Team 5

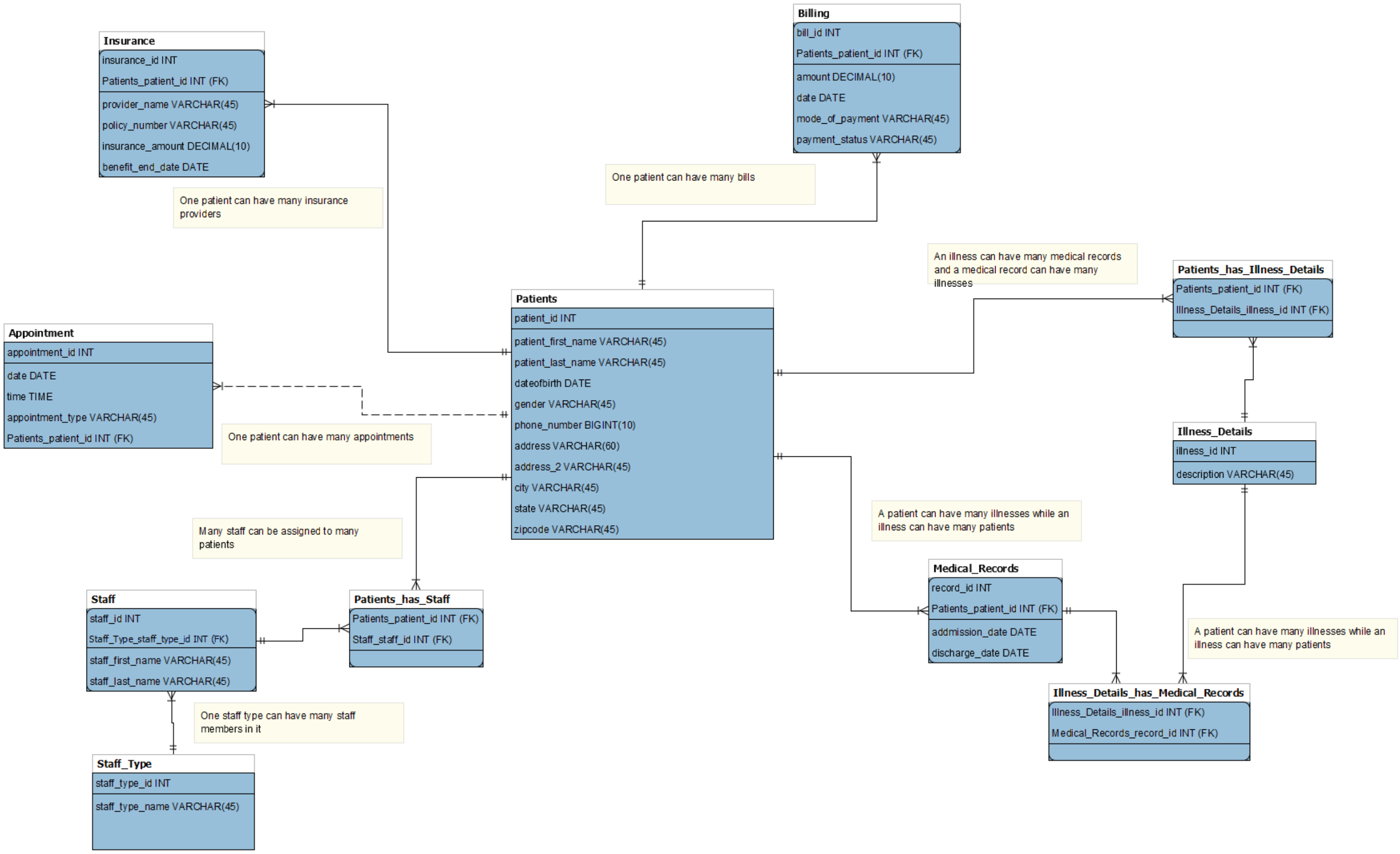
Hospital Management System

Team Members :-

- Abhishek Namdev Sawant
- Garrett Ringler
- Erdenetuya Namsrai
- Vivek Reddy Karra
- Sai Kishore Chintala

Milestone 2 Model:

Following simplification of the database our new model narrowed down to 11 tables with some corrections to relationships as well.



Insurance
insurance_id INT
Patients_patient_id INT (FK)
provider_name VARCHAR(45)
policy_number VARCHAR(45)
insurance_amount DECIMAL(10)
benefit_end_date DATE
One patient can have many insurance providers
Billing
bill_id INT
Patients_patient_id INT (FK)
amount DECIMAL(10)
date DATE
mode_of_payment VARCHAR(45)
payment_status VARCHAR(45)
One patient can have many bills
An illness can have many medical records and a medical record can have many illnesses
Patients_has_Illness_Details
Patients_patient_id INT (FK)
Illness_Details_illness_id INT (FK)
Appointment
appointment_id INT
date DATE
time TIME
appointment_type VARCHAR(45)
Patients_patient_id INT (FK)
One patient can have many appointments
Patients
patient_id INT
patient_first_name VARCHAR(45)
patient_last_name VARCHAR(45)
dateofbirth DATE
gender VARCHAR(45)
phone_number BIGINT(10)
address VARCHAR(60)
address_2 VARCHAR(45)
city VARCHAR(45)
state VARCHAR(45)
zipcode VARCHAR(45)
Illness_Details
illness_id INT
description VARCHAR(45)
Many staff can be assigned to many patients
A patient can have many illnesses while an illness can have many patients
Staff
staff_id INT
Staff_Type_staff_type_id INT (FK)
staff_first_name VARCHAR(45)
staff_last_name VARCHAR(45)
Patients_has_Staff
Patients_patient_id INT (FK)
Staff_staff_id INT (FK)
Medical_Records
record_id INT
Patients_patient_id INT (FK)
addmission_date DATE
discharge_date DATE
A patient can have many illnesses while an illness can have many patients
One staff type can have many staff members in it
Illness_Details_has_Medical_Records
Illness_Details_illness_id INT (FK)
Medical_Records_record_id INT (FK)
Staff_Type
staff_type_id INT
staff_type_name VARCHAR(45)

