

Database Implementation with MS Access – Lab

Introduction and Problem Description

The information below is the list of vendors (suppliers) of a fast food restaurant. The restaurant buys major items from these vendors, and has a database to facilitate internal record keeping. We will design an MS Access database that stores the information and answers "certain questions."

ID	Name	Contact person	Street address	city	post code	Phone	balance	date due
9621	Bakery A.S.		cayir cad. No 12 Istinye	Istanbul	80860	212-229-60-82	\$102.00	
4532	Bardak A.S.	Ali Kara	Organize sanayi bolgesi	Manisa	45070	242-321-12-67	(\$800.00)	29-Oct2006
2345	Pinar A.S.	Zeynep Sen	Talatpasa bulv. Alsancak	Izmir	35960	232-222-56-93	(\$500.00)	2-Nov 2006
8991	Sebze A.S.	Ayşe Yesil	Muhendis sk. Cengelkoy	Istanbul	80450	216-356-23-12	\$40.00	

General Concepts of MS Access

NOTE: Both MS Access and MS Access 2007 are in use, and are available in different labs. The user interface is slightly different, although the main concepts stay the same.

Objects.

Within a MS Access database, several objects can be created. They will be shown in a list on the left hand side (in MS Access 2007, the user can apply some filters, so not all objects might be visible at any moment). From this list, one or more objects can be opened for editing and viewing. The main objects are:

- Tables (this includes the data stored in the tables)
- Queries
- Forms
- Reports

Views.

For most objects, there are several views, which constitute possibilities of viewing and editing. In the top left corner, there is an icon to switch between views. The most important views are:

- Design View (used for editing objects in a GUI)
- Datasheet View (used for viewing the result, or entering data in tables)
- SQL View (displays the SQL equivalent to the design view)

MS Access Files.

All objects and data are saved to a single file. This file contains everything. The ending is .mdb (for MS Access 2003) or .accdb (for MS Access 2007).

Lesson 1: Setting Up A New Database

Starting Access.

- When you click on Access (from Start>Programs>MS Access or similar), unlike other programs you are familiar with, you will be given options to choose from. For this lab, we will build a new database using a **blank database**.
- Again unlike most application programs, Access asks you to immediately give a file name and save your database-to-be. Select a folder you can write to (preferably the desktop) and give a filename (e.g. firstlab.mdb). After you click OK, you will be in the *database view*.

Designing the table.

We have to design the data table first so that we can store the data in. We have to create our "datasheets" since Access does not give us such sheets automatically. To proceed, click on "Create table in Design view". In MS Access 2007, there is a "Create" tab which allows access to all creating all types of objects.

You are now in design view. Of course before coming to this point, one has to know what fields should be included in a table. For this exercise, that question is already answered, otherwise this information comes from the logical design of the database (ERD).

- To design a table, one has to decide on **field names** and **data types**. One could also add a description that explains what the field stands for. In addition, there are a number of **field properties** that can be specified for the fields in a table.
- In the table below, you see field names and data types. First, ignore the field properties and design your table by typing in field names and choosing the data types.
- When you complete entering the entire list, notice the following:
Step on one row that contains a field. On the lower half of the screen, you will see **field properties**. Field properties can be customized. To keep this simple, in this exercise we will **only change the field sizes for the text fields**.

Field name	Data Type	Field properties to be modified	
VendorID	Text	Change Size to 4, add caption " <i>ID</i> "	ID
Name	Text	Change Size to 20	Name
C_person	Text	Change Size to 30, add caption " <i>Contact Person</i> "	Contact person
Address	Text	Change Size to 60, add caption " <i>Street Address</i> "	Street address
City	Text	Change Size to 20	City
Pcode	Text	Change Size to 5, add caption " <i>Postcode</i> "	Postcode
Phone	Text	Change Size to 10, input mask: <i>000\-000\-00\-00</i>	Phone
Balance	Currency	Format <i>Currency</i>	Balance
Due	Date/time	Add caption " <i>Datedue</i> ", format <i>Medium date</i>	Datedue

Our table is almost ready, but not yet.

Setting the PRIMARY KEY.

Access needs a primary key for every table. The primary key is a field that identifies a record uniquely. That is, no two records can have the same value in a primary key. For instance, your student ID numbers identify you uniquely. A bank account number identifies that account uniquely. For this table the primary key should be VendorID. Highlight the field vendor ID in design view, go to Edit from the menu and check primary key (OR right click on the field and select the Primary Key option on the top). You will see a small key appearing next to it. Now the structure of our table is complete.

