



NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

**HOTEL RESERVATION SYSTEM
USING MS-ACCESS**

**Graduation Project
COM-400**

Student: Ahmed IQBAL

Student No: 20021325

Supervisor: Asst. Prof. Dr. Elbrus IMANOV

Nicosia-2006



ACKNOWLEDGMENT

First of all I will thank to mighty ALLAH.

*Second I wish to thanks my supervisor **Asst Prof Dr Elbrus IMANOV** for his in valuable advices, supporting, and encouragements.*

*Third, I would like to express my gratitude to **Mr. Umit ILHAN, Assoc.Prof.Dr. Rahib Abiyev and Mr. Okan donangil** for their excellent treatment during my graduation period.*

*Fourth, I would like to express my gratitude to **NEAR EAST UNIVERSITY** for providing me the opportunity for making my undergraduate program.*

Fifth, to my parent which sported me in all thick and thins they gave me courage in all fields of life s and sibling specially my cute sister Hania Iqbal and my wife

Then to all my friends specially to Mr. Muhammad Shahid NAWAZ for his invaluable efforts and advices during my project and all the other friends to giving me a very good time during my graduation period.

ABSTRACT

In this project, "HOTEL RESERVATION SYSTEM" is developed by using Microsoft Access. Microsoft Access is a database design application which is designed by Microsoft organization.

As we know, we are living in a computer era where almost every thing is operated by computers. Without computers it's very hard to manage the data but now-a-days with the help of computer we have different kind of data base management showers which helps to manage the data and in this project the Microsoft Access is used. It makes the processing of data faster and easier.

In application of design for "HOTEL RESERVATION SYSTEM" which cover all the needs for the hotel business as an administrative employee like 'Manager' which can do all the tasks to make the hotel management easier. This program is able to hold customer records, employee records, accounts; expenses, needs, transaction and we can get the report of all individual sides very easily.

The project uses Microsoft Database Access which is very user friendly, easier and reliable for making a Database.

CONTENTS

Acknowledgement	i
Abstract	ii
Table of contents	iii
Introduction	1
CHAPTER ONE: INTRODUCTION TO DATABASE	2
1.1 De-merits of absence of database	3
1.2 Merits of database	3
1.3 Introduction to database design	3
1.4 Database models	4
1.4.1 Flat Model	5
1.4.2 Network Model	5
1.4.3 Relational Model	5
1.4.3.1 Why we use a Relational Database Design	6
1.5 Relationships between Tables	7
1.5.2 One-To-One Relationships	7
1.5.3 One-To-Many Relationships	7
1.6 Data Modeling	8
1.6.1 Database Normalization	8
1.6.2 Primary Key	9
1.6.3 Foreign Key	10
1.6.4 Compound Key	10
CHAPTER TWO: VISUAL BASIC EDITOR AND STRUCTURED QUERY LANGUAGE	11
2.1 Visual Basic Editor	11
2.2 Structured Query Language	12
2.3 Description of SQL	13
2.4 SQL Keywords	14
2.4.1 Data Retrieval	14
2.4.2 Data Manipulation	15
2.4.3 Data Transaction	15
2.4.4 Data Definition	15
CHAPTER THREE: MICROSOFT ACCESS DATABASE SYSTEM	16
3.1 A Few Terms	16
3.2 Introductory Microsoft Access	17
3.3 Introduction to Tables	18
3.3.1 Table's data types	19
3.3.2 Creating a table using the wizard	22
3.3.3 Create a table with design views:	23
3.3.4 Creating Tables by entering data	24

3.3.5 Entering data in the Table:	25
3.3.5.1 Deleting data from a Table	25
3.3.5.2 Saving data in a Table	25
3.3.5.3 Closing a Table	25
3.4 Creating Forms	26
3.4.1 Create Form in Design View	26
3.4.1.1 Adding Records Using a Form	28
3.4.1.2 Editing a Forms	29
3.4.2 Creating Form by Using Wizard	31
3.5 Queries	34
3.5.1 Create a Query in Design View	35
3.5.2 Query Wizard	37
3.5.2.1 Find Duplicates Query	38
3.5.2.2 Saving a Query	40
3.5.2.3 Closing a Query	40
Chapter four: Description about the project	41
4.1 Starting project by Wizard	41
4.2 Pictorial representation of my project	45
4.3 Main Customer Record Switch Board	48
4.3.1 Customer reporce	52
4.4 Room information switch board	58
4.4.1 Check room availability	60
4.4.2 Reservation Form	63
4.5 Hotel Records	65
4.5.1 Employee Details	65
4.5.2 Check Employee Availability	66
4.5.3 Accounts Record	67
4.5.4 Accounts	69
4.6 Relation between tables	71
Future Developments	72
Conclusion	73
References	74

INTRODUCTION

Microsoft database Access is one of the best database management applications now-a-days. It is easier, reliable and user friendly. Making a database management program is very easy and efficient in Microsoft Access. Managing records, searching records, and adding new records, options are provided in the application and it is very easy for the end user to operate on these options.

The documentation is about the Hotel reservation system program which is made by using Microsoft Access database. The documentation is the full explanation about the program that how we created forms, queries, tables etc.

All the chapters included in the documentation demonstrate the working of my project and how I tackle all the problems while making my project.

Let us go through the overview of all the chapters in brief.

Chapter1: will discuss about the database definitions, merits and demerits, database design, relationship, data modeling etc.

Chapter2: This chapter is about visual basic editor in Microsoft Access, captions, properties, SQL, description of SQL, SQL keys etc.

Chapter3: Is about Microsoft Access, how to open, how create tables, how can create form by three ways in details, data types, and how can create Queries etc.

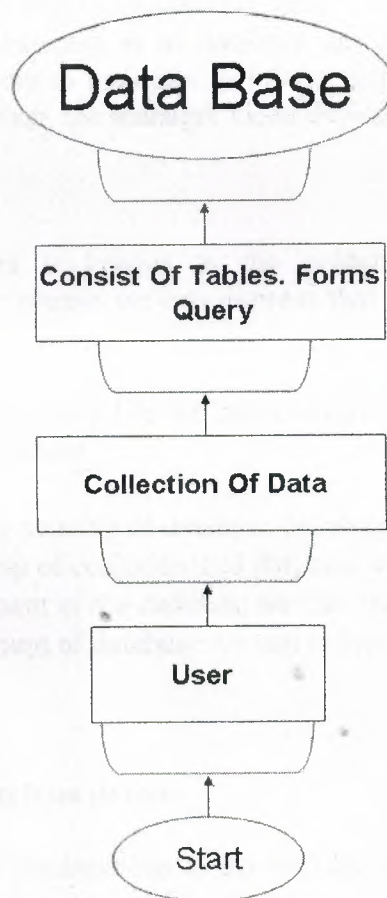
Chapter4: Will discuss about the working of the project, how user can store information, and what is the job of all the forms, and switchboards used in the program and relationship between tables.

CHAPTER ONE: INTRODUCTION TO DATABASE

Every thing around us has a particular identity. To identify anything system, actor or person in words we need a data or information. So this information is valuable and in this advanced era we can store it in database and access this data by the blink of eye.

For an instant if we go through the definitions of database we may find following definitions.

- A **database** is a collection of related information.
- A **database** is an organized body of related information.
- Each database has **objects** such as a table, query, form, or macro



Pictorial definition of the Database

1.1 De-merits of absence of database

A glance on the past will may help us to reveal the drawbacks in case of absence of database.

- In the past when there wasn't proper system of database, Much paper work was need to do and to handle great deal of written paper documentation was giant among the problems itself.
- In the huge networks to deal with equally bulky data, more workers are needed which affidavit cost much labor expanses.
- The old criteria for saving data and making identification was much time consuming such as if we want to search the particular data of a person.
- Before the Development of Computer database it was a great problem to search for some thing. Efforts to avoid the headache of search often results in new establishments of data.
- Before the development of database it seemed very unsafe to keep the worthy information. In Some situation some big organization had to employee the special persons in order to secure the data.
- Before the implementation of database any firm had to face the plenty of difficulties in order to maintain their Management. To hold the check on the expenses of the firm, the manager faced difficulties.

1.2 Merits of database

The modern era *is* known as the golden age computer sciences and technology. In a simple phrase we can express that the modern age is built on the foundation of database.

If we carefully watch our daily life we can examine that some how our daily life is being connected with database.

- There are several benefits of database developments.
- Now with the help of computerized database we can access data in a second.
- By the development of the database we can make data more secure.
- By the development of database we can reduce the cost.

1.3 Introduction to database design

The design of a database has to do with the way data is stored and how that data is related. The design process is performed after you determine exactly what information needs to be stored and how it is to be retrieved.

A collection of programs that enables you to store, modify, and extract information from a database. There are many different types of DBMS ranging from small systems that run on personal computers to huge systems that run on mainframes. The following are examples of database applications:

- Computerized library systems
- Automated teller machines
- Flight reservation systems
- Computerized parts inventory systems

From a technical standpoint, DBMS can differ widely. The terms *relational*, *network*, *flat*, and *hierarchical* all refer to the way a DBMS organizes information internally. The internal organization can affect how quickly and flexibly you can extract information.

Requests for information from a database are made in the form of a *query*.

Database design is a complex subject. A properly designed database is a model of a business, Country Database or some other in the real world. Like their physical model counterparts, *data models enable you to get answers about the facts that make up the objects being modeled. It's the questions that need answers that determine which facts need to be stored in the data model.*

In the relational model, data is organized in tables that have the following characteristics: every record has the same number of facts, every field contains the same type of facts (Data) in each record, and there is only one entry for each fact. *No two records are exactly the same.*

The more carefully you design, the better the physical database meets users' needs. In the process of designing a complete system, you must consider user needs from a variety of viewpoints.

1.4 Database Models

Various techniques are used to model data structures. Certain models are more easily implemented by some types of database management systems than others. For any one logical model various physical implementation may be possible. An example of this is the relational model: in larger systems the physical implementation often has indexes which point to the data; this is similar to some aspects of common implementations of the network model. But in small relational database the data is often stored in a set of files, one per table, in a flat, un-indexed structure. There is some confusion below and elsewhere in this article as to logical data model vs. its physical implementation.

1.4.1 Flat Model

The flat (or table) model consists of a single, two dimensional array of data elements, where all members of a given column are assumed to be similar values, and all members of a row are assumed to be related to one another. For instance, columns for name and password might be used as a part of a system security database. Each row would have the specific password associated with a specific user. Columns of the table often have a type associated with them, defining them as character data, date or time information, integers, or floating point numbers. This model is the basis of the spreadsheet.

1.4.2 Network Model

The network model allows multiple datasets to be used together through the use of pointers (or references). Some columns contain pointers to different tables instead of data. Thus, the tables are related by references, which can be viewed as a network structure. A particular subset of the network model, the hierarchical model, limits the relationships to a tree structure, instead of the more general directed graph structure implied by the full network model.

1.4.3 Relational Model

The relational data model was introduced in an academic paper by E.F. Cod in 1970 as a way to make database management systems more independent of any particular application. It is a mathematical model defined in terms of predicate logic and set theory.

Although the basic idea of a relational database has been very popular, relatively few people understand the mathematical definition and only a few obscure DBMSs implement it completely and without extension. Oracle, for example, can be used in a purely relational way, but it also allow tables to be defined that allow duplicate rows an extension (or violation) of the relational model. In common English usage, a DBMS is called relational if it supports relational operational operations, regardless of whether it enforces strict adherence to the relational model. The following is an

informal, not-technical explanation of how “relational” database management systems commonly work.

A relational database contains multiple tables, each similar to the one in the “flat” database model. However, unlike network databases, the tables are not linked by pointers. Instead, keys are used to match up rows of data in different tables. A key is just one or more columns in one table that correspond to columns in other tables. Any column can be a key, or multiple columns can be grouped together into a single key. Unlike pointers, it’s not necessary to define all the keys in advance; a column can be used as a key even if it wasn’t originally intended to be one.

A key that can be used to uniquely identify a row in a table is called a unique key. Typically one of the unique keys is the preferred way to refer to row; this is defined as the table’s primary key.

When a key consists of data that has an external, real-world meaning (such as a person’s name, a book’s ISBN, or a car’s serial number), it’s called a “natural” key. If no nature key is suitable, an arbitrary key can be assigned (such as by given employees ID numbers). In practice, most databases have both generated and natural keys, because generated keys can be used internally to create links between rows that can’t break, while natural keys can be used, less reliably, for searches and for integration with other databases. (For example, records in two independently developed databases could be matched up by social security number, except when the social security numbers are incorrect, missing, or have changed).

1.4.3.1 Why we use a Relational Database Design

Maintaining a simple, so-called flat database consisting of a single table doesn’t require much knowledge of database theory. On the other hand, most database worth maintaining are quite a bit more complicated than that. Real life databases often have hundreds of thousands or even millions of records, with data that are very intricately related. This is where using a full-fledged relational database program becomes essential. Consider, for example, the Library of Congress, which

has over 16 million books in its collection. For reasons that will become apparent soon, a single table simply will not do for this database.

1.5 Relationships between Tables

When you create tables for an application, you should also consider the relationships between them. These relationships give a relational database much of its power. There are three types of relationships between tables: one-to-one, one-to-many and many-to-many relationships.

1.5.2 One-To-One Relationships

In a one-to-one relationship, each record in one table corresponds to a single record in a second table. This relationship is not very common, but it can offer several benefits. First, you can put the fields from both tables into a single, combined table. One reason for using two tables is that each field is a property of a separate entity, such as owner operators and their trucks. Each operator can operate just one truck at a time, but the fields for the operator and truck tables refer to different entities.

A one-to-one relationship can also reduce the time needed to open a large table by placing some of the table's columns in a second, separate table. This approach makes particular sense when a table has some fields that are used infrequently. Finally, a one-to-one relationship can support in a table requires security, placing them in a separate table lets your application restrict to certain fields. Your application can link the restricted table back to the main table via a one-to-one relationship so that people with proper permissions can edit, delete, and add new records to these fields.

1.5.3 One-To-Many Relationships

A one-to-many relationship, in which a row from one table corresponds to one or more rows from a second table, is more common. This kind of relationship can form the basis for a **Many-To-Many** relationship as well.

1.6 Data Modeling

In information system design, data modeling is the analysis and design of the information in the system, concentrating on the logical entities and the logical dependencies between these entities. Data modeling is an abstraction activity in that the details of the values of individual data observations are ignored in favor of the structure, relationships, names and formats of the data of interest, although a list of valid values is frequently recorded. It is by the data model that definitions of what the data means is related to the data structures.

While a common term for this activity is "Data Analysis" the activity actually has more in common with the ideas and methods of **synthesis** (putting things together), than it does in the original meaning of the term **analysis** (taking things apart). This is because the activity strives to bring the data structures of interest together in a cohesive, inseparable, whole by eliminating unnecessary data redundancies and relating data structures by relationships. In the early phases of a software development project, emphasis will be on the design of a conceptual data model. This can be detailed into a logical data model sometimes called functional data model. In later stages, this model may be translated into physical data model.

1.6.1 Database Normalization

Database normalization is a series of steps followed to obtain a database design that allows for consistent storage and efficient access of data in a relational database. These steps reduce data redundancy and the risk of data becoming inconsistent.

However, many relational DBMS lack sufficient separation between the logical database design and the physical implementation of the data store, such that queries against a fully normalized database often perform poorly. In this case denormalizations are sometimes used to improve performance, at the cost of reduced consistency.

1.6.2 Primary Key

In database design, a **primary key** is a value that can be used to identify a particular row in a table. Attributes are associated with it. Examples are names in a telephone book (to look up telephone numbers), words in a dictionary (to look up definitions) and Dewey Decimal Numbers (to look up books in a library).

In the relational model of data, a **primary key** is a candidate key chosen as the main method of uniquely identifying a relation. Practical telephone books, dictionaries and libraries can not use names, words or Dewey Decimal System Numbers as candidate keys because they do not uniquely identify telephone numbers, word definitions or books. In some design situations it is impossible to find a natural key that uniquely identifies a relation. A **surrogate key** can be used as the **primary key**. In other situations there may be more than one candidate key for a relation, and no candidate key is obviously preferred. A surrogate key may be used as the primary key to avoid giving one candidate key artificial primacy over the others. In addition to the requirement that the primary key be a candidate key, there are several other factors which may make a particular choice of key better than others for a given relation.

The **primary key** should generally be short to minimize the amount of data that needs to be stored by other relations that reference it. A compound key is usually not appropriate. (However, this is a design consideration, and some database management systems may be better than others in this regard.)

The **primary key** should be immutable, meaning its value should not be changed during the course of normal operations of the database. (Recall that a **primary key** is the means of uniquely identifying a tuple, and that identity by definition, never changes.) This avoids the problem of dangling references or orphan records created by other relations referring to a tuple whose primary key has changed. If the **primary key** is immutable, this can never happen.

1.6.3 Foreign Key

A **foreign key (FK)** is a field in a database record under one primary key that points to a key field of another database record in another table where the **foreign key** of one table refers to the **primary key** of the other table. This way references can be made to link information together and it is an essential part of database normalization.

For example, a person sending an e-mail needs not to include the entire text of a book in the e-mail. Instead, they can include the ISBN of the book, and interested persons can then use the number to get information about the book, or even the book itself. The ISBN is the **primary key** of the book, and it is used as a **foreign key** in the e-mail.

Note that using a **foreign key** often assumes its existence as a **primary key** somewhere else. Improper **foreign key/primary key** relationships are the source of many database problems.

1.6.4 Compound Key

In database design, a **compound key** (also called a **composite key**) is a key that consists on 2 or more attributes.

No restriction is applied to the attribute regarding their (initial) ownership within the data model. This means that any one, none or all, of the multiple attributes within the **compound key** can be **foreign keys**. Indeed, a foreign key may, itself, be a compound key.

