назван **Структурированный язык запросов** или SQL. Операции вставки данных (INSERT) и их выборки (SELECT) – части этого самого языка. Ниже приведен пример запроса на выборку данных и его результат.



The screenshot shows a database query interface. At the top, a query editor contains the text `SELECT * FROM website_user`, which is highlighted with a red box. Below the editor, a toolbar includes buttons for 'Result', 'Profiler', 'Messages', 'Table Data', 'Info', and 'History'. The 'Result' tab is active, displaying a table of data. The table has columns: id, username, email, password, usergroup, and birthday. There are four rows of data, each with a checkbox in the first column. The entire table is enclosed in a red border.

	id	username	email	password	usergroup	birthday
<input type="checkbox"/>	1	john	john@example.com	MD1jdfjoi@20jo	2	(NULL)
<input type="checkbox"/>	2	brad	brad@example.com	oih98urij3102ij	1	(NULL)
<input type="checkbox"/>	3	mike	mike@example.com	83u9df9jo3fijo3f	4	(NULL)
<input type="checkbox"/>	4	kim	kim@example.com	e983hoi3dio32d	0	(NULL)

SQL – большая тема для повествования и его рассмотрение выходит за рамки данного руководства. Данная статья строго сфокусирована на изложении **процесса проектирования баз данных**. Позднее, в отдельном руководстве, я расскажу об основах SQL.

Реляционная модель.

В этом руководстве я покажу вам как создавать реляционную модель данных.

Компанию Lenovo атаковали в отместку за шпионскую программу Superfish

«Hero Image» — баннеры в параллаксе

Правоохранительные органы обрушили ботнет Ramnit

OpenCage — самый мощный инструмент для геокодирования

все лучшие

Лучшее на Geektimes

ИИ от Google самостоятельно освоил 49 старых игр Atari

Как я имплантировал RFID себе в руку, а потом еще NFC. Часть 1

Krita 2.9: релиз, осуществленный благодаря Kickstarter

Вышел Unreal Engine 4.7 с поддержкой HTML5 и WebGL

Заря электромобилей: XIX век

Премьера фильма «Вселенная Стивена Хокинга» в РФ

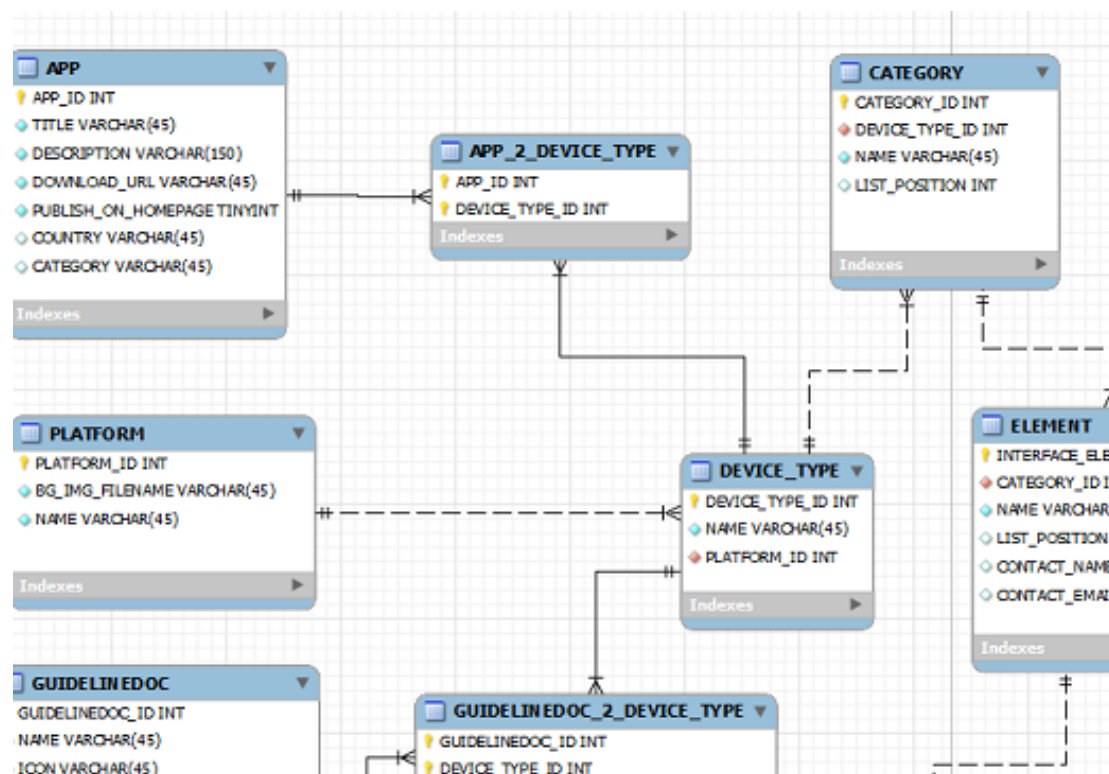
В Великобритании запретили «сексуально оскорбительную» рекламу смартфона

