TABLE CREATIONS AND MODIFICATIONS

```
CREATE TABLE Drugs
(
DID int NOT NULL,
Generic_Name varchar2(30),
Brand_Name varchar2(30),
Dosage varchar2(5),
PRIMARY KEY(DID)
)
CREATE TABLE PATIENT(
PAT_ID int NOT NULL,
PAT_FNAME varchar2(25),
PAT_LNAME varchar2(25),
ADDRESS varchar2(40),
ZIP VARCHAR2(10),
PAT_EMAIL VARCHAR2(100),
PHARM_ID int NOT NULL,
PRIMARY_PHYSICIAN int NOT NULL;
PRIMARY KEY (PAT_ID),
FOREIGN KEY (PHARM_ID) REFERENCES PHARMACY(PHARM_ID)
FOREIGN KEY (PRIMARY_PHYSICIAN) REFERENCES PRESCRIBER(PRESR_ID)
);
CREATE TABLE P_ALLERGIES
(
P_ALLERGY_ID int NOT NULL,
PRIMARY KEY (P_ALLERGY_ID),
PAT_ID int NOT NULL,
FOREIGN KEY (PAT_ID) REFERENCES PATIENT(PAT_ID),
ALLERGY_TO_DRUG VARCHAR2(25) NOT NULL
)
```

Project #2

```
CREATE TABLE PRESCRIBER(
PRESR_ID int NOT NULL,
PRIMARY KEY (PRESR_ID),
PRE_FNAME VARCHAR2(25) not NULL,
PRE_LNAME VARCHAR2(25) not NULL,
LICENSE_NUMBER int not NULL;
);
CREATE TABLE PRESCRIBER_PHONES(
PPHONE_ID int NOT NULL,
PRIMARY KEY (PPHONE_ID),
PHONE VARCHAR2(12),
PRESR_ID int NOT NULL,
FOREIGN KEY (PRESR_ID) REFERENCES PRESCRIBER(PRESR_ID),
);
CREATE TABLE PRESCRIPTION(
PRESN_ID int NOT NULL,
PRIMARY KEY (PRESN_ID),
DID int NOT NULL,
FOREIGN KEY (DID) REFERENCES DRUGS(DID),
PAT_ID int NOT NULL,
FOREIGN KEY (PAT_ID) REFERENCES PATIENT(PAT_ID)
PRESCRIPTION_FNAME varchar(25) NOT NULL,
PRESCRIPTION_LNAME varchar(25) NOT NULL,
DATE_PRESCRIBED date NOT NULL,
NUMBER_REFILLS int NOT NULL,
PRESR_ID int NOT NULL,
FOREIGN KEY (PRESR_ID) REFERENCE PRESCRIBER(PRESR_ID)
);

CREATE TABLE ORGANIZATION
```

Project #2

```
(
ORG_ID int NOT NULL,
ORG_TYPE varchar2(15) NOT NULL,
ORG_NAME varchar2(40) NOT NULL,
ADDRESS varchar2(40) NOT NULL,
ZIP varchar2(10) NOT NULL,
FOREIGN KEY (ZIP) REFERENCES ZIP_CODES(ZIP),
PRIMARY KEY(ORG_ID)
);

ALTER TABLE ORGANIZATION
ADD CHECK (ORG_TYPE='Hospital' or ORG_TYPE='Nursing Home' or
ORG_TYPE='Clinic' or ORG_TYPE='Other')

CREATE TABLE PRESCRIBER_IN_ORG(
ORG_ID int not NULL,
PRSER_ID int not NULL,
PRIMARY KEY (ORG_ID,PRESR_ID),
FOREIGN KEY (ORG_ID) REFERENCES ORGANIZATION(ORG_ID),
FOREIGN KEY (PRESR_ID) REFERENCES PRESCRIBER(PRESR_ID)
)
```

