Introduction to SAS SQL Language Proc SQL

Dr. Majed Al-Ghandour

Ph.D., P.E., CNA, MCP, MCSD, MCT, CPM

malghandour@ncdot.gov

NCDOT- Planning and Programming Division

For

State of North Carolina SAS Users Group

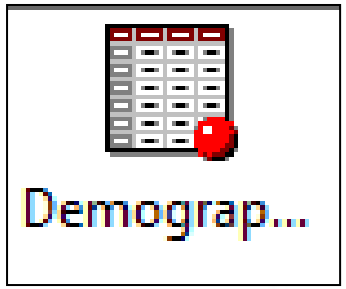
November 15, 2016

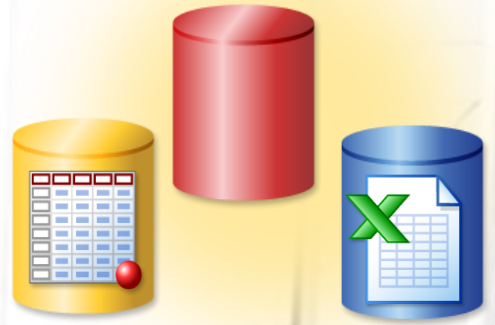
Outline

- SQL (Structured Query Language)
- Proc SQL
- Examples
- Advanced Topics… JOIN

What is SQL?

- It is standard query language for databases.
- In SAS:
  - SAS Data set →Table
  - Observations →Row
  - Variable →Column





VIEWTABLE: Learn.Demographic

	Subj	DOB	Gender	Name
1	001	10/15/1960	M	Friedman
2	002	08/01/1955	M	Stern
3	003	12/25/1988	F	McGoldrick
4	005	05/28/1949	F	Chien

SQL Types

1. Report →SELECT
2. Table →CREATE TABLE
3. Proc SQL View →CREATEVIEW

Why SQL?

Performance to retrieve Data from and SAS Tables

Using calculated values, adding character constants

Rich built-in functions, sum, avg, count, max

Add or modify, delete data values in a table

Add, modify, or drop a columns in a table

Create tables and views

Join multiple tables, subqueries

Generate reports using where, like, group by, summarizing, etc.

Using case expressions

Multi-Threads SQL

Proc SQL Syntax

Syntax:

```
Proc SQL;
….
Quit;
```

- PROC SQL FEEDBACK;

```
Select * from
sasuser.tablename;
QUIT;
```

- Proc SQL Step contains TWO Statements:
  - 1- Proc SQL
  - 2- SELECT
- SELECT statement contains clauses.
- It does not need Run statement.

FEEDBACK is the debugging option

INOBS and OUTOBS – These options control the number of rows to be read in or write out

The conditions that control the filtering of the query exist in a rough (if not quite exact) hierarchal order:

```
SELECT   *
FROM Table
[WHERE
GROUP BY
HAVING ]
```

Examples

```
/* For State of North Carolina SAS Users Group Demo
*/

title "Local Car Dealer Inventory at NewCars.com";
title2 "Today: &sysdate9 - Time: &systime ";
title3 "Print all Records";
proc print data=sashelp.cars;
run;
title3; *reset;
```

Results Viewer - SAS Output

Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
Print all Records

Obs	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
1	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6	265	17	23	4451	106	189
2	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4	200	24	31	2778	101	172
3	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4	200	22	29	3230	105	183
4	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6	270	20	28	3575	108	186
5	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6	225	18	24	3880	115	197
6	Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	$46,100	$41,100	3.5	6	225	18	24	3893	115	197
7	Acura	NSX coupe 2dr manual S	Sports	Asia	Rear	$89,765	$79,978	3.2	6	290	17	24	3153	100	174
8	Audi	A4 1.8T 4dr	Sedan	Europe	Front	$25,940	$23,508	1.8	4	170	22	31	3252	104	179
9	Audi	A41.8T convertible 2dr	Sedan	Europe	Front	$35,940	$32,506	1.8	4	170	23	30	3638	105	180
10	Audi	A4 3.0 4dr	Sedan	Europe	Front	$31,840	$28,846	3.0	6	220	20	28	3462	104	179
11	Audi	A4 3.0 Quattro 4dr manual	Sedan	Europe	All	$33,430	$30,366	3.0	6	220	17	26	3583	104	179
12	Audi	A4 3.0 Quattro 4dr auto	Sedan	Europe	All	$34,480	$31,388	3.0	6	220	18	25	3627	104	179
13	Audi	A6 3.0 4dr	Sedan	Europe	Front	$36,640	$33,129	3.0	6	220	20	27	3561	109	192
14	Audi	A6 3.0 Quattro 4dr	Sedan	Europe	All	$39,640	$35,992	3.0	6	220	18	25	3880	109	192
15	Audi	A4 3.0 convertible 2dr	Sedan	Europe	Front	$42,490	$38,325	3.0	6	220	20	27	3814	105	180
16	Audi	A4 3.0 Quattro convertible 2dr	Sedan	Europe	All	$44,240	$40,075	3.0	6	220	18	25	4013	105	180
17	Audi	A6 2.7 Turbo Quattro 4dr	Sedan	Europe	All	$42,840	$38,840	2.7	6	250	18	25	3836	109	192
18	Audi	A6 4.2 Quattro 4dr	Sedan	Europe	All	$49,690	$44,936	4.2	8	300	17	24	4024	109	193
19	Audi	A8 L Quattro 4dr	Sedan	Europe	All	$69,190	$64,740	4.2	8	330	17	24	4399	121	204
20	Audi	S4 Quattro 4dr	Sedan	Europe	All	$48,040	$43,556	4.2	8	340	14	20	3825	104	179
21	Audi	RS 6 4dr	Sports	Europe	Front	$84,600	$76,417	4.2	8	450	15	22	4024	109	191
22	Audi	TT 1.8 convertible 2dr (coupe)	Sports	Europe	Front	$35,940	$32,512	1.8	4	180	20	28	3131	95	159
23	Audi	TT 1.8 Quattro 2dr (convertible)	Sports	Europe	All	$37,390	$33,891	1.8	4	225	20	28	2921	96	159
24	Audi	TT 3.2 coupe 2dr (convertible)	Sports	Europe	All	$40,590	$36,739	3.2	6	250	21	29	3351	96	159
25	Audi	A6 3.0 Avant Quattro	Wagon	Europe	All	$40,840	$37,060	3.0	6	220	18	25	4035	109	192
26	Audi	S4 Avant Quattro	Wagon	Europe	All	$49,090	$44,446	4.2	8	340	15	21	3936	104	179
27	BMW	X3 3.0i	SUV	Europe	All	$37,000	$33,873	3.0	6	225	16	23	4023	110	180
28	BMW	X5 4.4i	SUV	Europe	All	$52,195	$47,720	4.4	8	325	16	22	4824	111	184
29	BMW	325i 4dr	Sedan	Europe	Rear	$28,495	$26,155	2.5	6	184	20	29	3219	107	176

```
title3 "First SQL";
proc sql;
    select Make,
           Model,
           MSRP,
           Invoice,
           Horsepower
    from sashelp.cars
    where Horsepower gt 200;
quit;
title3; *reset;
```

Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
First SQL

Make	Model	MSRP	Invoice	Horsepower
Acura	MDX	$36,945	$33,337	265
Acura	TL 4dr	$33,195	$30,299	270
Acura	3.5 RL 4dr	$43,755	$39,014	225
Acura	3.5 RL w/Navigation 4dr	$46,100	$41,100	225
Acura	NSX coupe 2dr manual S	$89,765	$79,978	290
Audi	A4 3.0 4dr	$31,840	$28,846	220
Audi	A4 3.0 Quattro 4dr manual	$33,430	$30,366	220
Audi	A4 3.0 Quattro 4dr auto	$34,480	$31,388	220
Audi	A6 3.0 4dr	$36,640	$33,129	220
Audi	A6 3.0 Quattro 4dr	$39,640	$35,992	220
Audi	A4 3.0 convertible 2dr	$42,490	$38,325	220
Audi	A4 3.0 Quattro convertible 2dr	$44,240	$40,075	220
Audi	A6 2.7 Turbo Quattro 4dr	$42,840	$38,840	250
Audi	A6 4.2 Quattro 4dr	$49,690	$44,936	300
Audi	A8 L Quattro 4dr	$69,190	$64,740	330
Audi	S4 Quattro 4dr	$48,040	$43,556	340
Audi	RS 6 4dr	$84,600	$76,417	450
Audi	TT 1.8 Quattro 2dr (convertible)	$37,390	$33,891	225
Audi	TT 3.2 coupe 2dr (convertible)	$40,590	$36,739	250
Audi	A6 3.0 Avant Quattro	$40,840	$37,060	220
Audi	S4 Avant Quattro	$49,090	$44,446	340
BMW	X3 3.0i	$37,000	$33,873	225
BMW	X5 4.4i	$52,195	$47,720	325
BMW	330i 4dr	$35,495	$32,525	225
BMW	330Ci 2dr	$36,995	$33,890	225
BMW	330xi 4dr	$37,245	$34,115	225
BMW	330Ci convertible 2dr	$44,295	$40,530	225
BMW	530i 4dr	$44,995	$41,170	225
BMW	545iA 4dr	$54,995	$50,270	325
BMW	745i 4dr	$69,195	$63,190	325

