

COMP23111 2016 - 2017 EX4

Sameul Da Costa - 9708530

December 3, 2016

1 Comments

I have made the assumption that columns should be formatted in the individual exercises and that the formatting should be removed afterwards. This has lead to some repetition but overall I believe it splits the exercises up more effectively.

Furthermore certain things in part 1g *should* throw errors, please see comments for details

2 Code

```
SQL> -- 1a -----
SQL> CREATE OR REPLACE VIEW customers_with_shopping_carts
  2 AS
  3     SELECT
  4     DISTINCT   firstName ,
  5               lastName
  6     FROM       customerInfo
  7     INNER JOIN orderCartInfo
  8     ON         orderCartInfo.customerID=customerInfo.loginName;
```

View created.

```
SQL>
SQL> SELECT * FROM customers_with_shopping_carts;
```

FIRSTNAME	LASTNAME
Ophelia	Elsinore
Mary	Folio
Percy	Byron

```
SQL>
SQL> -- 1b -----
SQL>
SQL> CREATE OR REPLACE VIEW items_to_reorder
  2 AS
  3     SELECT   code ,
  4             inventoryItem.itemNum,
```

```

5          belongsTo AS categoryID ,
6          qtyInstock AS quantityInStock
7  FROM    inventoryItem
8  INNER JOIN itemType
9  ON      inventoryItem.itemNum = itemType.itemNum
10 WHERE   qtyInstock < 25;

```

View created.

```

SQL>
SQL> COLUMN CODE FORMAT A5
SQL> SELECT * FROM items_to_reorder;

```

CODE	ITEMNUM	CATEGORYID	QUANTITYINSTOCK
hbk	C2	H	20
ebk	A0	P	15

```

SQL> CLEAR COLUMNS
SQL>
SQL> -- 1c -----
SQL> CREATE OR REPLACE VIEW order_details
2  AS
3      SELECT      loginName ,
4                  firstName ,
5                  lastName ,
6                  orderCartInfo.orderCartId ,
7                  SUM(qtyOrdered * orderPrice) AS TotalPrice
8  FROM
9  INNER JOIN      orderCartInfo
10 ON              orderCartInfo.customerID = customerInfo.loginName
11 INNER JOIN      lineItems
12 ON              lineItems.orderCartId=orderCartInfo.orderCartId
13 GROUP BY       orderCartInfo.orderCartId ,
14                loginName ,
15                firstName ,
16                lastName;

```

View created.

```

SQL>
SQL> COLUMN FIRSTNAME FORMAT A10
SQL> COLUMN LASTNAME FORMAT A10
SQL> COLUMN ORDERCARTID FORMAT A12
SQL> SELECT * FROM order_details;

```

LOGINNAME	FIRSTNAME	LASTNAME	ORDERCARTID	TOTALPRICE
shkeandco	Mary	Folio	5	2.99

astrab	Percy	Byron	4	14.97
shkeandco	Mary	Folio	1	8.97
dramab	Ophelia	Elsinore	2	11.97
astrab	Percy	Byron	8	7.98
astrab	Percy	Byron	3	73.83
shkeandco	Mary	Folio	6	13.98
dramab	Ophelia	Elsinore	7	1.99

8 rows selected.

SQL> CLEAR COLUMNS

SQL>

SQL> — 1d ———

SQL> COLUMN FIRSTNAME FORMAT A10

SQL> COLUMN LASTNAME FORMAT A10

SQL> CREATE OR REPLACE VIEW customer_order_details

```

2 AS
3     SELECT      loginName ,
4                 firstName ,
5                 lastName ,
6                 SUM(totalPrice) AS totalPrice
7     FROM        order_details
8     GROUP BY    loginName ,
9                 firstName ,
10                lastName;
```

View created.

