# Lab No:04

**Title: Relationships, Simple Query Designing and Advance Queries (Custom Calculations in Queries).**

# Objective:

* How to create relationships between tables and how to design a query.

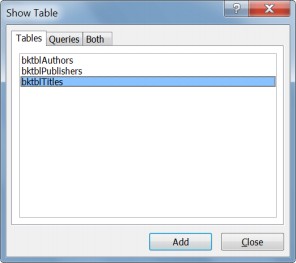
Using a query makes it easier to view, add, delete, or change data in your database. The objective of query is:

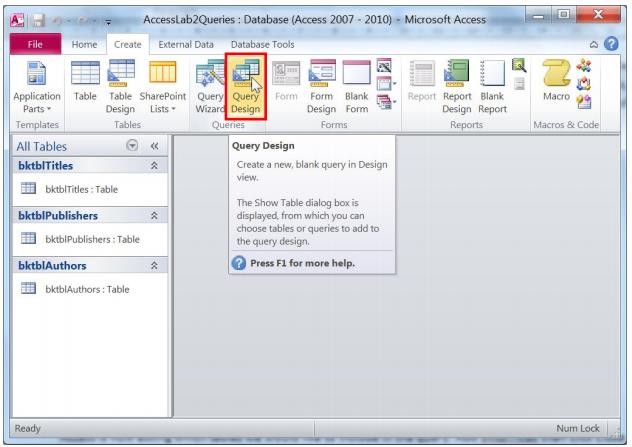
* Find specific data quickly by filtering on specific criteria (conditions).
* Calculate or summarize data.
* Automate data management tasks, such as reviewing the most current data on a recurring basis.

# Introduction:

Database relationships are associations between tables that are created using join statements to retrieve data. The following table describes the database relationships. Both tables can have only one record on each side of the relationship. Each primary key value relates to none or only one record in the related table

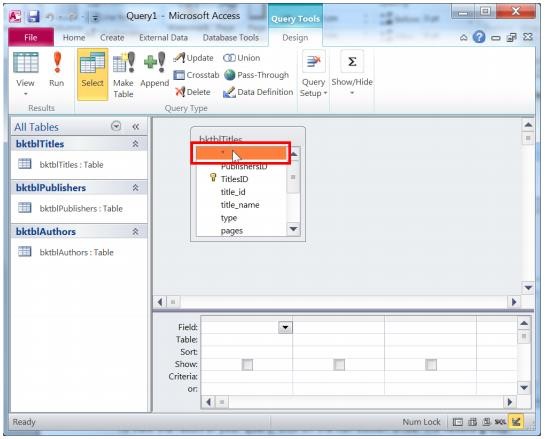
Relationships are the established associations between two or more tables. Relationships are based on common fields from more than one table, often involving primary and foreign keys. A primary key is the field (or fields) that is used to uniquely identify each record in a table **Queries**

Queries are questions submitted to the database. They can be used to filter, perform calculations on, and summarize data.

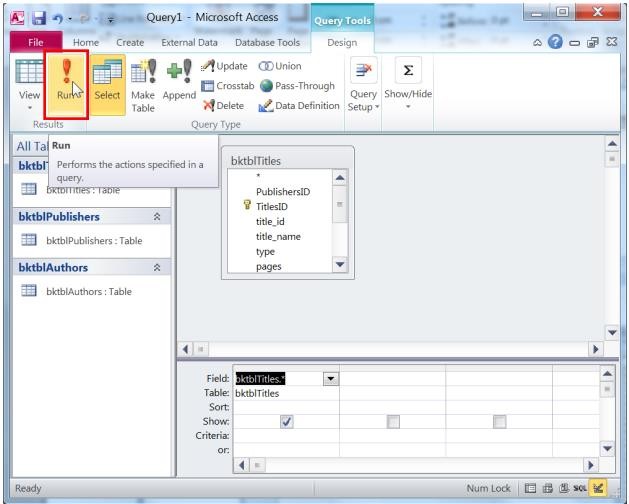


Access is now asking which tables we would like to include in the query. Add bktblTitles and press Close.

