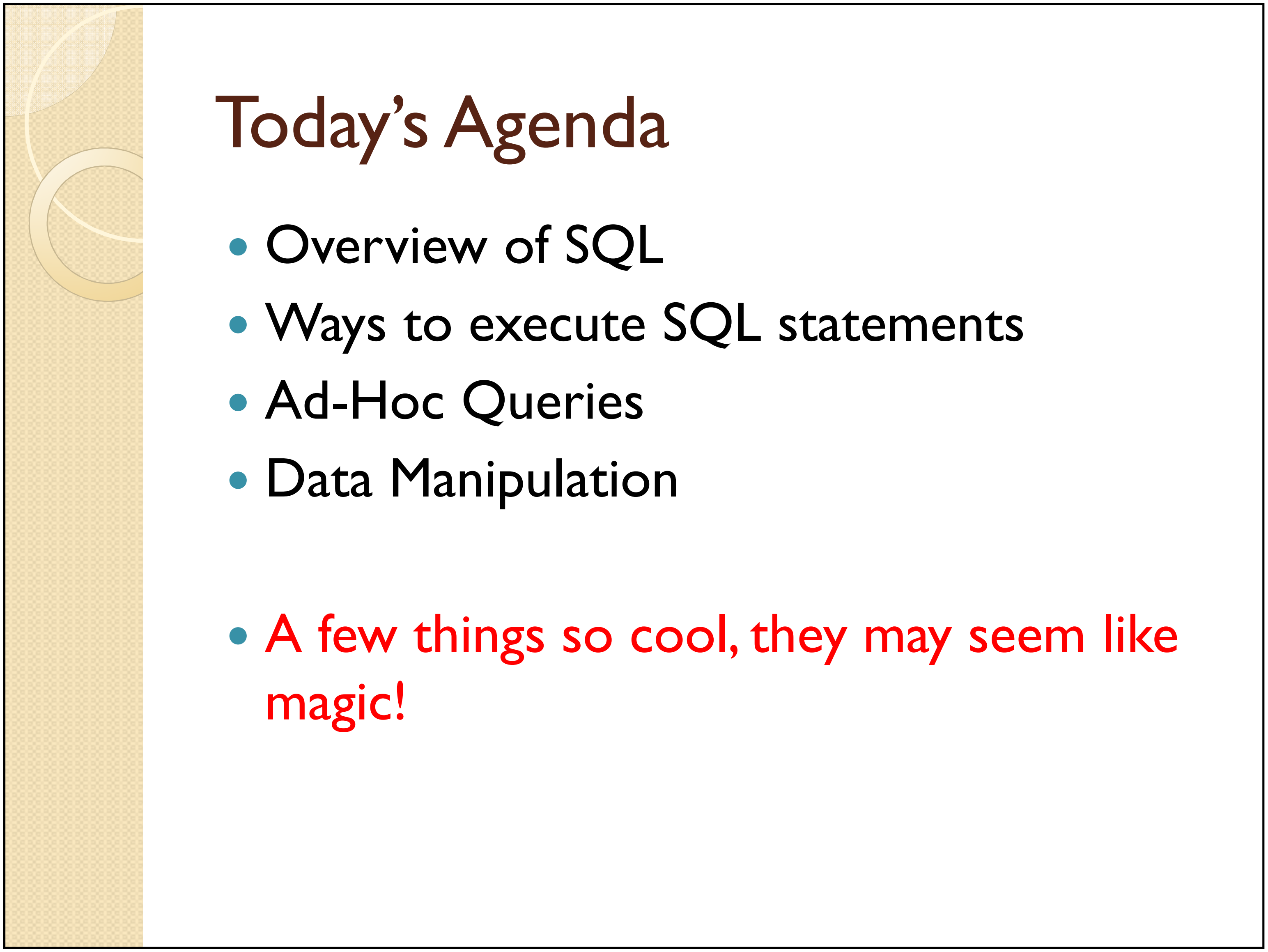
SQL Easy as Magic

Steven Wolk
Chief Technology Officer
PC Richard & Son
SWolk@PCRichard.com



Today’s Agenda

- Overview of SQL
- Ways to execute SQL statements
- Ad-Hoc Queries
- Data Manipulation

- A few things so cool, they may seem like magic!

Disclaimer

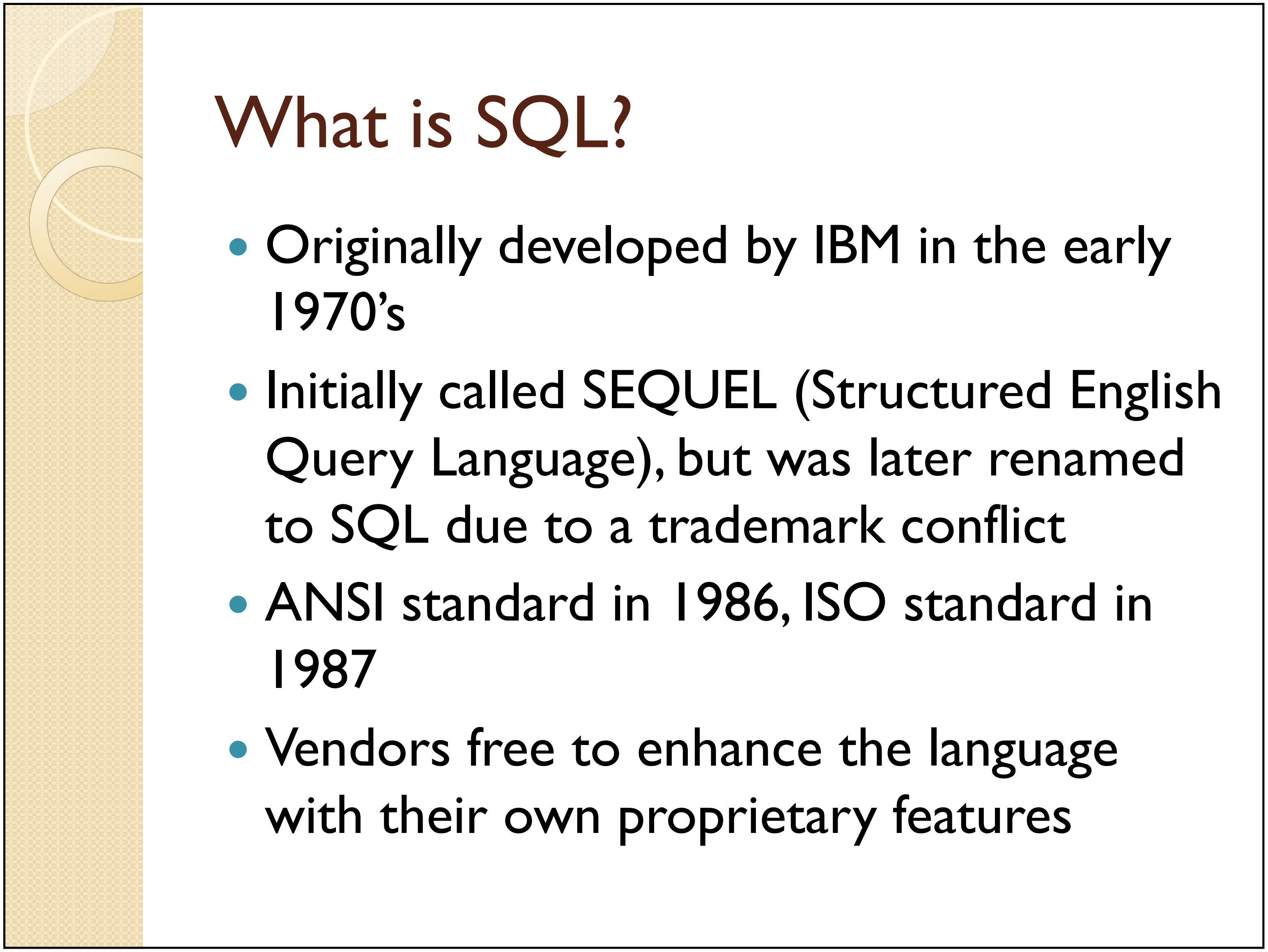
- **ALL SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY WARRANTY OF ANY NATURE WHATSOEVER. THE PROVIDER OF THIS SOFTWARE HEREBY DISCLAIMS ALL WARRANTIES, REPRESENTATIONS, AND CONDITIONS, STATUATORY OR OTHERWISE, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTY OF TITLE AND THE IMPLIED WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE PROVIDER SHALL NOT BE LIABLE FOR ANY DAMAGES ARISING FROM OR AS A RESULT OF YOUR USE OF THIS SOFTWARE. USE IT AS YOUR OWN RISK.**

Disclaimer

- **This product is meant for educational purposes only. Any resemblance to real persons, living or dead is purely coincidental. Void where prohibited. Some assembly required. Batteries not included. Contents may settle during shipment. Use only as directed. No other warranty expressed or implied. Objects in the mirror are closer than they appear. Baby tested, mother approved. Melts in your mouth, not in your hand. Past performance does not guarantee future results. Do not use while operating a motor vehicle or heavy equipment. Postage will be paid by addressee. Subject to approval. This is not an offer to sell securities. Apply only to affected area. May be too intense for some viewers. For recreational use only. All models over 18 years of age. If condition persists, consult your physician. No user-serviceable parts inside. Freshest if eaten before date on carton. Subject to change without notice. Times approximate. Simulated picture. No postage necessary if mailed in the United States. Breaking seal constitutes acceptance of agreement. For off-road use only. As seen on TV. One size fits all. Many suitcases look alike. Contains a substantial amount of non-tobacco ingredients. Colors may, in time, fade. Slippery when wet. For office use only. Not affiliated with the American Red Cross. Drop in any mailbox. Edited for television. Keep cool; process promptly. Post office will not deliver without postage. List was current at time of printing. Not responsible for direct, indirect, incidental or consequential damages resulting from any defect, error or failure to perform. At participating locations only. Penalty for private use. Substantial penalty for early withdrawal. Do not write below this line. Falling rocks. Lost ticket pays maximum rate. Your cancelled check is your receipt. Avoid contact with skin. Sanitized for your protection. Be sure each item is properly endorsed. Sign here without admitting guilt. Slightly higher west of the Mississippi. Employees and their families are not eligible. Beware of dog. Contestants have been briefed on some questions before the show. Limited time offer, call now to ensure prompt delivery. You must be present to win. No passes accepted for this engagement. No purchase necessary. Shading within a garment may occur. Use only in well-ventilated area. Keep away from fire or flame. Replace with same type. Check here if tax deductible. Some equipment shown is optional. Price does not include taxes. No Canadian coins. Not recommended for children. Prerecorded for this time zone. Reproduction strictly prohibited. No solicitors. No alcohol, dogs, or horses. Restaurant package, not for resale. List at least two alternate dates. First pull up, then pull down. Call toll free before digging. Driver does not carry cash. Some of the trademarks mentioned in this product appear for identification purposes only. Record additional transactions on back of previous stub. Decision of judges is final.**

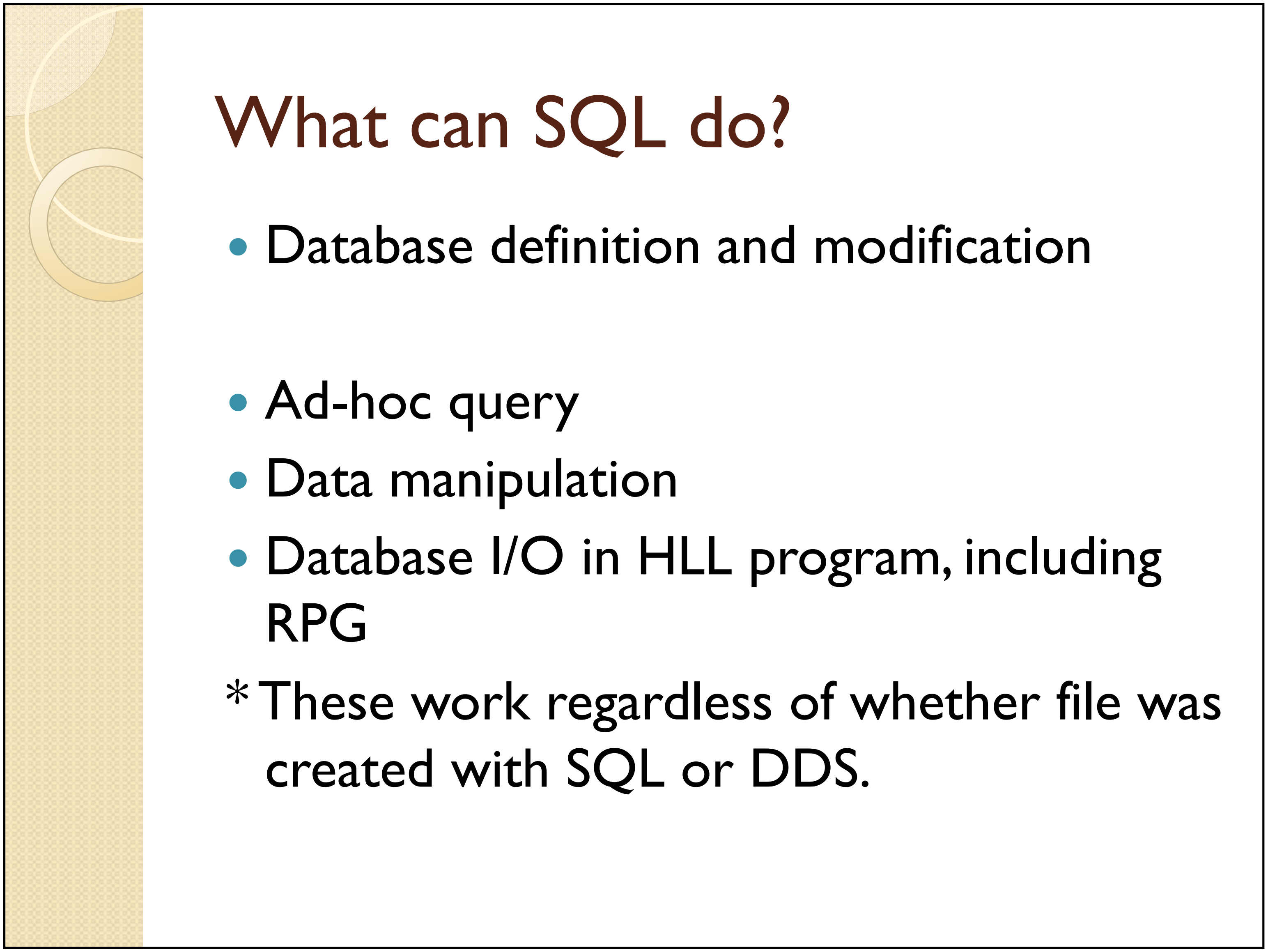
Obligatory Brag Slide

- PC Richard & Son
  - Founded in 1909 as a hardware store in Bensonhurst, Brooklyn.
  - Began transition to appliances by selling the first electric iron, and later the first washing machines.
  - Currently own and operate 65 retail showrooms throughout NY, NJ, CT, & PA.
  - Family owned and operated – 5th Generation
- Steve Wolk
  - Joined PCR in 1986
  - Became company's first CTO in 2000.



What is SQL?

- Originally developed by IBM in the early 1970's
- Initially called SEQUEL (Structured English Query Language), but was later renamed to SQL due to a trademark conflict
- ANSI standard in 1986, ISO standard in 1987
- Vendors free to enhance the language with their own proprietary features

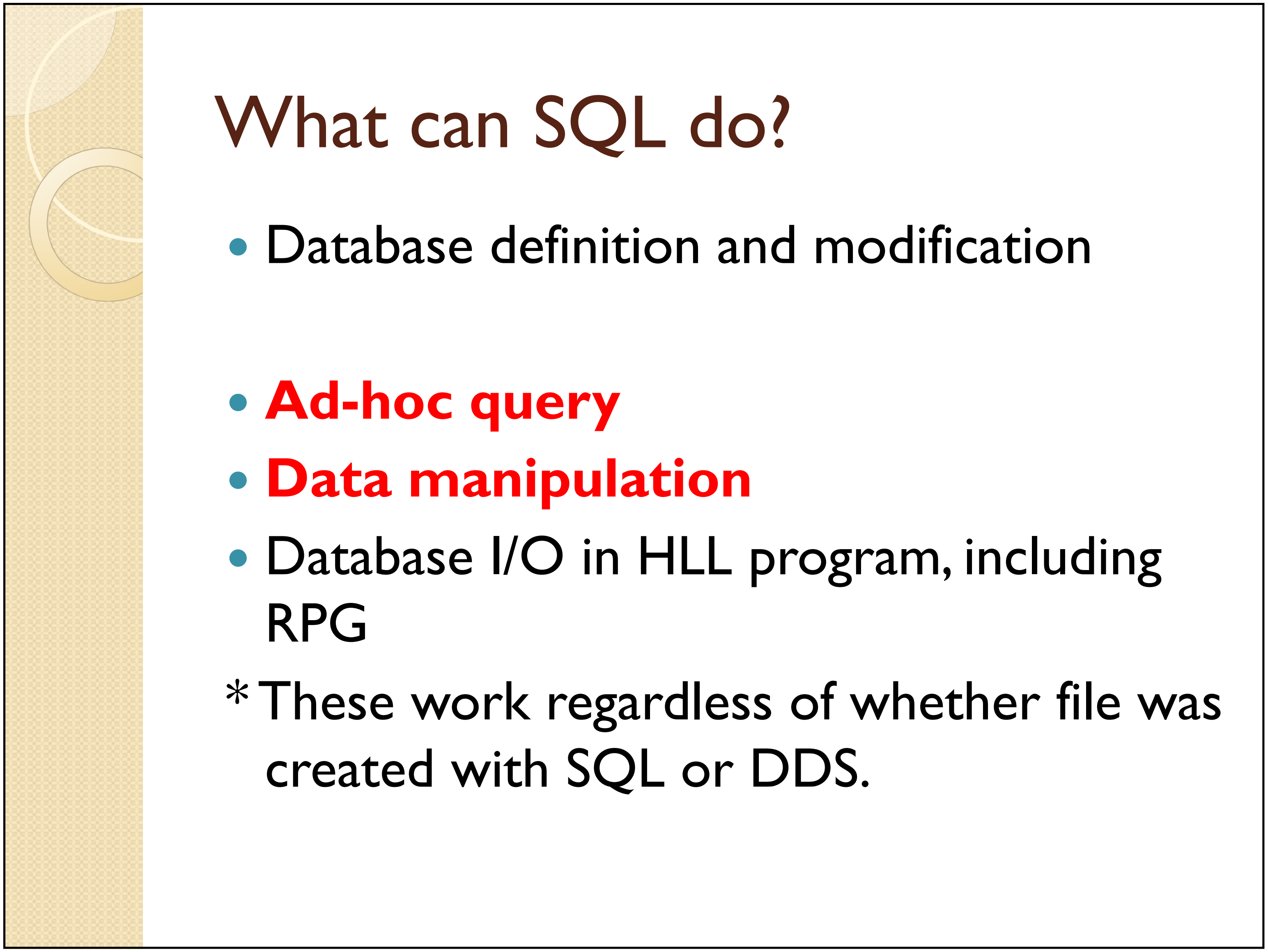


What can SQL do?

- Database definition and modification

- Ad-hoc query
- Data manipulation
- Database I/O in HLL program, including RPG

* These work regardless of whether file was created with SQL or DDS.

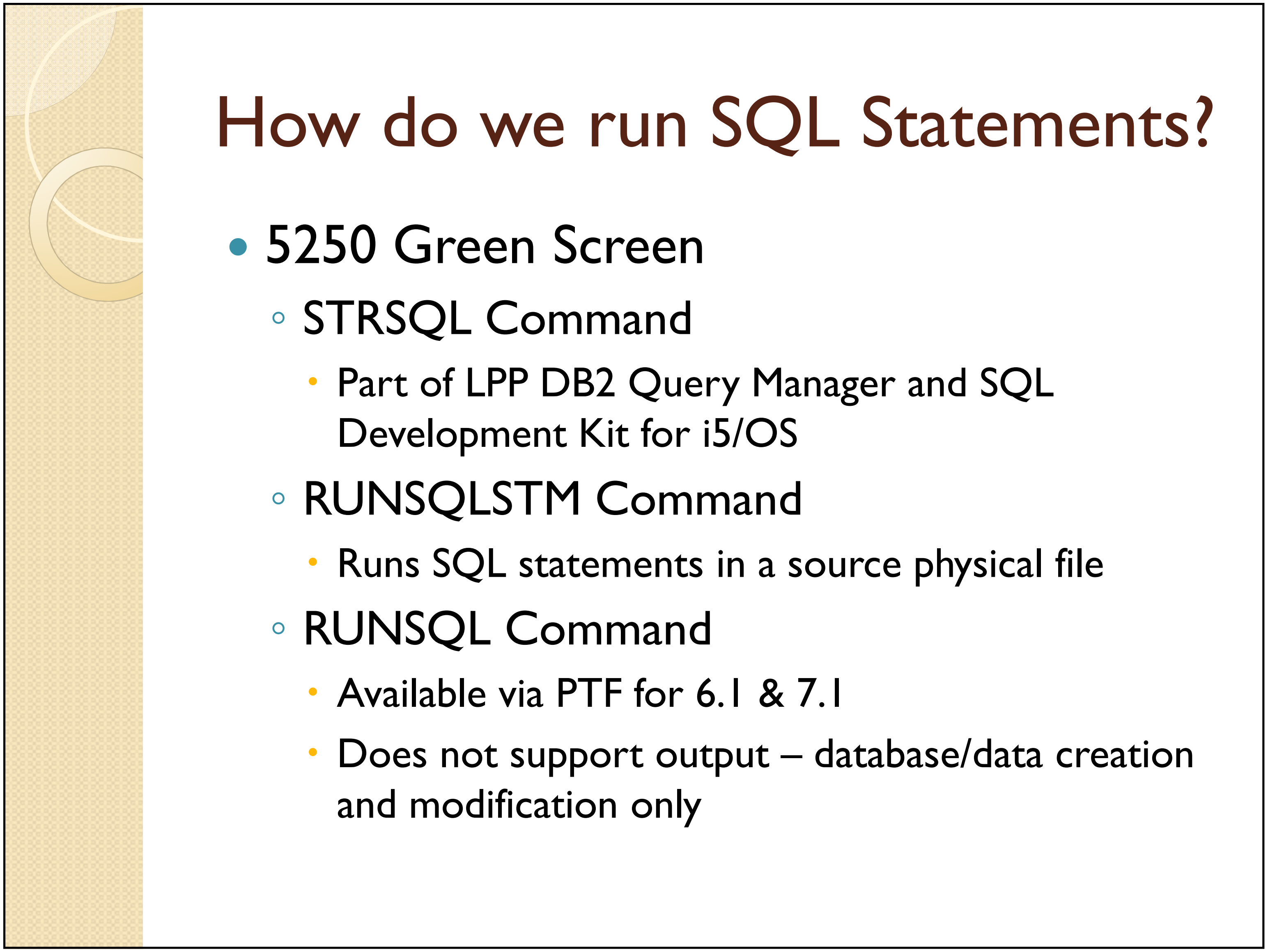


What can SQL do?

- Database definition and modification

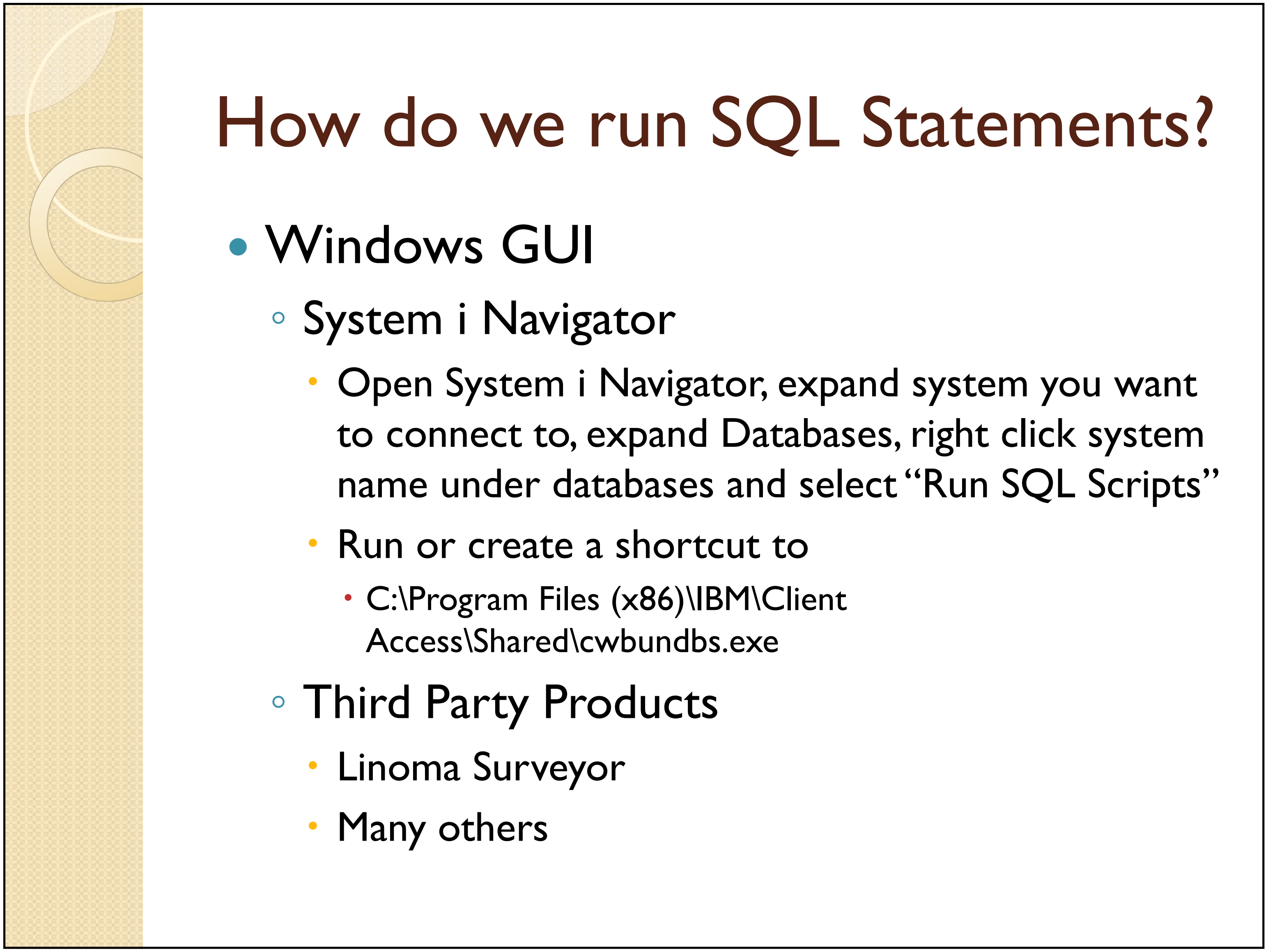
- **Ad-hoc query**
- **Data manipulation**
- Database I/O in HLL program, including RPG

*These work regardless of whether file was created with SQL or DDS.



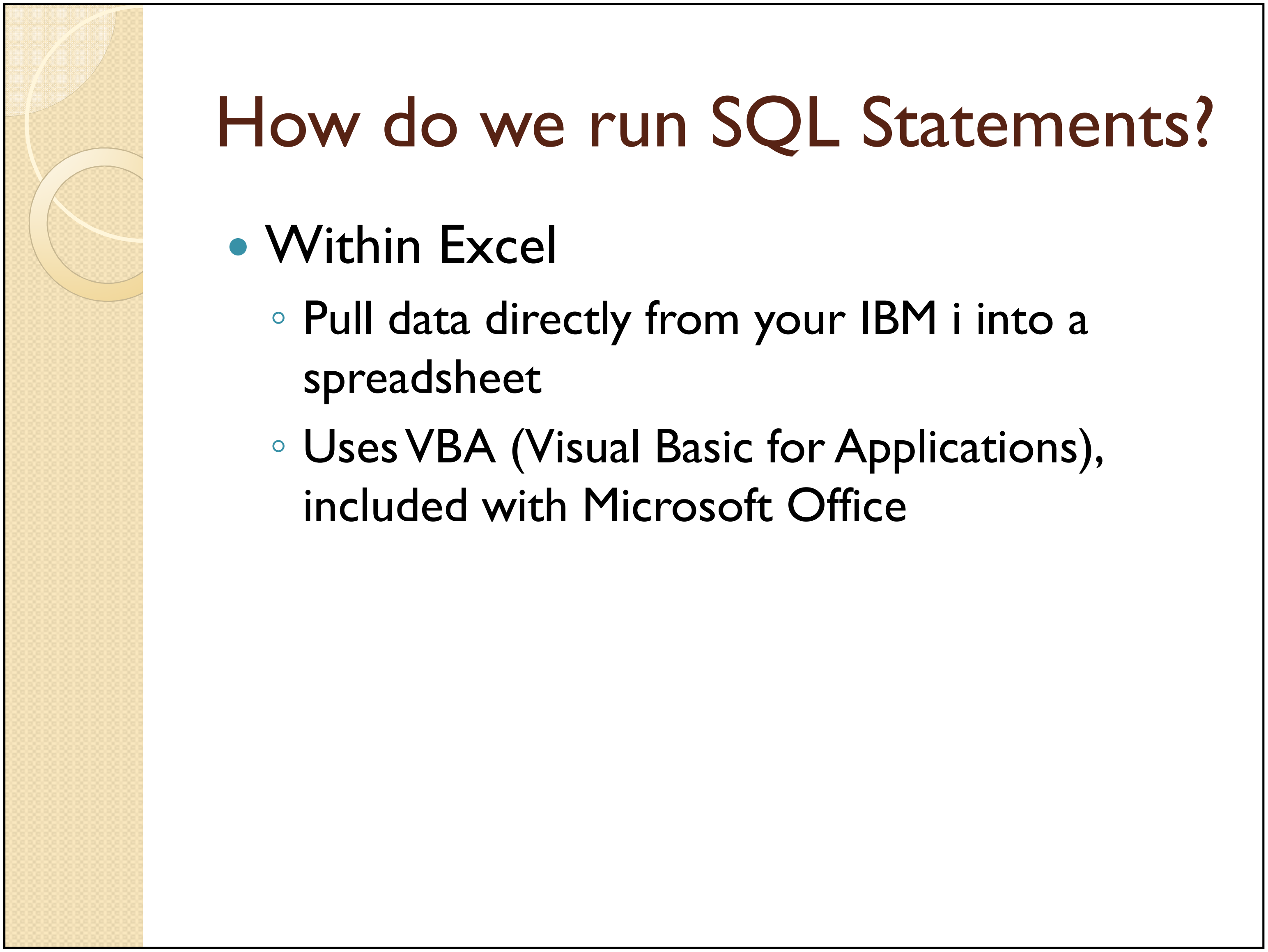
How do we run SQL Statements?

