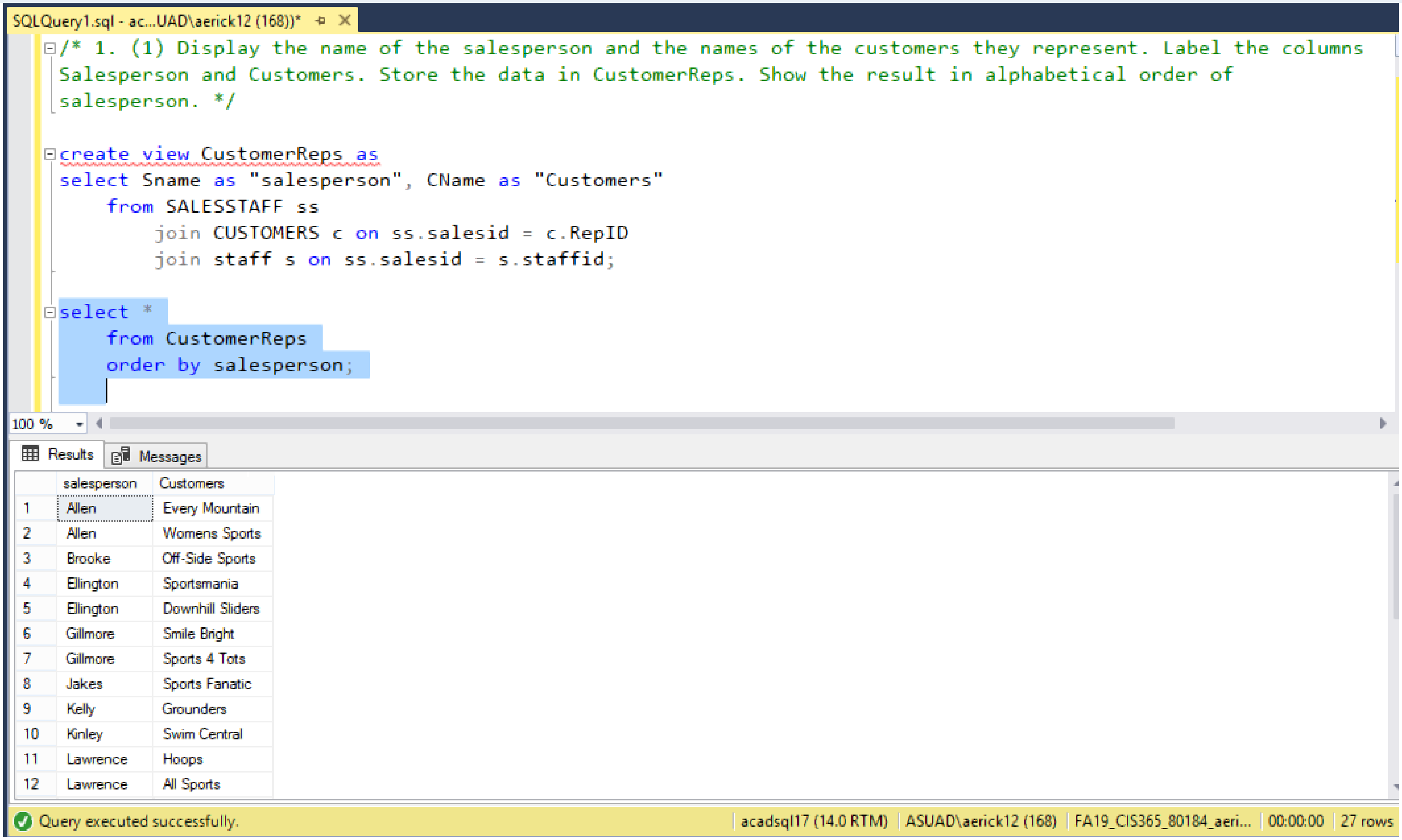
CIS365 10:30 AM – SQL VIEWS/TRANSACTIONS

