

SQL Basics Part II

Jonathan Miller
Learning SQL Series

Items To be Covered

- Review of SQL Basics I
- Join Types
 - Inner & Outer Joins
 - Self-Joins
 - Working With Multiple Queries
 - UNION, INTERSECT, MINUS
 - Subqueries & Correlated Subqueries
 - EXISTS & NOT EXISTS
- Functions
 - Group
 - String
 - Date
 - Number
- Participants Questions & Answers (hopefully!)

Review of SQL Basics I

- Structure of a SQL Statement

SELECT *column(s)*

FROM *table(s)*

WHERE condition(s)

ORDER BY column(s)

Optional Elements

Cartesian / Simple Join

```
SELECT mo_id, poc, parameter_desc  
FROM monitors, parameters
```

PARAMETERS

Parameter_Code	Parameter_Desc
----------------	----------------

44201	Ozone
-------	-------

Ozone

42101	CO
-------	----

CO

42401	SO2
-------	-----

SO2

81102	PM10
-------	------

PM10

MONITORS

MO_ID	SI_SI_ID	PA_PARAMETER_CODE	POC
-------	----------	-------------------	-----

SI_SI_ID

PA_PARAMETER_CODE

POC

1	1	44201	1
---	---	-------	---

1

44201

1

2	1	42101	1
---	---	-------	---

1

42101

1

3	1	42101	2
---	---	-------	---

1

42101

2

4	2	81102	1
---	---	-------	---

2

81102

1

5	2	44201	1
---	---	-------	---

2

44201

1

6	3	42401	1
---	---	-------	---

3

42401

1

Primary & Foreign Keys

- Primary Key (s)
 - Columns on a Table that Uniquely Identify a Record on the Table
 - Can be composed of 1 or more columns
- Foreign Key (s)
 - Column on a table that references the Primary Key of another Table
 - Can be composed of one or more columns

Inner Join Between 2 Tables

```
SELECT mo_id, poc, parameter_desc  
FROM monitors, parameters  
WHERE pa_parameter_code = parameter_code
```

PARAMETERS		MONITORS			
Parameter_Code	Parameter_Desc	MO_ID	SI_SI_ID	PA_PARAMETER_CODE	POC
44201	Ozone	1	1	44201	1
42101	CO	2	1	42101	1
42401	SO2	3	1	42101	2
81102	PM10	4	2	81102	1
		5	2	44201	1
		6	3	42401	61

The diagram illustrates an inner join between two tables: PARAMETERS and MONITORS. The PARAMETERS table has columns Parameter_Code and Parameter_Desc. The MONITORS table has columns MO_ID, SI_SI_ID, PA_PARAMETER_CODE, and POC. Yellow arrows show the join condition: PA_PARAMETER_CODE in MONITORS equals Parameter_Code in PARAMETERS. Red circles highlight MO_ID and POC in the MONITORS table and Parameter_Desc in the PARAMETERS table.