- 5250 Green Screen
  - STRSQL Command
    - Part of LPP DB2 Query Manager and SQL Development Kit for i5/OS
  - RUNSQLSTM Command
    - Runs SQL statements in a source physical file
  - RUNSQL Command
    - Available via PTF for 6.1 & 7.1
    - Does not support output – database/data creation and modification only



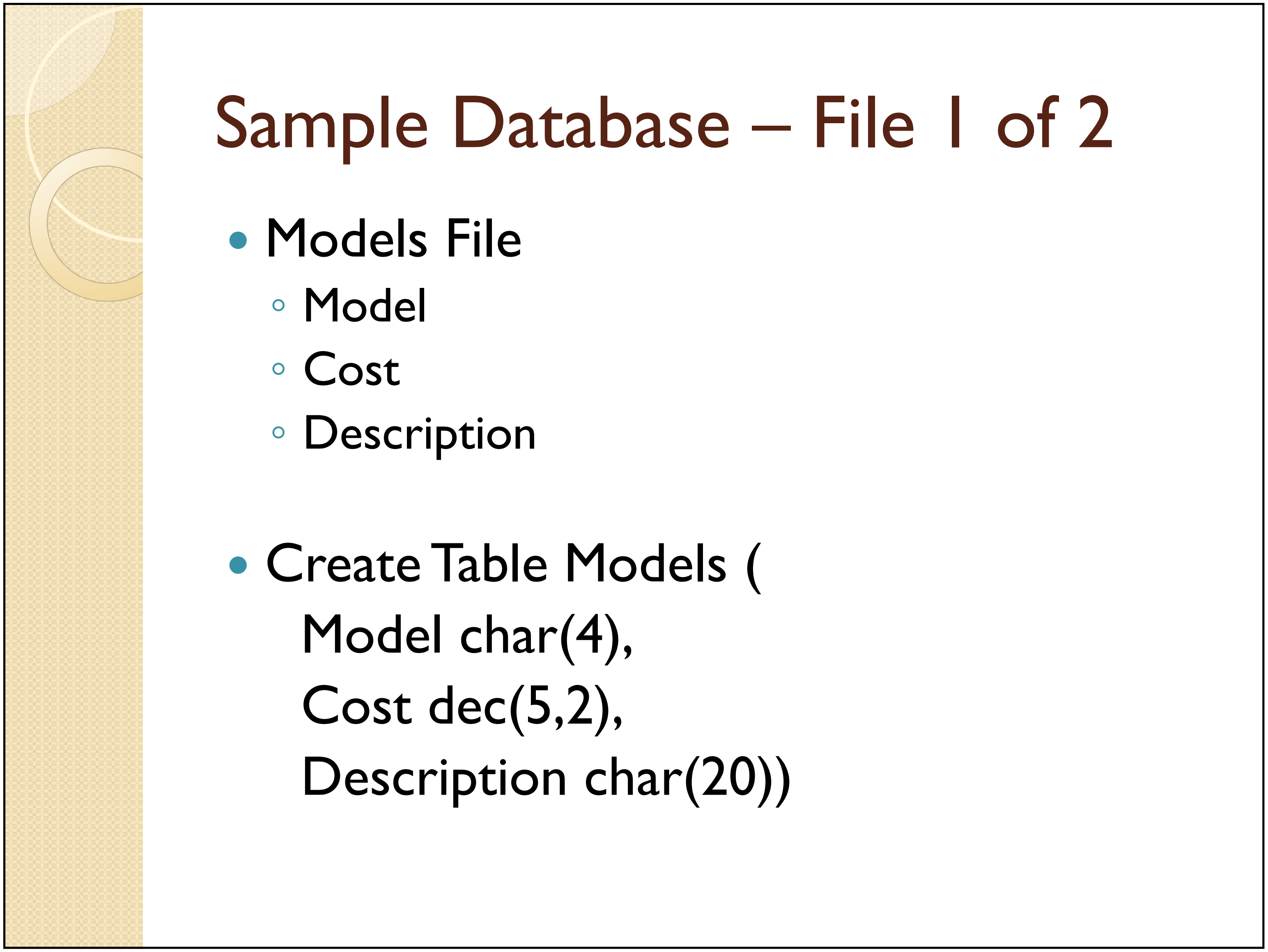
How do we run SQL Statements?

- Windows GUI
  - System i Navigator
    - Open System i Navigator, expand system you want to connect to, expand Databases, right click system name under databases and select "Run SQL Scripts"
    - Run or create a shortcut to
      - C:\Program Files (x86)\IBM\Client Access\Shared\cwbundbs.exe
  - Third Party Products
    - Linoma Surveyor
    - Many others



How do we run SQL Statements?

- Within Excel
  - Pull data directly from your IBM i into a spreadsheet
  - Uses VBA (Visual Basic for Applications), included with Microsoft Office



Sample Database – File 1 of 2

- Models File
  - Model
  - Cost
  - Description

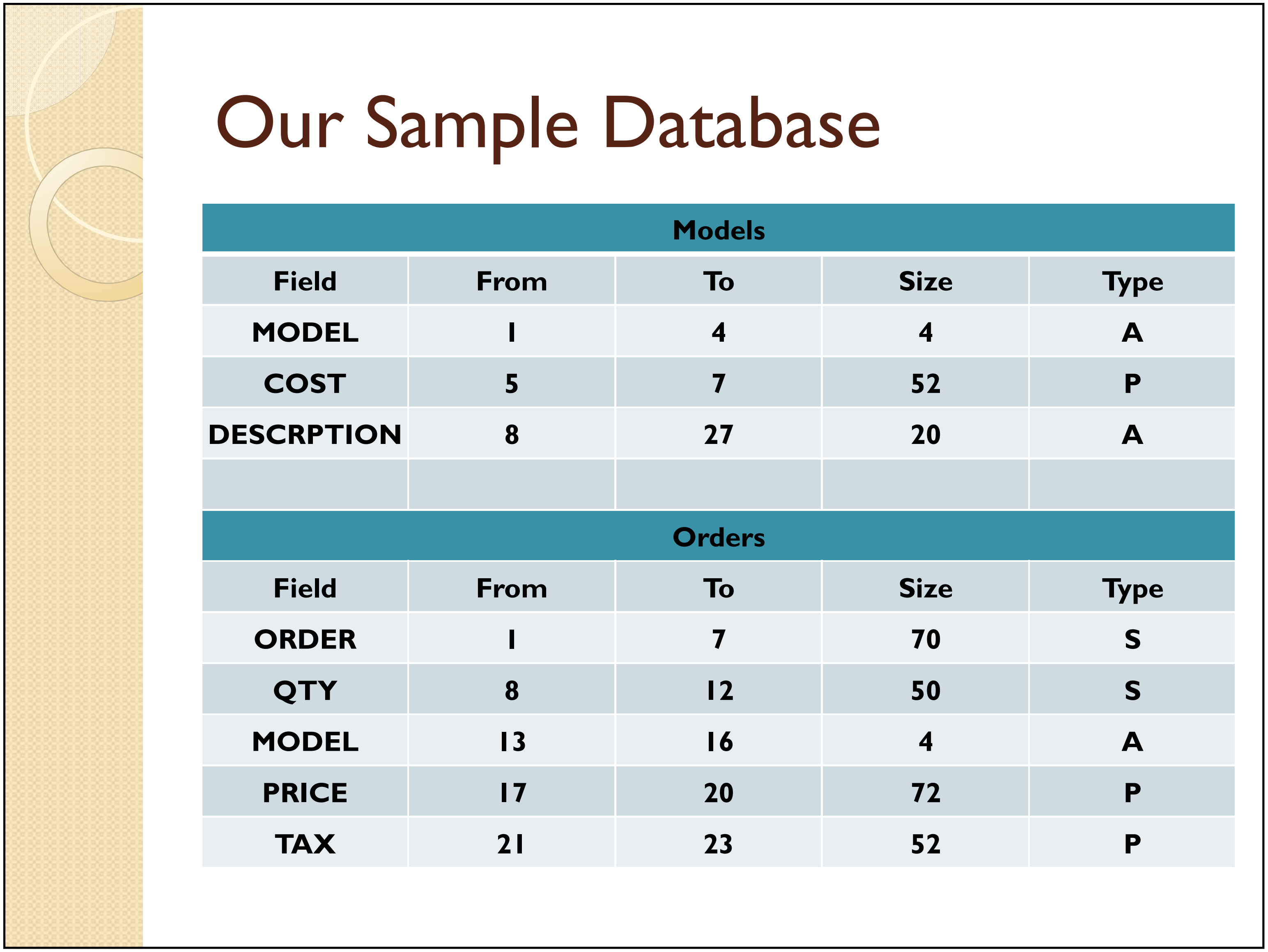
- Create Table Models (

```
  Model char(4),
  Cost dec(5,2),
  Description char(20))
```

Sample Database – File 2 of 2

- Orders File
  - Order #
  - Quantity
  - Model
  - Sales Price
  - Tax

-
  ```
  create table orders (
      Order num(7),
      Qty num(5),
      Model char(4),
      Price dec(7,2),
      Tax dec(5,2))
  ```



Our Sample Database

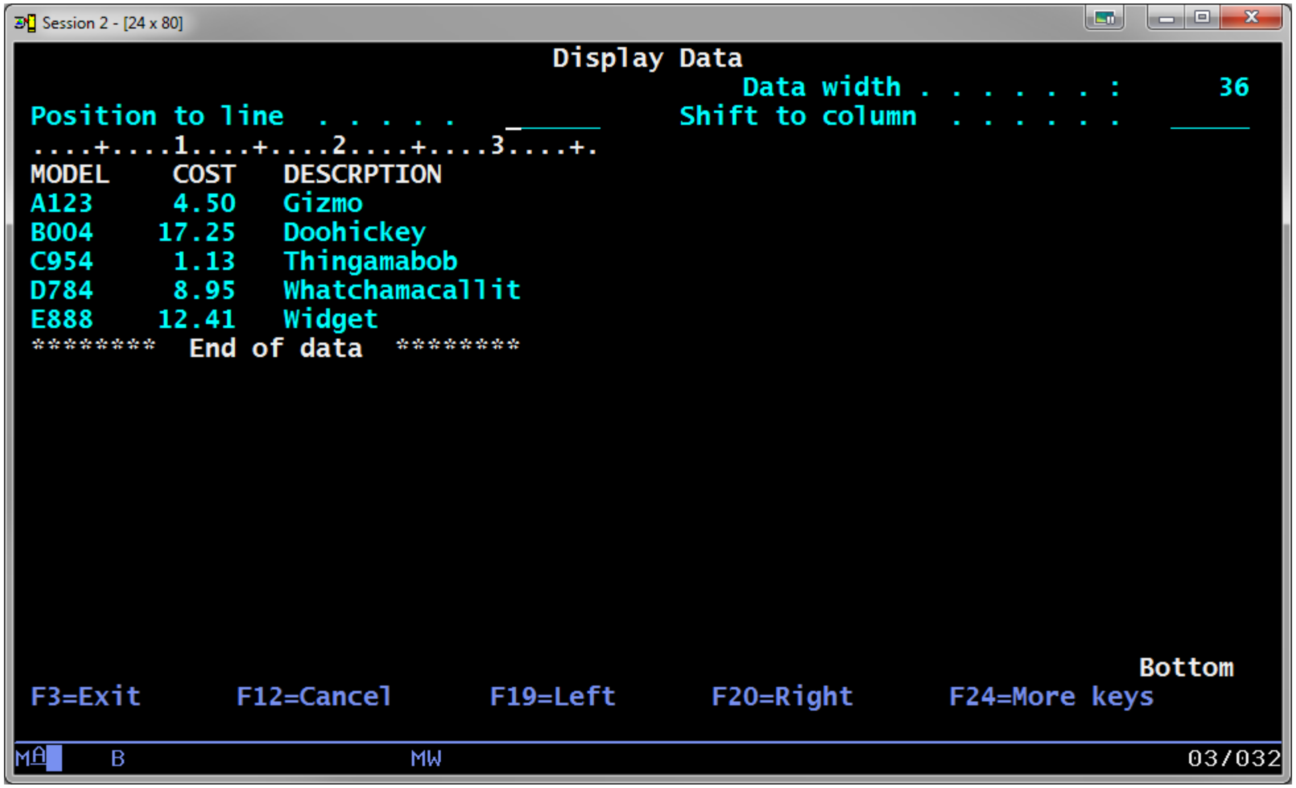
Models				
Field	**From**	**To**	**Size**	**Type**
MODEL	**1**	**4**	**4**	**A**
COST	**5**	**7**	**52**	**P**
DESCRPTION	**8**	**27**	**20**	**A**

Orders				
Field	**From**	**To**	**Size**	**Type**
ORDER	**1**	**7**	**70**	**S**
QTY	**8**	**12**	**50**	**S**
MODEL	**13**	**16**	**4**	**A**
PRICE	**17**	**20**	**72**	**P**
TAX	**21**	**23**	**52**	**P**

Simple Select Statement

Using STRSQL, let's look at our MODELS

select * from models



```
                             Display Data
                                               Data width . . . . . . :     36
Position to line  . . . . .  ______      Shift to column  . . . . . .   _____
....+....1....+....2....+....3....+.
MODEL     COST    DESCRPTION
A123      4.50    Gizmo
B004     17.25    Doohickey
C954      1.13    Thingamabob
D784      8.95    Whatchamacallit
E888     12.41    Widget
********  End of data  ********




                                                                      Bottom
F3=Exit      F12=Cancel      F19=Left      F20=Right      F24=More keys

MA     B                  MW                                          03/032
```

Simple Select Statement

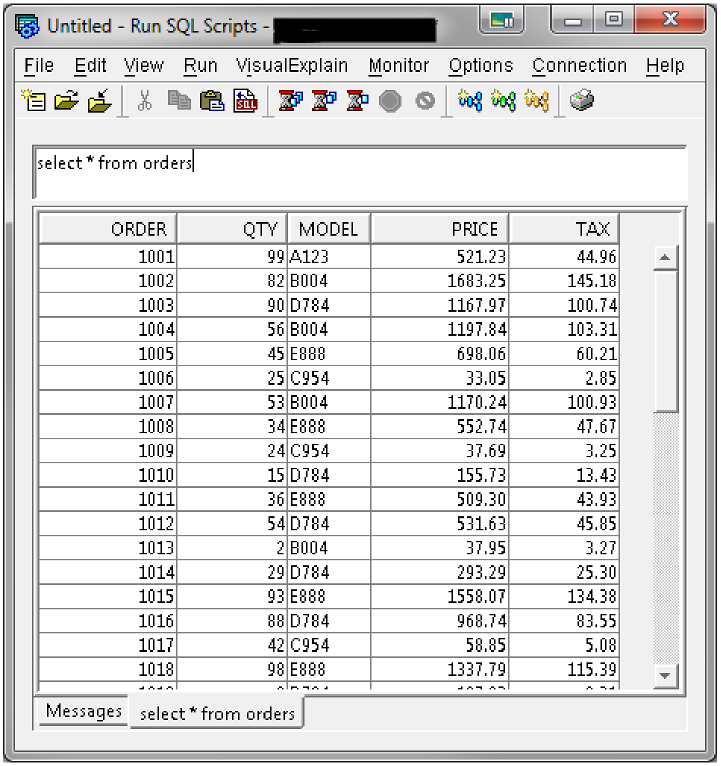
Using

System i Navigator,

let's look at our

ORDERS

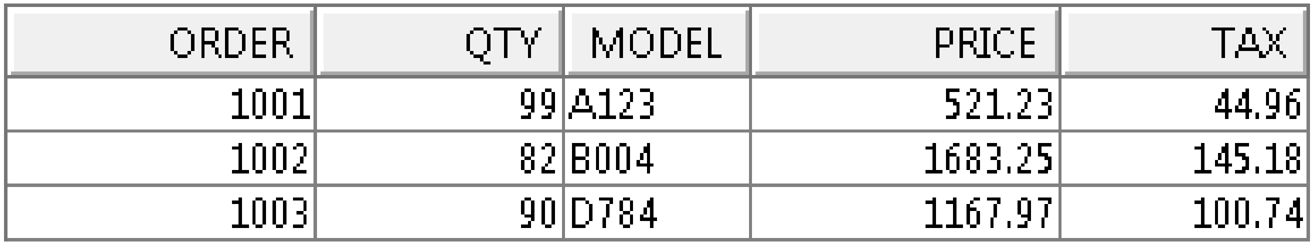
select * from orders



```
select * from orders
```

ORDER	QTY	MODEL	PRICE	TAX
1001	99	A123	521.23	44.96
1002	82	B004	1683.25	145.18
1003	90	D784	1167.97	100.74
1004	56	B004	1197.84	103.31
1005	45	E888	698.06	60.21
1006	25	C954	33.05	2.85
1007	53	B004	1170.24	100.93
1008	34	E888	552.74	47.67
1009	24	C954	37.69	3.25
1010	15	D784	155.73	13.43
1011	36	E888	509.30	43.93
1012	54	D784	531.63	45.85
1013	2	B004	37.95	3.27
1014	29	D784	293.29	25.30
1015	93	E888	1558.07	134.38
1016	88	D784	968.74	83.55
1017	42	C954	58.85	5.08
1018	98	E888	1337.79	115.39

select * from orders

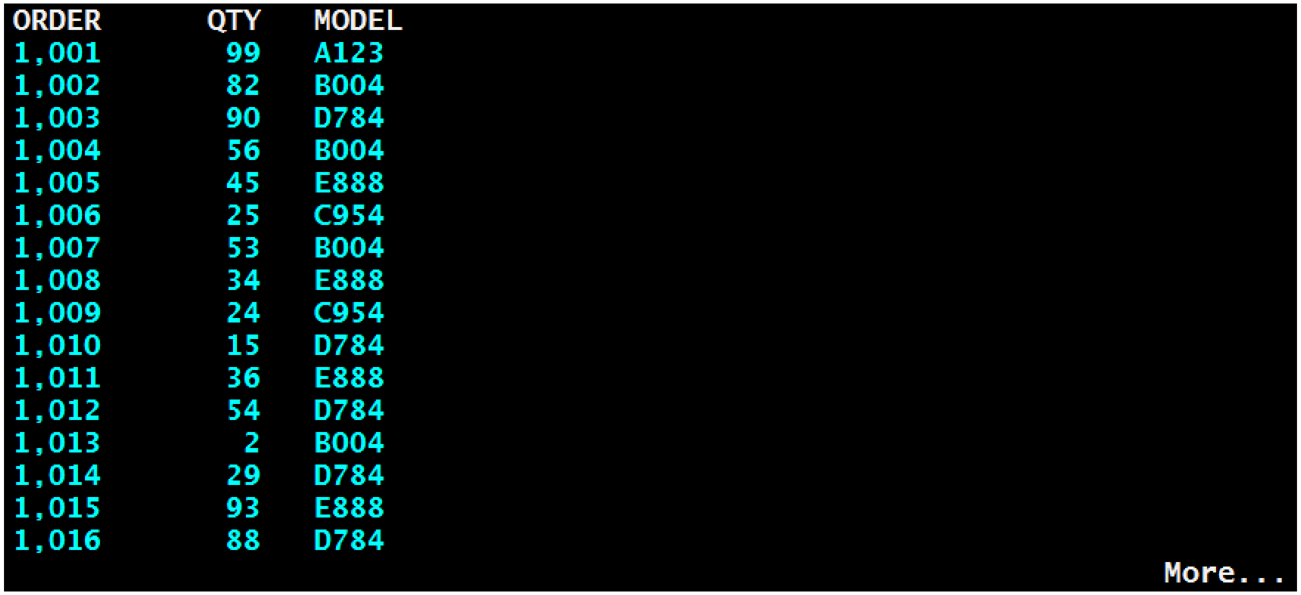


ORDER	QTY	MODEL	PRICE	TAX
1001	99	A123	521.23	44.96
1002	82	B004	1683.25	145.18
1003	90	D784	1167.97	100.74

Specifying Field Names

select order, qty, model from orders

select orders.order, orders.qty,
orders.model from orders

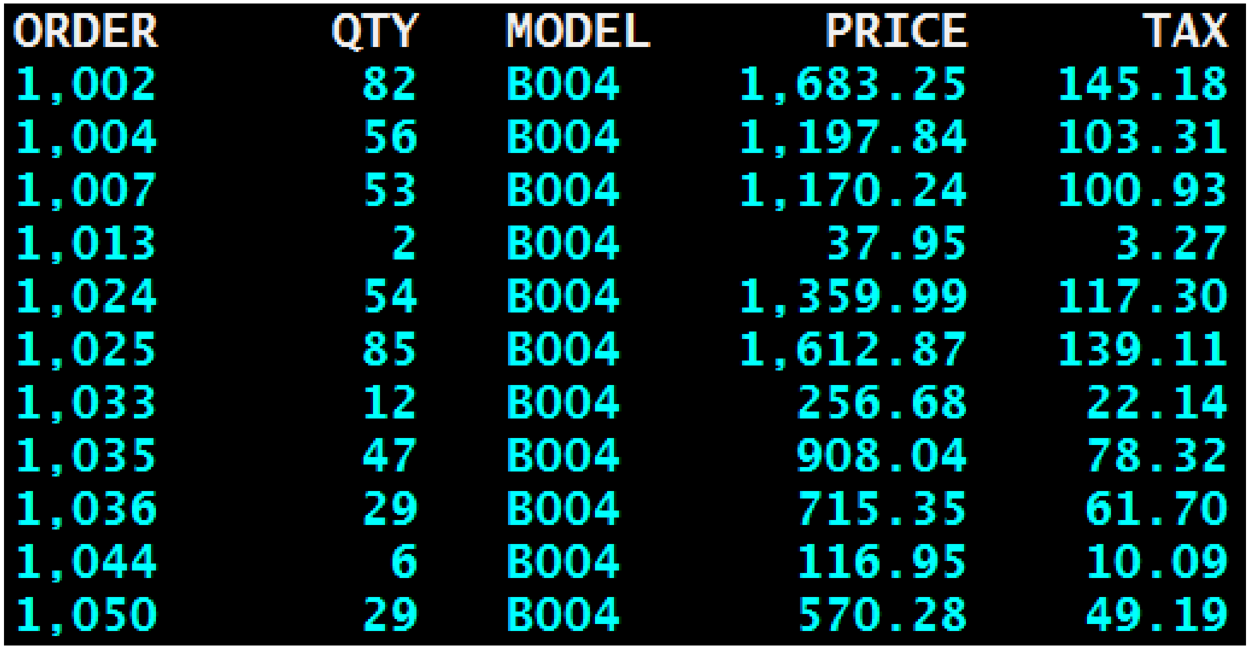


```
ORDER        QTY    MODEL
1,001         99    A123
1,002         82    B004
1,003         90    D784
1,004         56    B004
1,005         45    E888
1,006         25    C954
1,007         53    B004
1,008         34    E888
1,009         24    C954
1,010         15    D784
1,011         36    E888
1,012         54    D784
1,013          2    B004
1,014         29    D784
1,015         93    E888
1,016         88    D784
                                                                            More...
```

Where Clause

We want all orders for model B004

select * from orders where model='B004'

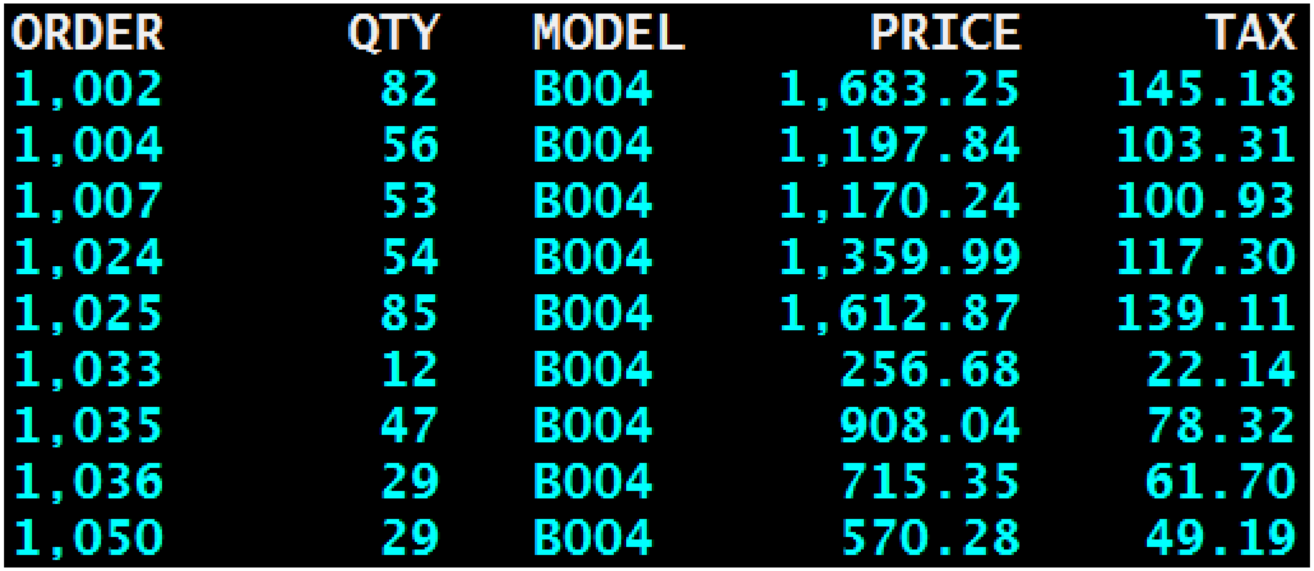


```
ORDER          QTY   MODEL       PRICE      TAX
1,002           82   B004     1,683.25   145.18
1,004           56   B004     1,197.84   103.31
1,007           53   B004     1,170.24   100.93
1,013            2   B004        37.95     3.27
1,024           54   B004     1,359.99   117.30
1,025           85   B004     1,612.87   139.11
1,033           12   B004       256.68    22.14
1,035           47   B004       908.04    78.32
1,036           29   B004       715.35    61.70
1,044            6   B004       116.95    10.09
1,050           29   B004       570.28    49.19
```

