

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

РАЗРАБОТЧИК БАЗ ДАННЫХ

РЕФЕРАТ

Студентки 1 курса 151 группы
направления 09.03.04 Программная инженерия
факультета компьютерных наук и информационных технологий
Стюхиной Ангелины Сергеевны

Саратов 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Базы данных и СУБД	4
1.1 Определение основных понятий: система база данных и СУБД.....	4
1.2 Назначение СУБД.....	4
1.3 Архитектура системы баз данных	5
2 Принципы проектирования баз данных	6
2.1 Общая нотация модели «Сущность-связь»	6
2.1.1 Представление сущностей.....	6
2.1.2 Представление связей	7
2.2 Реляционная модель данных	7
3 Запросы и управление базой данных	8
3.1 Языки запроса и управления данными.....	8
3.2 Язык запросов SQL	8
4 Современные тенденции в разработке баз данных	10
4.1 BigData и NoSQL базы данных	10
4.2 Облачные базы данных	11
ЗАКЛЮЧЕНИЕ	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	14

ВВЕДЕНИЕ

XXI век по праву считается веком информации и информационных технологий. Трудно представить сферу деятельности, в которой ни коем образом не приходится работать с большими объемами данных. Исходя из этого, мы закономерно сталкиваемся со следующими задачами: в каком виде хранить, как быстро искать и обрабатывать, как распространять между различными пользователями информацию, обеспечивая при этом её безопасность и целостность.

Наиболее эффективным ответом на перечисленные вопросы является создание базы данных. База данных — структурированное поименованное хранилище информации. Они используются в науке, здравоохранении, образовании, бизнесе и не только.

Базы данных могут быть разработаны на различных языках программирования, включая Java, Python, C++, Ruby, PHP, JavaScript и другие. Кроме того, существует множество специализированных языков запроса и управления данными, таких как: SQL, NoSQL или MongoDB. Но, очевидно, недостаточно спроектировать базу данных. Её необходимо поддерживать, обновлять, чистить, каким-то образом к ней обращаться.

Неудивительно, что разработчик баз данных — это довольно востребованная специальность на рынке IT-специалистов. Давайте подробнее рассмотрим, что входит в обязанности такого сотрудника и какими знаниями и навыками он должен обладать.

1 Базы данных и СУБД

1.1 Определение основных понятий: система база данных и СУБД

Система баз данных — это компьютеризированная система хранения записей, т.е. компьютеризированная система, основное назначение которой хранить информацию, предоставляя пользователям средства её извлечения и модификации.

Любая система баз данных состоит из 4 основных частей: данные, аппаратное обеспечение, пользователи и программное обеспечение. Наиболее интересным является последний из перечисленных компонент.

Между физической базой данных (т.е. данными, которые хранятся на компьютере) и пользователями системы располагается уровень программного обеспечения, который можно называть по-разному: диспетчер базы данных (database manager), сервер базы данных (database server) или, что более привычно, система управления базами данных, СУБД (DataBase Management System, DBMS).

Все запросы пользователей на получение доступа к базе данных обрабатываются СУБД. Все имеющиеся средства добавления файлов (или таблиц), выборки и обновления данных в этих файлах или таблицах также предоставляет СУБД. Основная задача СУБД — дать пользователю базы данных возможность работать с ней, не вникая во все подробности работы на уровне аппаратного обеспечения.

СУБД — это наиболее важный, но не единственный программный компонент системы. В числе других можно назвать утилиты, средства разработки приложений, средства проектирования, генераторы отчетов и диспетчер транзакций.

Наиболее распространенными являются две СУБД — MySQL и Oracle. Зачастую, саму СУБД также называют базой данных, из-за чего возникает путаница в терминах. Это, безусловно, неправильно, но ошибка широко распространена [1].

1.2 Назначение СУБД

Любая СУБД обязана обеспечить реализацию следующих требований:

1. Позволять пользователям создавать новые базы данных и определять их схемы с помощью некоторого специализированного языка, называемого языком определения данных (Data Definition Language, DDL).

