

рый справился бы со всеми существующими проблемами отслеживания объектов. Эта статья призвана проанализировать существующие проблемы слежения за объектами, а также предложить способы их решения.

Ключевые слова: обнаружение объекта, отслеживание объекта, вычитание фона, вычитание изображения, оптический поток, ускоренные надежные функции

Авторлар туралы ақпарат:

Бектемысова Г.У. – техника ғылымдарының кандидаты, «Компьютерлік инженерия және ақпараттық қауіпсіздік» кафедрасының қауымдастырылған профессоры, Халықаралық ақпараттық технологиялар университеті, Алматы, Қазақстан

Айнабек Ж.Б. – бакалавр дәрежесін аяқтады және Халықаралық ақпараттық технологиялар университетінде магистратурада оқиды.

Сведения об авторах:

Бектемысова Г.У. – к.т.н., ассоциированный профессор кафедры «Компьютерная инженерия и информационная безопасность», Международный университет информационных технологий, Алматы.

Айнабек Ж.Б. – магистр Международного университета информационных технологий.

About authors:

Bektemisova G.U. – candidate of technical sciences, associate professor of the department of Computer Engineering and Information Security, International Information Technologies University, Almaty, Kazakhstan.

Ainabek Zh.B. – completed bachelor's degree and is studying for a master's degree at the International Information Technologies University.

УДК 004.65.004.7

Алданазар А.А.

Международный университет информационных технологий, Алматы, Казахстан

ПРОЕКТИРОВАНИЕ СХЕМЫ БАЗЫ ДАННЫХ ДЛЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПО СБОРУ ДАННЫХ

Аннотация. В статье приведен процесс разработки схемы БД для системы по работе с динамическими данными, а также основная концепция создания структуры таблиц для интернет магазина.

Ключевые слова: разработка, схема базы данных, форматирование данных, тестирование системы, база данных

Введение

Схема базы данных - это ее структура, описанная на формальном языке, поддерживаемом системой управления базами данных (СУБД). Термин «схема» относится к организации данных, как к схеме построения базы данных (разделенной на таблицы базы данных в случае реляционных баз данных). Формальное определение схемы базы данных - это набор формул, называемых ограничениями целостности, наложенными на базу данных. Эти ограничения целостности обеспечивают совместимость между частями схемы. Все ограничения выражаются на одном языке. База данных может рассматриваться как структура в реализации языка баз данных [1]. Состояния созданной концептуальной схемы преобразуются в явное

отображение, схему базы данных. Это описывает, как реальные сущности моделируются в базе данных.

«Схема базы данных определяется, основываясь на знаниях администратора базы данных о возможных приложениях, факты, которые могут войти в базу данных, или факты, представляющие интерес для возможных конечных пользователей». Понятие схемы базы данных играет ту же роль, что и понятие теории в исчислении предикатов. Модель этой «теории» близко соответствует базе данных, которую можно рассматривать в любой момент времени как математический объект. Таким образом, схема может содержать формулы, представляющие ограничения целостности специально для приложения и ограничения специально для типа базы данных, выраженные на одном языке базы данных. В реляционной базе данных схема определяет таблицы, поля, отношения, представления, индексы, пакеты, процедуры, функции, очереди, триггеры, типы, последовательности, материализованные представления, синонимы, ссылки на базы данных, каталоги, схемы XML и другие элементы.

База данных обычно хранит свою схему в словаре данных. Хотя схема определяется на языке текстовой базы данных, этот термин часто используется для обозначения графического изображения структуры базы данных. Другими словами, схема - это структура базы данных, которая определяет объекты в базе данных.

Основная концепция разработки структуры базы данных

Процесс проектирования базы данных требует тщательного изучения предметной области. Продуманная схема может в дальнейшем дать огромную гибкость при накладывании записей в таблицах, а также:

- Экономить дисковое пространство за счет устранения избыточных данных.
- Сохранять точность и целостность данных.
- Предоставлять доступ к данным полезными способами.

Для создания эффективной и полезной базы данных необходимо соблюдать надлежащий процесс, включая следующие этапы:

- Анализ требований или определение цели вашей базы данных.
- Организация данных в таблицы.
- Определение первичных ключей и анализ отношений.
- Нормализация для стандартизации таблиц.

Понимание цели базы данных будет влиять на выбор на протяжении всего процесса проектирования. Очень важно убедиться, что вы рассматриваете базу данных со всех сторон. Например, в нашем случае база данных должна быть спроектирована таким образом, чтобы в дальнейшем можно было гибко внедрять неструктурированные данные под формат интернет магазина.

