## Lab Assignment # 3

## Data Retrieval Language, SELECT from Several Logical Linked Tables

**Charu Bisht**

In this lab you will work with SELECT statements against several logical linked tables.

**Create the tables for this lab**

------------------Create tables START COPY----------------------------------------------

CREATE TABLE customer(

cust\_id NUMBER(6) PRIMARY KEY,

first\_name VARCHAR2(20),

last\_name VARCHAR2(25),

address VARCHAR2(30),

zip\_code VARCHAR2(8),

city VARCHAR2(20),

area\_code VARCHAR2(6),

telephone VARCHAR2(12));

CREATE TABLE cust\_order(

ord\_id NUMBER(9) PRIMARY KEY,

cust\_id REFERENCES customer(cust\_id),

order\_date DATE);

CREATE TABLE prod\_group(

group\_id NUMBER(4) PRIMARY KEY,

group\_name VARCHAR2(30));

CREATE TABLE product(

prod\_id NUMBER(8) PRIMARY KEY,

group\_id REFERENCES prod\_group(group\_id),

prod\_name VARCHAR2(25),

price NUMBER(9,2));

CREATE TABLE cart(

row\_id NUMBER(9) PRIMARY KEY,

ord\_id REFERENCES cust\_order(ord\_id),

prod\_id REFERENCES product(prod\_id),

quantity NUMBER(6));

CREATE TABLE prod\_pict(

pict\_id NUMBER(9) PRIMARY KEY,

prod\_id REFERENCES product(prod\_id),

file\_type VARCHAR2(5),

path VARCHAR2(80),

width NUMBER(4),

height NUMBER(4));

------------------Create tables END COPY------------------------------------------------

**Now we have a table structure representing sales activities. If you look at the next page you will see a data model of the tables with primary- and foreign keys.**

**Data model**

CUSTOMER

# cust\_id

\* first\_name

\* last\_name

\* address

\* zip

o phone

CUST\_ORDER

# ord\_id

(#)cust\_id

\* order\_date

CART

# row\_id

(#)ord\_id

(#)prod\_id

\* quantity

PRODUCT

# prod\_id

(#)group\_id

\* prod\_name

\* price

PROD\_GROUP

# group\_id

\* group\_name

PROD\_PICT

# pict\_id

(#)prod\_id

\* file\_type

\* path

\* width

\* height

**Explanation of notation**

# = Primary key

(#) = Foreign key

\* = Mandatory (must contain a value => NOT NULL)

o = Optional (must not contain a value can be NULL)

**Next step is to fill the tables with data. Do that by copy and paste the following into SQL Live, and hit Run.**

------------------Fill the tables with data START COPY----------------------------------

INSERT INTO customer VALUES(1,'olof','andersson','box144','79100','falun','023','225478');

INSERT INTO customer VALUES(2,'maria','andersson','storgatan 23',79123,'falun','023','445599');

INSERT INTO customer VALUES(3,'tomas','kvist','box1','54784','gagnef','0246','11122');

INSERT INTO customer VALUES(4,'hans','rosenboll','sommarvagen 36','78458','borlange','0243','228869');

INSERT INTO customer VALUES(5,'yvette','porpoix','sadelgatan 10','79100','falun','023','147858');

INSERT INTO customer VALUES(6,'gustav','moller','box33','78547','gustafs','0243','122099');

INSERT INTO customer VALUES(7,'zoltan','habbervic','paradisvagen 12','78523','borlange','0243','45877');

INSERT INTO customer VALUES(8,'lena','larsson','sandgatan 13','73100','sater','0225','43251');

INSERT INTO customer VALUES(9,'ollas','bullas','korkhuvudvagen 1','79100','falun','023','11477');

INSERT INTO customer VALUES(10,'roger','nyberg','soldatvagen 25','79100','falun','023','225499');

INSERT INTO cust\_order VALUES(100,1,TO\_DATE('2001-02-14','YYYY-MM-DD'));

INSERT INTO cust\_order VALUES(101,4,TO\_DATE('2001-02-14','YYYY-MM-DD'));

INSERT INTO cust\_order VALUES(289,4,TO\_DATE('2003-03-04','YYYY-MM-DD'));

INSERT INTO cust\_order VALUES(125,2,TO\_DATE('2001-05-24','YYYY-MM-DD'));

INSERT INTO cust\_order VALUES(147,3,TO\_DATE('2001-12-11','YYYY-MM-DD'));

INSERT INTO cust\_order VALUES(152,5,TO\_DATE('2001-12-15','YYYY-MM-DD'));

INSERT INTO cust\_order VALUES(458,6,TO\_DATE('2004-05-08','YYYY-MM-DD'));

INSERT INTO cust\_order VALUES(489,6,TO\_DATE('2004-06-10','YYYY-MM-DD'));

INSERT INTO cust\_order VALUES(324,10,TO\_DATE('2003-08-22','YYYY-MM-DD'));