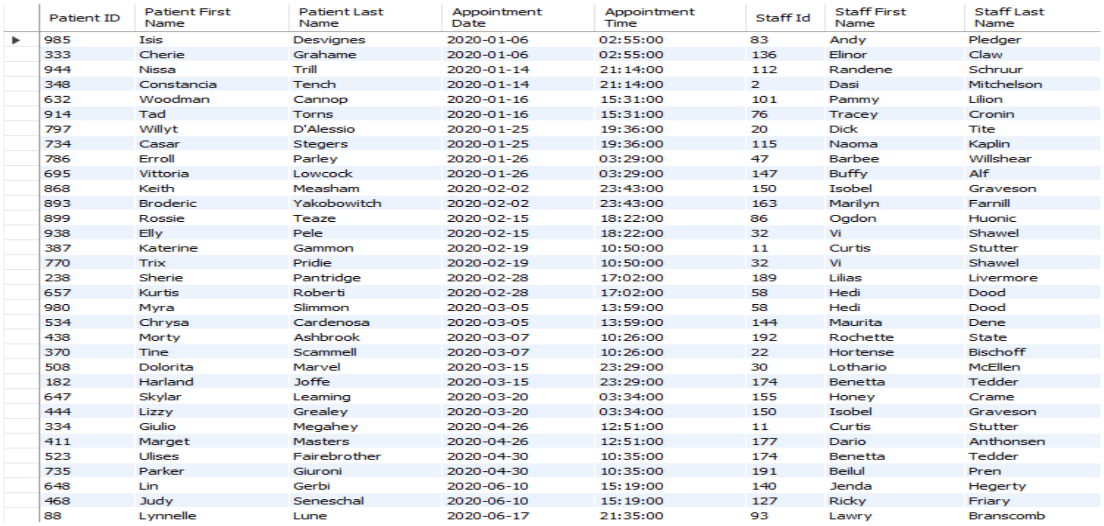
Milestone 2 SQL:

1. List the details of patients who had appointments scheduled with staff between the dates 2020-01-01 and 2020-06-30.

Query : -

```
SELECT

   p.patient_id AS 'Patient ID',

   p.patient_first_name AS 'Patient First Name',

   p.patient_last_name AS 'Patient Last Name',

   a.date AS 'Appointment Date',

   a.time AS 'Appointment Time',

   s.staff_id AS 'Staff Id',

   s.staff_first_name AS 'Staff First Name',

   s.staff_last_name AS 'Staff Last Name'

FROM Patients AS p

JOIN Appointment AS a ON p.patient_id = a.Patients_patient_id

JOIN Patients_has_Staff AS ps ON p.patient_id = ps.Patients_patient_id

JOIN Staff AS s ON s.staff_id = ps.Staff_staff_id

WHERE a.date BETWEEN '2020-01-01' AND '2020-06-30'

ORDER BY a.date;
```

Output :-



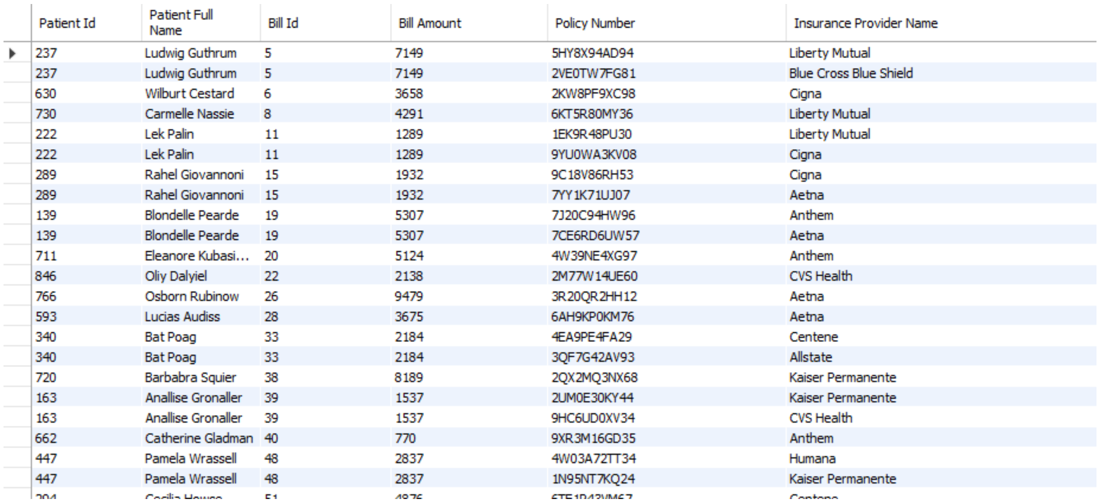
	Patient ID	Patient First Name	Patient Last Name	Appointment Date	Appointment Time	Staff Id	Staff First Name	Staff Last Name
▶	985	Isis	Desvignes	2020-01-06	02:55:00	83	Andy	Pledger
	333	Cherie	Grahame	2020-01-06	02:55:00	136	Elinor	Claw
	944	Nissa	Trill	2020-01-14	21:14:00	112	Randene	Schruur
	348	Constancia	Tench	2020-01-14	21:14:00	2	Dasi	Mitchelson
	632	Woodman	Cannop	2020-01-16	15:31:00	101	Pammy	Lilion
	914	Tad	Torns	2020-01-16	15:31:00	76	Tracey	Cronin
	797	Willyt	D'Alessio	2020-01-25	19:36:00	20	Dick	Tite
	734	Casar	Stegers	2020-01-25	19:36:00	115	Naoma	Kaplin
	786	Erroll	Parley	2020-01-26	03:29:00	47	Barbee	Willshear
	695	Vittoria	Lowcock	2020-01-26	03:29:00	147	Buffy	Alf
	868	Keith	Measham	2020-02-02	23:43:00	150	Isobel	Graveson
	893	Broderic	Yakobowitch	2020-02-02	23:43:00	163	Marilyn	Farnill
	899	Rossie	Teaze	2020-02-15	18:22:00	86	Ogdon	Huonic
	938	Elly	Pele	2020-02-15	18:22:00	32	Vi	Shawel
	387	Katerine	Gammon	2020-02-19	10:50:00	11	Curtis	Stutter
	770	Trix	Pridie	2020-02-19	10:50:00	32	Vi	Shawel
	238	Sherie	Pantridge	2020-02-28	17:02:00	189	Lilias	Livermore
	657	Kurtis	Roberti	2020-02-28	17:02:00	58	Hedi	Dood
	980	Myra	Slimmon	2020-03-05	13:59:00	58	Hedi	Dood
	534	Chrysa	Cardenosa	2020-03-05	13:59:00	144	Maurita	Dene
	438	Morty	Ashbrook	2020-03-07	10:26:00	192	Rochette	State
	370	Tine	Scammell	2020-03-07	10:26:00	22	Hortense	Bischoff
	508	Dolorita	Marvel	2020-03-15	23:29:00	30	Lothario	McEllen
	182	Harland	Joffe	2020-03-15	23:29:00	174	Benetta	Tedder
	647	Skylar	Leaming	2020-03-20	03:34:00	155	Honey	Crame
	444	Lizzy	Grealey	2020-03-20	03:34:00	150	Isobel	Graveson
	334	Giulio	Megahey	2020-04-26	12:51:00	11	Curtis	Stutter
	411	Marget	Masters	2020-04-26	12:51:00	177	Dario	Anthonsen
	523	Ulises	Fairebrother	2020-04-30	10:35:00	174	Benetta	Tedder
	735	Parker	Giuroni	2020-04-30	10:35:00	191	Beilul	Pren
	648	Lin	Gerbi	2020-06-10	15:19:00	140	Jenda	Hegerty
	468	Judy	Seneschal	2020-06-10	15:19:00	127	Ricky	Friary
	88	Lynnelle	Lune	2020-06-17	21:35:00	93	Lawry	Branscomb

2. This query will list details of patients, billing amounts, and insurance details of the patient for bills covered by insurance as a payment method.

