

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ УЧЕТА ПРОДАЖ КНИЖНОГО МАГАЗИНА

В рамках работы рассмотрены методы работы с информацией: запросы, функции, процедуры, и т. д., разработан программный код и интерфейс базы данных, проект базы данных учета информации о продажах книжного магазина при помощи Microsoft SQL Server.

Ключевые слова: база данных, сервер, система баз данных, учет данных, SQL server

В настоящее время информационные системы (ИС) являются неотъемлемой частью бизнес-процессов многих компаний. Они позволяют автоматизировать многие задачи, упростить управление процессами и повысить эффективность работы. Одним из бизнес-секторов, где используются информационные системы, является книжный рынок.

Основой современных ИС являются базы данных (БД). Одним из основных преимуществ баз данных является то, что пользователи БД получают возможность эффективной работы с таблицами, данными, запросами без потребности знания реальной организации данных во внешней памяти.

Основной целью проектируемой базы данных является управление информацией по заказам.

Проектирование базы данных является очень важным этапом, от которого зависят последующие этапы разработки СУБД.

Предлагаемую базу данных для оптимизации работы книжного магазина разработаем при помощи Microsoft SQL Server,

На основании анализа, были выделены следующие сущности и их атрибуты [1]:

1. Книги (Books):

- Код книги (BookID) – number;
- Код издательства (PublishingHouseID) – number;
- Название книги (BookName) – string;
- Автор книги (BookAuthor) – string;
- Жанр книги (BookGenre) – string;
- Цена (Price) – number.

2. Издательства (PublishingHouses):

- Код издательства (PublishingHouseID) – number;
- Название издательства (PublishingHouseName) – string.

3. Заказы (Orders):

- Код заказа (OrderID) – number;
- Код книги (BookID) – number;
- Код покупателя (CustomerID) – number;
- Дата продажи (OrderDate) – data;
- Количество (Amount) – number.

4. Клиенты (Customers):

- Код покупателя (CustomerID) – number;
- Фамилия покупателя (CustomerSurname) – string;
- Имя покупателя (CustomerName) – string;
- Отчество покупателя (CustomerPatronymic) – string;
- Номер телефона (PhoneNumber) – number.

В базе данных имеется тип множественной связи «один ко многим» (1:n):

– связь между сущностью «Customers» – «Orders», то есть один клиент может совершить несколько заказов;

– связь между сущностью «Books» – «Orders», то есть одно наименование книги может участвовать в нескольких заказах;

– связь между сущностью «PublishingHouses» – «Books», то есть одно издательство может предоставлять несколько различных книг.

Логический и физический уровни модели базы данных были разработаны в Miro.

Выразим концептуальную модель предметной области в терминах выбранной модели данных применительно к нашей СУБД. Создадим диаграмму базы данных в MS SQL Server [2]. Результаты даталогического моделирования представлены на рис. 1.

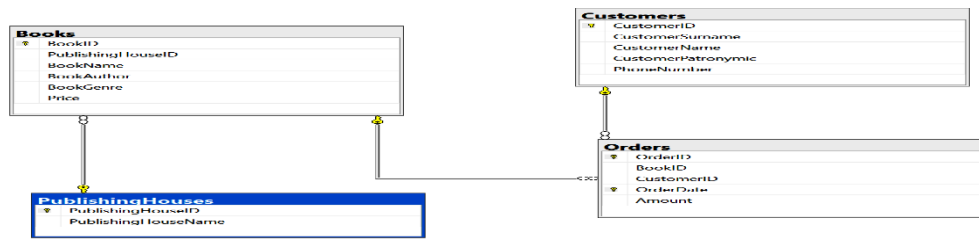


Рис. 1. Модель базы данных

Исходя из модели разработаем запросы на языке SQL для создания базы данных, определим места хранения, размер и названия файлов базы данных.

Далее осуществим разработку запросов на языке SQL для создания всех таблиц.

На данном этапе были разработаны запросы для создания таблиц [3]. Все связи между таблицами имеют свойства «Ограничения целостности данных», ограничений на значения, и т. д. Также, установлено свойство «Каскадное обновление».

