**Laboratory work 1**

LEARN ESSENTIALS OF WORK WITH DBMS ACCESS

## I. Problem description

Some enterprise purchases products from various suppliers (both legal entities and individual entrepreneurs). Purchasing is performed using batches and formalized as supply contracts. Each supply contract has unique number and might be concluded with a single supplier. Documents for each contract include product name, supplied amount, and price (in UAH).

Database with following tables might be used to store and process such information using DBMS Access.

1. “Suppliers” table

Each table entry contains following fields:

| Field name | Data type | Field size | Description |
|---|---|---|---|
| SupplierName | Text | 50 | Supplier name |
| SupplierID | Number | Integer | Supplier ID |
| Note | Memo | | Note |

2. “LegalEntities” table

Each table entry contains following fields:

| Field name | Data type | Field size | Description |
|---|---|---|---|
| SupplierID | Number | Integer | Supplier ID |
| TaxNumber | Text | 20 | Tax number |
| VATNumber | Text | 20 | VAT certificate number |

3. “IndividualEntrepreneurs” table

Each table entry contains following fields:

| Field name | Data type | Field size | Description |
|---|---|---|---|
| SupplierID | Number | Integer | Supplier ID |
| LastName | Text | 20 | Last name |

| | | | |
|---|---|---|---|
| FirstName | Text | 20 | First name |
| SecondName | Text | 20 | Second name |
| RegistrationNumber | Text | 20 | Registration certificate number |

4. “Contracts” table

Each table entry contains following fields:

| Field name | Data type | Field size | Description |
|---|---|---|---|
| ContractNumber | Number | Integer | Contract number |
| ContractDate | Date/Time | Short Date | Contract conclusion date |
| SupplierID | Number | Integer | Supplier ID |
| ContractName | Text | 50 | Contract name |
| Comment | Memo | | Note |

5. “Supplied” table

Each table entry contains following fields:

| Field name | Data type | Field size | Description |
|---|---|---|---|
| ContractNumber | Number | Integer | Contract number |
| Product | Text | 50 | Product name |
| Amount | Number | Long Integer | Batch size (items) |
| PricePerItem | Number | Single with 2 decimal places | Price per item (in UAH) |

## II. Implementation steps

1. Create working directory (e.g., D:\ACCLAB).

2. Run DBMS Access.

3. Create new database with the name “sk.mdb”.

4. If the database was created earlier – open it.

5. Create database tables according to the structure outlined above. To create each table:

1) Click “Create”;

2) Select “Table” and enable design view;

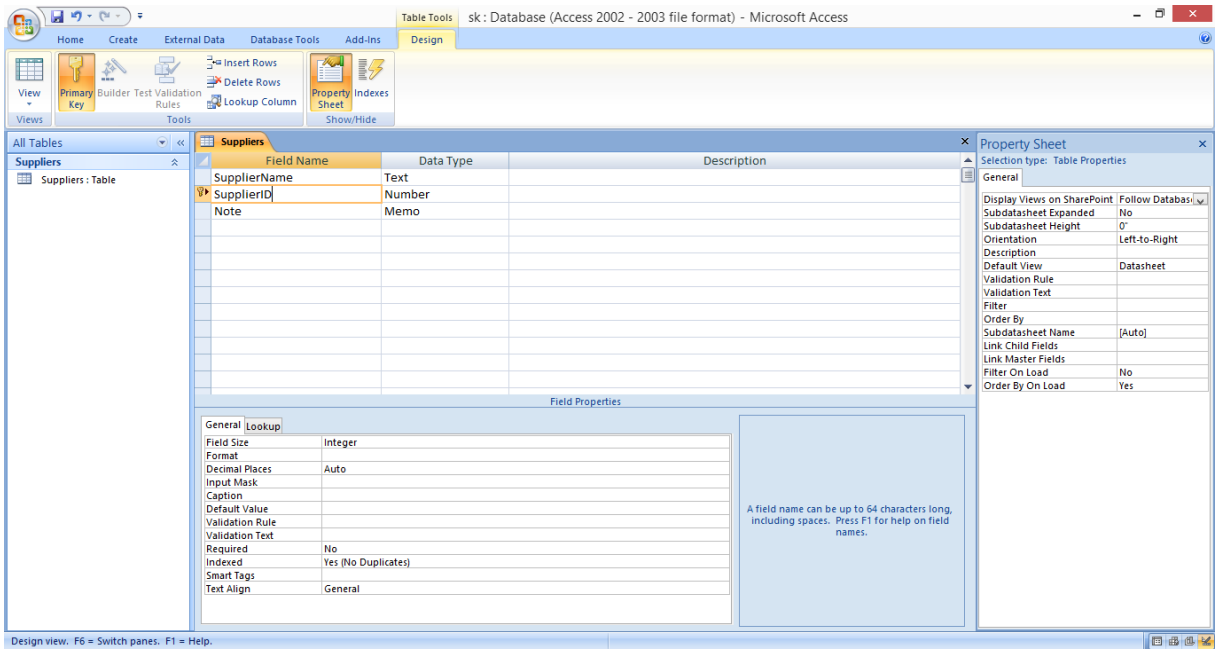



Figure 1.1

3) Type table name and then input table structure information (from problem description section);

4) Input primary key (Pk) information. To make the field primary select it and click “Primary Key”. Attention! Hence, PK might include only a single field!

5) For the table “Suppliers” include the field “SupplierID” into the PK (figure 1.1);

6) For the tables “LegalEntities” and “IndividualEntrepreneurs” include the field “SupplierID” into the PK;

7) For the table “Contracts” include the field “ContractNumber” into the PK;

8) For the table “Supplied” include the field “ContractNumber” into the PK. Unlike the other tables, this table requires a composite primary key that will allow to control unique pair of contract number and product name. Therefore, the field “Product” should be included into the PK. For this click “Indexes” and add “Product” field into the “PrimaryKey” index (figure 1.2);

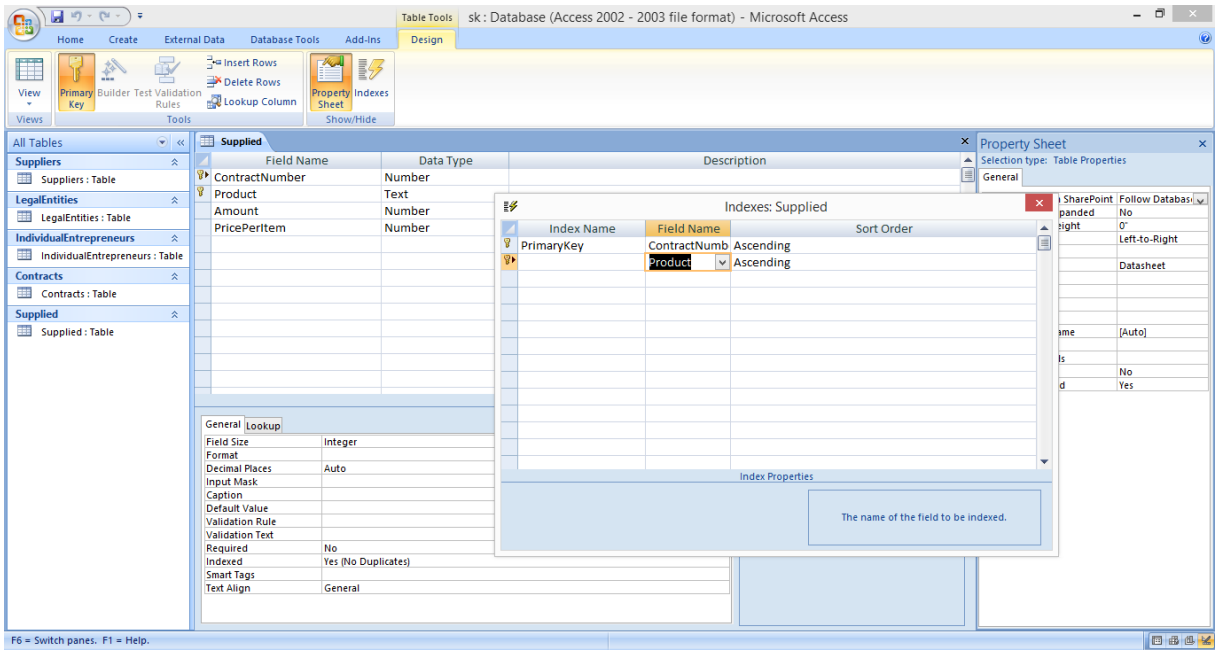



Figure 1.2

9) Save created table for which close design tab.

6. Create relationships between tables for which:

1) Open “Database Tools” tab;

2) Click “Relationships”;

3) Add all created tables to the data scheme;

4) Drag the field “ContractNumber” of the table “Contracts” to the field “ContractNumber” of the table “Supplied”. As the result, relationship between tables will appear (figure 1.3);

5) In the same way create relationship between the field ‘SupplierID” of the table “Suppliers” and the field “SupplierID” of the table “Contracts”;

6) Create relationship between the field “SupplierID” of the table “Suppliers” and the fields “SupplierID” of the tables “LegalEntities” and “IndividualEntrepreneurs”;

7) Double click on each relationship between database tables and check “Enforce Referential Integrity” (figure 1.4);

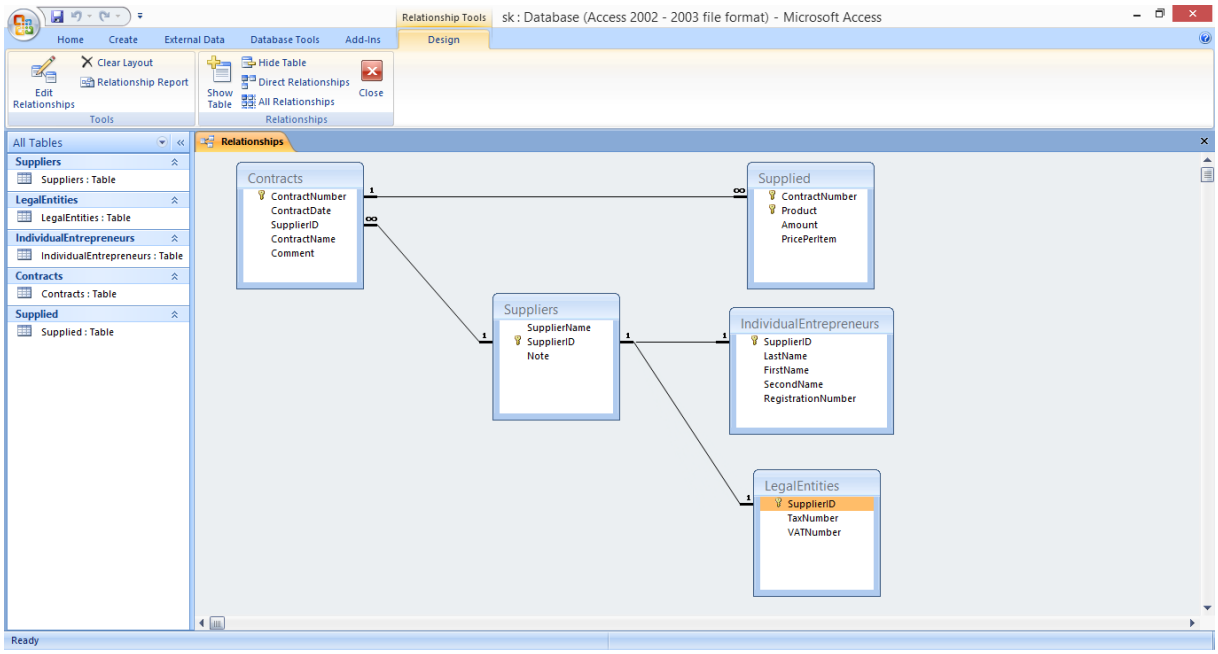



Figure 1.3

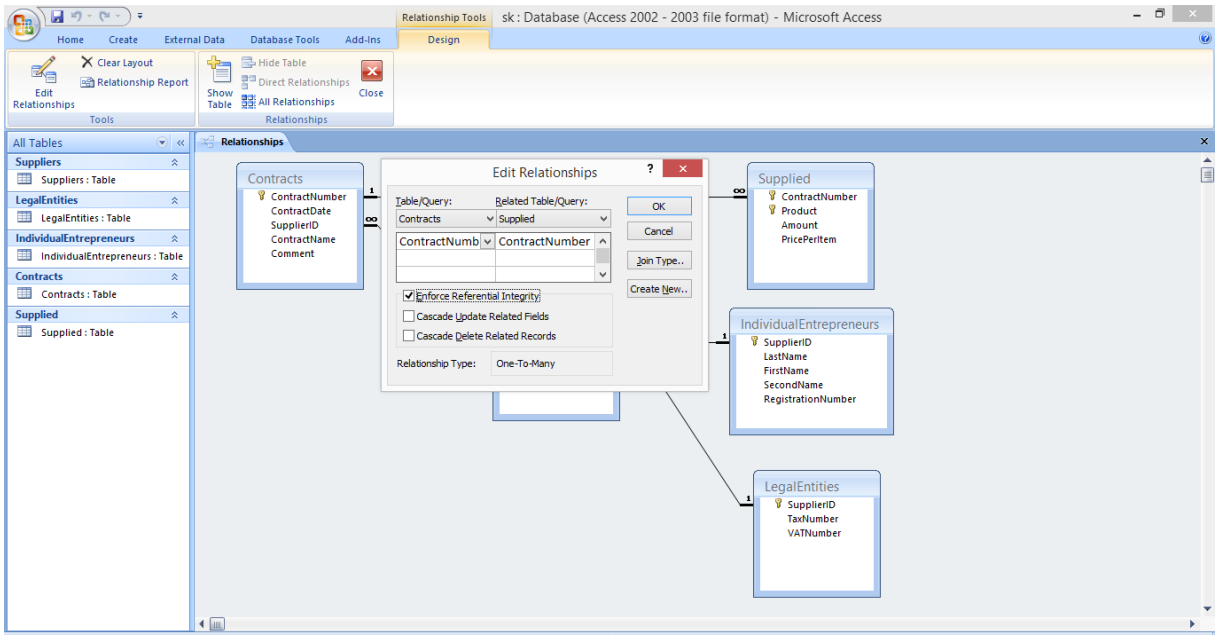



Figure 1.4

8) Save changes and close “Relationships” tab.

7. Enter data into tables for which right click on table and choose “Open”. In case of any troubles check “Tips and tricks” below.

Tables should contain following information:

Attention! Data in the field “Note” of the table “Suppliers” and data in the field “ContractName” of the table “Contracts” are single line strings!

Table “Suppliers”

| SupplierName | SupplierID | Note |
|---|---|---|
| Petrov P. P. PE | 3 | Kharkiv, Nauky Av., 55, office 108, tel. 32-18-44 |
| “Interfrut” LLC | 2 | Kyiv, Peremohy Av., 154, office 3 |
| Ivanov I. I. PE | 1 | Kharkiv, Pushkinska Str., 77 (tel. 33-33-44, 12-34-56, fax 22-12-33) |
| “Transservis” LLC | 4 | Odesa, Deribasivska Str., 75 |
| Sidorov S. S. PE | 5 | Poltava, Svobody Str., 15, apt. 43 |

Table “LegalEntities”

| SupplierID | TaxNumber | VATNumber |
|---|---|---|
| 2 | 00123987 | 19848521 |
| 4 | 29345678 | 25912578 |

Table “IndividualEntrepreneurs”

| SupplierID | LastName | FirstName | SecondName | RegistrationNumber |
|---|---|---|---|---|
| 1 | Ivanov | Illia | Illich | 00123987 |
| 3 | Petrov | Pavlo | Petrovych | 12345678 |
| 5 | Sidorov | Serhii | Stepanovych | 09876541 |

Attention!

Date representation format in DBMS Access corresponds to the Windows system format. To check and modify date representation format change your system date/time configuration in “Time and language” settings.

Table “Contracts” (Attention! Dates are shown in the U.S. format)

| ContractNumber | ContractDate | SupplierID | ContractName | Comment |
|---|---|---|---|---|
| 1 | 9/1/1999 | 1 | Contract 1 | Invoice 34 from 8/30/99 |
| 2 | 9/10/1999 | 1 | Contract 2 | Invoice 08-78 from |

| | | | | |
|---|---|---|---|---|
| | | | | 8/28/99 |
| 3 | 9/10/1999 | 3 | Contract 3 | Invoice 08-78 from 8/28/99 |
| 4 | 9/23/1999 | 3 | Contract 4 | Order 56 from 8/28/99 |
| 5 | 9/24/1999 | 2 | Contract 5 | Invoice 74 from 9/11/99 |
| 6 | 10/1/1999 | 1 | Contract 6 | Invoice 9-12 from 9/28/99 |
| 7 | 10/2/1999 | 2 | Contract 7 | Invoice 85 from 9/21/99 |

Table “Supplied”

| ContractNumber | Product | Amount | PricePerItem |
|---|---|---|---|
| 1 | TV | 10 | 1253.45 |
| 1 | Audio Player | 25 | 655.12 |
| 1 | Video Player | 12 | 722.33 |
| 2 | Stereo System | 11 | 511.43 |
| 2 | Audio Player | 5 | 455.14 |
| 2 | Video Player | 8 | 450.67 |
| 1 | Stereo System | 12 | 220.45 |
| 1 | PC | 24 | 1554.22 |
| 2 | PC | 43 | 1453.18 |
| 3 | TV | 52 | 899.99 |
| 3 | Audio Player | 11 | 544.00 |
| 3 | Display | 85 | 545.32 |
| 4 | TV | 56 | 990.56 |
| 4 | Audio Player | 22 | 323.19 |
| 4 | Printer | 41 | 350.77 |
| 5 | TV | 14 | 860.33 |
| 5 | Audio Player | 33 | 585.67 |
| 5 | Video Player | 17 | 850.12 |
| 4 | Stereo System | 27 | 330.55 |
| 5 | Display | 44 | 590.23 |
| 6 | TV | 34 | 810.15 |
| 6 | PC | 32 | 1850.24 |
| 6 | Display | 51 | 520.95 |
| 7 | TV | 62 | 900.58 |
| 7 | PC | 15 | 1234.56 |
| 7 | Display | 22 | 389.75 |

**Tips and tricks.**

**A.** To add new record into a table select the last empty row of the table. After the input is done leave this row and move to the next.

**B.** Right click on the record and click "Delete Record" to remove it.

**C.** Use blocks of data while copy-pasting it to a database table.

8. Take a look at the referential integrity features. Make some delete/update operations in the related tables.

9. Save file "SK.mdb".

**Report requirements:**

1) Briefly describe main steps of this work;

2) Depict structure of the created database including relations between the tables;

3) Describe the referential integrity features that you analyzed.

## Questions

1. Explain why the proposed database structure was selected to store information. Which shortcomings this database structure has?

2. Is it possible to change this database structure? How?

3. Briefly describe each DBMS Access data type.

4. What is purpose of the "AutoNumber" field type? What are advantages and disadvantages of this field type? Why it was not used to design database tables in this work?

5. How to create a database using DBMS Access?

6. How to modify a database created using DBMS Access (create new table or change structure of the existing table)?

7. How to add new record into a table?

8. How to delete one or several records from a table?

9. How to sort data within a table by any field?

10. How to change date representation format?

11. What is required to enable referential integrity between tables?

12. How to modify referential integrity type?

13. Advantages and disadvantages of the “Restrict” referential integrity.

14. Advantage and disadvantage of the “Cascade” referential integrity.

15. What is index?

16. List basic index types.

17. List basic index functions.

18. What is a table primary key?

19. How to setup a primary key for a table?

20. How to create a composite primary key (this key contains several fields)?

21. How to make table field unique (do not allow to input repeating values)?

22. How to change fields order within a table structure?

## Laboratory work 2

CREATE MULTITABLE FORMS USING DBMS ACCESS

**Prepare to work**

1. Move the database file **SK.mdb** created during the previous work into another folder (e.g., D:\ACC_LAB_2).

2. Run DBMS Access.

3. Open the database created earlier.

**Follow the steps below**

1. Create main form using designer:

1) Open “Create” tab, click “Form Design”, select table “Contracts” as the “Record Source” in properties sheet;

2) Resize form (approximately: 8x4 inches), type “Supply information” into “Caption” within “Format” tab in “Property Sheet” window;

3) Click “Add Existing Fields” and drag all “Contracts” table fields to the form, change size and place of these elements (figure 2.1);

4) Switch to the form view and view records for all contracts.

2. Create combo box to look up the records:

1) Switch to the design view, select “Combo Box” control and drag it to the form, select first option, check “Suppliers” table, select all available fields, name this combo box “Suppliers”;

2) Bound created combo box to the supplier ID on the main form. Select combo box, select “SupplierID” as the control source on the “Data” tab in the property sheet, click “After Update” on the “Event” tab, and select “Code builder”, then type inside procedure Combo8_AfterUpdate():

SupplierID = Combo8.Column(0)

Close Visual Basic window and back to the Access workspace;

3) Place the combo box and its label as it is shown in figure 2.2, switch to the form view and ensure that combo box works correct (figure 2.3).

