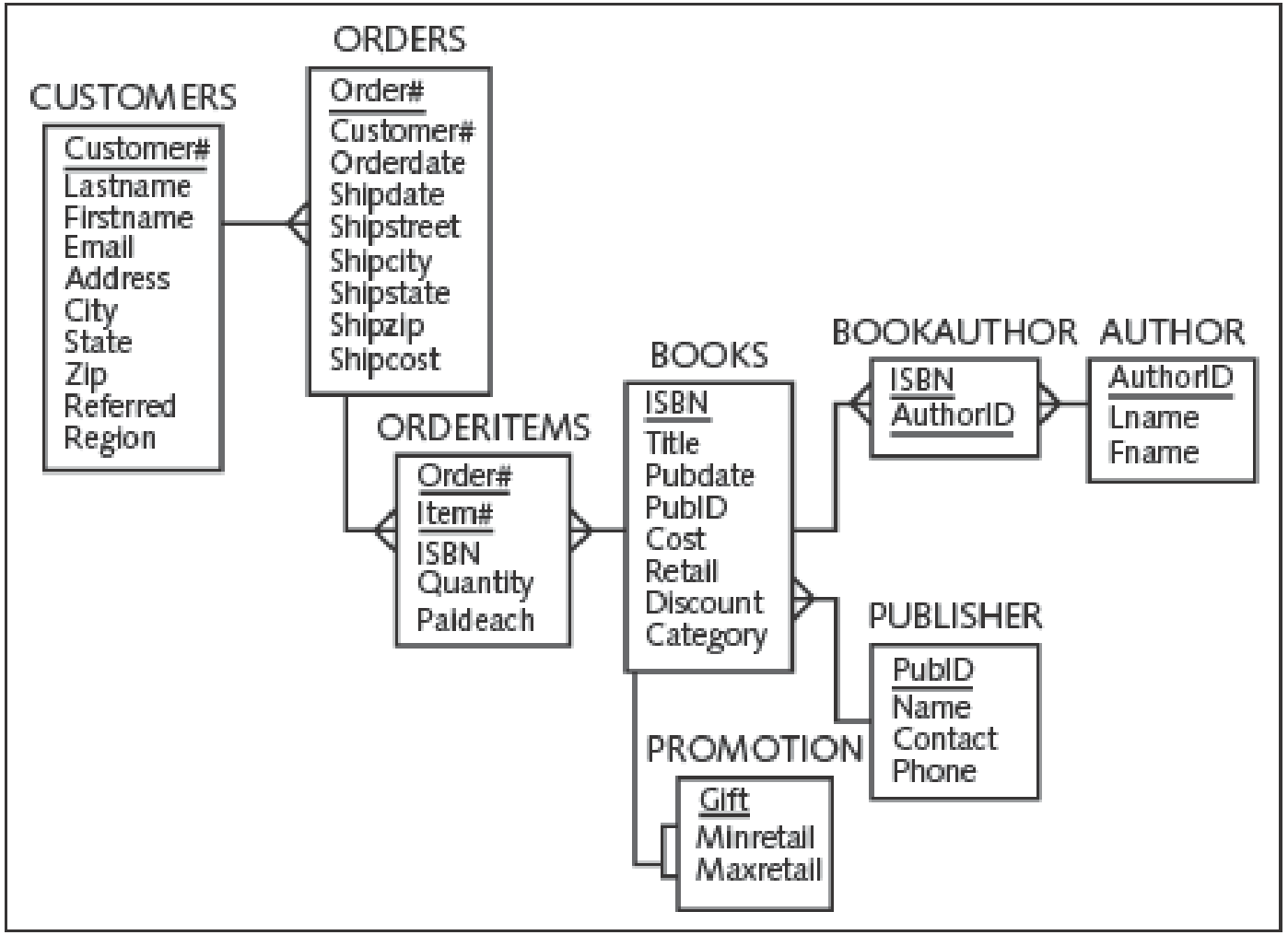
Database Systems
Other Stuff

CS 630 Database Systems

Professor Nardi

KING GRADUATE SCHOOL MONROE COLLEGE

Normalized JustLee Books Database…



Terminology

- **Function :** Predefined Block Of Code That Accepts Arguments…
- **Single-Row Function :** Returns One Row of Results For Each Record Processed…
- **Multiple-Row Function :** Returns One Result Per Group Of Data Processed…

Summary of Types of Functions…

Type of Function	Functions
Case Conversion Functions	UPPER, LOWER, INITCAP
Character Manipulation Functions	SUBSTR, INSTR, LENGTH, LPAD/RPAD, LTRIM/RTRIM, REPLACE, TRANSLATE, CONCAT
Numeric Functions	ROUND, TRUNC, MOD, ABS, POWER
Date Functions	MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY, TO_DATE, ROUND, TRUNC, CURRENT_DATE
Regular Expressions	REGEXP_LIKE, REGEXP_SUBSTR
Other Functions	NVL, NVL2, NULLIF, TO_CHAR, DECODE, CASE expression, SOUNDEX, TO_NUMBER

Case Conversion Functions…

- Alters the Case of Data Stored in a Column or Character String…
- **LOWER :** Used To Convert Characters To Lowercase Letters…
- **UPPER :** Used To Convert Characters To Uppercase Letters…
- Use in a *SELECT* Clause to Alter the Display of Characters in the Results Set…
- Use in a *WHERE* Clause to Modify the Case of Characters in a Search Condition…
- Syntax For UPPER (and LOWER) is UPPER(c) Where “c” is the String or Field to Be Converted Into Uppercase (or Lowercase) Characters…

Case Conversion Examples…

Will Convert Value of *lastname* to Lower Case and Then Perform the Where Clause…

This Example Can Return a Last Name of NELSON, nelson, Nelson, etc….