Compound keys almost always originate from attributive or associative entities (tables) within the model, but this is not an absolute value.

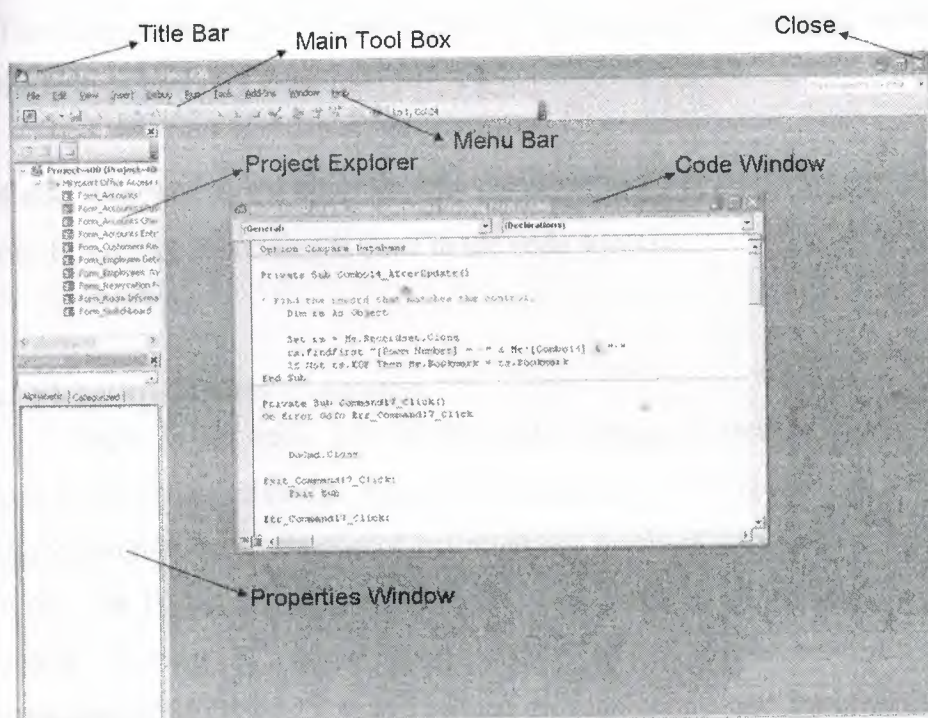
CHAPTER TWO: VISUAL BASIC EDITOR AND STRUCTURED QUERY LANGUAGE

2.1 Visual Basic Editor

The visual basic editor can be used in MS Access with different tools e.g. command button, Text box, combo box etc. Now let we see how we can open the Visual Basic editor in MS Access.

- Open MS Access
- Create a new data base
- Enter a form in the data base and open the form in design view.
- Go to the Tools from main menu.
- Select Macro option.
- Select Visual Basic Editor

Important: The Visual Basic Editor can not be opened in the Form view. After selecting the Visual Basic Editor this window will be displayed.



MS Access Visual Basic Editor

Let us see how to work with these windows.

Main Tool Box: It contains different main options e.g. save, open, run etc.

Project Explorer: This is a very important window if we are using a multi form project it helps us to explore the forms and the links which are related to these forms.

Properties Windows: The properties windows show all the properties of the selected object the most common properties are as follows.

Caption: This property can be found with every object in the Visual Basic Editor. It is the label presented on the objects when they run.

Name: This is a very important property because it is used in the coding of the program in our program we can access the objects with the name which is given to it in the property windows.

There are several more common properties which can be assigning to the objects by the help of property window.

Code Window: In the code window we write different codes for our objects used in our forms. All the coding is done in the code window.

2.2 Structured Query Language

SQL is the most popular computer language used to create, modify and retrieve data from relational management systems. The language has evolved beyond its original purpose to support object-relational database management systems.

During the 1970s, a group at IBM's San Jose research center developed a database system "System R" based upon Codd's model. **Structured English Query Language ("SEQUEL")** was designed to manipulate and retrieve data stored in System R. The acronym sequel was later condensed to SQL due to a trademark

dispute (the word "sequel" was held as a trade mark by the Hawker-Siddeley aircraft company of the UK).

In 1979, Relational Software, Inc. (now Oracle Corporation) introduced the first commercially available implementation of SQL and, soon many vendors developed dialects of it.

SQL was adopted as a standard by the ANSI (American National Standard Institute) in 1986 and IOS (International Organization for Standardization) in 1987. ANSI has declared that the official pronunciation for SQL is "es queue el", although many English-speaking database professionals still pronounce it as SEQUEL.

2.3 Description of SQL

SQL allows the specification of queries in a high level, declarative manner. For example, to select rows from a database, the user need only specify the criteria that want to search by; the details of performing the search operation efficiently is left up to the database system, and is invisible to the user.

Compared to general purpose programming languages, this structure allows the user/programmer to be less familiar with the information contained in the data. This blurs the line between user and programmer, appealing to individuals who fall more into the 'business' or 'research' area and less in the 'information technology' area. The original vision for SQL was to allow non-technical users to write their own database queries. While this has been realized to some extent, the complexity of querying an advanced database system using SQL can still require a significant learning curve.

SQL contrasts with the more powerful database-oriented fourth-generation programming languages such as focus, however, in its relative functional simplicity and simpler command set. This greatly reduces the degree of difficulty involved in maintaining the worst SQL source code, but it also makes programming such questions as 'Who had the top ten scores?' more difficult, leading to the development of procedural extensions, discussed above. However, it also makes it possible for SQL source code to be produced (and optimized) by software, leading to the development of a number of natural language database query languages, as well as

'drag and drop' database programming packages with 'object oriented' interfaces. Often these allow the resultant SQL source code to be examined, for educational purposes, further enhancement, or to be used in a different environment.

2.4 SQL Keywords

SQL keywords fall into several groups, like:

2.4.1 Data Retrieval

The most frequently used operation in transactional databases is the data retrieval operation.

- SELECT is used to retrieve zero or more rows from one or more tables in a database. In most applications, SELECT is the most commonly used DML command. In specifying a select query, the user specifies a description of the desired result set, but they do not specify what physical operations must be executed to produce that result set. Translating the query into an optimal query plan is to leave the database system, more specifically to the query optimizer.
- Commonly available keywords related to SELECT includes:
 - FROM is used to indicate which tables the data is to be taken from, as well as how the tables join to each other.
 - WHERE is used to identify which rows to be retrieved, or applied to GROUP BY.
 - GROUP BY is used to combine rows with related values into elements of a smaller set of rows.
 - ORDER BY is used to identify which columns are used to sort the resulting data.

2.4.2 Data Manipulation

First there are the standard Data Manipulation Language (DML) elements. DML is the subset of the language used to add, update and delete data.

- INSERT is used to add the zero or more rows (formally tuples) to an existing table.
- UPDATE is used to modify the values of a set of existing table rows.
- DELETE removes zero or more existing rows from a table.

2.4.3 Data Transaction

Transaction, if available, can be used to wrap around the DML operations. BEGIN WORK (or START TRANSACTION, depending on SQL dialect) can be used to mark the start of a database transaction, which either completes completely or not at all

COMMIT causes all data changes in a transaction to be made permanent.

ROLLBACK causes all data changes since the last COMMIT or ROLLBACK to be discarded, so that the state of the data is "Rolled Back" to the way it was prior to those changes being requested.

COMMIT and ROLLBACK interact with areas such as transaction control and locking. Strictly, both terminate any open transaction and release any locks held on data. In the absence of a BEGIN WORK or similar statement, the semantics of SQL are implementation-dependent.

2.4.4 Data Definition

The second group of keywords is the Data Definition Language (DDL). DDL allows the user to define new tables and associated elements. Most commercial SQL databases have proprietary extensions in their DDL, which allow control over nonstandard features of the database system.

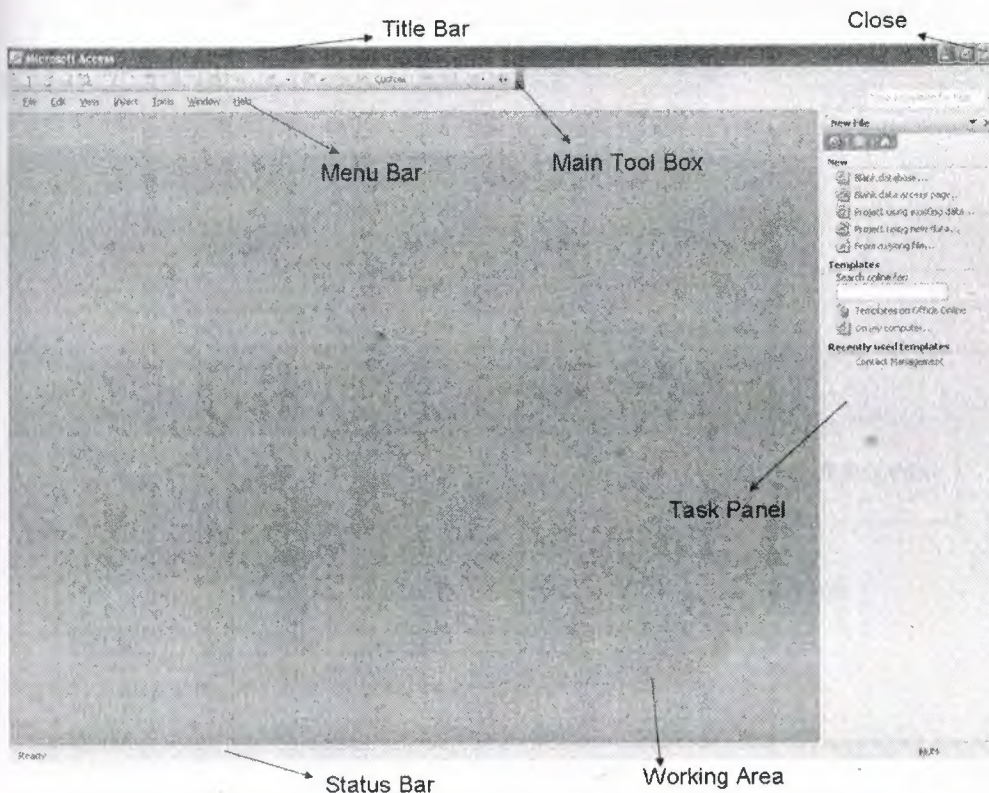
The most basic items of DDL are the CREATE and DROP commands. CREATE causes an object (a table, for example) to be created within the database. DROP causes an existing object within the database to be deleted, usually irretrievably.

CHAPTER THREE: MICROSOFT ACCESS DATABASE SYSTEM

Some terms are used often in access. Some of them are stated below.

3.1 A Few Terms

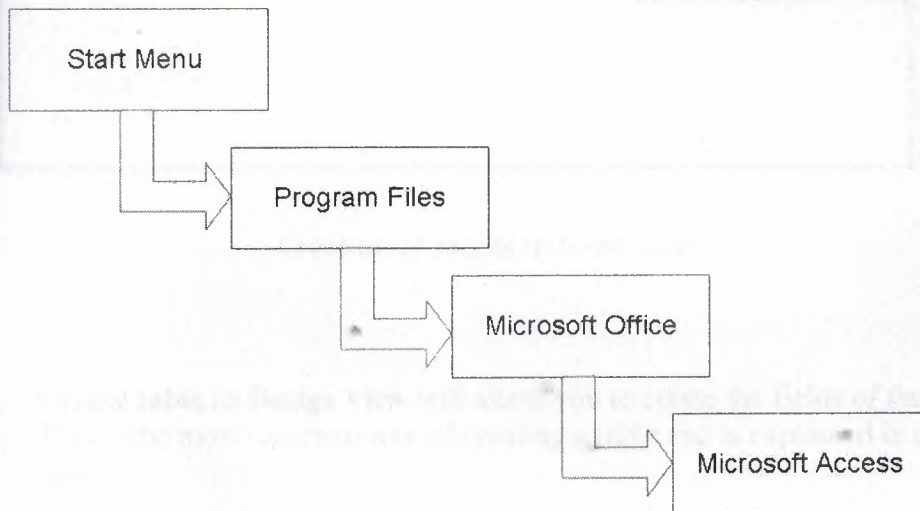
- (1) An **object** is a competition in the database such as a table, query, form, or macro.
- (2) A **table** is a grouping of related data organized in fields (columns) and records (rows) on a datasheet. By using a common field in two tables, the data can be combined. Many tables can be stored in a single database.
- (3) A **field** is a column in database and defines a data type for a set of values in a table
- (4) All rows except first are **records**.
- (5) **Design View** provides the tools for creating fields in a table.
- (6) **Datasheet View** allows you to update, edit, and delete in formation from a table.



Main window of Microsoft Access

3.2 Introductory Microsoft Access

- Go to start menu
- Opening Access from Microsoft office
- Creating a Database
- Click File, New or click the new icon on the standard toolbar
- Select Blank Database from the Task Pane menu
- Type a name for the Database in the File Name window
- Click Create and wait for some time
- Closing a database click File, Close
- Opening a database click File, Open or click the open icon on the standard toolbar
- Browse, where to save the database
- Click the name of the database
- Click Open



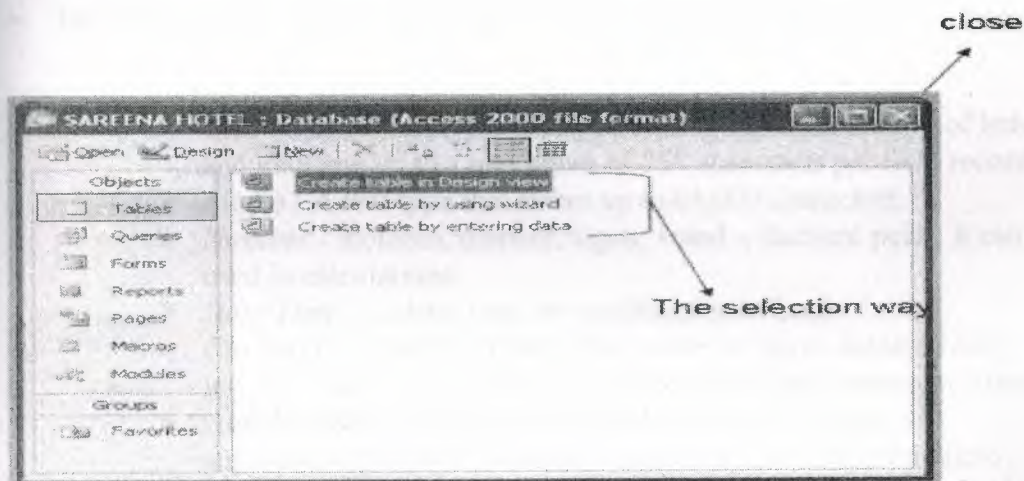
Pictorial representation of opening MS Access

3.3 Introduction to Tables

A **table** is a grouping of related data organized in fields (columns) and records (rows) on a datasheet. A database may have one or more tables.

Tables are grids that store information in a database

Access provides *three* ways to create a table for which there are icons in the Database Window. Double-click on the icons (how u want) to create a table.



Creation of tabels in three ways

1. **Create table in Design view** will allow you to create the fields of the table. This is the most common way of creating a table and is explained in detail below.
2. **Create table using wizard** will step you through the creation of a table.
3. **Create table by entering data** will give you a blank datasheet with unlabelled columns that looks much like an Excel worksheet. Enter data into the cells and click the **Save** button. You will be prompted to add a **primary key** field. After the table is saved, the empty cells of the datasheet are

trimmed. The fields are given generic names such as "Field1", "Field2", etc. To rename them with more descriptive titles that reflect the content of the fields, select **Format Rename Column** from the menu bar or highlight the column, right-click on it with the mouse, and select **Rename Column** from the shortcut menu.

3.3.1 Table's data types

First we introduce the following terms related to table design.

- **Field Name** - This is the name of the field and should represent the contents of the field such as "Name", "Address", etc. The name can not exceed 64 characters in length and may include spaces.
- **Data Type** is the type of value that will be entered into the fields.
 - **Text** - The default type, text type allows any combination of letters and numbers up to a maximum of 255 characters per field record.
 - **Memo** - A text type that stores up to 64,000 characters.
 - **Number** - includes, number, signs: + and -, decimal point, it can be used in calculations
 - **Date/Time** - A date, time, or combination of both.
 - **Currency** - Monetary values that can be set up to automatically include a dollar sign (\$) and correct decimal and comma positions.
 - **AutoNumber** - When a new record is created, Access will automatically assign a unique integer to the record in this field. Since every record in a datasheet must include at least one field that distinguishes it from all others, this is a useful data type to use if the existing data will not produce such values, it can be a primary key.
 - **Yes/No** - Use this option for True/False, Yes/No, On/Off, or other values that must be only one of two.
 - **OLE Object** - An OLE (Object Linking and Embedding) object is a sound, picture, or other object such as a Word document or Excel spreadsheet that is created in another program. Use this data type to embed an OLE object or link to the object in the database.
 - **Hyperlink** - A hyperlink will link to an Internet or Intranet site, or another location in the database (URL).
- **Description (optional)** - Enter a brief description of what the contents of the field are.