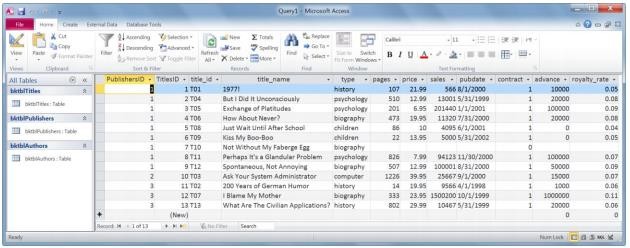
Access now shows you the design view of your query. This form of queries is called query by example or QBE. To add all the fields of the table into the query, double-click the \* under bktblTitles.



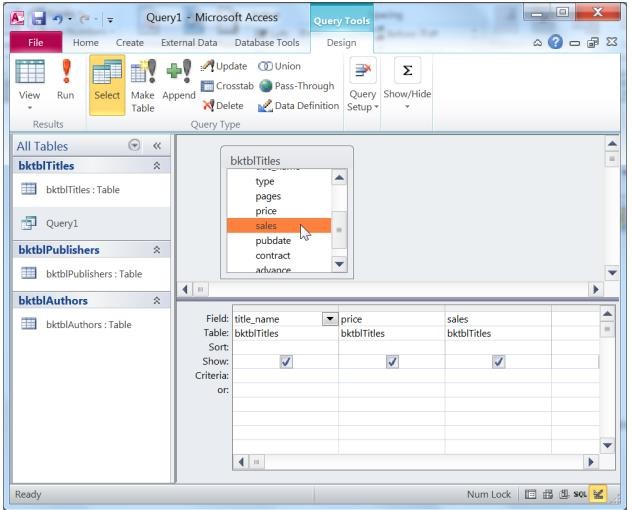
Now, to see the result of your query, click on the Run button under the Results group.



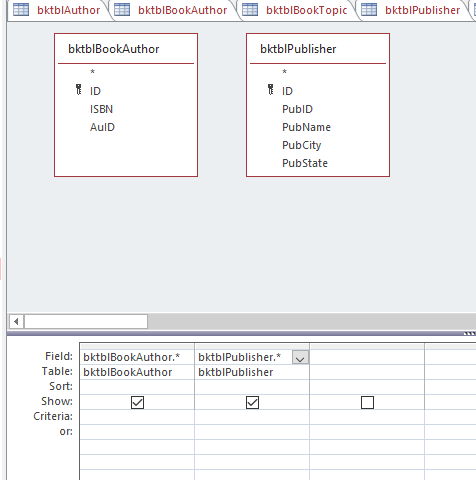
The query result now shows exactly what you asked for: the entire bktblTitles table



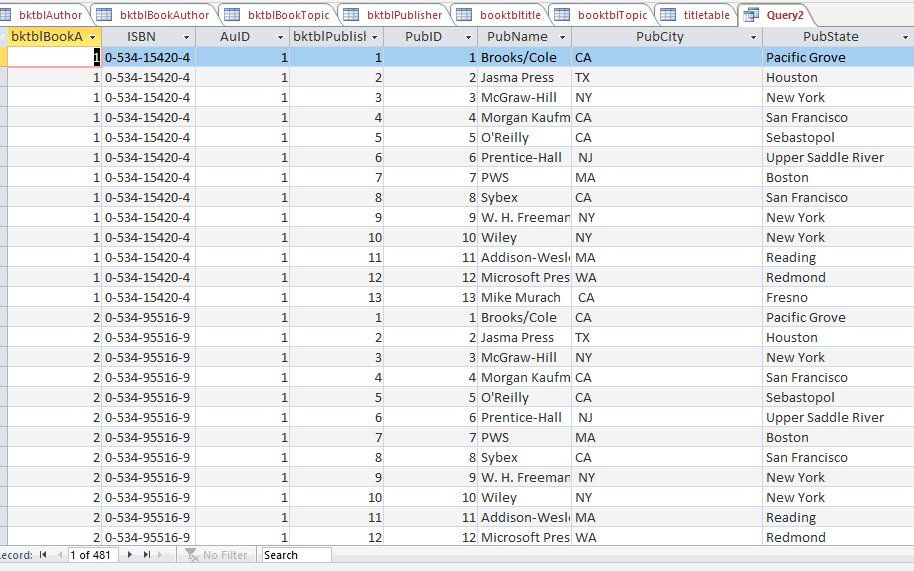
Go back to the design view of the query. Now only include the title name, price, and sales.