INSERT INTO cust\_order VALUES(198,9,TO\_DATE('2002-01-12','YYYY-MM-DD'));

INSERT INTO cust\_order VALUES(348,1,TO\_DATE('2004-07-17','YYYY-MM-DD'));

INSERT INTO prod\_group VALUES(1,'beard care');

INSERT INTO prod\_group VALUES(2,'hunting');

INSERT INTO prod\_group VALUES(3,'farmhouse');

INSERT INTO prod\_group VALUES(4,'leisure');

INSERT INTO product VALUES(1434,1,'trimmer deluxe',189.50);

INSERT INTO product VALUES(1724,1,'fungicides',198.5);

INSERT INTO product VALUES(113,2,'hatchet',795);

INSERT INTO product VALUES(1447,2,'knife',349.5);

INSERT INTO product VALUES(5896,3,'pig feed',240);

INSERT INTO product VALUES(5542,3,'potato fertilizer',128);

INSERT INTO product VALUES(1333,4,'dartboard',49.50);

INSERT INTO product VALUES(1888,4,'peasant trap',788.50);

INSERT INTO product VALUES(1141,4,'hammock',181.50);

INSERT INTO prod\_pict VALUES(1,1434,'jpg','/images/1/',480,640);

INSERT INTO prod\_pict VALUES(2,113,'jpg','/images/2/',480,640);

INSERT INTO prod\_pict VALUES(3,5896,'jpg','/images/3/',480,640);

INSERT INTO prod\_pict VALUES(4,1888,'gif','/images/4/',480,640);

INSERT INTO cart VALUES(1,100,1141,1);

INSERT INTO cart VALUES(2,101,1434,3);

INSERT INTO cart VALUES(3,101,1724,4);

INSERT INTO cart VALUES(4,289,1434,1);

INSERT INTO cart VALUES(5,289,1724,5);

INSERT INTO cart VALUES(6,125,1333,1);

INSERT INTO cart VALUES(7,125,1141,1);

INSERT INTO cart VALUES(8,147,5896,4);

INSERT INTO cart VALUES(9,147,5542,4);

INSERT INTO cart VALUES(10,152,113,2);

INSERT INTO cart VALUES(11,458,5896,3);

INSERT INTO cart VALUES(12,458,1447,1);

INSERT INTO cart VALUES(13,489,5542,3);

INSERT INTO cart VALUES(14,324,113,3);

INSERT INTO cart VALUES(15,324,1447,3);

INSERT INTO cart VALUES(16,324,1888,1);

INSERT INTO cart VALUES(17,198,1141,7);

INSERT INTO cart VALUES(18,348,113,3);

COMMIT;

------------------ Fill the tables with data END COPY-----------------------------------

**Task 1:**

Show **cust\_id**, **first\_name**, **last\_name** and the **number** of customer orders that each customer has in the system.

***Correct answer:***

CUST\_ID FIRST\_NAME LAST\_NAME NUMBER\_OF\_ORDERS

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

1 olof andersson 2

2 maria andersson 1

3 tomas kvist 1

4 hans rosenboll 2

5 yvette porpoix 1

6 gustav moller 2

9 ollas bullas 1

10 roger nyberg 1

**Solution 1:**

select c.cust\_id, first\_name, last\_name, count(co.ord\_id) as number\_of\_orders from customer c, cust\_order co where c.cust\_id = co.cust\_id group by c.cust\_id, first\_name, last\_name order by c.cust\_id;

**Task 2:**

Show **cust\_id**, **first\_name**, **last\_name** for those customers who have bought products that belong to the product groups: 'farmhouse' and 'beard care'. **Solve this task by using nested search** (i.e., using sub queries)

***Correct answer:***

CUST\_ID FIRST\_NAME LAST\_NAME

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

3 tomas kvist

4 hans rosenboll

6 gustav moller

**Solution 2:**

select cust\_id, first\_name, last\_name from customer where cust\_id in

(select cust\_id from cust\_order where ord\_id in  
(select ord\_id from cart where prod\_id in

(select prod\_id from product where group\_id in

(select group\_id from prod\_group where group\_name in

('farmhouse', 'beard care')))))order by cust\_id;

**Task 3:**

Show **cust\_id**, **first\_name**, **last\_name** for those customers who have bought products that belong to the product groups: 'farmhouse' and 'beard care'. **Solve this task by using join search** (i.e., using equi-join)

***Correct answer:***

CUST\_ID FIRST\_NAME LAST\_NAME

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

3 tomas kvist

4 hans rosenboll

6 gustav moller

**Solution 3:**

select c.cust\_id,first\_name,last\_name

from customer c, cust\_order co, cart ca, product p, prod\_group pg where c.cust\_id = co.cust\_id and co.ord\_id = ca.ord\_id and ca.prod\_id = p.prod\_id and p.group\_id = pg.group\_id and pg.group\_name in ('farmhouse','beard care') group by c.cust\_id,first\_name,last\_name

order by c.cust\_id;

**Task 4:**

Show **cust\_id**, **first\_name**, **last\_name** and the **total amount** that customers have shopped for.