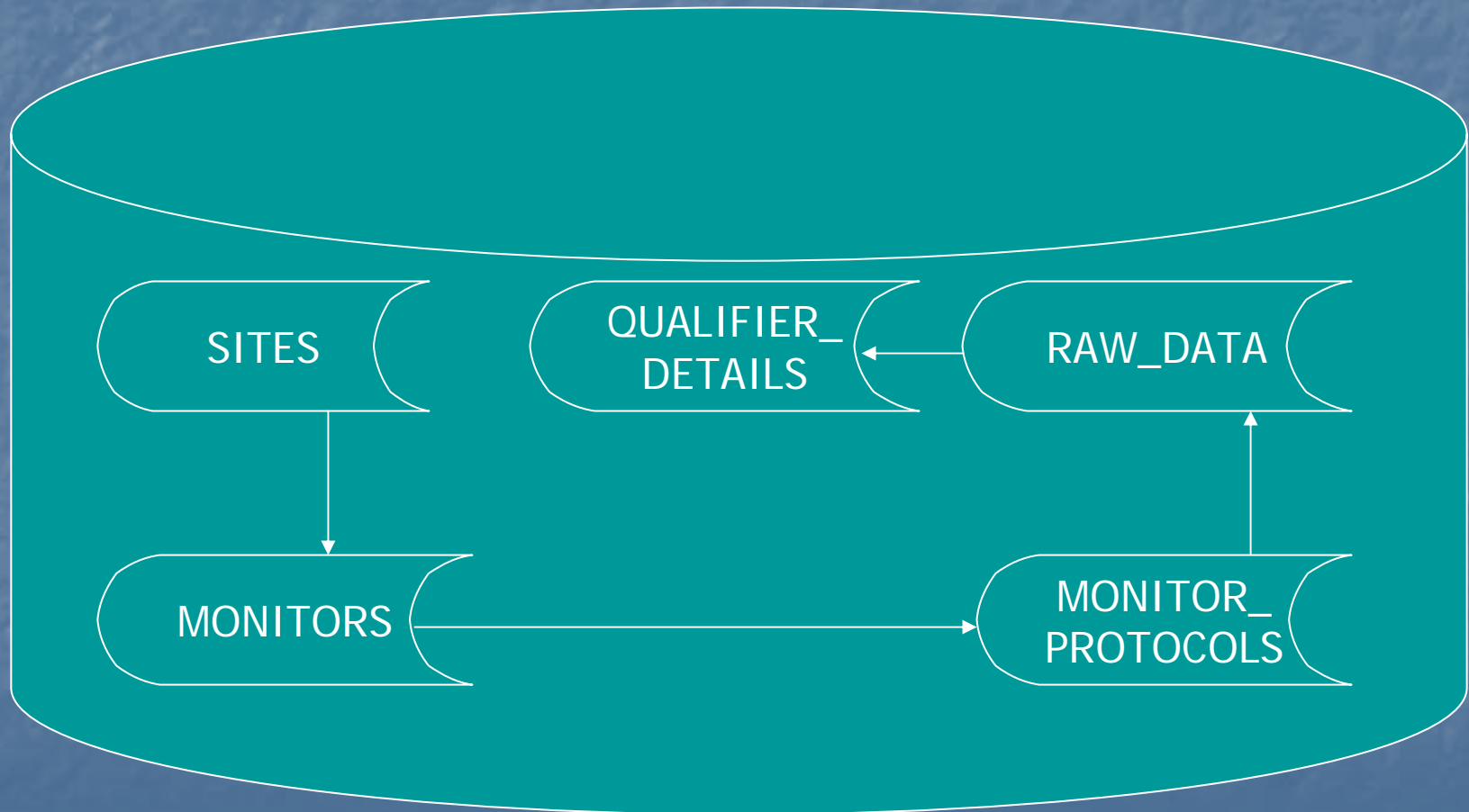
Joining Tables Together

- Joins Between Tables are Usually Based on Primary / Foreign Keys
- Make Sure Joins Between All Tables in the FROM Clause Exist
- List Joins Between Tables Before Other Selection Elements

Aliases

- “Shorthand” for Table or Column References
- SELECT Aliases Appear as Column Headers in the Output
- Aliases Cannot be Keywords (SELECT, FROM, WHERE, etc)

