Name: Dev Jatinbhai

Patel Roll No: CE076

Id No: 20CEUOS018

DBMS LAB 9

1. Create a trigger product_i, which should ensure that the product_no given at the time of insertion, in sales_order_details should be present in product_master.

PL/SQL:

```
CREATE OR REPLACE TRIGGER product_i BEFORE
INSERT ON sales_order_details FOR EACH ROW

DECLARE
C NUMBER(2); BEGIN

SELECT COUNT(*) into C FROM product_master WHERE product_master.product_no = :new.product_no;

IF (C=0) THEN
dbms_output.put_line('product no is not present in product_master'); END IF;

END;
```

```
insert into sales_order_details values('O19001','P00002',2,2,530);
```

Script Output × sod × po ×

Task completed in 0.154 seconds

```
Trigger PRODUCT_I compiled


Trigger PRODUCT_I compiled

product no is not present in product_master
```

2. Create a trigger sales_u, which should ensure that if client_no or s_order_no is modified in the sales_order table it should do the corresponding changes in the other tables. PL/SQL:

```
CREATE OR REPLACE TRIGGER sales_u AFTER
UPDATE OF client_no, s_order_no ON sales_order FOR EACH ROW

BEGIN

UPDATE client_master SET client_no = :new.client_no WHERE client_no = :old.client_no;
UPDATE s_order_details SET s_order_no = :new.s_order_no WHERE s_order_no = :old.s_order_no;

END;
```

```
update sales_order set client_no = 1 where client_no = 0001;
```

Script Output | sod | po | so 1 | cm

SQL | All Rows Fetched: 6 in 0.006 seconds

	S_ORDER_NO	SALESMAN_NO	ORDER_DATE	CLIENT_NO	DELY_ADD	DELY_TYPE	BILL_YN	DELY_DATE	ORDER_STATUS
1	O19001	S00001	12-01-96	1	-	F	N	20-01-96	in process
2	O19002	S00002	25-01-96	0002	-	P	N	27-01-96	cancelled
3	O16865	S00003	18-02-96	0003	-	F	Y	20-02-96	fulfilled
4	O19003	S00001	03-04-96	1	-	F	Y	07-04-96	fulfilled
5	O46866	S00002	20-05-96	0004	-	P	N	22-05-96	cancelled
6	O10008	S00004	24-05-96	0005	-	F	N	26-05-96	in process

3. Create a trigger sales_u1, which should ensure that if s_order_no is modified in sales_order_details table then it should not allow it and display an error message. PL/SQL:

```
create or replace trigger sales_u1 before update of s_order_no on
sales_order_details for each row

begin
RAISE_APPLICATION_ERROR(-20001,'order_no modification is not allowed'); end;
```

```
update sales_order_details set s_order_no = 'O19105' where s_order_no='O19002';
```

Script Output | sod | po | so 1 | cm

Task completed in 0.068 seconds

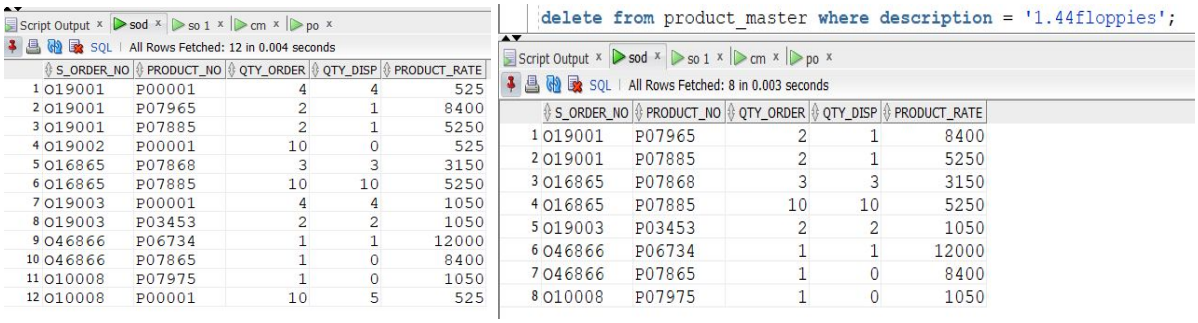
```
Error starting at line : 180 in command -
update sales_order_details set s_order_no = 'O19105' where s_order_no='O19002'
Error report -
ORA-20001: order_no modification is not allowed
ORA-06512: at "PAVAN.SALES_U1", line 2
ORA-04088: error during execution of trigger 'PAVAN.SALES_U1'
```

4. Create a trigger product_d, which should ensure that if a product is deleted from product_master table all corresponding entries should be removed from the sales_order_details table.

PL/SQL:

```
create or replace trigger product_d after delete on product_master
for each row begin

delete from sales_order_details where product_no = :old.product_no;
delete from challan_details where product_no = :old.product_no; end;
```



Script Output | sod | so 1 | cm | po

SQL | All Rows Fetched: 12 in 0.004 seconds

	S_ORDER_NO	PRODUCT_NO	QTY_ORDER	QTY_DISP	PRODUCT_RATE
1	O19001	P00001	4	4	525
2	O19001	P07965	2	1	8400
3	O19001	P07885	2	1	5250
4	O19002	P00001	10	0	525
5	O16865	P07868	3	3	3150
6	O16865	P07885	10	10	5250
7	O19003	P00001	4	4	1050
8	O19003	P03453	2	2	1050
9	O46866	P06734	1	1	12000
10	O46866	P07865	1	0	8400
11	O10008	P07975	1	0	1050
12	O10008	P00001	10	5	525

```
delete from product_master where description = '1.44floppies';
```

Script Output | sod | so 1 | cm | po

SQL | All Rows Fetched: 8 in 0.003 seconds

	S_ORDER_NO	PRODUCT_NO	QTY_ORDER	QTY_DISP	PRODUCT_RATE
1	O19001	P07965	2	1	8400
2	O19001	P07885	2	1	5250
3	O16865	P07868	3	3	3150
4	O16865	P07885	10	10	5250
5	O19003	P03453	2	2	1050
6	O46866	P06734	1	1	12000
7	O46866	P07865	1	0	8400
8	O10008	P07975	1	0	1050

5. Write down one trigger example. PL/SQL:

```
create or replace trigger qtn_reminder before update of qty_on_hand on product_master
for each row begin

if(:new.qty_on_hand < 5) then
dbms_output.put_line('now, you have less than 5 quantity for this product'); end if;

end;
```

```
update product_master set qty_on_hand = 4 where description='1.44floppies';
```

Script Output | sod | so 1 | cm | po

Task completed in 0.064 seconds

```
now, you have less than 5 quantity for this product


1 row updated.
```

Thank
you.....