- **Field Properties** - Select any pertinent properties for the field from the bottom pane of Design View window
- **Field Size** is used to set the number of characters needed in a text or number field. The default field size for the text type is 50 characters. Maximum size is 255.
 - **Byte** - Positive integers between 1 and 255
 - **Integer** - Positive and negative integers between -32,768 and 32,768
 - **Long Integer (default)** - Larger positive and negative integers between -2 billion and 2 billion.
 - **Single** - Single-precision floating-point number
 - **Double** - Double-precision floating-point number
 - **Decimal** - Allows for Precision and Scale property control
- **Format** changes the way a field is displayed or printed, but does not affect the stored value.
- **Number format.** Select one of the preset options from the drop down menu or construct a custom format using symbols explained below:
 - **Date format.** In the table below, the value "1/1/01" is entered into the datasheet, and the following values are displayed as a result of the different assigned formats.
 - **Yes/No.** fields are displayed as check boxes by default on the datasheet. To change the formatting of these fields, first click the Lookup tab and change the Display Control to a text box. Go back to the General tab choices to make formatting changes. The formatting is designated in three sections separated by semicolons. The first section does not contain anything but the semicolon must be included. The second section specifies formatting for Yes values and the third for No values.
- **Default Value.** There may be cases where the value of a field will usually be the same for all records. In this case, a changeable default value can be set to prevent typing the same thing numerous times. Set the Default Value property.

- **Primary Key.** Every record in a table must have a **primary key** that differentiates it from every other record in the table. A social security number is an example of a record whose values will only appear once in a database table.

Designate the primary key field by right-clicking on the record and selection **Primary Key** from the shortcut menu or select **Edit Primary Key** from the menu bar. The primary key field will be noted with a key image to the left. To remove a primary key, repeat one of these steps.

If none of the existing fields in the table will produce unique values for every record, a separate field must be added. Access will prompt you to create this type of field at the beginning of the table the first time you save the table and a primary key field has not been assigned. The field is named "ID" and the data type is "auto number". Since this extra field serves no purpose to you as the user, the auto number type automatically updates whenever a record is added so there is no extra work on your part. You may also choose to hide this column in the datasheet as explained on a later page in this tutorial.

- **Indexes.** Creating indexes allows Access to query and sort records faster. To set an indexed field, select a field that is commonly searched and change the Indexed property to **Yes (Duplicates OK)**, if multiple entries of the same data value are allowed or **Yes (No Duplicates)** to prevent duplicates.
- **Field Validation Rules.** Validation Rules specify requirements (change word) for the data entered in the worksheet. A customized message can be displayed to the user when data that violates the rule setting is entered. Click the expression builder ("...") button at the end of the Validation Rule box to write the validation rule. Examples of field validation rules include $\neq 0$ to not allow zero values in the record, and $\text{Len} < 3$ To only all data strings three characters in length.
- **Input Masks.** An input mask controls the value of a record and sets it in a specific format. They are similar to the Format property, but instead display the format on the datasheet before the data is entered. For example, a telephone number field can formatted with an input mask to accept ten digits that are automatically formatted as "(555) 123-4567". The blank field would look like () - . An input mask to a field by following these steps:

1. In design view, place the cursor in the field that the input mask will be applied to.
2. Click in the white space following **Input Mask** under the **General** tab.

3.3.2 Creating a table using the wizard:

- Double click on **Create table by using wizard**
- Select the type of table (**Business** or **Personal**)
- Choose a table from the **Sample Tables** list
- Select a data field to include in the table
- Click the **single arrow**
- Repeat these steps or if we want to get all the fields to include we have to click on the **double arrow**
- Click **Next** if we want to make any field as the Primary Key field
- Then click **Next** if we want to make any **form** for the table to enter the data straight in the table
- Click **Finish**

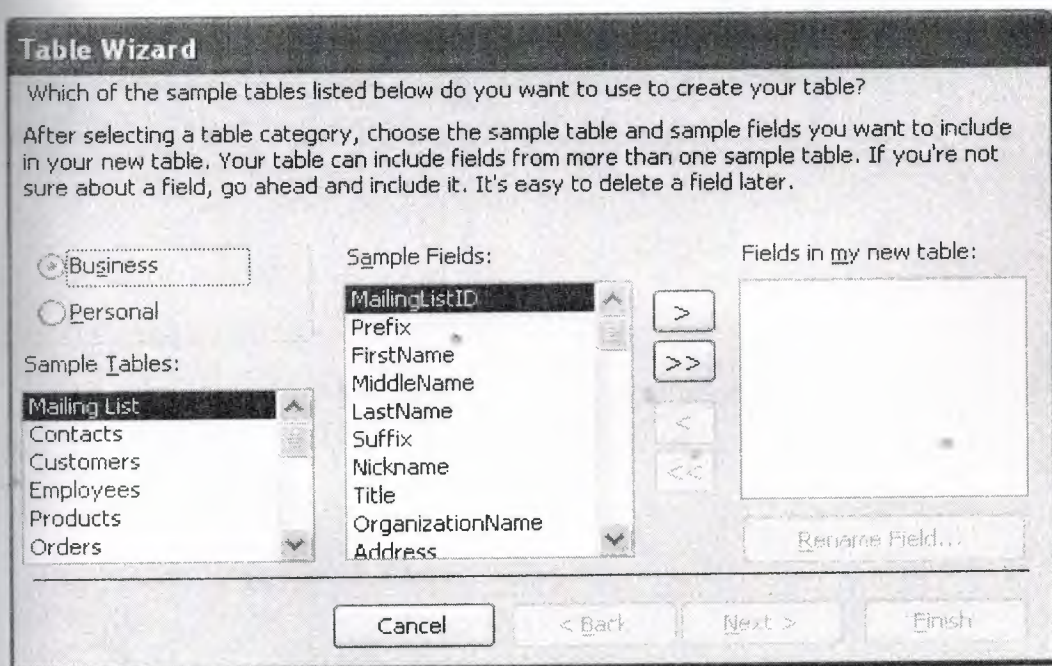


Table wizard

3.3.3 Create a table with design views:

- Double click on **Create table in design view**
- In the **Field Name** column type the name of data field
- In the **Data Type** column select the type of data to be entered in the field.
- To save the table, click **File**, then click **Save** or click the save icon on the Standard toolbar
- Type the name of the **Table**
- Click OK
- If we do not want any **Primary Key**, click no

3.3.4 Creating Tables by entering data

To create a table quickly, we double click on the **Create Table in Design View** icon in the database window or to open a blank table window in the design view. You can also create a table by clicking on the **Create Table in Design View** icon in the database window.

To add a field to the table, click on the **Field Name** column in the design view. The **Field Name** column will be highlighted. You can then type the name of the field. The **Data Type** column will be highlighted. You can then select the data type for the field. The **Table** column will be highlighted. You can then type the name of the table. The **Primary Key** column will be highlighted. You can then click on the **Primary Key** icon to set the primary key for the table.

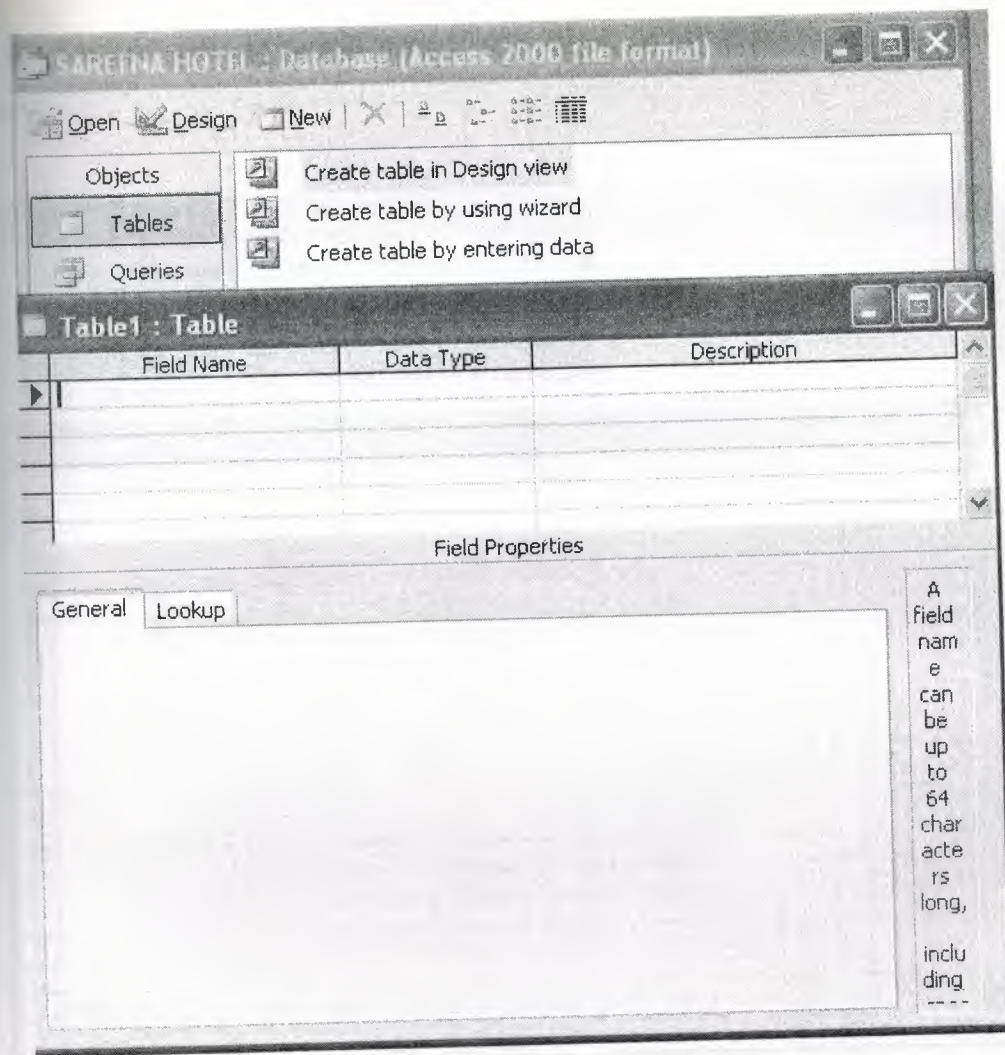


Table in design view

3.3.4 Creating Tables by entering data

To create a table manually, we double click on the Create Table in Design View icon in the database window to open a blank table window in the design view. From here, we can add any fields, a Primary or a Composite Key or an index.

To add a field, type the field's name in a blank field name column in the Design view window. Field names follow normal Visual Basic for Applications (VBA) naming conventions. They can be up to 64 characters long, and the characters can be letters, numbers, spaces, and special characters except the period, the exclamation mark, square brackets and the grave accent character ('). Also we can not start a field name with a space or a control character (ASCII values 0 through 31). While we can

Include internal spaces in field names, they must be bracketed in expressions and queries.

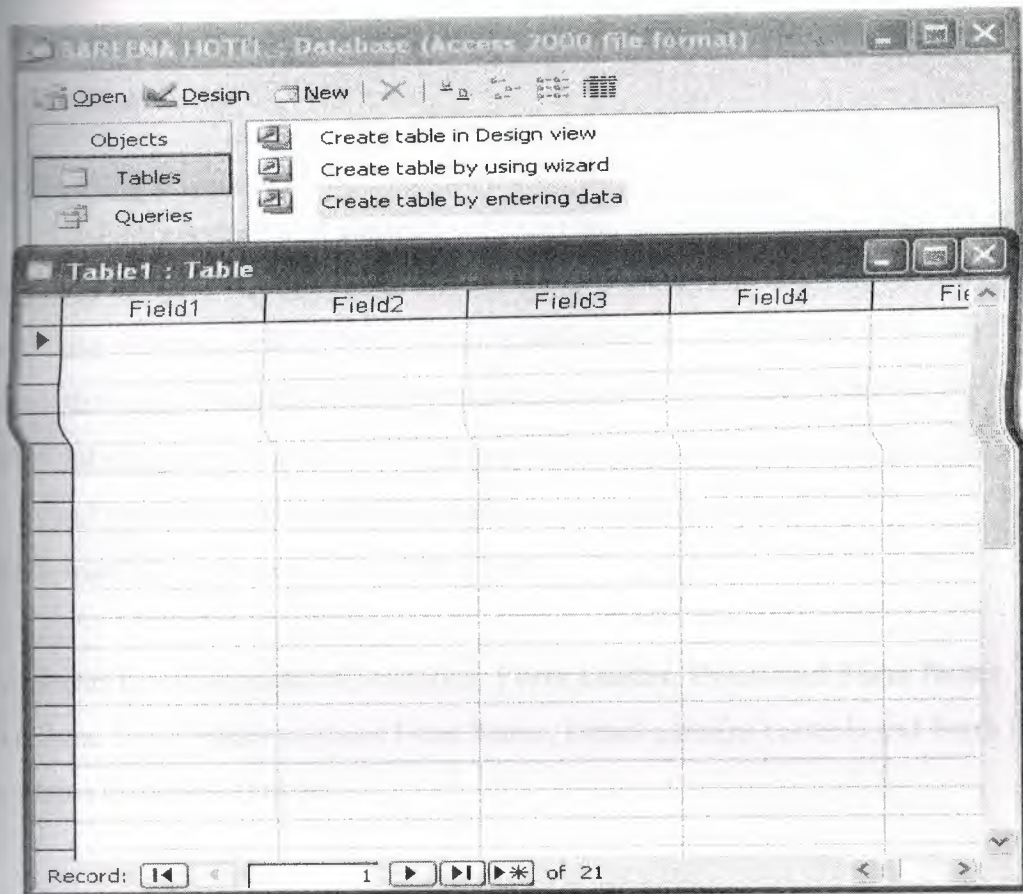


Table by entering data

3.3.5 Entering data in the Table:

- Double click on the table name
- Click in the field you want to enter data
- Press Tab to move to the next field
- Click in the data field
- Click a sort icon on the Standard Toolbar
- Or click Records
- Select the sort order

3.3.5.1 Deleting data from a Table:

- Click in the record we wish to delete
- Click Edit, Delete Record

3.3.5.2 Saving data in a Table:

- Click File, Save or click the save icon on the Standard Toolbar

3.3.5.3 Closing a Table:

- Click File, Close. Or click on the cross on the upper right corner of the window.

3.4 Creating Forms

All forms consist of **controls** (objects) that accept and display data. There are 3 types of controls - **bound, unbound, and calculated**. A **bound control** (text box) has a data source (a field in the underlying table) and is used to enter or modify the data in that table. An **unbound control** has no data source. They are used to display titles, labels, graphics and pictures. Every bound control is associated with one unbound control. A **calculated control** has as its data sourced a mathematical expression rather than a field.

All forms layout consists of 3 sections: **Form header, Detail and Form footer sections**. Form header contains Form Name, Detail contains controls and Form footer contains command buttons.

3.4.1 Create Form in Design View

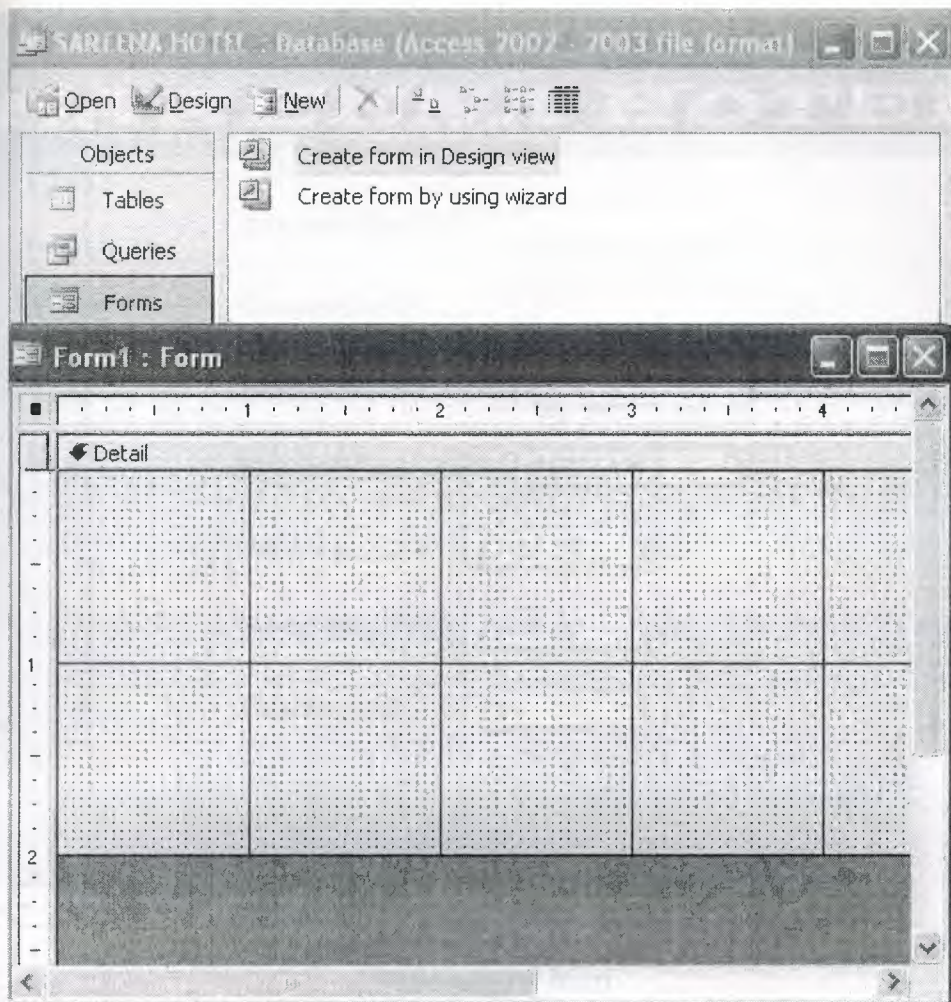
To create a form using design view, we have to obey following steps.

- Click the **New** button on the form database window.
- Select "Design View" and choose the table or query the form will be associated with the form from the drop-down menu.
- Select **View Toolbox** from the menu bar to view the floating toolbar with additional options.




Tool bar

Add controls to the form by clicking and dragging the field names from the Field List floating window. Access creates a text box for the value and label for the field name when this action is accomplished. To add controls for all of the fields in the Field List, double-click the Field List window's title bar and drag all of the highlighted fields to the form.



