

Дудник Олег Игоревич, студент,

Санкт-Петербургский государственный экономический университет,

г. Санкт-Петербург

Журба Мария Владимировна, студент,

Санкт-Петербургский государственный экономический университет,

г. Санкт-Петербург

РЕАЛИЗАЦИИ БАЗЫ ДАННЫХ ДЛЯ РЕСТОРАННОГО БИЗНЕСА

Аннотация: Ресторанный бизнес является одним из самых рискованных видов предпринимательской деятельности, в соответствии со справочником банкротств Dun & Bradstreet. В связи с этим, данный вид бизнеса наиболее остро нуждается в применении современных информационных технологий, способных автоматизировать информационные процессы, улучшить ведение учета и снизить нагрузку на менеджеров и управленцев, чтобы минимизировать риски разорения.

Одним из самых простых, но в то же время дающим наибольший результат, методов решения этой задачи является введение технологий баз данных (БД). Базы данных позволяют структурировать информацию, получаемую во время ведения бизнеса, чтобы на её основе строить прогнозы или принимать управленческие решения.

Целью данной работы является проектирование базы данных, которая подходит для ресторанного бизнеса, реализация интерфейса, для пользовательской работы с базой данных, и разработка запросов, которые позволили бы удовлетворить потребности оперативной и управленческой аналитики.

Ключевые слова: ресторанный бизнес, проектирование базы данных, MySQL, php, html, css, js.

Annotation: According to Dun & Bradstreet bankruptcy handbook, the restaurant business is considered one of the riskiest forms of entrepreneurial activity. Due to this, it is crucially in need of modern information technologies capable of automating processes, improving accounting practices, and reducing the burden on managers and executives to minimize the risk of bankruptcy.

One of the simplest yet most effective methods to address this challenge is the implementation of database technologies. Databases allow for the structured organization of information gathered during business operations, enabling businesses to make forecasts and informed managerial decisions based on that data

The aim of this project is to design a database suitable for the restaurant business, implement an interface for user interaction with the database, and develop queries to meet the needs of operational and managerial analytics.

Key words: Restaurant business, database design, MySQL, PHP, HTML, CSS, JS.

Ресторанный бизнес является одним из самых рискованных видов предпринимательской деятельности, в соответствии со справочником банкротств Dun & Bradstreet. В связи с этим, данный вид бизнеса наиболее остро нуждается в применении современных информационных технологий, способных автоматизировать информационные процессы, улучшить ведение учета и снизить нагрузку на менеджеров и управленцев, чтобы минимизировать риски разорения [2].

Одним из самых простых, но в то же время дающим наибольший результат, методов решения этой задачи является введение технологий баз данных (БД). Базы данных позволяют структурировать информацию, получаемую во время ведения бизнеса, чтобы на её основе строить прогнозы или принимать управленческие решения.

В ресторанном бизнесе самым важным, безусловно, является приготовление блюд, поэтому в базе данных должна содержаться информация о

блюдах, нормативные рецепты и ингредиенты. Необходимо так же вести учет о продажах этих блюд, путем сохранения информации о чеках.

Следует так же предусмотреть возможность учета поступления ингредиентов, при помощи записи о поставках и поставщиках.

Определение информационных потребностей

Для более формального описания проблем и постановки задачи в этой работе применяется метод пользовательских историй (user stories) и пользовательских кейсов (user cases), представленный в таблице 1.

Таблица 1. Пользовательские кейсы для БД ресторанного бизнеса

Кто?	Что?	Зачем?
Владелец ресторана	Финансовая отчетность	Отслеживать финансовое положение дел ресторана, как бизнеса
Менеджер по закупкам	Список поставщиков и записи о закупках	Ускорить процесс заказа ингредиентов
Шеф-повар	Блюда и нормативные рецепты	Для создания и ведение рецептуры ресторана
Повар	Ингредиенты на складе	Учет использованных ингредиентов и предотвращения их порчи

Проектирование базы данных. Определение информационных объектов

Перед тем, как начать проектировать и реализовывать базу данных, необходимо определить информационные объекты, для которых эта база данных будет строиться.

