Jinwei Lin

Last Four Digits of Student ID#: 3389

CSCI 331-32

Submission Date: 12/22/2014
Due Date: 12/22/2014

Project Three

SQL commands to create table, insert and alter table

customer

```
Create table customer
(CID number primary key,
cfirst varchar(50) not null,
clast varchar(50) not null,
cemail varchar(50) not null,
caddress varchar(50) not null,
ccity varchar(20) not null,
cstate char(2) not null,
czip char(5) not null);

insert into customer
values('1','Jinwei','Lin','jinwei.lin8@gmail.com','44-35 Colden St
Apt5A','Flushing','NY','11355');

insert into customer
values(2,'Luke','Lee','luke.lee@gmail.com','3303 48th St Apt
3D','Sunnyside','NY','11104');

insert into customer
values('3','Binquan','Wang','lanyanzhanshen@gmail.com','11-17 46th
St','Flushing','NY','11354');

insert into customer
values('4','Bo','Li','boli@gmail.com','20-14 43th St Apt 7B','New York','NY','10017');

insert into customer
values('5','Corey','Malone','coreymalone@gmail.com','70 60th
St','Woodside','NY','11377');

insert into customer
values('6','Maria','Sharapova','mariasharapova@gmail.com','24-34 48th St','New
York','NY','10017');

insert into customer
values('7','Disha','Gupta','disha.gupta@gmail.com','29 E 34th St','New
York','NY','10001');

insert into customer
values('8','Adam','Najman','adam.najman@yahoo.com','18-23 64th St Apt 4A','New
York','NY','10065');

insert into customer
```

```
values('9','Jared','Sakamoto','jared.saka@yahoo.com','34-11 Union
St','Flushing','NY','11354');

insert into customer
values('10','Sau','Putt','sau.putt@hotmail.com','55-32 82 St Apt 2E','Jackson
Heights','NY','11370');

insert into customer
values('11','Kevin','Dugan','kevin.dugan@gmail.com','78-34 College
Point','Flushing','NY','11354');

insert into customer
values('12','Caitilin','King','caitilin.king@gmail.com','847 135th','Jamaica','NY','11412');

insert into customer
values('13','Jie','Zheng','jie.zheng@gmail.com','31-11 41th St Apt
2B','Wayne','NJ','07469');
```

Department

```
Create table department
(DEPTID varchar(4) primary key,
Dname varchar(25) not null);

insert into department
values('d1','Mac');

insert into department
values('d2','iphone');

insert into department
values('d3','ipad');

insert into department
values('d4','ipod');

insert into department
values('d5','apple tv');

insert into department
values('d6','Software');

insert into department
values('d7','Device Care');
```

Store

```
Create table store
```

```
(STOREID varchar(4) primary key,
Store_address varchar(25) not null,
Store_city varchar(25) not null,
Store_state char(2) not null,
Store_zip char(5) not null);

alter table store modify
(store_city null,
store_state null,
 store_zip null);

insert into store
values('s1','767 Fifth Avenue','New York','NY','10153');

insert into store
values('s2','1981 Broadway','New York','NY','10023');

insert into store
values('s3','45 Grand Central Terminal','New York','NY','10017');

insert into store
values('s4','401 W 14th Street','New York','NY','10014');

insert into store
values('s5','103 Prince Street','New York','NY','10012');

insert into store
values('s6','1 Garden State Plaza','Paramus','NJ','07652');

insert into store
values('s7','1400 Willowbrook Mall','Wayne','NJ','07470');

insert into store
values('s8','Web',null,null,null);
```

Staff

```
Create table staff
(SID varchar(4) primary key,
SFirst varchar(50) not null,
SLast varchar(50) not null);

insert into staff
values('sf1','Wen','Jiang');

insert into staff
values('sf2','Chris','Dugas');
```

```
insert into staff
values('sf3','Brian','Zitro');

insert into staff
values('sf4','Rainie','Yang');

insert into staff
values('sf5','Thom','Su');

insert into staff
values('sf6','Gilbert','Galindo');
```

Staff_Store

```
Create table staff_store
(SID varchar(4) not null,
Storeid varchar(4) not null);

alter table staff_store
add constraint staff_storeid primary key(sid, storeid);

insert into staff_store
values('sf1','s2');

insert into staff_store
values('sf1','s4');

insert into staff_store
values('sf2','s1');

insert into staff_store
values('sf3','s3');

insert into staff_store
values('sf4','s6');

insert into staff_store
values('sf4','s5');

insert into staff_store
values('sf5','s7');

insert into staff_store
values('sf6','s1');
```