AKIKO OTSUKA

Q1



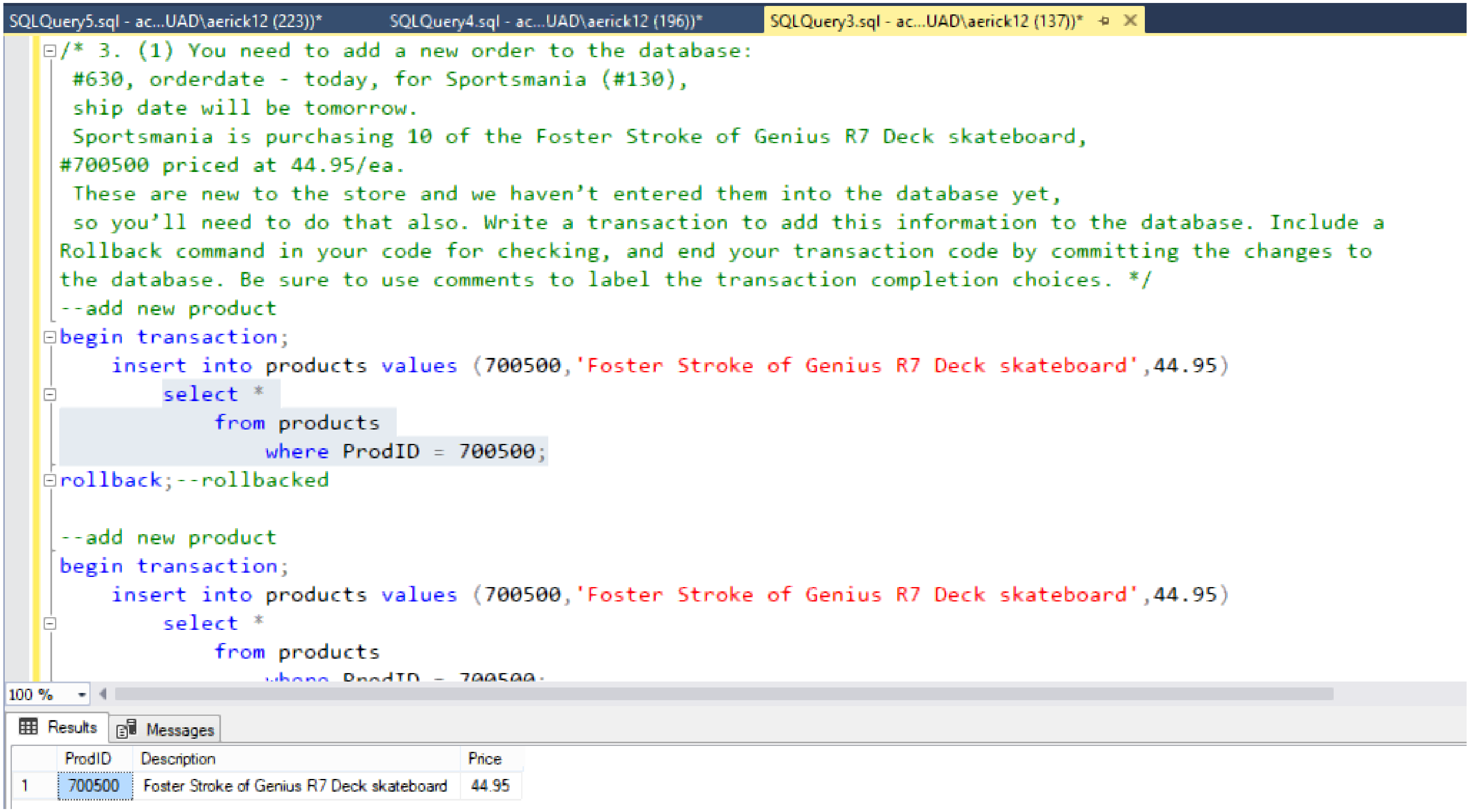
```
/* 1. (1) Display the name of the salesperson and the names of the customers they represent. Label the columns
Salesperson and Customers. Store the data in CustomerReps. Show the result in alphabetical order of
salesperson. */

create view CustomerReps as
select Sname as "salesperson", CName as "Customers"
    from SALESSTAFF ss
        join CUSTOMERS c on ss.salesid = c.RepID
        join staff s on ss.salesid = s.staffid;

select *
    from CustomerReps
    order by salesperson;
```

	salesperson	Customers
1	Allen	Every Mountain
2	Allen	Womens Sports
3	Brooke	Off-Side Sports
4	Ellington	Sportsmania
5	Ellington	Downhill Sliders
6	Gillmore	Smile Bright
7	Gillmore	Sports 4 Tots
8	Jakes	Sports Fanatic
9	Kelly	Grounders
10	Kinley	Swim Central
11	Lawrence	Hoops
12	Lawrence	All Sports

```
Query executed successfully.    acadsql17 (14.0 RTM) | ASUAD\aerick12 (168) | FA19_CIS365_80184_aeri... | 00:00:00 | 27 rows
```

