1. Identify the medication history of [child patient name]. Display the patient name,
parent names, physician, drug, date of prescription and dosage. Order chronologically by date.

```
select pat_fname||', '||pat_lname Patient_Name, guard_fname||', '||guard_lname
Parent_Name, pre_fname||', '||pre_lname Physician,

brand_name Drug_Name, date_prescribed, dosage

FROM patient a INNER JOIN prescription b

ON a.pat_id = b.pat_id

INNER JOIN guardians_children c

ON a.pat_id = c.pat_id

INNER JOIN drugs d

ON d.DID = b.DID

INNER JOIN prescriber e

ON b.presr_id = e.presr_id

INNER JOIN guardians f

ON c.guard_id = f.guard_id

WHERE pat_fname = 'Snow' AND pat_lname = 'White'

ORDER BY date_prescribed
```

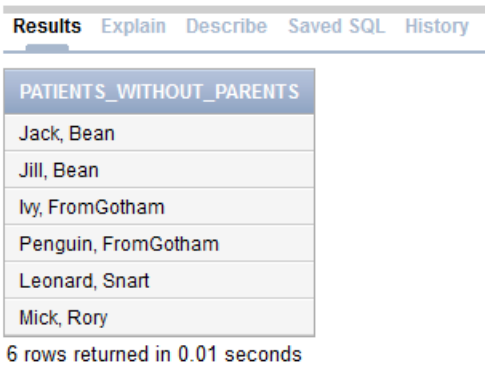
Results Explain Describe Saved SQL History

PATIENT_NAME	PARENT_NAME	PHYSICIAN	DRUG_NAME	DATE_PRESCRIBED	DOSAGE
Snow, White	SomeGuard, TheGuard	Dr., Phil	Differin	02/15/2016	50mg
Snow, White	SomeGuard, TheGuard	Dr., Phil	Differin	05/13/2016	200mg

2 rows returned in 0.01 seconds

2. Identify child patients without parents in the database. Display the child patient
name. Use a nested select to answer this question.

```
SELECT pat_fname||', '||pat_lname Patients_Without_Parents
from patient
where ischild = 'yes' AND pat_id NOT IN(
  SELECT pat_id
  FROM guardians_children
)
order by pat_id
```



Results Explain Describe Saved SQL History

PATIENTS_WITHOUT_PARENTS
Jack, Bean
Jill, Bean
Ivy, FromGotham
Penguin, FromGotham
Leonard, Snart
Mick, Rory

6 rows returned in 0.01 seconds

3. Identify pharmacy staff that dispensed the most prescriptions in the last year.
Display the pharmacy staff name, store address and number of medications. Display one row for each pharmacy staff. The staff with the most medications will be displayed first. Use a nested select to answer this question.

```
SELECT *

FROM (

SELECT c.pharm_s_fname||', '||c.pharm_s_lname Pharmacy_Staff, b.address,
count(pharm_s_fname) COUNT

FROM staff_in_pharmacy a INNER JOIN pharmacy b

ON a.pharm_id = b.pharm_id

INNER JOIN pharm_staff c

ON a.pharm_staff_id = c.pharm_staff_id

INNER JOIN prescription d

ON d.dispensed_by = c.pharm_staff_id

GROUP BY pharm_s_fname, pharm_s_lname, address

ORDER BY 3 DESC

)
```

Results Explain Describe Saved SQL History

PHARMACY_STAFF	ADDRESS	COUNT
Vision, From_Avengers	783 Manhattan Ave	6
Bruce, Wayne	783 Manhattan Ave	2
Nikita, Kruschev	559 Driggs Ave	1
Bruce, Banner	559 Driggs Ave	1
Hulk, Hogan	559 Driggs Ave	1

5 rows returned in 0.01 seconds

4. Identify pharmacies with more than three staff. Display the store name and
number of staff. Display one row for each store. The store with the most staff will
be displayed first.