Staff_dept

```
Create table staff_dept
(SID varchar(4) not null,
deptid varchar(4) not null);

alter table staff_dept
add constraint staff_deptid primary key(sid, deptid);

insert into staff_dept
values('sf1','d1');

insert into staff_dept
values('sf1','d2');

insert into staff_dept
values('sf1','d3');

insert into staff_dept
values('sf2','d2');

insert into staff_dept
values('sf3','d7');

insert into staff_dept
values('sf4','d4');

insert into staff_dept
values('sf5','d2');

insert into staff_dept
values('sf5','d3');

insert into staff_dept
values('sf6','d6');
```

Hardware

```
Create table hardware
(serial_number varchar(9) primary key,
 processor_type varchar(30),
 memory varchar(5),
 storage varchar(5));
alter table hardware modify
(processor_type varchar2(33));

/* serial_number contains E for computers, I for iphone, IP for ipad,
    S for software*/
```

```
insert into hardware
values('1129E401','1.4GHz dual-core Intel Core i5','4GB','128GB');

insert into hardware
values('1129E402','1.4GHz dual-core Intel Core i5','4GB','128GB');

insert into hardware
values('1129E403','1.4GHz dual-core Intel Core i5','4GB','256GB');

insert into hardware
values('1129E404','1.4GHz dual-core Intel Core i5','8GB','256GB');

insert into hardware
values('1129E405','1.7GHz dual-core Intel Core i7','8GB','128GB');

insert into hardware
values('1129E406','1.4GHz dual-core Intel Core i5','8GB','500GB');

insert into hardware
values('1129E407','2.7GHz quad-core Intel Core i5','8GB','1TB');

insert into hardware
values('1129I101','A8 chip with 64-bit architecture','null','16GB');

insert into hardware
values('1129I102','A8 chip with 64-bit architecture','null','64GB');

insert into hardware
values('1129I103','A8 chip with 64-bit architecture','null','16GB');

insert into hardware
values('1129IP101','A8X chip with 64-bit architecture','null','16GB');

insert into hardware
values('1129IP102','A8X chip with 64-bit architecture','null','64GB');

insert into hardware
values('1129IP103','A7 chip with 64-bit architecture','null','16GB');

insert into hardware
values('1129IP104','A7 chip with 64-bit architecture','null','64GB');

insert into hardware
values('1129S01','null','null','null');

insert into hardware
```

```
values('1129S02','null','null','null');

insert into hardware
values('1129S03','null','null','null');

insert into hardware
values('1129S04','null','null','null');

alter table hardware
drop column memory;

alter table hardware
add memory_gb number;

update hardware
set memory_gb = 8
where serial_number='1129E405' or  serial_number='1129E406' or
serial_number='1129E407' or  serial_number='1129E404';

update hardware
set memory_gb = 4
where serial_number='1129E401' or  serial_number='1129E402' or
serial_number='1129E403' or  serial_number='1129E404';
```

Product

```
Create table product
(PID varchar(4) primary key,
Pname varchar(25) not null,
Price number(10,2) not null,
Deptid varchar(4) not null,
serial_number varchar(9) not null,
foreign key (deptid) references department(deptid),
foreign key (serial_number) references hardware(serial_number));
alter table product modify
(pname varchar2(50));

insert into product
values('p1','Microsoft Office for Mac Home and Student 2011','139.95','d6','1129S01');

insert into product
values('p2','Microsoft Office for Mac Home and Business 2011','219.95','d6','1129S02');

insert into product
values('p3','Microsoft Office 365 University','79.95','d6','1129S03');

insert into product
```

```
values('p10','11-inch MacBook Air: 128GB','899.00','d1','1129E401');

insert into product
values('p11','13-inch MacBook Air: 128GB','999.00','d1','1129E402');

insert into product
values('p12','21.5-inch iMac: 500GB','1099.00','d1','1129E406');

insert into product
values('p13','21.5-inch iMac: 1TB','1299.00','d1','1129E407');

insert into product
values('p100','iPhone 6: 16GB ','649.00','d2','1129I101');

insert into product
values('p101','iPhone 6 Plus: 16GB ','749.00','d2','1129I103');

insert into product
values('p110','iPad Air 2: 64GB','599.00','d3','1129IP102');

insert into product
values('p111','iPad mini 3: 64GB','499.00','d3','1129IP104');
```

