Egyptian Russian University

Faculty of Management, Economic and Business Technology

Pharmacy Management System

Database System 2 Project
IST207
Group 6

Supervised by
Lecturer Assistant Reham Abdallah
Doctor Ghada Nabil

Team Drop Database

Alhosna Ezzat Elshenawy	224222
Mariam Tamer Mohammed	224284
Eriny Daood Zakaria	224144
Mariem Hany Monir	224186
Ahmed Ebrahim Ahmed	224257

❖ *Requirement*

a) Orders are accepted by patients.
b) Orders are made by patients.
c) Insrance covers prescriptions.
d) Bills contain details of orders.
e) Inventory contains drugs.
f) Patients receive prescriptions.
g) Doctors read medical histories.
h) Doctors prescribe drugs.
i) Suppliers supply drugs.
j) Drug companies manufacture drugs.

❖ *Business Rules*

Welcome in our pharmacy management system which its members are (Doctors, employee, patient, supplier)

The pharmacy management system has person each on has a unique number, first name, last name, phone, address, email which they are multi value attribute.

Person may be patients, doctor, supplier, or employee.

An employee accepts many orders that have attributes salary, performance, and Date of hiring the orders attribute's identity number, status, total price, and order date and order contain one bill.

Bill contains one order the bill has attributes identifier Id Payment _Method, Discounts, total amount.

Patient has attributes medical history patient make one order and order made by many patients.

Patients receive many Prescriptions and that have attributes identifier P_ID, SSN, P_Date.

Prescriptions read by many doctors and doctors attributes are salary, specialization, and doctor read one Prescriptions.

Doctors describe many drugs which its items are identifier Drug_ID unit_price, Expiry date, description, Drugname. And drugs are being described by many doctors.

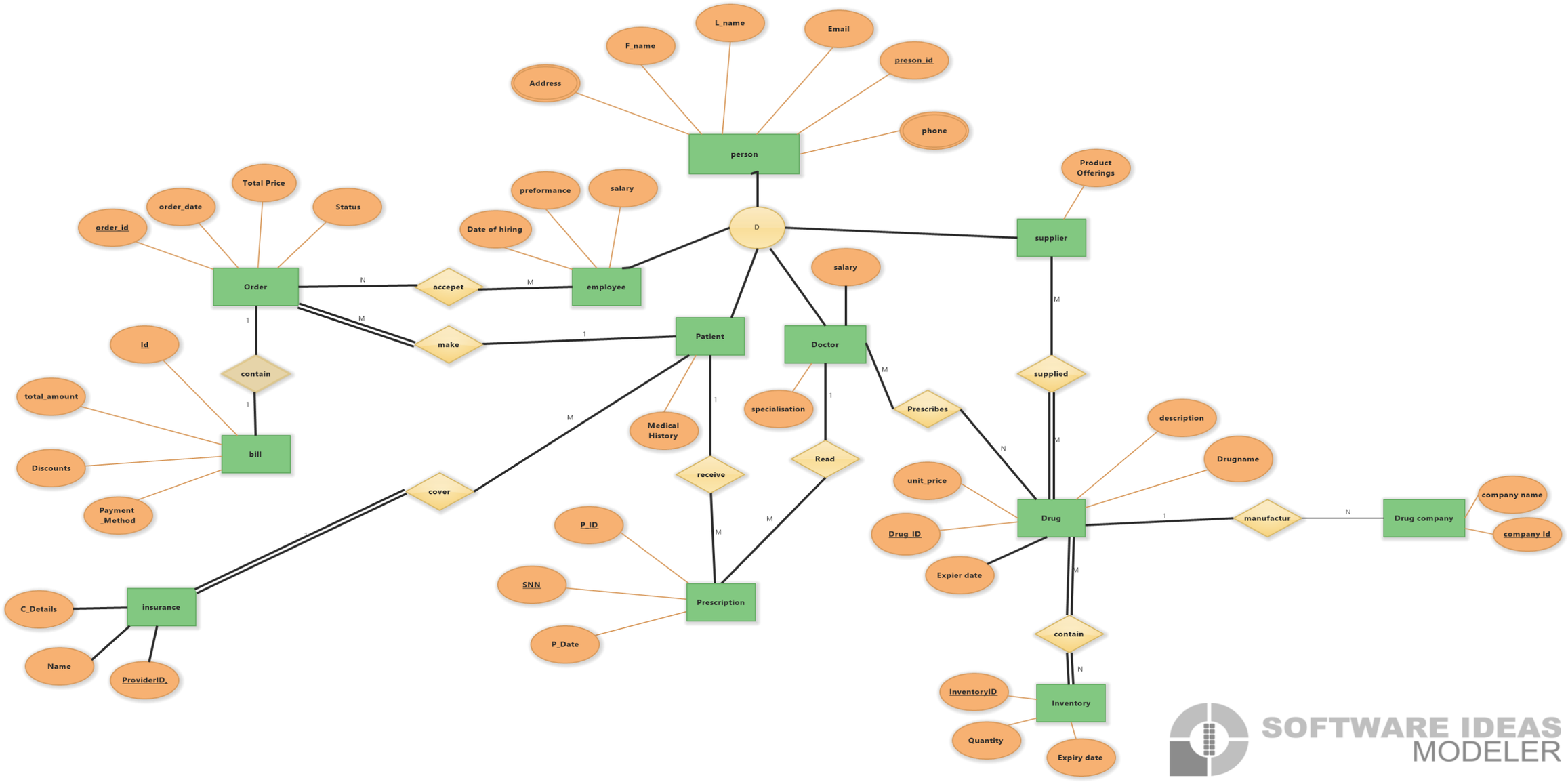
Drugs supplied by many suppliers and suppliers has product offering.

