

Denver Area Access  
Users Group

Newsletter

August 2, 2001

Patricia Chase Riley

Newsletter Editor

[DAAUG's Website](#)



## Regular Monthly Meetings

Normal Meeting Locations:

**Date:** 2<sup>nd</sup> Thursday of each month.

**Time:** 6:30 to 8:15 pm

**Locations:** alternate between 2 different locations.

**Odd-numbered months** (Jan, Mar, May, Jul, Sep, Nov) we meet **Downtown** at:

**ING Security Life Building**

1290 Broadway (SE Corner of 13th & Broadway - across the street from the Main Library)

**In even-numbered months** (Feb, Apr, Jun, Aug, Oct, Dec) we meet in **Highlands Ranch** at:

**Red Rocks Federal Credit Union**

200 W. Plaza Drive (C-470 to Broadway, South to 2<sup>nd</sup> light - West (right) on Plaza Drive 2 blocks - building is on your left)

Monthly Lab Meetings

**Date:** 4<sup>th</sup> Wednesday of each month (except December! - **No Lab in December**)

**Time:** 6:30 to 8:15 pm

**Location:** [ExecuTrain](#) of Denver Tech Center

6160 So Syracuse Way, Suite 160B, Englewood, CO 80111

**Phone:** (303) 436-1000 or (303) 749-2060 **Fax:** (303) 749-2068

(Updated 11/30/00)

**Directions:**

From I-25: Take I-25 to Arapahoe Road Exit. Turn West onto Arapahoe Road. Turn right (North) onto Syracuse Way. Go North to 6160 So Syracuse Way. Executrain is in the building on the Southeast corner of Syracuse Way and Maplewood Ave, in Suite 160B. Please call Executrain @ (303) 436-1000 if you have questions about the location.

Page	Inside this Issue
1	<a href="#">Times and Locations</a>
2	<a href="#">The "President's Corner"</a>
2	<a href="#">This Month's Presenter</a>
2	<a href="#">Relationships in Access Databases</a>
3	<a href="#">SeQueL 5 – Queries Talk to the ORACLE!</a>
5	<a href="#">Just for Fun</a>
5	<a href="#">Recap of Last Month's Meeting</a>
5	<a href="#">DAAUG Officers</a>
5	<a href="#">Library Notes</a>
5	<a href="#">New Members</a>
6	<a href="#">2001 Presentation Schedule</a>

## “President’s Corner”

by Jim Pilcher

Don’t miss this month’s DAAUG presentation by Carl Dubler, *Access 2002—What’s New?* We have a whole new version of our favorite software available now and Carl will show us how to get the most from it’s new features. We will also have a few give-aways relating to Access 2002 and general computing. I hope to see you at the Red Rocks Credit Union on August 9<sup>th</sup>. Your Denver Area Access User Group is a volunteer non-profit organization. None of your board members are affiliated in any way with Microsoft or any of our major sponsors. We rely upon the resources of our members for things such as meeting space, projection equipment, etc... As far as sponsorship goes, FMS regularly provides us with copies of their products for our semi-regular monthly drawings. TriSys Software provides with web programming resources. ViaWest hosts our web site. ExecuTrain provides hands-on lab space. And of course, Microsoft provides us with our reason to exist as well as occasionally providing a speaker and meeting space. With all that said, I want to put out a request to our membership. One of the most expensive needs is the occasional projector. It can cost your user group as much as \$250 to rent a projector for an evening. DAAUG wants to put together a list of members that are willing to provide a projector once or twice per year. Typically we need a projector five times a year. If we can get two or three DAAUG members to provide a projector, we can save up to \$1000 a year in rental. Give it a thought and contact Bruce Benninghoff or me to offer use of this necessary piece of equipment. Thanks, folks.

---

### This Month’s Presenter

**Carl Dubler** is a Senior Technical Specialist in Microsoft's Rocky Mountain district, helping customers understand and buy SQL Server and Business Intelligence solutions. Prior to joining Microsoft, Carl ran his own SQL Server and BI training and consulting company and has an IT background going back to 1987. With extensive experience technically and in public speaking and debate, Carl has presented seminars and classes in 3 countries on DBA, developer, public speaking, and professional development topics.