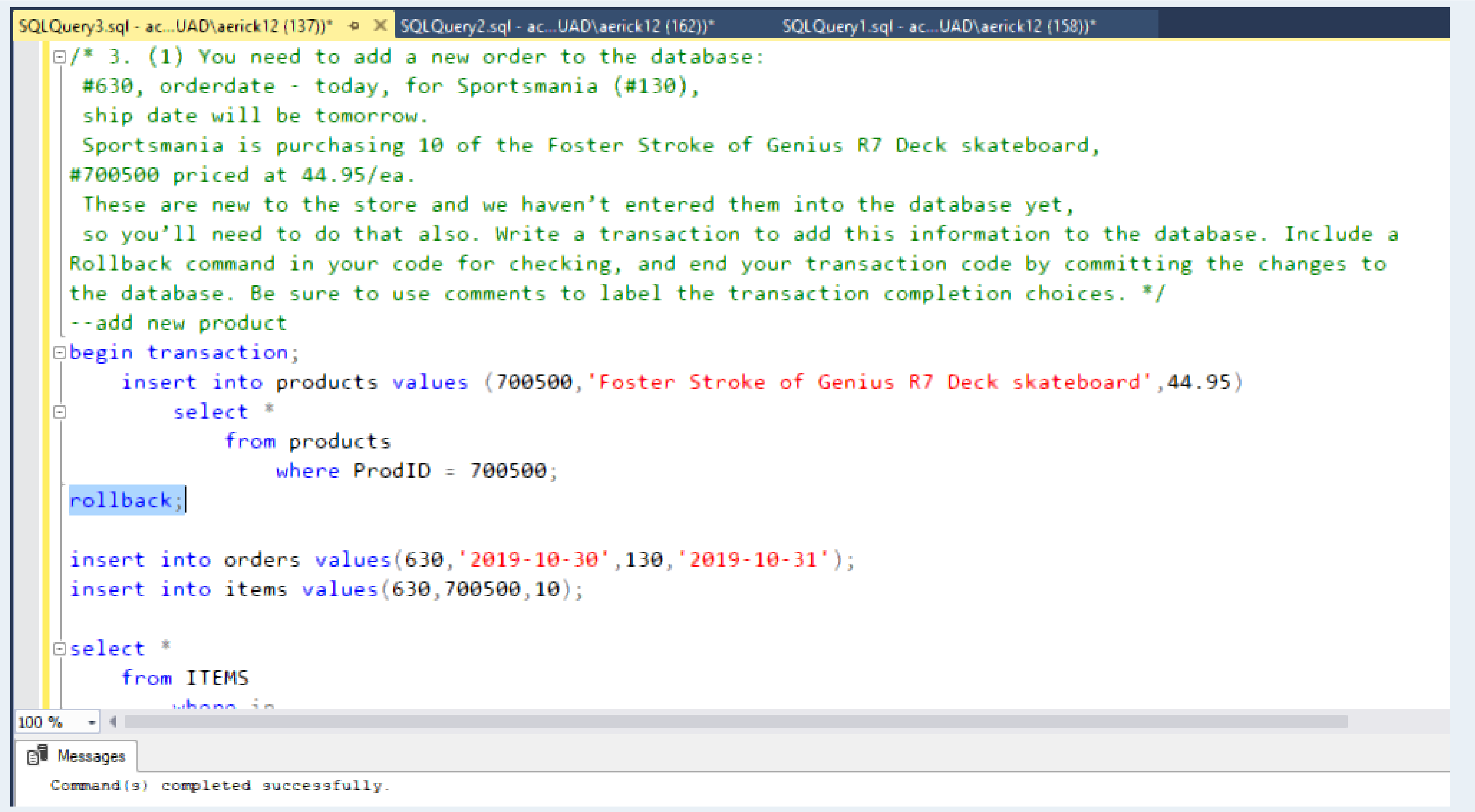
Q2

Q3


SQLQuery5.sql - ac...UAD\aerick12 (223))*
SQLQuery4.sql - ac...UAD\aerick12 (196))*
SQLQuery3.sql - ac...UAD\aerick12 (137))*
100 %
Results
Messages

