Contents

What is a database?

A database is a tool for collecting and organizing information. Databases can store information about people, products, orders, or anything else. Many databases start as a list in a word-processing program or spreadsheet. As the list grows bigger, redundancies and inconsistencies begin to appear in the data. The data becomes hard to understand in list form, and there are limited ways of searching or pulling subsets of data out for review. Once these problems start to appear, it's a good idea to transfer the data to a database created by a database management system (DBMS), such as Office Access 2007.

A computerized database is a container of objects. One database can contain more than one table. For example, an inventory tracking system that uses three tables is not three databases, but one database that contains three tables. Unless it has been specifically designed to use data or code from another source, an Access database stores its tables in a single file, along with other objects, such as forms, reports, macros, and modules. Databases created in the Access 2007 format have the file extension .accdb, and databases created in earlier Access formats have the file extension .mdb. You can use Access 2007 to create files in earlier file formats (for example, Access 2000 and Access 2002-2003).

Using Access, you can:

- Add new data to a database, such as a new item in an inventory
- Edit existing data in the database, such as changing the current location of an item
- Delete information, perhaps if an item is sold or discarded
- Organize and view the data in different ways
- Share the data with others via reports, e-mail messages, an intranet, or the Internet

What is data, database (DB), DBMS and DBS?

In computer science, data is anything in a form suitable for use with a computer. Data is often distinguished from programs. A program is a set of instructions that detail a task for the computer to perform. In this sense, data is thus everything that is not program code.

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images.

In computing, databases are sometimes classified according to their organizational approaches. The most prevalent approach is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways. A distributed database is one that can

be dispersed or replicated among different points in a network. An object-oriented programming database is one that is congruent with the data defined in object classes and subclasses.

As one of the oldest components associated with computers, the database management system (DBMS), is a computer software program that is designed as the means of managing all databases that are currently installed on a system hard drive or network. Different types of database management

systems exist, with some of them designed for the oversight and proper control of databases that are configured for specific purposes. Here are some examples of the various incarnations of DBMS technology that are currently in use, and some of the basic elements that are parts of DBMS software applications.

What is an Electronic Database?

An electronic database is a collection of data arranged in a systematic way to make the search easy and fast. In other words it is a computer-based collection or listing of information, usually organized with searchable elements or fields. The most common type of library database consists of records describing articles in journals or newspapers. Retrieval from this information store is basically accomplished through a matching process. The process of matching customer's query against information in databases is the essence of computerized information retrieval.

Why use electronic databases?

Electronic Databases are the major sources of information. Increasingly, information is being published only in electronic formats. There is coverage in virtually all areas of knowledge: science, engineering, mathematics, medicine, agriculture, psychology, sociology, philosophy, law, business, economics, education and more. A single database may refer to a variety of sources, including periodical articles, books, government documents, industry reports, papers at meetings, newspaper items, films, video recordings etc.

With a terminal with Internet connection you can be seconds away from receiving valuable information on any imaginable topic. These rich and voluminous databases are stored in by the information companies or agencies known as retrieval services. Once you are connected to them, much of the world's knowledge literally will be at your fingertips.

As an information source a database may be shared by thousands of users simultaneously, and it is available whenever the retrieval service is in operation. There is no limit to the number of times a database can be searched or the number of times an item can be displayed. Unlike a library book, databases do not deteriorate physically, nor can they be misplaced, stolen, or vandalized.

Database - Advantages & Disadvantages

Advantages

1. Reduced data redundancy
2. Reduced updating errors and increased consistency
3. Greater data integrity and independence from applications programs
4. Improved data access to users through use of host and query languages
5. Improved data security
6. Reduced data entry, storage, and retrieval costs
7. Facilitated development of new applications program

Disadvantages

1. Database systems are complex, difficult, and time-consuming to design
2. Substantial hardware and software start-up costs

3. Damage to database affects virtually all applications programs
4. Extensive conversion costs in moving form a file-based system to a database system
5. Initial training required for all programmers and users

Database Management System:

In addition to the data itself, a set of program is necessary to facilitate adding new data as well as modifying and retrieving existing data within a database. This set of programs is referred to as database management system (DBMS).

Generally, the user of a database management system accesses the database via a special query language or via applications programs written in a high-level language. The applications program utilizes special host- or command-language instructions to communicate requests for data to the portion of the DBMS referred to as the database processing system. Basically, the database processing system consists of a series of programs that translate requests from users or user programs into the instructions necessary to access, add to, change, or delete records or items of data from database. Thus, an applications program need include only a command to access the desired data element. The actual mechanics of locating and accessing the data element are accomplished by the database processing programs.

1. Accessing Data from a database in response to an applications program request.
2. Application program instruction initiates a request for data
3. Control unit transfers control to DBMS
4. DBMS verifies that the request is valid and determines location of data element in database
5. DBMS issues command to access data from secondary storage
6. Data element is accessed and stored in a buffer of the DBMS
7. DBMS transfers data element to application program storage area
8. Next instruction in application program is accessed and executed

DBMS Functions

There are several functions that a DBMS performs to ensure data integrity and consistency of data in the database. The ten functions in the DBMS are: data dictionary management, data storage management, data transformation and presentation, security management, multiuser access control, backup and recovery management, data integrity management, database access languages and application programming interfaces, database communication interfaces, and transaction management.

1. Data Dictionary Management

Data Dictionary is where the DBMS stores definitions of the data elements and their relationships (metadata). The DBMS uses this function to look up the required data component structures and relationships. When programs access data in a database they are basically going through the DBMS. This function removes structural and data dependency and provides the user with data abstraction. In turn, this makes things a lot easier on the end user. The Data Dictionary is often hidden from the user and is used by Database Administrators and Programmers.

2. Data Storage Management

This particular function is used for the storage of data and any related data entry forms or screen definitions, report definitions, data validation rules, procedural code, and structures that can handle video and picture formats. Users do not need to know how data is stored or manipulated. Also involved with this structure is a term called performance tuning that relates to a database's efficiency in relation to storage and access speed.

3. Data Transformation and Presentation

This function exists to transform any data entered into required data structures. By using the data transformation and presentation function the DBMS can determine the difference between logical and physical data formats.

4. Security Management

This is one of the most important functions in the DBMS. Security management sets rules that determine specific users that are allowed to access the database. Users are given a username and password or sometimes through biometric authentication (such as a fingerprint or retina scan) but these types of authentication tend to be more costly. This function also sets restraints on what specific data any user can see or manage.

5. Multiuser Access Control

Data integrity and data consistency are the basis of this function. Multiuser access control is a very useful tool in a DBMS, it enables multiple users to access the database simultaneously without affecting the integrity of the database.

6. Backup and Recovery Management

Backup and recovery is brought to mind whenever there is potential outside threats to a database. For example if there is a power outage, recovery management is how long it takes to recover the database after the outage. Backup management refers to the data safety and integrity; for example backing up all your mp3 files on a disk.

7. **Data Integrity Management**

The DBMS enforces these rules to reduce things such as data redundancy, which is when data is stored in more than one place unnecessarily, and maximizing data consistency, making sure database is returning correct/same answer each time for same question asked.

8. **Database Access Languages and Application Programming Interfaces**

A query language is a nonprocedural language. An example of this is SQL (structured query language). SQL is the most common query language supported by the majority of DBMS vendors. The use of this language makes it easy for user to specify what they want done without the headache of explaining how to specifically do it.

9. **Database Communication Interfaces**

This refers to how a DBMS can accept different end user requests through different network environments. An example of this can be easily related to the internet. A DBMS can provide access to the database using the Internet through Web Browsers (Mozilla Firefox, Internet Explorer, Netscape).

10. **Transaction Management**

This refers to how a DBMS must supply a method that will guarantee that all the updates in a given transaction are made or not made. All transactions must follow what is called the ACID properties.
What is Table in database?

In database terms, a table is responsible for storing data in the database. Database tables consist of rows and columns.

In the following example, the second row is highlighted in black:

DATA and DATABASE security

Database security can be defined as a system or process by which the "Confidentiality, Integrity, and Availability," or CIA, of the database can be protected. Unauthorized entry or access to a database server signifies a loss of confidentiality; unauthorized alteration to the available data signifies loss of integrity; and lack of access to database services signifies loss of availability. Loss of one or more of these basic facets will have a significant impact on the security of the database.

Threats and risks to databases have increased and therefore, the need for securing databases has also increased. Let's learn about the basic facets of database security, including assurance, integrity, availability, and confidentiality.

The majority of the companies store sensitive data in databases. However, database security is sometimes not given as much thought and effort as other areas of computer security. Hackers have been able to target large databases in recent years to obtain sensitive information like credit card

numbers and other personal information. It is important to protect databases against these risks, and this is where database security comes into place.

Data security is critical for most businesses and even home computer users. Client information, payment information, personal files, bank account details - all of this information can be hard to replace and potentially dangerous if it falls into the wrong hands. Data lost due to disasters such as a flood or fire is crushing, but losing it to hackers or a malware infection can have much greater consequences.

Thorough data security begins with an overall strategy and risk assessment. This will enable you to identify the risks you are faced with and what could happen if valuable data is lost through theft, malware infection or a system crash. Other potential threats you want to identify include the following:

1. Physical threats such as a fire, power outage, theft or malicious damage
2. Human error such as the mistaken processing of information, unintended disposal of data or input errors
3. Exploits from corporate espionage and other malicious activity
4. You can then identify areas of vulnerability and develop strategies for securing your data and information systems. Here are several aspects that need to be considered:
5. Just who has access to what data
6. Who uses the internet, email systems and how they access it
7. Who will be allowed access and who will be restricted
8. Whether or not to use passwords and how they will be maintained
9. What type of firewalls and anti-malware solutions to put in place
10. Properly training the staff and enforcing data security.

After the above analysis, you can then prioritize specific data along with your more critical systems and determine those that require additional security measures. It is also a good idea to layout a BCP (Business Continuity Plan) so that your staff is still able to work effectively if the systems happen to fail. Company risks and security implementations should be reviewed frequently to support changes such as the growth of your business and other circumstances.

