Государственное автономное профессиональное образовательное учреждение   
Чувашской Республики «Межрегиональный центр компетенций –  
 Чебоксарский электромеханический колледж» Министерства образования Чувашской Республики

Дисциплина МДК.11.01 Технология разработки и защита базы данных

КУРСОВОЙ ПРОЕКТ

Разработка и защиты базы данных

автоматизированной информационной системы службы доставки ресторана

КП.Ип5-21.12.МДК.11.01.ПЗ

Выполнил студент 3 курса, группы Ип5-21

Никитин Константин Евгеньевич

(Фамилия И. О.)

(подпись) (чч.мм.гггг)

Преподаватель Ордяков Д.Е.

(Фамилия И. О.)

Защищен

(чч.мм.гггг)

с оценкой

Подпись

(подпись) (расшифровка подписи)

2023

СОДЕРЖАНИЕ

[Введение 3](#_Toc151719161)

ВВЕДЕНИЕ

В современных условиях стало обычным наличие обширного объема информации у предприятий. Для эффективного управления этой информацией необходима информационная система, способная оперировать данными в соответствии с определенными параметрами. Одним из эффективных инструментов для этой цели является база данных.

База данных (БД) представляет собой организованный набор данных, отражающих состояние объектов и их взаимосвязи в рассматриваемой предметной области.

Ресторанный сектор, не отставая от этой тенденции, в настоящее время является одной из отраслей с наиболее динамичным развитием, требующей эффективных и надежных информационных систем для управления своими комплексами.

Использование автоматизированных информационных систем в ресторанном бизнесе ускоряет и упрощает обработку базы данных, что в конечном итоге приводит к повышению эффективности управления предприятием и улучшению уровня обслуживания клиентов.

Интеграция работы с базами данных в ресторанной сфере обеспечивает возможность эффективно обрабатывать обширные объемы информации. С увеличением объема данных в базе становится жизненно важным обеспечение надежной защиты от несанкционированного доступа и утечек. Разработка современных технологий защиты баз данных становится неотъемлемым аспектом работы специалистов в данной области.

Это делает тему — "Технология разработки и защиты базы данных" актуальной. Создание приложения для базы данных предоставит возможность управлять данными: просматривать, редактировать, добавлять, удалять и фильтровать информацию.

Цель: разработать базу данных для автоматизированной информационной системы ресторана.

Задачи:

1. Проанализировать и описать предметную область;
2. Разработать концептуальную и логическую модели; составить список сущностей и атрибутов, построить реляционную модель на основе логической модели;
3. Cпроектировать базу данных с использованием средств SQL;
4. Разработать приложение для управления базой данных;

1 Инфологическое проектирование

1.1 Анализ и описание предметной области

Концептуальное проектирование – это построение модели предметной   
области, ориентированной на восприятие человека и независимой от типа системы управления базами данных (СУБД) и типа самой базы данных.

Под предметной областью (ПО) понимается совокупность связанных между собой функций, задач управления в некоторой области деятельности предприятия, с помощью которых достигается выполнение поставленной цели.

Множество фрагментов составляют из себя предметную область. Каждый фрагмент предметной области характеризуется множеством объектов и процессов, использующих объекты, а также множеством пользователей, с разными взглядами на ПО.

Модель предметной области представляет собой знания о предметной   
области. Знания могут быть неформальными и формальными. Модель предметной области описывает скорее процессы, происходящие в предметной области и   
данные, используемые этими процессами. От того, насколько правильно   
смоделирована предметная область, зависит успех дальнейшей разработки   
приложений.

1.2 Разработка концептуальной модели

Для разработки базы данных необходимо определить основные сущности и необходимые поля. Так же следует выделить сопоставимые поля для определения связей между таблицами.

База данных должна содержать данные:

* о комнатах в гостинице;
* о клиентах;
* о заказах;
* о работниках;
* о курьерах.

Учитывая все вышеперечисленное, база данных должна содержать следующие сущности и их атрибуты.

Сущность «Clients» (Клиенты):

* «ID\_Client» (Персональный код клиента);
* «ClientName» (Имя клиента);
* «Adress» (Адрес клиента);
* «PhoneNumber» (Телефонный номер клиента).

Сущность «Order» (Заказы):

* «ID\_Order» (Персональный код заказа);
* «Status» (Статус заказа);
* «OrderContent» (Содержание заказа);
* «Date» (Дата заказа);
* «Cost» (Стоимость заказа).

Сущность «Workers» (Работники):

* «WorkerID» (Персональный код сотрудника);
* «WorkerName» (Имя сотрудника);
* «Post » (Должность)
* «WorkPlace» (Место работы в ресторане).