Form in design view

3.4.1.1 Adding Records Using a Form

Input data into the table by filling out the fields of the form. Press the **Tab** key to move from field to field and create a new record by clicking **Tab** after the last field of the last record. A new record can also be created at any time by clicking the **New Record** button  at the bottom of the form window. Records are automatically saved as they are entered so no additional manual saving needs to be executed.

Reservation Form

SAREENA HOTEL

Reservation ID: 123

Customer ID: 111

Employee ID: 21

Reservation Date: 05/21/06

Amount Paid: \$100.00

Total Due: \$100.00

Confirmed: Yes

Record: 1 of 4

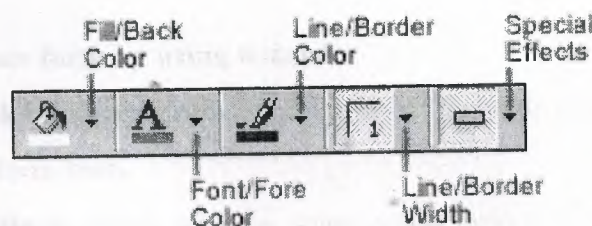
Reservation form

3.4.1.2 Editing a Forms

The follow points may be helpful when modifying forms in Design View.

- **Grid lines** - By default, a series of lines and dots underlay the form in Design View so form elements can be easily aligned. To toggle this feature on and off select **View Grid** from the menu bar.
- **Snap to Grid** - Select **Format Snap to Grid** to align form objects with the grid to allow easy alignment of form objects or uncheck this feature to allow objects to float freely between the grid lines and dots.
- **Resizing Objects** - Form objects can be resized by clicking and dragging the handles on the edges and corners of the element with the mouse.

- **Change form object type** - To easily change the type of form object without having to create a new one, right click on the object with the mouse and select **Change To** and select an available object type from the list.
- **Label/object alignment** - Each form object and its corresponding label are bounded and will move together when either one is moved with the mouse. However, to change the position of the object and label in relation to each other (to move the label closer to a text box, for example), click and drag the large handle at the top, left corner of the object or label.
- **Tab order** - Alter the tab order of the objects on the form by selecting **View Tab Order** from the menu bar. Click the gray box before the row you would like to change in the tab order, drag it to a new location, and release the mouse button.
- **Form Appearance** - Change the background color of the form by clicking the **Fill/Back Color** button on the formatting toolbar and click one of the color swatches on the palette. Change the color of individual form objects by highlighting one and selecting a color from the **Font/Fore Color** palette on the formatting toolbar. The font and size, font effect, font alignment, border around each object, the border width, and a special effect can also be modified using the formatting toolbar:



- **Page Header and Footer** - Headers and footers added to a form will only appear when it is printed. Access these sections by selecting **View Page Header/Footer** on the menu bar. Page numbers can also be added to these sections by selecting **Insert Page Numbers**. A date and time can be added from **Insert Date and Time....** Select **View Page Header/Footer** again to hide these sections from view in Design View.

The screenshot shows a Microsoft Access form titled "SAREENA HOTEL" with a "Customer's Records" title bar. The form contains two columns of text boxes for data entry. The left column includes fields for Customer ID, First Name, Middle Name, Last Name, Country, City, and Address. The right column includes fields for Staying From, Staying Till, Paid, Due, Home Phone, Mobile Phone, and Email Address. The data entered in the form is as follows:

Field	Value
Customer ID	123
First Name	Ahmed
Middle Name	
Last Name	Iqbal
Country	Pakistan
City	Multan
Address	Multan
Staying From	3/5/2006
Staying Till	3/22/2006
Paid	\$1,000.00
Due	\$0.00
Home Phone	0092614
Mobile Phone	0542-8727817
Email Address	ahmed.iqbal@hotel

At the bottom of the form, there is a status bar showing "Record: 14" and "1 of 16".

Customer record form

3.4.2 Creating Form by Using Wizard

To create a form using the assistance of the wizard, follow these steps:

- From the main database window, click **Forms** under **Objects** on the left menu
- Click **Create form by using wizard**
- In the **Tables/Queries** drop down list, select the table or query we wish to create the form from
- In the **Available Fields** window, select a data field to include in the form and click the **Single arrow** to select one data field in the Selected Fields
- Or click the **Double arrow** to include all data fields in the form's field
- Click **Next**
- Select a layout for the form, then click **Next**
- Select a style for the form, then click **Next**
- Type a name for the **Form**
- Click **Finish**

Form Wizard

Which fields do you want on your form?
You can choose from more than one table or query.

Tables/Queries
Table: Employees

Available Fields:

- DepartmentName
- SocialSecurityNumber
- EmployeeNumber
- NationalEmplNumber
- MiddleName
- Title
- EmailName
- Extension

Selected Fields:

- EmployeeID
- FirstName
- LastName

Buttons: Cancel, < Back, Next >, Finish

Form wizard

Form Wizard

What layout would you like for your form?

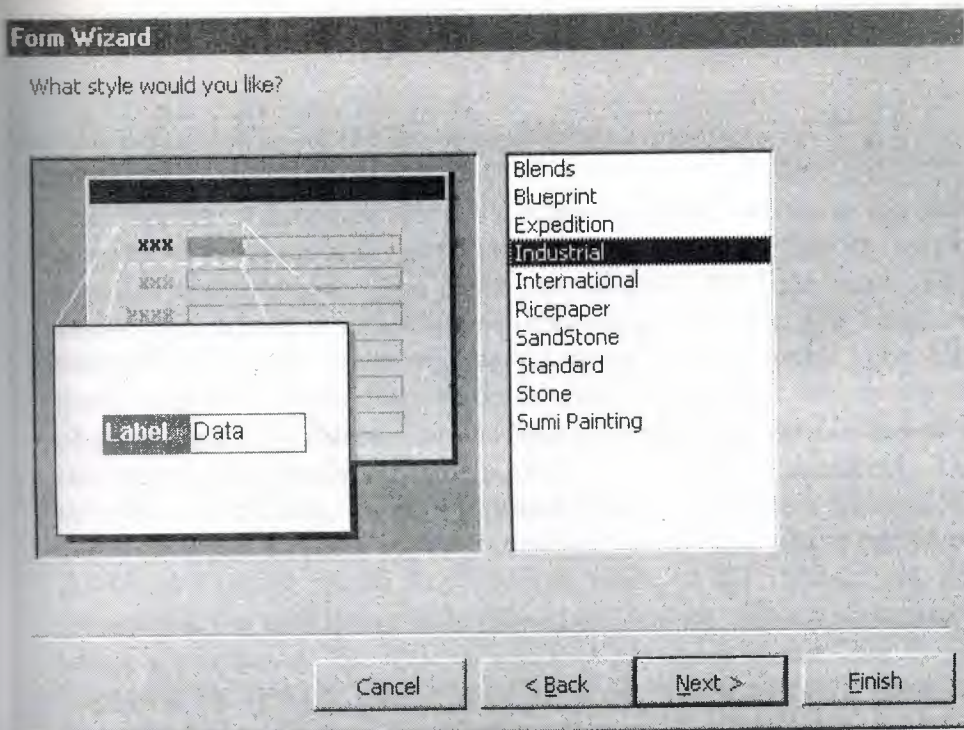
Layout preview: A grid showing a sample data layout with columns for EmployeeID, FirstName, LastName, and other fields.

Layout Options:

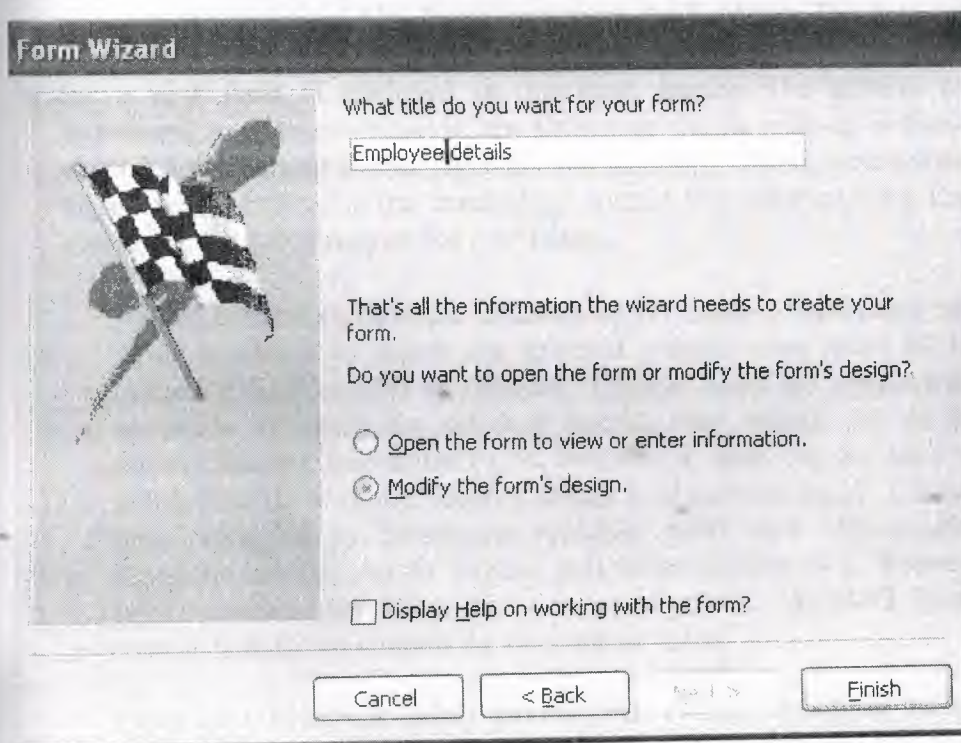
- ☒ Columnar
- ☐ Tabular
- ☐ Datasheet
- ☐ Justified

Buttons: Cancel, < Back, Next >, Finish

Form wizard 2



Form wizard 3



Form wizard 4

3.5 Queries

Query is a question about DB and answer for this question.

The question is developed by using a graphical tool known as the **design grid**. It consists of columns and rows. Each field in the query has its own column and contains multiple rows. The **Field row** displays the field name. The **Sort row** enables you to sort in ascending or descending sequence. The **Show row** controls whether or not the field will be displayed in the dynast. The **Criteria row** determines the records that will be selected. The answer selects records from one or more tables in a database on the base of criteria specified in design grid so they can be viewed, analyzed, and sorted on a common datasheet. The resulting collection of records, called a **dynaset** (short for dynamic subset), is saved as a database object and like a table, it enables you to enter a new record or modify or delete an existing record. Any changes made in the dynaset are automatically reflected in the underlying table. The query will be updated whenever the original tables are updated. The **Query window** has three views. The **Design view** is displayed by default and is used to create (or modify) a select query. The **Datasheet view** displays the resulting dynaset. The **SQL view** enables you to use SQL (Structured Query Language) statements to modify the query. The upper portion of the Design view window contains the field list for the tables on which the query is based. The lower portion of the window displays the design grid, which is where the **specifications for the select query** are entered. A field is added to the design grid by dragging it from the field list. The data type of a field determines the way in which the criteria are specified for that field. The criterion for a **text field** is enclosed in quotation marks. The criteria for **number, currency, and counter** fields are shown as digits with or without a decimal point. (Commas and dollar signs are not allowed). **Dates** are enclosed in **pound signs** and are entered in the **mm/dd/yy** format. The criterion for a **Yes/No** field is entered as **Yes(or True) or No (or False)**.

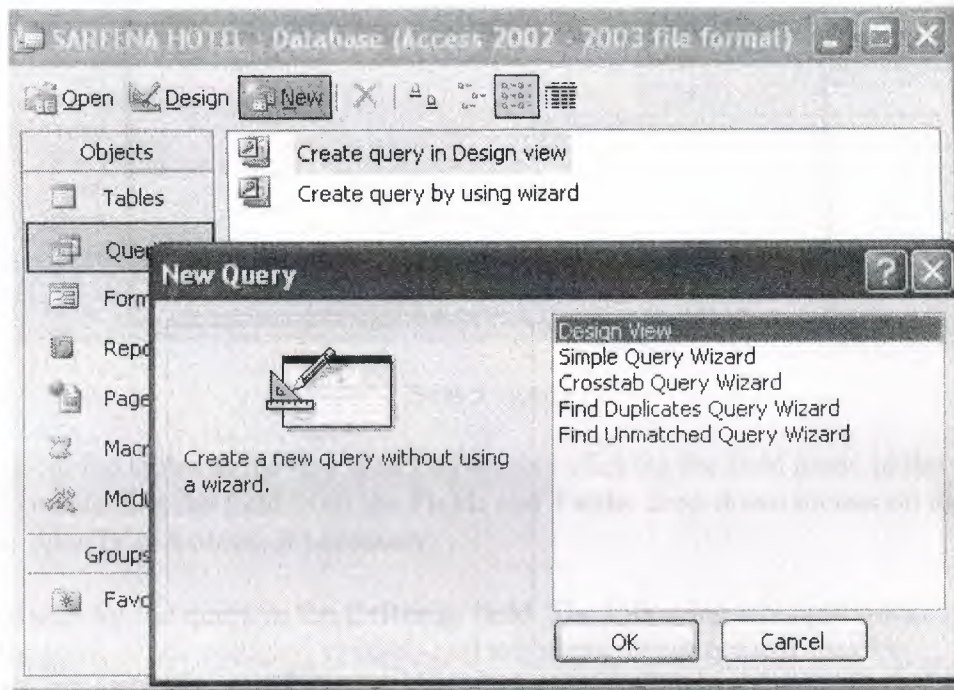
Values entered in multiple columns of the same Criteria row implement an **AND** condition in which the selected records must meet all the specified criteria. Values entered in different Criteria rows are connected by an **OR** condition in which the selected records may satisfy any of the indicated criteria. **Relation operators** (>, <, >=, <=, =, and <>) are used with date or number fields to return records within a designated range. Criteria can grow more complex by combining multiple **AND** and **OR** conditions. Other functions enable you to impose still other criteria. The **between function** selects records that fall within a range of values. The **NOT** function selects records that do not contain the designated value.

Types of queries are **select queries** that extract data from tables based on specified values, **find duplicate** queries that display records with duplicate values for one or more of the specified fields, and **find unmatched** queries display records from one table that do not have corresponding values in a second table.

3.5.1 Create a Query in Design View

Follow these steps to create a new query in Design View:

From the Queries page on the Database Window, click the **New** button.




Query design view

- Select Design View and click **OK**.
- Select tables and existing queries from the **Tables** and **Queries** tabs and click the **Add** button to add each one to the new query.
- Click **Close** when all of the tables and queries have been selected.



Select query

Add fields from the tables to the new query by double-clicking the field name in the table boxes or selecting the field from the **Field:** and **Table:** drop-down menus on the query form. Specify sort orders if necessary.

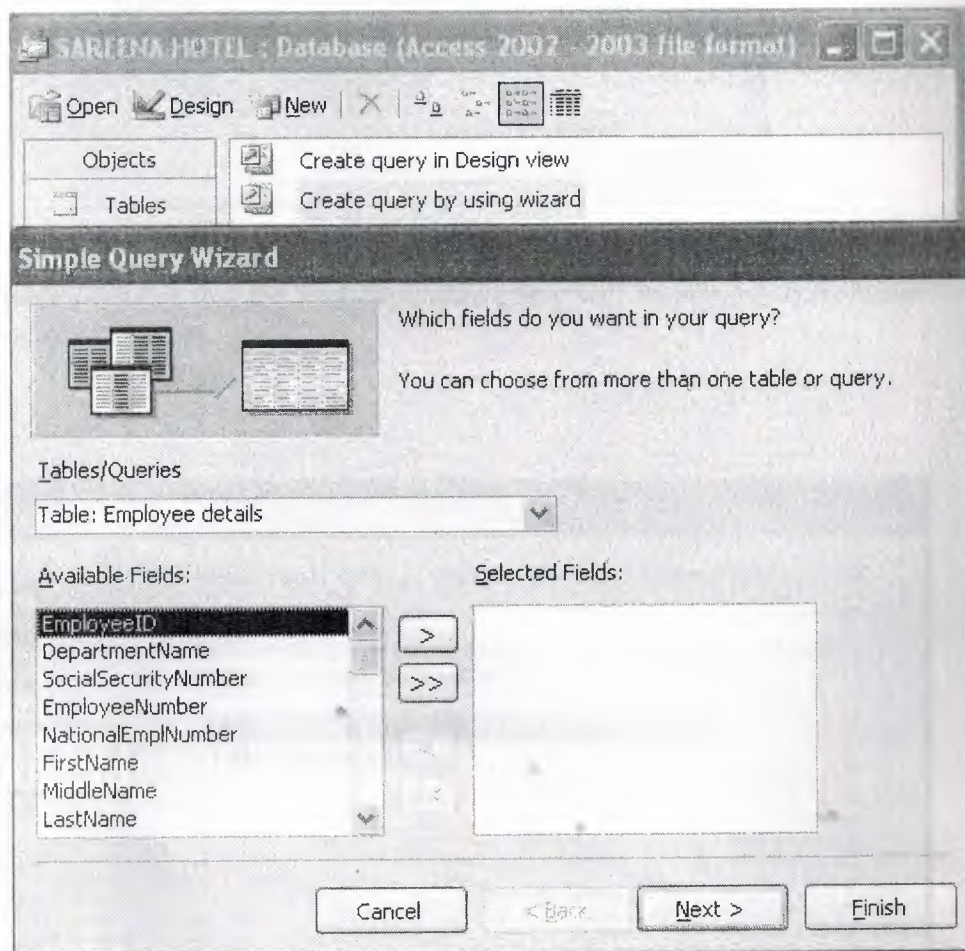
Enter the criteria for the query in the **Criteria:** field. The following table provides examples for some of the wildcard symbols and arithmetic operators that may be used. The **Expression Builder**  can also be used to assist in writing the expressions.