SQL>

SQL> SELECT * FROM customer_order_details;

LOGINNAME	FIRSTNAME	LASTNAME	TOTALPRICE
shkeandco	Mary	Folio	25.94
astrab	Percy	Byron	96.78
dramab	Ophelia	Elsinore	13.96

SQL> CLEAR COLUMNS

SQL>

SQL> — 1e ———

SQL>

SQL> CREATE OR REPLACE VIEW carts_per_customer

```

2 AS
3     SELECT      loginName ,
4                 COUNT(orderCartId) AS numOrders
5     FROM        customerInfo
6     INNER JOIN  orderCartInfo
7     ON          customerInfo.loginName = orderCartInfo.customerId
8     GROUP BY    loginName;
```

View created.

```
SQL>
SQL> SELECT  loginName ,
      2      CASE
      3          WHEN    numOrders <= 2
      4              THEN    'BR-1 satisfied '
      5              ELSE    'BR-1 violated '
      6      END AS BR_1
      7 FROM    carts_per_customer ;
```

LOGINNAME	BR_1
dramab	BR-1 satisfied
shkeandco	BR-1 violated
astrab	BR-1 violated

```
SQL>
SQL> -- 1f -----
SQL>
SQL> SELECT  itemNum ,
      2      itemColor ,
      3      itemSize
      4 FROM
      5     (
      6         SELECT  itemNum ,
      7             itemColor ,
      8             itemSize ,
      9             CASE
     10             WHEN    COUNT_COL_SIZE <= 1
     11                 THEN    'BR-2 satisfied '
     12                 ELSE    'BR-2 violated '
     13             END AS BR_2
     14         FROM
     15             (
     16                 SELECT  itemNum ,
     17                     itemColor ,
     18                     itemSize ,
     19                     COUNT(itemColor) AS COUNT_COL_SIZE
     20                 FROM    inventoryItem
     21                 GROUP BY itemNum, itemColor , itemSize
     22             )
     23     )
     24 WHERE BR_2='BR-2 violated ';
```

ITEMNUM	ITEMCOLOR	ITEMSIZE
C2	black	3

SQL>

green

4

SQL>

SQL> — 1g —————

SQL>

```
SQL> SHOW ERRORS TRIGGER check_item_value
```

No errors.

```
SQL> SET PAGES 999;
```

```
SQL> SET LONG 999999;
```

```
SQL> COLUMN "Picture" FORMAT A7;
```

SQL>

```
SQL> CREATE OR REPLACE TRIGGER check_item_value
```

2 AFTER INSERT OR UPDATE ON itemType

3 DECLARE

```
4         min_price      float DEFAULT 0;
```

```
5         max_price         float DEFAULT 0;
```

6 BEGIN

```
7      SELECT  MIN(price) INTO min_price
```

```
8      FROM      itemType;
```

9

```
10      SELECT  MAX(price) INTO max_price
```

```

11      FROM      itemType;

```

12

```
13      IF max_price > min_price * 4
```

14 THEN

```
15         RAISE_APPLICATION_ERROR(num => -20000,  
16                                msg => 'Price too high');
```

```
msg => 'Price too high');
```

17 END IF;

18 END check_i

19 /

Trigger created.

SQL>

SQL> — Initial state of table

```
SQL> SELECT * FROM itemType;
```

ITEMNUM	NAME	PICTURE	PRICE	BELONGSTO
A0	The Anarchy of Mask	***	10.99	P
A1	The Butler Did	—	11.99	C
A2	The Abolished Man	==	15.99	SF
A3	Lyrical Bullets	+ + +	20.99	P
B1	The Postlude	=\=\=	34.99	P
C1	The August of Guns	**--	10.99	H
C2	The Expectant Mirror	^ - ^	12.99	H