2. Предлагать пользователям возможности задания запросов (т.е. вопросов, затрагивающих те или иные аспекты информации, хранящейся в базе данных) и модификации данных средствами соответствующего языка запросов (query language), или языка управления данными (Data Manipulation Language, DML).
3. Поддерживать способность сохранения больших объемов информации до многих гигабайтов и более на протяжении длительных периодов времени, предотвращая опасность несанкционированного доступа к данным и гарантируя эффективность операций их просмотра и изменения.
4. Управлять единовременным доступом к данным со стороны многих пользователей, исключая возможность влияния действий одного пользователя на результаты, получаемые другим, и запрещая совместное обращение к данным, чреватое их порчей [2].

1.3 Архитектура системы баз данных

Большинство систем баз данных описывается архитектурой, предложенной исследовательской группой ANSI/SPARC (Study Group on Data Management Systems), так называемой архитектурой ANSI/SPARC.

Архитектура ANSI SPARC включает три уровня:

1. Внутренний уровень (физический) наиболее близок к физическому хранению информации, т.е. связан со способами сохранения информации на физических устройствах.
2. Внешний уровень (пользовательский логический) наиболее близок к пользователям, т.е. связан со способами представления данных для отдельных пользователей.
3. Концептуальный уровень (общий логический или просто логический) является «промежуточным» уровнем между двумя первыми.

Если внешний уровень связан с индивидуальными представлениями пользователей, то концептуальный уровень связан с обобщенным представлением пользователей. Большинству пользователей нужна не вся база данных, а только ее небольшая часть, поэтому может существовать несколько внешних представлений, каждое из которых состоит из более или менее абстрактного представления определенной части базы данных, и только одно концептуальное представление, состоящее из абстрактного представления базы данных в целом [3].

2 Принципы проектирования баз данных

Поговорим о том, как проектируются базы данных на самом верхнем уровне — концептуальном, когда рассматривается только суть хранимых данных, их свойства и связи между различными элементами без привязки к особенностям физической реализации и конкретной СУБД.

Словарь Вебстера определяет модель так: «модель — это описание или аналогия, используемая для визуализации чего-либо, что не может наблюдаться непосредственно. Основное предназначение модели облегчить понимание структур данных и их свойств, связей и ограничений.

Проектировщик БД использует модели данных как средство коммуникации, обеспечивающее взаимодействие проектировщиков, прикладных программистов и конечных пользователей. Если модель данных тщательно проработана, то она поможет лучше понять положение дел в организации.

Рассмотрим наиболее популярную модель «Сущность-связь» и подробно разберем, как описываются в терминах этой модели объекты (сущности), их атрибуты и связи [4].

2.1 Общая нотация модели «Сущность-связь»

ER-модель — это представление базы данных в виде наглядных графических диаграмм. ER-модель визуализирует процесс, который определяет некоторую предметную область. Диаграмма «сущность»-«связь» — это диаграмма, которая представляет в графическом виде сущности, атрибуты и связи.

2.1.1 Представление сущностей

Сущность — это абстрактный объект определенного вида с уникальным именем. Под сущностью на уровне ER-моделирования на самом деле подразумевается множество сущностей (entity set), не содержащее дубликатов, а не единственная сущность. Иначе говоря, слово «сущность» в ER-моделировании соответствует таблице. В ER-модели отдельная строка таблицы называется экземпляром сущности (entity instance).

Атрибуты описывают свойства сущностей. Например, сущность STUDENT может включать в себя атрибуты NAME (фio) и NUMBER (номер зачетки).

У атрибутов имеются домены. Домен — это набор возможных значений атрибута. Например, домен для числового атрибута средней оценки студента

(Grand Point Average, GPA) может быть записан в виде интервала [1,5), поскольку минимальное значение GPA равно 1, а максимальное значение — 5.

2.1.2 Представление связей

Одно из требований к организации базы данных — обеспечение возможности отыскания одних сущностей по значениям других.

Связь (relationship) — это ассоциирование сущностей. Сущности, участвующие в связи, называются участниками (participants). Связи не могут существовать без связываемых сущностей. Они могут быть бинарными, тернарными, N-арными и рекурсивными.

В качестве названия связи используется глагол в активной или пассивной форме. Например, «студент (STUDENT) занимается в группе (CLASS)» или «преподаватель (PROFESSOR) ведет занятия в группе (CLASS)».

Связи между сущностями всегда действуют в обоих направлениях, поэтому связь трудно классифицировать, если известна только одна ее сторона [5].

2.2 Реляционная модель данных

Модель «сущность-связь» используется для проектирования баз данных, которые затем могут быть реализованы в виде реляционной, иерархической, сетевой или иной модели данных. Реляционная модель является одной из самых популярных среди современных информационных систем.

