ITC 5104 Database Design and SQL

Lecture 1

Chapter 2 Oracle 12c: SQL

Basic SQL SELECT Statements

Objectives

- Create initial database
- Identify keywords, mandatory clauses, and optional clauses in a SELECT statement
- Select and view all, one or multiple columns of a table
- Use a column alias to clarify contents of a column
- Perform basic arithmetic operations in a SELECT clause
- Remove duplicate items using either DISTINCT or UNIQUE keywords
- Some basic settings for SQL Developer

Create Initial Database

- For this course we will use the SQL Developer interface as a client software tool
- SQL Developer will be used to connect to the Oracle 12c database residing on Dilbert
- For those of you with the textbook you can refer to Appendix B for an overview of SQL Developer (is based on previous version)
- These steps were outlined in both Lab Exercise 1 and the notes on installing SQL Developer on your own system

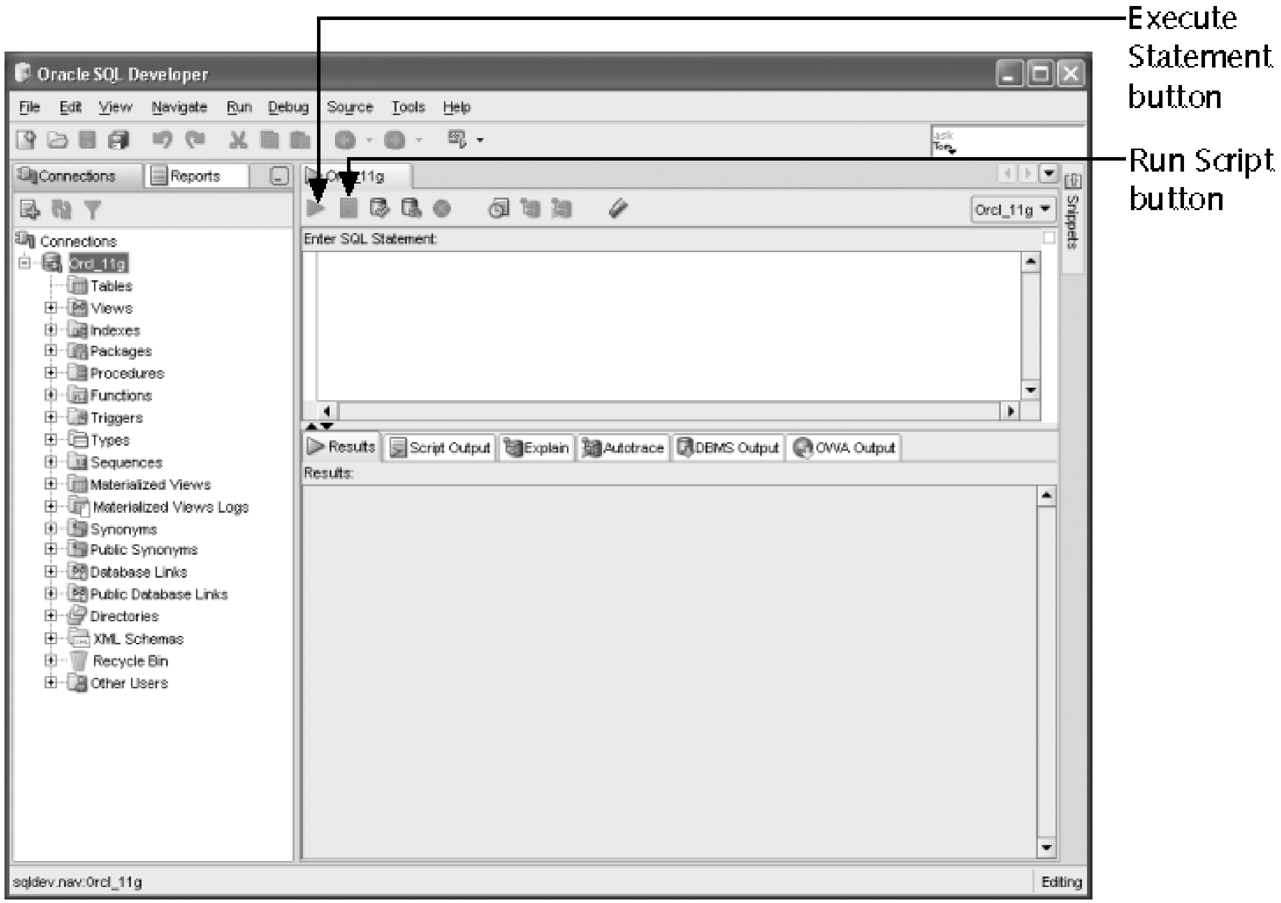
Some Facts About Data

- 2.5 quintillion bytes of data generated each day 2.5 exabytes
- This equates to about 1 Zettabytes per year or (1000 Exabyte)
- Facebook scans roughly **105 terabytes** of data each half hour

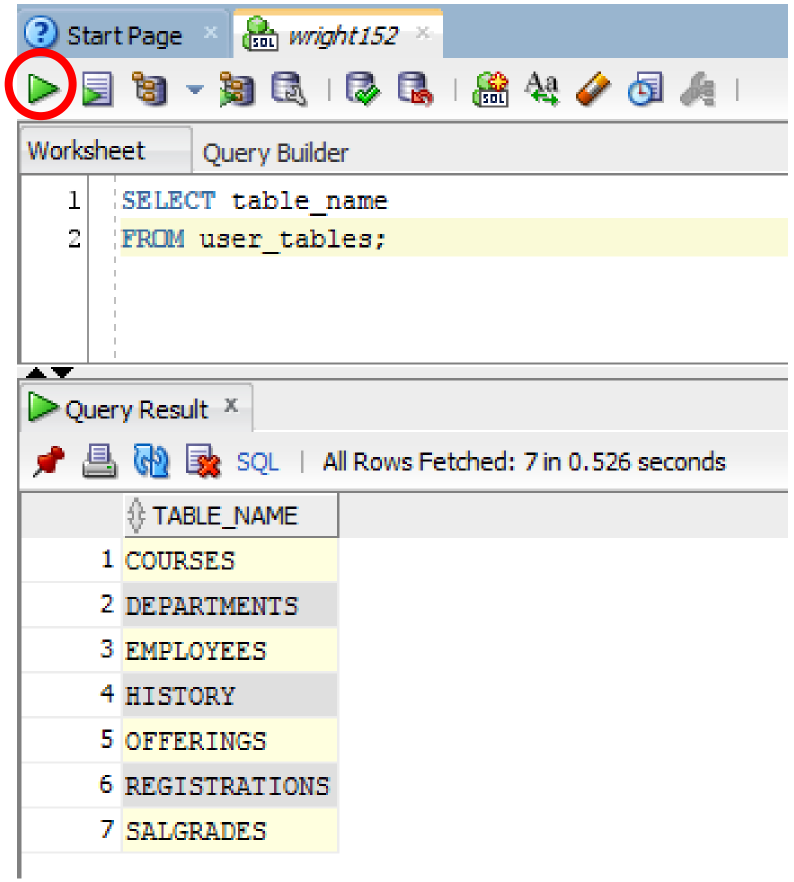
Some Facts About Data

- Walmart handles 2.5 petabytes every hour in transactions
- Walmart building cloud database to process 2.5 petabytes per hour
- Business data doubles every 1.2 years
- As of July 2017 over I.2 Billion active users for Gmail

SQL Developer Interface



What Tables Do I Own?



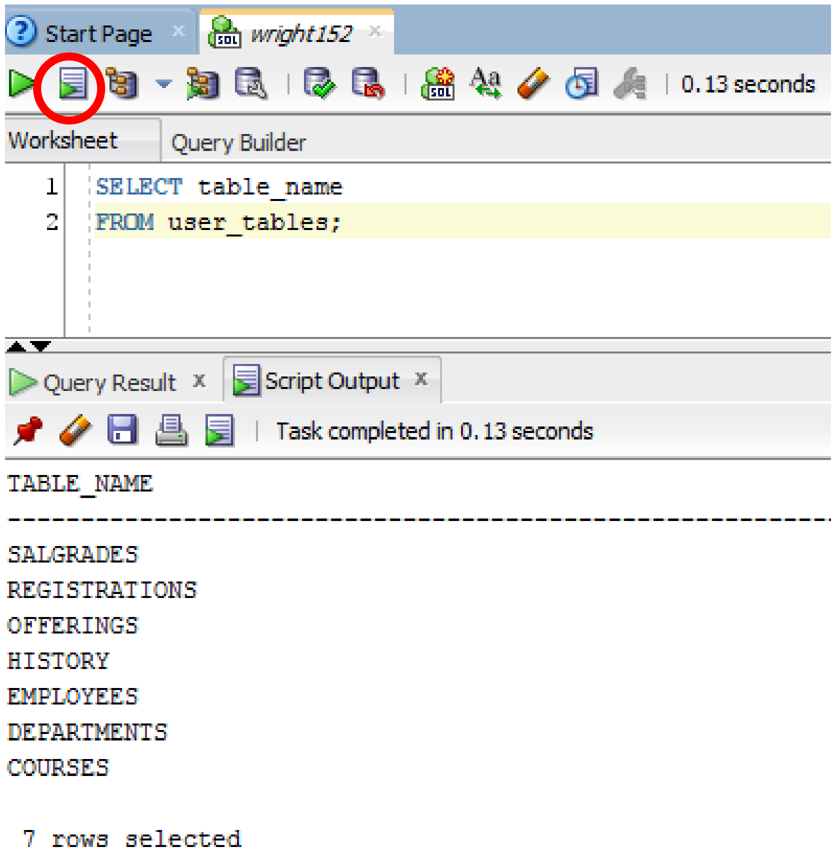
```
SELECT table_name
FROM user_tables;
```

All Rows Fetched: 7 in 0.526 seconds

	TABLE_NAME
1	COURSES
2	DEPARTMENTS
3	EMPLOYEES
4	HISTORY
5	OFFERINGS
6	REGISTRATIONS
7	SALGRADES

- user_tables is a data dictionary view that will display the tables you own
- To execute this SQL statement the Execute Statement button is clicked
- You can also press the Ctrl+Enter keys to execute a statement
- You have to have your cursor in the statement you wish to execute, other wise nothing happens

What Tables Do I Own?



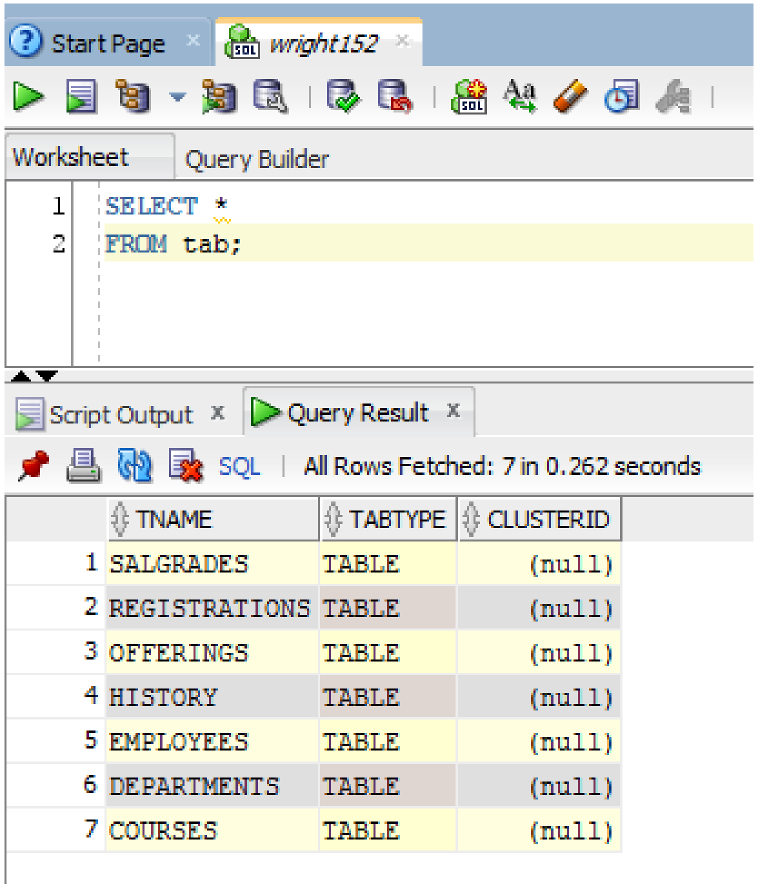
```sql
SELECT table_name
FROM user_tables;
```

```
TABLE_NAME
--------------------------------------------------------
SALGRADES
REGISTRATIONS
OFFERINGS
HISTORY
EMPLOYEES
DEPARTMENTS
COURSES

 7 rows selected
```

- The output is the same, just in a different format
- This output was produced by clicking the Run Script button or by pressing the F5 function key
- This produces a text based output not as structured as the previous method
- The window in this case does not clear each time the button is pressed, instead it will accumulate and show all outputs that have been generated

What Tables Do I Own?

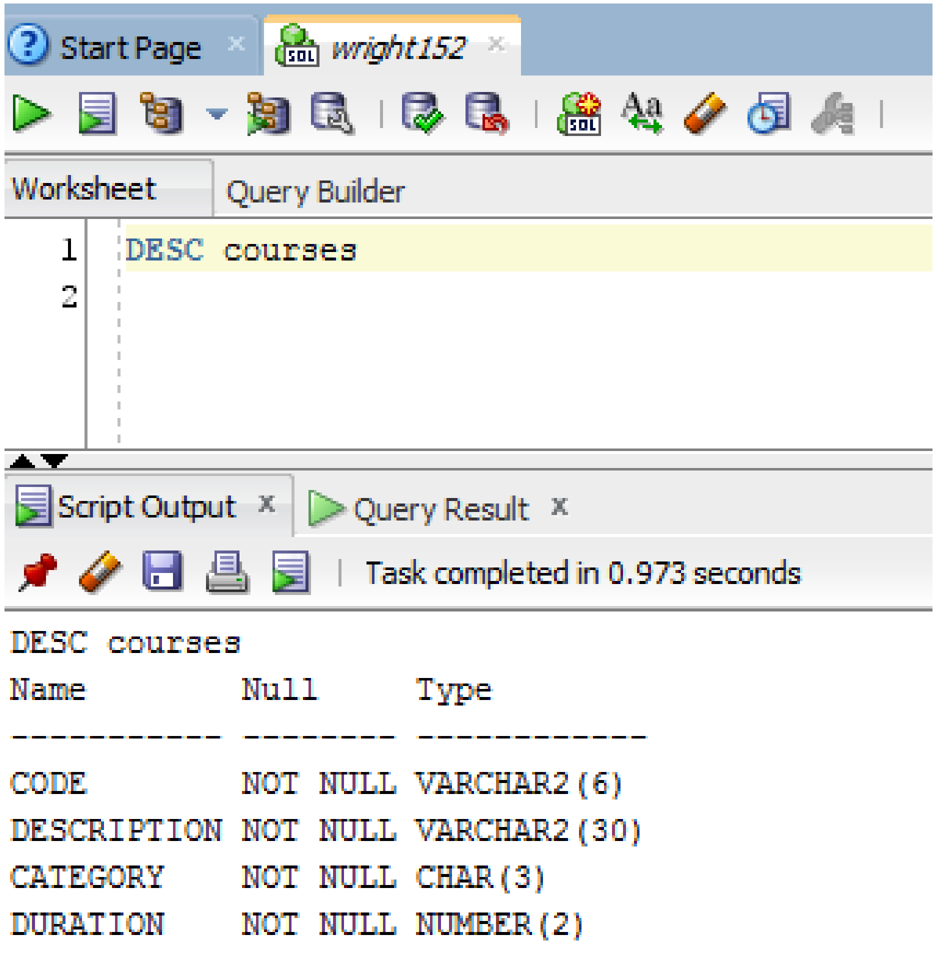


```
SELECT *
FROM tab;
```

	TNAME	TABTYPE	CLUSTERID
1	SALGRADES	TABLE	(null)
2	REGISTRATIONS	TABLE	(null)
3	OFFERINGS	TABLE	(null)
4	HISTORY	TABLE	(null)
5	EMPLOYEES	TABLE	(null)
6	DEPARTMENTS	TABLE	(null)
7	COURSES	TABLE	(null)