Securing Data

Once you draw up a plan and assess your risks, it is time to put your data security system into action. Since data can be compromised in many ways, the best security against misuse or theft involves a combination of technical measures, physical security and a well educated staff. You should implement clearly defined polices into your infrastructure and effectively present them to the staff. Here are things that you may do:

1. Protect your office or data center with alarms and monitoring systems

2. Keep computers and associated components out of public view
3. Enforce restrictions on internet access
4. Ensure that your anti-malware solution is up to date
5. Ensure that your operating system is up to date
6. Fight off hacking attacks with intrusion detection technology
7. Utilize a protected power supply and backup energy sources

Mobile Data Security

Hand-held devices and laptop computers have become popular in the business environment. However, mobile computers are at a much greater risk of data loss through damage and theft. For this reason, different safeguards need to implement in addition to the security measures listed above.

1. Regularly backup data on removable media and safely store multiple copies
2. Activate password protection whenever the device is left alone
3. Never leave the device alone and visible in a vehicle
4. Protect the device from physical damage by transporting it in protective casing

Efficient data security involves numerous steps, many of which can be downright time consuming. On the other hand, I am sure you will agree that actually losing this important data could be much worse.

Relational Database Management System (RDBMS)

A DBMS that is based on relational model is called as RDBMS. Relation model is most successful mode of all three models. Designed by E.F. Codd, relational model is based on the theory of sets and relations of mathematics.
Relational model represents data in the form a table. A table is a two dimensional array containing rows and columns. Each row contains data related to an entity such as a student. Each column contains the data related to a single attribute of the entity such as student name.

One of the reasons behind the success of relational model is its simplicity. It is easy to
Understand the data and easy to manipulate. Another important advantage with relational model, compared with remaining two models is, it doesn't bind data with relationship between data item. Instead it allows you to have dynamic relationship between entities using the values of the columns. Almost all Database systems that are sold in the market, now- a-days, have either complete or partial implementation of relational model.

RDBMS = DBMS + REFERENTIAL INTEGRITY

META DATA

In database management system (DBMS), data files are the files that store the database information, whereas other files, such as index files and data dictionaries, store administrative information, known as metadata.

Database system is a system to achieve an organized, store a large number of dynamical associated data, facilitate for multi-user accessing to computer hardware, software and data, that it is a computer system with database technology.

Data Dictionary:

Data dictionary is simply a repository for information about the database- data definitions and characteristic such as usage, physical representations, ownership, authorization, and security.

1. The DBMS can access the data dictionary to determine the information it needs to operate. For example, the DBMS can access the data dictionary to
2. Determine if a data element already exists before adding. This reduces data redundancy
3. Change the description of a data field. For Example, to change the description of a 20-position alphanumeric field to a 25-character alphanumeric field, only the description in the data dictionary need to be modified.
4. Determine what relationship exists between the elements.
5. Determine what applications programs can access what data elements.

6. The data dictionary is also useful to programmers and system analysts.
7. A programmer can copy a definition directly from the data dictionary for use in an applications program. This guarantees greater accuracy with less work from the programmer.
8. If a data description is changed, the data dictionary can be consulted to determine all affected applications programs.

Relationship

A very simple computer system may be able to be supported by a very simple database design that only includes a single table. However, if the database design needs to be enhanced to support more complex requirements, the single table design would almost always end up being normalized into multiple tables linked together through relationships. This is required to reduce data redundancy and to improve efficiency.

There are 3 types of table relationships:

1. One-to-one relationships

2. One-to-many relationships

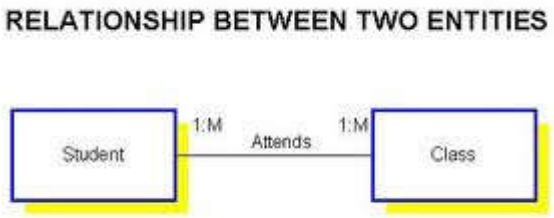
3. Many-to-many relationships

One-to-One Relationships

In a one-to-one relationship, each row in one database table is linked to one and only one other row in another table. In a one-to-one relationship between Table A and Table B, each row in Table A is

linked to another row in Table B. The number of rows in Table A must equal the number of rows in Table B.

It would be apparent that one-to-one relationships are not very useful since the database designer might as well simply merge both tables into a single table. This is true in general. However, there are some situations in which the one-to-one relationship may improve performance. For example, if a database table contains a few columns of data that is frequently used and the remaining columns being infrequently used, the database designer may split the single table into 2 tables linked through a one-to-one relationship. Such a design would reduce the overhead needed to retrieve the infrequently used columns whenever query is performed on the contents of the database table.

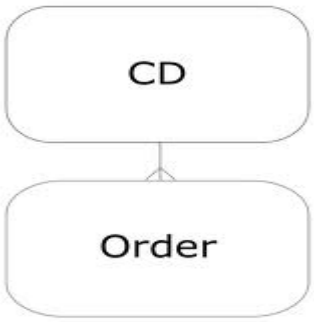


One to one relationship

One-to-Many Relationships

In a one-to-many relationship, each row in the related to table can be related to many rows in the relating table. This effectively save storage as the related record does not need to be stored multiple times in the relating table.

For example, all the customers belonging to a business is stored in a customer table while all the customer invoices are stored in an invoice table. Each customer can have many invoices but each invoice can only be generated for a single customer.

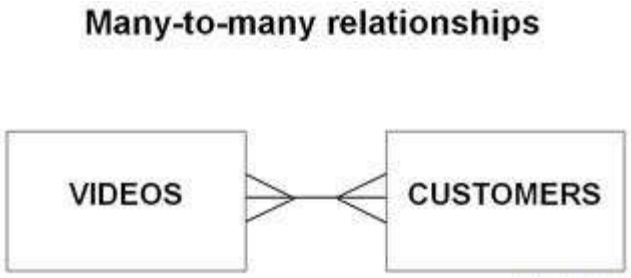


One to Many relationships

Many-to-Many Relationships

In a many-to-many relationship, one or more rows in a table can be related to 0, 1 or many rows in another table. A mapping table is required in order to implement such a relationship.

For example, all the customers belonging to a bank is stored in a customer table while all the bank's products are stored in a product table. Each customer can have many products and each product can be assigned to many customers



Referential Integrity:

In order to remove redundant data in database tables, the design of the database is normalized - almost always resulting in multiple tables linked together using foreign keys. This introduces a new problem because the database engine will need to keep track of the relationship to ensure that each foreign key must actually link to a valid value in another table. This is call referential integrity.

When referential integrity is enforced, all foreign keys are properly matched with existing data in the referenced table. This simply means that if Table A contains a foreign key that references to a record in Table B, the record in Table B must exist.

For example, consider the following tables: product table

==========

product_id (primary key)

description

category_id (foreign key)

category table

==========

category_id (primary key)

description

Let's assume that the above tables contain the following data:product_id description category_id

---------- ---------- ----------

0001 table 0001

0002 chair 0001

0003 fan 0002

category_id description

---------- ----------

0001 furniture

0002 electrical

What happens if the user tries to create a new record in the product table with a category_id of "0003"? There is no such record in the category table. As such, such a record in the product table will contain an invalid reference to the category table.

What happens if the user tries to delete the 1st record in the category table? Deleting the 1st record will effectively invalidate the category reference in the first 2 records in the product table.

If referential integrity is applied, both the above transactions would not be allowed by the database engine. The database engine enforces referential integrity through the use of referential constraints and triggers. Referential constraints prevent situations as describe in scenario 1 above from happening while triggers prevent situations as described in scenario 2 above from happening.

Entities

These are the people, places, things, events and concepts of interest to an organization. In short, anything which an organization needs to store data about. Entities are represented on the diagram by labeled boxes. Entities represent collections of things. For example, an EMPLOYEE entity might represent a collection of all the employees that work for an organization. Individual members (employees) of the collection are called occurrences of the EMPLOYEE entity.

Because the available space for naming the entity is restricted to the size of the box, Entities should always have detailed descriptions. These detailed descriptions are usually short paragraphs of text describing the entity in more detail but for some important entities, a lengthy description may be required.

Attributes

Entities are the things you're keeping track of. In an video rental database, you keep track of customers, invoices, titles, copies, etc. These are represented in tables where the rows are individual instances of a customer or title. The columns are the attributes, the things that tell us about the instance in the row.

The customer's name, address, city, balance, etc. are attributes that help identify the customer. An invoice's attributes might be price, number, date, paid/unpaid, etc.

Primary key

The primary key of a relational table uniquely identifies each record in the table. It can either be a normal attribute that is guaranteed to be unique (such as Social Security Number in a table with no more than one record per person) or it can be generated by the DBMS. Primary keys may consist of a single attribute or multiple attributes in combination.

Examples:

Imagine we have a STUDENTS table that contains a record for each student at a university. The student's unique student ID number would be a good choice for a primary key in the STUDENTS table. The student's first and last name would not be a good choice, as there is always the chance that more than one student might have the same name

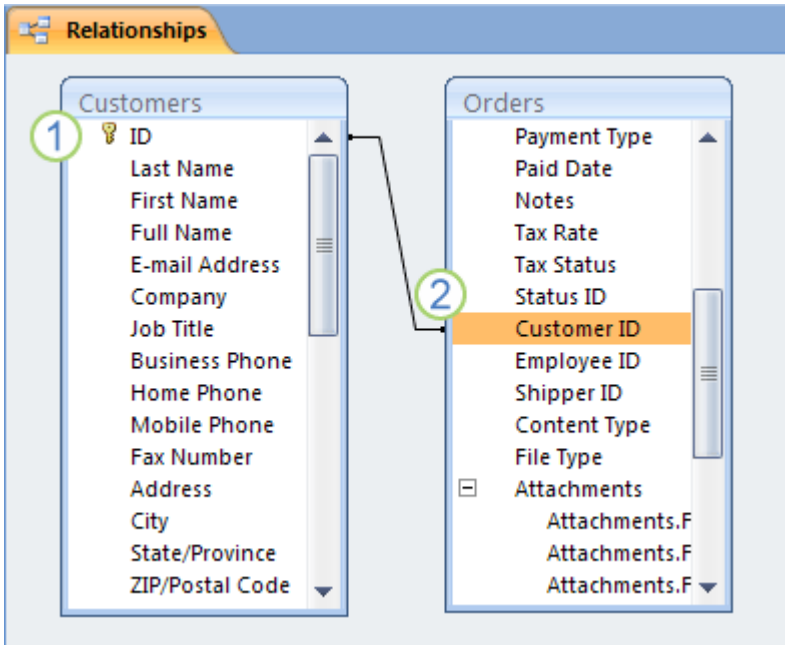
Foreign key

A foreign key is a field in a relational table that matches the primary key column of another table. The foreign key can be used to cross-reference tables.

