

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»**

Институт компьютерных наук и кибербезопасности
Высшая школа технологий искусственного интеллекта
Направление 02.03.01 Математика и Компьютерные науки

КУРСОВАЯ РАБОТА
по дисциплине «Управление знаниями и технологии баз данных»
**Разработка и программирование базы
метаданных**

Студент: _____

Сергиенко Кирилл Александрович

Преподаватель: _____

Попов Сергей Геннадьевич

«_____» _____ 20__ г.

РЕФЕРАТ

На 1 с., 1 рисунков, 1 приложения.

КЛЮЧЕВЫЕ СЛОВА: Метаданные, базы данных, СУБД, MySQL, клиент-серверная архитектура, веб-приложение.

Тема курсовой работы: «Разработка и программирование базы метаданных»

ABSTRACT

1 pages, 1 figures, 1 appendices

KEYWORDS: Metadata, databases, MSDB, MySQL, client-server architecture, web application.

Title of the thesis: "Development and programming of a metadata database."

Содержание

ВВЕДЕНИЕ	4
ПОСТАНОВКА ЗАДАЧИ	5
1 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ	6
2 ХРАНЕНИЕ МЕТАДАННЫХ	8
2.1 База метаданных	8
2.2 Заполнение базы метаданных	10
2.3 Примеры исходных данных	11
3 ПОЛЬЗОВАТЕЛЬСКИЙ ВЕБ-ИНТЕРФЕЙС	15
Список использованных источников	15

ВВЕДЕНИЕ

В эпоху взрывного роста количества и разнообразия информации критически важным становится вопрос грамотного управления данными о структуре баз данных. Метаданные — то есть информация, описывающая другие данные, — выступают ключевым инструментом для обеспечения прозрачности процессов, контроля за ресурсами и их эффективного использования. Для современных информационных систем необходимы инструменты, позволяющие автоматизировать сбор, хранение и анализ таких сведений, что напрямую влияет на качество администрирования данных и обоснованность принимаемых решений.

Целью данной курсовой работы является проектирование и программная реализация специализированной базы метаданных. Разрабатываемая система будет предназначена для автоматического извлечения, каталогизации и управления метаданными из различных источников. Итогом работы станет универсальное программное решение, умеющее взаимодействовать с множеством баз данных MySQL, независимо от сложности и особенностей их схем.

ПОСТАНОВКА ЗАДАЧИ

Цель курсовой работы — разработать масштабируемое программное обеспечение для автоматического извлечения, хранения и управления метаданными баз данных с поддержкой пользовательским псевдонимов для обращения к хранимым базам.

Задачи:

1. Предусмотреть возможность подключения и работы с несколькими базами данных MySQL, содержащими произвольные структуры таблиц и данных.
2. Разработать и реализовать схему базы данных для хранения метаданных, включающую информацию о:
 - базах данных,
 - таблицах,
 - сущностях,
 - ключах.
3. Реализовать механизм автоматизированной выгрузки метаданных из подключаемых баз данных и сохранения их в разработанную базу.
4. Реализовать интерфейс, позволяющий пользователю задавать псевдонимы для баз данных и их таблиц, а также выполнять запросы вида `SELECT FROM WHERE` с использованием данных псевдонимов.

1 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

Схема взаимодействия компонент приложения представлена на рисунке 1.1. Архитектура программного решения включает следующие основные подсистемы: внешние базы данных (Outer DBs), внутренняя база данных хранения метаданных (Metadata DB), серверное приложение и клиентское веб-приложение.

1. Внешние базы данных (Outer DBs) — произвольный набор реляционных баз данных MySQL, подлежащих анализу структуры. Серверное приложение устанавливает соединение с указанными источниками с использованием стандартного протокола SQL через библиотеку `mysql.connector`. Основная функция взаимодействия — получение информации о количестве записей в базе данных.

2. Metadata DB — реляционная база MySQL, предназначенная для хранения метаданных;

Функции:

- сохранение извлечённых метаданных;
- хранение псевдонимов;
- предоставление данных серверу.