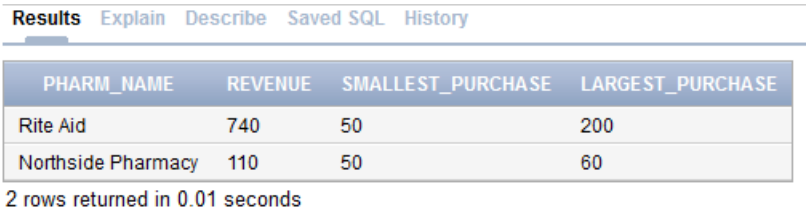
```
SELECT b.pharm_name, count(1) COUNT

FROM staff_in_pharmacy a INNER JOIN pharmacy b

ON a.pharm_id = b.pharm_id

INNER JOIN pharm_staff c

ON a.pharm_staff_id = c.pharm_staff_id

GROUP BY pharm_name

HAVING count(1) > 3

ORDER BY 2 DESC
```

PHARM_NAME	COUNT
Rite Aid	7
Northside Pharmacy	4

2 rows returned in 0.01 seconds

5. Identify stores with the most sales in 2016. Display one row for each store. Display the store address, city, total revenue, smallest sale and largest sale. Use
functions to answer this question. The store with the highest revenue will display
first.

```
select b.pharm_name, sum(a.sale_amount), min(a.sale_amount),
max(a.sale_amount)

from sales a INNER JOIN pharmacy b

ON a.pharm_id = b.pharm_id

where date_purchased >= '01012016'

group by b.pharm_name

order by sum(a.sale_amount) desc
```



Results Explain Describe Saved SQL History

PHARM_NAME	REVENUE	SMALLEST_PURCHASE	LARGEST_PURCHASE
Rite Aid	740	50	200
Northside Pharmacy	110	50	60

2 rows returned in 0.01 seconds

6. Increase the price of Differin by 25% at all stores. Identify the SQL commands to perform this operation.

```
update pharmacy_inventory
set drug_price = drug_price * (drug_price * .25)
WHERE DID = (
select DID
from drugs
where brand_name = 'Differin' AND
dosage='50mg'
)
```

3 rows updated.

7. The pharmacist doesn’t know how to spell a drug name, but the first few letters
are *adap*. Identify all drugs with a similar spelling. Display the brand name, generic name and dosage.

```
select generic_name, brand_Name, dosage

from drugs

where lower(generic_name) LIKE lower('%adap%')
```

Results Explain Describe Saved SQL History

GENERIC_NAME	BRAND_NAME	DOSAGE
Adapalene	Differin	50mg
Adapalene	Differin	100mg
Adapalene	Differin	200mg

3 rows returned in 0.00 seconds

8. The drug Adapelene will no longer be sold at all stores. What is the best process
to implement. Identify the SQL commands to perform this operation.

```
delete from pharmacy_inventory
where DID =  (
select DID
from drugs
where brand_name = 'Differin' AND
dosage='50mg'
)
```

3 row(s) deleted…

9. In one SQL window, change the staff salary for record 1. Don’t commit. In another SQL window, change the staff salary for record 1. Don’t commit. Resolve
the problem. Disable the auto commit flag at the top of the windows before performing this operation. Explain your results.

When performing this operation on two separate users there appears to be a deadlock. When the first account performs a SQL command for updating record 1, everything completes fine. When the second user attempts to perform the same SQL operation there is a deadlock. The reason for this is because the first user does not commit their update, therefore the second user cant access that row for modification. The solution for this would be for the first user to commit right after performing the SQL operation, so that the row write-lock is disengaged. At that point the second user can perform their changes to the same row. A rollback from the first user, would also free up the row write-lock.

10. In one SQL window, delete all drugs. Don’t commit. In another SQL window,
increase the price of all drugs 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. Disable the auto commit flag at the top of the windows before performing this operation.

The first problem that occurs are foreign key constraints. In order for the first user to delete all drugs from the table the foreign key constraints would need to be removed from other tables. Assuming this wasn’t the case, then if the second user changed the prices of all the drugs first by 5% and didn’t commit, and then the first user tried to delete all the drugs, then there would be a deadlock on the first user due to the second user having a table lock. Therefore it wouldn’t be possible to delete all the drugs without releasing the table lock. To solve this, the user that that increased the price would have to commit or rollback to release this lock. This would resolve this deadlock.