A table can also have one or more foreign keys. A foreign key contains values that correspond to values in the primary key of another table. For example, you might have an Orders table in which each order has a customer ID number that corresponds to a record in a Customers table. The customer ID field is a foreign key of the Orders table.

The correspondence of values between key fields forms the basis of a table relationship. You use a table relationship to combine data from related tables. For example, suppose that you have a Customers table and an Orders table. In your Customers table, each record is identified by the primary key field, ID.

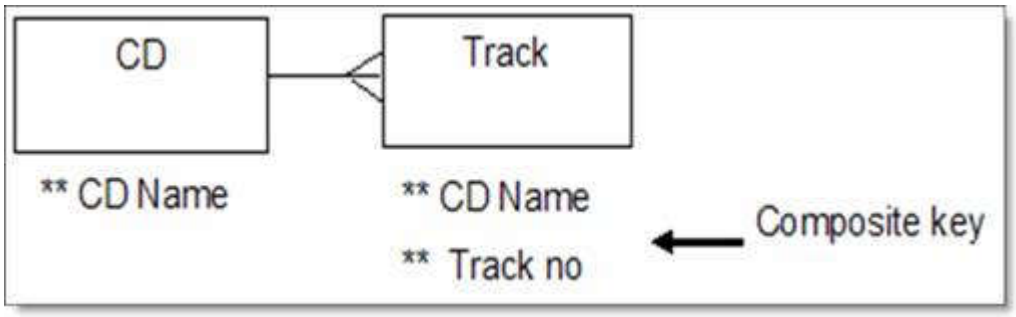
To associate each order with a customer, you add a foreign key field to the Orders table that corresponds to the ID field of the Customers table, and then create a relationship between the two keys. When you add a record to the Orders table, you use a value for customer ID that comes from the Customers table. Whenever you want to view any information about an order's customer, you use the relationship to identify which data from the Customers table corresponds to which records in the Orders table.



A table relationship, shown in the Relationships window.

Composite Key

A composite key consists of more than one attribute to uniquely identify an entity occurrence. This differs from a compound key in that one or more of the attributes, which make up the key, are not simple keys in their own right.



CD name in the track entity is a simple key, linking to the CD entity, but track number is not a simple key in its own right.

For example, you have a database holding your CD collection. One of the entities is called tracks, which holds details of the tracks on a CD. This has a composite key of CD name, track number.

Tuples (rows or records)

Definition: Tuple is a term from set theory which refers to a collection of one or more attributes.

Note: Using a (Relational) Database Software you define Relation constructing a TABLE, which you populate of Tuples that correspond to ROWS of such a table with COLUMNS as Attributes.

What is an Entity Relationship Diagram (ERD)?

ER-modeling is a data modeling technique used in software engineering to produce a conceptual data model of a information system. Diagrams created using this ER-modeling technique are called Entity-Relationship Diagrams, or ER diagrams or ERDs. So you can say that Entity Relationship Diagrams illustrate the logical structure of databases.

Dr. Peter Chen is the originator of the Entity-Relationship Model. His original paper about ER-modeling is one of the most cited papers in the computer software field. Currently the ER model serves as the foundation of many system analysis and design methodologies, computer-aided software engineering (CASE) tools, and repository systems.

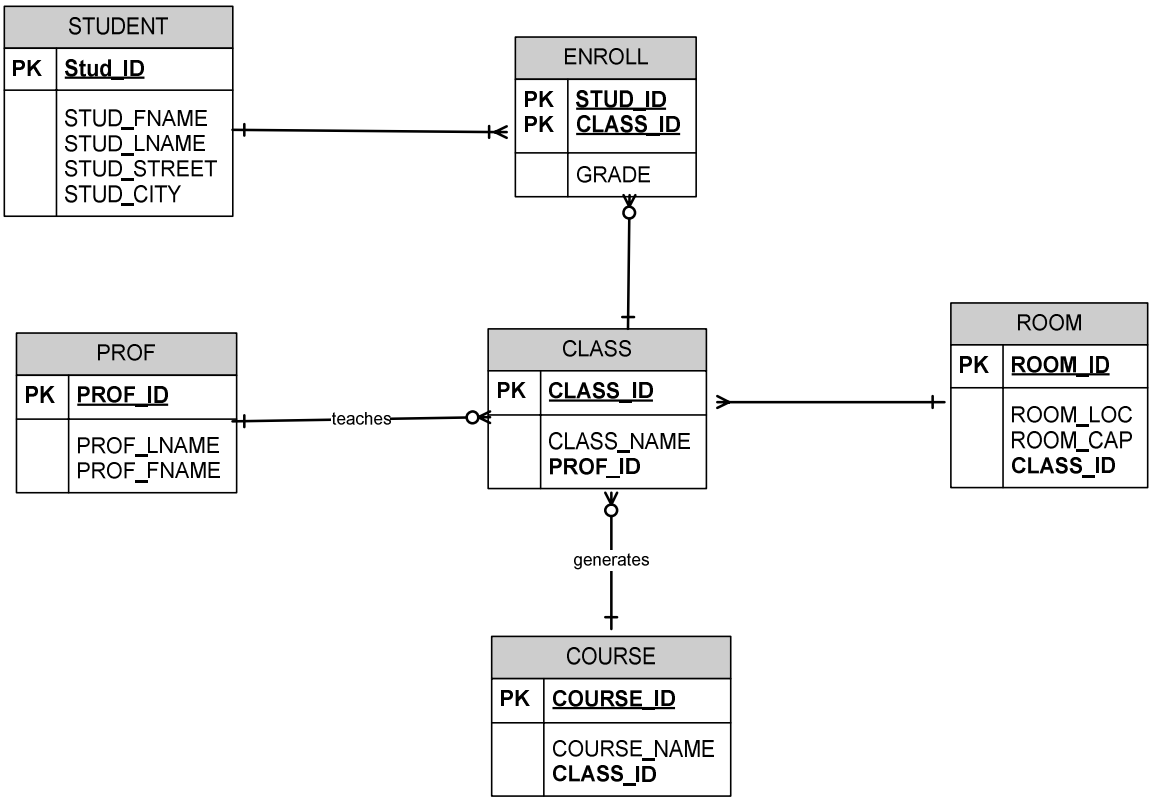
The original notation for ER-Diagrams uses rectangles to represent entities, and diamonds to represent relationships. Alternative notations include:

Crow's Foot

There are three basic elements in ER-Diagrams:

1. Entities are the "things" for which we want to store information. An entity is a person, place, thing or event.
2. Attributes are the data we want to collect for an entity.
3. Relationships describe the relations between the entities.

ERDs show entities in a database and relationships between tables within that database. It is essential to have ER-Diagrams if you want to create a good database design. The diagrams help focus on how the database actually works.



Wildcard characters

Use this set of wildcard characters when you use the Find and Replace dialog box to find and optionally replace data in an Access database or an Access project. You also use these characters when you run select and update queries against an Access database, but you do not use them in queries run against an Access project. Character Description Example

* Matches any number of characters. You can use the asterisk anywhere in a character string. wh* finds what, white, and why, but not awhile or watch.

? Matches any single alphabetic character. B?ll finds ball, bell, and bill

[ ] Matches any single character within the brackets. B[ae]ll finds ball and bell but not bill

! Matches any character not in the brackets. b[!ae]ll finds bill and bull but not ball or bell

\- Matches any one of a range of characters. You must specify the range in ascending order (A to Z, not Z to A). b[a-c]d finds bad, bbd, and bcd

Data Models

Data model is a way of storing and retrieving the data. There are three different data models. Data models differ in the way they allow users to view and manipulate relationships between entities. Each has its own way of storing the data. The following are the three different data models:

_ Hierarchical
_ Network
_ Relational

Hierarchical

In this model, data is stored in the form of a tree. The data is represented by parentchild relation ship. Each tree contains a single root record and one or more subordinate records. For example, each batch is root and students of the batch will be
subordinates. This model supports only one-to-many relationship between entities.

This was used in IBM's Information management system, IMS.

Network

Data is stored along with pointers, which specify the relationship between entities. This was used in Honeywell's Integrated Data Store, IDS. This model is complex. It is difficult to understand both the way data is stored and the way data is manipulated. It is capable of supporting many-to-many relationship
between entities, which hierarchical model doesn't.

Relational

This stores data in the form of a table. Table is a collection of rows and columns. RDBMS has been discussed in detail above.

Normalization

Normalization is a method for organizing data elements in a database into tables.

Normalization Avoids

- Duplication of Data – The same data is listed in multiple lines of the database
- Insert Anomaly – A record about an entity cannot be inserted into the table without first inserting information about another entity – Cannot enter a customer without a sales order
- Delete Anomaly – A record cannot be deleted without deleting a record about a related entity. Cannot delete a sales order without deleting all of the customer's information.
- Update Anomaly – Cannot update information without changing information in many places. To update customer information, it must be updated for each sales order the customer has placed

Normalization is a three stage process – After the first stage, the data is said to be in first normal form, after the second, it is in second normal form, after the third, it is in third normal form

Before Normalization

1. Begin with a list of all of the fields that must appear in the database. Think of this as one big table.
2. Do not include computed fields
3. One place to begin getting this information is from a printed document used by the system.
4. Additional attributes besides those for the entities described on the document can be added to the database.