Saving the table structure.

If you click on **save** button, Access will ask for a table name. Call this table "**vendors**". If you try to save a table before setting the primary key, Access will recommend creating one for you. This might sometimes complicate your design. Therefore it is usually recommended to set the primary key before saving a table structure.

Entering data.

Our table is still blank. To enter data, click on **datasheet view** button (the left-most on the toolbar). You can now enter the data just as you would enter it into an Excel sheet. Enter at least 5 different vendors, at least two of which should be from Istanbul.

Warning: The balance column above shows the account balance with that particular vendor. Values in parentheses are negative numbers, which are vendors owed by the store, and the due dates are listed in the next column. To enter these values, enter -800, -500, etc. The formatting should be automatic.

Lesson 2: Designing A Query

Start.

Access has some features that make it superior to Excel for certain problems. An integral part of a database is the query: a "question" that you direct to the database. There are different types of queries: the ones that just answer questions (e.g. select), the ones that perform certain operations while answering questions.

In order to start designing queries you should open the query window from the database menu (or the "Create" tab). Select "Create query in design view."

Now, you should select the table you will create queries on, so choose the table: vendor and close the "Show table" window.

Running a query and views.

The query window in design view is open now. You can switch windows to see the results of your query by clicking on the *view* button on the upper left corner just like in the tables, or using the RUN icon which runs (starts) the query. You can also switch to the SQL view.

Specifying the query.

The first step is to select which fields (attributes) are to be used in the query. In the GUI, one is used in every row. In order to select the fields you are interested in, you can simply drag the items from table to the field row. In order to select all fields you can just click on *. Alternatively, you can select them from the menu in the field cell.

Any of the selected fields can be used for output or for specifying the query itself (or both). Whether it will be part of the output is selected by the "Show" checkbox in each column. To be used in the query as a criteria, the criteria row is to be filled for the attribute.

Query 1: Select the vendors in İstanbul.

- In the database view, first click on **query**, then **new** on the right, and choose **design view**. **Add** the table vendors we have just created and then **close** the **show table** dialog box. We are now in the query design view.
- In the first column, choose **name**. In the second column, choose **city**.
- When you click on datasheet view in the left-most corner (or run the query), you will see our table, but only with two fields. Still, this shows all vendors, this is not what we want.
- In order to select the vendors in İstanbul, go back to design view by clicking on the left-most button. Now, under the city column, in the criteria row you should write İstanbul and go back to datasheet view again.

Note: If you are not sure about the record you are calling, you can use *.

Query 2: Show the vendors with negative balances.

In order to see the vendors with negative balances you should write <0 to the criteria row.

Query 3: Show the vendors from Istanbul with negative balances.

Lesson 3: Queries Using Grouping and Aggregation

An important feature is grouping (group by in SQL). This means that all records are categorized according to the attribute(s) chosen as grouping variables. This means that there are as many resulting groups as there are different values for this attribute(s) in the data. Other attributes included can then be reported in an aggregated manner for each group (including sum, average etc., but also count, which gives the number of distinct values for the attribute within the group. In order to access this functionality, use the Totals button (sum sign). This opens a new row for every fields, which can be set either to group by, or some aggregation.

Query 4: Calculate the sum of balances according to cities.

Lesson 4: Multiple Tables - Establishing Relationships

Refer to the Movie House database from the website. The database is a prototype that is designed to keep track of the records of Movie House (MH), a store that rents movies, and consists of 4 tables: movies, inventory (tapes), transactions and members.

Important! First, copy the file moviehouse.mdb from the above location into your work area . Then open your copy of the file and work on it. Rename this file as *yourusername.mdb*.

Primary keys and relationships.

While a relational model has foreign keys, this has not been specified in the database as yet. Any foreign key has as yet been a simple field. Therefore these relationships have to be established to be able to have queries that span several tables or to enforce referential integrity (like cascade on delete). In the GUI, the tab Database Tools has an icon for relationships.

Establish relationships.

In this database, the primary keys are already installed. To establish relationships, choose relationships from the tools menu. You can establish relationships between two fields by simply dragging the “from” field towards the “to” field. For example, to establish a relationship between the movieID field in the movies table and movieID field in the inventory table, drag the movieID field in the movies table towards the movieID field in the inventory table.

