Белорусский национальный технический университет

Факультет технологий управления и гуманитаризации

Кафедра «Таможенное дело»

КУРСОВАЯ РАБОТА

по дисциплине

”Компьютерные информационные технологии”

|  |  |
| --- | --- |
| Тема: | Разработка и управление базой данных «Турфирма» |
|  | |
|  | |

|  |  |
| --- | --- |
| Исполнитель: | Якубенко Филипп Анатольевич |

*(фамилия, имя, отчество)*

|  |  |  |  |
| --- | --- | --- | --- |
| студент | I | курса ФТУГ, группа | 108410 |

*Руководитель: ст.преподаватель Разоренова Т. Р.*

Минск 2011

Белорусский национальный технический университет

Факультет технологий управления и гуманитаризации

Кафедра «Таможенное дело»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе по дисциплине**

# ”Компьютерные информационные технологии”

|  |  |
| --- | --- |
| Тема |  |
| Разработка и управление базой данных «Турфирма» | |
|  | |
|  | |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Исполнитель:** |  | ( | ***Якубенко Ф.А*** | ) |

(подпись) (фамилия, инициалы)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cтудент** | I | **курса** | 108410 | **группы** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Руководитель:** |  | ( | ***Разоренова Т.Р.*** | ) |

(подпись) (фамилия, инициалы)

**Минск 2011**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **БелорусскИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ** | | | | | | | |
| **Факультет \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Технологий управления и гуманитаризации**  **Кафедра « \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ »**  **Таможенное дело** | | | | | | | |
| «УТВЕРЖДАЮ» |  | | | | | | |
| Заведующий кафедрой | | | | | |  | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Б.Б. Синьков\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | | |  |  | | |
| подпись инициалы и фамилия |  | | | | | | |
| «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 г. |  |  |  | | | |  |

**ЗАДАНИЕ**

**по курсовому проектированию**

|  |  |  |
| --- | --- | --- |
| Студенту |  | |
| 1.Тема работы | **Разработка и управление базой данных** | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| 2. Сроки сдачи студентом завершенной работы | | . .20 |
| 3. Исходные данные к работе:  Теоретический раздел: | |  |
| Практический раздел: База данных должна описывать следующие основные объекты и их связи:   * Разработать структуру таблиц и схему данных с обеспечением целостности. * Допускается добавление других информационных объектов, таблиц-справочников и описательных полей. * Заполнить таблицы данными, достаточными для демонстрации приложения. * Разработать не менее пяти запросов разных типов, имеющих практическое значение для созданной базы данных. Описать краткий алгоритм создания каждого типа запроса, назначение и вид каждого разработанного запроса в режиме конструктора и SQL. * Разработать формы и отчеты (назначение, краткий алгоритм создания, вид в режиме конструктора и описание элементов управления для каждой разработанной формы или отчета). * Разработать управляющую форму, применить элементы автоматизации и макросы для демонстрации приложения по управлению базой данных. | | |
| 4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат | | |
| рассмотрению) | | |
| Титульный лист | | |
| Задание | | |
| Содержание | | |
| Введение | | |

СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 4](#_Toc293678348)

[ВВЕДЕНИЕ 5](#_Toc293678349)

[1. ТЕОРИТИЧЕСКИЙ РАЗДЕЛ «Нормализация модели» 6](#_Toc293678350)

[1.1. Обзор понятий «Нормализация модели» 6](#_Toc293678351)

[1.2. Типы «Нормальных форм» 7](#_Toc293678352)

[2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ «Турфирма» 11](#_Toc293678353)

[2.1. Анализ предметной области и составление спецификаций данных 11](#_Toc293678354)

[2.2. Информационно-логическая модель базы данных «Турфирма» 12](#_Toc293678355)

[2.3. Физическая модель базы данных «Турфирма» 14](#_Toc293678356)

[3. СОЗДАНИЕ ЗАПРОСОВ, ОТЧЕТОВ, ФОРМ БАЗЫ ДАННЫХ «Турфирма» 15](#_Toc293678357)

[3.1. Создание запросов 15](#_Toc293678358)

[3.2. Создание форм 17](#_Toc293678359)

[3.3. Создание отчетов 18](#_Toc293678361)

[3.4. Создание кнопочной формы 22](#_Toc293678362)

[ЗАКЛЮЧЕНИЕ 24](#_Toc293678363)

[ЛИТЕРАТУРА 25](#_Toc293678364)

[ПРИЛОЖЕНИЯ 26](#_Toc293678365)

**ВВЕДЕНИЕ**

В настоящее время среди разработчиков базы данных (БД) большой популярностью пользуется реляционная СУБД ACCESS, входящая в состав пакета Microsoft Office 2007. Дружественный интерфейс и простота настройки, эффективные средства создания таблиц, форм, запросов, интеграция с другими приложениями пакета, средства организации работы с базами данных и защита информации - вот далеко не полный перечень достоинств этого приложения.

Основные функции СУБД – это описание структуры базы данных, обработка данных и управление данными.

База данных – это совокупность сведений о реальных объектах, процессах, событиях или явлениях, относящихся к определённой теме или задаче, организованная таким образом, чтобы обеспечить удобное представление этой совокупности, как в целом, так и любой её части. Реляционная база данных представляет собой множество взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определённого типа. Каждая строка таблицы содержит данные об одном объекте (например, клиенте, автомобиле, документе), а столбцы таблицы содержат различные характеристики этих объектов – атрибуты (например, наименования и адреса клиентов, марки и цены автомобилей). Строки таблицы называются записями, все записи имеют одинаковую структуру – они состоят из полей, в которых хранятся атрибуты объекта. Каждое поле в записи содержит одну характеристику объекта и имеет строго определённый тип данных (например, текстовая строка, число, дата). Все записи имеют одни и те же поля, только в них содержатся разные значения атрибутов.