Before Normalization – Example

See Sales Order from below:

Sales Order

Fiction Company
202 N. Main
Mahattan, KS 66502

CustomerNumber:	*1001*	**Sales Order Number:**	405
Customer Name:	*ABC Company*	**Sales Order Date:**	2/1/2000
Customer Address:	*100 Points*	**Clerk Number:**	*210*
	Manhattan, KS 66502	**Clerk Name:**	*Martin Lawrence*

Item Ordered	Description	Quantity	Unit Price	Total
800	*widgit small*	*40*	*60.00*	*2,400.00*
801	*tingimajigger*	*20*	*20.00*	*400.00*
805	*thingibob*	*10*	*100.00*	*1,000.00*
	Order Total			*3,800.00*

Fields in the original data table will be as follows:

SalesOrderNo, Date, CustomerNo, CustomerName, CustomerAdd, ClerkNo, ClerkName, ItemNo, Description, Qty, UnitPrice

Think of this as the baseline – one large table

Normalization: First Normal Form

- Separate Repeating Groups into New Tables.
- ***Repeating Groups*** Fields that may be repeated several times for one document/entity
- Create a new table containing the repeating data
- The primary key of the new table (repeating group) is always a composite key; Usually document number and a field uniquely describing the repeating line, like an item number.

First Normal Form Example

The new table is as follows:

SalesOrderNo, ItemNo, Description, Qty, UnitPrice

The repeating fields will be removed from the original data table, leaving the following.

SalesOrderNo, Date, CustomerNo, CustomerName, CustomerAdd, ClerkNo, ClerkName

These two tables are a database in first normal form

What if we did not Normalize the Database to First Normal Form?

Repetition of Data – SO Header data repeated for every line in sales order.

Normalization: Second Normal Form

- Remove Partial Dependencies.
- ***Functional Dependency*** The value of one attribute in a table is determined entirely by the value of another.
- ***Partial Dependency*** A type of functional dependency where an attribute is functionally dependent on only part of the primary key (primary key must be a composite key).
- Create separate table with the functionally dependent data and the part of the key on which it depends. Tables created at this step will usually contain descriptions of resources.

Second Normal Form Example

The new table will contain the following fields:

ItemNo, Description

All of these fields except the primary key will be removed from the original table. The primary key will be left in the original table to allow linking of data:

SalesOrderNo, ItemNo, Qty, UnitPrice

Never treat price as dependent on item. Price may be different for different sales orders (discounts, special customers, etc.)

Along with the unchanged table below, these tables make up a database in second normal form:

SalesOrderNo, Date, CustomerNo, CustomerName, CustomerAdd, ClerkNo, ClerkName

What if we did not Normalize the Database to Second Normal Form?

- Repetition of Data – Description would appear every time we had an order for the item
- Delete Anomalies – All information about inventory items is stored in the SalesOrderDetail table. Delete a sales order, delete the item.
- Insert Anomalies – To insert an inventory item, must insert sales order.
- Update Anomalies – To change the description, must change it on every SO.

Normalization: Third Normal Form

- Remove transitive dependencies.
- ***Transitive Dependency*** A type of functional dependency where an attribute is functionally dependent on an attribute other than the primary key. Thus its value is only indirectly determined by the primary key.
- Create a separate table containing the attribute and the fields that are functionally dependent on it. Tables created at this step will usually contain descriptions of either resources or agents. Keep a copy of the key attribute in the original file.

Third Normal Form Example

The new tables would be:

CustomerNo, CustomerName, CustomerAdd

ClerkNo, ClerkName

All of these fields except the primary key will be removed from the original table. The primary key will be left in the original table to allow linking of data as follows:

SalesOrderNo, Date, CustomerNo, ClerkNo

Together with the unchanged tables below, these tables make up the database in third normal form.

ItemNo, Description

SalesOrderNo, ItemNo, Qty, UnitPrice

What if we did not Normalize the Database to Third Normal Form?

- Repetition of Data – Detail for Cust/Clerk would appear on every SO

- Delete Anomalies – Delete a sales order, delete the customer/clerk
- Insert Anomalies – To insert a customer/clerk, must insert sales order.
- Update Anomalies – To change the name/address, etc, must change it on every SO.

Completed Tables in Third Normal Form

Customers: CustomerNo, CustomerName, CustomerAdd

Clerks: ClerkNo, ClerkName

Inventory Items: *ItemNo*, Description

Sales Orders: SalesOrderNo, Date, CustomerNo, ClerkNo

SalesOrderDetail: SalesOrderNo, ItemNo, Qty, UnitPrice

Access 2007 Environment

Control Buttons:

Located in the upper-right corner of your window, these buttons (in order) will minimize, maximize/restore and close your window.

Object Window:

The Object window displays the opened objects as a series of tabbed documents so you can quickly switch between multiple, open Access objects.

Office Menu:

A button that displays a menu of common functions such as open and print.

Page View Controls:

Located in the lower-right corner of your window, the Page View Controls include quick access to page views include a zoom slider control.

Quick Access Toolbar:

Contains buttons to your most frequently used tasks such as Save, Undo and Print.

Ribbon:

Displays commands and features that were previously hidden under menus in easy-to-access task-oriented tabs such as Insert, Page Layout and Formulas.

Scroll Bars:

Vertical and horizontal bars allowing you the view other areas of the worksheet.

Status Bar:

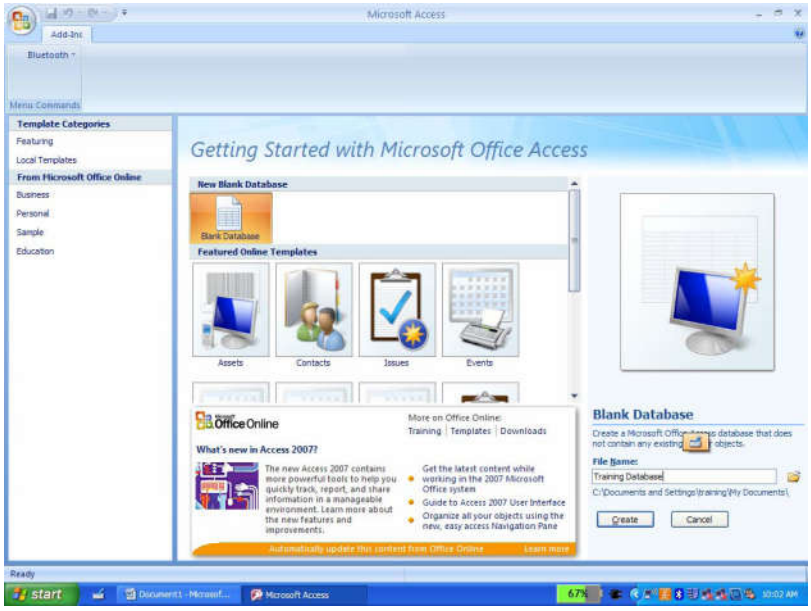
In the Status Bar, you see a document page and word count, plus macro and editing indicators.

Title Bar:

Located at the very top of your screen, the Title Bar displays the name of the program you are running and the name of the document you are working in.

Navigation Pane:

The Navigation Pane replaces the old Database Window and can be used to quickly navigate between the different Access objects.



Exploring the Ribbon

The Access 2007 Ribbon contains four tabs across the top of the Access window. Each tab contains groups of core tasks and each Group contains related sets of commands.

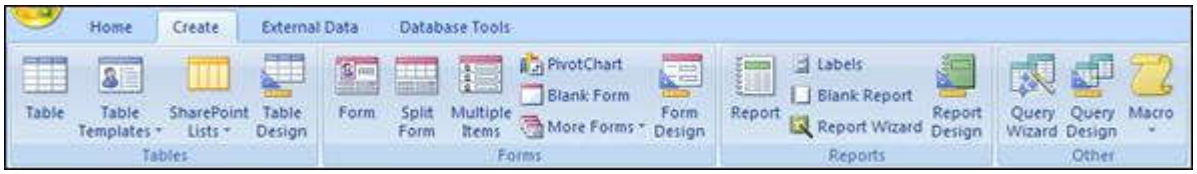
Many of the Ribbon groups contain a Dialog Box Launcher arrow ().This arrow is always found in the lower-right corner of a group and typically opens a Dialog Box with familiar Access settings. Occasionally, this button options a Task Pane instead of a Dialog Box.

Home Tab

The Home tab contains your most often-used commands including cut, copy and paste. You'll also see groups for Views, Font, Rich Text, Records, Sort & Filter and Find.

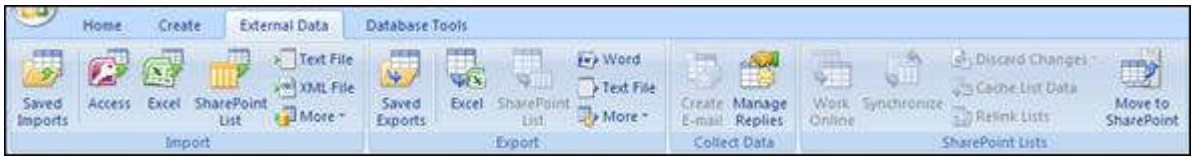
Create Tab

The Create tab contains commands related to creating new database objects including tables, forms, reports and queries. The Create tab contains groups for Tables, Forms, Reports and Other.



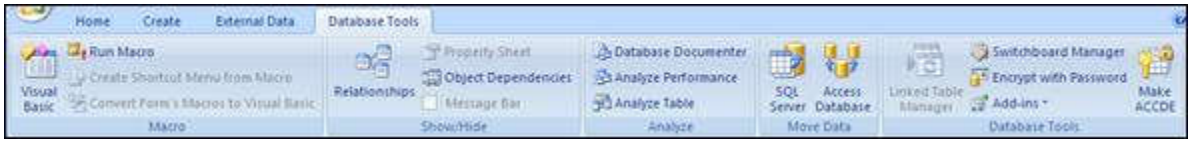
External Data Tab

The External Data tab contains commands related to importing and exporting data as well as working with SharePoint lists. This tab contains groups for Import, Export, Collect Data and SharePoint Lists.