- This is an alternate table called TAB
- TAB is a pseudo table, you will not see it but you can use it
- Its purpose is to display the table objects and other objects called VIEWS
- Will also use to see any dropped tables you may have in your schema

DESCRIBE the Structure of a Table

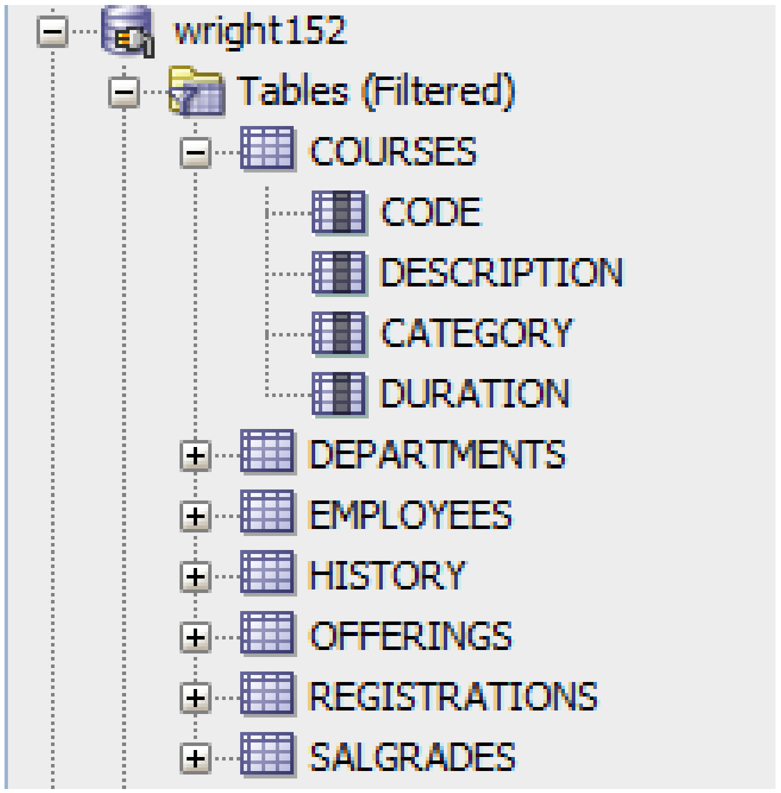


```
DESC courses
```

```
DESC courses
Name        Null     Type
----------- -------- ------------
CODE        NOT NULL VARCHAR2(6)
DESCRIPTION NOT NULL VARCHAR2(30)
CATEGORY    NOT NULL CHAR(3)
DURATION    NOT NULL NUMBER(2)
```

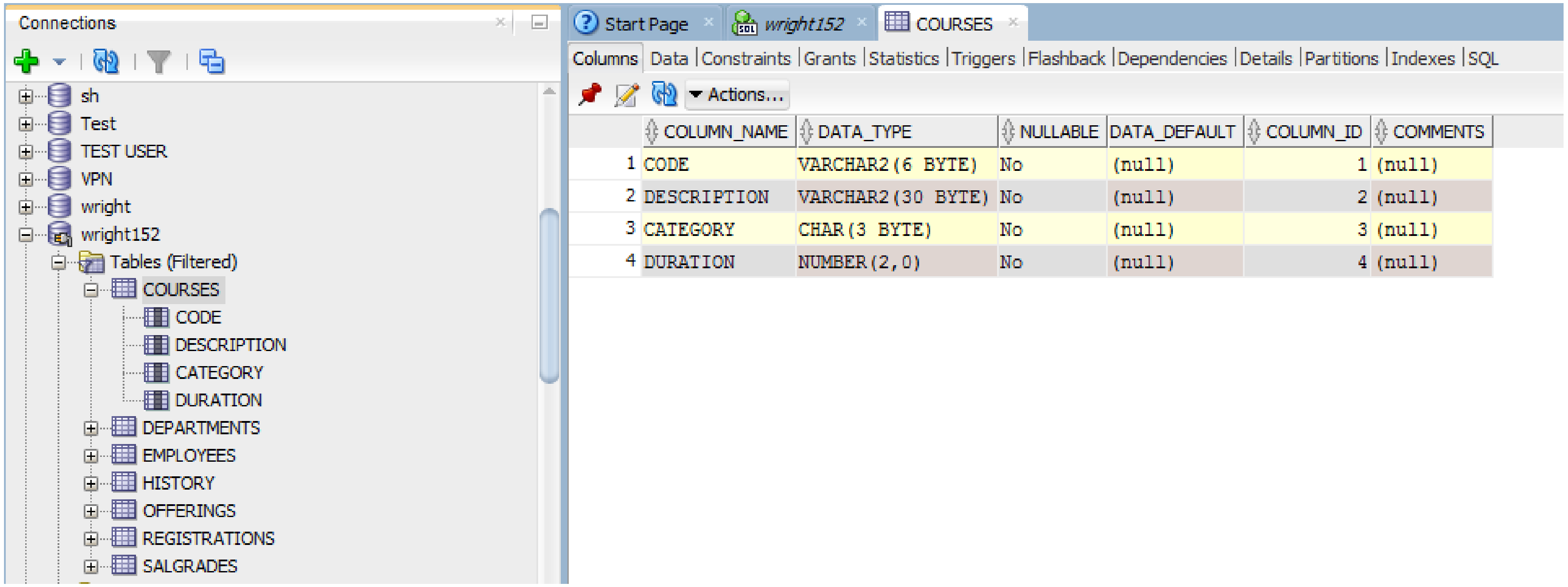
- The DESC or DESCRIBE command followed by a table name will show the structure of that table
- The structure is the names of the columns and their data types with sizes
- The NULL column shows any columns that have been defined as NOT NULL

Viewing Tables in SQL Developer



- In SQL you can expand the tree where it says Tables, this is on the left side of the window
- The tables you have in your database account will be displayed
- You can click on the actual table name to see details of the table

Viewing Tables in SQL Developer



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CODE	VARCHAR2(6 BYTE)	No	(null)	1	(null)
2	DESCRIPTION	VARCHAR2(30 BYTE)	No	(null)	2	(null)
3	CATEGORY	CHAR(3 BYTE)	No	(null)	3	(null)
4	DURATION	NUMBER(2,0)	No	(null)	4	(null)

- When you click on a table name the columns in the table are shown in the tree and details of the table appear to the right in the upper pane

- You will also notice a series of Tabs above the output, you can click these to see various aspects of the table

SELECT Statement

- The majority of the SQL operations performed on a database are done with the SELECT statement
- The SELECT statement allows a user to retrieve data from existing tables
- Note that SELECT does not change data, it simply retrieves it from the database
- The user can ask for all of the fields and records in a table or can request only certain fields and records, up to you as the user
- The SELECT statement asks the database a question, also known as a query

SELECT Statement

- After querying the database, the results are displayed
- What is displayed is the answer to the question (query) asked by the user
- Keywords begin a section of a SELECT statement are called a clause:
    - SELECT clause
    - FROM clause
    - WHERE clause
    - ORDER BY clause
    - There are more of them

SELECT Statement Syntax

```
SELECT  [DISTINCT | UNIQUE] (*, columnname [ AS alias], …)
        FROM      tablename
        [WHERE    condition]
        [GROUP BY group_by_expression]
        [HAVING   group_condition]
        [ORDER BY columnname];
```

This is a syntax diagram for the SELECT statement. We will discuss all of these pieces and even a few that are not here over the course of the semester

We will make reference to this syntax diagram again and again as we build the SELECT statement

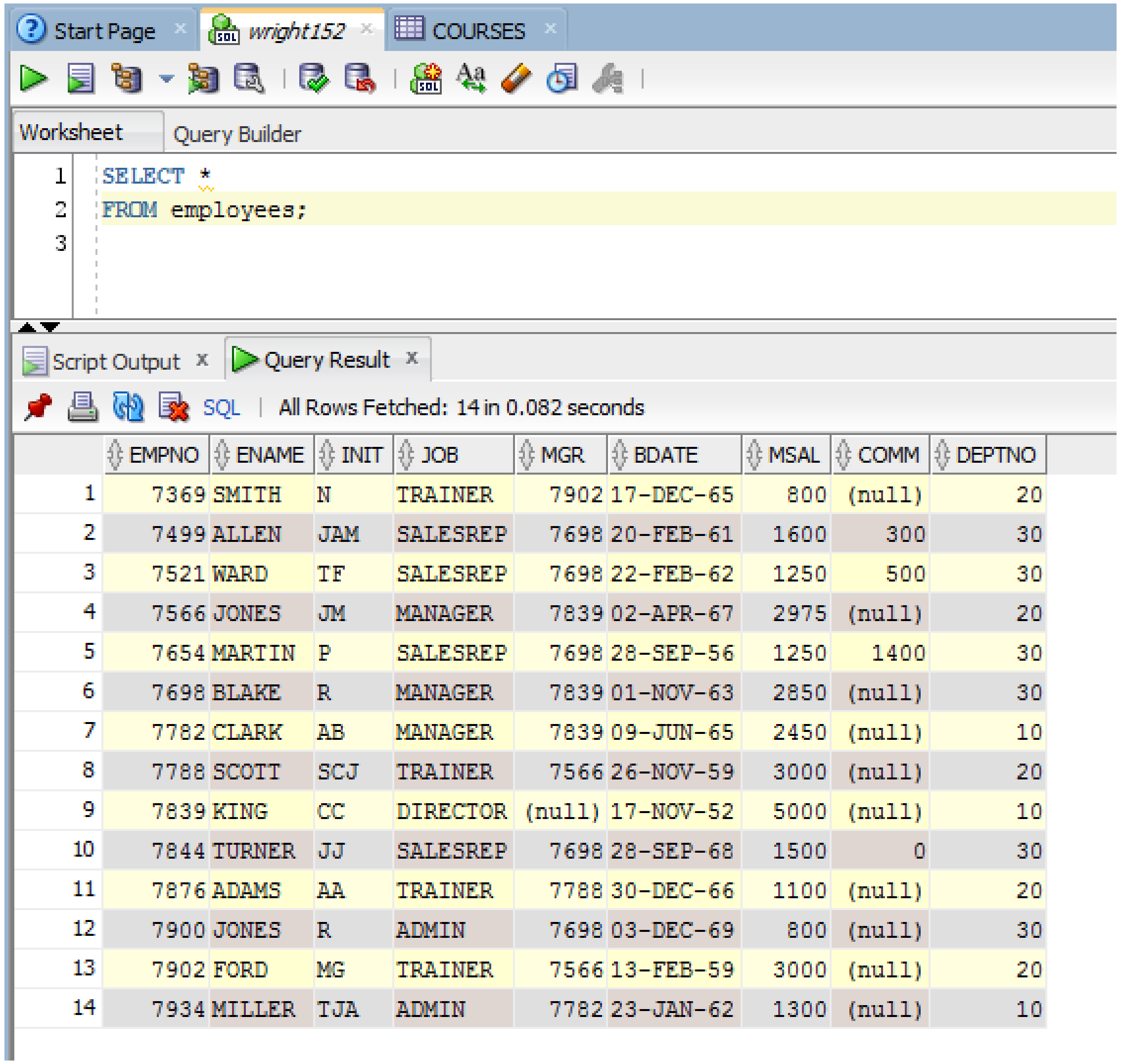
SELECT Statement Syntax Style

- The other capitalized components in the SELECT statement also begin clauses
- The only required clauses are SELECT and FROM
- The asterisk character is used to indicate that all columns available are to be displayed
- To select the entire table, the syntax is:

  SELECT *

  FROM *table_name*;
- By default, column headings displayed in a query are capitalized
- *** It is best to enter your SQL command over several lines, beginning each line with a keyword ***
- *** Please enter query over multiple lines, it is not a contest to see who can use the fewest lines, make it clear and readable, and easier to debug ***

Selecting all Rows in a Table



```
SELECT *
FROM employees;
```

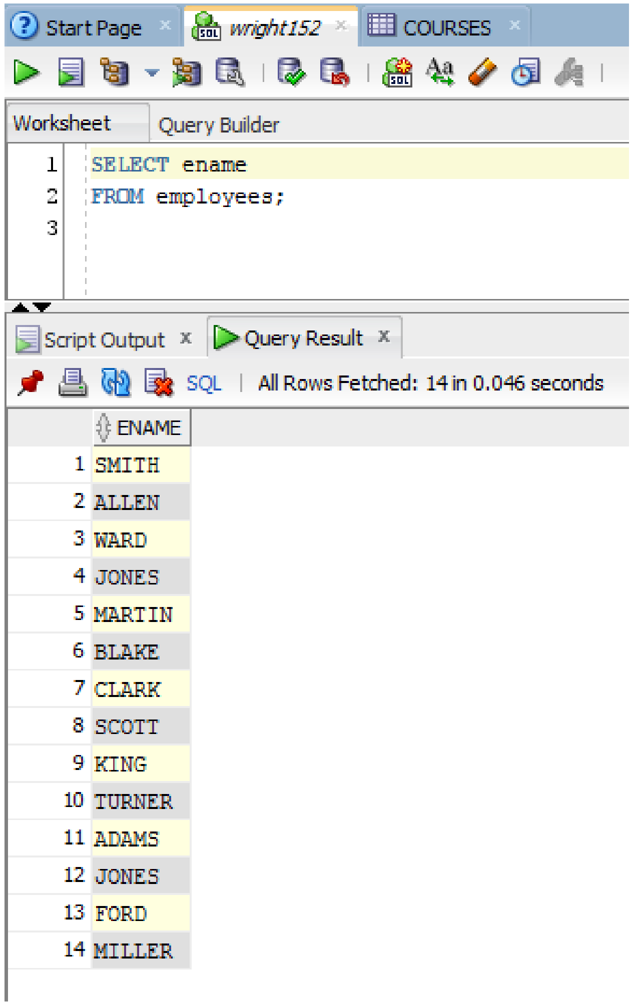
All Rows Fetched: 14 in 0.082 seconds

	EMPNO	ENAME	INIT	JOB	MGR	BDATE	MSAL	COMM	DEPTNO
1	7369	SMITH	N	TRAINER	7902	17-DEC-65	800	(null)	20
2	7499	ALLEN	JAM	SALESREP	7698	20-FEB-61	1600	300	30
3	7521	WARD	TF	SALESREP	7698	22-FEB-62	1250	500	30
4	7566	JONES	JM	MANAGER	7839	02-APR-67	2975	(null)	20
5	7654	MARTIN	P	SALESREP	7698	28-SEP-56	1250	1400	30
6	7698	BLAKE	R	MANAGER	7839	01-NOV-63	2850	(null)	30
7	7782	CLARK	AB	MANAGER	7839	09-JUN-65	2450	(null)	10
8	7788	SCOTT	SCJ	TRAINER	7566	26-NOV-59	3000	(null)	20
9	7839	KING	CC	DIRECTOR	(null)	17-NOV-52	5000	(null)	10
10	7844	TURNER	JJ	SALESREP	7698	28-SEP-68	1500	0	30
11	7876	ADAMS	AA	TRAINER	7788	30-DEC-66	1100	(null)	20
12	7900	JONES	R	ADMIN	7698	03-DEC-69	800	(null)	30
13	7902	FORD	MG	TRAINER	7566	13-FEB-59	3000	(null)	20
14	7934	MILLER	TJA	ADMIN	7782	23-JAN-62	1300	(null)	10

Selecting a Single Column

- In most cases, you will only want to see the data in certain columns
- This may be because some data is sensitive and you do not want others to see it, or the results wrap because of the number of columns being displayed
- Could also be that there is just too much data to see and you do not need to see all the data
- Selecting specific columns in a SELECT statement is called *projection*. You may select one or many columns, as desired

Selecting a Single Column



```
SELECT ename
FROM employees;
```

All Rows Fetched: 14 in 0.046 seconds

	ENAME
1	SMITH
2	ALLEN
3	WARD
4	JONES
5	MARTIN
6	BLAKE
7	CLARK
8	SCOTT
9	KING
10	TURNER
11	ADAMS
12	JONES
13	FORD
14	MILLER