## Task 3:

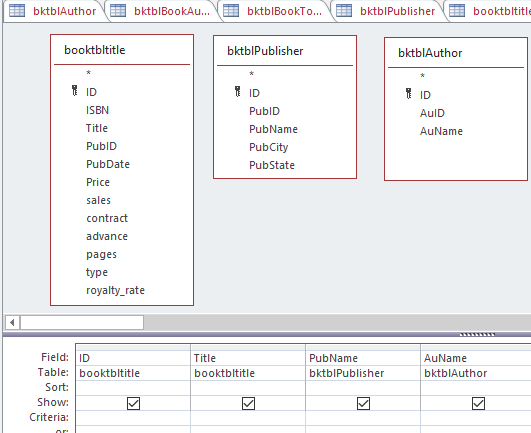
Create a new query that shows all the information in the bktblAuthors and bktblpublisher tables.

This is the query we made to show all the entries in the both Author’s and Publisher’s table. Both tables are selected and all entries in both tables are selected to show.



This is the running form of query that we made to show all the attributes of both tables and all the entries of both tables are shown in the running form.

## Task 4:

Create a new query that displays title ID, title name, the publisher’s name, and the author’s names.

This is the query we made to show the only entries in the three tables that we select title, author and publisher. This query shows the title id then title name

publisher’s

than name

from the publisher table and on last it shows the author’s name from the table of

author. entries

All these

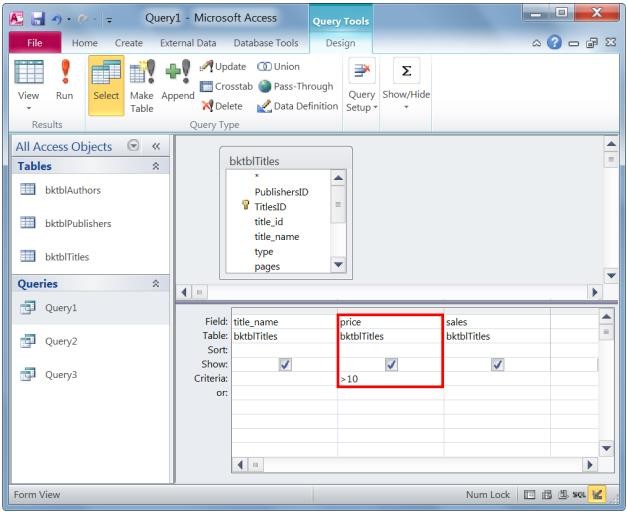
are

shown

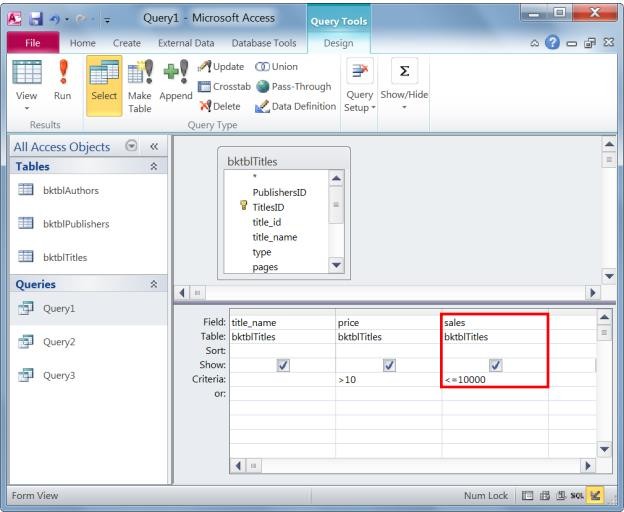
sequence wise.