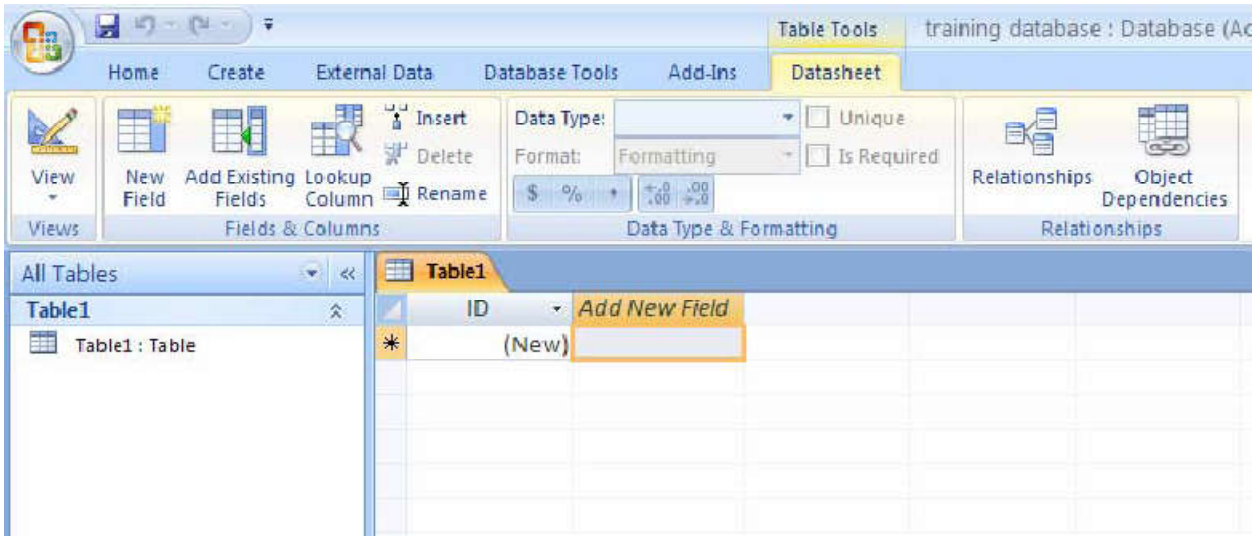
Database Tools Tab

The Database Tools tab contains commands related to advanced database commands such as macros and passwords. This tab contains groups for Macro, Show/Hide, Analyze, Move Data and Database Tools.



1) Select **Blank Database**
2) In the File Name field enter a name for the database
3) Click **Create**

Microsoft Access automatically creates a new table in the database called **Table1**. This is a temporary name until the table is saved.

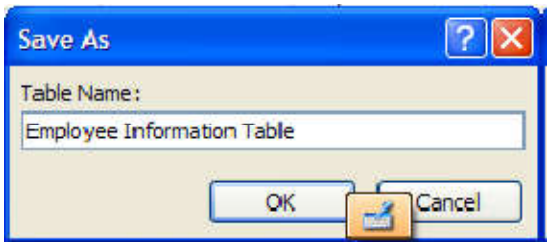


Understanding the Views

There are 2 basic views when you work in a table: **Design View** and **Datasheet View.** Design View is used to set the data types, insert or delete fields, and set the Primary key. Datasheet View is used to enter the data for the records. By default, Access places you in Datasheet View.

To Switch to Design view:

1) Click the **View** button on the Home Ribbon
2) Type a name for the table
3) Click **OK**

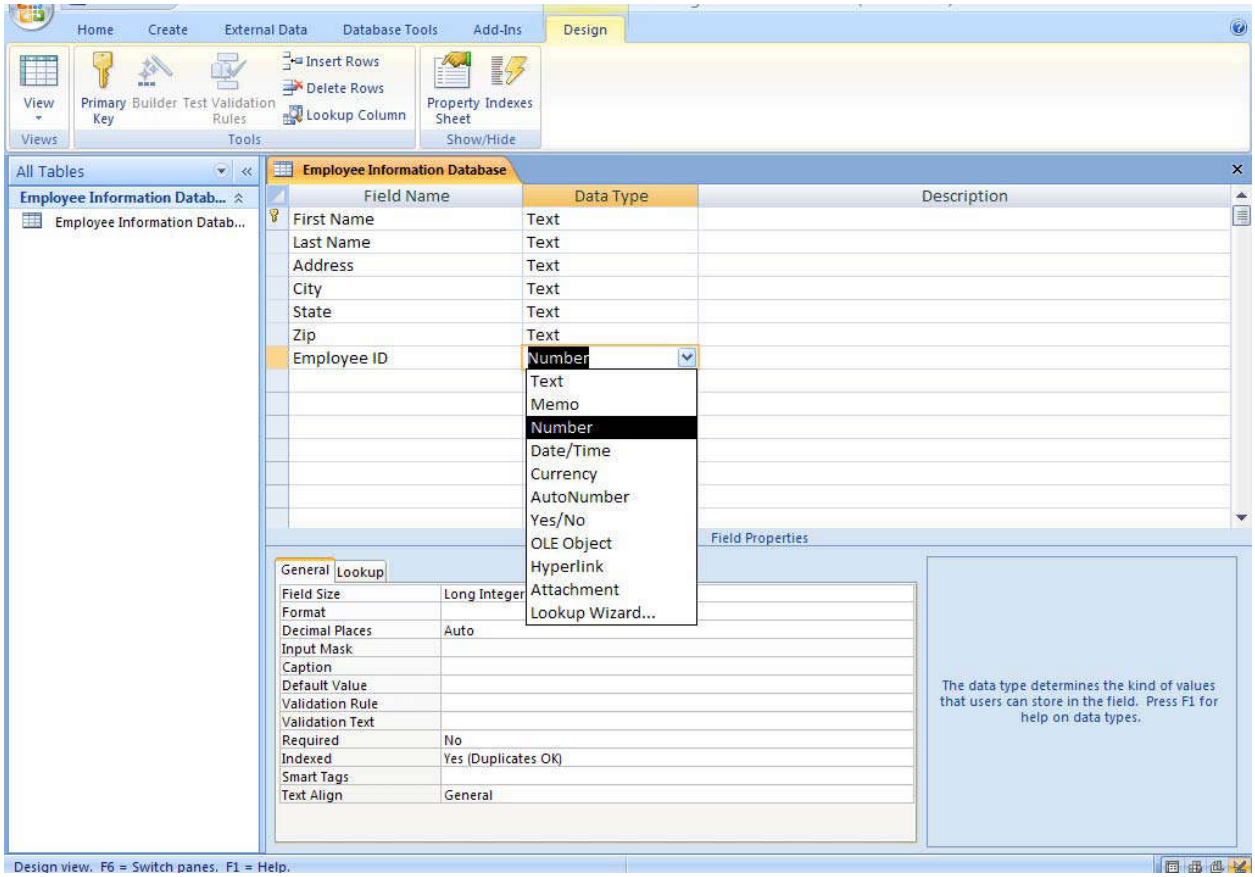


Before proceeding, it is important to understand common Microsoft Access Data Types. (Explained in the table below)

Data Types

Data Type	Description	Size
Text (most common data type)	Alphanumeric data	Up to 255 characters
Memo	Alphanumeric data; sentences and paragraphs	Up to 64,000 characters
Number	Numeric data	1,2,4, or 8 bytes.
Date/ Time	Dates and times	8 bytes
Currency	Currency data, stored with 4 decimal places	8 bytes
Auto-Number	Unique value generated by Access for each new record	4 bytes
Yes/No	Boolean (true/false) data	1 bit
OLE Object	Pictures, graphs, or other ActiveX objects from another Windows-based application	Up to about 1 gigabyte
Hyperlink	A link "address" to a document or file on the Web, local network, or on your computer	Up to 2048 characters

To Enter Fields in a Table:
1) Type a name for the first field in the table
2) Press **Enter**
3) Select a data type
4) Press **Enter**
5) Type a description for the field
6) Press **Enter**
Continue this until all necessary fields have been entered into the table.
Note: The order that you enter the field names is the order the fields will Appear in the table and on a form.



To View the Datasheet:

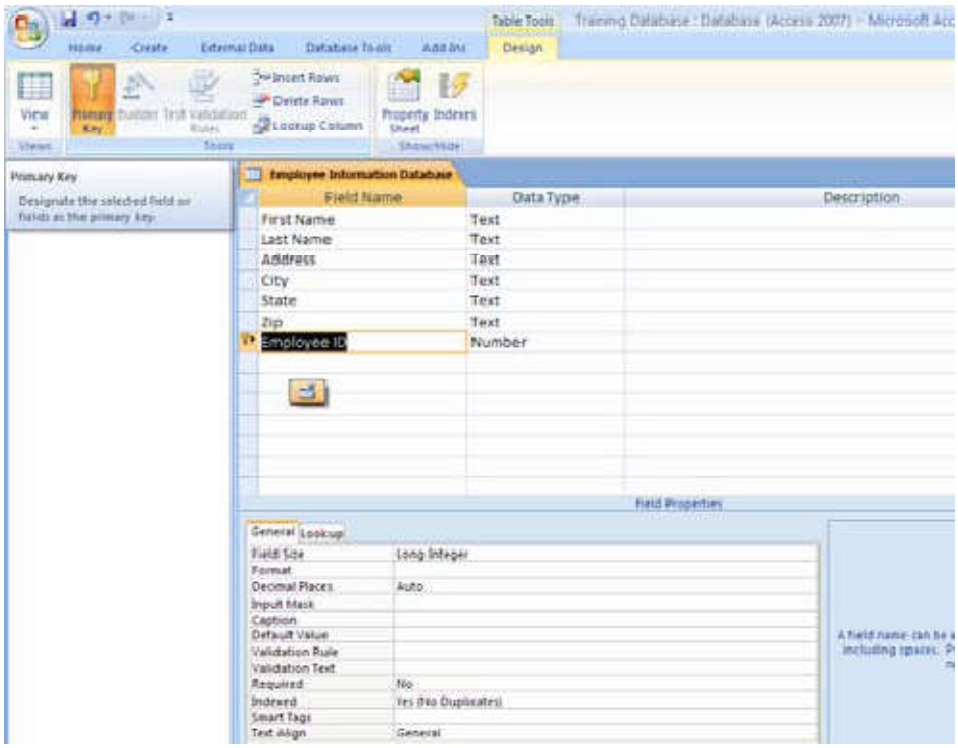
Click the **View** button on the Ribbon

Setting a Primary Key

The **Primary Key** is the unique identifier for each record in a table. Access will not allow duplicate entries in a Primary Key field. By default, Access sets the first field in the table as the Primary Key field. An example of a Primary Key would be your Social Security Number. This is something unique about you and should not be duplicated.

