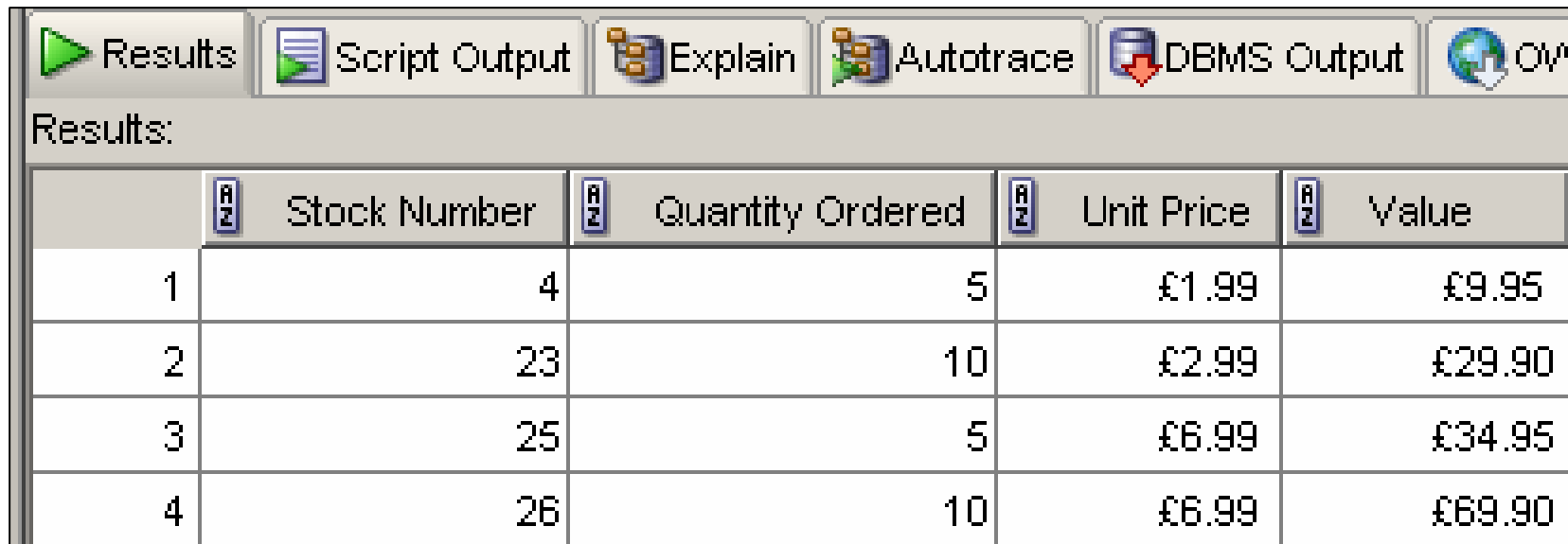


## Exercise Six



6.1 Using the SQL in exone4\_1.sql, limit the records to only those where the quantity ordered is between 5 and 10. Order by od\_stock\_number. Save as c:\course\exone6\_1.sql

A screenshot of the SQL Developer Results window. The window has a toolbar with buttons for 'Results', 'Script Output', 'Explain', 'Autotrace', 'DBMS Output', and 'OW'. Below the toolbar, the word 'Results:' is displayed. A table with four columns is shown: 'Stock Number', 'Quantity Ordered', 'Unit Price', and 'Value'. The table contains four rows of data.

	Stock Number	Quantity Ordered	Unit Price	Value
1	4	5	£1.99	£9.95
2	23	10	£2.99	£29.90
3	25	5	£6.99	£34.95
4	26	10	£6.99	£69.90



## Exercise Six

6.2 Using the previous SQL, include the od\_order\_date in the output, limit the selection to only those orders placed between the 10<sup>th</sup> and 15<sup>th</sup> January 2001. Order by Date and within that quantity ordered. Save as c:\course\exone6\_2.sql

	A Z	Date Ordered	A Z	Stock Number	A Z	Quantity Ordered	A Z	Unit Price	A Z	Value
1		12-JAN-01		3		1		£15.99		£15.99
2		12-JAN-01		15		1		£14.99		£14.99
3		12-JAN-01		4		5		£1.99		£9.95
4		13-JAN-01		17		1		£8.99		£8.99



## Exercise Six

6.3 Open the c:\course\exone4\_2.sql file. Amend this to retrieve only those sales which took place on a Monday. Hint: A wildcard is better than an equals comparison.