3. Серверное приложение реализовано на Python с использованием фреймворков FastAPI (веб-приложение) и Uvicorn (ASGI-сервер). Функции:

- управление подключениями к внешним MySQL-базам данных;
- выгрузка метаданных и запись результатов в Metadata DB;
- чтение метаданных из Metadata DB;
- логика присвоения псевдонимов внешним базам данных и их таблицам;
- обработка HTTP-запросов клиентского приложения и генерация ответов в формате JSON.

4. Клиентское веб-приложение разработано на JavaScript с использованием React. Взаимодействие с сервером организовано через библиотеку `axios` по протоколу HTTP. Функции:

- отправка запросов на получение метаданных и статистической информации;

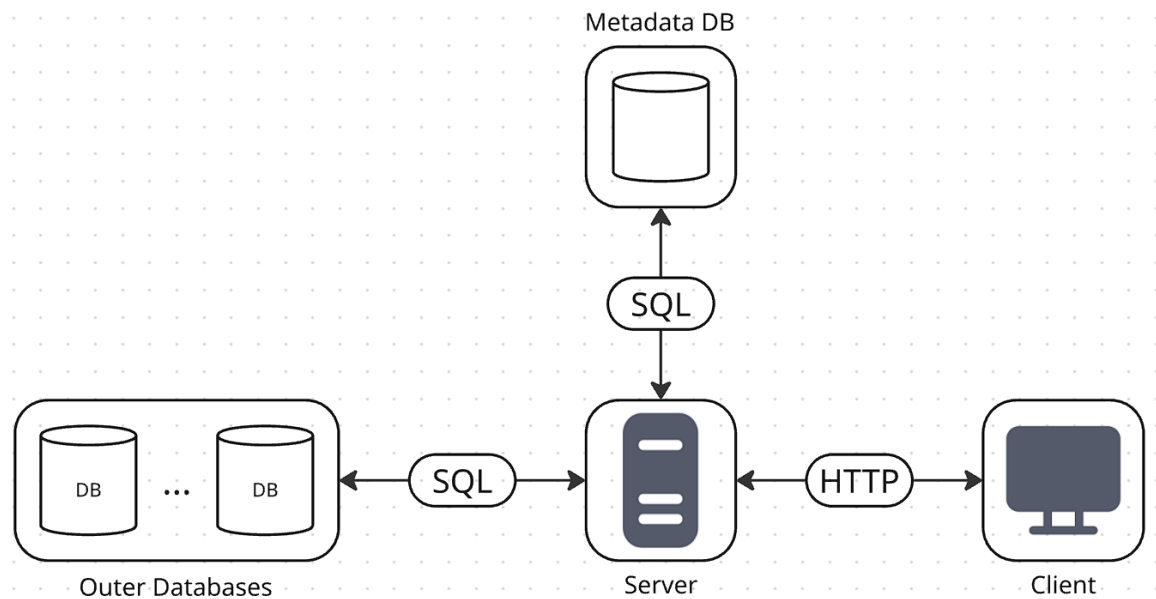


Рис. 1.1: Функциональная схема взаимодействия компонент приложения.

- визуализация результатов и предоставление интерфейса пользователю;
- инициирование операций по добавлению новых баз данных в базу метаданных;
- инициирование операций по добавлению псевдонимов в базу метаданных и удалению из нее;
- инициирование запросов вида `SELECT FROM WHERE` с учетом возможности использования псевдонимов.

2 ХРАНЕНИЕ МЕТАДААННЫХ

2.1 База метаданных

Эффективная работа с информацией в компьютерных системах — от хранения и обработки до обмена и пользовательского доступа — требует четкого описания свойств данных. Эти описания критически важны как для корректной работы программного обеспечения, так и для людей, которым необходимо понимать структуру информации, анализировать ее, строить поисковые запросы и оценивать пригодность данных для решения конкретных задач. Подобные описания называются метаданными и представляют собой отдельную, значимую категорию информационных ресурсов.

