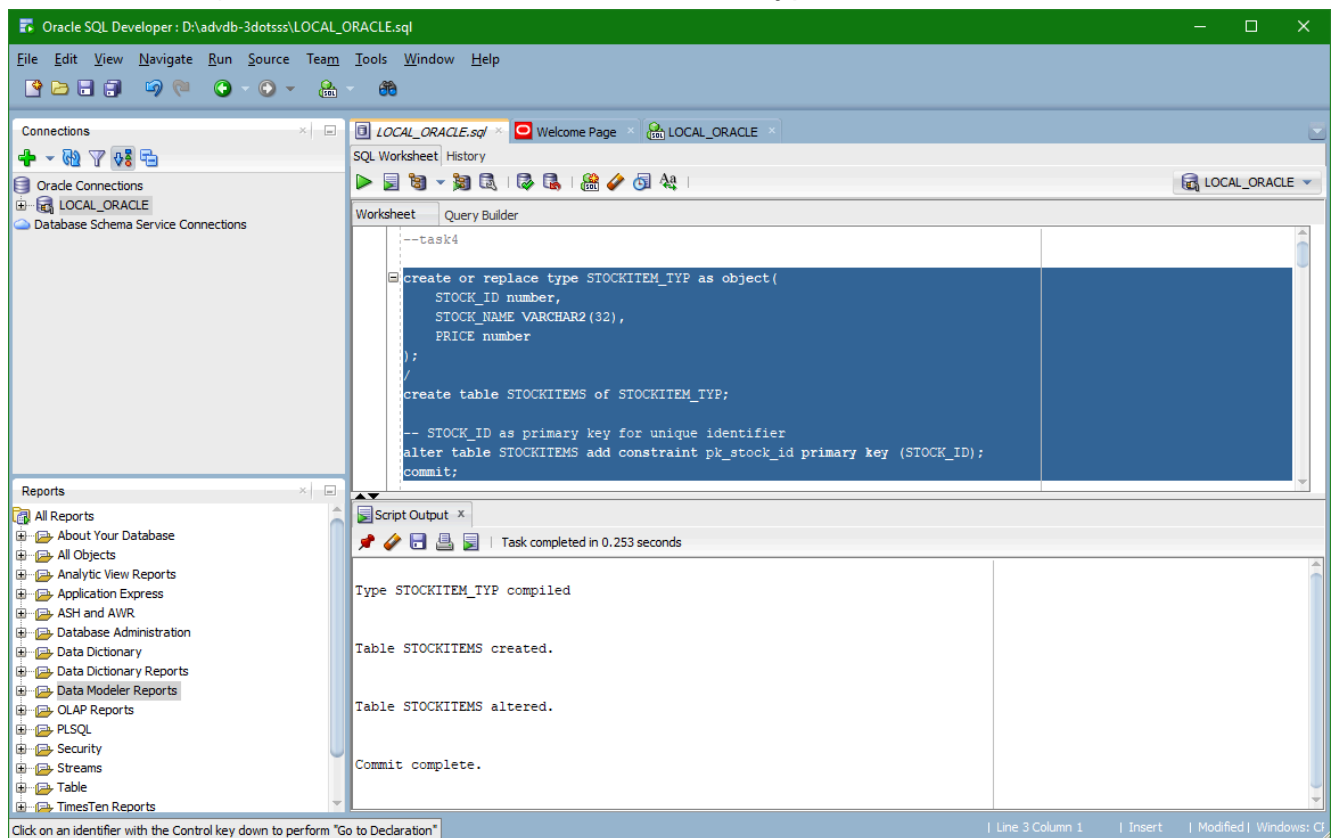


Name: Shaheer Shamsi
Username: shaheer-shamsi

3dotsss advdb Task 04

Modeling and Referencing Stock and Line Items in Oracle ORDBMS

a) Define a Stock Item as a Custom Type and Create a Table



b) Define a Line Item Type That References a Stock Item

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL script in the 'Worksheet' tab, which defines a new type and a table. The script is as follows:

```
create or replace type lineitem_typ as object(  
    lineitem_no number,  
    stockref ref STOCKITEM_TYP,  
    quantity NUMBER  
);  
  
create table lineitems of lineitem_typ;  
  
-- lineitem_no as primary key for unique identifier  
alter table lineitems add constraint pk_lineitem_no primary key (lineitem_no);  
  
commit;
```