Query Wildcards and Expression Operators	
Wildcard / Operator	Explanation
? Street	The question mark is a wildcard that takes the place of a single letter.
43th *	The asterisk is the wildcard that represents a number of characters.
<100	Value less than 100
>=1	Value greater than or equal to 1
<>"FL"	Not equal to (all states besides Florida)
Between 1 and 10	Numbers between 1 and 10
Is Null Is Not Null	Finds records with no value or all records that have a value
Like "a*"	All words beginning with "a"
>0 And <=10	All numbers greater than 0 and less than 10
"Bob" Or "Jane"	Values are Bob or Jane

- After you have selected all of the fields and tables, click the **Run** button on the toolbar.
- Save the query by clicking the **Save** button

3.5.2 Query Wizard

- Double click on the **Create query by using wizard** option
- In the **Tables/Queries** drop-down list, select the table or query we wish to create for query
- In the **Available Fields** window, select a data field. To select the field click the **single arrow**



Query by Wizard

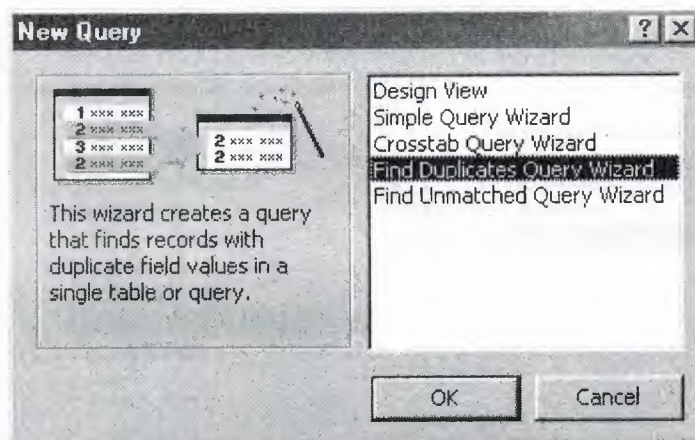
Or to select **all the fields** we need to click the **double arrow**, which will include all the fields in the query. We can chose fields from more than one table or query.

Click **Next**

Type a name for the query, then click **Finish**

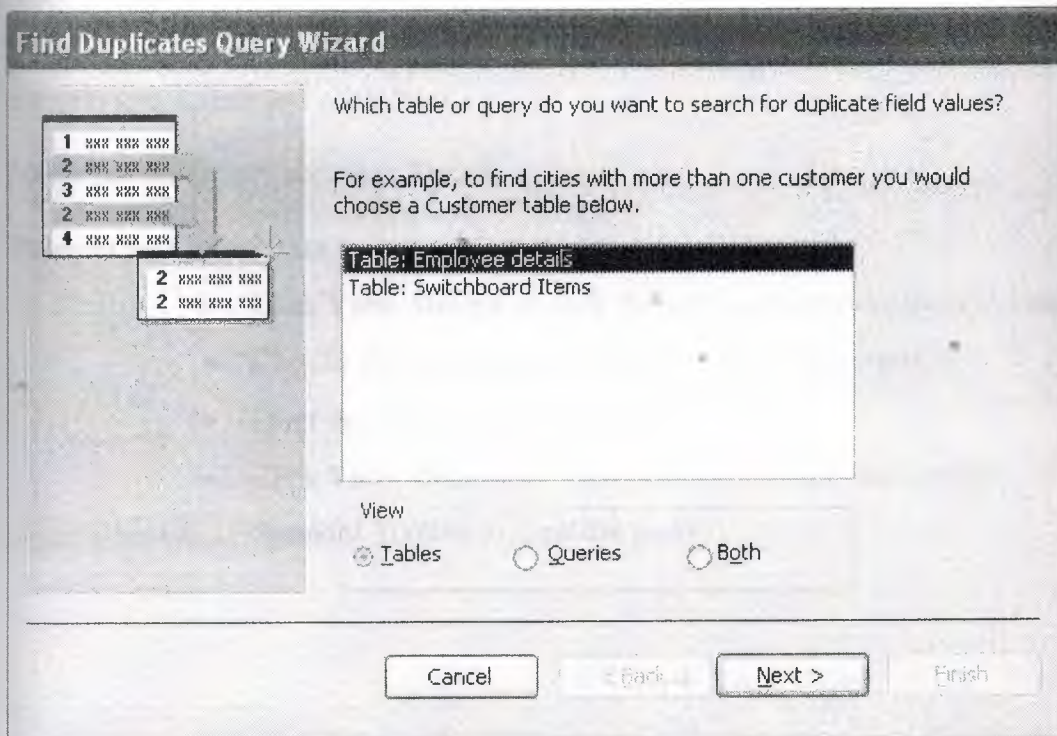
3.5.2.1 Find Duplicates Query

- This query will filter out records in a single table that contain duplicate values in a field.
 - Click the **New** button on the Queries database window, select **Find Duplicates Query Wizard** from the **New Query** window and click **OK**



New Query

Select the table or query that the find duplicates query will be applied to from the list provided and click **Next >**.



Finding the Duplicates Query wizard

Select the fields that may contain duplicate values by highlighting the names in the Available fields list and clicking the > button to individually move the fields to the Duplicate-value fields list or >> to move all of the fields. Click **Next >** when all fields have been selected.

Find Duplicates Query Wizard

Which fields might contain duplicate information?

For example, if you are looking for cities with more than one customer, you would choose City and Region fields here.

Available fields:

- NationalEmpINumber
- MiddleName
- Title
- EmailName
- Extension
- City
- StateOrProvince
- Region

Duplicate-value fields:

- FirstName
- LastName
- Address

Buttons: >, >>, <, <<, Cancel, < Back, Next >, Finish

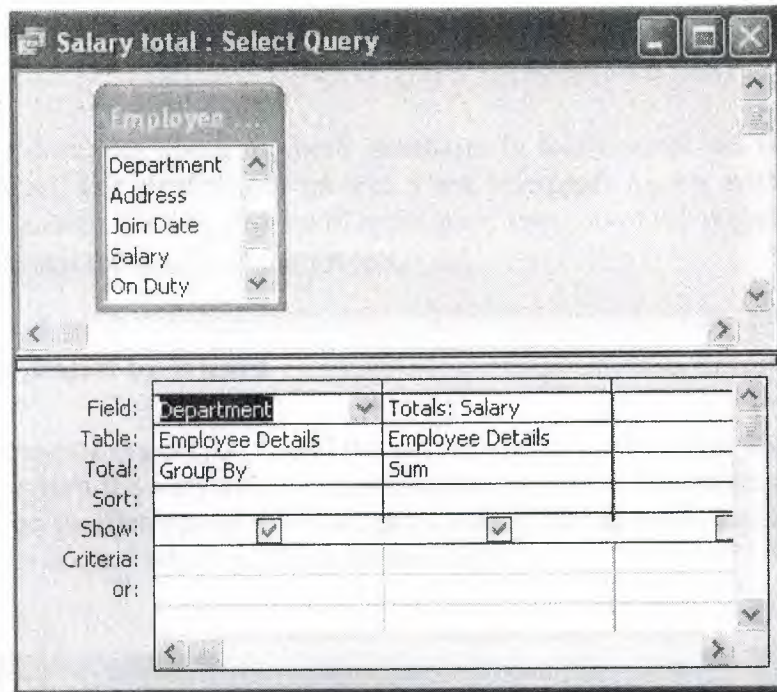
Finding the Duplicates Query wizard 2

Select the fields that should appear in the new query along with the fields selected on the previous screen and click **Next**.

Name the new query and click **Finish**.

Restricting Query Data

- Click **View, Design** or click the view icon on the Standard Toolbar
- Click in the **criteria box** of the field we wish to restrict
- Enter the value for the **criteria**
- Click **View, Datasheet View** or click the view icon on the Standard Toolbar to view the query



Example of Query

3.5.2.2 Saving a Query

Click **File, Save** or click on the save icon on the Standard Toolbar

3.5.2.3 Closing a Query

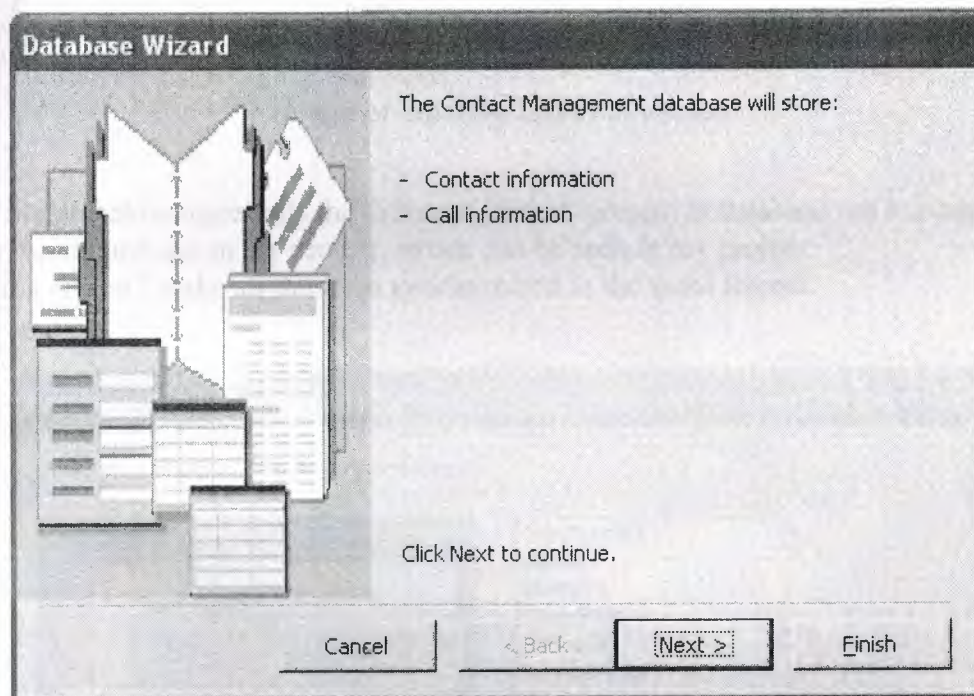
Click **File, Close**

CHAPTER FOUR: DESCRIPTION ABOUT THE PROJECT

This chapter describes about the hotel database. In this chapter we will discuss our project in detail. For creating our project, I use Microsoft Access with Visual basic editor, SQL queries. Step by step I will explain the creation of my project. Let us go through the details of our project.

4.1 Starting project by Wizard

For creating my project I used the wizard option of Microsoft access. By using this wizard option it's very easy to create tables and form. For selecting this option we have to go to after opening the access, Contact Management and after this new wizard will be opened which guide us further more.



Database wizard for form and tables

The next image displays the creation of table with different ways of different fields. The left hand window of the database wizard display the name of the table for the database, while the right arm window of the database wizard displays the fields of the tables, and in the end we have some control buttons.

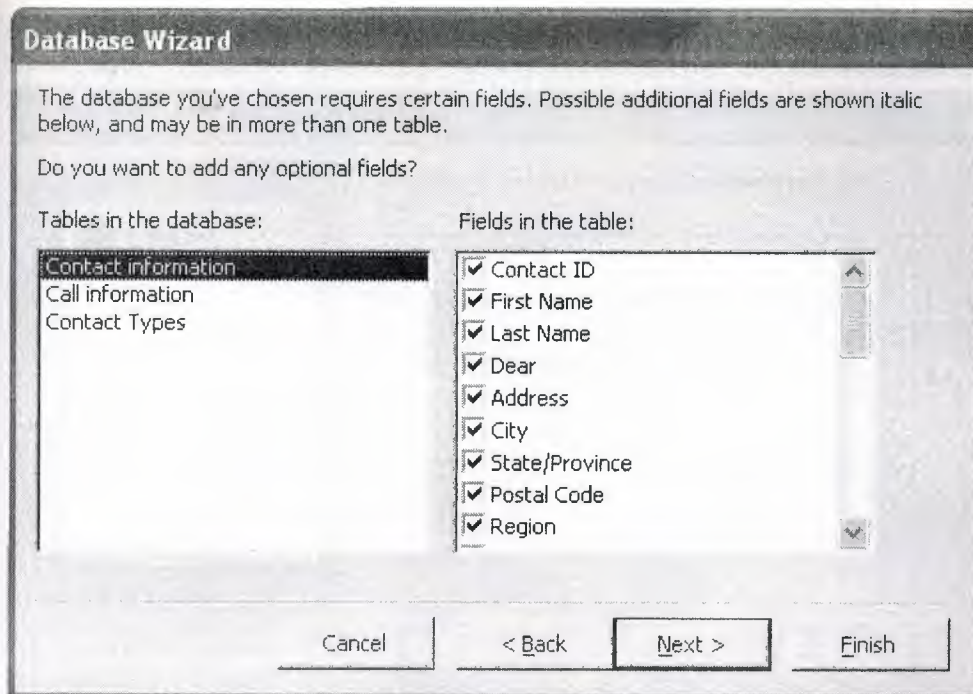
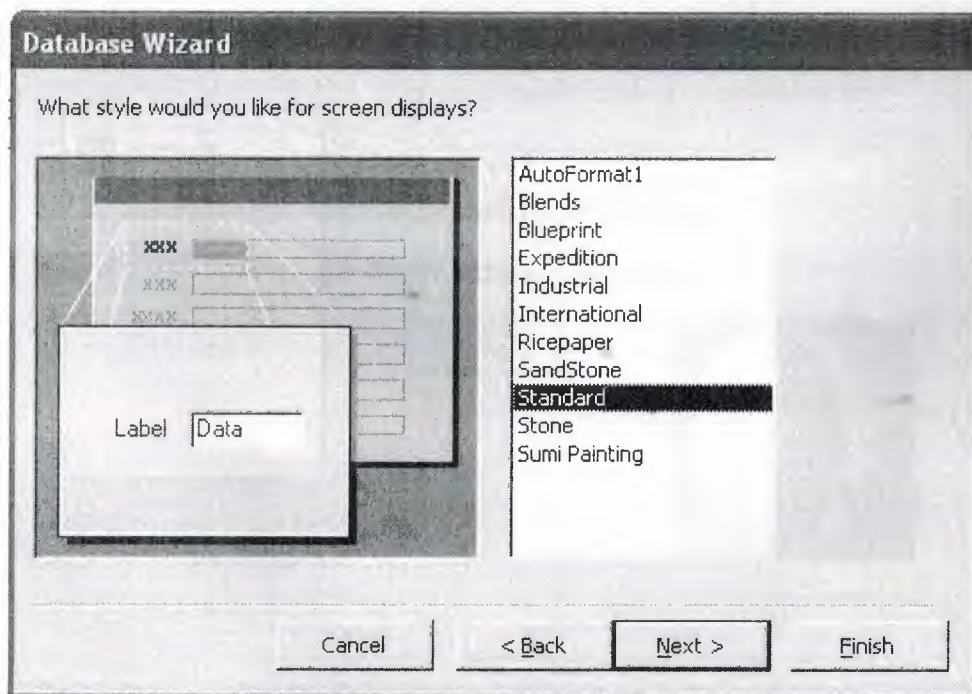


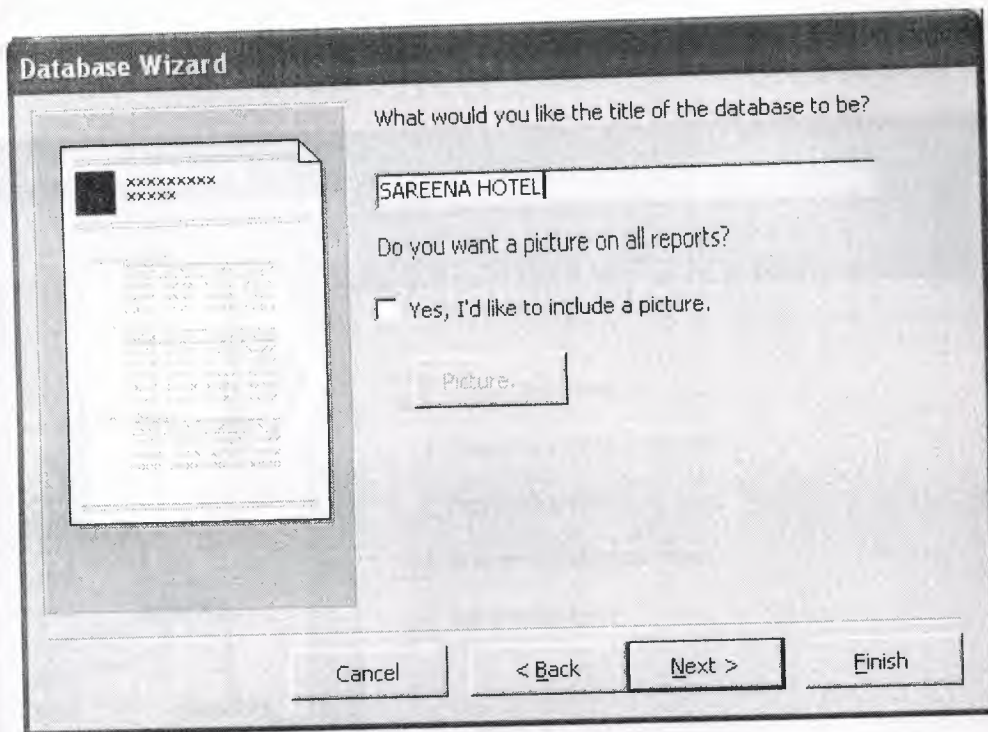
Image of creating tables in wizard

The image below represents the different formats present in database but I created my own format and use in my project, which can be seen in my project. By this option I make all the form synchronized in the same format.



