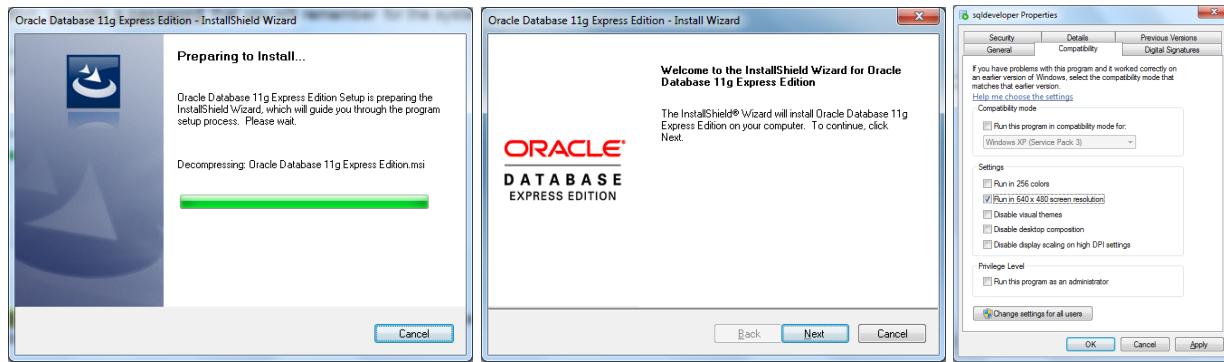
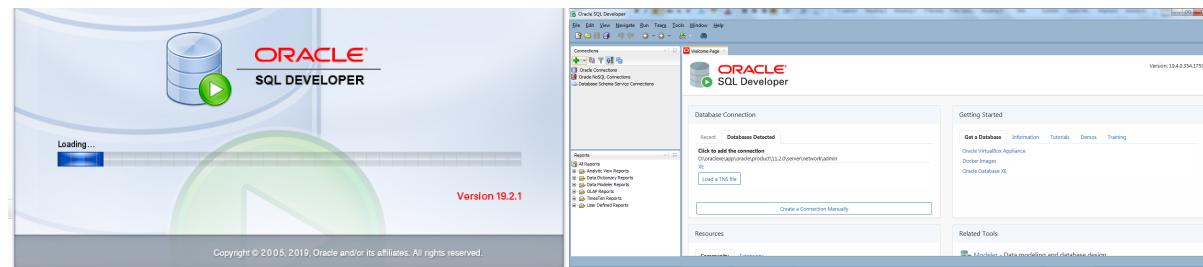


Downloading and Installing ORACLE SQL DEVELOPER

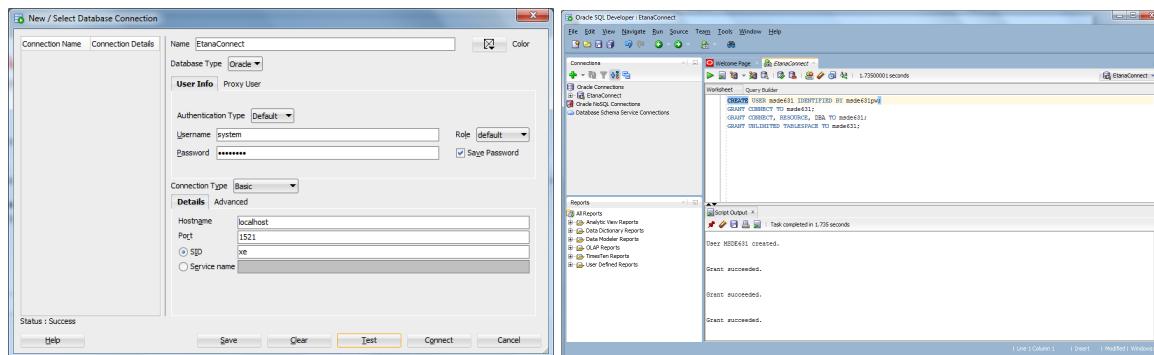


Initiating ORACLE SQL DEVELOPER



Connecting Database

```
CREATE USER msde631 IDENTIFIED BY msde631pw;
GRANT CONNECT TO msde631;
GRANT CONNECT, RESOURCE, DBA TO msde631;
GRANT UNLIMITED TABLESPACE TO msde631;
```