Enter SQL Statement:

```
SELECT firstname, lastname
 FROM customers
 WHERE LOWER(lastname) = 'nelson';
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	FIRSTNAME	LASTNAME
1	BECCA	NELSON

Case Conversion Examples…

Will Convert the Return Values of *lastname* and *firstname* to Lower Case When They Are Selected From the Query…

So For This Example, *lastname* Will Always Return nelson Even If It is Stored as NELSON, nelson, Nelson, etc….

Enter SQL Statement:

```
SELECT LOWER(firstname), LOWER(lastname)
 FROM customers
 WHERE LOWER(lastname) = 'nelson';
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	LOWER(FIRSTNAME)	LOWER(LASTNAME)
1	becca	nelson

INITCAP Function…

- Used to Convert Characters to Mixed Case…That Is, the First Letter of Each World is Capitalized and ALL Other Letters Are Lower Case…
- So For Example:

```
SELECT INITCAP( firstname || ' ' || lastname )
FROM contacts;
```

- This Would Return ‘John Smith’ Even If the Data Was Stored as ‘john’ and ‘smith’…

Character Manipulation – SUBSTR, INSTR…

- Manipulate Data By Extracting Substrings, Counting The Number Of Characters, Replacing Strings, Etc.
- **SUBSTR :** Used To Return a Substring or Portion of a String…You Can Start Your Search for the Substring From Either the Start of the String or the End of the String…
- **INSTR :** Searches For a Substring in a String and Returns the Position of the Substring in a String…

Character Manipulation SUBSTR, INSTR Examples…

Enter SQL Statement:

```
SELECT DISTINCT zip, SUBSTR(zip, 1, 3), SUBSTR(zip, -3, 2)
 FROM customers
 WHERE SUBSTR(zip, -3, 2) < 30;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	ZIP	SUBSTR(ZIP,1,3)	SUBSTR(ZIP,-3,2)
1	98115	981	11
2	12211	122	21
3	82003	820	00
4	02110	021	11
5	49006	490	00
6	31206	312	20
7	33111	331	11

SUBSTR (zip, 1, 3) Means Take the Field ‘zip’, Start At Position 1 and Take 3 Characters…

SUBSTR (zip, -3, 2) Means Take the Field ‘zip’, Start At the END of the String, Go Backwards 3 Positions, and Take the Next 2 Characters…

Enter SQL Statement:

```
SELECT name, INSTR(name, ',') "First comma",
             INSTR(name, ',', 10) "Start read position 10",
             INSTR(name, ',', 1, 2) "Second comma"
 FROM contacts;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	NAME	First comma	Start read position 10	Second comma
1	LaFodant,Mike,934-555-3493	9	14	14
2	Harris,Annette,727-555-2739	7	15	15
3	Crew,Ben,352-555-3638	5	0	9

Note the Comment After Each INSTR in the Example…

KING GRADUATE SCHOOL MONROE COLLEGE

LENGTH and CONCAT Functions…

- **LENGTH :** Used to Determine the Number of Characters in a String…

Enter SQL Statement:

```
SELECT DISTINCT LENGTH(address)
 FROM customers
 ORDER BY LENGTH(address) DESC;
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	LENGTH(ADDRESS)
1	20
2	18
3	17
4	16
5	13
6	12
7	11

- **CONCAT :** Used to Concatenate or Combine Two Character Strings…

Enter SQL Statement:

```
SELECT firstname, lastname,
        CONCAT('Customer number: ', customer#) "Number"
 FROM customers
 WHERE state = 'FL';
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	FIRSTNAME	LASTNAME	Number
1	BONITA	MORALES	Customer number: 1001
2	LEILA	SMITH	Customer number: 1003
3	NICHOLAS	NGUYEN	Customer number: 1013
4	STEVE	SCHELL	Customer number: 1015

LPAD, RPAD, LTRIM, RTRIM…

- **LPAD, RPAD :** Used to Pad or Fill In a Character String to a Fixed Width…

Enter SQL Statement:

```
SELECT firstname, LPAD(firstname, 12, ' '), LPAD(firstname, 12, '*')
 FROM customers
 WHERE firstname LIKE 'J%';
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	FIRSTNAME	LPAD(FIRSTNAME,12,'')	LPAD(FIRSTNAME,12,'*')
1	JORGE	JORGE	*******JORGE
2	JAKE	JAKE	********JAKE
3	JASMINE	JASMINE	*****JASMINE
4	JENNIFER	JENNIFER	****JENNIFER

- **LTRIM, RTRIM :** Used to Remove a Specific String of Characters…

Enter SQL Statement:

```
SELECT lastname, address, LTRIM(address, 'P.O. BOX')
 FROM customers
 WHERE state = 'FL';
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	LASTNAME	ADDRESS	LTRIM(ADDRESS,'P.O.BOX')
1	MORALES	P.O. BOX 651	651
2	SMITH	P.O. BOX 66	66
3	NGUYEN	357 WHITE EAGLE AVE.	357 WHITE EAGLE AVE.
4	SCHELL	P.O. BOX 677	677

REPLACE and TRANSLATE Functions…

- **REPLACE :** Substitutes One String With Another String…

Enter SQL Statement:

```
SELECT address, REPLACE(address, 'P.O.', 'POST OFFICE')
 FROM customers
 WHERE state = 'FL';
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA

Results:

	ADDRESS	REPLACE(ADDRESS,'P.O.','POSTOFFICE')
1	P.O. BOX 651	POST OFFICE BOX 651
2	P.O. BOX 66	POST OFFICE BOX 66
3	357 WHITE EAGLE AVE.	357 WHITE EAGLE AVE.
4	P.O. BOX 677	POST OFFICE BOX 677

- **TRANSLATE :** Makes Single-Character, One to One Translations or Substitutions… Format Has the String to Be Translated, the Original Character, and the Character You Want to Replace It With…

Enter SQL Statement:

```
SELECT name, TRANSLATE(name, ',', '-'), TRANSLATE(name, ',A', '-a')
 FROM contacts;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	NAME	TRANSLATE(NAME,',','-')	TRANSLATE(NAME,',A','-A')
1	LaFodant,Mike,934-555-3493	LaFodant-Mike-934-555-3493	LaFodant-Mike-934-555-3493
2	Harris,Annette,727-555-2739	Harris-Annette-727-555-2739	Harris-annette-727-555-2739
3	Crew,Ben,352-555-3638	Crew-Ben-352-555-3638	Crew-Ben-352-555-3638

Number Functions…

- Allows Manipulation of Numeric Data…
- **ROUND :** Used to Round Numeric Columns to a Stated Precision…
- **TRUNC :** Used to Truncate a Numeric Value to a Specific Function…
- **MOD :** Gives the Remainder of a Division…MOD (7/2) is 1…
- **ABS :** Returns the Absolute Value of a Number…ABS (2.25) Returns 2.25…ABS (-2.25) Returns 2.25…

Number Function Examples…

Enter SQL Statement:

```
SELECT title, retail, ROUND(retail,1), ROUND(retail,0), ROUND(retail,-1)
 FROM books;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	TITLE	RETAIL	ROUND(RETAIL,1)	ROUND(RETAIL,0)	ROUND(RETAIL,-1)
1	BODYBUILD IN 10 MINUTES A DAY	30.95	31	31	30
2	REVENGE OF MICKEY	22	22	22	20
3	BUILDING A CAR WITH TOOTHPICKS	59.95	60	60	60
4	DATABASE IMPLEMENTATION	55.95	56	56	60
5	COOKING WITH MUSHROOMS	19.95			

Enter SQL Statement:

```
SELECT title, retail, TRUNC(retail,1), TRUNC(retail,0), TRUNC(retail,-1)
 FROM books;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	TITLE	RETAIL	TRUNC(RETAIL,1)	TRUNC(RETAIL,0)	TRUNC(RETAIL,-1)
1	BODYBUILD IN 10 MINUTES A DAY	30.95	30.9	30	30
2	REVENGE OF MICKEY	22	22	22	20
3	BUILDING A CAR WITH TOOTHPICKS	59.95	59.9	59	50
4	DATABASE IMPLEMENTATION	55.95	55.9	55	50
5	COOKING WITH MUSHROOMS	19.95	19.9	19	10

Number Function Examples…

Enter SQL Statement:

```
SELECT pubdate, SYSDATE, ROUND(pubdate-SYSDATE) "Days",
        ABS(ROUND(pubdate-SYSDATE)) "ABS Days"
 FROM books
 WHERE category = 'CHILDREN';
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	PUBDATE	SYSDATE	Days	ABS Days
1	18-MAR-06	24-JAN-09	-1044	1044
2	08-NOV-05	24-JAN-09	-1174	1174

Enter SQL Statement:

```
SELECT 235/16
 FROM DUAL;

SELECT TRUNC(235/16,0) LBS, MOD(235,16) OZ
 FROM DUAL;
```

Results | Script Output | Explain | Autotrace | DBMS Output

```
235/16
----------------------
14.6875

1 rows selected

LBS                    OZ
---------------------- ----------------------
14                     11

1 rows selected
```

KING GRADUATE SCHOOL MONROE COLLEGE

Date Functions…

- Used to Perform Data Calculations or Format Date Values…
- For Example, Subtract Two Dates to Determine the Number of Days Between the Dates…

Enter SQL Statement:

```
SELECT order#, shipdate, orderdate, (shipdate-orderdate) Delay
 FROM orders
 WHERE order# = 1004;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	ORDER#	SHIPDATE	ORDERDATE	DELAY
1	1004	05-APR-09	01-APR-09	4

MONTHS_BETWEEN Function…

- Determines the Number of Months Between Two Dates…
- Don’t Worry About the JOIN…We Will Get to It…

Enter SQL Statement:

```
SELECT title, MONTHS_BETWEEN(orderdate,pubdate) MTHS
 FROM books JOIN orderitems USING (isbn)
       JOIN orders USING (order#)
 WHERE order# = 1004;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	TITLE	MTHS
1	PAINLESS CHILD-REARING	56.4838709677419354838709677419354838709 7

ADD_MONTHS Function…

- Adds a Specified Number of Months to a Date…

Enter SQL Statement:

```
SELECT title, pubdate, ADD_MONTHS('01-DEC-08',18) "Renegotiate Date",
        ADD_MONTHS(pubdate,84) "Drop Date"
 FROM books
 WHERE category = 'COMPUTER'
 ORDER BY "Renegotiate Date";
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	TITLE	PUBDATE	Renegotiate Date	Drop Date
1	DATABASE IMPLEMENTATION	04-JUN-03	01-JUN-10	04-JUN-10
2	HOLY GRAIL OF ORACLE	31-DEC-05	01-JUN-10	31-DEC-12
3	HANDCRANKED COMPUTERS	21-JAN-05	01-JUN-10	21-JAN-12
4	E-BUSINESS THE EASY WAY	01-MAR-06	01-JUN-10	01-MAR-13

NEXT_DAY Function…

- Determines the Next Occurrence of a Specified Day of the Week After a Given Date…
- So Looking at the Example…the SQL Will Display the Next MONDAY After the orderdate…

Enter SQL Statement:

```
SELECT order#, orderdate, NEXT_DAY(orderdate, 'MONDAY')
 FROM orders
 WHERE order# = 1018;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	ORDER#	ORDERDATE	NEXT_DAY(ORDERDATE,'MONDAY')
1	1018	05-APR-09	06-APR-09

TO_DATE Function…

- Converts Various Date Formats to the Internal Format (DD-MON-YY) Used By Oracle…

Enter SQL Statement:

```
SELECT order#, orderdate, shipdate
 FROM orders
 WHERE orderdate = TO_DATE('March 31, 2009','Month DD, YYYY');
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	ORDER#	ORDERDATE	SHIPDATE
1	1000	31-MAR-09	02-APR-09
2	1001	31-MAR-09	01-APR-09
3	1002	31-MAR-09	01-APR-09