Gemalto отвергла обвинения в массовой краже ключей доступа к SIM-картам

Госорганам РФ предлагают закупать только отечественный софт с 1 июля 2015 года

Цифровой аудиоформат 24/192, и почему в

Реляционная модель – это модель, которая описывает как организовать данные в таблицах и как определить связи между этими таблицами.



Правила реляционной модели диктуют, как информация должна быть организована в таблицах и как таблицы связаны друг с другом. В конечном счете результат можно предоставить в виде диаграммы базы данных или, если точнее, диаграммы «сущность-связь», как на рисунке (Пример взят из MySQL Workbench).

Примеры.

В качестве примеров в руководстве я использовал ряд приложений.

РСУБД.

нем нет смысла. Часть 4 (и последняя)

все публикации

Лучшее на Мегамозге

5 главных ошибок или почему ваши рациональные решения не работают?

Инвестиционный фонд Runa Capital вкладывается в СУБД MariaDB

Философия системы Asana. 4 принципа работы в системе

Апофеоз неудачника или чем мне нравятся провальные стартапы и их основатели

Parallels приобрела разработчика приложений для удаленного доступа 2X Software

Управление ретроспективой или взгляд в прошлое: Руководство Gov.uk

По итогам 2014 года, игры и социальные сервисы – основные точки роста доходов Mail.ru Group

Forbes составил рейтинг 20 самых дорогих компаний Рунета

Mail.ru Group сообщила о неопределенности в продаже HeadHunter

5 шагов к созданию привлекательного сервиса для приложения

все публикации

Вопросы по теме

PCУБД, которую я использовал для создания таблиц примеров – MySQL. MySQL – наиболее популярная PCУБД и она бесплатна.

Утилита для администрирования БД.

После установки MySQL вы получаете только интерфейс командной строки для взаимодействия с MySQL. Лично я предпочитаю графический интерфейс для управления моими базами данных. Я часто использую SQLyog. Это бесплатная утилита с графическим интерфейсом. Изображения таблиц в данном руководстве взяты оттуда.

Визуальное моделирование.

Существует отличное бесплатное приложение MySQL Workbench. Оно позволяет спроектировать вашу базу данных графически. Изображения диаграмм в руководстве сделаны в этой программе.

Проектирование независимо от PCУБД.

Важно знать, что хотя в данном руководстве и приведены примеры для MySQL, проектирование баз данных независимо от PCУБД. Это значит, что информация применима к реляционным базам данных в общем, не только к MySQL. Вы можете применить знания из этого руководства к любым реляционным базам данных, подобным Mysql, Postgresql, Microsoft Access, Microsoft Sql or Oracle.

В следующей части я коротко расскажу об эволюции баз данных. Вы узнаете откуда взялись базы данных и реляционная модель данных.

2. История.

В 70-х – 80-х годах, когда компьютерные ученые все еще носили коричневые

ВОПРОСЫ ПО ТЕМЕ

Знает кто хорошие онлайн курсы по БД и СУБД?

Как сделать выборку из SQL эффективнее всего ?

Что тратит больше ресурсов: JOIN или SELECT?

Стоит ли создавать отдельную таблицу для картинок в БД ?

Как в SQL выбрать пользователей, у которых день рождения в ближайшую неделю?

Как оптимизировать БД?

Как при использовании EAV сделать добавление полей по категориям?

Как правильно составить SQL запрос?

Как задать каскадное удаление записей в MySQL?

Как построить такой запрос?

Посоветуйте веб-интерфейс для выполнения заранее сохраненных MySQL-запросов с параметрами?