Etana Disasa

MSDE 631 Lab 1: ORACLE SQL Developer



Downloading the product.sql and emp.sql to MSDE631 database

The screenshot shows two Oracle SQL Developer windows side-by-side. The left window displays the 'product.sql' script, which contains SQL commands for creating tables like PRODUCT, CUSTOMER, and LINEITEM, and inserting data into them. The right window displays the 'product.ed' script, which contains similar SQL commands for creating tables and inserting data. Both windows show the connection 'MSDE631' selected.

The screenshot shows a single Oracle SQL Developer window displaying the 'emp.sql' script. This script creates the DEPT and EMP tables and inserts data into them. The connection 'MSDE631' is selected. The 'Script Output' pane shows the results of the insert statements.

Exploring the data we got

The screenshot shows the Oracle SQL Developer interface with the 'Tables' tab selected. A table named 'product' is displayed with columns DEPTNO, DNAME, LOC, and DEPT_ID. The data shows three rows corresponding to the departments ACCOUNTING, RESEARCH, and SALES.

Retrieving data from the table

Etana Disasa

MSDE 631 Lab 1: ORACLE SQL Developer

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar on the left lists 'EtanaConnect' and 'MSDE631'. Under 'MSDE631', the 'Tables (Filtered)' section shows tables: CONTACT, CUSTOMER, CUSTOMERID, DEPT, EMP, LINEITEM, ORDER_DATA, and PRODUCT. The 'Reports' sidebar shows various report types. The 'Worksheet' tab in the center has the SQL query: 'select * from product;'. The 'Query Result' tab displays the following data:

PRODUCTID	CODE	NAME	UNITPRICE
1	10 AB123	Leather Sofa	1000
2	20 AB456	Baby Chair	200.25
3	30 AB789	Sport Shoes	250.6
4	40 PQ123	Sony Digital Camera	399
5	50 PQ456	Hitachi HandyCam	1050
6	60 PQ789	GM Saturn	2250.99

The screenshot shows the Oracle SQL Developer interface, identical to the first one but with a different query in the Worksheet tab: 'select productid,code,name, unitprice from product;'. The 'Query Result' tab displays the same data as the first screenshot:

PRODUCTID	CODE	NAME	UNITPRICE
1	10 AB123	Leather Sofa	1000
2	20 AB456	Baby Chair	200.25
3	30 AB789	Sport Shoes	250.6
4	40 PQ123	Sony Digital Camera	399
5	50 PQ456	Hitachi HandyCam	1050
6	60 PQ789	GM Saturn	2250.99

Etana Disasa

MSDE 631 Lab 1: ORACLE SQL Developer

Restricting Selected Information/logical relationships

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays connections and tables. The central workspace shows a query in the 'Query Builder' tab:

```
select *
from product
where unitprice >= 100 and unitprice <= 500;
```

The 'Query Result' tab shows the fetched data:

PRODUCTID	CODE	NAME	UNITPRICE
1	20 AB456	Baby Chair	200.25
2	30 AB789	Sport Shoes	250.6
3	40 PQ123	Sony Digital Camera	399

Order/Sort results

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays connections and tables. The central workspace shows a query in the 'Query Builder' tab:

```
select *
from product
where unitprice >= 100 and unitprice <= 500
order by unitprice DESC;
```