TABLE States was renamed from DEMO_STATES in the original database. The Following commands were used:

```
RENAME table DEMO_STATES to STATES

ALTER TABLE STATES
ADD PRIMARY KEY(ST)

ALTER TABLE STATES
MODIFY ST int NOT NULL;
```

Project #2

TABLE ZIP_CODES was imported from a separate file. The following commands were used to alter the table:

```
Alter Table ZIP_CODES
ADD PRIMARY KEY (ZIP)

ALTER TABLE ZIP_CODES
MODIFY ZIP varchar(5) not NULL

ALTER TABLE ZIP_CODES
MODIFY CITY varchar(30) not NULL

ALTER TABLE ZIP_CODES
MODIFY ST varchar(2) not NULL;

ALTER TABLE ZIP_CODES
MODIFY COUNTY_BOROUGH varchar(30) not NULL

ALTER TABLE ZIP_CODES
ADD FOREIGN KEY(ST) REFERENCES STATES (ST)
```

INSERTIONS AND CREATION OF SEQUENCES

```
CREATE SEQUENCE
DRUGS_SEQUENCE
START WITH 100
INCREMENT BY 1;
```

Project #2

```
CREATE SEQUENCE
LICENSE_SEQUENCE
START WITH 10000
INCREMENT BY 1

CREATE SEQUENCE
ORGANIZATION_SEQUENCE
START WITH 200
INCREMENT BY 1

CREATE SEQUENCE
PATIENT_SEQUENCE
START WITH 1000
INCREMENT BY 1

CREATE SEQUENCE
PHARMACY_SEQUENCE
START WITH 50
INCREMENT BY 1

CREATE SEQUENCE
PRESCRIBER_SEQUENCE
START WITH 500
INCREMENT BY 1

CREATE SEQUENCE
PRESCRIPTION_SEQUENCE
START WITH 5000
INCREMENT BY 1
```

Project #2

CREATE SEQUENCE

PRESC_PHONE_SEQUENCE

START WITH 500

INCREMENT BY 1

CREATE SEQUENCE

P_ALLERGY_SEQUENCE

START WITH 70

INCREMENT BY 1

INSERT INTO ORGANIZATION

VALUES (ORGANIZATION_SEQUENCE.NEXTVAL, Hospital, Dexter General Hospital, 80-80 Starling Avenue, 48130)

INSERT INTO ORGANIZATION

VALUES (ORGANIZATION_SEQUENCE.NEXTVAL, Hospital, Mount Sinai Beth Israel, 10 Nathan D Perlman Pl, 48130)

203	Hospital	Queens Hospital Center	82-68 164th St	11432
204	Clinic	Manhattan Medical	934 Manhattan Ave # 1	11222
205	Clinic	CityMD	2398 Broadway	10024
206	Clinic	New York University Medical Clinic	160 E 34th St	10016
207	Clinic	Amen Clinics - New York	16 E 40th St	10016
208	Nursing Home	Forest View Center	7120 110th St	11375
209	Nursing Home	Rego Park Health Care	11126 Corona Ave	11368
210	Other	Dr. Frankenstein Home Practice	100 Bedford Avenue	11249

Eventually this is what the table looked like.

Project #2

INSERT ALL

INTO PRESCRIBER

VALUES(PRESCRIBER_SEQUENCE.NEXTVAL, DEREK, SHEPHERD, LICENSE_SEQUENCE.NEXTVAL)

INTO PRESCRIBER_IN_ORG

VALUES(200, PRESCRIBER_SEQUENCE.NEXTVAL)
So on and so forth for each of the doctors. I'm trying to conserved space by not showing all the inserting. The above tables eventually looked like this:

PRESR_ID	PRE_FNAME	PRE_LNAME	LICENSE_NUMBER
522	Derek	Shepherd	10022
523	James	Parkinson	10023
525	Elizabeth	Blackwell	10025
528	Benjamin	Spock	10028
529	Hippocrates	Kos	10029
530	Ben	Carson	10030
531	Jack	Kevorkian	10031
532	Alex	Durant	10032
534	Viktor	Fries	10034
535	Hugo	Strange	10035
500	Viktor	Frankenstein	10000
501	Dr.	Mundo	10001
502	Dr.	Jekkyl	10002
503	Dr.	Oz	10003
504	Dr.	Phil	10004
505	Gregory	House	10005
524	Virginia	Apgar	10024
527	Joseph	Lister	10027
533	Benjamin	Rush	10033
536	Mary	Shelley	10036