Join Types and Putting Multiple SQL Select Statements Together



Join Types

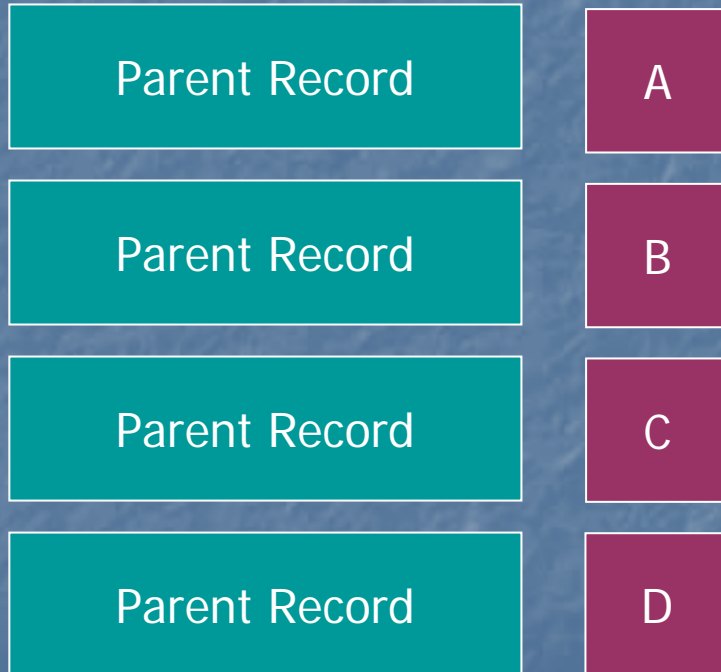
- Simple Join:
 - No links made between multiple tables
 - RESULT: Cartesian Product
- Inner Join
 - TABLE1.Foreign Key = TABLE2.Primary Key
 - RESULT: 1 record for each match between the 2 tables
- Outer Join
 - TABLE1.Foreign Key = TABLE2.Primary Key(+)
 - RESULT: 1 record for each match between the 2 tables AND 1 record for each where the record only exists in TABLE1

Outer Joins

- Place a “(+)” Next to the Column Name in the WHERE clause of the **SHORTER** table
- Where to Use:
 - When you have null values in a column that you are joining on
 - When a parent record may not have a child record
- A Table Can Only Be Outer Joined to One Other Table within a Given Query