Информационными объектами в этой работе будут выступать:

1. Блюда: информация об используемых ингредиентах, энергетической ценности, выхода блюда, размере порций и времени приготовления
2. Ингредиенты: информация о сорте и классе ингредиента

3. Склад: информация о количестве ингредиентов и их сроки годности
4. Меню: информация о ценах блюд
5. Чеки: информация о проданных блюдах, их стоимости и дате продажи
6. Закупки: информация о закупаемых ингредиентах, поставщиках и закупочной спецификации

Нормализация информационных объектов

Для реализации базы данных необходимо построить информационно-логическую модель данных, для этого нужно произвести нормализацию над информационными объектами, с целью получить отношения в третьей нормальной форме. Для этого каждая таблица будет содержать поле id, которое будет содержать уникальное число для каждой записи в таблице, таким образом поле id выступает в качестве уникального ключа каждой записи таблицы. Связи между таблицами будут выполнены путем добавления в зависимую таблицу поля с уникальным ключом (id) независимой таблицы.

Стоит отметить, что ингредиенты и блюда, а также закупки и ингредиенты имеют отношения вида много ко многому. Третья нормальная форма не допускает такие отношения, поэтому для реализации этих отношений будут применены составные таблицы, которые будут содержать ключи связываемых таблиц.

Построенная модель данных представлена на Рисунке 1:

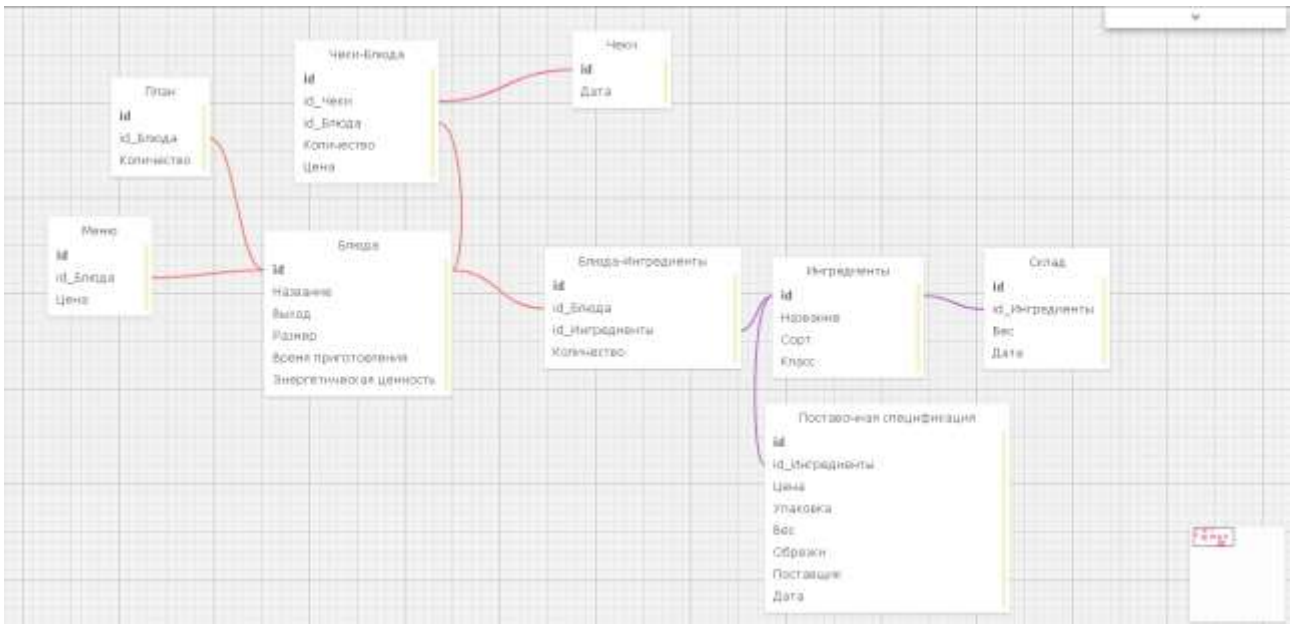


Рисунок 1. Модель данных

Разработка схемы базы данных

У представленной в прошлом разделе модели данных есть существенный недостаток. Рассмотрим его на примере. Пусть у нас есть несколько поставщиков:

1. ООО РУКАРТОН
2. Группа компаний Фуд Сторис
3. A4business
4. MRG

Тогда таблица поставочной спецификации, построенная по модели данных может выглядеть следующим образом:

Таблица 2. Поставочная спецификация

Id	Ингредиент	Цена	Упаковка	Вес	Обрезки	Поставщик	Дата
1	Картофель	10	Коробка	10	0.01	A4business	Сегодня
2	Морковь	25	Пакет	5	0.01	A4business	Сегодня
3	Курица	100	Пакет	4	0.02	MRG	Сегодня
4	Говядина	400	Контейнер	3	0.02	MRG	Сегодня
5	Свинина	250	Контейнер	2	0.02	MRG	Сегодня

Как видно из таблицы 2, название поставщиков дублируются: A4business встречается два раза, а MRG – три. Минимальный размер памяти, занимаемый одним символом, равен 1 байту, тогда как для строки в базе данных принято отводить размер в 256 символов. Таким образом в базе данных для одна текстовая запись будет содержать 256 байт. Количество поставок в день в больших ресторанах может измеряться десятками, а количество ингредиентов – сотнями. Таким образом, вполне естественно ожидать, что таблица для поставочной спецификации в один момент может иметь более 100 000 строк.

Для того, чтобы уменьшить количество занимаемой памяти, можно выделить название поставщиков в отдельную таблицу, а в таблице спецификаций, вместо указания имени поставщика, указывать ключ в таблице поставщиков. Ключ размером всего в 4 байта способен принимать более 4 миллиардов значений. Таким образом, мы сможем уменьшить объем данных, отводимых для поставщика в $256/4 = 64$ раза.

При построении схемы базы данных аналогичный трюк был проделан для:

1. Поставщиков
2. Типа упаковки
3. Размера порций
4. Энергетической ценности блюда
5. Единиц измерения

Для этого были созданы дополнительные таблицы: providers, packs, sizes, energy, units.

Формирование пользовательского интерфейса для ввода данных и диалога

В качестве приложения, которое позволяет работать с базой данных, был реализован web-сайт. В качестве системы управления базой данных была выбрана MySQL, серверный язык – php, драйвера для работы с СУБД – MySQLi [5; 6]. Интерфейс был написан в связке SCC + JS и с использованием фреймворка uikit [3; 4; 7].

Для доступа к функциям сайта нужно пройти авторизацию.

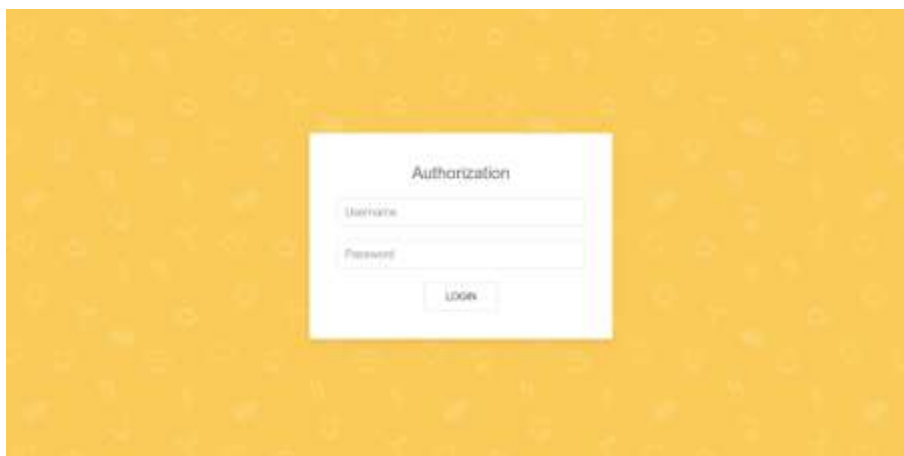


Рисунок 2. Авторизация

В левой части сайта расположено меню навигации. Рассмотрим страницы сайта:

1. Действия:

1.1. Добавить блюдо – позволяет добавить информацию о новом блюде в базу данных.