- Only the ename column is requested in the query result
- This query shows all of the ename values in the EMPLOYEES table

Writing SQL Statements – Syntax Rules

- SQL statements are **not** case sensitive
- They can be entered on one or many lines
- Keywords cannot be split across lines or abbreviated
- Clauses are usually placed on separate lines for readability and ease of editing
- Tabs and indents make code more readable
- The preferred style is for Keywords to be entered in uppercase, while all other words such as table names and columns to be entered in lowercase (companies usually have coding standards)
- SQL statements end with a semi-colon (**;**)
- Use of the **;** is shown on the previous slide
- **If you are executing only a single command in SQL Developer you can omit the semi-colon, good practice to include the semi-colon though**

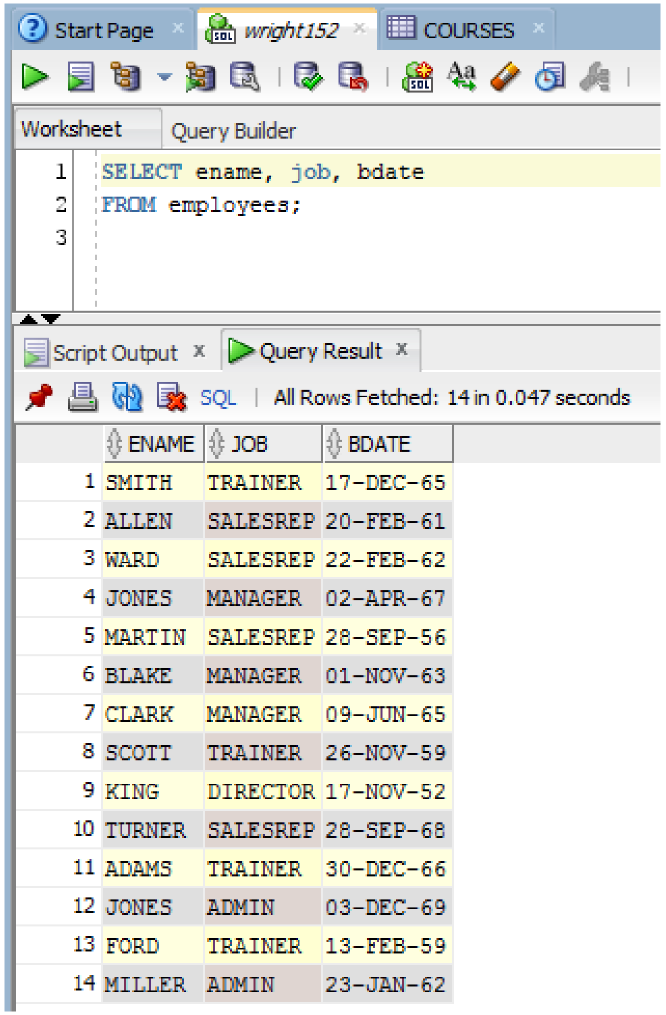
Writing SQL Statements – Best Practices

- Based on the previous information, will the following SQL statements produce the same results as the previous slide?
  1. SELECT ENAME FROM EMPLOYEES;
  2. select ename from employees;
  3. SELECT ename FROM employees;
  4. SELECT ename
     FROM employees;
- Which statement do I expect to use in this case?

Selecting Multiple Columns From a Table

- The SELECT statement can also be used to retrieve multiple columns from a table
- A comma is used to separate the column names, a space is not necessary but is used to improve readability
- The order the columns are listed is the order in which the columns will display

Selecting Multiple Columns From a Table



```
SELECT ename, job, bdate
FROM employees;
```

All Rows Fetched: 14 in 0.047 seconds

	ENAME	JOB	BDATE
1	SMITH	TRAINER	17-DEC-65
2	ALLEN	SALESREP	20-FEB-61
3	WARD	SALESREP	22-FEB-62
4	JONES	MANAGER	02-APR-67
5	MARTIN	SALESREP	28-SEP-56
6	BLAKE	MANAGER	01-NOV-63
7	CLARK	MANAGER	09-JUN-65
8	SCOTT	TRAINER	26-NOV-59
9	KING	DIRECTOR	17-NOV-52
10	TURNER	SALESREP	28-SEP-68
11	ADAMS	TRAINER	30-DEC-66
12	JONES	ADMIN	03-DEC-69
13	FORD	TRAINER	13-FEB-59
14	MILLER	ADMIN	23-JAN-62

- The ename, job and bdate columns appear in the result table
- The ename was specified first in the SELECT CLAUSE so it appears first followed by the job, then by the bdate column
- A comma separates the two column names
- The statement is terminated with a semicolon
- They columns display in the order they are specified

Operations Within the SQL Statement

- We will now look at some other operations that can be done with the SELECT statement:
  - Column aliases
  - Arithmetic operations
  - Elimination of duplicate output
  - Display rows on multiple lines

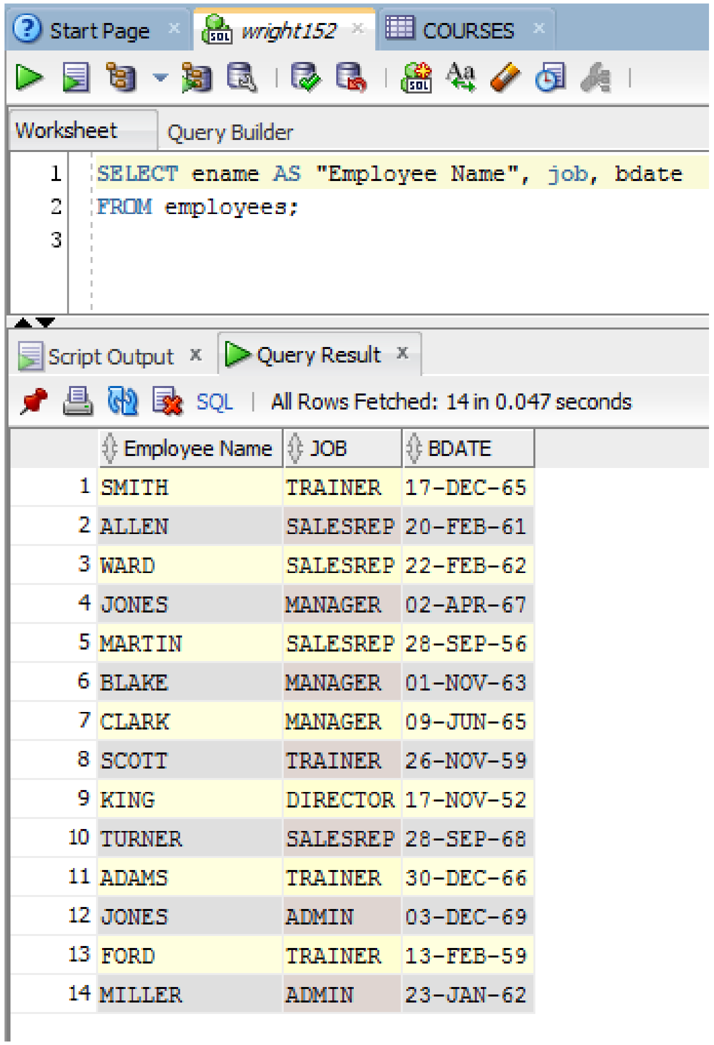
Using Column Aliases

- In some cases, the column (attribute) name may not be a good descriptor of the data displayed
- To better describe the data listed, you can substitute a column alias for the column name in the results of a query
- To use a column alias, place the name of the alias beside the column name in the query

Using Column Aliases

- The optional keyword AS can be included to indicate that what follows is the column alias. It distinguishes the column name from the column alias
- If the alias name contains any spaces or special symbols, or if you want to maintain the case of the alias, you must enclose the alias in *double quotation marks*

Using Column Aliases



```
SELECT ename AS "Employee Name", job, bdate
FROM employees;
```

All Rows Fetched: 14 in 0.047 seconds

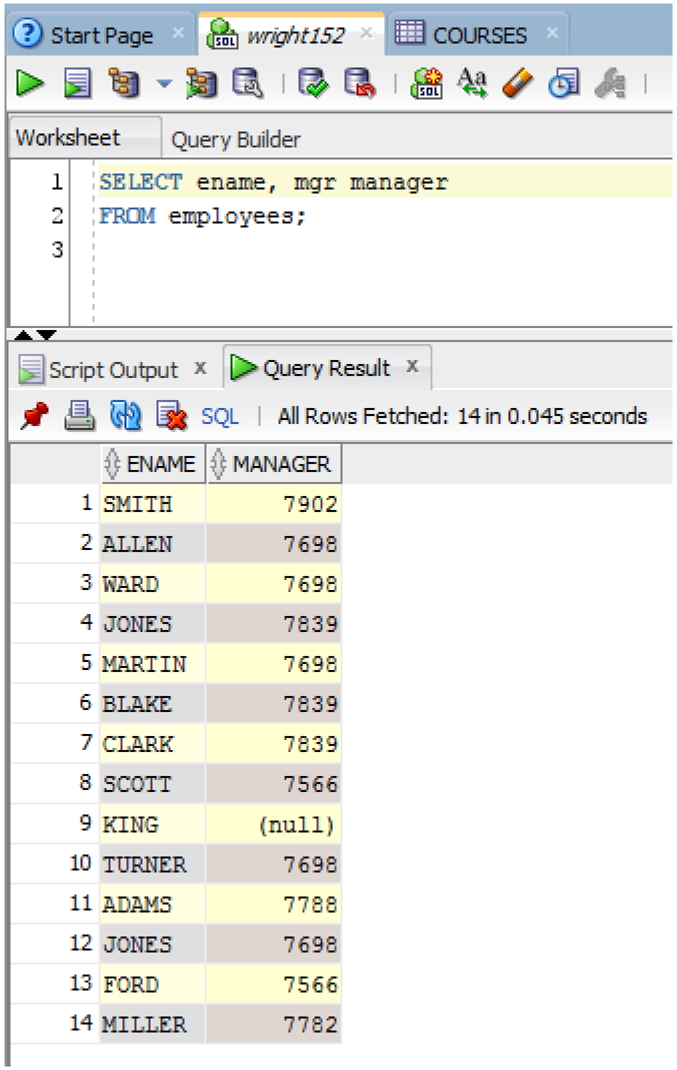
	Employee Name	JOB	BDATE
1	SMITH	TRAINER	17-DEC-65
2	ALLEN	SALESREP	20-FEB-61
3	WARD	SALESREP	22-FEB-62
4	JONES	MANAGER	02-APR-67
5	MARTIN	SALESREP	28-SEP-56
6	BLAKE	MANAGER	01-NOV-63
7	CLARK	MANAGER	09-JUN-65
8	SCOTT	TRAINER	26-NOV-59
9	KING	DIRECTOR	17-NOV-52
10	TURNER	SALESREP	28-SEP-68
11	ADAMS	TRAINER	30-DEC-66
12	JONES	ADMIN	03-DEC-69
13	FORD	TRAINER	13-FEB-59
14	MILLER	ADMIN	23-JAN-62

- The column heading ename has been replaced with the alias Employee Name
- Notice JOB and BDATE, are displayed in upper case still
- Since "Employee Name" is enclosed in double-quotations it displays in the case specified as well I am allowed to use the space character
- If I wanted all column headings to appear the same how would I accomplish this?

Using Column Aliases

- As you can see from the next slide, it is sometimes difficult to tell the column name from the alias without the AS keyword – retail is the column name, price is the alias!

Using Column Aliases



```
SELECT ename, mgr manager
FROM employees;
```

	ENAME	MANAGER
1	SMITH	7902
2	ALLEN	7698
3	WARD	7698
4	JONES	7839
5	MARTIN	7698
6	BLAKE	7839
7	CLARK	7839
8	SCOTT	7566
9	KING	(null)
10	TURNER	7698
11	ADAMS	7788
12	JONES	7698
13	FORD	7566
14	MILLER	7782

- The AS keyword sometimes clarifies that an alias was actually requested
- Here it could be confusing since MANAGER could have been an actual column and a comma was omitted in the SELECT statement
- MANAGER is the alias for MGR

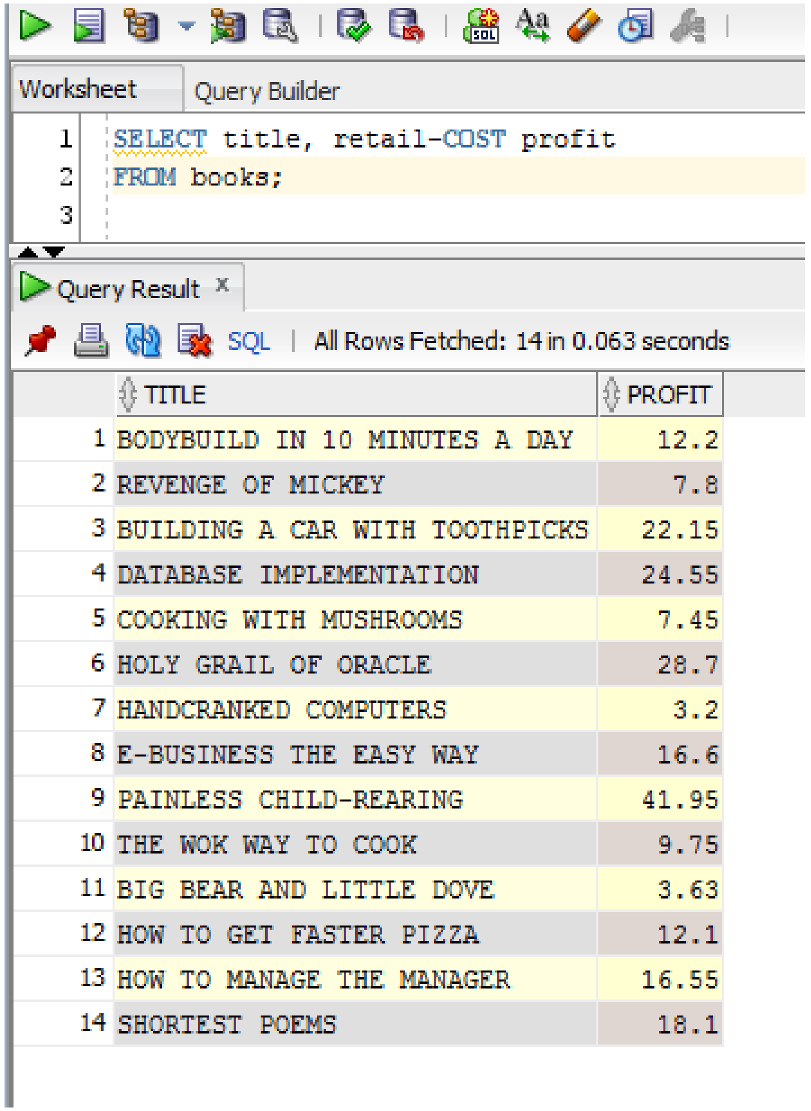
Column Alignment

- Data for text or character fields is left-aligned
- Data for date fields is left-aligned
- Data for numeric data is right-aligned
- By default, Oracle does not display insignificant zeroes.
- If there was a value of 54.50, the zero on he right side is insignificant so it is suppressed in the result, you would only see 54.5 display
- 25 is would actually be displayed for 25.00, the zeroes are insignificant so they are suppressed in the result

Arithmetic Expressions

- You can create expressions on NUMBER and DATE data by using arithmetic operators:
  - + Add
  - - Subtract
  - * Multiply
  - / Divide
- Arithmetic operators can be used in ***any clause*** of a SQL statement ***except*** the FROM clause
- Exponents are not supported, use multiplication
- Arithmetic operations follow the standard order of operations (BEDMAS, or PEMDAS if you learned either of these acronyms)