Любая СУБД позволяет выполнять четыре простейшие операции с данными:

* добавить в таблицу одну или несколько записей;
* удалить из таблицы одну или несколько записей;
* обновить значения некоторых полей в одной или нескольких записях;
* найти одну или несколько записей, удовлетворяющих заданному условию.

Для выполнения этих операций используется механизм запросов. Результатом выполнения запросов является либо отобранное по определённым критериям множество записей, либо изменение в таблицах.

1. ТЕОРИТИЧЕСКИЙ РАЗДЕЛ «Нормализация модели»
   1. Обзор понятий «Нормализация модели»

Нормализация – это процесс преобразования отношений базы данных к виду, отвечающему нормальным формам, процесс проверки и реорганизации сущностей и атрибутов с целью удовлетворения требований к реляционной модели данных. Нормализация предназначена для приведения структуры базы данных к виду, обеспечивающему минимальную логическую избыточность, и не имеет цель уменьшения или увеличения производительности работы или же уменьшение или увеличение физического объёма БД. Конечной целью нормализации является уменьшение потенциальной противоречивости хранимой в БД информации. В процессе нормализации элементы данных группируются в таблицы, представляющие объекты и их взаимосвязи. Теория нормализации основана на том, что определенный набор таблиц обладает лучшими свойствами при включении, модификации и удалении данных, чем все остальные наборы таблиц, с помощью которых могут быть представлены те же данные. Введение нормализации отношений при разработке информационной модели обеспечивает минимальный объем физической, то есть записанной на каком-либо носителе БД и ее максимальное быстродействие, что впрямую отражается на качестве функционирования информационной системы.

Общее назначение процесса нормализации заключается в следующем:

* исключение некоторых типов избыточности;
* устранение некоторых аномалий обновления;
* разработка проекта базы данных, который является достаточно «качественным» представлением реального мира, интуитивно понятен и может служить хорошей основой для последующего расширения;
* упрощение процедуры применения необходимых ограничений целостности.

Устранение избыточности производится, как правило, за счёт декомпозиции отношений таким образом, чтобы в каждом отношении хранились только первичные факты (то есть факты, не выводимые из других хранимых фактов).

Притом, что идеи нормализации весьма полезны для проектирования баз данных, они отнюдь не являются универсальным или исчерпывающим средством повышения качества БД. Это связано с тем, что существует слишком большое разнообразие возможных ошибок и недостатков в структуре БД, которые нормализацией не устраняются. Несмотря на эти рассуждения, теория нормализации является очень ценным достижением реляционной теории и практики, поскольку она даёт научно строгие и обоснованные критерии качества проекта БД и формальные методы для усовершенствования этого качества. Этим теория нормализации резко выделяется на фоне чисто эмпирических подходов к проектированию, которые предлагаются в других моделях данных. Более того, можно утверждать, что во всей сфере информационных технологий практически отсутствуют методы оценки и улучшения проектных решений, сопоставимые с теорией нормализации реляционных баз данных по уровню формальной строгости.

Теория нормализации реляционных баз данных была разработана в конце 70-х годов 20 века. Согласно ей, выделяются шесть нормальных форм, пять из которых так и называются: первая, вторая, третья, четвертая, пятая нормальная форма, а также нормальная форма Бойса-Кодда, находящаяся между третьей и четвертой.

Нормальная форма — свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, которая потенциально может привести к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

* 1. Типы «Нормальных форм»
* **первая нормальная форма:**

Отношение находится в первой нормальной форме тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов.

В реляционной модели отношение всегда находится в первой нормальной форме по определению понятия отношение.

Что же касается различных таблиц, то они могут не быть правильными представлениями отношений и, соответственно, могут не находиться в 1NF. В соответствии с определением К. Дж. Дейта для такого случая, таблица нормализована (эквивалентно — находится в первой нормальной форме) тогда и только тогда, когда она является прямым и верным представлением некоторого отношения. Конкретнее, рассматриваемая таблица должна удовлетворять следующим пяти условиям:

1. нет упорядочивания строк сверху-вниз.
2. нет упорядочивания столбцов слева-направо.
3. нет повторяющихся строк.
4. каждое пересечение строки и столбца содержит ровно одно значение из соответствующего домена (и больше ничего).
5. все столбцы являются обычными.

«Обычность» всех столбцов таблицы означает, что в таблице нет «скрытых» компонентов, которые могут быть доступны только в вызове некоторого специального оператора взамен ссылок на имена регулярных столбцов, или которые приводят к побочным эффектам для строк или таблиц при вызове стандартных операторов. Таким образом, например, строки не имеют идентификаторов кроме обычных значений потенциальных ключей (без скрытых «идентификаторов строк» или «идентификаторов объектов»). Они также не имеют скрытых временных меток.

* **вторая нормальная форма:**

Отношение находится во второй нормальной форме, если оно находится в первой нормальной форме, и при этом любой его атрибут, не входящий в состав потенциального ключа, функционально полно зависит от каждого потенциального ключа. Функционально полная зависимость означает, что атрибут функционально зависит от всего составного потенциального ключа, но при этом не находится в функциональной зависимости от какой-либо из входящих в него частей. Или другими словами: в 2NF нет неключевых атрибутов, зависящих от части составного потенциального ключа.

Второе, важное значение, второй нормальной формы состоит в том, что она по определению запрещает наличие неключевых атрибутов, которые вообще не зависят от потенциального ключа. Таким образом, 2NF запрещает создавать отношения как несвязанные (хаотические, случайные) наборы атрибутов.