Существует строгое определение реляционной базы данных и теоретический фундамент для операций, которые могут быть выполнены над ней. Однако в данном случае более полезно неформальное определение.

Реляционной называется база данных, в которой все данные, доступные пользователю, организованы в виде таблиц, а все операции базы данных выполняются над этими таблицами.

Связи между данными устанавливаются с помощью ключей. Первичный ключ (Primary key) — это уникальный идентификатор каждой записи в таблице реляционной базы данных. Внешний ключ (Foreign key) — это поле или группа полей в таблице, которые связывают ее с другой таблицей. Он ссылается на первичный ключ другой таблицы и используется для создания связей между таблицами. Существуют и другие типы ключей [6].

3 Запросы и управление базой данных

3.1 Языки запроса и управления данными

Языки запроса и управления данными — это специальные языки, которые используются для работы с базами данных. Они позволяют выполнять различные операции, такие как создание, изменение, удаление и поиск данных в базе данных. Языки запроса используются для поиска и извлечения данных из базы данных, а языки управления данными — для управления самой базой данных, создания и настройки таблиц, индексов, ограничений целостности и других объектов базы данных. Знание этих языков является необходимым для работы с большинством современных информационных систем и приложений.

Среди наиболее популярных языков запроса можно выделить SQL (Structured Query Language), который широко используется в реляционных базах [7].

3.2 Язык запросов SQL

Первые версии языка SQL, тогда ещё называвшегося SEQUEL, были разработаны компанией IBM в рамках реализации реляционной СУБД System R в 1974 году. SQL является стандартным языком для работы с реляционными базами данных и в настоящее время поддерживается практически всеми продуктами, представленными на рынке.

Официальный стандарт языка SQL был опубликован Американским национальным институтом стандартов (American National Standards Institute, ANSI) и Международной организацией по стандартизации (International Standards Organization, ISO) в 1986 году, после чего был расширен в 1989 году, а затем — в 1992, 1999, 2003 и 2006 годах.

SQL является языком реляционных баз данных, поэтому он стал популярным тогда, когда популярной стала реляционная модель представления данных. Табличная структура реляционной базы данных со строками и столбцами интуитивно понятна пользователям, поэтому язык SQL является простым для изучения.

Инструкции SQL похожи на обычные предложения английского языка, что упрощает их понимание. Частично это обусловлено тем, что инструкции SQL описывают данные, которые необходимо получить, а не способ их поиска.

Перечислим основные инструкции SQL:

1. Выборка информации из базы данных с помощью инструкции SELECT.

Можно извлечь все данные из таблицы или лишь часть из них, отсортировать их и получить итоговые значения, вычисляя суммы и средние величины.

2. Изменение информации в базе данных. Инструкция INSERT добавляет данные, инструкция DELETE удаляет их, а инструкция UPDATE обновляет существующие данные.
3. Создание и изменение базы данных путем определения структуры новых таблиц и удаления таблиц, ставших ненужными, для чего применяются инструкции CREATE И DROP.

Также SQL используется для управления доступом к базе данных. С помощью инструкций SQL предоставляются и отменяются разного рода привилегии для различных пользователей.

Пример запроса на языке SQL:

```
SELECT CITY  
FROM OFFICES;
```

Ответом на данный запрос будут все данные из столбца CITY таблицы OFFICES (рис. 1) [6].

Таблица OFFICES

OFFICE	CITY	REGION	TARGET	SALES
22	Denver	Western	\$300,000.00	\$186,042.00
11	New York	Eastern	\$575,000.00	\$692,637.00
12	Chicago	Eastern	\$800,000.00	\$735,042.00
13	Atlanta	Eastern	\$350,000.00	\$367,911.00
21	Los Angeles	Western	\$725,000.00	\$835,915.00

Рисунок 1 – Простая реляционная база данных

К плюсам языка SQL можно отнести его относительную простоту и низкий порог входа, универсальность, эффективность при работе с большими объемами данных. Также SQL имеет широкую поддержку и множество ресурсов для помощи при работе с языком.

Но одновременно с этим неопытному пользователю может быть достаточно сложно составить многоуровневый запрос. Сам SQL может иметь проблемы с производительностью при выполнении таких сложных запросов и с масштабированием на большие базы данных. А также при работе с SQL необходимо предпринимать дополнительные меры безопасности [8].