<u>Query :-</u>

```
SELECT
   p.patient_id AS 'Patient Id',
   CONCAT(p.patient_first_name, ' ', p.patient_last_name) AS 'Patient Full Name',
   b.bill_id AS 'Bill Id',
   b.amount AS 'Bill Amount',
   i.policy_number AS 'Policy Number',
   i.provider_name AS 'Insurance Provider Name'
FROM Patients AS p
JOIN Billing AS b ON p.patient_id = b.Patients_patient_id
JOIN Insurance AS i ON p.patient_id = i.Patients_patient_id
WHERE b.mode_of_payment LIKE 'insurance';
```

<u>Output:-</u>



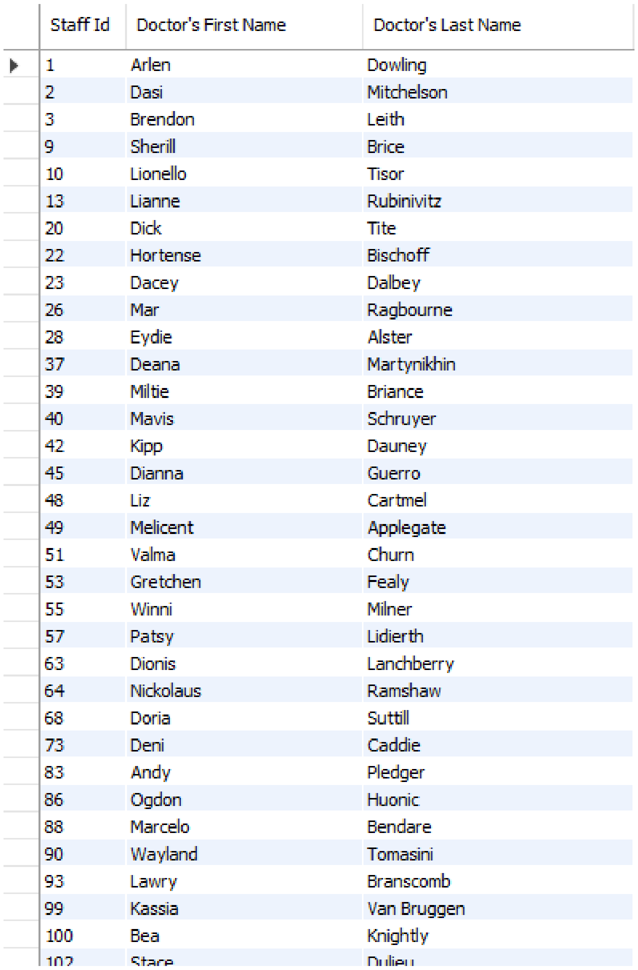
Patient Id	Patient Full Name	Bill Id	Bill Amount	Policy Number	Insurance Provider Name
237	Ludwig Guthrum	5	7149	5HY8X94AD94	Liberty Mutual
237	Ludwig Guthrum	5	7149	2VE0TW7FG81	Blue Cross Blue Shield
630	Wilburt Cestard	6	3658	2KW8PF9XC98	Cigna
730	Carmelle Nassie	8	4291	6KT5R80MY36	Liberty Mutual
222	Lek Palin	11	1289	1EK9R48PU30	Liberty Mutual
222	Lek Palin	11	1289	9YU0WA3KV08	Cigna
289	Rahel Giovannoni	15	1932	9C18V86RH53	Cigna
289	Rahel Giovannoni	15	1932	7YY1K71UJ07	Aetna
139	Blondelle Pearde	19	5307	7J20C94HW96	Anthem
139	Blondelle Pearde	19	5307	7CE6RD6UW57	Aetna
711	Eleanore Kubasi...	20	5124	4W39NE4XG97	Anthem
846	Oliy Dalyiel	22	2138	2M77W14UE60	CVS Health
766	Osborn Rubinow	26	9479	3R20QR2HH12	Aetna
593	Lucias Audiss	28	3675	6AH9KP0KM76	Aetna
340	Bat Poag	33	2184	4EA9PE4FA29	Centene
340	Bat Poag	33	2184	3QF7G42AV93	Allstate
720	Barbabra Squier	38	8189	2QX2MQ3NX68	Kaiser Permanente
163	Anallise Gronaller	39	1537	2UM0E30KY44	Kaiser Permanente
163	Anallise Gronaller	39	1537	9HC6UD0XV34	CVS Health
662	Catherine Gladman	40	770	9XR3M16GD35	Anthem
447	Pamela Wrassell	48	2837	4W03A72TT34	Humana
447	Pamela Wrassell	48	2837	1N95NT7KQ24	Kaiser Permanente
204	Cecilia Howse	51	4876	6TE1P43VM67	Centene

3. This query will get staff details for all the doctor staff.

Query :-

```
SELECT

  s.staff_id AS 'Staff Id',

  s.staff_first_name AS 'Doctor\'s First Name',

  s.staff_last_name AS 'Doctor\'s Last Name'

FROM Staff AS s

JOIN Staff_Type AS st ON s.staff_id = st.Staff_staff_id

WHERE st.staff_type_name LIKE 'doctor';
```