* **третья нормальная форма:**

Согласно определению Кодда, таблица находится в 3NF тогда и только тогда, когда выполняются следующие условия:

1. отношение R (таблица) находится во второй нормальной форме;
2. каждый непервичный атрибут R находится в нетранзитивной (то есть прямой) зависимости от каждого ключа R.

Непервичный (неключевой) атрибут R — это атрибут, который не принадлежит ни одному из возможных (альтернативных) ключей R. Транзитивная зависимость — это функциональная зависимость, при которой X → Z (X определяет Z) не напрямую, а посредством отношения X → Y и Y → Z (отношение Y → X не является обязательным условием).

Определение 3NF, эквивалентное определению Кодда, но по-другому сформулированное, дал Карло Заниоло в 1972 году. Согласно ему, таблица находится в 3NF тогда и только тогда, когда для каждой из ее функциональных зависимостей X → A выполняется хотя бы одно из следующих условий:

1. Х содержит А (то есть X → A — тривиальная функциональная зависимость);
2. Х — суперключ;
3. А — первичный атрибут (то есть А входит в состав альтернативного ключа).

Определение Заниоло четко определяет разницу между 3NF и более строгой нормальной формой Бойса-Кодда (НФБК): НФБК исключает третье условие («А — первичный атрибут»).

* **нормальная форма Бойса-Кодда:**

Отношение находится в BCNF тогда и только тогда, когда каждая его нетривиальная и неприводимая слева функциональная зависимость имеет в качестве своего детерминанта некоторый потенциальный ключ.

Менее формально, переменная отношения находится в нормальной форме Бойса-Кодда тогда и только тогда, когда детерминанты всех ее функциональных зависимостей являются потенциальными ключами.

Для определения BCNF следует понимать понятие функциональной зависимости атрибутов отношения.

Пусть R является переменной отношения, а X и Y — произвольными подмножествами множества атрибутов переменной отношения R. Y функционально зависимо от X тогда и только тогда, когда для любого допустимого значения переменной отношения R, если два кортежа переменной отношения R совпадают по значению X, они также совпадают и по значению Y. Подмножество X называют детерминантом, а Y — зависимой частью.

Функциональная зависимость тривиальна тогда и только тогда, когда ее правая (зависимая) часть является подмножеством ее левой части (детерминанта).

Ситуация, когда отношение будет находиться в 3NF, но не в BCNF, возникает, например, при условии, что отношение имеет два (или более) потенциальных ключа, которые являются составными и имеют общий атрибут. На практике такая ситуация встречается достаточно редко, для всех прочих отношений 3NF и BCNF эквивалентны.

* **четвертая нормальная форма:**

Отношение находится в 4NF, если оно находится в НФБК и не содержит нетривиальных многозначных зависимостей. То есть все многозначные зависимости являются, по сути, функциональными зависимостями от ключей отношения.

* **пятая нормальная форма:**

Отношение находится в пятой нормальной форме (иначе — в проекционно-соединительной нормальной форме) тогда и только тогда, когда каждая нетривиальная зависимость соединения в нём определяется потенциальным ключом (ключами) этого отношения.

При этом:

1. зависимость соединения \*{A, B,..., Z} является тривиальной тогда и только тогда, когда по крайнем мере одно из подмножеств A, B,..., Z множества атрибутов является множество всех атрибутов отношения;
2. зависимость соединения \*{A, B,..., Z} определяется потенциальным ключом (ключами) тогда и только тогда, когда каждое из подмножеств A, B,..., Z множества атрибутов является суперключом отношения.

Условие «каждое из подмножеств A, B,..., Z множества атрибутов является суперключом отношения» можно эквивалентно сформулировать так: «каждое из подмножеств A, B,..., Z множества атрибутов включает некоторый потенциальный ключ отношения».

1. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ «Турфирма»
   1. Анализ предметной области и составление спецификаций данных

В данном курсовом проекте в качестве предметной области рассматривается туристическая фирма. Наша база данных решает следующие задачи: учёт заказов, представление данных о доступных турах, услугах и клиентах.

База данных «Турфирма» позволяет упорядочить работу туристического агентства: все туры, заказы, данные туристов и т.д. хранятся в электронном виде в одном файле на компьютере и легко могут быть найдены и воспроизведены на бумаге. База данных позволяет печатать данные о путешествии на типографском бланке – нет необходимости заполнять бланк от руки, намного облегчается ведение и анализ клиентской базы, упрощается работа и расчеты с клиентами:  
расчетные документы (например, приходные кассовые ордера) легко могут быть оформлены и учтены. Данные клиента и информация о путешествии вносятся в базу один раз и используются для оформления всех необходимых документов: туров, заказов и т.д. При этом значительно уменьшается вероятность ошибок при заполнении документов (например, ошибок в ФИО туристов или адресных данных). База данных позволяет легко собирать и анализировать статистические данные о количестве и характере путешествий туристов, качестве работы менеджеров агентства, популярности туристических маршрутов, отелей и т.д.

Спроектируем базу данных, содержащую информацию о турах, странах, клиентах, заказах и предоставляемых услугах. При проектировании таблиц лучше разработать структуру на бумаге и только затем начинать работу с СУБД Access. При проектировке таблиц, рекомендуется руководствоваться следующими основными принципами:

* не должно быть повторений между таблицами. Когда определенная информация храниться только в одной таблице, то и изменять ее придется только в одном месте. Это делает работу более эффективной, а также исключает возможность несовпадения информации в разных таблицах. Например, в одной таблице должны содержаться адреса и фамилии клиентов;
* каждая таблица должна содержать информацию только на одну тему. Сведения на каждую тему обрабатываются намного легче, если содержаться они в независимых друг от друга таблицах. Например, адреса и заказы клиентов хранятся в разных таблицах, с тем, чтобы при удалении заказа информация о клиенте осталась в базе данных.