Format Model Elements – Dates…

Element	Description	Example
MONTH	Name of the Month Spelled Out and Padded With Blank Spaces to a Total Width of 9 Characters	APRIL
MON	Three-Letter Abbreviation for the Name of the Month	APR
MM	Two-Digit Numeric Value for the Month	04
RM	Roman Numeral Representing the Month	IV
D	Numeric Value for the Day of the Week	Wednesday = 4
DD	Numeric Value for the Day of the Month	28
DDD	Numeric Value for the Day of the Year	December 31 = 365
DAY	Name of the Day of the Week Padded With Blank Spaces to a Length of 9 Characters	WEDNESDAY
DY	Three-Letter Abbreviation for the Day of the Week	WED
YYYY	Displays the Four-Digit Numeric Value of the Year	2009
YYY or YY or Y	The Last Three, Two, or Single Digits of the Year	2009 = 009; 2009 = 09; 2009 = 9
YEAR	Spelled Out Version of the Year	TWO THOUSAND NINE
B.C. OR A.D.	Value Indicating B.C. or A.D.	2009 A.D.

ROUND and TRUNC Functions With Dates…

- **ROUND :** Returns a Date Rounded to a Specific Unit (i.e., Year, Month, Day)…

Enter SQL Statement:

```
SELECT pubdate, ROUND(pubdate,'MONTH'), ROUND(pubdate,'YEAR')
 FROM books;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Outp

Results:

	PUBDATE	ROUND(PUBDATE,'MONTH')	ROUND(PUBDATE,'YEAR')
1	21-JAN-05	01-FEB-05	01-JAN-05
2	14-DEC-05	01-DEC-05	01-JAN-06
3	18-MAR-06	01-APR-06	01-JAN-06
4	04-JUN-03	01-JUN-03	01-JAN-03
5	28-FEB-04	01-MAR-04	01-JAN-04

- **TRUNC :** Returns a Date Truncated to a Specific Unit (i.e., Year, Month, Day)…

Enter SQL Statement:

```
SELECT title, TRUNC(MONTHS_BETWEEN(orderdate,pubdate),0) MTHS
 FROM books JOIN orderitems USING (isbn)
       JOIN orders USING (order#)
 WHERE order# = 1004;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	TITLE	MTHS
1	PAINLESS CHILD-REARING	56

Regular Expressions…

- Allows the Description of Complex Patterns in Textual Data…

REGEXP_LIKE Function…

- Advanced Version of the LIKE Function…Returns Rows That Match a Regular Expression Pattern…
- Syntax is : REGEXP_LIKE (Source_String, Search_Pattern [, Match_Parameter]);
  - Source_String : String to Be Searched…
  - Search_Pattern : Literal String That Represents the “Regular Expression” Pattern to Be Matched…
  - Match_Parameter : Literal String That Changes the Default Matching Behavior Of The REGEXP_LIKE() Function.
- Returns Rows That Match the Regular Expression Pattern….

```
SELECT      first_name
FROM        employees
WHERE       REGEXP_LIKE( first_name, 'c' )
ORDER BY    first_name;
```

	FIRST_NAME
1	Alice
2	Florence
3	Frederick
4	Grace
5	Gracie
6	Jackson
7	Jessica
8	Lucy
9	Scarlett

Other Functions…

- **NVL :** Substitutes a Value for a NULL Value…
- **NVL2 :** Allows a Different Action Based on Whether a Value is NULL…
- **NULLIF :** Accepts Two Arguments…Returns a NULL Value If the Two Arguments Are Equal…If the Arguments Are Not Equal, the Function Returns the First Argument…
- **TO_CHAR :** Converts Dates and Numbers to a Formatted Character String…
- **DECODE :** Determines Action Based Upon Values in a List…
- **CASE :** Adds "If-Else" Logic to SQL Without Calling a Procedure…Evaluates a List of Conditions and Returns One of the Possible Results…
- **SOUNDEX :** References Phonetic Representation of Words…
- **TO_NUMBER :** Converts a String to a Number…

Other Functions Examples…

Enter SQL Statement:

```
SELECT order#, orderdate, shipdate,
       NVL(shipdate,'06-APR-09')-orderdate "Ship Days"
 FROM orders
 WHERE orderdate >= '03-APR-09';
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	ORDER#	ORDERDATE	SHIPDATE	Ship Days
1	1009	03-APR-09	05-APR-09	2
2	1010	03-APR-09	04-APR-09	1
3	1011	03-APR-09	05-APR-09	2
4	1012	03-APR-09	(null)	3
5	1013	03-APR-09	04-APR-09	1
6	1014	04-APR-09	05-APR-09	1
7	1015	04-APR-09	(null)	2
8	1016	04-APR-09	(null)	2
9	1017	04-APR-09	05-APR-09	1
10	1018	05-APR-09	(null)	1
11	1019	05-APR-09	(null)	1
12	1020	05-APR-09	(null)	1

Enter SQL Statement:

```
SELECT order#, orderdate,
       NVL2(shipdate, 'Shipped', 'Not Shipped') "Status"
 FROM orders
 WHERE orderdate >= '03-APR-09';
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	ORDER#	ORDERDATE	Status
1	1009	03-APR-09	Shipped
2	1010	03-APR-09	Shipped
3	1011	03-APR-09	Shipped
4	1012	03-APR-09	Not Shipped
5	1013	03-APR-09	Shipped
6	1014	04-APR-09	Shipped
7	1015	04-APR-09	Not Shipped
8	1016	04-APR-09	Not Shipped
9	1017	04-APR-09	Shipped
10	1018	05-APR-09	Not Shipped
11	1019	05-APR-09	Not Shipped
12	1020	05-APR-09	Not Shipped