Where Clause – using AND

We want all orders for more than 10 B004's

select * from orders where model='B004' and qty>10

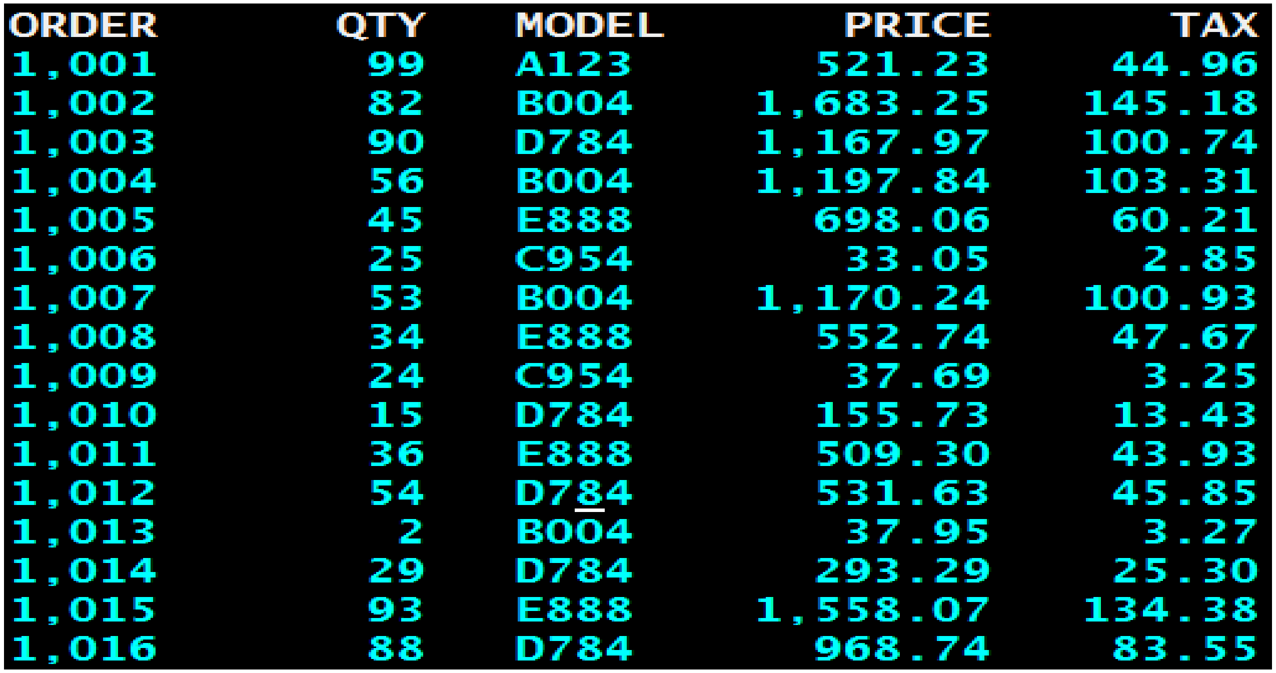


```
ORDER          QTY   MODEL       PRICE      TAX
1,002           82   B004     1,683.25   145.18
1,004           56   B004     1,197.84   103.31
1,007           53   B004     1,170.24   100.93
1,024           54   B004     1,359.99   117.30
1,025           85   B004     1,612.87   139.11
1,033           12   B004       256.68    22.14
1,035           47   B004       908.04    78.32
1,036           29   B004       715.35    61.70
1,050           29   B004       570.28    49.19
```

Where Clause – using OR

We want all orders for model B004 or where a quantity of greater than 10 was sold

select * from orders where model='B004' or qty>10

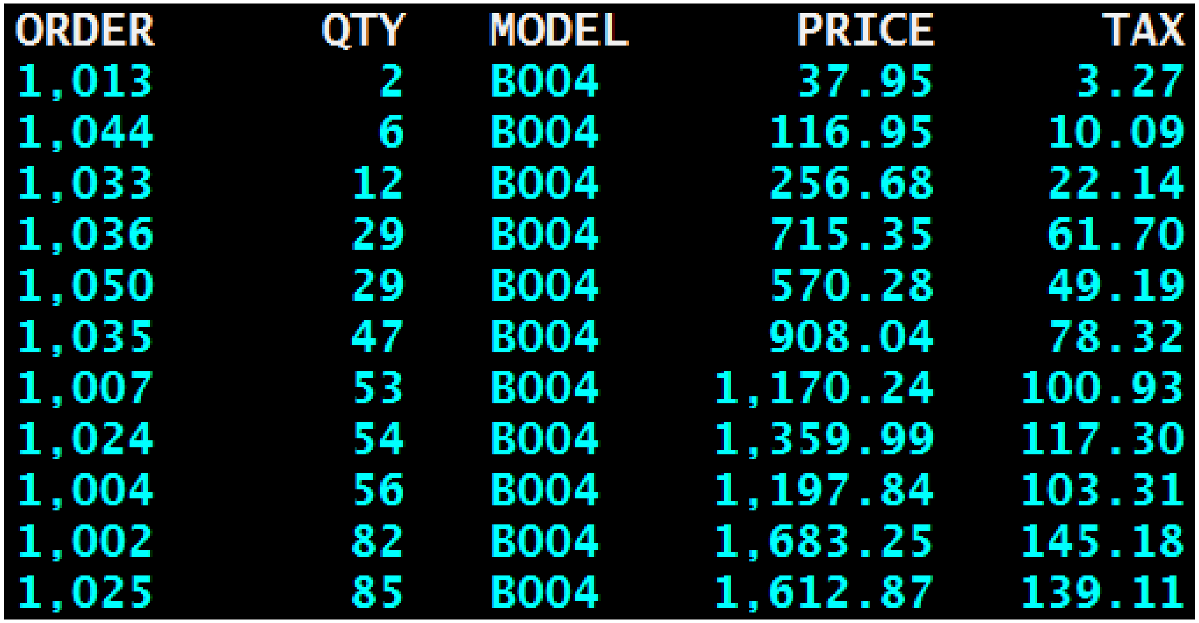


```
ORDER           QTY   MODEL          PRICE       TAX
1,001            99   A123          521.23     44.96
1,002            82   B004        1,683.25    145.18
1,003            90   D784        1,167.97    100.74
1,004            56   B004        1,197.84    103.31
1,005            45   E888          698.06     60.21
1,006            25   C954           33.05      2.85
1,007            53   B004        1,170.24    100.93
1,008            34   E888          552.74     47.67
1,009            24   C954           37.69      3.25
1,010            15   D784          155.73     13.43
1,011            36   E888          509.30     43.93
1,012            54   D784          531.63     45.85
1,013             2   B004           37.95      3.27
1,014            29   D784          293.29     25.30
1,015            93   E888        1,558.07    134.38
1,016            88   D784          968.74     83.55
```

Sorting your results

We want to sort our results by quantity

select * from orders where
model='B004' order by qty

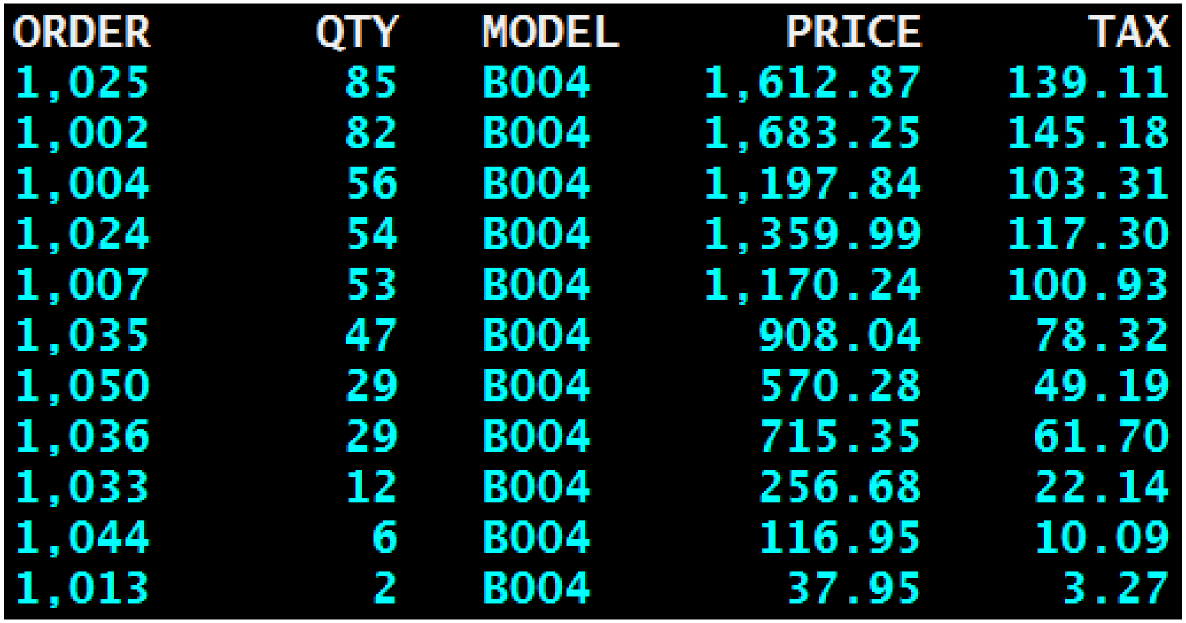


```
ORDER          QTY   MODEL          PRICE        TAX
1,013            2   B004           37.95       3.27
1,044            6   B004          116.95      10.09
1,033           12   B004          256.68      22.14
1,036           29   B004          715.35      61.70
1,050           29   B004          570.28      49.19
1,035           47   B004          908.04      78.32
1,007           53   B004        1,170.24     100.93
1,024           54   B004        1,359.99     117.30
1,004           56   B004        1,197.84     103.31
1,002           82   B004        1,683.25     145.18
1,025           85   B004        1,612.87     139.11
```

Sorting your results - Descending

We want to sort by descending quantity

select * from orders where
model='B004' order by qty desc



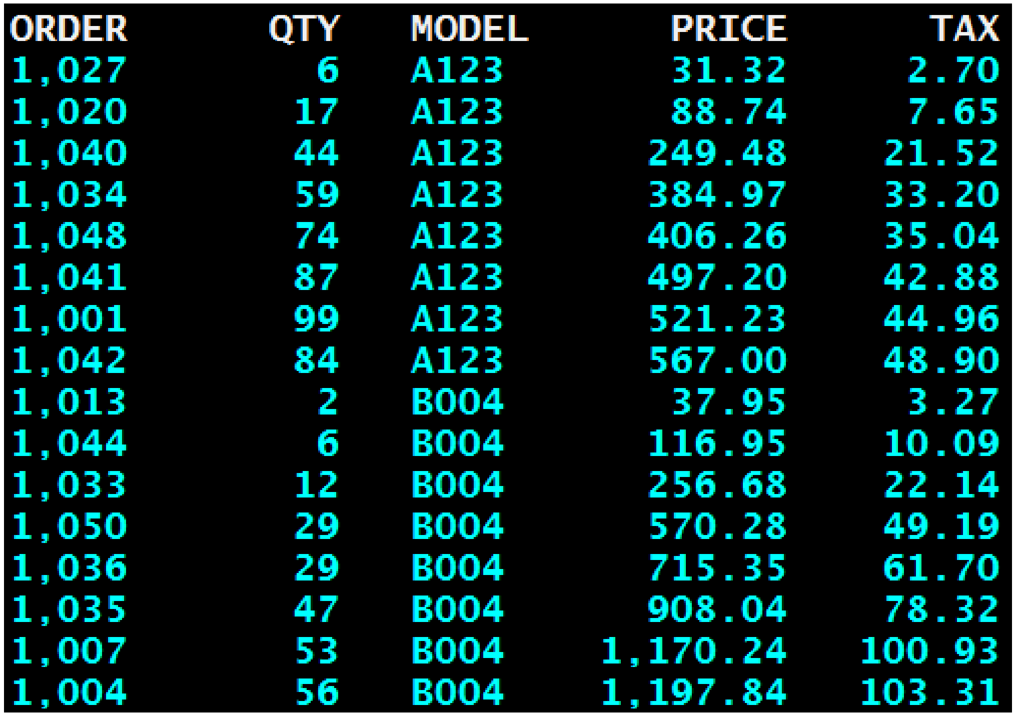
```
ORDER        QTY   MODEL       PRICE      TAX
1,025         85   B004     1,612.87   139.11
1,002         82   B004     1,683.25   145.18
1,004         56   B004     1,197.84   103.31
1,024         54   B004     1,359.99   117.30
1,007         53   B004     1,170.24   100.93
1,035         47   B004       908.04    78.32
1,050         29   B004       570.28    49.19
1,036         29   B004       715.35    61.70
1,033         12   B004       256.68    22.14
1,044          6   B004       116.95    10.09
1,013          2   B004        37.95     3.27
```

Sorting your results by multiple fields

We want to sort by model and by price

select * from orders

order by model, price



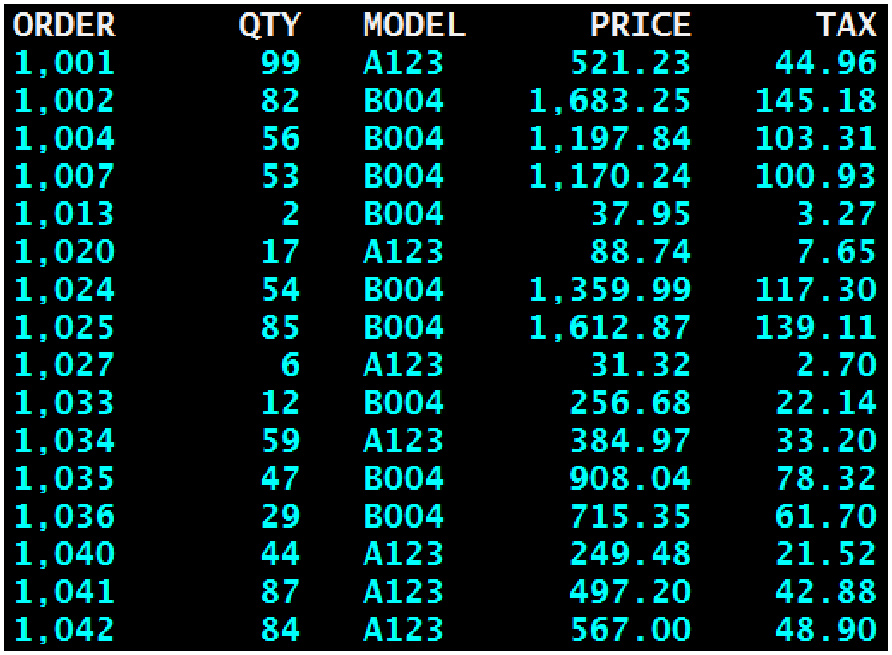
```
ORDER      QTY   MODEL      PRICE      TAX
1,027        6   A123       31.32     2.70
1,020       17   A123       88.74     7.65
1,040       44   A123      249.48    21.52
1,034       59   A123      384.97    33.20
1,048       74   A123      406.26    35.04
1,041       87   A123      497.20    42.88
1,001       99   A123      521.23    44.96
1,042       84   A123      567.00    48.90
1,013        2   B004       37.95     3.27
1,044        6   B004      116.95    10.09
1,033       12   B004      256.68    22.14
1,050       29   B004      570.28    49.19
1,036       29   B004      715.35    61.70
1,035       47   B004      908.04    78.32
1,007       53   B004    1,170.24   100.93
1,004       56   B004    1,197.84   103.31
```

Selecting Based on a List

We want all orders for A123 and B004

select * from orders where

model **in** **('A123','B004')**

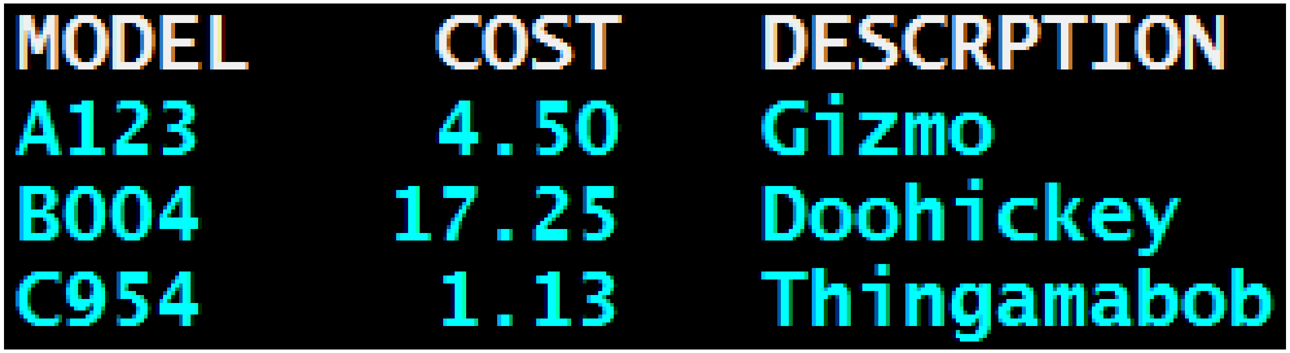


```
ORDER        QTY   MODEL        PRICE       TAX
1,001         99   A123        521.23     44.96
1,002         82   B004      1,683.25    145.18
1,004         56   B004      1,197.84    103.31
1,007         53   B004      1,170.24    100.93
1,013          2   B004         37.95      3.27
1,020         17   A123         88.74      7.65
1,024         54   B004      1,359.99    117.30
1,025         85   B004      1,612.87    139.11
1,027          6   A123         31.32      2.70
1,033         12   B004        256.68     22.14
1,034         59   A123        384.97     33.20
1,035         47   B004        908.04     78.32
1,036         29   B004        715.35     61.70
1,040         44   A123        249.48     21.52
1,041         87   A123        497.20     42.88
1,042         84   A123        567.00     48.90
```

Selecting Based on Text Patterns

We want all models with the letter "o" in the description

select * from models where descrption like '%o%'



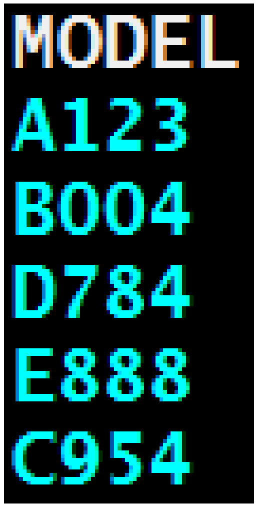
```
MODEL       COST    DESCRPTION
A123        4.50    Gizmo
B004       17.25    Doohickey
C954        1.13    Thingamabob
```

Note: The % sign acts as a wildcard, and matches anything.

Selecting Distinct Values

We want a list of all the unique models in our orders file

select distinct model from orders

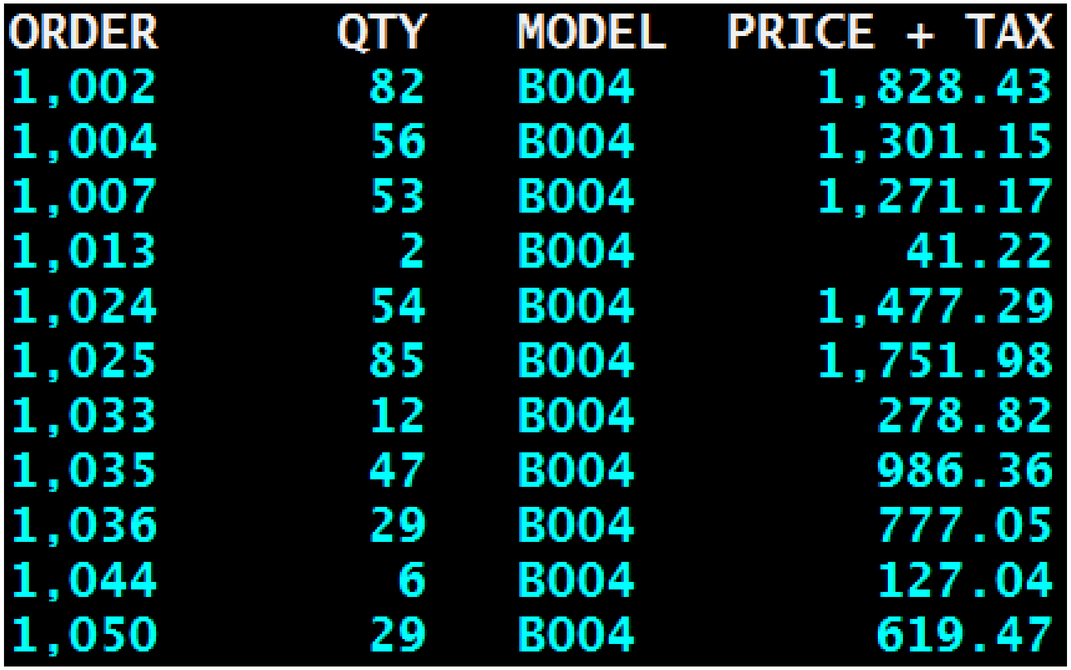


```
MODEL
A123
B004
D784
E888
C954
```

Simple Math

We want to add Price + Tax

select order, qty, model, price+tax
from orders where model='B004'

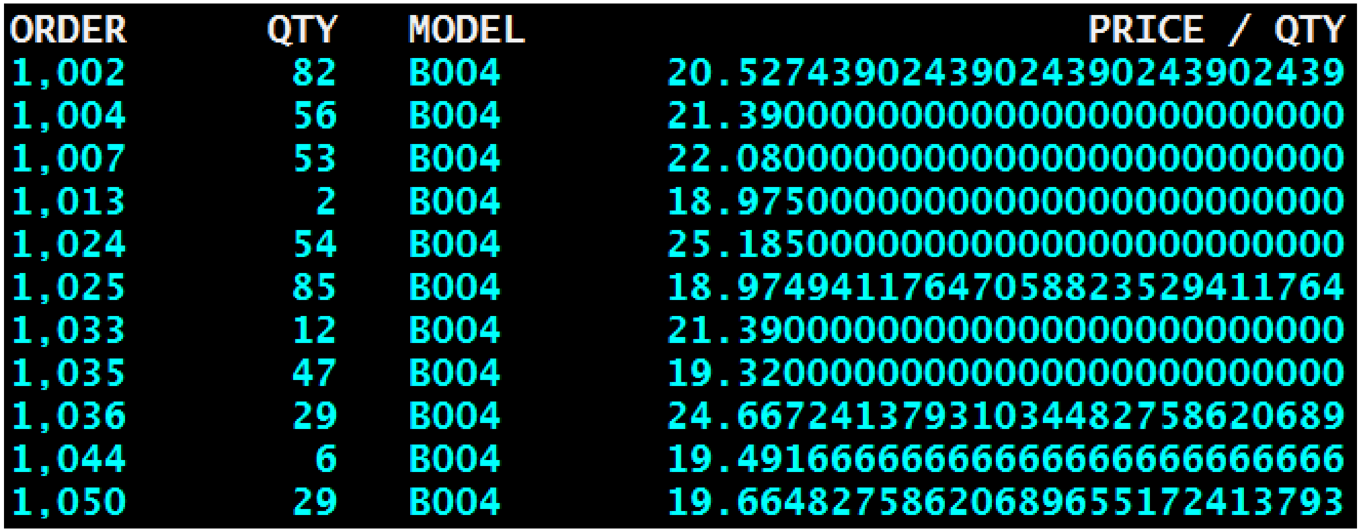


```
ORDER          QTY     MODEL    PRICE + TAX
1,002           82     B004        1,828.43
1,004           56     B004        1,301.15
1,007           53     B004        1,271.17
1,013            2     B004           41.22
1,024           54     B004        1,477.29
1,025           85     B004        1,751.98
1,033           12     B004          278.82
1,035           47     B004          986.36
1,036           29     B004          777.05
1,044            6     B004          127.04
1,050           29     B004          619.47
```

More Simple Math

We want to find unit sales price

**select order, qty, model, price/qty
from orders where model='B004'**

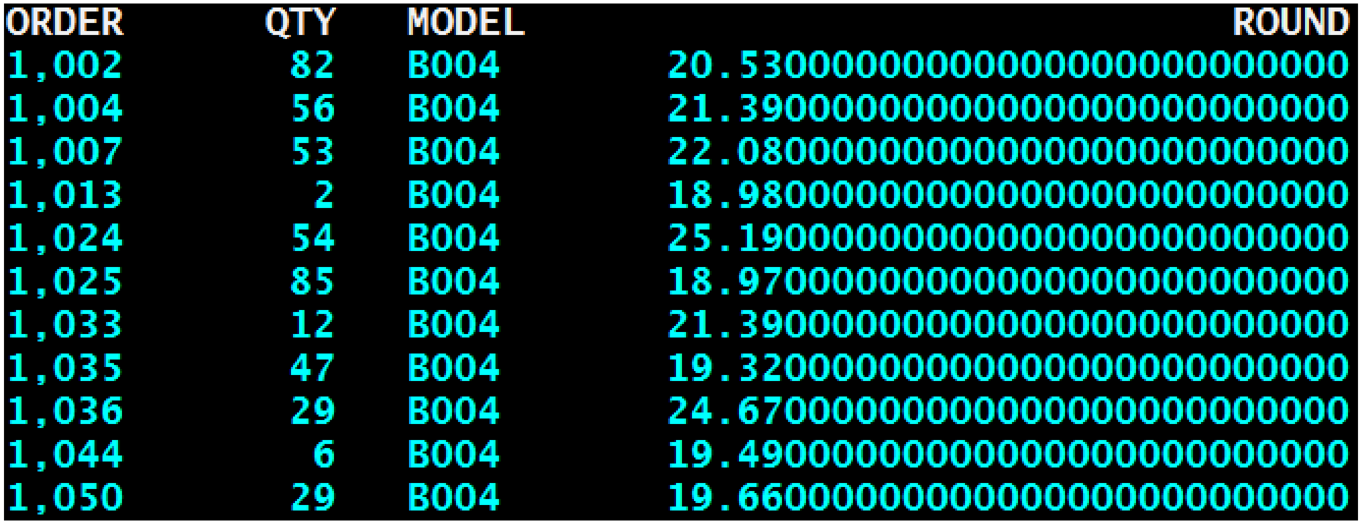


```
ORDER      QTY    MODEL                        PRICE / QTY
1,002       82    B004       20.52743902439024390243902439
1,004       56    B004       21.39000000000000000000000000
1,007       53    B004       22.08000000000000000000000000
1,013        2    B004       18.97500000000000000000000000
1,024       54    B004       25.18500000000000000000000000
1,025       85    B004       18.97494117647058823529411764
1,033       12    B004       21.39000000000000000000000000
1,035       47    B004       19.32000000000000000000000000
1,036       29    B004       24.66724137931034482758620689
1,044        6    B004       19.49166666666666666666666666
1,050       29    B004       19.66482758620689655172413793
```

More Simple Math – with Rounding

We want to find rounded unit sales price

select order, qty, model, round(price/qty,2)
from orders where model='B004'

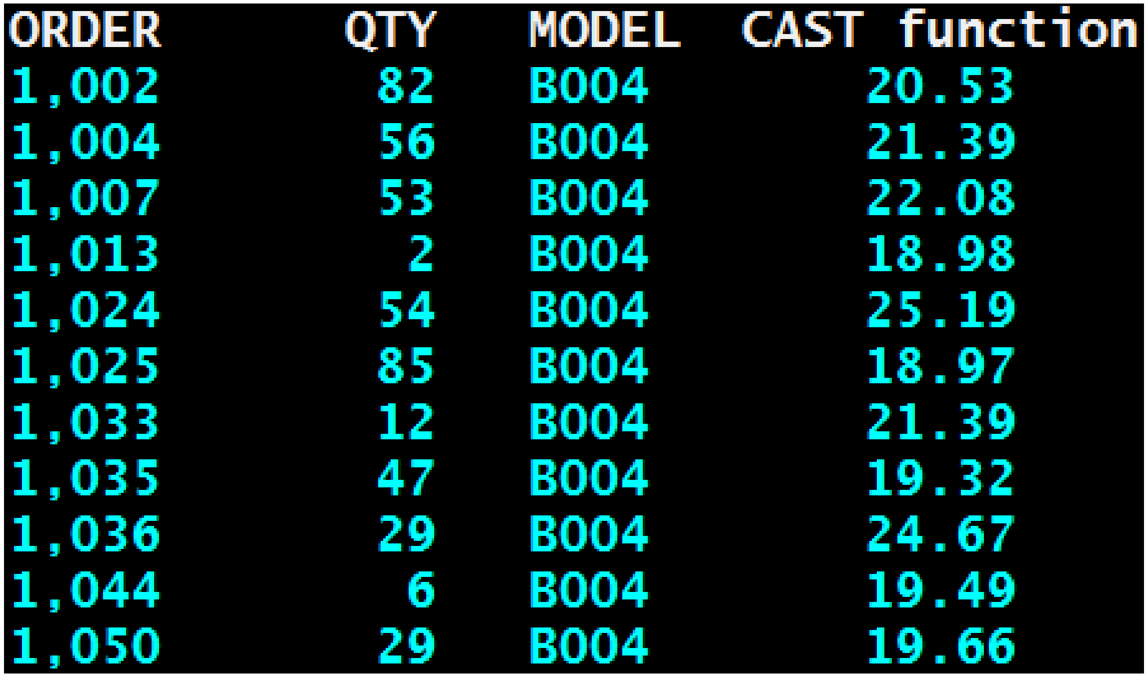


```
ORDER        QTY    MODEL                               ROUND
1,002         82    B004         20.53000000000000000000000000
1,004         56    B004         21.39000000000000000000000000
1,007         53    B004         22.08000000000000000000000000
1,013          2    B004         18.98000000000000000000000000
1,024         54    B004         25.19000000000000000000000000
1,025         85    B004         18.97000000000000000000000000
1,033         12    B004         21.39000000000000000000000000
1,035         47    B004         19.32000000000000000000000000
1,036         29    B004         24.67000000000000000000000000
1,044          6    B004         19.49000000000000000000000000
1,050         29    B004         19.66000000000000000000000000
```

Casting Data Types

We want to find unit sales price in money format

select order, qty, model,
cast(round(price/qty,2) as dec(4,2))
from orders where model='B004'