Сущность «Couriers» (Курьеры):

* «ID\_Courier» (Персональный код курьера);
* «DeliveryStatus» (Статус заказа);
* «CourierName» (Имя курьера).

Таким образом, были определены основные объекты будущей информационной системы.

Установим связи между сущностями и распишем их с помощью ERD-инструкции:

«Clients» и «Order», связь «один-ко-многим». Бизнес-правила:

1. каждый клиент (Clients) должен иметь один или более заказов (Order).
2. каждый заказ (Order) должен иметь один и только одного клиента (Clients).

«Order» и «Couriers», связь «один-ко-многим». Бизнес-правила:

1. каждый курьер (Couriers) может иметь один или более заказов (Order).
2. каждый заказ (Order) должен быть у одного и только одного клиента (Clients).

«Order» и «Workers», связь «один-к-одному». Бизнес-правила:

1. каждый заказ (Order) должен готовить один или более сотрудников (Workers).
2. каждый сотрудник (Workers) может готовить один или более заказов (Order).

При осуществлении нормализации текущих отношений было осуществлено приведение к нормальным формам:

* первая нормальная форма (1NF) — все атрибуты должны иметь только одно значение;
* вторая нормальная форма (2NF) — отношения находятся в первой нормальной форме и все не ключевые атрибуты зависят от первичного ключа;
* третья нормальная форма (3NF) — отношения находятся во второй нормальной форме и все неключевые атрибуты не транзитивно зависят от первичного ключа.

Нормализация – это процесс преобразования отношений базы данных к виду, отвечающему нормальным формам.

Нормализация предназначена для приведения структуры баз данных к виду, обеспечивающему минимальную логическую избыточность, и не имеет целью уменьшение или увеличение производительности работы или же уменьшение, или увеличение физического объёма базы данных.

Под избыточностью понимают повторение данных в разных строках одной таблицы или в разных таблицах БД. Итогом нормализации является уменьшение потенциальной противоречивости, хранимой в базе данных информации.

В заключение главы выделим все выполненные в процессе создания концептуальной модели действия:

* были определены объекты базы данных и их атрибуты;
* были определены связи между объектами;
* были определены действия необходимые для приведения отношений в базе данных к нормальной форме.

2. Определение требований к операционной системе

По данной формуле будет вычисляться объем памяти, требуемый для хранения данных, на основании данных из таблицы 2.1:

𝑛

Мд = 2 ∑ 𝑙𝑖 ∗ (𝑁𝑖 + 𝑁𝑎𝑖 )

𝑖=1

где 𝑙𝑖 - длина записи в i-ой таблице (в байтах);

𝑁𝑖 - примерное (максимально возможное) количество записей в i-ой таблице;

𝑁𝑎 – количество записей в архиве i-ой таблицы;

*n* – количество таблиц в БД.

Таблица 2.1 – Атрибуты, отображающие длину записи в каждой таблице

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Отношения | Атрибут | Тип данных | Размер,  байт | Среднее  кол-во | Объем,  байт |
| Clients | ID\_Client | INT | 2 |  |  |
| ClientName | NVARCHAR | 120 |
| Adress | NVARCHAR | 100 |
| PhoneNumber | NVARCHAR | 22 |
| Couriers | ID\_Courier | INT | 2 |  |  |
| DeliveryStatus | BIT | 1 |
| CourierName | NVARCHAR | 140 |
| Order | ID\_Order | INT | 2 |  |  |
| Status | BIT | 1 |
| OrderContent | NVARCHAR | 600 |
| Date | DATE | 3 |
| Client\_ID\_Client | INT | 2 |
| Workers\_WorkerID | INT | 2 |
| Couriers\_ID\_Courier | INT | 2 |
| Cost | FLOAT | 4 |

Продолжение таблицы 2.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Отношения | Атрибут | Тип данных | Размер,  байт | Среднее  кол-во | Объем,  байт |
| Workers | WorkerID | INT | 2 |  |  |
| WorkerName | NVARCHAR | 140 |  |  |
| Post | NVARCHAR | 80 |  |  |
| WorkPlace | NVARCHAR | 80 |  |  |

Таким образом, объем памяти хранения данных за один год, составит:

МС = () = байт

Полученный объем памяти не будет являться постоянным, потому что он увеличивается или уменьшается по ходу работы с базой данных.

После проведенных вычислений становится ясно, что объем данных в информационной системе небольшой. Следовательно, для реализации проекта подойдет любой современный компьютер.

Объем оперативной памяти, который необходим, будет определяться требованиями выбранной системой управления баз данных, следовательно, СУБД MS SQL подходит для введения данной базы данных.