Output :-



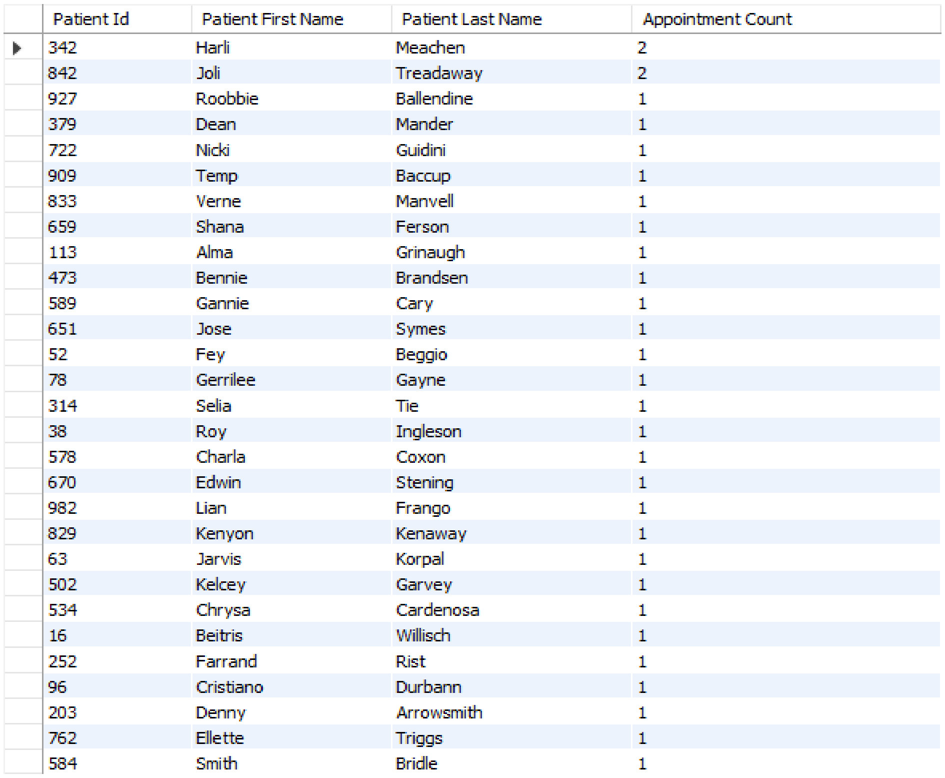
Staff Id	Doctor's First Name	Doctor's Last Name
1	Arlen	Dowling
2	Dasi	Mitchelson
3	Brendon	Leith
9	Sherill	Brice
10	Lionello	Tisor
13	Lianne	Rubinivitz
20	Dick	Tite
22	Hortense	Bischoff
23	Dacey	Dalbey
26	Mar	Ragbourne
28	Eydie	Alster
37	Deana	Martynikhin
39	Miltie	Briance
40	Mavis	Schruyer
42	Kipp	Dauney
45	Dianna	Guerro
48	Liz	Cartmel
49	Melicent	Applegate
51	Valma	Churn
53	Gretchen	Fealy
55	Winni	Milner
57	Patsy	Lidierth
63	Dionis	Lanchberry
64	Nickolaus	Ramshaw
68	Doria	Suttill
73	Deni	Caddie
83	Andy	Pledger
86	Ogdon	Huonic
88	Marcelo	Bendare
90	Wayland	Tomasini
93	Lawry	Branscomb
99	Kassia	Van Bruggen
100	Bea	Knightly
102	Stace	Dulieu

4. This query lists the details of patients and the number of appointments that they made in the year 2018.

<u>Query :-</u>

```
SELECT
  p.patient_id AS 'Patient Id',
  p.patient_first_name AS 'Patient First Name',
  p.patient_last_name AS 'Patient Last Name',
  COUNT(a.date) AS 'Appointment Count'
FROM Patients AS p
JOIN Appointment AS a ON p.patient_id = a.patients_patient_id
WHERE a.date BETWEEN '2018-01-01' AND '2018-12-31'
GROUP BY p.patient_id
ORDER BY COUNT(a.date) DESC;
```

<u>Output :-</u>



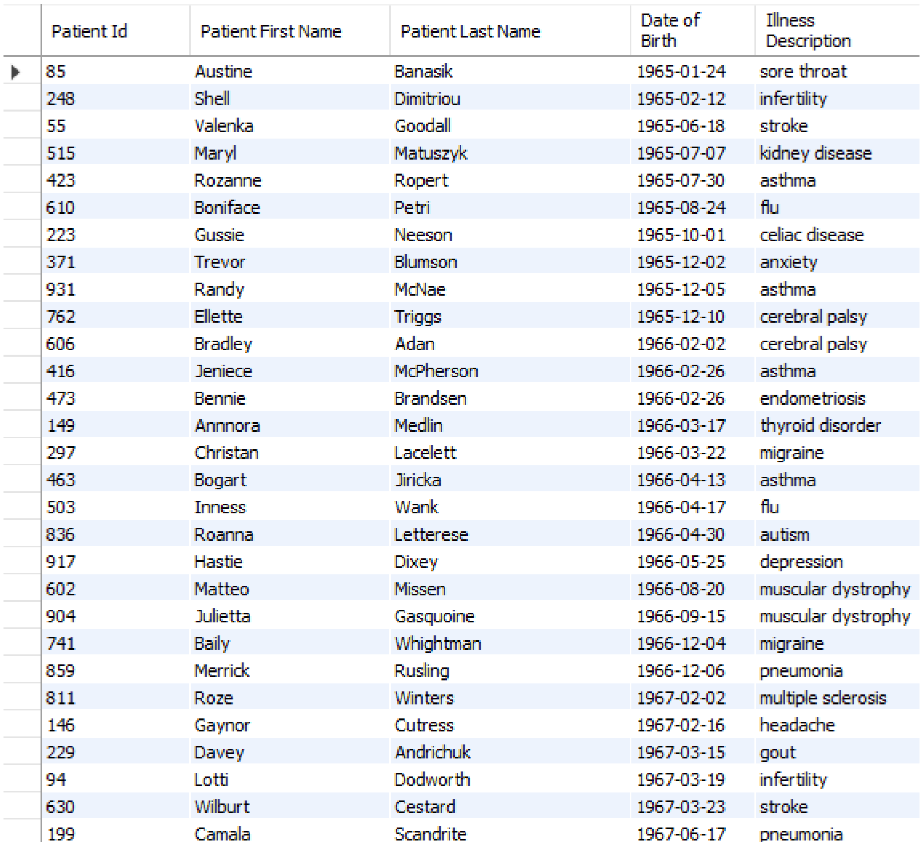
Patient Id	Patient First Name	Patient Last Name	Appointment Count
342	Harli	Meachen	2
842	Joli	Treadaway	2
927	Roobbie	Ballendine	1
379	Dean	Mander	1
722	Nicki	Guidini	1
909	Temp	Baccup	1
833	Verne	Manvell	1
659	Shana	Ferson	1
113	Alma	Grinaugh	1
473	Bennie	Brandsen	1
589	Gannie	Cary	1
651	Jose	Symes	1
52	Fey	Beggio	1
78	Gerrilee	Gayne	1
314	Selia	Tie	1
38	Roy	Ingleson	1
578	Charla	Coxon	1
670	Edwin	Stening	1
982	Lian	Frango	1
829	Kenyon	Kenaway	1
63	Jarvis	Korpal	1
502	Kelcey	Garvey	1
534	Chrysa	Cardenosa	1
16	Beitris	Willisch	1
252	Farrand	Rist	1
96	Cristiano	Durbann	1
203	Denny	Arrowsmith	1
762	Ellette	Triggs	1
584	Smith	Bridle	1