Also, supplier supply many drugs, and inventory identifier InventoryID, Quantity, Expiry date.

Drugs are being manufactured by company drug, one drug is being manufactured by many companies drug, and many company drug manufactured by one drug.

❖ *Databases Design*

✓ **ERD**



✓ DB schema (Mapping)

Person (super entity)

Person_id [1]	F_name	L_name	email

Person address

Person_id [1]	address

Person phone

Person_id[1]	phone

Employee (sub entity)

Employee_id [2]	Person_id[1]	salary	performance	Date of hiring

Doctor (sub entity)

Doctor_id [3]	Person_id[1]	specialization	salary

insurance

Provider_ID [4]	Name	C_Details

Patient (sub entity)

Patient_id [5]	Person_id[1]	Medical History	Provider_ID [4]

Order

Order_id [6]	Order_data	Total_price	status	patient_id [5]

Employee_order

Employee_id [2]	Order_id [6]

Bill

Bill_id [7]	Discounts	total_amount	Payment _Method	Order_id [6]

Prescription

P_id [8]	SNN	P_date	patient_id [5]	doctor_id [3]

Supplier (sub entity)

Supplier_id [9]	Person_id [1]	Product Offerings

Durg

Drug_ID [10]	Durg_name	description	Unit_price

Drug company

Company Id [11]	company name	Drug_ID [10]