---

### Relationships in Access Databases

by Jim Pilcher

One of the most common flaws I find in databases I’m asked to evaluate is the incomplete or improper use of relationships. In fact, I’ve reviewed many “finished” databases that had no defined relationships at all! It makes an experienced database designer want to run down the street screaming and pulling out his hair (in my case, pulling out what’s left of my hair). I’ll

offer here a short discussion of some important concepts surrounding relationships in Microsoft Access.

There are three basic types of relationships in relational database design. The first, and most common, is the one-to-many relationship. An example of this type of relationship is a table with student names and another table that contains all the courses for which a student has registered. The relationship, probably defined through the use of a student ID number in both tables, is one-to-many because one student can take many classes. A technical description describes this as a relationship between the primary key of the “parent” table and a foreign key of the “child” table.

A second type of relationship is the one-to-one variety. This is a relationship through the primary keys of two tables. Typically, if you have the need to create a one-to-one relationship, you should consider putting all of the information in one table during the design phase of a project. However, business rules, performance, or technical limitations of the database engine might make a one-to-one relationship desirable. For example, Microsoft Access does not allow security at the field level in a table. An employee list may include salary information that only the payroll department should see. By moving salary information into a second table with a one-to-one relationship to the first table on employee ID or social security number, security can be maintained. That’s because Access *does* allow security at the table level. The primary employee table can be left open to most users, while the payroll table can be locked down so only the payroll group has access to the information. Another use for a one-to-one relationship is a case where a primary table might have millions of records, but only a few thousand have certain pieces of information. By taking that information into a secondary table and establishing a one-to-one relationship between them there is a potential for significant performance gain and disk space savings. After the split, we do not have millions of empty fields in the primary table.

The final relationship in this trio of types is the many-to-many relationship. Relational design does not directly support the concept of many-to-many relationships, so they are actually composed, in their simplest form, of two one-to-many relationships. Returning to our example of students and classes, recall that one student can have many classes. Also, one class can have many students. That means that the relationship between students and classes is many-to-many. The database construct to model this relationship is to place a linking entity (table) between students and classes. The linking table has exactly two fields, the student ID and the Class ID. By making this combination of two fields the primary key of the linking table, we have created a very effective many-to-many relationship. It is now easy to print a class schedule for a student and print a class roster for the teacher, all without duplicating any information about the students or classes!

Next month I’ll discuss referential integrity, inner and outer joins, and how to establish relationships between queries and tables. Until then, happy computing!

---

# SeQueL 5 – Queries

## Talk to the ORACLE!

by Clark Anderson

For quite a while, I wondered just how different SQL for ORACLE would be from SQL for MS Access. I will show some of the similarities and differences. To provide examples I will revisit the bagel shop of my earlier articles. Here are two tables I used:

BagelsBaked Table

BakedID	BagelID	Name	Quantity
1	1	Plain	360
2	4	Sesame Seed	240
3	7	Poppy Seed	240
4	2	Egg	240

BagelsOrdered Table

OrderID	BagelID	Name	Quantity
1	1	Plain	240
2	2	Egg	120
3	5	Whole Wheat	120
4	3	Everything	60

The MS Access SQL, shown below, brings data from two tables together.

```
SELECT BagelsBaked.Name, BagelsBaked.Quantity,
BagelsOrdered.Quantity, BagelsOrdered.Name
FROM BagelsBaked INNER JOIN BagelsOrdered ON
BagelsBaked.Name = BagelsOrdered.Name;
```

Here is an equivalent query for ORACLE:

```
SELECT BagelsBaked.Name, BagelsBaked.Quantity,
BagelsOrdered.Quantity, BagelsOrdered.Name
FROM BagelsBaked, BagelsOrdered WHERE
BagelsBaked.Name = BagelsOrdered.Name;
```

Notice that the SELECT clause is the same, but the FROM clause merely lists the table names. Also note that the information in the MS Access JOIN clause is converted to a WHERE clause for ORACLE.