Transaction

```
Create table transaction
(tid varchar(5) primary key,
cid number not null,
pid varchar(5) not null,
storeid varchar(5) not null,
dop date not null,
price number(10,2) not null,
rating number,
method_of_purchase varchar(7) not null,
foreign key (cid) references customer(cid),
foreign key (pid) references product(pid),
foreign key (storeid) references store(storeid));

insert into transaction
values('t1','8','p1','s8','3/12/2014','139.95','3','website');

insert into transaction
values('t2','3','p3','s3','11/2/2013','79.95','5','store');

insert into transaction
values('t3','6','p3','s1','12/20/2014','79.95','5','store');
```

```
insert into transaction
values('t4','1','p3','s5','12/2/2014','79.95','4','phone');

insert into transaction
values('t5','12','p3','s7','10/7/2014','79.95','3','store');

insert into transaction
values('t6','7','p3','s8','8/11/2014','79.95','5','website');

insert into transaction
values('t7','4','p2','s8','8/11/2014','219.95','2','website');

insert into transaction
values('t8','2','p3','s8','8/2/2014','79.95','5','website');

insert into transaction
values('t9','5','p3','s8','4/16/2014','79.95','4','website');

insert into transaction
values('t10','6','p3','s6','4/16/2014','79.95','5','phone');

insert into transaction
values('t11','10','p3','s4','6/19/2014','79.95','4','store');

insert into transaction
values('t12','9','p3','s5','7/6/2014','79.95','5','store');

insert into transaction
values('t13','13','p13','s7','7/6/2014','1299',null,'store');

insert into transaction
values('t14','4','p100','s1','11/25/2014','649','5','store');

insert into transaction
values('t15','1','p11','s8','09/04/2014','999','5','website');

insert into transaction
values('t16','11','p111','s3','11/11/2014','499','3','store');

insert into transaction
values('t17','9','p12','s2','01/20/2014','1099',null,'phone');

insert into transaction
values('t18','5','p101','s1','07/18/2014','749',null,'store');

insert into transaction
```

```
values('t19','6','p110','s8','10/15/2014','599',null,'website');

insert into transaction
values('t20','4','p100','s6','5/2/2014','649',null,'store');

insert into transaction
values('t21','11','p12','s3','5/29/2014','1099',null,'store');

insert into transaction
values('t22','7','p111','s4','12/13/2014','499',null,'store');

insert into transaction
values('t23','3','p10','s5','09/10/2014','899',null,'store');

insert into transaction
values('t24','4','p101','s8','11/8/2014','749',null,'website');

insert into transaction
values('t25','1','p13','s1','06/20/2014','1299',null,'store');

insert into transaction
values('t26','7','p110','s6','04/30/2014','599',5,'store');

insert into transaction
values('t27','2','p101','s5','08/10/2014','749',null,'store');

insert into transaction
values('t28','12','p100','s7','10/16/2014','649',5,'store');

insert into transaction
values('t29','8','p100','s4','05/13/2014','649',5,'store');

insert into transaction
values('t30','6','p111','s8','08/25/2014','499',5,'website');
```

Store_product

```
create table store_product
(storeid varchar(4) not null,
 pid varchar(4) not null);
alter table store_product
add constraint pk_store_productid primary key(storeid, pid);

/* I only insert certain products to NJ stores to do question 4, is there a faster way to
insert all products id's to all stores id's?*/

insert into store_product
```

```
values('s6','p1');

insert into store_product
values('s6','p2');

insert into store_product
values('s6','p3');

insert into store_product
values('s6','p10');

insert into store_product
values('s6','p11');

insert into store_product
values('s6','p12');

insert into store_product
values('s6','p13');

insert into store_product
values('s7','p1');

insert into store_product
values('s7','p2');

insert into store_product
values('s7','p3');

insert into store_product
values('s7','p10');

insert into store_product
values('s7','p11');

insert into store_product
values('s7','p12');

insert into store_product
values('s7','p13');
```

Table Definition

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CID	NUMBER	22	-	-	1	-	-	-
	CFIRST	VARCHAR2	50	-	-	-	-	-	-
	CLAST	VARCHAR2	50	-	-	-	-	-	-
	CEMAIL	VARCHAR2	50	-	-	-	-	-	-
	CADDRESS	VARCHAR2	50	-	-	-	-	-	-
	CCITY	VARCHAR2	20	-	-	-	-	-	-
	CSTATE	CHAR	2	-	-	-	-	-	-
	CZIP	CHAR	5	-	-	-	-	-	-
									1 - 8

