

# Assignment

Course unit: COMP23111

Academic year: 2017-2018

Exercise number: EX04

Student ID: 10407315

School of Computer Science

2018.12.4

```
SQL> -- sends everything to EX04-10407315.log
SQL> -- here you can set the SQL*Plus parameters, such as column width,
SQL> -- that will allow the script to produce readable answers in the spool
SQL> -- file
SQL>
SQL> -- [body]
SQL> start /opt/info/courses/COMP23111/create-Eclectic-Ecommerce-tables.sql
SQL> create table customerInfo (
2      loginName varchar(10) primary key,
3      password  varchar(15) not null,
4      firstName varchar(20) not null,
5      lastName  varchar(20) not null,
6      email     varchar(20) not null,
7      cStreet   varchar(20) not null,
8      cState    varchar(2)  not null,
9      cZip      varchar(6)  not null
10 );
```

Table created.

```
SQL>
SQL> create table orderCartInfo (
2      orderCartId varchar(10) primary key,
3      oType      varchar(15) not null check (oType in ('S', 'O')),
4      oYear      int        not null,
5      oMonth     int        not null,
6      oDay       int        not null,
7      customerID varchar(10) not null
8      references customerInfo(loginName) on delete cascade
9 );
```

Table created.

SQL>

```
SQL> create table category (  
2     categoryId varchar(10) primary key,  
3     description varchar(50)  
4 );
```

Table created.

SQL>

```
SQL> create table itemType (  
2     itemNum  varchar(10) primary key,  
3     name     varchar(20) not null,  
4     picture  varchar(50),  
5     price   float  not null,  
6     belongsTo varchar(10) not null  
7           references category(categoryId) on delete cascade  
8 );
```

Table created.

SQL>

```
SQL> create table inventoryItem (  
2     itemNum  varchar(10) not null  
3           references itemType(itemNum) on delete cascade,  
4     code     varchar(20) not null,  
5     qtyInstock float  not null,  
6     itemColor varchar(15),  
7     itemSize  float,  
8     primary key (itemNum, code)  
9 );
```

Table created.

SQL>

# University of Manchester

---

```
SQL> create table lineItems (  
2      orderCartId varchar(10) not null  
3          references orderCartInfo(orderCartId) on delete cascade,  
4      itemNum varchar(10) not null  
5          references itemType(itemNum) on delete cascade,  
6      code      varchar(20) not null,  
7      qtyOrdered float      not null,  
8      orderPrice float      not null,  
9      primary key (orderCartId, itemNum, code)  
10 );
```

Table created.

SQL>

```
SQL> start /opt/info/courses/COMP23111/populate-Eclectic-Ecommerce-tables.sql
```

SQL>

```
SQL> insert into customerInfo values ('choizeznyc', 'KJH*GS9', 'John', 'Booker', 'jb@choizez-nyc.com',  
'202 East 78th', 'NY', '10021');
```

1 row created.

```
SQL> insert into customerInfo values ('shkeandco', 'JHKG*!9S', 'Mary', 'Folio', 'mf@shkeandco.com', '993  
Lexington Ave', 'NY', '10065');
```

1 row created.

```
SQL> insert into customerInfo values ('dramab', 'KG454*56Sk1', 'Ophelia', 'Elsinore', 'oe@dramab.com',  
'502 West 40th', 'NY', '10018');
```

1 row created.

```
SQL> insert into customerInfo values ('astrab', '4!2!3kjn3e', 'Percy', 'Byron', 'pb@astrab.com', '92  
31st', 'NY', '11106');
```

1 row created.

SQL>

SQL> insert into orderCartInfo values ('1', 'S', 2015, 12, 23, 'shkeandco');

1 row created.

SQL> insert into orderCartInfo values ('2', 'S', 2015, 08, 03, 'dramab');

1 row created.

SQL> insert into orderCartInfo values ('3', 'S', 2015, 12, 23, 'astrab');

1 row created.

SQL> insert into orderCartInfo values ('4', 'O', 2015, 08, 03, 'astrab');

1 row created.

SQL> insert into orderCartInfo values ('5', 'S', 2015, 09, 18, 'shkeandco');

1 row created.