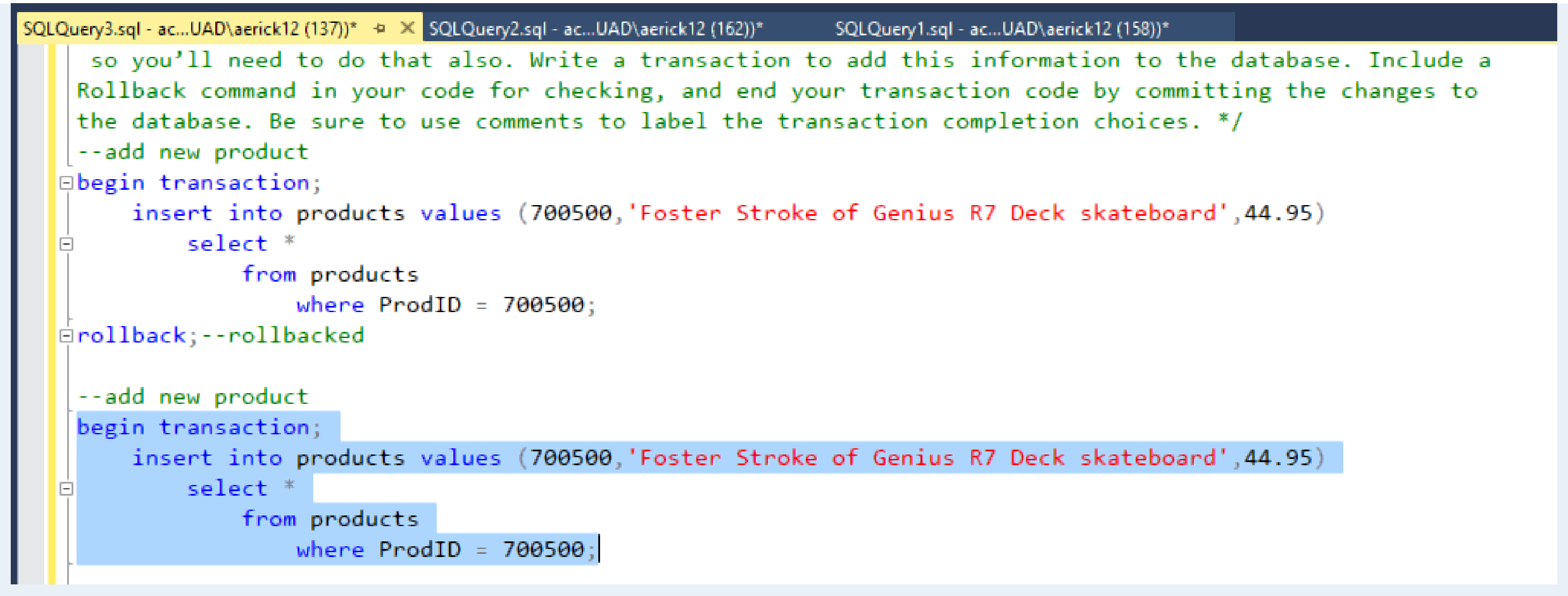
```
/* 3. (1) You need to add a new order to the database:
 #630, orderdate - today, for Sportsmania (#130),
 ship date will be tomorrow.
 Sportsmania is purchasing 10 of the Foster Stroke of Genius R7 Deck skateboard,
#700500 priced at 44.95/ea.
 These are new to the store and we haven’t entered them into the database yet,
 so you’ll need to do that also. Write a transaction to add this information to the database. Include a
Rollback command in your code for checking, and end your transaction code by committing the changes to
the database. Be sure to use comments to label the transaction completion choices. */
--add new product
begin transaction;
    insert into products values (700500,'Foster Stroke of Genius R7 Deck skateboard',44.95)
        select *
            from products
                where ProdID = 700500;
rollback;--rollbacked

--add new product
begin transaction;
    insert into products values (700500,'Foster Stroke of Genius R7 Deck skateboard',44.95)
        select *
            from products
                where ProdID = 700500;
```