ORG_ID	PRESR_ID
200	523
200	524
201	525
203	527
203	528
204	529
204	530
205	522
205	531
206	532
206	533
207	501
207	534
208	503
208	504
208	505
209	502
209	535
210	500
210	536

INSERT INTO PATIENT

VALUES(PATIENT_SEQUENCE.NEXTVAL, 'Miley, 'Cyrus', '6 Pennsylvania Plaza', 10001, 'MileyCyrus@wreckingballs.org', 55, 555)

So on and so forth. The final table looked like this:

PAT_ID	PAT_FNAME	PAT_LNAME	ADDRESS	ZIP	PAT_EMAIL	PHARM_ID	PRIMARY_PHYSICIAN
1002	Miley	Cyrus	6 Pennsylvania Plaza	10001	MileyCyrus@wreckingballs.org	55	522
1003	Donald	Duck	80-24 184th Street	11432	DonaldDuck@disney.com	59	527
1004	Mickey	Mouse	514 Driggs Ave	11222	MickeyMouse@disney.com	53	528
1005	Bill	Clinton	512 Norman Ave	11222	BillClinton@ExWhitehouse.com	53	528
1006	Hillary	Clinton	514 Norman Ave	11222	BillClinton@thinkshesbetterthantrump.org	53	528
1007	Larry	TheCableGuy	134 Greenpoint Ave	11222	LarryTheCableGuy@blueCollarTv.com	54	529
1008	Cookie	Monster	150 Box St	11222	NomNomNomNom@SesameStreet.com	54	529
1009	Kermit	TheFrog	152 Box St	11222	ThatGreenFrog@SesameStreet.com	54	529
1010	Bert	FromSesame	160 Box St	11222	bert@SesameStreet.com	54	529
1011	Ernie	FromSesame	160 Box St	11222	Ernie@SesameStreet.com	54	529
1012	TheMostInteresting	ManInTheWorld	11126 Corona Ave	11368	IDontAlwaysDrinkBeer@ButWhenIDoIDrink.com	60	535
1013	NotTheMostInteresting	ManInTheWorld	11126 Corona Ave	11368	IDontAlwaysDrinkBeer@ButWhenIDoIDontDrink.com	60	535
1014	Betty	Crocker	11126 Corona Ave	11368	BettyCrocker@Crocker.com	59	502
1015	Herbet	ThePervert	7120 110th St	11375	IGotSomePopsiclesInMyBasement@GetYourFatAssBackHere.com	54	504
1016	Nester	TheMolester	7120 110th St	11375	SomeCreep@ICantComeUpWithCreativeEmailAddresses.com	52	505
1017	Henry	TheGreat	84 Wythe Ave	11249	HenryIV@kingsthatareprobablydead.com	52	503
1018	Ivan	TheTerrible	84 Wythe Ave	11249	IvanTheTerribly@motherRussia.ru	59	504
1023	Catherine	TheGreat	84 Wythe Ave	11249	CatherineTheGreat@motherRussia.ru	55	505
1000	Monster	Frankenstein	45 Bedford Avenue	11249	FrankensteinTheMonster1818@modernprometheus.org	54	500
1001	Sarah Jessica	Parker	700 Manhattan Avenue	11222	SarahJessicaParker@horsewhisperer.us	52	502

Insert into p_allergy

Values(p_allergy_sequence.nextval, 1001, ‘Hydrocodone’)

Insert into p_allergy

Values(p_allergy_sequence.nextval, 1000, ‘Rompun’)

Insert into p_allergy

Values(p_allergy_sequence.nextval, 1018, ‘Dextroamphetamine’)