Horsepower gt 200;

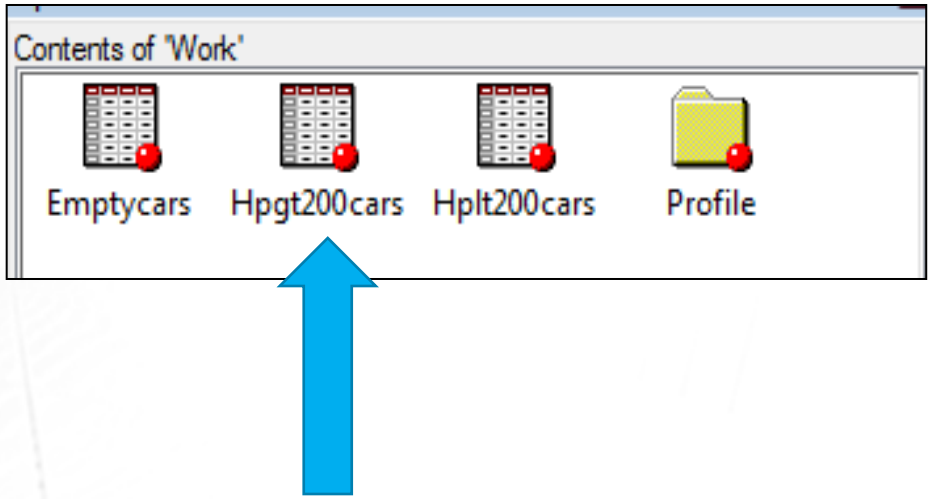
```
title3 "SQL: Renaming variables, Format=, Label=,
order by";
proc sql;
    select Make as Car_Make,
           Model as Car_Model,
           MSRP label ='Manufacturers Suggested Retail
Price',
           MSRP-Invoice as Car_Dealer_Margin format=
Dollar10.0,
            Horsepower
    from sashelp.cars
    where Horsepower gt 200
order by Horsepower desc;
quit;
title3; *reset;
```

Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
SQL: Renaming variables, Format=, Label=, order by

Car_Make	Car_Model	Manufacturers Suggested Retail Price	Car_Dealer_Margin	Horsepower
Dodge	Viper SRT-10 convertible 2dr	$81,795	$7,344	500
Mercedes-Benz	SL600 convertible 2dr	$126,670	$8,816	493
Mercedes-Benz	SL55 AMG 2dr	$121,770	$8,382	493
Mercedes-Benz	CL600 2dr	$128,420	$8,820	493
Porsche	911 GT2 2dr	$192,465	$18,905	477
Audi	RS 6 4dr	$84,600	$8,183	450
Volkswagen	Phaeton W12 4dr	$75,000	$5,870	420
Jaguar	XJR 4dr	$74,995	$6,689	390
Jaguar	XKR coupe 2dr	$81,995	$7,319	390
Jaguar	S-Type R 4dr	$63,120	$5,621	390
Jaguar	XKR convertible 2dr	$86,995	$7,769	390
Chevrolet	Corvette convertible 2dr	$51,535	$6,342	350
Chevrolet	Corvette 2dr	$44,535	$5,467	350
Mercedes-Benz	C32 AMG 4dr	$52,120	$3,598	349
Mercedes-Benz	SLK32 AMG 2dr	$56,170	$3,881	349
Cadillac	Escalade EXT	$52,975	$4,434	345
Pontiac	GTO 2dr	$33,500	$2,790	340
Audi	S4 Quattro 4dr	$48,040	$4,484	340
Porsche	Cayenne S	$56,665	$6,800	340
Infiniti	M45 4dr	$42,845	$4,053	340
Audi	S4 Avant Quattro	$49,090	$4,644	340
Infiniti	Q45 Luxury 4dr	$52,545	$4,970	340
Volkswagen	Phaeton 4dr	$65,000	$5,088	335
BMW	M3 coupe 2dr	$48,195	$4,025	333
BMW	M3 convertible 2dr	$56,595	$4,780	333
Audi	A8 L Quattro 4dr	$69,190	$4,450	330
BMW	745i 4dr	$69,195	$6,005	325
BMW	545iA 4dr	$54,995	$4,725	325

```
*This sql statement will create a table HPgt200Cars
from cars;
title3 "SQL: Create a table HPgt200Cars from cars";
proc sql;
    create table work.HPgt200Cars as
   select Make as Car_Make,
          Model as Car_Model,
          MSRP label ='Manufacturers Suggested Retail
Price',
           MSRP-Invoice as Car_Dealer_Margin format=
Dollar10.0,
           Horsepower
    from sashelp.cars
    where Horsepower gt 200
order by Horsepower desc;
quit;
```

```
proc print data=work.HPgt200Cars;
run;
title3; *reset;
```

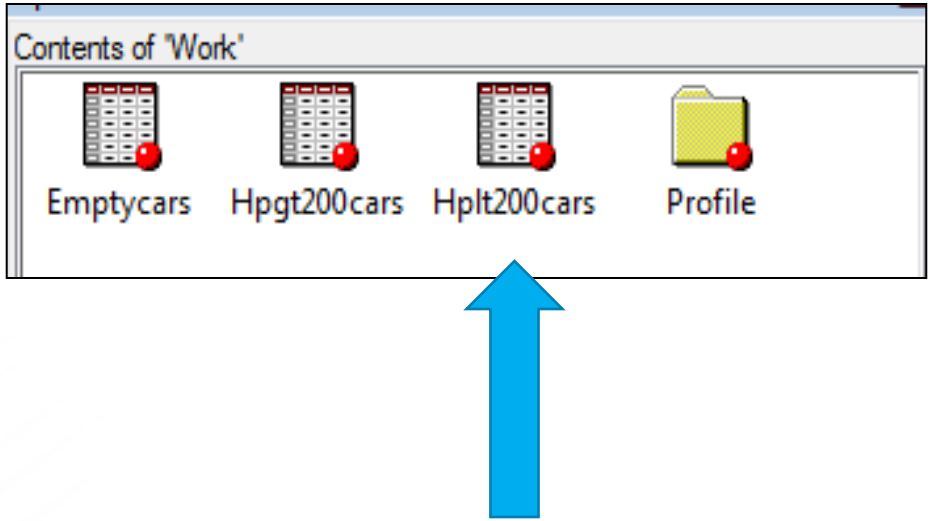


Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
SQL: Create a table HPgt200Cars from cars

Obs	Car_Make	Car_Model	MSRP	Car_Dealer_Margin	Horsepower
1	Dodge	Viper SRT-10 convertible 2dr	$81,795	$7,344	500
2	Mercedes-Benz	SL600 convertible 2dr	$126,670	$8,816	493
3	Mercedes-Benz	SL55 AMG 2dr	$121,770	$8,382	493
4	Mercedes-Benz	CL600 2dr	$128,420	$8,820	493
5	Porsche	911 GT2 2dr	$192,465	$18,905	477
6	Audi	RS 6 4dr	$84,600	$8,183	450
7	Volkswagen	Phaeton W12 4dr	$75,000	$5,870	420
8	Jaguar	XJR 4dr	$74,995	$6,689	390
9	Jaguar	XKR coupe 2dr	$81,995	$7,319	390
10	Jaguar	S-Type R 4dr	$63,120	$5,621	390
11	Jaguar	XKR convertible 2dr	$86,995	$7,769	390
12	Chevrolet	Corvette convertible 2dr	$51,535	$6,342	350
13	Chevrolet	Corvette 2dr	$44,535	$5,467	350
14	Mercedes-Benz	C32 AMG 4dr	$52,120	$3,598	349
15	Mercedes-Benz	SLK32 AMG 2dr	$56,170	$3,881	349
16	Cadillac	Escalade EXT	$52,975	$4,434	345
17	Pontiac	GTO 2dr	$33,500	$2,790	340
18	Audi	S4 Quattro 4dr	$48,040	$4,484	340
19	Porsche	Cayenne S	$56,665	$6,800	340
20	Infiniti	M45 4dr	$42,845	$4,053	340
21	Audi	S4 Avant Quattro	$49,090	$4,644	340
22	Infiniti	Q45 Luxury 4dr	$52,545	$4,970	340
23	Volkswagen	Phaeton 4dr	$65,000	$5,088	335
24	BMW	M3 coupe 2dr	$48,195	$4,025	333
25	BMW	M3 convertible 2dr	$56,595	$4,780	333
26	Audi	A8 L Quattro 4dr	$69,190	$4,450	330
27	BMW	745i 4dr	$69,195	$6,005	325
28	BMW	545iA 4dr	$54,995	$4,725	325

```
*This sql statement will create a table HPlt200Cars
from cars;
title3 "SQL: Create a table HPlt200Cars from cars";
proc sql;
    create table work.HPlt200Cars as
   select Make as Car_Make,
            Model as Car_Model,
             MSRP label ='Manufacturers Suggested
Retail Price',
            MSRP-Invoice as Car_Dealer_Margin format=
Dollar10.0,
             Horsepower
    from sashelp.cars
    where Horsepower lt 200
order by Horsepower desc;
```