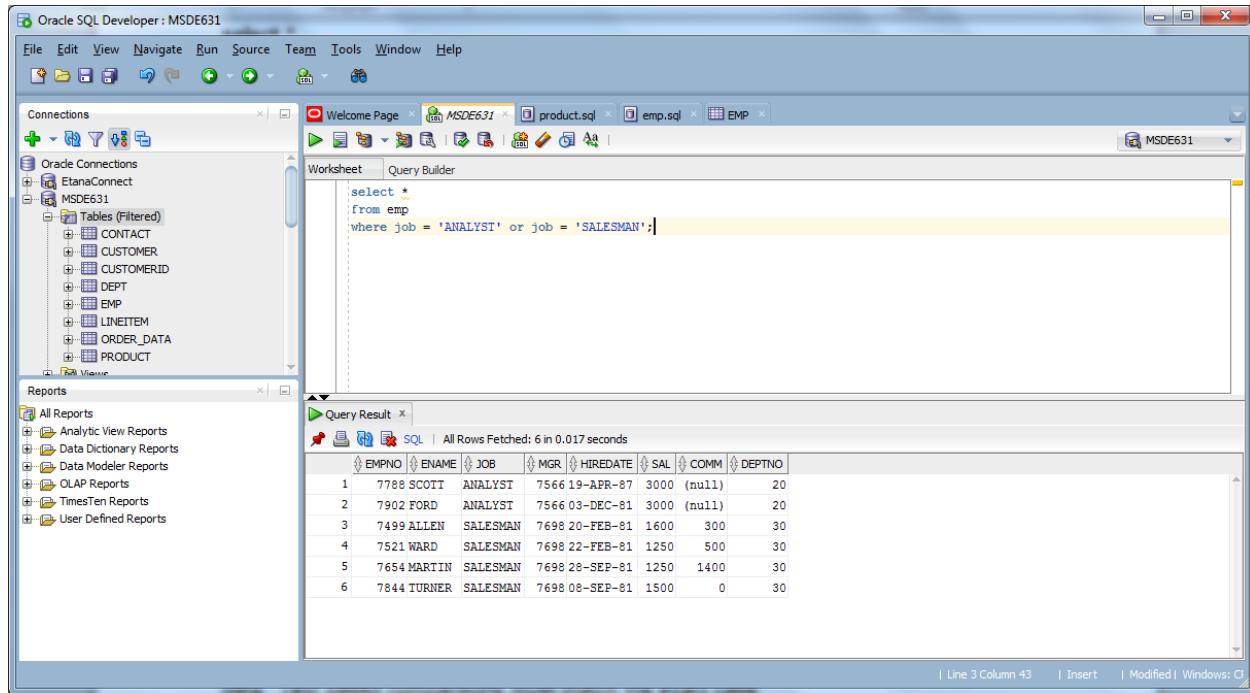
The 'Query Result' tab shows the fetched data, ordered by unit price descending:

PRODUCTID	CODE	NAME	UNITPRICE
1	20 AB456	Baby Chair	200.25
2	30 AB789	Sport Shoes	250.6
3	40 PQ123	Sony Digital Camera	399