# Criteria in queries:

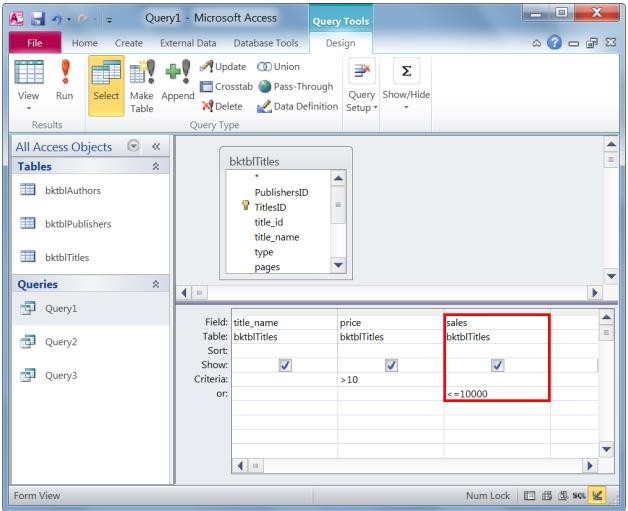
We can specify criteria for fields to limit query results based on their values. Let us display only the book entries with a price that is more than 10.



Now, what if we wanted to view all book titles whose price was greater than 10 AND at least 10000 copies?

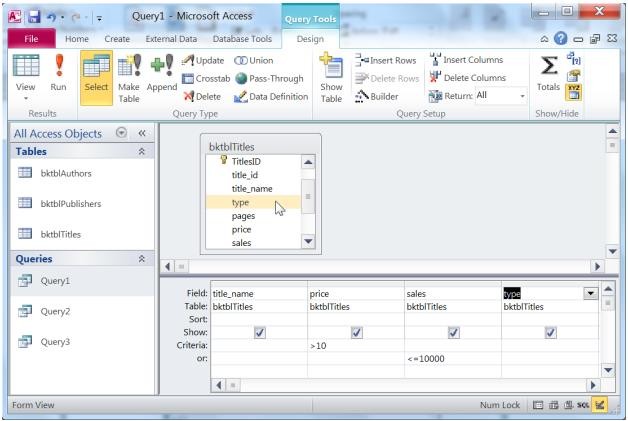


What if we want to display any book title with a price >10 or one with a sales amount <=10000? This is where the or field comes in handy.

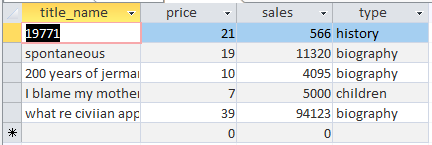


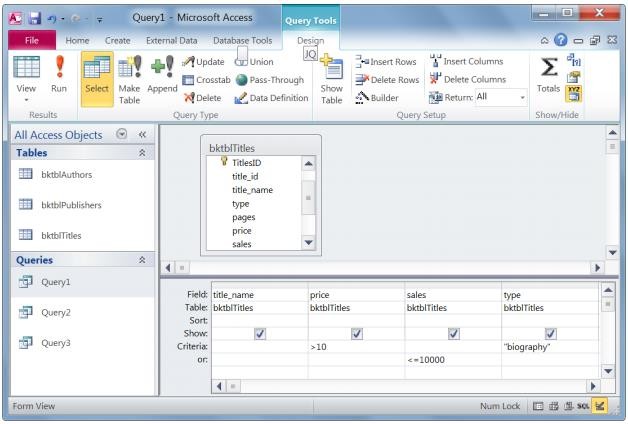
Now the query results show any book title with a price >10 or a sales amount <=10000.

We can also add query criteria to text fields. Add the field type from bktblTitles to the query



Let us only show book titles of type “biography”