The 'Script Output' window at the bottom shows the execution results:

```
Type LINEITEM_TYP compiled  
  
Table LINEITEMS created.  
  
Table LINEITEMS altered.  
  
Commit complete.
```

The 'Connections' panel on the left shows the 'LOCAL_ORACLE' connection selected. The 'Reports' panel on the bottom left lists various report categories, with 'Data Modeler Reports' currently selected.

c) Populate the Stock Items Table

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL script in the 'Worksheet' tab, which is used to populate the 'STOCKITEMS' table. The script consists of four 'insert into' statements, each adding a new row with a unique ID, a description, and a quantity. The script concludes with a 'commit;' statement. The 'Script Output' window at the bottom shows the execution results, indicating that each insert statement successfully added one row and the commit was completed. The 'Connections' panel on the left shows the 'LOCAL_ORACLE' connection selected. The 'Reports' panel on the left lists various report categories, with 'Data Modeler Reports' currently selected. The status bar at the bottom indicates the current position is at Line 52, Column 8.

```
insert into STOCKITEMS values(
  101,
  'Matador Hi-School Pen',
  5
);
insert into STOCKITEMS values(
  102,
  'University 120 Page Script',
  60
);
insert into STOCKITEMS values(
  103,
  '12 Inch Ruler',
  40
);
insert into STOCKITEMS values(
  104,
  'Mechanical Pencil 0.5mm',
  80
);
commit;
```

Script Output x

Task completed in 0.231 seconds

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

Commit complete.

Click on an identifier with the Control key down to perform "Go to Declaration" | Line 52 Column 8 | Insert | Modified | Windows: CJ

d) Populate the Line Items Table

The screenshot shows the Oracle SQL Developer interface. The main window displays an SQL script with four insert statements and a commit. The script is executed, and the Script Output pane shows the results of the execution.

```
insert into lineitems
select 1001, ref(s), 15 from stockitems s where STOCK_ID = 101;

insert into lineitems
select 1002, ref(s), 10 from stockitems s where STOCK_ID = 102;

insert into lineitems
select 1003, ref(s), 5 from stockitems s where STOCK_ID = 103;

insert into lineitems
select 1004, ref(s), 3 from stockitems s where STOCK_ID = 104;

commit;
```

Script Output:

```
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
Commit complete.
```

Click on an identifier with the Control key down to perform "Go to Declaration" | Line 67 Column 1 | Insert | Modified | Windows: C

e) Display the Stock Name of a Specific Line Item (by line item number 1002)

The screenshot shows the Oracle SQL Developer interface. The main window displays an SQL query to select the stock name of line item 1002. The query is executed, and the Query Result pane shows the results.

```
-- display stock name of item 1002
select
  l.lineitem_No,
  l.stockref.STOCK_NAME as stock_name,
  --showing price and qty of lineitem 1002 as well
  l.stockref.PRICE as price,
  l.quantity as Qty
from lineitems l
where l.lineitem_No = 1002;
```

Query Result:

LINEITEM_NO	STOCK_NAME	PRICE	QTY
1	1002 University 120 Page Script	60	10

Click on an identifier with the Control key down to perform "Go to Declaration" | Line 91 Column 1 | Insert | Modified | Windows: C

f) Find All Line Items Referencing a Specific Stock Item (stock item number 102)

The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left lists 'LOCAL_ORACLE'. The 'SQL Worksheet' pane contains the following SQL query:

```
-- find all ref 102
select
  l.lineitem_No,
  l.stockref.STOCK_NAME as stock_name,
  l.stockref.PRICE as price,
  l.quantity as Qty
from lineitems l
where l.stockref.STOCK_ID = 102;
```

The 'Query Result' pane shows the results of the query:

LINEITEM_NO	STOCK_NAME	PRICE	QTY
1	1002 University 120 Page Script	60	10

The status bar at the bottom indicates 'Line 99 Column 1 | Insert | Modified | Windows: C'.