	ProdID	Description	Price
1	700500	Foster Stroke of Genius R7 Deck skateboard	44.95

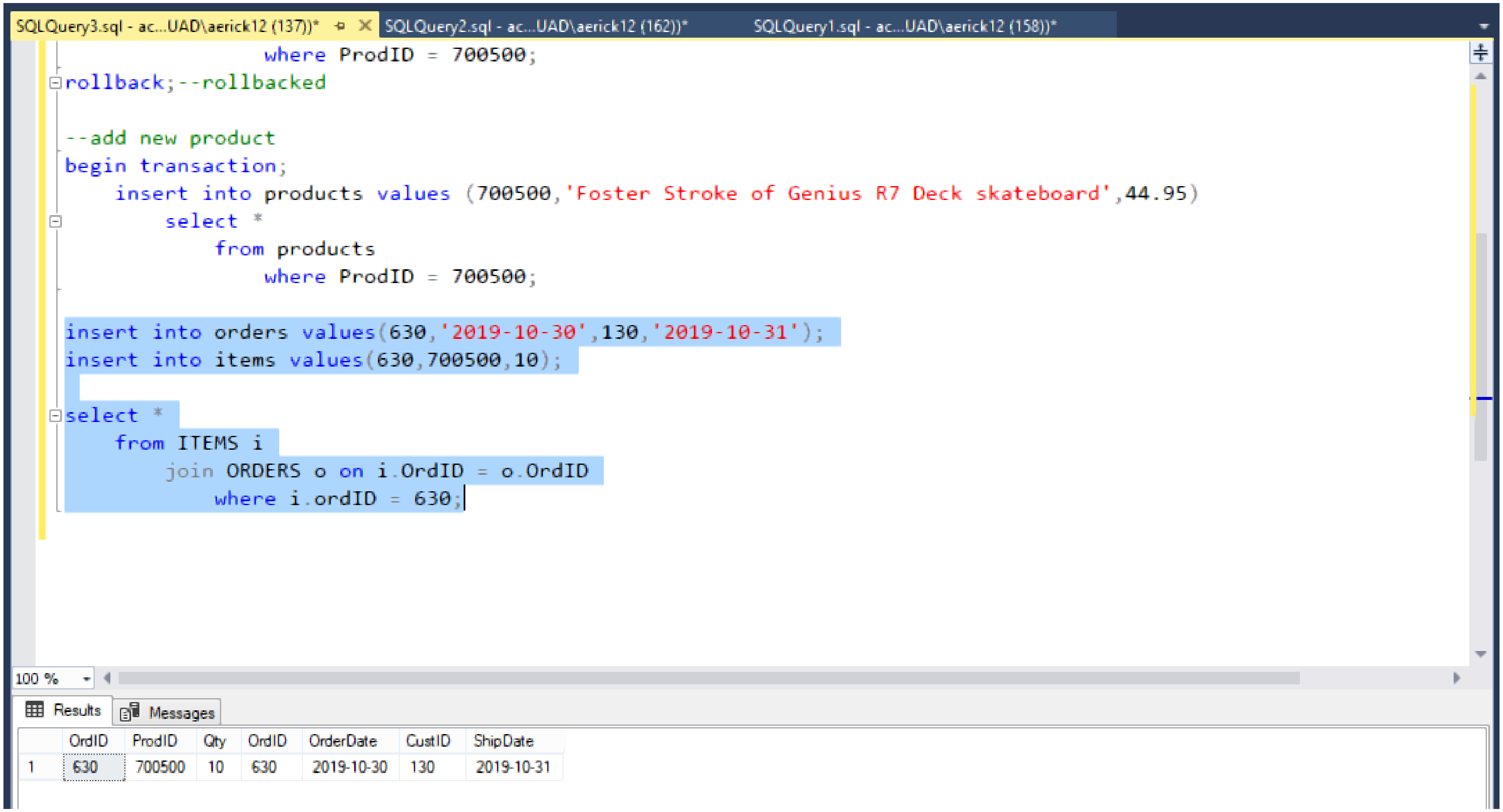

SQLQuery3.sql - ac...UAD\aerick12 (137))*
SQLQuery2.sql - ac...UAD\aerick12 (162))*
SQLQuery1.sql - ac...UAD\aerick12 (158))*
100 %
Messages

```
/* 3. (1) You need to add a new order to the database:
 #630, orderdate - today, for Sportsmania (#130),
 ship date will be tomorrow.
 Sportsmania is purchasing 10 of the Foster Stroke of Genius R7 Deck skateboard,
#700500 priced at 44.95/ea.
 These are new to the store and we haven't entered them into the database yet,
 so you'll need to do that also. Write a transaction to add this information to the database. Include a
Rollback command in your code for checking, and end your transaction code by committing the changes to
the database. Be sure to use comments to label the transaction completion choices. */
--add new product
begin transaction;
    insert into products values (700500,'Foster Stroke of Genius R7 Deck skateboard',44.95)
        select *
            from products
                where ProdID = 700500;
rollback;

insert into orders values(630,'2019-10-30',130,'2019-10-31');
insert into items values(630,700500,10);

select *
    from ITEMS
        where in
```

```
Command(s) completed successfully.
```

