МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| к.э.н., доцент |  |  |  |  | Т.Н.Елина |
| должность, уч. степень, звание |  | подпись, дата |  |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4 |
| **Построение ER- диаграммы физического уровня** по дисциплине: Базы данных |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 3843 |  |  |  | А.П. Конева |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2021

1. **Задача**

Построить ER-диаграмму физического уровня.

1. **Цель работы**

Провести и обосновать выбор СУБД. На основе выбранной СУБД спроектировать физическую модель данных. Описать отличия физической от логической модели.

1. **Ход работы**

Для выбора СУБД рассмотрим и сравним три наиболее популярные реляционные системы управления базами данных: [SQLite](https://www.sqlite.org/), [MySQL](https://www.mysql.com/) и [PostgreSQL](https://www.postgresql.org/).

**SQLite**

SQLite — это мощная СУБД, встраиваемая в приложение, которое её использует. Будучи файловой БД, она предоставляет отличный набор инструментов для более простой (в сравнении с серверными БД) обработки любых видов данных.

Когда приложение использует SQLite, их связь производится с помощью функциональных и прямых вызовов файлов, содержащих данные (например, баз данных SQLite), а не какого-то интерфейса, что повышает скорость и производительность операций.

***Преимущества***

* **Файловая:**вся база данных хранится в одном файле, что облегчает перемещение.
* **Стандартизированная:**SQLite использует SQL; некоторые функции опущены (RIGHT OUTER JOIN или FOR EACH STATEMENT), однако, есть и некоторые новые.
* **Отлично подходит для разработки и даже тестирования:**во время этапа разработки большинству требуется масштабируемое решение. SQLite, со своим богатым набором функций, может предоставить более чем достаточный функционал, при этом будучи достаточно простой для работы с одним файлом.

***Недостатки***

* **Отсутствие пользовательского управления:**продвинутые БД предоставляют пользователям возможность управлять связями в таблицах в соответствии с привилегиями, но у SQLite такой функции нет.
* **Невозможность дополнительной настройки:**SQLite нельзя сделать более производительной, изменив параметры в настройках.

***Когда стоит использовать SQLite***

* **Встроенные приложения:**все портируемые, не предназначенные для масштабирования приложения — например, локальные однопользовательские приложения, мобильные приложения или игры.
* **Система доступа к дисковой памяти:**в большинстве случаев приложения, часто производящие прямые операции чтения/записи на диск, можно перевести на SQLite для повышения производительности.
* **Тестирование:**отлично подойдёт для большинства приложений, частью функционала которых является тестирование бизнес-логики.

***Когда не стоит использовать SQLite***

* **Многопользовательские приложения:**если вы работаете над приложением, доступом к БД в котором будут одновременно пользоваться несколько человек, лучше выбрать полнофункциональную СУБД — например, MySQL.
* **Приложения, записывающие большие объёмы данных:**одним из ограничений SQLite являются операции записи. Эта СУБД допускает единовременное исполнение лишь одной операции записи.

**MySQL**

MySQL — это самая популярная из всех крупных серверных БД. Хотя MySQL и не пытается полностью реализовать SQL-стандарты, она предлагает широкий функционал.

***Преимущества***

* **Простота:**MySQL легко устанавливается. Существует много сторонних инструментов, включая визуальные, облегчающих начало работы с БД. MySQL прост и удобен в использовании. Вы можете создавать MySQL и взаимодействовать с ним, обладая только базовыми знаниями MySQL и несколькими простыми операторами SQL. В MySQL используется стандартизированный язык - благодаря документации и многолетнему опыту создания, он предоставляет единую платформу во всем мире для всех своих пользователей.
* **Много функций:**MySQL поддерживает большую часть функционала SQL. В MySQL реализован поиск по фразам, словам, есть возможность создавать ключевые индексы.
* **Безопасность:** MySQL достаточно безопасен, поскольку он состоит из надежного уровня безопасности данных для защиты конфиденциальных данных от злоумышленников, а пароли в MySQL зашифрованы.
* **Мощность и масштабируемость:** MySQL масштабируем и способен обрабатывать более 50 миллионов строк.
* MySQL имеет архитектуру клиент-сервер. Может быть любое количество клиентов или прикладных программ, которые обмениваются данными с сервером базы данных (MySQL) для запроса данных, сохранения изменений и т. д.
* **Скорость:**пренебрежение некоторыми стандартами позволяет MySQL работать производительнее. MySQL имеет уникальную архитектуру механизма хранения, которая делает его более быстрым, дешевым и надежным. MySQL дает разработчикам более высокую производительность за счет использования представлений, триггеров и хранимых процедур. большой объем данных извлекается быстро и эффективно. Такие операции, как вставка, удаление и манипулирование данными, также выполняются практически мгновенно.
* MySQL совместим с большинством операционных систем, включая Windows, Linux.

***Недостатки***

* **Известные ограничения:**присутствуют определённые ограничения функциональности.
* **Застой в разработке:**хотя MySQL и является open-source продуктом, работа над ней сильно заторможена.