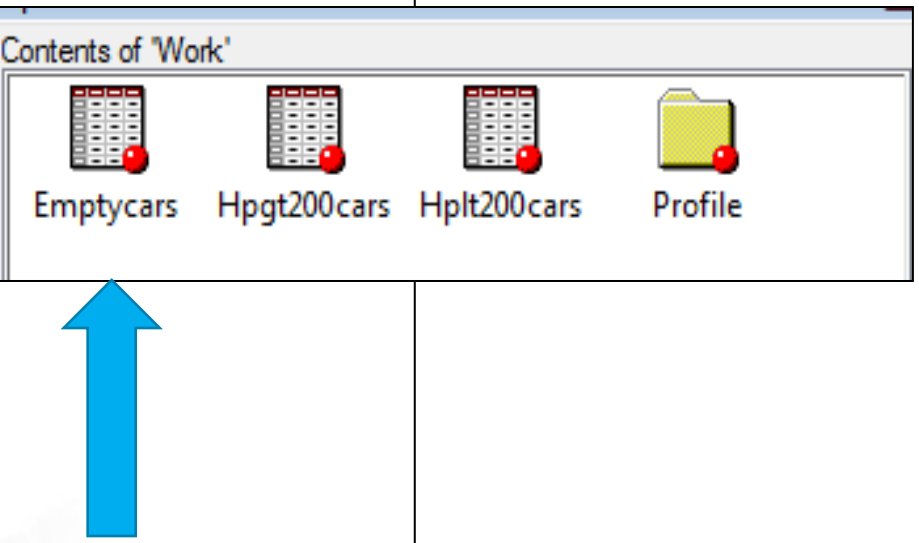
```
quit;
proc print
data=work.HPlt200Cars;
run;
title3; *reset;
```



Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
SQL: Create a table HPlt200Cars from cars

Obs	Car_Make	Car_Model	MSRP	Car_Dealer_Margin	Horsepower
1	Volvo	C70 LPT convertible 2dr	$40,565	$2,362	197
2	Mazda	RX-8 4dr automatic	$25,700	$1,906	197
3	Jeep	Grand Cherokee Laredo	$27,905	$2,219	195
4	Kia	Amanti 4dr	$26,000	$2,236	195
5	Kia	Sedona LX	$20,615	$1,215	195
6	Volvo	S80 2.5T 4dr	$37,885	$2,197	194
7	Hyundai	XG350 4dr	$24,589	$2,534	194
8	Hyundai	XG350 L 4dr	$26,189	$2,703	194
9	Isuzu	Rodeo S	$20,449	$1,188	193
10	Ford	Mustang 2dr (convertible)	$18,345	$1,402	193
11	Ford	Freestar SE	$26,930	$2,432	193
12	Mercedes-Benz	SLK230 convertible 2dr	$40,320	$2,772	192
13	Jaguar	X-Type 2.5 4dr	$29,995	$2,640	192
14	Kia	Sorento LX	$19,635	$1,005	192
15	GMC	Sonoma Crew Cab	$25,395	$2,352	190
16	Jeep	Wrangler Sahara convertible 2dr	$25,520	$2,245	190
17	GMC	Safari SLE	$25,640	$2,425	190
18	Toyota	Tundra Regular Cab V6	$16,495	$1,517	190
19	Toyota	Tundra Access Cab V6 SR5	$25,935	$2,415	190
20	Volkswagen	Passat GLX V6 4MOTION 4dr	$33,180	$2,597	190
21	Chevrolet	Astro	$26,395	$2,441	190
22	Mercedes-Benz	C230 Sport 2dr	$26,060	$1,811	189
23	Chevrolet	Venture LS	$27,020	$2,502	185
24	Pontiac	Aztekt	$21,595	$1,785	185
25	Buick	Rendezvous CX	$26,545	$2,460	185
26	Pontiac	Montana EWB	$31,370	$2,916	185
27	Oldsmobile	Silhouette GL	$28,790	$2,670	185
28	Pontiac	Montana	$23,845	$2,201	185
29	Suzuki	XL-7 EX	$23,699	$1,392	185

```
*This sql statement will create an empty table have
columns similar to cars;
title3 "SQL: Create an empty table have columns
similar to cars";
proc sql;
   create table work.EmptyCars
   like sashelp.cars;
quit;
proc contents data=work.EmptyCars;
run;
title3; *reset;
```



Results Viewer - sashtml.htm

179	Volkswagen	Jetta GLS TDI 4dr	$21,055	$1,417	100
180	Honda	Civic Hybrid 4dr manual (gas/electric)	$20,140	$1,689	93
181	Honda	Insight 2dr (gas/electric)	$19,110	$1,199	73

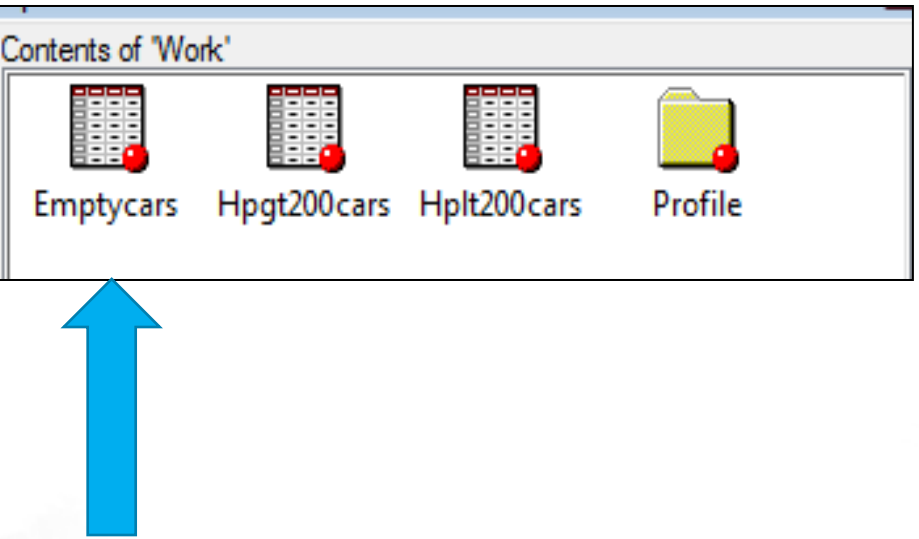
Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
SQL: Create an empty table have columns similar to cars

The CONTENTS Procedure

Data Set Name	WORK.EMPTYCARS	Observations	0
Member Type	DATA	Variables	15
Engine	V9	Indexes	0
Created	11/14/2016 13:57:25	Observation Length	152
Last Modified	11/14/2016 13:57:25	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	430
Obs in First Data Page	0
Number of Data Set Repairs	0
ExtendObsCounter	YES
Filename	C:\Users\MALGHA~1\AppData\Local\Temp\1\SAS Temporary Files_TD2264_FD279324_\emptycars.sas7bdat
Release Created	9.0401M3
Host Created	W32_7PRO

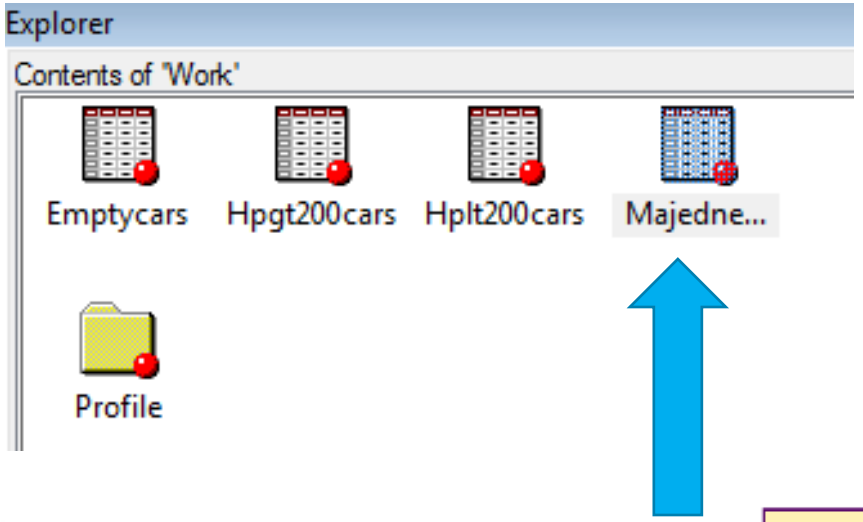
Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Label
9	Cylinders	Num	8		
5	DriveTrain	Char	5		



Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Label
9	Cylinders	Num	8		
5	DriveTrain	Char	5		
8	EngineSize	Num	8		Engine Size (L)
10	Horsepower	Num	8		
7	Invoice	Num	8	DOLLAR8.	
15	Length	Num	8		Length (IN)
11	MPG_City	Num	8		MPG (City)
12	MPG_Highway	Num	8		MPG (Highway)
6	MSRP	Num	8	DOLLAR8.	
1	Make	Char	13		
2	Model	Char	40		
4	Origin	Char	6		
3	Type	Char	8		
13	Weight	Num	8		Weight (LBS)
14	Wheelbase	Num	8		Wheelbase (IN)

```
*This sql statement will create an empty table by
defining columns;
title3 "SQL: Create an empty table by defining
columns";
proc sql;
    create table work.MajedNewEmptyCars(
      car_id char(16) label='VIN',
       destination char(5),
       Mileage num label='Miles not KM' ,
       discount float,
       datepurchased date format=date9.
);
quit;
proc contents data=work.MajedNewEmptyCars;
run;
```

Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16

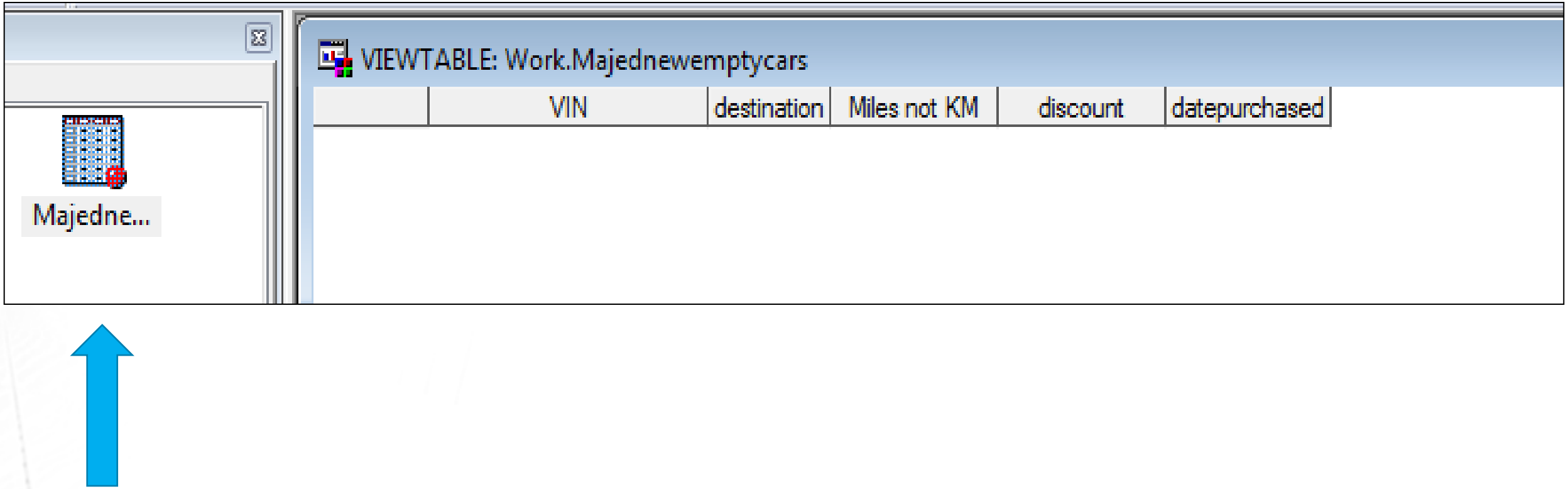


The CONTENTS Procedure

Data Set Name	WORK.MAJEDNEWEMPTYCARS	Observations	0
Member Type	DATA	Variables	5
Engine	V9	Indexes	0
Created	11/14/2016 14:01:04	Observation Length	48
Last Modified	11/14/2016 14:01:04	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	1361
Obs in First Data Page	0
Number of Data Set Repairs	0
ExtendObsCounter	YES
Filename	C:\Users\MALGHA~1\AppData\Local\Temp\1\SAS Temporary Files_TD2264_FD279324_\majednewemptycars s as 7bdat
Release Created	9.0401M3
Host Created	W32_7PRO

Alphabetic List of Variables and Attributes						
#	Variable	Type	Len	Format	Informat	Label
3	Mileage	Num	8			Miles not KM
1	car_id	Char	16			VIN
5	datepurchased	Num	8	DATE9.	DATE.	
2	destination	Char	5			
4	discount	Num	8			


Majedne...
VIEWTABLE: Work.Majednewemptycars
VIN
destination
Miles not KM
discount
datepurchased

```
proc sql;
    describe table work.MajedNewEmptyCars;
/* it will describe table on the sas log */
quit;
title3; *reset;
```

Log - (Untitled)

```
NOTE: SQL table WORK.MAJEDNEWEMPTYCARS was created like:

create table WORK.MAJEDNEWEMPTYCARS( bufsize=65536 )
  (
   car_id char(16) label='VIN',
   destination char(5),
   Mileage num label='Miles not KM',
   discount num,
   datepurchased num format=DATE9. informat=DATE.
  );

84   /* it will describe table on the sas log */
85   quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.06 seconds
      cpu time            0.01 seconds


86   title3; *reset;
```

```
proc sql;
   describe table work.MajedNewEmptyCars;
/* it will describe table on the sas log */
quit;
title3; *reset;
```

```
*This sql statement will use built in function and
group clause ;
title3 "SQL: Using built in function and Group
Clause";
proc sql;
    select Make,
           mean(Invoice) as Car_Mean_Invoice
format=dollar10.2
    from sashelp.cars
    group by Make;
quit;
title3; *reset;
```

Results Viewer - sashtml.htm

Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
SQL: Using built in function and Group Clause

group by Make;

Make	Car_Mean_Invoice
Acura	$38,590.86
Audi	$39,330.11
BMW	$39,620.65
Buick	$27,854.89
Cadillac	$46,426.88
Chevrolet	$24,060.81
Chrysler	$25,270.07
Dodge	$24,160.08
Ford	$21,953.00
GMC	$26,289.38
Honda	$19,512.76
Hummer	$45,815.00
Hyundai	$16,035.33
Infiniti	$32,880.00
Isuzu	$24,619.00
Jaguar	$56,098.42
Jeep	$22,644.67
Kia	$14,890.36
Land Rover	$41,851.00
Lexus	$38,760.00
Lincoln	$39,135.78
MINI	$16,787.00
Mazda	$20,192.45
Mercedes-Benz	$56,453.31
Mercury	$25,657.56
Mitsubishi	$21,834.77
Nissan	$22,997.47
Oldsmobile	$21,749.00
Pontiac	$22,159.64
Porsche	$73,662.86
Saab	$35,620.29
Saturn	$16,068.00
Scion	$12,910.00

```
*This sql statement will use group/having clauses;
title3 "SQL: Using Group/Having Clauses";
proc sql;
    select Make,
           mean(Invoice) as Car_Mean_Invoice
format=dollar10.2
    from sashelp.cars
    group by Make
    having mean(Invoice) lt 20000;
quit;
title3; *reset;
```

Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
SQL: Using Group/Having Clauses

Make	Car_Mean_Invoice
Honda	$19,512.76
Hyundai	$16,035.33
Kia	$14,890.36
MINI	$16,787.00
Saturn	$16,068.00
Scion	$12,910.00
Suzuki	$15,891.25

CASE-WHEN Statement Syntax

```
CASE case_operand/expression_on_case_operand
WHEN  val1 THEN result1
WHEN val2 THEN result2
ELSE result3
END;
```

```
*This sql statement will use group clause and case when;
title3 "SQL: Using Group Clause and Case When";
proc sql;
   select Make,
          mean(Invoice) as Car_Mean_Invoice format=dollar10.2,
             case
                    when 10000 le mean(Invoice) le 19990 then
'Standard Cars'
                    when 20000 le mean(Invoice) le 29990 then
'Loaded Cars'
                    when 30000 le mean(Invoice) le 80000 then
'Luxury Cars'
                    else 'out'
             end
                as Car_Features
   from sashelp.cars
   group by Make;
  quit;
title3; *reset;
```

Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
SQL: Using Group Clause and Case When