***Correct answer:***

CUST\_ID FIRST\_NAME LAST\_NAME TOTAL

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

1 olof andersson 2566,5

2 maria andersson 231

3 tomas kvist 1472

4 hans rosenboll 2544,5

5 yvette porpoix 1590

6 gustav moller 1453,5

9 ollas bullas 1270,5

10 roger nyberg 4222

**Solution 4:**

select c.cust\_id, c.first\_name, c.last\_name, sum(ca.quantity\*p.price) as

TOTAL from customer c, cust\_order co, cart ca, product p

where c.cust\_id = co.cust\_id and co.ord\_id = ca.ord\_id and ca.prod\_id = p.prod\_id group by c.cust\_id, c.first\_name, c.last\_name order by c.cust\_id;

**Task 5:**

Show **cust\_id**, **first\_name**, **last\_name** and the **total amount** that customers have shopped for. Sort the result, so the customer with the highest total amount comes first.

Show the total amount without any decimals. **Hint!** The ROUND()Function.

***Correct Answer:***

CUST\_ID FIRST\_NAME LAST\_NAME TOTAL

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

10 roger nyberg 4222

1 olof andersson 2567

4 hans rosenboll 2545

5 yvette porpoix 1590

3 tomas kvist 1472

6 gustav möller 1454

9 ollas bullas 1271

2 maria andersson 231

**Solution 5:**

select c.cust\_id,first\_name,last\_name, round(sum(ca.quantity\*p.price)) as TOTAL from customer c, cust\_order co, cart ca, product p where c.cust\_id = co.cust\_id and co.ord\_id = ca.ord\_id and ca.prod\_id = p.prod\_id group by c.cust\_id,first\_name,last\_name order by TOTAL desc;

**Task 6:**

Same as in task 5, but show only customers who have a total over 1500.

***Correct answer:***

CUST\_ID FIRST\_NAME LAST\_NAME TOTAL

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

10 roger nyberg 4222

1 olof andersson 2567

4 hans rosenboll 2545

5 yvette porpoix 1590

**Solution 6:**

select \* from (select c.cust\_id, first\_name, last\_name, round(sum(price\* quantity)) as TOTAL from customer c, cust\_order co, cart ca, product p where c.cust\_id = co.cust\_id and co.ord\_id = ca.ord\_id and ca.prod\_id = p.prod\_id group by c.cust\_id, first\_name, last\_name order by TOTAL desc)

where TOTAL > 1500;

**Task 7:**

Show **first\_name** and **last\_name** **capitalized**, for those customers who **have no** orders.

***Correct answer:***

FIRST\_NAME LAST\_NAME

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

Zoltan Habbervic

Lena Larsson

**Solution 7:**

select initcap(first\_name) as FIRST\_NAME, initcap(last\_name) as LAST\_NAME

from customer where cust\_id not in (select cust\_id from cust\_order);

**Task 8:**

Show **group\_name** and the **price** for the most expensive product in that product group

***Correct answer:***

GROUP\_NAME MOST\_EXPENSIVE

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

farmhouse 240

leisure 788,5

hunting 795

beard care 198,5

**Solution 8:**

Since the question requires most expensive products, we display the data in descending order of price.

select group\_name, max(price) as most\_expensive from prod\_group pg, product p where pg.group\_id = p.group\_id group by group\_name order by most\_expensive desc;

**Task 9:**

Show **prod\_id**, and the **full path** to the image of the products that have an image. You get the full path by **concatenate** *path*, *pict\_id* and *file\_type*.

***Correct answer:***

PROD\_ID FULL\_PATH

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

1434 /images/1/1.jpg

113 /images/2/2.jpg

5896 /images/3/3.jpg

1888 /images/4/4.gif

**Solution 9:**

select prod\_id, (path || pict\_id ||'.'|| file\_type) as FULL\_PATH

from prod\_pict;

**Task 10:**

Show **first name** and **last name** for those customers who owns a customer order that was created during 2004.

***Correct answer:***

FIRST\_NAME LAST\_NAME

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

olof andersson

gustav moller

**Solution 10**

select first\_name, last\_name from customer where cust\_id in (select cust\_id from cust\_order where extract(year from order\_date) = 2004);

**Optional task:**

Show ord\_id and total order value for the most expensive customer order.

***Correct answer:***

ORD\_ID TOTAL

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

324 4222

**Solution:**

select ord\_id, sum(quantity\*price) as Total from cart c, product p where c.prod\_id = p.prod\_id group by ord\_id order by sum(quantity\*price) desc FETCH FIRST 1 ROWS ONLY ;