To Set a Primary Key:

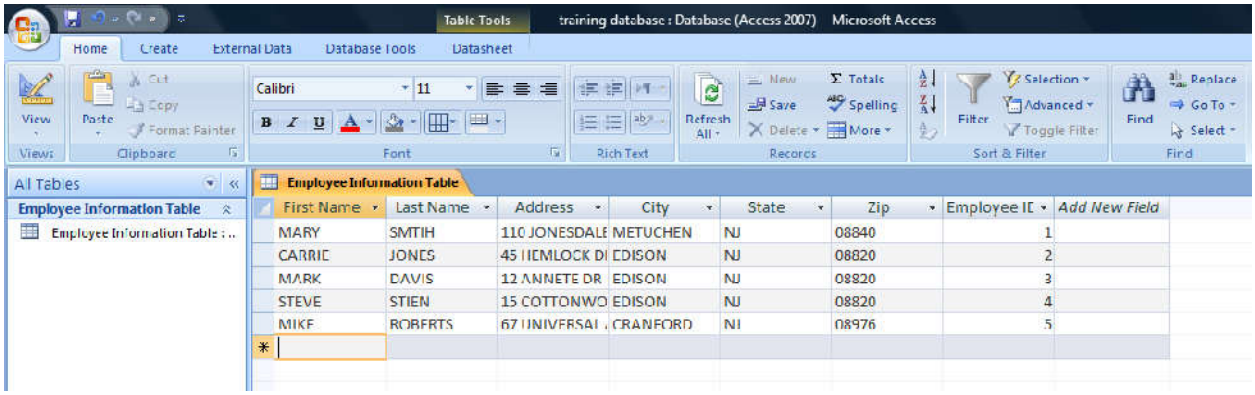
1) Switch to **Design View**
2) Position your cursor in the field you wish to set as the Primary Key
3) Click the **Primary Key** button on the Ribbon



To Switch Back to Datasheet View to Enter your Records:
Click the **View** button on the Ribbon.
Entering Data in a Table
Once you have entered the fields and set the data types it is now time to enter the records in a table.

To Enter Data in a Table:

1) Make sure you are in **Datasheet View**
2) Enter the data into the table by pressing the tab key to move from one cell to another
4) When you have completed the record (row), press **Enter**



When inputting data into the table, Access automatically saves the data after

each new record.

Input Masks

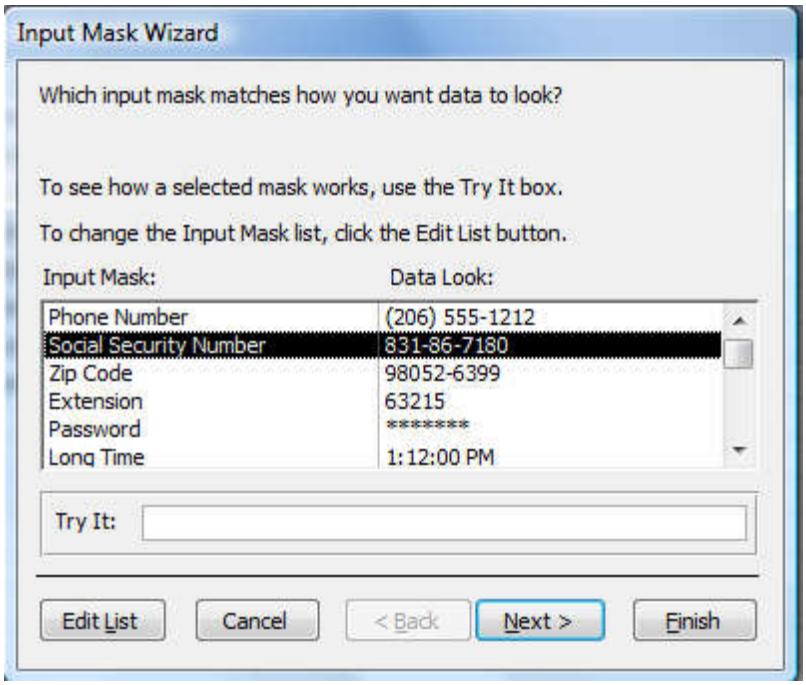
An **Input Mask** is used to pre-format a field to "look/act" a certain way when a user inputs data.

Example: You could create an input mask for a Social Security Number field that automatically inserts the dash.

The Input Mask data can either be stored in the table or simply displayed and not stored. (The latter is preferred)

To Create an Input Mask for a Field

1) Open a table in Design View
2) Click in a field for which you'd like to create an input mask
3) In the Field Properties section at the bottom of the screen, click in the **Input Mask** line and notice the **Build** button that appears at the right end of the line (see below)

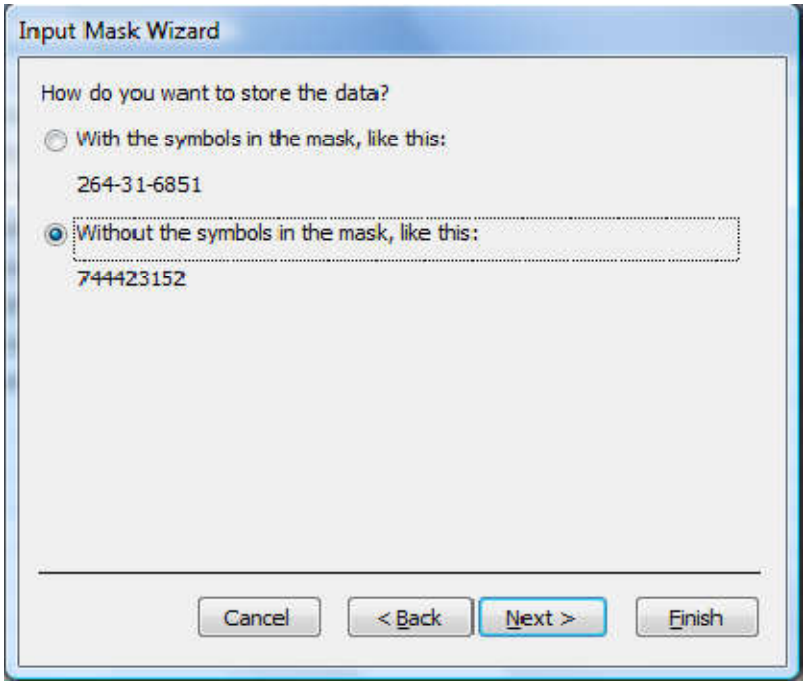


4) Click the **Build** button
5) Select **Input Mask**
6) Click **Next**
6
7) Select a **Placeholder character**
8) Click **Next**

9) Select **Without the symbols in the mask**

10) Click **Next**

11) Click **Finish**

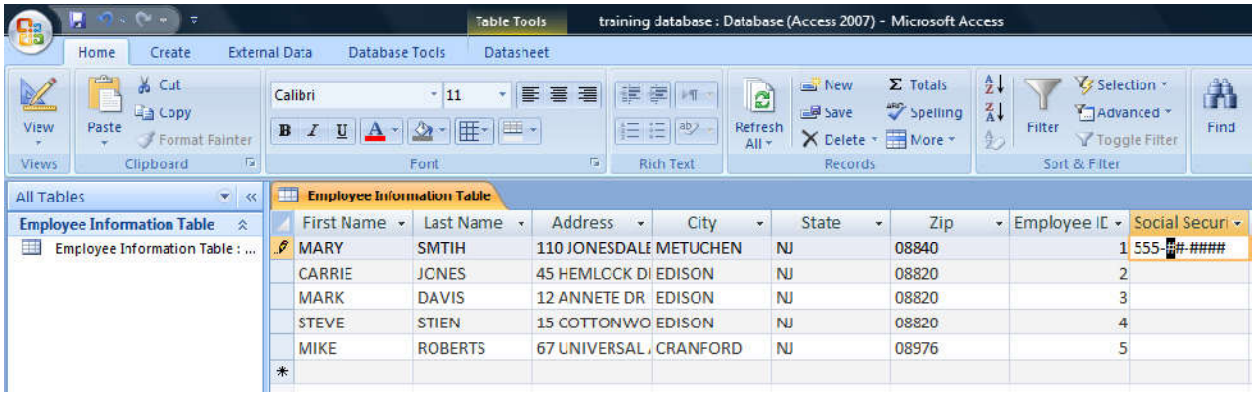


Now, when entering data that has been formatted with an Input Mask, you do not have to type the format into the record.

7

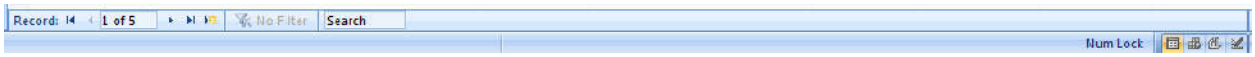
The following is an example of a table with a field that has been formatted with an Input Mask on the Social Security Field.