***Когда стоит использовать MySQL***

* **Распределённые операции:**когда вам нужен функционал больший, чем может предоставить SQLite, стоит использовать MySQL.
* **Высокая безопасность:**функции безопасности MySQL предоставляют надёжную защиту доступа и использования данных.
* **Веб-сайты и приложения:**большая часть веб-ресурсов вполне может работать с MySQL, несмотря на ограничения. Этот инструмент весьма гибок и прост в обращении, что только на руку в длительной перспективе.
* **Кастомные решения:**если вы работаете над очень специфичным продуктом, MySQL подстроится под ваши потребности благодаря широкому спектру настроек и режимов работы.

***Когда не стоит использовать MySQL***

* **SQL-совместимость:**поскольку MySQL не пытается полностью реализовать стандарты SQL, она не является полностью совместимой с SQL. Из-за этого могут возникнуть проблемы при интеграции с другими СУБД.
* **Конкурентность:**хотя MySQL неплохо справляется с операциями чтения, одновременные операции чтения-записи могут вызвать проблемы.

**PostgreSQL**

PostgreSQL — это самая продвинутая РСУБД, ориентирующаяся в первую очередь на полное соответствие стандартам и расширяемость. PostgreSQL, или Postgres, пытается полностью соответствовать SQL-стандартам ANSI/ISO. PostgreSQL отличается от других СУБД тем, что обладает объектно-ориентированным функционалом, в том числе полной поддержкой концепта ACID (Atomicity, Consistency, Isolation, Durability). Будучи основанным на мощной технологии, Postgres отлично справляется с одновременной обработкой нескольких заданий. Поддержка конкурентности реализована с использованием MVCC (Multiversion Concurrency Control), что также обеспечивает совместимость с ACID. Хотя эта СУБД не так популярна, как MySQL, существует много сторонних инструментов и библиотек для облегчения работы с PostgreSQL.

***Преимущества***

* **Полная SQL-совместимость**.
* **Сообщество:**PostgreSQL поддерживается опытным сообществом 24/7.
* **Поддержка сторонними организациями:**несмотря на очень продвинутые функции, PostgreSQL используется во многих инструментах, связанных с СУБД.
* **Расширяемость:** PostgreSQL можно программно расширить за счёт хранимых процедур.
* **Объектно-ориентированность:**PostgreSQL — не только реляционная, но и объектно-ориентированная СУБД.

***Недостатки***

* **Производительность:**в простых операциях чтения PostgreSQL может уступать своим соперникам.
* **Популярность:**из-за своей сложности инструмент не очень популярен.

***Когда стоит использовать PostgreSQL***

* **Целостность данных:**если приоритет стоит на надёжность и целостность данных, PostgreSQL — лучший выбор.
* **Сложные процедуры:**если ваша БД должна выполнять сложные процедуры, стоит выбрать PostgreSQL в силу её расширяемости.
* **Интеграция:**если в будущем вам предстоит перемещать всю базу на другое решение, меньше всего проблем возникнет с PostgreSQL.

***Когда не стоит использовать PostgreSQL***

* **Скорость:**если всё, что нужно — это быстрые операции чтения, не стоит использовать PostgreSQL.
* **Простые ситуации:**если не требуется повышенная надёжность, поддержка ACID.

В силу простоты использования, достаточной безопасности и разнообразного функционала для выполнения данной лабораторной работы была выбрана бесплатная СУБД с открытым исходным кодом MySQL.

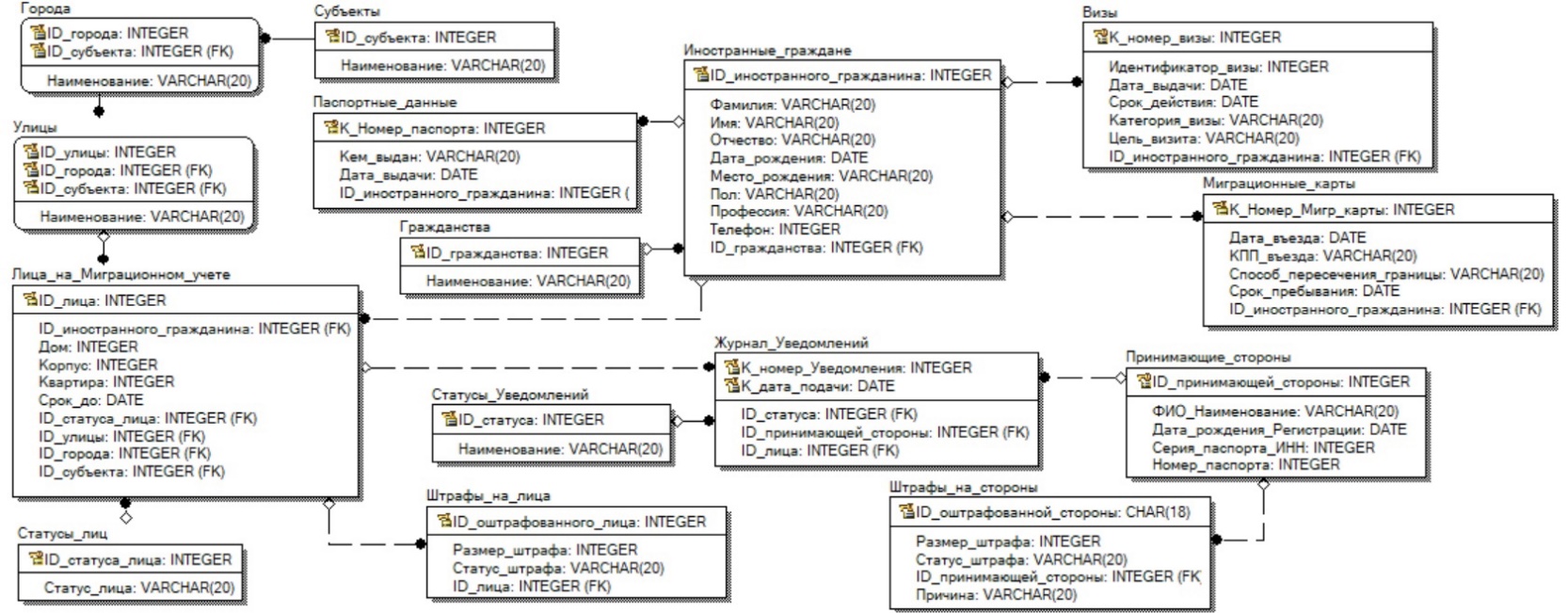
С учетом особенностей выбранной СУБД разработали ER-диаграмму физического уровня, которая представлена на Рис.1.

