Ryan Devlin

Prof Crawford

COMP281

HW 9-1

set echo on;

set serveroutput on;

--1. Create an SQL query that uses set operations to display the union of the last names in table CUSTOMER and

--the last names in table EMPLOYEE. Do not include duplicates in the output.

SELECT CUS\_LNAME FROM CUSTOMER

UNION

SELECT EMP\_LNAME FROM EMPLOYEE;

--2. Create an SQL query that uses set operations to display the union of the last names in table CUSTOMER and

--the last names in table EMPLOYEE. This time, include duplicates in the output.

SELECT CUS\_LNAME FROM CUSTOMER

UNION ALL

SELECT EMP\_LNAME FROM EMPLOYEE;

--3. Create an SQL query that USES AN UNCORRELATED SUBQUERY2 AND NO JOINS3

--to display the

--descriptions for products provided by a vendor in area code 615.

SELECT P\_DESCRIPT FROM PRODUCT

WHERE V\_CODE IN (SELECT V\_CODE FROM VENDOR

WHERE V\_AREACODE = '615');

--4. Create an SQL query that USES A CORRELATED SUBQUERY to display the descriptions for products

--provided by a vendor in area code 615 (a join is okay here).

SELECT P\_DESCRIPT FROM PRODUCT

WHERE V\_CODE = (SELECT V\_CODE FROM VENDOR

WHERE VENDOR.V\_CODE = PRODUCT.V\_CODE

AND VENDOR.V\_AREACODE = '615');

/\*

SQL> set echo on

SQL> set serveroutput on

SQL> --1. Create an SQL query that uses set operations to display the union of the last names in table CUSTOMER and

SQL> --the last names in table EMPLOYEE. Do not include duplicates in the output.

SQL> SELECT CUS\_LNAME FROM CUSTOMER

UNION

SELECT EMP\_LNAME FROM EMPLOYEE;

CUS\_LNAME

---------------

Brandon

Brown

Diante

Dunne

Farriss

Genkazi

Johnson

Jones

Kolmycz

Lange

Lewis

CUS\_LNAME

---------------

O'Brian

Olowski

Orlando

Ramas

Saranda

Smith

Smythe

Vandam

Washington

Wiesenbach

Williams

22 rows selected

SQL> --2. Create an SQL query that uses set operations to display the union of the last names in table CUSTOMER and

SQL> --the last names in table EMPLOYEE. This time, include duplicates in the output.

SQL> SELECT CUS\_LNAME FROM CUSTOMER

UNION ALL

SELECT EMP\_LNAME FROM EMPLOYEE;

CUS\_LNAME

---------------

Brown

Dunne

Farriss

O'Brian

Olowski

Orlando

Ramas

Smith

Smith

Williams

Kolmycz

CUS\_LNAME

---------------

Lewis

Vandam

Jones

Williams

Lange

Smith

Diante

Wiesenbach

Smith

Genkazi

Washington

CUS\_LNAME

---------------

Johnson

Smythe

Brandon

Saranda

Smith

27 rows selected

SQL> --3. Create an SQL query that USES AN UNCORRELATED SUBQUERY2 AND NO JOINS3

SQL> --to display the

SQL> --descriptions for products provided by a vendor in area code 615.

SQL> SELECT P\_DESCRIPT FROM PRODUCT

WHERE V\_CODE IN (SELECT V\_CODE FROM VENDOR

WHERE V\_AREACODE = '615');

P\_DESCRIPT

-----------------------------------

7.25-in. pwr. saw blade

9.00-in. pwr. saw blade

B&D jigsaw, 12-in. blade

B&D jigsaw, 8-in. blade

Claw hammer

Rat-tail file, 1/8-in. fine

Hicut chain saw, 16 in.

1.25-in. metal screw, 25

2.5-in. wd. screw, 50

9 rows selected

SQL> --4. Create an SQL query that USES A CORRELATED SUBQUERY to display the descriptions for products

SQL> --provided by a vendor in area code 615 (a join is okay here).

SQL> SELECT P\_DESCRIPT FROM PRODUCT

WHERE V\_CODE = (SELECT V\_CODE FROM VENDOR

WHERE VENDOR.V\_CODE = PRODUCT.V\_CODE

AND VENDOR.V\_AREACODE = '615');

P\_DESCRIPT

-----------------------------------

7.25-in. pwr. saw blade

9.00-in. pwr. saw blade

B&D jigsaw, 12-in. blade

B&D jigsaw, 8-in. blade

Claw hammer

Rat-tail file, 1/8-in. fine

Hicut chain saw, 16 in.