Save as c:\course\exone6\_3.sql.

Results:						
	Day Ordered	Stock Number	Quantity Ordered	Unit Price	Value	
1	Monday	15	3	£14.99	£44.97	
2	Monday	14	1	£79.99	£79.99	
3	Monday	23	10	£2.99	£29.90	
4	Monday	21	1	£4.70	£4.70	

## Exercise Six



6.4 Using the Store\_Locations table, Select sl\_location\_id and sl\_location\_name where the location name is either Dover, Hyde or Penrith. Save as c:\course\exone6\_4.sql

Results:

	A Z	Location Id	A Z	Location Name
1		91		DOVER
2		96		HYDE
3		104		PENRITH

## Exercise Six



6.5 Amend exone6\_4.sql to display only those records where the sl\_parent\_location\_id does not exist.  
Save as c:\course\exone6\_5.sql

The screenshot shows the SQL Developer interface with a toolbar at the top containing buttons for 'Results', 'Script Output', 'Explain', and 'Autotrace'. Below the toolbar, the 'Results' tab is active, displaying a table with two columns: 'Location Id' and 'Location Name'. The table contains one row of data.

	Location Id	Location Name
1	1	WEST NORWOOD



In this section the User will learn...

- How to perform more complex aggregate functions on a table



## Exercise Seven

7.1 Using a tree walk, display the sl\_location\_id, sl\_location\_address, sl\_location\_type from the Store\_Locations table, starting at location 110 traverse upwards to find all of its parents ...

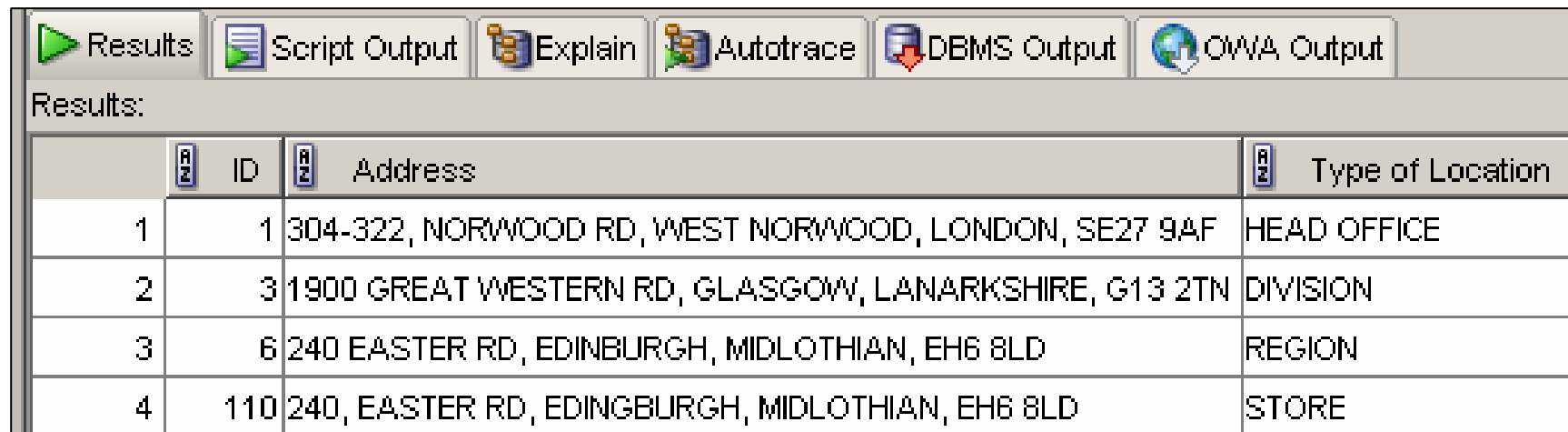
Save as c:\course\extwo7\_1.sql

	ID	Address	Type
1	110	240, EASTER RD, EDINGBURGH, MIDLOTHIAN, EH6 8LD	S
2	6	240 EASTER RD, EDINBURGH, MIDLOTHIAN, EH6 8LD	R
3	3	1900 GREAT WESTERN RD, GLASGOW, LANARKSHIRE, G13 2TN	D
4	1	304-322, NORWOOD RD, WEST NORWOOD, LONDON, SE27 9AF	H