SQL> insert into orderCartInfo values ('6', 'S', 2015, 06, 03, 'shkeandco');

1 row created.

SQL> insert into orderCartInfo values ('7', 'S', 2015, 12, 12, 'dramab');

1 row created.

SQL> insert into orderCartInfo values ('8', 'O', 2015, 08, 28, 'astrab');

1 row created.

SQL>

SQL> insert into category values ('F', 'fiction');

1 row created.

SQL> insert into category values ('H', 'history');

1 row created.

SQL> insert into category values ('P', 'poetry');

1 row created.

SQL> insert into category values ('C', 'crime');

1 row created.

SQL> insert into category values ('SF', 'sci-fi');

1 row created.

SQL>

SQL> insert into itemType values ('A0', 'The Anarchy of Mask', '\*\*\*', 10.99, 'P');

1 row created.

SQL> insert into itemType values ('A1', 'The Butler Did', '---', 11.99, 'C');

1 row created.

SQL> insert into itemType values ('A2', 'The Abolished Man', '===', 15.99, 'SF');

1 row created.

```
SQL> insert into itemType values ('A3', 'Lyrical Bullets', '+++', 20.99, 'P');
```

1 row created.

```
SQL> insert into itemType values ('B1', 'The Postlude', '=\\=\\=', 34.99, 'P');
```

1 row created.

```
SQL> insert into itemType values ('C1', 'The August of Guns', '**--', 10.99, 'H');
```

1 row created.

```
SQL> insert into itemType values ('C2', 'The Expectant Mirror', '^--^', 12.99, 'H');
```

1 row created.

```
SQL>
```

```
SQL> insert into inventoryItem values ('A0', 'pbk', 30, 'black', 3.0);
```

1 row created.

```
SQL> insert into inventoryItem values ('B1', 'hbk', 50, 'white', 4.0);
```

1 row created.

```
SQL> insert into inventoryItem values ('A0', 'hbk', 25, 'green', 4.0);
```

1 row created.

```
SQL> insert into inventoryItem values ('C2', 'hbk', 20, 'black', 3.0);
```

1 row created.

# University of Manchester

---

```
SQL> insert into inventoryItem values ('C1', 'ebk', 70, 'black', 4.0);
```

```
1 row created.
```

```
SQL> insert into inventoryItem values ('A3', 'pbk', 30, 'black', 2.0);
```

```
1 row created.
```

```
SQL> insert into inventoryItem values ('A2', 'pbk', 50, 'green', 3.0);
```

```
1 row created.
```

```
SQL> insert into inventoryItem values ('A0', 'ebk', 15, 'green', 4.0);
```

```
1 row created.
```

```
SQL> insert into inventoryItem values ('C2', 'pbk', 30, 'black', 3.0);
```

```
1 row created.
```

```
SQL>
```

```
SQL> insert into lineItems values ('1', 'A0', 'pbk', 1, 2.99);
```

```
1 row created.
```

```
SQL> insert into lineItems values ('8', 'A3', 'pbk', 2, 3.99);
```

```
1 row created.
```

```
SQL> insert into lineItems values ('3', 'A2', 'pbk', 5, 5.99);
```

```
1 row created.
```

```
SQL> insert into lineItems values ('2', 'A2', 'pbk', 3, 3.99);
```



1 row created.

```
SQL> insert into lineItems values ('1', 'C1', 'ebk', 2, 2.99);
```

1 row created.

```
SQL> insert into lineItems values ('3', 'C2', 'hbk', 4, 2.99);
```

1 row created.

```
SQL> insert into lineItems values ('3', 'A3', 'pbk', 8, 3.99);
```

1 row created.

```
SQL> insert into lineItems values ('4', 'B1', 'hbk', 3, 4.99);
```

1 row created.

```
SQL> insert into lineItems values ('6', 'B1', 'hbk', 2, 6.99);
```

1 row created.

```
SQL> insert into lineItems values ('5', 'A2', 'pbk', 1, 2.99);
```

1 row created.

```
SQL> insert into lineItems values ('7', 'A1', 'pbk', 1, 1.99);
```