Other Functions Examples…

Enter SQL Statement:

```
SELECT o.customer#, order#, isbn, oi.paideach, b.retail,
       NULLIF(oi.paideach, b.retail)
 FROM orders o JOIN orderitems oi
          USING (order#)
       JOIN books b USING (isbn)
 WHERE order# IN(1001, 1007)
 ORDER BY order#;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	CUSTOMER#	ORDER#	ISBN	PAIDEACH	RETAIL	NULLIF(OI.PAIDEACH,B.RETAIL)
1	1010	1001	9247381001	31.95	31.95	(null)
2	1010	1001	2491748320	85.45	89.95	85.45
3	1007	1007	3957136468	72.15	75.95	72.15
4	1007	1007	9959789321	54.5	54.5	(null)
5	1007	1007	8117949391	8.95	8.95	(null)
6	1007	1007	8843172113	55.95	55.95	(null)

Enter SQL Statement:

```
SELECT title,
     TO_CHAR(pubdate, 'MONTH DD, YYYY') "Publication Date",
     TO_CHAR(retail, '$999.99') "Retail Price"
 FROM books
 WHERE isbn = 0401140733;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	TITLE	Publication Date	Retail Price
1	REVENGE OF MICKEY	DECEMBER 14, 2005	$22.00

Other Functions Examples…

Enter SQL Statement:

```
SELECT customer#, state,
        DECODE (state, 'CA', .08,
                       'FL', .07,
                             0) "Sales Tax Rate"
 FROM customers
 WHERE state IN('CA','FL','GA','TX');
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	CUSTOMER#	STATE	Sales Tax Rate
1	1001	FL	0.07
2	1002	CA	0.08
3	1003	FL	0.07

Enter SQL Statement:

```
SELECT empno, lname, fname,
    ROUND(MONTHS_BETWEEN('01-JUL-09', hiredate)/12,2) "Years",
    CASE
      WHEN (MONTHS_BETWEEN('01-JUL-09', hiredate)/12) < 4 THEN 'Level 1'
      WHEN (MONTHS_BETWEEN('01-JUL-09', hiredate)/12) < 8 THEN 'Level 2'
      WHEN (MONTHS_BETWEEN('01-JUL-09', hiredate)/12) < 11 THEN 'Level 3'
      WHEN (MONTHS_BETWEEN('01-JUL-09', hiredate)/12) < 15 THEN 'Level 4'
      ELSE 'Level 5'
    END "Retire Level"
  FROM employees;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	EMPNO	LNAME	FNAME	Years	Retire Level
1	7839	KING	BEN	17.62	Level 5
2	8888	JONES	LARRY	10.62	Level 3
3	7344	SMITH	SAM	13.62	Level 4
4	7355	POTTS	JIM	13.62	Level 4
5	8844	STUART	SUE	10.62	Level 3

Other Functions Examples…

Enter SQL Statement:

```
SELECT customer#, lastname, firstname
 FROM customers
 WHERE SOUNDEX(lastname) = SOUNDEX('smyth');
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	CUSTOMER#	LASTNAME	FIRSTNAME
1	1003	SMITH	LEILA
2	1019	SMITH	JENNIFER

Enter SQL Statement:

```
SELECT title, pubdate,
    TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) - TO_NUMBER(TO_CHAR(pubdate,'YYYY'))
"Yrs"
 FROM books
 WHERE category = 'COMPUTER';
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	TITLE	PUBDATE	Yrs
1	DATABASE IMPLEMENTATION	04-JUN-03	6
2	HOLY GRAIL OF ORACLE	31-DEC-05	4
3	HANDCRANKED COMPUTERS	21-JAN-05	4
4	E-BUSINESS THE EASY WAY	01-MAR-06	3

Format Model Elements – Time and Number…

	Time Elements	
SS	Seconds	Value Between 0 - 59
SSSS	Seconds Past Midnight	Value Between 0 - 86399
MI	Minutes	Value Between 0 - 59
HH or HH12	Hours	Value Between 1 - 12
HH24	Hours – 24 Hour Clock	Value Between 0 - 23
A.M. or P.M.	Value Indicating Morning or Evening Hours	A.M. – Before Noon or P.M. – After Noon

	Number Elements	
9	Indicates Width of Display With a Series of 9s, But Insignificant Leading Zeros Are NOT Displayed	99999
0	Displays Insignificant Leading Zeros	0009999
$	Displays a Floating Dollar Sign	$99999
.	Indicates Number of Decimals to Display	999.99
,	Displays a Comma in the Position Indicated	9,999

DUAL Table…

- Dummy Table…
- Consists of One Column and One Row…
- Can Be Used For Table Reference in the FROM Clause…

Enter SQL Statement:

```
SELECT ROUND(4769.43, -2), Length('Hello')
 FROM DUAL;
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	ROUND(4769.43,-2)	LENGTH('HELLO')
1	4800	5

Group Functions…

- Return One Result Per Group of Rows Processed…
- Also Called Multiple-Row and Aggregate Functions…
- All Group Functions Ignore NULL Values Except COUNT(*)…
- Use DISTINCT to Suppress Duplicate Values…

```
SELECT *|columnname, columnname…
FROM tablename
[WHERE condition]
[GROUP BY columnname, columnname…]
[HAVING group condition];
```

SUM Function…

- Calculates Total Amount Stored in a Numeric Column for a Group of Rows…

Enter SQL Statement:

```
SELECT SUM((paideach-cost)*quantity) "Total Profit"
 FROM orderitems JOIN books USING (isbn)
 WHERE order# = 1007;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	Total Profit
1	119.48

AVG Function…

- Calculates the Average of Numeric Values in a Specified Column…
- Can Be Used on Values With Character, Numeric, and Date Datatypes…

Enter SQL Statement:

```
SELECT AVG(retail-cost) "Average Profit"
 FROM books
 WHERE category = 'COMPUTER';
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	Average Profit
1	18.2625

KING GRADUATE SCHOOL MONROE COLLEGE

COUNT Function…

- COUNT Function for Non-NULL Values : Include Column Name in Argument to Count Number of Occurrences…
- COUNT Function for NULL Values : Include Asterisk in Argument to Count Number of Rows…
- Can Be Used on Values With Character, Numeric, and Date Datatypes…

Enter SQL Statement:

```
SELECT COUNT(DISTINCT category)
 FROM books;
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	COUNT(DISTINCTCATEGORY)
1	8

Enter SQL Statement:

```
SELECT COUNT(*)
 FROM orders
 WHERE shipdate IS NULL;
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	COUNT(*)
1	6

MAX and MIN Function…

- MAX Returns the Largest Value…
- MIN Returns the Smallest Value…
- Can Be Used on Values With Character, Numeric, and Date Datatypes…

Enter SQL Statement:

```
SELECT MAX(retail-cost) "Highest Profit"
 FROM books;
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	Highest Profit
1	41.95

Enter SQL Statement:

```
SELECT MIN(pubdate)
 FROM books;
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	MIN(PUBDATE)
1	09-MAY-03

GROUPING Data…

- GROUP BY Clause…
  - Used to Group Data…
  - Must Be Used For Any Individual Column in the SELECT Clause With a Group Function…
  - Cannot Reference Column Aliases…

Enter SQL Statement:

```
SELECT category, TO_CHAR(AVG(retail-cost), '999.99') "Profit"
 FROM books
 GROUP BY category;
```

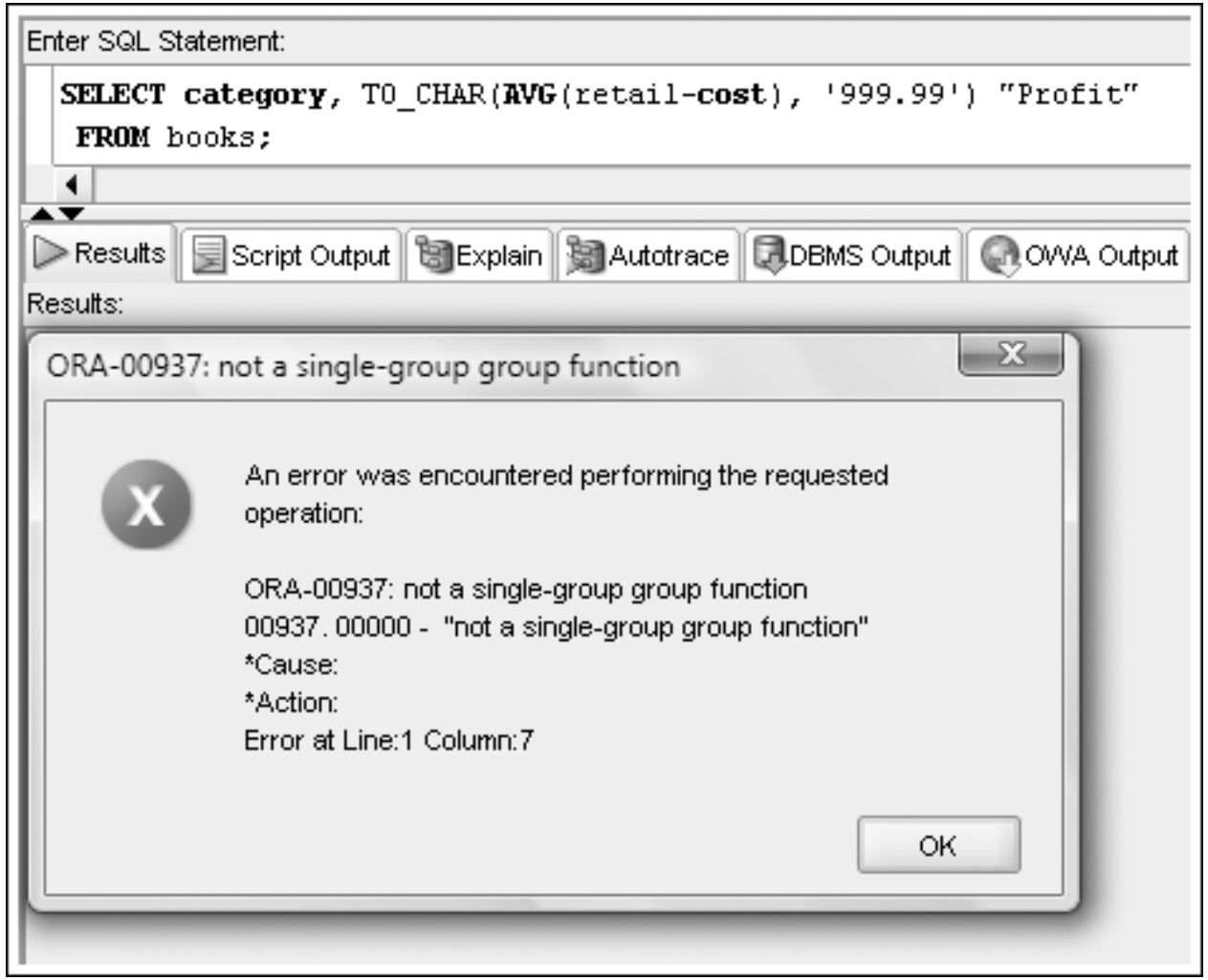
Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	CATEGORY	Profit
1	COMPUTER	18.26
2	COOKING	8.60
3	CHILDREN	12.89
4	LITERATURE	18.10
5	BUSINESS	16.55
6	FITNESS	12.20
7	FAMILY LIFE	24.88
8	SELF HELP	12.10

GROUP BY Common Error…

- A Common Error Is Missing a GROUP BY Clause For Non-Aggregated Columns in the SELECT Clause…



KING GRADUATE SCHOOL MONROE COLLEGE

Restricting Aggregated Output…

- HAVING clause serves as the WHERE clause for grouped data
- When included in the same SELECT statement, the clauses are evaluated in the order of:
  - WHERE
  - GROUP BY
  - HAVING

Enter SQL Statement:

```
SELECT category, TO_CHAR(AVG(retail-cost), '999.99') "Profit"
 FROM books
 GROUP BY category
 HAVING AVG(retail-cost) > 15;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	CATEGORY	Profit
1	COMPUTER	18.26
2	LITERATURE	18.10
3	BUSINESS	16.55
4	FAMILY LIFE	24.88