Outer Join Examples

Self Joins

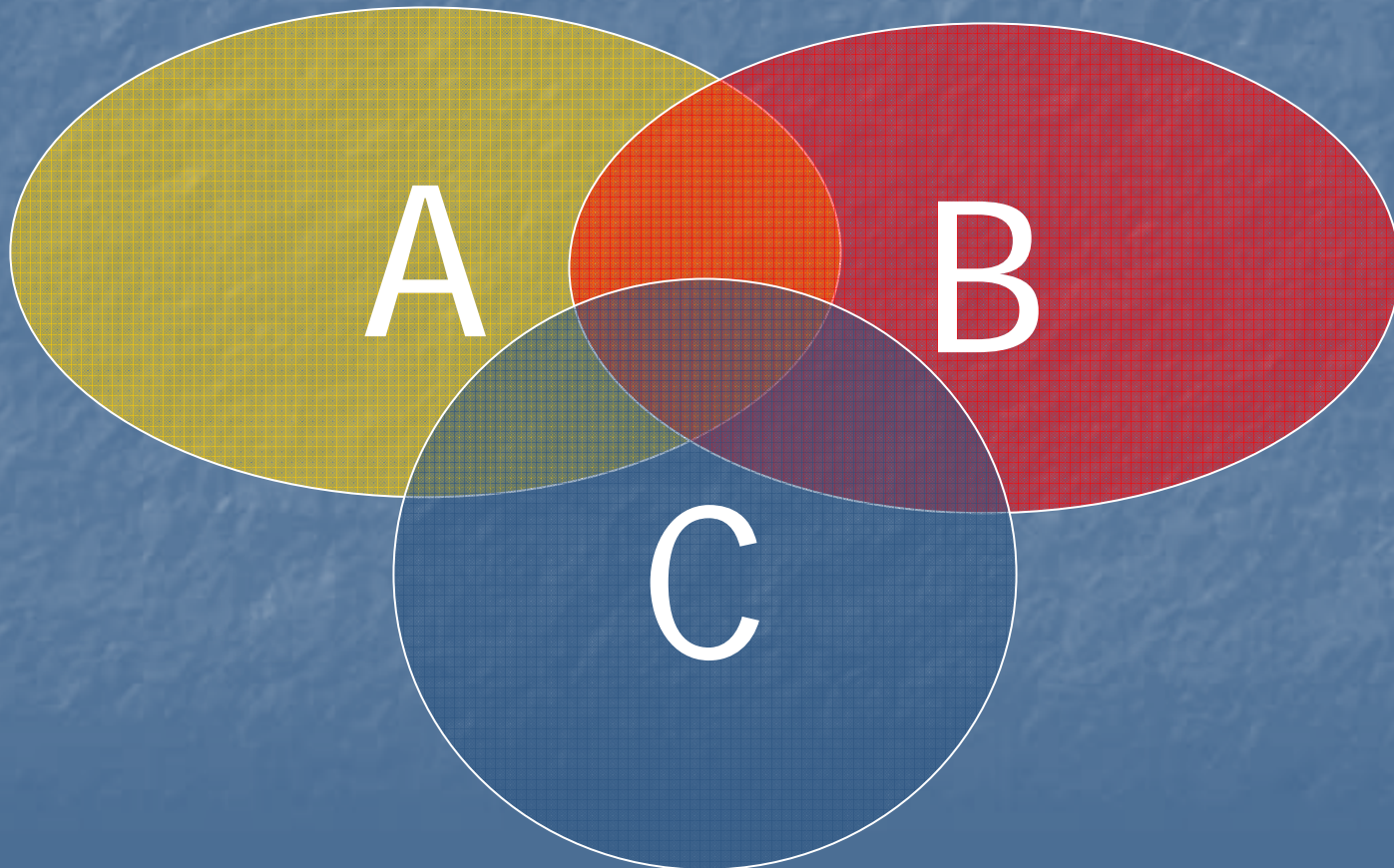


What is It

- Special Case of an Inner Join
- Use When You need "Rows" of a Child Record to Appear as "Columns"
- Specify the Same Table Multiple Times in the "FROM" Clause
 - Give Each Instance an Alias
 - In the "WHERE" Clause, Specify the Specific Value Desired

Self Join Example

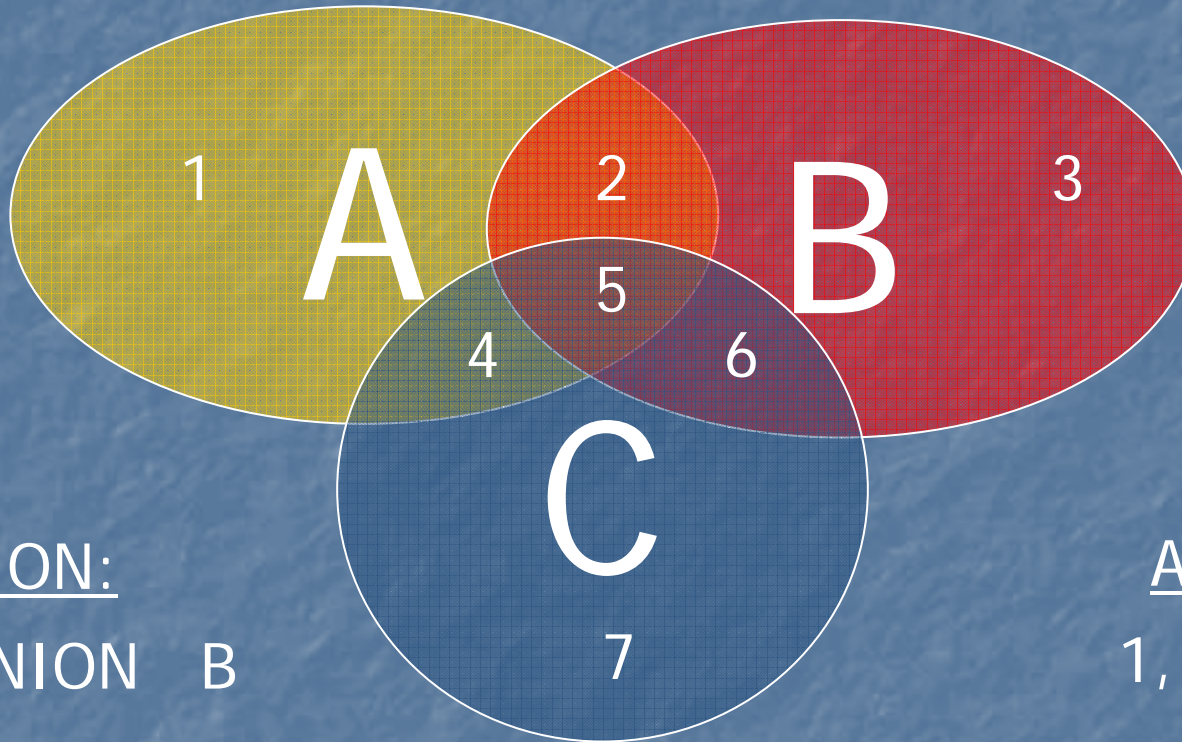
Union, Intersection, Minus



What Does it Do?