1.25-in. metal screw, 25

2.5-in. wd. screw, 50

9 rows selected

\*/

set echo on;

set serveroutput on;

CREATE OR REPLACE PROCEDURE prc\_inv\_delete (INV\_NUM IN NUMBER)

IS

BEGIN

DELETE from INVOICE where INV\_NUMBER = INV\_NUM;

END;

/

commit;

select \* from invoice where inv\_number in (1001,1008);

select \* from line where inv\_number in (1001,1008);

exec prc\_inv\_delete(1001);

exec prc\_inv\_delete(1008);

select \* from invoice where inv\_number in (1001,1008);

select \* from line where inv\_number in (1001,1008);

rollback;

--The corresponding lines in LINE did get deleted for invoices 1001 and 1008

--This is because of the CASCADE DELETE clause on the foreign key on the LINE table

--CASCADE DELETE will delete any records in the child table (LINE) if the corresponding record in the parent table (INVOICE) is deleted

/\*

SQL> set echo on

SQL> set serveroutput on

SQL> CREATE OR REPLACE PROCEDURE prc\_inv\_delete (INV\_NUM IN NUMBER)

IS

BEGIN

DELETE from INVOICE where INV\_NUMBER = INV\_NUM;

END;

/

Procedure PRC\_INV\_DELETE compiled

SQL> commit;

Commit complete.

SQL> select \* from invoice where inv\_number in (1001,1008);

INV\_NUMBER CUS\_CODE INV\_DATE

---------- ---------- ---------

1001 10014 16-JAN-12

1008 10011 17-JAN-12

SQL> select \* from line where inv\_number in (1001,1008);

INV\_NUMBER LINE\_NUMBER P\_CODE LINE\_UNITS LINE\_PRICE

---------- ----------- ---------- ---------- ----------

1001 1 13-Q2/P2 1 14.99

1001 2 23109-HB 1 9.95

1008 3 23109-HB 1 9.95

1008 1 PVC23DRT 5 5.87

1008 2 WR3/TT3 3 119.95

SQL> exec prc\_inv\_delete(1001);

PL/SQL procedure successfully completed.

SQL> exec prc\_inv\_delete(1008);

PL/SQL procedure successfully completed.

SQL> select \* from invoice where inv\_number in (1001,1008);

no rows selected

SQL> select \* from line where inv\_number in (1001,1008);

no rows selected

SQL> rollback;

Rollback complete.

\*/

set echo on;

set serveroutput on;

CREATE OR REPLACE TRIGGER trg\_prod\_QOH\_on\_line\_add

AFTER INSERT ON LINE FOR EACH ROW

BEGIN

UPDATE PRODUCT

SET P\_QOH = P\_QOH - :new.LINE\_UNITS

where P\_CODE = :new.P\_CODE;

END;

/

COMMIT;

SELECT P\_CODE, P\_QOH FROM PRODUCT WHERE P\_CODE='54778-2T';

SELECT \* FROM LINE WHERE P\_CODE='54778-2T';

INSERT INTO LINE VALUES(1005,2,'54778-2T',12,5.87);

SELECT P\_CODE, P\_QOH FROM PRODUCT WHERE P\_CODE='54778-2T';

SELECT \* FROM LINE WHERE P\_CODE='54778-2T';

rollback;

/\*

SQL> set serveroutput on

SQL> CREATE OR REPLACE TRIGGER trg\_prod\_QOH\_on\_line\_add

AFTER INSERT ON LINE FOR EACH ROW

BEGIN

UPDATE PRODUCT

SET P\_QOH = P\_QOH - :new.LINE\_UNITS

where P\_CODE = :new.P\_CODE;

END;

/

Trigger TRG\_PROD\_QOH\_ON\_LINE\_ADD compiled

SQL> SELECT P\_CODE, P\_QOH FROM PRODUCT WHERE P\_CODE='54778-2T';

P\_CODE P\_QOH

---------- ----------

54778-2T 43

SQL> SELECT \* FROM LINE WHERE P\_CODE='54778-2T';

INV\_NUMBER LINE\_NUMBER P\_CODE LINE\_UNITS LINE\_PRICE

---------- ----------- ---------- ---------- ----------

1002 1 54778-2T 2 4.99

1004 1 54778-2T 3 4.99

1007 2 54778-2T 1 4.99

SQL> INSERT INTO LINE VALUES(1005,2,'54778-2T',12,5.87);

1 row inserted.

SQL> SELECT P\_CODE, P\_QOH FROM PRODUCT WHERE P\_CODE='54778-2T';

P\_CODE P\_QOH

---------- ----------

54778-2T 31

SQL> SELECT \* FROM LINE WHERE P\_CODE='54778-2T';

INV\_NUMBER LINE\_NUMBER P\_CODE LINE\_UNITS LINE\_PRICE

---------- ----------- ---------- ---------- ----------

1002 1 54778-2T 2 4.99

1004 1 54778-2T 3 4.99

1007 2 54778-2T 1 4.99

1005 2 54778-2T 12 5.87

SQL> rollback;

Rollback complete.