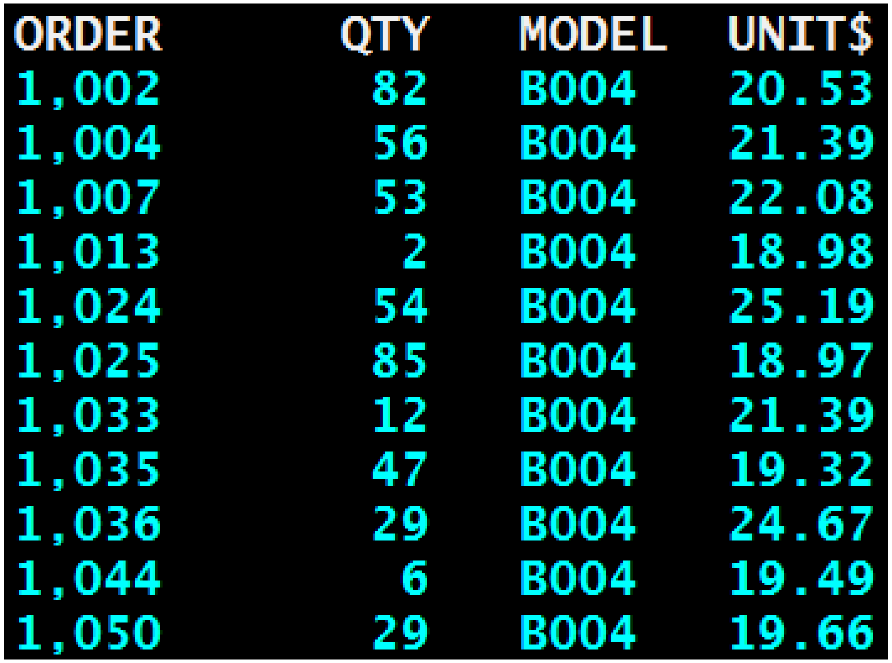


```
ORDER           QTY     MODEL   CAST  function
1,002            82     B004         20.53
1,004            56     B004         21.39
1,007            53     B004         22.08
1,013             2     B004         18.98
1,024            54     B004         25.19
1,025            85     B004         18.97
1,033            12     B004         21.39
1,035            47     B004         19.32
1,036            29     B004         24.67
1,044             6     B004         19.49
1,050            29     B004         19.66
```

Renaming Expressions

We want to find unit sales price in money format

select order, qty, model,
  cast(round(price/qty,2) as dec(4,2)) Unit$
  from orders where model='B004'

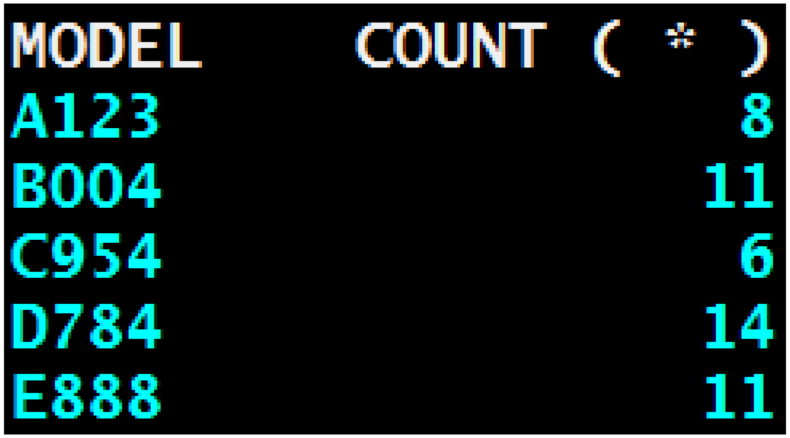


```
ORDER         QTY    MODEL   UNIT$
1,002          82    B004    20.53
1,004          56    B004    21.39
1,007          53    B004    22.08
1,013           2    B004    18.98
1,024          54    B004    25.19
1,025          85    B004    18.97
1,033          12    B004    21.39
1,035          47    B004    19.32
1,036          29    B004    24.67
1,044           6    B004    19.49
1,050          29    B004    19.66
```

Simple Data Analysis - Grouping

We want to find the number of orders written for each model

select model, count(*) from orders group by model order by model

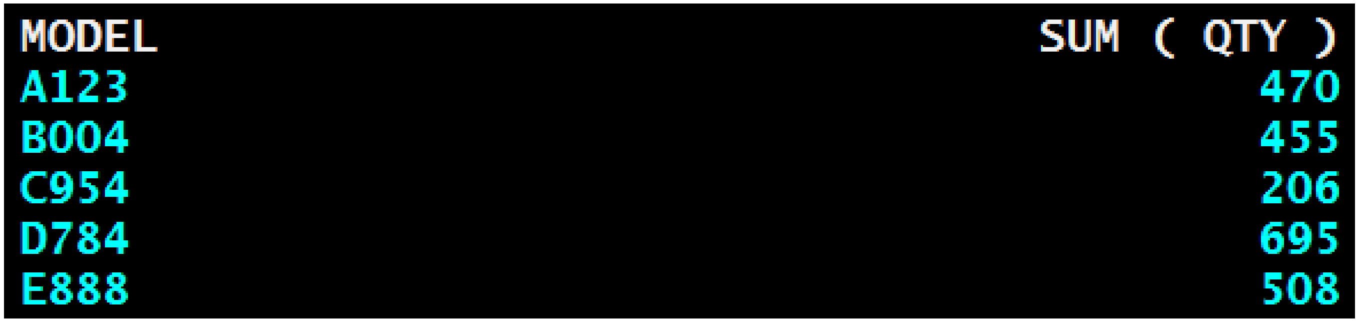


```
MODEL      COUNT(*)
A123              8
B004             11
C954              6
D784             14
E888             11
```

Simple Data Analysis - Grouping

We want to find total quantity sold for each model

select model, sum(qty) from orders group by model order by model

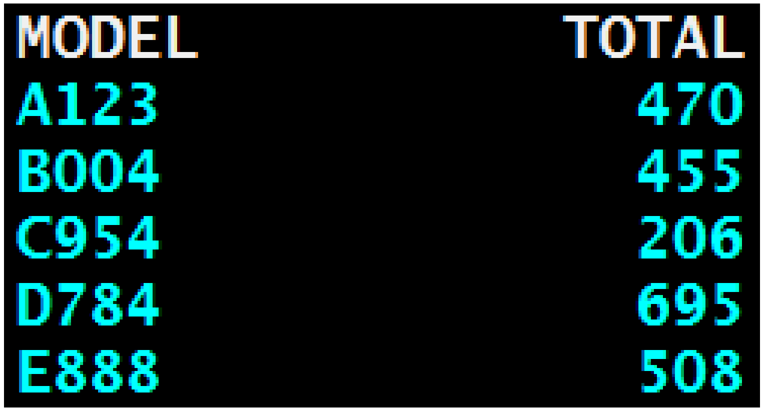


```
MODEL                                    SUM ( QTY )
A123                                             470
B004                                             455
C954                                             206
D784                                             695
E888                                             508
```

Simple Data Analysis - Grouping

We want to find total quantity sold for each model, formatted and renamed for reporting.

select model, int(sum(qty)) Total from orders group by model order by model

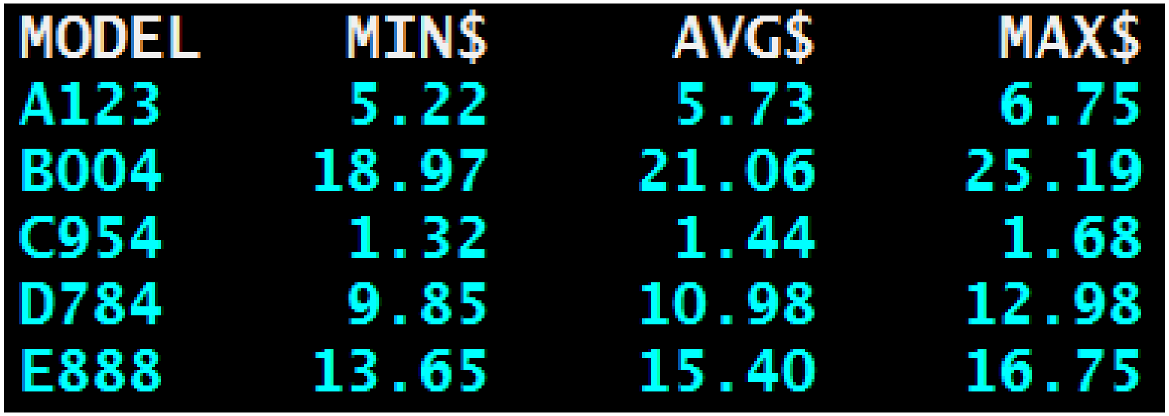


```
MODEL                TOTAL
A123                   470
B004                   455
C954                   206
D784                   695
E888                   508
```

Simple Data Analysis - Grouping

We want to find minimum, maximum, and average unit sales $ for each model

```
select model,
Cast( round(min(price/qty),2) as dec(5,2)) Min$,
Cast( round(avg(price/qty),2) as dec(5,2)) Avg$,
Cast( round(max(price/qty),2) as dec(5,2)) Max$
from orders group by model order by model
```



```
MODEL       MIN$       AVG$       MAX$
A123        5.22       5.73       6.75
B004       18.97      21.06      25.19
C954        1.32       1.44       1.68
D784        9.85      10.98      12.98
E888       13.65      15.40      16.75
```

Simple Data Analysis - Grouping

We want to find minimum, maximum, and average unit sales $ for each model

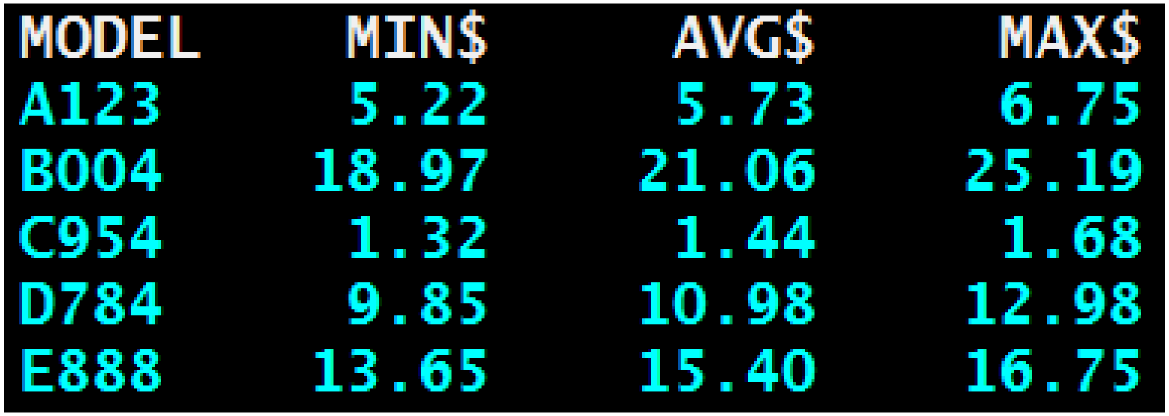
select model,

Cast(round(min(price/qty),2) as dec(5,2)) Min$,

Cast(round(avg(price/qty),2) as dec(5,2)) Avg$,

Cast(round(max(price/qty),2) as dec(5,2)) Max$

from orders group by model order by model

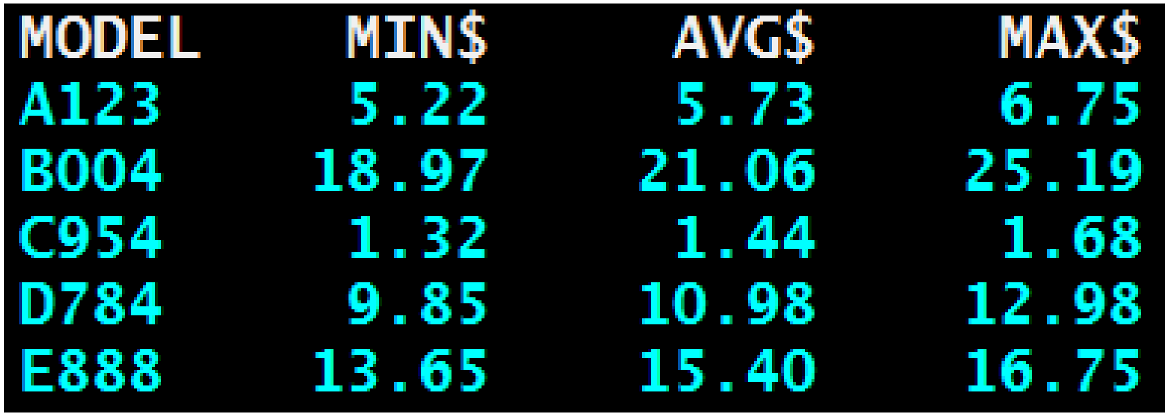


```
MODEL        MIN$       AVG$       MAX$
A123         5.22       5.73       6.75
B004        18.97      21.06      25.19
C954         1.32       1.44       1.68
D784         9.85      10.98      12.98
E888        13.65      15.40      16.75
```

Simple Data Analysis - Grouping

We want to find minimum, maximum, and average unit sales $ for each model

```
select model,
Cast( round(min(price/qty),2) as dec(5,2)) Min$,
Cast( round(avg(price/qty),2) as dec(5,2)) Avg$,
Cast( round(max(price/qty),2) as dec(5,2)) Max$
from orders group by model order by model
```

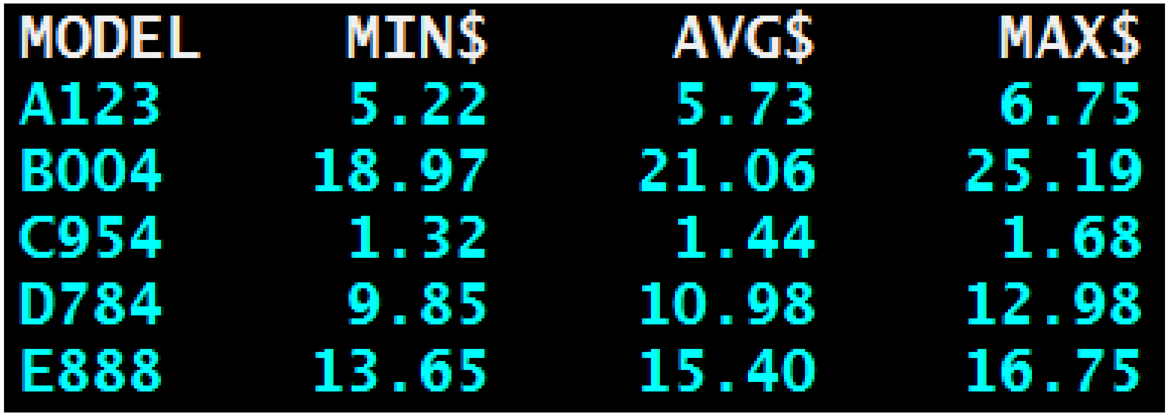


```
MODEL        MIN$      AVG$      MAX$
A123         5.22      5.73      6.75
B004        18.97     21.06     25.19
C954         1.32      1.44      1.68
D784         9.85     10.98     12.98
E888        13.65     15.40     16.75
```

Simple Data Analysis - Grouping

We want to find minimum, maximum, and average unit sales $ for each model

```
select model,
Cast( round(min(price/qty),2) as dec(5,2)) Min$,
Cast( round(avg(price/qty),2) as dec(5,2)) Avg$,
Cast( round(max(price/qty),2) as dec(5,2)) Max$
from orders group by model order by model
```

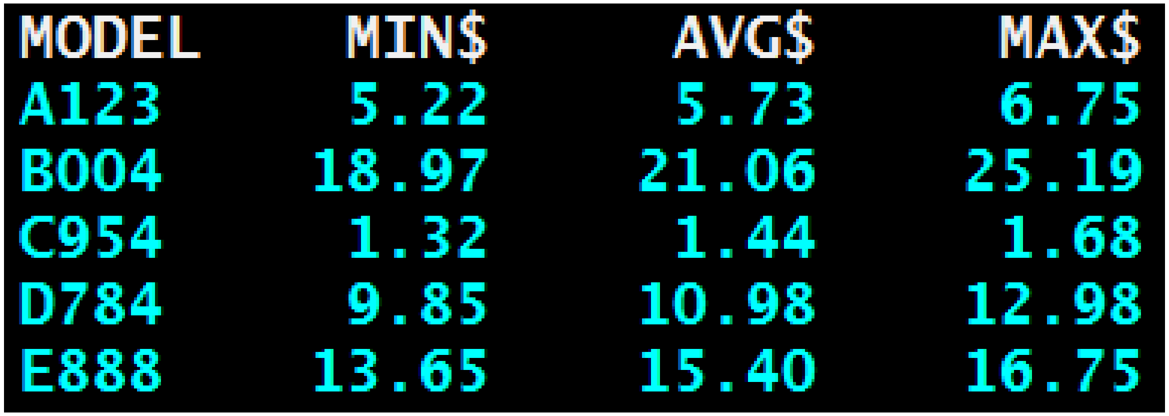


```
MODEL       MIN$      AVG$      MAX$
A123        5.22      5.73      6.75
B004       18.97     21.06     25.19
C954        1.32      1.44      1.68
D784        9.85     10.98     12.98
E888       13.65     15.40     16.75
```

Simple Data Analysis - Grouping

We want to find minimum, maximum, and average unit sales $ for each model

```
select model,
Cast( round(min(price/qty),2) as dec(5,2)) Min$,
Cast( round(avg(price/qty),2) as dec(5,2)) Avg$,
Cast( round(max(price/qty),2) as dec(5,2)) Max$
from orders group by model order by model
```

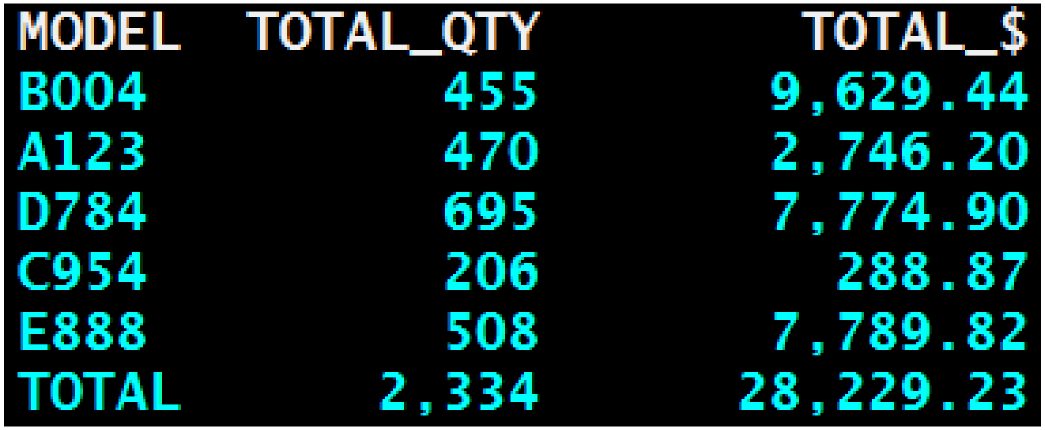


```
MODEL        MIN$       AVG$       MAX$
A123         5.22       5.73       6.75
B004        18.97      21.06      25.19
C954         1.32       1.44       1.68
D784         9.85      10.98      12.98
E888        13.65      15.40      16.75
```

Grouping with Totals - Union

We want to summarize quantity & sales $ by model, including grand totals

```
select model,
        cast(sum(qty) as dec(7,0)) Total_Qty,
        cast(sum(price) as dec(9,2)) Total_$
        from orders group by model
union
select 'TOTAL',
        cast(sum(qty) as dec(7,0)),
        cast(sum(price) as dec(9,2))
        from orders
```



```
MODEL   TOTAL_QTY        TOTAL_$
B004          455       9,629.44
A123          470       2,746.20
D784          695       7,774.90
C954          206         288.87
E888          508       7,789.82
TOTAL       2,334      28,229.23
```

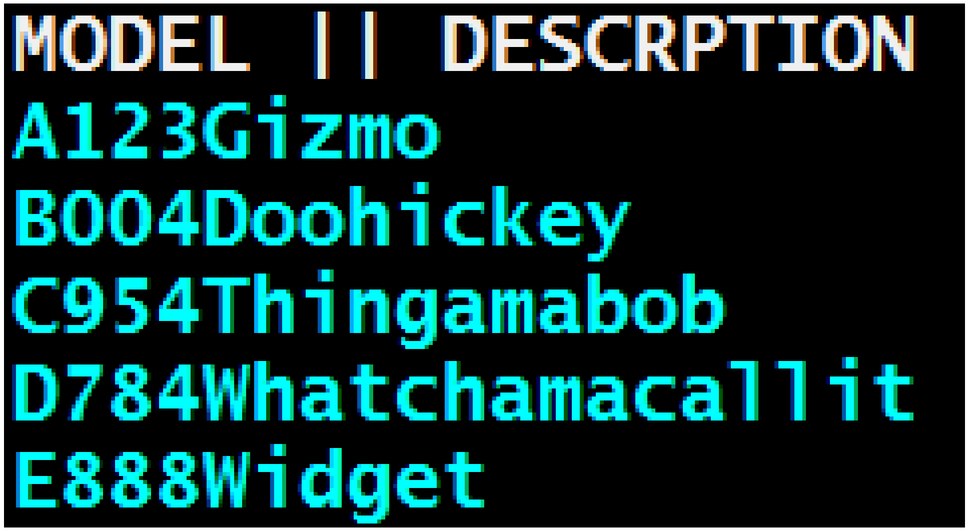
String Functions - Concatenation

Concatenate Model & Description Together

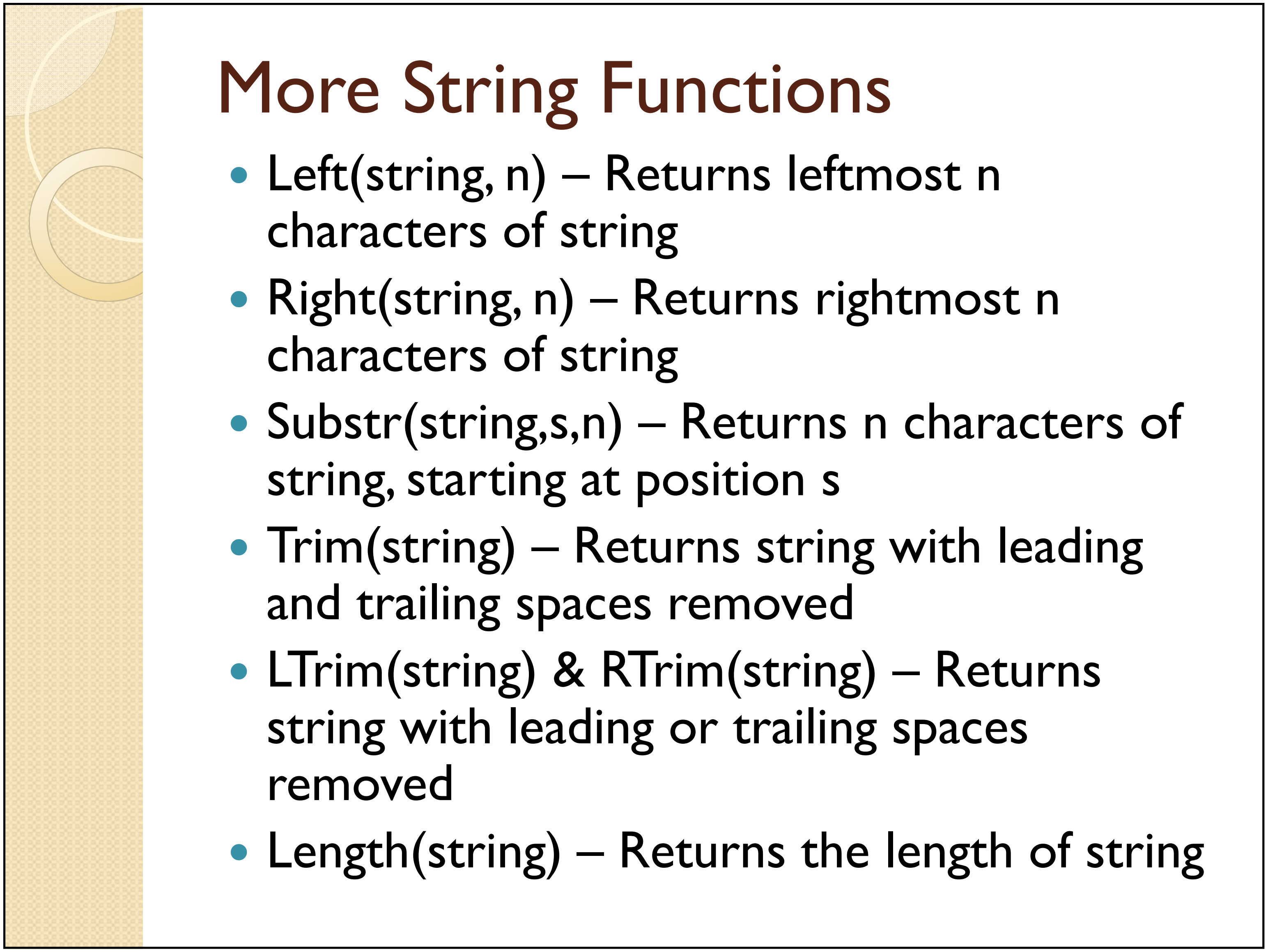
select model || descrption from models

or

select concat(model,descrption) from models



```
MODEL || DESCRPTION
A123Gizmo
B004Doohickey
C954Thingamabob
D784Whatchamacallit
E888Widget
```

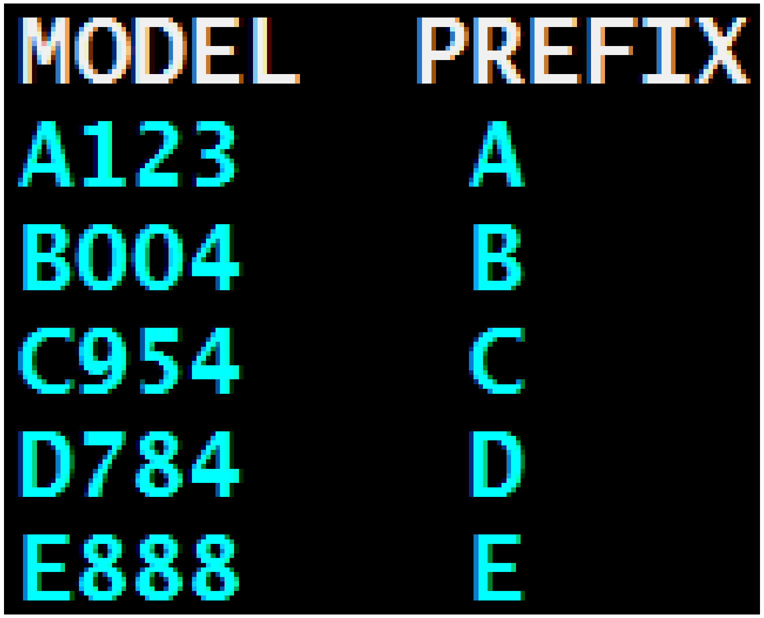


More String Functions

- Left(string, n) – Returns leftmost n characters of string
- Right(string, n) – Returns rightmost n characters of string
- Substr(string,s,n) – Returns n characters of string, starting at position s
- Trim(string) – Returns string with leading and trailing spaces removed
- LTrim(string) & RTrim(string) – Returns string with leading or trailing spaces removed
- Length(string) – Returns the length of string

String Function Examples - LEFT

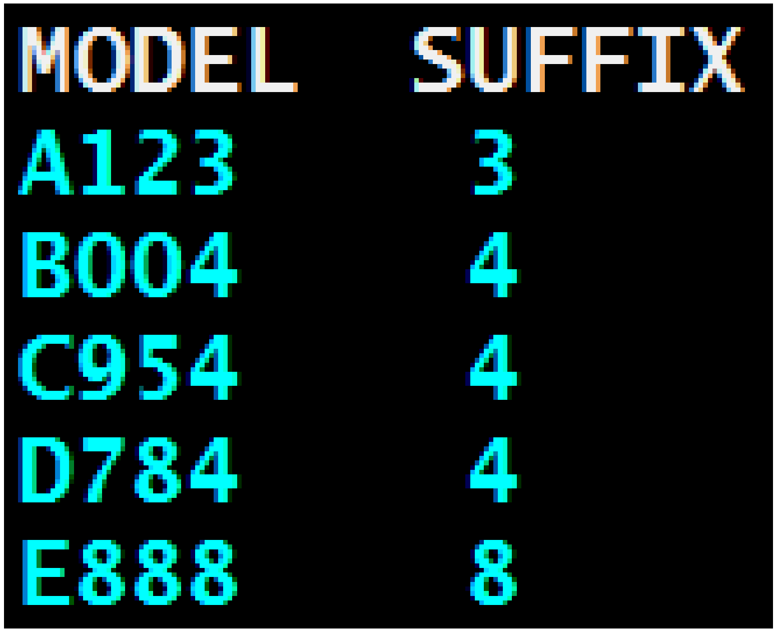
select model, left(model,1) prefix from models



```
MODEL   PREFIX
A123      A
B004      B
C954      C
D784      D
E888      E
```