Using Arithmetic Operations

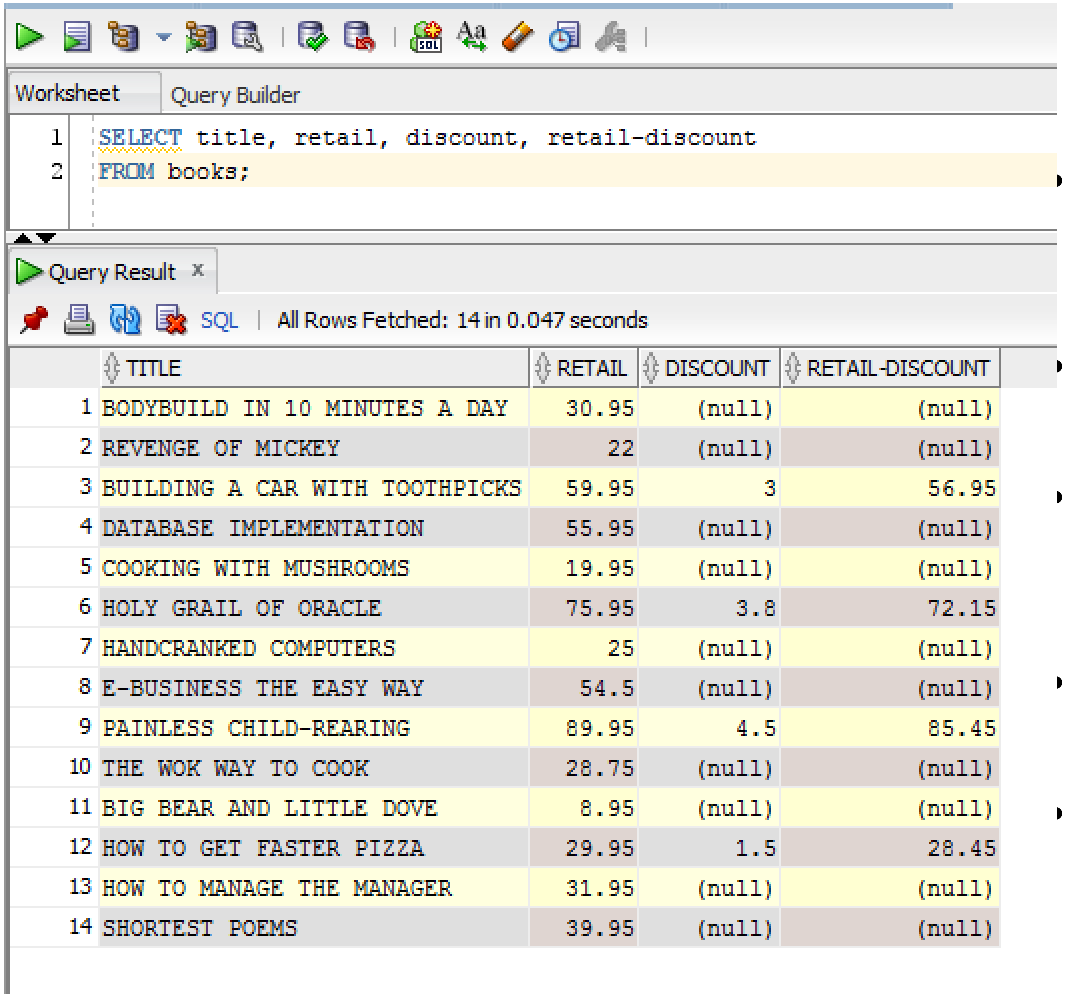


```sql
SELECT title, retail-COST profit
FROM books;
```

	TITLE	PROFIT
1	BODYBUILD IN 10 MINUTES A DAY	12.2
2	REVENGE OF MICKEY	7.8
3	BUILDING A CAR WITH TOOTHPICKS	22.15
4	DATABASE IMPLEMENTATION	24.55
5	COOKING WITH MUSHROOMS	7.45
6	HOLY GRAIL OF ORACLE	28.7
7	HANDCRANKED COMPUTERS	3.2
8	E-BUSINESS THE EASY WAY	16.6
9	PAINLESS CHILD-REARING	41.95
10	THE WOK WAY TO COOK	9.75
11	BIG BEAR AND LITTLE DOVE	3.63
12	HOW TO GET FASTER PIZZA	12.1
13	HOW TO MANAGE THE MANAGER	16.55
14	SHORTEST POEMS	18.1

- Notice the use of the column alias
- The second heading appears as PROFIT
- If the column alias was not used how would the output appear?

Using Arithmetic Operations

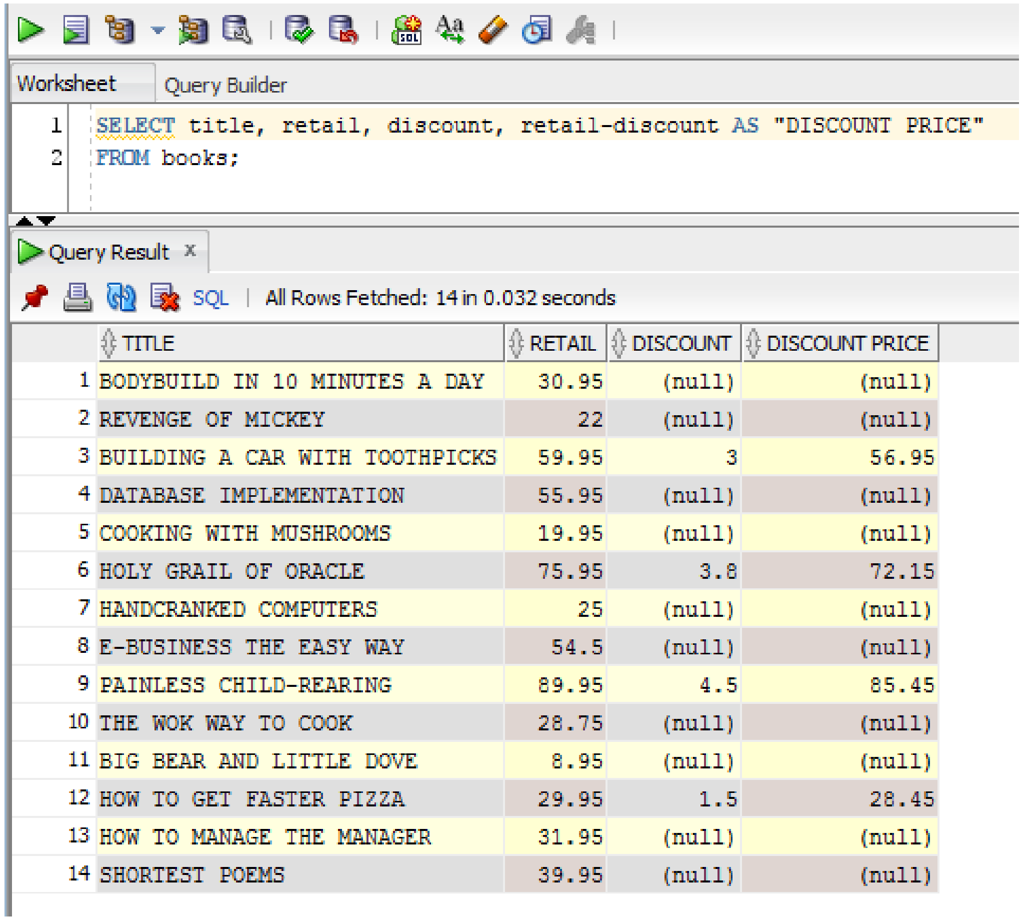


```sql
SELECT title, retail, discount, retail-discount
FROM books;
```

	TITLE	RETAIL	DISCOUNT	RETAIL-DISCOUNT
1	BODYBUILD IN 10 MINUTES A DAY	30.95	(null)	(null)
2	REVENGE OF MICKEY	22	(null)	(null)
3	BUILDING A CAR WITH TOOTHPICKS	59.95	3	56.95
4	DATABASE IMPLEMENTATION	55.95	(null)	(null)
5	COOKING WITH MUSHROOMS	19.95	(null)	(null)
6	HOLY GRAIL OF ORACLE	75.95	3.8	72.15
7	HANDCRANKED COMPUTERS	25	(null)	(null)
8	E-BUSINESS THE EASY WAY	54.5	(null)	(null)
9	PAINLESS CHILD-REARING	89.95	4.5	85.45
10	THE WOK WAY TO COOK	28.75	(null)	(null)
11	BIG BEAR AND LITTLE DOVE	8.95	(null)	(null)
12	HOW TO GET FASTER PIZZA	29.95	1.5	28.45
13	HOW TO MANAGE THE MANAGER	31.95	(null)	(null)
14	SHORTEST POEMS	39.95	(null)	(null)

- The heading shows what appears in the Select CLAUSE, RETAIL-DISCOUNT
- An alias would help to make the result set more understandable
- Which is better to see an alias of DICOUNT PRICE or RETAIL-DISCOUNT
- DISCOUNT PRICE shows a clearer picture of the data in the column
- Alias values are used quite frequently when arithmetic statements are performed

Using Arithmetic Operations



Worksheet Query Builder

```
SELECT title, retail, discount, retail-discount AS "DISCOUNT PRICE"
FROM books;
```

Query Result

SQL | All Rows Fetched: 14 in 0.032 seconds

	TITLE	RETAIL	DISCOUNT	DISCOUNT PRICE
1	BODYBUILD IN 10 MINUTES A DAY	30.95	(null)	(null)
2	REVENGE OF MICKEY	22	(null)	(null)
3	BUILDING A CAR WITH TOOTHPICKS	59.95	3	56.95
4	DATABASE IMPLEMENTATION	55.95	(null)	(null)
5	COOKING WITH MUSHROOMS	19.95	(null)	(null)
6	HOLY GRAIL OF ORACLE	75.95	3.8	72.15
7	HANDCRANKED COMPUTERS	25	(null)	(null)
8	E-BUSINESS THE EASY WAY	54.5	(null)	(null)
9	PAINLESS CHILD-REARING	89.95	4.5	85.45
10	THE WOK WAY TO COOK	28.75	(null)	(null)
11	BIG BEAR AND LITTLE DOVE	8.95	(null)	(null)
12	HOW TO GET FASTER PIZZA	29.95	1.5	28.45
13	HOW TO MANAGE THE MANAGER	31.95	(null)	(null)
14	SHORTEST POEMS	39.95	(null)	(null)

- Alias being used to clarify result in column

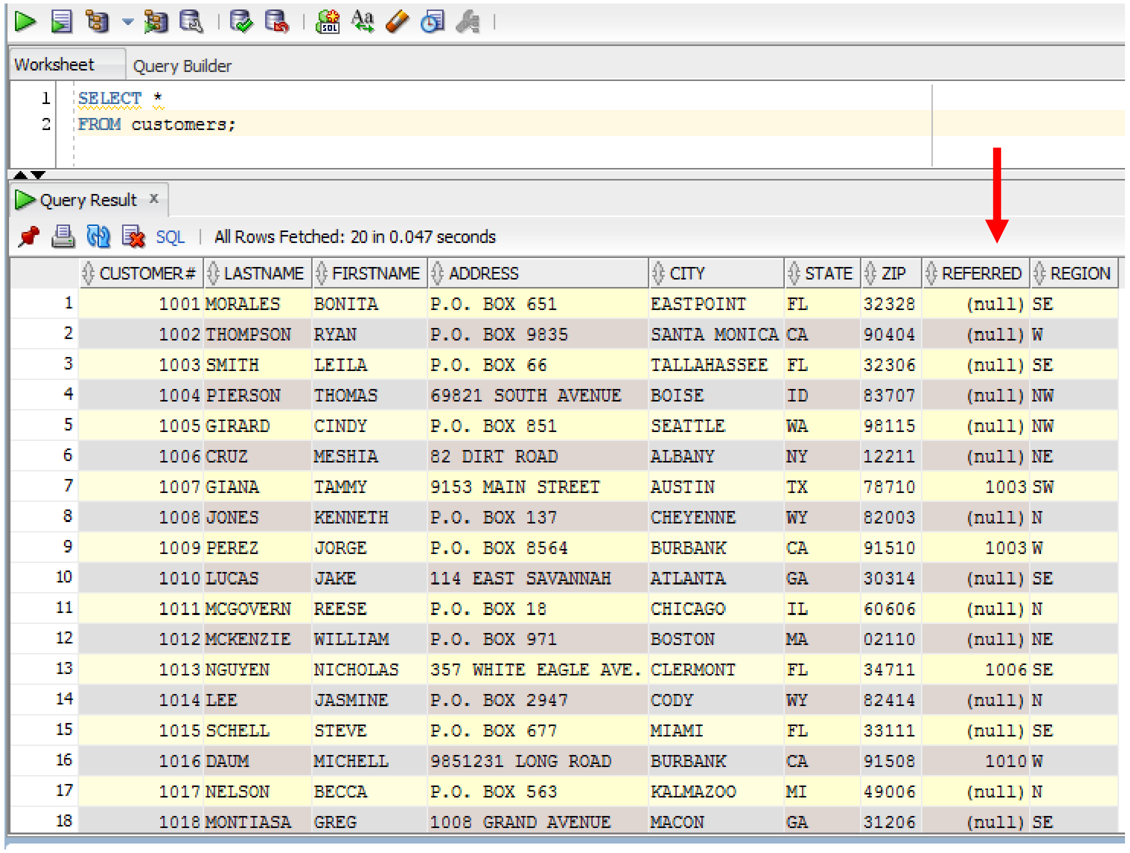
Order of Operations

- Moving from left to right in the arithmetic equation, any required multiplication and division operations are solved first
- Addition and subtraction operations are solved after multiplication and division, again moving form left to right in the equation
- To override this order of operations, you use parentheses to enclose a portion that should be calculated first

NULL Values

- If no value is entered for a column in a row of data the value is considered NULL
- It indicates the absence of data
- In the next slide you will notice in the COMM column the word (null) for any row value that has a NULL value for the COMM