Restricting Aggregated Output…

Enter SQL Statement:

```
SELECT category, TO_CHAR(AVG(retail-cost), '999.99') "Profit"
 FROM books
 GROUP BY category
 HAVING AVG(retail-cost) > 15;
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	CATEGORY	Profit
1	COMPUTER	18.26
2	LITERATURE	18.10
3	BUSINESS	16.55
4	FAMILY LIFE	24.88

Enter SQL Statement:

```
SELECT category, TO_CHAR(AVG(retail-cost), '999.99') "Profit"
 FROM books
 WHERE pubdate > '01-JAN-05'
 GROUP BY category
 HAVING AVG(retail-cost) > 15;
```

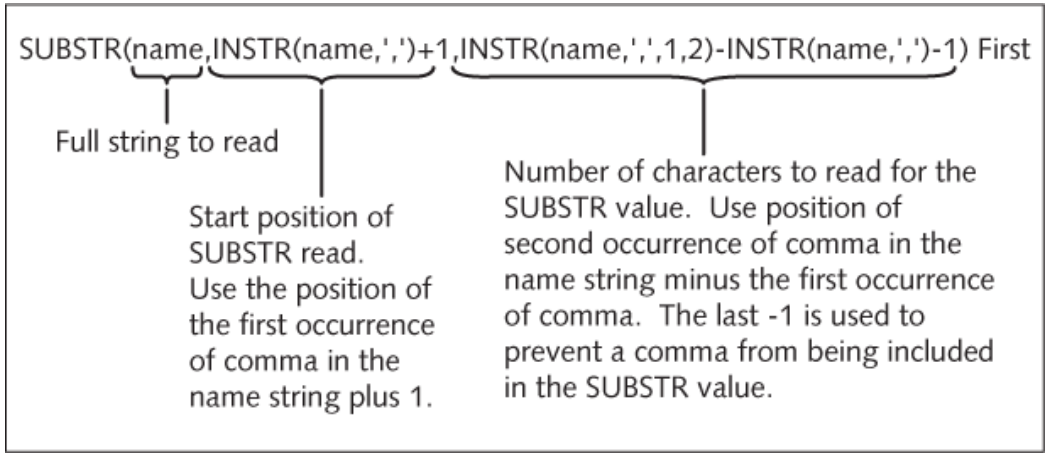
Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	CATEGORY	Profit
1	COMPUTER	16.17
2	LITERATURE	18.10

Nesting Functions…

- Functions Within a Function…
- INNER Functions Are Resolved First…
- Maximum Nesting Depth is 2…



Enter SQL Statement:

```
SELECT AVG(SUM(quantity*paideach)) "Average Order Total"
 FROM orders JOIN orderitems USING (order#)
 GROUP BY order#;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	Average Order Total
1	80.3833333333333333333333333333333333333

KING GRADUATE SCHOOL MONROE COLLEGE

Statistical Group Functions…

- Based on Normal Distribution…
- Includes…
  - **STDEV :** Returns the Statistical Standard Deviation of All Values in the Specified Expression….That Is, the Amount of Deviation Between All the Values of a Field or Expression…
  - **VARIANCE :** Returns the Differences in Values Between Rows…For Example, the Variance in Salary…

Statistical Group Functions Examples…

Enter SQL Statement:

```
SELECT category, COUNT(*), TO_CHAR(AVG(retail-cost),'999.99') "Avg",
       TO_CHAR(STDDEV(retail-cost),'999.9999') "Stddev"
 FROM books
 GROUP BY category;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	CATEGORY	COUNT(*)	Avg	Stddev
1	COMPUTER	4	18.26	11.2267
2	COOKING	2	8.60	1.6263
3	CHILDREN	2	12.89	13.0956
4	LITERATURE	1	18.10	.0000
5	BUSINESS	1	16.55	.0000
6	FITNESS	1	12.20	.0000
7	FAMILY LIFE	2	24.88	24.1477
8	SELF HELP	1	12.10	.0000

Enter SQL Statement:

```
SELECT category, TO_CHAR(VARIANCE(retail-cost),'999.99') "Var",
   MIN(retail-cost) "Min", MAX(retail-cost) "Max"
 FROM books
 GROUP BY category;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	CATEGORY	Var	Min	Max
1	COMPUTER	126.04	3.2	28.7
2	COOKING	2.65	7.45	9.75
3	CHILDREN	171.50	3.63	22.15
4	LITERATURE	.00	18.1	18.1
5	BUSINESS	.00	16.55	16.55
6	FITNESS	.00	12.2	12.2
7	FAMILY LIFE	583.11	7.8	41.95
8	SELF HELP	.00	12.1	12.1

Enhanced Aggregation for Reporting…

- Oracle Provides Extensions to the GROUP BY Clause, Which Allow Both Aggregation Across Multiple Dimensions or the Generation of Increasing Levels of Subtotals With a Single SELECT Statement…
- A Dimension is a Term Used to Describe Any Category Used in Analyzing Data, Such as Time, Geography, and Product Line…
- Each Dimension Could Contain Various Levels of Aggregation…
- For Example, Time Dimension May Include Aggregation By Month, Quarter, and Year…

Similar to an EXCEL Pivot Table…

Book1

	A	B	C
1	# Books Available by Publisher & Category		
2			
3	Sum Of Count		
4	Publisher	Category	Total
5	AMERICAN PUBLISHING	COMPUTER	3
6	AMERICAN PUBLISHING Total		3
7	PRINTING IS US	BUSINESS	1
8		FAMILY LIFE	1
9	PRINTING IS US Total		2
10	PUBLISH OUR WAY	CHILDREN	1
11		COMPUTER	1
12	PUBLISH OUR WAY Total		2
13	READING MATERIALS INC.	COOKING	2
14		FITNESS	1
15		SELF HELP	1
16	READING MATERIALS INC. Total		4
17	REED-N-RITE	CHILDREN	1
18		FAMILY LIFE	1
19		LITERATURE	1
20	REED-N-RITE Total		3
21	Grand Total		14