String Function Examples - RIGHT

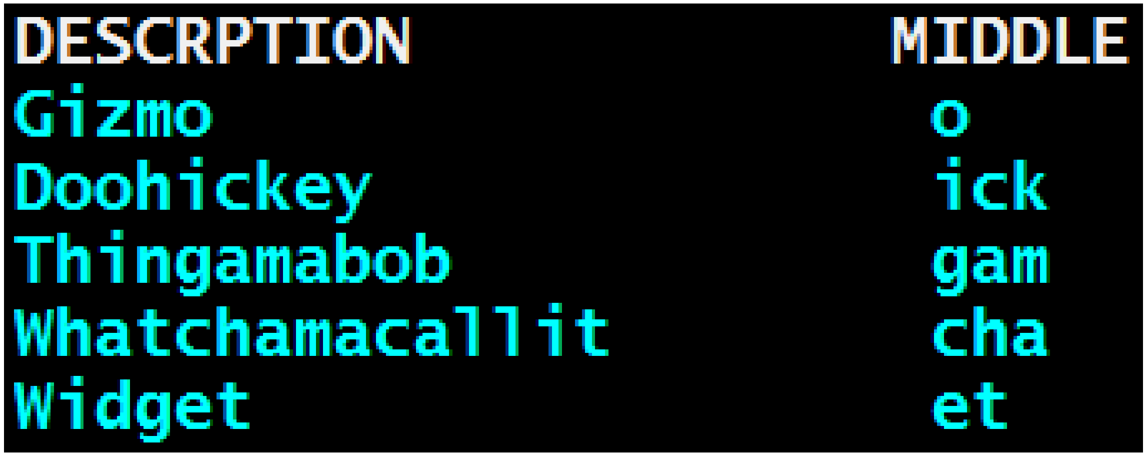
select model, right(model,1) suffix from models



```
MODEL   SUFFIX
A123      3
B004      4
C954      4
D784      4
E888      8
```

String Function Examples - SUBSTR

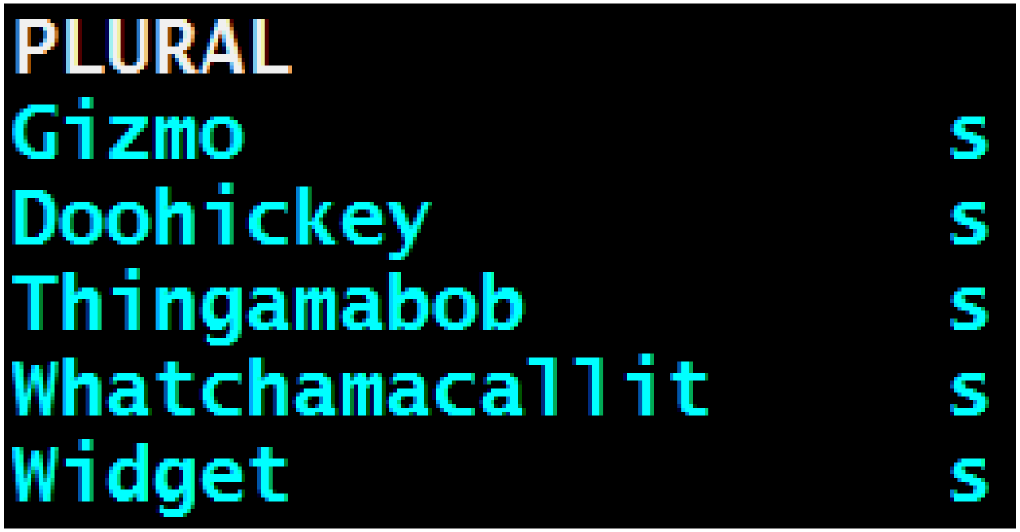
select descrption, substr(descrption,5,3) middle from models



```
DESCRPTION              MIDDLE
Gizmo                    o
Doohickey                ick
Thingamabob              gam
Whatchamacallit          cha
Widget                   et
```

String Function Examples - TRIM

select descrption || 's' plural from models

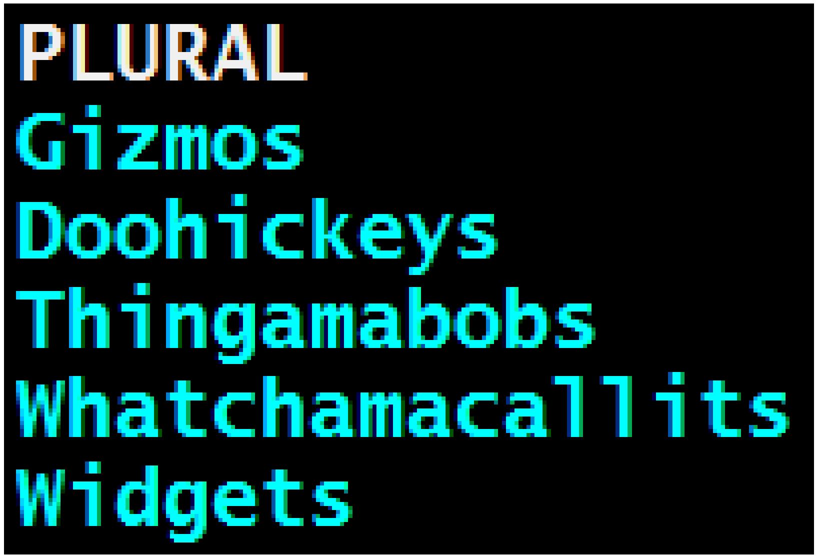


```
PLURAL
Gizmo               s
Doohickey           s
Thingamabob         s
Whatchamacallit     s
Widget              s
```

How do we remove the extra spaces? With TRIM

String Function Examples - TRIM

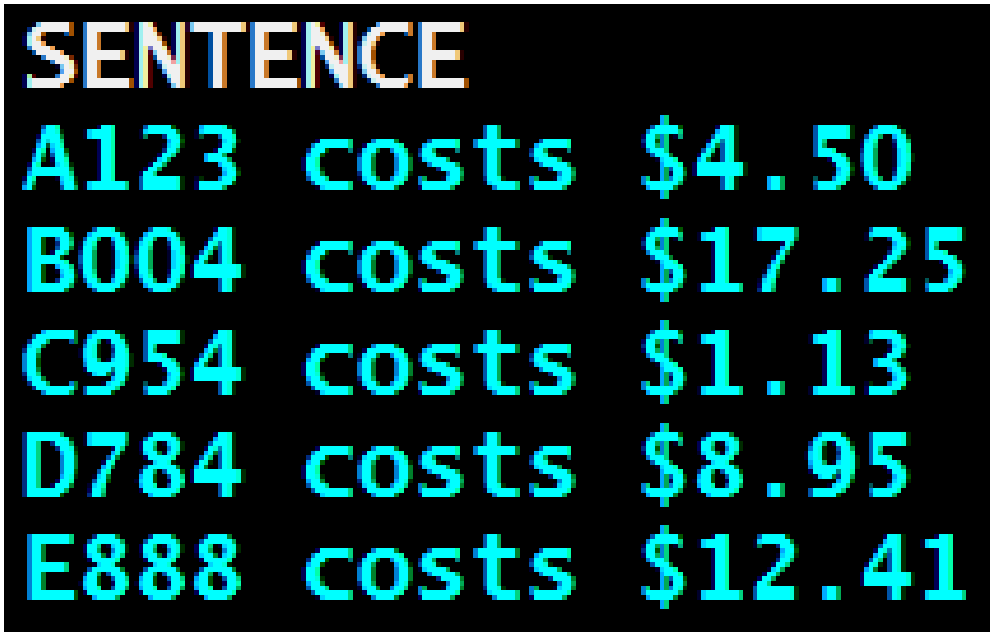
select trim(descrption) || 's' plural from models



```
PLURAL
Gizmos
Doohickeys
Thingamabobs
Whatchamacallits
Widgets
```

Converting Numbers to Characters

select trim(model) || ' costs $' || char(cost) sentence
from models



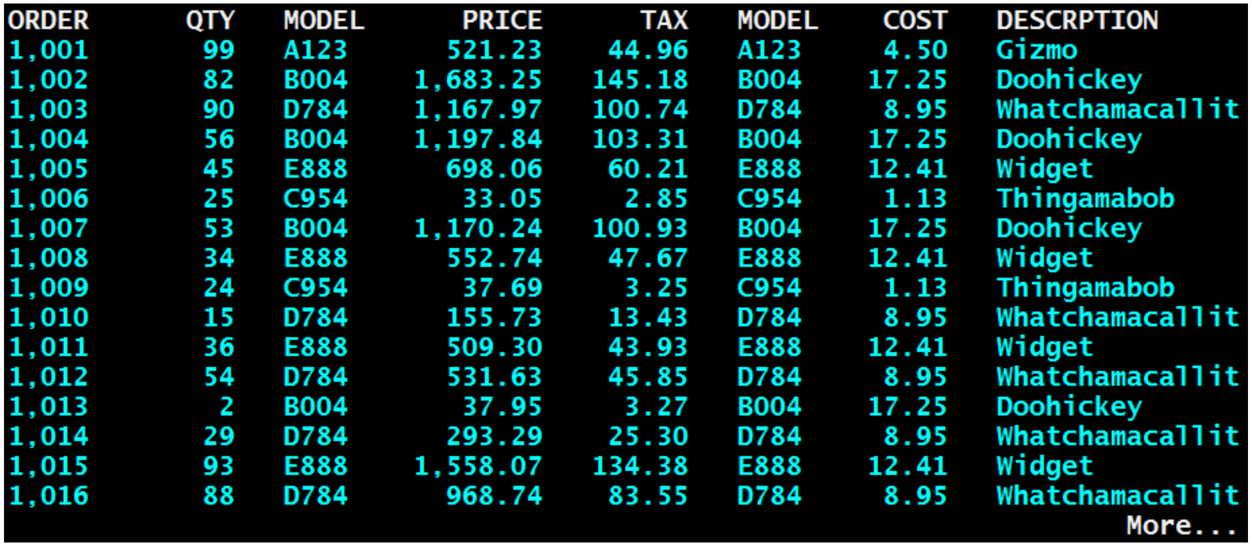
```
SENTENCE
A123 costs $4.50
B004 costs $17.25
C954 costs $1.13
D784 costs $8.95
E888 costs $12.41
```

Joining two Files or Tables

select * from orders join models on orders.model=models.model

File.Field Notation

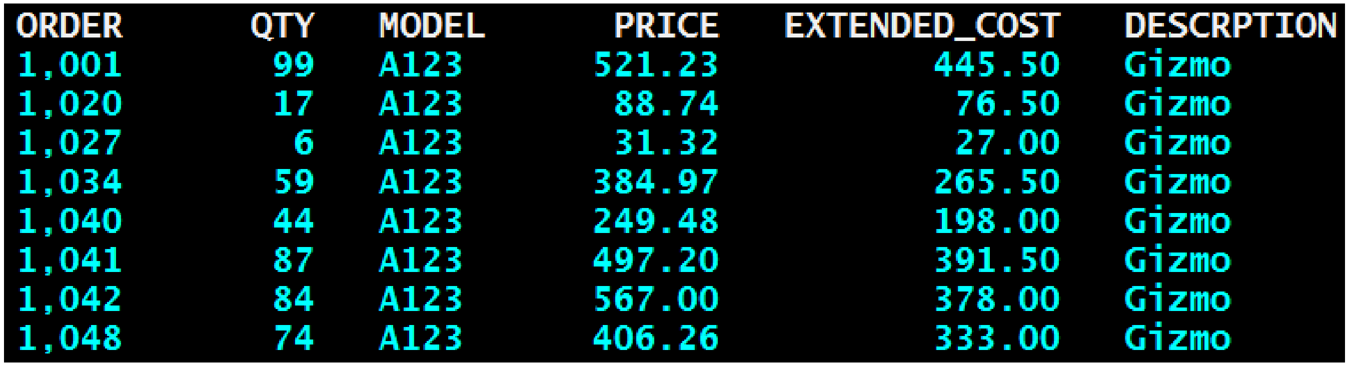
Fields from orders file **Fields from models file**



```
ORDER      QTY  MODEL       PRICE      TAX  MODEL     COST  DESCRPTION
1,001       99  A123       521.23    44.96  A123      4.50  Gizmo
1,002       82  B004     1,683.25   145.18  B004     17.25  Doohickey
1,003       90  D784     1,167.97   100.74  D784      8.95  Whatchamacallit
1,004       56  B004     1,197.84   103.31  B004     17.25  Doohickey
1,005       45  E888       698.06    60.21  E888     12.41  Widget
1,006       25  C954        33.05     2.85  C954      1.13  Thingamabob
1,007       53  B004     1,170.24   100.93  B004     17.25  Doohickey
1,008       34  E888       552.74    47.67  E888     12.41  Widget
1,009       24  C954        37.69     3.25  C954      1.13  Thingamabob
1,010       15  D784       155.73    13.43  D784      8.95  Whatchamacallit
1,011       36  E888       509.30    43.93  E888     12.41  Widget
1,012       54  D784       531.63    45.85  D784      8.95  Whatchamacallit
1,013        2  B004        37.95     3.27  B004     17.25  Doohickey
1,014       29  D784       293.29    25.30  D784      8.95  Whatchamacallit
1,015       93  E888     1,558.07   134.38  E888     12.41  Widget
1,016       88  D784       968.74    83.55  D784      8.95  Whatchamacallit
                                                                 More...
```

Sales Report for Model A123

```
select  order, qty, orders.model, price,
        cost*qty extended_cost, descrption
        from orders join models on
        orders.model=models.model
        where orders.model='A123'
```

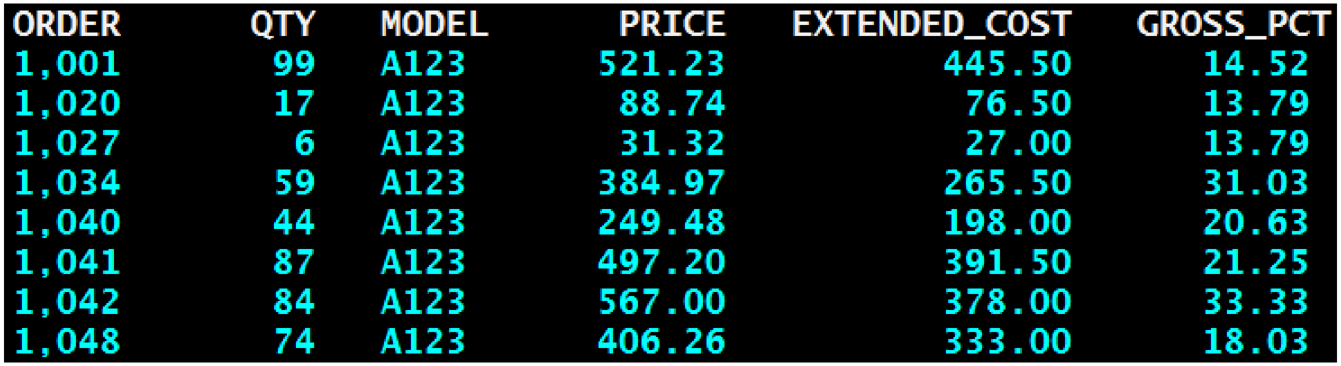


```
ORDER       QTY    MODEL        PRICE    EXTENDED_COST    DESCRPTION
1,001        99    A123        521.23           445.50    Gizmo
1,020        17    A123         88.74            76.50    Gizmo
1,027         6    A123         31.32            27.00    Gizmo
1,034        59    A123        384.97           265.50    Gizmo
1,040        44    A123        249.48           198.00    Gizmo
1,041        87    A123        497.20           391.50    Gizmo
1,042        84    A123        567.00           378.00    Gizmo
1,048        74    A123        406.26           333.00    Gizmo
```

Gross Profit % for Model A123

Gross Profit is defined as (Sales – Cost) / Sales

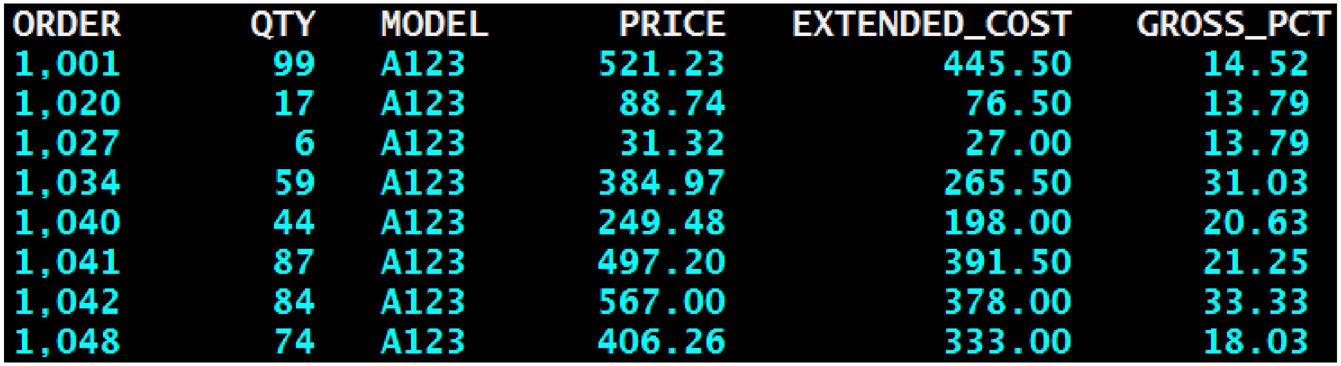
```
select  order, qty, orders.model, price,
        cost*qty extended_cost,
        cast((price - cost*qty) / price * 100 as
                dec(5,2)) gross_pct
        from orders join models on
        orders.model=models.model
        where orders.model='A123'
```



```
ORDER         QTY    MODEL       PRICE    EXTENDED_COST    GROSS_PCT
1,001          99    A123       521.23           445.50        14.52
1,020          17    A123        88.74            76.50        13.79
1,027           6    A123        31.32            27.00        13.79
1,034          59    A123       384.97           265.50        31.03
1,040          44    A123       249.48           198.00        20.63
1,041          87    A123       497.20           391.50        21.25
1,042          84    A123       567.00           378.00        33.33
1,048          74    A123       406.26           333.00        18.03
```

Updating Data

Let’s say we decide that the 6 A123’s on order 1027 should have been given to the customer for free.



```
ORDER        QTY   MODEL        PRICE   EXTENDED_COST   GROSS_PCT
1,001         99   A123        521.23          445.50       14.52
1,020         17   A123         88.74           76.50       13.79
1,027          6   A123         31.32           27.00       13.79
1,034         59   A123        384.97          265.50       31.03
1,040         44   A123        249.48          198.00       20.63
1,041         87   A123        497.20          391.50       21.25
1,042         84   A123        567.00          378.00       33.33
1,048         74   A123        406.26          333.00       18.03
```

update orders set price=0 where order=1027

CAREFUL! We’re now updating data!

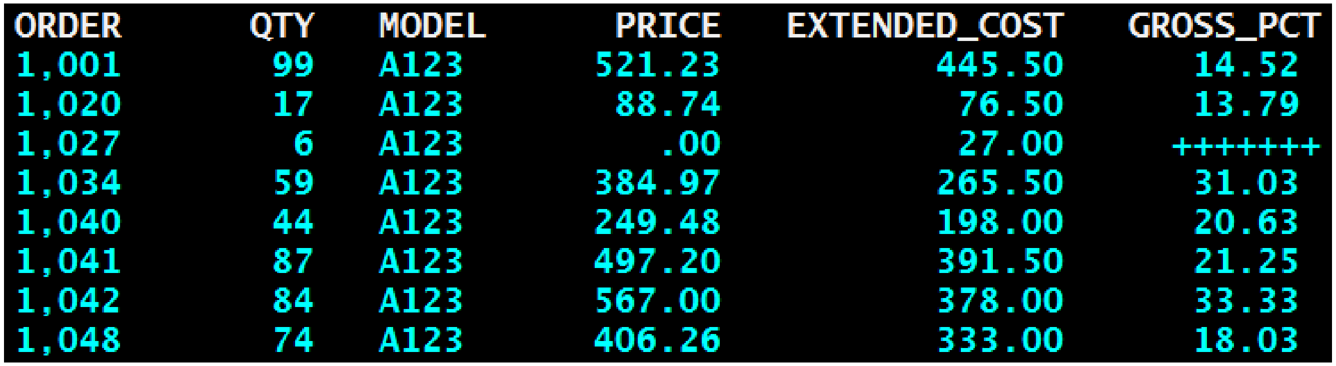
… and SQL returns with:

1 rows updated in ORDERS.

Gross Profit % for Model A123

Gross Profit is defined as (Sales – Cost) / Sales

```
select  order, qty, orders.model, price,
        cost*qty extended_cost,
        cast((price - cost*qty) / price * 100 as
              dec(5,2)) gross_pct
        from orders join models on
        orders.model=models.model
        where orders.model='A123'
```

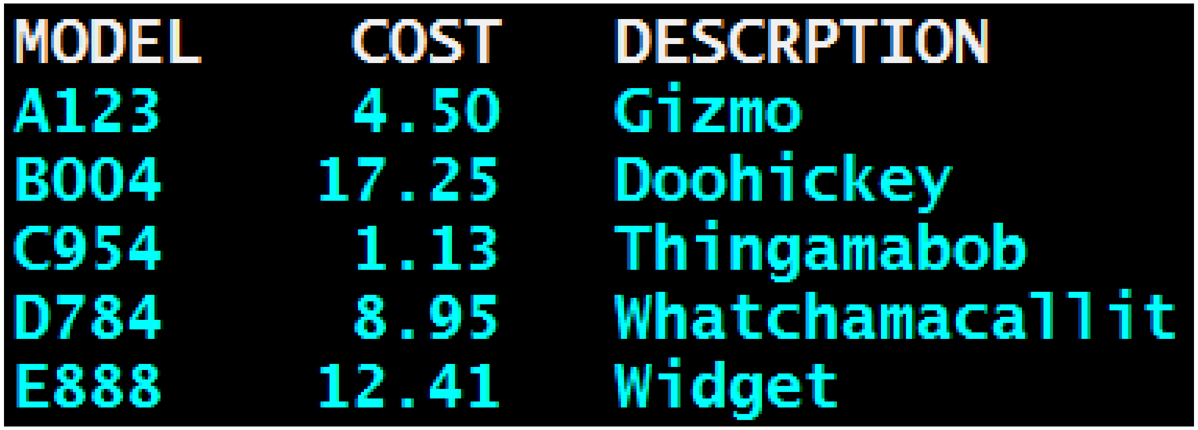


```
ORDER         QTY    MODEL          PRICE    EXTENDED_COST      GROSS_PCT
1,001          99    A123          521.23           445.50          14.52
1,020          17    A123           88.74            76.50          13.79
1,027           6    A123             .00            27.00        +++++++
1,034          59    A123          384.97           265.50          31.03
1,040          44    A123          249.48           198.00          20.63
1,041          87    A123          497.20           391.50          21.25
1,042          84    A123          567.00           378.00          33.33
1,048          74    A123          406.26           333.00          18.03
```

More Data Updates - Strings

We've decided we want all of our model descriptions to be plural. Let's review what we currently have:

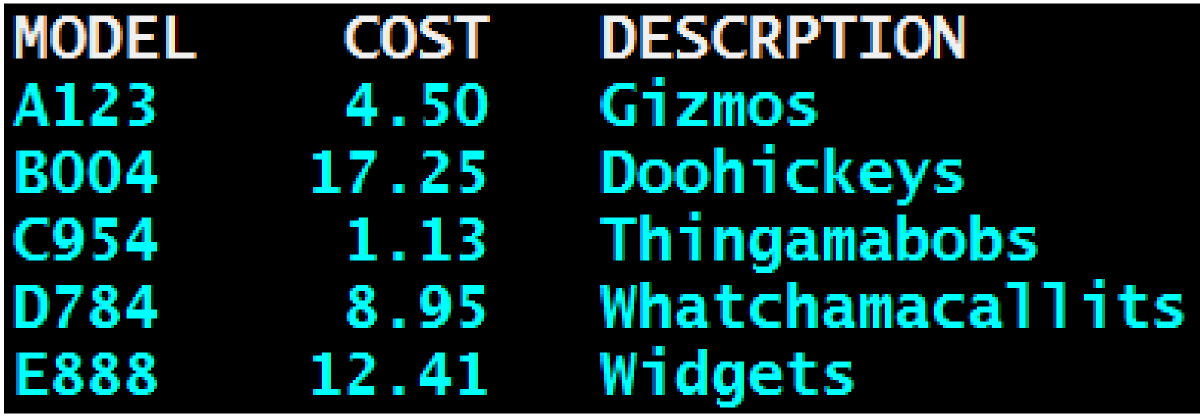
select * from models



```
MODEL      COST    DESCRPTION
A123       4.50    Gizmo
B004      17.25    Doohickey
C954       1.13    Thingamabob
D784       8.95    Whatchamacallit
E888      12.41    Widget
```

More Data Updates - Strings

```
update models
set descrption=trim(descrption)||'s'

select * from models
```



```
MODEL      COST    DESCRPTION
A123       4.50    Gizmos
B004      17.25    Doohickeys
C954       1.13    Thingamabobs
D784       8.95    Whatchamacallits
E888      12.41    Widgets
```

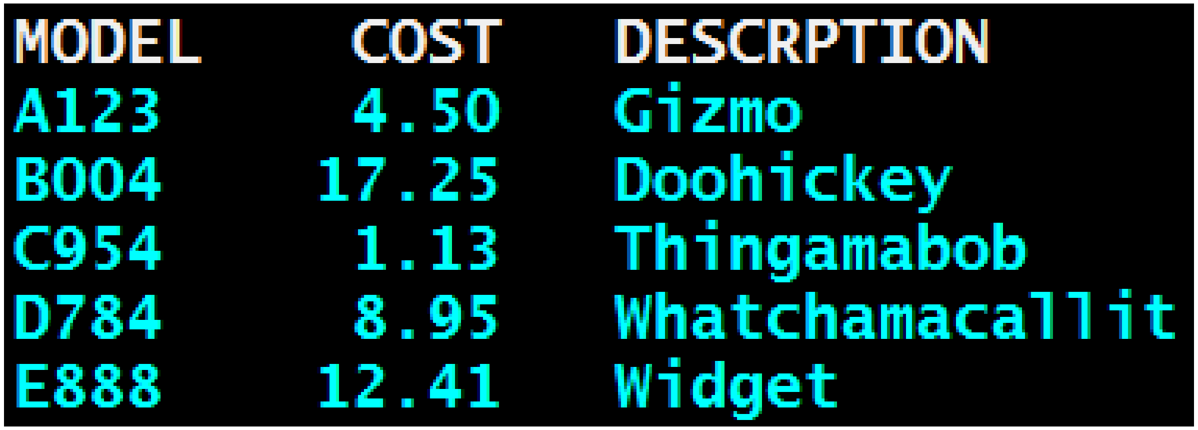
More Data Updates - Strings

Now we’ve changed our minds, and decided we’d rather stick with the singular:

```
update models
set   descrption = left(descrption,
      length(trim(descrption))-1)
```

More Data Updates - Strings

select * from models



```
MODEL       COST    DESCRPTION
A123        4.50    Gizmo
B004       17.25    Doohickey
C954        1.13    Thingamabob
D784        8.95    Whatchamacallit
E888       12.41    Widget
```

Another way - REPLACE

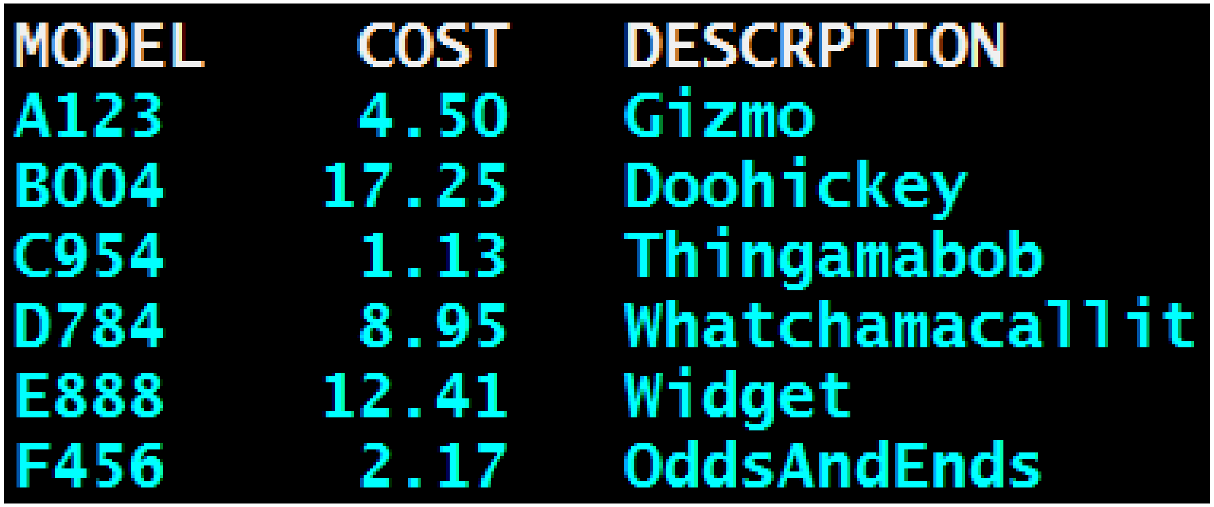
The REPLACE function replaces all occurrences of a text string with a different text string

```
update models set
descrption=replace(descrption,'s ','  ')
```

