HOSPITAL INPATIENT DATABASE END-TO-END PROJECT

CREATING ERD, TABLES, PROCEDURES, TRIGGERS, AND RUNNING QUERIES

TABLE OF CONTENTS

PROJECT DIRECTION OVERVIEW

The project is to create a database for hospital admissions (In-patients). The database will contain data about patients who get admitted to the hospital. It will be used by doctors, nurses, technicians, administrative staff, and patients. Not everyone will have access to all data and this will be customized for each type of user category.

What kind of data it will contain – as it is an in-patient records database it will contain

1. Patient's personal details and history

2. Patient's admission details

3. Patient's tests and results

4. Patient's diagnosis and management

5. Hospital's doctors and staff database

6. Hospital's test database

7. Hospital's procedure database

8. Patients' payment billing details

USE CASES AND FIELDS

The use cases and fields are many, and 2 examples are

1. Patient Admission

Field	What it stores	Why it is needed
Patient ID	Unique alphanumeric value to identify patients	Instead of the patient's name, this will be used for record-keeping to maintain confidentiality For logging in to an app made for patients.
Admission ID	Number value assigned for each visit	For getting details of a single visit (each patient can have multiple admissions)
Date of admission	Date of admission into the hospital	Record, to calculate the duration of stay, billing for room
Date of discharge	Date of discharge from the hospital	Record, to calculate the duration of stay, billing for room

Duration of stay	Number of days in the hospital	Record, billing
Admitting physician (Doctor ID)	The doctor under whom the patient primarily got admitted	Record and billing (cost structure of that department or doctor)
Room number	Rooms in which patient admitted during the stay – this may be broken down into 3-4 fields as long stays can have different rooms.	Record of this information
Bill amount	Final bill amount at end of stay	Record of this information for the insurance and accounts department of the hospital.

2. Healthcare Staff data

Field	What it stores	Why it is needed
Employee ID	Unique alphanumeric value to identify staff	For record keeping and identification of staff. Will be used to access patient records like test results. Can be used to access their own records (salary, attendance)
First Name	First name (from a govt ID)	For record-keeping, identification, and communication.
Last Name	Last name (from a govt ID)	For record-keeping, identification, and communication.
Date of birth	DoB from a govt issued ID	For record-keeping, identification, and communication.
Age	Age calculated from DoB	For record-keeping and identification
Gender	Staff's gender	For record-keeping and identification
Address	Staff's address	For record-keeping, identification, and communication.
Employee number	Serial number of the employee	Data for hospital
Medical license/registration	Registration or license number	Record keeping and medico-legal reasons
Educational qualification	Degrees, diplomas, and certifications of healthcare staff	Record keeping and patient information.
Department	Department in which staff primarily works (lab, ER, Medicine, etc.)	Record keeping and information to authorized users

STRUCTURAL DATABASE RULES

In our scenario, each patient has one account so patient and patient account are the same for this exercise

1. Each patient (patient account) can have one to many admissions, each admission has only one patient.

Each patient will have at least one admission as it is an in-patient account database and an account will be created only when hospital admission takes place. Each admission will only have one patient.

2. In each admission patient will have one patient tests list (patient tests - list of tests patient undergoes), and each test list (patient tests) will be for one admitted patient (patient admissions)

3. In each admission patient will have one procedures list (patient procedures - list of procedures patient undergoes), and each procedure list (patient procedures) will be for one admitted patient (patient admissions)

An admitted patient will have one list for all tests and one for all procedures, as these lists are only for one patient, they may be done to only admit patient (patient admission)

4. Patient tests may have zero to many testIDs (actual test), and each test may be done to zero to many patients.

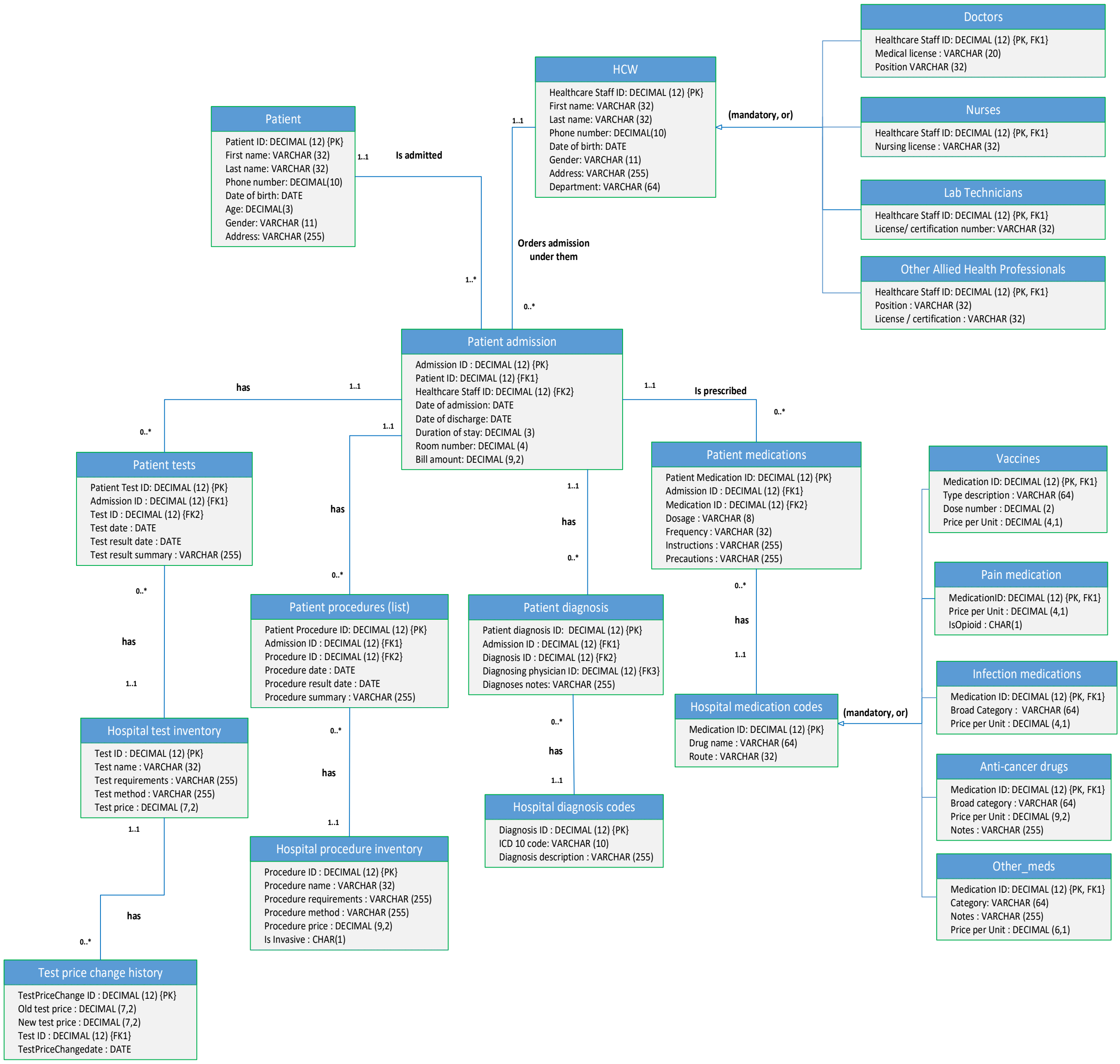
5. Patient procedures may have zero to many procedureIDs (actual procedure), and each procedure may be done to zero to many patients.

Each patient may undergo several tests and procedures. The hospital has a list of tests and procedures available and each of them may be done to zero to several patients and so, may appear in zero to many patient tests and patient procedure lists.