- Processes the Results of Separate Queries Into a Single Return
- **UNION:** Everything from Query 1 + Everything from Query 2
- **INTERSECTION:** Only records where the results between Query 1 and Query 2 are the same
- **MINUS:** Everything from Query 1 – Matching Records from Query 2

What Would be Returned?



QUESTION:

1. A UNION B
2. A MINUS B
3. A INTERSECT (B MINUS C)
4. C UNION (B INTERSECT A)

ANSWER:

1, 2, 3, 4, 5, 6

1, 4

2

2, 4, 5, 6, 7

Rules for These Queries

- Each Query Must Contain the same Number of and Type of Data Elements

Select arithmetic_mean From Annual_Summaries

Union

Select max_value From Summary_Maximums

- Only Unique Occurrences of the Records will be Returned Unless you Specify the "ALL" Keyword

SELECT ALL arithmetic_mean FROM Annual_Summaries

Union

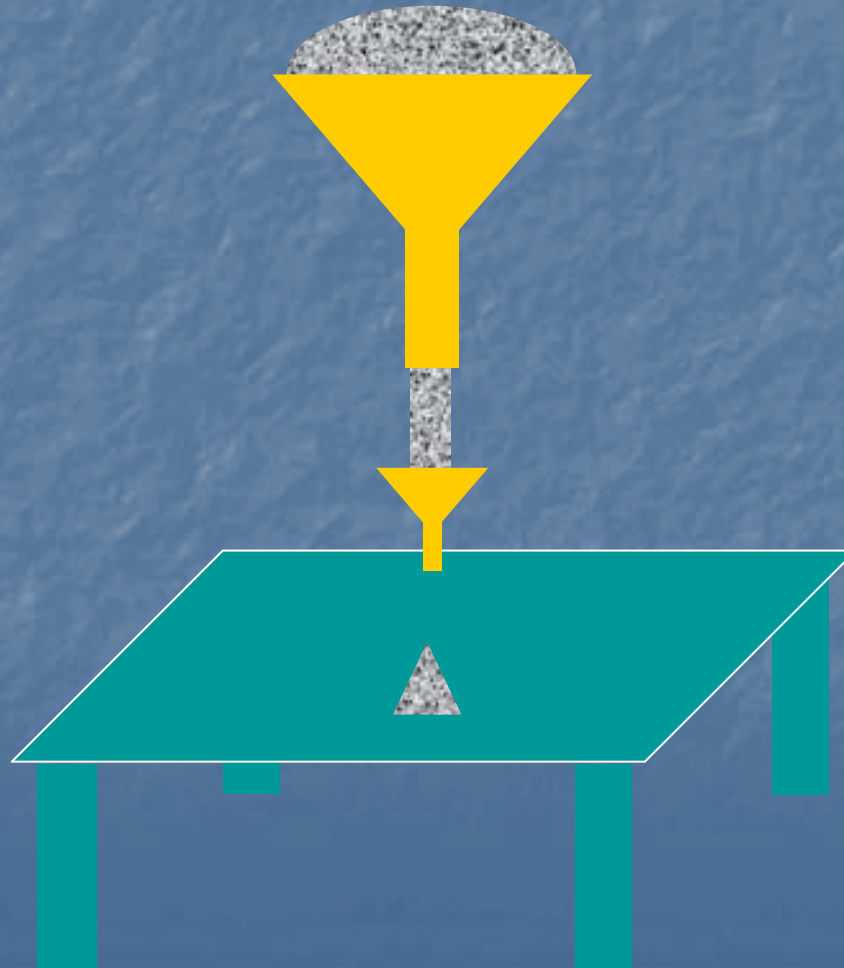
SELECT max_value FROM Summary_Maximums

- If you want the Data Sorted, Must be Referenced by Column Position

UNION, INTERSECT, MINUS

Examples

Subqueries & Correlated Subqueries

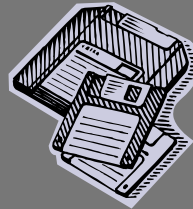
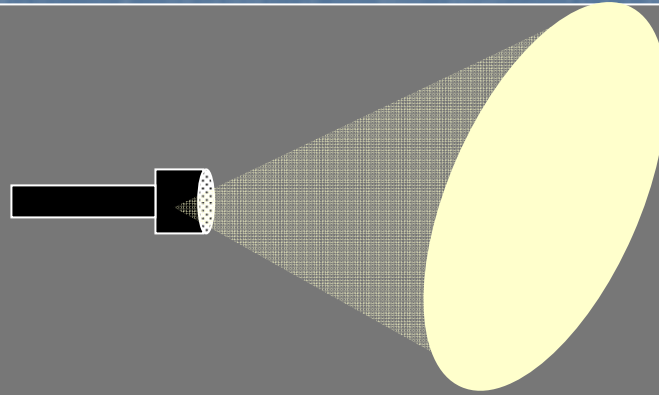


Subqueries

- A Subquery is Simply a Query Within Another Query
 - Used as a Filtering Mechanism
 - Additional Queries Appear in the WHERE Clause
 - Can be Nested up to 16 Levels
- A Correlated Subquery is a Subquery When the 2nd Query References a column From the Primary Query

Subquery Examples

Exists And NOT EXISTS



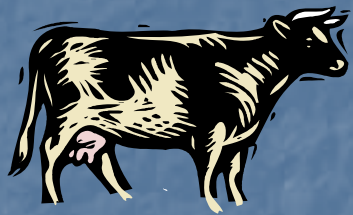
EXISTS and NOT EXISTS

- Special Subquery Used in the WHERE Clause
- Looks for the Existence of ANY Record in the Subquery