1.2. Изменить меню – позволяет изменить цены на блюда

1.3. Продать блюдо – записывает информацию о продаже блюда, создает чек, списывает ингредиенты со склада, которые необходимы для приготовления этого блюда

1.4. Принять товары – позволяет записать информацию о поступлении ингредиентов, привозимых поставщиком, добавляется информация о поступлениях ингредиентов на склад.

2. Таблицы:

2.1. Блюда – показывает список всех блюд, указанных в базе данных, и информацию о них.

2.2. Ингредиенты – агрегированная таблица, указывает на общее число ингредиентов на складе.

3. Снабжение:

3.1. Склад – показывает информацию об ингредиентах на складе, с

учетом даты их поставки.

3.2. План – позволяет составить план продаж блюд на следующий день.

3.3. Закупки – указывает, какие ингредиенты нужно закупить, в соответствии с планом продаж блюд

3.4. Поставки – информация о поставках

3.5. Излишки – показывает ингредиенты, которые находятся на складе, но, тем не менее, не участвуют ни в одном из рецептов

4. Отчеты:

4.1. Финансы – информация о закупках и продажах, а также общий итог по этим операциям

4.2. Чеки – информация о чеках

5. Система – системные таблицы, которые не нуждаются в изменениях в коротком периоде, но необходимы для работы системы, к ним относятся: размеры порций, виды упаковок, размерности и поставщики.

БЛЮДО	ЦЕНА
Бургер овощной	100
Бруни	110
Сырники из творога	120
Сырный суп по-французски с курицей	130
Спагетти карбонара с красным соусом	140
Газанья классическая с мясом	150
Бургер овощной	100

Рисунок 3. Страница сайта Меню

Разработка запросов в соответствии с информационными потребностями в данных для оперативной и управленческой аналитики

Для удобства пользователей были разработаны многочисленные SQL-запросы, позволяющие удобно работать с базой данных. Данные запросы применяются в том числе и на сайте. Рассмотрим наиболее интересные из них.

1. Запрос для подсчета всех ингредиентов на складе:

```
SELECT name, grade, class, SUM(warehouse.weight) FROM
ingredients JOIN warehouse ON
ingredients.id=warehouse.ingredient_id GROUP BY
ingredients.id
```

2. Запрос на формирования информации о финансовых операциях:

```
SELECT * FROM (SELECT 0, 'Покупка ингредиентов',
price, day as data FROM purchasing_specification
UNION
SELECT 1, 'Продажа блюда', SUM(checks_dishes.count *
checks_dishes.price), checks.day as data FROM checks
JOIN checks_dishes ON checks_dishes.check_id =
checks.id
JOIN dishes ON dishes.id = checks_dishes.dish_id
GROUP BY checks.id) as t
ORDER BY t.data DESC
```

3. Запрос для подсчета дохода финансовых операций:

```
SELECT SUM(total) FROM
(SELECT SUM(checks_dishes.count *
checks_dishes.price) as total FROM checks
JOIN checks_dishes ON checks_dishes.check_id =
checks.id
JOIN dishes ON dishes.id = checks_dishes.dish_id
GROUP BY checks.id) as t
```

4. Запрос на формирование краткой информации о блюдах:

```
SELECT dishes.id, dishes.name,
sizes.name, dishes.output,
dishes.cooking_time, energy.calories
FROM dishes
JOIN sizes ON dishes.size_id=sizes.id
```

```
JOIN energy ON energy.dish_id=dishes.id
```

5. Запрос для подсчета общей стоимости блюд в чеке:

```
SELECT checks.id,
SUM(checks_dishes.count*checks_dishes.price) FROM checks
JOIN checks_dishes ON checks_dishes.check_id = checks.id
JOIN dishes ON dishes.id = checks_dishes.dish_id
GROUP BY checks.id
```