6. Each admission has one management data (diagnosis and medicines), each management data has one admission (one admit patient)

7. Each admission is under one doctor (HCW) and each doctor can have many admissions under them.

FULL DBMS PHYSICAL ERD



Doctors
Healthcare Staff ID: DECIMAL (12) {PK, FK1}
Medical license : VARCHAR (20)
Position VARCHAR (32)

HCW
Healthcare Staff ID: DECIMAL (12) {PK}
First name: VARCHAR (32)
Last name: VARCHAR (32)
Phone number: DECIMAL(10)
Date of birth: DATE
Gender: VARCHAR (11)
Address: VARCHAR (255)
Department: VARCHAR (64)

Patient
Patient ID: DECIMAL (12) {PK}
First name: VARCHAR (32)
Last name: VARCHAR (32)
Phone number: DECIMAL(10)
Date of birth: DATE
Age: DECIMAL(3)
Gender: VARCHAR (11)
Address: VARCHAR (255)

Nurses
Healthcare Staff ID: DECIMAL (12) {PK, FK1}
Nursing license : VARCHAR (32)

Lab Technicians
Healthcare Staff ID: DECIMAL (12) {PK, FK1}
License/ certification number: VARCHAR (32)

Other Allied Health Professionals
Healthcare Staff ID: DECIMAL (12) {PK, FK1}
Position : VARCHAR (32)
License / certification : VARCHAR (32)

Patient admission
Admission ID : DECIMAL (12) {PK}
Patient ID: DECIMAL (12) {FK1}
Healthcare Staff ID: DECIMAL (12) {FK2}
Date of admission: DATE
Date of discharge: DATE
Duration of stay: DECIMAL (3)
Room number: DECIMAL (4)
Bill amount: DECIMAL (9,2)

Patient tests
Patient Test ID: DECIMAL (12) {PK}
Admission ID : DECIMAL (12) {FK1}
Test ID : DECIMAL (12) {FK2}
Test date : DATE
Test result date : DATE
Test result summary : VARCHAR (255)

Patient medications
Patient Medication ID: DECIMAL (12) {PK}
Admission ID : DECIMAL (12) {FK1}
Medication ID : DECIMAL (12) {FK2}
Dosage : VARCHAR (8)
Frequency : VARCHAR (32)
Instructions : VARCHAR (255)
Precautions : VARCHAR (255)

Vaccines
Medication ID: DECIMAL (12) {PK, FK1}
Type description : VARCHAR (64)
Dose number : DECIMAL (2)
Price per Unit : DECIMAL (4,1)

Pain medication
MedicationID: DECIMAL (12) {PK, FK1}
Price per Unit : DECIMAL (4,1)
IsOpioid : CHAR(1)

Patient procedures (list)
Patient Procedure ID: DECIMAL (12) {PK}
Admission ID : DECIMAL (12) {FK1}
Procedure ID : DECIMAL (12) {FK2}
Procedure date : DATE
Procedure result date : DATE
Procedure summary : VARCHAR (255)

Patient diagnosis
Patient diagnosis ID: DECIMAL (12) {PK}
Admission ID : DECIMAL (12) {FK1}
Diagnosis ID : DECIMAL (12) {FK2}
Diagnosing physician ID: DECIMAL (12) {FK3}
Diagnoses notes: VARCHAR (255)

Infection medications
Medication ID: DECIMAL (12) {PK, FK1}
Broad Category : VARCHAR (64)
Price per Unit : DECIMAL (4,1)

Hospital medication codes
Medication ID: DECIMAL (12) {PK}
Drug name : VARCHAR (64)
Route : VARCHAR (32)

Hospital test inventory
Test ID : DECIMAL (12) {PK}
Test name : VARCHAR (32)
Test requirements : VARCHAR (255)
Test method : VARCHAR (255)
Test price : DECIMAL (7,2)

Anti-cancer drugs
Medication ID: DECIMAL (12) {PK, FK1}
Broad category : VARCHAR (64)
Price per Unit : DECIMAL (9,2)
Notes : VARCHAR (255)

Hospital diagnosis codes
Diagnosis ID : DECIMAL (12) {PK}
ICD 10 code: VARCHAR (10)
Diagnosis description : VARCHAR (255)

Hospital procedure inventory
Procedure ID : DECIMAL (12) {PK}
Procedure name : VARCHAR (32)
Procedure requirements : VARCHAR (255)
Procedure method : VARCHAR (255)
Procedure price : DECIMAL (9,2)
Is Invasive : CHAR(1)

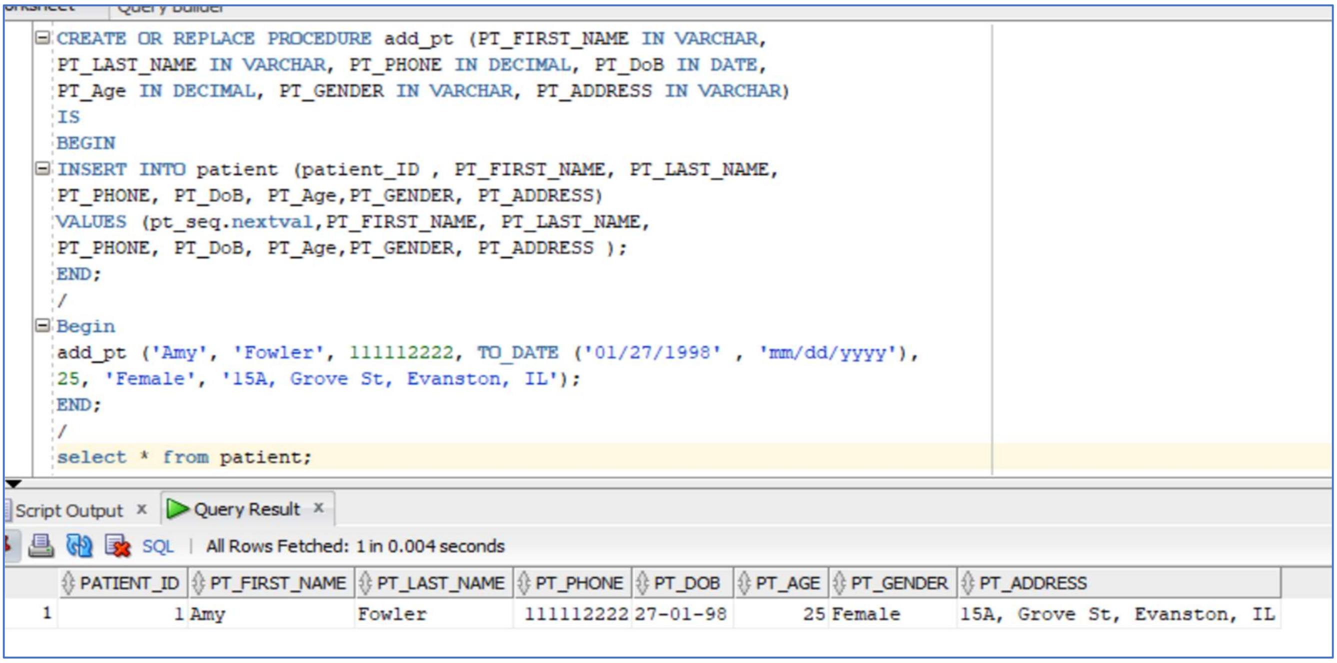
Other_meds
Medication ID: DECIMAL (12) {PK, FK1}
Category: VARCHAR (64)
Notes : VARCHAR (255)
Price per Unit : DECIMAL (6,1)

Test price change history
TestPriceChange ID : DECIMAL (12) {PK}
Old test price : DECIMAL (7,2)
New test price : DECIMAL (7,2)
Test ID : DECIMAL (12) {FK1}
TestPriceChangedate : DATE

STORED PROCEDURE EXECUTION AND EXPLANATIONS

There are 6 stored procedures in this project of which 2 are shown here

1. The ‘add_pt’ procedure to enter patient data in their account

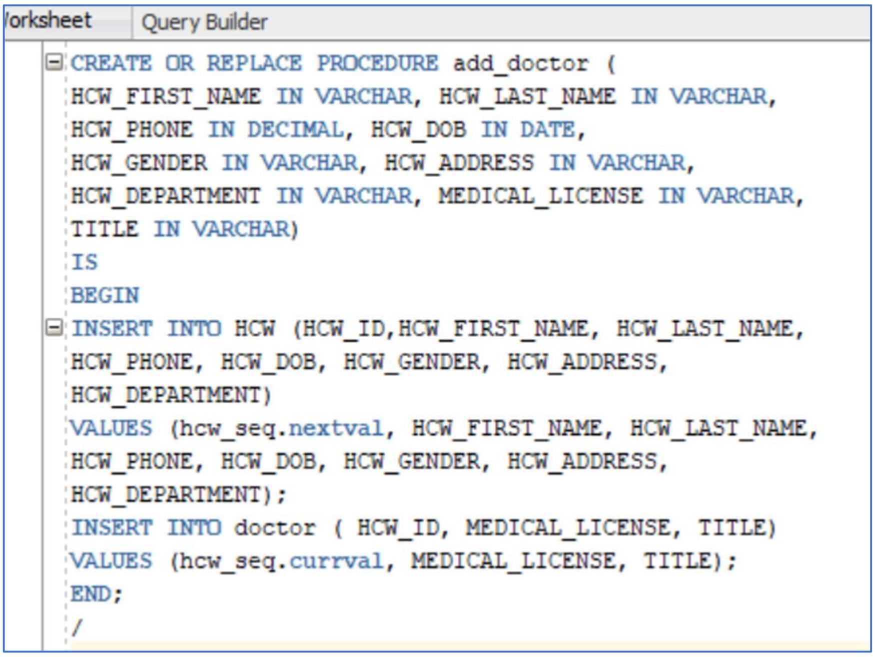


```
CREATE OR REPLACE PROCEDURE add_pt (PT_FIRST_NAME IN VARCHAR,
PT_LAST_NAME IN VARCHAR, PT_PHONE IN DECIMAL, PT_DoB IN DATE,
PT_Age IN DECIMAL, PT_GENDER IN VARCHAR, PT_ADDRESS IN VARCHAR)
IS
BEGIN
INSERT INTO patient (patient_ID , PT_FIRST_NAME, PT_LAST_NAME,
PT_PHONE, PT_DoB, PT_Age,PT_GENDER, PT_ADDRESS)
VALUES (pt_seq.nextval,PT_FIRST_NAME, PT_LAST_NAME,
PT_PHONE, PT_DoB, PT_Age,PT_GENDER, PT_ADDRESS );
END;
/
Begin
add_pt ('Amy', 'Fowler', 111112222, TO_DATE ('01/27/1998' , 'mm/dd/yyyy'),
25, 'Female', '15A, Grove St, Evanston, IL');
END;
/
select * from patient;
```