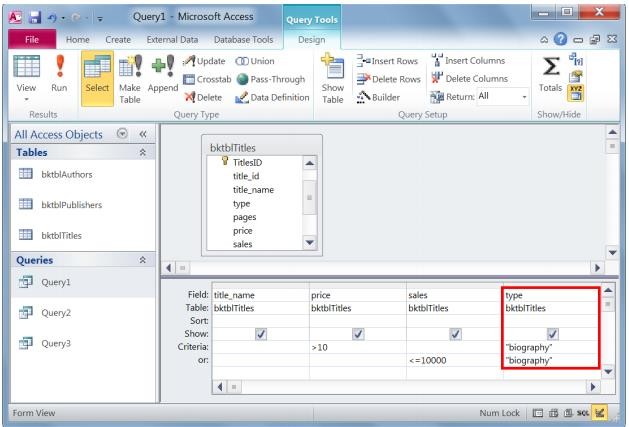




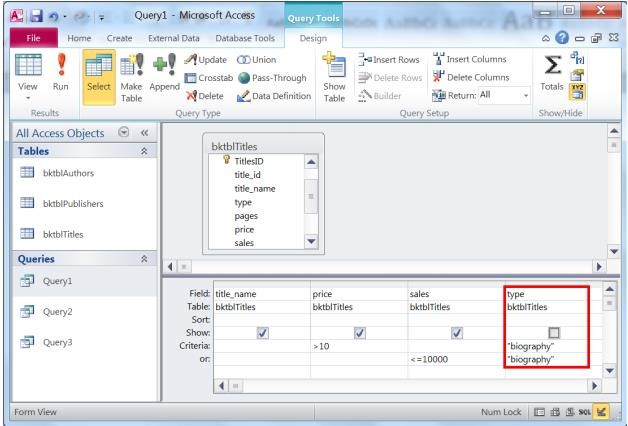
Now take a look at the results.

Notice how the or clause was applied in this query. The results show book titles with a price >10 and of type biography, or ones with sales <=10000.

If we would like to view only biography books with price>10 or biography books with sales amount <=10000.



We can also use a field to limit our query results without displaying that field in the query. We do this by unchecking the Show box. Do this for the type field.

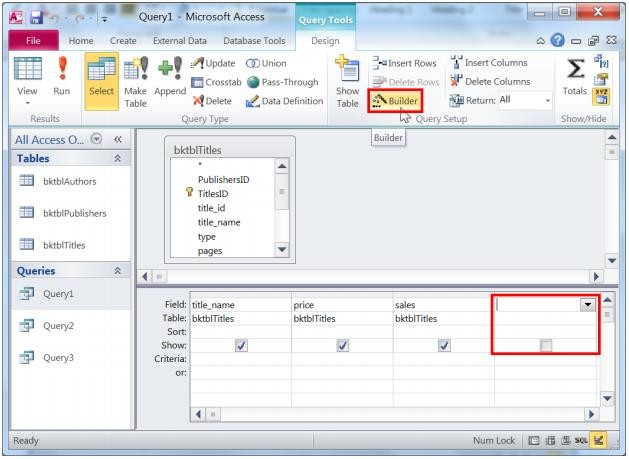


Now the type is still included in the query criteria but does not show up as a field in the query result. To include in the result, check the Show box under type.

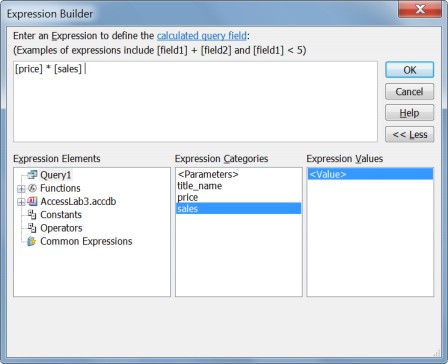
# Advanced Queries:

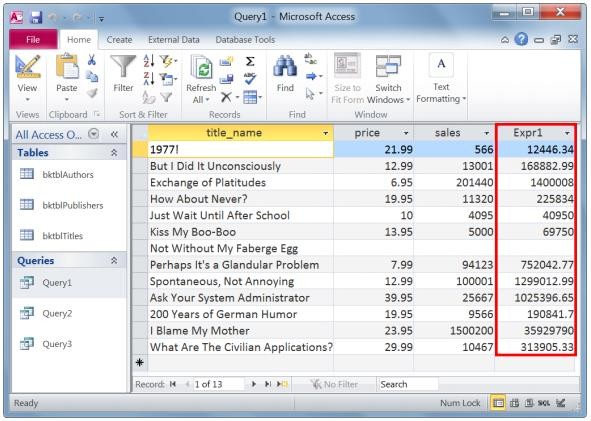
## Custom calculations in queries:

Sometimes, you want to specify certain calculated variables that would show up in your query result. Access allows you to create custom expressions in your query, similar to the formulas used in Excel.



Double-click price and then type in \* (the asterisk is the multiplication sign). Next, double-click on the sales field. Then, press the OK button.