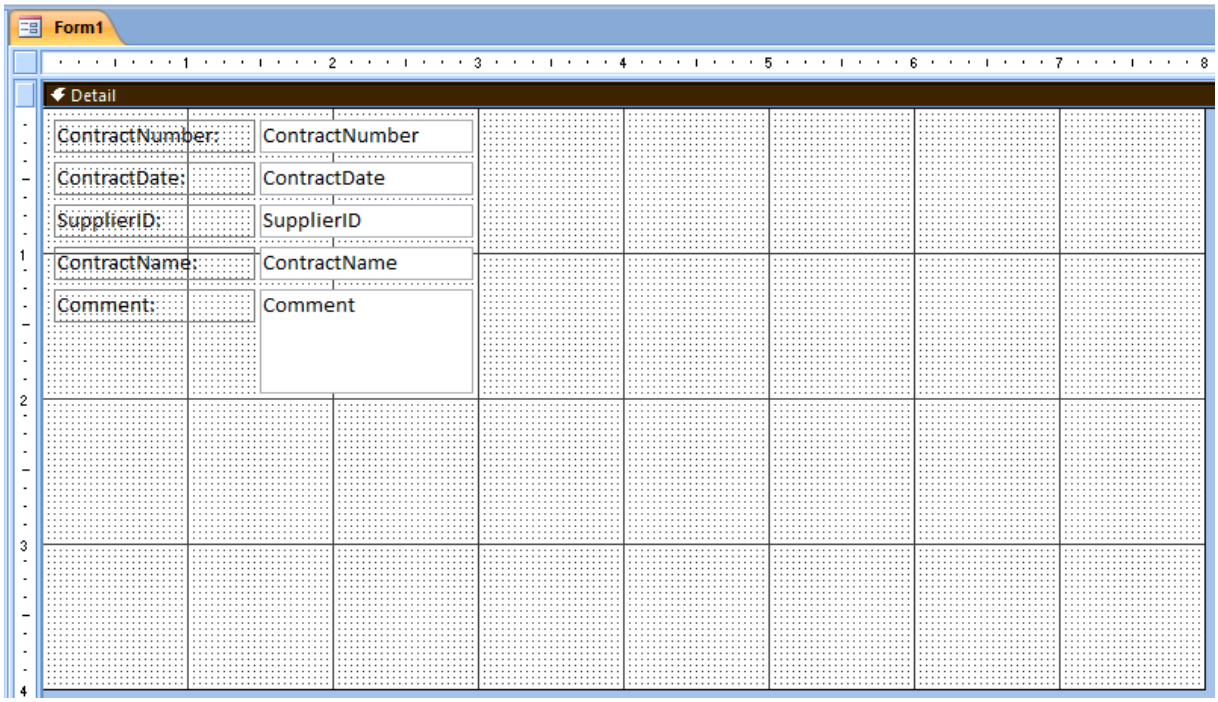



Figure 2.1

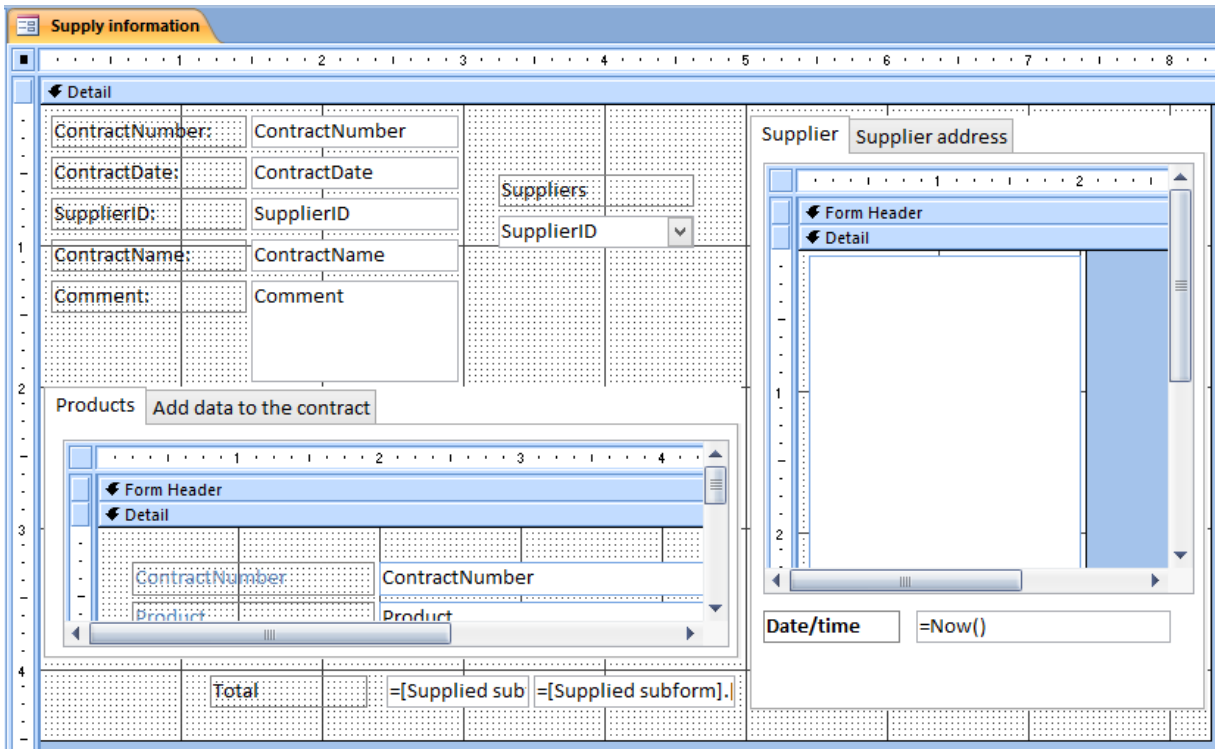



Figure 2.2

3. Add “Tab Control” object to the main form:

1) Drag “Tab Control” to the form, place it where it is necessary and resize. Add another tab control in the same way (figure 2.2);

2) Rename the first tab page using “Other” tab on the “Property Sheet” window, type “Products” to the “Name” field. Call other tabs: “Add data to the contract”, “Supplier”, “Supplier address”.

4. Place subfrom “Supplied subform” to the “Products” tab:

1) Drag “Supplied” table to the “Products” tab, click “Next” and then “Finish” in the wizard window;

2) Remove subform name and resize it (figure 2.2). Set “No” for the “Navigation Buttons” property on the “Format” tab, as well as for the “Allow Additions” and “Data Entry” on the “Data” tab.

3) Resize columns and check bounding of the main and subform in the form view (figure 2.3).

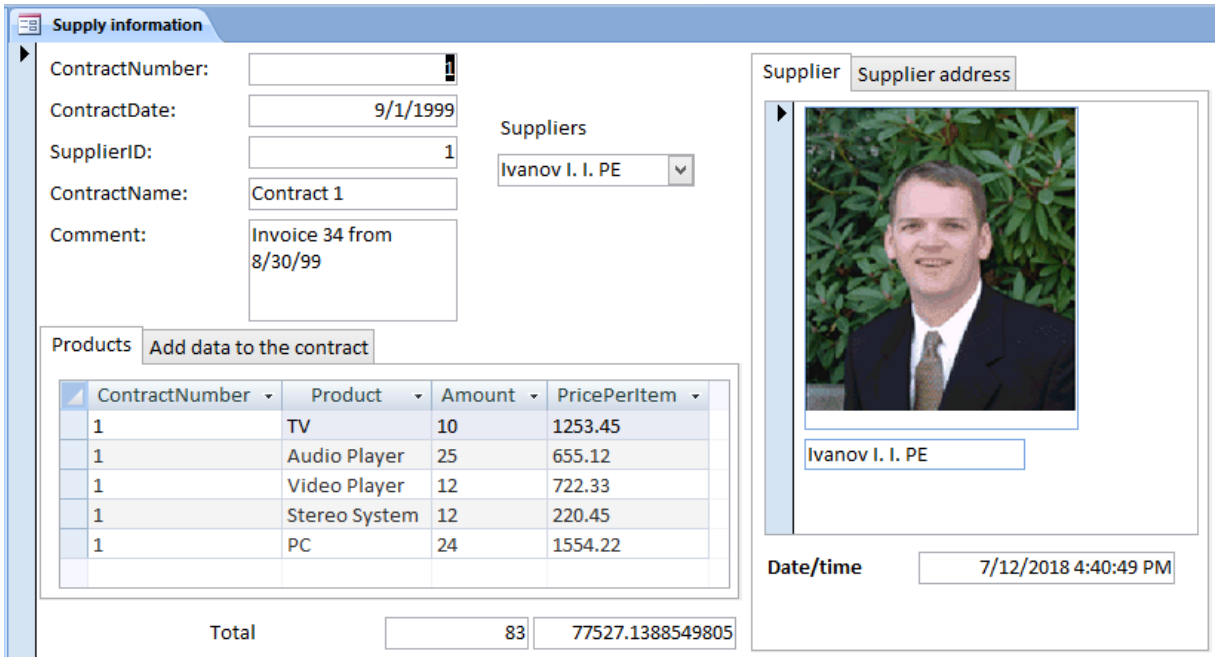



Figure 2.3

5. Add subform “AddData” to the “Add data to the contract” tab except the “ContractNumber” field (figure 2.4). Set value “Yes” for the “Data Entry” property. Switch to the form view and add data to the contract (figure 2.5).

6. Add “OLE Object” field to the “Suppliers” table:

1) Open “Suppliers” table in the design view;

2) Add new field “PhotoOLE” of type “OLE Object”, set “Supplier photo” as the description and save changes;

3) Open “Suppliers” table;

4) Right click on the “PhotoOLE” field of the first entry, click “Insert Object”, then select “Create from file” and upload supplier’s photo;

**Attention! Ask teacher about the suppliers’ photos.**

5) Also add photo to the “Petrov P. P. PE” supplier.

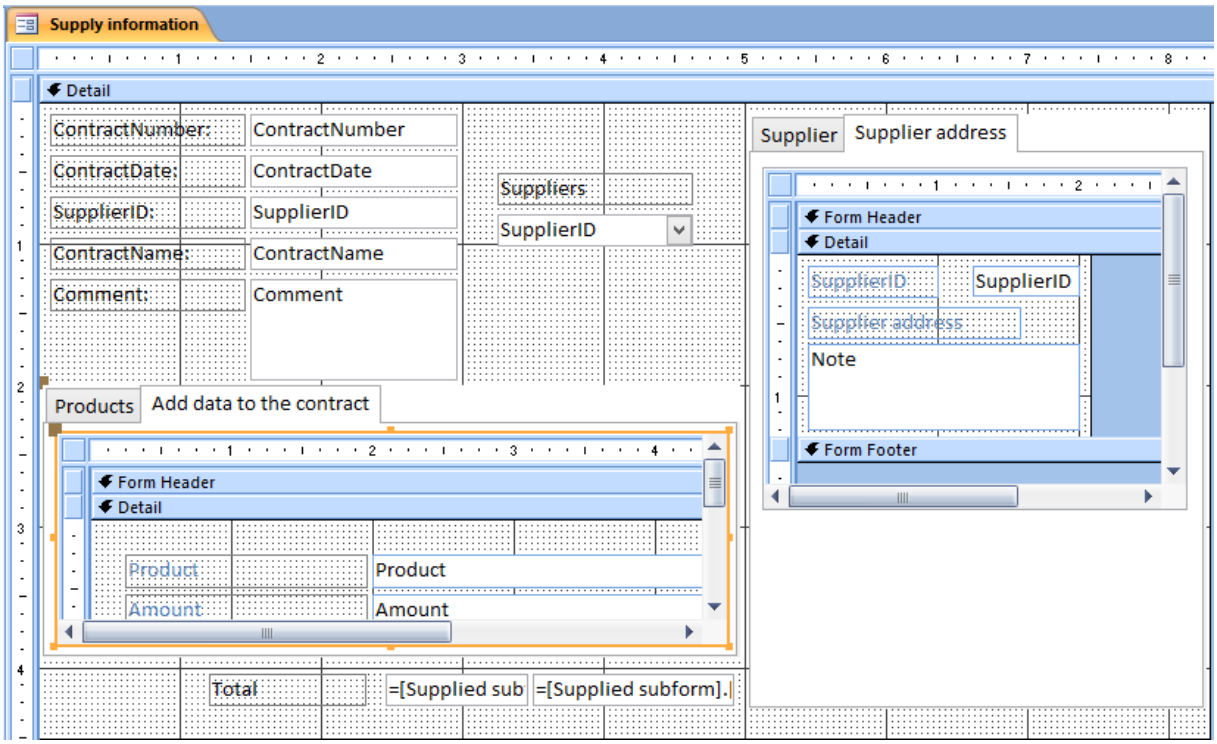



Figure 2.4

7. Create subform “sbfSupplier” using form wizard:

1) Click “More Forms” on the “Create” tab, select “Form Wizard”, select “Suppliers” table as the source;

2) Select “SupplierName” and “PhotoOLE” fields;

3) Select “Justified” form layout;

4) After form created, switch to the design view and remove “Form Header”, then change form size (figure 2.2). Set “Navigation Buttons” property to “No” and save changes.

8. Drag the subform “sbfSupplier” to the “Supplier” tab of the main form. Remove subform header and resize the subform if necessary.

9. Create subform “sbfSupplierAddress” using the form wizard:

1) Select “Suppliers” table as the data source;

2) Select “SupplierID” and “Note” fields;

3) Select “Justified” layout;

4) After form created, switch to the design view, then remove form header and change “Caption” property for the “Note” label to “Supplier address”. Resize this form if necessary (figure 2.4). Set “No” for the “Navigation Buttons” property. Set “Neither” for the “Scroll Bars” property. Save changes.

10. Drag the subform “sbfSupplierAddress” to the “SupplierAddress” tab of the main form. Remove subform header and resize the subform if necessary. Switch to the form view and check that form works correctly (figure 2.5).

11. Create the calculated field to display current date and time:

1) Drag the “Text Box” control to the bottom of the “Supplier” tab;

2) Rename label to “Date/time” (bold);

3) Input =Now to the field in order to show current date and time (figure 2.2);

4) Switch to the form view and check changes (figure 2.3).

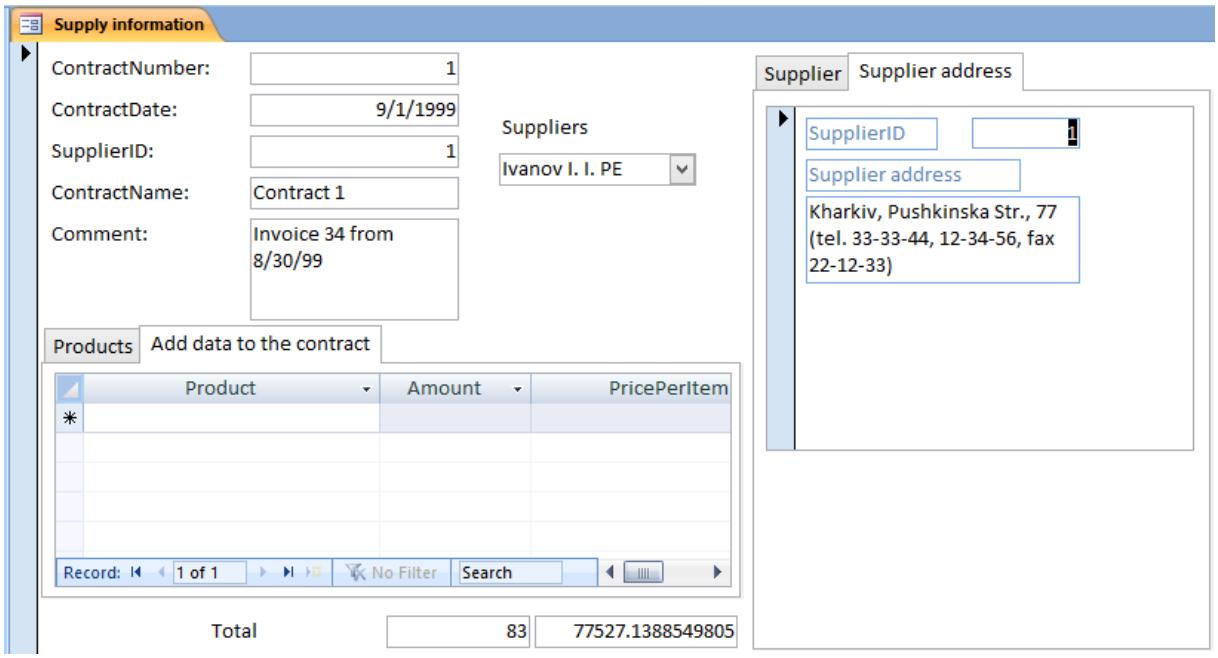



Figure 2.5

12. Check the form correctness by adding records about supplied products. It is obvious that form does not work correctly. If you switch to the "Products" tab right after input new record, created record would be displayed only after navigating through the contracts. To fix this defect select subfrom on the "Add data to the contract" tab, choose "After Insert" event, open "Code Builder", and insert following code to the procedure Form_AfterInsert():

Me.Parent![Supplied subform].Requery

Me.Parent![Supplied subform].SetFocus

Attention! Check subform name to ensure correct execution of the edited procedure.

13. Close Visual Basic editor and return to the Access environment. Check data insert correctness.

To fix products delete operation open code builder of the "After Del Confirm" event of the subform "Supplied subform" and insert following coed to the procedure Form_AfterDelConfirm():

Me.Parent [Supplied subform].Requery

Attention! Check subform name to ensure correct execution of the edited procedure.

14. Developed form provides main data operations (view, add, remove) but does not provide user with the total data (amount of supplied products and supply cost) for each contract. To fix this issue:

1) Open “Supplied subform” in the design view;

2) Create two text boxes with names “ContractAmount” and “ContractCost” in the “Form Footer” section (enable this section using right click on the form space if necessary) (figure 2.6). Insert following expressions to this text boxes:

=Sum([Amount])

=Sum([Amount]*[PricePerItem])

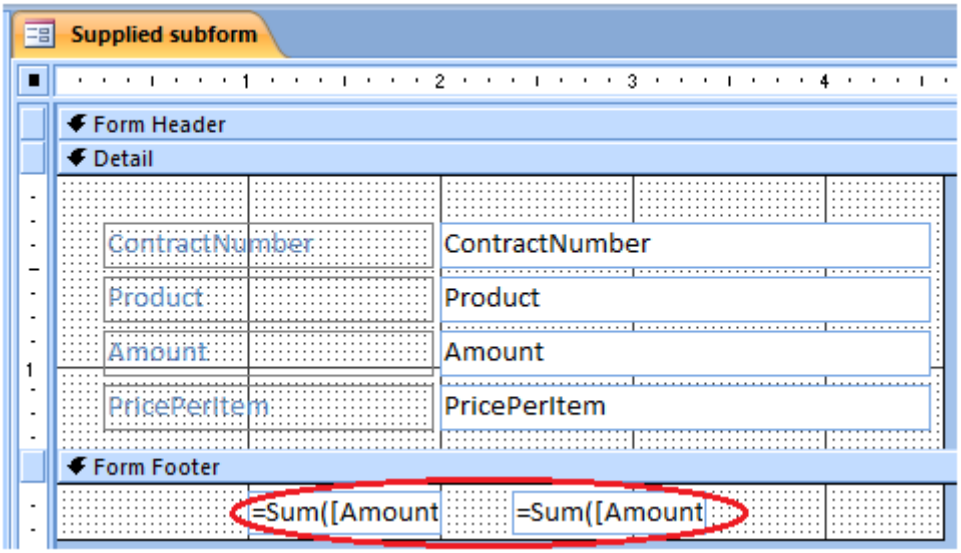



Figure 2.6

3) Save changes and close this subform;

4) Open the “Supply information” form in the design view;

5) Create one label and two text boxes as it is shown in figure 2.7. Insert following expressions to this text boxes:

=[Supplied subform].[Form]![ContractAmount]

=[Supplied subform].[Form]![ContractCost]

Attention! Check subform name to ensure correct execution of the edited procedure.

6) Save changes and close form;

7) Check form correctness. Its appearance should be as it is shown in figure 2.3.

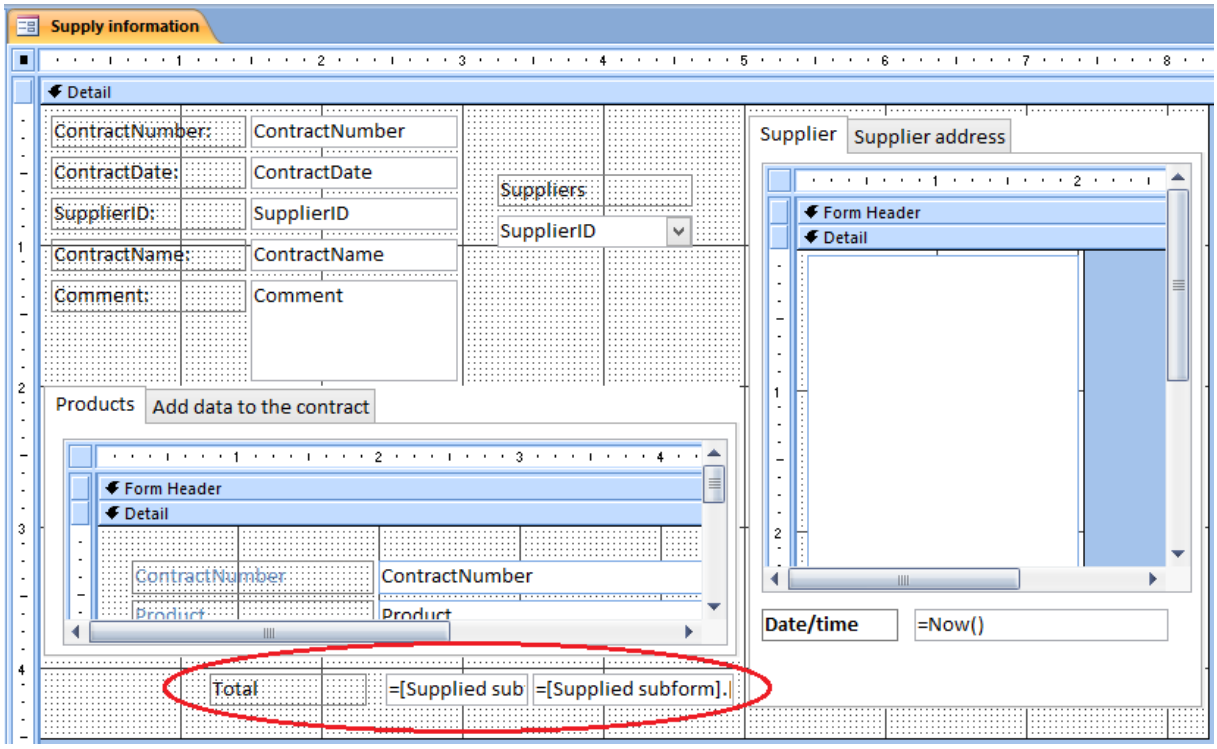



Figure 2.7

**Report requirements:**

1) Briefly describe basic steps of this work;

2) Depict the main form both in the design and final view;

3) Describe form controls: text box, combo box, OLE object, calculated fields;

4) Describe how to create and use tab controls;

5) Describe how to create and use subforms;

6) Describe usage of the properties sheets while creating form in the design view;

7) Describe usage of forms to work with data.

## Questions

1. How to use toolbox? Make an example.

2. How to change form size using designer? How to justify elements on a form?

3. Describe purpose of properties tabs “Events” and “Others”. How they have been used during this work?

4. How to add subform to that main form?

5. How to bind main form with subform by common field?

6. How to use the list of fields in the design view?

7. How to create combo box to look up records?

8. What is the purpose of “Bound Column” property of a combo box?

9. What is the purpose of “Row Source” property of a combo box?

10. List features of “Row Source” property you have used in this work?

11. Describe usage of properties tab “Form”.

12. How to add OLE object field to the table?

13. How to add tab control to the main form?

14. How to create calculated field to display current date and time?

15. How to create calculated field to display total amount of products supplied by a certain contract?

16. How to create calculated field to display total cost of products supplied by a certain contract?

17. Explain the purpose of the AfterUpdate() procedure for a combo box.

18. Explain how the AfterUpdate() procedure works for the combo box with the list of suppliers.

19. Explain the purpose of the Form_AfterInsert() procedure.

20. Explain how the Form_AfterInsert() procedure works in this work.

21. Explain the purpose of the Form_AfterDelConfirm() procedure.

22. Explain how the Form_ AfterDelConfirm () procedure works in this work.

# Laboratory work 3

LEARN BASIC SQL DATA MANIPULATION COMMANDS

**Prepare to work**

1. Move the database file **SK.mdb** created during the previous work into another folder (e.g., D:\ACC_LAB_3).

2. Run DBMS Access.

3. Open the database created earlier.

**Follow the steps below**

## I. SELECT SQL command. Data processing using SELECT queries.

To implement following SELECT SQL queries open “Create” tab in the Access window.