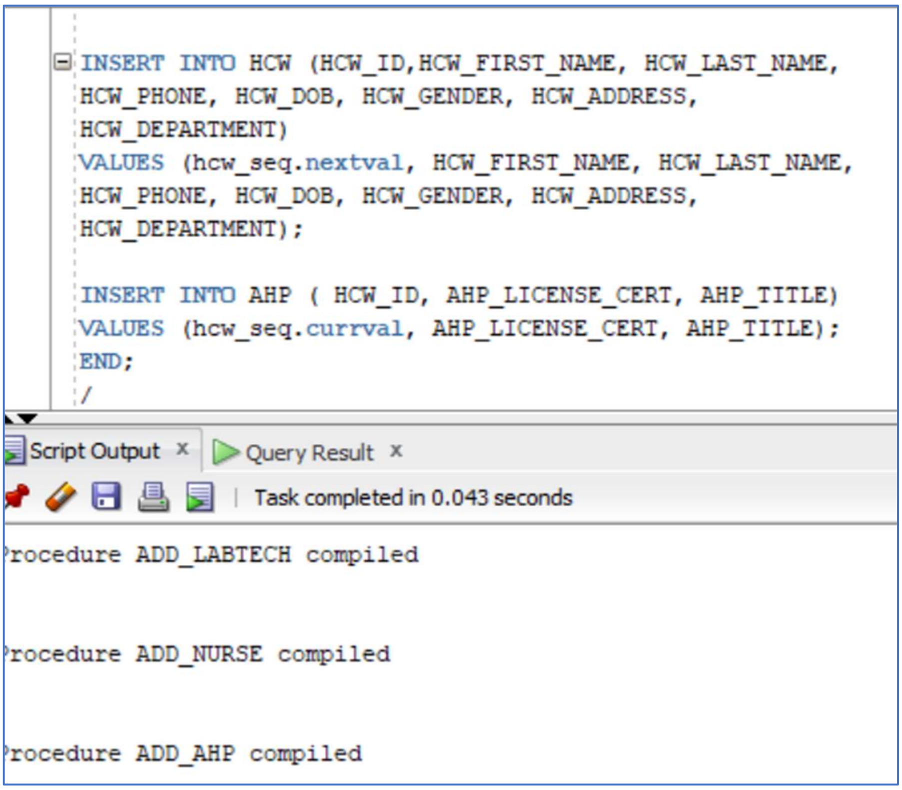
All Rows Fetched: 1 in 0.004 seconds

```
    PATIENT_ID  PT_FIRST_NAME  PT_LAST_NAME  PT_PHONE   PT_DOB    PT_AGE  PT_GENDER  PT_ADDRESS
1            1  Amy            Fowler        111112222  27-01-98      25  Female     15A, Grove St, Evanston, IL
```

2. Procedure to add Health Care Workers data – 4 subtypes (Doctor, Nurse, Labtech, AHP)



```
CREATE OR REPLACE PROCEDURE add_doctor (
HCW_FIRST_NAME IN VARCHAR, HCW_LAST_NAME IN VARCHAR,
HCW_PHONE IN DECIMAL, HCW_DOB IN DATE,
HCW_GENDER IN VARCHAR, HCW_ADDRESS IN VARCHAR,
HCW_DEPARTMENT IN VARCHAR, MEDICAL_LICENSE IN VARCHAR,
TITLE IN VARCHAR)
IS
BEGIN
INSERT INTO HCW (HCW_ID,HCW_FIRST_NAME, HCW_LAST_NAME,
HCW_PHONE, HCW_DOB, HCW_GENDER, HCW_ADDRESS,
HCW_DEPARTMENT)
VALUES (hcw_seq.nextval, HCW_FIRST_NAME, HCW_LAST_NAME,
HCW_PHONE, HCW_DOB, HCW_GENDER, HCW_ADDRESS,
HCW_DEPARTMENT);
INSERT INTO doctor ( HCW_ID, MEDICAL_LICENSE, TITLE)
VALUES (hcw_seq.currval, MEDICAL_LICENSE, TITLE);
END;
/
```

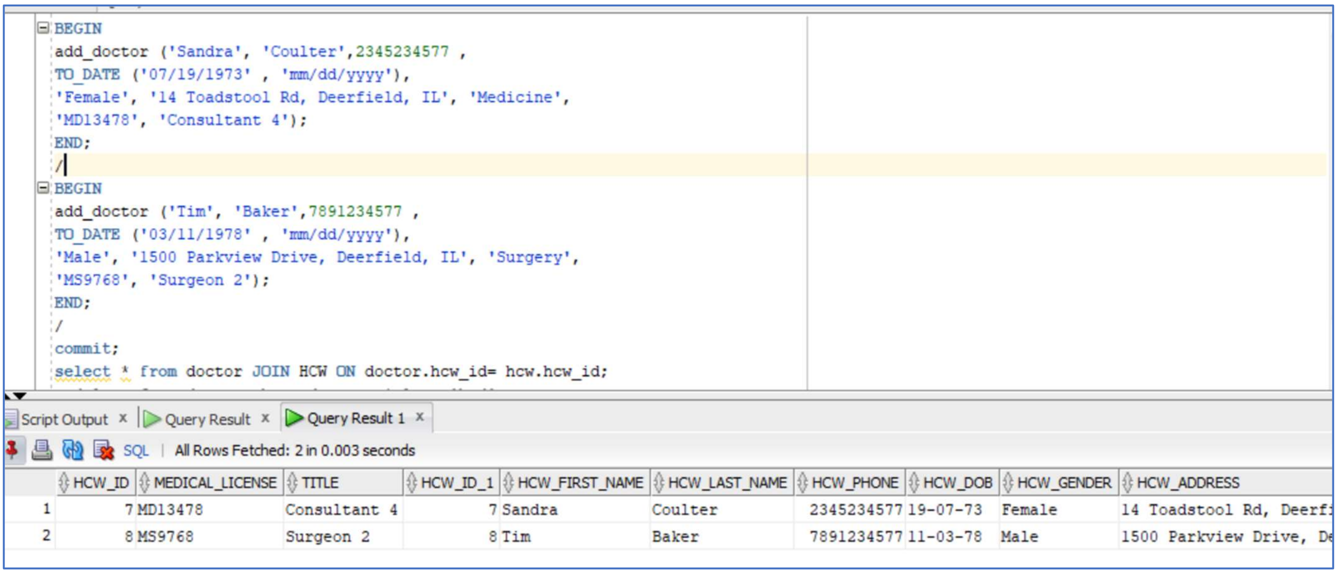


```
INSERT INTO HCW (HCW_ID,HCW_FIRST_NAME, HCW_LAST_NAME,
HCW_PHONE, HCW_DOB, HCW_GENDER, HCW_ADDRESS,
HCW_DEPARTMENT)
VALUES (hcw_seq.nextval, HCW_FIRST_NAME, HCW_LAST_NAME,
HCW_PHONE, HCW_DOB, HCW_GENDER, HCW_ADDRESS,
HCW_DEPARTMENT);

INSERT INTO AHP ( HCW_ID, AHP_LICENSE_CERT, AHP_TITLE)
VALUES (hcw_seq.currval, AHP_LICENSE_CERT, AHP_TITLE);
END;
/
```

Task completed in 0.043 seconds

```
Procedure ADD_LABTECH compiled


Procedure ADD_NURSE compiled


Procedure ADD_AHP compiled
```