Insert into p_allergy

Values(p_allergy_sequence.nextval, 1018, ‘Rompun’)

Insert into prescription

Values(prescription_sequence.nextval, 102, 1001, ‘Sarah J.’, ‘Parker’, ‘04122016’, 2, 534)

So on and so forth. The 5000 presn_id values were inserted by mistake because I used another sequence accidentally. The table eventually looked like this:

PRESN_ID	DID	PAT_ID	PRESCRIPTION_FNAME	PRESCRIPTION_LNAME	DATE_PRESCRIBED	NUMBER_REFILLS	PRESR_ID
1028	102	1001	Sarah J.	Parker	04/12/2016	2	534
1029	103	1001	Sarah Jessica	Parker	01/02/2016	2	534
1030	104	1008	Cookie	Monster	04/04/2014	1	535
1031	105	1008	Cookie	Monster	05/23/2014	2	535
1032	106	1008	Cookie	Monster	07/15/2015	2	535
1033	107	1008	Cookie	Monster	12/15/2015	8	535
1034	108	1000	Monster	Frankenstein	12/15/2015	2	500
1035	109	1000	Monster	Frankenstein	03/04/2016	2	500
1036	110	1000	Monster	Frankenstein	02/05/2016	4	505
1037	111	1000	Monstah	Frankenstein	12/04/2016	4	505
1038	116	1018	Ivan	TheTerrible	03/14/2013	2	536
1040	115	1002	Miley	Cyrus	06/12/2015	1	533
1041	115	1002	Miley	Cyrus	01/14/2016	2	534
1042	120	1002	Miley	Cyrus	01/14/2016	1	534
1043	122	1002	Miley	Cyrus	01/14/2016	1	534
1044	125	1002	Miley	Cyrus	01/14/2016	2	534
1047	130	1005	Bill	Clinton	03/14/2016	1	531
1048	125	1005	Bill	Clinton	03/14/2016	1	531
1049	128	1005	Bill	Clinton	04/20/2015	1	501
1050	122	1005	Bill	Clinton	06/20/2015	2	504
1051	125	1006	Hillary	Clinton	06/20/2015	3	504
1052	125	1006	Hillary	Clinton	04/12/2016	3	505
1058	121	1013	NotTheMostInteresting	ManInTheWorld	04/03/2016	1	522
1059	127	1012	TheMostInteresting	ManInTheWorld	04/03/2016	1	535
1061	117	1003	Donald	Duck	01/24/2015	1	522
1062	117	1003	Donald	Duk	08/14/2015	2	529
1063	117	1009	Kermit	TheFrog	12/11/2015	1	524
1064	118	1009	Kermit	Frog	04/11/2016	2	525
1065	128	1010	Bert	FromSesame	04/13/2016	2	530
1066	128	1011	Ernie	FromSesame	04/13/2016	2	530
5007	109	1001	Sarah Jessica	Parker	01/24/2015	1	505
5008	103	1000	Monster	Frankenstein	04/20/2015	1	505
5009	108	1008	Cookie	Monster	04/13/2016	2	505

Insert into prescriber_phones

Values(presc_phone_sequence.nextval, ‘646-703-1341’, 500)

Project #2

Insert into prescriber_phones

Values(presc_phone_sequence.nextval, ‘646-503-1341’, 501)

So on and so forth. Table looked like this:

PPHONE_ID	PHONE	PRESR_ID
350	646-703-1341	500
355	646-503-1341	501
360	646-503-1331	502
365	646-503-1231	503
370	347-503-1231	504
375	347-503-1211	505
380	347-103-1211	522
385	347-203-1211	523
390	347-303-1211	524
395	347-403-1211	525
400	347-413-1211	527
405	347-413-2211	528
410	347-414-2299	529
415	347-414-3399	530
420	347-414-3309	531
425	347-415-3309	532
430	347-415-3409	533
435	327-415-3409	534
440	317-415-3409	534
445	317-315-3409	535
450	317-315-2409	536
465	718-315-4409	536
470	718-315-4109	530
480	718-327-4109	525