11. In one SQL window, null all patient addresses. Don’t commit. In another SQL
window, null all patient allergies. Don’t commit. Quit both Oracle sessions. Login to Oracle and search for this information. Explain your results. Disable the
auto commit flag at the top of the windows before performing this operation.

Neither of the sessions’ changed the data from their SQL commands when they ended their respective sessions. The reason for this is simple, the commands were not committed therefore all uncommitted DML changes were not applied, leading to no data changes. Upon logging back in, the state of the data was returned to the very last commit (if there ever was one).

12. Use the SQL DESCRIBE operation to list the table structure for all tables.

Column Name	Data Type	Nullable	Default	Primary Key
ORG_ID	NUMBER	No	-	1
ORG_TYPE	VARCHAR2(15)	No	-	-
ORG_NAME	VARCHAR2(40)	No	-	-
ADDRESS	VARCHAR2(40)	No	-	-
ZIP	VARCHAR2(10)	No	-	-

1 - 5

Column Name	Data Type	Nullable	Default	Primary Key
P_ALLERGY_ID	NUMBER	No	-	1
PAT_ID	NUMBER	No	-	-
ALLERGIC_TO_DRUG	VARCHAR2(25)	Yes	-	-

1 - 3

Column Name	Data Type	Nullable	Default	Primary Key
PAT_ID	NUMBER	No	-	1
PAT_FNAME	VARCHAR2(25)	No	-	-
PAT_LNAME	VARCHAR2(25)	No	-	-
ADDRESS	VARCHAR2(40)	No	-	-
ZIP	VARCHAR2(10)	No	-	-
PAT_EMAIL	VARCHAR2(100)	No	-	-
PHARM_ID	NUMBER	No	-	-
PRIMARY_PHYSICIAN	NUMBER	No	-	-
ISCHILD	VARCHAR2(3)	Yes	-	-

1 - 9

Column Name	Data Type	Nullable	Default	Primary Key
PHARM_STAFF_ID	NUMBER	No	-	1
PHARM_S_FNAME	VARCHAR2(20)	No	-	-
PHARM_S_LNAME	VARCHAR2(20)	No	-	-
HIRE_DATE	DATE	No	-	-
TITLE	VARCHAR2(15)	Yes	-	-
SALARY	NUMBER(10,2)	No	-	-

1 - 6

Column Name	Data Type	Nullable	Default	Primary Key
PHARM_ID	NUMBER	No	-	1
PHARM_NAME	VARCHAR2(30)	No	-	-
ADDRESS	VARCHAR2(30)	No	-	-
ZIP	VARCHAR2(5)	No	-	-
PHONE	VARCHAR2(12)	No	-	-

1 - 5

Column Name	Data Type	Nullable	Default	Primary Key
PHARM_ID	NUMBER	No	-	1
DID	NUMBER	No	-	2
DRUG_PRICE	NUMBER(8,2)	Yes	-	-
AMOUNT_LEFT	NUMBER	Yes	-	-

1 - 4

Column Name	Data Type	Nullable	Default	Primary Key
PRESR_ID	NUMBER	No	-	1
PRE_FNAME	VARCHAR2(25)	No	-	-
PRE_LNAME	VARCHAR2(25)	No	-	-
LICENSE_NUMBER	NUMBER	No	-	-

1 - 4

Column Name	Data Type	Nullable	Default	Primary Key
ORG_ID	NUMBER	No	-	1
PRESR_ID	NUMBER	No	-	2

1 - 2

Column Name	Data Type	Nullable	Default	Primary Key
PPHONE_ID	NUMBER	No	-	1
PHONE	VARCHAR2(12)	No	-	-
PRESR_ID	NUMBER	No	-	-

1 - 3

Column Name	Data Type	Nullable	Default	Primary Key
PRESN_ID	NUMBER	No	-	1
DID	NUMBER	No	-	-
PAT_ID	NUMBER	No	-	-
PRESCRIPTION_FNAME	VARCHAR2(25)	No	-	-
PRESCRIPTION_LNAME	VARCHAR2(25)	No	-	-
DATE_PRESCRIBED	DATE	No	-	-
NUMBER_REFILLS	NUMBER	No	-	-
PRESR_ID	NUMBER	No	-	-
DISPENSED_BY	NUMBER	Yes	-	-