\*/

set echo on;

set serveroutput on;

CREATE OR REPLACE TRIGGER trg\_prod\_QOH\_on\_line\_delete

AFTER DELETE ON LINE FOR EACH ROW

BEGIN

UPDATE PRODUCT

SET P\_QOH = P\_QOH + :old.LINE\_UNITS

where P\_CODE = :old.P\_CODE;

END;

/

COMMIT;

SELECT P\_CODE, P\_QOH FROM PRODUCT WHERE P\_CODE='54778-2T';

SELECT \* FROM LINE WHERE P\_CODE='54778-2T';

DELETE FROM LINE WHERE INV\_NUMBER = 1004 AND P\_CODE ='54778-2T';

SELECT P\_CODE, P\_QOH FROM PRODUCT WHERE P\_CODE='54778-2T';

SELECT \* FROM LINE WHERE P\_CODE='54778-2T';

rollback;

/\*

SQL> set echo on

SQL> set serveroutput on

SQL> CREATE OR REPLACE TRIGGER trg\_prod\_QOH\_on\_line\_delete

AFTER DELETE ON LINE FOR EACH ROW

BEGIN

UPDATE PRODUCT

SET P\_QOH = P\_QOH + :old.LINE\_UNITS

where P\_CODE = :old.P\_CODE;

END;

/

Trigger TRG\_PROD\_QOH\_ON\_LINE\_DELETE compiled

SQL> SELECT P\_CODE, P\_QOH FROM PRODUCT WHERE P\_CODE='54778-2T';

P\_CODE P\_QOH

---------- ----------

54778-2T 43

SQL> SELECT \* FROM LINE WHERE P\_CODE='54778-2T';

INV\_NUMBER LINE\_NUMBER P\_CODE LINE\_UNITS LINE\_PRICE

---------- ----------- ---------- ---------- ----------

1002 1 54778-2T 2 4.99

1004 1 54778-2T 3 4.99

1007 2 54778-2T 1 4.99

SQL> DELETE FROM LINE WHERE INV\_NUMBER = 1004 AND P\_CODE ='54778-2T';

1 row deleted.

SQL> SELECT P\_CODE, P\_QOH FROM PRODUCT WHERE P\_CODE='54778-2T';

P\_CODE P\_QOH

---------- ----------

54778-2T 46

SQL> SELECT \* FROM LINE WHERE P\_CODE='54778-2T';

INV\_NUMBER LINE\_NUMBER P\_CODE LINE\_UNITS LINE\_PRICE

---------- ----------- ---------- ---------- ----------

1002 1 54778-2T 2 4.99

1007 2 54778-2T 1 4.99

SQL> rollback;

Rollback complete.

\*/

set echo on;

set serveroutput on;

CREATE OR REPLACE TRIGGER trg\_prod\_QOH\_on\_line\_update

AFTER UPDATE ON LINE FOR EACH ROW

BEGIN

UPDATE PRODUCT

SET P\_QOH = P\_QOH + (:new.LINE\_UNITS - :old.LINE\_UNITS)

where P\_CODE = :old.P\_CODE;

END;

/

COMMIT;

SELECT P\_CODE, P\_QOH FROM PRODUCT WHERE P\_CODE='54778-2T';

SELECT \* FROM LINE WHERE P\_CODE='54778-2T';

UPDATE LINE SET LINE\_UNITS=LINE\_UNITS\*2 where P\_CODE='54778-2T';

SELECT P\_CODE, P\_QOH FROM PRODUCT WHERE P\_CODE='54778-2T';

SELECT \* FROM LINE WHERE P\_CODE='54778-2T';

rollback;

/\*

SQL> set echo on

SQL> set serveroutput on

SQL> CREATE OR REPLACE TRIGGER trg\_prod\_QOH\_on\_line\_update

AFTER UPDATE ON LINE FOR EACH ROW

BEGIN

UPDATE PRODUCT

SET P\_QOH = P\_QOH + (:new.LINE\_UNITS - :old.LINE\_UNITS)

where P\_CODE = :old.P\_CODE;

END;

/

Trigger TRG\_PROD\_QOH\_ON\_LINE\_UPDATE compiled

SQL> SELECT P\_CODE, P\_QOH FROM PRODUCT WHERE P\_CODE='54778-2T';

P\_CODE P\_QOH

---------- ----------

54778-2T 43

SQL> SELECT \* FROM LINE WHERE P\_CODE='54778-2T';

INV\_NUMBER LINE\_NUMBER P\_CODE LINE\_UNITS LINE\_PRICE

---------- ----------- ---------- ---------- ----------

1002 1 54778-2T 2 4.99

1004 1 54778-2T 3 4.99

1007 2 54778-2T 1 4.99

SQL> UPDATE LINE SET LINE\_UNITS=LINE\_UNITS\*2 where P\_CODE='54778-2T';

3 rows updated.

SQL> SELECT P\_CODE, P\_QOH FROM PRODUCT WHERE P\_CODE='54778-2T';

P\_CODE P\_QOH

---------- ----------

54778-2T 49

SQL> SELECT \* FROM LINE WHERE P\_CODE='54778-2T';

INV\_NUMBER LINE\_NUMBER P\_CODE LINE\_UNITS LINE\_PRICE

---------- ----------- ---------- ---------- ----------

1002 1 54778-2T 4 4.99

1004 1 54778-2T 6 4.99

1007 2 54778-2T 2 4.99

SQL> rollback;

Rollback complete.

\*/