NULL Values



```
SELECT *
FROM customers;
```

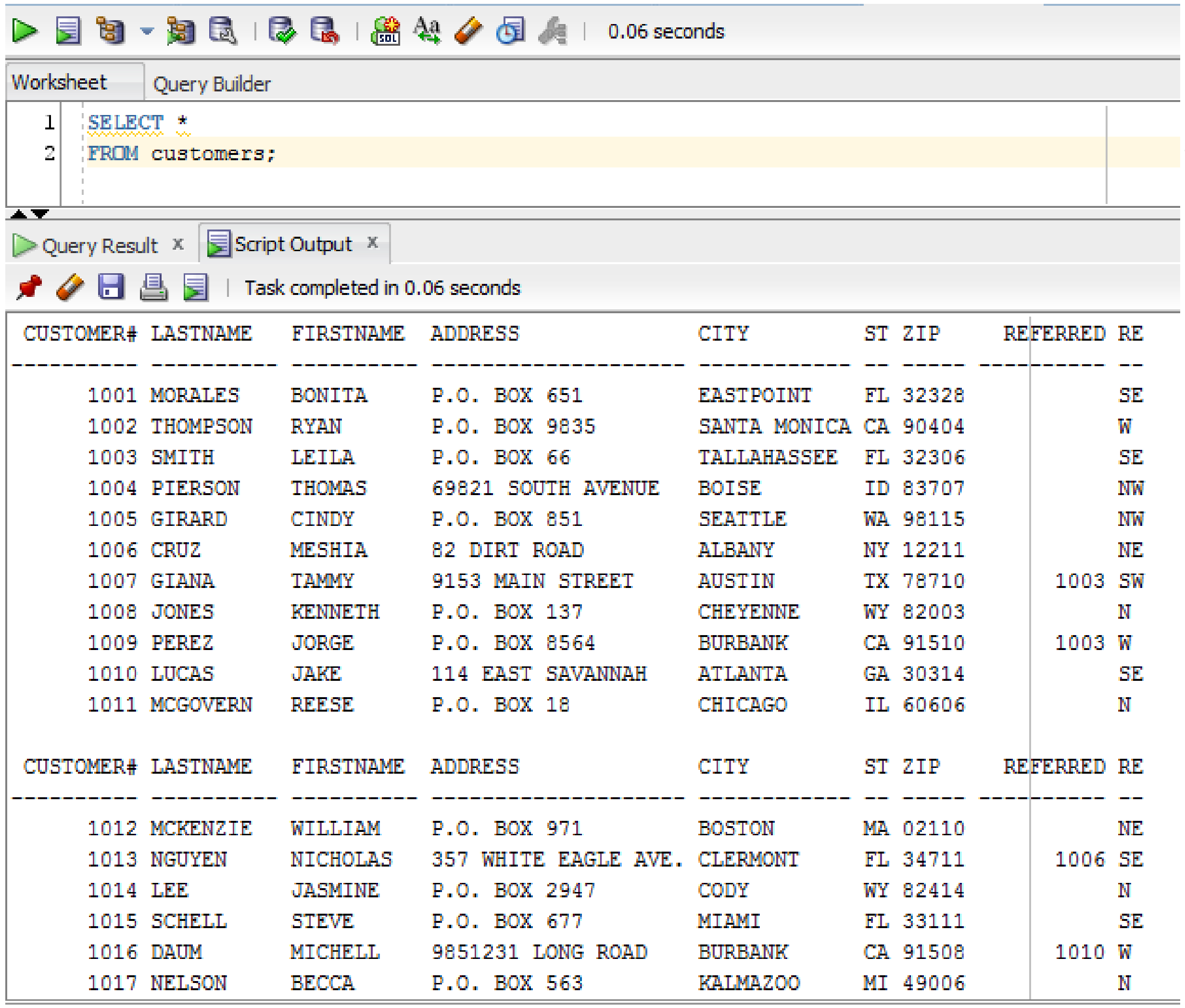
All Rows Fetched: 20 in 0.047 seconds

	CUSTOMER#	LASTNAME	FIRSTNAME	ADDRESS	CITY	STATE	ZIP	REFERRED	REGION
1	1001	MORALES	BONITA	P.O. BOX 651	EASTPOINT	FL	32328	(null)	SE
2	1002	THOMPSON	RYAN	P.O. BOX 9835	SANTA MONICA	CA	90404	(null)	W
3	1003	SMITH	LEILA	P.O. BOX 66	TALLAHASSEE	FL	32306	(null)	SE
4	1004	PIERSON	THOMAS	69821 SOUTH AVENUE	BOISE	ID	83707	(null)	NW
5	1005	GIRARD	CINDY	P.O. BOX 851	SEATTLE	WA	98115	(null)	NW
6	1006	CRUZ	MESHIA	82 DIRT ROAD	ALBANY	NY	12211	(null)	NE
7	1007	GIANA	TAMMY	9153 MAIN STREET	AUSTIN	TX	78710	1003	SW
8	1008	JONES	KENNETH	P.O. BOX 137	CHEYENNE	WY	82003	(null)	N
9	1009	PEREZ	JORGE	P.O. BOX 8564	BURBANK	CA	91510	1003	W
10	1010	LUCAS	JAKE	114 EAST SAVANNAH	ATLANTA	GA	30314	(null)	SE
11	1011	MCGOVERN	REESE	P.O. BOX 18	CHICAGO	IL	60606	(null)	N
12	1012	MCKENZIE	WILLIAM	P.O. BOX 971	BOSTON	MA	02110	(null)	NE
13	1013	NGUYEN	NICHOLAS	357 WHITE EAGLE AVE.	CLERMONT	FL	34711	1006	SE
14	1014	LEE	JASMINE	P.O. BOX 2947	CODY	WY	82414	(null)	N
15	1015	SCHELL	STEVE	P.O. BOX 677	MIAMI	FL	33111	(null)	SE
16	1016	DAUM	MICHELL	9851231 LONG ROAD	BURBANK	CA	91508	1010	W
17	1017	NELSON	BECCA	P.O. BOX 563	KALMAZOO	MI	49006	(null)	N
18	1018	MONTIASA	GREG	1008 GRAND AVENUE	MACON	GA	31206	(null)	SE

Notice the word NULL for any value that is not defined

NULL is not stored in the database, it appears for our use to give a visual indication that the column contains a NULL value

NULL Values



```
SELECT *
FROM customers;
```

```
 CUSTOMER# LASTNAME   FIRSTNAME  ADDRESS              CITY         ST ZIP     REFERRED RE
---------- ---------- ---------- -------------------- ------------ -- ----- ---------- --
      1001 MORALES    BONITA     P.O. BOX 651         EASTPOINT    FL 32328            SE
      1002 THOMPSON   RYAN       P.O. BOX 9835        SANTA MONICA CA 90404            W
      1003 SMITH      LEILA      P.O. BOX 66          TALLAHASSEE  FL 32306            SE
      1004 PIERSON    THOMAS     69821 SOUTH AVENUE   BOISE        ID 83707            NW
      1005 GIRARD     CINDY      P.O. BOX 851         SEATTLE      WA 98115            NW
      1006 CRUZ       MESHIA     82 DIRT ROAD         ALBANY       NY 12211            NE
      1007 GIANA      TAMMY      9153 MAIN STREET     AUSTIN       TX 78710       1003 SW
      1008 JONES      KENNETH    P.O. BOX 137         CHEYENNE     WY 82003            N
      1009 PEREZ      JORGE      P.O. BOX 8564        BURBANK      CA 91510       1003 W
      1010 LUCAS      JAKE       114 EAST SAVANNAH    ATLANTA      GA 30314            SE
      1011 MCGOVERN   REESE      P.O. BOX 18          CHICAGO      IL 60606            N

 CUSTOMER# LASTNAME   FIRSTNAME  ADDRESS              CITY         ST ZIP     REFERRED RE
---------- ---------- ---------- -------------------- ------------ -- ----- ---------- --
      1012 MCKENZIE   WILLIAM    P.O. BOX 971         BOSTON       MA 02110            NE
      1013 NGUYEN     NICHOLAS   357 WHITE EAGLE AVE. CLERMONT     FL 34711       1006 SE
      1014 LEE        JASMINE    P.O. BOX 2947        CODY         WY 82414            N
      1015 SCHELL     STEVE      P.O. BOX 677         MIAMI        FL 33111            SE
      1016 DAUM       MICHELL    9851231 LONG ROAD    BURBANK      CA 91508       1010 W
      1017 NELSON     BECCA      P.O. BOX 563         KALMAZOO     MI 49006            N
```

- If I use the alternate method to execute a query, notice the NULL values appear with no visible value or any indication the column is NULL

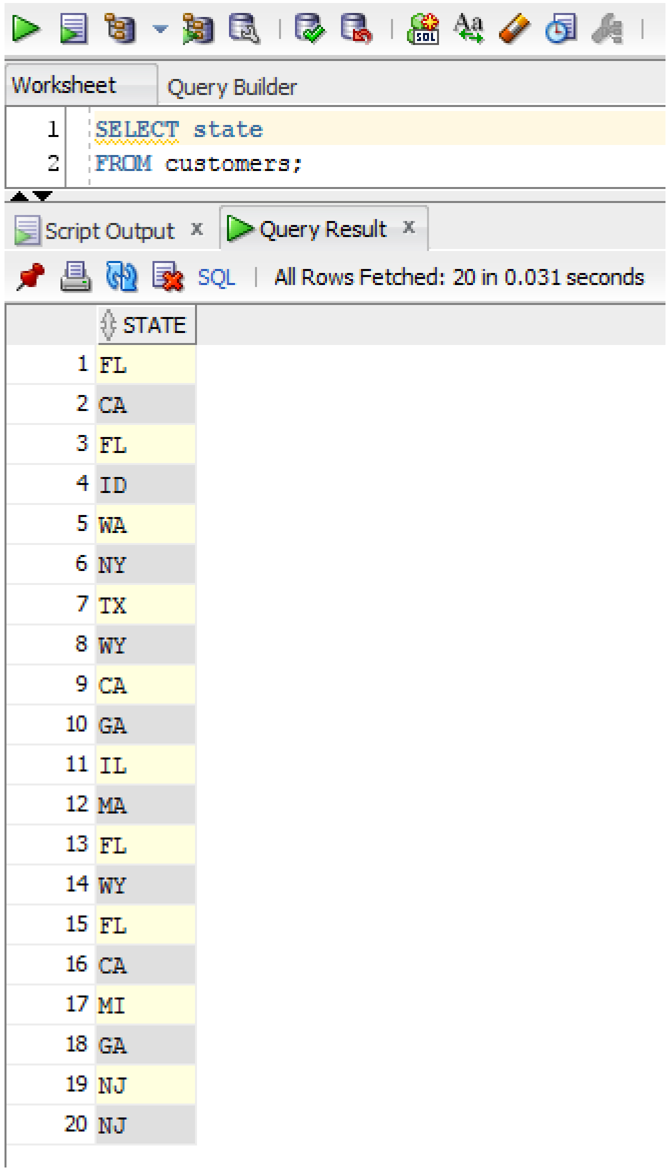
NULL Values

- NULL values can lead to undesirable results in operations
- For example, what if you need to an employee's total salary, as we saw this was difficult to determine
- Seems simple enough, but check out the result for every row where the REFERRED value was NULL
- This can be fixed, we shall see later on a function can be used to allow you to substitute a value for a NULL

Using DISTINCT and UNIQUE

- Suppose you want to know which are the various jobs the employees can have
- You want to list only the jobs, not the employee information. This could be done by listing only the JOB column of the EMPLOYEES table
- This is done on the next slide. Do you notice a problem?

Using DISTINCT and UNIQUE



Worksheet | Query Builder

```
SELECT state
FROM customers;
```

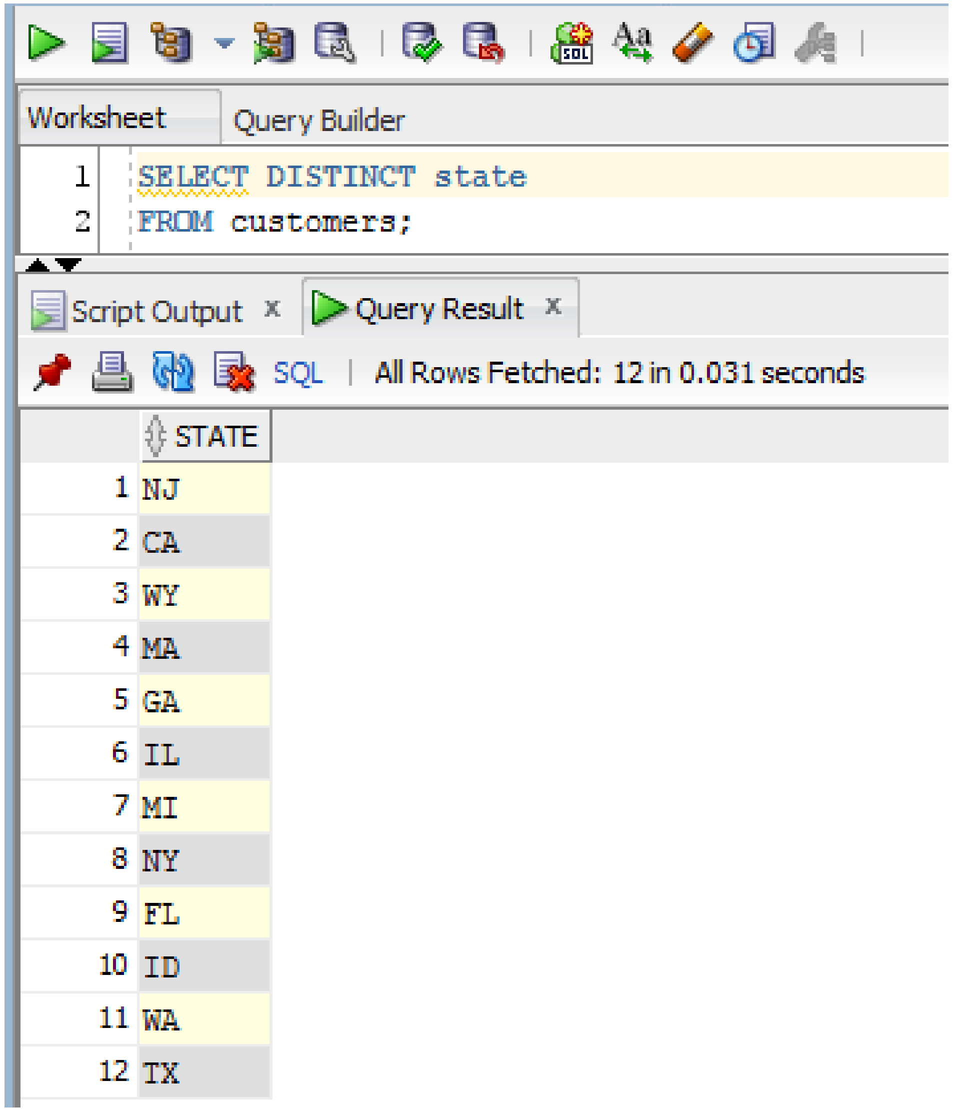
Script Output | Query Result

SQL | All Rows Fetched: 20 in 0.031 seconds

	STATE
1	FL
2	CA
3	FL
4	ID
5	WA
6	NY
7	TX
8	WY
9	CA
10	GA
11	IL
12	MA
13	FL
14	WY
15	FL
16	CA
17	MI
18	GA
19	NJ
20	NJ

- The output shows one row of output for each row in the table
- 20 rows shows 20 state values, one for each row
- If the question was asked, show the jobs that my employees may have, would I want to see the value for each employee or would I only need to see individual state values?
- I am glad I only have 14 customers showing and not 50,000 customers
- How can I modify the query to show each state only once?

Using DISTINCT and UNIQUE



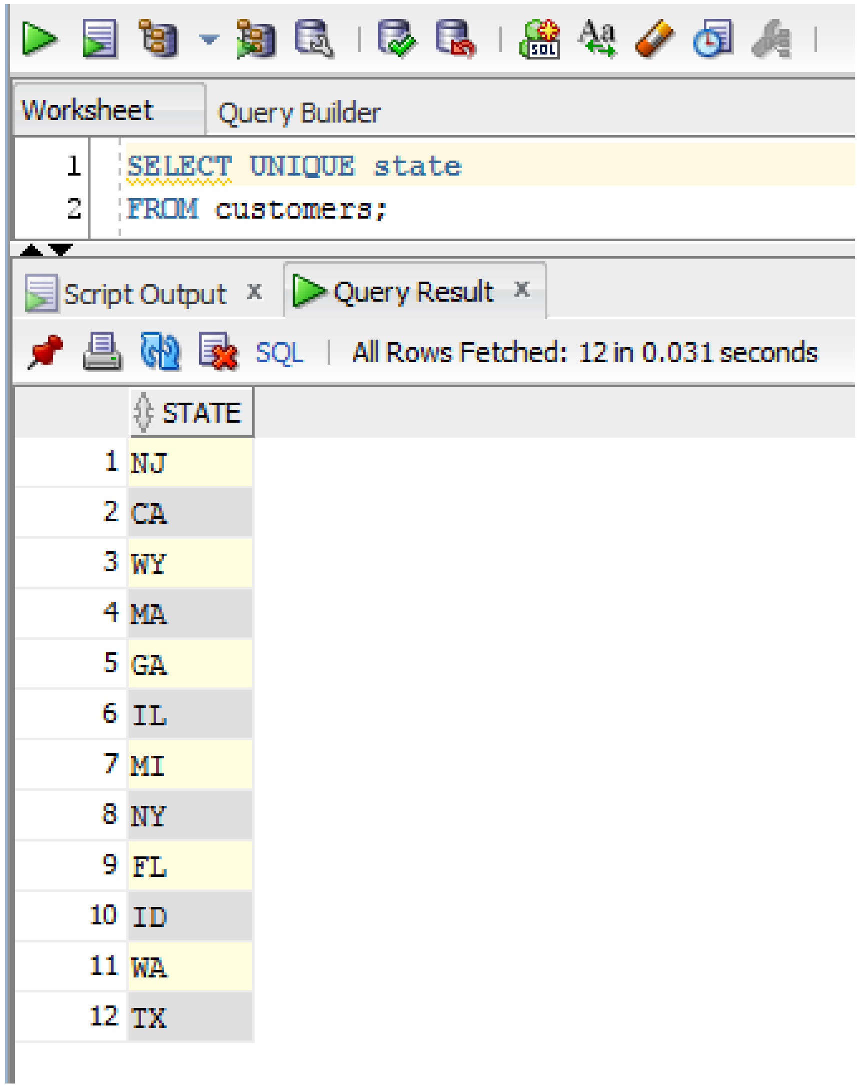
```
SELECT DISTINCT state
FROM customers;
```