BagelsBaked		BagelsOrdered	
Name	Quantity	Quantity	Name
Egg	240	120	Egg

Plain	360	240	Plain
-------	-----	-----	-------

We can adjust the SQL to use the AS keyword providing different, alias, names for the columns:

```
SELECT BagelsBaked.Name AS BakBagels,
BagelsBaked.Quantity AS BakedQty,
BagelsOrdered.Quantity AS OrderedQty,
BagelsOrdered.Name AS OrdBagels
FROM BagelsBaked INNER JOIN BagelsOrdered ON
BagelsBaked.Name = BagelsOrdered.Name
ORDER BY BagelsBaked.Name
```

Here is an equivalent query for ORACLE:

```
SELECT BagelsBaked.Name BakBagels,
BagelsBaked.Quantity BakedQty,
BagelsOrdered.Quantity OrderedQty,
BagelsOrdered.Name OrdBagels
FROM BagelsBaked, BagelsOrdered WHERE
BagelsBaked.Name = BagelsOrdered.Name
ORDER BY BagelsBaked.Name
```

ORACLE SQL achieves the same result by placing the alias name immediately after the original name with just a space instead of the AS keyword.

BagelsBaked		BagelsOrdered	
BakBagels	BakedQty	OrderedQty	OrdBagels
Egg	240	120	Egg
Plain	360	240	Plain

ORACLE can take this alias syntax a step further with the table names listed in the FROM clause:

```
SELECT BB.Name BakBagels, BB.Quantity BakedQty,
BO.Quantity OrderedQty, BO.Name OrdBagels
FROM BagelsBaked BB, BagelsOrdered BO WHERE
BB.Name = BO.Name;
ORDER BY BakBagels
```

The BB alias, defined in the FROM clause, for the BagelsBaked table can be used every where else in the query! Of course the same is true for the alias defined for the BagelsOrdered table, BO. I find this feature very handy. I have seen some very long table and column names in ORACLE databases. Also, please note that the ORDER BY clause can refer to the column's alias name, BakBagels.

Now I will recall an example from another article. Here is part of a reference table in a LookUp database:

```
SELECT StateCode, StateName, Country
FROM [D:\SCRATCH\LookUp.mdb].LUStates
```

WHERE StateCode IN  
('CO','CT','WA','KS','ON','FL','AB');  
( continued)

LUStates Table

StateCode	StateName	Country
CO	Colorado	USA
CT	Connecticut	USA
FL	Florida	USA
KS	Kansas	USA
WA	Washington	USA
AB	Alberta	Canada
ON	Ontario	Canada

And another table:

CoffeeHouses Table

Shop	City	State
Moe's	Bloomfield	CT
Real Coffee	Seattle	WA
Cafe Luna	Longmont	CO
Espresso Roma	Boulder	CO
Central Café	Lyons	KS
Aggie's Diner	Ottawa	ON

I can JOIN these tables from their separate databases:

```
SELECT CoffeeHouses.Shop, CoffeeHouses.City,
LUStates.StateName AS State
FROM CoffeeHouses INNER JOIN
[D:\SCRATCH\LookUp.mdb].LUStates ON
CoffeeHouses.State = LUStates.StateCode;
(continued)
```

Shop	City	State
Moe's	Bloomfield	Connecticut
Real Coffee	Seattle	Washington
Cafe Luna	Longmont	Colorado
Espresso Roma	Boulder	Colorado
Central Café	Lyons	Kansas

Aggie's Diner	Ottawa	Ontario
---------------	--------	---------

Whereas MS Access often has separate database files, ORACLE and, I have been told, many other “true SQL” databases tend to combine these separate databases into one and refer to each as separate and independent SCHEMA. To convert this query into ORACLE, I will assume two SCHEMAS, COFFEE and LOOKUP.

```
SELECT CoffeeHouses.Shop, CoffeeHouses.City,
LUStates.StateName State
FROM COFFEE.CoffeeHouses, LOOKUP.LUStates
WHERE CoffeeHouses.State = LUStates.StateCode;
```

Again, the FROM clause completely identifies the tables involved in this INNER JOIN. The syntax is SCHEMA name, dot, table name. I have learned, that it is a very good idea, to always specify the SCHEMA name for each table in the FROM clause.