Inventory

Inventory-id [12]	Quantity	Expiry date

Inventory_durg

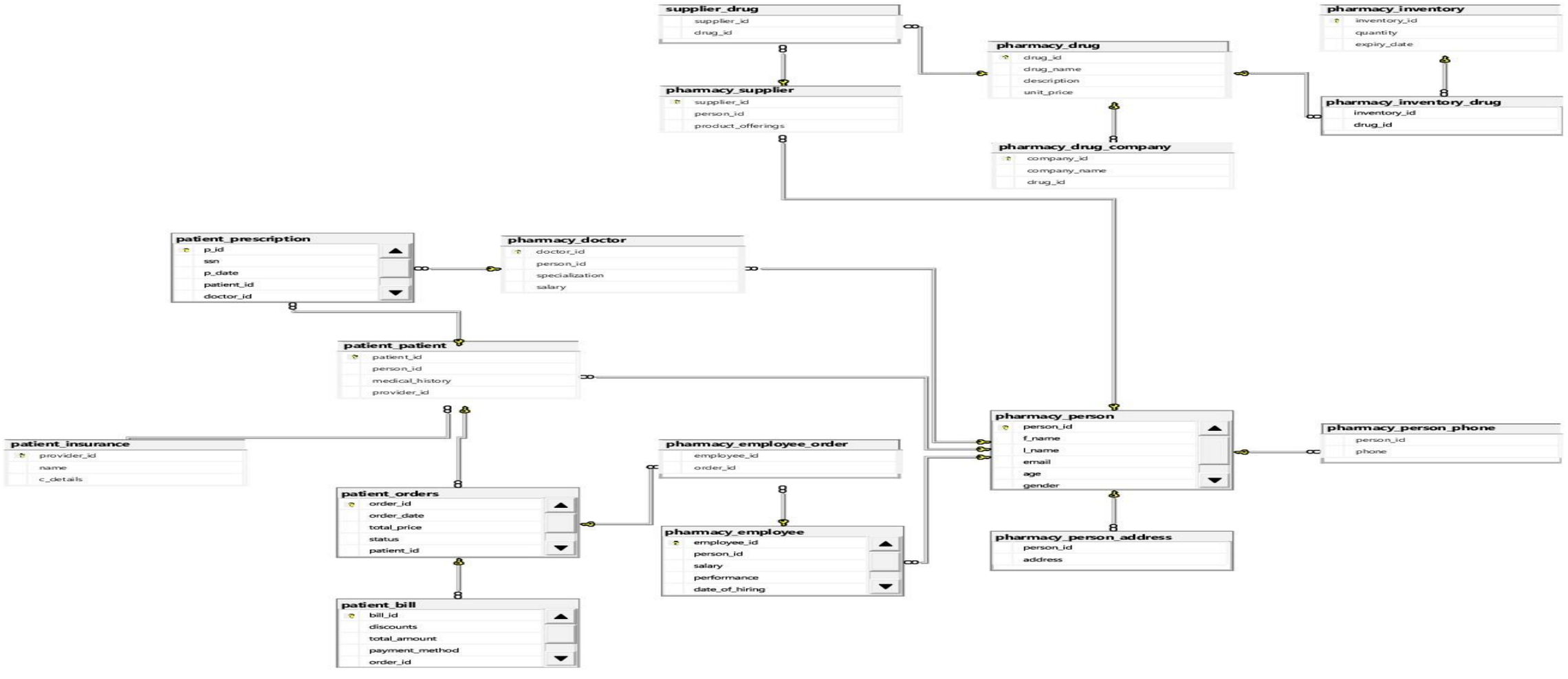
Inventory-id [12]	Drug_ID [10]

Supplier_durg

supplier_id [9]	Drug_ID [10]

DB Daigram

❖ *Database Implementation*



✓ SQL Code

```
create database pharmacy_management_system
create schema pharmacy
create schema patient
--super entity
create table pharmacy_person (
    person_id int primary key,
    f_name varchar(50),
    l_name varchar(50),
    email varchar(100)
);

create table pharmacy_person_address (
    person_id int,
    address varchar(255),
    foreign key (person_id) references pharmacy_person(person_id)
);


create table pharmacy_person_phone (
    person_id int,
    phone varchar(20),
    foreign key (person_id) references pharmacy_person(person_id)
);
--sub entity
create table pharmacy_employee (
    employee_id int primary key,
    person_id int,
    salary decimal(10, 2),
    performance varchar(50),
    date_of_hiring date,
    foreign key (person_id) references pharmacy_person(person_id)
);

-- sub entity
create table pharmacy_doctor (
    doctor_id int primary key,
    person_id int,
    specialization varchar(100),
    salary decimal(10, 2),
    foreign key (person_id) references pharmacy_person(person_id)
);

create table patient_insurance (
    provider_id int primary key,
    name varchar(100),
    c_details varchar(255)
);

-- sub entity
create table patient_patient (
    patient_id int primary key,
```

```
    person_id int,
    medical_history text,
    provider_id int,
    foreign key (person_id) references pharmacy_person(person_id),
    foreign key (provider_id) references patient_insurance(provider_id)
);

create table patient_orders (
    order_id int primary key,
    order_date date,
    total_price decimal(10, 2),
    status varchar(50),
    patient_id int,
    foreign key (patient_id) references patient_patient(patient_id)
);

create table pharmacy_employee_order (
    employee_id int,
    order_id int,
    foreign key (employee_id) references pharmacy_employee(employee_id),
    foreign key (order_id) references patient_orders(order_id)
);

create table patient_bill (
    bill_id int primary key,
    discounts decimal(10, 2),
    total_amount decimal(10, 2),
    payment_method varchar(50),
    order_id int,
    foreign key (order_id) references patient_orders(order_id)
);

create table patient_prescription (
    p_id int primary key,
    ssn varchar(20),
    p_date date,
    patient_id int,
    doctor_id int,
    foreign key (patient_id) references patient_patient(patient_id),
    foreign key (doctor_id) references pharmacy_doctor(doctor_id)
);

--sub entity
create table pharmacy_supplier (
    supplier_id int primary key,
    person_id int,
    product_offerings text,
    foreign key (person_id) references pharmacy_person(person_id)
);

create table pharmacy_drug (
    drug_id int primary key,
    drug_name varchar(100),
```

```
    description text,
    unit_price decimal(10, 2)
);


create table pharmacy_drug_company (
    company_id int primary key,
    company_name varchar(100),
    drug_id int,
    foreign key (drug_id) references pharmacy_drug(drug_id)
);


create table pharmacy_inventory (
    inventory_id int primary key,
    quantity int,
    expiry_date date
);

create table pharmacy_inventory_drug (
    inventory_id int,
    drug_id int,
    foreign key (inventory_id) references pharmacy_inventory(inventory_id),
    foreign key (drug_id) references pharmacy_drug(drug_id)
);
create table supplier_drug (
    supplier_id int,
    drug_id int,
    foreign key (supplier_id) references pharmacy_supplier(supplier_id),
    foreign key (drug_id) references pharmacy_drug(drug_id)
);
-- Alter
ALTER TABLE pharmacy_Person
ADD age int;
ALTER TABLE pharmacy_Person
ADD gender varchar(99);
```