Каждая таблица содержит информацию на отдельную тему, а каждое поле в таблице содержит отдельные сведения по теме таблицы. Например, в таблице с данными о поставщиках могут содержаться поля с названием компании, адресом и номером телефона. При разработке полей для каждой таблицы необходимо помнить:

* каждое поле должно быть связано с темой таблицы;
* не рекомендуется включать в таблицу данные, которые являются результатом выражения;
* в таблице должна присутствовать вся необходимая информация;
* информацию следует разбивать на наименьшие логические единицы.

В процессе анализа предметной области создано пять таблиц: клиенты, заказы, услуги, туры и страны. В таблице клиенты получилось шесть полей: фамилия, имя, отчество, код клиента, адрес и телефон. В таблице заказы получилось девять полей: код заказа, код тура, код клиента, дата отправления, дата прибытия, стоимость тура, тип питания, тип проживания и транспорт. В таблице услуги получилось два поля: название услуги и код поля. В таблице страны получилось также два поля: название страны и наличие посольства РБ. В таблице туры получилось семь полей: код тура, название страны, курорт, визовое обслуживание, количество дней, экскурсии и код услуги.

* 1. Информационно-логическая модель базы данных «Турфирма»

Первым этапом и самым главным этапом в процессе проектирования и создания базы данных, является разработка инфологической модели. Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

В процессе создания ER-диаграммы базы данных «Учебный процесс», необходимо проанализировать все связи между таблицами. При анализе таблиц становится известно, что заказы осуществляют клиенты и каждый клиент может сделать несколько заказов. Получаем связь «ОДИН-КО-МНОГИМ» между таблицами Клиенты и Заказы по полю «Код клиента». Продолжая анализировать таблицы, выяснили, что один и тот же тур могут заказать несколько раз, тогда получим вторую связь, также «ОДИН-КО-МНОГИМ», только между таблицами Туры и Заказы по полю «Код тура». Следующей выявленной связью оказалась связь «ОДИН-КО-МНОГИМ» между таблицами Туры и Страны по полю «Название страны», так как имеет место такое утверждение, что может быть несколько туров в одну и ту же страну. Итак, осталась одна несвязанная таблица – это таблица Услуги, проанализировав её и таблицу Туры, становится ясно, что неизвестное количество туров может быть с одной и той же услугой, то есть мы, снова, получаем связь «ОДИН-КО-МНОГИМ», но в этот раз уже между таблицами Туры и Услуги по полю «Код услуги». Таким образом мы получили четыре связи «ОДИН-КО-МНОГИМ» между таблицами в базе данных. Так же существуют и другие связи, к примеру, такие как «ОДИН К ОДНМУ», то есть одной записи в первой таблице (ключу) соответствует единственная запись во второй таблице. Данный вид связи нарушает логическую целостность данных, но позволяет разбить одну большую таблицу на несколько маленьких, при этом, обычно, поля связи называются одинаково. В разбитом виде для человека информация воспринимается легче, «МНОГИЕ-КО-МНОГИМ» - абстрактный вид связи, когда в первой таблице множеству записей соответствует множество записей связанной таблицы. Такой вид связи используется редко, так как в реальной жизни связь многие ко многим редко встречается и фактически трактует отсутствие логической связи.

Реализованную информационно-логическую модель базы данных «Турфирма» можно увидеть на рисунке 1.

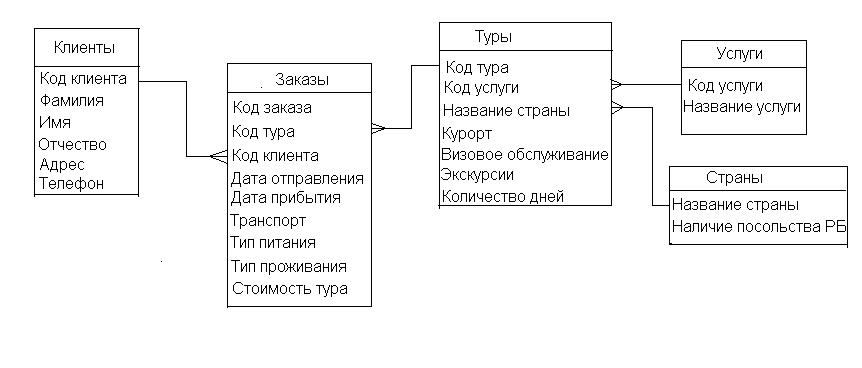


Рисунок 1 – Структурная схема базы данных «Учебный процесс»

* 1. Физическая модель базы данных «Турфирма»

На основании проведенного анализа предметной области и полученных спецификаций данных была переходим к созданию физической модели базы данных «Турфирма». Для создания базы данных нужно сначала запустить программу Microsoft Office Access 2007, для этого нужно проделать следующие действия: Пуск → Программы → Microsoft Office → Microsoft Office Access 2007. После этого откроется окно программы. Для создания новой базы данных нужно нажать кнопку «office» → Создать, выбираем «Новая база данных», вводим название базы данных и нажимаем кнопку «Создать». На основании созданных в п. 2.1 таблиц, создаём их в базе данных. Для этого нужно нажать на вкладку Создание → Конструктор таблиц.

Каждое поле таблицы характеризуется наименованием, типом и шириной поля. При задании типа дан­ных поля можно также указать размер, формат и другие параметры, влияющие на отображение значения поля и точность числовых данных. Определим, какие типы данных будут присутствовать в наших таблицах. По окончании заполнения через кнопку «office» → Сохранить как сохраняем полученную таблицу с требуемым именем. На рисунке 1 в Приложении А можно увидеть таблицы базы данных «Турфирмы» в режиме конструктора. Далее заполняем таблицы необходимыми данными.