Adding Records to a File

insert into models values('F456',2.17,'OddsAndEnds')

select * from models

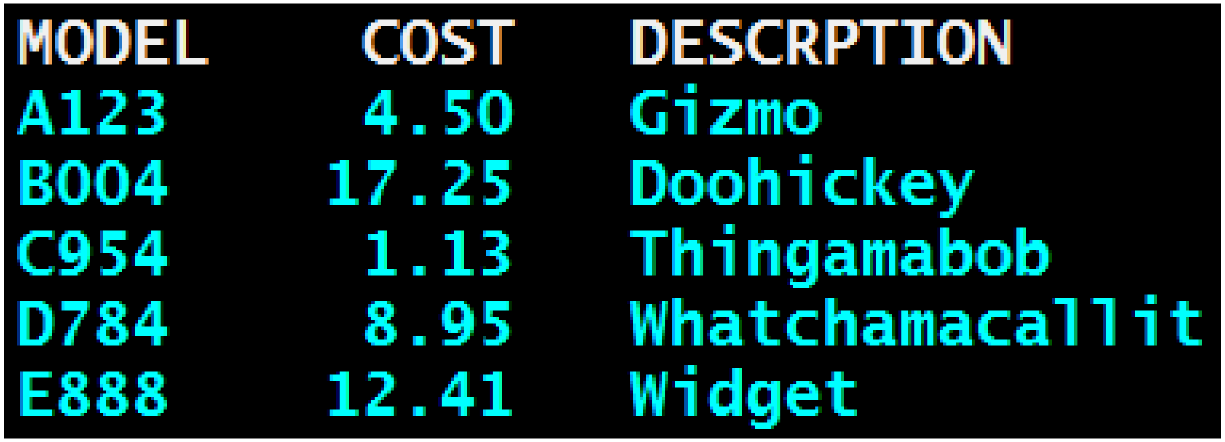


```
MODEL      COST    DESCRPTION
A123       4.50    Gizmo
B004      17.25    Doohickey
C954       1.13    Thingamabob
D784       8.95    Whatchamacallit
E888      12.41    Widget
F456       2.17    OddsAndEnds
```

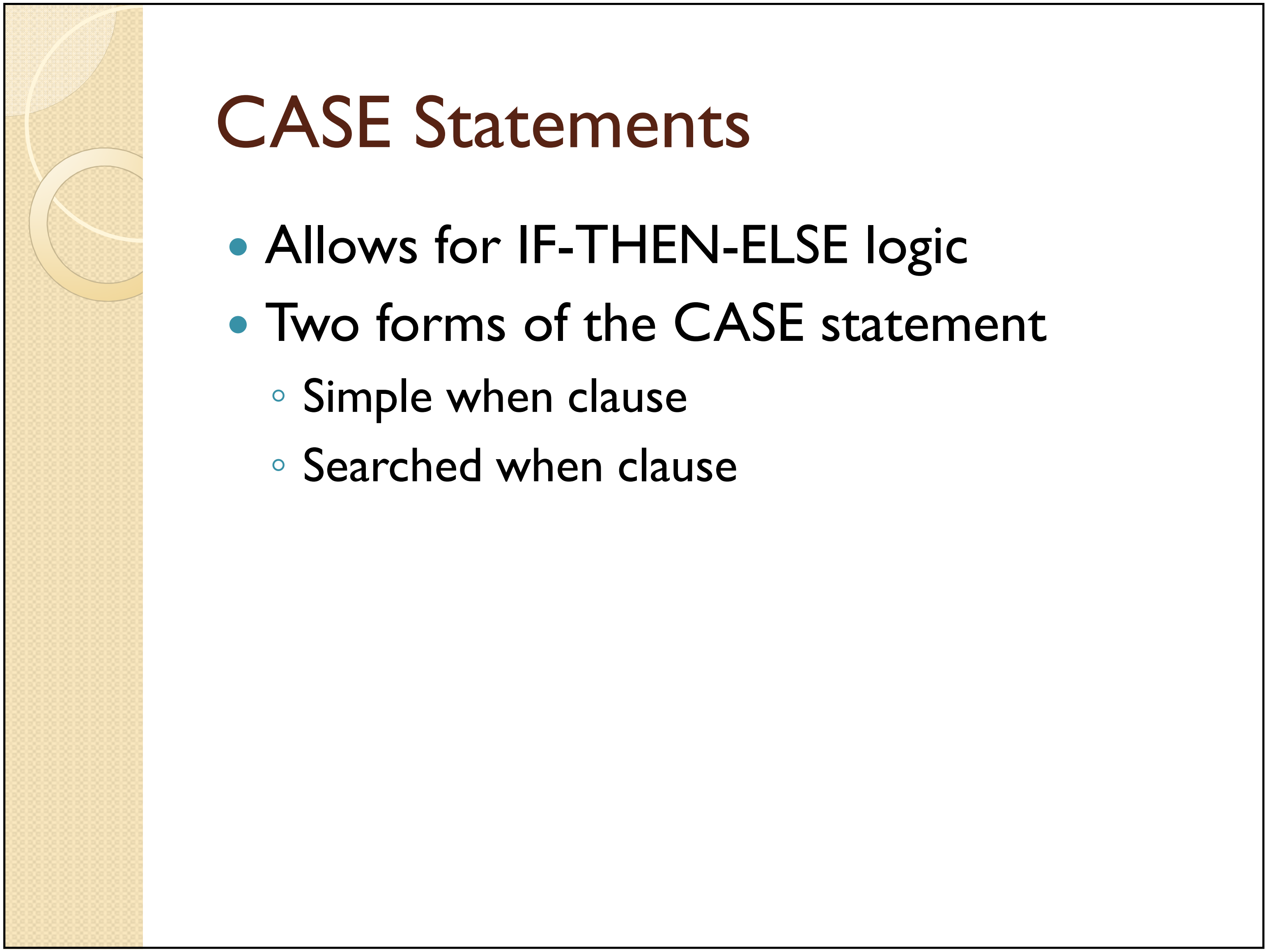
Deleting Records from a File

delete from models where model='F456‘

select * from models



```
MODEL      COST    DESCRPTION
A123       4.50    Gizmo
B004      17.25    Doohickey
C954       1.13    Thingamabob
D784       8.95    Whatchamacallit
E888      12.41    Widget
```

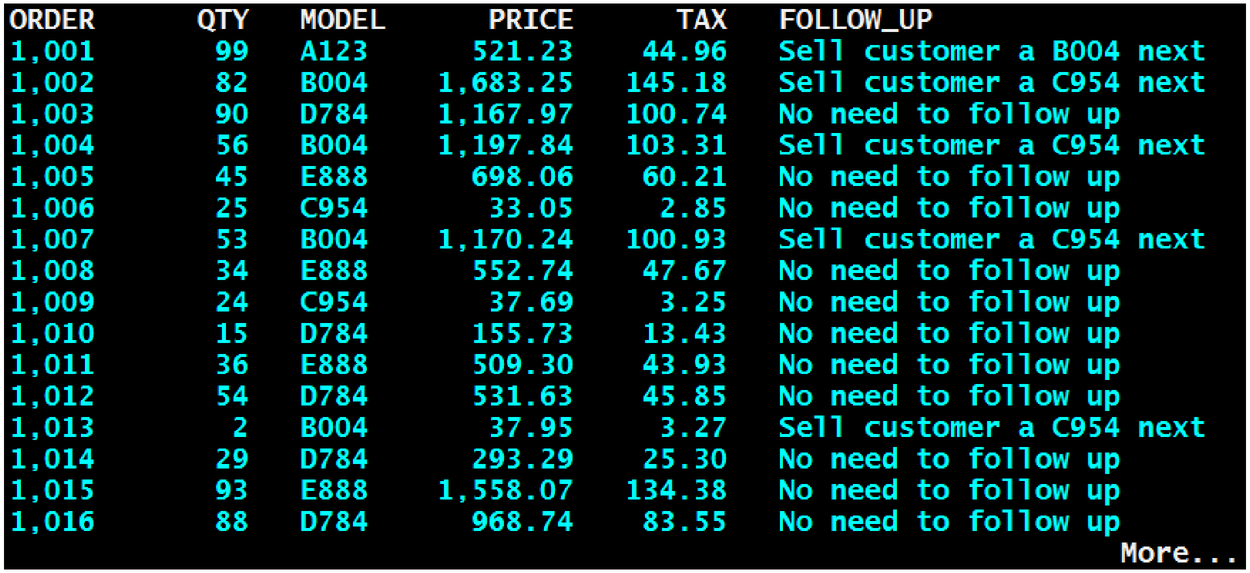


CASE Statements

- Allows for IF-THEN-ELSE logic
- Two forms of the CASE statement
  - Simple when clause
  - Searched when clause

Case – Simple When Clause

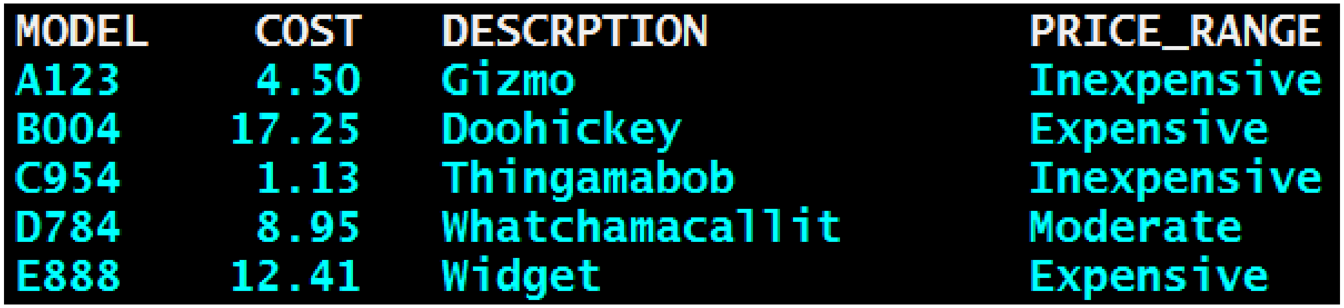
```
select orders.*, case model
        when 'A123' then 'Sell customer a B004 next'
        when 'B004' then 'Sell customer a C954 next'
        else 'No need to follow up'
    end as follow_up
    from orders
```



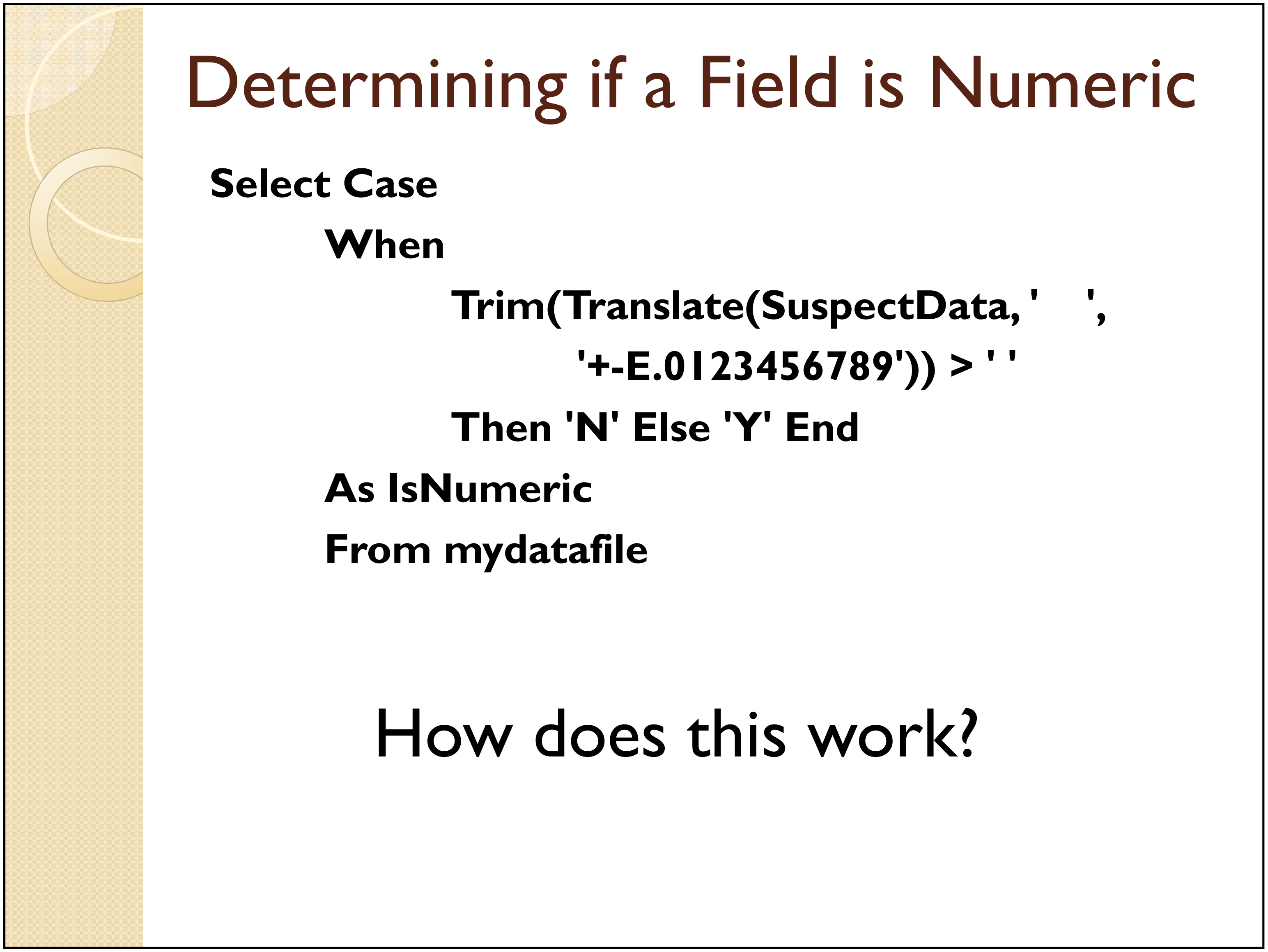
```
ORDER          QTY   MODEL       PRICE       TAX   FOLLOW_UP
1,001           99   A123       521.23     44.96   Sell customer a B004 next
1,002           82   B004     1,683.25    145.18   Sell customer a C954 next
1,003           90   D784     1,167.97    100.74   No need to follow up
1,004           56   B004     1,197.84    103.31   Sell customer a C954 next
1,005           45   E888       698.06     60.21   No need to follow up
1,006           25   C954        33.05      2.85   No need to follow up
1,007           53   B004     1,170.24    100.93   Sell customer a C954 next
1,008           34   E888       552.74     47.67   No need to follow up
1,009           24   C954        37.69      3.25   No need to follow up
1,010           15   D784       155.73     13.43   No need to follow up
1,011           36   E888       509.30     43.93   No need to follow up
1,012           54   D784       531.63     45.85   No need to follow up
1,013            2   B004        37.95      3.27   Sell customer a C954 next
1,014           29   D784       293.29     25.30   No need to follow up
1,015           93   E888     1,558.07    134.38   No need to follow up
1,016           88   D784       968.74     83.55   No need to follow up
                                                                       More...
```

Case – Searched When Clause

```
select models.*, case
        when cost <=5 then 'Inexpensive'
        when cost <10 then 'Moderate'
        else 'Expensive'
     end as price_range
     from models
     order by model
```



```
MODEL      COST    DESCRPTION                    PRICE_RANGE
A123       4.50    Gizmo                         Inexpensive
B004      17.25    Doohickey                     Expensive
C954       1.13    Thingamabob                   Inexpensive
D784       8.95    Whatchamacallit               Moderate
E888      12.41    Widget                        Expensive
```



Determining if a Field is Numeric

```
Select Case
        When
                Trim(Translate(SuspectData, '    ',
                        '+-E.0123456789')) > ' '
                Then 'N' Else 'Y' End
        As IsNumeric
        From mydatafile
```

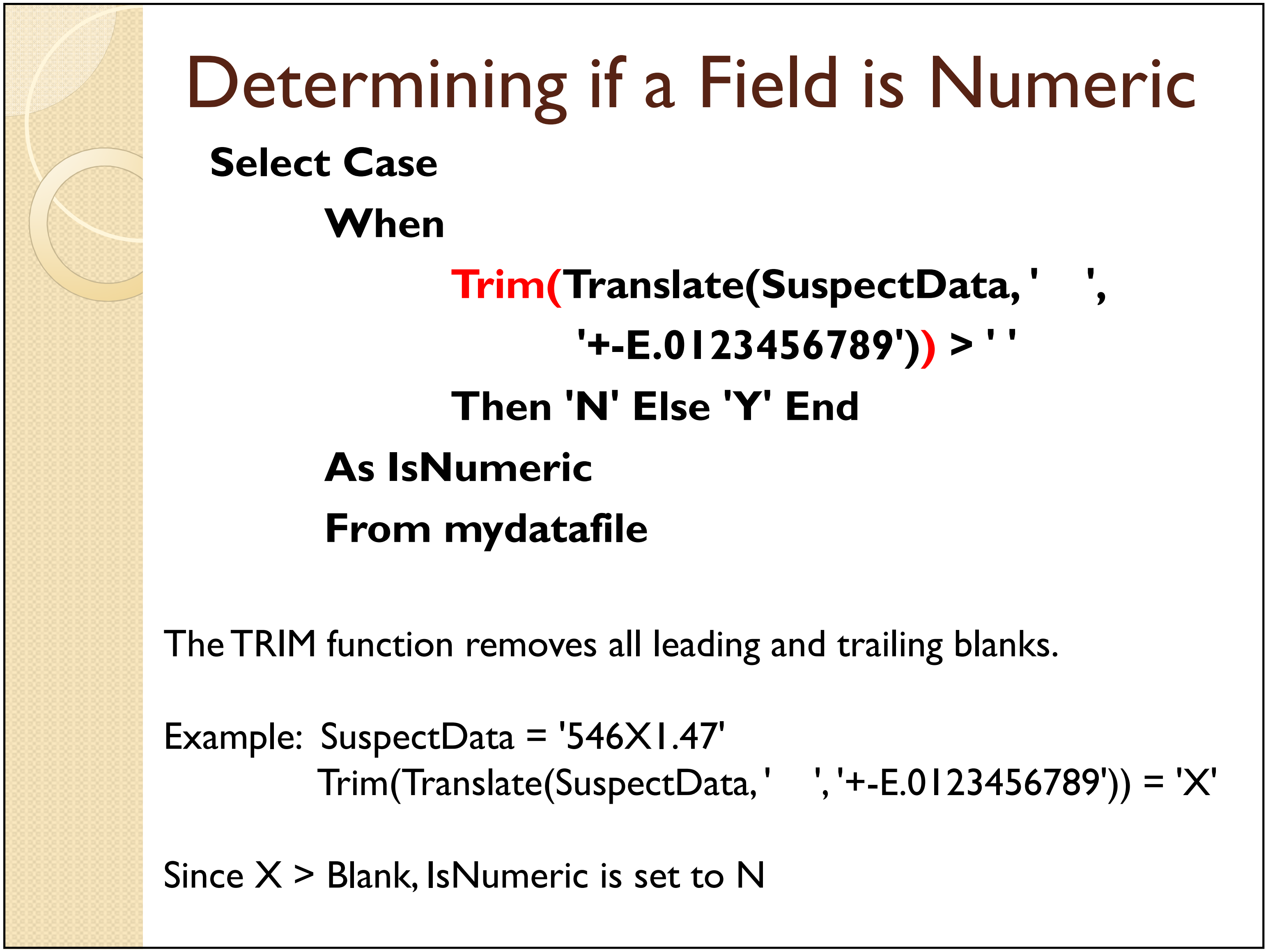
How does this work?

Determining if a Field is Numeric

```
Select Case
        When
                Trim(Translate(SuspectData, '    ',
                        '+-E.0123456789')) > ' '
                Then 'N' Else 'Y' End
        As IsNumeric
        From mydatafile
```

The TRANSLATE function changes all characters in SuspectData that ARE NOT one of +-E.0123456789 to a blank (value specified in second parameter).

```
Example:   SuspectData = '546X1.47'
           Translate(SuspectData, '    ', '+-E.0123456789') = '   X    '
                (3 spaces before the X, 4 spaces after)
```



Determining if a Field is Numeric

```
Select Case
     When
          Trim(Translate(SuspectData, '    ',
               '+-E.0123456789')) > ' '
          Then 'N' Else 'Y' End
     As IsNumeric
     From mydatafile
```

The TRIM function removes all leading and trailing blanks.

Example: SuspectData = '546X1.47'
          Trim(Translate(SuspectData, '   ', '+-E.0123456789')) = 'X'

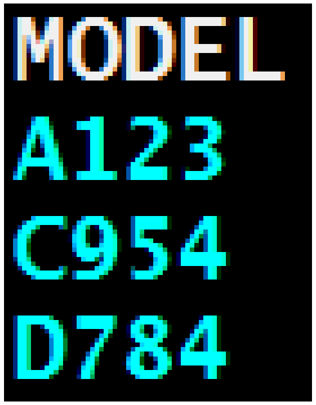
Since X > Blank, IsNumeric is set to N

Subselects

We want all of our orders where the unit cost < $10

Unit cost is in our MODELS file

Select model from models where cost < 10



```
MODEL
A123
C954
D784
```

Subselects

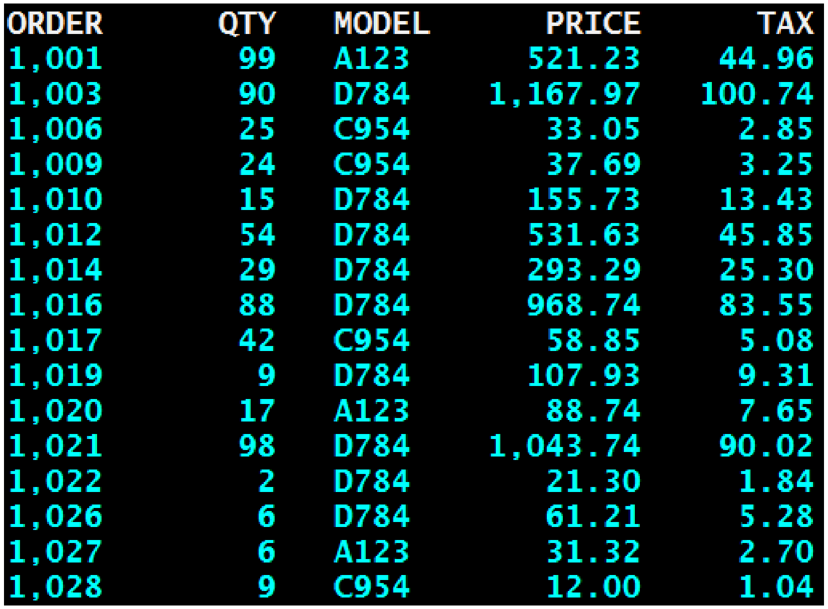
We want all of our orders where the unit cost < $10

Unit cost is in our MODELS file

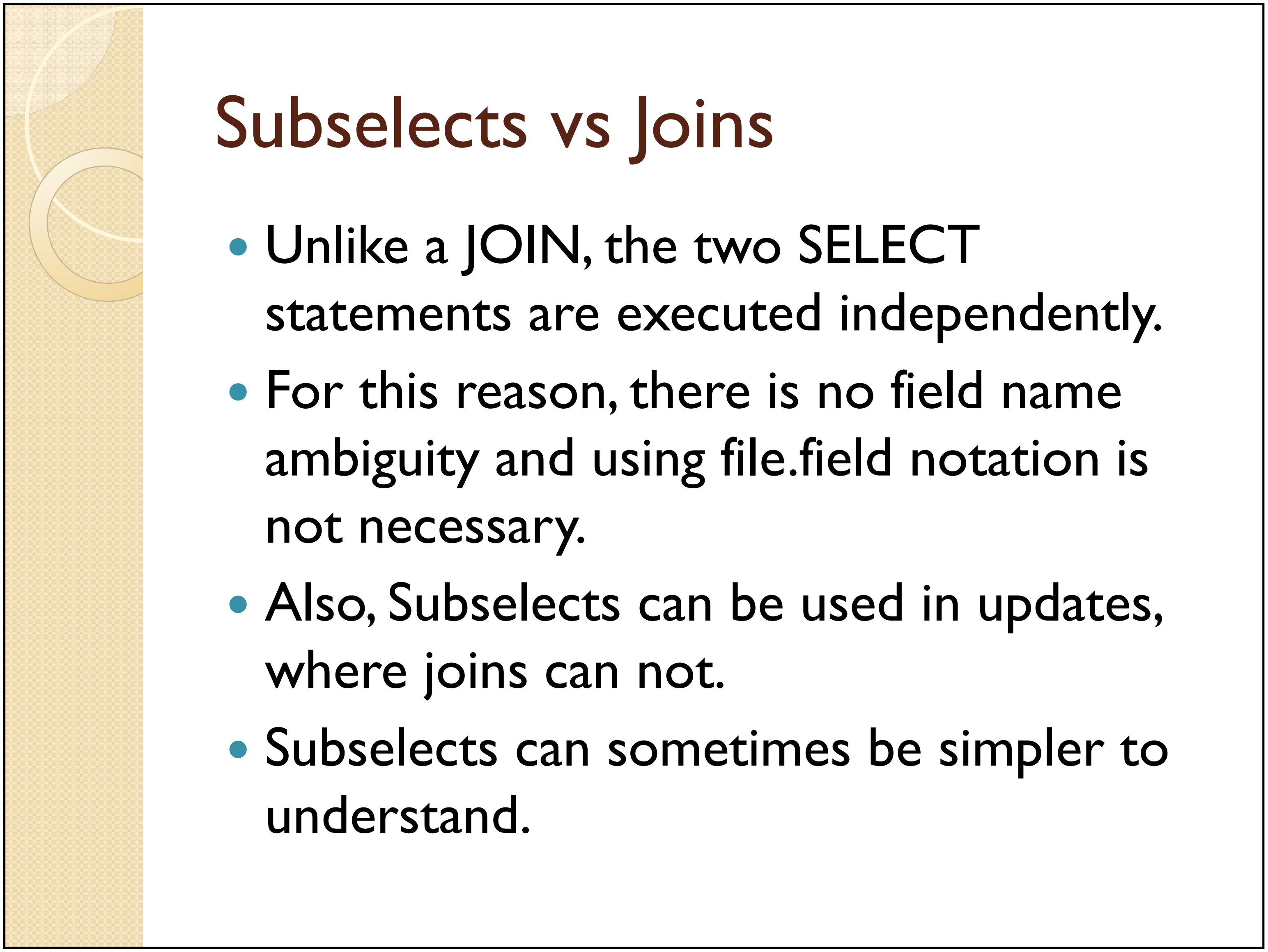
Orders are in our ORDERS file

Select * from orders where model in

(Select model from models where cost < 10)



```
ORDER         QTY   MODEL        PRICE       TAX
1,001          99   A123        521.23     44.96
1,003          90   D784      1,167.97    100.74
1,006          25   C954         33.05      2.85
1,009          24   C954         37.69      3.25
1,010          15   D784        155.73     13.43
1,012          54   D784        531.63     45.85
1,014          29   D784        293.29     25.30
1,016          88   D784        968.74     83.55
1,017          42   C954         58.85      5.08
1,019           9   D784        107.93      9.31
1,020          17   A123         88.74      7.65
1,021          98   D784      1,043.74     90.02
1,022           2   D784         21.30      1.84
1,026           6   D784         61.21      5.28
1,027           6   A123         31.32      2.70
1,028           9   C954         12.00      1.04
```



Subselects vs Joins

- Unlike a JOIN, the two SELECT statements are executed independently.
- For this reason, there is no field name ambiguity and using file.field notation is not necessary.
- Also, Subselects can be used in updates, where joins can not.
- Subselects can sometimes be simpler to understand.

Scalar Subselects

- Just like a subselect, but the second SELECT statement is based on data returned from the first SELECT statement
- In our previous example,

```
Select * from orders where model in
        (Select model from models where cost < 10)
```

the second SELECT statement could be run totally independent of the first SELECT statement.