Two dimensions used on same row

Analyzing count of books

This pivot table requires totals by:
1. Publisher & Category
2. Publisher
3. Grand Total

One dimension used on row

One dimension used on column

Book1

	A	B	C	D	E	F	G	H	I	J
1	# Books Available by Publisher & Category									
2										
3	Sum Of Count	Category								
4	Publisher	BUSINESS	CHILDREN	COMPUTER	COOKING	FAMILY LIFE	FITNESS	LITERATURE	SELF HELP	Grand Total
5	AMERICAN PUBLISHING			3						3
6	PRINTING IS US	1				1				2
7	PUBLISH OUR WAY		1	1						2
8	READING MATERIALS INC.				2		1		1	4
9	REED-N-RITE		1			1		1		3
10	Grand Total	1	2	4	2	2	1	1	1	14
11										

This pivot table requires totals by:
1. Publisher & Category (intersection of column & row)
2. Publisher
3. Category
4. Grand Total

KING GRADUATE SCHOOL MONROE COLLEGE

Grouping Sets…

- For Another Time…

Enter SQL Statement:

```
SELECT name, category, COUNT(isbn),
  TO_CHAR(AVG(retail),'99.99') "Avg Retail"
 FROM publisher JOIN books USING (pubid)
 WHERE pubid IN (2,3,5)
 GROUP BY GROUPING SETS(name, category, (name, category),())
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	NAME	CATEGORY	COUNT(ISBN)	Avg Retail
1	REED-N-RITE	CHILDREN	1	8.95
2	PUBLISH OUR WAY	CHILDREN	1	59.95
3	PUBLISH OUR WAY	COMPUTER	1	54.50
4	AMERICAN PUBLISHING	COMPUTER	3	52.30
5	REED-N-RITE	LITERATURE	1	39.95
6	REED-N-RITE	FAMILY LIFE	1	89.95
7	(null)	CHILDREN	2	34.45
8	(null)	COMPUTER	4	52.85
9	(null)	LITERATURE	1	39.95
10	(null)	FAMILY LIFE	1	89.95
11	AMERICAN PUBLISHING	(null)	3	52.30
12	PUBLISH OUR WAY	(null)	2	57.23
13	REED-N-RITE	(null)	3	46.28
14	(null)	(null)	0	51.20

Name and Category

Category

Name

Overall total

Cube…

- For Another Time…

Enter SQL Statement:

```
SELECT name, category, COUNT(isbn),
  TO_CHAR(AVG(retail),'99.99') "Avg Retail"
 FROM publisher JOIN books USING (pubid)
 WHERE pubid IN (2,3,5)
 GROUP BY CUBE(name, category)
 ORDER BY name, category;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	NAME	CATEGORY	COUNT(ISBN)	Avg Retail
1	AMERICAN PUBLISHING	COMPUTER	3	52.30
2	AMERICAN PUBLISHING	(null)	3	52.30
3	PUBLISH OUR WAY	CHILDREN	1	59.95
4	PUBLISH OUR WAY	COMPUTER	1	54.50
5	PUBLISH OUR WAY	(null)	2	57.23
6	REED-N-RITE	CHILDREN	1	8.95
7	REED-N-RITE	FAMILY LIFE	1	89.95
8	REED-N-RITE	LITERATURE	1	39.95
9	REED-N-RITE	(null)	3	46.28
10	(null)	CHILDREN	2	34.45
11	(null)	COMPUTER	4	52.85
12	(null)	FAMILY LIFE	1	89.95
13	(null)	LITERATURE	1	39.95
14	(null)	(null)	8	51.28

ROLLUP…

- For Another Time…

Enter SQL Statement:

```
SELECT name, category, COUNT(isbn),
   TO_CHAR(AVG(retail),'99.99') "Avg Retail"
 FROM publisher JOIN books USING (pubid)
 WHERE pubid IN (2,3,5)
 GROUP BY ROLLUP(name, category)
 ORDER BY name, category;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	NAME	CATEGORY	COUNT(ISBN)	Avg Retail
1	AMERICAN PUBLISHING	COMPUTER	3	52.30
2	AMERICAN PUBLISHING	(null)	3	52.30
3	PUBLISH OUR WAY	CHILDREN	1	59.95
4	PUBLISH OUR WAY	COMPUTER	1	54.50
5	PUBLISH OUR WAY	(null)	2	57.23
6	REED-N-RITE	CHILDREN	1	8.95
7	REED-N-RITE	FAMILY LIFE	1	89.95
8	REED-N-RITE	LITERATURE	1	39.95
9	REED-N-RITE	(null)	3	46.28
10	(null)	(null)	8	51.28

Name subtotals

Grand total

Pattern Matching…

- For Another Time…

Worksheet | Query Builder

```
SELECT *
  FROM webhits  MATCH_RECOGNIZE (
   PARTITION BY wpage
   ORDER BY whdate
   MEASURES STRT.whdate AS BEGIN_wdate,
      LAST(DOWN.whdate) AS LOW_wdate,
      LAST(UP.whdate) AS END_wdate
   ONE ROW PER MATCH
   AFTER MATCH SKIP TO LAST UP
   PATTERN (STRT DOWN+ UP+)
   DEFINE
     DOWN AS DOWN.total < PREV(DOWN.total),
     UP AS UP.total > PREV(UP.total)
     ) WH
   ORDER BY WH.wpage, WH.BEGIN_wdate;
```

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0 seconds

	WPAGE	BEGIN_WDATE	LOW_WDATE	END_WDATE
1	PRD0201	11-NOV-14	21-NOV-14	01-DEC-14
2	PRD0201	01-DEC-14	11-DEC-14	21-DEC-14
3	PRD0485	08-NOV-14	21-NOV-14	28-NOV-14
4	PRD0485	28-NOV-14	11-DEC-14	18-DEC-14

Questions…