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPARTMENT	DEPTID	VARCHAR2	4	-	-	1	-	-	-
	DNAME	VARCHAR2	25	-	-	-	-	-	-
									1 - 2

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STORE	STOREID	VARCHAR2	4	-	-	1	-	-	-
	STORE_ADDRESS	VARCHAR2	25	-	-	-	-	-	-
	STORE_CITY	VARCHAR2	25	-	-	-	✓	-	-
	STORE_STATE	CHAR	2	-	-	-	✓	-	-
	STORE_ZIP	CHAR	5	-	-	-	✓	-	-
									1 - 5

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STAFF	SID	VARCHAR2	4	-	-	1	-	-	-
	SFIRST	VARCHAR2	50	-	-	-	-	-	-
	SLAST	VARCHAR2	50	-	-	-	-	-	-
									1 - 3

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STAFF_STORE	SID	VARCHAR2	4	-	-	1	-	-	-
	STOREID	VARCHAR2	4	-	-	2	-	-	-
									1 - 2

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STAFF_DEPT	SID	VARCHAR2	4	-	-	1	-	-	-
	DEPTID	VARCHAR2	4	-	-	2	-	-	-
									1 - 2

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
HARDWARE	SERIAL_NUMBER	VARCHAR2	9	-	-	1	-	-	-
	PROCESSOR_TYPE	VARCHAR2	33	-	-	-	✓	-	-
	STORAGE	VARCHAR2	5	-	-	-	✓	-	-
	MEMORY_GB	NUMBER	22	-	-	-	✓	-	-
									1 - 4

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT	PID	VARCHAR2	4	-	-	1	-	-	-
	PNAME	VARCHAR2	50	-	-	-	-	-	-
	PRICE	NUMBER	-	10	2	-	-	-	-
	DEPTID	VARCHAR2	4	-	-	-	-	-	-
	SERIAL_NUMBER	VARCHAR2	9	-	-	-	-	-	-
									1 - 5

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comme
TRANSACTION	TID	VARCHAR2	5	-	-	1	-	-	-
	CID	NUMBER	22	-	-	-	-	-	-
	PID	VARCHAR2	5	-	-	-	-	-	-
	STOREID	VARCHAR2	5	-	-	-	-	-	-
	DOP	DATE	7	-	-	-	-	-	-
	PRICE	NUMBER	-	10	2	-	-	-	-
	RATING	NUMBER	22	-	-	-	✓	-	-
	METHOD_OF_PURCHASE	VARCHAR2	7	-	-	-	-	-	-
									1 - 8

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STORE_PRODUCT	STOREID	VARCHAR2	4	-	-	1	-	-	-
	PID	VARCHAR2	4	-	-	2	-	-	-
									1 - 2

Queries

1. Identify customers who have not purchased a new computer recently. Display the customer name and email address. You identify the criteria for a recent purchase. Use a nested select to answer the question.

```
Select cfirst, clast, cemail
From customer
Where cid not in
        (select cid
            From transaction
         Where dop >='9/20/2014' and pid in
                    (select pid
                      From product
                 Where deptid in
                    (select deptid
                     From department
                     Where dname='Mac')));
```

CFIRST	CLAST	CEMAIL
Corey	Malone	coreymalone@gmail.com
Caitilin	King	caitilin.king@gmail.com
Binquan	Wang	lanyanzhanshen@gmail.com
Adam	Najman	adam.najman@yahoo.com
Sau	Putt	sau.putt@hotmail.com
Disha	Gupta	disha.gupta@gmail.com
Bo	Li	boli@gmail.com
Jared	Sakamoto	jared.saka@yahoo.com
Jie	Zheng	jie.zheng@gmail.com
Luke	Lee	luke.lee@gmail.com
Kevin	Dugan	kevin.dugan@gmail.com
Maria	Sharapova	mariasharapova@gmail.com

12 rows returned in 0.04 seconds Download

2. Identify the highest rated Microsoft office software. The product needs at least 10 reviews from customers who live in NY. Display the product name, price and average rating.

```
Select p.pname, p.price, avg(t.rating) "AVERAGE RATING"
From product p, transaction t
where p.pid = t.pid and p.pname like'%Microsoft%'
and cid in
  (select cid
   from customer
   where cstate = 'NY')
group by p.pname, p.price
having count(rating)>=10;
```