6. Запрос для подсчета необходимой закупки ингредиентов в соответствии с планом:

```
SELECT name, ware.count, need.count, ware.count-
need.count FROM
(SELECT ingredients.id as id,
SUM(plan.count * dishes_ingredients.count * units.si)
as count
FROM plan
JOIN dishes ON dishes.id=plan.dish_id
JOIN dishes_ingredients ON
dishes_ingredients.dish_id=dishes.id
JOIN units ON dishes_ingredients.units_id=units.id
JOIN ingredients ON
dishes_ingredients.ingredient_id=ingredients.id
GROUP BY ingredients.id
) as need
JOIN (
SELECT ingredients.id as id, ingredients.name as name,
SUM(warehouse.weight) as count FROM ingredients
JOIN warehouse ON
ingredients.id=warehouse.ingredient_id
GROUP BY ingredients.id
) as ware ON ware.id=need.id
```

7. Запрос для подсчета лишних ингредиентов:

```
SELECT ingredients.name, warehouse.weight FROM
ingredients JOIN warehouse ON
warehouse.ingredient_id=ingredients.id
WHERE ingredients.id NOT IN (SELECT ingredients.id
FROM ingredients
JOIN
dishes_ingredients ON
dishes_ingredients.ingredient_id=ingredients.id
JOIN dishes ON dishes.id=dishes_ingredients.dish_id)
```

8. Функция для удаления ингредиентов со склада, при покупке блюда:

```
function deleteIngredients($dish_id, $dish_count)
{
    global $mysqli;
    $stmt = $mysqli->prepare("SELECT ingredient_id,
count*si FROM dishes_ingredients JOIN units ON
units.id=units_id WHERE dish_id=?");
    $stmt->bind_param("i", $dish_id);
    $stmt->execute();
    $res = $stmt->get_result();
    $res = $res->fetch_all();
    $ingredients = array();
    foreach ($res as $value)
        $ingredients[$value[0]] =
$value[1]*$dish_count;
    foreach ($ingredients as $key => $count) {
        $stmt = $mysqli->prepare("SELECT id, weight,
spoil FROM warehouse WHERE ingredient_id=? ORDER BY
spoil");
        $stmt->bind_param("i", $key);
```

```

$stmt->execute();
$res = $stmt->get_result();
$all = $res->fetch_all();
foreach ($all as $value) {
    if($count <= 0) break;
    if($value[1] <= $count) {
        $count -= $value[1];
        $stmt = $mysqli->prepare("DELETE FROM
warehouse WHERE id=?");
        $stmt->bind_param("i", $value[0]);
        $stmt->execute();
    } else
    {
        $stmt = $mysqli->prepare("UPDATE
warehouse SET weight=?, spoil=? WHERE id=?");
        $diff = $value[1] - $count; $count =
0;
        $stmt-
>bind_param("dsi", $diff, $value[2], $value[0]);
        $stmt->execute();
    }
}
}
}

```

Заключение

В данной работе был произведен анализ ресторанного бизнеса и его потребностей. Для удовлетворения этих потребностей были выделены информационные объекты, произведена их нормализация, построена модель данных.

На основе модели данных была построена схема базы данных, с введением

дополнительных таблиц и, с целью уменьшения хранимых данных и описанием этого момента.

В заключительной части был создан сайт, позволяющий работать пользователям с базой данных и сформированы основные запросы к базе данных, которые функционально доступны на сайте.

Библиографический список:

1. Адитья Бхаргава – Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих – СПб,: Питер, 2017.
2. Кристофер Эгертон-Томас – Ресторанный бизнес. Как открыть и успешно управлять рестораном – М.: РосКонсульт, 1999. - 272 с.
3. CSS [Электронный ресурс] / Режим доступа: – <https://developer.mozilla.org/ru/docs/Web/css>.
4. JavaScript [Электронный ресурс] / Режим доступа: – <https://developer.mozilla.org/ru/docs/Web/JavaScript>.
5. MySQLi [Электронный ресурс] / Режим доступа: – <https://www.php.net/manual/ru/intro.mysql.php>.
6. HTML [Электронный ресурс] / Режим доступа: – <https://developer.mozilla.org/ru/docs/Web/html>.
7. Uikit [Электронный ресурс] / Режим доступа: – <https://getuikit.com/>.