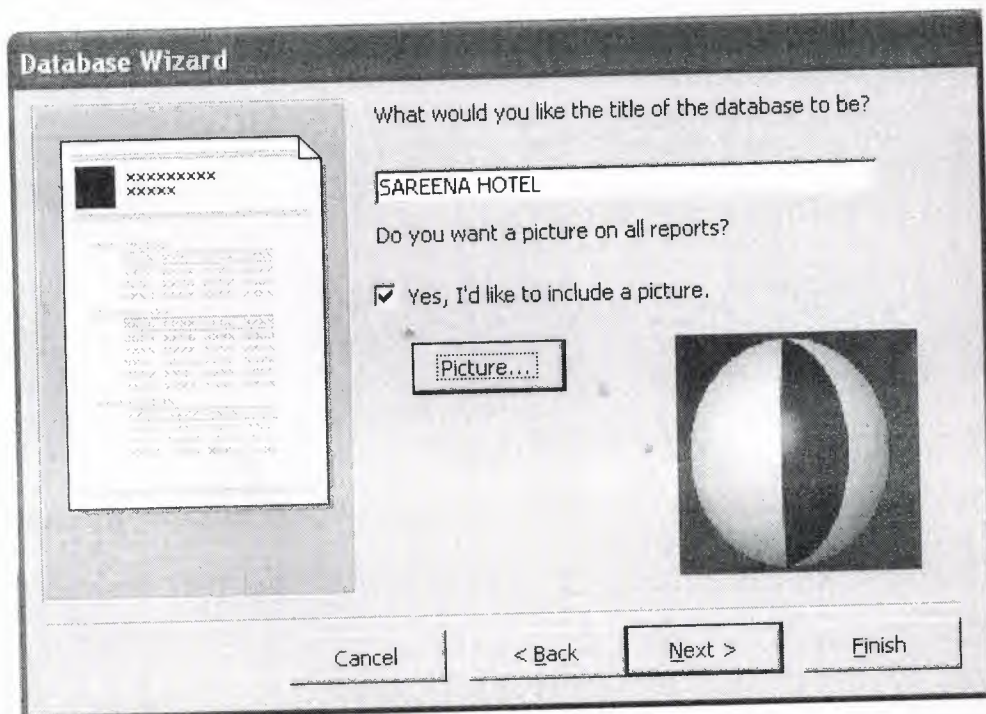
Style selection for the forms

The up coming image of the database wizard is about the Title of my project.



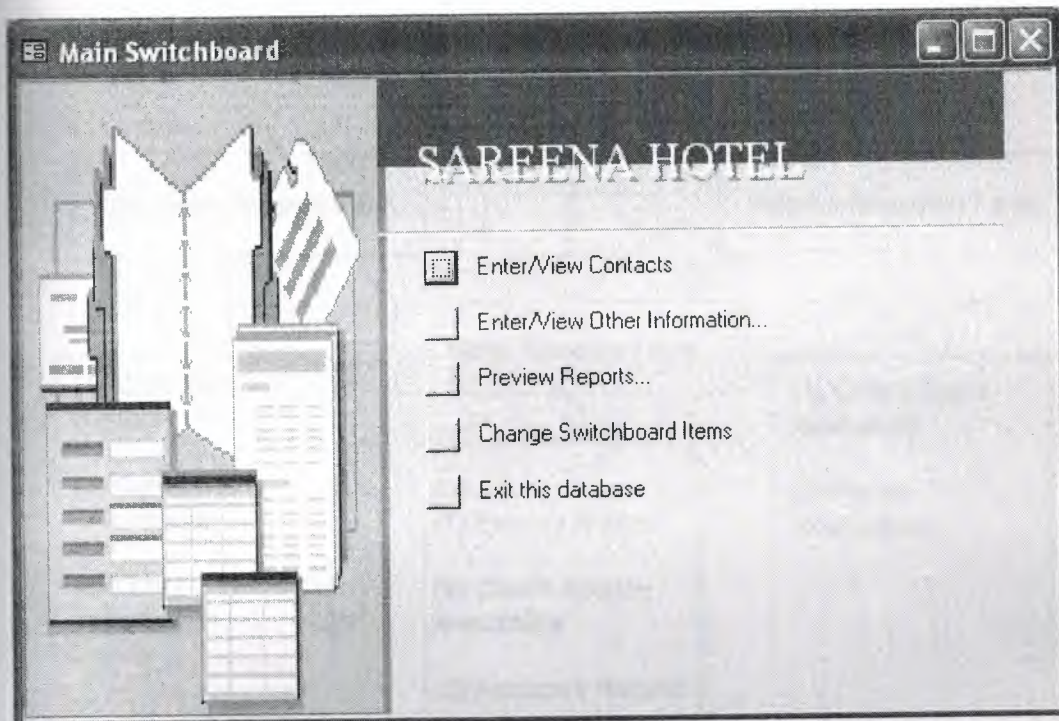
Title window

I added the image of my project of Sareena Hotel by using the same window mentioned above. Let us see how I did that.



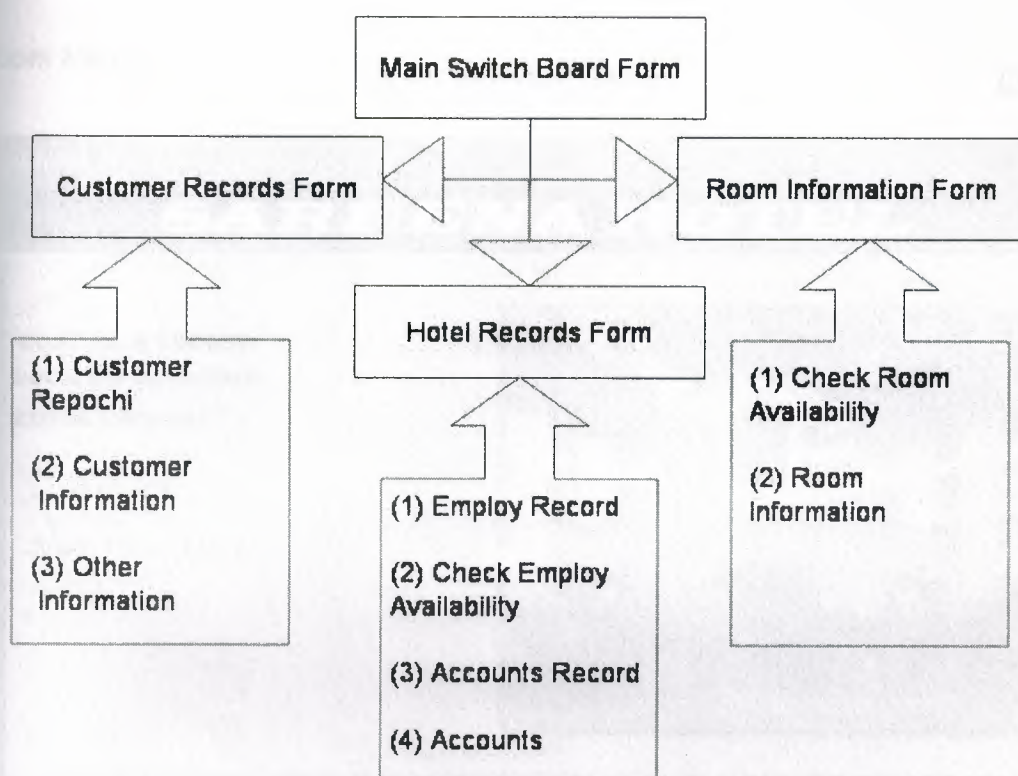
The title window with Image

In the above mentioned database wizard, we can choose different options for table and forms easily. The format of the forms can be chosen and we can also upload the desired format. After the whole process end, we can see a new window given below.



Window displayed after completing database wizard

4.2 Pictorial representation of my project



The above picture represents my project. Here I linked main switch board to three main sub forms. The idea of my project is about the Reservation system. So I decide that first the customer will come and the receptionist will check either the room is available or not. After this he/ she will enter the other data's regarding to reservation system.

I also provide special and personal information regarding to just about the hotels. Such like the accounts information. And I provide the facility in this reservation system that the manager can check either employee is available or not. And some personnel information about the employee.

In the room information form I provide a facility that the customer can check the room preview. I provide different types of room.

In this project I tried to make an easy and efficient way to make the reservation.

Starting my project

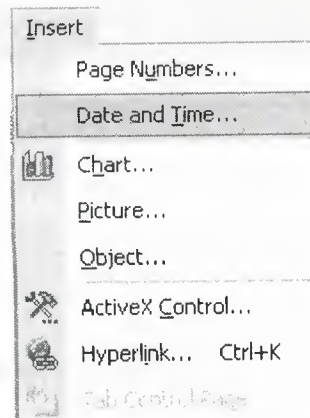
When I open the project the first thing you can see is the main Switchboard which consists of three buttons, a pictures of Sareena Hotel, an exit button and the current time and the date of the system. Every thing you can see is designed from the form I discussed above which comes after finishing the database wizard. The main

switch board form was designed by me and then I saved the format of the form and in the next forms I used the same format for the synchronization of my project. Let me show you the pictures of the main Switchboard of my project.



Main switch board form

The silver border on the left and bottom are labels and the border which contains the name of the hotel is also label. The date and time are the built in option by Microsoft access which I get from the following way.



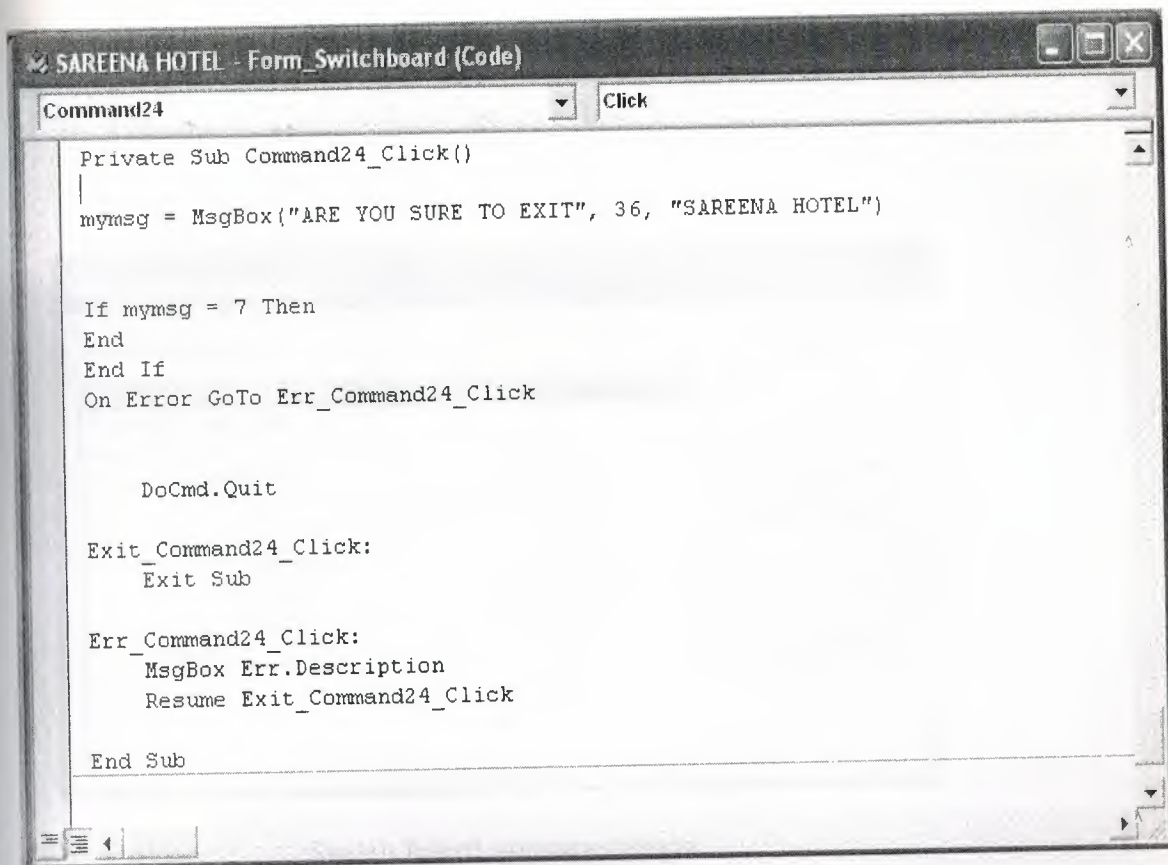
Option of inserting date and time

The Button on the right end bottom is for exiting from the application when the button is clicked the message box to the user is displayed having two options of YES or NO. If the user clicks on the yes option the application will be quit. And if the user select the option No the message box will be disappear and the user comes back to the main form. The picture of this event is as follows.



Exiting from the application through the exit button

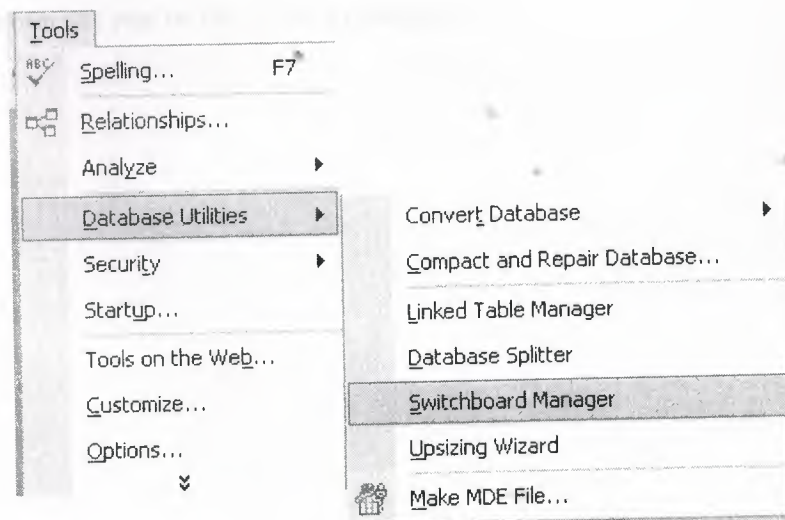
The coding of the exit button is done in the visual basic editor of Microsoft access which is mentioned in chapter 2. The coding for the exit button is as follows.



Visual basic editor coding for the exit button

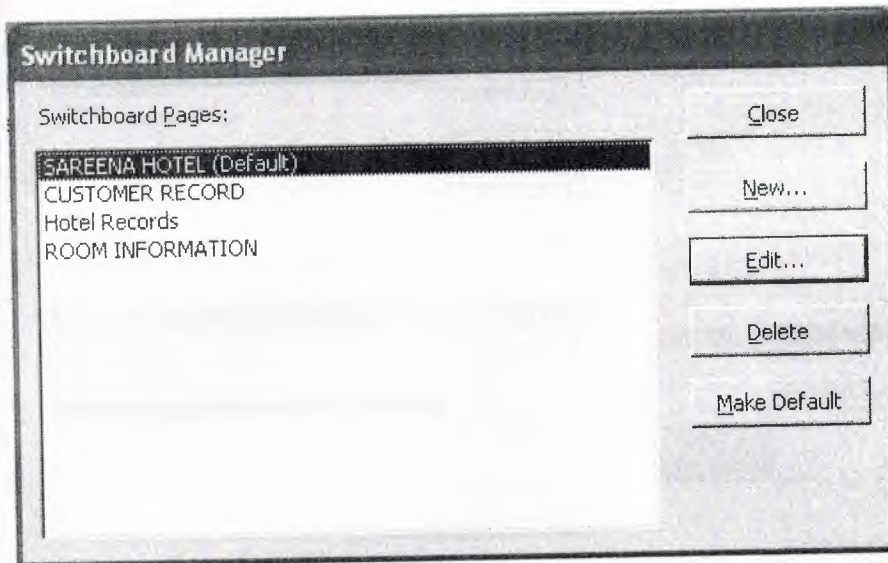
4.3 Main Customer Record Switch Board

The customer record switch board is linked with the main switch board by the help of switch board manager we can access the switch board manager from the menu bar.



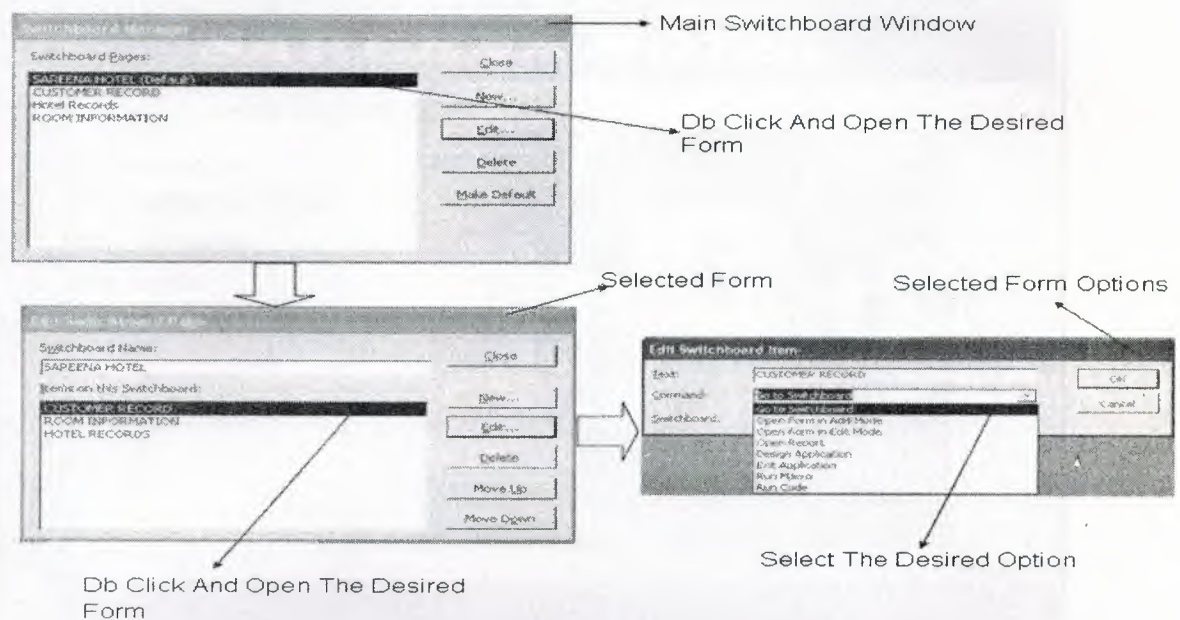
The way of accessing the switch board manager

After selecting the switch board manager option which is present in the sub menu of the tools Menu. You can see a new window shown below.