A window then opens where additional data on the relationship (which will be one-to-many) can be specified, e.g. If referential integrity is to be enforced, and with which policy.

Build all relationships as seen on the attached sheet for the database.

Members Table

Membershi	name	lname	straddress	city	postcode	Phone	...
94341001	Ahmet	Kara	yesilbaglar sk.	Istanbul	80325	212-265-22-12	...
94341002	Ali	Ak	kavaklaralti sk.,	Istanbul	80815	212-266-13-13	...
95341003	Zeynep	Tiryaki	Cumhuriyet cad.	Istanbul	80466	216-354-12-32	...
96341004	Fatma	Gulsever	papatya sk.	Istanbul	80860	212-229-54-54	...

Movies Table (sample records)

movieID	title	company	category	relasedate	rating
101	Casablanca	Warner B.	CL	11-Sep-42	none
102	African Queen	20th century	CL	12-Oct-51	none
103	Dirty Harry	warner B.	AC	18-Mar-71	R
104	Star Wars	20th century	SF	25-May-77	PG
105	Friday the 13 th	Paramount	HO	09-Jan-80	R
106	Star Trek	Paramount	SF	20-Dec-79	G
107	Raiders of the Lost	Paramount	AC	23-Jul-81	G

Inventory Table (sample records)

Tapeno	movieID	purchasedate	purchaseprice	rentalperday
01	101	01-May-95	\$39.75	\$1.99
02	101	05-Jan-95	\$39.75	\$1.99
03	107	05-Mar-95	\$42.95	\$2.49
07	102	03-Mar-94	\$39.99	\$1.99
08	102	03-Mar-94	\$39.99	\$1.99
09	102	03-Mar-94	\$39.99	\$1.99
10	105	07-Mar-95	\$34.99	\$2.49
11	105	07-Mar-95	\$34.99	\$2.49

Transactions Table (sample records)

ID	Tapen	membno	rentaldate	returndate
1	02	95341003	09-Sep-99	11-Sep-99
2	05	95341003	09-Sep-99	11-Sep-99
3	04	94341001	05-Dec-99	06-Dec-99
12	01	96341004	04-Oct-99	06-Oct-99
13	02	94341002	05-Nov-99	07-Nov-99
14	21	94341002	31-Jan-00	
15	05	94341002	31-Jan-00	

Lesson 5: Queries That Combine Data From Multiple Tables

Sometimes we need to have queries that span several tables (e.g. all rentals of a person we only know by name). Therefore we need to have several tables in our query. Based on the established relationships, those are then “joined” (there are several types of joins like inner, outer etc. which have slightly different results). This actually means that internally the tables are de-normalized (e.g. the name of persons are added to their ID for each rental incident).

Query 5: Suppose that a customer walks into the store and asks if the movie "Casablanca" is available for rental in MH right at that time.

To find out, we have to build a query that uses multiple tables.

- In the database view, click on queries, then choose design view.
- In the “*Show Table*” dialog box, add movies, inventory and transactions, and *close* the Show Table dialog box.
- In query design, include title from movies table, tapeno from inventory table, rental date and return date from the transactions table.
- Type “Casablanca” in *Criteria* row under title.

Note: This output is not appropriate in “real-life” since it lists all the transaction information about the movie Casablanca.

Lesson 6: Queries With Calculated Fields

You can have additional fields beyond those contained in the tables in any query. This is done by entering information, like most often an expression (formula) into an empty column. Access puts brackets [] around the field names. It will also give a name to this calculated field (e.g. Expr1). This name can be changed to something more meaningful, for instance, age, by editing the column in the design view. This is used for the output, but can also be used as a variable name in further expressions. There is also an entry in the context menu, Build, which helps in constructing expressions.

Query 6: Suppose that one of the owners wants to see a list of her customers and how old they are, i.e. she wants to list the name, last name, occupation and age of each customer. We can build a query with a calculated field to bring this list.

- In the database view, step on queries, new and design view, and OK.
- In the *show table dialog box* that comes up, add members table and *close the dialog box*. Now, bring the fields name, last name, occupation.
- Next to the occupation column, in the field row, type `year(now())-year(bdate)`.
- There are many ways to compute the age of a member based on the birth date field in the *members* table. The above formula uses the year() function, which returns the year of a given date supplied to it as an argument inside the parentheses. This computes the age of the person in years.