- Can be Much Faster than Using an "IN" Clause Since it Only Needs to Find 1 Record to Satisfy the Condition

EXISTS and NOT EXISTS

Example

Functions



What is a Function?

- Stored Software that Manipulates Submitted Elements and Returns Some Value
 - May be a SQL Standard Function (SUBSTR)
 - Could be Application Specific (GET_METHOD)
- Generally Two Categories
 - Creation of New Objects From Old Ones
 - Descriptions of Objects

Function Notation

- FUNCTION(parameter {datatype}, [optional parameters])
- Datatypes
 - VARCHAR2 ('Apple', 'Sample Text *', '1234')
 - NUMBER (1, 1.03, -15)
 - DATE – We'll talk about this one later...

Types of Functions

- String Functions
- Mathematical Functions
 - Single-Value Functions
 - Group-Value Functions
 - List Functions
- Conversion / Transformation Functions
- Date Functions

Where Can I Use Functions?

- SELECT Statement

- SELECT INITCAP(agency_desc)
FROM agencies

- WHERE Statement

- SELECT *

- FROM sites

- WHERE SUBSTR(lut_land_use_type, 1) = 'MOBILE'

DUAL Table

- Real Oracle Table With a Single Row
- Used When All Elements in the SELECT and WHERE Clauses do not Reference Any Tables

```
SELECT 'A'  
FROM DUAL
```

- Result: "A"

Common String Functions

- ||
- INITCAP(String)
- INSTR(String, set [, start [, occurrence]])
- LENGTH(String)
- LOWER(String) / UPPER(String)
- LPAD(String, Length [, 'set']) / RPAD(String, Length [, 'set'])
- LTRIM(String [, 'set']) / RTRIM(String [, 'set'])
- SUBSTR(String, Start [,count])

Concatenate (||)

- Glues two Strings Together

```
SELECT 'Go' || ' WOLFPACK'  
FROM DUAL
```

- Results: "Go WOLFPACK"