Question 1

Project #2

Identify the most popular medication prescribed in New York in the last 10 months. Display the drug name and number prescribed. Display one row for each drug. Order alphabetically by drug name.

```
select generic_name, count(1) as Amount
from prescription NATURAL JOIN drugs NATURAL JOIN prescriber_in_org
NATURAL JOIN organization NATURAL JOIN zip_codes NATURAL JOIN states
where date_prescribed >= '06132015' AND date_prescribed <= '04132016' AND
st='NY'
group by generic_name
having count(1) > 2
order by 1 asc;
```

GENERIC_NAME	AMOUNT
Alprazolam	8
Amphetamines	3
Hydrocodone	3
Oxycodone	3

Project #2

Question 2

Identify the medication history of Cookie Monster. Display the patient name, physician, drug, date of prescription and dosage. Order chronologically by date.

```
SELECT  prescription_fname||', '||prescription_lname Patient_Name,
pre_fname||', '||pre_lname prescriber_name,  generic_name,
date_prescribed, dosage
FROM prescriber INNER JOIN prescription
ON prescriber.presr_ID=prescription.presr_id
INNER JOIN drugs
ON prescription.DID = drugs.DID
WHERE prescription_fname='Cookie' and prescription_lname='Monster'
ORDER by date_prescribed asc;
```

PATIENT_NAME	PRESCRIBER_NAME	GENERIC_NAME	DATE_PRESCRIBED	DOSAGE
Cookie, Monster	Hugo, Strange	Metadate CD	04/04/2014	20mg
Cookie, Monster	Hugo, Strange	Metadate CD	05/23/2014	50mg
Cookie, Monster	Hugo, Strange	Metadate ER	07/15/2015	10mg
Cookie, Monster	Hugo, Strange	Metadate ER	12/15/2015	20mg

Project #2

Question 3

Identify prescriptions affiliated with Amen Clinics in the last 12 months. Display the doctor name, organization, drug name, patient name and date. Order by doctor name, drug and patient name.

```
SELECT pre_fname||', '||pre_lname Prescriber_Name, org_name Organization,
generic_name Drug_Name, prescription_fname||', '||prescription_lname
Patient_Name, date_prescribed
FROM prescriber a INNER JOIN prescription b
ON a.presr_ID = b.presr_ID
INNER JOIN prescriber_in_org c
ON c.presr_ID = b.presr_id
NATURAL JOIN organization d
NATURAL JOIN prescriber_in_org e
INNER JOIN drugs f
ON f.DID = b.DID
WHERE org_name LIKE '%Amen Clinics%' AND date_prescribed >= '04132015' AND
date_prescribed <= '04132016'
ORDER by pre_fname, generic_name, prescription_fname
```

PRESCRIBER_NAME	ORGANIZATION	DRUG_NAME	PATIENT_NAME	DATE_PRESCRIBED
Dr., Mundo	Amen Clinics – New York	Amphetamines	Bill, Clinton	04/20/2015
Viktor, Fries	Amen Clinics – New York	Alprazolam	Miley, Cyrus	01/14/2016
Viktor, Fries	Amen Clinics – New York	Alprazolam	Miley, Cyrus	01/14/2016
Viktor, Fries	Amen Clinics – New York	Alprazolam	Miley, Cyrus	01/14/2016
Viktor, Fries	Amen Clinics – New York	Oxycodone	Miley, Cyrus	01/14/2016
Viktor, Fries	Amen Clinics – New York	Rompun	Sarah J., Parker	04/12/2016
Viktor, Fries	Amen Clinics – New York	Rompun	Sarah Jessica, Parker	01/02/2016