✓ Data Manipulation Language (DML)

○ Insert

```
-- Insert data into pharmacy_person table
INSERT INTO pharmacy_person (person_id, f_name, l_name, email, age, gender) VALUES
(1, 'John', 'Doe', 'john@example.com', 30, 'Male'),
(2, 'Jane', 'Smith', 'jane@example.com', 25, 'Female'),
(3, 'Michael', 'Johnson', 'michael@example.com', 40, 'Male'),
(4, 'Emily', 'Williams', 'emily@example.com', 35, 'Female'),
(5, 'David', 'Brown', 'david@example.com', 28, 'Male'),
(6, 'Sarah', 'Jones', 'sarah@example.com', 32, 'Female'),
(7, 'Matthew', 'Davis', 'matthew@example.com', 45, 'Male'),
(8, 'Jessica', 'Martinez', 'jessica@example.com', 27, 'Female'),
(9, 'Christopher', 'Rodriguez', 'christopher@example.com', 38, 'Male'),
(10, 'Amanda', 'Taylor', 'amanda@example.com', 29, 'Female');

-- Insert data into pharmacy_person_address table
INSERT INTO pharmacy_person_address (person_id, address) VALUES
(1, '123 Main St'),
(2, '456 Elm St'),
(3, '789 Oak St'),
(4, '101 Pine St'),
(5, '222 Maple St'),
(6, '333 Cedar St'),
(7, '444 Birch St'),
(8, '555 Walnut St'),
(9, '666 Elm St'),
(10, '777 Pine St');

-- Insert data into pharmacy_person_phone table
INSERT INTO pharmacy_person_phone (person_id, phone) VALUES
(1, '555-1234'),
(2, '555-5678'),
(3, '555-9876'),
(4, '555-5432'),
(5, '555-6789'),
(6, '555-4321'),
(7, '555-8765'),
(8, '555-2109'),
(9, '555-3456'),
```

```
    (10, '555-7890');

    -- Insert data into pharmacy_employee table
    INSERT INTO pharmacy_employee (employee_id, person_id, salary, performance,
date_of_hiring) VALUES
    (101, 1, 50000.00, 'Excellent', '2023-01-15'),
    (102, 2, 60000.00, 'Good', '2023-02-20'),
    (103, 3, 55000.00, 'Excellent', '2022-11-10'),
    (104, 4, 65000.00, 'Good', '2022-12-05'),
    (105, 5, 52000.00, 'Excellent', '2023-03-18'),
    (106, 6, 62000.00, 'Good', '2023-04-22'),
    (107, 7, 54000.00, 'Excellent', '2022-09-28'),
    (108, 8, 63000.00, 'Good', '2022-10-30'),
    (109, 9, 56000.00, 'Excellent', '2023-05-14'),
    (110, 10, 64000.00, 'Good', '2023-06-25');

    -- Insert data into pharmacy_doctor table
    INSERT INTO pharmacy_doctor (doctor_id, person_id, specialization, salary) VALUES
    (201, 2, 'Cardiologist', 80000.00),
    (202, 1, 'Pediatrician', 75000.00),
    (203, 4, 'Dermatologist', 85000.00),
    (204, 3, 'Neurologist', 90000.00),
    (205, 6, 'Orthopedic Surgeon', 95000.00),
    (206, 5, 'Psychiatrist', 85000.00),
    (207, 8, 'Oncologist', 100000.00),
    (208, 7, 'Gynecologist', 82000.00),
    (209, 10, 'Urologist', 88000.00),
    (210, 9, 'Endocrinologist', 86000.00);

    -- Insert data into patient_insurance table
    INSERT INTO patient_insurance (provider_id, name, c_details) VALUES
    (1, 'HealthCare Inc.', 'Contact: healthcare@example.com'),
    (2, 'MediCare Ltd.', 'Contact: medicare@example.com'),
    (3, 'Wellness Insurance', 'Contact: wellness@example.com'),
    (4, 'SafeGuard Insurance', 'Contact: safeguard@example.com'),
    (5, 'Coverage Plus', 'Contact: coverage@example.com'),
    (6, 'Total Care Insurance', 'Contact: totalcare@example.com'),
    (7, 'Premier Health', 'Contact: premierhealth@example.com'),
    (8, 'FirstChoice Insurance', 'Contact: firstchoice@example.com'),
    (9, 'Unity Health Insurance', 'Contact: unityhealth@example.com'),
    (10, 'Optimum Insurance', 'Contact: optimum@example.com');
```

```
    -- Insert data into patient_patient table
    INSERT INTO patient_patient (patient_id, person_id, medical_history, provider_id)
VALUES
    (301, 1, 'No significant medical history', 1),
    (302, 2, 'Allergic to penicillin', 2),
    (303, 3, 'Hypertension', 3),
    (304, 4, 'Asthma', 4),
    (305, 5, 'Depression', 5),
    (306, 6, 'Diabetes', 6),
    (307, 7, 'High cholesterol', 7),
    (308, 8, 'Thyroid disorder', 8),
    (309, 9, 'Obesity', 9),
    (310, 10, 'Arthritis', 10);

    -- Insert data into patient_orders table
    INSERT INTO patient_orders (order_id, order_date, total_price, status, patient_id)
VALUES
    (401, '2024-05-20', 150.00, 'Pending', 301),
    (402, '2024-05-21', 200.00, 'Shipped', 302),
    (403, '2024-05-22', 180.00, 'Delivered', 303),