## Exercise Seven

7.2 Amend the previous SQL statement to include the Location\_Type table to display the Lt\_Location\_Description column, change the order of the records to display the Head Office, Division, Region and Store in that order ...



Results: ▶ Results 📄 Script Output 🗃 Explain 📊 Autotrace 📄 DBMS Output 🌐 OWA Output

	 ID	 Address	 Type of Location
1	1	304-322, NORWOOD RD, WEST NORWOOD, LONDON, SE27 9AF	HEAD OFFICE
2	3	1900 GREAT WESTERN RD, GLASGOW, LANARKSHIRE, G13 2TN	DIVISION
3	6	240 EASTER RD, EDINBURGH, MIDLOTHIAN, EH6 8LD	REGION
4	110	240, EASTER RD, EDINGBURGH, MIDLOTHIAN, EH6 8LD	STORE



## Exercise Seven

7.3 Create a new Tree Walk on the Store\_Locations table, start at the first record (ie where the sl\_parent\_location\_id is null) and traverse down all records using lpad and level to indent the displayed location id to show its position relative to other records. Use Order Siblings.

Save as c:\course\extwo7\_3.sql

The screenshot shows the SQL Developer Results window with a tree walk of the Store\_Locations table. The results are displayed in a table with columns: Name, Address, and Type of Location. The records are indented based on their parent-child relationships.

	Name	Address	Type of Location
1	1 WEST NORWOOD	304-322, NORWOOD RD, WEST NORWOOD, LONDON, SE27 9AF	HEAD OFFICE
2	-3 GLASGOW	1900 GREAT WESTERN RD, GLASGOW, LANARKSHIRE, G13 2TN	DIVISION
3	--6 EDINBURGH	240 EASTER RD, EDINBURGH, MIDLOTHIAN, EH6 8LD	REGION
4	-109 DUNDEE	SOUTH RD, LOCHEE, DUNDEE, ANGUS, DD2 4SR	STORE
5	-106 DUNFERMLINE	CARNEGIE DRIVE, RETAIL PARK, DUNFERMLINE, FIFE, KY12 7AU	STORE
6	-108 KIRKCALDY	439, ESPLANADE WEST, KIRKCALDY FIFE, KY1 1SL	STORE

# Exercise Seven



7.4 If there is sufficient time, amend c:\course\extwo7\_3.sql to use the sys\_connect\_by\_path function to change the display as follows ... Save as c:\course\extwo7\_4.sql

The screenshot shows a SQL Developer window with a query result table. The table has three columns: Name, Address, and Type of Location. The data is as follows:

	Name	Address	Type of Location
1	/WEST NORWOOD	304-322, NORWOOD RD, WEST NORWOOD, LONDON, SE27 9AF	HEAD OFFICE
2	/WEST NORWOOD/ GLASGOW	1900 GREAT WESTERN RD, GLASGOW, LANARKSHIRE, G13 2TN	DIVISION
3	/WEST NORWOOD/ GLASGOW/ EDINBURGH	240 EASTER RD, EDINBURGH, MIDLOTHIAN, EH6 8LD	REGION
4	/WEST NORWOOD/ GLASGOW/ EDINBURGH/ DUNDEE	SOUTH RD, LOCHEE, DUNDEE, ANGUS, DD2 4SR	STORE
5	/WEST NORWOOD/ GLASGOW/ EDINBURGH/ DUNFERMLINE	CARNEGIE DRIVE, RETAIL PARK, DUNFERMLINE, FIFE, KY12 7AU	STORE
6	/WEST NORWOOD/ GLASGOW/ EDINBURGH/ KIRKCALDY	439, ESPLANADE WEST, KIRKCALDY FIFE, KY1 1SL	STORE