The same use of alias names still applies:

```
SELECT CH.Shop, CH.City, LUS.StateName State
FROM COFFEE.CoffeeHouses CH, LOOKUP.LUStates
LUS WHERE CH.State = LUS.StateCode;
```

I have provided aliases for each of the tables, CH and LUS, as well as the last column, State.

One of my friends, an Oracle DBA, has pointed out that there are other formats available on Oracle SQL to indicate aliases:

```
StateName AS State
StateName AS "State Name"
```

The AS keyword is optional and double quotes are used, instead of square brackets, to identify a name that contains spaces or other special characters. These variations on the alias syntax indicate that in the SQL used by ORACLE, “there are many ways to skin a cat”.

There is certainly a lot more to learn about ORACLE’s “Standard SQL”, but this is a pretty good start for using this variation of SQL.

I have delivered a lot of software products using MS Access SQL, but it is only this year that I have worked with the SQL used by ORACLE.

I have recently learned the syntax for outer joins, but I want to save that for another article. I will also get more experience with the use of wild cards in ORACLE’s Standard SQL and present that in a future article. I am happy to share my discoveries!

It is even more encouraging to discover that what I am learning in my ORACLE SQL experience has broader use.

If someone says "I know you are skilled in MS Access SQL, but we have this task involving ORACLE / SQL Server / InterBase / Informix / DB2..." I hope this article will help you respond, "I can do that!"

---

## Just for Fun

Dilbert's "Salary Theorem" states that "Engineers and scientists can never earn as much as business executives and sales people." This theorem can now be supported by a mathematical analysis based on the following two postulates

Postulate 1: Knowledge is Power.

Postulate 2: Time is Money.

As every engineer knows:  $\text{Power} = \text{Work} / \text{Time}$

Since:  $(\text{Knowledge} = \text{Power}) \text{ AND } (\text{Time} = \text{Money})$

Then:  $\text{Knowledge} = \text{Work} / \text{Money}$

Solving for Money, we get:  $\text{Money} = \text{Work} / \text{Knowledge}$ .

Thus, as Knowledge approaches zero, Money approaches infinity, regardless of the amount of work done.

Conclusion: The less you know, the more you make.

---

## Recap of Last Month's Meeting

by Patricia Chase Riley

Last month DAAUG welcomed two experienced presenters. We first heard from Mel Harris, a dynamic and much sought-after trainer and speaker. Mel's elegant phraseology and ability to simply explain difficult concepts like XML made for an enlightening overview. His entire presentation can be obtained by visiting his website at [toolman.jncd.com](http://toolman.jncd.com).

Our second speaker was one of DAAUG's long-time favorites: Sco Scofield. Sco managed to deliver a winning presentation on Access Data Projects. We learned about the client/server mindset, architecture, SQL. The server wishes to share resources with the clients. Individual clients also want to share but desire an element of autonomy. How can you manage the competing demands and scarce resources? One piece of advice for serious Access/SQL developers: Don't use the "baby SQL" (that's the desktop version that ships with Access 2000/2002); there are no database management capabilities available. Get the full

product if you are serious about SQL computing.

Both presenters have way of making complex "target topics" attainable and entertaining. If you missed last month, you missed a great evening. We all hope to see you on August 9th.