Разработаем схему данных, для этого:

1. Нажимаем на вкладку Работа с базами данных, там жмем по кнопку Кнопка_19 «Схема данных» на панели инструментов. На экране появится окно «Схема данных»;
2. Щёлкаем по кнопке Кнопка_20 «Отобразить таблицы» на панели инструментов**.** На экране появится окно «Добавление таблицы»;
3. Выбираем таблицу и нажимаем на кнопку «Добавить», выбираем следующую таблицу и нажимаем на кнопку «Добавить». Аналогично добавляем оставшиеся таблицы;
4. Закрываем окно, щелкнув по кнопке «3акрыть»;
5. Создадим связь между таблицами Клиентыи Заказы*.* Для этого курсором мыши перетаскиваем поле «Код клиента» в таблице Клиентына поле «Код клиента» в таблицу Заказы*.* На экране откроется окно «Изменение связей»;
6. Устанавливаем флажки в свойствах:

* обеспечение целостности данных
* каскадное обновление связанных полей
* каскадное удаление связанных записей

1. Нажимаем на кнопку «Создать». Связь будет создана;
2. Аналогично создаем связи между полем «Код тура» в таблице Турыи полем «Код тура» в таблице Заказыи так далее;
3. Закрываем окно схемы данных, ответив *«Да»* на вопрос о сохранении макета.
4. На рисунке 1 в Приложении Б можно увидеть полученную схему базы данных «Турфирма».
5. СОЗДАНИЕ ЗАПРОСОВ, ОТЧЕТОВ, ФОРМ БАЗЫ ДАННЫХ «Турфирма»
   1. Создание запросов

Запросы позволяют выбирать данные из одной или нескольких связанных таблиц. Результатом выполнения запроса является результирующая таблица, которая наряду с другими таблицами может быть использована при обработке данных. С помощью запросов можно также обновлять, удалять или добавлять данные в таблицы. Основным назначением запросов является отбор данных по критериям поиска.

Создадим запрос на выборку об информации, о заказах. Для создания этого запроса необходимо: перейти на вкладку Создание → Конструктор запросов → в появившемся окне Добавление таблиц выбрать таблицу Заказы и нажать кнопку «Добавить», таким же образом выбрать таблицы: Туры, Клиенты и Страны → выбираем нужные поля из таблиц (Код заказа, Код тура и Код клиента из таблицы Заказы, Фамилия, Имя и Отчество из таблицы Клиенты, Название услуги из таблицы Услуги и Название страны и Курорт из таблицы Туры), перетягиваем их курсором мыши → в столбце Код заказа, в поле Условие отбора требуется ввести «[Введите код заказа]». При выполнении запроса (значок Кнопка_17 на панели управления) появится окно с предложением ввести код заказа для того, чтобы узнать информацию о том заказе, о котором нужно. Результат запроса и запрос в режиме конструктора можно просмотреть на рисунке 1 в Приложении В.

Создадим запрос на добавление информации о новых заказах в таблицу заказы. Для создания этого запроса необходимо: создать таблицу Новые заказы по примеру создания таблицы Заказы → перейти на вкладку Создание → Конструктор запросов → в появившемся окне Добавление таблиц выбрать таблицу Новые заказы и нажать кнопку «Добавить» → выбирать поля и курсором мыши перетаскивать их по очереди → изменить на панели инструментов вид запроса с выборки на добавление → в появившемся окне Добавление, в поле Имя таблицы, выбрать таблицу Заказа и нажать кнопку «OK». Результат запроса и запрос в режиме конструктора можно просмотреть на рисунке 2 в Приложении В.

Создадим запрос на обновление: десятипроцентная скидка на все заказы стоимостью до 3 миллионов рублей. Для создания этого запроса необходимо: перейти на вкладку Создание → Конструктор запросов → в появившемся окне Добавление таблиц выбрать таблицу Заказы и нажать кнопку «Добавить» → выбрать поле Стоимость тура и перетащить его курсором мыши → изменить на панели инструментов вид запроса с выборки на обновление → в столбце Стоимость тура, в строке Обновление нужно ввести выражение: «[Стоимость тура]\*0,9» → в столбце Стоимость тура, в строке Условие отбора нужно ввести выражение: «<3000000». Результат запроса и запрос в режиме конструктора можно просмотреть на рисунке 3 в Приложении В.

Создадим запрос на добавление таблицы экскурсионных туров по Европе. Для создания этого запроса необходимо: перейти на вкладку Создание → Конструктор запросов → в появившемся окне Добавление таблиц выбрать таблицу Туры и нажать кнопку «Добавить» → выбрать нужные поля (Код тура, Код услуги, Количество дней, Название страны и Курорт) → изменить на панели инструментов вид запроса с выборки на создание таблицы → в появившемся окне Создание таблицы, в поле Имя таблицы, напечатать имя таблицы и нажать кнопку «OK». Результат запроса и запрос в режиме конструктора можно просмотреть на рисунке 4 в Приложении В.

Создадим запрос на удаление информации о бывших клиентах из таблицы Клиенты. Для создания этого запроса необходимо: перейти на вкладку Создание → Конструктор запросов → в появившемся окне «Добавление таблиц» выбрать таблицу Клиенты и нажать кнопку «Добавить» → выбрать поля и курсором мыши перетащить их по очереди → изменить на панели инструментов вид запроса с выборки на удаление → в столбце Код клиента, в строке Условие отбора нужно ввести выражение: «<6». Результат запроса и запрос в режиме конструктора можно просмотреть на рисунке 5 в Приложении В.