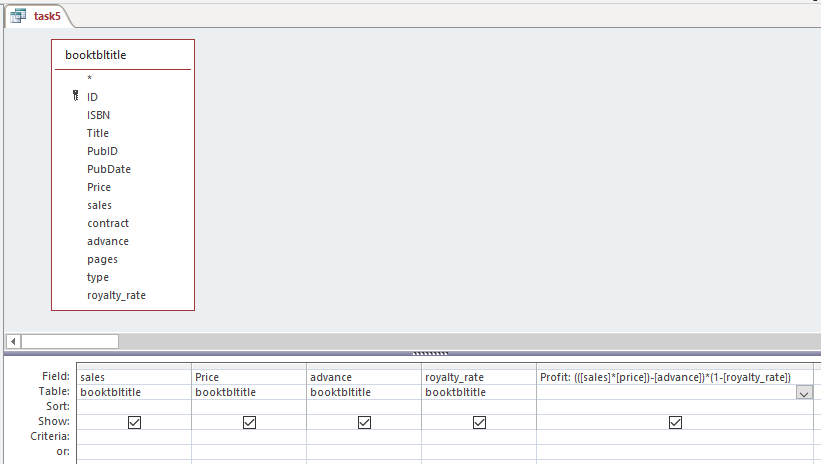
Run your query and notice how the new field is named Expr1 by default



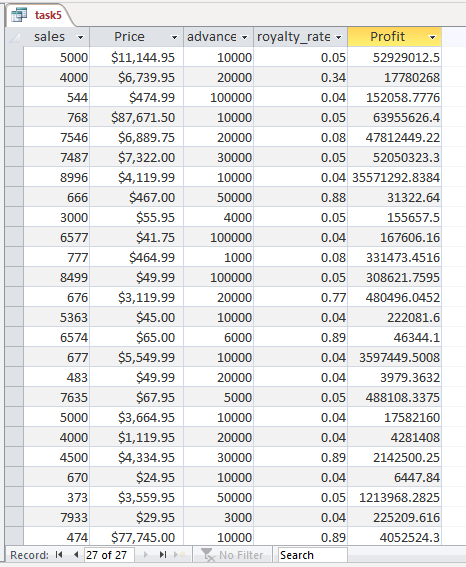
To give it a more meaningful name, return to the design view and change your expression from Expr1: [price]\*[sales] to Profit: [price]\*[sales]. The name before the colon is the heading of the field.

## Task 5:

The expression we created for profit is too simplified. Modify it so that the profit is calculated as: ((sales\*price)-advance) \* (1-royalty rate).



This is the query we write to find out the profit through the given formula in the table. This query actually makes an attribute named as “Profit” and then apply the formula to find out the profit and display.

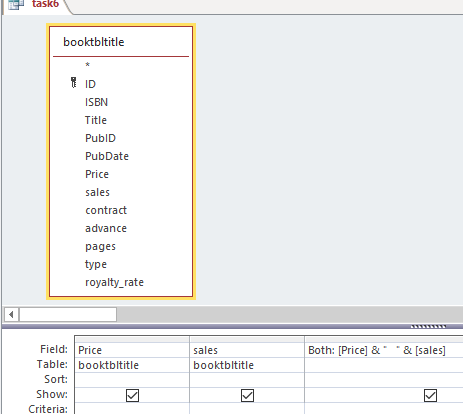


This is the output of the given formula in the question. It is showing all the data in the attributes we are using and also makes an attribute named as “Profit” and displays the profit using formula.

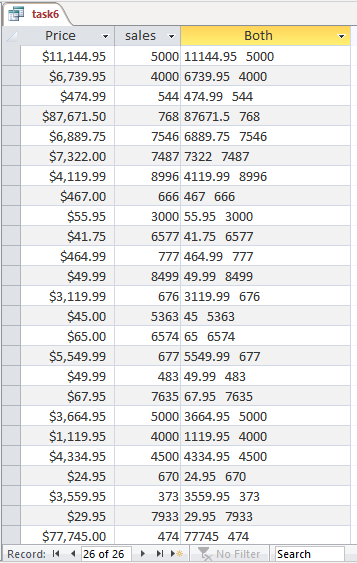
Profit = ((sales\*price)- advance) \* (1-royalty rate).

## Task 6:

Combine both fields (price and sales) in one column with a space.



We write the query to join both fields. First, we display both fields separately and then made the field named as “Both” which contains data of both with a space between them without changing in their values.



The task was to combine the both fields of “Price” and “Sales”. So, we make a query to first display the attributes of “Price” and then the attributes of “Sales” separately. Then we make the attribute named as both which contain the data of both attributes with a single space between without changing the values.