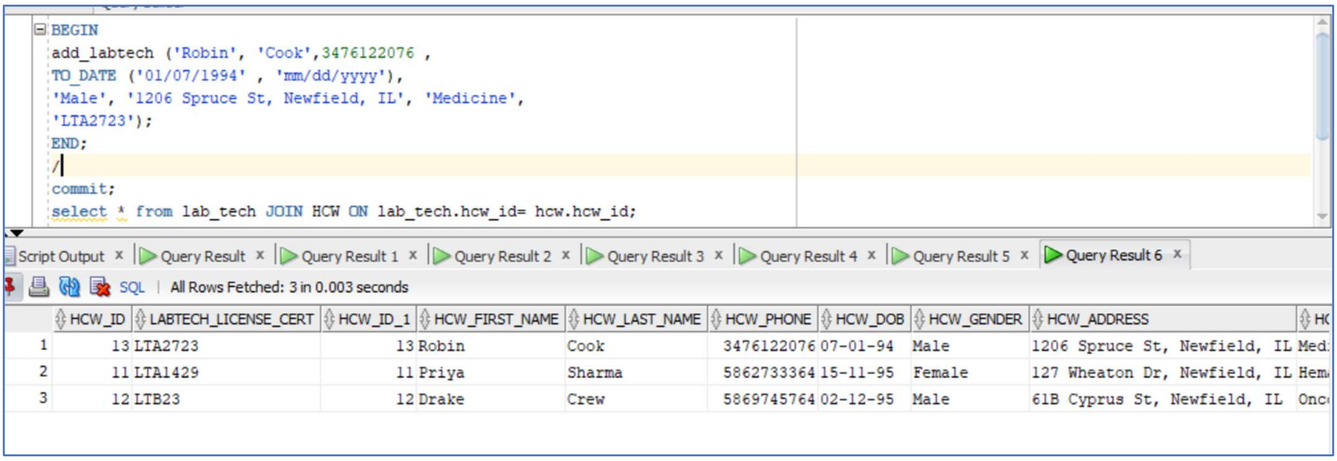
Using the procedure



```
BEGIN
add_doctor ('Sandra', 'Coulter',2345234577 ,
TO_DATE ('07/19/1973' , 'mm/dd/yyyy'),
'Female', '14 Toadstool Rd, Deerfield, IL', 'Medicine',
'MD13478', 'Consultant 4');
END;
/
BEGIN
add_doctor ('Tim', 'Baker',7891234577 ,
TO_DATE ('03/11/1978' , 'mm/dd/yyyy'),
'Male', '1500 Parkview Drive, Deerfield, IL', 'Surgery',
'MS9768', 'Surgeon 2');
END;
/
commit;
select * from doctor JOIN HCW ON doctor.hcw_id= hcw.hcw_id;
```

All Rows Fetched: 2 in 0.003 seconds

	HCW_ID	MEDICAL_LICENSE	TITLE	HCW_ID_1	HCW_FIRST_NAME	HCW_LAST_NAME	HCW_PHONE	HCW_DOB	HCW_GENDER	HCW_ADDRESS
1	7	MD13478	Consultant 4	7	Sandra	Coulter	2345234577	19-07-73	Female	14 Toadstool Rd, Deerf:
2	8	MS9768	Surgeon 2	8	Tim	Baker	7891234577	11-03-78	Male	1500 Parkview Drive, De



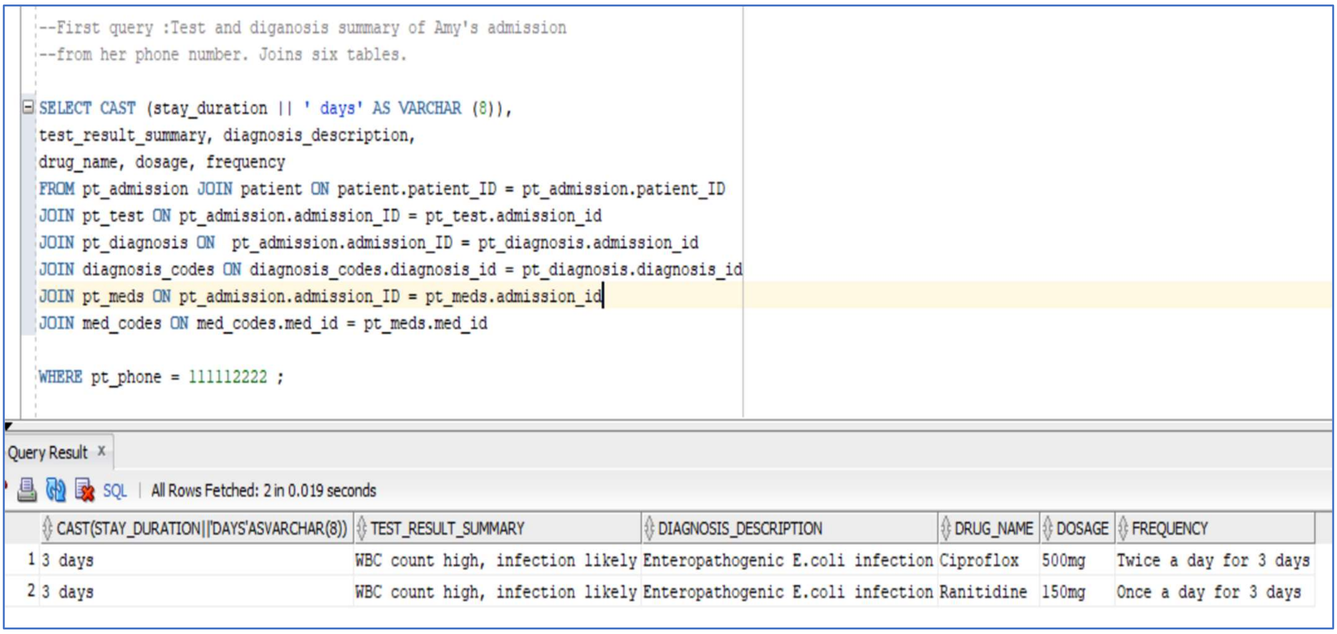
```
BEGIN
add_labtech ('Robin', 'Cook',3476122076 ,
TO_DATE ('01/07/1994' , 'mm/dd/yyyy'),
'Male', '1206 Spruce St, Newfield, IL', 'Medicine',
'LTA2723');
END;
/
commit;
select * from lab_tech JOIN HCW ON lab_tech.hcw_id= hcw.hcw_id;
```

All Rows Fetched: 3 in 0.003 seconds

	HCW_ID	LABTECH_LICENSE_CERT	HCW_ID_1	HCW_FIRST_NAME	HCW_LAST_NAME	HCW_PHONE	HCW_DOB	HCW_GENDER	HCW_ADDRESS	HC
1	13	LTA2723	13	Robin	Cook	3476122076	07-01-94	Male	1206 Spruce St, Newfield, IL	Med:
2	11	LTA1429	11	Priya	Sharma	5862733364	15-11-95	Female	127 Wheaton Dr, Newfield, IL	Hem
3	12	LTB23	12	Drake	Crew	5869745764	02-12-95	Male	61B Cyprus St, Newfield, IL	Onc

QUERY EXECUTIONS AND EXPLANATIONS

Query 1 - A patient’s test, diagnosis, and medication details from their Phone number

