IS 312 Database (with Access) Tutorial 3 and Demo 3:   
**Total Query: Creation, Interpretation, Planning/Design***Dr. Yüe “Jeff” Zhang*Version 17, February 11, 2018

We have learned about basic and calculated field queries. There are another type of queries that have more significant role in business analysis and decision making: Total query.

In business, we often need to find out the **summary characteristics** of a selected group, such as

* **average** GPA (summary characteristics) of Accounting students (selected group);
* **average** purchase amount (summary char.) of customers in zip 91324 (selected group);
* **earliest** opened date of a restaurant (summary char.) in Miami (selected group);
* **average** number of members (summary char.) in gyms in Northridge (group);
* **highest** sales amount (summary char.) in each product line (group);

About a group!

* …

The total query would deliver the answers to the above questions.

Cannot over emphasize

NOTE: Grouping is critical! Always determine the **grouping field first!**

**I. Total Query Creation**

|  |  |
| --- | --- |
| In **Query Design View**, move the mouse cursor within the Design Grid, right click.  A pop-up menu will appear, with the first item being “∑” – TotalS.  Click to select ∑ – Totals, a new row “Total” for Total Query will be added to the Query Design view – see the figure on top of next page. |  |

The default operation or function for the total row for fields selected to display is “Group By”.

We will change the Group By default to some of the following functions:

SUM, COUNT, AVG, MIN, MAX, StDev, … 【these are group summaries performed on A GROUP】, the group determined by the **Group By** field.

About a pre-specified group

**The results of total queries describe the characteristics of a GROUP (SUM, AVG, MIN/ MAX, COUNT, … of the GROUP);**

**The results are NOT about individual records: the results will only have as many rows as there are \_\_\_\_\_?\_\_\_\_\_\_s.**

|  |  |
| --- | --- |
|  | Options for the total query operations:  SUM, AVG, MIN, MAX, COUNT –covered here  StDev, Var, First, Last, Expression |

Demo 3-1 (“RestaurantsDB”): Creating a total query involving SUM, COUNT, AVG, MIN, and MAX.

Watch and think: Which field to group, and which fields to SUM, COUNT, AVG, …

We usually group on ONE field – grouping on two or more fields are rare and must be handled with extra care. \*\* Think: What kind of fields to group upon?

All fields – Group By or aggr function fields - in total query can have cri – Ex: AVG>3.0, Major=”Acct”, etc

Key! Most important and useful!

**II. Total Query Result Interpretation/Analysis**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Type of fields | Type-field | **Unique** field | Number field | Number/ Date field | Number/ Date field | **Calculated** field |
| Total: | Group By | COUNT | AVG | MAX | MIN | AVG/MIN/MAX |
| Interpre-tation/remark | **Usually text** | # members in the group | AVG of the values in group | MAX of the values in group | MIN of the values in group | Av/Mi/Mx of calc field value in group |
| Can have their cri – Example: | (Major) Acct;  (Zip) 9132\* | COUNT can have criteria:  >=100 | AVG can have cri:  >=3.5 | MAX can have criteria:  >=#1/1/2010# | Same as MAX | Profit:Price-Cost  In Criteria:  >=7.50 |

NOTE:

1, Group on fields that have “group nature” (such as City, Type, Zipcode, Major, Productline, …) – that has several rows having the same “type”-value (same zipcode, same product line, same major, for example)

* 1. AVOID (!!!!!!!!) group by fields that have unique values – why? Compared with “1” above.

2. All the math/stat functions must be used only on number/currency fields – makes sense right?

Demo 3-2: Given the query outcome (results) of the total query as created in Demo 3-1, try to interpret the results.

\*\*\* Interpretation is important: Imagine facing the Board or the customer.

One more exercise:

Given the following total query design, interpret the outcome of the query.

The interpretations must be -

* necessarily specific – “tell something in the scenario”; logically correct; theoretically accurate/rigorous; relatively concise.

Design grid using STUDENT table -

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field: | Major | Stud\_ID | GPA | GPA | DOB | CatYear | ExpGradYr | Duration:GradYr-CatYear |
| Total: | Group By | COUNT | AVG | MAX | MIN | MIN | MAX | AVG |
| Interpret |  |  |  |  |  |  |  |  |

**III. Given Business Questions, Design Total Query to Provide Relevant Results**

Example: Given ORDER\_LINE table as follows: [Discussion: What is primary key here?]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cust\_ID** | **Prod\_ID** | **Pdt\_Line** | **Price** | **Quant\_Purchased** | **Date\_Purchased** | **Zipcode** |
| C11012 | KitA13 | Kitchen | 12.34 | 6 | 11/22/2017 | 91324 |
| C14321 | LivC06 | Living | 23.45 | 4 | 11/22/2017 | 91325 |
| C13131 | KitA13 | Kitchen | 12.34 | 2 | 12/01/2017 | 91324 |
| C15644 | BthB07 | Bath | 56.78 | 2 | 12/02/2017 | 91326 |
| C11012 | KitB11 | Kitchen | 67.89 | 1 | 12/02/2018 | 91324 |
| C14321 | BedA02 | Bedroom | 178.90 | 2 | 1/04/2018 | 91325 |
| C11012 | KitA01 | Kitchen | 116.78 | 1 | 1/05/2018 | 91324 |
| C15644 | KitB11 | Kitchen | 67.89 | 1 | 1/05/2018 | 91326 |
| C13131 | BedA02 | Bedroom | 178.90 | 2 | 1/10/2018 | 91324 |

Answer the following questions with total queries:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Number of items purchased? | Total purchase amount? | Highest amount of purchase? | Average purchase amount? | When first purchase? | When last purchase? |  |
|  |  |  |  |  |  |  |  |
| Group By |  |  |  |  |  |  |  |

I want to see these types of total queries in your DB project

More examples/cases:

* average purchase amount from customers of a certain zip code (Think!);
* total sales amount of a certain product;
* total purchase on a certain date;
* average salary of employees of each job title;
* earliest available date of apartments of certain size (1-bed, 2-bed, etc);
* average commission rate of sales reps in a certain sales region, and total commission (dollars) they earned;
* … (the above list can also be used to stimulate your own design of total queries for your DB project!)