Etana Disasa

MSDE 631 Lab 1: ORACLE SQL Developer

OR operator



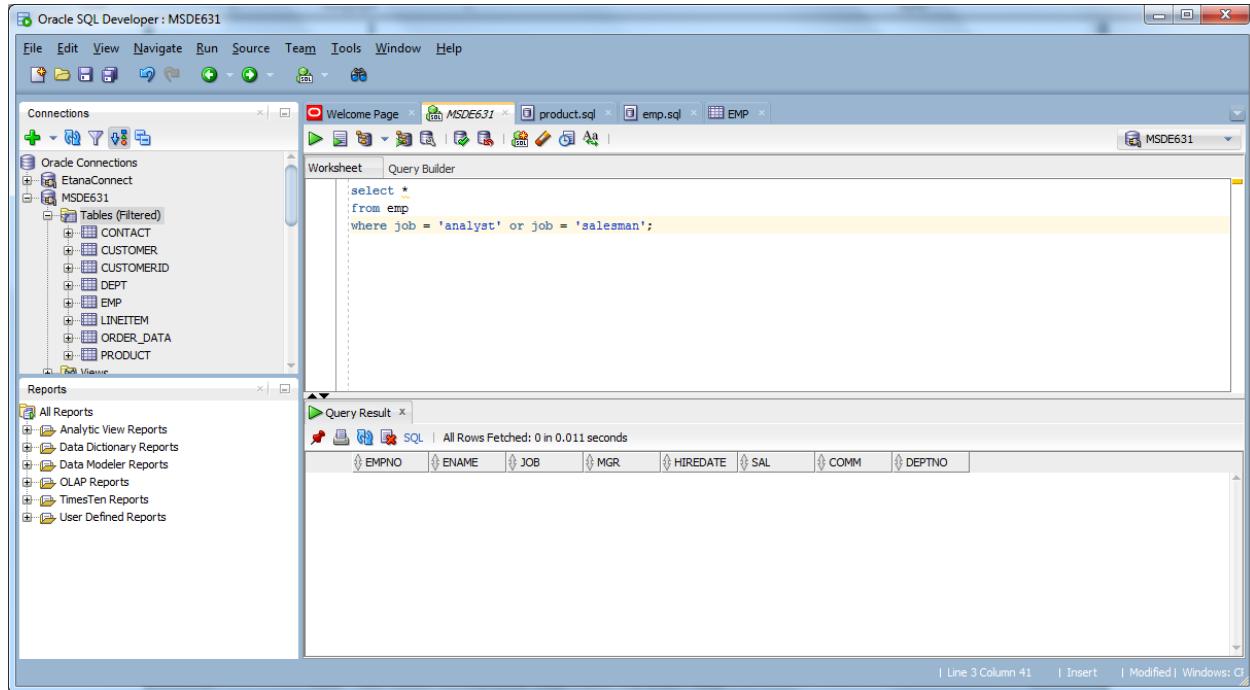
The screenshot shows the Oracle SQL Developer interface. In the central workspace, a query is being run:

```
select *
from emp
where job = 'ANALYST' or job = 'SALESMAN';
```

The results show six rows from the EMP table where the job is either 'ANALYST' or 'SALESMAN':

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7788	SCOTT	ANALYST	7566 19-APR-87	3000	(null)	20
2	7902	FORD	ANALYST	7566 03-DEC-81	3000	(null)	20
3	7499	ALLEN	SALESMAN	7698 20-FEB-81	1600	300	30
4	7521	WARD	SALESMAN	7698 22-FEB-81	1250	500	30
5	7654	MARTIN	SALESMAN	7698 28-SEP-81	1250	1400	30
6	7844	TURNER	SALESMAN	7698 08-SEP-81	1500	0	30

Case sensitivity in ORACLE



The screenshot shows the Oracle SQL Developer interface. In the central workspace, a query is being run:

```
select *
from emp
where job = 'analyst' or job = 'salesman';
```

The results show zero rows, indicating that the OR operator does not consider case sensitivity.

Remedy for the case sensitive issue to use the 'upper' function

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar shows a connection named 'EtanaConnect' and a schema named 'MSDE631'. The 'Tables (Filtered)' section lists several tables: CONTACT, CUSTOMER, CUSTOMERID, DEPT, EMP, LINEITEM, ORDER_DATA, and PRODUCT. The 'Worksheet' tab contains a SQL query:

```
select *
from emp
where job = upper('Analyst') or job = upper('Salesman');
```

The 'Query Result' tab displays the output of the query, which is a table with columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The data is as follows:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7788 SCOTT	ANALYST	7566 19-APR-87	3000 (null)	20		
2	7902 FORD	ANALYST	7566 03-DEC-81	3000 (null)	20		
3	7499 ALLEN	SALESMAN	7698 20-FEB-81	1600	300	30	
4	7521 WARD	SALESMAN	7698 22-FEB-81	1250	500	30	
5	7654 MARTIN	SALESMAN	7698 28-SEP-81	1250	1400	30	
6	7844 TURNER	SALESMAN	7698 08-SEP-81	1500	0	30	