Project #2

Question 4

A doctor creates a new prescription for Cookie Monster. What SQL operations are required to create a new prescription.

```
INSERT into prescription
VALUES(prescription_sequence.nextval,
    (SELECT drugs.DID
     FROM drugs
     WHERE drugs.generic_name = 'Hydrocodone' AND drugs.dosage='50mg'),
    (SELECT patient.pat_id
     FROM patient
     WHERE patient.pat_fname='Cookie' and patient.pat_lname='Monster'),
    'Cookie', 'Monster', '04132016', 2,
    (SELECT a.presr_id
     FROM prescriber a
     WHERE a.PRE_FNAME='Gregory' and a.PRE_LNAME='House')
)
```

```
1 row(s) inserted.
```

PRESN_ID	DID	PAT_ID	PRESCRIPTION_FNAME	PRESCRIPTION_LNAME	DATE_PRESCRIBED	NUMBER_REFILLS	PRESR_ID
5008	103	1000	Monster	Frankenstein	04/20/2015	1	505
5009	108	1008	Cookie	Monster	04/13/2016	2	505

Question 5

Identify patients with an allergy to the medication prescribed. Display the patient name, drug, doctor name and doctor phone number.

```
SELECT b.prescription_fname||', '||b.prescription_lname Patient_Name,
a.allergic_to_drug, c.generic_name, d.pre_fname||', '||pre_lname
Doctor_Name, e.phone
from p_allergies a INNER JOIN prescription b
ON a.pat_id = b.pat_id
INNER JOIN drugs c
ON a.allergic_to_drug = c.generic_name
INNER JOIN prescriber d
ON d.presr_id = b.presr_id
INNER JOIN prescriber_phones e
ON e.presr_id = b.presr_id
WHERE b.DID = c.DID
```

PATIENT_NAME	ALLERGIC_TO_DRUG	GENERIC_NAME	DOCTOR_NAME	PHONE
Monster, Frankenstein	Rompun	Rompun	Gregory, House	347-503-1211
Sarah Jessica, Parker	Hydrocodone	Hydrocodone	Gregory, House	347-503-1211

Project #2

Question 6

Identify the prescriber who assigned the most Hydrocodone in the last 12 months. Display the prescriber, license number, drug name and number of prescriptions. Display one row for each prescriber, license number and drug name. Display the prescriber and drug with the highest number first.

```
SELECT *
FROM(
SELECT a.pre_fname||', '||a.pre_lname Doctor_Name, generic_name,
a.license_number, count(pre_lname) as counter
FROM prescriber a INNER JOIN prescription b
ON a.presr_id = b.presr_id
INNER JOIN drugs c
ON b.DID = c.DID
WHERE c.generic_name='Hydrocodone' AND b.date_prescribed >= '04132015' AND
b.date_prescribed <= '04132016'
GROUP BY pre_fname, pre_lname, generic_name, license_number
Order by Counter desc
)
WHERE ROWNUM <=1
```

DOCTOR_NAME	GENERIC_NAME	LICENSE_NUMBER	COUNTER
Viktor, Frankenstein	Hydrocodone	10000	2

Project #2

Question 7

Identify patients with no allergies on file. Display the patient name and email. Used a nested select to answer this question.

```
SELECT pat_fname||', '||pat_lname, pat_email
from patient
where pat_id NOT IN(
   SELECT pat_id
   FROM p_allergies
)
```

PAT_FNAME||','||PAT_LNAME	PAT_EMAIL
Miley, Cyrus	MileyCyrus@wreckingballs.org
Donald, Duck	DonaldDuck@disney.com
Mickey, Mouse	MickeyMouse@disney.com
Bill, Clinton	BillClinton@ExWhitehouse.com
Hillary, Clinton	BillClinton@thinkshesbetterthantrump.org
Larry, TheCableGuy	LarryTheCableGuy@blueCollarTv.com
Cookie, Monster	NomNomNomNom@SesameStreet.com
Kermit, TheFrog	ThatGreenFrog@SesameStreet.com
Bert, FromSesame	bert@SesameStreet.com
Ernie, FromSesame	Ernie@SesameStreet.com
More than 10 rows available. Increase rows selector to view more rows.	