7 rows selected.

```

SQL>
SQL> — Valid insert
SQL> INSERT into itemType VALUES ('D4', 'The 5th thing', '++--', 10.88, 'P');

1 row created.

SQL>
SQL> — Invalid inserts
SQL> INSERT into itemType VALUES ('C3', 'The thing', '++--', 500.99, 'P');
INSERT into itemType VALUES ('C3', 'The thing', '++--', 500.99, 'P')
      *
ERROR at line 1:
ORA-20000: Price too high
ORA-06512: at "MBAX4SD3.CHECK_ITEM_VALUE", line 13
ORA-04088: error during execution of trigger 'MBAX4SD3.CHECK_ITEM_VALUE'

SQL> INSERT into itemType VALUES ('D1', 'The other thing', '++--', 43.97, 'P');
INSERT into itemType VALUES ('D1', 'The other thing', '++--', 43.97, 'P')
      *
ERROR at line 1:
ORA-20000: Price too high
ORA-06512: at "MBAX4SD3.CHECK_ITEM_VALUE", line 13
ORA-04088: error during execution of trigger 'MBAX4SD3.CHECK_ITEM_VALUE'

SQL> INSERT into itemType VALUES ('D2', 'The 3rd thing', '++--', 60.99, 'P');
INSERT into itemType VALUES ('D2', 'The 3rd thing', '++--', 60.99, 'P')
      *
ERROR at line 1:
ORA-20000: Price too high
ORA-06512: at "MBAX4SD3.CHECK_ITEM_VALUE", line 13
ORA-04088: error during execution of trigger 'MBAX4SD3.CHECK_ITEM_VALUE'

SQL> INSERT into itemType VALUES ('D3', 'The 4th thing', '++--', 90.88, 'P');
INSERT into itemType VALUES ('D3', 'The 4th thing', '++--', 90.88, 'P')
      *
ERROR at line 1:
ORA-20000: Price too high
ORA-06512: at "MBAX4SD3.CHECK_ITEM_VALUE", line 13
ORA-04088: error during execution of trigger 'MBAX4SD3.CHECK_ITEM_VALUE'

SQL>
SQL> — State after insertions
SQL> SELECT * FROM itemType;

```

ITEMNUM	NAME	PICTURE	PRICE BELONGSTO
---------	------	---------	-----------------

A0	The Anarchy of Mask	***	10.99	P
A1	The Butler Did	—	11.99	C
A2	The Abolished Man	==	15.99	SF
A3	Lyrical Bullets	+—+	20.99	P
B1	The Postlude	=\=\=	34.99	P
C1	The August of Guns	**—	10.99	H
C2	The Expectant Mirror	^—^	12.99	H
D4	The 5th thing	++—+	10.88	P

8 rows selected.

SQL>

SQL> — Valid update

SQL> UPDATE itemType SET price=10.15 WHERE itemNum='A1';

1 row updated.

SQL>

SQL> — Invalid updates

SQL> UPDATE itemType SET price=61.56 WHERE itemNum='A1';

UPDATE itemType SET price=61.56 WHERE itemNum='A1'

*

ERROR at line 1:

ORA-20000: Price too high

ORA-06512: at "MBAX4SD3.CHECK_ITEM_VALUE", line 13

ORA-04088: error during execution of trigger 'MBAX4SD3.CHECK_ITEM_VALUE'

SQL> UPDATE itemType SET price=86.56 WHERE itemNum='A2';

UPDATE itemType SET price=86.56 WHERE itemNum='A2'

*

ERROR at line 1:

ORA-20000: Price too high

ORA-06512: at "MBAX4SD3.CHECK_ITEM_VALUE", line 13

ORA-04088: error during execution of trigger 'MBAX4SD3.CHECK_ITEM_VALUE'

SQL> UPDATE itemType SET price=150.23 WHERE itemNum='A3';

UPDATE itemType SET price=150.23 WHERE itemNum='A3'

*

ERROR at line 1:

ORA-20000: Price too high

ORA-06512: at "MBAX4SD3.CHECK_ITEM_VALUE", line 13

ORA-04088: error during execution of trigger 'MBAX4SD3.CHECK_ITEM_VALUE'

SQL>

SQL> — State after updating

```
SQL> SELECT * FROM itemType;
```

ITEMNUM	NAME	PICTURE	PRICE	BELONGSTO
A0	The Anarchy of Mask	***	10.99	P
A1	The Butler Did	—	10.15	C
A2	The Abolished Man	==	15.99	SF
A3	Lyrical Bullets	+—+	20.99	P
B1	The Postlude	=\=\=	34.99	P
C1	The August of Guns	**--	10.99	H
C2	The Expectant Mirror	^—^	12.99	H
D4	The 5th thing	++—+	10.88	P

```
8 rows selected.
```

```
SQL>
```

```
SQL>
```

```
SQL>
```

```
SQL> SPPOOL OFF
```