PNAME	PRICE	AVERAGE RATING
Microsoft Office 365 University	79.95	4.5

1 rows returned in 0.02 seconds Download

3. Identify stores with the most sales in 2014. Display one row for each store. Display the store address, city, total revenue, smallest sale and largest sale. Use functions to answer this question. Rename the columns so they are descriptive. The store with the highest revenue will display first.

```
select store_address,store_city, sum(price) "REVENUE", min(price) "SMALLEST
SALE", max(price) "LARGEST SALE"
from store s, transaction t
where s.storeid = t.storeid and dop >= '1/1/2014'
group by store_address,store_city
order by 3 desc;
```

STORE_ADDRESS	STORE_CITY	REVENUE	SMALLEST SALE	LARGEST SALE
Web	-	4094.75	79.95	999
767 Fifth Avenue	New York	2776.95	79.95	1299
1400 Willowbrook Mall	Wayne	2027.95	79.95	1299
103 Prince Street	New York	1807.9	79.95	899
45 Grand Central Terminal	New York	1598	499	1099
1 Garden State Plaza	Paramus	1327.95	79.95	649
401 W 14th Street	New York	1227.95	79.95	649
1981 Broadway	New York	1099	1099	1099

8 rows returned in 0.47 seconds

4. Double the memory of all 2014 iMac computers available at NJ stores. Identify the SQL commands to perform this operation.

```
alter table hardware
drop column memory;

alter table hardware
add memory_gb number;

update hardware
set memory_gb = 8
where serial_number='1129E405' or  serial_number='1129E406' or
serial_number='1129E407' or  serial_number='1129E404';

update hardware
set memory_gb = 4
where serial_number='1129E401' or  serial_number='1129E402' or
serial_number='1129E403' or  serial_number='1129E404';

update hardware
set memory_gb = memory_gb*2
where serial_number in
  (select serial_number
   from product
   where pname like '%iMac%'
   and pid in
     (select pid
      from store_product
      where storeid in
        (select storeid
         from store
         where store_state = 'NJ')));
```

SERIAL_NUMBER	PROCESSOR_TYPE	STORAGE	MEMORY_GB
1129E405	1.7GHz dual-core Intel Core i7	128GB	8
1129E406	1.4GHz dual-core Intel Core i5	500GB	16
1129E407	2.7GHz quad-core Intel Core i5	1TB	16

PID	PNAME	PRICE	DEPTID	SERIAL_NUMBER
p1	Microsoft Office for Mac Home and Student 2011	139.95	d6	1129S01
p12	21.5-inch iMac: 500GB	1099	d1	1129E406
p13	21.5-inch iMac: 1TB	1299	d1	1129E407

5. Products need to be recalled. Identify computers with a serial number starting with 1129E4 sold in the last 12 month in the US. Display the customer name and email address. Use a nested select to answer this question.

```
select cfirst, clast, cemail
from customer
where cid in
  (select cid
   from transaction
   where dop between '12/21/2013' and '12/21/2014' and pid in
     (select pid
      from product
      where serial_number like '1129E4%'));
```

CFIRST	CLAST	CEMAIL
Kevin	Dugan	kevin.dugan@gmail.com
Binquan	Wang	lanyanzhanshen@gmail.com
Jinwei	Lin	jinwei.lin8@gmail.com
Jie	Zheng	jie.zheng@gmail.com
Jared	Sakamoto	jared.saka@yahoo.com

5 rows returned in 0.08 seconds Download

6. Identify neighborhoods with the most iPhones sold in 2014. Display one row for each neighborhood. Use a function and nested select to answer this question. Display the neighborhood and number of phones sold. Rename the columns so they are descriptive. Display the neighborhood with the most iPhone's first.

```
select store_city neighborhood, count(*) "NUMBER OF PHONES SOLD"
from store
where storeid <> 's8' and storeid in
  (select storeid
   from transaction
   where dop >= '1/1/2014' and pid in
     (select pid
      from product
      where pname like '%iPhone%'))
group by store_city
order by 2 desc;
```

NEIGHBORHOOD	NUMBER OF PHONES SOLD
New York	3
Paramus	1
Wayne	1

3 rows returned in 0.04 seconds Download

7. Purchase one iPhone. Identify the SQL operations to perform this request.

```
insert into transaction
values('t31','1','p100','s8','12/20/2014','649',5,'website');

select *
from transaction
where tid = 't31';
```

TID	CID	PID	STOREID	DOP	PRICE	RATING	METHOD_OF_PURCHASE
t31	1	p100	s8	12/20/2014	649	5	website

1 rows returned in 0.50 seconds Download

8. In one SQL window, change the iPad price for record 1. Don't commit. In another SQL window, change the iPad price for record 1. Don't commit. Explain your results. Resolve the problem.