Так как в нашей системе присутствует логика множественных магазинов было принято решение добавить таблицу магазинов:

- shops (магазины);
- Id (уникальный идентификатор записи в таблице);
- created_at (дата создания записи);
- updated_at (дата обновления записи);
- deleted_at (дата удаления записи);
- name (наименование);
- url (ссылка на магазин);
- main_image_id (ссылка на запись в таблице);
- description (описание).

Для хранения данных файлов используется отдельная таблица, где указываются типы, а также путь нахождения файла:

- system_files (системные файлы);
- Id (уникальный идентификатор записи в таблице);
- Bucket (папка хранения файла);
- Name (наименование файла);
- type (тип файла).

Таблица, хранящая данные о товарах:

- shop_items (товары);
- Id (уникальный идентификатор записи в таблице);
- created_at (дата создания записи);
- updated_at (дата обновления записи);
- deleted_at (дата удаления записи);
- currency (валюта);
- name (наименование);
- price (цена);
- is_visible (видимость товара в магазине для пользователей);
- category_id (внешний ключ категорий);
- main_image_id (внешний ключ файла);
- url (ссылка на товар);
- description (описание товара);
- shop_id (уникальный идентификатор магазина).

Таблица для хранения данных о категориях товаров;

- shop_categories (категории товаров);
- id (уникальный идентификатор записи в таблице);
- created_at (дата создания записи);
- updated_at (дата обновления записи);
- deleted_at (дата удаления записи);
- name (наименование категории).

Как вы могли заметить таблица магазинов (shops) связана с таблицей файлов (system_files) через колонну “main_image_id”, которая в свою очередь является внешним ключом. Таблица товары (shop_items) связана с таблицей магазинов (shops) через колонну “shop_id”. Таблица товаров также связана с таблицей категорий товаров (shop_categories) через колонну (category_id).

Тестирование разработанной схемы базы данных

Когда у вас есть предварительный проект для вашей базы данных, вы можете применить правила нормализации, чтобы убедиться, что таблицы структурированы правильно [2]. Считайте эти правила отраслевыми стандартами. Тем не менее, не все базы данных подходят для нормализации. В общем, базы данных оперативной обработки транзакций (сокращенно OLTP), в которых пользователи занимаются созданием, чтением, обновлением и удалением записей, должны быть нормализованы. Базы данных онлайн-аналитической обработки (OLAP), которые отдают предпочтение анализу и составлению отчетов, могут работать лучше при определенной степени денормализации, поскольку упор делается на скорость вычислений. К ним относятся приложения поддержки принятия решений, в которых данные необходимо быстро анализировать, но не изменять. Каждая форма или уровень нормализации включает правила, связанные с более низкими формами.

Первая нормальная форма

Первая нормальная форма (сокращенно 1NF) указывает, что каждая ячейка в таблице может иметь только одно значение, а не список значений.

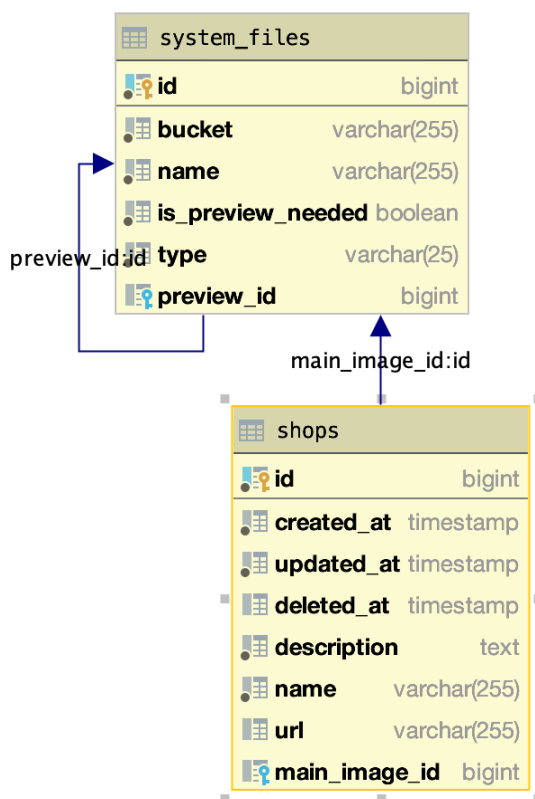


Рис. 1 – Схема таблицы магазинов

Как отмечено на рисунке 1, есть таблица магазинов с надлежащими к ней колоннами. Каждая ячейка будет хранить в себе лишь одно значение. Если бы сложилась такая ситуация, при которой, например, мы бы хранили фотографии списком в одной колонне с разделителями между каждым значением, то это считалось бы неправильной логикой структурирования таблиц, а именно денормализованной структурой таблиц.