Make	Car_Mean_Invoice	Car_Features
Acura	$38,590.86	Luxury Cars
Audi	$39,330.11	Luxury Cars
BMW	$39,620.65	Luxury Cars
Buick	$27,854.89	Loaded Cars
Cadillac	$46,426.88	Luxury Cars
Chevrolet	$24,060.81	Loaded Cars
Chrysler	$25,270.07	Loaded Cars
Dodge	$24,160.08	Loaded Cars
Ford	$21,953.00	Loaded Cars
GMC	$26,289.38	Loaded Cars
Honda	$19,512.76	Standard Cars
Hummer	$45,815.00	Luxury Cars
Hyundai	$16,035.33	Standard Cars
Infiniti	$32,880.00	Luxury Cars
Isuzu	$24,619.00	Loaded Cars
Jaguar	$56,098.42	Luxury Cars
Jeep	$22,644.67	Loaded Cars
Kia	$14,890.36	Standard Cars
Land Rover	$41,851.00	Luxury Cars
Lexus	$38,760.00	Luxury Cars
Lincoln	$39,135.78	Luxury Cars
MINI	$16,787.00	Standard Cars
Mazda	$20,192.45	Loaded Cars
Mercedes-Benz	$56,453.31	Luxury Cars
Mercury	$25,657.56	Loaded Cars
Mitsubishi	$21,834.77	Loaded Cars
Nissan	$22,997.47	Loaded Cars
Oldsmobile	$21,749.00	Loaded Cars
Pontiac	$22,159.64	Loaded Cars
Porsche	$73,662.86	Luxury Cars

```
*This sql statement will use distinct function;
title3 "SQL: Using distinct function";
proc sql;
    select distinct Make as Car_Make
    from sashelp.cars
    order by 1;
quit;
title3; *reset;
```

Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
SQL: Using distinct function

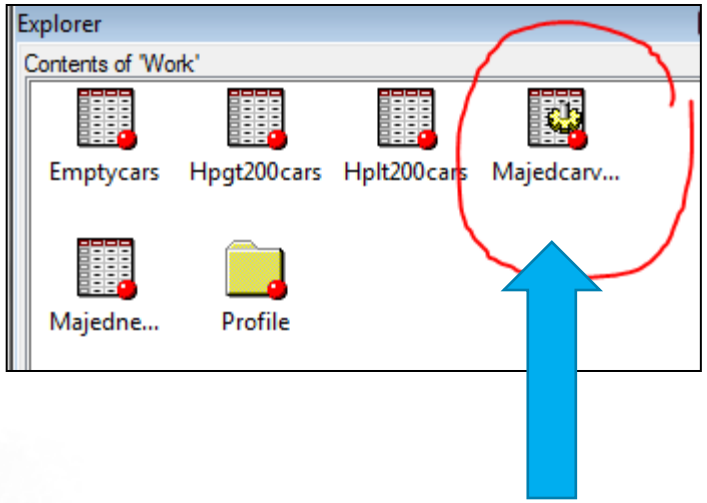
Car_Make
Acura
Audi
BMW
Buick
Cadillac
Chevrolet
Chrysler
Dodge
Ford
GMC
Honda
Hummer
Hyundai
Infiniti
Isuzu
Jaguar
Jeep
Kia
Land Rover
Lexus
Lincoln
MINI
Mazda
Mercedes-Benz
Mercury
Mitsubishi
Nissan
Oldsmobile
Pontiac

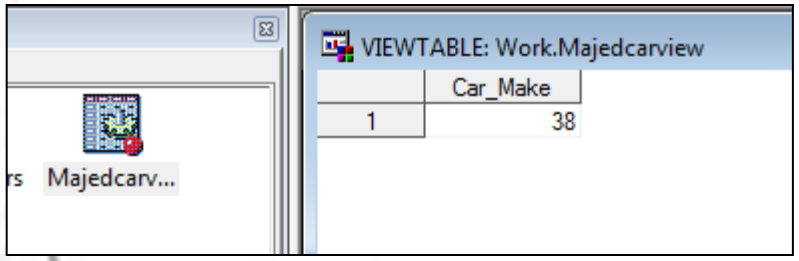
```
*This sql statement will use count function;
title3 "SQL: Using count function";
proc sql;
    select count(distinct Make) as Car_Make
    from sashelp.cars
    order by 1;
quit;
title3; *reset;
```

Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
SQL: Using count function

Car_Make
38

```
*This sql statement will create a view;
title3 "SQL: Create a view";
proc sql;
    create view work.MajedCarView as
     select count(distinct Make) as Car_Make
     from sashelp.cars
    order by 1;
quit;
proc contents data=work.MajedCarView;
run;
title3; *reset;
```





Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
SQL: Create a view

The CONTENTS Procedure

Data Set Name	WORK.MAJEDCARVIEW	Observations	.
Member Type	VIEW	Variables	1
Engine	SQLVIEW	Indexes	0
Created	11/14/2016 15:10:51	Observation Length	8
Last Modified	11/14/2016 15:10:51	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	YES
Label			
Data Representation	Default		
Encoding	Default		

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Flags
1	Car_Make	Num	8	P--

Sort Information

Sortedby	Car_Make
Validated	YES
Character Set	ANSI

```
title3 "SQL: Show the view";
proc sql;
     select *
     from work.MajedCarView;
quit;
title3; *reset;
```

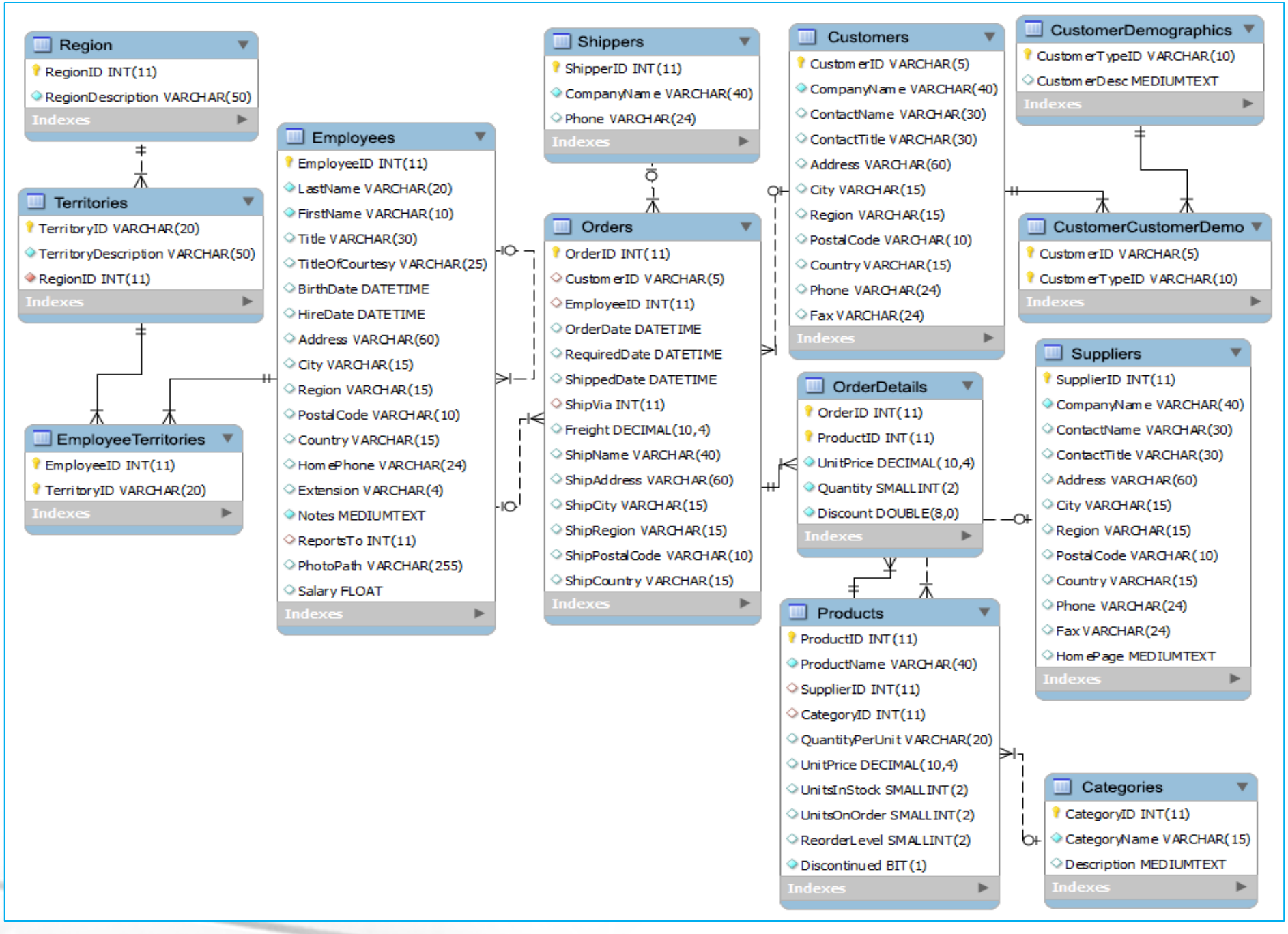
Local Car Dealer Inventory at NewCars.com
Today: 14NOV2016 - Time: 13:16
SQL: Show the view