Что делать в данном случае?

Как составить такой запрос?

Рекомендуется ли делать такую выборку из базы?

Как добавить внешний ключ в таблицу на колонку с null с уже существующими записями?

Как расширить DISTINCTIVE?

.. ~ ^

смокинги и очки с большими, квадратными оправами, данные хранились бесструктурно в файлах, которые представляли собой текстовый документ с данными, разделенными (обычно) запятыми или табуляциями.



Так выглядели профессионалы в сфере информационных технологий в 70-е. (Слева внизу находится Билл Гейтс).

Текстовые файлы и сегодня все еще используются для хранения малых объемов простой информации. Comma-Separated Values (CSV) — значения, разделённые запятыми, очень популярны и широко поддерживаются сегодня различным программным обеспечением и операционными системами. Microsoft Excel – один из примеров программ, которые могут работать с CSV-файлами. Данные, сохраненные в таком файле могут быть считаны компьютерной программой.

Можно ли сделать такой запрос?

MySQL Как удалить от символа до символа?

MYSQL сложная выборка многое ко многим, как организовать?

Что почитать про нормализацию БД?

Что обсуждают?

MVC и Модель 2. Знания и обязанности компонентов **17**

Как мы домены мониторить начали и что из этого получилось **2**

Фрактальное пламя — алгоритм построения **8**

Правоохранительные органы обрушили ботнет Ramnit **17**

Странности реализации Wi-Fi в метро Москвы **31**

Ещё один программный UART на ATtiny13 **10**

Роутер от оператора? – Нет, спасибо! **87**

Честные приватные свойства в прототипе **29**

WEB Server на базе ENC28j60 + Arduino — проще не бывает **47**

Генерация текстур планет как в игре Star Control 2 **7**

все публикации


```
'tutorial_id', 'title', 'category'  
'1','Access Tutorial','Software'  
'2','Excel Tutorial','Software'  
'3','Database design tutorial','Software'  
'4','Oracle DBA Course ','Software'  
'5','Raid Storage Tutorial ','Hardware'  
'6','Network Security Tutorial','Networks'
```

Выше приведен пример того, как такой файл мог бы выглядеть. Программа, производящая чтение данного файла, должна быть уведомлена о том, что данные разделены запятыми. Если программа хочет выбрать и вывести категорию, в которой находится урок '**Database Design Tutorial**', то она должна строка за строкой производить чтение до тех пор, пока не будут найдены слова '**Database Design Tutorial**' и затем ей нужно будет прочитать следующее за запятой слово для того, чтобы вывести категорию **Software**.

Таблицы баз данных.

Чтение файла строка за строкой не является очень эффективным. В реляционной базе данных данные хранятся в таблицах. Таблица ниже содержит те же самые данные, что и файл. Каждая строка или "запись" содержит один урок. Каждый столбец содержит какое-то свойство урока. В данном случае это заголовок (title) и его категория (category).

Компания дня ?

 Zfort Group

Последняя публикация: РНР-Дайджест № 57
– интересные новости, материалы и
инструменты (9 – 22 февраля 2015)

4657 подписчиков

tutorial_id	title	category
1	Access Tutorial	Software
2	Excel Tutorial	Software
3	Database design tutorial	Software
4	Oracle DBA Course	Software
5	Raid Storage Tutorial	Hardware
6	Network Security Tutorial	Networks

Компьютерная программа могла бы осуществить поиск в столбце tutorial_id данной таблицы по специфическому идентификатору tutorial_id для того, чтобы быстро найти соответствующие ему заголовок и категорию. Это намного быстрее, чем поиск по файлу строка за строкой, подобно тому, как это делает программа в текстовом файле.

Современные реляционные базы данных созданы так, чтобы позволять делать выборку данных из специфических строк, столбцов и множественных таблиц, за раз, очень быстро.

История реляционной модели.

Реляционная модель баз данных была изобретена в 70-х Эдгаром Коддом (Ted Codd), британским ученым. Он хотел преодолеть недостатки сетевой модели баз данных и иерархической модели. И он очень в этом преуспел. Реляционная модель баз данных сегодня всеобща принята и считается мощной моделью для эффективной организации данных.