### 1. Print a list of products delivered by the supplier 1 (Ivanov I.I. PE) for the contract 1.

Steps:

1) Open “Query Design” window and select “Contracts”, “Supplied”, and “Suppliers” tables (figure 3.1);

2) Add the following fields to the query (TableName.FieldName): Contracts.ContractNumber, Supplied.Product, Supplied.Amount, Supplied.PricePerItem, Suppliers.SupplierName, Suppliers.SupplierID (figure 3.1);

3) Set value 1 as the “Criteria” for the fields “ContractNumber” and “SupplierID” (figure 3.1);

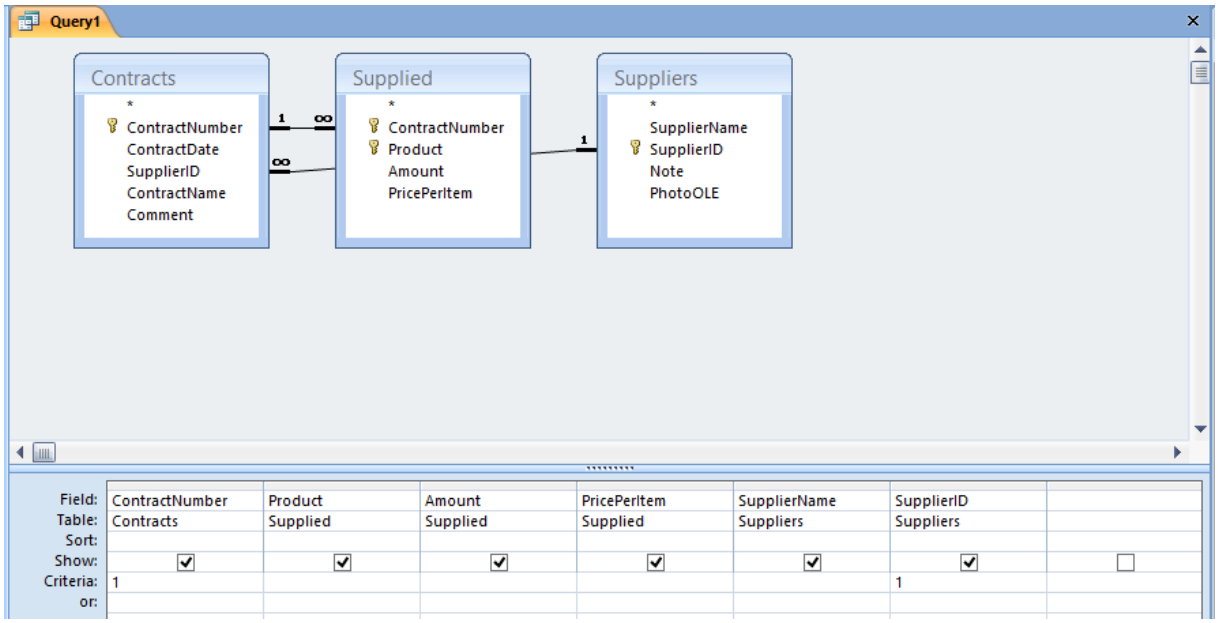



Figure 3.1

4) Switch “View” to “SQL View” and check SQL code of the SELECT command:

```
SELECT
      Contracts.ContractNumber,
      Supplied.Product,
      Supplied.Amount,
      Supplied.PricePerItem,
      Suppliers.SupplierName,
      Suppliers.SupplierID
FROM
      Suppliers INNER JOIN
            (Contracts INNER JOIN Supplied ON Contracts.ContractNumber =
            Supplied.ContractNumber)
            ON Suppliers.SupplierID = Contracts.SupplierID
WHERE
      (((Contracts.ContractNumber)=1) AND ((Suppliers.SupplierID)=1));
```

5) Switch “View” to “Datasheet View” and check result of query execution;

6) Save this query as “Query1” and close.

**2. Print a list of products delivered by supplier 1 (Ivanov I.I. PE) between 9/1/1999 and 9/12/1999 (using "mm/dd/yyyy" date format).**

Use the following SQL command to implement this query:

```
SELECT
      Supplied.ContractNumber,
      Contracts.ContractDate,
      Supplied.Product,
      Supplied.Amount,
      Supplied.PricePerItem,
      Suppliers.SupplierName
FROM
      Suppliers INNER JOIN
            (Contracts INNER JOIN Supplied ON Contracts.ContractNumber =
            Supplied.ContractNumber)
            ON Suppliers.SupplierID = Contracts.SupplierID
WHERE
      (((Contracts.ContractDate) Between #9/5/1999# And #9/12/1999#) AND
((Suppliers.SupplierID)=1));
```

You can use the query designer (figure 3.3) as well, but entering SQL commands is much more preferable and may prevent various mistakes.

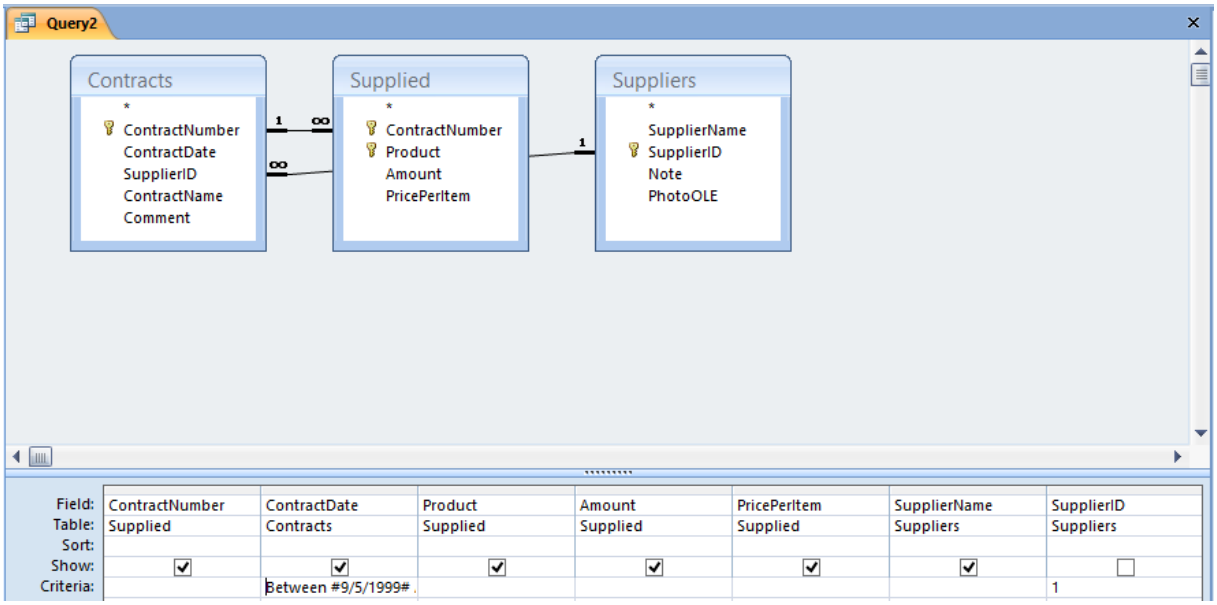



Figure 3.2

Save created query as “Query2” and close.

**3. Print a list of products supplied in September of 1999 with supplier name and supply date.**

Use the following SQL command or use the query designer (figure 3.3):

```
SELECT
      Contracts.ContractNumber,
      Contracts.ContractDate,
      Supplied.Product,
      Supplied.Amount,
      Supplied.PricePerItem,
      (Supplied.Amount * Supplied.PricePerItem) AS Total,
      Suppliers.SupplierName
FROM
      Suppliers INNER JOIN
            (Contracts INNER JOIN Supplied ON Contracts.ContractNumber =
            Supplied.ContractNumber)
            ON Suppliers.SupplierID = Contracts.SupplierID
WHERE
      Month(Contracts.ContractDate) = 9 AND Year(Contracts.ContractDate) =
1999
```

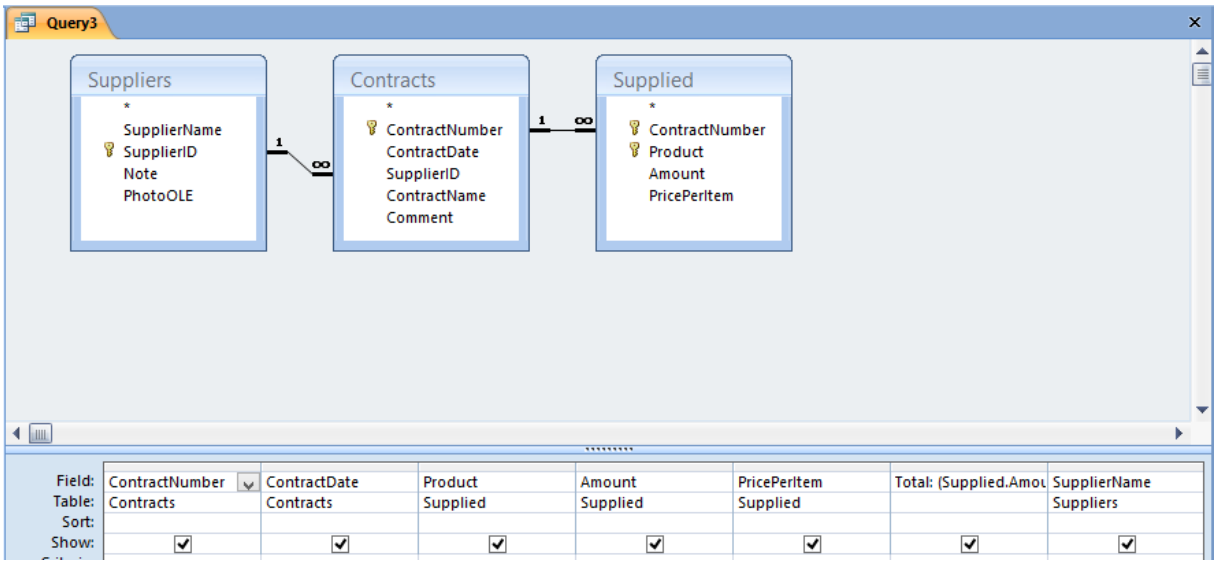



Figure 3.3

Save created query as “Query3” and close.

**4. Print a list of contracts (number, date), total amount of the supplied products and total price for each contract (multiply and sum amount and price for each contract). The list should be sorted by contract numbers (ascending).**

Use the following SQL command or use the query designer (figure 3.4):

```
SELECT
      Contracts.ContractNumber,
      Contracts.ContractDate,
      Sum(Supplied.Amount) AS [TotalAmount],
      Sum(Supplied.Amount * Supplied.PricePerItem) AS [TotalPrice]
FROM
      Contracts INNER JOIN
            Supplied ON Contracts.ContractNumber = Supplied.ContractNumber
GROUP BY Contracts.ContractNumber, Contracts.ContractDate
ORDER BY Contracts.ContractNumber
```

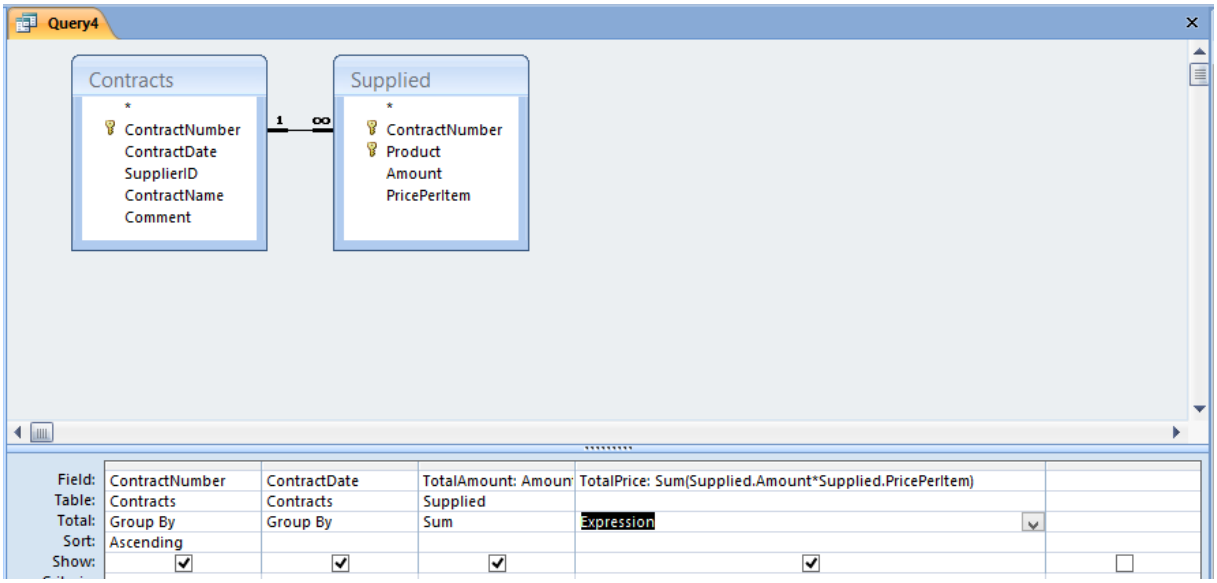



Figure 3.4

Save created query as “Query4” and close.

**5. Print a list of contracts (number, date) with total price for each contract. The list should be sorted by total price for each contract. Records**

**for which contract number is greater than 3 should be excluded from query results.**

Use the following SQL command or use the query designer (figure 3.5):

```
SELECT
      Contracts.ContractNumber,
      Contracts.ContractDate,
      Sum(Supplied.Amount * Supplied.PricePerItem) AS [TotalPrice]
FROM
      Contracts INNER JOIN
            Supplied ON Contracts.ContractNumber = Supplied.ContractNumber
GROUP BY Contracts.ContractNumber, Contracts.ContractDate
HAVING Contracts.ContractNumber <= 3
ORDER BY Sum(Supplied.Amount * Supplied.PricePerItem)
```

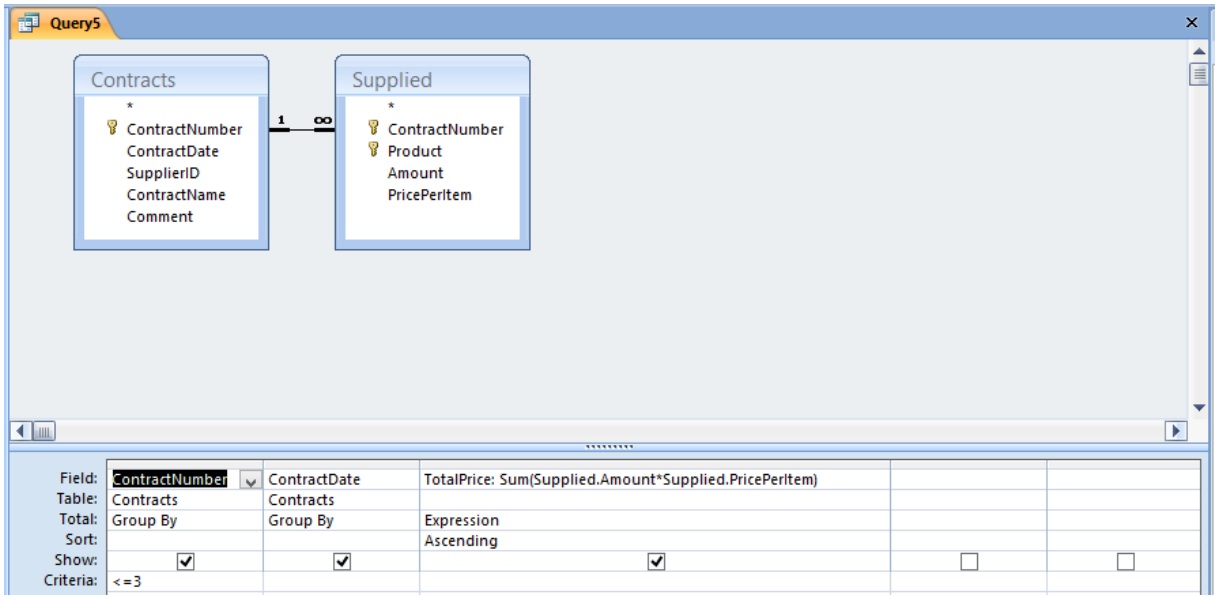



Figure 3.5

Save created query as “Query5” and close.

**6. Print information about the largest product batch among all contracts. Include information about supplier, contract number, and date.**

Use the following SQL command or use the query designer (figure 3.6):

```
SELECT
      Contracts.ContractNumber,
```

```
        Contracts.ContractDate,
        Supplied.Product,
        Supplied.Amount,
        Suppliers.SupplierName
FROM
        Suppliers INNER JOIN
              (Contracts INNER JOIN Supplied ON Contracts.ContractNumber =
              Supplied.ContractNumber)
              ON Suppliers.SupplierID = Contracts.SupplierID
WHERE
        Supplied.Amount = (SELECT Max(Amount) FROM Supplied)
```

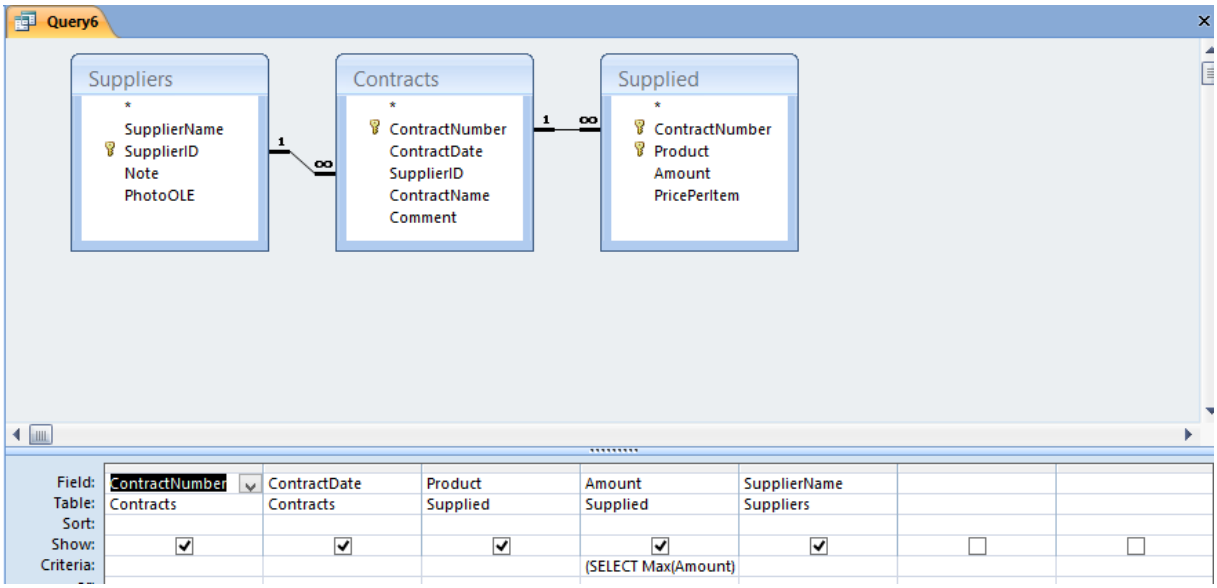



Figure 3.6

Save created query as “Query6” and close.

**7. Print a list of suppliers (name and ID) that have not concluded any contracts.**

Use the following SQL command or use the query designer (figure 3.7):

*Option 1*

```
SELECT
        Suppliers.SupplierName,
        Suppliers.SupplierID
FROM
```

```
        Suppliers
WHERE
        Suppliers.SupplierID NOT IN (SELECT SupplierID FROM Contracts)
```

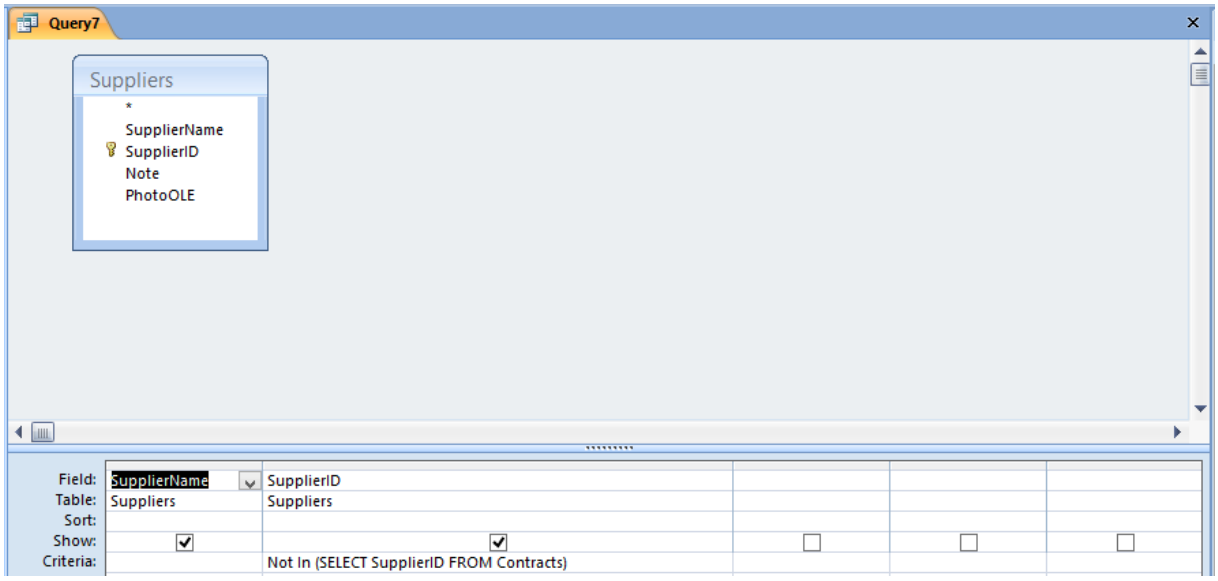



Figure 3.7a

Save created query as “Query7a” and close.

*Option 2*

```
SELECT
        Suppliers.SupplierName,
        Suppliers.SupplierID
FROM
        Suppliers
WHERE
        Suppliers.SupplierID <> ANY(SELECT SupplierID FROM Contracts)
```

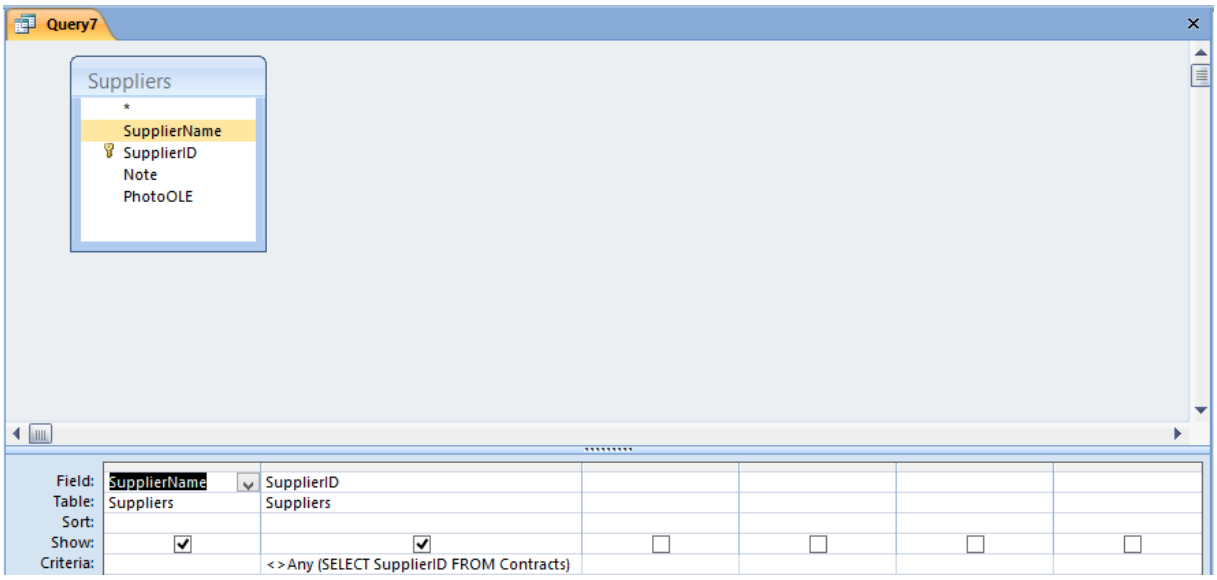



Figure 3.7b

Save created query as “Query7b” and close.