Создадим перекрестный запрос о средней стоимости туров. Для создания этого запроса необходимо: перейти на вкладку Создание → Конструктор запросов → в появившемся окне Добавление таблиц выбрать таблицу Заказы и нажать кнопку «Добавить» → выбрать поля (Название страны из таблицы Туры, Название услуги из таблицы Услуги и Стоимость тура из таблицы Заказы) и курсором мыши перетащить их по очереди → на панели инструментов нажать кнопку Итоги → изменить на панели инструментов вид запроса с выборки на перекрестный → в столбце Название страны, в строке Перекрестная таблица выбрать «Заголовки строк» → в столбце Название услуги, в строке Перекрестная таблица выбрать «Заголовки столбцов» → в столбце Стоимость тура, в строке Перекрестная таблица выбрать «Значение» → в столбце Название страны, в строке Групповая операция выбрать функцию «Avg».

* 1. Создание форм

Формы служат для ввода и просмотра данных в удобном для пользователя виде, который соответствует привычному для него документу. При выводе данных с помощью форм можно применять специальные средства оформления.

Создадим форму «Услуги». Для создания этой формы необходимо:

1. перейти на вкладку Создание;
2. нажать кнопку «Конструктор форм»;
3. нажать левую кнопку мыши в «Области данных» и выбрать пункт «Заголовок/примечание формы»;
4. на панели инструментов нажать кнопку «Надпись» и курсором мыши добавить надпись «Услуги» в Заголовок формы;
5. на панели инструментов нажать кнопку «Рисунок», курсором мыши добавить его в Заголовок отчета, в появившемся окне выбрать рисунок из компьютера и нажать кнопку «ОК»;
6. на панели инструментов нажать кнопку «Рисунок», курсором мыши добавить его в Область данных, в появившемся окне выбрать рисунок из компьютера и нажать кнопку «ОК»;
7. на панели инструментов нажать кнопку «Добавить поле», курсором мыши перетащить нужные поля (Код услуги и Название услуги из таблицы Услуги) в Область данных;
8. на панели инструментов нажать кнопку «Кнопка», курсором мыши добавить кнопки в Заголовок отчёта, в появившемся окне выбрать «Переход по записям → Предыдущая / Следующая запись»;
9. Оформить форму (цвет фона, текста и т.д.).

Полученную форму и форму в режиме конструктора можно просмотреть на рисунке 1 в Приложении Г.

Создадим форму «Услуги» на основе запроса. Для создания этой формы необходимо:

1. Создать запрос «Для формы Заказы», по примеру создания запроса на выборку, из таблиц выбрать нужные поля(Код тура, Код заказа, Стоимость тура из таблицы Заказы и Название страны из таблицы Туры);
2. перейти на вкладку Создание;
3. нажать кнопку «Конструктор форм»;
4. нажать левую кнопку мыши в «Области данных» → выбрать пункт «Свойства» → в появившемся окне выбрать свойства формы, выбрать раздел «данные» → нажать на поле «Источники данных» → в появившемся окне выбрать запрос «Заказы» и, курсором мыши, перетащить все поля;
5. на панели инструментов нажать кнопку «Добавить поле» и, курсором мыши, перетащить все поля;
6. на панели инструментов нажать кнопку «Кнопка» → курсором мыши добавить кнопки в Область данных → в появившемся окне выбрать нужные функции для кнопок;
7. на панели инструментов нажать кнопку «Диаграмма» → курсором мыши добавить ее в Область данных → в появившемся окне выбрать запрос «Для формы Заказы» → выбрать нужные доступные поля (Код заказа, Название страны, Стоимость тура) → выбрать тип диаграммы → нажать кнопку «Готово»;
8. Оформить форму (цвет фона, текста и т.д.).

Полученную форму и форму в режиме конструктора можно просмотреть на рисунке 2 в Приложении Г.

* 1. Создание отчетов

Отчёты предназначены для формирования выходных документов и вывода их на печать. По своим свойствам и структуре отчёты во многом подобны формам. Основное их отличие заключается в том, что в отчёте отображаются все данные и в них предусмотрена возможность группировать данные по различным критериям. Отчёты в отличие от форм могут содержать специальные элементы оформления, характерные для печати документов: колонтитулы, номера страниц и т.д.

Создадим отчет по заказам. Для создания этого отчета необходимо:

1. перейти на вкладку Создание;
2. нажать кнопку «Конструктор форм»;
3. нажать левую кнопку мыши в «Области данных» и выбрать пункт «Заголовок/примечание отчёта»;
4. нажать левую кнопку мыши в «Области данных» → выбрать пункт «Свойства» → в появившемся окне выбрать свойства отчета, выбрать раздел «данные» → нажать на поле «Источники данных» → в появившемся окне выбрать таблицы Туры и Заказы и, курсором мыши, перетащить все нужные поля (Название страны из таблицы Туры и все поля из таблицы Заказы);
5. нажать на кнопку «Группировка» → выбрать «Название страны»;
6. в окне Группировка нажать на кнопку «Больше» → выбрать в пункте «С разделом примечания»;
7. на панели инструментов нажать кнопку «Надпись» и курсором мыши добавить надпись «Отчёт по заказам» в Заголовок отчёта;
8. на панели инструментов нажать кнопку «Добавить поле» → курсором мыши перетащить поле Название страны в раздел Заголовок группы «Название страны» → название этого поля вырезать и вставить в раздел Верхний колонтитул;
9. на панели инструментов нажать кнопку «Добавить поле» → курсором мыши перетащить все оставшиеся поля в раздел Область данных → название этих полей вырезать и вставить в раздел Верхний колонтитул;
10. на панели инструментов нажать кнопку «Поле» → курсором мыши добавить поле в раздел Примечание группы «Название страны» → в поле через свойства в строке Данные открыть построитель выражений → в построителе выражений ввести «=Count([Код заказа])» → в названии поля ввести «Всего заказов:»;
11. на панели инструментов нажать кнопку «Поле» → курсором мыши добавить поле в раздел Нижний колонтитул → в поле через свойства в строке Данные открыть построитель выражений → в построителе выражений ввести «=Now()» → название поля удалить;
12. на панели инструментов нажать кнопку «Поле» → курсором мыши добавить поле в раздел Нижний колонтитул → в поле через свойства в строке Данные открыть построитель выражений → в построителе выражений ввести «="Страница " & [Page] & " из " & [Pages]» → название поля удалить;
13. на панели инструментов нажать кнопку «Поле» → курсором мыши добавить поле в раздел Примечание отчета → в поле через свойства в строке Данные открыть построитель выражений → в построителе выражений ввести «=Count([Код заказа])» → в названии поля ввести «Всего заказов:»;
14. Оформить отчет (цвет фона, текста и т.д.).