На рисунке 2.1 изображена структурная схема хранилища метаданных. Оно предназначено для хранения метаданных баз данных, таблиц, колонок и ключей. Составные ключи хранятся через связи с колонками соответствующей таблицы, где указывается позиция каждой колонки в ключе. Внешние ключи реализованы через связи с ключами референтной таблицы. Хранение псевдонимов реализовано посредством соответствующих атрибутов таблиц.

На основе СУБД MySQL запрограммирована база данных `metadata`, соответствующая описанной выше схеме. SQL-скрипт создания описанной схемы приведен в Приложении А.

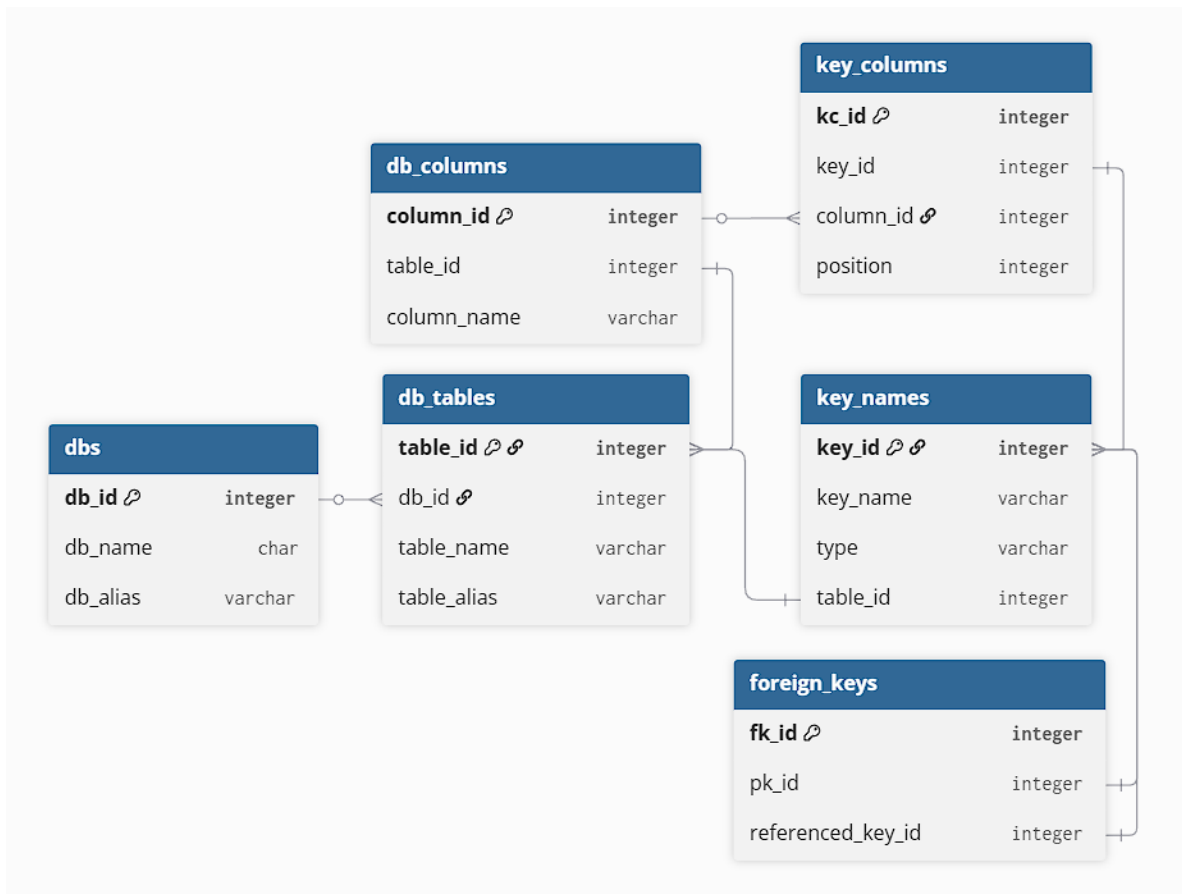


Рис. 2.1: Структурная схема хранилища метаданных.