Сегодня доступен широкий выбор систем управления базами данных: от небольших десктопных приложений до многофункциональных серверных систем с высокооптимизированными методами поиска. Вот некоторые из наиболее известных систем управления реляционными базами данных (РСУБД):

- **Oracle** – используется преимущественно для профессиональных, больших приложений.

- **Microsoft SQL server** – РСУБД компании Microsoft. Доступна только для операционной системы Windows.

- **Mysql** – очень популярная РСУБД с открытым исходным кодом. Широко используется как профессионалами, так и новичками. Что еще нужно?! Она бесплатна.

- **IBM** – имеет ряд РСУБД, наиболее известна DB2.

- **Microsoft Access** – РСУБД, которая используется в офисе и дома. На самом деле – это больше, чем просто база данных. MS Access позволяет создавать базы данных с пользовательским интерфейсом.

В следующей части я расскажу кое-что о характеристиках реляционных баз данных.

3. Характеристики реляционных баз данных.

Реляционные базы данных разработаны для быстрого сохранения и получения больших объемов информации. Ниже приведены некоторые характеристики реляционных баз данных и реляционной модели данных.

Использование ключей.

Каждая строка данных в таблице идентифицируется уникальным “ключом”, который называется первичным ключом. Зачастую, первичный ключ это автоматически увеличиваемое (автоинкрементное) число (1,2,3,4 и т.д). Данные в различных таблицах могут быть связаны вместе при использовании

ключей. Значения первичного ключа одной таблицы могут быть добавлены в строки (записи) другой таблицы, тем самым, связывая эти записи вместе.

Используя структурированный язык запросов (SQL), данные из разных таблиц, которые связаны ключом, могут быть выбраны за один раз. Для примера вы можете создать запрос, который выберет все заказы из таблицы заказов (orders), которые принадлежат пользователю с идентификатором (id) 3 (Mike) из таблицы пользователей (users). О ключах мы поговорим далее, в следующих частях.

	id	username	email	password	usergroup
<input type="checkbox"/>	1	john	john@example.com	MD1jdfjoi@20	2
<input type="checkbox"/>	2	brad	brad@example.com	oih98urij310	1
<input type="checkbox"/>	3	mike	mike@example.com	83u9df9jo3fi	4
<input type="checkbox"/>	4	kim	kim@example.com	e983hoi3dio3	1

Столбец id в данной таблице является первичным ключом. Каждая запись имеет уникальный первичный ключ, часто число. Столбец usergroup (группы пользователей) является внешним ключом. Судя по ее названию, она видимо ссылается на таблицу, которая содержит группы пользователей.

Отсутствие избыточности данных.

В проекте базы данных, которая создана с учетом правил реляционной модели данных, каждый кусочек информации, например, имя пользователя, хранится только в одном месте. Это позволяет устранить необходимость работы с данными в нескольких местах. Дублирование данных называется избыточностью данных и этого следует избегать в хорошем проекте базы данных.

Ограничение ввода.

Используя реляционную базу данных вы можете определить какой вид данных позволено сохранять в столбце. Вы можете создать поле, которое

содержит целые числа, десятичные числа, небольшие фрагменты текста, большие фрагменты текста, даты и т.д.

Field Name	Datatype	Len	Def
id	int	11	
username	varchar	50	
email	varchar	50	
password	varchar	100	
usergroup	int	10	
birthday	date		
*			

Когда вы создаете таблицу базы данных вы предоставляете тип данных для каждого столбца. К примеру, `varchar` – это тип данных для небольших фрагментов текста с максимальным количеством знаков, равным 255, а `int` – это числа.

Помимо типов данных РСУБД позволяет вам еще больше ограничить возможные для ввода данные. Например, ограничить длину или принудительно указать на уникальность значения записей в данном столбце. Последнее ограничение часто используется для полей, которые содержат регистрационные имена пользователей (логины), или адреса электронной почты.

Эти ограничения дают вам контроль над целостностью ваших данных и предотвращают ситуации, подобные следующим:

- ввод адреса (текста) в поле, в котором вы ожидаете увидеть число
- ввод индекса региона с длиной этого самого индекса в сотню символов
- создание пользователей с одним и тем же именем
- создание пользователей с одним и тем же адресом электронной почты
- ввод веса (числа) в поле дня рождения (дата)

Поддержание целостности данных.