1 row created.

```
SQL>
```

```
SQL> -- (a)
```

```
SQL> create or replace view cast_cart as
```

```
2  select firstName, lastName
3  from customerInfo
4  where customerInfo.loginName in(
5      select customerID
6      from orderCartInfo
7  );
```

View created.

```
SQL> select * from cast_cart;
```

FIRSTNAME	LASTNAME
Mary	Folio
Ophelia	Elsinore
Percy	Byron

```
SQL> drop view cast_cart;
```

View dropped.

```
SQL>
```

```
SQL> -- (b)
```

```
SQL> create or replace view reorder_needed as
```

```
2  select inventoryItem.code, inventoryItem.itemNum, itemType.belongsTo, inventoryItem.qtyInstock
3  from inventoryItem, itemType
4  where inventoryItem.qtyInstock<25
5  and inventoryItem.itemNum = itemType.itemNum;
```

View created.

```
SQL> select * from reorder_needed;
```

CODE	ITEMNUM	BELONGSTO	QTYINSTOCK
------	---------	-----------	------------

---

# University of Manchester

---

hbk	C2	H	20
ebk	A0	P	15

```
SQL> drop view reorder_needed;
```

View dropped.

```
SQL>
```

```
SQL> -- (c)
```

```
SQL> create or replace view id_ttl as
```

```
2   select orderCartInfo.orderCartId, sum(lineItems.orderPrice) as total_price
3   from orderCartInfo, lineItems
4   where orderCartInfo.orderCartId=lineItems.orderCartId
5   group by orderCartInfo.orderCartId;
```

View created.

```
SQL> create or replace view ord_cust_ttl as
```

```
2   select customerInfo.loginName, customerInfo.firstName,
3   customerInfo.lastName, id_ttl.orderCartId, id_ttl.total_price
4   from customerInfo, id_ttl, orderCartInfo
5   where customerInfo.loginName=orderCartInfo.customerID
6   and orderCartInfo.orderCartId=id_ttl.orderCartId;
```

View created.

```
SQL> select * from ord_cust_ttl;
```

LOGINNAME	FIRSTNAME	LASTNAME	ORDERCARTI	TOTAL_PRICE
shkeandco	Mary	Folio	1	5.98
astrab	Percy	Byron	3	12.97
shkeandco	Mary	Folio	6	6.99

## University of Manchester

---

astrab	Percy	Byron	8	3.99
shkeandco	Mary	Folio	5	2.99
dramab	Ophelia	Elsinore	7	1.99
dramab	Ophelia	Elsinore	2	3.99
astrab	Percy	Byron	4	4.99

8 rows selected.

```
SQL> drop view id_ttl;
```

View dropped.

```
SQL> drop view ord_cust_ttl;
```

View dropped.

```
SQL>
```

```
SQL> -- (d)
```

```
SQL> -- total_price for John Brooker with loginName choizeznyc does not even have a cart.
```

```
SQL> create or replace view id_ttl as
```

```
2  select customerInfo.loginName, sum(lineItems.orderPrice)
3  as total_price
4  from customerInfo
5  left join orderCartInfo
6  on customerInfo.loginName=orderCartInfo.customerID
7  left join lineItems
8  on orderCartInfo.orderCartId=lineItems.orderCartId
9  group by customerInfo.loginName;
```

View created.

```
SQL> create or replace view ord_cust_ttl as
```

```
2  select customerInfo.loginName, customerInfo.firstName,
3  customerInfo.lastName, id_ttl.total_price
```

---

# University of Manchester

---

```
4 from customerInfo, id_ttl where customerInfo.loginName=id_ttl.loginName;
```

View created.

```
SQL> select * from ord_cust_ttl;
```

LOGINNAME	FIRSTNAME	LASTNAME	TOTAL_PRICE
dramab	Ophelia	Elsinore	5.98
choizeznyc	John	Booker	
shkeandco	Mary	Folio	15.96
astrab	Percy	Byron	21.95

```
SQL> drop view id_ttl;
```

