УДК 00.004.4’2

РАЗРАБОТКА БАЗЫ ДАННЫХ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ «ОНЛАЙН-СТОЛОВКА»

Слинько И.А., Абдулвелеева Р.Р.
Новотроицкий филиал НИТУ «МИСИС», г. Новотроицк

Аннотация. В данной статье разработана база данных для мобильного приложения «Онлайн-Столовка», позволяющего осуществлять онлайн-заказы в столовой НФ НИТУ «МИСиС». Также в статье рассмотрена ER-диаграмма этой базы данных и подробно описана её структура.

Ключевые слова: разработка, база данных, столовая, мобильное приложение, заказ еды, ER-диаграмма, PostgreSQL.

Современный мир, насыщенный постоянно развивающимися инновационными технологиями, подвергается неустанному влиянию перемен. Благодаря научно-техническому прогрессу, мобильные устройства, которыми владеет сейчас практически каждый, становятся неотъемлемой частью этой технологической эволюции, превращаясь из простых коммуникационных средств в универсальных помощников в повседневных задачах.

Сегодня использование смартфонов выходит далеко за рамки совершения звонков и отправки сообщений. Одной из значимых функций, которую предоставляют современные мобильные приложения, является возможность заказать еду из различных заведений. В быту такие приложения имеют довольно весомое значение, т.к. предоставляют удобство и возможность выбора из многообразия блюд.

Большинство людей предпочитает дистанционный способ заказа еды, учитывая его широкое распространение, поскольку он обеспечивает быстроту оформления заказа и избавляет от необходимости стоять в очереди, а также предлагает бонусы и акции постоянным клиентам. Ряд известных ресторанов быстрого питания, таких как «KFC», «Burger King», «Додо Пицца» и другие, успешно интегрировали в свой бизнес мобильные приложения.

В столовой НФ НИТУ «МИСИС» отсутствует система дистанционного заказа питания, что вкупе с одновременным выполнением кассиром функций накладывания пищи и оформления заказов приводит к формированию долгих очередей во время единственного обеденного перерыва. Для решения этой проблемы было предложено разработать мобильное приложение «Онлайн-Столовка», которое позволит сотрудникам и студентам ВУЗа заказывать блюда заранее, оплачивать их онлайн и затем получать уже готовый заказ в столовой.

В центре внимания настоящей статьи находится процесс разработки базы данных, который является одним из основных этапов создания мобильного приложения.

Первым делом необходимо тщательно спроектировать её структуру. В это входит создание ER-диаграммы (*entity-relationship diagram*), которая визуально представляет устройство базы данных и отображает все её компоненты и их взаимосвязи. Построить ERD можно и на бумаге, однако современные программные средства будут более удобным и функциональным вариантом для этого.

В CASE-системе Visual Paradigm можно создавать ER-диаграмму (а также множество других диаграмм), и автоматически конвертировать её в SQL-код, что значительно упрощает процесс разработки базы данных. В качестве СУБД была выбрана PostgreSQL, так как она предоставляет широкие возможности для масштабирования и управления данными.

Рассмотрим основные аспекты ER-диаграммы, представленной на рисунке 1.

1. Сущности (*entities*) – это реальные или представляемые объекты, информация о которых должна сохраняться в базе данных и быть доступной.

2. Отношения или связи (*relationships*) – это графически изображаемые ассоциации, устанавливаемые между двумя типами сущностей.

3. Атрибуты (*attributes*) – это характеристики сущностей, которые нужны для уточнения, идентификации, классификации или выражения их состояния.