Рис.1 ER- диаграмма физического уровня

Необходимо отметить следующие отличия логической модели от физической:

* + Физическая модель данных зависит от используемой СУБД, в отличие от логической модели данных, которая является универсальной. Одной и той же логической модели могут соответствовать несколько разных физических моделей.
  + Логическая модель данных включает сущности, атрибуты, отношения и ключи. Не имеет значения, какой конкретно тип данных имеет атрибут. Физическая модель данных включает таблицы, столбцы, типы и формат данных, ограничения первичного и внешнего ключей, триггеры и хранимые процедуры.
  + В логической модели данных для сущностей и атрибутов используются длинные неформальные имена. Однако в физических данных для имен таблиц и столбцов используются сокращенные формальные имена, которые имеют некоторые ограничения.
  + Логическая модель данных сначала выводится из описания. После этого выводится физическая модель данных.
* Логическая модель данных нормализована обычно до 3НФ. При необходимости физическая модель базы данных будет деформализована для соответствия требованиям.
* Логическая модель может определять отношения «многие ко многим», в то время как физические потребности используют ассоциативные таблицы.
* Значения также могут быть реализованы в виде отдельной таблицы словарей (иногда общей таблицы для многих словарей), при этом логическая модель не должна указывать такие детали.
* Физическая модель часто имеет системные столбцы или столбцы метаданных (например, *created\_by* , *modified\_date* , *deleted\_flag* ), которые логической не требуются.

**Вывод**

Проанализировали существующие СУБД и после изучения их недостатков и преимуществ выбрали MySQL. Основными преимуществами этой СУБД являются: безопасность, простота и удобство в использовании в связи с применением только базовых знаний MySQL и нескольких простых операторов SQL.

На основе выбранной СУБД MySQL спроектировали ER-диаграмму физического уровня, изображенную на Рис.1.

Среди отличий физической от логической модели, описанных выше, наиболее значимыми являются следующие: 1) Логическая модель данных включает сущности, атрибуты, отношения и ключи, а физическая модель данных включает таблицы, столбцы, типы и формат данных, ограничения первичного и внешнего ключей, триггеры и хранимые процедуры. 2) В логической модели данных для сущностей и атрибутов используются длинные неформальные имена, однако в физических данных для имен таблиц и столбцов используются сокращенные формальные имена, которые имеют некоторые ограничения. 3) Логическая модель данных нормализована обычно до 3НФ, но при необходимости физическая модель базы данных будет деформализована для соответствия требованиям. 4) Логическая модель может определять отношения «многие ко многим», в то время как физические потребности используют ассоциативные таблицы.

**Список литературы:**

1. MySQL Advantages and Disadvantages – Режим доступа: <https://www.w3spoint.com/mysql-advantages-disadvantages>
2. Проектирование реляционных баз данных: методические указания – Режим доступа: <https://www.spbgasu.ru/documents/docs_317.pdf>
3. О прибытии иностранного гражданина – Режим доступа: <https://support.kontur.ru/pages/viewpage.action?pageId=4589781>
4. Logical vs Physical Data Dictionary – Режим доступа: <https://dataedo.com/blog/logical-vs-physical-data-dictionary>
5. Data Modelling: Conceptual, Logical, Physical Data Model Types – Режим доступа: <https://www.guru99.com/data-modelling-conceptual-logical.html>
6. Физическая и логическая модели данных – Режим доступа: <https://helpiks.org/8-15367.html>
7. Разница между логической и физической моделью данных – Режим доступа: <https://ru.strephonsays.com/logical-and-vs-physical-data-model-11564>