Scalar Subselects

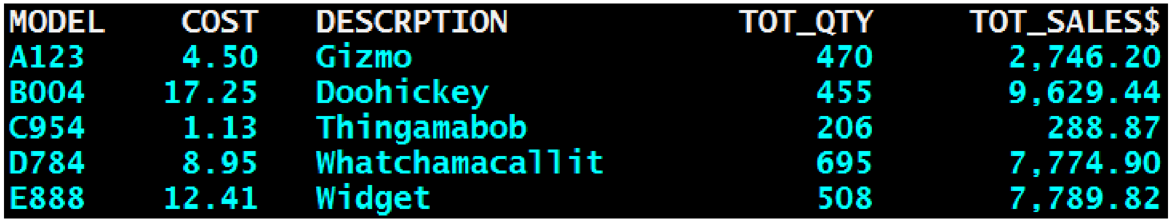
We want to see our models file, but include summary sales data.

select models.*,

**(select cast(sum(qty) as dec(5,0))
from orders where models.model =
orders.model) TOT_QTY,**

**(select cast(sum(price) as dec(9,2))
from orders where models.model =
orders.model) TOT_SALES$**

from models order by model



```
MODEL      COST    DESCRPTION              TOT_QTY       TOT_SALES$
A123       4.50    Gizmo                       470         2,746.20
B004      17.25    Doohickey                   455         9,629.44
C954       1.13    Thingamabob                 206           288.87
D784       8.95    Whatchamacallit             695         7,774.90
E888      12.41    Widget                      508         7,789.82
```

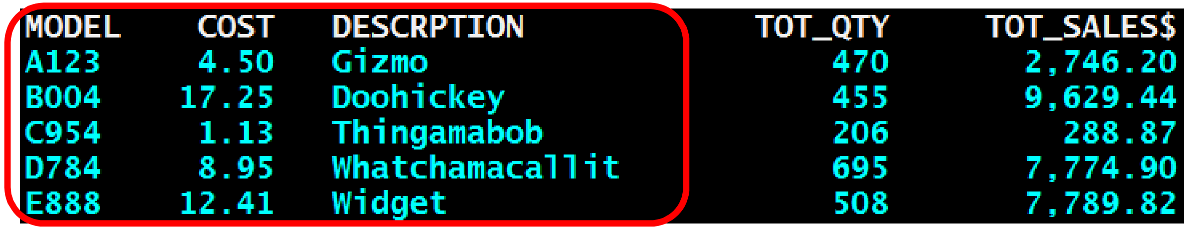
Scalar Subselects

We want to see our models file, but include summary sales data.

Outer SELECT

Outer SELECT

```
select models.*,
       (select cast(sum(qty) as dec(5,0))
       from orders where models.model =
       orders.model) TOT_QTY,
       (select cast(sum(price) as dec(9,2))
       from orders where models.model =
       orders.model) TOT_SALES$
from models order by model
```



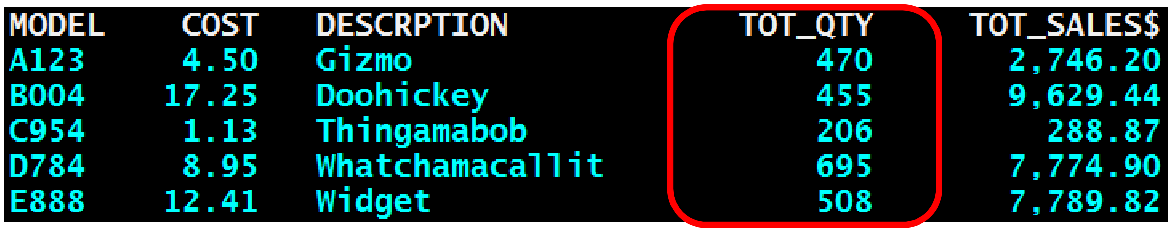
```
MODEL      COST   DESCRPTION                 TOT_QTY       TOT_SALES$
A123       4.50   Gizmo                          470         2,746.20
B004      17.25   Doohickey                      455         9,629.44
C954       1.13   Thingamabob                    206           288.87
D784       8.95   Whatchamacallit                695         7,774.90
E888      12.41   Widget                         508         7,789.82
```

Scalar Subselects

We want to see our models file, but include summary sales data.

Inner
SELECT # 1

```
select models.*,
      (select cast(sum(qty) as dec(5,0))
      from orders where models.model =
      orders.model) TOT_QTY,
      (select cast(sum(price) as dec(9,2))
      from orders where models.model =
      orders.model) TOT_SALES$
from models order by model
```



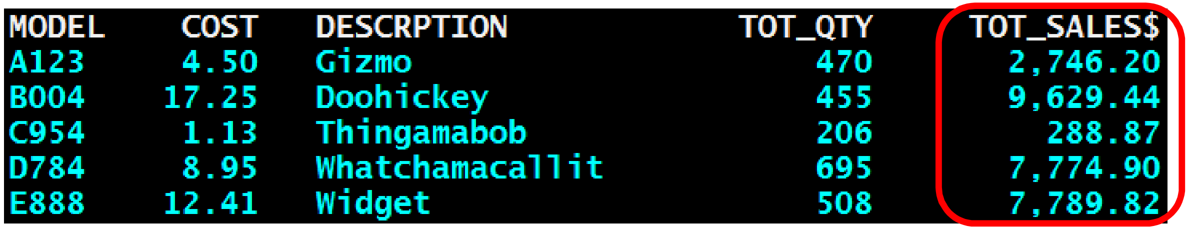
```
MODEL      COST    DESCRPTION                TOT_QTY        TOT_SALES$
A123       4.50    Gizmo                         470          2,746.20
B004      17.25    Doohickey                     455          9,629.44
C954       1.13    Thingamabob                   206            288.87
D784       8.95    Whatchamacallit               695          7,774.90
E888      12.41    Widget                        508          7,789.82
```

Scalar Subselects

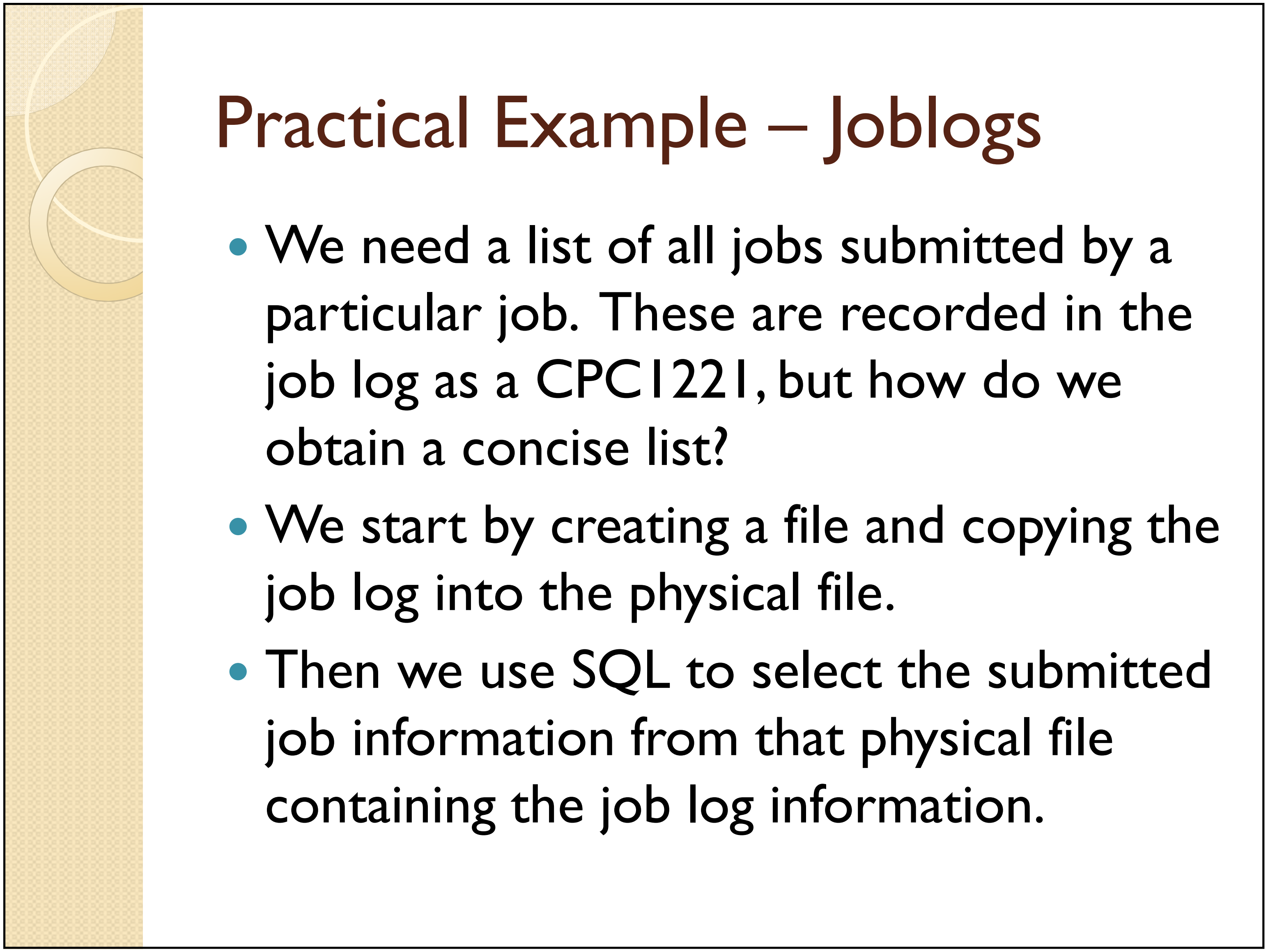
We want to see our models file, but include summary sales data.

```
select models.*,
        (select cast(sum(qty) as dec(5,0))
        from orders where models.model =
        orders.model) TOT_QTY,
        (select cast(sum(price) as dec(9,2))
        from orders where models.model =
        orders.model) TOT_SALES$
from models order by model
```

Inner SELECT # 2

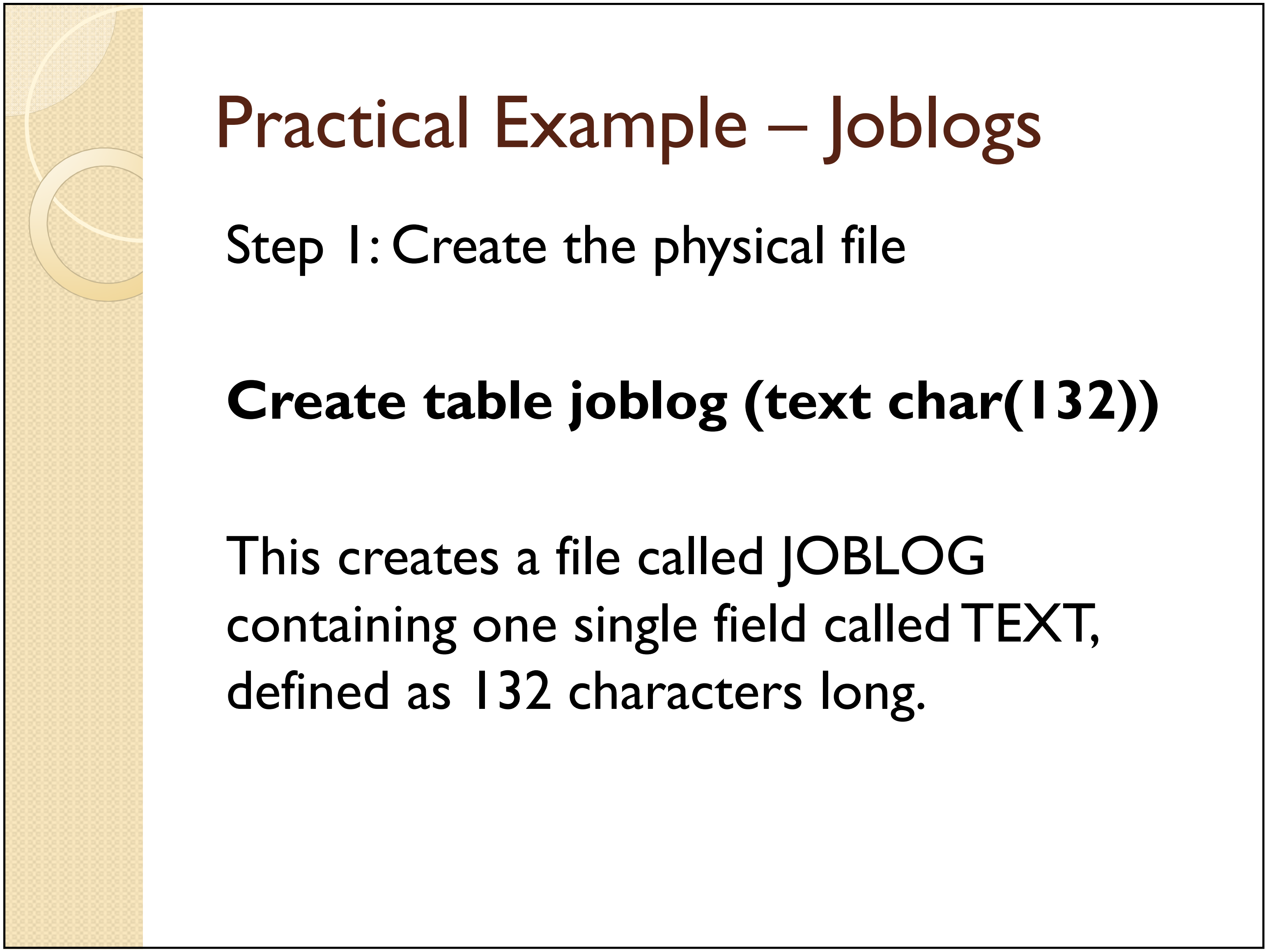


```
MODEL      COST    DESCRPTION               TOT_QTY      TOT_SALES$
A123       4.50    Gizmo                        470        2,746.20
B004      17.25    Doohickey                    455        9,629.44
C954       1.13    Thingamabob                  206          288.87
D784       8.95    Whatchamacallit              695        7,774.90
E888      12.41    Widget                       508        7,789.82
```



Practical Example – Joblogs

- We need a list of all jobs submitted by a particular job. These are recorded in the job log as a CPC1221, but how do we obtain a concise list?
- We start by creating a file and copying the job log into the physical file.
- Then we use SQL to select the submitted job information from that physical file containing the job log information.



Practical Example – Joblogs

Step 1: Create the physical file

Create table joblog (text char(132))

This creates a file called JOBLOG containing one single field called TEXT, defined as 132 characters long.

Practical Example – Joblogs

Step 2: Copy the joblog spool file to the physical file

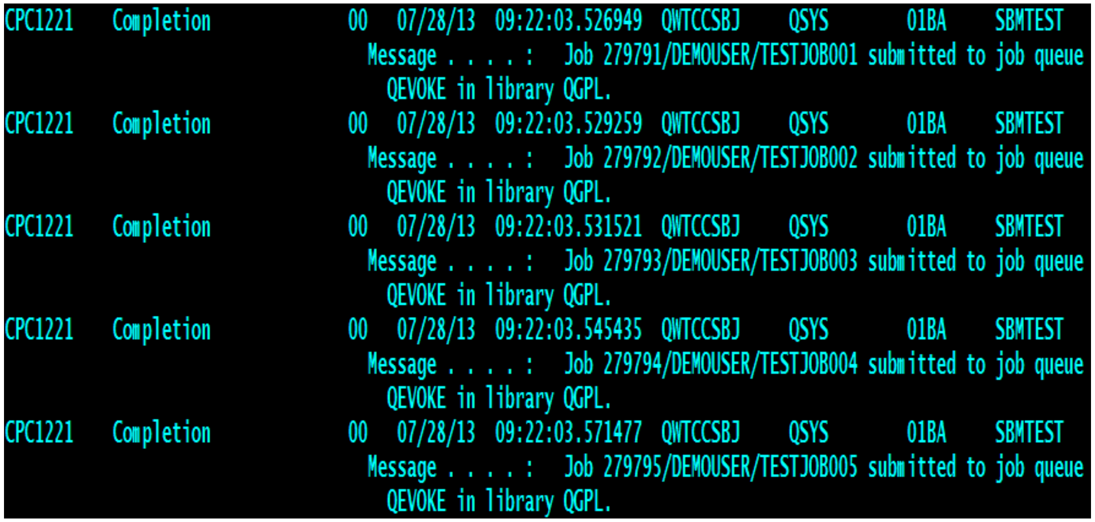
```
CPYSPLF FILE(QPJOBLOG)
TOFILE(JOBLOG)
JOB(279788/DEMOUSER/MAINJOB)
```

Practical Example – Joblogs

Step 2: Copy the joblog spool file to the physical file

Let's see what we have so far…

Select * from joblog

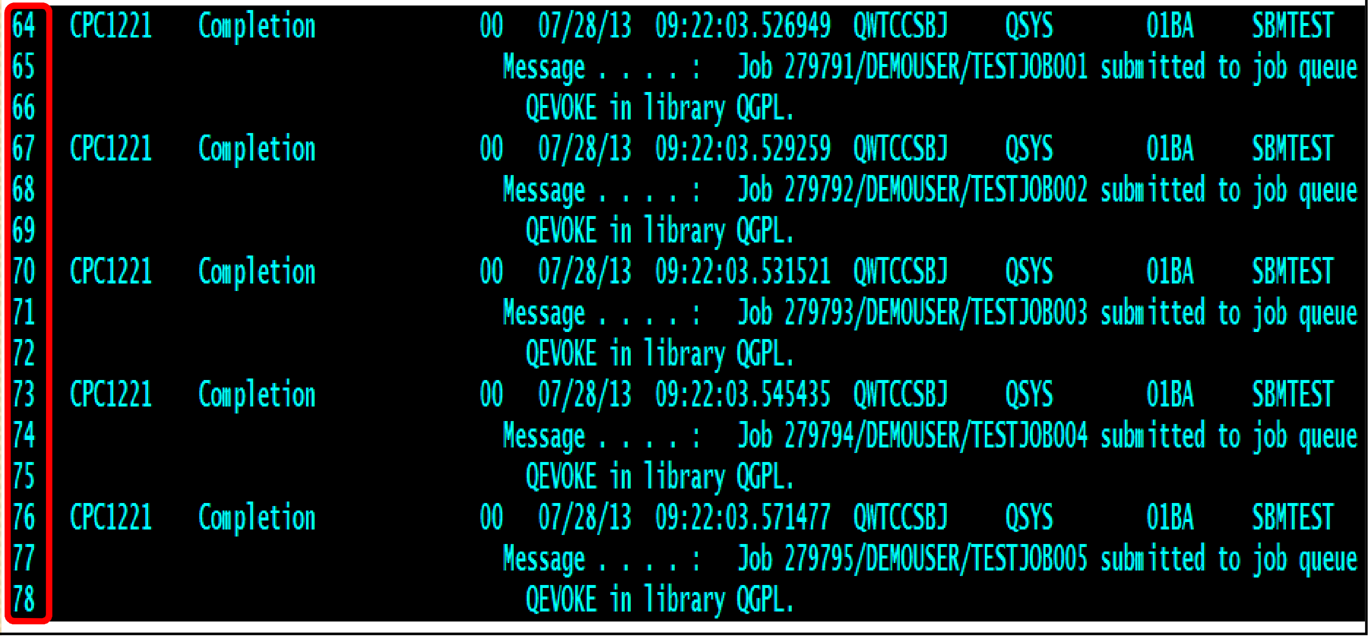


```
CPC1221   Completion         00  07/28/13  09:22:03.526949  QWTCCSBJ    QSYS      01BA    SBMTEST
                               Message . . . . :   Job 279791/DEMOUSER/TESTJOB001 submitted to job queue
                                 QEVOKE in library QGPL.
CPC1221   Completion         00  07/28/13  09:22:03.529259  QWTCCSBJ    QSYS      01BA    SBMTEST
                               Message . . . . :   Job 279792/DEMOUSER/TESTJOB002 submitted to job queue
                                 QEVOKE in library QGPL.
CPC1221   Completion         00  07/28/13  09:22:03.531521  QWTCCSBJ    QSYS      01BA    SBMTEST
                               Message . . . . :   Job 279793/DEMOUSER/TESTJOB003 submitted to job queue
                                 QEVOKE in library QGPL.
CPC1221   Completion         00  07/28/13  09:22:03.545435  QWTCCSBJ    QSYS      01BA    SBMTEST
                               Message . . . . :   Job 279794/DEMOUSER/TESTJOB004 submitted to job queue
                                 QEVOKE in library QGPL.
CPC1221   Completion         00  07/28/13  09:22:03.571477  QWTCCSBJ    QSYS      01BA    SBMTEST
                               Message . . . . :   Job 279795/DEMOUSER/TESTJOB005 submitted to job queue
                                 QEVOKE in library QGPL.
```

Practical Example – Joblogs

Let’s try using relative record numbers:

select rrn(joblog), text from joblog

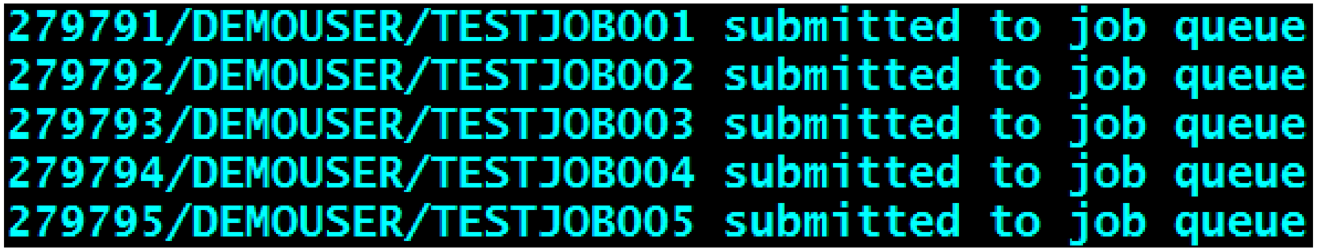


```
64  CPC1221   Completion            00  07/28/13  09:22:03.526949  QWTCCSBJ   QSYS     01BA    SBMTEST
65                                     Message . . . . :   Job 279791/DEMOUSER/TESTJOB001 submitted to job queue
66                                       QEVOKE in library QGPL.
67  CPC1221   Completion            00  07/28/13  09:22:03.529259  QWTCCSBJ   QSYS     01BA    SBMTEST
68                                     Message . . . . :   Job 279792/DEMOUSER/TESTJOB002 submitted to job queue
69                                       QEVOKE in library QGPL.
70  CPC1221   Completion            00  07/28/13  09:22:03.531521  QWTCCSBJ   QSYS     01BA    SBMTEST
71                                     Message . . . . :   Job 279793/DEMOUSER/TESTJOB003 submitted to job queue
72                                       QEVOKE in library QGPL.
73  CPC1221   Completion            00  07/28/13  09:22:03.545435  QWTCCSBJ   QSYS     01BA    SBMTEST
74                                     Message . . . . :   Job 279794/DEMOUSER/TESTJOB004 submitted to job queue
75                                       QEVOKE in library QGPL.
76  CPC1221   Completion            00  07/28/13  09:22:03.571477  QWTCCSBJ   QSYS     01BA    SBMTEST
77                                     Message . . . . :   Job 279795/DEMOUSER/TESTJOB005 submitted to job queue
78                                       QEVOKE in library QGPL.
```

Practical Example – Joblogs

Step 3: Extract the data we need.

```
select substr(text,62,100) from joblog
        where rrn(joblog)-1 in
        (select rrn(joblog) from joblog
         where text like '%CPC1221%')
```



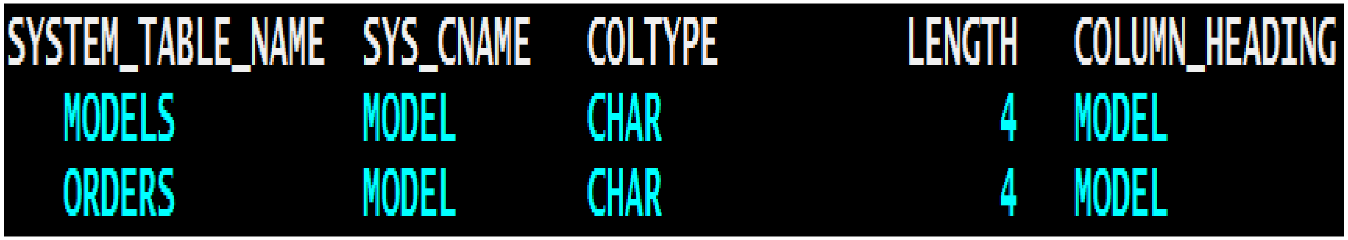
```
279791/DEMOUSER/TESTJOB001 submitted to job queue
279792/DEMOUSER/TESTJOB002 submitted to job queue
279793/DEMOUSER/TESTJOB003 submitted to job queue
279794/DEMOUSER/TESTJOB004 submitted to job queue
279795/DEMOUSER/TESTJOB005 submitted to job queue
```

Searching for a Specific Field

We want to find all files with the field MODEL.

File SYSCOLUMNS in QSYS2 contains all fields in all files on the system i.

```
select  sys_tname, sys_cname,
        coltype, length, column_heading
        from syscolumns
        where column_name = 'MODEL'
```

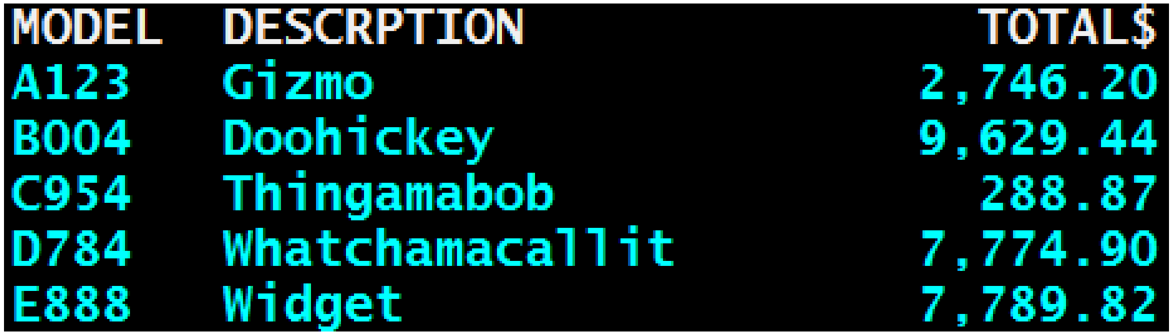


```
SYSTEM_TABLE_NAME  SYS_CNAME   COLTYPE          LENGTH  COLUMN_HEADING
   MODELS          MODEL       CHAR                  4  MODEL
   ORDERS          MODEL       CHAR                  4  MODEL
```

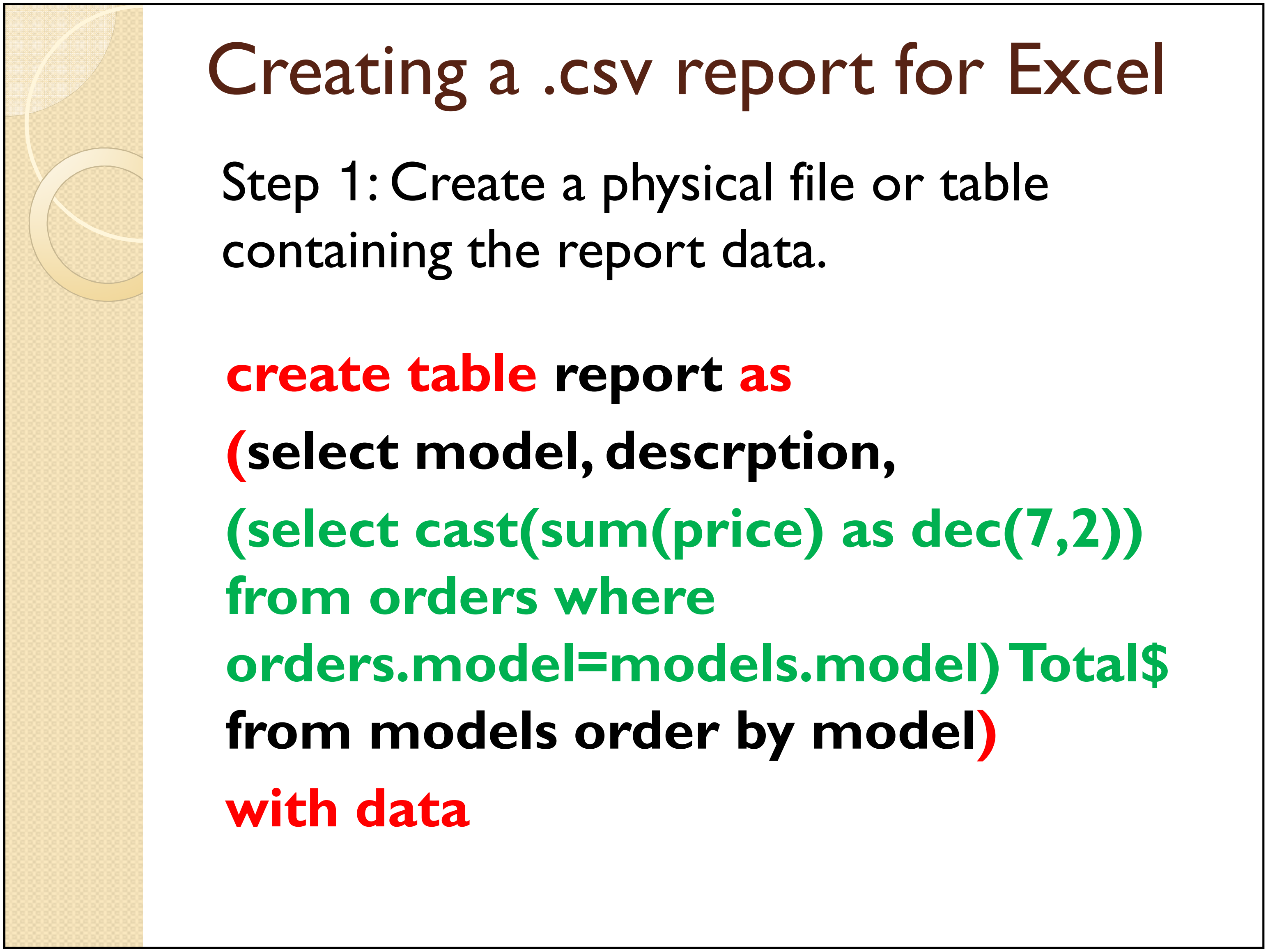
Creating a .csv report for Excel

We'd like to export a sales summary report to be viewed in Excel.

```
select model, descrption,
(select cast(sum(price) as dec(7,2))
from orders where
orders.model=models.model) Total$
from models order by model
```



```
MODEL  DESCRPTION               TOTAL$
A123   Gizmo                  2,746.20
B004   Doohickey              9,629.44
C954   Thingamabob              288.87
D784   Whatchamacallit        7,774.90
E888   Widget                 7,789.82
```



Creating a .csv report for Excel

Step 1: Create a physical file or table containing the report data.