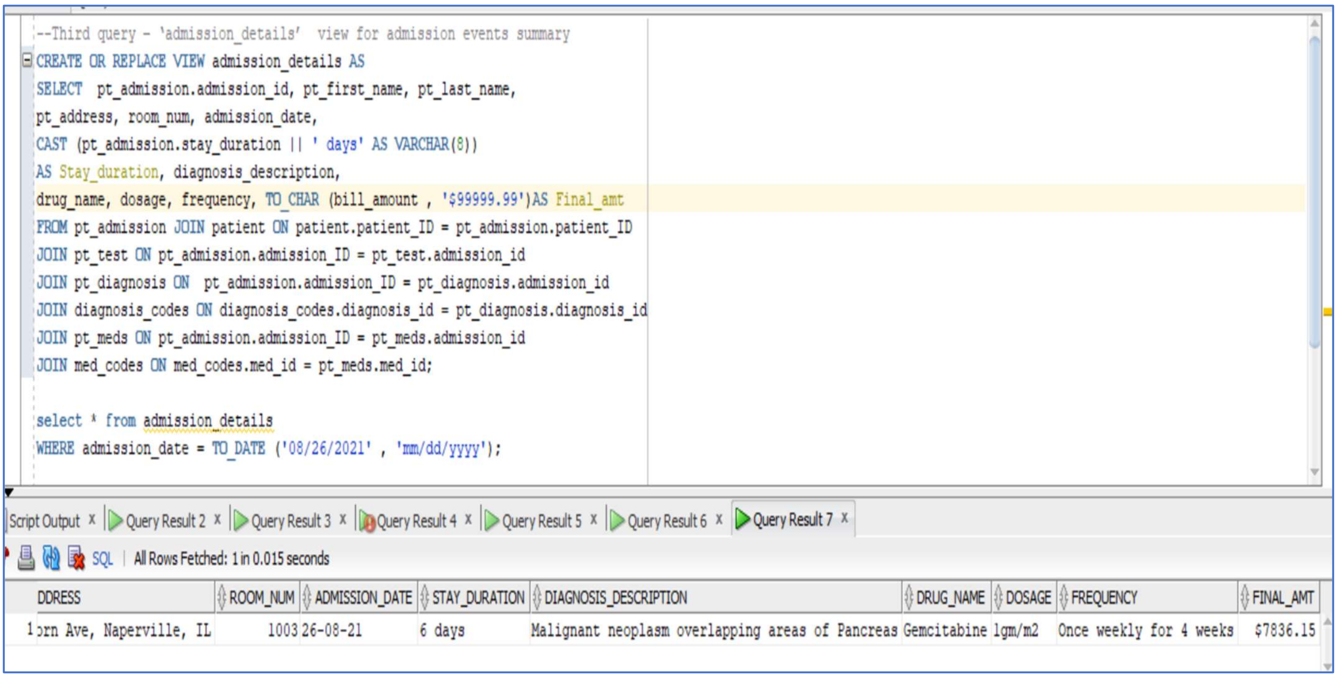


```
--First query :Test and diganosis summary of Amy's admission
--from her phone number. Joins six tables.

SELECT CAST (stay_duration || ' days' AS VARCHAR (8)),
test_result_summary, diagnosis_description,
drug_name, dosage, frequency
FROM pt_admission JOIN patient ON patient.patient_ID = pt_admission.patient_ID
JOIN pt_test ON pt_admission.admission_ID = pt_test.admission_id
JOIN pt_diagnosis ON  pt_admission.admission_ID = pt_diagnosis.admission_id
JOIN diagnosis_codes ON diagnosis_codes.diagnosis_id = pt_diagnosis.diagnosis_id
JOIN pt_meds ON pt_admission.admission_ID = pt_meds.admission_id
JOIN med_codes ON med_codes.med_id = pt_meds.med_id

WHERE pt_phone = 111112222 ;
```

All Rows Fetched: 2 in 0.019 seconds

	CAST(STAY_DURATION||'DAYS'ASVARCHAR(8))	TEST_RESULT_SUMMARY	DIAGNOSIS_DESCRIPTION	DRUG_NAME	DOSAGE	FREQUENCY
1	3 days	WBC count high, infection likely	Enteropathogenic E.coli infection	Ciproflox	500mg	Twice a day for 3 days
2	3 days	WBC count high, infection likely	Enteropathogenic E.coli infection	Ranitidine	150mg	Once a day for 3 days

Query 2 – Patient admission details using a view



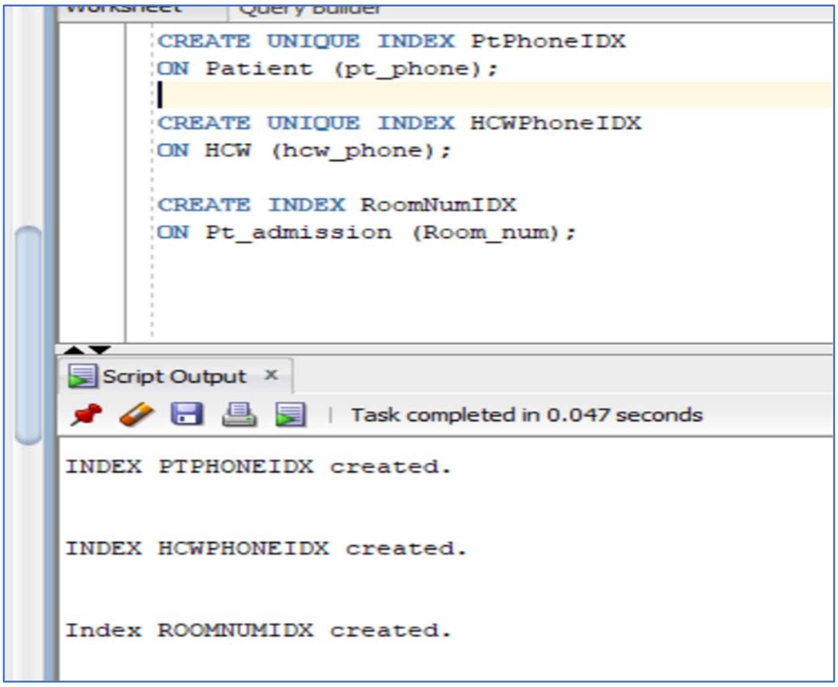
```
--Third query - 'admission_details'  view for admission events summary
CREATE OR REPLACE VIEW admission_details AS
SELECT  pt_admission.admission_id, pt_first_name, pt_last_name,
pt_address, room_num, admission_date,
CAST (pt_admission.stay_duration || ' days' AS VARCHAR(8))
AS Stay_duration, diagnosis_description,
drug_name, dosage, frequency, TO_CHAR (bill_amount , '$99999.99')AS Final_amt
FROM pt_admission JOIN patient ON patient.patient_ID = pt_admission.patient_ID
JOIN pt_test ON pt_admission.admission_ID = pt_test.admission_id
JOIN pt_diagnosis ON  pt_admission.admission_ID = pt_diagnosis.admission_id
JOIN diagnosis_codes ON diagnosis_codes.diagnosis_id = pt_diagnosis.diagnosis_id
JOIN pt_meds ON pt_admission.admission_ID = pt_meds.admission_id
JOIN med_codes ON med_codes.med_id = pt_meds.med_id;

select * from admission_details
WHERE admission_date = TO_DATE ('08/26/2021' , 'mm/dd/yyyy');
```

All Rows Fetched: 1 in 0.015 seconds

	DDRESS	ROOM_NUM	ADMISSION_DATE	STAY_DURATION	DIAGNOSIS_DESCRIPTION	DRUG_NAME	DOSAGE	FREQUENCY	FINAL_AMT
1	orn Ave, Naperville, IL	1003	26-08-21	6 days	Malignant neoplasm overlapping areas of Pancreas	Gemcitabine	1gm/m2	Once weekly for 4 weeks	$7836.15

INDEX IDENTIFICATION AND CREATIONS

Creating Index-es – Phone and Room number are 2 examples; they are unique and used often. Other indexes can be admission date and discharge date.



```
CREATE UNIQUE INDEX PtPhoneIDX
ON Patient (pt_phone);

CREATE UNIQUE INDEX HCWPhoneIDX
ON HCW (hcw_phone);

CREATE INDEX RoomNumIDX
ON Pt_admission (Room_num);
```

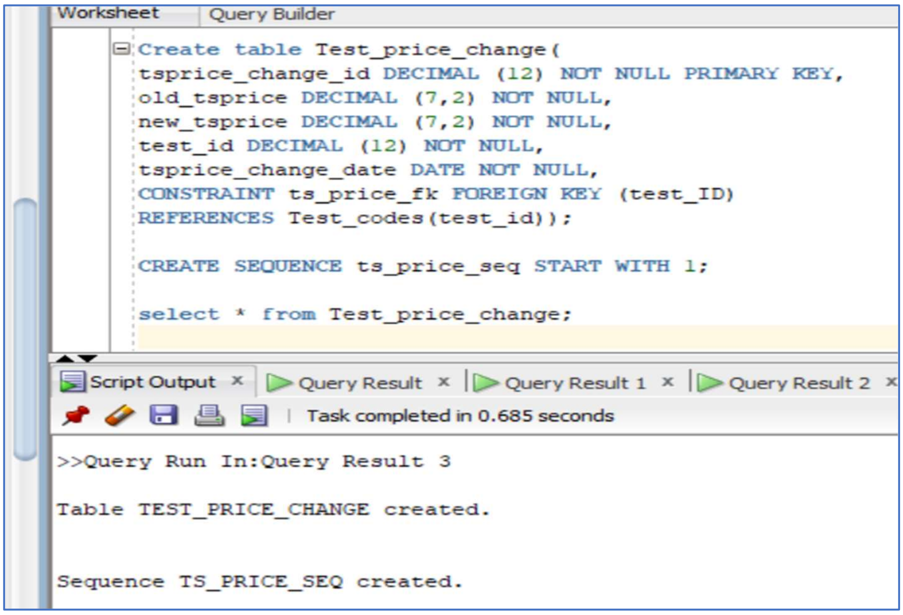
Task completed in 0.047 seconds

```
INDEX PTPHONEIDX created.


INDEX HCWPHONEIDX created.


Index ROOMNUMIDX created.
```

HISTORY TABLE AND TRIGGERS

Example shown – if price of a test is changed, then it is recorded in history table, and it will trigger when test_price is changed.