Настраивая свойства полей, связывая таблицы между собой и настраивая ограничения, вы можете увеличить надежность ваших данных.



Назначение прав.

Большинство РСУБД предлагают настройку прав доступа, которая позволяет назначать определенные права определенным пользователям. Некоторые действия, которые могут быть позволены или запрещены пользователю: SELECT (выборка), INSERT (вставка), DELETE (удаление), ALTER (изменение), CREATE (создание) и т.д. Это операции, которые могут быть выполнены с помощью структурированного языка запросов (SQL).

Структурированный язык запросов (SQL).












Для того, чтобы выполнять определенные операции над базой данных, такие, как сохранение данных, их выборка, изменение, используется структурированный язык запросов (SQL). SQL относительно легок для понимания и позволяет в т.ч. и уложенные выборки, например, выборка связанных данных из нескольких таблиц с помощью оператора SQL JOIN. Как и упоминалось ранее, SQL в данном руководстве обсуждаться не будет. Я сосредоточусь на проектировании баз данных.

То, как вы спроектируете базу данных будет оказывать непосредственное влияние на запросы, которые вам будет необходимо выполнить, чтобы получить данные из базы данных. Это еще одна причина, почему вам необходимо задуматься о том, какой должна быть ваша база. С хорошо спроектированной базой данных ваши запросы могут быть чище и проще.

Query  

1 SELECT * FROM USER

1 Result 2 Profiler 3 Messages 4 Table Data 5 Info 6 History

(Read Only)           ☒ Limit rows First row:  0

	id	username	email	password	usergroup	birthday
<input type="checkbox"/>	1	john	john@example.com	MD1jdfjoi@20	2	(NULL)
<input type="checkbox"/>	2	brad	brad@example.com	oih98urij310	1	(NULL)
<input type="checkbox"/>	3	mike	mike@example.com	83u9df9jo3fi	4	(NULL)
<input type="checkbox"/>	4	kim	kim@example.com	e983hoi3dio3	0	(NULL)










Переносимость.

Реляционная модель данных стандартна. Следуя правилам реляционной модели данных вы можете быть уверены, что ваши данные могут быть перенесены в другую РСУБД относительно просто.

Как говорилось ранее, проектирование базы данных – это вопрос идентификации данных, их связи и помещение результатов решения данного вопроса на бумагу (или в компьютерную программу). Проектирование базы данных независимо от РСУБД, которую вы собираетесь использовать для ее создания.

В следующей части подробнее рассмотрим первичные ключи.

 MySQL, SQL, СУБД, рсубд

 +6
 
 80845
  505
  hbuser^{1,3}





Похожие публикации

Интересные моменты работы Linq to Sql 19 февраля в 12:08

[Москва, 19.02.2015] Дмитрий Ленев — Менеджеры блокировок в MySQL 13 февраля в 10:08

Русская документация по языку SQL СУБД Firebird 2.5 13 ноября 2014 в 21:58

Раскрываем магию MySQL или о строгости и мягкости MySQL 21 января 2013 в 12:20

GreenSQL: Защита SQL-серверов от инъекций 13 апреля 2011 в 12:54

Сайт MySQL.com скомпрометирован через внедрение SQL-кода 28 марта 2011 в 13:30

Расчет периодов стажа в MySQL 13 марта 2009 в 16:39

Оптимизация работы с MySQL 14 февраля 2008 в 00:09

EXPLAIN — Самая мощная команда MySQL 22 июня 2007 в 17:28

Как выявить медленные SQL запросы? 3 июня 2007 в 03:02

Комментарии (27)



VitaZheltyakov 10 сентября 2013 в 14:22 #

+7 ↑ ↓

Сначала прочитал этот ужас (сама статья, а не перевод) — всё скомкано и непонятно. А потом прочитал, что это цикл из 15 статей, и мне стало страшно.