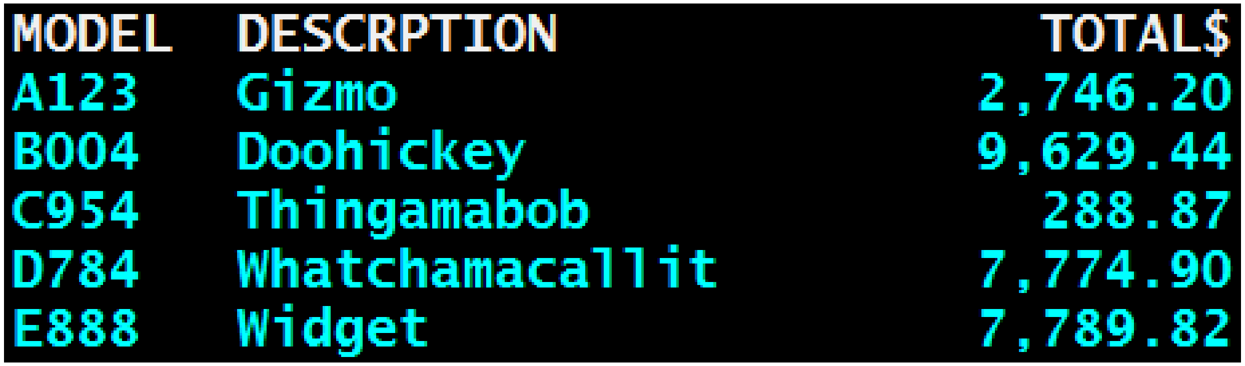
```
create table report as
(select model, descrption,
(select cast(sum(price) as dec(7,2))
from orders where
orders.model=models.model) Total$
from models order by model)
with data
```

Creating a .csv report for Excel

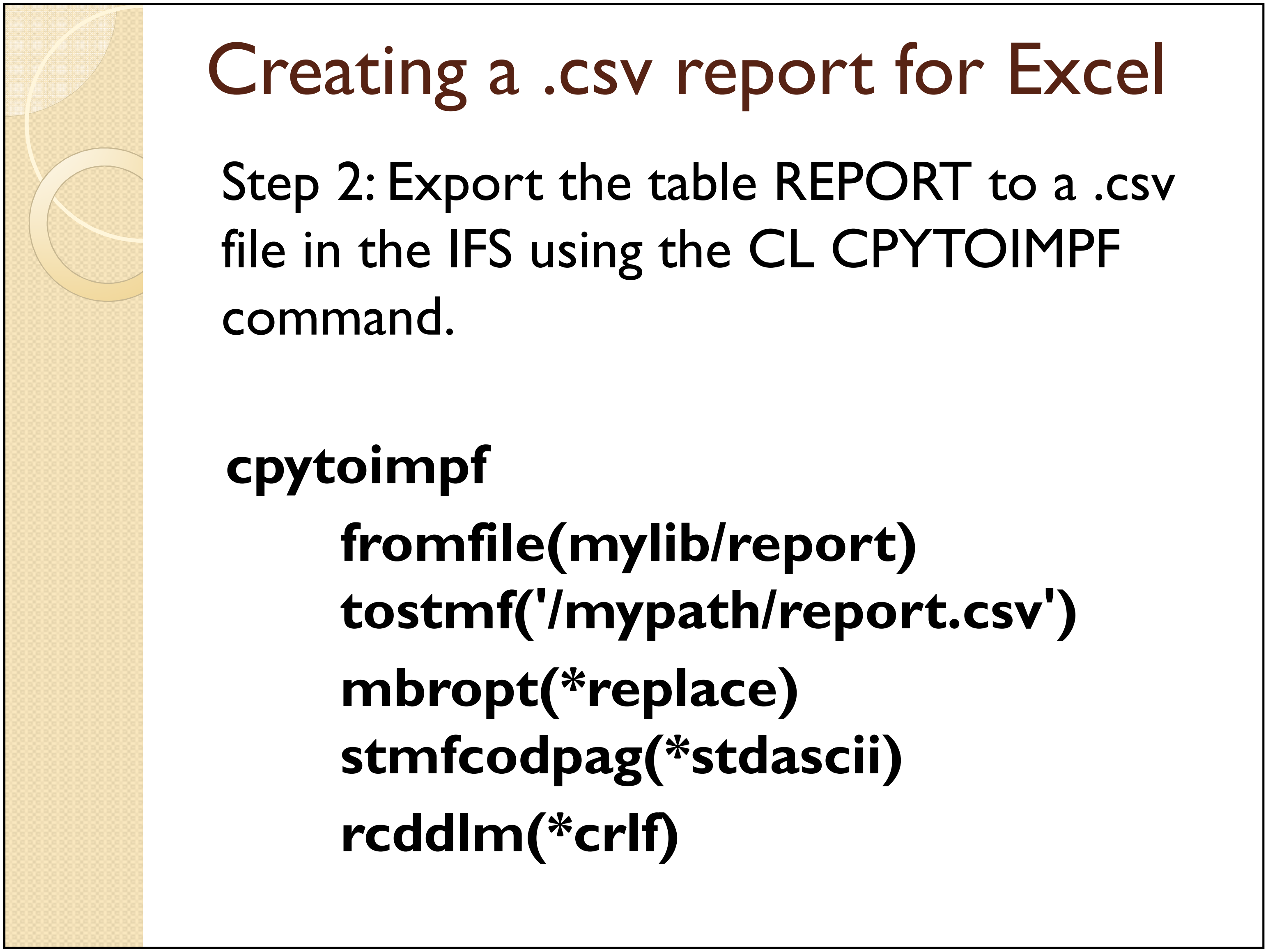
Step 1: Create a physical file or table containing the report data.

Let’s see what we have.

Select * from report



```
MODEL   DESCRPTION                  TOTAL$
A123    Gizmo                     2,746.20
B004    Doohickey                 9,629.44
C954    Thingamabob                 288.87
D784    Whatchamacallit           7,774.90
E888    Widget                    7,789.82
```



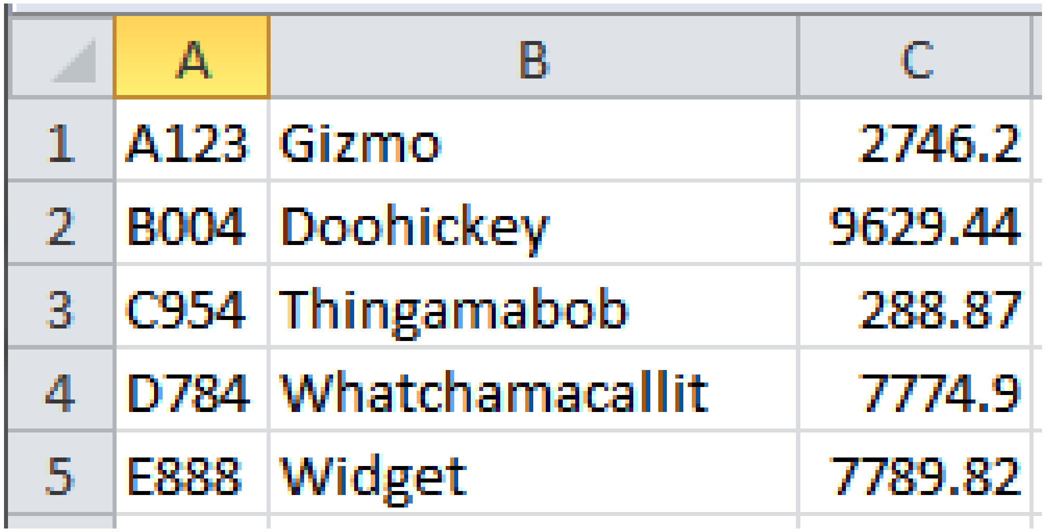
Creating a .csv report for Excel

Step 2: Export the table REPORT to a .csv file in the IFS using the CL CPYTOIMPF command.

```
cpytoimpf
        fromfile(mylib/report)
        tostmf('/mypath/report.csv')
        mbropt(*replace)
        stmfcodpag(*stdascii)
        rcddlm(*crlf)
```

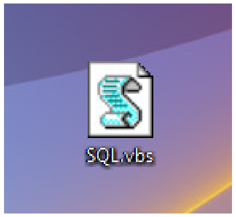
Creating a .csv report for Excel

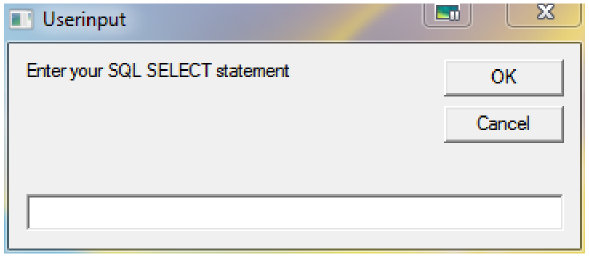
Step 3: Open the .csv file in Excel



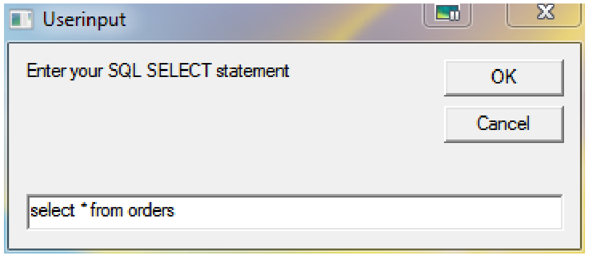
	A	B	C
1	A123	Gizmo	2746.2
2	B004	Doohickey	9629.44
3	C954	Thingamabob	288.87
4	D784	Whatchamacallit	7774.9
5	E888	Widget	7789.82

One Click to Pull IBM i Data into Excel





Enter your SQL SELECT statement

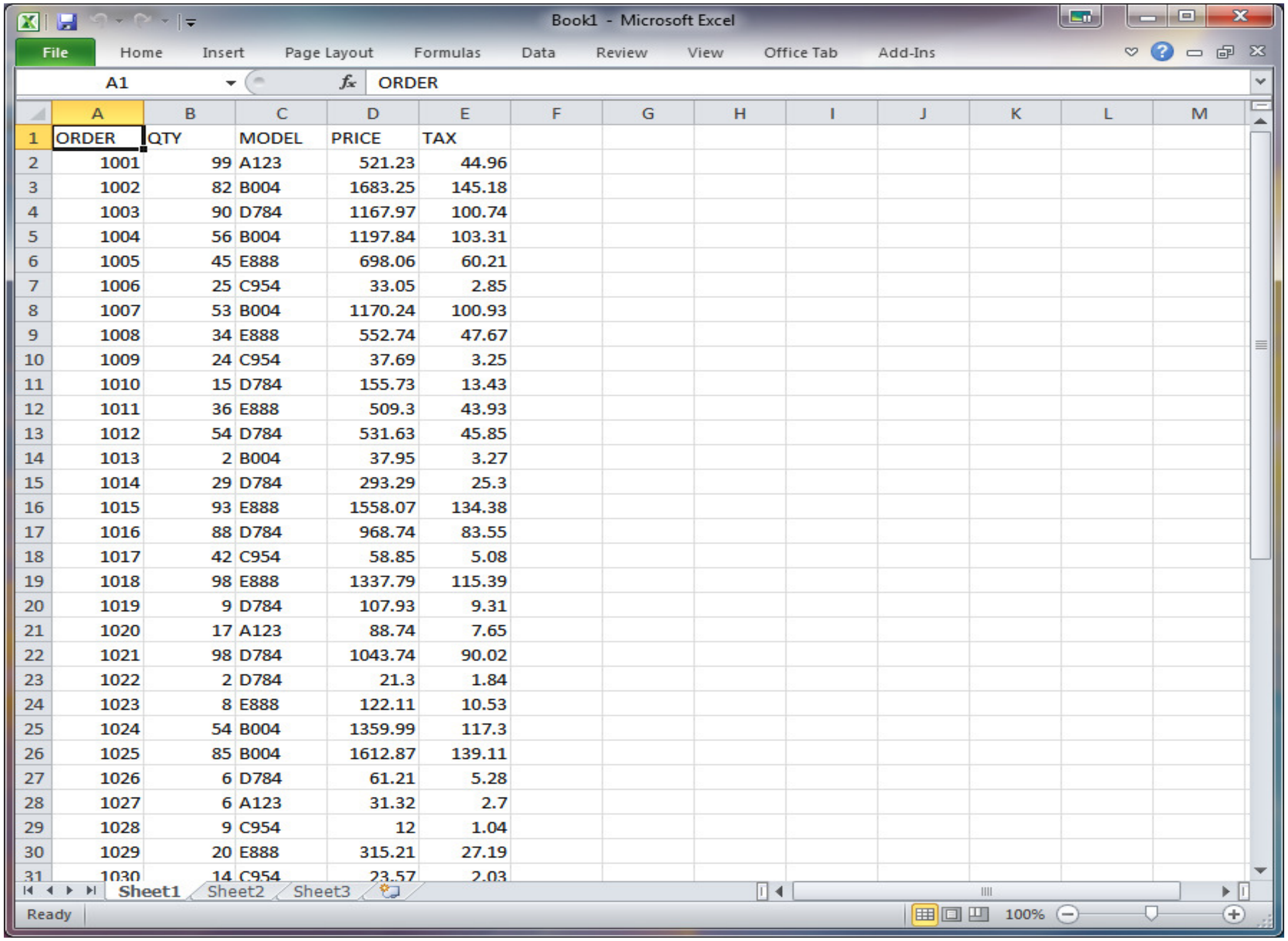


Enter your SQL SELECT statement

select * from orders

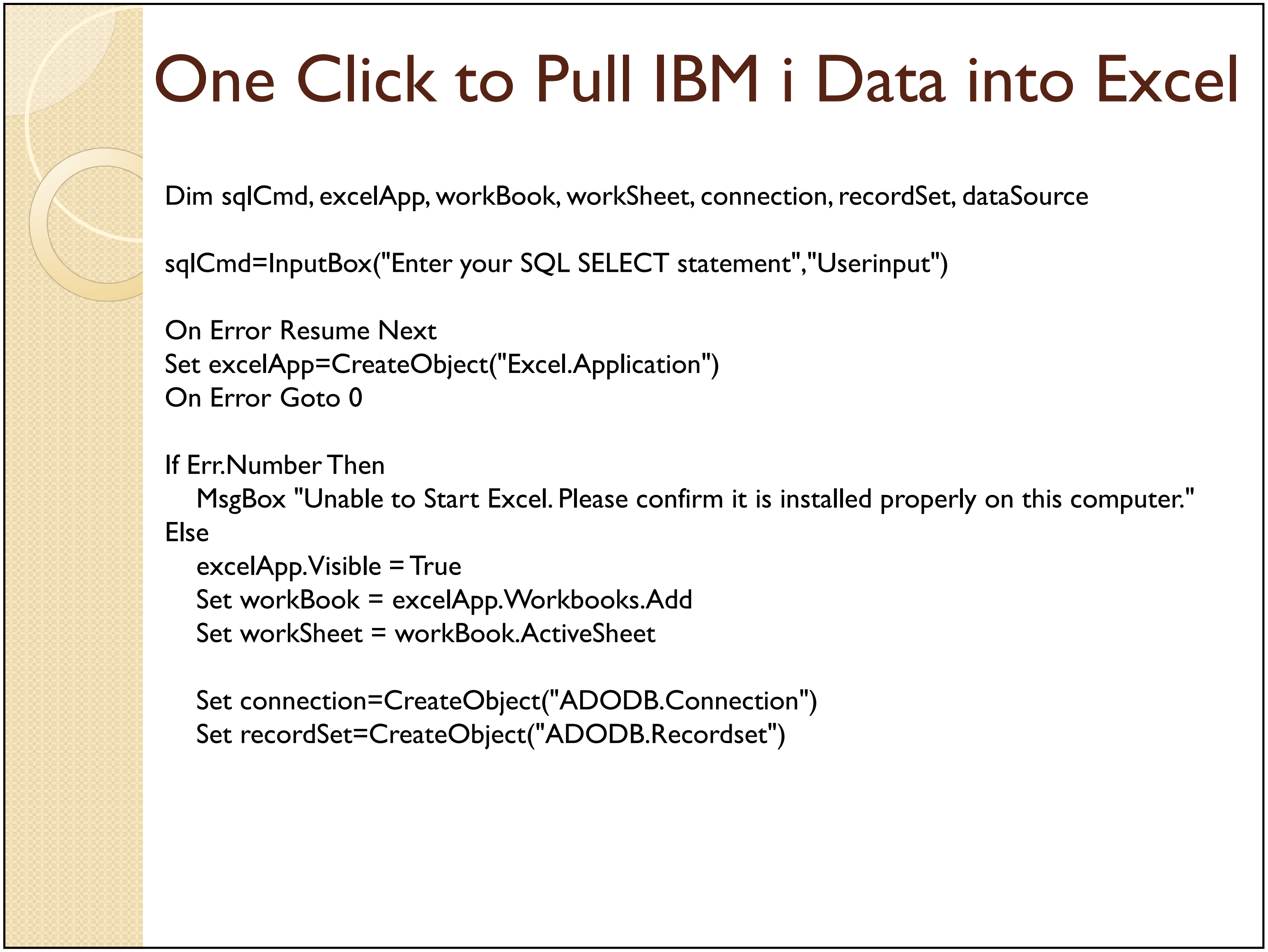
And easy as magic…

One Click to Pull IBM i Data into Excel



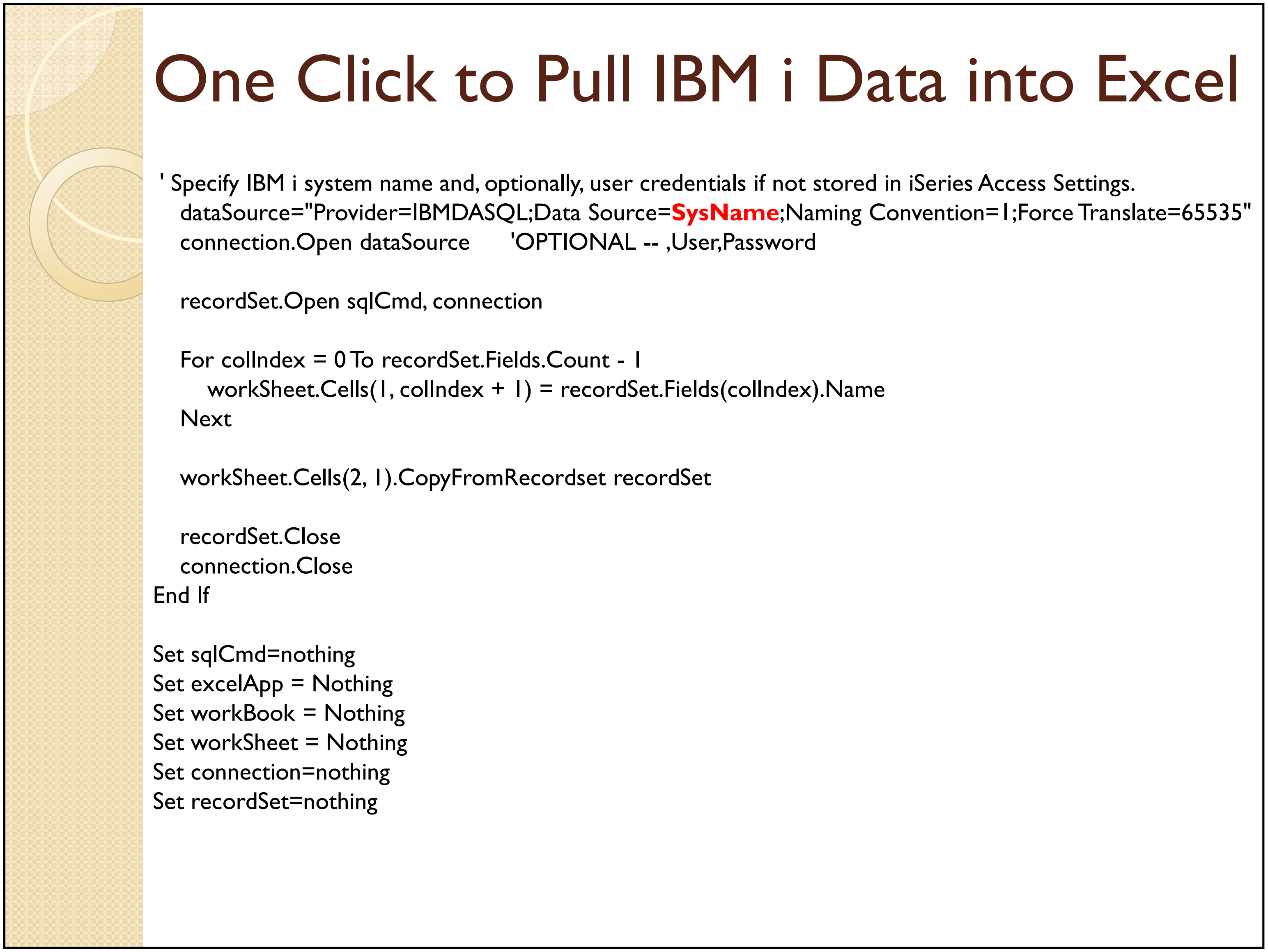
A1: ORDER

	A	B	C	D	E
1	ORDER	QTY	MODEL	PRICE	TAX
2	1001	99	A123	521.23	44.96
3	1002	82	B004	1683.25	145.18
4	1003	90	D784	1167.97	100.74
5	1004	56	B004	1197.84	103.31
6	1005	45	E888	698.06	60.21
7	1006	25	C954	33.05	2.85
8	1007	53	B004	1170.24	100.93
9	1008	34	E888	552.74	47.67
10	1009	24	C954	37.69	3.25
11	1010	15	D784	155.73	13.43
12	1011	36	E888	509.3	43.93
13	1012	54	D784	531.63	45.85
14	1013	2	B004	37.95	3.27
15	1014	29	D784	293.29	25.3
16	1015	93	E888	1558.07	134.38
17	1016	88	D784	968.74	83.55
18	1017	42	C954	58.85	5.08
19	1018	98	E888	1337.79	115.39
20	1019	9	D784	107.93	9.31
21	1020	17	A123	88.74	7.65
22	1021	98	D784	1043.74	90.02
23	1022	2	D784	21.3	1.84
24	1023	8	E888	122.11	10.53
25	1024	54	B004	1359.99	117.3
26	1025	85	B004	1612.87	139.11
27	1026	6	D784	61.21	5.28
28	1027	6	A123	31.32	2.7
29	1028	9	C954	12	1.04
30	1029	20	E888	315.21	27.19
31	1030	14	C954	23.57	2.03



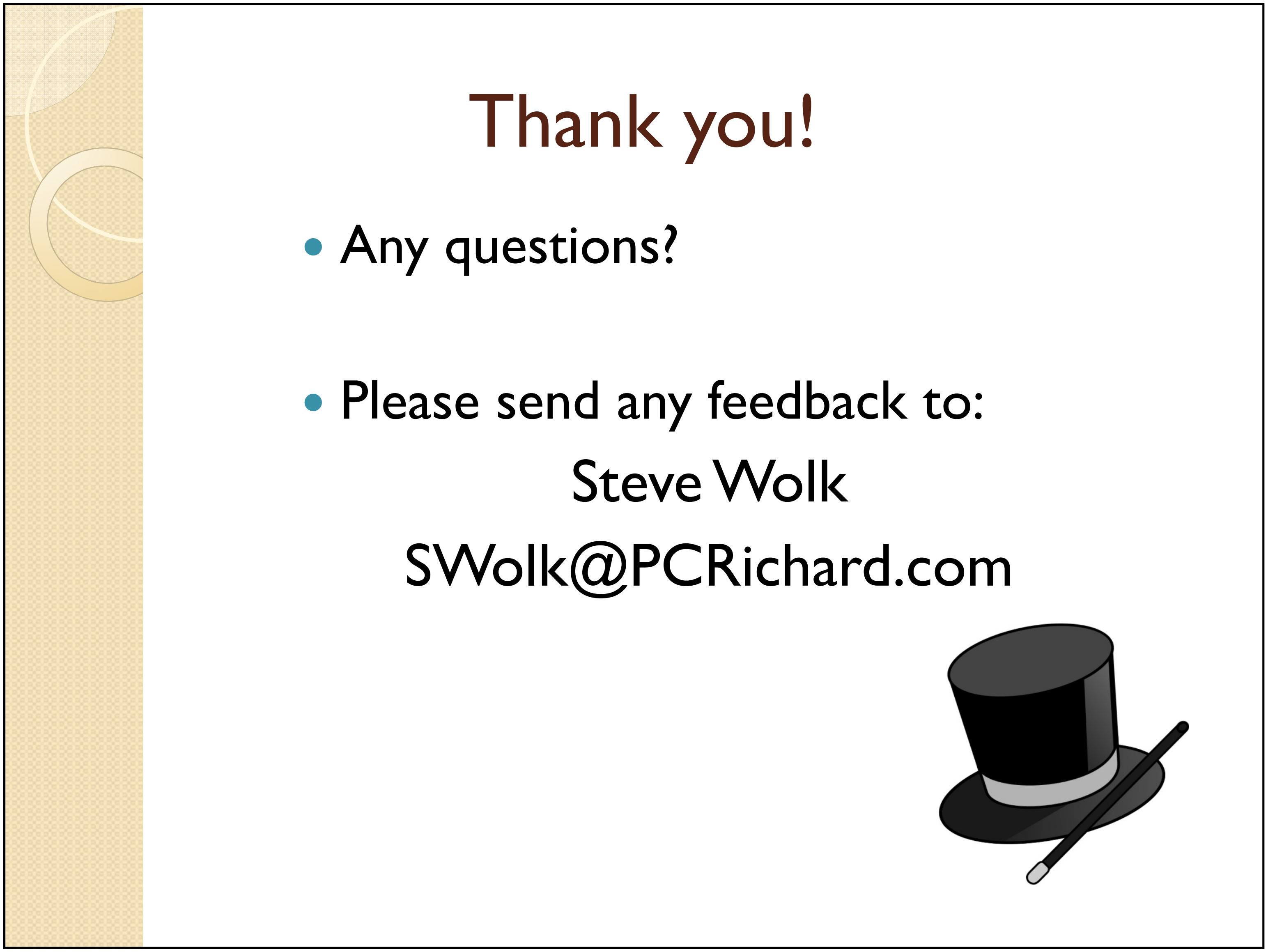
One Click to Pull IBM i Data into Excel

```
Dim sqlCmd, excelApp, workBook, workSheet, connection, recordSet, dataSource

sqlCmd=InputBox("Enter your SQL SELECT statement","Userinput")

On Error Resume Next
Set excelApp=CreateObject("Excel.Application")
On Error Goto 0

If Err.Number Then
    MsgBox "Unable to Start Excel. Please confirm it is installed properly on this computer."
Else
    excelApp.Visible = True
    Set workBook = excelApp.Workbooks.Add
    Set workSheet = workBook.ActiveSheet

    Set connection=CreateObject("ADODB.Connection")
    Set recordSet=CreateObject("ADODB.Recordset")
```



One Click to Pull IBM i Data into Excel

```
 ' Specify IBM i system name and, optionally, user credentials if not stored in iSeries Access Settings.
   dataSource="Provider=IBMDASQL;Data Source=SysName;Naming Convention=1;Force Translate=65535"
   connection.Open dataSource      'OPTIONAL -- ,User,Password

   recordSet.Open sqlCmd, connection

   For colIndex = 0 To recordSet.Fields.Count - 1
      workSheet.Cells(1, colIndex + 1) = recordSet.Fields(colIndex).Name
   Next

   workSheet.Cells(2, 1).CopyFromRecordset recordSet

   recordSet.Close
   connection.Close
End If

Set sqlCmd=nothing
Set excelApp = Nothing
Set workBook = Nothing
Set workSheet = Nothing
Set connection=nothing
Set recordSet=nothing
```



Thank you!

- Any questions?

- Please send any feedback to:

Steve Wolk

SWolk@PCRichard.com