Как результат, в обозревателе объектов базы данных мы видим созданные нами таблицы (рис. 2).

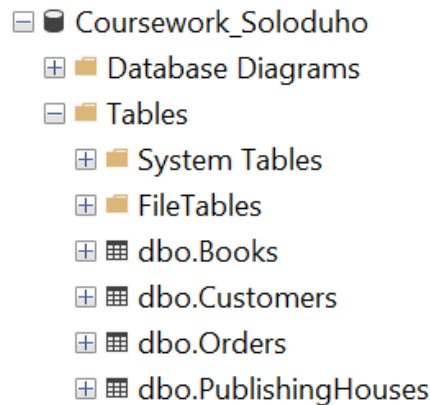


Рис. 2. Таблицы базы данных

Следующим этапом разработаем запросы на языке SQL для создания индексов и ограничений для таблиц БД.

Уникальный индекс гарантирует, что ключ индекса не будет содержать одинаковых значений, а значит, каждая строка в таблице будет уникальна [4].

На данном этапе, были созданы запросы на языке SQL для создания уникальных и неуникальных индексов соответствующих таблиц.

Создание схемы данных (диаграмму), включая в нее необходимые таблицы.

На данном этапе разработки была создана схема данных, включающая в себя все таблицы, созданные ранее. Все связи имеют свойство «Обеспечение целостности данных» и свойство «Каскадное обновление».

Чтобы освоить навык работы с запросами, мы создаем тестовую таблицу TestTable, выбрав все столбцы из таблицы Orders [5].

Далее мы изменяем структуру созданной таблицы, добавив в нее новый столбец BookName, в котором будет храниться информация о названии книги.

Заполняем новое поле с помощью запроса на обновление данных на языке SQL.

Далее обновляем значения в поле BookName с помощью запроса на языке SQL с учетом условия – BookID=2.

Создаем запрос на языке SQL на добавление в таблицу TestTable новых записей.

Составляем запрос на языке SQL на удаление записей из созданной таблицы, где количество книг в заказе меньше 5 [6].

Составляем запрос на языке SQL, отображающий все поля одной таблицы с отбором по одному условию и сортировкой. Запрос отображает все поля из таблицы Books. Условие отбора – цена > 30, и сортировкой по BookID.

Далее был создан запрос, отображающий поля OrderDate, Amount, OrderID из таблицы Orders с отбором по сложному условию (если заказ был оформлен в феврале или если количество книг в заказе больше 10) и сортировкой по полям OrderDate, OrderID и Amount.

Составили запрос с выборкой данных из таблиц Books и Orders [7], при условии, что OrderDate находится в промежутке от 1 до 28 февраля, выполнили сортировку данных по total_amount в обратном порядке.

Разработали запрос на языке SQL, результирующая таблица которого соответствует по структуре записи таблице Orders, но коды в полях внешнего ключа заменили соответствующими им наименованиями из родительских таблиц (BookID заменили на BookName, а CustomerID заменили на CustomerSurname, CustomerName, CustomerPatronymic) [8]. Выполнили сортировку по OrderData, OrderID.

Разработали запрос на языке SQL, результирующая таблица которого соответствует по структуре записи таблице Orders, но коды в полях внешнего ключа заменили соответствующими им наименованиями из родительских таблиц (BookID заменили на BookName, а CustomerID заменили на CustomerSurname, CustomerName, CustomerPatronymic). Отсортировали данные по ФИО клиента и количеству книг в заказе. Добавили столбец OrderCost, в котором рассчитали стоимость заказа. А также рассчитали стоимость всех заказов.

Создаем хранимую процедуру, которая выводит значения OrderDate, CustomerSurname, CustomerName, CustomerPatronymic, BookName, Amount и OrderCost по ID заказа.

Создаем еще одну хранимую процедуру, которая выводит значения BookAuthor, BookName, Price, Sales (количество проданных экземпляров книги) и Revenue (выручка по книге) по наименованию издательства.

Далее была создана пользовательская функция типа Inline Table-valued, которая вычисляет общую стоимость заказов по каждому клиенту за определенный период времени.