Если действительно хотите подтянуть знания по проектированию баз данных, то читаем — Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. «Базы данных: Учебник для высших учебных заведений». Именно раздел о проектировании, остальная часть уже прилично устарела.



hbuser 10 сентября 2013 в 14:32 (комментарий был изменён) # ↗ ↑

0 ↑ ↓

Спасибо за наводку. Буду смотреть. А по поводу статьи. Каждый воспринимает информацию по-своему. При прочтении различных источников что-то может оказаться неясным и в данных статьях, если отделять зерна от плевел, я нашел для себя недостающие кусочки информационной мозаики, при всей, порой, простоте (может и не в лучшем смысле этого слова) изложения. Возможно кому-то

это поможет, все мы разные. А не поможет... c'est la vie



norguhtar 10 сентября 2013 в 15:18 #

0 ↑ ↓

Читайте классику citforum.ru/database/osbd/contents.shtml



zBit 10 сентября 2013 в 15:25 #

+1 ↑ ↓

К слову про фрагмент:

Существует отличное бесплатное приложение MySQL Workbench. Оно позволяет спроектировать вашу базу данных графически. Изображения диаграмм в руководстве сделаны в этой программе.

Воркбенч позволяет не только «спроектировать вашу базу данных графически», но и всё остальное что вы делали другими инструментами (SQLyog) и даже больше. Вообще можно было выкинуть всё и использовать только воркбенч.



hbuser 10 сентября 2013 в 17:44 # ↗ ↑

0 ↑ ↓

А ведь никто и не спорит. Позволяет, да, равно как существуют и другие программные продукты. Использование тех или иных программ одной направленности — дело вкуса. Я, например, не пользуюсь SQLyog. Но автор статей, очевидно, пользуется и использовать MySQL Workbench в качестве средства графического моделирования — его выбор.








zBit 10 сентября 2013 в 17:53 (комментарий был изменён) # ↗ ↑

0 ↑ ↓

Просто в своё время очень долго искал путёвый универсальный инструмент для работы с MySQL и понял, что лучше Воркбенча вообще ничего нет, а перепробовал я очень многое. Workbench как швейцарский нож.

Вот сейчас работаю преимущественно с PostgreSQL и не хватает инструмента, который был бы как и Workbench, своего рода швейцарским ножом для PostgreSQL. Но к сожалению, пока ничего лучше, чем EMS SQL Manager (Lite версия) не нашёл. Хотя и он не плох, очень даже не плох.

 **hbuser** 10 сентября 2013 в 18:22 (комментарий был изменён) #   0  

| лучше Воркбенча вообще ничего нет

Частично согласен (в абсолюте судить не берусь), универсальный инструмент. Тоже им пользуюсь, наряду еще с одним попроще.

 **vazir** 11 сентября 2013 в 01:37 # 0  

Я бы это публиковать на хбре постеснялся.

 **hbuser** 11 сентября 2013 в 02:06 #   0  

Вы все не обижайтесь, но если вам это не интересно, то ваш уровень знаний далеко впереди.

Серия статей — для новичков.

Кому-то, надеюсь, это поможет, а это самое главное.

 **GlukKazan** 11 сентября 2013 в 09:46 #   0  

Как раз для новичков эта статья вредна (кстати, я не заметил в статье ссылки на первоисточник). В принципе, отдельные тезисы верны, но изложение... Статью можно разбирать на цитаты:

| База данных создается для хранения в ней информации и получения этой информации при необходимости. Это значит, что мы должны иметь возможность помещать, вставлять (INSERT) информацию в базу данных и мы хотим иметь возможность делать выборку информации из базы данных (SELECT).

Хороший пример. Все верно, не придерешься. Но куда делись UPDATE с DELETE-ом?

| РСУБД, которую я использовал для создания таблиц примеров – MySQL.

MySQL – наиболее популярная РСУБД и она бесплатна.

Аналогично. Относительно бесплатности MySQL и его открытости (особенно после покупки Oracle-ом) есть разные мнения. По поводу наибольшей популярности, тезис также требует доказательств.

- Oracle – используется преимущественно для профессиональных, больших приложений.
 - Microsoft SQL server – РСУБД компании Microsoft. Доступна только для операционной системы Windows.
 - Mysql – очень популярная РСУБД с открытым исходным кодом. Широко используется как профессионалами, так и новичками. Что еще нужно?! Она бесплатна.
 - IBM – имеет ряд РСУБД, наиболее известна DB2.
 - Microsoft Access – РСУБД, которая используется в офисе и дома. На самом деле – это больше, чем просто база данных. MS Access позволяет создавать базы данных с пользовательским интерфейсом.
- В следующей части я расскажу кое-что о характеристиках реляционных баз данных.