1 - 9

Column Name	Data Type	Nullable	Default	Primary Key
SALE_ID	NUMBER	No	-	1
SALE_AMOUNT	NUMBER(10,2)	No	-	-
PHARM_ID	NUMBER	No	-	-
DATE_PURCHASED	DATE	No	-	-

1 - 4

Column Name	Data Type	Nullable	Default	Primary Key
PHARM_ID	NUMBER	No	-	1
PHARM_STAFF_ID	NUMBER	No	-	2

1 - 2

Column Name	Data Type	Nullable	Default	Primary Key
ST	VARCHAR2(30)	No	-	1
STATE_NAME	VARCHAR2(30)	Yes	-	-

1 - 2

Column Name	Data Type	Nullable	Default	Primary Key
ZIP	VARCHAR2(5)	No	-	1
CITY	VARCHAR2(30)	No	-	-
ST	VARCHAR2(2)	No	-	-
COUNTY_BOROUGH	VARCHAR2(30)	No	-	-

1 - 4

Column Name	Data Type	Nullable	Default	Primary Key
DID	NUMBER	No	-	1
GENERIC_NAME	VARCHAR2(30)	No	-	-
BRAND_NAME	VARCHAR2(30)	No	-	-
DOSAGE	VARCHAR2(7)	No	-	-

1 - 4

Column Name	Data Type	Nullable	Default	Primary Key
DID	NUMBER	No	-	-
GENERIC_NAME	VARCHAR2(30)	No	-	-
BRAND_NAME	VARCHAR2(30)	No	-	-
DOSAGE	VARCHAR2(7)	No	-	-

1 - 4

Column Name	Data Type	Nullable	Default	Primary Key
GUARD_ID	NUMBER	No	-	1
GUARD_FNAME	VARCHAR2(30)	Yes	-	-
GUARD_LNAME	VARCHAR2(30)	Yes	-	-
GUARD_EMAIL	VARCHAR2(100)	Yes	-	-
ADDRESS	VARCHAR2(30)	Yes	-	-
ZIP	VARCHAR2(10)	Yes	-	-

1 - 6

Column Name	Data Type	Nullable	Default	Primary Key
GUARD_ID	NUMBER	No	-	1
PAT_ID	NUMBER	No	-	2

1 - 2

Column Name	Data Type	Nullable	Default	Primary Key
GUARD_PHONE_ID	NUMBER	No	-	1
GUARD_PHONE_NUM	VARCHAR2(12)	Yes	-	-
GUARD_ID	NUMBER	No	-	-

1 - 3

CREATE TABLE/ALTER/MODIFY

```
create table guardians(
guard_id int NOT NULL,
guard_fname varchar2(30),
guard_lname varchar2(30),
guard_email varchar(100),
address varchar2(30),
zip varchar(10),
PRIMARY KEY(guard_id)
)

create table guardians_phone_number(
guard_phone_id int NOT NULL,
PRIMARY KEY(guard_phone_id),
guard_phone_num varchar2(12),
guard_id int NOT NULL,
FOREIGN KEY (guard_id) REFERENCES guardians(guard_id)
)

create table pharm_staff(
pharm_staff_id int NOT NULL,
PRIMARY KEY (pharm_staff_id),
pharm_s_fname varchar2(20) NOT NULL,
pharm_s_lname varchar2(20) NOT NULL,
hire_date date NOT NULL,
title varchar(15),
salary NUMBER(5,2) NOT NULL
)
```

```
create table staff_in_pharmacy(
pharm_staff_id int NOT NULL,
pharm_id int NOT NULL,
PRIMARY KEY (pharm_id, pharm_staff_id),
FOREIGN KEY(pharm_staff_id) REFERENCES pharm_staff(pharm_staff_id),
FOREIGN KEY(pharm_id) REFERENCES pharmacy(pharm_id)
)

alter table prescription
add dispensed_by int

alter table prescription
add FOREIGN KEY(dispensed_by) REFERENCES pharm_staff(pharm_staff_id)

create sequence
pharmacist_sequence
start with 250
increment by 1;

create sequence
pharm_sales
start with 1000
increment by 1;

create sequence
guard_sequence
start with 200
increment by 1
```

```
alter table patient
add ischild varchar2(3)

create table sales(
sale_id int NOT NULL,
sale_amount number(10,2) NOT NULL,
pharm_id int NOT NULL,
PRIMARY KEY(sale_id),
FOREIGN KEY(pharm_id) REFERENCES pharmacy(pharm_id)
)

alter table sales
add date_purchased date NOT NULL

create table guardians_children(
guard_id int NOT NULL,
pat_id int NOT NULL,
PRIMARY KEY(guard_id, pat_id),
FOREIGN KEY(guard_id) REFERENCES guardians(guard_id),
FOREIGN KEY(pat_id) REFERENCES patient(pat_id)
)
create table pharmacy_inventory(
pharm_id int NOT NULL,
DID int NOT NULL,
drug_price number(8,2),
amount_left int,
PRIMARY KEY(pharm_id, DID),
FOREIGN KEY(pharm_id) REFERENCES pharmacy(pharm_id),
FOREIGN KEY(DID) REFERENCES drugs(DID)
)
```