Полученный отчет и отчет в режиме конструктора можно просмотреть на рисунке 1 в Приложении Д.

Создадим отчет по налогам с продам. Для создания этого отчета необходимо:

1. перейти на вкладку Создание;
2. нажать кнопку «Конструктор форм»;
3. нажать левую кнопку мыши в «Области данных» и выбрать пункт «Заголовок/примечание отчёта»;
4. нажать левую кнопку мыши в «Области данных» → выбрать пункт «Свойства» → в появившемся окне выбрать свойства отчета, выбрать раздел «данные» → нажать на поле Источники данных → в появившемся окне выбрать таблицу Заказы и, курсором мыши, перетащить все нужные поля (Код тура, Код заказа, Кож клиента, Стоимость тура из таблицы Заказы);
5. на панели инструментов нажать кнопку «Надпись» и курсором мыши добавить надпись «Налоги с продаж» в Заголовок отчёта;
6. на панели инструментов нажать кнопку «Добавить поле» → курсором мыши перетащить все поля в раздел Область данных → название этих полей вырезать и вставить в раздел Верхний колонтитул;
7. на панели инструментов нажать кнопку «Поле» → курсором мыши добавить поле в раздел Область данных → в поле через свойства в строке Данные открыть построитель выражений → в построителе выражений ввести «=IIf ([Стоимость тура]<2000000;100000;IIf([Стоимость тура]<3000000; 150000; 200000))» → название поля вырезать и вставить в раздел Верхний колонтитул;
8. на панели инструментов нажать кнопку «Поле» → курсором мыши добавить поле в раздел Нижний колонтитул → в поле через свойства в строке Данные открыть построитель выражений → в построителе выражений ввести «=Now()» → название поля удалить;
9. на панели инструментов нажать кнопку «Поле» → курсором мыши добавить поле в раздел Нижний колонтитул → в поле через свойства в строке Данные открыть построитель выражений → в построителе выражений ввести «="Страница " & [Page] & " из " & [Pages]» → название поля удалить;
10. на панели инструментов нажать кнопку «Поле» → курсором мыши добавить поле в раздел Примечание отчета → в поле через свойства в строке Данные открыть построитель выражений → в построителе выражений ввести «=IIf ([Стоимость тура]<2000000;100000;IIf([Стоимость тура]<3000000; 150000; 200000))» → в названии поля ввести «Всего налог с продаж:»;
11. Оформить отчет (цвет фона, текста и т.д.).

Полученный отчет и отчет в режиме конструктора можно просмотреть на рисунке 2 в Приложении Д.

Создадим отчет по клиентам на основе запроса «Для отчета клиенты». Для создания этого отчета необходимо:

1. создать запрос «Для отчета клиенты» по примеру создания запроса «Клиенты»
2. перейти на вкладку Создание;
3. нажать кнопку «Конструктор форм»;
4. нажать левую кнопку мыши в «Области данных» и выбрать пункт «Заголовок/примечание отчёта»;
5. нажать левую кнопку мыши в «Области данных» → выбрать пункт «Свойства» → в появившемся окне выбрать свойства отчета, выбрать раздел «данные» → нажать на поле «Источники данных» → в появившемся окне выбрать запрос Клиенты и, курсором мыши, перетащить все нужные поля (Фамилия, Имя, Отчество, Адрес, Телефон из таблицы Клиенты);
6. на панели инструментов нажать кнопку «Надпись» и курсором мыши добавить надпись «Клиенты» в Заголовок отчёта;
7. на панели инструментов нажать кнопку «Добавить поле» → курсором мыши перетащить все поля в раздел Область данных → название этих полей вырезать и вставить в раздел Верхний колонтитул;
8. на панели инструментов нажать кнопку «Поле» → курсором мыши добавить поле в раздел Нижний колонтитул → в поле через свойства в строке Данные открыть построитель выражений → в построителе выражений ввести «=Now()» → название поля удалить;
9. на панели инструментов нажать кнопку «Поле» → курсором мыши добавить поле в раздел Нижний колонтитул → в поле через свойства в строке Данные открыть построитель выражений → в построителе выражений ввести «="Страница " & [Page] & " из " & [Pages]» → название поля удалить;
10. на панели инструментов нажать кнопку «Поле» → курсором мыши добавить поле в раздел Примечание отчета → в поле через свойства в строке Данные открыть построитель выражений → в построителе выражений ввести «=Count([Фамилия])» → в названии поля ввести «Всего клиентов:»;
11. Оформить отчет (цвет фона, текста и т.д.).

Полученный отчет и отчет в режиме конструктора можно просмотреть на рисунке 3 в Приложении Д.