View dropped.

```
SQL> drop view ord_cust_ttl;
```

View dropped.

```
SQL>
```

```
SQL> -- (e)
```

```
SQL> -- case for John Brooker is not printed as he or she does not even has a cart.
```

```
SQL> create or replace view ca_cu as
```

```
2 select customerInfo.loginName, count(orderCartInfo.orderCartId) as cnt_id
3 from customerInfo, orderCartInfo
4 where customerInfo.loginName=orderCartInfo.customerID
5 group by customerInfo.loginName;
```

View created.

```
SQL> create or replace view ca_per_cu as
```

```
2 select sum(cnt_id)/count(distinct loginName)
```

```
3   as res
4   from ca_cu;
```

View created.

```
SQL> select loginName, case
2   when cnt_id<=2 then 'BR-2 satisfied'
3   when cnt_id>2 then 'BR-2 violated'
4   end cnt_id
5   from ca_cu;
```

```
LOGINNAME  CNT_ID
```

```
-----
dramab      BR-2 satisfied
shkeandco   BR-2 violated
astrab      BR-2 violated
```

```
SQL>
SQL> -- (f)
SQL> -- select *
SQL> -- from (
SQL> --   with temp as(
SQL> --       select count(*) as count, itemNum, itemSize, itemColor
SQL> --       from inventoryItem
SQL> --       group by itemNum, itemSize, itemColor having count(*)>0
SQL> --   )select itemNum, itemSize, itemColor, case
SQL> --       when count>1 then 'BR-2 violated'
SQL> --       when count=1 then 'BR-2 satisfied'
SQL> --   end count
SQL> --   from temp
SQL> -- )where count='BR-2 violated';
SQL>
SQL> select *
2   from(
```

```
3  select itemNum, itemSize, itemColor, case
4  when count>1 then 'BR-2 violated'
5  when count=1 then 'BR-2 satisfied'
6  end count
7  from(
8    select count(*) as count, itemNum, itemSize, itemColor
9    from inventoryItem
10   group by itemNum, itemSize, itemColor having count(*)>0
11  )
12 )where count='BR-2 violated';
```

ITEMNUM	ITEMSIZE	ITEMCOLOR	COUNT
A0	4	green	BR-2 violated
C2	3	black	BR-2 violated

SQL>

SQL>

SQL> -- (g)

SQL> create or replace trigger temp

```
2  before insert or update on itemType
3  for each row
4  declare
5    tmp_val float(126);
6  PRAGMA AUTONOMOUS_TRANSACTION;
7  begin
8    select min(price) into tmp_val from itemType;
9    if :new.price>4*tmp_val then
10      raise_application_error(-20000,'ERROR TRIGGERED!');
11    end if;
12  end;
13 /
```

Trigger created.

# University of Manchester

---

SQL>

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> insert into itemType (price) values(170);

insert into itemType (price) values(170)

\*

ERROR at line 1:

ORA-20000: ERROR TRIGGERED!

ORA-06512: at "Z26937HC.TEMP", line 7

ORA-04088: error during execution of trigger 'Z26937HC.TEMP'

SQL> col name format a25;

SQL> col picture format a7;

SQL> select \* from itemType;

ITEMNUM	NAME	PICTURE	PRICE	BELONGSTO
A0	The Anarchy of Mask	***	10.99	P
A1	The Butler Did	---	11.99	C



# University of Manchester

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> update itemType set price=170 where price=15.99;

update itemType set price=170 where price=15.99

\*

ERROR at line 1:

ORA-20000: ERROR TRIGGERED!

ORA-06512: at "Z26937HC.TEMP", line 7

ORA-04088: error during execution of trigger 'Z26937HC.TEMP'

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>

SQL> -- [close]

SQL>

SQL> start /opt/info/courses/COMP23111/drop-Eclectic-Ecommerce-tables.sql

SQL> drop table lineItems;

Table dropped.

SQL> drop table inventoryItem;

Table dropped.

SQL> drop table itemType;

Table dropped.

SQL> drop table category;

Table dropped.

SQL> drop table orderCartInfo;

Table dropped.

SQL> drop table customerInfo;

Table dropped.

SQL>

SQL> -- [footer]

SQL> --

SQL> -- End of Exercise 04 by Haorui Chen

SQL> SPPOOL OFF