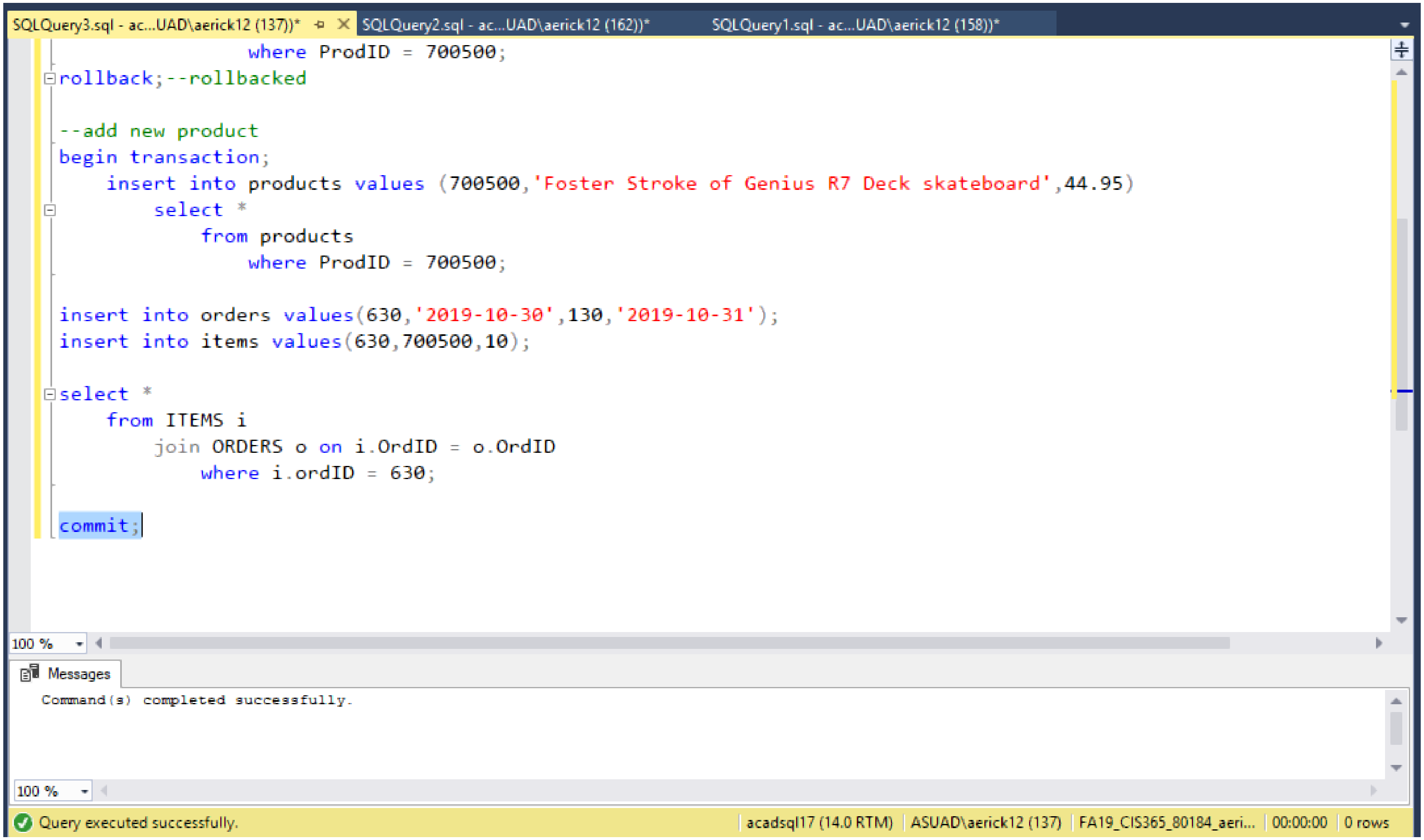

SQLQuery3.sql - ac...UAD\aerick12 (137))*
SQLQuery2.sql - ac...UAD\aerick12 (162))*
SQLQuery1.sql - ac...UAD\aerick12 (158))*

```
 so you'll need to do that also. Write a transaction to add this information to the database. Include a
Rollback command in your code for checking, and end your transaction code by committing the changes to
the database. Be sure to use comments to label the transaction completion choices. */
--add new product
begin transaction;
    insert into products values (700500,'Foster Stroke of Genius R7 Deck skateboard',44.95)
        select *
            from products
                where ProdID = 700500;
rollback;--rollbacked

--add new product
begin transaction;
    insert into products values (700500,'Foster Stroke of Genius R7 Deck skateboard',44.95)
        select *
            from products
                where ProdID = 700500;
```


SQLQuery3.sql - ac...UAD\aerick12 (137))*
SQLQuery2.sql - ac...UAD\aerick12 (162))*
SQLQuery1.sql - ac...UAD\aerick12 (158))*
100 %
Results
Messages

```
                where ProdID = 700500;
rollback;--rollbacked

--add new product
begin transaction;
    insert into products values (700500,'Foster Stroke of Genius R7 Deck skateboard',44.95)
        select *
            from products
                where ProdID = 700500;

insert into orders values(630,'2019-10-30',130,'2019-10-31');
insert into items values(630,700500,10);

select *
    from ITEMS i
        join ORDERS o on i.OrdID = o.OrdID
            where i.ordID = 630;
```

	OrdID	ProdID	Qty	OrdID	OrderDate	CustID	ShipDate
1	630	700500	10	630	2019-10-30	130	2019-10-31