Для поддержания целостности данных выполняется резервное копирование баз данных. План резервного копирования приведен в таблице 2.2.

Таблица 2.2 – План резервного копирования

|  |  |
| --- | --- |
| День недели | Вид резервного копирования |
| Понедельник | Разностное резервное копирование (клиентов и заказов) |
| Вторник | Разностное резервное копирование (клиентов и заказов) |

Продолжение таблицы 2.2

|  |  |
| --- | --- |
| Среда | Разностное резервное копирование (клиентов и заказов) |
| Четверг | Разностное резервное копирование (клиентов и заказов) |
| Пятница | Разностное резервное копирование (клиентов и заказов) |
| Суббота | Разностное резервное копирование (клиентов и заказов) |
| Воскресенье | Полное резервное копирование всей БД |

В Microsoft SQL Server принята практика разных типов резервного копирования. Пользователям доступно:

1. полное – делается резервная копия всей БД;
2. дифференциальное или разностное – осуществляется копирование данных с того момента, когда осуществлялось ее последнее полное резервирование;
3. логов или инкрементальное.

3. Выбор СУБД и других программных средств

При создании базы данных и клиентского приложения были решены важные и сложные задачи, для успешного выполнения которых нужно было правильно сделать выбор средства реализации проекта. Для создания концептуальной и логической моделей базы данных Oracle SQL Developer Data Modeler предоставил много возможностей. Приложение является бесплатным графическим программным обеспечением, с помощью которого можно сконструировать различные модели данных, такие как логические, реляционные, многомерные, сетевые и т. д.

СУБД MS SQL Server стала отличным выбором для создания, использования и управления будущей базы данных. Обладая не только всеми необходимыми функциями для создания клиентского приложения, но и имея высокую производительность, она стала самым удачным выбором, так как другие приложения компании Microsoft могут работать с ней без каких-либо проблем.

Следовательно, наилучшим вариантом после выбора СУБД стала программная среда Microsoft Visual Studio с использованием объектно-ориентированного языка программирования C# для разработки пользовательского интерфейса и программного кода, и комфортной работы пользователей с базой данных.

Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows.

Язык программирования C# был выбран из-за его перспективности среди других языков и обстоятельств, облегчающих дальнейшее сопровождение программного обеспечения.

4. Логическое проектирование реляционной базы данных

Логическое проектирование является созданием схемы баз данных на основе конкретной модели данных, то есть в данном случае реляционной модели данных, представленных на рисунках 4.1 и 4.2.

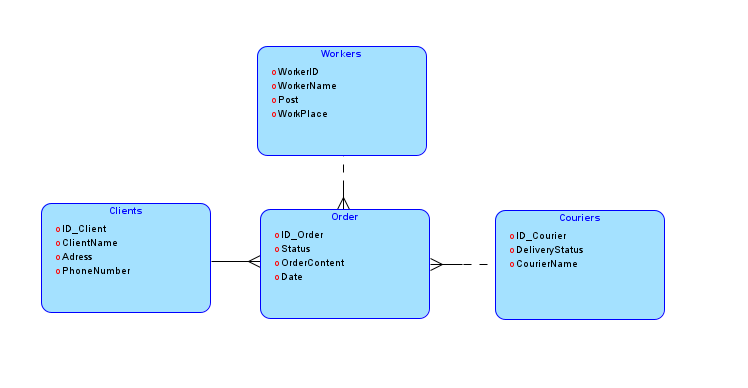


Рисунок 4.1 – Логическая модель базы данных

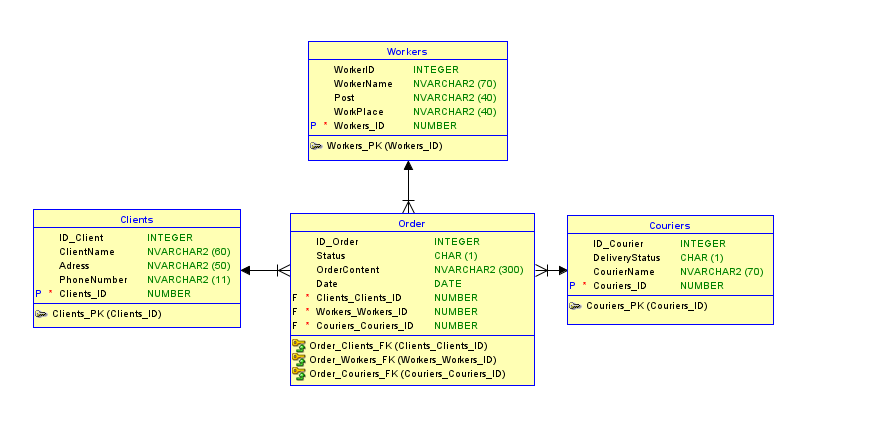


Рисунок 4.2 – Реляционная модель базы данных

Сущностью в базе данных может являться практически любой объект, который возможно логически выделить из предметной области, а ее атрибутом именованный столбец таблицы. Особыми атрибутами считаются первичные и внешние ключи.