5. This query gives the details of patients with their illness who are born between 1965-01-01 and 1975-12-31.

Query :-

```
SELECT
   p.patient_id AS 'Patient Id',
   p.patient_first_name AS 'Patient First Name',
   p.patient_last_name AS 'Patient Last Name',
   p.dateofbirth AS 'Date of Birth',
   ill.description AS 'Illness Description'
FROM Patients AS p
JOIN Patients_has_Illness_Details AS pd ON p.patient_id = pd.Patients_patient_id
JOIN Illness_Details AS ill ON ill.illness_id = pd.Illness_details_illness_id
WHERE dateofbirth BETWEEN '1965-01-01' AND '1975-12-31'
ORDER BY dateofbirth;
```

Output :-



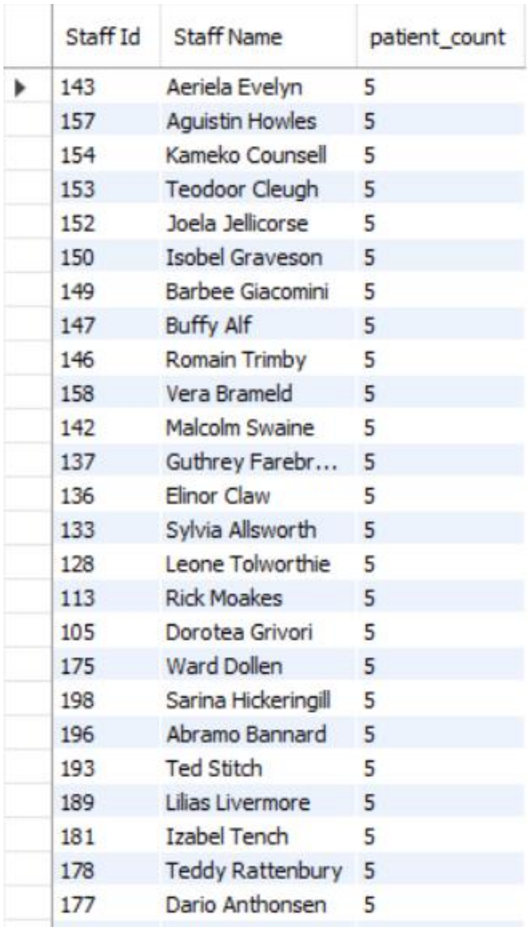
Patient Id	Patient First Name	Patient Last Name	Date of Birth	Illness Description
85	Austine	Banasik	1965-01-24	sore throat
248	Shell	Dimitriou	1965-02-12	infertility
55	Valenka	Goodall	1965-06-18	stroke
515	Maryl	Matuszyk	1965-07-07	kidney disease
423	Rozanne	Ropert	1965-07-30	asthma
610	Boniface	Petri	1965-08-24	flu
223	Gussie	Neeson	1965-10-01	celiac disease
371	Trevor	Blumson	1965-12-02	anxiety
931	Randy	McNae	1965-12-05	asthma
762	Ellette	Triggs	1965-12-10	cerebral palsy
606	Bradley	Adan	1966-02-02	cerebral palsy
416	Jeniece	McPherson	1966-02-26	asthma
473	Bennie	Brandsen	1966-02-26	endometriosis
149	Annnora	Medlin	1966-03-17	thyroid disorder
297	Christan	Lacelett	1966-03-22	migraine
463	Bogart	Jiricka	1966-04-13	asthma
503	Inness	Wank	1966-04-17	flu
836	Roanna	Letterese	1966-04-30	autism
917	Hastie	Dixey	1966-05-25	depression
602	Matteo	Missen	1966-08-20	muscular dystrophy
904	Julietta	Gasquoine	1966-09-15	muscular dystrophy
741	Baily	Whightman	1966-12-04	migraine
859	Merrick	Rusling	1966-12-06	pneumonia
811	Roze	Winters	1967-02-02	multiple sclerosis
146	Gaynor	Cutress	1967-02-16	headache
229	Davey	Andrichuk	1967-03-15	gout
94	Lotti	Dodworth	1967-03-19	infertility
630	Wilburt	Cestard	1967-03-23	stroke
199	Camala	Scandrite	1967-06-17	pneumonia

6. This query gives the list of doctors by the number of patients they have served, in descending order.

Query :-

```
SELECT
  st.Staff_staff_id AS 'Staff Id',
  CONCAT(s.staff_first_name, ' ', s.staff_last_name) AS 'Staff Name',
  COUNT(DISTINCT phs.Patients_patient_id) AS Patient_count
FROM Staff_Type st
JOIN Patients_has_Staff phs ON st.Staff_staff_id = phs.Staff_staff_id
JOIN Staff s ON s.staff_id = st.Staff_staff_id
WHERE staff_type_name = 'Doctor'
GROUP BY st.Staff_staff_id , st.staff_type_name
ORDER BY Patient_count DESC;
```

Output :-



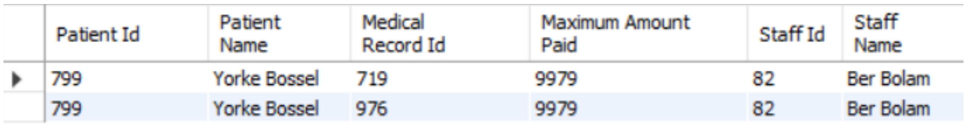
Staff Id	Staff Name	patient_count
143	Aeriela Evelyn	5
157	Aguistin Howles	5
154	Kameko Counsell	5
153	Teodoor Cleugh	5
152	Joela Jellicorse	5
150	Isobel Graveson	5
149	Barbee Giacomini	5
147	Buffy Alf	5
146	Romain Trimby	5
158	Vera Brameld	5
142	Malcolm Swaine	5
137	Guthrey Farebr...	5
136	Elinor Claw	5
133	Sylvia Allsworth	5
128	Leone Tolworthie	5
113	Rick Moakes	5
105	Dorotea Grivori	5
175	Ward Dollen	5
198	Sarina Hickeringill	5
196	Abramo Bannard	5
193	Ted Stitch	5
189	Lilias Livermore	5
181	Izabel Tench	5
178	Teddy Rattenbury	5
177	Dario Anthonsen	5

7. This query Identifies the patient with the highest bill payment and lists the services they received, as well as the names of the staff's who treated them.