Window 1

```
Connected.
SQL> update product
  2  set price = price - 50
  3  where pname like '%iPad%';

2 rows updated.

SQL> select pname, price
  2  from product
  3  where pname like '%iPad%';

PNAME                                              PRICE
-------------------------------------------------- ----------
iPad mini 3: 64GB                                     449
iPad Air 2: 64GB                                      549
```

Window2

```
Connected.
SQL> update product
  2  set price = price - 20
  3  where pname like '%iPad%';
```

Window 1 updates successfully whereas window 2 is not updating. The reason why this happened is that window 1 put the write lock on two iPad rows. To resolve the problem, type in commit or rollback on window 1.

Window1

```
SQL> rollback;

Rollback complete.
```

Window2

```
Connected.
SQL> update product
  2  set price = price - 20
  3  where pname like '%iPad%';

2 rows updated.
```

9. In one SQL window, delete all products. Don't commit. In another SQL window, increase the price of all iPad's by 5%. Don't commit. Explain your results. Resolve the problem. Create a backup of your table before implementing. To create a backup table, enter CREATE TABLE <NEWTABLE> AS SELECT * FROM <ORIGINALTABLE>; COMMIT; Then you can rename a table using the RENAME TABLE commit.

Window1

```
SQL> create table product_backup as select* from product;

Table created.

SQL> commit;

Commit complete.

SQL> delete from product
  2
SQL> delete from product;
delete from product
*
ERROR at line 1:
ORA-02292: integrity constraint (LIN.SYS_C007349) violated - child record found


SQL> delete from product_backup;

11 rows deleted.
```

Window2

```
SQL> update product_backup
  2  set price = price + price*0.05
  3  where pname like '%iPad%';
```

Window 1 updates successfully whereas window 2 is not updating. The reason why this happened is that window 1 put the lock on table product_backup. To resolve the problem, type in rollback on window 1.

Window1

```
SQL> rollback;

Rollback complete.
```

Window2

```
SQL> update product_backup
  2  set price = price + price*0.05
  3  where pname like '%iPad%';

2 rows updated.
```

10. In one SQL window, decrease the price of all products by 50%. Don't commit. In another SQL window, double the memory in all computers. Don't commit. Quit both Oracle sessions. Login to Oracle and display all information. Explain your results.

Window 1

```
SQL> update product
  2  set price = price - price*0.5;

11 rows updated.
```

Window 2

```
SQL> update hardware
  2  set memory_gb = memory_gb*2
  3  where serial_number in
  4    (select serial_number
  5     from product
  6     where pname like '%Mac%');

6 rows updated.
```

Login and display

```
PNAME                                              PRICE
-------------------------------------------------- ----------
Microsoft Office for Mac Home and Student 2011         139.95
21.5-inch iMac: 500GB                                    1099
21.5-inch iMac: 1TB                                      1299
iPad mini 3: 64GB                                         499
Microsoft Office for Mac Home and Business 2011        219.95
Microsoft Office 365 University                         79.95
11-inch MacBook Air: 128GB                                899
13-inch MacBook Air: 128GB                                999
iPhone 6: 16GB                                            649
iPhone 6 Plus: 16GB                                       749
iPad Air 2: 64GB                                          599

11 rows selected.
```

```
SQL> select pname, memory_gb
  2  from product p, hardware h
  3  where p.serial_number = h.serial_number;

PNAME                                              MEMORY_GB
-------------------------------------------------- ----------
21.5-inch iMac: 500GB                                      16
21.5-inch iMac: 1TB                                        16
iPhone 6: 16GB
11-inch MacBook Air: 128GB                                  4
13-inch MacBook Air: 128GB                                  4
iPad Air 2: 64GB
iPad mini 3: 64GB
Microsoft Office for Mac Home and Student 2011
Microsoft Office for Mac Home and Business 2011
Microsoft Office 365 University
iPhone 6 Plus: 16GB

11 rows selected.
```

All data do not change. Since updates weren't committed before quitting both sessions, changes would not get reflected to the database. To resolve the problem, make sure to commit before quitting the session.