History table creation

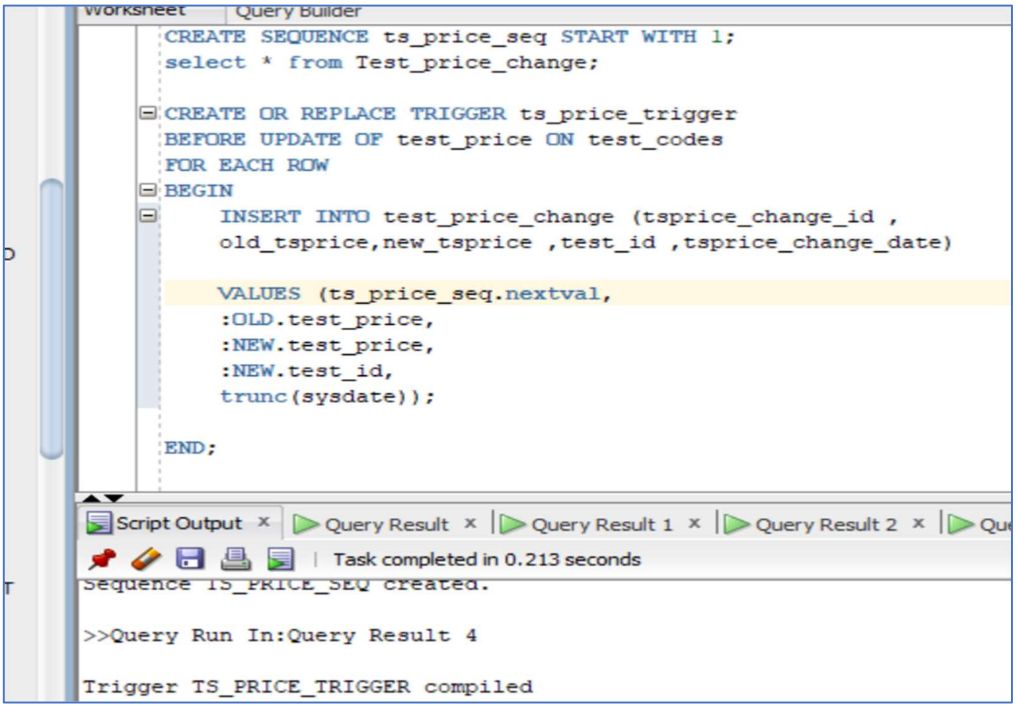


```
Create table Test_price_change(
tsprice_change_id DECIMAL (12) NOT NULL PRIMARY KEY,
old_tsprice DECIMAL (7,2) NOT NULL,
new_tsprice DECIMAL (7,2) NOT NULL,
test_id DECIMAL (12) NOT NULL,
tsprice_change_date DATE NOT NULL,
CONSTRAINT ts_price_fk FOREIGN KEY (test_ID)
REFERENCES Test_codes(test_id));

CREATE SEQUENCE ts_price_seq START WITH 1;

select * from Test_price_change;
```

Task completed in 0.685 seconds

```
>>Query Run In:Query Result 3

Table TEST_PRICE_CHANGE created.


Sequence TS_PRICE_SEQ created.
```

Trigger creation

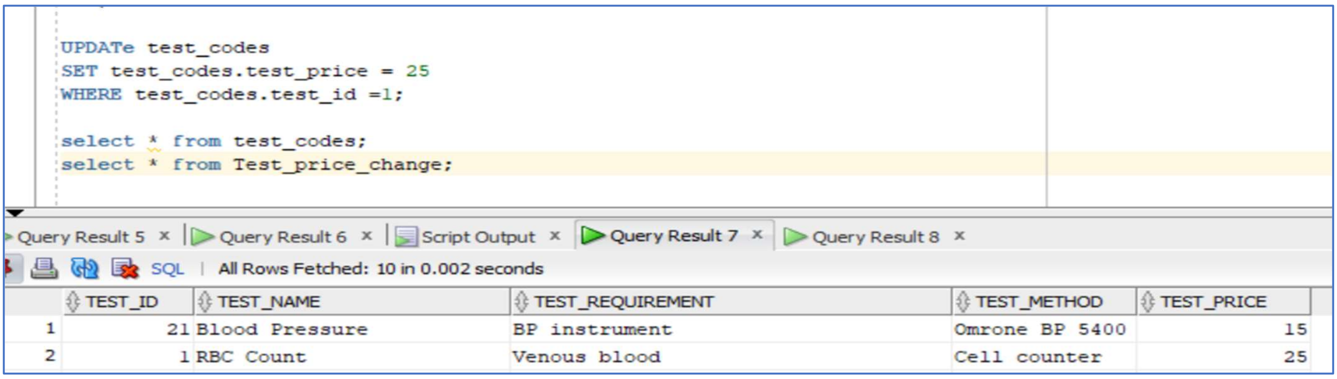


```
CREATE SEQUENCE ts_price_seq START WITH 1;
select * from Test_price_change;

CREATE OR REPLACE TRIGGER ts_price_trigger
BEFORE UPDATE OF test_price ON test_codes
FOR EACH ROW
BEGIN
    INSERT INTO test_price_change (tsprice_change_id ,
    old_tsprice,new_tsprice ,test_id ,tsprice_change_date)

    VALUES (ts_price_seq.nextval,
    :OLD.test_price,
    :NEW.test_price,
    :NEW.test_id,
    trunc(sysdate));

END;
```

Task completed in 0.213 seconds

```
Sequence TS_PRICE_SEQ created.

>>Query Run In:Query Result 4

Trigger TS_PRICE_TRIGGER compiled
```

Changing price of test_id = 1 from 20 to 25

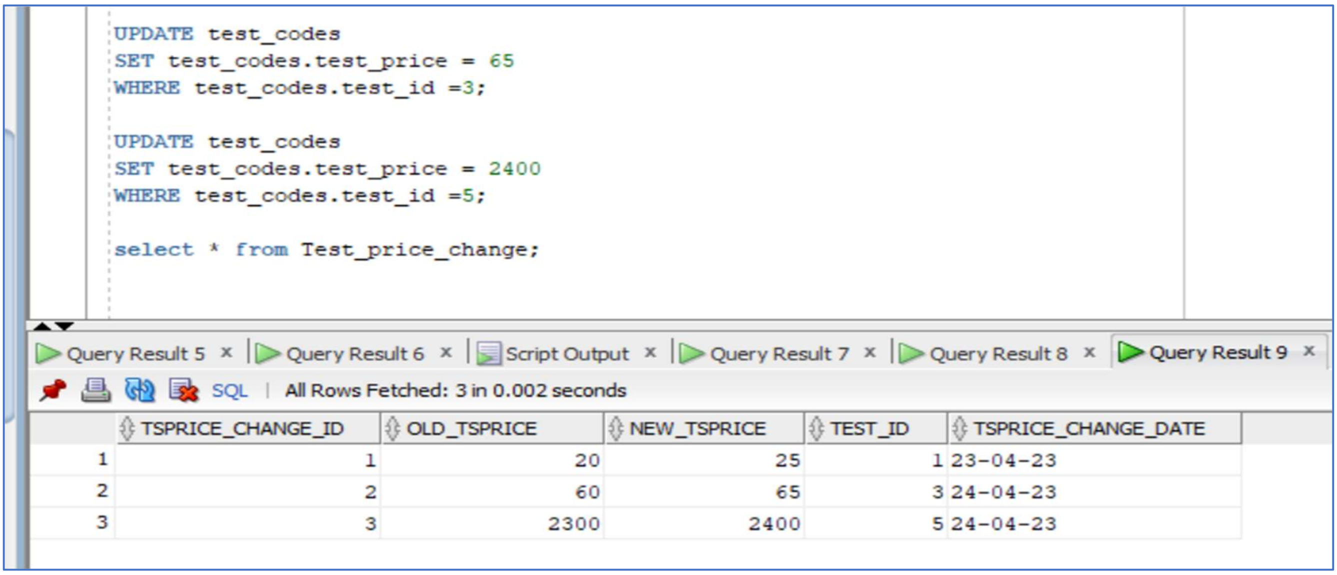


```
UPDATe test_codes
SET test_codes.test_price = 25
WHERE test_codes.test_id =1;

select * from test_codes;
select * from Test_price_change;
```

All Rows Fetched: 10 in 0.002 seconds

	TEST_ID	TEST_NAME	TEST_REQUIREMENT	TEST_METHOD	TEST_PRICE
1	21	Blood Pressure	BP instrument	Omrone BP 5400	15
2	1	RBC Count	Venous blood	Cell counter	25

3 changes in history table



```
UPDATE test_codes
SET test_codes.test_price = 65
WHERE test_codes.test_id =3;

UPDATE test_codes
SET test_codes.test_price = 2400
WHERE test_codes.test_id =5;

select * from Test_price_change;
```