Query :-

```
SELECT
  p.patient_id AS 'Patient Id',
  CONCAT(p.patient_first_name, ' ', p.patient_last_name) AS 'Patient Name',
  mr.record_id AS 'Medical Record Id',
  b.max_paid AS 'Maximum Amount Paid',
  phs.Staff_staff_id AS 'Staff Id',
CONCAT(s.staff_first_name, ' ', s.staff_last_name) AS 'Staff Name'
FROM
  Patients AS p
JOIN
  (SELECT
    Patients_patient_id, MAX(amount) AS max_paid
  FROM
    Billing AS b
  GROUP BY Patients_patient_id
  ORDER BY max_paid DESC
  LIMIT 1) AS b ON p.patient_id = b.Patients_patient_id
JOIN Medical_Records AS mr ON p.patient_id = mr.Patients_patient_id
JOIN Patients_has_Staff AS phs ON p.patient_id = phs.Patients_patient_id
JOIN Staff AS s ON phs.Staff_staff_id = s.staff_id
JOIN Staff_Type AS st ON s.staff_id = st.Staff_staff_id;
```

Output :-



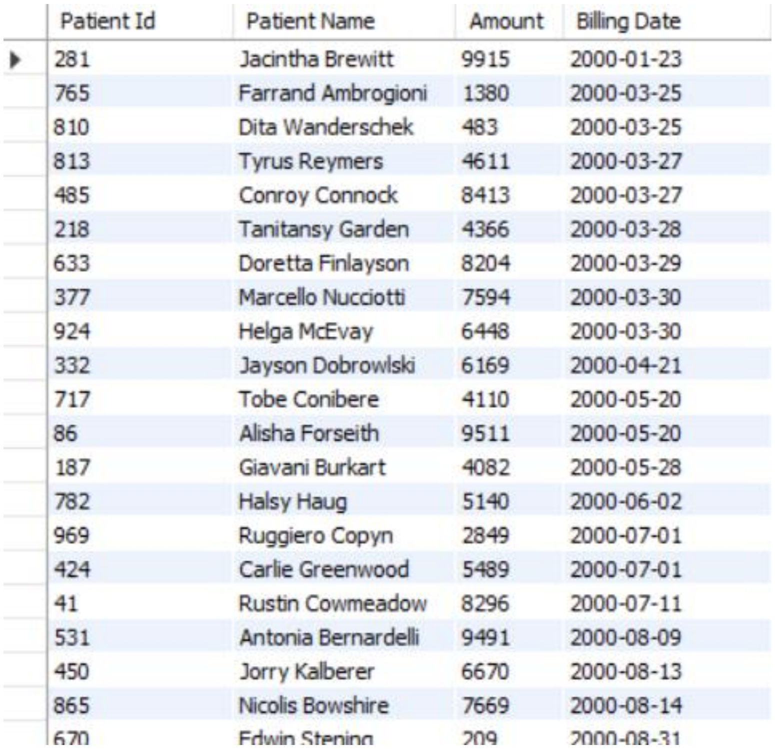
	Patient Id	Patient Name	Medical Record Id	Maximum Amount Paid	Staff Id	Staff Name
▶	799	Yorke Bossel	719	9979	82	Ber Bolam
	799	Yorke Bossel	976	9979	82	Ber Bolam

8. This query finds all the billing records for patients with payment status as Unpaid.

Query :-

```
SELECT
    p.patient_id AS 'Patient Id',
     CONCAT(p.patient_first_name, ' ', p.patient_last_name) AS 'Patient Name',
    b.amount AS 'Amount',
    b.date AS 'Billing Date'
FROM Patients AS p
JOIN Billing AS b ON p.patient_id = b.Patients_patient_id
WHERE b.payment_status = 'unpaid'
ORDER BY b.date;
```

Output :-



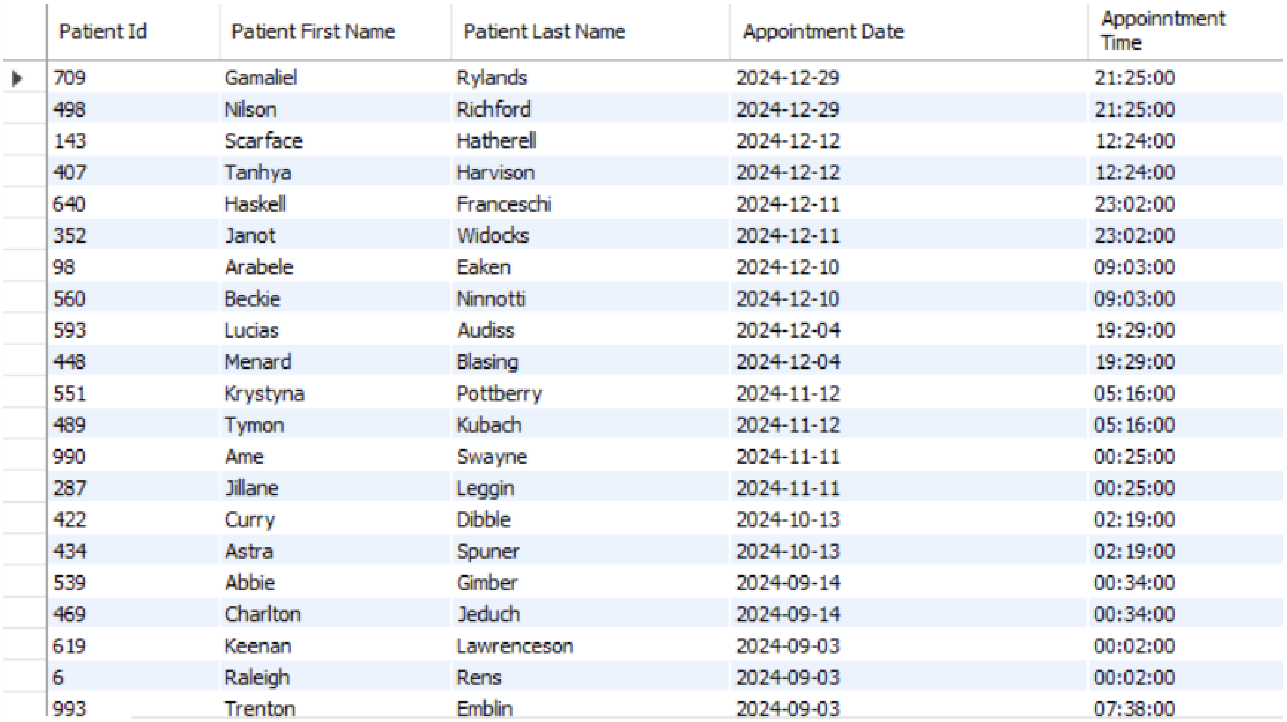
Patient Id	Patient Name	Amount	Billing Date
281	Jacintha Brewitt	9915	2000-01-23
765	Farrand Ambrogioni	1380	2000-03-25
810	Dita Wanderschek	483	2000-03-25
813	Tyrus Reymers	4611	2000-03-27
485	Conroy Connock	8413	2000-03-27
218	Tanitansy Garden	4366	2000-03-28
633	Doretta Finlayson	8204	2000-03-29
377	Marcello Nucciotti	7594	2000-03-30
924	Helga McEvay	6448	2000-03-30
332	Jayson Dobrowlski	6169	2000-04-21
717	Tobe Conibere	4110	2000-05-20
86	Alisha Forseith	9511	2000-05-20
187	Giavani Burkart	4082	2000-05-28
782	Halsy Haug	5140	2000-06-02
969	Ruggiero Copyn	2849	2000-07-01
424	Carlie Greenwood	5489	2000-07-01
41	Rustin Cowmeadow	8296	2000-07-11
531	Antonia Bernardelli	9491	2000-08-09
450	Jorry Kalberer	6670	2000-08-13
865	Nicolis Bowshire	7669	2000-08-14
670	Edwin Stening	209	2000-08-31