Project #2

Question 8

Identify patients with misspelled names – where the name on a new prescription doesn't match the name on a previously assigned prescription. Use other fields to identify this patient in the database. Display the patient name and address.

```
SELECT PAT_FNAME, PAT_LNAME, PAT_EMAIL
FROM(
  SELECT PRESCRIPTION_FNAME, PRESCRIPTION_LNAME, PAT_FNAME, PAT_LNAME,
PAT_EMAIL
  FROM prescription a INNER JOIN patient b
  ON a.pat_id = b.pat_id
  MINUS
  SELECT PRESCRIPTION_FNAME, PRESCRIPTION_LNAME, PAT_FNAME, PAT_LNAME,
PAT_EMAIL
  FROM prescription a INNER JOIN patient b
  ON a.pat_id = b.pat_id
  WHERE (a.prescription_fname=b.pat_fname and a.prescription_lname
=b.pat_lname)
)
```

Project #2

PAT_FNAME	PAT_LNAME	PAT_EMAIL
Donald	Duck	DonaldDuck@disney.com
Kermit	TheFrog	ThatGreenFrog@SesameStreet.com
Monster	Frankenstein	FrankensteinTheMonster1818@modernprometheus.org
Sarah Jessica	Parker	SarahJessicaParker@horsewhisperer.us

Question 9

Identify pharmacies near Cookie Monster. Display the pharmacy name, address, city, state, zip code and phone number. Use a nested select to answer this question.

```
select *
from(
   SELECT b.pharm_name as Pharmacy_Name, b.address, c.city, c.zip,
d.state_name as State, b.phone
   FROM patient a INNER JOIN pharmacy b
   ON a.zip=b.zip
   INNER JOIN zip_codes c
   on b.zip = c.zip
   INNER JOIN states d
   on c.st=d.st
   where a.pat_fname LIKE 'Cookie' and a.pat_lname LIKE 'Monster'
   )
```

PHARMACY_NAME	ADDRESS	CITY	ZIP	STATE	PHONE
Duane Reade	893 Manhattan Ave	BROOKLYN	11222	NEW YORK	718-689-2642
Rite Aid	783 Manhattan Ave	BROOKLYN	11222	NEW YORK	718-339-1641
Rite Aid	723-725 Manhattan Ave	BROOKLYN	11222	NEW YORK	718-329-1641

Question 10

Identify patients without a prescription in the last 2 months. Display patient name. Use a nested select to answer this question.

```
SELECT *
FROM(
   SELECT b.pat_fname||', '||b.pat_lname as Patient_Name
   FROM prescription a INNER JOIN patient b
   ON a.pat_id = b.pat_id
   WHERE a.date_prescribed <= '04132016' and a.date_prescribed >=
'02132016'
   )
```

PATIENT_NAME
Sarah Jessica, Parker
Bill, Clinton
Bill, Clinton
Hillary, Clinton
NotTheMostInteresting, ManInTheWorld
TheMostInteresting, ManInTheWorld
Kermit, TheFrog
Bert, FromSesame
Ernie, FromSesame
Monster, Frankenstein

Question 11

- Notice: Images are used instead of plain text to keep formatting style

Project #2

Object Type **TABLE** Object **PRESCRIBER_IN_ORG**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRESCRIBER_IN_ORG	ORG_ID	NUMBER	22	-	0	1	-	-	-
	PRESR_ID	NUMBER	22	-	0	2	-	-	-
									1 - 2

Object Type **TABLE** Object **PRESCRIBER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRESCRIBER	PRESR_ID	NUMBER	22	-	0	1	-	-	-
	PRE_FNAME	VARCHAR2	25	-	-	-	-	-	-
	PRE_LNAME	VARCHAR2	25	-	-	-	-	-	-
	LICENSE_NUMBER	NUMBER	22	-	0	-	-	-	-
									1 - 4