INITCAP, UPPER, LOWER

- Deals with the Capitalization of Strings
 - INITCAP – Capitalizes first letter of a string as well as after spaces, periods. Lower case for all other letters
 - UPPER – All Upper Case
 - LOWER – All Lower Case

```
SELECT INITCAP('gO woLFpack'),  
       UPPER ('gO woLFpack'),  
       LOWER('gO woLFpack')  
FROM dual
```

- RESULTS: "Go Wolfpack", "GO WOLFPACK", "go wolfpack"

"In String" - INSTR

- Finds the Position of a String Within a String

```
SELECT INSTR('Go Wolfpack', 'ac')  
FROM dual
```

- Results: 9

- Optionally can tell it Where to Start and What Occurrence to Find

```
SELECT INSTR('Go Wolfpack', 'o', 1, 2)  
FROM dual
```

- Results: 5

LENGTH

- Tells you How Long a String Is (Spaces Count)

```
SELECT LENGTH('Go Wolfpack')  
FROM dual
```

- Result: 11

LPAD / RPAD

- “Pads” Out a String To a Given Length
 - Defaults to Pad with a Space
 - You May Optionally Specify What Character to Use

```
SELECT LPAD('Carrot',10)
FROM dual
```

- Result: “ Carrot”

```
SELECT RPAD('Carrot',10, 'x')
FROM dual
```

- Result: “Carrotxxxxx”

LTRIM / RTRIM

- Trims off a Set of Characters from Either the Left (LTRIM) or Right (RTRIM) of a String
 - Keeps Trimming Until No More of the SET of Characters are Found

```
SELECT RTRIM('Go Wolfpack', 'k')  
FROM DUAL
```

- Results: "Go Wolfpac"

```
SELECT LTRIM('Go Wolfpack', 'oGfW')  
FROM dual
```

- Results: "lfpack"

Substring - SUBSTR

- Clips out a Portion of a String
 - May Tell it How Many Characters to Clip (Default is to clip to end of the string)

```
SELECT SUBSTR('Go Wolfpack', 7)  
FROM dual
```

- Results: "pack"

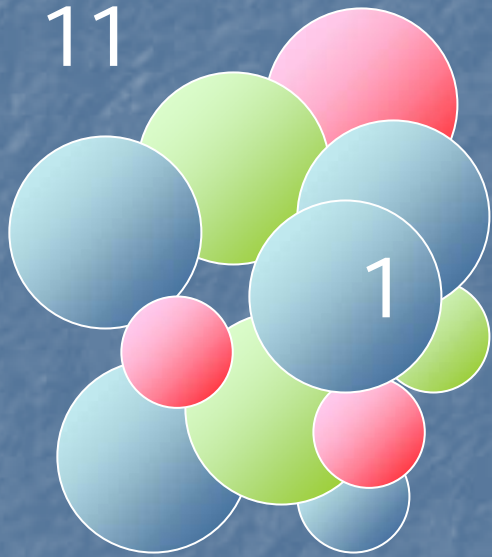
```
SELECT SUBSTR('Go Wolfpack', 7,3)  
FROM dual
```

- Results: "pac"

Mathematic Functions

- ABS – Absolute Value
- CEIL / FLOOR – Smallest Integer Above / Below Value
- MOD – Modulus
- POWER
- ROUND
- SQRT – Square Root
- TRUNC – Truncate
- List Functions
 - GREATEST (List)
 - LEAST (List)
- Group Functions
 - AVG, COUNT, MAX, MIN, STDDV, SUM, VARIANCE

Group Functions



By Size

7

4

2

*Total Count
By Color*

1

1

3

5

3

2

4

By Color & Size

Group Functions Example

- When you use the GROUP functions, you must specify the breaking elements in a GROUP BY clause
 - These must include all non-group function items in your SELECT statement
- Select the maximum arithmetic mean per year

```
SELECT annual_summary_year, MAX(annual_arithmetic_mean)  
FROM annual_summaries  
GROUP BY annual_summary_year
```