9. This query lists the Patients appointment history in descending order.

Query :-

```
SELECT
   p.patient_id AS 'Patient Id',
   p.patient_first_name AS 'Patient First Name',
   p.patient_last_name AS 'Patient Last Name',
   a.date AS 'Appointment Date',
   a.time AS 'Appoinntment Time'
FROM Patients AS p
JOIN Appointment AS a ON p.patient_id = a.Patients_patient_id
ORDER BY a.date DESC;
```

Output :-



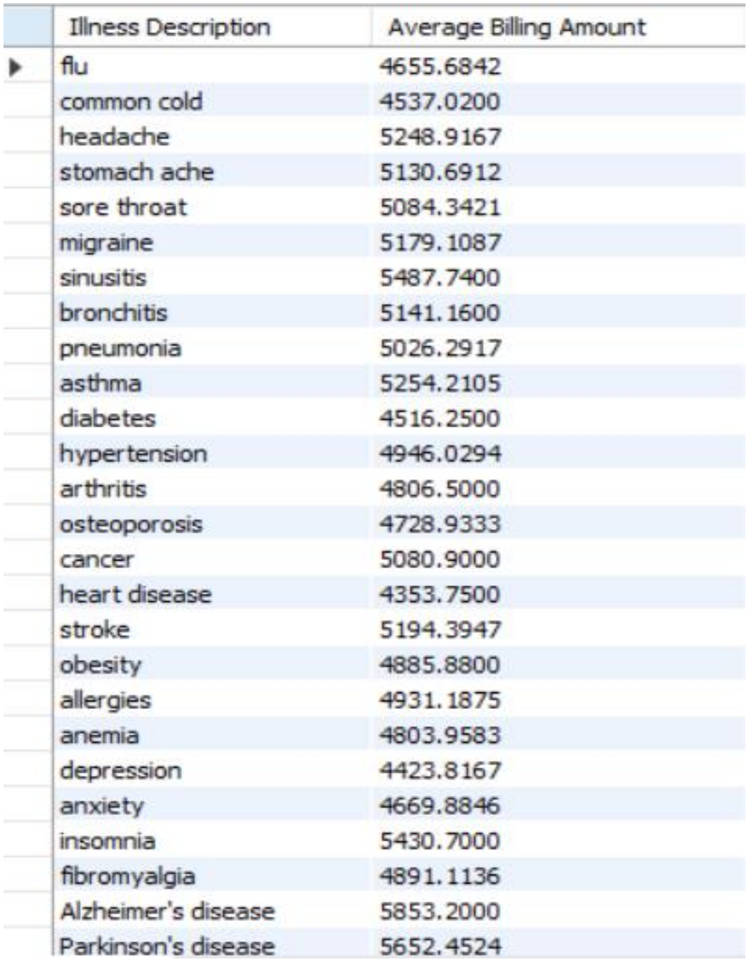
Patient Id	Patient First Name	Patient Last Name	Appointment Date	Appoinntment Time
709	Gamaliel	Rylands	2024-12-29	21:25:00
498	Nilson	Richford	2024-12-29	21:25:00
143	Scarface	Hatherell	2024-12-12	12:24:00
407	Tanhya	Harvison	2024-12-12	12:24:00
640	Haskell	Franceschi	2024-12-11	23:02:00
352	Janot	Widocks	2024-12-11	23:02:00
98	Arabele	Eaken	2024-12-10	09:03:00
560	Beckie	Ninnotti	2024-12-10	09:03:00
593	Lucias	Audiss	2024-12-04	19:29:00
448	Menard	Blasing	2024-12-04	19:29:00
551	Krystyna	Pottberry	2024-11-12	05:16:00
489	Tymon	Kubach	2024-11-12	05:16:00
990	Ame	Swayne	2024-11-11	00:25:00
287	Jillane	Leggin	2024-11-11	00:25:00
422	Curry	Dibble	2024-10-13	02:19:00
434	Astra	Spuner	2024-10-13	02:19:00
539	Abbie	Gimber	2024-09-14	00:34:00
469	Charlton	Jeduch	2024-09-14	00:34:00
619	Keenan	Lawrenceson	2024-09-03	00:02:00
6	Raleigh	Rens	2024-09-03	00:02:00
993	Trenton	Emblin	2024-09-03	07:38:00

10. This query gets the average billing amount per illness type.

Query :-

```
SELECT
il.description AS 'Illness Description',
        AVG(b.amount) AS 'Average Billing Amount'
FROM Illness_Details AS il
JOIN Patients_has_Illness_Details AS pid ON il.illness_id = pid.Illness_Details_illness_id
JOIN Billing AS b ON pid.Patients_patient_id = b.Patients_patient_id
GROUP BY il.description;
```

Output :-



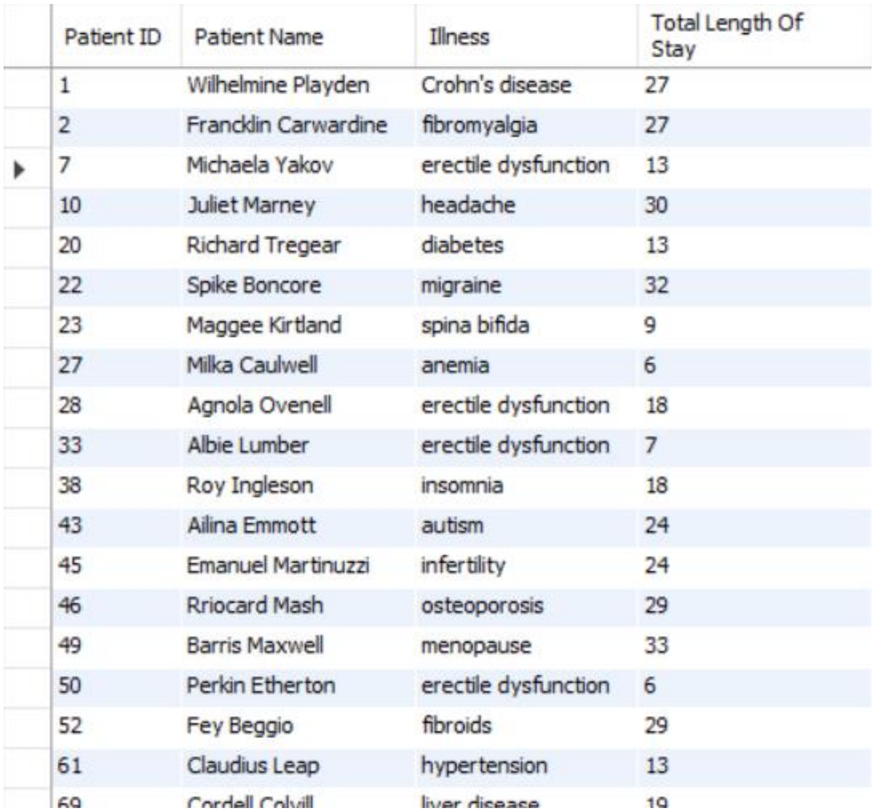
Illness Description	Average Billing Amount
flu	4655.6842
common cold	4537.0200
headache	5248.9167
stomach ache	5130.6912
sore throat	5084.3421
migraine	5179.1087
sinusitis	5487.7400
bronchitis	5141.1600
pneumonia	5026.2917
asthma	5254.2105
diabetes	4516.2500
hypertension	4946.0294
arthritis	4806.5000
osteoporosis	4728.9333
cancer	5080.9000
heart disease	4353.7500
stroke	5194.3947
obesity	4885.8800
allergies	4931.1875
anemia	4803.9583
depression	4423.8167
anxiety	4669.8846
insomnia	5430.7000
fibromyalgia	4891.1136
Alzheimer's disease	5853.2000
Parkinson's disease	5652.4524