Первичный ключ – это столбец, значения которого однозначно определяет строку.

Внешний ключ – это столбец в данной таблице, значения которого совпадает со значениями столбца, являющийся первичным ключем в другой таблице.

Во время разработки структуры базы данных были созданы таблицы, представления, хранимые процедуры и триггеры.

В базе данных таблица является совокупностью, связанной между собой информации, хранящейся в определенном порядке. Любая таблица состоит из кортежей (строк) и атрибутов (столбцов).

Представленная база данных состоит из четырех таблиц:

1. Таблица «Clients»;
2. Таблица «Order»;
3. Таблица «Workers»;
4. Таблица «Couriers».

Таблица «Clients» представляет собой совокупность информации о клиентах ресторана. Таблица «Order» хранит информацию о заказах, таблица «Workers» хранит информацию о сотрудниках ресторана. Таблица «Couriers» хранит данные о курьерах.

Все таблицы созданы в одной базе данных. Описание структуры таблиц и их полей приведены в таблицах 4.1 – 4.5.

Таблица 4.1 – Структура таблицы «Clients»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Длина | Содержание | Ограничения |
| ID\_Clients | Int | 2 | Идентифицирующий  номер клиента | Первичный  ключ |
| ClientName | Varchar(30) | 30 | Имя клиента | NOT NULL |
| Adress | Varchar(30) | 30 | Адрес | NOT NULL |
| PhoneNumber | Varchar(30) | 30 | Номер телефона | NOT NULL |

Таблица 4.2 – Структура таблицы «Couriers»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Длина | Содержание | Ограничения |
| ID\_Courier | Int | 2 | Идентифицирующий  номер курьера | Первичный  ключ |
| DeliveryStatus | Bool | 1 | Статус доставки | NOT NULL |
| CourierName | Varchar(30) | 30 | Имя курьера | NOT NULL |

Таблица 4.3 – Структура таблицы «Workers»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Длина | Содержание | Ограничения |
| WorkerID | Int | 2 | Идентифицирующий  номер транзакции | Первичный  ключ |
| WorkerName | Varchar(30) | 30 | Имя сотрудника | NOT NULL |
| Post | Varchar(30) | 30 | Должность | NOT NULL |
| WorkPlace | Varchar(30) | 30 | Место работы | NOT NULL |

Таблица 4.4 – Структура таблицы «Order»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Длина | Содержание | Ограничения |
| ID\_Order | Int | 2 | Идентифицирующий  номер заказа | Первичный  ключ |
| Status | Bool | 1 | Статус | NOT NULL |

Продолжение таблицы 4.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OrderContent | Varchar(200) | 200 | Содержимое заказа | NOT NULL |
| Date | Date | 8 | Дата заказа | NOT NULL |
| Cost | Varchar(30) | 30 | Стоимость заказа | NOT NULL |

После создания таблиц и добавления данных в MS SQL Server база данных выглядит следующим образом (рисунки 4.1-4.2).

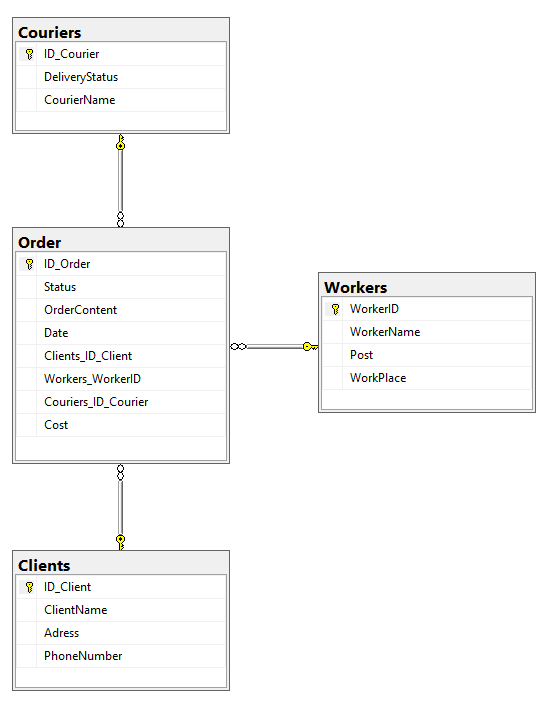


Рисунок 4.1 – Диаграмма базы данных