2.2 Заполнение базы метаданных

Метаданные о хранимых базах данных получаются путем выполнения запросов к системной базе INFORMATION_SCHEMA. Эта база данных предоставляет доступ к служебной информации о структуре базы данных и параметрах сервера MySQL — таким как названия баз и таблиц, типы данных столбцов и права доступа пользователей. В технической литературе для обозначения этой информации также используются термины «словарь данных» и «системный каталог»

В качестве примера на листинге 2.1 показан запрос к INFORMATION_SCHEMA, который возвращает список таблиц базы данных hotel_management.

```
1 SELECT TABLE_NAME
2 FROM information_schema.TABLES
3 WHERE TABLE_SCHEMA = "hotel_management";
```

Листинг 2.1: Пример выполнения запроса для получения списка таблиц базы данных.

В листинге 2.2 продемонстрировано получение списка колонок базы данных hotel_management.

```
1 SELECT TABLE_NAME, COLUMN_NAME
2 FROM information_schema.COLUMNS
3 WHERE TABLE_SCHEMA = "hotel_management";
4 ORDER BY TABLE_NAME
```

Листинг 2.2: Пример выполнения запроса для получения списка колонок базы данных.

В листинге 2.3 продемонстрировано получение списка ключей базы данных hotel_management.

```
1 SELECT TABLE_NAME, CONSTRAINT_NAME, CONSTRAINT_TYPE
2 FROM information_schema.TABLE_CONSTRAINTS
3 WHERE TABLE_SCHEMA = "hotel_management";
```

Листинг 2.3: Пример выполнения запроса для получения списка ключей базы данных.

В листинге 2.4 продемонстрировано получение списка колонок в ключах базы данных hotel_management.

```
1 SELECT CONSTRAINT_NAME, TABLE_NAME, COLUMN_NAME,
2     ORDINAL_POSITION
3 FROM information_schema.KEY_COLUMN_USAGE
4 WHERE TABLE_SCHEMA = "hotel_management"
5 ORDER BY CONSTRAINT_NAME, ORDINAL_POSITION;
```

Листинг 2.4: Пример выполнения запроса для получения списка колонок в ключах базы данных.

В листинге 2.5 продемонстрировано получение связей внешних и первичных ключей базы данных hotel_management.

```
1 SELECT
2 rc.CONSTRAINT_NAME AS fk_name,
```

```

3      rc.CONSTRAINT_SCHEMA AS fk_schema,
4      kcu.TABLE_NAME AS fk_table_name,
5      kcu.REFERENCED_TABLE_NAME AS referenced_table_name
6  FROM information_schema.REFERENTIAL_CONSTRAINTS rc
7  JOIN information_schema.KEY_COLUMN_USAGE kcu
8  ON rc.CONSTRAINT_NAME = kcu.CONSTRAINT_NAME
9  AND rc.CONSTRAINT_SCHEMA = kcu.CONSTRAINT_SCHEMA
10 WHERE rc.CONSTRAINT_SCHEMA = "hotel_management"
11 GROUP BY rc.CONSTRAINT_NAME, kcu.TABLE_NAME, kcu.
      REFERENCED_TABLE_NAME

```

Листинг 2.5: Пример выполнения запроса для получения связей внешних и первичных ключей базы данных.