Using Deref:

The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left lists 'LOCAL_ORACLE'. The 'SQL Worksheet' pane contains the following SQL query:

```
-- or using Deref. I prefer using dot
select
  l.lineitem_No,
  Deref(l.stockref).STOCK_NAME as stock_name,
  Deref(l.stockref).PRICE as price,
  l.quantity
from lineitems l
where Deref(l.stockref).STOCK_ID = 102;
```

The 'Query Result' pane shows the results of the query:

LINEITEM_NO	STOCK_NAME	PRICE	QUANTITY
1	1002 University 120 Page Script	60	10

The status bar at the bottom indicates 'Line 109 Column 1 | Insert | Modified | Windows: C'.

SQL Source Code:

--task4

```
create or replace type STOCKITEM_TYP as object(  
    STOCK_ID number,  
    STOCK_NAME VARCHAR2(32),  
    PRICE number  
);  
/  
create table STOCKITEMS of STOCKITEM_TYP;  
-- STOCK_ID as primary key for unique identifier  
alter table STOCKITEMS add constraint pk_stock_id primary key (STOCK_ID);  
commit;  
create or replace type lineitem_typ as object(  
    lineitem_No number,  
    stockref ref STOCKITEM_TYP,  
    quantity NUMBER  
);  
/  
create table lineitems of lineitem_typ;  
-- lineitem_No as primary key for unique identifier  
alter table lineitems add constraint pk_lineitem_no primary key  
(lineitem_No);  
commit;  
insert into STOCKITEMS values(  
    101,  
    'Matador Hi-School Pen',  
    5  
);  
insert into STOCKITEMS values(  
    102,  
    'University 120 Page Script',  
    60  
);  
  
insert into STOCKITEMS values(  
    103,  
    '12 Inch Ruler',  
    40  
);  
insert into STOCKITEMS values(  
    104,  
    'Mechanical Pencil 0.5mm',  
    80  
);  
commit;
```

```
insert into lineitems
select 1001, ref(s), 15 from stockitems s where STOCK_ID = 101;
```

```
insert into lineitems
select 1002, ref(s), 10 from stockitems s where STOCK_ID = 102;
```

```
insert into lineitems
select 1003, ref(s), 5 from stockitems s where STOCK_ID = 103;
```

```
insert into lineitems
select 1004, ref(s), 3 from stockitems s where STOCK_ID = 104;
```

```
commit;
```

```
-- test query from 3dotsss Class
select * from lineitems;
```

```
select lineitem_no, l.stockref.STOCK_NAME as stock_name,
l.stockref.PRICE, quantity as Qty
from lineitems l
where lineitem_no = 1002;
```

```
select lineitem_no, l.stockref.STOCK_NAME as stock_name,
l.stockref.PRICE, quantity
from lineitems l;
```

```
commit;
```

```
-- display stock name of item 1002
select
    l.lineitem_No,
    l.stockref.STOCK_NAME as stock_name,
    --showing price and qty of lineitem 1002 as well
    l.stockref.PRICE as price,
    l.quantity as Qty
from lineitems l
where l.lineitem_No = 1002;
select
    l.lineitem_No,
    Deref(l.stockref).STOCK_NAME as stock_name,
    Deref(l.stockref).PRICE as price,
    l.quantity
```

```
from lineitems l
where Deref(l.stockref).STOCK_ID = 102;
```

```
-- find all ref 102
```

```
select
    l.lineitem_No,
    l.stockref.STOCK_NAME as stock_name,
    l.stockref.PRICE as price,
    l.quantity as Qty
from lineitems l
where l.stockref.STOCK_ID = 102;
```

```
-- or using Deref. I prefer using dot
```

```
select
    l.lineitem_No,
    Deref(l.stockref).STOCK_NAME as stock_name,
    Deref(l.stockref).PRICE as price,
    l.quantity
from lineitems l
where Deref(l.stockref).STOCK_ID = 102;
```

```
-- view
```

```
select
    l.lineitem_No,
    l.stockref.STOCK_ID as stock_id,
    l.stockref.STOCK_NAME as stock_name,
    l.stockref.PRICE as price,
    l.quantity
from lineitems l;
```