**Warning! Both queries should give the same results.**

But second option might give invalid result (e.g. print all suppliers) in various Access versions. To fix this issue modify the second query as following:

```
SELECT
      Suppliers.SupplierName,
      Suppliers.SupplierID
FROM
      Suppliers
WHERE
      NOT(Suppliers.SupplierID) = ANY(SELECT SupplierID FROM Contracts)
```

Ensure that both queries give the same results.

**8. Print a list of supplied product names with the average price per item (regardless of supplier).**

Use the following SQL command or use the query designer (figure 3.8):

```
SELECT
      Supplied.Product,
      Avg(Supplied.PricePerItem) AS [AvgPricePerItem]
FROM
```

```
      Supplied
GROUP BY Supplied.Product
```

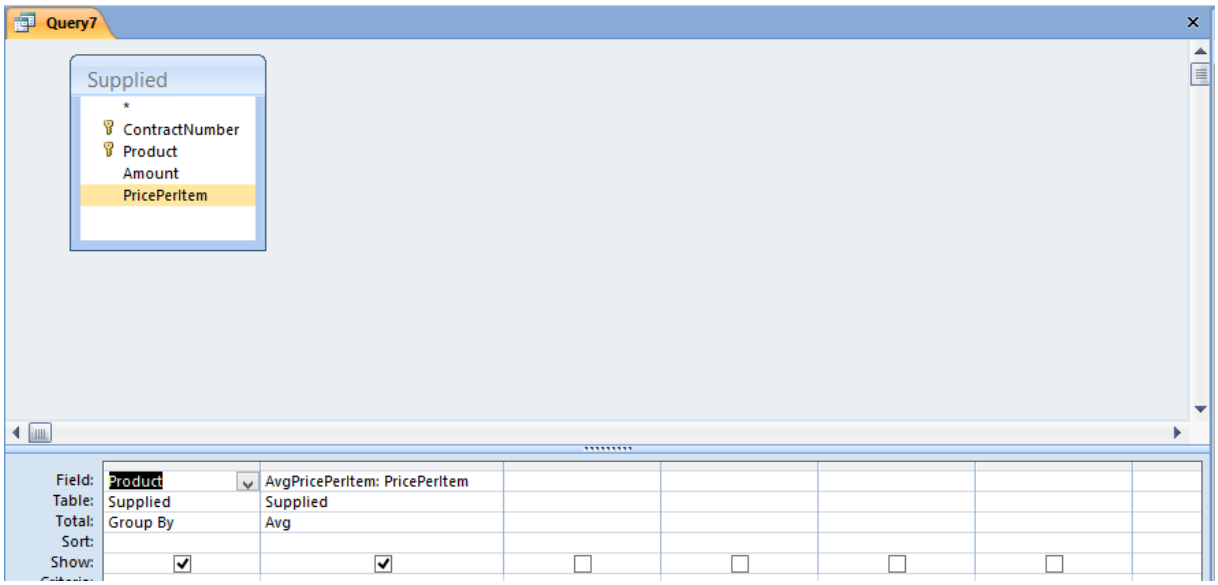



Figure 3.8

Save created query as “Query8” and close.

**9. Print a list of products (name, amount and price, supplier) for which price per item is greater than average.**

Use the following SQL command or use the query designer (figure 3.9):

```
SELECT
      Supplied.Product,
      Supplied.Amount,
      Supplied.PricePerItem,
      Suppliers.SupplierName
FROM
      Suppliers INNER JOIN
            (Contracts INNER JOIN Supplied ON Contracts.ContractNumber =
            Supplied.ContractNumber)
      ON Suppliers.SupplierID = Contracts.SupplierID
WHERE
      Supplied.PricePerItem > (SELECT Avg(PricePerItem) FROM Supplied)
```

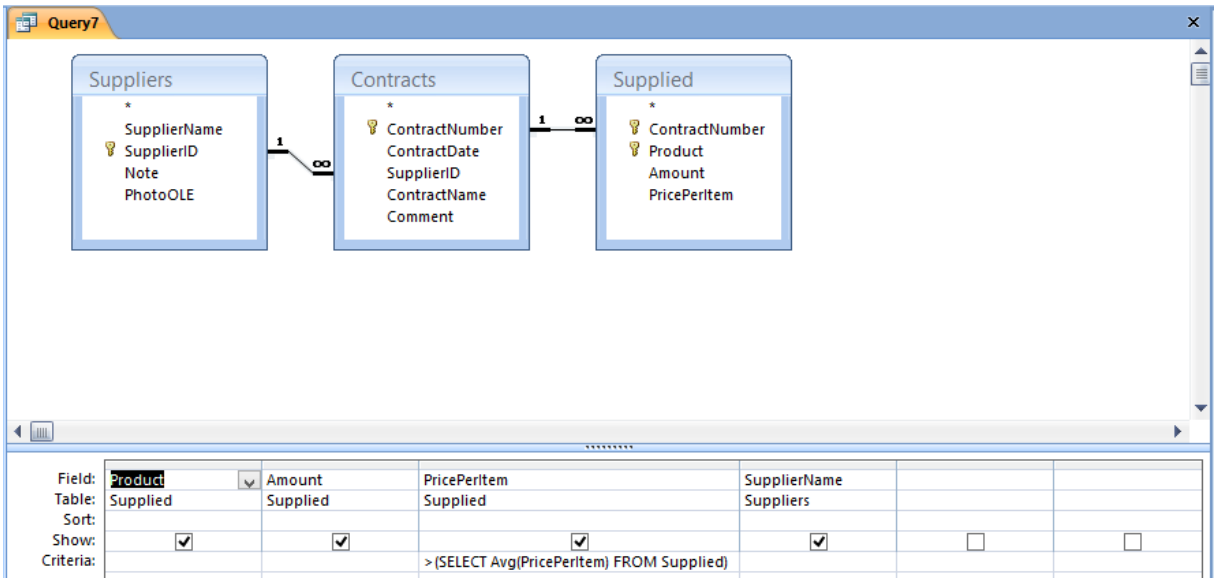



Figure 3.9

Save created query as “Query9” and close.

**10. Print information about top five expensive products (name, price per item, supplier).**

Use the following SQL command or use the query designer (figure 3.10):

```
SELECT TOP 5
      Supplied.Product,
      Supplied.PricePerItem,
      Suppliers.SupplierName
FROM
      (Suppliers INNER JOIN Contracts ON Suppliers.SupplierID = Con-
      tracts.SupplierID) INNER JOIN
      Supplied ON Contracts.ContractNumber = Supplied.ContractNumber
ORDER BY Supplied.PricePerItem DESC
```

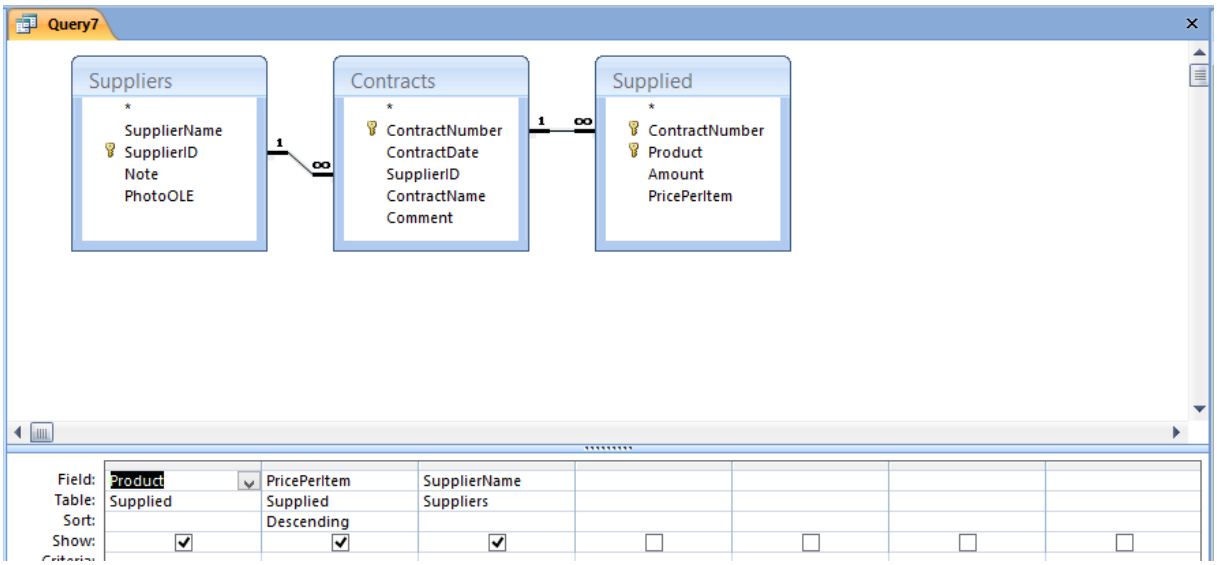



Figure 3.10

Save created query as “Query10” and close.

**11. For each day of September 1999 define price of products delivered by each supplier (include only delivery days).**

Use the following SQL command or use the query designer (figure 3.11):

```
TRANSFORM Sum(Supplied.Amount * Supplied.PricePerItem) AS [Total]
SELECT
      Suppliers.SupplierName
FROM
      Suppliers INNER JOIN
      (Contracts INNER JOIN Supplied ON Contracts.ContractNumber = Sup-
      plied.ContractNumber)
      ON Suppliers.SupplierID = Contracts.SupplierID
WHERE
      Month(Contracts.ContractDate) = 9 AND Year(Contracts.ContractDate) =
1999
GROUP BY
      Suppliers.SupplierName,
      Month(Contracts.ContractDate),
      Year(Contracts.ContractDate)
PIVOT Day(Contracts.ContractDate)
```

This is crosstab query, which defines [Suppliers.SupplierName] as row headings, Day(Contracts.ContractDate) as column headings, and [Total] as cell values.

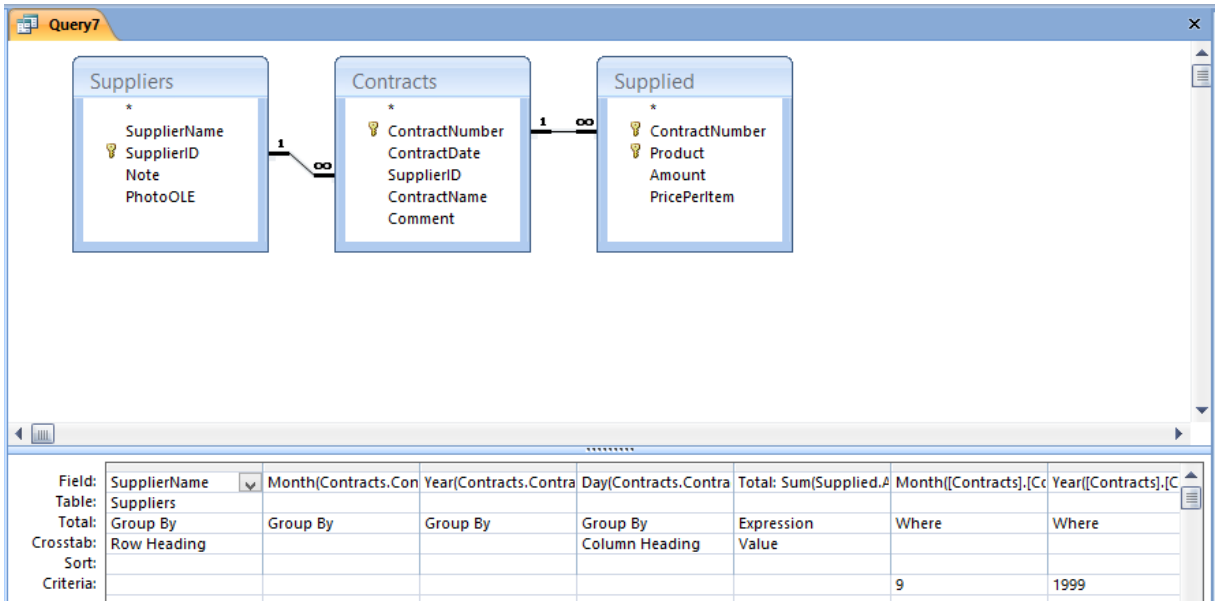



Figure 3.11

Save created query as “Query11” and close.

**12. Create a list of contracts (only numbers), total amount of the supplied products, and total price for each contract. Print full names (last name, first name, and second name) of suppliers that are private entrepreneurs, as well as tax numbers of legal entities.**

Use the following SQL command or use the query designer (figure 3.12):

```
SELECT
      Contracts.ContractNumber,
      Sum(Supplied.Amount) AS [TotalAmount],
      Sum(Supplied.Amount * Supplied.PricePerItem) AS [TotalPrice],
      LastName & " " & FirstName & " " & SecondName AS [SupplierFullName],
      TaxNumber
FROM
      ((Contracts LEFT JOIN IndividualEntrepreneurs ON Contracts.SupplierID
      = IndividualEntrepreneurs.SupplierID) LEFT JOIN
```

```
            LegalEntities          ON          Contracts.SupplierID          =
            LegalEntities.SupplierID) INNER JOIN
            Supplied ON Contracts.ContractNumber = Supplied.ContractNumber
GROUP BY
      Contracts.ContractNumber,
      LastName & " " & FirstName & " " & SecondName,
      TaxNumber
ORDER BY Contracts.ContractNumber
```

Here LEFT JOIN is used to join all records from “Contracts” and only matching records from “IndividualEntrepreneurs” and “LegalEntities” tables.

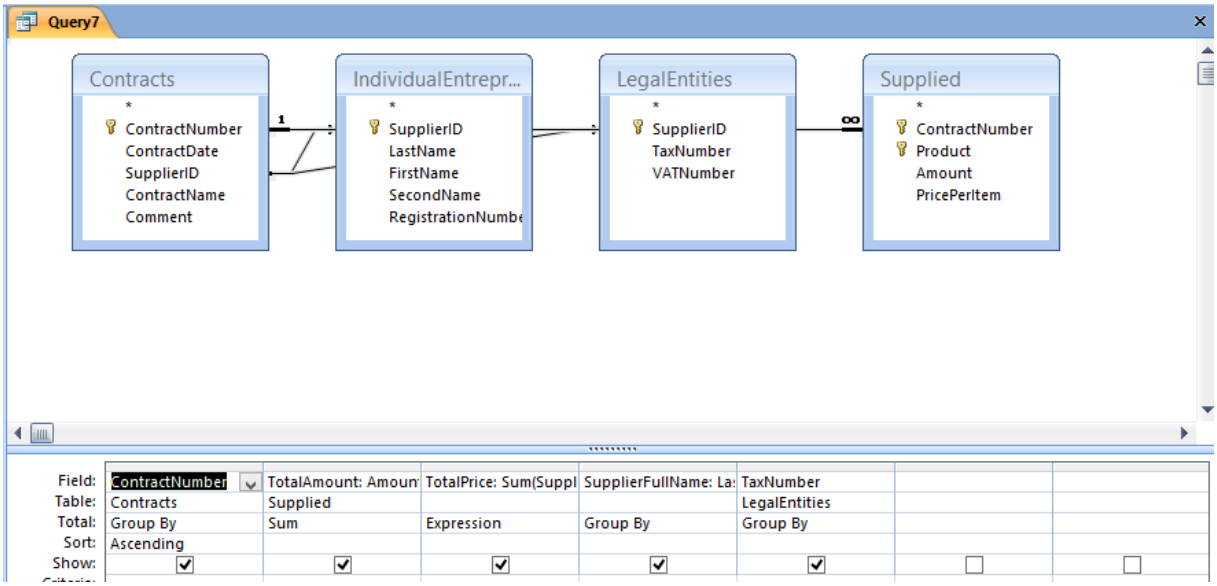



Figure 3.12

Save created query as “Query12” and close.

### 13. Define amounts of each delivered product by each supplier.

Use the following SQL command or use the query designer (figure 3.13):

```
TRANSFORM Sum(Supplied.Amount) AS [TotalAmount]
SELECT
      Suppliers.SupplierName
FROM
      Suppliers INNER JOIN
```

```
            (Contracts INNER JOIN Supplied ON Contracts.ContractNumber =
            Supplied.ContractNumber)
            ON Suppliers.SupplierID = Contracts.SupplierID
GROUP BY Suppliers.SupplierName
PIVOT Supplied.Product
```

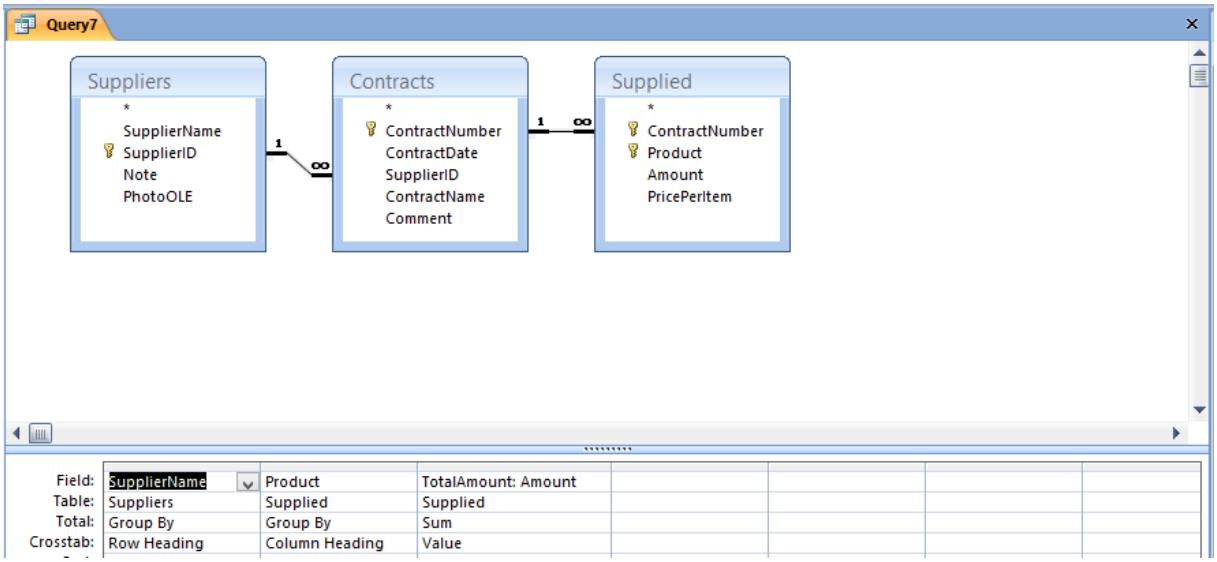



Figure 3.13

Save created query as “Query13” and close.

**14. Print a list of contracts (number, date) and total price for each contract. The list should be sorted by total price for each contract. Exclude records for which the contract number is greater than a given value from the query result.**

Use the following SQL command or use the query designer (figure 3.14):

```
PARAMETERS NumParam Short;
SELECT
      Contracts.ContractNumber,
      Contracts.ContractDate,
      Sum(Supplied.Amount) AS [TotalAmount],
      Sum(Supplied.Amount * Supplied.PricePerItem) AS [TotalPrice]
FROM
      Contracts INNER JOIN
      Supplied ON Contracts.ContractNumber = Supplied.ContractNumber
```

```
GROUP BY Contracts.ContractNumber, Contracts.ContractDate
HAVING Contracts.ContractNumber <= NumParam
ORDER BY Contracts.ContractNumber
```

This query is quite similar to the query 5. But in this query the contract number value that is used as criteria is not defined in the query. Instead the contract number is requested from a user when query is executes. This variable is called query parameter.

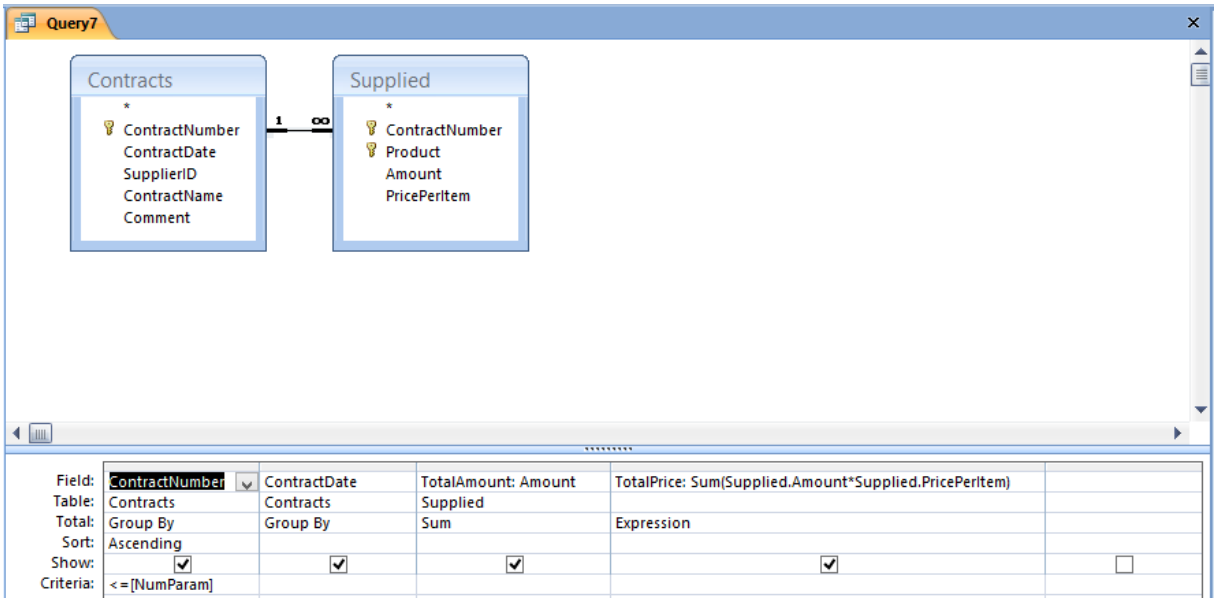



Figure 3.14

Save created query as “Query14” and close.

**15. Create a list of products delivered by the suppliers 1 and 2 (“Interfruit” LLC).**

Use the following SQL command (figure 3.15):

```
SELECT
      Product,
      Amount,
      PricePerItem,
      (Amount * PricePerItem) AS [TotalPrice]
FROM
```

```
      Contracts, Suppliers, Supplied
WHERE
      Contracts.ContractNumber = Supplied.ContractNumber AND
      Contracts.SupplierID = Suppliers.SupplierID AND
      Suppliers.SupplierID = 1
UNION
SELECT
      Product,
      Amount,
      PricePerItem,
      (Amount * PricePerItem) AS [TotalPrice]
FROM
      Contracts, Suppliers, Supplied
WHERE
      Contracts.ContractNumber = Supplied.ContractNumber AND
      Contracts.SupplierID = Suppliers.SupplierID AND
      Suppliers.SupplierID = 2
```

This query demonstrates usage of the UNION command that is used to merge results of several queries. It is not necessary to use UNION in order to solve considered task.