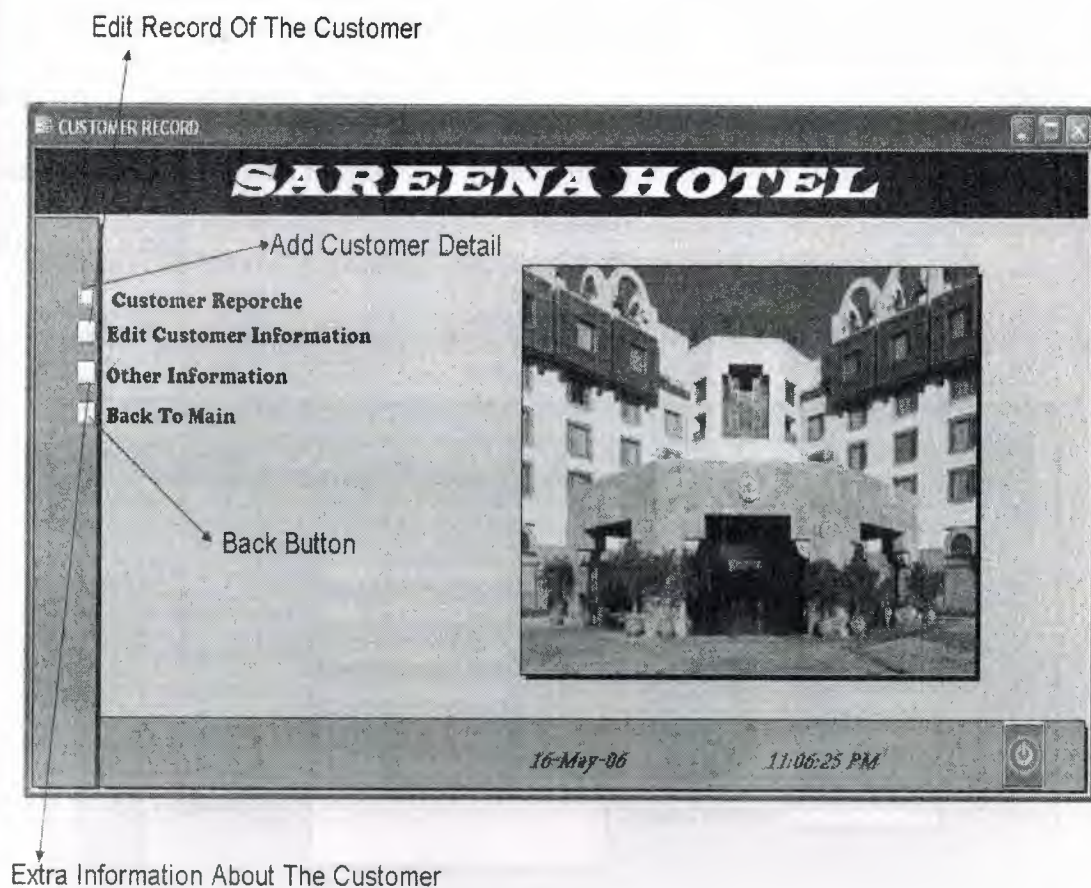
Switch board manager image

In the switch board manager window we can access all the switch board which we use in my project. New switch boards can also be added and can be deleted from the project. Each switch board contains different options which can also be viewed in the switch board manager. All the linking in my project is done by the switch board manager. Let us see one of the linking processes.



Linking of switch board and forms

Above picture shows us the complete way of linking forms and switch board with the help of switch board manager. The customer record switch board is also linked by the same process mentioned above and all the forms of the customer record switch board are linked with the same option of Microsoft access switch board manager. The image of the customer record switch board is given below.



Customer record switch board

The same format is followed which is applied in the main switch board. The behavior of the exit button is same as we study above and the date and time is also inserted in the same pattern. The only difference we can see is that the customer form switch board has its on following options.

- Customer reporche
- Edit customer information
- Other information
- Back to main

Let us see the options in details.

4.3.1 Customer reporche

The first option we see in the customer record switch board is customer reporche. The customer reporche contains the total detail of the customer e.g. Name, surname address phone number and country etc. Let us have a look on the customer reporche form.

The screenshot shows a software window titled "Customer Records" with a header "SAREENA HOTEL". The form contains the following fields:

Field	Value
Customer ID	
First Name	
Middle Name	
Last Name	
Country	
City	
Address	
Staying From:	
Staying Till:	
Paid:	\$0.00
Dues:	\$0.00
Home Phone	
Mobile Phone	
Email Address	

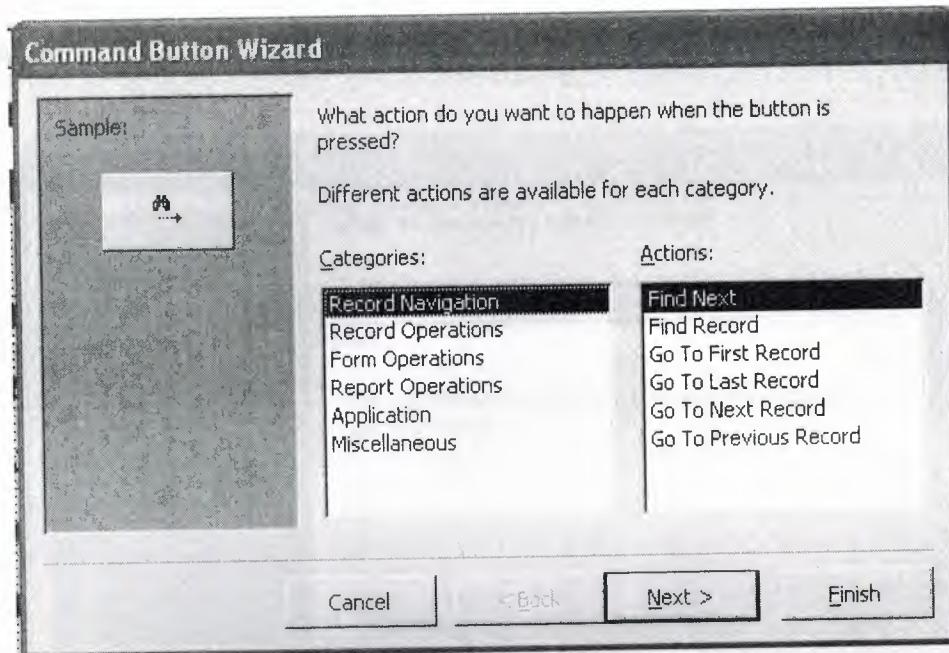
At the bottom left, there are "Record Navigators" with a "Record: 1 of 1" indicator. At the bottom right, there is a "Back To Sub Menu" button.

Customer reporche form

The form carries all the information about the customer. As you can see in the lower right corner of the form we have a back button. The job of the back button is to exit from the current form and go back to the linking form. The back button is inserted by the following way.

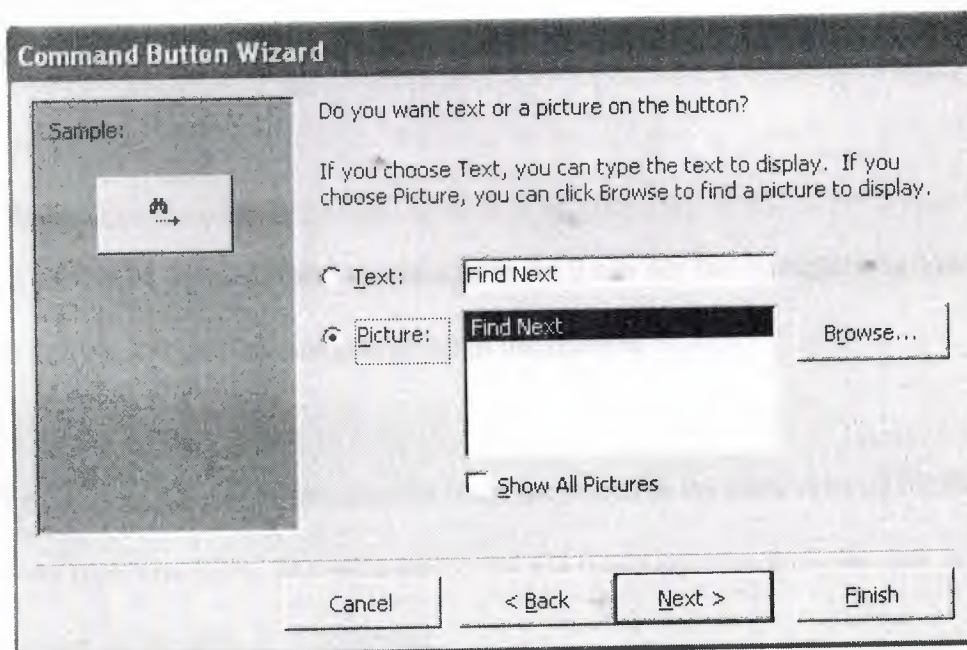
- Open the form in the design view
- Select the command button from the tool box at the left side of the window

- Bring the mouse pointer on the form and drag the pointer
- When the click is released you will see a window.



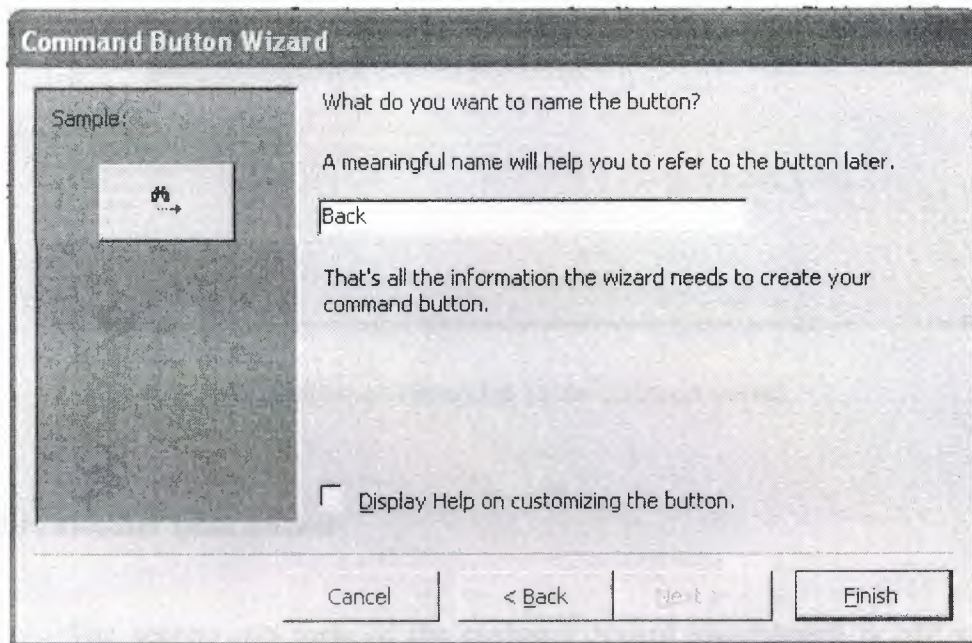
Command button wizard

- Select the desired option and click next, this kind of window will display.



Command button wizard step2

- Choose the picture or browse the pictures for command button and click next.
- I made a new image for my project and used that image with the help of command button wizard.



Command button wizard step3

- Give the name for the command button which is used in the coding.
- At the last click finish button.

At the bottom of the customer reporce form we u can see the Navigator buttons. These buttons are used to navigate through the records.

All the fields in the customer reporce form are stored in the table created for the customer reporce form. The table carries all the fields and stores all the data in it. The picture of the table is as followed.

Field Name	Data Type	Description
CustomerID	Number	
FirstName	Text	
MiddleName	Text	
LastName	Text	
Address	Text	
City	Text	
Country/Region	Text	
HomePhone	Text	
MobilePhone	Text	
EmailAddress	Text	
Staying From	Date/Time	
Staying Till	Date/Time	
Paid	Currency	
Dues	Currency	

Field Properties	
General	Lookup
Format	Currency
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	No
Indexed	No
Smart Tags	

This property cannot be modified in linked tables.

Customer reporcche table (Design view)

Edit customer information

The second sub form of the customer record switchboard is edit customer information. This form is similar to the customer reporcche form the only difference is that we can only add the customer data in the customer reporcche form but in editcustomer information form we can view the customer records and can also edit the records. All of the otter things are same as we see in the customer reporcche form.

The image of the form is as fallows.

SAREENA HOTEL

Customer ID: 12 Staying From: 12/1/2006

First Name: ahmed Staying Till: 2/24/2006

Middle Name: Paid: \$1,000.00

Last Name: iqbal Dues: \$0.00

Country: cyprus Home Phone: Mobile Phone: Email Address:

City: Jeskosa

Address: cyp

Record: 14 of 16

Annotations: Last Record (button), Total Records (16), Record Navigators (navigation buttons)

Edit Customer Record Form

We can see in the bottom of the form that there are record navigators and there is another button which is highlighted which take the control to the last record and the total records are also shown at the bottom of the form. The rest of the pattern is same as we see in the customer reporché form. The table of customer reporché is used for this form because the job of this form is only to display the data not to add it in the tables.

Other information

The other information form consists the extra information about the customer. It holds information like Service charges, paying method such like how a customer wants to make his payment, Transaction Id etc.

The picture of all the fields on form is given below.

Accounts Checking Form

SAREENA HOTEL

Transaction ID	<input type="text"/>	Service Charge	<input type="text"/>
Payment ID	<input type="text"/>	Tax	<input type="text" value="\$0.00"/>
Transaction Number	<input type="text"/>	Employee ID	<input type="text"/>
Transaction Date	<input type="text"/>		
Transaction Description	<input type="text"/>		
Transaction Amount	<input type="text"/>		
Account ID	<input type="text"/>		
Reference Number	<input type="text"/>		
Withdrawal Amount	<input type="text" value="\$0.00"/>		
Deposit Amount	<input type="text" value="\$0.00"/>		

Accounts Entry Form

Record: 2 of 2

Record Navigators

Open Accounts Entry Form

Back To Submenu

Accounts checking form

The table of other information opened in the design view is as followed.

Accounts : Table		
Field Name	Data Type	Description
TransactionID	Number	
PaymentID	Number	
TransactionNumber	Number	
TransactionDate	Date/Time	
TransactionDescription	Text	
TransactionAmount	Currency	
AccountID	Number	
ReferenceNumber	Text	
WithdrawalAmount	Currency	
DepositAmount	Currency	
ServiceCharge	Currency	
Tax	Currency	
Employee ID	Number	

Field Properties	
General	Lookup
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	Transaction ID
Default Value	
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (No Duplicates)

This property cannot be modified in linked tables.

Table of other information

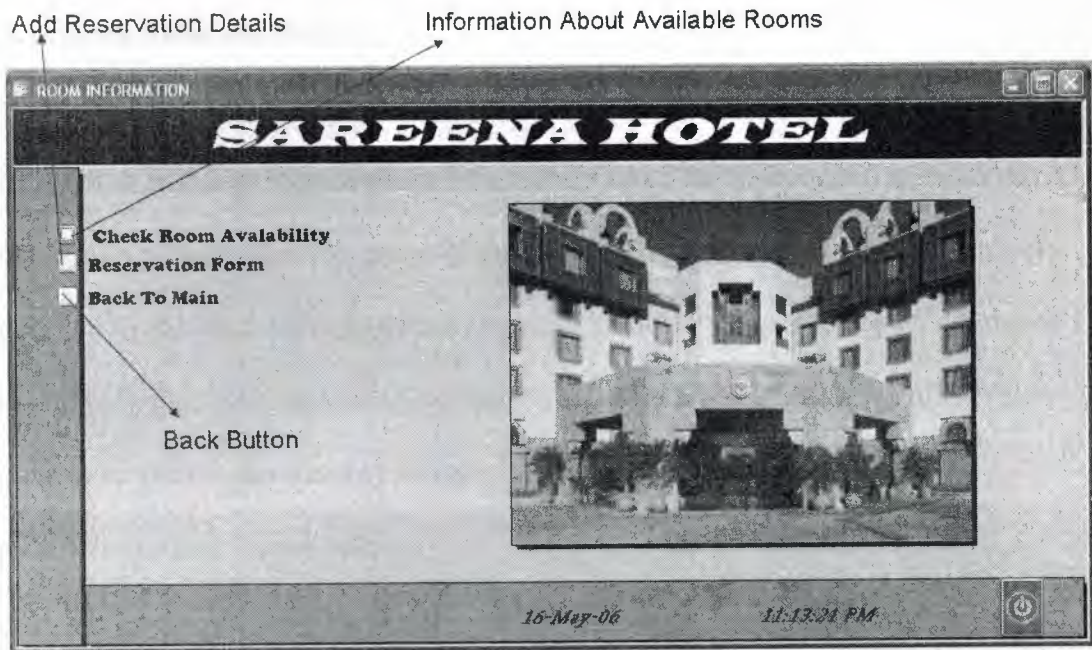
Back to main

The back to main option on the customer record switch board is linked with the main switch board when the user click the button back to main the customer record switch board exit and the main switch board appears.