Query Exercises

Query 7: List the number of movies in each category.

Query 8: List the tape numbers and number of rentals for each movie by movie name.

Query 9: List the minimum, average and maximum number of rental days for each movie by movie name.

Query 10: List the revenue for each movie by movie name.

Query 11: List the total number of rental days per customer.

Query 12: List the mean age of members renting movies per movie category.

Query 13: For each customer, list the number of rentals per movie category.

Lesson 7: Designing Forms

Refer to the Hotel database from the website. **Important!** First, copy the file hotel.mdb from the above location **into your work area** or the Desktop. Then open **your copy** of the file and work on it. Rename this file (e.g. as *yourusername.mdb*).

Forms.

A form is a template that fits the data in the underlying table(s). You scroll through the data in the table by using this form. You can actually enter new data using this form. To enter a new record, at the end of the records a blank form appears.

Designing a simple form using the wizard.

The Wizard is a guided menu which helps in designing forms. Mostly, this is based on choosing between a few standard layouts, and especially on choosing the relevant attributes.

- In the database view (main view), select Forms from Objects (or the Create tab), and choose Form wizard.
- In the tables/queries box, choose which attributes from which tables to include. In our example, choose the Room table. Choose all of the fields of the Room table (can also be done by clicking >>).
- Next choose a layout / format like columnar.
- Finally, select a style. This only controls visual aspects.
- Lastly, the form is saved.

Designing a form with subform using the wizard.

More complicated forms can also be developed with the wizard. Especially, using relationships set up, a form with a subform for related records can be developed. As another exercise, we want to develop a form to view reservations of each customer and enter new reservation data for a particular customer.

- Create a new form, again using the wizard.
- In the first dialog box, select **Guest** table (the "one" side of the one-to-many relationship) from the list. Here we would like to create a Guests form that displays reservations for each Guest in a subform. Select the relevant fields.
- In the same dialog, change to a different table (reservations - the "many" side of the one-to-many relationship), and again select the relevant fields.
- **Note:** It doesn't matter which table or query you choose first.
- In the next step, the wizard asks which table or query you want to view by (we use by Guest).
- There is also an option for Form with Subform(s), which we select.
- The remaining steps are identical to simple forms.

- Note that this results in two forms which are saved separately.

Lesson 8: Designing Reports

Designing reports using the wizard.

Reports are prefabricated, and nicely formatted queries. Actually, a report can be based on the output of a query, in addition to field from tables. Reports can also be exported and used in MS Word etc. We want to have a report displaying the monthly number of guests.

- In the database view (main view), select Reports from Objects (or the Create tab), and choose Report wizard.
- Next, the table to base the report on needs to be selected (we use Reservations).
- We also need to specify the fields to be used (ArrivalDate and NoOfGuests).
- Next, any grouping has to be specified. In our example, we choose ArrivalDate, which has month as default grouping options. Other options can be selected as well.
- For the remaining variables, in the next step the summary options are specified (like sum, average, count etc.).
- The next steps are similar to forms and select layout and style.
- The report is saved, and can be changed further.

Lesson 9: Lookup

A lookup means that the values for a field are limited to the values contained in another field, in general from another table or a query. This can be used to limit the possibilities for inputs to existing values (in most cases this will be used for foreign key fields).

- To enable this, go to the design view of a table and select the field to be used in that way.
- Then, go to the lookup tab in the lower part of the screen.
- There, change input to combo, this will open additional rows for defining that.
- Choose source (table/query) and field from that source that contains the usable values.
- There are some other possible options for that (e.g. whether the user could enter new values which are then added in the other table).
- Try that for the hotel database: A reservation is only allowed for an existing room.

Query Exercises (continued)

For these queries, we are using the database “vendors” from the website.

Query 14: List the balances of each month of every year.

Query 15: List the maximum balance and the number of vendors of each city.

Query 16: Show the balances of people whose due dates are in 2006. Hint: You will need to use a function for this.

Query 17: Show the inflows, outflows and balances according to cities. Hint: You will need to use a logical function (similar to if in MS Excel).

Integration Exercise

Your task is to implement a simple database for your book collection. Each book has a title, company, number of pages and a price. Each book is written by one author, each of which has a name, and a date of birth. Implement this database and fill with at least 5 rows per table.

Query 18: List the number of books and total pages for each author.

Query 19: List the average price of books authored by people born before and after 1970.