All Rows Fetched: 12 in 0.031 seconds

	STATE
1	NJ
2	CA
3	WY
4	MA
5	GA
6	IL
7	MI
8	NY
9	FL
10	ID
11	WA
12	TX

- In this output I only see an individual row for each state
- There are no duplicate values displayed for the state column
- Even if I had 50,000 customers in the table, and all customers were in one of the 12 different states showing then I would only see 12 rows of output still
- Any duplicate values are suppressed

Using DISTINCT and UNIQUE



```
SELECT UNIQUE state
FROM customers;
```

All Rows Fetched: 12 in 0.031 seconds

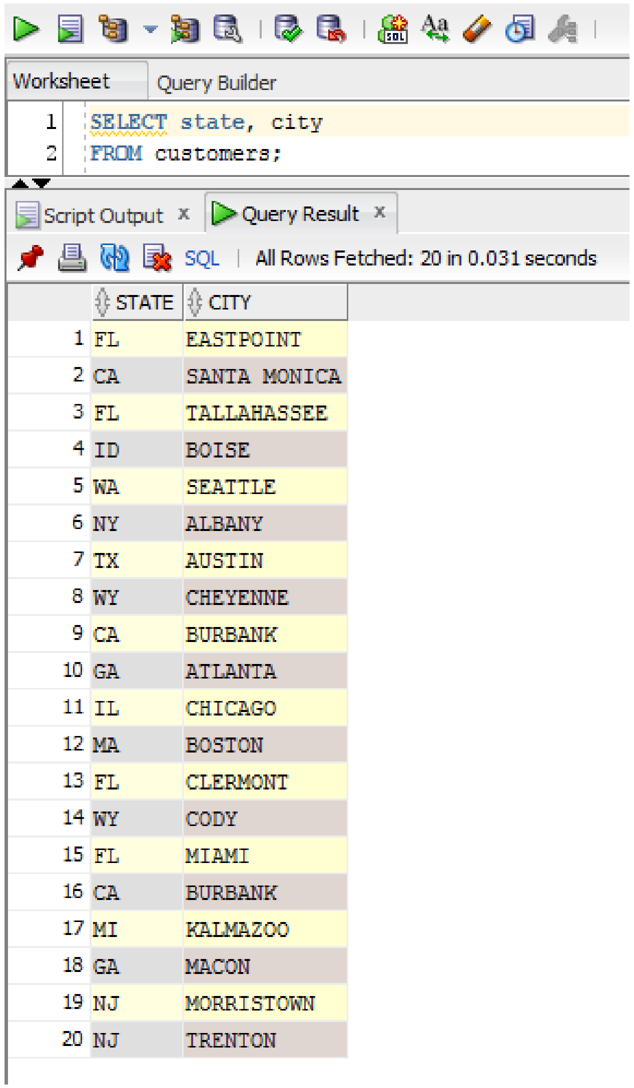
	STATE
1	NJ
2	CA
3	WY
4	MA
5	GA
6	IL
7	MI
8	NY
9	FL
10	ID
11	WA
12	TX

- The UNIQUE keyword displays the same output as the DISTINCT keyword
- Both perform the same function and are interchangeable

Using DISTINCT and UNIQUE

- The DISTINCT keyword is applied to **all** columns listed in the SELECT statement
- So, if multiple columns were used in the SELECT, **each** possible result would be tested for uniqueness. Only the ones that are unique for **all** columns will be returned

Using DISTINCT



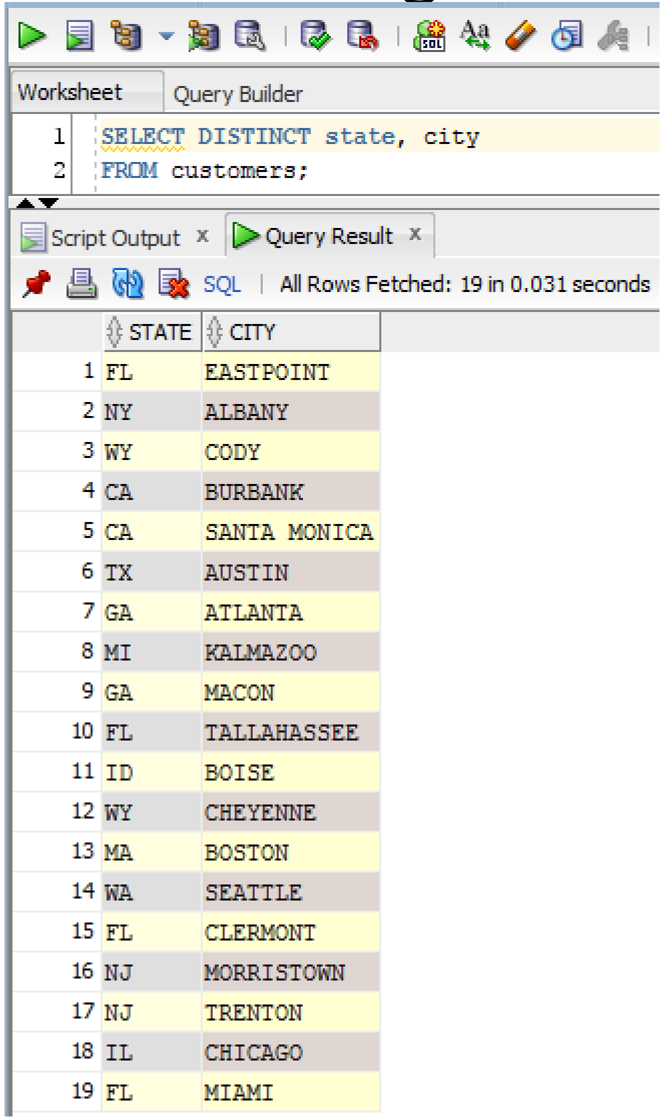
```
SELECT state, city
FROM customers;
```

All Rows Fetched: 20 in 0.031 seconds

	STATE	CITY
1	FL	EASTPOINT
2	CA	SANTA MONICA
3	FL	TALLAHASSEE
4	ID	BOISE
5	WA	SEATTLE
6	NY	ALBANY
7	TX	AUSTIN
8	WY	CHEYENNE
9	CA	BURBANK
10	GA	ATLANTA
11	IL	CHICAGO
12	MA	BOSTON
13	FL	CLERMONT
14	WY	CODY
15	FL	MIAMI
16	CA	BURBANK
17	MI	KALMAZOO
18	GA	MACON
19	NJ	MORRISTOWN
20	NJ	TRENTON

Here I am showing the STATE and CITY values, there is one row for every CUSTOMER

Using DISTINCT and UNIQUE



```
SELECT DISTINCT state, city
FROM customers;
```

All Rows Fetched: 19 in 0.031 seconds

	STATE	CITY
1	FL	EASTPOINT
2	NY	ALBANY
3	WY	CODY
4	CA	BURBANK
5	CA	SANTA MONICA
6	TX	AUSTIN
7	GA	ATLANTA
8	MI	KALMAZOO
9	GA	MACON
10	FL	TALLAHASSEE
11	ID	BOISE
12	WY	CHEYENNE
13	MA	BOSTON
14	WA	SEATTLE
15	FL	CLERMONT
16	NJ	MORRISTOWN
17	NJ	TRENTON
18	IL	CHICAGO
19	FL	MIAMI

- Instead of the 20rows for all customers or the 12 rows for the 12 different states shown before or the unique results of CITY and STATE combinations
- This tells me that two of my customers have the same STATE and CITY
- I do not know which ones are the same, only that two must be the same

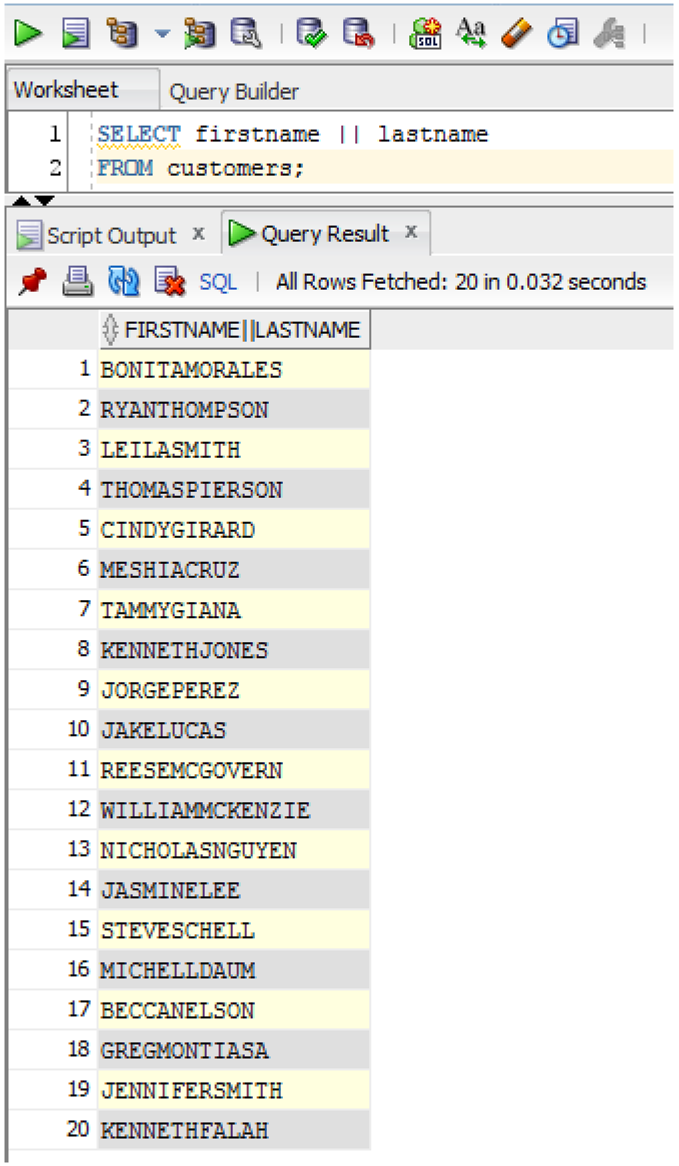
DISTINCT versus UNIQUE

- I was asked, “what is the difference between DISTINCT and UNIQUE?”
- UNIQUE was Oracle’s traditional method for removing duplicate values, and only Oracle uses this method
- DISTINCT was added when Oracle adopted the ANSI specifications
- So DISTINCT will work in other SQL versions, where UNIQUE will only function with Oracle

Concatenation

- So far, each field has been placed in a column of its own
- In certain situations, you might want to display the contents of each field so they appear right next to each other, separated by a single space or comma
- Combining columns is called concatenation
- The concatenation operator in Oracle is two vertical bars beside one another ||

Concatenation

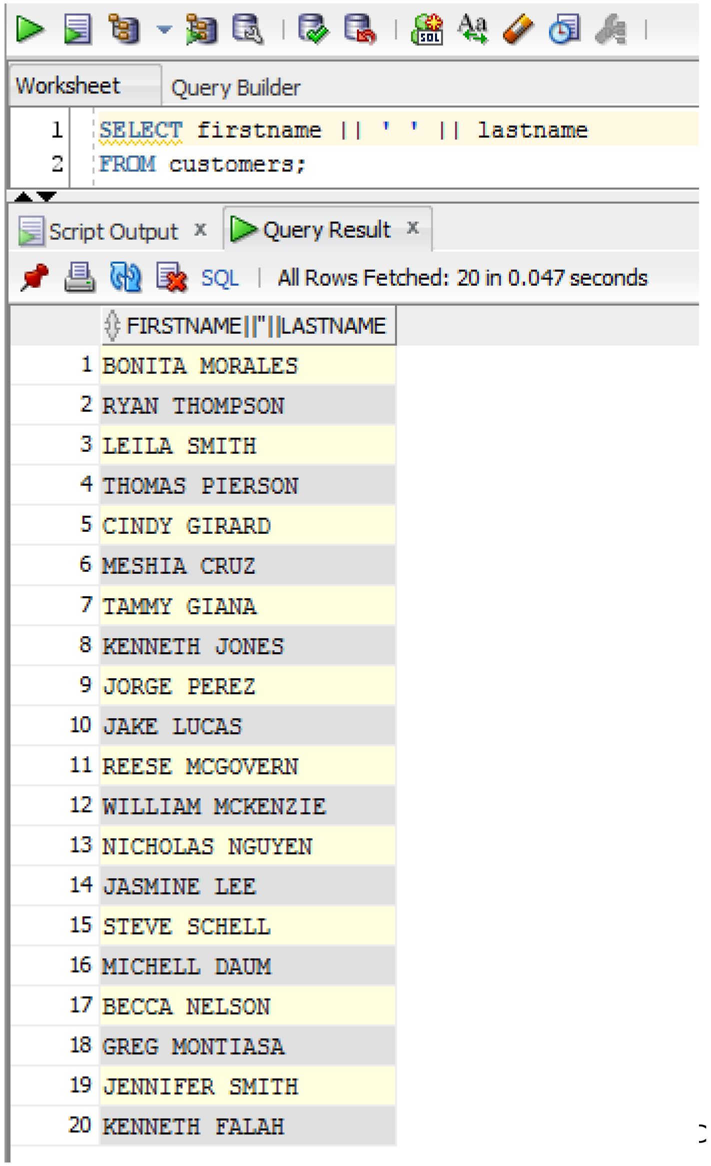


```
SELECT firstname || lastname
FROM customers;
```

	FIRSTNAME||LASTNAME
1	BONITAMORALES
2	RYANTHOMPSON
3	LEILASMITH
4	THOMASPIERSON
5	CINDYGIRARD
6	MESHIACRUZ
7	TAMMYGIANA
8	KENNETHJONES
9	JORGEPEREZ
10	JAKELUCAS
11	REESEMCGOVERN
12	WILLIAMMCKENZIE
13	NICHOLASNGUYEN
14	JASMINELEE
15	STEVESCHELL
16	MICHELLDAUM
17	BECCANELSON
18	GREGMONTIASA
19	JENNIFERSMITH
20	KENNETHFALAH

- This did as I requested, it concatenated the first name and the last name together
- It is not easy to read, can it be improved?
- There should be a space between the two values
- Time to try again to improve the readability of our result

Concatenation



```
SELECT firstname || ' ' || lastname
FROM customers;
```

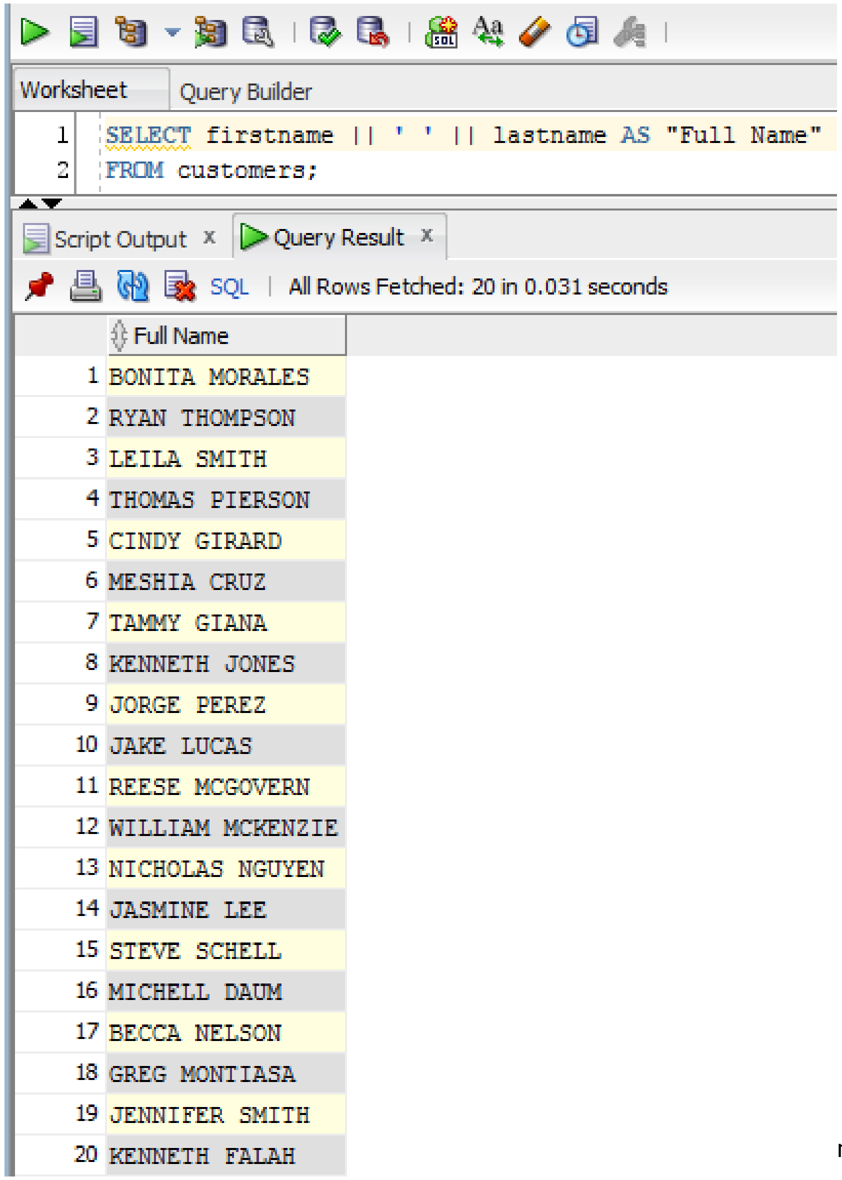
```
   FIRSTNAME||''||LASTNAME
 1 BONITA MORALES
 2 RYAN THOMPSON
 3 LEILA SMITH
 4 THOMAS PIERSON
 5 CINDY GIRARD
 6 MESHIA CRUZ
 7 TAMMY GIANA
 8 KENNETH JONES
 9 JORGE PEREZ
10 JAKE LUCAS
11 REESE MCGOVERN
12 WILLIAM MCKENZIE
13 NICHOLAS NGUYEN
14 JASMINE LEE
15 STEVE SCHELL
16 MICHELL DAUM
17 BECCA NELSON
18 GREG MONTIASA
19 JENNIFER SMITH
20 KENNETH FALAH
```

- This looks better, I have the space between two values
- Could it still be improved?
- What about the headings?
- How can I fix the heading?