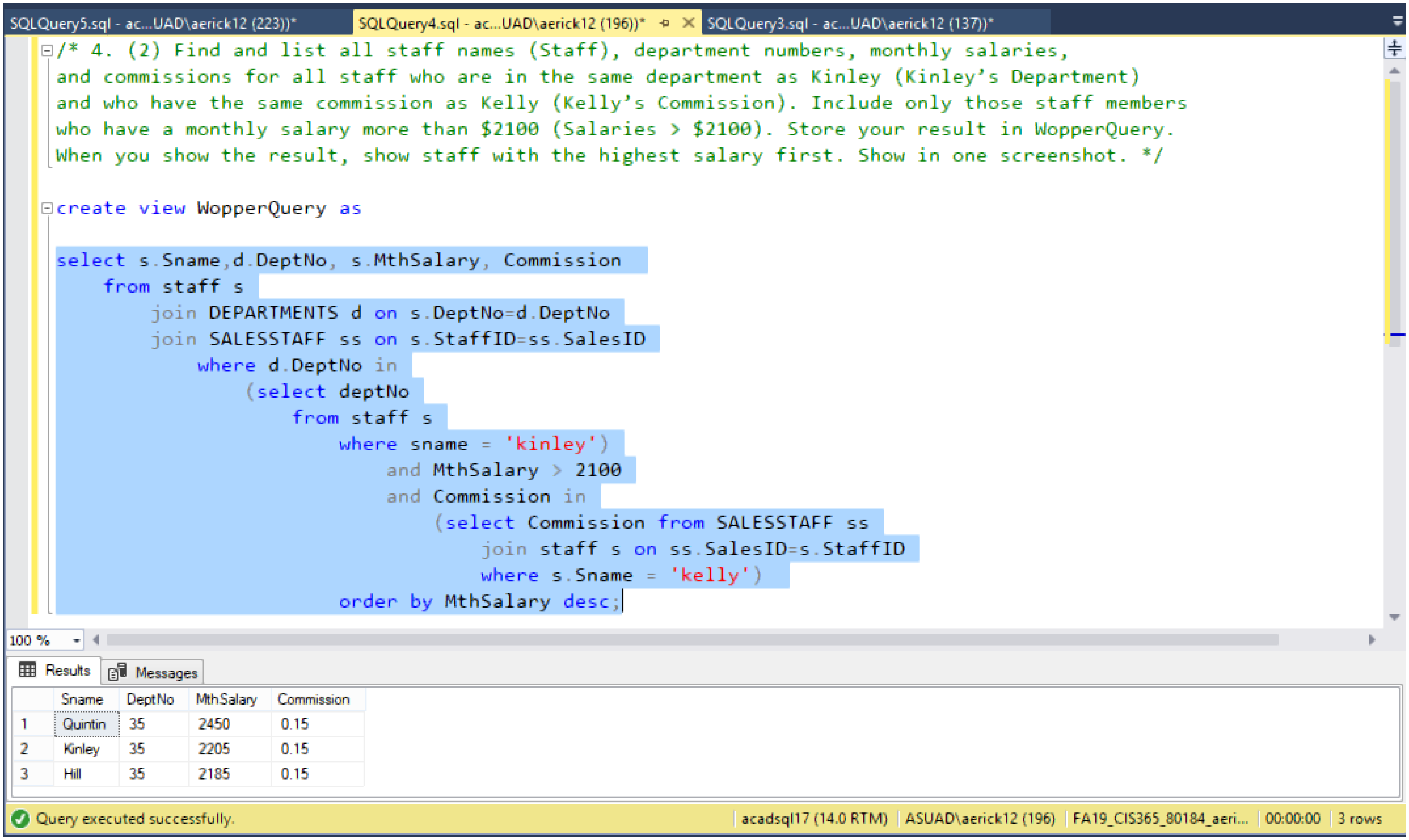
SQLQuery3.sql - ac...UAD\aerick12 (137))*
SQLQuery2.sql - ac...UAD\aerick12 (162))*
SQLQuery1.sql - ac...UAD\aerick12 (158))*
100 %
Messages
100 %

```sql
                where ProdID = 700500;
rollback;--rollbacked

--add new product
begin transaction;
    insert into products values (700500,'Foster Stroke of Genius R7 Deck skateboard',44.95)
        select *
            from products
                where ProdID = 700500;

insert into orders values(630,'2019-10-30',130,'2019-10-31');
insert into items values(630,700500,10);

select *
    from ITEMS i
        join ORDERS o on i.OrdID = o.OrdID
            where i.ordID = 630;

commit;
```

```
Command(s) completed successfully.
```

Query executed successfully. | acadsql17 (14.0 RTM) | ASUAD\aerick12 (137) | FA19_CIS365_80184_aeri... | 00:00:00 | 0 rows

Q4


SQLQuery5.sql - ac...UAD\aerick12 (223))*
SQLQuery4.sql - ac...UAD\aerick12 (196))*
SQLQuery3.sql - ac...UAD\aerick12 (137))*
100 %
Results
Messages

```
/* 4. (2) Find and list all staff names (Staff), department numbers, monthly salaries,
and commissions for all staff who are in the same department as Kinley (Kinley’s Department)
and who have the same commission as Kelly (Kelly’s Commission). Include only those staff members
who have a monthly salary more than $2100 (Salaries > $2100). Store your result in WopperQuery.
When you show the result, show staff with the highest salary first. Show in one screenshot. */

create view WopperQuery as

select s.Sname,d.DeptNo, s.MthSalary, Commission
    from staff s
        join DEPARTMENTS d on s.DeptNo=d.DeptNo
        join SALESSTAFF ss on s.StaffID=ss.SalesID
            where d.DeptNo in
                (select deptNo
                    from staff s
                        where sname = 'kinley')
                            and MthSalary > 2100
                            and Commission in
                                (select Commission from SALESSTAFF ss
                                    join staff s on ss.SalesID=s.StaffID
                                    where s.Sname = 'kelly')
                        order by MthSalary desc;
```

	Sname	DeptNo	MthSalary	Commission
1	Quintin	35	2450	0.15
2	Kinley	35	2205	0.15
3	Hill	35	2185	0.15

Query executed successfully. | acadsql17 (14.0 RTM) | ASUAD\aerick12 (196) | FA19_CIS365_80184_aeri... | 00:00:00 | 3 rows