4 Современные тенденции в разработке баз данных

4.1 BigData и NoSQL базы данных

Стремительно растущий объем информации ставит перед нами новые сложные задачи по организации ее хранения и обработки. По оценкам IDC размер «цифровой вселенной» в 2006 году составлял 0.18 зеттабайт, а к 2011 году должен был достигнуть 1.8 зеттабайт, продемонстрировав десятикратный рост за 5 лет.

Закономерно появление термина «Big Data» или «большие данные» — это структурированные или неструктурированные массивы данных большого объема. Большие данные имеют четыре основные характеристики: объем, разнообразие, скорость и ценность.

Классическая реляционная архитектура не подходит для работы с большими данными. Попытки приспособить реляционную СУБД к работе с большими данными приводят к следующему:

1. Отказ от строгой согласованности.
2. Уход от нормализации и внедрение избыточности.
3. Потеря выразительности языка SQL и необходимость моделировать часть его функций программно.
4. Существенное усложнение клиентского программного обеспечения.
5. Сложность поддержания работоспособности и отказоустойчивости получившегося решения.

Производители реляционных СУБД осознают все эти проблемы и уже начали предлагать масштабируемые кластерные решения. Однако стоимость внедрения и сопровождения подобных решений зачастую не окупается.

Тогда почему бы не спроектировать архитектуру, способную адаптироваться к возрастающим объемам данных и эффективно их обрабатывать? Подобные мысли привели к появлению движения NoSQL.

NoSQL — это семейство баз данных, которые отличаются от традиционных реляционных баз данных (SQL) тем, что они не используют таблицы, схемы и SQL-язык запросов. Вместо этого они используют другие модели данных, такие как документы, графы или ключ-значение.

Среди самых популярных NoSQL баз данных можно выделить MongoDB, Cassandra, Redis и Couchbase. Они используются в различных областях, от веб-разработки и аналитики больших данных до хранения данных в облаке.

NoSQL не подразумевает бездумного отказа от всех принципов реляционной модели. Более того, термин «NoSQL» впервые был использован в 1998 году для описания реляционной базы данных, не использовавшей SQL.

Популярность NoSQL стал набирать в 2009 году, в связи с появлением большого количества веб-стартапов, для которых важнейшей задачей является поддержание постоянной высокой пропускной способности хранилища при неограниченном увеличении объема данных.

Рассмотрим основные особенности NoSQL подхода:

1. Исключение излишнего усложнения.
2. Высокая пропускная способность.
3. Неограниченное горизонтальное масштабирование.
4. Рост производительности за счет пренебрежения согласованностью данных.

Но все же, при выборе инструментария необходимо отталкиваться от задачи, поэтому нельзя сказать, что NoSQL может полностью заменить реляционную модель. Поэтому разработчик баз данных должен обладать достаточной компетентностью для выбора правильной технологии в том или ином случае [9].

4.2 Облачные базы данных

Облачные вычисления являются наиболее популярной технологией в последнее время, поэтому неудивительно, что их стали использовать и в проектировании баз данных.

Облачная база данных (Cloud Database) — это служба базы данных, созданная и доступная через облачную платформу. Она выполняет те же функции, что и традиционная база данных, с дополнительной гибкостью облачных вычислений. Пользователи устанавливают программное обеспечение в облачной инфраструктуре для реализации базы данных.

Существует две основные модели развертывания облачной базы данных:

1. Традиционная. Организация приобретает пространство виртуальной машины у поставщика облачных услуг, а база данных развертывается в облаке.
2. База данных как услуга (DBaaS). Организация заключает договор с поставщиком облачных услуг и база данных работает во внешней инфраструктуре. Модель DBaaS предоставляет организациям наибольшую ценность, позволяя им использовать внешнее управление базами данных, оптимизированное за счет автоматизации программного обеспечения, вместо того,

чтобы нанимать собственных специалистов по базам данных.

Новейший и наиболее инновационный тип облачной базы данных — это самоуправляемая (self-driving) облачная база данных. В то время как локальным базам данных требуется специальный администратор баз данных для управления ими, такие же глубокие знания не требуются для управления автономной облачной базой данных. Этот тип облачной базы данных использует машинное обучение для автоматизации настройки, обеспечения безопасности, резервного копирования, обновления и других рутинных задач управления, которые традиционно выполнялись администраторами баз данных.