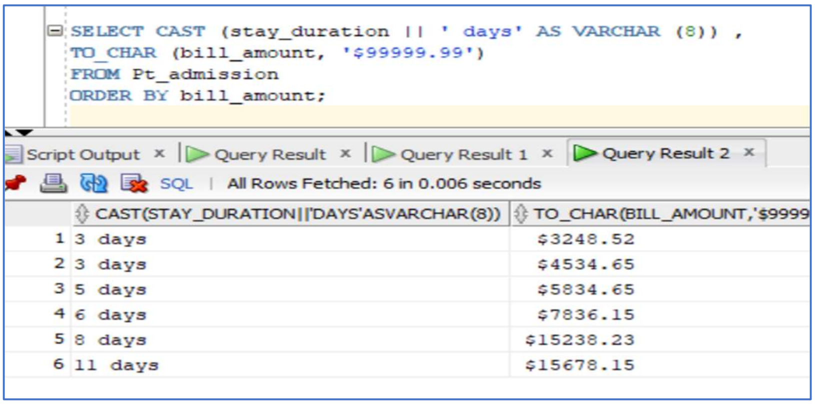
All Rows Fetched: 3 in 0.002 seconds

	TSPRICE_CHANGE_ID	OLD_TSPRICE	NEW_TSPRICE	TEST_ID	TSPRICE_CHANGE_DATE
1	1	20	25	1	23-04-23
2	2	60	65	3	24-04-23
3	3	2300	2400	5	24-04-23

DATA VISUALIZATIONS

Using our database to write queries and gain useful insights from hospital inpatient data.

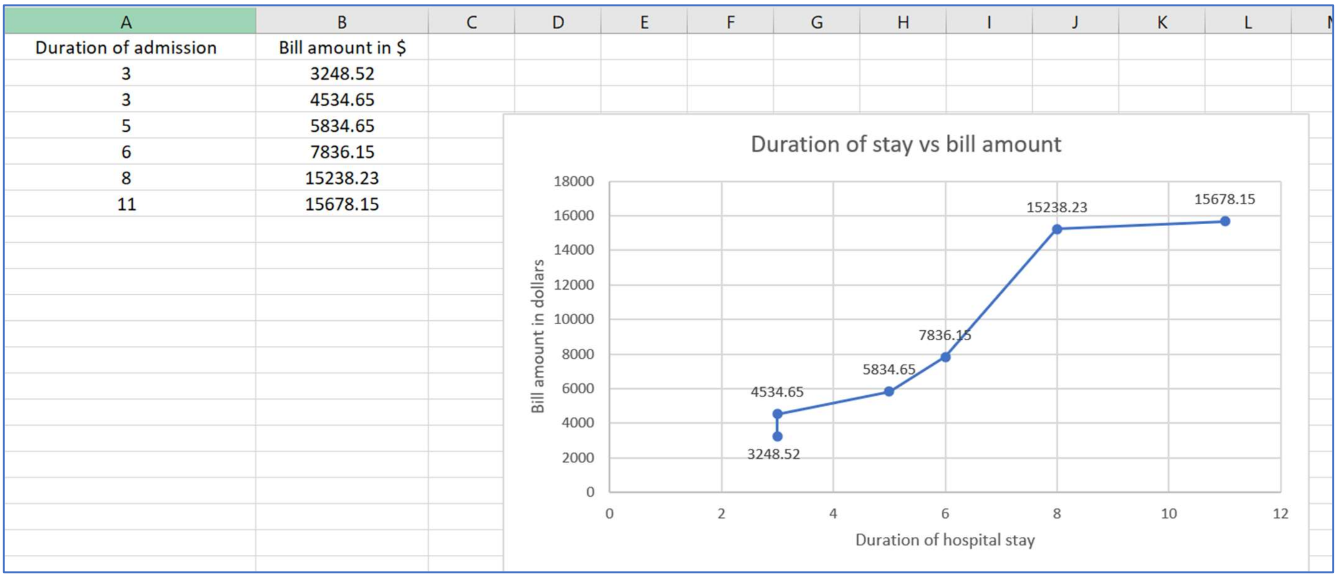
1. The bill amount and duration of hospital stay



```
SELECT CAST (stay_duration || ' days' AS VARCHAR (8)) ,
TO_CHAR (bill_amount, '$99999.99')
FROM Pt_admission
ORDER BY bill_amount;
```

All Rows Fetched: 6 in 0.006 seconds

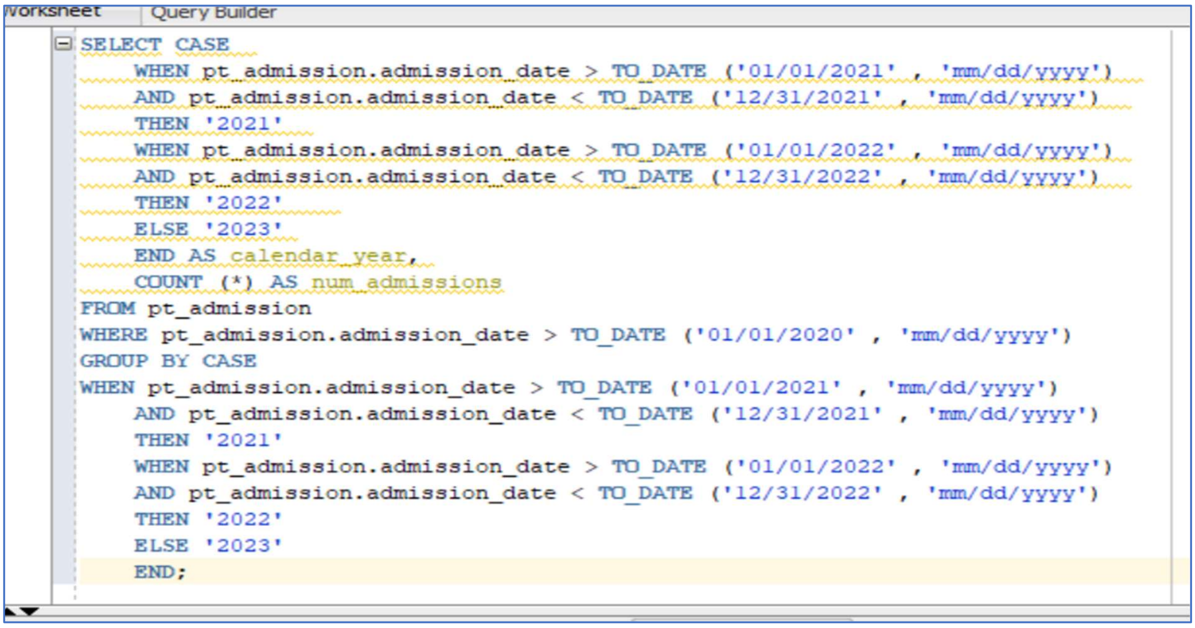
```
    CAST(STAY_DURATION||'DAYS'ASVARCHAR(8))   TO_CHAR(BILL_AMOUNT,'$9999
  1 3 days                                     $3248.52
  2 3 days                                     $4534.65
  3 5 days                                     $5834.65
  4 6 days                                     $7836.15
  5 8 days                                    $15238.23
  6 11 days                                   $15678.15
```



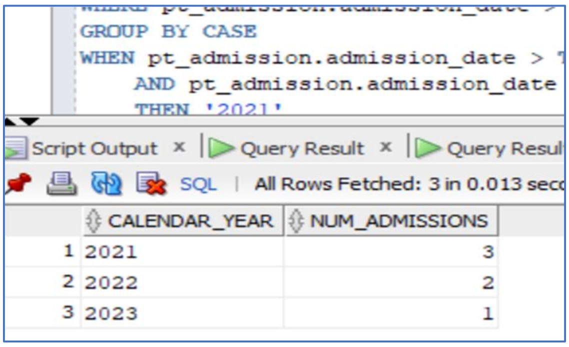
A	B
Duration of admission	Bill amount in $
3	3248.52
3	4534.65
5	5834.65
6	7836.15
8	15238.23
11	15678.15

We can see that duration of stay more than 8 days does not mean significant in crease in cost, and most admissions and procedures take less than a week.