Car_Make
38

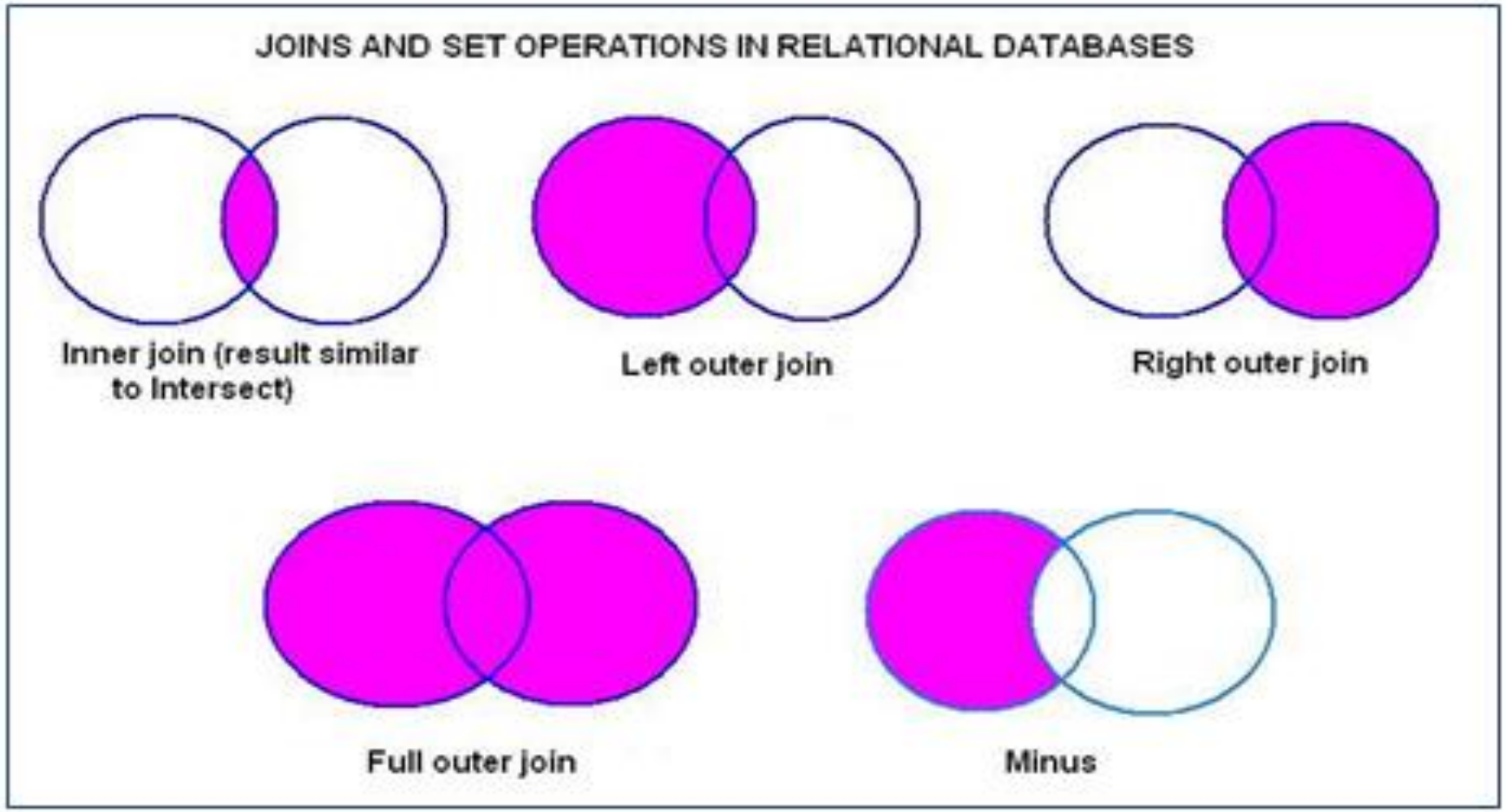
Advanced Topics

Joins

ERD


Region
RegionID INT(11)
RegionDescription VARCHAR(50)
Indexes
Territories
TerritoryID VARCHAR(20)
TerritoryDescription VARCHAR(50)
RegionID INT(11)
Indexes
EmployeeTerritories
EmployeeID INT(11)
TerritoryID VARCHAR(20)
Indexes
Employees
EmployeeID INT(11)
LastName VARCHAR(20)
FirstName VARCHAR(10)
Title VARCHAR(30)
TitleOfCourtesy VARCHAR(25)
BirthDate DATETIME
HireDate DATETIME
Address VARCHAR(60)
City VARCHAR(15)
Region VARCHAR(15)
PostalCode VARCHAR(10)
Country VARCHAR(15)
HomePhone VARCHAR(24)
Extension VARCHAR(4)
Notes MEDIUMTEXT
ReportsTo INT(11)
PhotoPath VARCHAR(255)
Salary FLOAT
Indexes
Shippers
ShipperID INT(11)
CompanyName VARCHAR(40)
Phone VARCHAR(24)
Indexes
Orders
OrderID INT(11)
CustomerID VARCHAR(5)
EmployeeID INT(11)
OrderDate DATETIME
RequiredDate DATETIME
ShippedDate DATETIME
ShipVia INT(11)
Freight DECIMAL(10,4)
ShipName VARCHAR(40)
ShipAddress VARCHAR(60)
ShipCity VARCHAR(15)
ShipRegion VARCHAR(15)
ShipPostalCode VARCHAR(10)
ShipCountry VARCHAR(15)
Indexes
Customers
CustomerID VARCHAR(5)
CompanyName VARCHAR(40)
ContactName VARCHAR(30)
ContactTitle VARCHAR(30)
Address VARCHAR(60)
City VARCHAR(15)
Region VARCHAR(15)
PostalCode VARCHAR(10)
Country VARCHAR(15)
Phone VARCHAR(24)
Fax VARCHAR(24)
Indexes
CustomerDemographics
CustomerTypeID VARCHAR(10)
CustomerDesc MEDIUMTEXT
Indexes
CustomerCustomerDemo
CustomerID VARCHAR(5)
CustomerTypeID VARCHAR(10)
Indexes
OrderDetails
OrderID INT(11)
ProductID INT(11)
UnitPrice DECIMAL(10,4)
Quantity SMALLINT(2)
Discount DOUBLE(8,0)
Indexes
Products
ProductID INT(11)
ProductName VARCHAR(40)
SupplierID INT(11)
CategoryID INT(11)
QuantityPerUnit VARCHAR(20)
UnitPrice DECIMAL(10,4)
UnitsInStock SMALLINT(2)
UnitsOnOrder SMALLINT(2)
ReorderLevel SMALLINT(2)
Discontinued BIT(1)
Indexes
Suppliers
SupplierID INT(11)
CompanyName VARCHAR(40)
ContactName VARCHAR(30)
ContactTitle VARCHAR(30)
Address VARCHAR(60)
City VARCHAR(15)
Region VARCHAR(15)
PostalCode VARCHAR(10)
Country VARCHAR(15)
Phone VARCHAR(24)
Fax VARCHAR(24)
HomePage MEDIUMTEXT
Indexes
Categories
CategoryID INT(11)
CategoryName VARCHAR(15)
Description MEDIUMTEXT
Indexes

Join Types: It may varies per Data Engines!



```
PROC SQL;
Select table1.columns, table2.columns
from
table1 INNER JOIN / LEFT JOIN / RIGHT JOIN /
FULL JOIN table2
on {table1.ID_COLUMNS =
table2.ID_COLUMNS}
QUIT;
```

```
libname learn 'F:\0-
MajedData\MajedSAS\MajedSASClass2016';
*Data set SOCIAL;
data learn.social1;
    input SS1 $11.;
datalines;
123-45-6789
001-34-9876
007-77-6767
102-43-9182
;
```

Natural Joins

```
*Data set SOCIAL2;
data learn.social2;
    input SS2 $11.;
datalines;
123-45-6789
001-43-9876
007-77-6767
485-46-1182
102-43-9188
;
```