Concatenation

- The previous slide took two concatenation operations
- In the previous slide, a string literal was inserted into the output to put a blank space between the two fields
- All string literals are enclosed in single quotation marks, the single quotes have a space inside so I end up with a space in my result
- On the previous slide, the lastname is concatenated to the blank space, then the blank space is concatenated to the firstname

Concatenation



Worksheet Query Builder

```
SELECT firstname || ' ' || lastname AS "Full Name"
FROM customers;
```

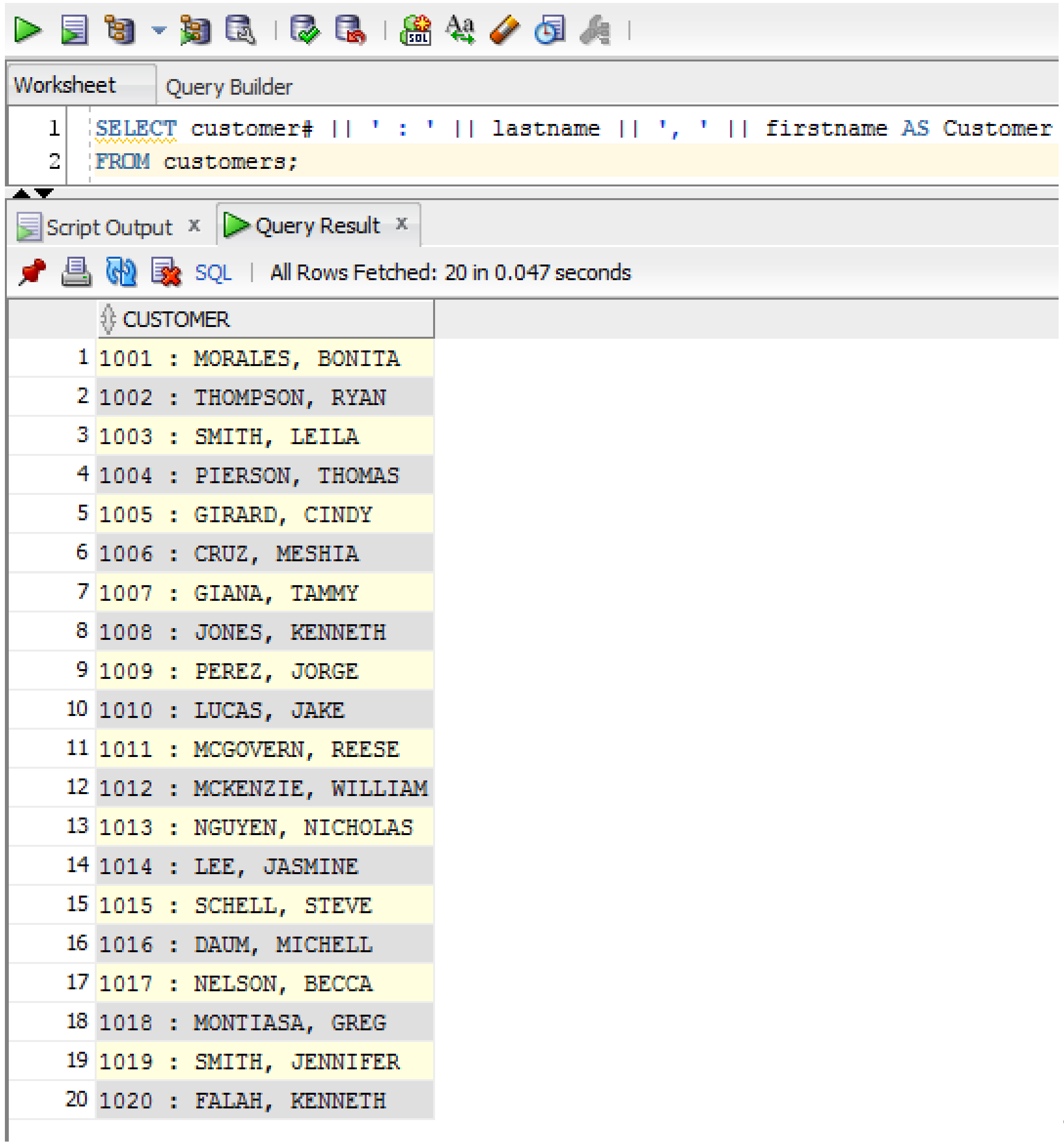
Script Output Query Result

SQL | All Rows Fetched: 20 in 0.031 seconds

	Full Name
1	BONITA MORALES
2	RYAN THOMPSON
3	LEILA SMITH
4	THOMAS PIERSON
5	CINDY GIRARD
6	MESHIA CRUZ
7	TAMMY GIANA
8	KENNETH JONES
9	JORGE PEREZ
10	JAKE LUCAS
11	REESE MCGOVERN
12	WILLIAM MCKENZIE
13	NICHOLAS NGUYEN
14	JASMINE LEE
15	STEVE SCHELL
16	MICHELL DAUM
17	BECCA NELSON
18	GREG MONTIASA
19	JENNIFER SMITH
20	KENNETH FALAH

- Here an alias is used to make the output more readable
- It is always advisable to use column aliases to replace headings for expressions or concatenated values that could appear in the headings when concatenation is used

Concatenation



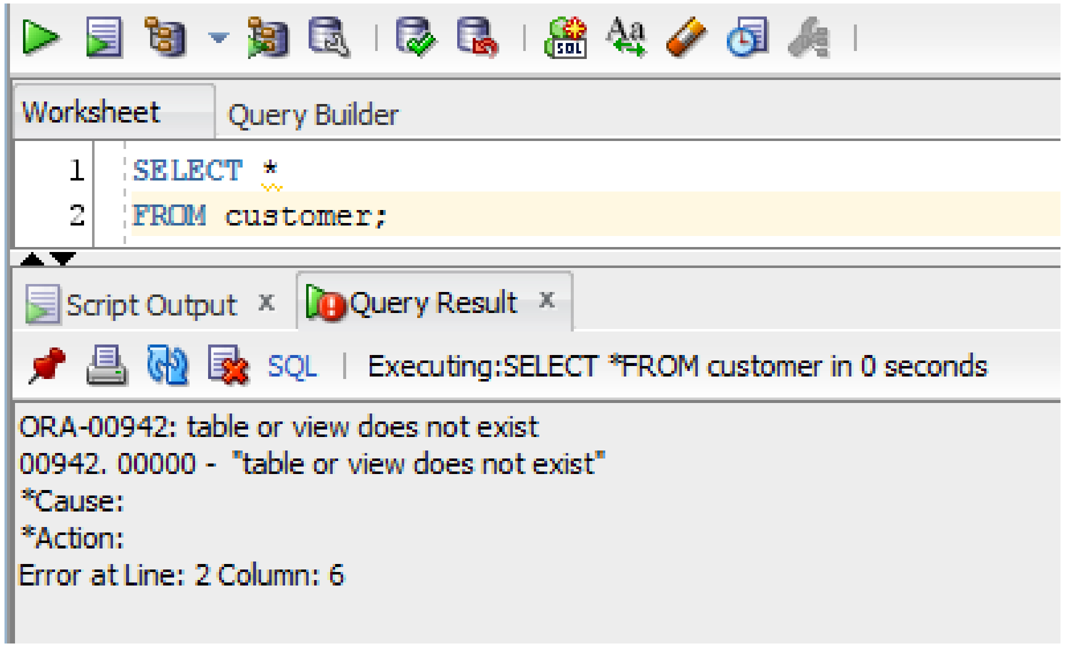
```
SELECT customer# || ' : ' || lastname || ', ' || firstname AS Customer
FROM customers;
```

All Rows Fetched: 20 in 0.047 seconds

```
    CUSTOMER
 1  1001 : MORALES, BONITA
 2  1002 : THOMPSON, RYAN
 3  1003 : SMITH, LEILA
 4  1004 : PIERSON, THOMAS
 5  1005 : GIRARD, CINDY
 6  1006 : CRUZ, MESHIA
 7  1007 : GIANA, TAMMY
 8  1008 : JONES, KENNETH
 9  1009 : PEREZ, JORGE
10  1010 : LUCAS, JAKE
11  1011 : MCGOVERN, REESE
12  1012 : MCKENZIE, WILLIAM
13  1013 : NGUYEN, NICHOLAS
14  1014 : LEE, JASMINE
15  1015 : SCHELL, STEVE
16  1016 : DAUM, MICHELL
17  1017 : NELSON, BECCA
18  1018 : MONTIASA, GREG
19  1019 : SMITH, JENNIFER
20  1020 : FALAH, KENNETH
```

- This is another variation where string literals have been placed into the result

Syntax Errors



```
SELECT *
FROM customer;
```

```
Executing:SELECT *FROM customer in 0 seconds
```

```
ORA-00942: table or view does not exist
00942. 00000 -  "table or view does not exist"
*Cause:
*Action:
Error at Line: 2 Column: 6
```

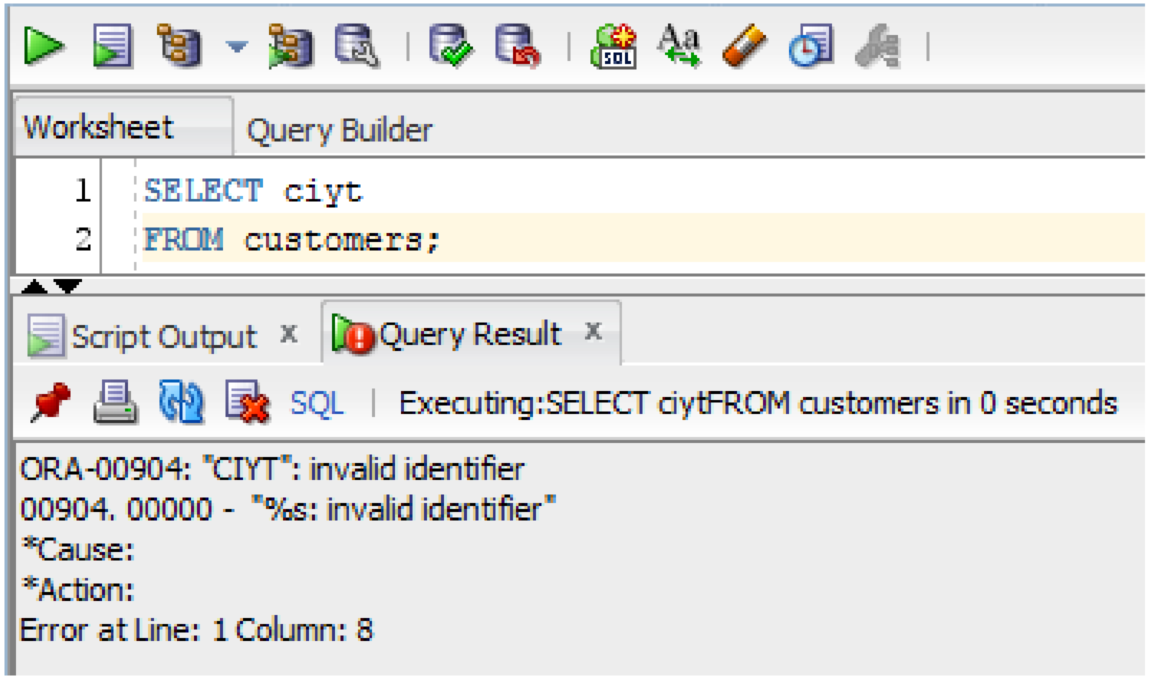
The query to the right produces an error ORA-00942 table or view does not exist

Table name in this case is incorrect it is CUSTOMERS not CUSTOMER

Correct and the re-execute the command

Check your tree on the left to see the table names

Syntax Errors



```
SELECT ciyt
FROM customers;
```

```
Executing:SELECT ciytFROM customers in 0 seconds
ORA-00904: "CIYT": invalid identifier
00904. 00000 -  "%s: invalid identifier"
*Cause:
*Action:
Error at Line: 1 Column: 8
```

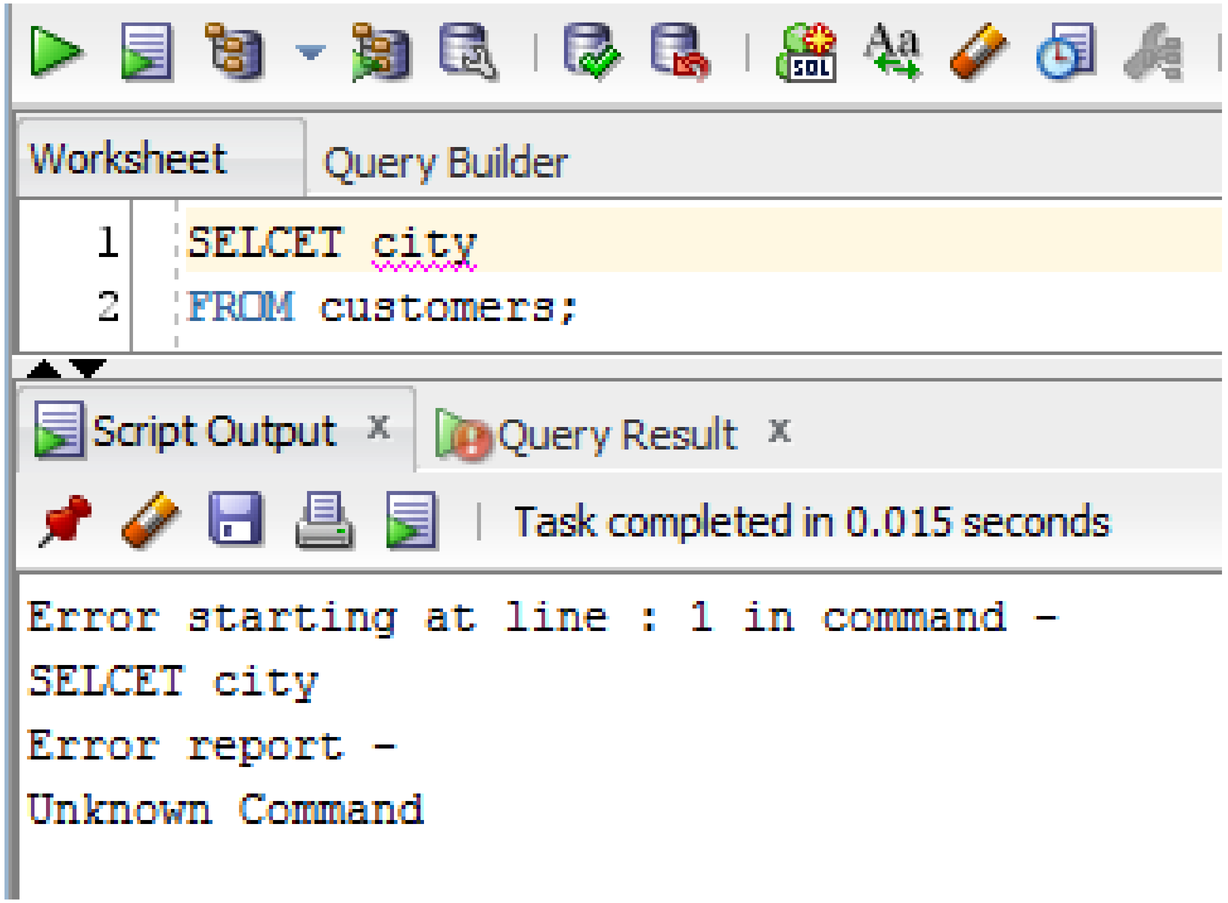
The query to the right has produced an error ORA-00904: CIYTinvalid identifier