Figure 3.15

Save created query as “Query15” and close.

### 16. Create a list of products supplied more than once.

Use the following SQL command or use the query designer (figure 3.16):

*Option 1*

```
SELECT
	Supplied.Product,
	Count(Supplied.Product) AS [CountProducts]
FROM
	Supplied
GROUP BY Supplied.Product
HAVING Supplied.Product IN
	(SELECT
		Product
	FROM
		Supplied
	GROUP BY Product
	HAVING Count(Product) > 1)
```

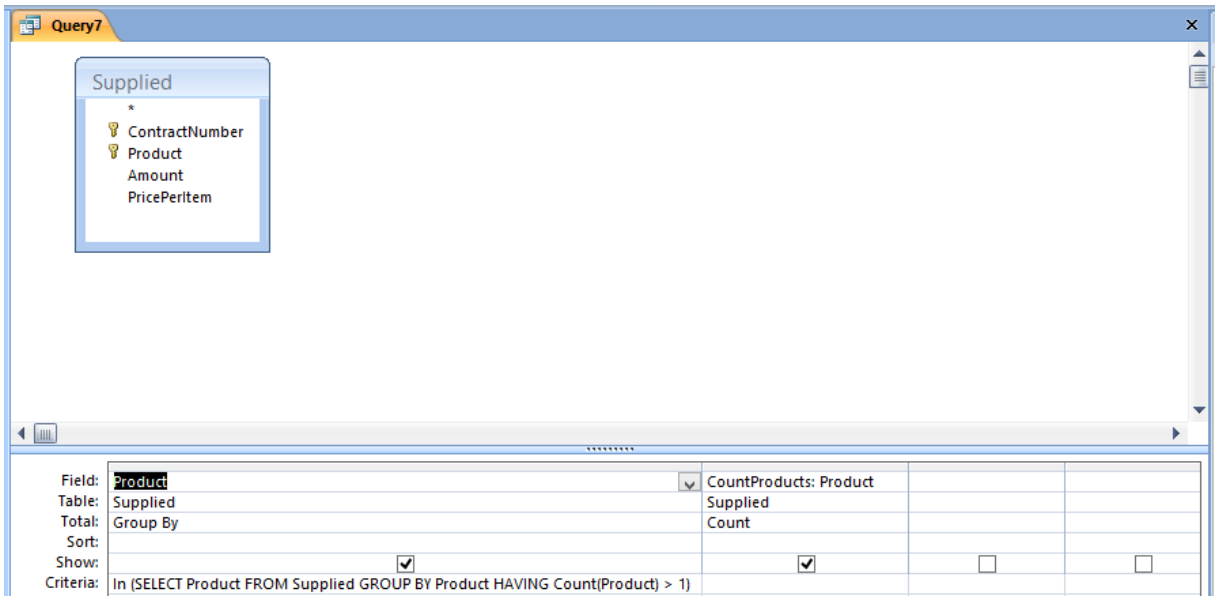



Figure 3.16a

Save created query as “Query16a” and close.

*Option 2*

```
SELECT
	Supplied.Product,
```

```
      Count(Supplied.Product) AS [CountProduct]
FROM
      Supplied
GROUP BY Supplied.Product
HAVING Count(Product) > 1
```

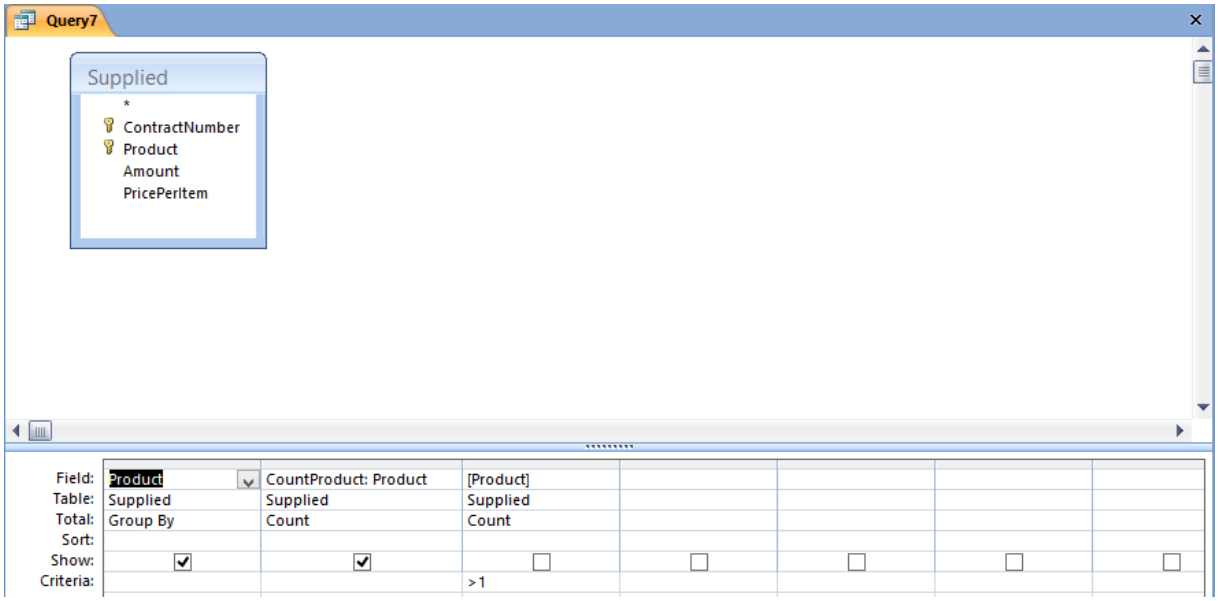



Figure 3.16b

Save created query as “Query16b” and close.

## II. Data manipulation using SQL commands UPDATE and DELETE.

### 17. Increase amount of each product delivered by the supplier 1 by 10.

Use the following SQL command or use the query designer (figure 3.17):

```
UPDATE
      Supplied
SET
      Supplied.Amount = Supplied.Amount + 10
WHERE
Supplied.ContractNumber IN
      (SELECT
            ContractNumber
```

```
FROM
	Contracts
WHERE
	SupplierID = 1)
```

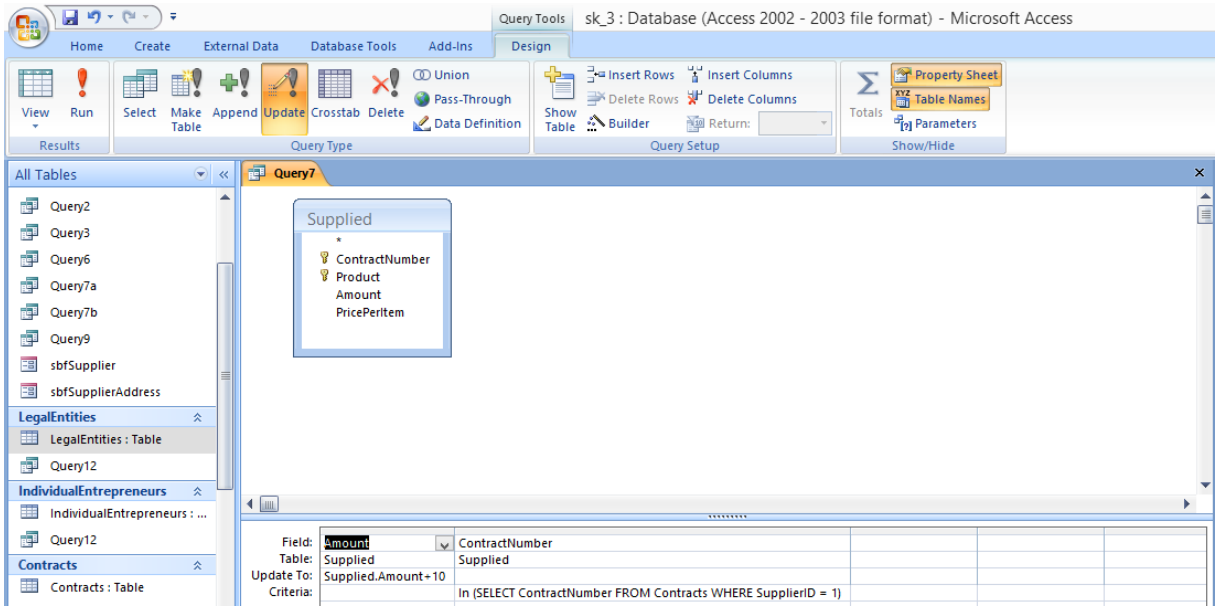



Figure 3.17

Save created query as “Query17” and close.

**18. Delete all “empty” contracts (with no records about supplied products).**

1) Create new record in “Contracts” table:

ContractNumber 8

ContractDate 7/27/2002

SupplierID 3

2) Use the following SQL command or use the query designer (figure 3.18):

```
DELETE
	Contracts.ContractNumber
FROM
	Contracts
WHERE
```

```
Contracts.ContractNumber NOT IN (SELECT ContractNumber FROM Supplied)
```

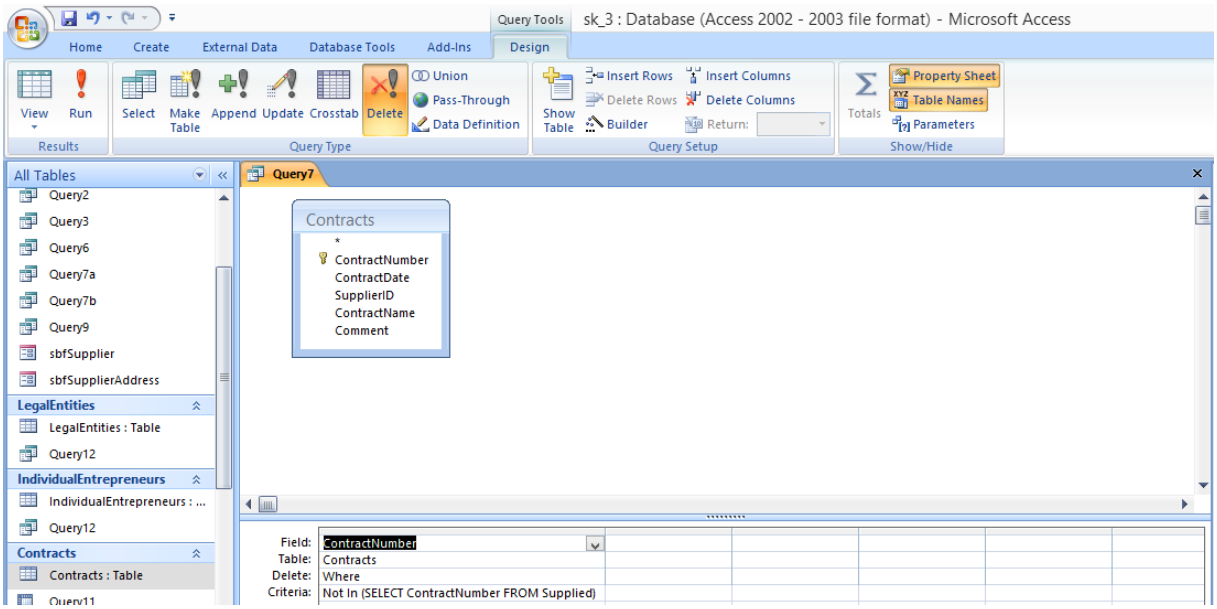



Figure 3.18

Save created query as “Query18” and close.

**III. Finish the work. Save SK.mdb file.**

**Report requirements:**

1) Briefly describe main steps of this work;

2) Depict query code and execution result for each SQL command.

## Questions

1. SELECT SQL command. Description, purpose, and usage.
2. SELECT SQL command. Provided relational operations.
3. SELECT SQL command. Columns in query results. Expressions.
4. SELECT SQL command. FROM statement. Purpose and usage.
5. SELECT SQL command. WHERE statement. Purpose and usage.

6. SELECT SQL command. WHERE statement. FILTER conditions and their usage.

7. SELECT SQL command. WHERE statement. Multiple tables queries. JOIN conditions and their usage.

8. SELECT SQL command. Join of tables in multiple tables queries. INNER JOIN, LEFT JOIN, RIGHT JOIN and their features.

9. SELECT SQL command. Crosstab queries and their features.

10. SELECT SQL command. DISTINCT argument. Purpose and usage.

11. SELECT SQL command. TOP argument. Purpose and usage.

12. SELECT SQL command. Boolean operators AND, OR, NOT, and their usage.

13. SELECT SQL command. Special operators IN, BETWEEN. Purpose and usage.

14. SELECT SQL command. Aggregate functions COUNT, SUM, AVG, MAX, MIN. Purpose and usage.

15. SELECT SQL command. GROUP BY statement. Purpose and usage.

16. SELECT SQL command. ORDER BY statement. Purpose and usage.

17. SELECT SQL command. HAVING statement. Purpose and usage.

18. SELECT SQL command. Subqueries. Purpose and usage.

19. SELECT SQL command. Subqueries. Types and features.

20. SELECT SQL command. Query parameters and their usage.

21. INSERT SQL command. Description, purpose, and usage.

22. DELETE SQL command. Description, purpose, and usage.

23. UPDATE SQL command. Description, purpose, and usage.

24. How to implement query 15 without UNION operation?

## Laboratory work 4

LEARN DBMS ACCESS REPORT DEVELOPMENT TOOLS

### Prepare to work

1. Move the database file **SK.mdb** created during the previous work into another folder (e.g., D:\ACC_LAB_4).

2. Run DBMS Access.

3. Open the database created earlier.

### Follow the steps below

1. Create report to print result of the “Query1” execution for which:

1) Click “Report Design” in “Create” tab;

2) Report designer window will appear (figure 4.1);

3) Select “Query1” as the record source in property sheet tab “Data”;

4) Create report layout using “Label”, “Text Box”, and “Line” controls as it is shown in figure 4.1;

5) Check created report by switching to the ‘Report View” mode;

6) Save report as “Report1” and close.

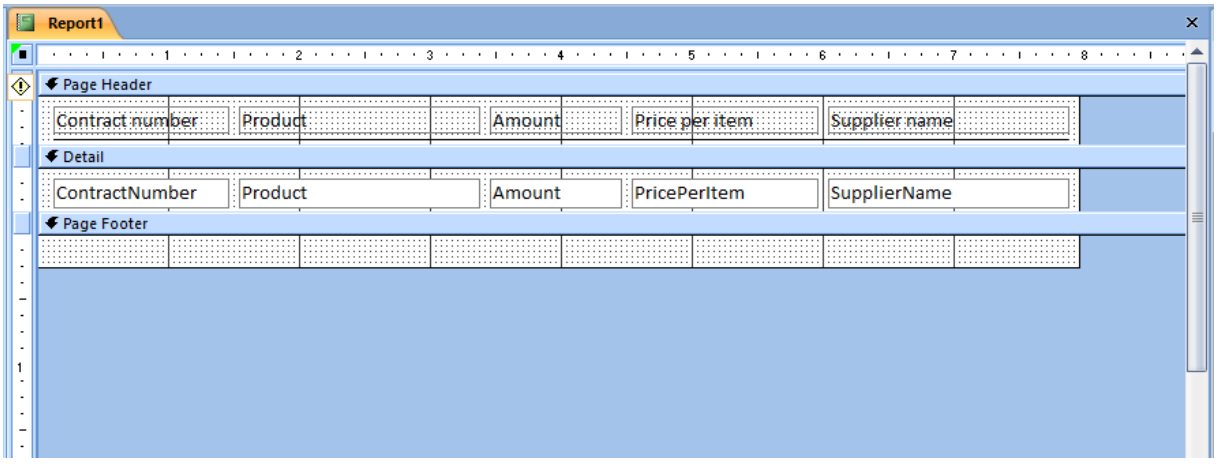



Figure 4.1

2. Create report to print result of the “Query2” execution for which:

1) Click “Report Design” in “Create” tab;

2) Report designer window will appear (figure 4.2);

3) Select “Query2” as the record source in property sheet tab “Data”;

4) Create report layout using “Label”, “Text Box”, and “Line” controls as it is shown in figure 4.2;

5) Check created report by switching to the ‘Report View” mode;

6) Save report as “Report2” and close.

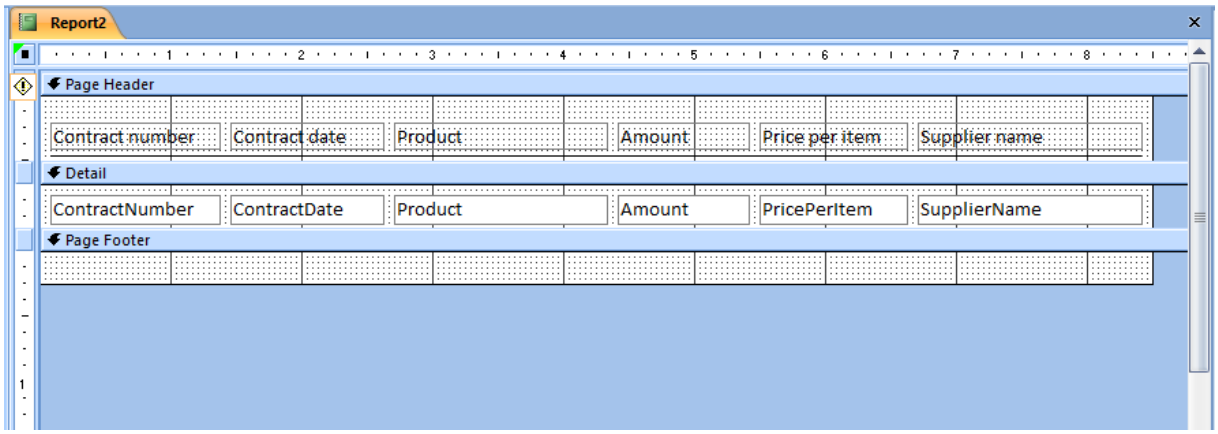



Figure 4.2

3. Create report to print result of the “Query3” execution. Report should provide display of total data for the “Amount” and “Price” columns. Follow the steps below to implement this report:

1) Click “Report Design” in “Create” tab;

2) Report designer window will appear (figure 4.2);

3) Select “Query2” as the record source in property sheet tab “Data”;

4) Enable report header and footer by clicking “Report Header/Footer” in the “Report Design Tools/Arrange” tab;

5) Create report layout using “Label”, “Text Box”, and “Line” controls to display query fields “Product”, “SupplierName”, “Amount”, and “PricePerItem” as it is shown in figure 4.3;

6) Use the “Text Box” control to create object that provides calculation of total cost for each row. Enter “=[Amount]*[PricePerItem]” expression into the text box. Also create the label “Total cost” for this field or use the label that created automatically with the text box (figure 4.3). It was not necessary to create the field with expression. The query field “Total” might be use instead;

7) Use the “Text Box” control to create object that provides calculation of total amount of the supplied products. Use “=Sum([Amount])” expression to calculate this value. Place the control into the “Report Footer” section and create the “Total” label as well;

8) Create the control to calculate the sum of total costs for each row in the same manner. Use “=Sum([Amount]*[PricePerItem])” expression;

9) Check created report by switching to the ‘Report View” mode;

10) Save report as “Report3” and close.

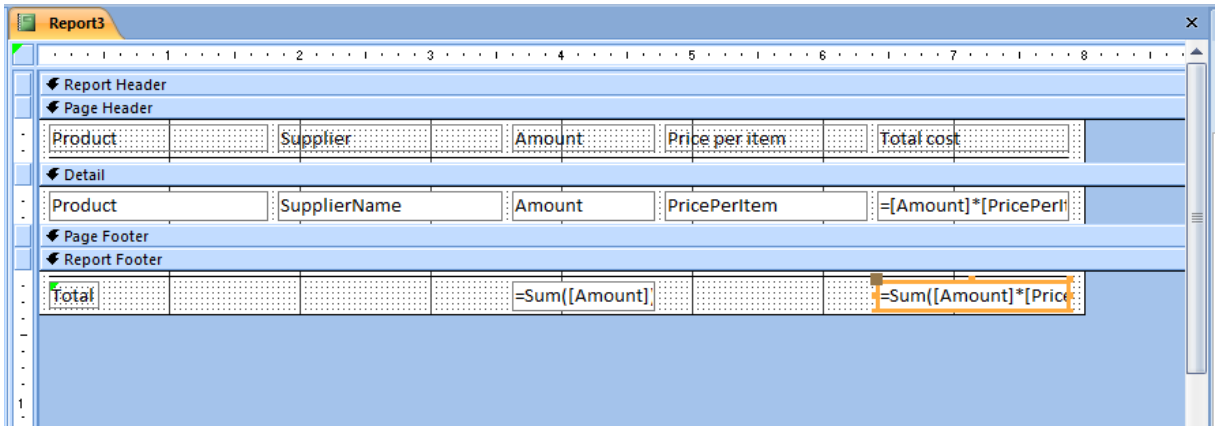



Figure 4.3

4. Create reports for the queries 4 – 16 in the same manner. These reports should contain all the fields displayed in the considered reports. Close DBMS Access after reports will be created and save the **SK.mdb** file.

**Report requirements:**

1) Briefly describe basic steps of this task;

2) Depict structure of each created report and describe purpose of used objects;

3) Show results obtained from the created reports.

## Questions

1. List basic objects used to create reports. Briefly describe each object.
2. Purpose and basic features of the “Label” object.
3. Purpose and basic features of the “Text Box” object.
4. Purpose and basic features of the “Line” object.
5. Purpose and basic features of the “Rectangle” object.
6. How to enable/disable report header and footer?
7. How to enable/disable page header and footer?
8. Purpose and basic features of the “Report Header” section.
9. Purpose and basic features of the “Page Header” section.
10. Purpose and basic features of the “Detail” section.
11. Purpose and basic features of the “Page Footer” section.
12. Purpose and basic features of the “Report Footer” section.
13. Usage of grid while creating report. How to enable/disable the grid?
14. How to define data source for a report?
15. How to change data source for a report?
16. How to enable/disable a list of available fields?
17. How to switch into the “Print Preview” mode?
18. What to do if report size is greater that a paper?
19. Where to define margins of a report? How to change margins?
20. How to change page orientation (portrait/landscape)?
21. How to change a number of columns on a page?
22. How to select a printer to print a report?
23. How to change a paper type to print a report?

24. How to select pages you need to print (not all report)?

25. How to select a number of copies for a report (if you need to print multiple copies of a certain report)?

## Laboratory work 5

DESIGN OF COMPLEX REPORTS USING REPORT WIZARD AND THEIR MODIFICATION USING DBMS ACCESS TOOLS

**Prepare to work**

1. Move the database file **SK.mdb** created during the previous work into another folder (e.g., D:\ACC_LAB_5).

2. Run DBMS Access.

3. Open the database created earlier.

**Follow the steps below**

### I. Create report using “Report Wizard”

1. Open “Report Wizard” in the “Create” tab and select “Supplied” table.

2. Select available fields “Product”, “Amount”, and “PricePerItem”. Select “Suppliers” table and select available field “Supplier Name”, place it after the “Product” field.

3. Skip next step (click “Next”).

4. Select “Product” and click “>”, click “Grouping Options …” and select “Normal” option in the “Grouping intervals” control.