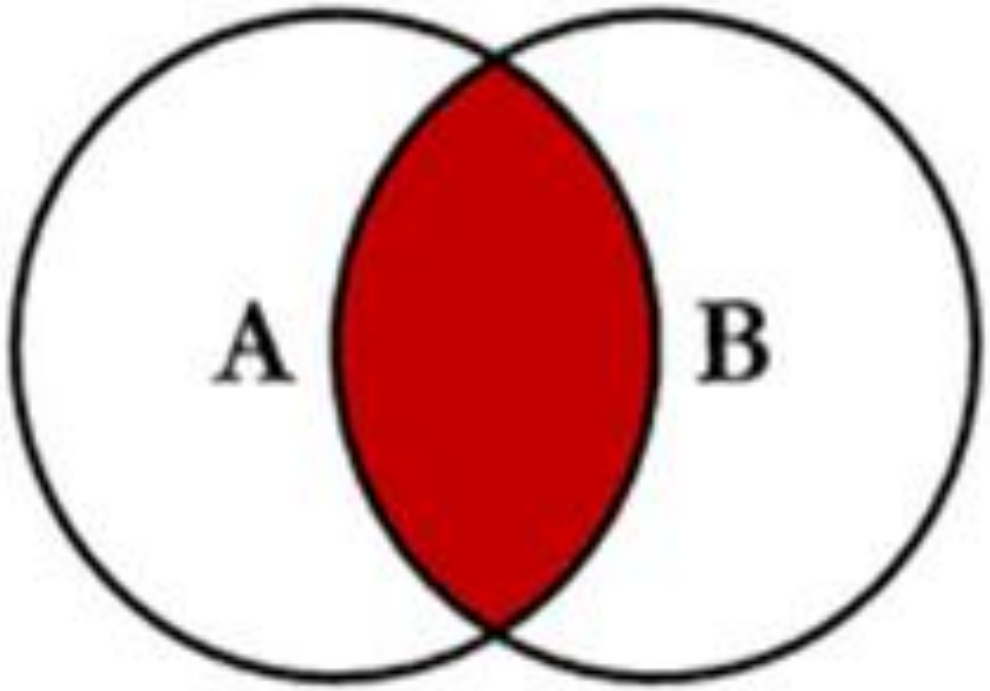
```
*This sql statement will create a table from two
dataset as natural;
proc sql;
    create table learn.social as
    select *
    from social1, social2;
quit;

proc print data=learn.social;
title "Summary Social";
run;
title;
```

Summary Social

Obs	SS1	SS2
1	123-45-6789	123-45-6789
2	123-45-6789	001-43-9876
3	123-45-6789	007-77-6767
4	123-45-6789	485-46-1182
5	123-45-6789	102-43-9188
6	001-34-9876	123-45-6789
7	001-34-9876	001-43-9876
8	001-34-9876	007-77-6767
9	001-34-9876	485-46-1182
10	001-34-9876	102-43-9188
11	007-77-6767	123-45-6789
12	007-77-6767	001-43-9876
13	007-77-6767	007-77-6767
14	007-77-6767	485-46-1182
15	007-77-6767	102-43-9188
16	102-43-9182	123-45-6789
17	102-43-9182	001-43-9876
18	102-43-9182	007-77-6767
19	102-43-9182	485-46-1182
20	102-43-9182	102-43-9188

INNER JOIN




```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```

VIEWTABLE: Learn.Health

	Subj	DOB	Weight	HR	SBP	DBP
1	001	12/25/1944	210	80	160	100
2	002	05/11/1966	102	88	122	76
3	003	08/03/2000	66	90	102	62

VIEWTABLE: Learn.Demographic

	Subj	DOB	Gender	Name
1	001	10/15/1960	M	Friedman
2	002	08/01/1955	M	Stem
3	003	12/25/1988	F	McGoldrick
4	005	05/28/1949	F	Chien

```
*Performing an INNER JOIN ;
title "Demonstrating an Inner Join (Merge)";
proc sql;
   select h.Subj as Subj_Health,
          d.Subj as Subj_Demog,
          Weight,
          Name,
          Gender
   from learn.health as h inner join
        learn.demographic as d
   on h.Subj eq d.Subj;
quit;
title;
```

VIEWTABLE: Learn.Health

	Subj	DOB	Weight	HR	SBP	DBP
1	001	12/25/1944	210	80	160	100
2	002	06/11/1966	102	88	122	76
3	003	08/03/2000	66	90	102	62

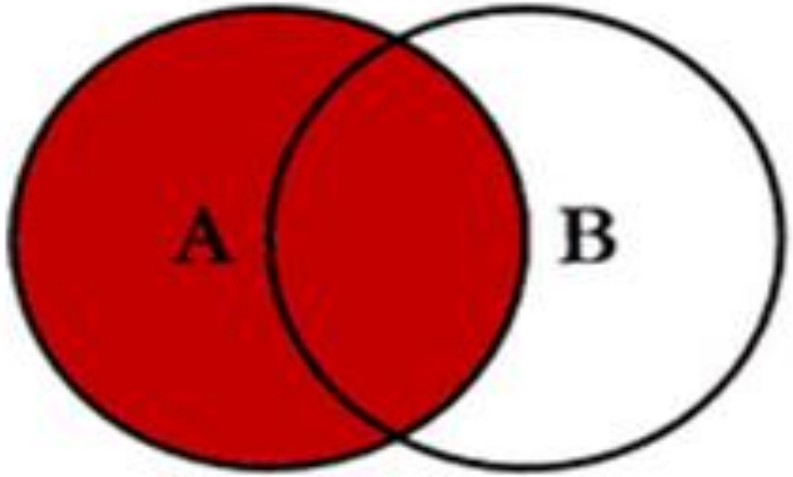
VIEWTABLE: Learn.Demographic

	Subj	DOB	Gender	Name
1	001	10/15/1960	M	Friedman
2	002	08/01/1955	M	Stern
3	003	12/25/1988	F	McGoldrick
4	005	05/28/1949	F	Chien

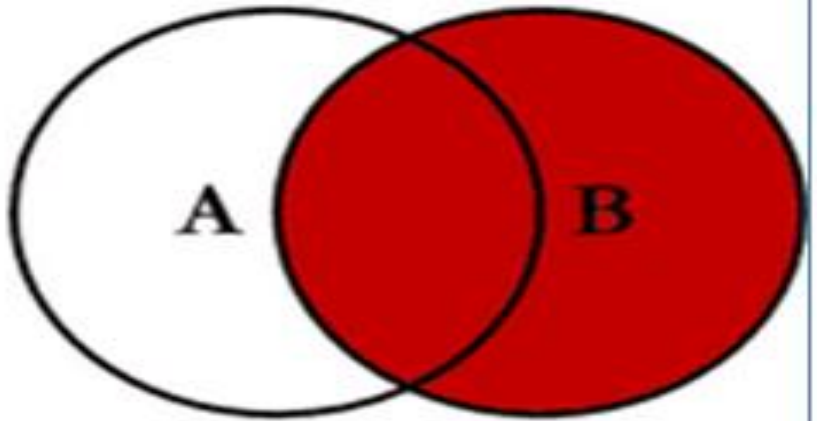
Demonstrating an Inner Join (Merge)

Subj_Health	Subj_Demog	Weight	Name	Gender
001	001	210	Friedman	M
002	002	102	Stern	M
003	003	66	McGoldrick	F

Left, Right, and Full Join




```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```




```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```

```
*Demonstrating a Left, Right, and Full Join;
proc sql;
    title "Left Join";
    select h.Subj as Subj_Health,
           d.Subj as Subj_Demog,
           Gender
    from learn.health as h left join
         learn.demographic as d
    on h.Subj eq d.Subj;
```

```
title "Right Join";
select h.Subj as Subj_Health,
          d.Subj as Subj_Demog,
          Gender
   from learn.health as h right join
        learn.demographic as d
   on h.Subj eq d.Subj;
```