The 'like' function (I understood this as a wildcard operation)

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar shows a connection named 'EtanaConnect' and a schema named 'MSDE631'. The 'Tables (Filtered)' section lists several tables: CONTACT, CUSTOMER, CUSTOMERID, DEPT, EMP, LINEITEM, ORDER_DATA, and PRODUCT. The 'Worksheet' tab contains a SQL query:

```
select *
from emp
where ename like 'M%';
```

The 'Query Result' tab displays the output of the query, which is a table with columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The data is as follows:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7654 MARTIN	SALESMAN	7698 28-SEP-81	1250	1400	30	
2	7934 MILLER	CLERK	7782 23-JAN-82	1300	(null)	10	

Etana Disasa

MSDE 631 Lab 1: ORACLE SQL Developer

BETWEEN/AND operator

The screenshot shows the Oracle SQL Developer interface. In the Worksheet pane, a query is written:

```
select *
from emp
where sal BETWEEN 1100 AND 2000
order by sal;
```

The Query Result pane displays the results of the query:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7876 ADAMS	CLERK	7788	23-MAY-87	1100	(null)	20
2	7521 WARD	SALESMAN	7698	22-FEB-81	1250	500	30
3	7654 MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
4	7934 MILLER	CLERK	7782	23-JAN-82	1300	(null)	10
5	7844 TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
6	7499 ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30

Joining Tables

The screenshot shows the Oracle SQL Developer interface. In the Worksheet pane, a query is written:

```
select a.ename, b.dname
FROM EMP a, DEPT b
WHERE a.deptno = b.deptno;
```

The Query Result pane displays the results of the query:

ENAME	DNAME
KING	ACCOUNTING
BLAKE	SALES
CLARK	ACCOUNTING
JONES	RESEARCH
SCOTT	RESEARCH
FORD	RESEARCH
SMITH	RESEARCH
ALLEN	SALES
WARD	SALES
MARTIN	SALES
TURNER	SALES
ADAMS	RESEARCH
JAMES	SALES
MILLER	ACCOUNTING

Etana Disasa

MSDE 631 Lab 1: ORACLE SQL Developer

Querying unique data

The DISTINCT option

A screenshot of the Oracle SQL Developer interface. The 'Connections' sidebar shows a connection named 'MSDE631'. The 'Tables (Filtered)' section lists several tables: CONTACT, CUSTOMER, CUSTOMERID, DEPT, EMP, LINEITEM, ORDER_DATA, and PRODUCT. The 'Worksheet' tab contains the following SQL query:

```
select DISTINCT deptno
FROM emp;
```

The 'Query Result' tab displays the output:

DEPTNO
1
2
3

All Rows Fetched: 3 in 0.031 seconds

Aggregate Functions

AVERAGE function

A screenshot of the Oracle SQL Developer interface. The 'Connections' sidebar shows a connection named 'MSDE631'. The 'Tables (Filtered)' section lists several tables: CONTACT, CUSTOMER, CUSTOMERID, DEPT, EMP, LINEITEM, ORDER_DATA, and PRODUCT. The 'Worksheet' tab contains the following SQL query:

```
select ROUND (AVG(sal), 2)
FROM emp;
```

The 'Query Result' tab displays the output:

ROUND(AVG(SAL),2)
2073.21

All Rows Fetched: 1 in 0.008 seconds

Etana Disasa

MSDE 631 Lab 1: ORACLE SQL Developer

COUNT function

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar shows 'EtanaConnect' and 'MSDE631'. The 'Tables (Filtered)' section under 'MSDE631' lists CONTACT, CUSTOMER, CUSTOMERID, DEPT, EMP, LINEITEM, ORDER_DATA, and PRODUCT. The 'Worksheet' tab contains the SQL query: `select count(*) FROM emp;`. The 'Query Result' tab shows the output:

COUNT(*)
14

MAX function

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar shows 'EtanaConnect' and 'MSDE631'. The 'Tables (Filtered)' section under 'MSDE631' lists CONTACT, CUSTOMER, CUSTOMERID, DEPT, EMP, LINEITEM, ORDER_DATA, and PRODUCT. The 'Worksheet' tab contains the SQL query: `select deptno, MAX(sal) FROM emp GROUP BY deptno;`. The 'Query Result' tab shows the output:

DEPTNO	MAX(SAL)
1	2850
2	3000
3	5000

Etana Disasa

MSDE 631 Lab 1: ORACLE SQL Developer

GROUP BY function along with aggregate functions

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar shows 'EtanaConnect' and 'MSDE631'. The 'Tables (Filtered)' section under 'MSDE631' lists CONTACT, CUSTOMER, CUSTOMERID, DEPT, EMP, LINEITEM, ORDER_DATA, and PRODUCT. The 'Worksheet' tab contains the following SQL code:

```
select deptno, ROUND(AVG(sal), 2)
FROM emp
GROUP BY deptno
HAVING AVG(sal) > 2000;
```

The 'Query Result' tab displays the output:

DEPTNO	ROUND(AVG(SAL),2)
1	20
2	10

LOWER function as a scalar function

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar shows 'EtanaConnect' and 'MSDE631'. The 'Tables (Filtered)' section under 'MSDE631' lists CONTACT, CUSTOMER, CUSTOMERID, DEPT, EMP, LINEITEM, ORDER_DATA, and PRODUCT. The 'Worksheet' tab contains the following SQL code:

```
select lower(ename) FROM emp;
```

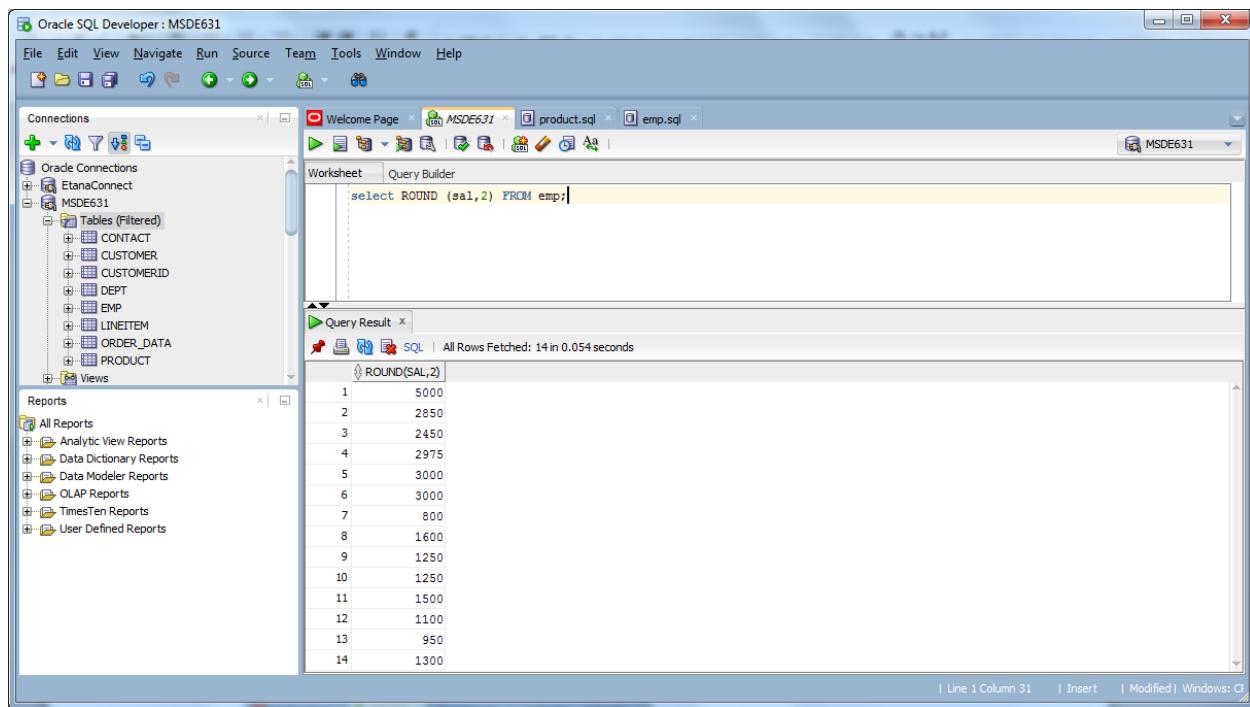
The 'Query Result' tab displays the output:

LOWER(ENAME)
king
blake
clark
jones
scott
ford
smith
allen
ward
martin
turner
adams
james
miller

Etana Disasa

MSDE 631 Lab 1: ORACLE SQL Developer

ROUND function



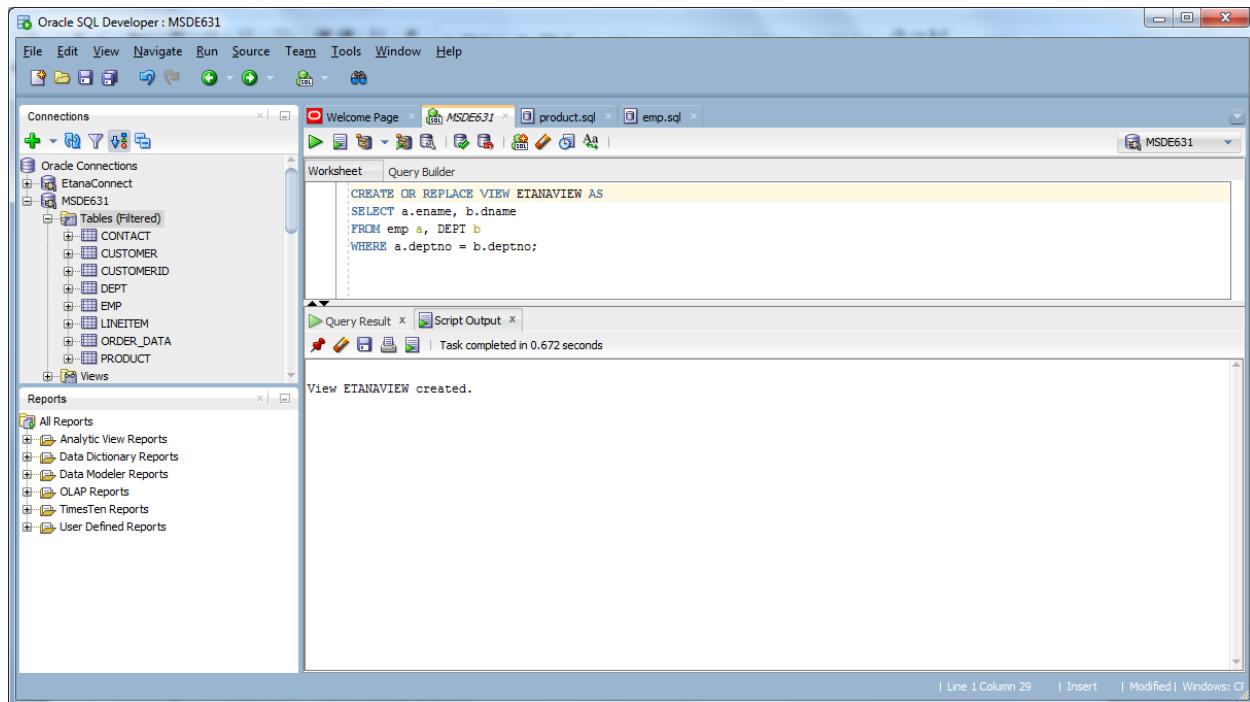
The screenshot shows the Oracle SQL Developer interface. In the Worksheet pane, the following SQL query is run:

```
select ROUND (sal,2) FROM emp;
```

The results are displayed in the Query Result pane:

	ROUND(SAL,2)
1	5000
2	2850
3	2450
4	2975
5	3000
6	3000
7	800
8	1600
9	1250
10	1250
11	1500
12	1100
13	950
14	1300

Views



The screenshot shows the Oracle SQL Developer interface. In the Worksheet pane, the following SQL code is run to create a view:

```
CREATE OR REPLACE VIEW ETANAVIEW AS
SELECT a.ename, b.dname
FROM emp a, DEPT b
WHERE a.deptno = b.deptno;
```

The results are displayed in the Script Output pane:

```
View ETANAVIEW created.
```

Etana Disasa

MSDE 631 Lab 1: ORACLE SQL Developer

Viewing ETANAVIEW (temporary read only table)

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays connections and reports. The main area has a 'Worksheet' tab open with the following SQL query:

```
SELECT * FROM ETANAVIEW;
```

The results are displayed in a table titled 'Query Result' with the following data:

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

PROBLEM

1. The first query will join the emp and dept tables together. The columns to include are shown below. Sort the results by ename. Your result set should match the results shown below exactly.

Answer:

```
SELECT b.ename, a.dname, b.job, b.empno, b.hiredate, a.loc
FROM dept a, emp b
WHERE a.deptno = b.deptno
ORDER BY b.ename ASC;
```

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains the SQL query:SELECT b.ename, a.dname, b.job, b.empno, b.hiredate, a.loc
FROM dept a, emp b
WHERE a.deptno = b.deptno
ORDER BY b.ename ASC;The 'Query Result' tab displays the output of the query, which is a table of employee information. The table has 14 rows and the following data:

	ENAME	DNAME	JOB	EMPNO	HIREDATE	LOC
1	ADAMS	RESEARCH	CLERK	7876	23-MAY-81	DALLAS
2	ALLEN	SALES	SALESMAN	7499	20-FEB-81	CHICAGO
3	BLAKE	SALES	MANAGER	7698	01-MAY-81	CHICAGO
4	CLARK	ACCOUNTING	MANAGER	7782	09-JUN-81	NEW YORK
5	FORD	RESEARCH	ANALYST	7900	03-DEC-81	DALLAS
6	JAMES	SALES	CLERK	7900	03-DEC-81	CHICAGO
7	JONES	RESEARCH	MANAGER	7566	02-APR-81	DALLAS
8	KING	ACCOUNTING	PRESIDENT	7839	17-NOV-81	NEW YORK
9	MARTIN	SALES	SALESMAN	7654	28-SEP-81	CHICAGO
10	MILLER	ACCOUNTING	CLERK	7934	23-JAN-82	NEW YORK
11	SCOTT	RESEARCH	ANALYST	7788	19-APR-87	DALLAS
12	SMITH	RESEARCH	CLERK	7369	17-DEC-80	DALLAS
13	TURNER	SALES	SALESMAN	7844	08-SEP-81	CHICAGO
14	WARD	SALES	SALESMAN	7521	22-FEB-81	CHICAGO

2. For the second query, you might have to Google Oracle SQL commands to write the SQL to match the result set shown below. You will join the emp and dept tables together. The columns to include are shown below, notice that I changed the column names using column aliases. Sort the results descending order by the column named count_of_employees. Your result set should match the results shown below exactly:

Answer:

```
SELECT a.dname, count(b.ename) as "COUNT_OF_EMPLOYEES",
       round(avg(b.sal), 2) as "AVERAGE_SALARY"
  FROM dept a, emp b
 WHERE a.deptno = b.deptno
   group by a.dname
 ORDER BY a.dname DESC;
```

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections panel shows a connection named 'MSDE631' with several tables listed under 'Tables (Filtered)'. The central area has a 'Worksheet' tab open with the provided SQL query. Below it, a 'Query Result' tab displays the execution output in a table:

DNAME	COUNT_OF_EMPLOYEES	AVERAGE_SALARY
1 SALES	6	1566.67
2 RESEARCH	5	2175
3 ACCOUNTING	3	2916.67