5. Click “Summary Options …” on the next step, pick “Sum” option for the “Amount” field and click “OK”.

6. Select “Stepped” layout on the fifth step, click “Next” and select “Median” style.

7. Enter the name “**reportProductList**” and click “Finish”. Preview created report (figure 5.1).

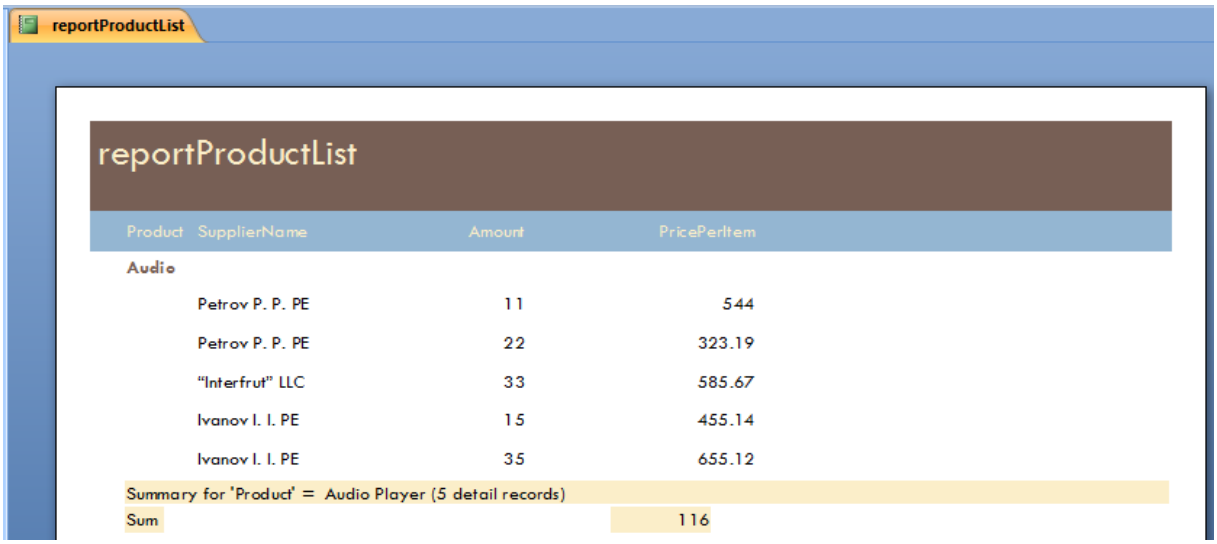



Figure 5.1

**II. Modify the report created using "Report Wizard"**

1. Open "Design View" of the report created on previous step, and open property sheet of the report (figure 5.2).

2. Enter "**Available Products**" into "Caption" field on the "Format" tab of the property sheet. Do the same for the label in the "Report Header" section. Set Arial Narrow bold font, 18 pt, and shrink "Report Header" section to save some space (figure 5.3). Save changes and preview the report.

3. Change labels in the "Page Header" section (figure 5.3). Select all labels and set bold font, 11 pt., "Align Text Left". Click "Arrange" tab and enable "To Widest" to setup size of the labels.

4. Set 10 pt. font and "To Widest" size for the text boxes in sections "Product Header" and "Detail". Set "Align Text Left" for the "Product" and "SupplierName" text boxes and "Align Text Right" for the "Amount" and "PricePerItem" text boxes (figure 5.3).

5. Set bold Arial Narrow font, 10 pt for labels and text boxes in the "Product Footer" section. Set left align for labels and right align for text boxes respectively. Change "Sum" label to "Total".

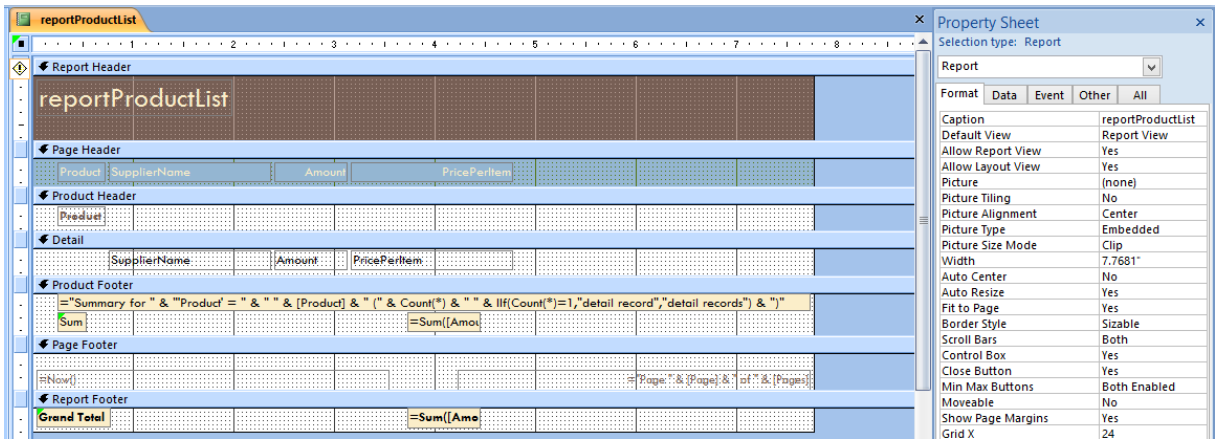



Figure 5.2

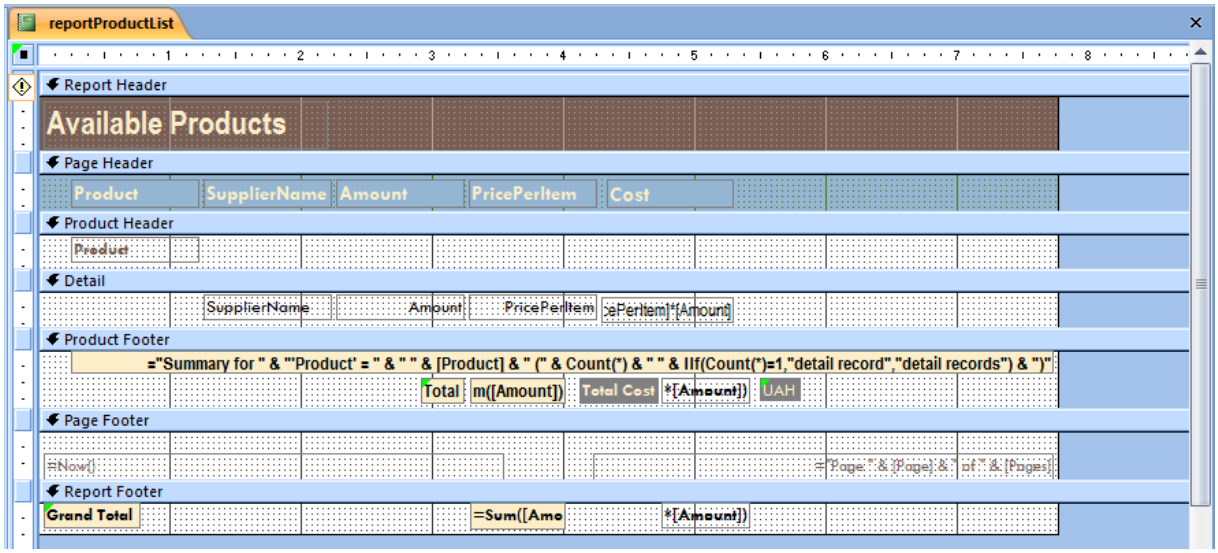



Figure 5.3

6. Add calculated field “Cost”:

1) Create label “Cost” in the “Page Header” section on the right of the “PricePerItem” label. Set bold font, 12 pt., left align;

2) Add text box to the “Detail” section under the “Cost” label;

3) Enter the “=[PricePerItem]*[Amount]” expression to the added text box. Set “Fixed” format, 2 decimal places, Arial Narrow font, 10 pt., “Align Text Right”;

4) Add text box and assigned label to the “Product Footer” section. Enter the “=Sum([PricePerItem]*[Amount])” to the text box. Set fixed format, 2 dec-

imal places. Set label text “Total Cost”, bold font, 10 pt., alight to right. Add “UAH” label to the right of the text box;

5) Repeat previous step to create the text field in the “Report Footer” section in order to calculate total cost of all products. Insert following “=Sum([PricePerItem]*[Amount])” expression to this field. Format this field in the same way as other controls.

7. Modified report is shown in figures 5.3 (designer mode) and 5.5 (preview mode). Check all report pages.

8. Resize margins or report width if preview shows blank pages after each report page. Use “Page Setup” tab controls to resize report/margins. Shrink top and bottom margins to place a lot of data on the report.

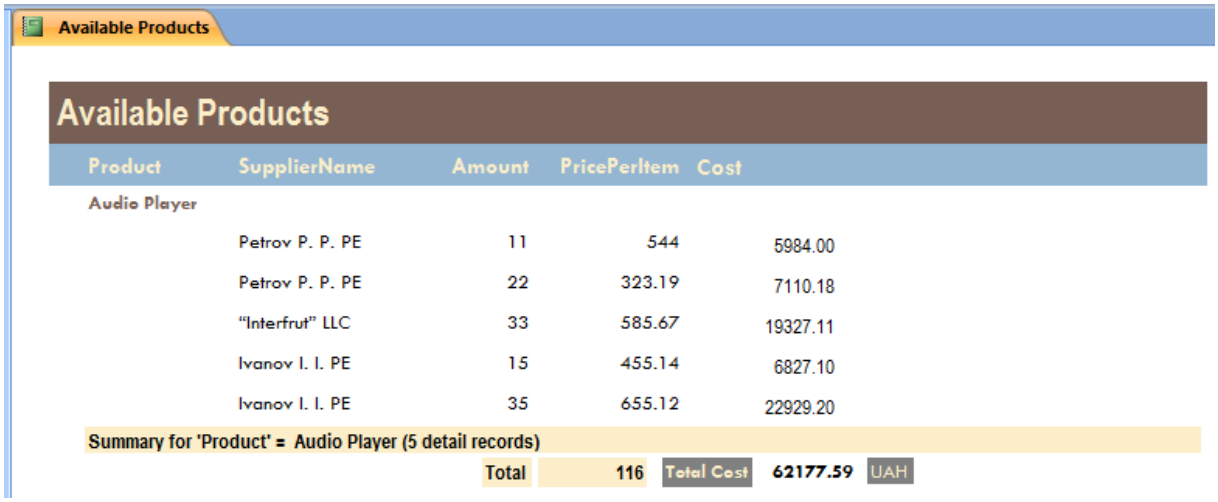



Figure 5.4

9. Use the group property “keep whole group together on one page” if necessary:

1) Switch to the “Design View” and click “Group & Sort”;

2) Select the “Product” field in the “Group on …” option (it is already selected);

3) Click “More >” and select “keep whole group together on one page”;

4) Close “Group & Sort” window, switch to the “Print Preview” mode and check report layout.

### III. Insert subreport

You can add subreports into reports as well as into forms. Unlike the form wizard, report wizard does not allow to create report with subreport. It is necessary to create report using wizard or designer and then insert it into another report.

Create report to display data about total product amounts in weekly supplies as the example. This report serves for two purposes: primary report (to estimate supplies) and subreport inside another report. Therefore, the report “**repWeeklyProductSupplies**” will be added as the subreport into the report “**reportProductList**”. Follow the steps to implement this task:

1. Create new query “query1999ProductSuppliesSQ” using the following query or query designer (figure 5.5):

```
TRANSFORM Sum(Supplied.Amount) AS TotalAmount
SELECT
      Supplied.Product
FROM
      Contracts INNER JOIN
      Supplied ON Contracts.ContractNumber = Supplied.ContractNumber
WHERE Contracts.ContractDate Like "*/*/1999"
GROUP BY Supplied.Product
PIVOT Format(Contracts.ContractDate,"ww") In (36,37,39,40,42);
```

Run query to check its correctness (figure 5.6). Save it with the name “**query1999ProductSuppliesSQ**”.

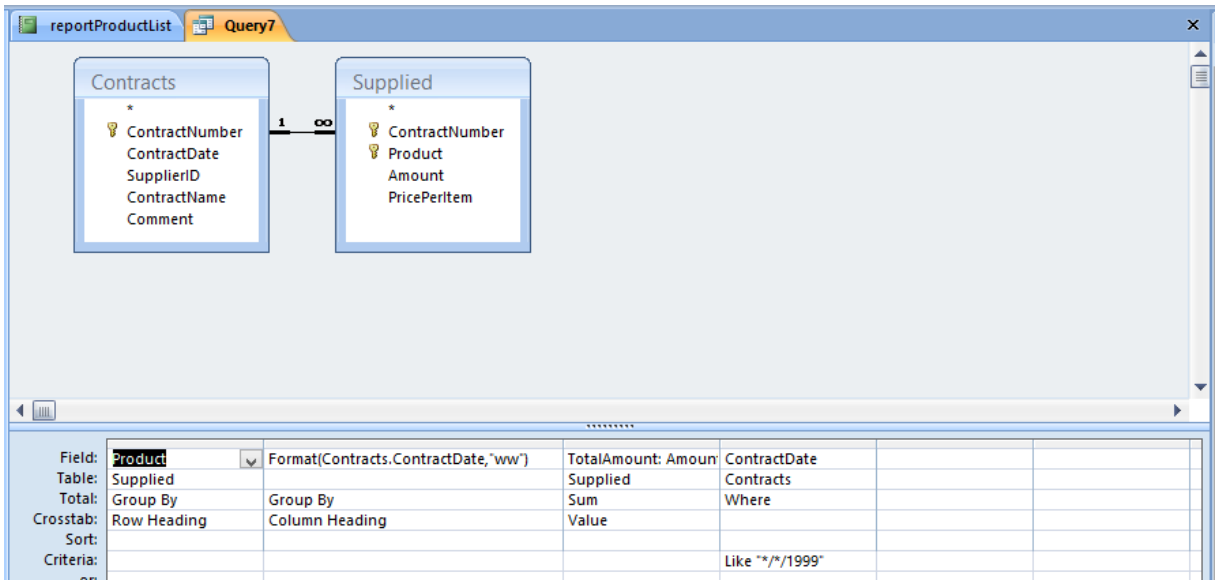
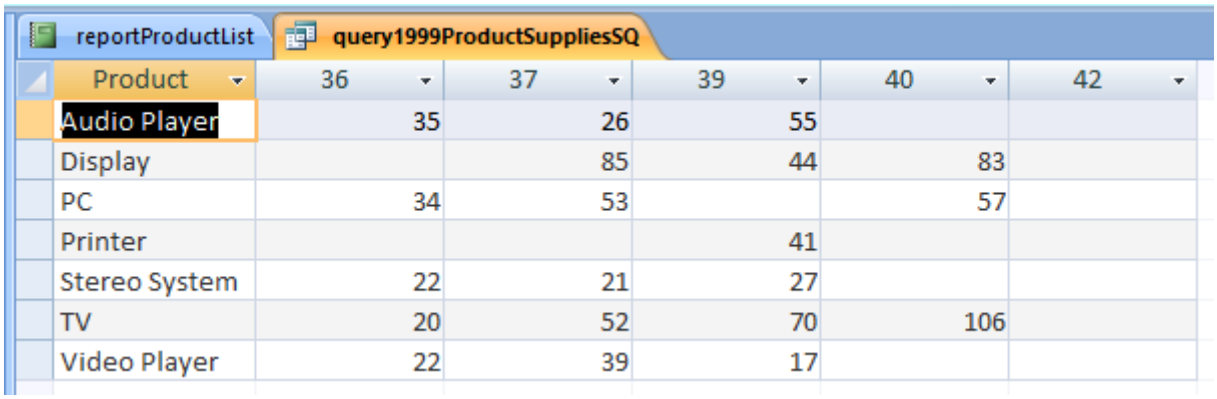



Figure 5.5




| Product | 36 | 37 | 39 | 40 | 42 |
|---|---|---|---|---|---|
| Audio Player | 35 | 26 | 55 | | |
| Display | | 85 | 44 | 83 | |
| PC | 34 | 53 | | 57 | |
| Printer | | | 41 | | |
| Stereo System | 22 | 21 | 27 | | |
| TV | 20 | 52 | 70 | 106 | |
| Video Player | 22 | 39 | 17 | | |

Figure 5.6

2. Create new report "**repWeeklyProductSupplies**":

1) Click "Report" in the 'Create" tab;

2) Access will automatically select the query "**query1999Product SuppliesSQ**" as the data source. Select "Design View";

3) Remove headers and footers. Set 5 to "Grid X" and "Grid Y" in the "Format" tab of the property sheet;

4) Increase report width to 8 inches;

5) Click "Group & Sort" and select the "Product" field using the "with A on top" option (ascending sort). Close this window;

6) Click “Add Existing Fields” and drag available fields to the “Details” section of the report. Place them as it is show in figure 5.7;

7) Add “Week” to each label that displays week number. Set bold fonts for all labels, set left align for the “Product” label and right align for the other labels;

8) Add lines above and below labels and text boxes;

9) Preview created report (figure 5.8). Save it with the name “**repWeekly ProductSupplies**” and close.

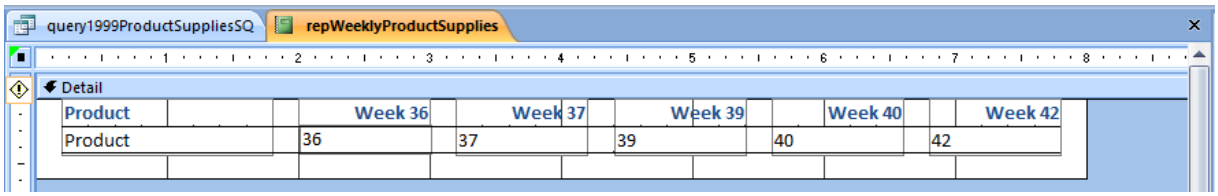



Figure 5.7

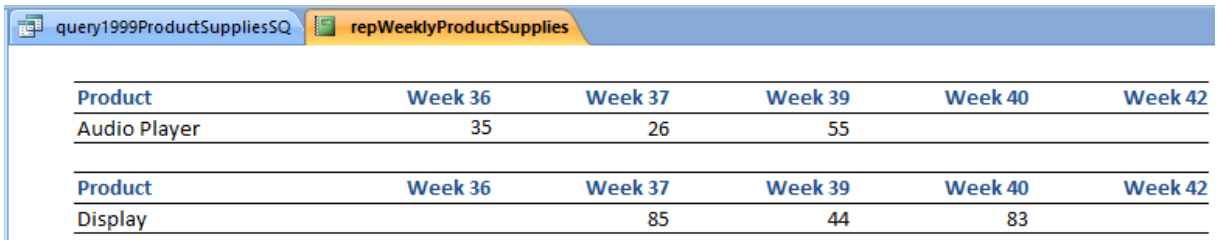



Figure 5.8

3. Add created subreport “**repWeeklyProductSupplies**” to the report “**reportProductList**” and bind theirs data:

1) Open the “**reportProductList**” in the designer mode, increase height of the “Product Footer” section;

2) Drag the report “**repWeeklyProductSupplies**” into the “Product Footer” section and place it as it is shown in figure 5.9;

3) Setup height of the “Product Footer” section to provide 1mm margins above and below the subreport;

4) Open the “Data” tab in property sheet and select the “Product” field in the “Link Master Fields” or “Link Child Fields” to bind data of the primary report and subreport;

5) Preview created report (figure 5.10), check its correctness;

6) Save this report as “**reportProductList2**” and save it.

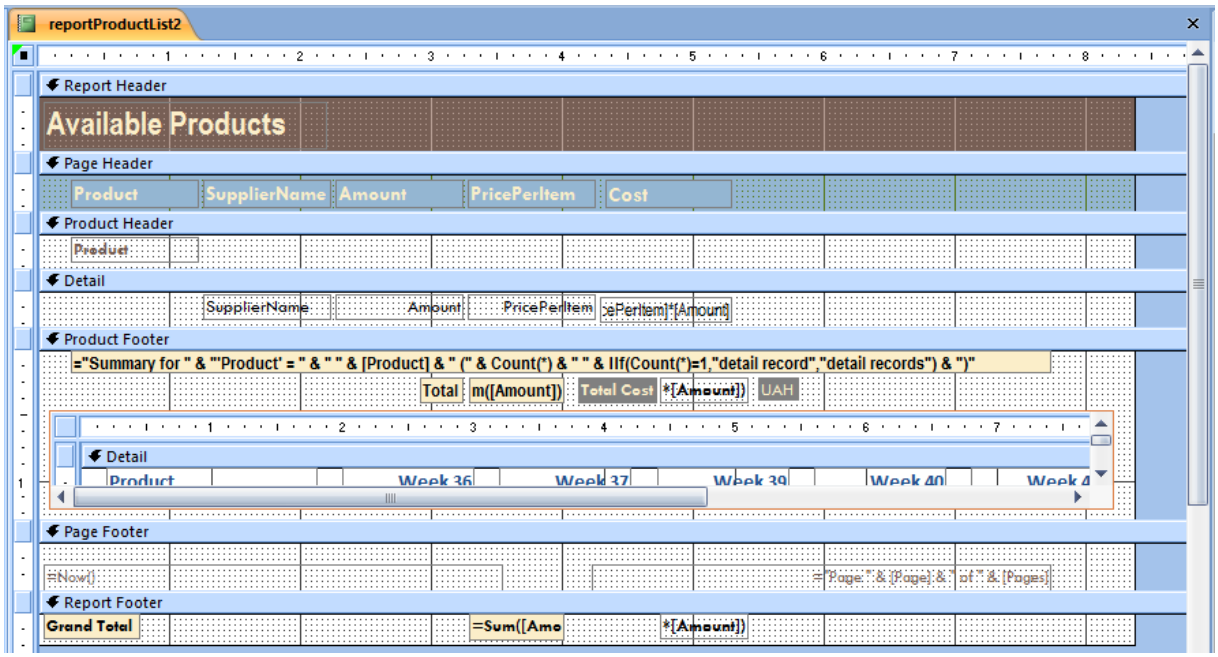



Figure 5.9

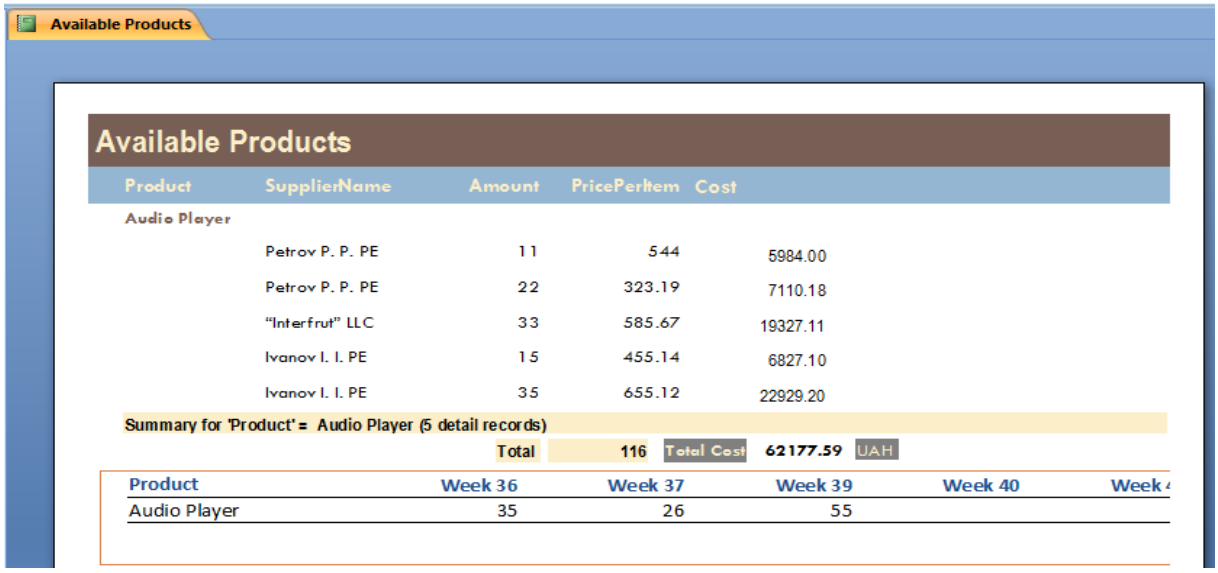



Figure 5.10

## IV. Save and print report

1. Select the report in database window and right click on it, then select “Export” and choose “PDF or XPS” option, and save report snapshot. Change its name if necessary.