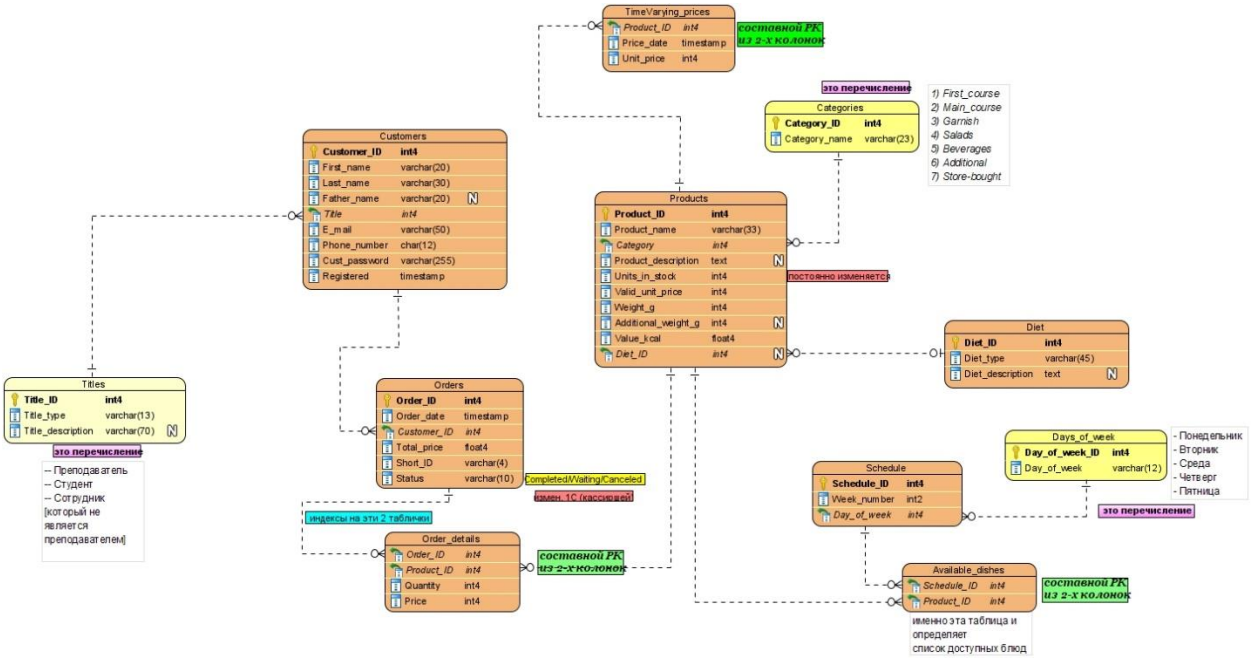

Рис. 1 – ER-диаграмма базы данных моб. приложения «Онлайн-Столовка»

1. Таблица **Customers** (*заказчики*) хранит информацию о зарегистрированных клиентах с уникальным идентификатором Customer_ID. Она связана с таблицей Orders по полю Customer_ID, позволяя отслеживать заказы клиента.

– Customer_ID: int, Primary Key - уникальный ID заказчика;

– First_name: varchar(20) - имя заказчика;

– Last_name: varchar(30) - фамилия заказчика;

– Father_name: varchar(20) - отчество заказчика (если есть);

– Title: Titles – «звание» заказчика (студент, преподаватель или сотрудник);

– E_mail: varchar(50) - электронная почта заказчика;

– Phone_number: char(12) - номер телефона заказчика;

– Cust_password: varchar(255) - пароль заказчика;

– Registered: timestamp - дата регистрации заказчика.

2. Таблица **Products** (*блюда*) хранит информацию о всех блюдах, имеющихся в меню столовой, предоставляя Product_ID в качестве уникального идентификатора блюда. Она связана с таблицей Order_details по полю Product_ID, определяя количество и цену каждого блюда в заказе.

– Product_ID: int, Primary Key - уникальный ID блюда;

– Product_name: varchar(60) - наименование блюда;

– Category: Categories - категория блюда (1-ое, 2-ое, гарнир, салат и т.д.);

– Product_description: text - описание/состав блюда (если есть);

– Units_in_stock: int - единиц товара в наличии;

– Valid_unit_price: int - действующая цена за единицу товара;

– Weight_g: int - вес (г);

– Additional_weight_g: int - дополнительный вес (г) (если есть);

– Value_kcal: float4 - пищевая энергетическая ценность (ккал);

– Diet_ID: int, Foreign Key - ID диеты из таблицы Diet.

3. Таблица **Diets** (*диеты*) содержит информацию о различных диетах, предоставляя Diet_ID в качестве уникального идентификатора диеты. Она связана с таблицей Products по полю Diet_ID, позволяя привязывать блюда к определённым диетам.

– Diet_ID: int, Primary Key - уникальный ID диеты;

– Diet_type: varchar(45) - номенклатура стандартной диеты;

– Diet_description: text - описание/состав диеты (если есть).

На данный момент эта таблица остается не задействованной, однако планируется ее активное внедрение в будущем.

4. Таблица **TimeVarying_prices** (*цены, изменяющиеся со временем*) содержит информацию о ценах на все блюда, а также

информацию об изменении цен на все блюда с течением времени. Она связана с таблицей Products по полю Product_ID, позволяя обновлять в последней актуальные цены на продукты с помощью функции update_valid_unit_price().

– Product_ID: int, Foreign Key - ID блюда из таблицы Products;

– Price_date: timestamp - дата установления цены на блюдо;

– Unit_price: float8 - цена за единицу товара.

Primary Key в этой таблице композитный – он состоит из пары полей: Product_ID и Price_date. Необходимость в композитном ключе обусловлена тем, что значения поля Product_ID не могут обеспечить уникальность записей в таблице, поскольку один и тот же продукт может присутствовать в ней неоднократно. Схожим образом, поле Price_date также не обладает уникальностью, поскольку допускается установка цены для нескольких продуктов в одинаковый временной интервал.

5. Таблица **Schedule** (*расписание*) представляет собой расписание.