Хммм... А где PostgreSQL? Собственно это в продолжение предыдущей цитаты.

В 70-х – 80-х годах, когда компьютерные ученые все еще носили коричневые смокинги и очки с большими, квадратными оправами, данные хранились бесструктурно в файлах, которые представляли собой текстовый документ с данными, разделенными (обычно) запятыми или табуляциями.

Да ладно, все было не так печально. Например индексы тогда уже были (деталей я за давностью лет уже не помню, но можно посмотреть [здесь](#), мы ведь говорим про CODASYL?). И потом, что значит «бесструктурно», если речь идет о CSV-файлах? Вполне себе структура. Кроме того, замалчивается тот факт, что данные, в то время, хранились на лентах. Соответственно, структуры данных, ориентированные на выборку с произвольным доступом, были не востребованы.

Компьютерная программа могла бы осуществить поиск в столбце tutorial_id данной таблицы по специфическому идентификатору tutorial_id для того, чтобы быстро найти соответствующие ему заголовок и категорию. Это намного быстрее, чем поиск по файлу строка за строкой, подобно тому, как это делает программа в текстовом файле.

Это быстрее далеко не во всех обстоятельствах. Очень часто полный просмотр выгоднее чем доступ по индексу.

В проекте базы данных, которая создана с учетом правил реляционной модели данных, каждый кусочек информации, например, имя пользователя, хранится только в одном месте. Это позволяет устранить необходимость работы с данными в нескольких местах. Дублирование данных называется избыточностью данных и этого следует избегать в хорошем проекте базы данных.

Это самое оригинальное, лаконичное (и очень поверхностное) изложение принципов нормализации БД, которое я видел.

И так вся статья. Изложение поверхностное, неряшливое и местами неточное. Я убежден, что эта статья вредна для начинающих разработчиков и совершенно бесполезна для чуть более опытных. Настоятельно рекомендую переключиться на перевод чего либо более полезного.



zBit 11 сентября 2013 в 12:33 # ↗ ↑

0 ↑ ↓

Добротно расписали)

Тоже подумал, «А где же PostgreSQL?». Осмелюсь предположить, что оригинальные статьи очень старые.

И я согласен с вами в том, что статья вредна для начинающих.



psykrab 15 сентября 2013 в 14:38 # ↗ ↑

0 ↑ ↓

а Sybase где? обидно однако...



GlukKazan 15 сентября 2013 в 15:25 # ↵ ↑

0 ↑ ↓

За Sybase тоже обидно, но он не бесплатный. А про MySQL утверждается что он чуть ли не самый популярный и бесплатный, что несколько не соответствует действительности.

Что еще надо ?

Надо PostgreSQL, дабы у новичков не складывалось неправильное представление об арсенале бесплатных РСУБД



zBit 15 сентября 2013 в 17:11 # ↵ ↑

+1 ↑ ↓

Вообще предпочтительнее начинать, как мне кажется, именно с PostgreSQL, а не с MySQL. Функционал значительно шире и зная PG можно будет быстро разобраться с остальными. Например, работа с теми же самыми хранимыми процедурами, которых нет в MySQL. Ещё вьюхи, не зависящая от типа таблиц возможность использовать внешние ключи и т.п.



Fortop 28 сентября 2013 в 15:55 # ↵ ↑

0 ↑ ↓

Вот вы являетесь наглядным примером того, что зная PG в MySQL не смогли разобраться

dev.mysql.com/doc/refman/5.0/en/stored-routines.html

Соответственно ценность вашего мнения близка к вашей оценке наличия хранимых процедур в MySQL — нулевая.



zBit 28 сентября 2013 в 21:52 # ↵ ↑

+1 ↑ ↓

Согласен, лажанул.

Сейчас настрочил длинное оправдание, но подумал, кому это надо и стёр его:-)

В общем, в последний раз с MySQL я работал довольно таки давно, но уже тогда была версия 5.x, вот точно непомню, скорее всего одна из первых. И в данный момент мои знания о MySQL, видимо, просто ничтожны, т.к. уже давно перестал работать с ним и слежу теперь больше за обновлениями PG.

После MySQL пришлось столкнуться с Firebird, SQLite и, собственно PG. И последний привёл меня в дикий восторг. PG в некоторых местах заставляет думать по-другому. И у меня возникла мысль, что, если бы я начал не с MySQL, а с PG, то, скорее всего у меня бы остались совершенно другие впечатления от MySQL.