    (404, '2024-05-23', 220.00, 'Pending', 304),
    (405, '2024-05-24', 190.00, 'Shipped', 305),
    (406, '2024-05-25', 210.00, 'Delivered', 306),
    (407, '2024-05-26', 160.00, 'Pending', 307),
    (408, '2024-05-27', 230.00, 'Shipped', 308),
    (409, '2024-05-28', 200.00, 'Delivered', 309),
    (410, '2024-05-29', 240.00, 'Pending', 310);

    INSERT INTO pharmacy_employee_order (employee_id, order_id) VALUES
    (101, 401),
    (102, 402),
    (103, 403),
    (104, 404),
    (105, 405),
    (106, 406),
    (107, 407),
    (108, 408),
    (109, 409),
    (110, 410);

    -- Insert data into patient_bill table
    INSERT INTO patient_bill (bill_id, discounts, total_amount, payment_method,
order_id) VALUES
```

```
    (501, 10.00, 140.00, 'Credit Card', 401),
    (502, 0.00, 200.00, 'Cash', 402),
    (503, 5.00, 175.00, 'Credit Card', 403),
    (504, 15.00, 205.00, 'Cash', 404),
    (505, 20.00, 170.00, 'Credit Card', 405),
    (506, 0.00, 210.00, 'Cash', 406),
    (507, 10.00, 150.00, 'Credit Card', 407),
    (508, 0.00, 230.00, 'Cash', 408),
    (509, 8.00, 192.00, 'Credit Card', 409),
    (510, 12.00, 228.00, 'Cash', 410);

    -- Insert data into patient_prescription table
    INSERT INTO patient_prescription (p_id, ssn, p_date, patient_id, doctor_id) VALUES
    (601, '123-45-6789', '2024-05-22', 301, 202),
    (602, '987-65-4321', '2024-05-21', 302, 201),
    (603, '456-78-9012', '2024-05-20', 303, 204),
    (604, '789-01-2345', '2024-05-19', 304, 203),
    (605, '234-56-7890', '2024-05-18', 305, 206),
    (606, '567-89-0123', '2024-05-17', 306, 205),
    (607, '890-12-3456', '2024-05-16', 307, 208),
    (608, '345-67-8901', '2024-05-15', 308, 207),
    (609, '678-90-1234', '2024-05-14', 309, 210),
    (610, '901-23-4567', '2024-05-13', 310, 209);

    -- Insert data into pharmacy_supplier table
    INSERT INTO pharmacy_supplier (supplier_id, person_id, product_offerings) VALUES
    (701, 1, 'Pharmaceuticals'),
    (702, 2, 'Medical equipment'),
    (703, 3, 'Medical supplies'),
    (704, 4, 'Laboratory equipment'),
    (705, 5, 'Diagnostic devices'),
    (706, 6, 'Surgical instruments'),
    (707, 7, 'Rehabilitation aids'),
    (708, 8, 'Orthopedic implants'),
    (709, 9, 'Dental supplies'),
    (710, 10, 'Veterinary products');

    -- Insert data into pharmacy_drug table
    INSERT INTO pharmacy_drug (drug_id, drug_name, description, unit_price) VALUES
    (801, 'Aspirin', 'Pain reliever', 5.00),
    (802, 'Amoxicillin', 'Antibiotic', 10.00),
    (803, 'Lisinopril', 'Blood pressure medication', 8.00),
```

```
    (804, 'Albuterol', 'Bronchodilator', 12.00),
    (805, 'Sertraline', 'Antidepressant', 15.00),
    (806, 'Metformin', 'Diabetes medication', 10.00),
    (807, 'Simvastatin', 'Cholesterol-lowering drug', 7.00),
    (808, 'Levothyroxine', 'Thyroid medication', 10.00),
    (809, 'Atorvastatin', 'Cholesterol-lowering drug', 8.00),
    (810, 'Omeprazole', 'Acid reflux medication', 9.00);

    -- Insert data into pharmacy_drug_company table
    INSERT INTO pharmacy_drug_company (company_id, company_name, drug_id) VALUES
    (901, 'PharmaCorp', 801),
    (902, 'MedEquip Ltd.', 802),
    (903, 'MedSupply Inc.', 803),
    (904, 'LabTech Solutions', 804),
    (905, 'DiagnosTech', 805),
    (906, 'MetaPharm', 806),
    (907, 'StatMed Laboratories', 807),
    (908, 'ThyroPharm', 808),
    (909, 'CholestaCare', 809),
    (910, 'AcidBlock Pharmaceuticals', 810);

    -- Insert data into pharmacy_inventory table
    INSERT INTO pharmacy_inventory (inventory_id, quantity, expiry_date) VALUES
    (1001, 100, '2025-01-01'),
    (1002, 50, '2024-12-01'),
    (1003, 75, '2024-11-01'),
    (1004, 120, '2024-10-01'),
    (1005, 90, '2024-09-01'),
    (1006, 110, '2024-08-01'),
    (1007, 85, '2024-07-01'),
    (1008, 130, '2024-06-01'),
    (1009, 95, '2024-05-01'),
    (1010, 105, '2024-04-01');

    -- Insert data into pharmacy_inventory_drug table
    INSERT INTO pharmacy_inventory_drug (inventory_id, drug_id) VALUES
    (1001, 801),
    (1002, 802),
    (1003, 803),
    (1004, 804),
    (1005, 805),
    (1006, 806),
```