Пользовательская функция типа Multi-statement table-valued, которая возвращает список всех заказов определенного клиента за определенный год. Данная информация может понадобиться для маркетингового исследования либо, когда клиенту необходимо что-то посоветовать на основе предыдущих заказов.

Разработали функцию с использованием курсора, которая выводит ФИО клиента, количество заказов, сделанных клиентом, количество книг во всех этих заказах и общую стоимость заказов за обозначенный период времени.

Разработали триггер для таблицы Orders [9], который запрещает обновление столбца OrderDate.

Следующий триггер создан для таблицы Books. При добавлении новой книги триггер выводит сообщение “Добавлена новая книга + название этой книги”.

Следующий триггер создан для таблицы PublishingHouses. При удалении издательства триггер выводит сообщение “Сотрудничество с издательством ‘имя издательства’ прекращено”.

Создаю триггер с курсором, с помощью которого можно изменить цену книги, если это требуется [10].

Не менее важным аспектом работы с базой данных является управление пользователями и системой безопасности базы данных.

На данном этапе были созданы 3 роли (менеджер по продажам, главный менеджер и системный администратор) и 4 пользователя. Также, были назначены полномочия на объекты базы данных этим пользователям.

Также, на случай непредвиденных обстоятельств, важно разработать план резервного копирования, создать резервную копию базы данных

Для разработки варианта интерфейса приложения можно использовать среду платформу .NET (язык C#).

Главная форма хранит в себе информацию о заказах. Именно с неё можно осуществлять переходы на другие формы при клике на соответствующие menuStrip. Также есть возможность поиска информации о заказе по ID заказа.

Первый раздел, вход в который можно осуществить с главной формы – это книги. В нем хранится не только таблица Books, но еще и таблица Orders для того, чтобы, выбирая определенную

книгу, можно было отслеживать, осуществлялись ли на нее заказы или нет. Таким образом, это форма с подчиненной таблицей.

Последняя созданная форма – «Издательства». В ней хранится информация об издательствах: ID издательства и название издательства. Есть возможность поиска издательства по наименованию издательства.

Таким образом, разработанная база данных выполняет следующий перечень требований:

- представление данных в удобном для пользователя виде;
- добавление, редактирование и удаление данных;
- фильтрация и сортировка данных;
- навигация по набору данных.

Преимущество использования автоматизированной системы состоит в том, затрачивается меньше времени на обработку данных, чем при ручной обработке, меньше места для хранения информации, так как в бумажном виде та же самая информация занимает большее пространство. Автоматизированная система отличается быстродействием, оперативностью обработки и наглядным и удобным интерфейсом.

Список использованных источников

1. *Осипов, Д.* Технологии проектирования баз данных / Д. Осипов Издательство ДМК-Пресс, 2019. – 66 с.
2. *Уолтер Шилдс.* SQL: быстрое погружение / Уолтер Шилдс – Издательство Питер, 2022. – 26 с.
3. Таблицы SQ [Электронный ресурс]. – 2022. Режим доступа: <https://learn.microsoft.com/ru-ru/sql/relational-databases/>. – Дата доступа: 15.04.2023.
4. Microsoft sql server: преимущества и недостатки [Электронный ресурс]. – 2022. Режим доступа: <https://astv.ru/news/materials/microsoft>. – Дата доступа: 11.04.2023.
5. *Болье, А.* Изучаем SQL / Алан Болье. – Издательство Диалектика, 2021. – 54 с.
6. *Моргунов, Е.* PostgreSQL. Основы языка SQL / Е. Моргунов. – Издательство BHV, 2018. – 224 с.
7. *Грабер, М.* SQL для простых смертных / М. Гарбер. – М. : ЗАО «Издательство «Лори», 2014 – 76 с.
8. *Сергей Тарасов.* СУБД для программиста. Базы данных изнутри. / С. Тарасов. – Издательство Солон-пресс, 2018. – 187 с.
9. Robin Dewson. Beginning SQL Server for Developers (The Expert's Voice in SQL Server) – 2014 – 567 с.
10. *Daniel Nichter.* Efficient MySQL Performance. Best Practices and Techniques / Daniel Nichter, 2021. – 225 с.