В общем можно гадать о том как было бы, но моё мнение (ИМХО), мне надо было начинать с того от чего мои впечатления были лучше.

P.S.: каюс, надо было пробежаться по документации MySQL, прежде чем писать.



Win32Sector 11 сентября 2013 в 08:19 #

+1 ↑ ↓

Спасибо за перевод.



hbuser 11 сентября 2013 в 21:20 # ↗ ↑

0 ↑ ↓

Приятно видеть простое «спасибо»... Вам спасибо за то, что читали.



NikitOS9 11 сентября 2013 в 19:29 #

0 ↑ ↓

еще вот хорошая и толстая книга
Системы баз данных. Полный курс — Гектор Гарсиа-Молина, Джеффри Ульман,
Дженнифер Уидом
www.ozon.ru/context/detail/id/1351096/



hbuser 11 сентября 2013 в 21:44 # ↗ ↑

0 ↑ ↓

Спасибо. Оглавление вызывает интерес.



ColorPrint 15 сентября 2013 в 00:52 # ↩ ↑

0 ↑ ↓

А на курсе еще есть стэнфордский курс по базам данных от Дженнифер Уидом
class.coursera.org/db/class/index



hbuser 16 сентября 2013 в 01:24 # ↩ ↑

0 ↑ ↓

Неплохой курс.
Не знал про данный ресурс. Спасибо.



FaNtAsY 12 сентября 2013 в 11:32 #

+1 ↑ ↓

Вы пишете, что SQLyog — бесплатная утилита.
Странно, я думал, что она — платная
www.webyog.com/product/sqlyog



m_z 18 сентября 2013 в 22:45 # ↩ ↑

0 ↑ ↓

code.google.com/p/sqlyog/downloads/list, не?



FaNTAsY 18 сентября 2013 в 23:03 # ↩ ↑

0 ↑ ↓

Обалдеть!
Я всегда думал, что она платная!
Интересно, а это одна и та же программа? Я обязательно проверю завтра.

Спасибо большое за ссылку!



Iutov 14 сентября 2013 в 00:28 #

0 ↑ ↓

Спасибо, несколько лет назад я искал такой текст, но ничего не нашел. Может, плохо искал. Так или иначе, общую картину пришлось собирать самостоятельно из разрозненных фрагментов.



Рад, что пригодилось.

Только зарегистрированные пользователи могут оставлять комментарии.
[Войдите](#), пожалуйста.

Яндекс.Метро следит за тобой

Как растащить class-файл

Встраиваем бэктор в
публичный ключ RSA

Brainstorage

Требуется ведущий программист asp

Project manager в Веб Студию

Ведущий HTML-Верстальщик

PHP Developer

Веб-программист (для CMS Joomla)

.NET разработчик

Веб-программист

Platform engineer

Андроид разработчик Рокетбанка

JavaScript developer

все вакансии

ФРИЛАНСИМ

Нужно разработать мобильное приложение на Xamarin

Нарисовать закрепленный пост для сайта storytut.ru

Разработать сайт с калькулятором на стороне клиента на...

Нужен SMM для проекта (5 проектов)

Прикладной протокол поверх udp, клиент и сервер

Разработать проект на Python/Django

Оптимизировать сайт. Внести исправления, уменьшить в...

Сайт - биржа предметов Steam

Верстка сайта на Bitrix

Дизайн портфолио для программиста

все заказы

[Войти](#)

[Регистрация](#)

[Разделы](#)

[Публикации](#)

[Инфо](#)

[О сайте](#)

[Услуги](#)

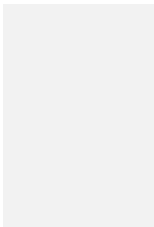
[Реклама](#)

[Разное](#)

[Приложения](#)

© ТМ

[Служба поддержки](#)



[Хабы](#)

[Компании](#)

[Пользователи](#)

[Q&A](#)

[Песочница](#)

[Правила](#)

[Помощь](#)

[Соглашение](#)

[Спецпроекты](#)

[Тарифы](#)

[Контент](#)

[Семинары](#)

[Тест-драйвы](#)

[Помощь стартапам](#)

[Мобильная версия](#)