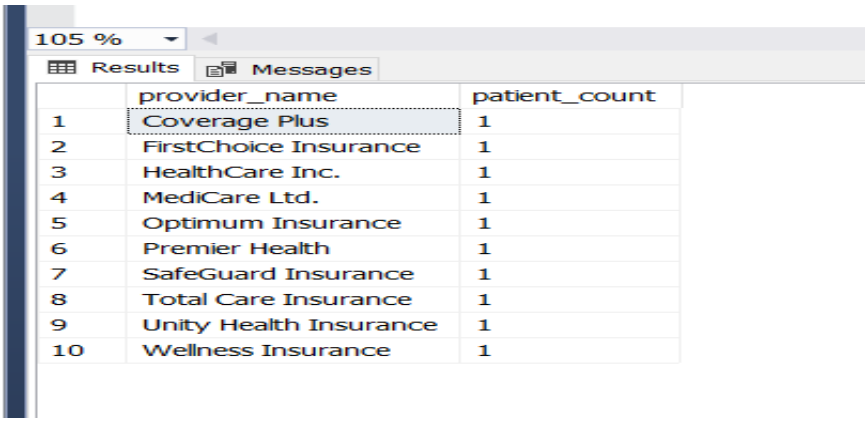
```
    (1007, 807),
    (1008, 808),
    (1009, 809),
    (1010, 810);
```