4.4 Room information switch board

The second option on the main switch board is room information when the user wants to see the information about the rooms availability or the user wants to reserve the room he / she approaches to this button. The button is linked with room

information switch board with the help of switch board manager. The way of linking the switch boards is mentioned above at the start of this chapter.



Room information switch board

We can see the format of the switch board is same like the main switch board. All the options on room information switch board are same only the difference is that in room information switch board we have two different options.

- Check room availability
- Reservation form
- Back to main

Let me describe these options in details.

4.4.1 Check room availability

The first option in the room information switch board is room availability. The room availability form consists information about the room types, availability of room, room preview, and the prices of the rooms. The room availability form is very important in the hotel database system because it allows the user to search the rooms very quickly that either the room is available or not. The preview option allows the customer to choose the room of his own likings. The preview of the Room availability form is shown below.

Select Room Preview

The screenshot shows a web application window titled "Room Information Checking Form" for "SAREENA HOTEL". The form contains several input fields: "Room Number", "Room Type", "Room Information", "Rent/Night", "Availability" (with a checkbox), "ReservationID", and "Search By Room Type" (a dropdown menu). To the right of these fields is a "Room Preview" section displaying four small images of hotel rooms. Below the form is a "Direct Access to Reservation Form" button. The window also has a status bar at the bottom showing "Record: 14 of 20".

Room Preview

Direct Access to Reservation Form

Room availability checking form

We can see a lot of new things in this form. Let us discuss these new option one by one.

- The preview of the room is displayed on the right side of the form which is done by the help of picture tool in the tool box of Microsoft access.
- The combo box in the form is quick access to see the rooms. I provided the different types of room and their pictures with prices. When the user selects the preview of the room is displayed at the right side of the form in the room preview.

Room Information Checking Form

SAREENA HOTEL

Room Number: 103

Room Type: Single

Room Information: Normal

Rent/Night: \$45.00

Availability: ☐

ReservationID:

Search By Room Type: Single (dropdown menu open showing: Single, Double, Single, Double, Single, Double, Double, Double)

Record: 2 of 20

Room information checking form

As we seen the picture above that when the user clicks the single option in the combo box the preview of the combo box changes and all the fields above are filled such like rent for room, room type, availability and the room status etc. Every room has its own property when the room is selected the fields which shows the property of the room also change.

Room Information Checking Form

SAREENA HOTEL

Room Number

Room Type

Room Information

Rent/Night

Availability ☐

ReservationID

Search By Room Type

Go To Reservation!

Record: 1 of 20

We can also see a new button at the bottom left in above picture. I provide a direct way to access the reservation form. When the user will write the Reservation ID the Go to reservation link will be activated automatically. Let's have a demonstration about this button.

Reservation Form

SAREENA HOTEL

Reservation ID

Customer ID

Employee ID

Reservation Date

Amount Paid

Go To Reservation!

Record: 1 of 4

Demonstration for Go to reservation form

As we see that when we give the reservation ID for the customer the reservation form will be opened automatically. The rest of the events have been described before such like back button that has been described before etc.

Now I will show you the table for this form.

Field Name	Data Type	Description
Room Number	Text	
Floor Number	Text	
Room Type	Text	
Room Information	Text	
Rent/Night	Currency	
Availability	Yes/No	
ReservationID	Number	
Room view	OLE Object	

Field Properties	
General	Lookup
Field Size	50
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None

This property cannot be modified in linked tables.

Room information table

4.4.2 Reservation Form

As we see above that there is immediately source to access the reservation form but in my project I give that access also in the main window of room information to make it easy as possible as. The contents of this form are exactly same. Now let's have a look for this form.

Reservation Form

SAREENA HOTEL

Reservation ID
Customer ID
Employee ID
Reservation Date
Amount Paid
Total Due
Confirmed

Record: 1 of 4

Reservation table

Reservations : Table

Field Name	Data Type	Description
ReservationID	Number	
CustomerID	Number	
EmployeeID	Number	
ReservationDate	Date/Time	
AmountPaid	Currency	
TotalDue	Currency	
Confirmed	Yes/No	
TransactionID	Number	

Field Properties

General Lookup

Field Size: Long Integer

Format: Auto

Input Mask: Reservation ID

Default Value:

Validation Rule:

Validation Text:

Required: No

Indexed: Yes (No Duplicates)

Smart Tags:

This property cannot be modified in linked tables.

Reservation table

4.5 Hotel Records

This form keeps the information about the hotel records. I provide this form in the main and there are four more sub forms which are being linked with this main form. The purpose of creating these forms is to keep all necessary information regarding to the hotel records such like details about employee, accounts etc. The fields related to this switchboard are as fallows.

- Employee Details
- Check Employee Availability
- Accounts Records
- Accounts

The details of these forms are given below.


4.5.1 Employee Details

This sub form keeps the detail for the employee i.e. employee name, address, salary etc. I also provide the picture format to immerge the picture of the employee. Now let's have a demonstration for this form.

Employee Preview

Employee Details

SAREENA HOTEL

Employee ID: Photograph: 

Name

Job Title

Department

Address

Join Date

Salary

Record: 14 of 6

Save Details

Employee detail sub form

For this form I made a button to save the details of the employee. The rest of things are same as previous such like a back button which help us to go to back main window of hotel's record. And also the navigators which shows the totals record.

Now I will show you the table for the employee details.

Field Name	Data Type	Description
Employee ID	Number	
Name	Text	
Job Title	Text	
Department	Text	
Address	Text	
Join Date	Date/Time	
Salary	Currency	
On Duty	Yes/No	
Room number	Text	
Photograph	OLE Object	

Field Properties	
General	Lookup
Field Size	Long Integer
Format	Auto
Decimal Places	0
Input Mask	<>0
Caption	Put A Non-Zero Value
Default Value	No
Validation Rule	Yes (No Duplicates)
Required	Yes
Indexed	Yes
Smart Tags	No

This property cannot be modified in linked tables.

Employee details table

4.5.2 Check Employee Availability

This sub form is also being linked to the hotel's record form. The purpose of creation for this form is that we can easily check either employee is available or not by his or her employee id. I also create a search option. By the help of this option we can access the availability of the employee. Now have a look for this form.

Employees Available Check Form

SAREENA HOTEL

Employee ID:

Name

Job Title

On Duty ☐

Room number

Search By ID

Record: 1 of 6

Employee available check form

I also provide the designation of the employee i.e. administrator, manager etc. and I also provide a combo box as room number. By the help of this we can check which employee is on duty at the specific room. And the rest of the things are same as previous forms such like a back button to go to back and the navigators, etc.

4.5.3 Accounts Record

This sub form keeps the information about the accounts. This form is related with the hotel so that way I linked this form with the hotel records. This form consists of many fields related to the transactions, the mode of the payment by the customer, some necessary charges etc. now I will show the preview of this form.

Accounts Entry form

SAREENA HOTEL

Transaction ID

Payment ID

Transaction Number

Transaction Date

Transaction Description

Transaction Amount

Account ID

Reference Number

Withdrawal Amount

Deposit Amount

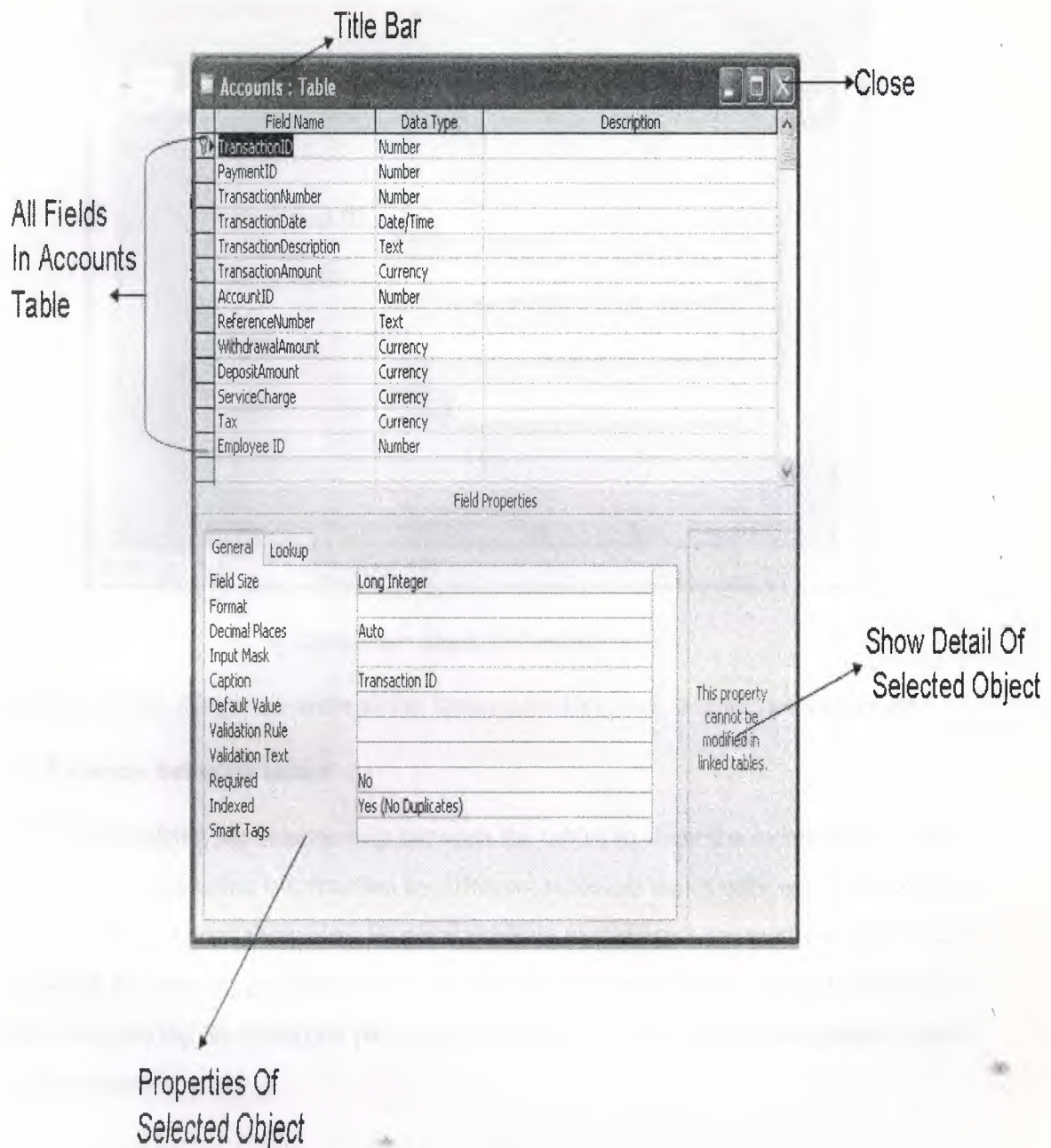
Service Charge

Tax

Record: 1 of 1

Accounts entry form

As I mentioned above that this form is related with accounts. It keeps all the information related with accounts such like account id, account date service charges etc. The rest of things are similar as the previous form e.g. back button, navigators and the save button to save the records. Now let's have a look on the table.



Accounts table

4.5.4 Accounts

This sub form is also being linked with the hotel record form. By the help of this form we can check the amount, name of the customer by entering the customer id. The purpose of creation of this form is that we can check the dues and paid amount of the customer immediately. Now let's have a look on the form.

The screenshot shows a software window titled "Accounts (Customer Payment Entry Form)" with a standard Windows-style title bar. Below the title bar is a header area with the text "SAREENA HOTEL" in a large, bold, serif font. The main content area of the form contains four labeled input fields: "Customer ID" with the value "1", "First Name" (empty), "Paid" (empty), and "Deus" with the value "\$0.00". At the bottom of the window, there is a status bar that reads "Record: 1 of 16" and includes navigation icons for first, previous, next, and last records.

Customer payment entry form

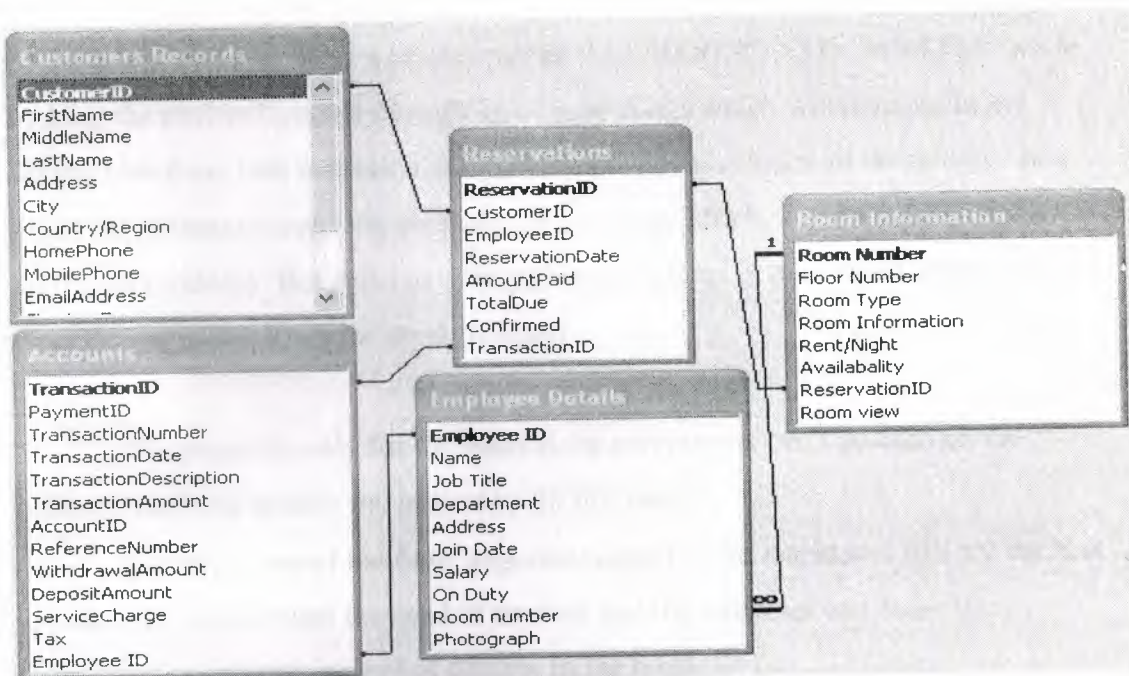
The rest of the things are same as the forms such like back button, navigators etc.

4.6 Relation between tables

I construct the relationship between the tables to share the information. Some time we do need some information by different tables so that's why we construct the relationship between the tables. Its good practice to construct the relationship because by doing this we can get information quickly and the same. In my project the relations are playing an important part let me describe the job of the relationships I made in the project.

- ❖ First of all we can see the relationship between the Customers Records form with the Reservation form. The Customer ID field in the Customer Records form is linked with the same field in the Reservation form. Actually the job of the relationship is to make the connection between two tables and with the help of related field the desired information is shared.
- ❖ The second related form is Accounts form with the Reservation form and with Employee Details form, so the Accounts form is multi related form because it is sharing information with two different forms. The transaction ID field of the Accounts form is related with the same field of the Reservation form and the

Employee ID field of Accounts form is related with the same field in the Employee Detail form as we can see in the figure below.



Forms Relationship Image

- ❖ The Reservation form is also related with the Room Information form which shares the field Reservation ID. The Reservation Form is also a multi linked form which is sharing information with three different forms Customer Records, Accounts and Room Information form.
- ❖ The relation between the Room Information form is different then the other relationships because Room Information field of the Room information form is linked with all the Employee Detail form as you can see it in the figure above.

FUTURE DEVELOPMENT

I have given a task to make a project on "HOTEL RESERVATION SYSTEM" while making the project I passed through lot of new things which will help me in my future. I tried my best to make a perfect project which includes all the things, I had given the options to make my project easy and more efficient and I tried to make it more users friendly. But still I have to add more options in my project which can meet the requirements of the modern world.

The project is only for the stand alone computer it can't be used on the network, so I will modify my project to fill this need.

Internet is one of the most important aspect of the world so I will try my best to make my project meet the need of internet and the customer can make there reservation on internet instead of coming to the hotel.

I will add more forms in my project so it can store more information about the hotel and customer record.

In my project the customer can only pay in cash but I will make some modification to make it more flexible for the customer in the payment.

The above mentioned points are the future development process I will make after my graduation degree to make my project more efficient and user friendly.

CONCLUSION

Microsoft access is a power full database management tool provided by Microsoft organization with the help of access we can create forms, tables, queries, etc. It is an easy way to approach the user requirements.

In the project HOTEL RESERVATION SYSTEM it can store information about the customer record, employee details and the accounts management system of hotel. The program is capable of storing all the information about the customer and it works like a 'Manager' of a hotel.

The HOTEL MANAGEMENT SYSTEM program is made for the small level of business, because Microsoft Access is not very efficient or reliable for handling big projects.

The Access is the easiest way for making a small level program and it provides a lot of options in editing and designing a project e.g. inserting button in the form, providing a GUI (Graphical User Interface) to the end user. The Access database management system is very easy to handle for the end user and for the new users.

I used SQL (Structured Quire Language) which is a very powerful language used for data base design now-a-days.

REFRANCES

Books:-

- | | |
|---|---------------|
| SQL an Interactive Approach | Ijaz Ahmed |
| Implementation RDBMS with SQL | APTECH |
| Microsoft Access 2002 Visual Basic: For Applications Step by Step | Evan Callahan |

WEB Pages

www.wikipedia.com

www.google.com

www.microsoft.com

www.aptech-pakistan.com

www.aptech-globaltraining.com