Полный код на языке Python модуля `metadata_service` заполнения базы метаданных представлен в приложении

2.3 Примеры исходных данных

Для тестирования и демонстрации работы всей системы используются базы данных ниже:

1. `composite_key_example` — синтетический пример для демонстрации хранения составных ключей, содержащий 4 таблицы. Схема базы данных представлена на рисунке 2.2;
2. `event_agency` — база данных агентства по управлению мероприятиями, содержащая 11 таблиц. Схема базы данных представлена на рисунке 2.3;
3. `hotel_management` — база данных управления отелями, содержащая 18 таблиц. Схема базы данных представлена на рисунке 2.4;

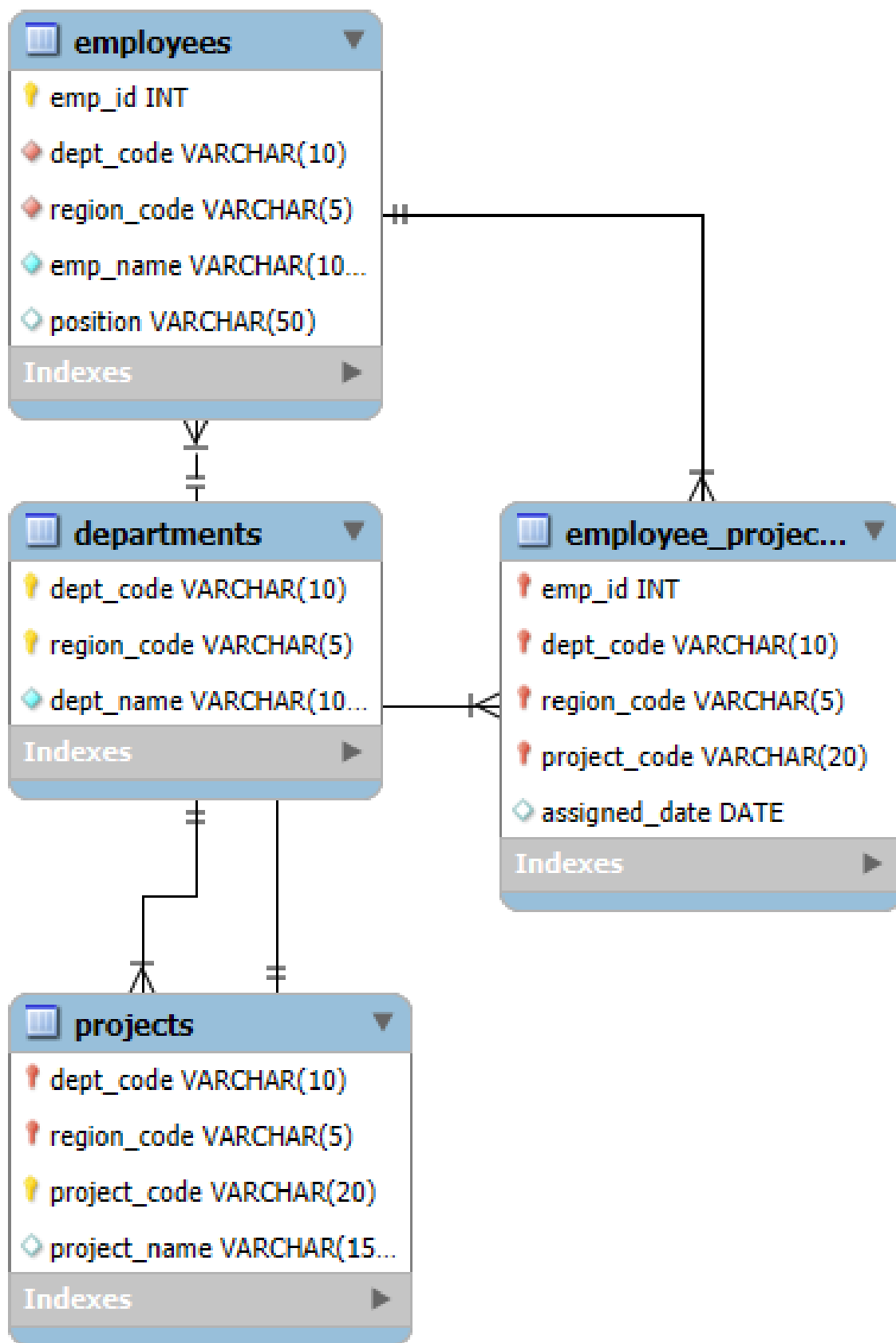


Рис. 2.2: Схема базы данных composite_key_example.