✓ Data Query Language (DQL)

○ Join (more than two tables)

1.

```
---patients have insurance from each provider
select pi.name as provider_name, COUNT(pp.patient_id) as patient_count
from patient_patient pp join patient_insurance pi
on pp.provider_id = pi.provider_id
group by pi.name;
```



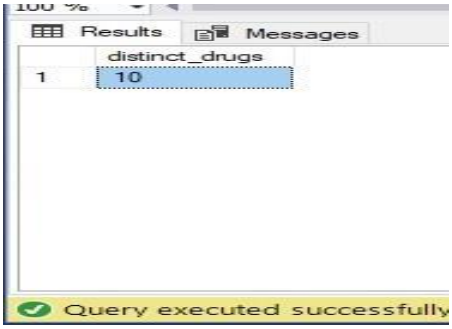
105 %

Results | Messages

	provider_name	patient_count
1	Coverage Plus	1
2	FirstChoice Insurance	1
3	HealthCare Inc.	1
4	MediCare Ltd.	1
5	Optimum Insurance	1
6	Premier Health	1
7	SafeGuard Insurance	1
8	Total Care Insurance	1
9	Unity Health Insurance	1
10	Wellness Insurance	1

2.

```
--- distinct drugs are there in the inventory
select count(distinct drug_id) as distinct_drugs
from pharmacy_inventory_drug;
```



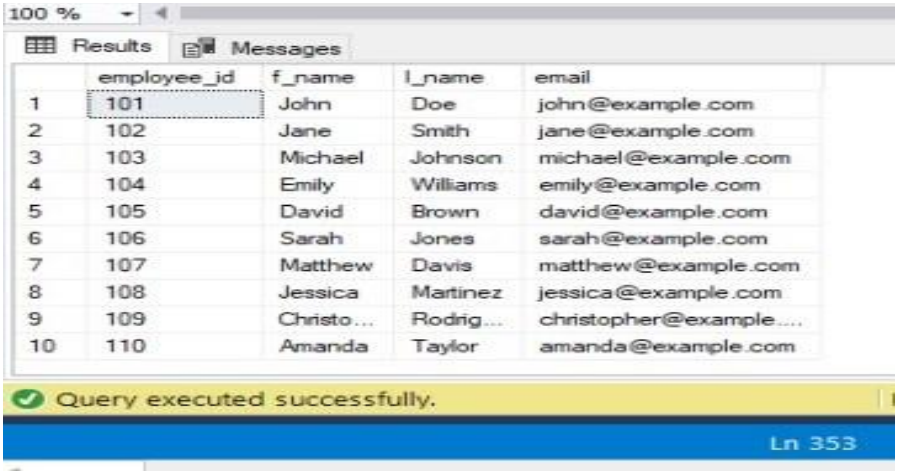
Results | Messages

	distinct_drugs
1	10

Query executed successfully

3.

```
--- retrieve a list of all employees along with their associated personal information
(first name, last name, and email)
select e.employee_id, p.f_name, p.l_name, p.email
from pharmacy_employee e join pharmacy_person p
on e.person_id = p.person_id;
```



100 %

Results | Messages

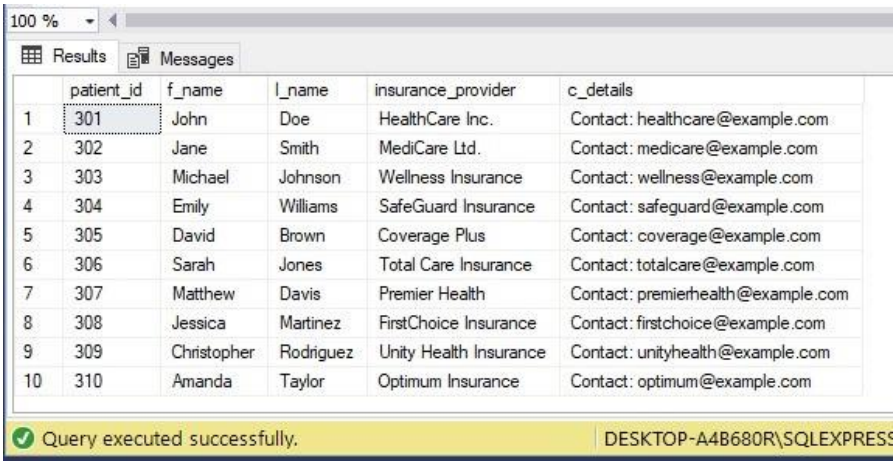
	employee_id	f_name	l_name	email
1	101	John	Doe	john@example.com
2	102	Jane	Smith	jane@example.com
3	103	Michael	Johnson	michael@example.com
4	104	Emily	Williams	emily@example.com
5	105	David	Brown	david@example.com
6	106	Sarah	Jones	sarah@example.com
7	107	Matthew	Davis	matthew@example.com
8	108	Jessica	Martinez	jessica@example.com
9	109	Christo...	Rodrig...	christopher@example....
10	110	Amanda	Taylor	amanda@example.com

Query executed successfully.