* 1. Создание кнопочной формы

Для создания кнопочной формы необходимо выполнить следующие действия:

1. перейти на вкладку Работа с базами данных;
2. нажать кнопку «Диспетчер кнопочных форм»;
3. в появившемся окне нажать кнопку «Создать» → в появившемся окне ввести название кнопочной формы «Турфирма «День и ночь»»;
4. в появившемся окне нажать кнопку «Создать» → в появившемся окне ввести название кнопочной формы «Формы»;
5. в появившемся окне нажать кнопку «Создать» → в появившемся окне ввести название кнопочной формы «Отчеты»;
6. выбрать кнопочную форму «Турфирма «День и ночь»» и нажать кнопку «изменить»;
7. в появившемся окне нажать кнопку «Создать» → в появившемся окне в строке Текст ввести «Формы», в строке Кнопочная форма выбрать кнопочную форму «Формы» и нажать «ОК»;
8. по примеру пункта 7 создать ещё две кнопки: «Отчёты» и «Выход»;
9. по примеру пунктов 6 и 7 изменить кнопочные формы «Формы» и «Отчеты» и создать в них кнопки «Предоставляемые услуги», «Принятые заказы», «Назад», «Выход» и «Заказы», «Налоги с продаж», «Клиенты», «Назад» и «Выход» соответственно;
10. Нажать кнопку «Закрыть» → в появившемся окне выбрать кнопочную форму «Турфирма «День и ночь» и нажать кнопку «По умолчанию»;
11. Оформить Кнопочную форму в режиме конструктора. Кнопочную форму можно посмотреть на рисунке 1 в Приложении Е.

# ЗАКЛЮЧЕНИЕ

База данных – это совокупность структурированных и взаимосвязанных данных и методов, обеспечивающих добавление выборку и отображение данных. Microsoft Access позволяет управлять всеми сведениями из одного файла базы данных. В рамках этого файла используются следующие объекты:

* таблицы для сохранения данных;
* запросы для поиска и извлечения только требуемых данных;
* формы для просмотра, добавления и изменения данных в таблицах;
* отчеты для анализа и печати данных в определенном формате.

Удачная разработка базы данных обеспечивает простоту ее поддержания.

В ходе проделанной работы, нами были созданы:

* 9 таблиц;
* 8 запросов;
* 3 формы;
* 3 отчета.

Microsoft Access, обладая всеми чертами классической СУБД, предоставляет и дополнительные возможности. Access - это не только мощная, гибкая и простая в использовании СУБД, но и система для разработки работающих с базами данных приложений. С помощью Access можно создать приложение, работающее в среде Windows и полностью соответствующее потребностям по управлению данными. Используя запросы, есть возможность выбирать и обрабатывать хранящуюся в таблицах информацию. Можно создавать формы для ввода, просмотра и обновления данных, а также использовать Access для создания как простых, так и сложных отчетов. Формы и отчеты «наследуют» свойства базовой таблицы или запроса, так что в большинстве случаев необходимо указать форматы, условия на значения и некоторые другие характеристики данных только один раз. К числу наиболее мощных средств Access относятся средства разработки объектов - Мастера, которые можно использовать для создания таблиц, запросов различных типов форм и отчетов, просто выбрав с помощью мыши нужные опции.

С помощью Access также легко импортировать данные из текстовых файлов, документов текстовых процессоров и электронных таблиц (и экспортировать данные в них).

Microsoft Access имеет множество областей применения: на малом предприятии, корпорации и даже в сфере домашнего применения.

# ЛИТЕРАТУРА

|  |
| --- |
| 1. Домашняя страница Access: справка и обучение http://office.microsoft.com/ru-ru/access/FX100487571049.aspx |
| 1. Мейер М. Теория реляционных баз данных. – М.: Мир, 1987. |
| 1. Основы проектирования реляционных баз данных. Электронное учебное пособие. |
| 1. Симонович С. В., Евсеев Г. А., Алексеев А. Г. Специальная информатика: Учебное пособие. – М.: АСТ-ПРЕСС: Инфорком-Пресс, 2000 |
| 1. Хаббард Дж. Автоматизированное проектирование баз данных. – М.: Мир, 1984. 2. Бондарева Г.А., Сахарова Е.В., Королькова Л.Н., Информатика. Методические указания. Ставрополь, СТИС, 2006 3. Гончаров А. Ю. Access 2003. Самоучитель с примерами., Москва, 2004г. 4. Григорьев В.А., Ревунков В.И. Банки данных. Учебник для вузов. – М., МВТУ им.Баумана, 2002.  ПРИЛОЖЕНИЯ ПРИЛОЖЕНИЕ А  рисунок1.JPG  Рисунок 1 – таблицы базы данных «Турфирмы»  ПРИЛОЖЕНИЕ Б  рисунок  1.JPG  Рисунок 2 **–** схема данных базы данных «Турфирмы»  ПРИЛОЖЕНИЕ В  рисунок  1.JPG  Рисунок 1 – Запрос об информации о заказах  рисунок 1.JPG  Рисунок 2 – Запрос на появившиеся новые заказы  рисунок1.JPG  Рисунок 3 – Запрос на 10%-ую скидка на все заказы до 3млн  рисунок1.JPG  Рисунок 4 – Запрос на создание таблицы Экскурсионные туры по Европе  рисунок 1.JPG  Рисунок 5 – Запрос на удаление информации он бывших клиентах  рисунок1.JPG  Рисунок 6 – Запрос о средней стоимость туров  ПРИЛОЖЕНИЕ Г  рисунок1.JPG  Рисунок 1 – Форма «Заказы»  рисунок1.JPG  Рисунок 2 – Форма «Услуги»  ПРИЛОЖЕНИЕ Д  рисунок1.JPG  Рисунок 1 – Отчет «Клиенты»  рисунок1.JPG  Рисунок 2 – Отчет «Налоги с продаж»  рисунок1.JPG  Рисунок 3 – Отчёт «Заказы»  ПРИЛОЖЕНИЕ Е  рисунок.JPG  Рисунок 1 – Кнопочная форма |