Can you come up with more? – good exercise

Demo 3-3: Use the T7\_BookFinder.accdb database to answer business questions. Please think about your solutions so as to be able to participate in class.

**IV. Given Data Structure, Design Total Query to Support Decision Analyses**

Open-ended discussion: Given the following data table structure, what kind of questions can we answer from possible total queries? 【Think BEFORE class; will have a discussion】

APARTMENT (Note: Type – 3-bed, 2-bed, 1-bed, studio)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Type | Apt# | SqFt | Rent | Contract# | LeaseStart | LeaseEnd | LeaseLength: | #\_occupant |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

**==========================================================**

**312 Access Database Demo 3: Creating and Interpreting Total Queries**

**Demo 3-1: Creating Total Queries**

**Demo question:**

In all the **cities**, find

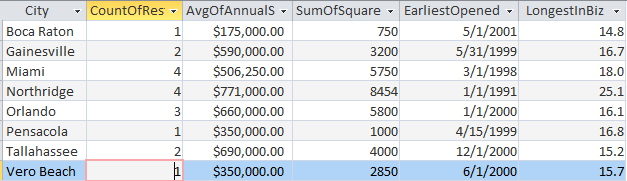
1. The number of restaurants **in each city**;
2. average annual sales of restaurants **in each city**;
3. total squared footage & largest SqFt of restaurants **in each city**;
4. earliest opening date of restaurants **in each city**,
5. longest time in business **for each city**,

Design:

**Group By** City,

**Count** \_\_\_\_\_\_\_\_\_, **Avg** \_\_\_\_\_\_\_\_\_\_\_\_\_, **Sum** \_\_\_\_, **MIN** \_\_\_\_\_\_\_\_\_\_, **MAX** \_\_\_\_\_\_\_\_\_\_

Result:



Question: In addition to the above, if you also **Group By** RestaurantD, what would happen? Group By DateOpened?

Observe, think, note the results, and act accordingly for your homework and project, as well as in future exams.

Do NOT Group By fields whose values are \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Group By only by fields whose values are \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

– Do NOT **Group by** every field!

Count only \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ type of fields

Sum/Avg only \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ type of fields

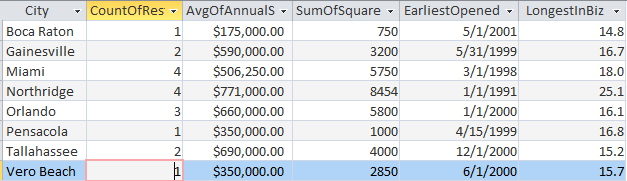
Some other scenarios:

CUSTOMER table: do we group by their date of first purchase? By their zip code? By their gender? Why or why not?

CAR table: do we group by VIN? By price? By make/model? By mileage? Why or why not?

**Demo 3-2: Interpreting Results of Total Queries**

Given the query outcome from Demo 3-1, interpret the outcome:



**Discussion:**

1. Are the outcomes about an individual restaurant?

2. If SUM is used, what range is it summed across? What does the result of COUNT mean?

3. **CRITICAL!!** – which field should be **Group By**? In the restaurant table, do we group by street address? Annual sales? Data opened? City? Type of service? Why or why not?

-- This is THE FIRST QUESTION you always need to ask when creating a total query.

**The results of total queries are about A GROUP, not about individual records.**

**Demo 3-3: Given Business Questions, Design Total Queries**

In this demo, we will use the T7\_BookFinder.accdb database to find the following information. Please think about your solutions, so as to be able to participate in class:

* Which authors tend to write longer books?
* Which author’s books re sold at higher prices?
* Books of which categories tend to be thicker?
* Books of which categories tend to sell at higher price?
* Questions similar to the above, but now regarding “pricePerPage”.

Think about your own total queries, and then compare them with the ones I will create in the demo. Check your solutions against mine, and answer or raise questions correspondingly.

**Demo 3-4: Given Data Structure, Design Total Queries**

Think about how to design total queries to support business analysis in the following situations (with the following given scenarios and attributes), i.e., Group By what field, COUNT/SUM/AVG /MIN/MAX what field(s), that makes the best business sense. Design the respective queries, and provide interpretations to the queries and their results.

1. Sales analysis

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| StoreID | Region | City | Quarter | numEmployees | salesAmt |  |

1. DVD rental

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MovieID | Title | Genre | Director | leadActor | leadActress | Year | numNomin | numAward | copiesRented |

1. Phone talks analysis

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| AreaCode | Number | callDate | callWeekDay | callTimeInDay | Length | OutGoing |  |

1. Sales rep performance analysis

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RepID | LName | FName | Title | Store | yearHired | Skill | salesAmt | satisfacRating |

1. Traffic violation analysis

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DriverLicNum | LName | FName | Sex | DOB | AlchoTestData | Date | Time | WeekDay |