Notice, the only thing that the user has to enter is the digits, not the symbols.



Navigating Records

Use the arrows at the bottom of the table to navigate among records.
You are able to navigate from the first record, previous record, next record, last record, and create a new record (as shown in the picture below).
Notice that the total number of records in the table is shown at the right end of the navigation arrows.

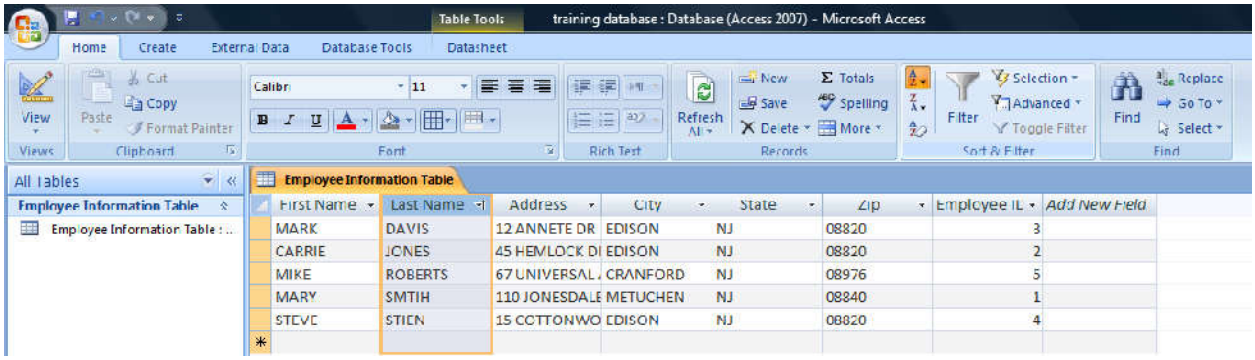


Sorting Records in a Table

By sorting your records in a table, you are easily able to view/locate records in your table.

To Sort Records in a Table:

1) Position your cursor in the field that you wish to sort by, by clicking on any record in that field.
5) Click either the **Sort Ascending** or **Sort Descending** icon



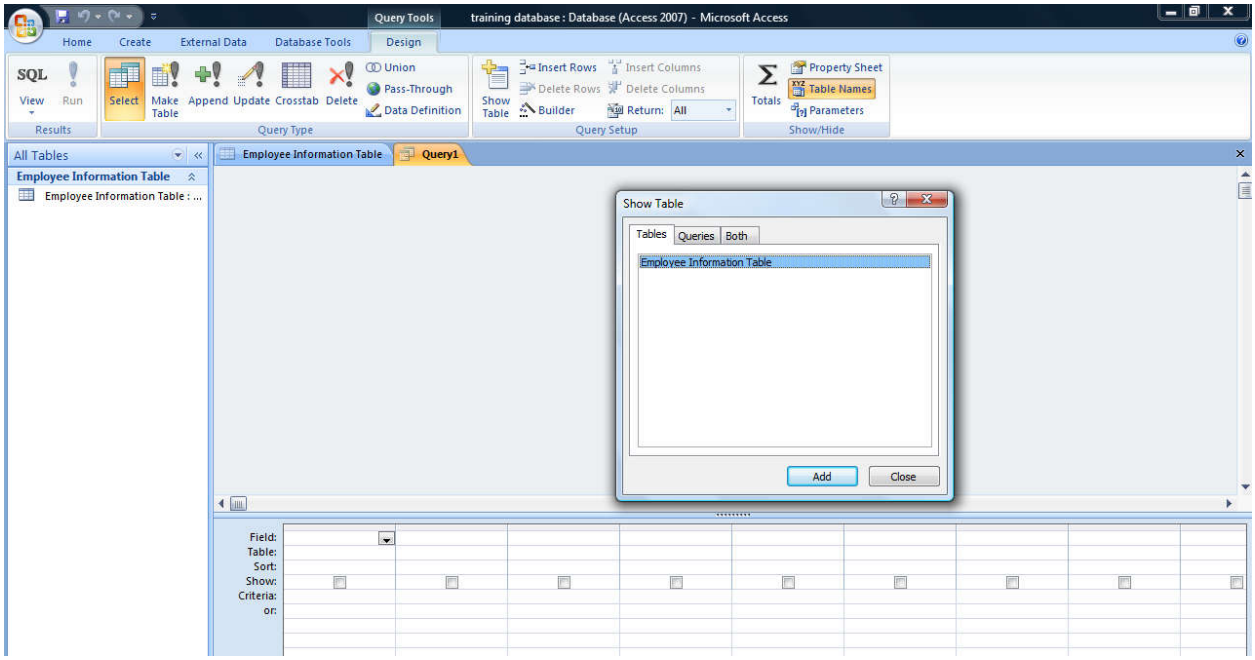
Notice, the table above has been sorted by the Last Name field in ascending order.

Queries

You use Queries to view, change, and analyze data in different ways. You can also use them as a source of records for forms and reports.

To Create a Query:

1) Click the **Create** tab on the Ribbon
2) Click **Query Design** icon



3) Double-click **Create Query** in Design View
4) Select the table that you would like to base your Query on
5) Click **Add**
6) Close the **Show Table** window

The table(s) will now be displayed in the upper part of the Query Design Screen by boxes containing the tables' fields.

7) Double click on the field names in the field list window which you would like to include in the Query

Defining Criteria in the Query

In order to control which records are displayed, you must define criteria in a Query. The most common type of Query is the **Select Records Query** which will be discussed below.

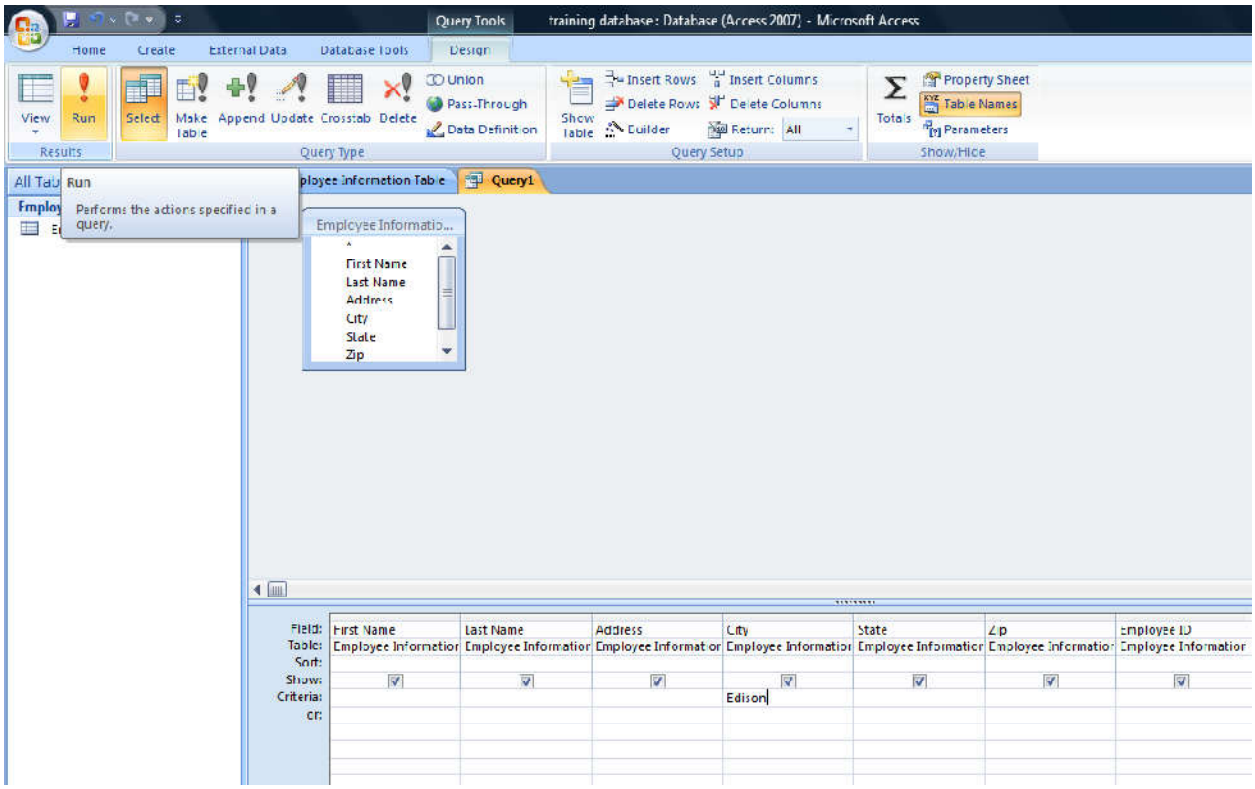
To Define Criteria for your Query:

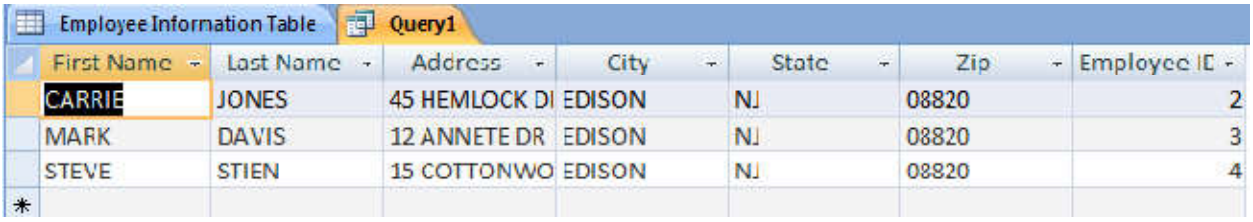
1) Position your cursor in the criteria row in the field for which you wish to define the criteria for
2) Enter the criteria

Example: To find all people it the table who live in Edison:

- Position your cursor in the **criteria row** of the City field
- Type **Edison**
- Click the **Run Query** button

Below is a picture of the results of the above query:

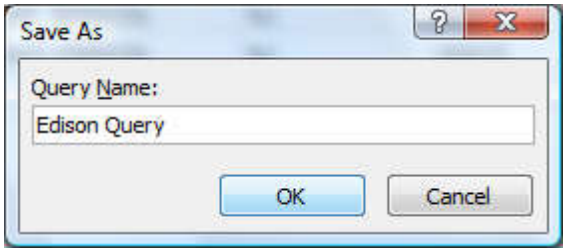




The result of a query is called a **recordset**. A recordset can be sorted, printed or Filtered in the same manner as a table.