Вторая нормальная форма

Вторая нормальная форма (2NF) требует, чтобы каждый из атрибутов полностью зависел от всего первичного ключа. Это означает, что каждый атрибут должен напрямую зависеть от первичного ключа, а не косвенно через какой-либо другой атрибут. Например, если атрибут «name», который зависит от «main_image_id», который, в свою очередь, зависит от «id», считается, что он имеет частичную функциональную зависимость, и таблица, содержащая эти атрибуты, не соответствует второй нормальной форме [3]. Более того, таблица с первичным ключом, состоящим из нескольких полей, нарушает вторую нормальную форму, если одно или несколько других полей не зависят от каждой части ключа.

Третья нормальная форма

Третья нормальная форма (3NF) добавляет к этим правилам требование, чтобы каждый неключевой столбец был независим от всех остальных столбцов. Если изменение значения в одном неключевом столбце вызывает изменение другого значения, эта таблица не соответствует третьей нормальной форме. Следовательно, между таблицами не должно быть транзитивных зависимостей.

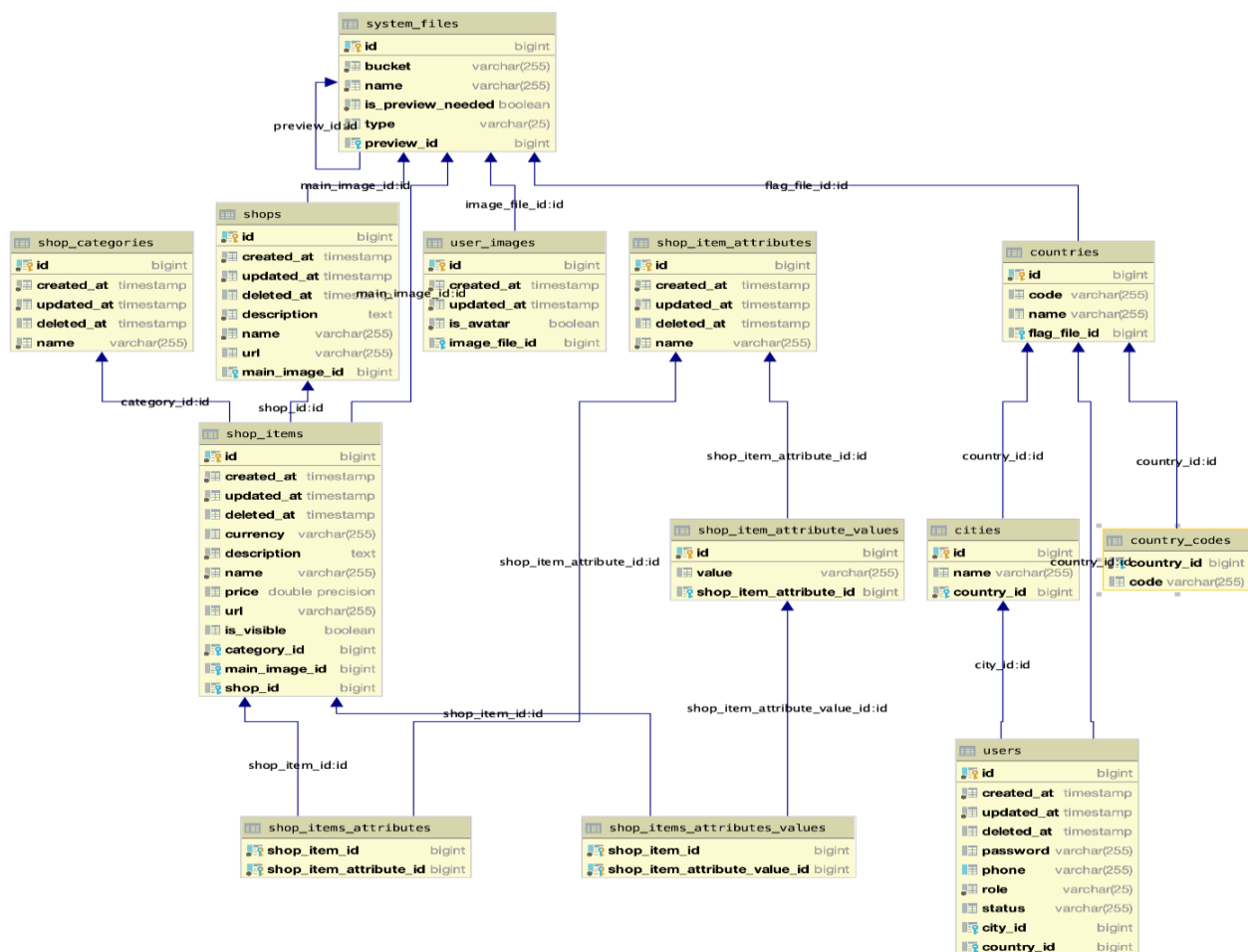


Рис. 2 – Общая схема базы данных

Заключение

Известные методы проектирования баз данных (БД) возникали в процессе разработки все более сложных информационных систем (ИС), которые должны были учитывать потребности не одного пользователя, а больших групп и коллективов [4]. Одна такая интегрированная база данных была создана для решения множества задач, каждая из которых использовала только «свою» часть данных, обычно перекрываясь с частями, используемыми в других задачах. Поэтому наиболее важными методами проектирования стали методы устранения избыточности данных. Эти методы были связаны с другими средствами обеспечения логической целостности данных. Было сформулировано фундаментальное требование отделить программы от интегрированных данных. Этот принцип направлен на отчуждение данных как корпоративного ресурса, а также важен тем, что консервативные по своей природе данные отделяются от приложений, которые могут подвергаться частым изменениям. Другой важной проблемой при проектировании базы данных было обеспечение необходимых рабочих параметров, таких как объем внешней памяти или время для выполнения различных операций.