2. Open saved report file in “Explorer” and check all its pages. Print it or send via email if necessary.

**Report requirements:**

1) Briefly describe main steps of this work;

2) Depict the report created using wizard;

3) Depict the modified report and describe how it was created;

4) Describe the way how to create crosstab and subreport;

5) Describe the way how to add subreport into the primary report;

6) Describe the way how to bound data between the primary report and subreport.

## Questions

1. How to change font and its size?

2. How to select all labels and set label size to fit widest text?

3. Describe the ways how to remove, replace, and edit controls.

4. How to add calculated controls into a report?

5. How to add a subreport into a report?

6. Why groups are necessary in reports?

7. How to group data in a report?

8. What is the purpose of “keep whole group together on one page” property and how to use it?

9. Describe a sequence of steps used to create, view, and print report.

10. What does the expression “LIKE */*/1999” mean?

11. How to create a crosstab?

12. How to define column headings of a crosstab?

13. How to define row headings of a crosstab?

14. What is the “Value” in a crosstab? How to define it?

15. What is the aggregate function? What aggregate functions are used to create a crosstab?

16. How to define filter conditions in a crosstab?

17. How to define fixed column headings in a crosstab?

18. What does the expression Format([ContractDate], “ww”) mean?

19. For which tasks the function Format() might be used?

# Laboratory work 6

INCLUDE CHARTS AND DIAGRAMS INTO REPORTS USING DBMS ACCESS TOOLS

**Prepare to work**

1. Move the database file **SK.mdb** created during the previous work into another folder (e.g., D:\ACC_LAB_6).

2. Run DBMS Access.

3. Open the database created earlier.

**Follow the steps below**

## I. Create a pivot chart

1. Create new report **Query1999ProductSupplyDiagram** using tables “Contracts” and “Supplied”. Add “Product”, “Amount”, and “ContractDate” fields. Setup criteria for the “ContractDate” field to select only deliveries of 1999 year using **LIKE “*/*/1999”** expression. Save this query.

2. Select created query and click “Pivot Chart” in the “Create” tab (figure 6.1).

3. Drag “ContractDate By Week” from the “Chart Field List” window to the X axis, drag “Amount” to the Y axis, and the drag “Product” to the legend.

4. Click “Legend” in the “Show/Hide” section to display legend with the list of supplied product names. Right click on the “Years” rectangle in the bottom of form and then click the “Expand” option in order to show detailed data by each week on the chart (figure 6.2).

5. Save created form as **Form1999ProductSupplyDiagram**.


Figure 6.1

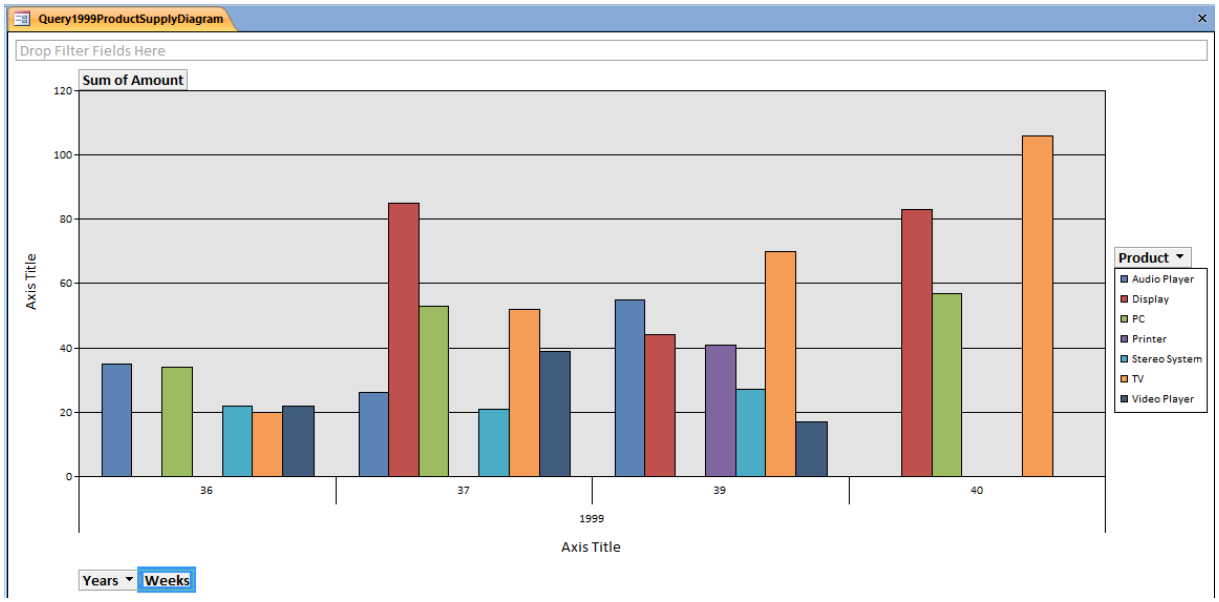



Figure 6.2

## II. Setup a diagram structure

1. Select X axis title and open property sheet. Open “Format” tab and type “Week ‘99” into the “Caption” field.

2. Repeat the same actions to change Y axis caption to “Products Supplied” (figure 6.3).

3. Open property sheet for the whole form and click “Add Title” in the “General” tab.

4. Open property sheet for the appeared title, open “Format” tab and type “Weekly Product Supplies in 1999” (figure 6.3).

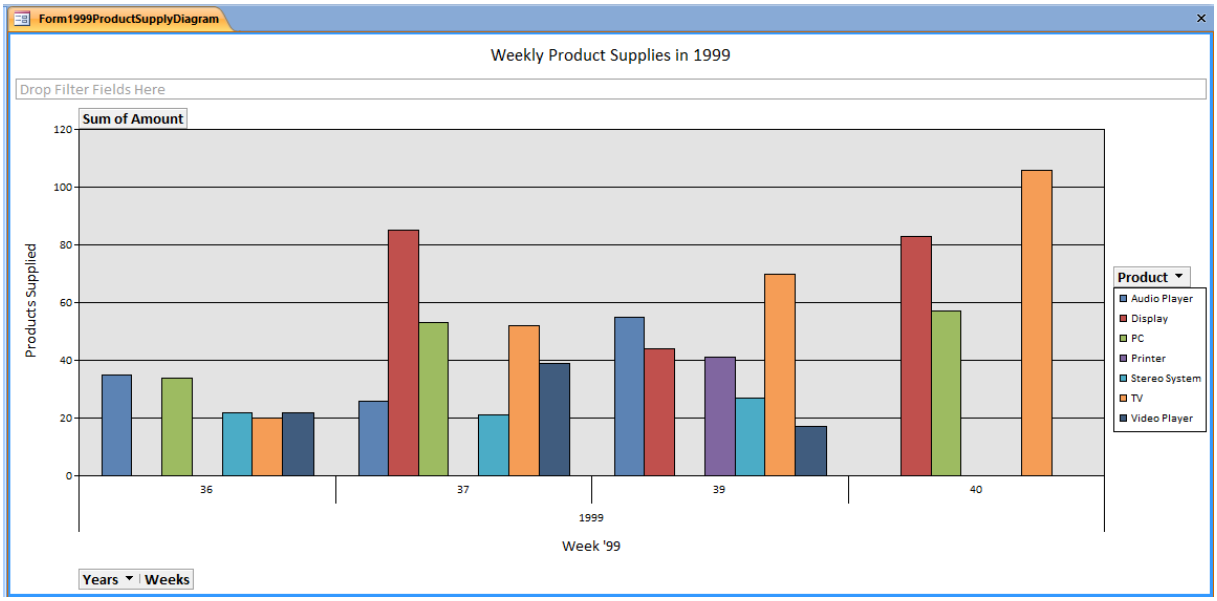



Figure 6.3

### III. Change chart type

You can change chart type if it is necessary. To change chart type follow the steps below:

1. Open form with diagram and select Pivot Chart View mode.

2. Click “Change Chart Type” in the “Type” section.

3. Select another chart type, i.e. “Line”, and then pick “Line with markers displayed at each data value” (figure 6.4).

4. To present query results using a stacked column chart, select the appropriate chart type using the “Type” tab in “Properties” window (figure 6.5).

Any type of chart depends on the specific needs might be selected using the property sheet tab “Type”.

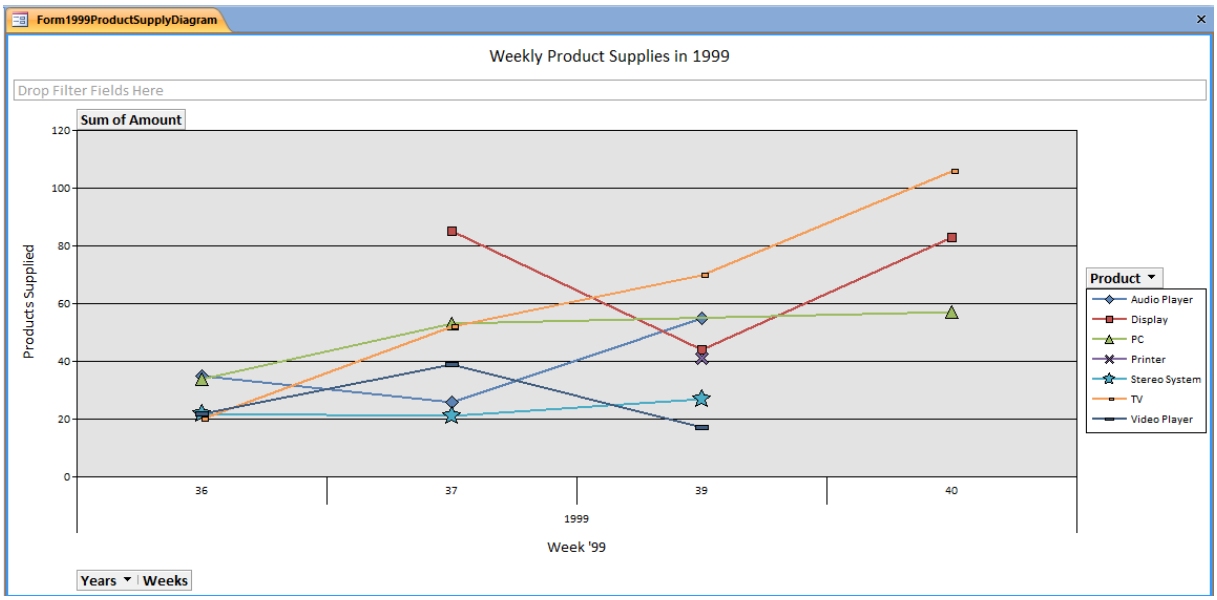



Figure 6.4

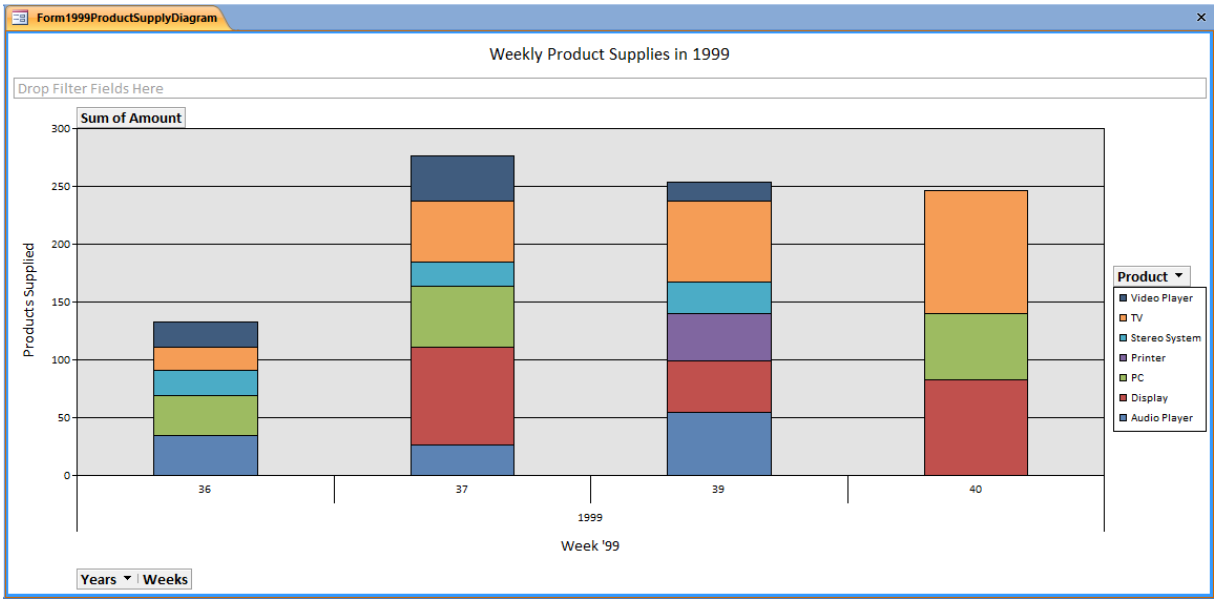



Figure 6.5

## IV. Print charts in reports

1. Open the **reportProductList** report created in previous laboratory work using the design mode.

2. Drag the **Form1999ProductSupplyDiagram** form to the "Report Footer" section. Remove assigned label and change its size (figure 6.6).

3. Save changes and switch to the “Print Preview” mode to verify changes (figure 6.7).

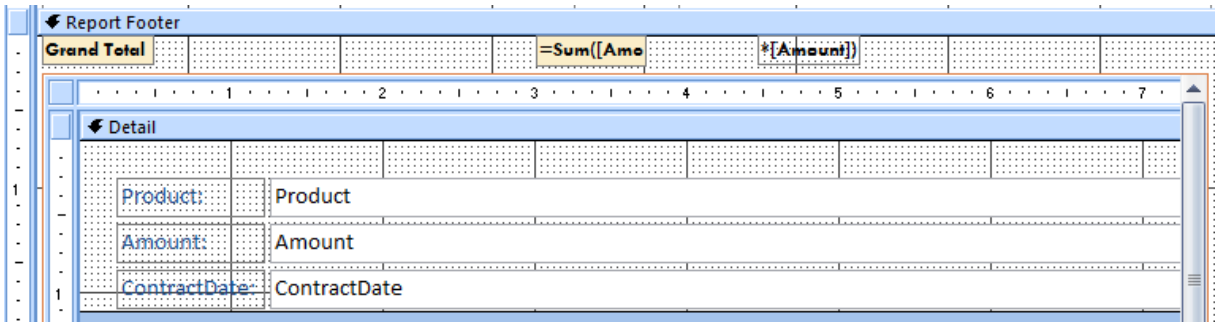



Figure 6.6

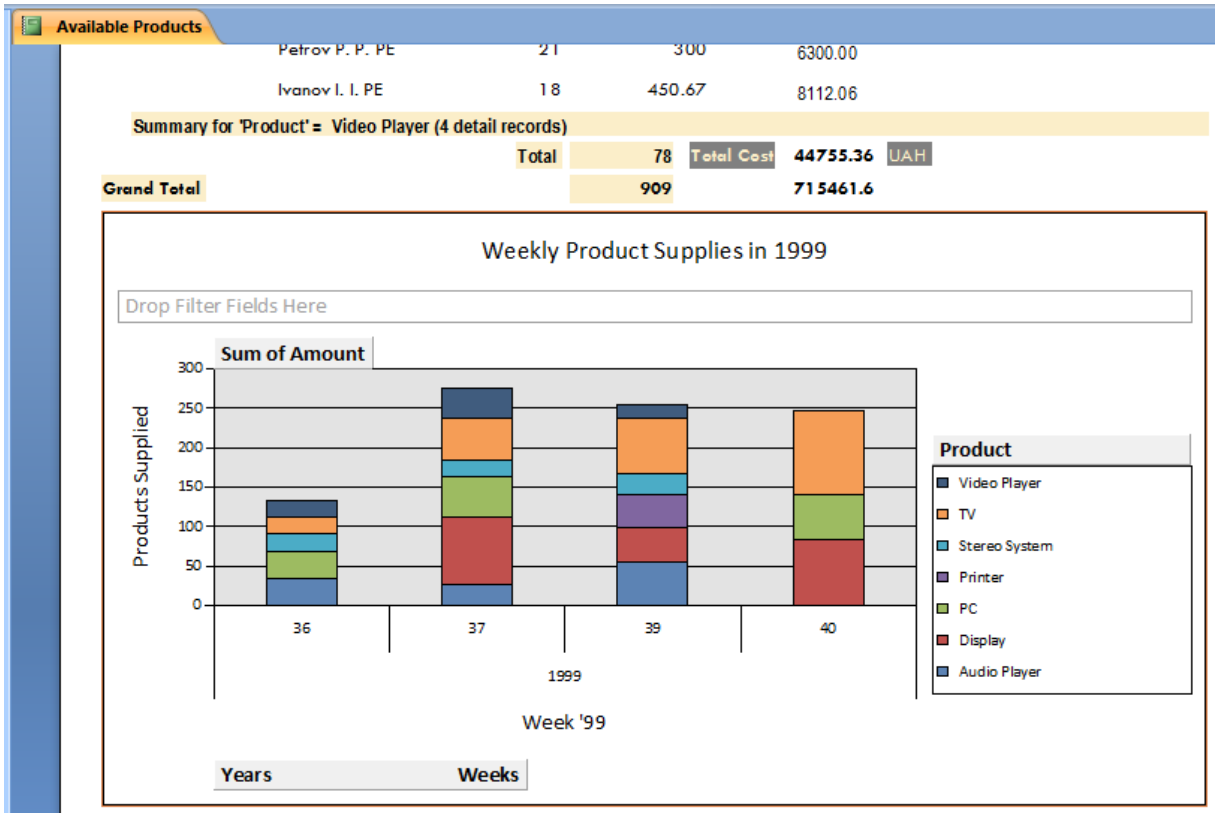



Figure 6.7

## V. Create a bonded chart based on a crosstab

First of all, to create bonded chart, we need to create crosstab that will be used as the data source for the chart. Required steps are outlined below.

1. Design a crosstab for a chart.

To create new query **query1999ProductSupplyDiagramCT** using the **Query1999ProductSupplyDiagram** query, the following actions are required:

1) Create new query using design mode, select the earlier created **Query1999ProductSupplyDiagram** query instead of table(s);

2) Click “Crosstab” in the “Query Type” section;

3) Select “Product” field and set it to be row heading;

4) Type **Week:Format([ContractDate],"ww")** expression into the second field and set it to be column heading;

5) Select “Amount” field, set the “Sum” operation as total, and set it to be value (figure 6.8);

6) Run created query to check correctness and save it (figure 6.9).

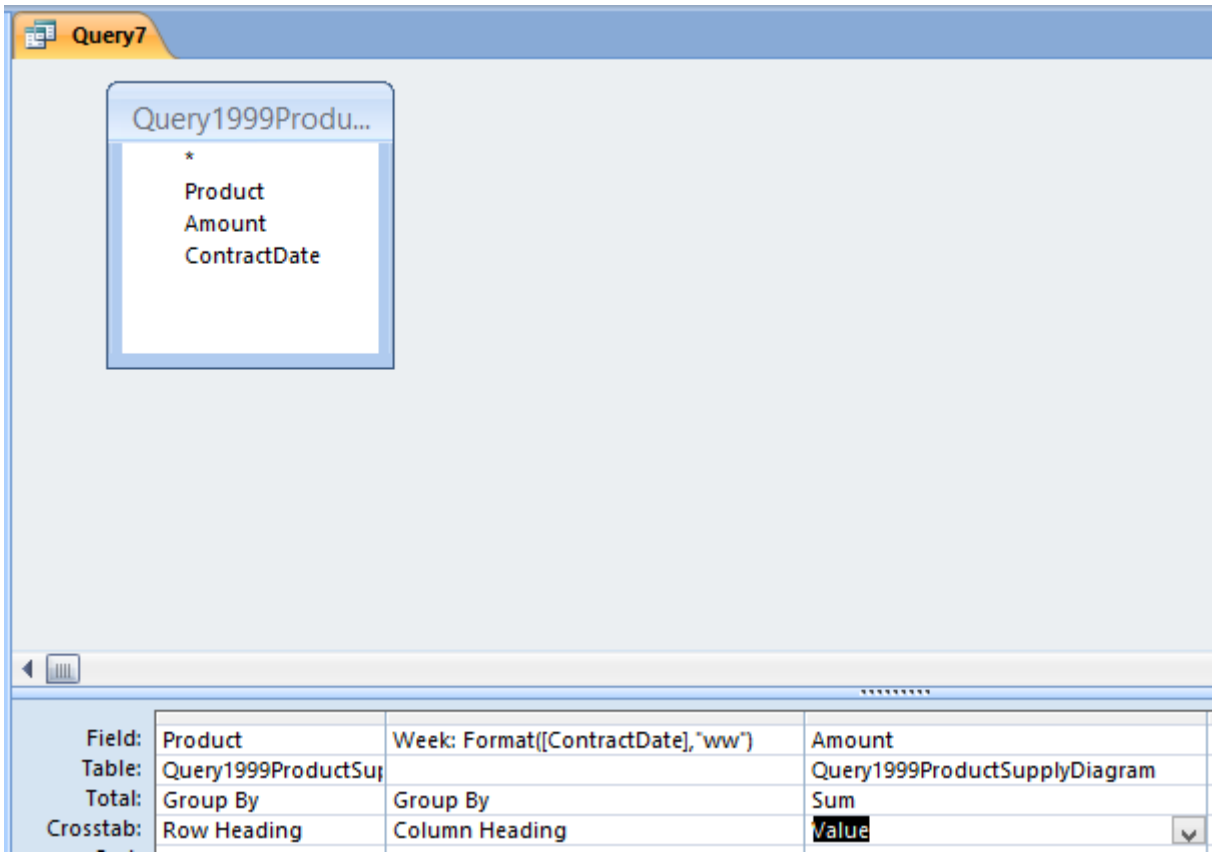



Figure 6.8

Query7

| Product | 36 | 37 | 39 | 40 |
|---|---|---|---|---|
| Audio Player | 35 | 26 | 55 | |
| Display | | 85 | 44 | 83 |
| PC | 34 | 53 | | 57 |
| Printer | | | 41 | |
| Stereo System | 22 | 21 | 27 | |
| TV | 20 | 52 | 70 | 106 |
| Video Player | 22 | 39 | 17 | |

Figure 6.9

2. Follow the steps below to assign created crosstab as the data source for the unbounded chart:

1) Create empty form and click “Insert Chart”;

2) Select created earlier query **Query1999ProductSupplyDiagram**;

3) Select all available fields, select line chart type, and setup axis and series in the same manner it was done for the **Form1999ProductSupplyDiagram** form (figure 6.10);

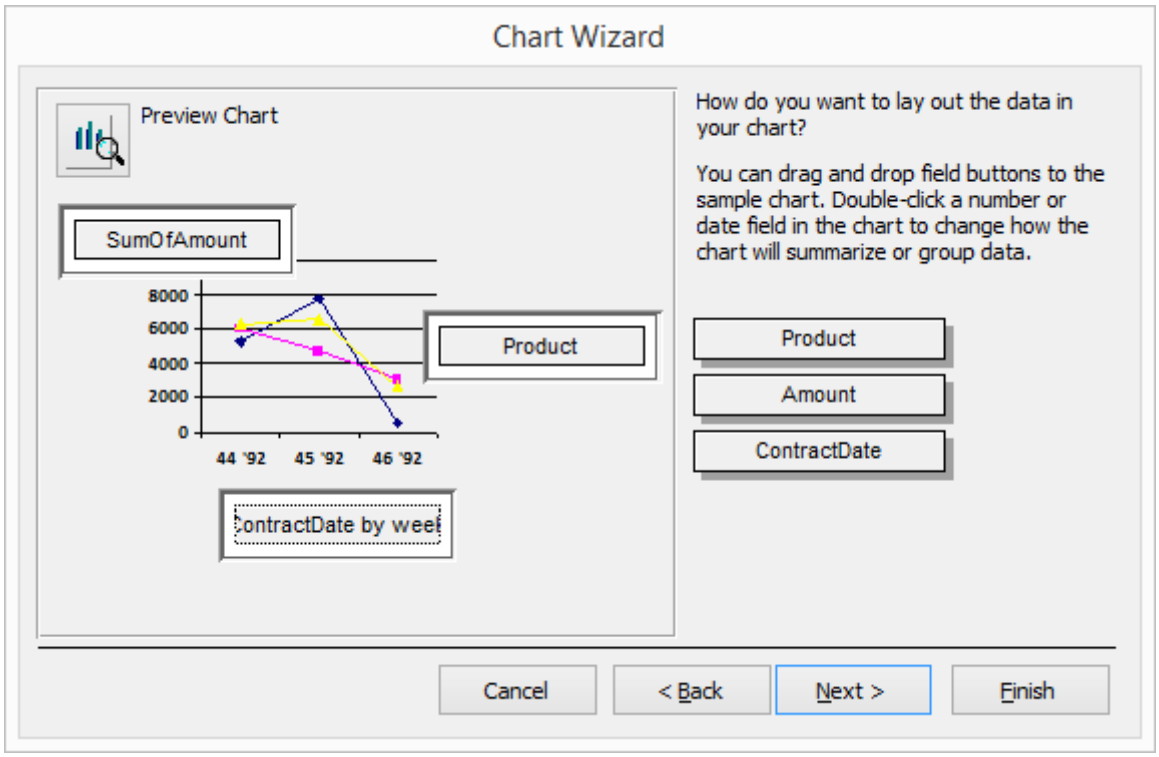



Figure 6.10

4) Save created form as **Form1999ProductSupplyDiagram2**.