INSERTIONS

```
insert into pharm_staff
values(pharmacist_sequence.nextval,
'Joseph',
'Stalin',
'05231924',
'RisingStar',
10);

insert into pharm_staff
values(pharmacist_sequence.nextval,
'Nikita',
'Kruschev',
'05231953',
'Sr. Pharmacist',
25);

insert into pharm_staff
values(pharmacist_sequence.nextval,
'Hulk',
'Hogan',
'03191997',
'DoesNotNeedOne',
25);

insert into pharm_staff
values(pharmacist_sequence.nextval,
'Bruce',
'Banner',
'03191997',
```

```
'Sr. Pharmacist',
40);

insert into pharm_staff
values(pharmacist_sequence.nextval,
'Bruce',
'Wayne',
'03052005',
'Pharmacist',
50);

insert into pharm_staff
values(pharmacist_sequence.nextval,
'Vision',
'From_Avengers',
'03052008',
'Best Pharmacist',
50);

insert into pharm_staff
values(pharmacist_sequence.nextval,
'Tony',
'Stark',
'04022009',
'Jr. Pharmacist',
40);

insert into pharm_staff
values(pharmacist_sequence.nextval,
'Brad',
```

```
'Pitt',
'04032014',
'Sr. Pharmacist',
65);

insert into pharm_staff
values(pharmacist_sequence.nextval,
'Shado',
'Wilson',
'10152009',
'Sr. Pharmacist',
70);

insert into pharm_staff
values(pharmacist_sequence.nextval,
'Slade',
'Wilson',
'10162009',
'Jr. Pharmacist',
60);

insert into pharm_staff
values(pharmacist_sequence.nextval,
'Harley',
'Quinn',
'11042010',
'Dangerous Pharm',
20);

insert into pharm_staff
```

```
values(pharmacist_sequence.nextval,
'The',
'Joker',
'11042010',
'Dangerous Pharm',
25);

insert into staff_in_pharmacy
values(250, 54)
insert into staff_in_pharmacy
values(251, 54)
insert into staff_in_pharmacy
values(252, 54)
insert into staff_in_pharmacy
values(253, 54)
insert into staff_in_pharmacy
values(254, 54)
insert into staff_in_pharmacy
values(261, 53)
insert into staff_in_pharmacy
values(262, 53)
insert into staff_in_pharmacy
values(263, 53)
insert into staff_in_pharmacy
values(264, 53)
insert into staff_in_pharmacy
values(256, 52)
insert into staff_in_pharmacy
values(258, 52)
insert into staff_in_pharmacy
```

values(259, 52)

insert into patient

values(patient_sequence.nextval, 'GirlFrom', 'TheRing', '534 Manhattan Avenue', 11222, 'girlfromthering@thewell.com', 54, 529, 'yes');

insert into patient

values(patient_sequence.nextval, 'Snow', 'White', '529 Manhattan Avenue', 11222, 'snowwhite@disney.com', 54, 529, 'yes');