– Schedule_ID: int, Primary Key - уникальный ID дня в расписании;

– Week_number: int - номер недели (1 или 2);

– Day_of_week: Days_of_week - день недели (с понедельника по пятницу).

В этой таблице содержится всего 10 записей, отражая две недели: четную и нечетную, следующие друг за другом. Каждая из этих недель включает 5 рабочих дней, исключая выходные (субботу и воскресенье), когда столовая не работает. Следовательно, первые 5 записей соответствуют дням первой недели, а последующие 5 – дням второй недели.

6. Таблица **Available_dishes** (*актуальные блюда*) представляет собой список доступных блюд на основе действующего расписания.

Она связана с таблицей Products по полю Product_ID.

– Schedule_ID: int, Foreign Key - ID дня в расписании из таблицы Schedule;

– Product_ID: int, Foreign Key - ID блюда из таблицы Products.

Primary Key в этой таблице композитный – он состоит из пары полей: Schedule_ID и Product_ID. Это обусловлено тем, что Schedule_ID сам по себе не является уникальным, так как в один день добавляется множество блюд. Product_ID сам по себе также может может дублироваться, учитывая наличие двух недель, где могут встречаться одни и те же блюда (например, в понедельник 1-ой недели и среду 2-ой недели).

7. Таблица **Orders** (*заказы*) хранит информацию о заказах с уникальным идентификатором Order_ID.

Она связана с таблицей Order_details по полю Order_ID, позволяя эффективно ассоциировать записи последней с соответствующими заказами и устанавливая, какие блюда были заказаны в каждом конкретном заказе.

– Order_ID: int, Primary Key - уникальный ID заказа;

– Order_date: timestamp - дата совершения заказа;

– Customer_ID: int, Foreign Key - ID заказчика из таблицы Customers;

– Total_price: float8 - итоговая цена заказа (руб.);

– Short_ID: varchar(4) - короткий временный ID, генерируемый системой автоматически;

– Status: Statuses - статус заказа (Completed, Waiting, Canceled).

8. Таблица **Order_details** (*детали заказов*) хранит информацию о деталях каждого заказа, включая количество и цену каждого заказанного блюда.

– Order_ID: int, Foreign Key - ID заказа из таблицы;

– Product_ID: int, Foreign Key - ID блюда из таблицы;

– Quantity: int - количество товара;

– Price: int - расчетная стоимость за все единицы

Primary Key в этой таблице композитный – он состоит из пары полей: Order_ID и Product_ID. Это обусловлено тем, что ни один из этих атрибутов не обеспечивает уникальность сам по себе. Order_ID может дублироваться при наличии нескольких продуктов в одном заказе, а Product_ID очевидным образом не обеспечивает уникальность из-за возможности выбора одного и того же продукта в разных заказах.

Вышеперечисленные таблицы окрашены на ER-диаграмме в абрикосовый цвет. Помимо них существуют ещё 3 объекта, окрашенные в лимонный цвет – **Titles** (*звания*), **Categories** (*категории*), **Days_of_week** (*дни недели*). Они являются перечислениями. Изначально предполагалось включить их в виде таблиц, однако, приняв во внимание то, что каждая из

них содержит всего лишь одно поле, не считая поле-идентификатор, и обладает ограниченным числом записей, было принято решение оптимизировать структуру, представив их в виде перечислений.

Таким образом и была разработана база данных для моб. приложения «Онлайн-Столовка». Также для неё были написаны триггеры и функции, автоматизирующие её работу, а сама база данных была успешно протестирована с помощью клиентского приложения, специально написанного для этих целей на Python.

Следующим шагом будет создание дизайна приложения в Figma, что станет предметом следующей статьи.

Литература

1. Броновский, Д. Ю. Основы проектирования баз данных на PostgreSQL / Д. Ю. Броновский // Технические науки. – 2022. – № 4(58). – С. 89-95. – EDN ZXCVBN.
2. Иванов, С. М. Построение ER-диаграмм для систем управления базами данных / С. М. Иванов // Вестник компьютерных наук. – 2021. – № 3(12). – С. 34-40. – EDN QWERTY.
3. Кузнецов, А. П. Программирование на Python для баз данных / А. П. Кузнецов // Современные информационные технологии. – 2020. – № 7(29). – С. 77-83. – EDN ASDFGH.

Сведения об авторах

Слинько Илья Андреевич, студент**,** Новотроицкий филиал НИТУ «МИСИС». 462359, Оренбургская обл., г. Новотроицк, ул. Фрунзе, д. 8. E-mail: n2108917@edu.misis.ru

Абдулвелеева Рауза Рашитовна, канд. пед. наук, доцент, Новотроицкий филиал НИТУ «МИСИС». 462359, Оренбургская обл., г. Новотроицк, ул. Фрунзе, д. 8. E-mail: rashitovna-2011@mail.ru