**3. Bound chart to the single query or table entry.**

1) Open property sheet and select created earlier crosstab query **query1999ProductSupplyDiagramCT** as the record source;

2) Select the chart object and open property sheet tab “Data”;

3) Type ‘Product” into “Link Master Fields” and “Link Child Fields”;

4) Switch to the form view in order to verify changes (figure 6.11);

5) Double click on chart to activate it, select “Chart” menu item and click “Add Trendline”, type custom trend line name “Supplies Trend” in the “Options” tab, and click OK to accept default trend line (figure 6.11).

6) Save created form and close it.

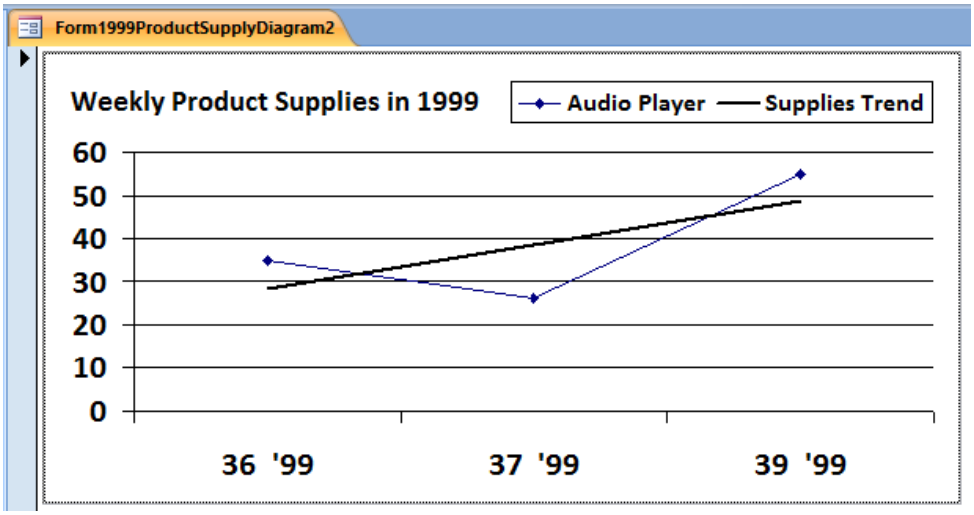



Figure 6.11

**Report requirements:**

1) Briefly describe main steps of this work;

2) Describe steps to create a pivot chart;

3) Describe steps to setup a pivot chart;

4) Describe steps to change chart type;

5) Describe how to put a chart into a report;

6) Describe how to create a bonded chart based on a crosstab.

## Questions

1. List basic chart types and their purpose.
2. How to create a pivot chart?
3. How to create a chart using the form designer?
4. How to activate a chart and change it size?
5. How to change a caption, axis titles, and legend of a chart?
6. How to change a font size for each label?
7. How to change chart type?
8. How to add a chart to a report?
9. How to create a crosstab based on a query?
10. How to change created crosstab to group data by days?
11. How to change created crosstab to group data by months?
12. How to change created crosstab to group data by years?
13. How to change created crosstab to demonstrate total price for each week?
14. How to change created crosstab to demonstrate number of deliveries for each week?
15. How to change created crosstab to demonstrate number of deliveries for each month?
16. How to change created crosstab to demonstrate number of deliveries for each year?
17. How to bond a chart with a single entry of a query or table?
18. How to add a trend line to a chart?
19. What does a trend line show?
20. How to setup a trend line before add it?

# Laboratory work 7

## MERGE ALL CREATED COMPONENTS TOGETHER AND CREATE APPLICATION. LEARN BASICS OF INFORMATION SECURITY IN DBMS ACCESS

### Prepare to work

1. Move the database file **SK.mdb** created during the previous work into another folder (e.g., D:\ACC_LAB_7).

2. Run DBMS Access.

3. Open the database created earlier.

### Follow the steps below

### I. Merge created earlier components and create application

Several user forms might be created to merge created components and create application. All of these components will be called using the certain buttons on the forms. Following steps are required to create such forms.

1. Create “Data” form. It will allow to run created earlier form “Supply information” and others.

Follow the instructions below to create “Data” form:

1) Create blank form;

2) Open property sheet for this form;

3) Disable navigation buttons, set border size to “Dialog”;

4) Place button on the created form;

5) Select “Form Operations” category and “Open Form” action;

6) Select “Supply information” form;

7) Open property sheet for the button, remove “(bitmap)” from “Picture” property and type “Supply contracts” into “Caption” property. Resize the button if necessary;

8) Create new button to close form;

9) Check out the form layout. It should conform to the layout is shown in figure 7.1;

10) Save created form with the name “Data”;

11) Run this form, verify that all buttons work correct and close the form.

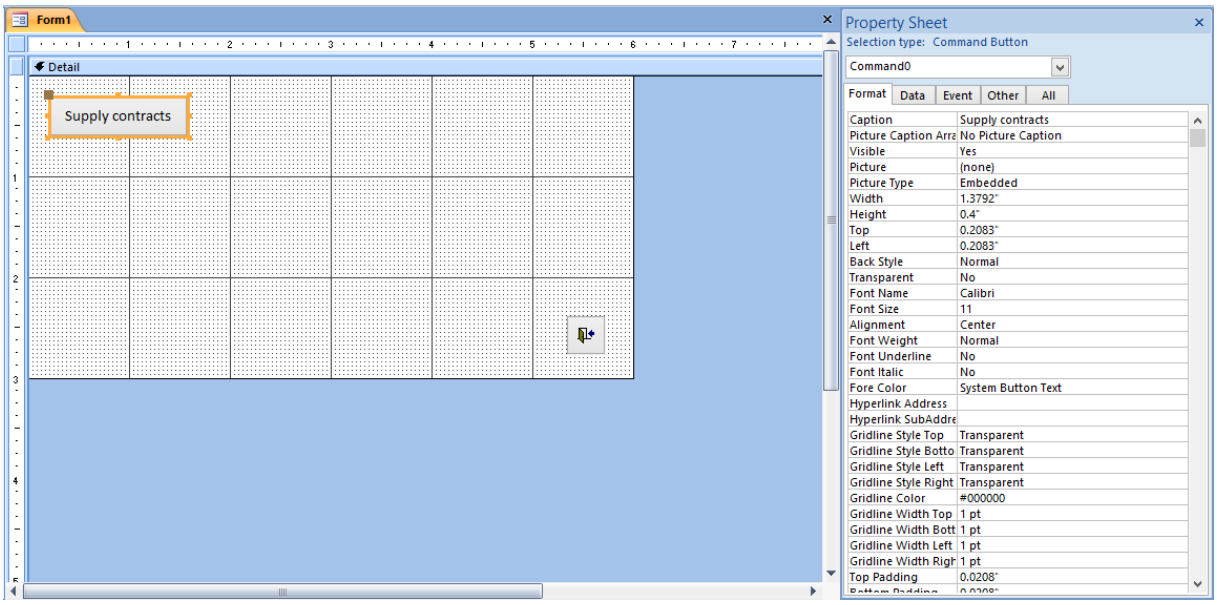



Figure 7.1

2. Create “Data processing” form.

Follow the steps below to implement this form:

1) Setup form properties according to the “Data” form;

2) Create “Exit” button;

3) Create new button to open the “Query1” query;

4) Verify correctness of the button;

5) Create new button to open the “Report1” report;

6) Verify correctness of the button;

7) Place buttons as it is shown in figure 7.2;

8) Save the form with the name “Data processing”.

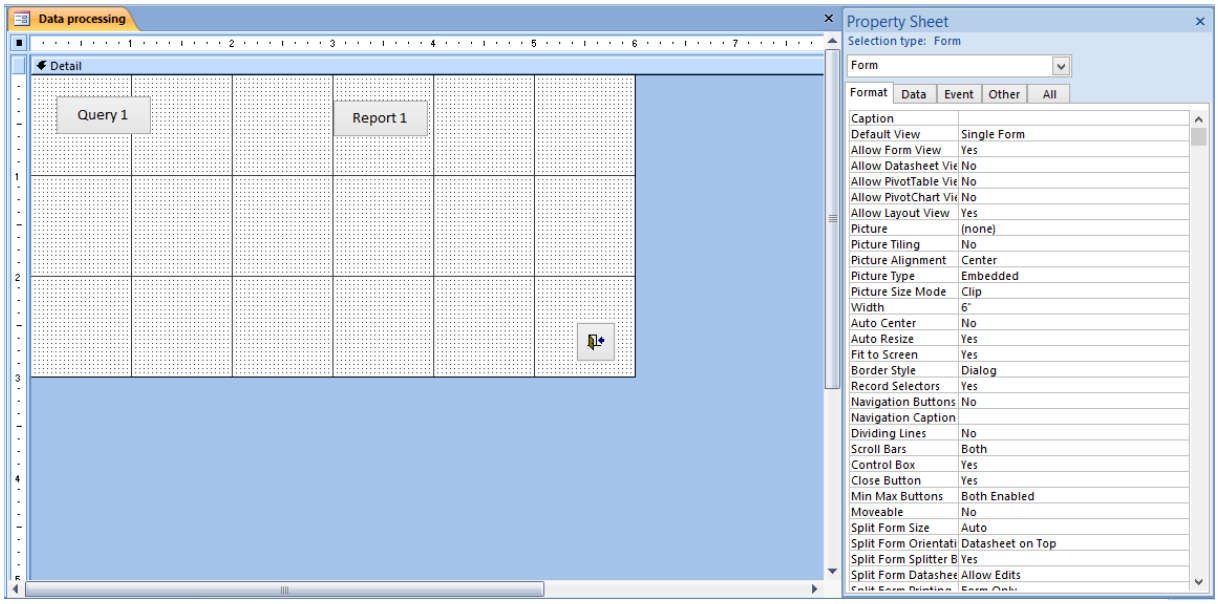



Figure 7.2

3. Create “Main” form.

Follow the steps below to implement this form:

1) Setup form properties in the same way you have done it for the previous forms;

2) Create “Quit Application” button;

3) Create new button used to open “Data” form;

4) Create new button used to open “Data processing” form;

5) Save the form using the name “Main” (figure 7.3);

6) Verify correctness of the buttons. “Stop” button should close DBMS Access.

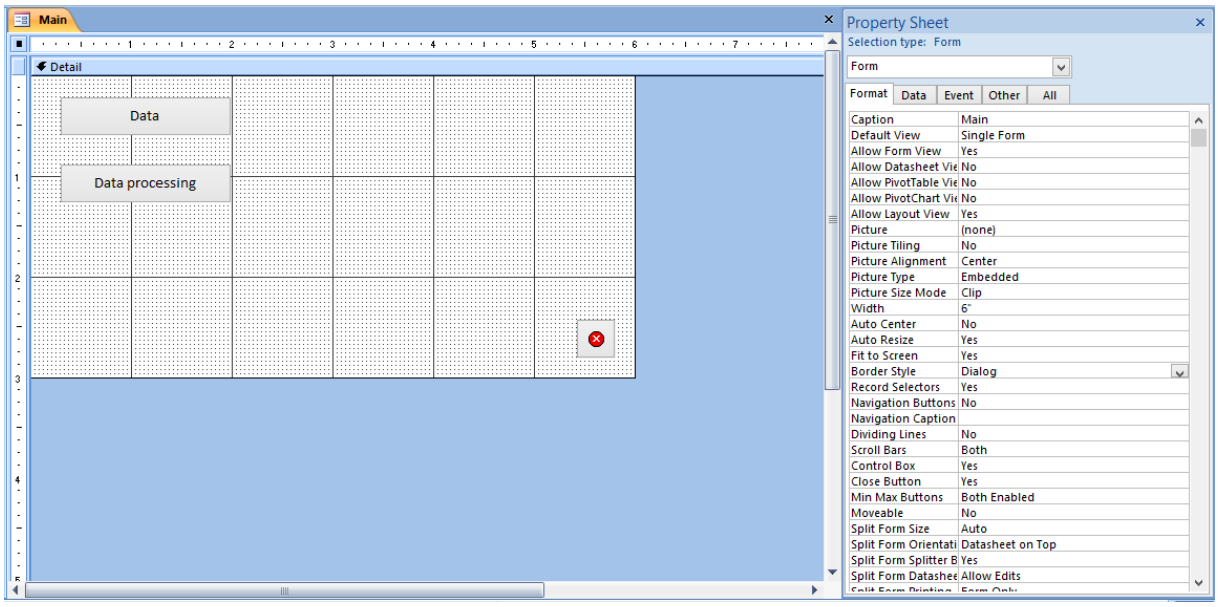



Figure 7.3

4. Setup the database to run the application automatically.

Follow the steps below to implement this setup:

1) Open “Access Options” window and select “Current Database”;

2) Type “Delivery” into “Application Title” field (figure 7.4);

3) Select “Main” form in the “Display Form” combo box (figure 7.4);

4) Disable “Display Status Bar”, “Display Navigation Pane”, and “Display Document Tabs”, select “Overlapping Windows” option (figure 7.4);

5) Save changes and restart the database. Main form will appear just after the database started.

## II. Learn basics of information security in DBMS Access.

Password setup is the one of the most common access control tools for DBMS Access database. Let’s see the sequence of actions used to enable and disable the database password.

1. Enable the database password.

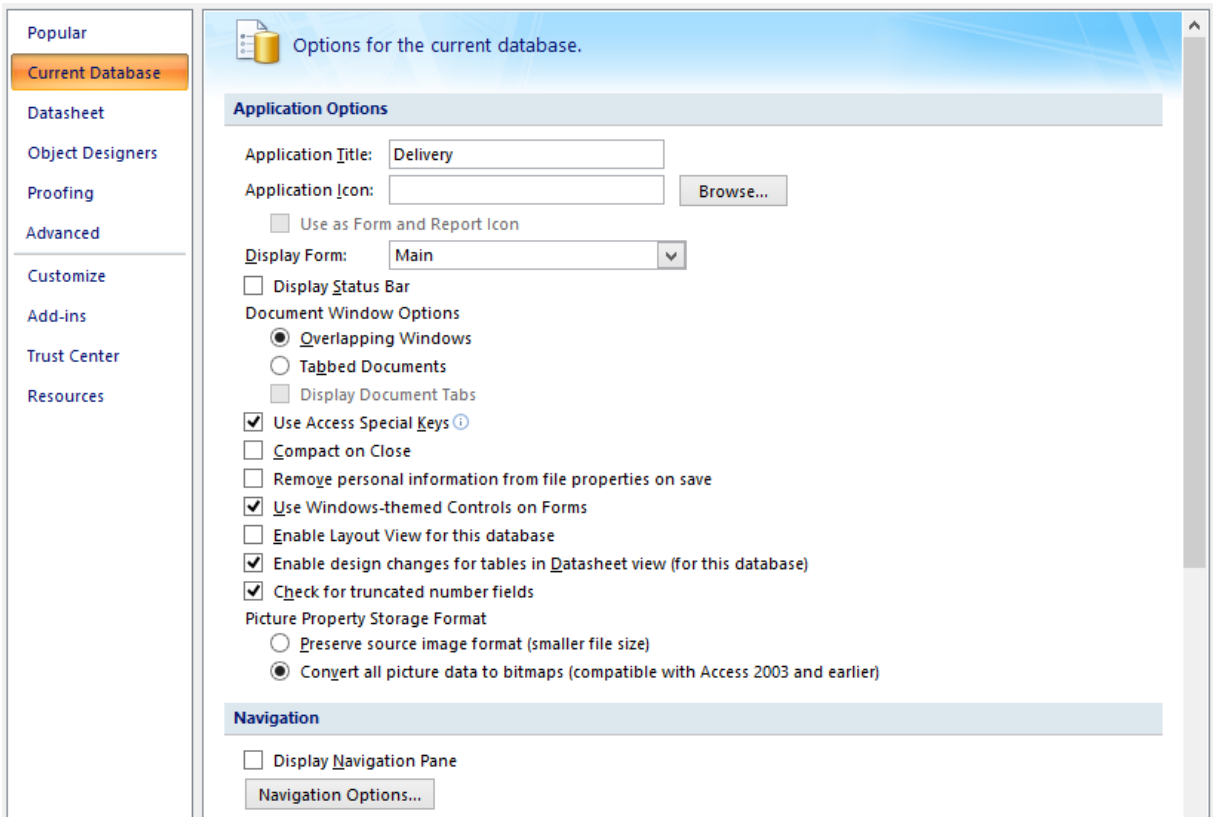



Figure 7.4

Follow the steps below to setup the database password:

1) Copy the database file;

2) Run DBMS Access;

3) Open exclusive the database copy (figure 7.5);

4) Open “Database Tools” tab and click “Set Database Password”;

5) Type “12345” into the password field and then type it again to verify the password;

6) Close database and DBMS Access;

7) Run database, enter the password into popup window and then try to unset database password. Password may be unset only in the exclusive mode.

2. Disable the database password.

1) Close the database but not DBMS Access;

2) Open the database using the exclusive mode;

3) Unset the password (it would be required to insert the password once again);

4) Close the database and the DBMS Access;

5) Run the database file to verify that the password is unset.

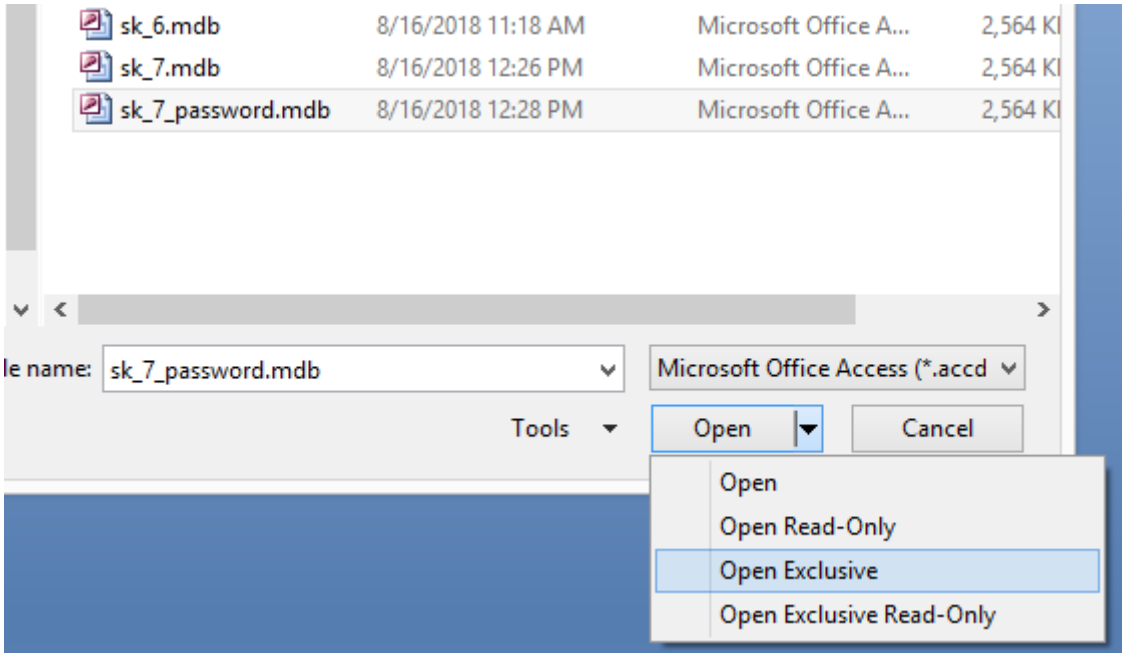



Figure 7.5

**Report requirements:**

1) Briefly describe main steps of this work;

2) Describe the purpose and structure of the created application;

3) Describe the sequence of steps required to merge application components;

4) Describe the sequence of steps required to setup application to run automatically;

5) Describe the sequence of steps required to setup and unset the database password.

## Questions

1. How to setup a button to open a form?
2. How to setup a button to open a query?
3. How to setup a button to open a report?
4. How to setup a database to automatically run an application?
5. How to disable automatically running setup?
6. How to setup a database password?
7. How to unset a database password?