```
   title "Full Join";
   select h.Subj as Subj_Health,
          d.Subj as Subj_Demog,
          Gender
   from learn.health as h full join
        learn.demographic as d
   on h.Subj eq d.Subj;
quit;
title;
```

Left Join

Subj_Health	Subj_Demog	Gender
001	001	M
002	002	M
003	003	F

Right Join

Subj_Health	Subj_Demog	Gender
001	001	M
002	002	M
003	003	F
	005	F

Full Join

Subj_Health	Subj_Demog	Gender
001	001	M
002	002	M
003	003	F
	005	F

VIEWTABLE: Learn.Health

	Subj	DOB	Weight	HR	SBP	DBP
1	001	12/25/1944	210	80	160	100
2	002	05/11/1966	102	88	122	76
3	003	08/03/2000	66	90	102	62

VIEWTABLE: Learn.Demographic

	Subj	DOB	Gender	Name
1	001	10/15/1960	M	Friedman
2	002	08/01/1955	M	Stern
3	003	12/25/1988	F	McGoldrick
4	005	05/28/1949	F	Chien

Left Join

Subj_Health	Subj_Demog	Gender
001	001	M
002	002	M
003	003	F

VIEWTABLE: Learn.Health

	Subj	DOB	Weight	HR	SBP	DBP
1	001	12/25/1944	210	80	160	100
2	002	05/11/1966	102	88	122	76
3	003	08/03/2000	66	90	102	62

VIEWTABLE: Learn.Demographic

	Subj	DOB	Gender	Name
1	001	10/15/1960	M	Friedman
2	002	08/01/1955	M	Stern
3	003	12/25/1988	F	McGoldrick
4	005	05/28/1949	F	Chien

Right Join

Subj_Health	Subj_Demog	Gender
001	001	M
002	002	M
003	003	F
	005	F

VIEWTABLE: Learn.Health

	Subj	DOB	Weight	HR	SBP	DBP
1	001	12/25/1944	210	80	160	100
2	002	05/11/1966	102	88	122	76
3	003	08/03/2000	66	90	102	62

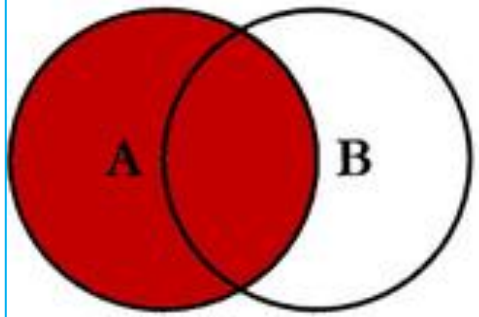
VIEWTABLE: Learn.Demographic

	Subj	DOB	Gender	Name
1	001	10/15/1960	M	Friedman
2	002	08/01/1955	M	Stern
3	003	12/25/1988	F	McGoldrick
4	005	05/28/1949	F	Chien

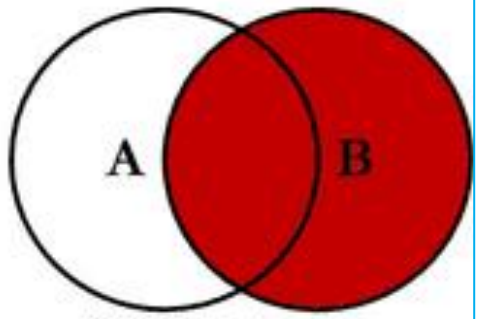
Full Join

Subj_Health	Subj_Demog	Gender
001	001	M
002	002	M
003	003	F
	005	F

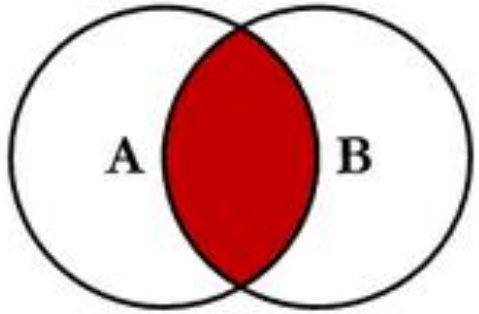
SQL JOINS



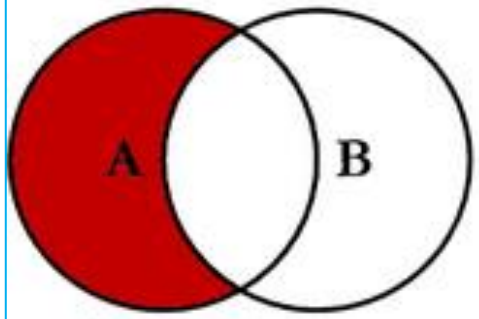

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



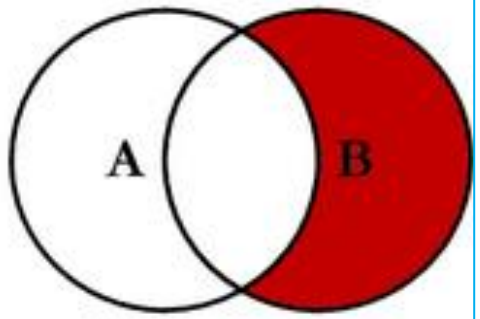

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



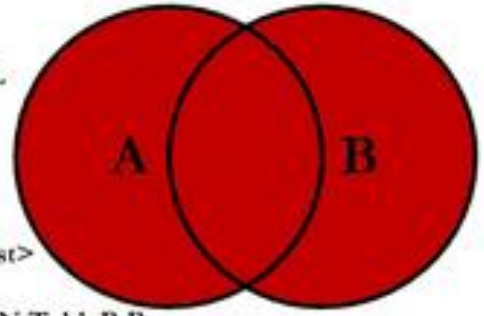

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



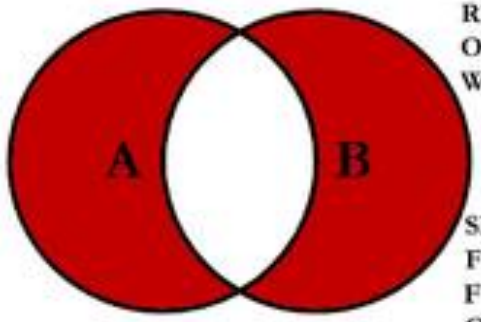

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

© C.L. Moffatt, 2008

Future Topics:

- Insert, Update, Delete!
- Altering columns in a Table
- Dropping Tables
- Combining Tables Horizontally or Vertically
- Indexing
- Multi-Threads SQL

.....

DDL

Data Definition Language (DDL) statements are used to define the database structure or schema.

CREATE - to create objects in the database

ALTER - alters the structure of the database

DROP - delete objects from the database

DML

Data Manipulation Language (DML) statements are used for managing data within schema objects.

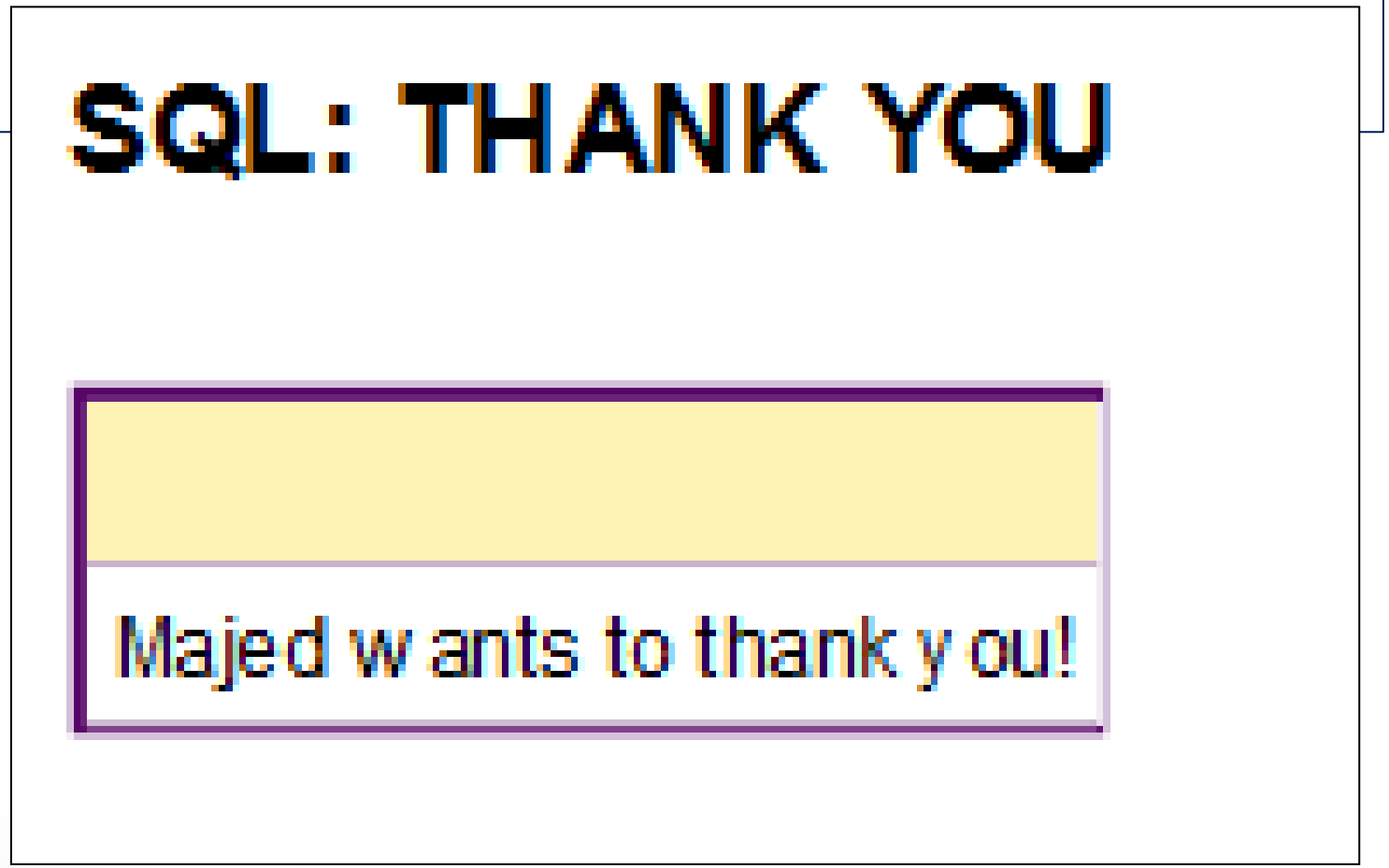
SELECT - retrieve data from the a database

INSERT - insert data into a table

UPDATE - updates existing data within a table

DELETE - deletes all records from a table, the space for the records remain

```
title3 "SQL: THANK YOU";
proc sql;
     select 'Majed wants to thank you!'
     from work.MajedCarView;
quit;
title3; *reset;
```



References and Citations

http://sas.com
https://sites.google.com/site/sasbuddy/proc-sql
http://support.sas.com/documentation/cdl/en/sqlproc/69049/PDF/default/sqlproc.pdf
Learning SAS by Example: A Programmer's Guide, By Ron Cody.