Transformation Functions

- NVL – Null Value Function
 - Default value to return if the column value is NULL
- DECODE – SQL version of an “IF” Statement
- Type Conversions
 - TO_DATE – Convert a String to a Date
 - TO_NUMBER – Convert a String to a Number
 - TO_CHAR – Convert a Date or Number to a String

NVL & DECODE Example

- How do you select a Tribal Site ID?
- IF a Tribal Code exists for a site, Use "TT" for the state code, the Tribal Code for the county and the site id. Otherwise use the State Code, County Code and Site ID

```
SELECT DECODE(ta_tribal_code, NULL, cn_stt_state_code, 'TT')  
      || '-' || NVL(ta_tribal_code, cc_cn_county_code)  
      || '-' || site_id "AIRS Site ID"  
FROM sites
```

Oracle Dates

- Dates stored in Oracle as a number
 - January 1, 4712 BC – December 31, 4712 AD
 - Use the TO_CHAR function to display the way you want it
 - System default is usually DD-MON-YY Format
 - Dates are stored down to the second
- "SYSDATE" = Right now (down to the second)

Date Comparisons

- If you compare two date values, the values must match down to the second
 - `TO_DATE('20041116', 'YYYYMMDD') = sysdate` is not a true statement.
- You can perform any arithmetic comparison between 2 dates
 - `>` , `<` , `=` , `!=` , `BETWEEN`

Date Functions

- You can convert a string to a date (TO_DATE) and a date to a string (TO_CHAR)
 - TO_DATE('2004-NOV-02', 'YYYY-MON-DD')
 - TO_CHAR(sysdate, 'YYYYMMDD')
- You can truncate a date (TRUNC) to make comparisons easier
 - TRUNC(SYSDATE) = Midnight of today's date
 - TRUNC(SYSDATE, 'YYYY') = January 1 of the current year
 - TRUNC(SYDATE, 'MM') = Day 1 of the current month

Date Comparison Example

- Select the MO_IDs, Reporting Agency Codes, and year for monitors that reported data

```
SELECT ans.mo_mo_id, ar.sa_ag_agency_code, ans.annual_summary_year year
FROM annual_summaries ans, agency_roles ar
WHERE ans.mo_mo_id = ar.mo_mo_id
      AND ar.rol_role = 'REPORTING'
      AND TO_DATE(ans.annual_summary_year || '0101', 'YYYYMMDD')
          BETWEEN TRUNC(ar.agency_role_begin_date, 'YYYY') AND
          NVL(ar.agency_role_end_date, sysdate)
```

Date Arithmetic

- If you subtract one date from another you get the number of days between the two dates
- If you add or subtract a number to a date you get a date that many days in the future (+) or past (-)

More Date Functions

- `ADD_MONTHS` – Move a date x months into the future
- `LAST_DAY` – Returns the date that is the last day of the month for the supplied month
- `MONTHS_BETWEEN` – Number of months between 2 dates
- `ROUND` – Like `TRUNC`, but will give you the higher formatted value if more than $\frac{1}{2}$ of the period has expired

Date Arithmetic & Function Example

- Select the time right now (Hour, minute, second), the number of days since the first of the year, the number of days until the end of the year, the number of days until the end of the month, number of hours left in the day

```
SELECT TO_CHAR(sysdate, 'HH24:MI:SS') right_now,  
       TRUNC(sysdate) – TRUNC(sysdate, 'YYYY') first_of_year,  
       TRUNC(ADD_MONTHS(sysdate,12), 'YYYY') – TRUNC(sysdate) end_of_year,  
       TRUNC(ADD_MONTHS(sysdate,1), 'MON') – TRUNC(sysdate) end_of_month,  
       ROUND((TRUNC(sysdate+1) – sysdate) * 24)  
FROM DUAL
```

Summary

- Functions are pre-written programs that may take variables and returns a value
- Oracle has several built-in functions
 - String
 - Math
 - Date
 - Translation
- Functions can be used in combination

Summary (Continued)

- A function may have multiple meanings based on the type of information provided (TRUNC for example)
- Oracle Dates
 - Be careful you know what time you have
 - Date arithmetic
 - Date functions

Any Questions?