Создание корпоративных баз данных в контексте Проекта новой системы - это деятельность, в которой используются многие методы классического дизайна, но требуется другая организация и множество дополнительных методов, а также новых, которые заменят некоторые из тех, что были разработаны 10 или больше лет назад. В соответствии с принципом защиты от компьютерных революций следует продолжать использовать классические методы проектирования баз данных, но только в тех областях, где они действительно полезны. Методы проектирования, рассматриваемые в конкретных проектах корпоративных ИБ и БД, и

соответствующие инструменты должны быть проверены на их способность предоставлять функции в соответствии с требованиями новой системы.

ЛИТЕРАТУРА

1. Database https://en.wikipedia.org/wiki/Database_normalization
2. Martin Kleppmann, Designing Data – Intensive Applications
3. Very Large Database. Endowment inc.
4. Margaret Rouse [https://searchsqlserver.techtarget.com/definition/database20%is%20a%](https://searchsqlserver.techtarget.com/definition/database20%is%20a%20)

Алданазар А.А.

Деректерді жинау үшін ақпараттық жүйенің деректер болмысын жасау

Аңдатпа. Мақалада динамикалық деректермен жұмыс істейтін жүйеге арналған мәліметтер қорының схемасын әзірлеу процесі, сондай-ақ интернет-дүкен үшін кесте құрылымын құрудың негізгі тұжырымдамасы сипатталған.

Түйінді сөздер: әзірлеу, мәліметтер базасының схемасы, мәліметтерді форматтау, жүйені тестілеу, деректер қоры

Aldanazar A.A.

Designing the database scheme for the information system for data collection

Annotation. The article describes the process of developing a database schema for a system for working with dynamic data, as well as the basic concept of creating a table structure for an online store.

Key words: development, database schema, data formatting, system testing, database

Автор туралы ақпарат:

Алданазар А.А. – Халықаралық ақпараттық технологиялар университетінде магистрант.

Сведения об авторе:

Алданазар А.А. – магистрант Международного университета информационных технологий.

УДК 004.65.004.75

Ускенбаева Р.К., Бектемысова Г.У., Керимбай Е.Т.

Международный университет информационных технологий, Алматы, Казахстан

РАЗРАБОТКА МОДУЛЯ ПЕРСОНАЛЬНЫХ ДАННЫХ ДЛЯ СИСТЕМЫ ИММИГРАЦИИ ГРАЖДАН

Аннотация. Рассматривается проблема внешней и внутренней иммиграции населения Республики Казахстан. Для этого разрабатывается интеллектуальная информационная система для анализа данных об иммиграции в реальном времени. На сегодняшний день разработано несколько моделей. В данной статье рассматривается разработка модуля персональных данных для системы иммиграции граждан.

Ключевые слова: персональные данные регистрация, портал, недвижимость, сведения, гражданин, обработка данных