11. This query will list patients that have not been seen by a doctor, what their illness was, and how long did they stay in the hospital.

Query:

```
  SELECT
    p.patient_id AS 'Patient ID',
    CONCAT(p.patient_first_name, ' ', p.patient_last_name) AS 'Patient Name',
    GROUP_CONCAT(DISTINCT id.description) AS Illness,
    SUM(DATEDIFF(mr.discharge_date, mr.addmission_date)) AS 'Total Length Of Stay'
  FROM
    Patients p
    INNER JOIN Patients_has_Illness_Details phi ON p.patient_id = phi.Patients_patient_id
    INNER JOIN Illness_Details id ON phi.Illness_Details_illness_id = id.illness_id
    INNER JOIN Medical_Records mr ON p.patient_id = mr.Patients_patient_id
    LEFT JOIN Patients_has_Staff phs ON p.patient_id = phs.Patients_patient_id
    INNER JOIN Staff s ON phs.Staff_staff_id = s.staff_id
    INNER JOIN Staff_Type st ON s.staff_id = st.Staff_staff_id
  WHERE
    st.staff_type_name IN ('Registered Nurse', 'Nurse Practitioner')
    AND NOT EXISTS (
      SELECT 1
      FROM Patients_has_Staff phs2
      INNER JOIN Staff s2 ON phs2.Staff_staff_id = s2.staff_id
      INNER JOIN Staff_Type st2 ON s2.staff_id = st2.Staff_staff_id
      WHERE
        phs2.Patients_patient_id = p.patient_id
        AND st2.staff_type_name = 'Doctor'
    )
  GROUP BY
    p.patient_id;
```

Output: -



Patient ID	Patient Name	Illness	Total Length Of Stay
1	Wilhelmine Playden	Crohn's disease	27
2	Francklin Carwardine	fibromyalgia	27
7	Michaela Yakov	erectile dysfunction	13
10	Juliet Marney	headache	30
20	Richard Tregear	diabetes	13
22	Spike Boncore	migraine	32
23	Maggee Kirtland	spina bifida	9
27	Milka Caulwell	anemia	6
28	Agnola Ovenell	erectile dysfunction	18
33	Albie Lumber	erectile dysfunction	7
38	Roy Ingleson	insomnia	18
43	Ailina Emmott	autism	24
45	Emanuel Martinuzzi	infertility	24
46	Rriocard Mash	osteoporosis	29
49	Barris Maxwell	menopause	33
50	Perkin Etherton	erectile dysfunction	6
52	Fey Beggio	fibroids	29
61	Claudius Leap	hypertension	13
69	Cordell Colvill	liver disease	19

Stored Procedures:

1. Procedure to quickly generate a monthly report table for Number of Transactions, Total Income, and Average Amount of each transaction within a set called Year and Month.

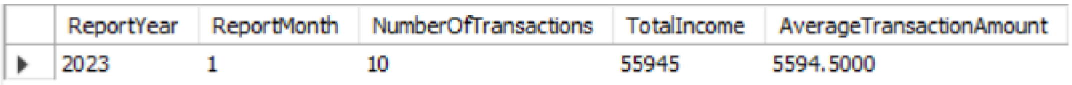
Procedure SQL:-

```
CREATE DEFINER=`mm_team05_01`@`%` PROCEDURE `GetPatientMedicalRecordsByIllness`(IN patient_id_param INT, IN illness_description_param VARCHAR(45))

BEGIN

   SELECT mr.record_id, mr.admission_date, mr.discharge_date, il.description

   FROM Medical_Records mr

   JOIN Illness_Details_has_Medical_Records idmr ON mr.record_id = idmr.Medical_Records_record_id

   JOIN Illness_Details il ON idmr.Illness_Details_illness_id = il.illness_id

   JOIN Patients_has_Illness_Details pid ON il.illness_id = pid.Illness_Details_illness_id

   WHERE pid.Patients_patient_id = patient_id_param AND il.description LIKE CONCAT('%', illness_description_param, '%');

END
```

Example Call:

```
CALL GetMonthlyIncomeReportDetailed(2023, 1)
```

Example Output:



	ReportYear	ReportMonth	NumberOfTransactions	TotalIncome	AverageTransactionAmount
▶	2023	1	10	55945	5594.5000

2. Procedure to quickly check if there are illnesses that patients had and haven't seen a doctor staff member and only have seen nurses to allow for better service in the future. This is purposefully setup as a simple call so anyone can use it if needed for quick use.

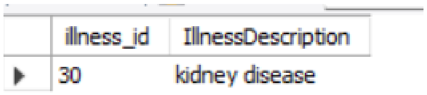
Procedure SQL:-

```
CREATE DEFINER=`mm_team05_01`@`%` PROCEDURE `GetIllnessesWithoutDoctorVisits`()
BEGIN
  SELECT
    id.illness_id,
    id.description AS IllnessDescription
  FROM
    Illness_Details id
  WHERE NOT EXISTS (
    SELECT 1
    FROM Patients_has_Illness_Details phi
    INNER JOIN Patients_has_Staff phs ON phi.Patients_patient_id = phs.Patients_patient_id
    INNER JOIN Staff s ON phs.Staff_staff_id = s.staff_id
    INNER JOIN Staff_Type st ON s.staff_id = st.Staff_staff_id
    WHERE
      phi.Illness_Details_illness_id = id.illness_id
      AND st.staff_type_name = 'Doctor'
  )
  GROUP BY
    id.illness_id, id.description;
END
```

Example Call:

```
CALL GetIllnessesWithoutDoctorVisits()
```

Example Output:



illness_id	IllnessDescription
30	kidney disease