insert into patient

values(patient_sequence.nextval, 'Alice', 'Wonder', '520 Manhattan Avenue', 11222, 'alicefromWonderland@alternateworld.com', 54, 529, 'yes');

insert into patient

values(patient_sequence.nextval, 'Sleeping', 'Beauty', '510 Manhattan Avenue', 11222, 'sleepingbeauty@sleepforever.com', 54, 529, 'yes');

insert into patient

values(patient_sequence.nextval, 'Jack', 'Bean', '200 Manhattan Avenue', 11222, 'jack@wentupthehill.edu', 54, 529, 'yes');

insert into patient

values(patient_sequence.nextval, 'Jill', 'Bean', '200 Manhattan Avenue', 11222, 'jill@wentupthehill.edu', 54, 529, 'yes');

insert into patient

values(patient_sequence.nextval, 'Ivy', 'FromGotham', '600 Manhattan Avenue', 11222, 'ivy@gotham.org', 52, 534, 'yes');

insert into patient

values(patient_sequence.nextval, 'Penguin', 'FromGotham', '601 Manhattan Avenue', 11222, 'penguin@gotham.org', 52, 534, 'yes');

insert into patient

values(patient_sequence.nextval, 'Leonard', 'Snart', '602 Manhattan Avenue', 11222, 'snart@dcverse.org', 52, 534, 'yes');

insert into patient

values(patient_sequence.nextval, 'Mick', 'Rory', '602 Manhattan Avenue', 11222, 'rory@dcverse.org', 52, 534, 'yes');

```
insert into drugs
values(drugs_sequence.nextval, 'Adapalene', 'Differin', '50mg');
insert into drugs
values(drugs_sequence.nextval, 'Adapalene', 'Differin', '100mg');
insert into drugs
values(drugs_sequence.nextval, 'Adapalene', 'Differin', '200mg');

insert into guardians
values(guard_sequence.nextval, 'Well', 'Guy', 'thewellguy@thering.com', '715 Manhattan Ave', '11222')
insert into guardians
values(guard_sequence.nextval, 'SomeGuard', 'TheGuard', 'guard1@guardians.com', '600 Manhattan Ave', '11222')
insert into guardians
values(guard_sequence.nextval, 'ImGettingLazy', 'WithTheNames', 'guard2@guardians.com', '610 Manhattan Ave', '11222')
insert into guardians
values(guard_sequence.nextval, 'SomeOtherGuard', 'ThatsAGuardian', 'guard3@guardians.com', '810 Manhattan Ave', '11222')

insert into prescription
values(prescription_sequence.nextval, 140, 1080, 'GirlFrom', 'TheRing', '02202016', 2, 504, 258);
insert into prescription
values(prescription_sequence.nextval, 140, 1081, 'Snow', 'White', '02152016', 2, 504, 258);
insert into prescription
values(prescription_sequence.nextval, 140, 1082, 'Alice', 'Wonder', '02122016', 2, 504, 258);
insert into prescription
values(prescription_sequence.nextval, 140, 1083, 'Sleeping', 'Beauty', '02132016', 2, 504, 256);
insert into prescription
```

values(prescription_sequence.nextval, 141, 1084, 'Jack', 'Bean', '02122016', 1, 504, 258);

insert into prescription

values(prescription_sequence.nextval, 141, 1085, 'Jill', 'Bean', '02112016', 1, 504, 254);

insert into prescription

values(prescription_sequence.nextval, 141, 1086, 'Ivy', 'FromGotham', '02102016', 1, 504, 253);

insert into prescription

values(prescription_sequence.nextval, 141, 1087, 'Penguin', 'FromGotham', '02102016', 1, 504, 251);

insert into prescription

values(prescription_sequence.nextval, 141, 1088, 'Leonard', 'Snart', '02102016', 1, 504, 258);

insert into prescription

values(prescription_sequence.nextval, 142, 1089, 'Mick', 'Rory', '02102016', 1, 504, 256);

insert into guardians_children

values(200, 1080)

insert into guardians_children

values(201, 1081)

insert into guardians_children

values(202, 1082)

insert into guardians_children

values(203, 1083)