Column name is misspelled, should be CITY

Again check the tree on the left to see the column names in the table

Could also do a DESCRIBE customers; to see correct columns

Syntax Errors



```
SELCET city
FROM customers;
```

```
Error starting at line : 1 in command -
SELCET city
Error report -
Unknown Command
```

The keyword SELECT is misspelled

Gives error as Unknown Command

Notice the keyword did turn BLUE indicating the error, you will notice all keywords will display in BLUE to indicate they are keywords

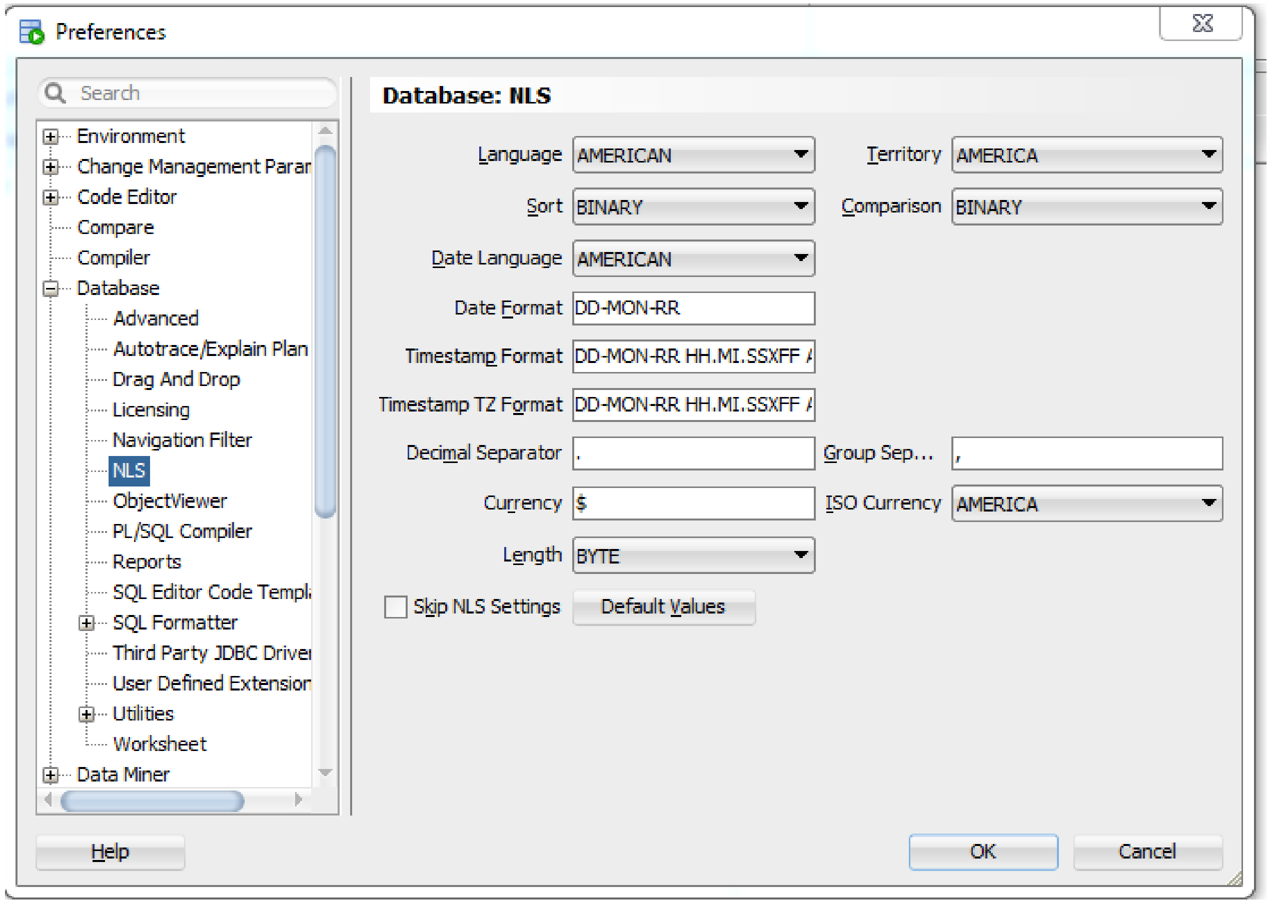
Fix and re-execute

Notice the red squiggle after the misspelled keyword

Settings

- It is important to make sure your date values are set correctly
- The default date format Oracle uses is DD-MON-RR
- When you install SQL Developer it looks at the Windows settings and sets the date format according to country
- ENGLISH CANADA is a different value
- Go to the Tools menu, select PREFERENCES, expand DATABASE then select NLS
- This is the important setting to check, please make sure otherwise you may get errors in your outputs involving date values

Settings



Database: NLS

Language	AMERICAN	Territory	AMERICA
Sort	BINARY	Comparison	BINARY
Date Language	AMERICAN		
Date Format	DD-MON-RR		
Timestamp Format	DD-MON-RR HH.MI.SSXFF A		
Timestamp TZ Format	DD-MON-RR HH.MI.SSXFF A		
Decimal Separator	.	Group Sep...	,
Currency	$	ISO Currency	AMERICA
Length	BYTE		

☐ Skip NLS Settings

Set the following:
Date Format
Timestamp Format
Timestamp TZ Format

They should all have DD-MON-RR

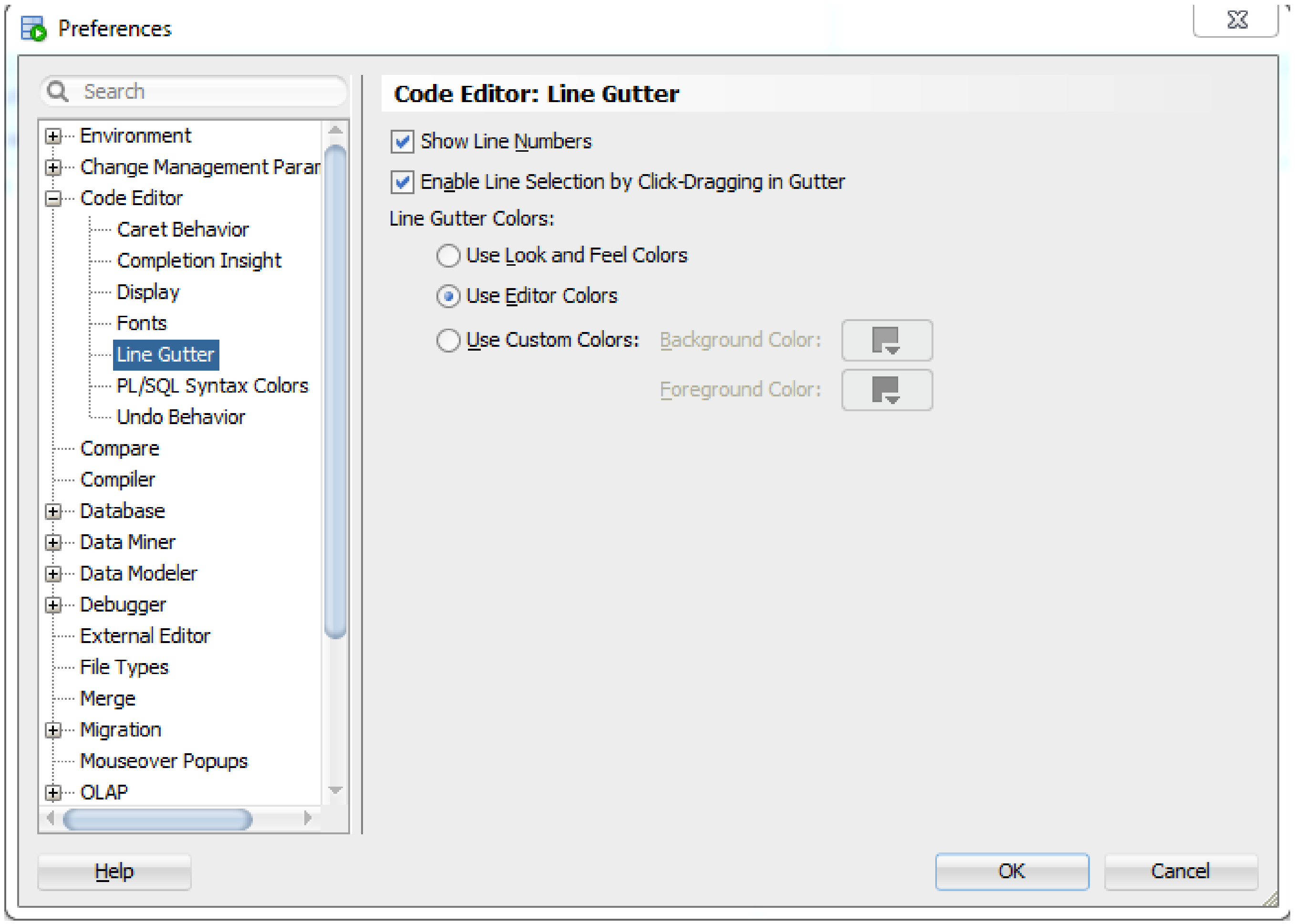
If yours is different please modify then click OK

On the Timestamps you only need to modify the left most values

Line Numbers

- While you are in the Preferences click on the Code Editor option
- Then select Line Gutter
- Check the Show Line Numbers check box
- This will then show line numbers in the editor

Line Numbers


Preferences

Search

Code Editor: Line Gutter

[✓] Show Line Numbers

[✓] Enable Line Selection by Click-Dragging in Gutter

Line Gutter Colors:

○ Use Look and Feel Colors

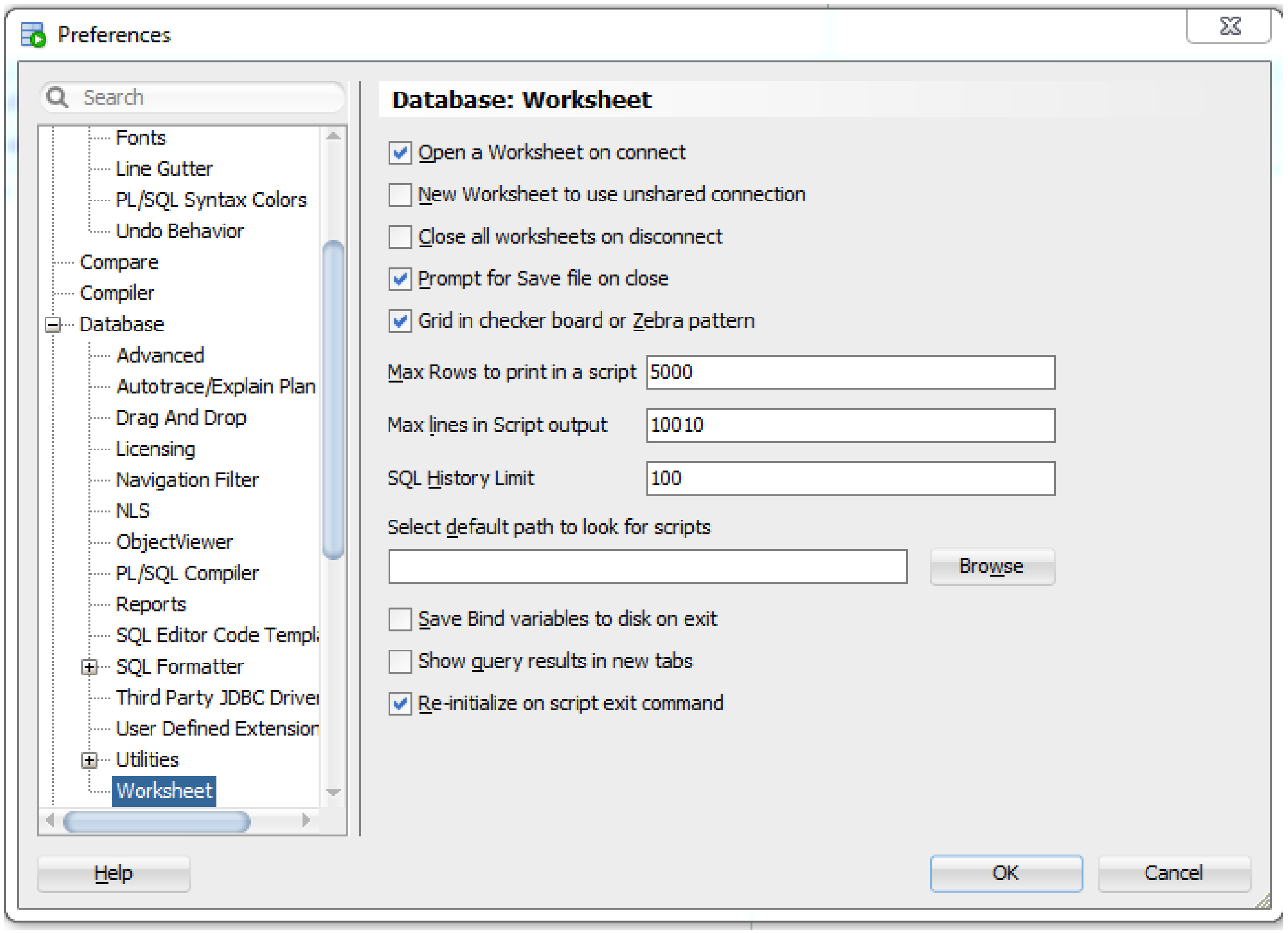
[✓ ● Use Editor Colors]

○ Use Custom Colors: Background Color: Foreground Color:

Zebra Pattern Print Output

- You may have noticed that the output in my queries used alternate colours for each line
- It alternated between a light grey and a cream colour
- If this is not done the output appears in one colour, white, still in a grid
- This method just makes it easier to read the output
- Again in the Preferences menu, select Database, scroll down till you see Worksheet, click the option Grid in checker board or Zebra pattern

Zebra Pattern Print Output



Database: Worksheet

- [✓] Open a Worksheet on connect
- [ ] New Worksheet to use unshared connection
- [ ] Close all worksheets on disconnect
- [✓] Prompt for Save file on close
- [✓] Grid in checker board or Zebra pattern

Max Rows to print in a script	5000
Max lines in Script output	10010
SQL History Limit	100

Select default path to look for scripts

- [ ] Save Bind variables to disk on exit
- [ ] Show query results in new tabs
- [✓] Re-initialize on script exit command

Summary

- Identify keywords, mandatory clauses, and optional clauses in a SELECT statement
- Select and view all, one or multiple columns of a table
- Use a column alias to clarify contents of a column
- Perform basic arithmetic operations in a SELECT clause
- Remove duplicate items using either DISTINCT or UNIQUE keywords
- Some basic settings for SQL Developer