To Save the Query:

1) Click the **Save** Icon
2) Enter a name for the Query
3) Click **OK**



Note: When saving a select Query, you are saving the question that you are asking, not the results that you see when you run the Query.

Creating a Form Using the Forms Wizard

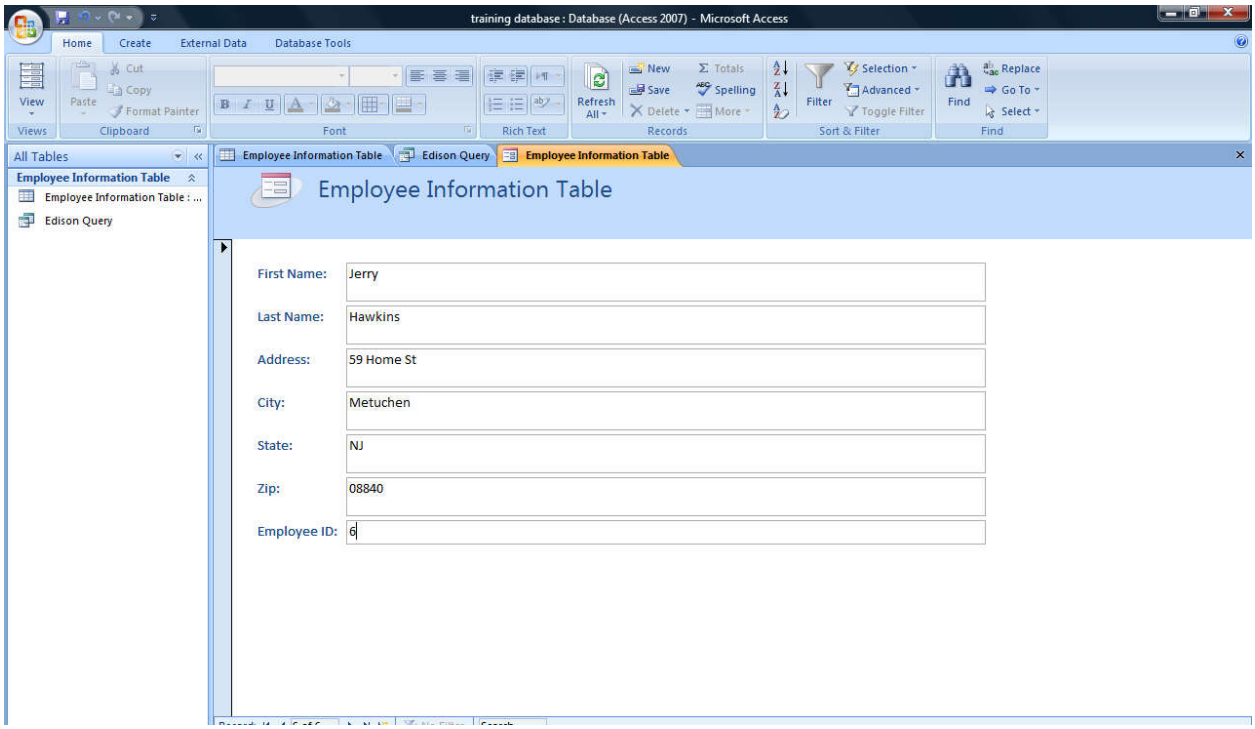
A form is a database object that is used to enter or display data in a database.

To Create a Form Using the Wizard:

1) Navigate to the table you want to base the form on
2) Click **Create** on the Ribbon
3) Click **Forms**

You are able to navigate using the navigation arrows at the bottom of the form.

Note: The form feeds the table. If you edit a record on the form, or create a New record that data will be passed to the table it is associated with.



To Enter a Record on the Form:

1) Click the **View** button on the Ribbon to switch from Layout View to Form View
2) Enter the data for each field in the record, pressing the **Enter** key to move to the next field

3) Press **Enter** after you have entered data for the last field

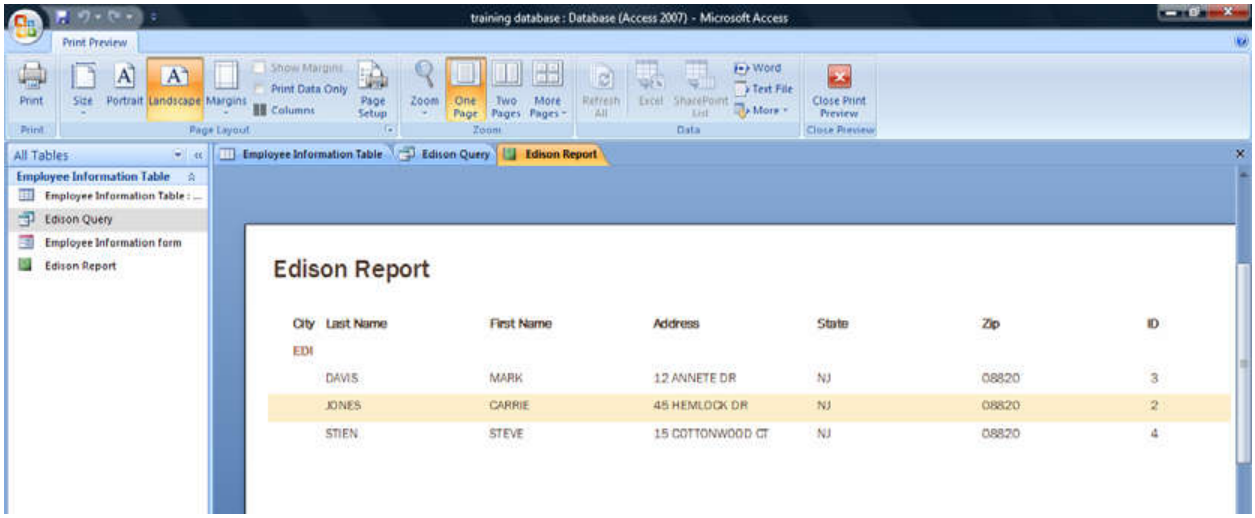
This will send the record to the table.

Reports

Reports can be based on tables or queries and can be made with the Report Wizard.

To Create a Report Using the Report Wizard:

1) Click the **Create** tab on the Ribbon
2) Click the **Report Wizard** icon
3) Select the table or query upon which the report will be based
4) Select the fields that you want to include on the report by double clicking on them
5) Click **Next**
6) If you would like to add grouping to your report, select the field you wish to group by double clicking on it (*Example: City*)
7) Click **Next**
8) Select a style for the report
9) Click **Next**
10) Type a title for the report
11) Click **Finish**



To Print a Report

1) Open the report by double clicking on the object in the **Navigation Pane**
2) By default, the report opens in Print Preview.

To Adjust the Orientation:

Click the portrait or landscape icon on the Print Preview Ribbon

To Adjust the Margins

1) Click them **Margins** icon on the Print Preview Ribbon
2) Select a margin size

To Print the Report

1) Click the **Print** Icon on the Print Preview Ribbon
2) Select the **Printer**
3) Click **OK**

Resize Columns

You can resize your Access columns by dragging. Dragging is quicker method of resize your column size

Freeze and unfreeze columns in a datasheet

You can freeze one or more of the columns (fields) on a datasheet so that they become the leftmost columns and

are visible at all times no matter where you scroll.

Open a table, query, form, view, or stored procedure in Datasheet view.

Select the columns you want to freeze.

To select one column, click the field selector for that column.

To select more than one column, click the column field selector and then, without releasing the mouse button, drag to extend the selection.

Do one of the following:

To freeze the selected columns, click Freeze Columns on the Format menu.

To unfreeze all columns, click Unfreeze All Columns on the Format menu.

Show or hide columns in a datasheet

Open a table, query, form, view, or stored procedure in Datasheet view.

Do one of the following:

Hide a column or columns

1. Select the columns you want to hide.
2. To select one column, click the field selector for that column.

3. To select adjacent columns, click a column field selector and then, without releasing the mouse button, drag to extend the selection.
4. On the Format menu, click Hide Columns.

Show a hidden column or columns

1. On the Format menu, click Unhide Columns.
2. In the Unhide Columns dialog box, select the names of the columns that you want to show.

Types of query in ms access

1. Create Query in design view
2. Create Query by wizard

How to Create All Types of Relationships in One Place in Microsoft Access

You can join data sources together with relationships.

Creating relationships in your Microsoft Access database prevents duplicate information, limits confusion and speeds up processes. Access contains several different relationship types and each depends on the kind of information listed in your sources. Two common relationship types include one-to-one and one-to-many. Access automatically defines relationships when you create a lookup field or combine tables in a query. You can make changes to these relationships, create your own or review the entire scheme in one unified location.

1. Click the "Database Tools" tab on the ribbon at the top of the page and select "Relationships." This opens the relationships window, which allows you to view, add and remove relationships between fields. Because forms and reports don't hold any records themselves, you won't see them listed here.
2. Click the "Show Table" button if you don't see a certain table or query to place it on the relationships grid. To create a relationship between two sources, drag the field from the first table to the related field of the second table. Then click on the "Edit Relationships" button on the ribbon to change the type of relationship.
3. Click the "Join Type" button to define the type of relationship between the two sources. Choose the first option to display a one-to-one join with the most restrictions. You will only be able to connect a single field in the first table to a single field in the second table.
4. Select the second option to define a one-to-many type of join. Your first table can use as many of the fields from the second table as you want, but the second table will only have one related record. Select the third join option to choose the reverse table assignment.