---

## 2001 Officers

President - James H Pilcher Jr. MCP,  
[JPilcher@qwest.net](mailto:JPilcher@qwest.net)

DataOne  
PO Box 261220  
Littleton, CO 80163-1220  
Phone 720-344-7155 business and mobile

Vice President - Bruce Benninghof,  
[bruce@prolynx.com](mailto:bruce@prolynx.com)

Secretary / Treasurer/Membership - Dean Haze,  
[DAAUGTREAS@aol.com](mailto:DAAUGTREAS@aol.com)

Lab / Education Director - Roger Bale,  
[rogerbale@worldnet.att.net](mailto:rogerbale@worldnet.att.net)

Newsletter - Patricia Chase Riley,  
[Chase\\_Patricia@hotmail.com](mailto:Chase_Patricia@hotmail.com)

Webmaster - Louie Walker, Jr.,  
[lwjr@trisysssoftware.net](mailto:lwjr@trisysssoftware.net)

Librarian - Mark Evans,  
[mt\\_evans@msn.com](mailto:mt_evans@msn.com)

---

## Library Notes

We no longer have charge account access, so remember to bring your checkbook if you want anything from the library. To check an item out of the library, you must leave a check for value of the item as the deposit. Your check will not be deposited if you bring the library item back when you agreed to return it. Come see what books and other learning materials are available at the next meeting.

Mark Evans, Librarian

---

## New Members

Kevin Bell

Clyde George

Ray Giacalone

James Peyton

David Sherrill

Welcome to the DAAUG!

---

## 2001 Schedule of Presentations – subject to change

	Location	Meeting Date	Main Presentation Topic	Presenter	10-Minute Topic	Lab Location	Lab Date
<b>January-01</b>	ING Building	11-Jan	Forms Basics	Chuck Robbins	AutoExec vs. Startup Form	ExecuTrain	24-Jan
<b>February-01</b>	Red Rocks Credit Union	8-Feb	Reports Basics	Bruce Benninghoff	Table-Level validation	ExecuTrain	28-Feb
<b>March-01</b>	ING Building	8-Mar	Special Presentation: Luke Chung from FMS	<b>Luke Chung</b>	none	ExecuTrain	28-Mar
<b>April-01</b>	Red Rocks Credit Union	12-Apr	So You Want To Be An Access Consultant?	Pilcher, Witt, Sco	Drag and Drop Objects	ExecuTrain	25-Apr
<b>May-01</b>	ING Building	10-May	Queries 301, Back To The Basics - Part 5"	Sco Scofield	A Myriad of Access Options - The Options Dialog	ExecuTrain	23-May
<b>June-01</b>	Red Rocks Credit Union	14-Jun	Macro Madness	Chris Kaiser	Tab Control Gotchas	ExecuTrain	27-Jun
<b>July-01</b>	ING Building	12-Jul	Access Data Projects and the Microsoft Data Engine	Sco Scofield	XML for Dummies Mel Harris	ExecuTrain	25-Jul
<b>August-01</b>	Red Rocks Credit Union	9-Aug	Access 2002	<b>Carl Dubler</b>	Cascading Combo Boxes	ExecuTrain	22-Aug
<b>September-01</b>	ING Building	13-Sep	Forms Part 2	Chuck Robbins	XML for Smarties Mel Harris	ExecuTrain	26-Sep
<b>October-01</b>	Red Rocks Credit Union	11-Oct	Stupid Report Tricks	Jim Pilcher	Hidden Columns in Combo Boxes	ExecuTrain	24-Oct
<b>November-01</b>	ING Building	8-Nov	Beyond Macros	Geller or Sco		ExecuTrain	28-Nov
<b>December-01</b>	Red Rocks Credit Union	13-Dec	Holiday Party	TBD			none
<b>January-02</b>	ING Building	10-Jan	Topic Open	TBD		ExecuTrain	23-Jan
<b>February-02</b>	Red Rocks Credit Union	14-Feb	Table Design Topics	TBD		ExecuTrain	27-Feb
<b>March-02</b>	ING Building	14-Mar	Queries	TBD		ExecuTrain	27-Mar
<b>April-02</b>	Red Rocks Credit Union	11-Apr	So You Want To Be An Access Consultant?	Pilcher et al		ExecuTrain	24-Apr