2. Number of admissions by year for 2021, 2022, 2023



```
SELECT CASE
    WHEN pt_admission.admission_date > TO_DATE ('01/01/2021' , 'mm/dd/yyyy')
    AND pt_admission.admission_date < TO_DATE ('12/31/2021' , 'mm/dd/yyyy')
    THEN '2021'
    WHEN pt_admission.admission_date > TO_DATE ('01/01/2022' , 'mm/dd/yyyy')
    AND pt_admission.admission_date < TO_DATE ('12/31/2022' , 'mm/dd/yyyy')
    THEN '2022'
    ELSE '2023'
    END AS calendar_year,
    COUNT (*) AS num_admissions
FROM pt_admission
WHERE pt_admission.admission_date > TO_DATE ('01/01/2020' , 'mm/dd/yyyy')
GROUP BY CASE
WHEN pt_admission.admission_date > TO_DATE ('01/01/2021' , 'mm/dd/yyyy')
    AND pt_admission.admission_date < TO_DATE ('12/31/2021' , 'mm/dd/yyyy')
    THEN '2021'
    WHEN pt_admission.admission_date > TO_DATE ('01/01/2022' , 'mm/dd/yyyy')
    AND pt_admission.admission_date < TO_DATE ('12/31/2022' , 'mm/dd/yyyy')
    THEN '2022'
    ELSE '2023'
    END;
```

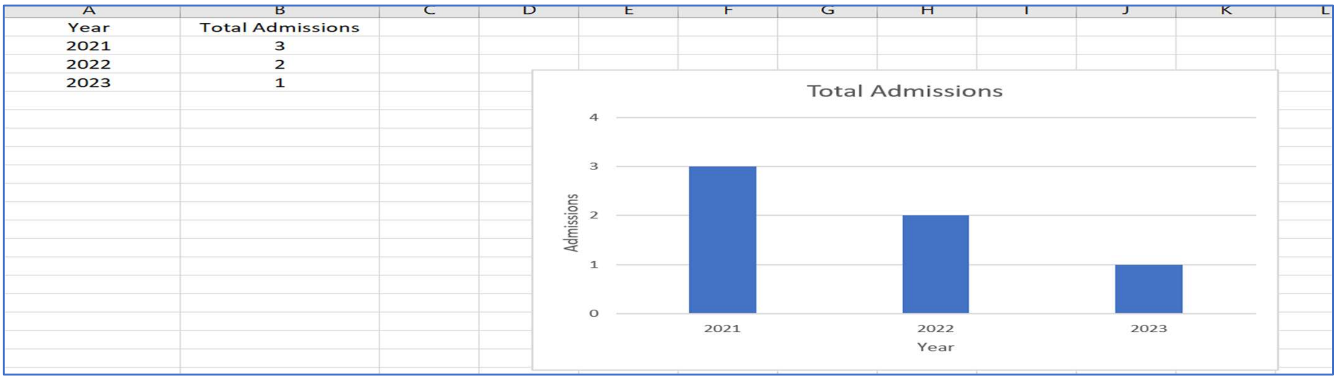


```
GROUP BY CASE
WHEN pt_admission.admission_date > T
    AND pt_admission.admission_date
    THEN '2021'
```

All Rows Fetched: 3 in 0.013 secc

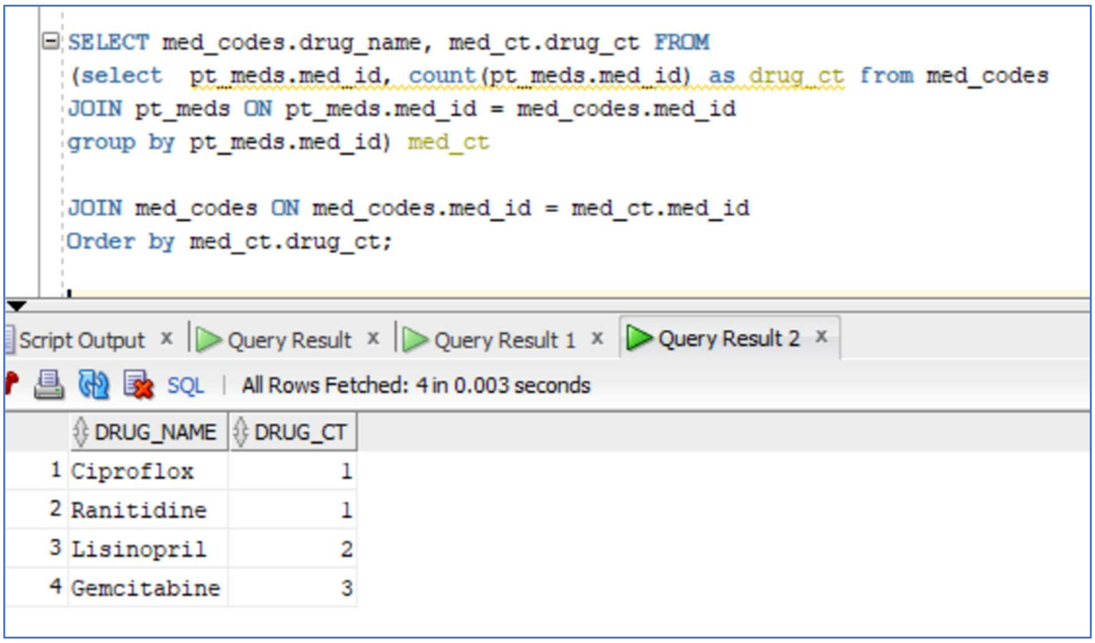
	CALENDAR_YEAR	NUM_ADMISSIONS
1	2021	3
2	2022	2
3	2023	1

Graph



Year	Total Admissions
2021	3
2022	2
2023	1

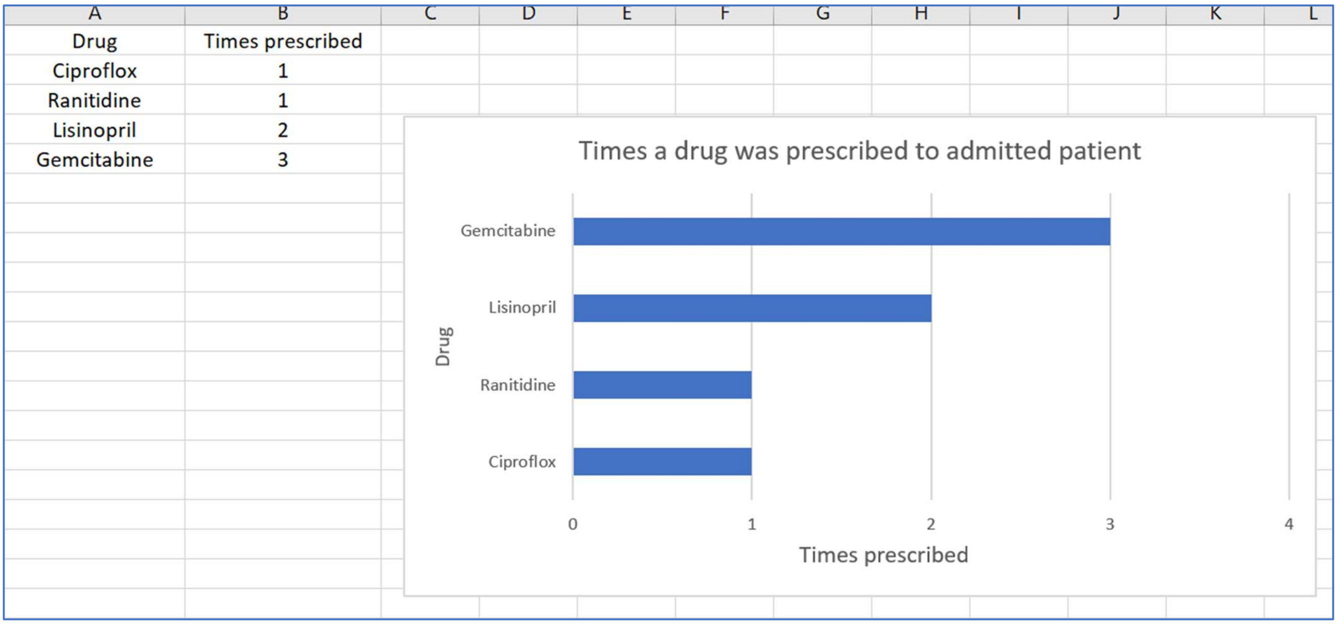
3. Number of times a drug is prescribed to admitted patients



```
SELECT med_codes.drug_name, med_ct.drug_ct FROM
(select  pt_meds.med_id, count(pt_meds.med_id) as drug_ct from med_codes
JOIN pt_meds ON pt_meds.med_id = med_codes.med_id
group by pt_meds.med_id) med_ct

JOIN med_codes ON med_codes.med_id = med_ct.med_id
Order by med_ct.drug_ct;
```

All Rows Fetched: 4 in 0.003 seconds

	DRUG_NAME	DRUG_CT
1	Ciproflox	1
2	Ranitidine	1
3	Lisinopril	2
4	Gemcitabine	3



Drug	Times prescribed
Ciproflox	1
Ranitidine	1
Lisinopril	2
Gemcitabine	3

SUMMARY

In this project I created an inpatient database, decided the attributes, tables, and relationships between entities, followed by populating tables, creating procedures, index-es, triggers, and history table, writing queries, and creating simple visualizations from this data.