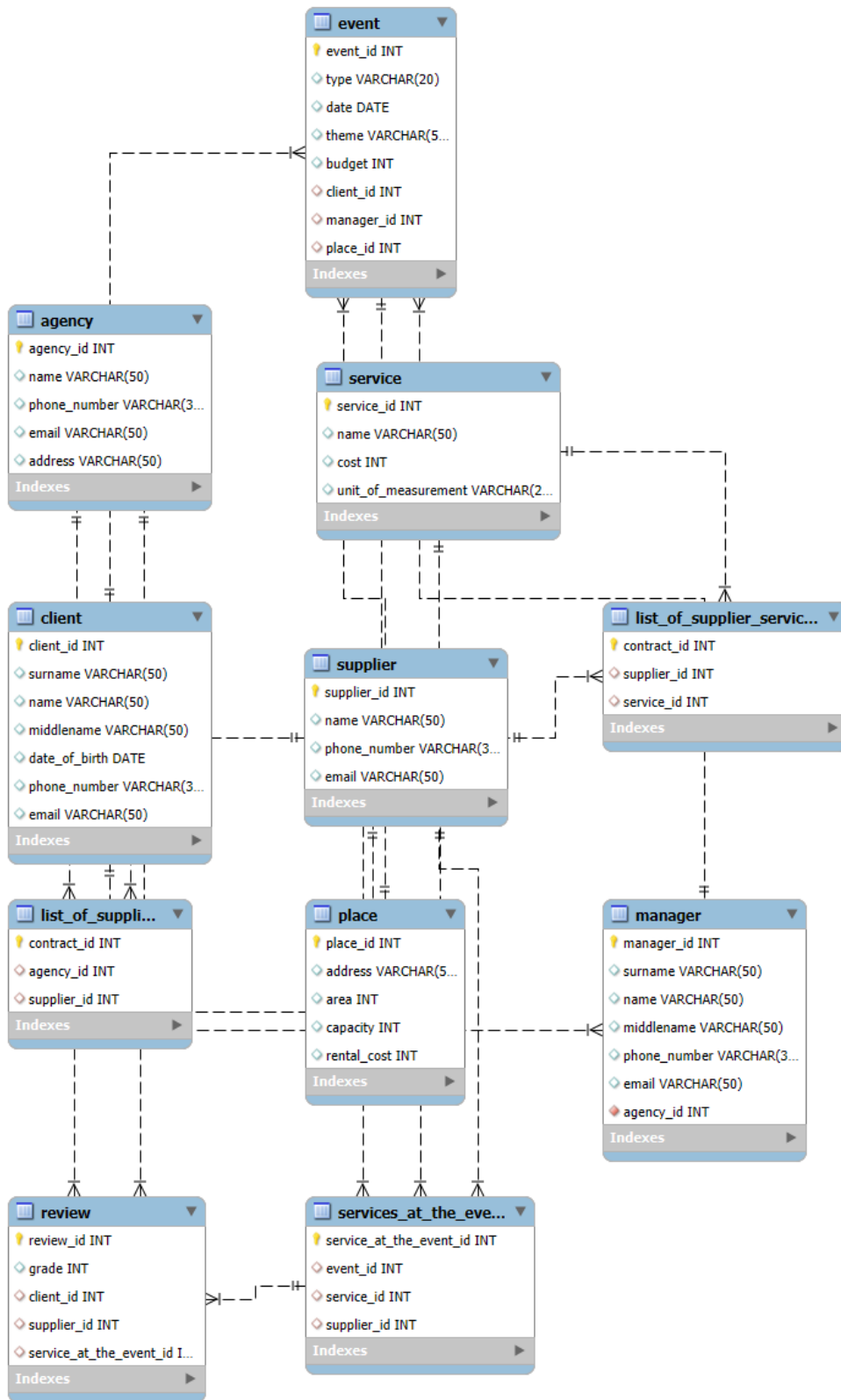
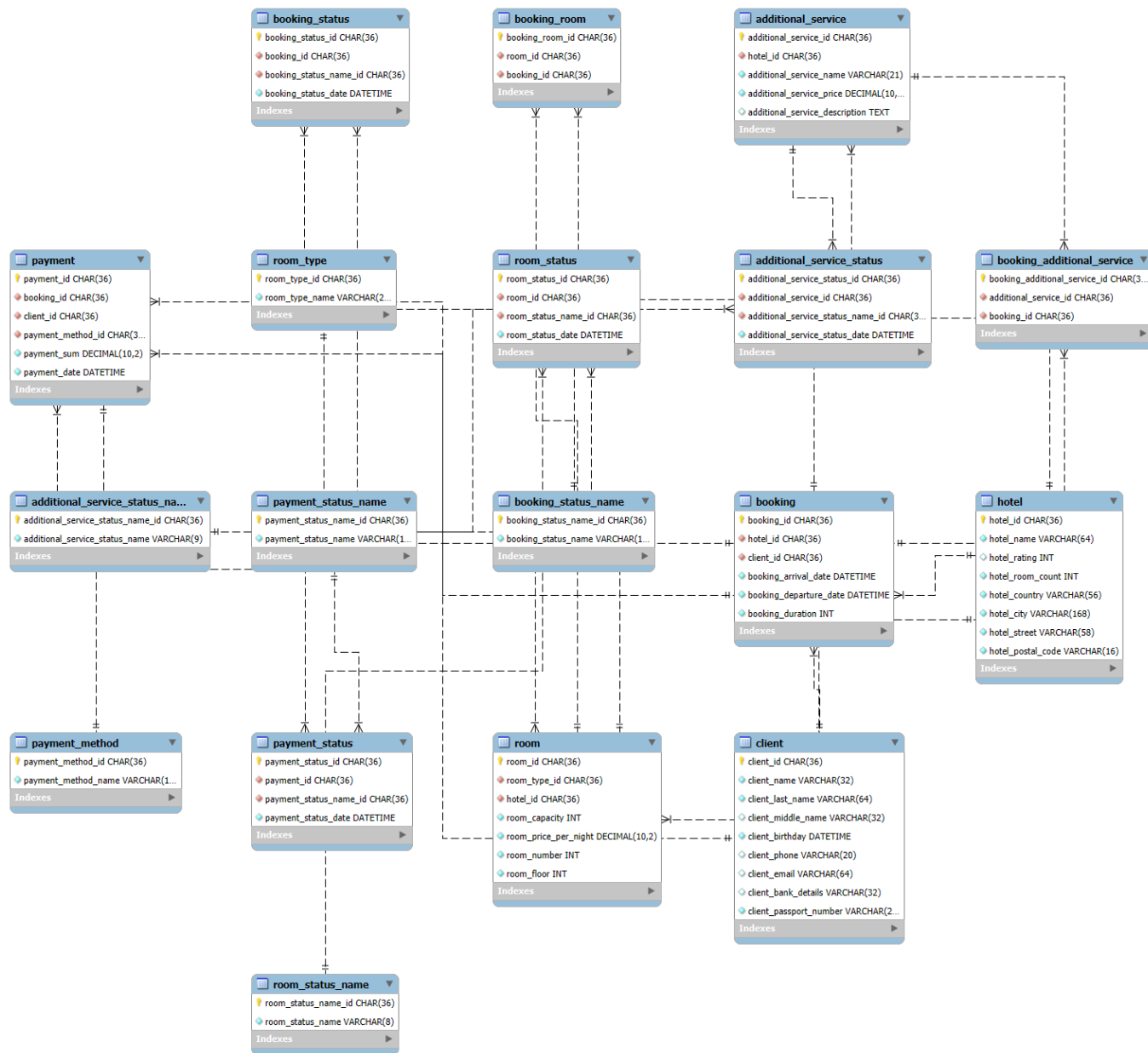


Рис. 2.3: Схема базы данных event_agency.

Рис. 2.4: Схема базы данных `hotel_management`.