Object Type **TABLE** Object **PRESCRIPTION**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRESCRIPTION	PRESN_ID	NUMBER	22	-	0	1	-	-	-
	DID	NUMBER	22	-	0	-	-	-	-
	PAT_ID	NUMBER	22	-	0	-	-	-	-
	PRESCRIPTION_FNAME	VARCHAR2	25	-	-	-	-	-	-
	PRESCRIPTION_LNAME	VARCHAR2	25	-	-	-	-	-	-
	DATE_PRESCRIBED	DATE	7	-	-	-	-	-	-
	NUMBER_REFILLS	NUMBER	22	-	0	-	-	-	-
	PRESR_ID	NUMBER	22	-	0	-	-	-	-
									1 - 8

Object Type **TABLE** Object **PRESCRIBER_PHONES**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRESCRIBER_PHONES	PPHONE_ID	NUMBER	22	-	0	1	-	-	-
	PHONE	VARCHAR2	12	-	-	-	-	-	-
	PRESR_ID	NUMBER	22	-	0	-	-	-	-
									1 - 3

Object Type **TABLE** Object **ORGANIZATION**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORGANIZATION	ORG_ID	NUMBER	22	-	0	1	-	-	-
	ORG_TYPE	VARCHAR2	15	-	-	-	-	-	-
	ORG_NAME	VARCHAR2	40	-	-	-	-	-	-
	ADDRESS	VARCHAR2	40	-	-	-	-	-	-
	ZIP	VARCHAR2	10	-	-	-	-	-	-
									1 - 5

Project #2

Object Type **TABLE** Object **STATES**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STATES	ST	VARCHAR2	30	-	-	1	-	-	-
	STATE_NAME	VARCHAR2	30	-	-	-	✓	-	-
									1 - 2

Object Type **TABLE** Object **ZIP_CODES**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ZIP_CODES	ZIP	VARCHAR2	5	-	-	1	-	-	-
	CITY	VARCHAR2	30	-	-	-	-	-	-
	ST	VARCHAR2	2	-	-	-	-	-	-
	COUNTY_BOROUGH	VARCHAR2	30	-	-	-	-	-	-
									1 - 4

Object Type **TABLE** Object **PHARMACY**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PHARMACY	PHARM_ID	NUMBER	22	-	0	1	-	-	-
	PHARM_NAME	VARCHAR2	30	-	-	-	-	-	-
	ADDRESS	VARCHAR2	30	-	-	-	-	-	-
	ZIP	VARCHAR2	5	-	-	-	-	-	-
	PHONE	VARCHAR2	12	-	-	-	-	-	-
									1 - 5

Object Type **TABLE** Object **DRUGS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DRUGS	DID	NUMBER	22	-	0	1	-	-	-
	GENERIC_NAME	VARCHAR2	30	-	-	-	-	-	-
	BRAND_NAME	VARCHAR2	30	-	-	-	-	-	-
	DOSAGE	VARCHAR2	7	-	-	-	-	-	-
									1 - 4

Project #2

Object Type **TABLE** Object **PATIENT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PATIENT	PAT_ID	NUMBER	22	-	0	1	-	-	-
	PAT_FNAME	VARCHAR2	25	-	-	-	-	-	-
	PAT_LNAME	VARCHAR2	25	-	-	-	-	-	-
	ADDRESS	VARCHAR2	40	-	-	-	-	-	-
	ZIP	VARCHAR2	10	-	-	-	-	-	-
	PAT_EMAIL	VARCHAR2	100	-	-	-	-	-	-
	PHARM_ID	NUMBER	22	-	0	-	-	-	-
	PRIMARY_PHYSICIAN	NUMBER	22	-	0	-	-	-	-

1 - 8

Object Type **TABLE** Object **P_ALLERGIES**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
P_ALLERGIES	P_ALLERGY_ID	NUMBER	22	-	0	1	-	-	-
	PAT_ID	NUMBER	22	-	0	-	-	-	-
	ALLERGIC_TO_DRUG	VARCHAR2	25	-	-	-	-	-	-

1 - 3