Исследование IDC показывает, что до 75% общих затрат предприятия на управление данными могут быть связаны только с оплатой труда. Самостоятельная база данных потенциально может сэкономить среднему предприятию сотни или, возможно, тысячи рабочих часов сотрудников, занятых полный рабочий день, ежегодно для каждой из основных корпоративных баз данных. Кроме того, было подсчитано, что 72% корпоративных ИТ-бюджетов уходит на поддержку существующих систем, а на инновации остается лишь 25%.

Самостоятельные базы данных могут иметь большое значение для устранения этих высоких затрат и предоставления предприятиям возможности использовать своих администраторов баз данных для более важных задач, таких как моделирование данных, помощь программистам с архитектурой данных и планирование будущих мощностей [10].

ЗАКЛЮЧЕНИЕ

Профессия разработчика баз данных является ключевой для эффективного управления информацией в современном мире. Спрос на нее будет продолжать расти в ближайшие годы, поскольку количество данных продолжает увеличиваться, а компании все больше заинтересованы в экспертах по их обработке и анализу. Ведь даже самоуправляемая база данных нуждается в администрировании, но уже более квалифицированными специалистами.

Подводя итог, разработчик баз данных должен как минимум знать непосредственно теорию баз данных и свободно владеть языком запросов и управления SQL. Также на практике появится необходимость работать с большими данными, высоконагруженными системами, нереляционными базами данных, облачными технологиями и так далее.

По данным компании интернет-рекрутмента HeadHunter¹ на июнь 2023 года в Саратове есть 9 вакансий на эту должность со средней заработной платой 90 тыс.р./месяц, а в Москве — 1003 вакансии, с заработной платой от 60 до 600 тыс.р./месяц. При этом, если переформулировать запрос как «программист SQL», то количество вакансий вырастет в Саратове до 92, а в Москве — до 4959. Такой результат неудивителен, ведь умение работать с базами данных хотя бы на начальном уровне является базовым для IT-специалиста. А по данным интернет-портала Habr² владение языком SQL входит в топ-3 необходимых навыков для программиста.

Обобщая вышесказанное, данная специальность является одной из самых перспективных и высокооплачиваемых в IT-индустрии.

¹<https://hh.ru>

²<https://habr.com>

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Дейт, К. Д.* Введение в системы баз данных / К. Д. Дейт. — Россия, г. Москва, ул. Лесная, д. 43: Вильямс, 2008. — С. 1328.
- 2 *Гарсиа-Молина, Г.* Системы баз данных: Полный курс / Г. Гарсиа-Молина, Д. Д. Ульман, Д. Уидом. — Россия, г. Москва, ул. Лесная, д. 43: Вильямс, 2003. — С. 1088.
- 3 *Ульман, Д.* Введение в системы баз данных / Д. Ульман, Д. Уидом // *М.: Лори.* — 2000. — Т. 374. — С. 379.
- 4 *Роб, П.* Системы баз данных: проектирование, реализация и управление / П. Роб, К. Коронел. — Россия, г. Санкт-Петербург, ул. Гончарная, д. 20: БХВ-Петербург, 2004. — С. 1024.
- 5 *Кузнецов, С. Д.* Основы баз данных / С. Д. Кузнецов // *М.: Бином. Лаборатория знаний.* — 2007. — С. 484.
- 6 *Грофф, Д.* SQL: Полное руководство / Д. Грофф, П. Вайнберг, Э. Оппель. — Россия, г. Санкт-Петербург: Диалектика, 2019. — С. 960.
- 7 *Тарасов, С.* СУБД для программиста. Базы данных изнутри / С. Тарасов. — Россия, г. Москва, ул. Садовая Кудринская д. 11: Солон-Пресс, 2015. — С. 320.
- 8 *Celko, J.* Joe Celko's thinking in sets: auxiliary, temporal, and virtual tables in SQL / J. Celko. — USA, Waltham: Morgan Kaufmann, 2008. — P. 384.
- 9 *Клеменков, П.* Большие данные: современные подходы к хранению и обработке / П. Клеменков, С. Кузнецов // *Труды Института системного программирования РАН.* — 2012. — Т. 23. — С. 143–158.
- 10 *Pavlo, A.* Self-driving database management systems // *CIDR.* — Vol. 4. — USA, Pennsylvania: Carnegie Mellon University, 2017. — P. 12.