Ln 353

4.

```
--- find all patients along with their insurance provider details
select pp.patient_id, p.f_name, p.l_name, pi.name as insurance_provider, pi.c_details
from patient_patient pp join pharmacy_person p
on pp.person_id = p.person_id join patient_insurance pi
on pp.provider_id = pi.provider_id;
```



100 %

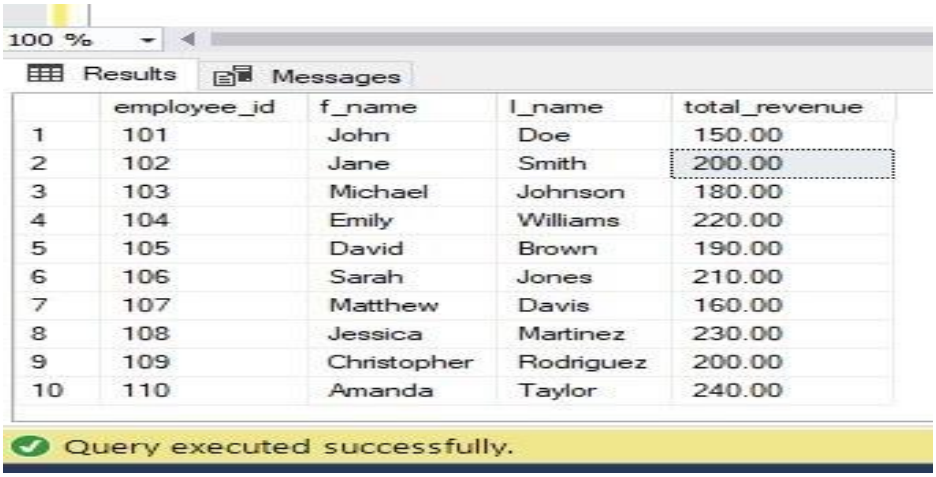
Results | Messages

	patient_id	f_name	l_name	insurance_provider	c_details
1	301	John	Doe	HealthCare Inc.	Contact: healthcare@example.com
2	302	Jane	Smith	MediCare Ltd.	Contact: medicare@example.com
3	303	Michael	Johnson	Wellness Insurance	Contact: wellness@example.com
4	304	Emily	Williams	SafeGuard Insurance	Contact: safeguard@example.com
5	305	David	Brown	Coverage Plus	Contact: coverage@example.com
6	306	Sarah	Jones	Total Care Insurance	Contact: totalcare@example.com
7	307	Matthew	Davis	Premier Health	Contact: premierhealth@example.com
8	308	Jessica	Martinez	FirstChoice Insurance	Contact: firstchoice@example.com
9	309	Christopher	Rodriguez	Unity Health Insurance	Contact: unityhealth@example.com
10	310	Amanda	Taylor	Optimum Insurance	Contact: optimum@example.com

Query executed successfully. DESKTOP-A4B680R\SQLEXPRESS

5.

```
--- total revenue generated from orders processed by each employee
select  e.employee_id, p.f_name, p.l_name, sum(o.total_price) as  total_revenue
from pharmacy_employee e join pharmacy_person p
on e.person_id = p.person_id join pharmacy_employee_order eo
on e.employee_id = eo.employee_id join patient_orders o
on eo.order_id = o.order_id
group by e.employee_id, p.f_name, p.l_name;
```



100 %

Results | Messages

	employee_id	f_name	l_name	total_revenue
1	101	John	Doe	150.00
2	102	Jane	Smith	200.00
3	103	Michael	Johnson	180.00
4	104	Emily	Williams	220.00
5	105	David	Brown	190.00
6	106	Sarah	Jones	210.00
7	107	Matthew	Davis	160.00
8	108	Jessica	Martinez	230.00
9	109	Christopher	Rodriguez	200.00
10	110	Amanda	Taylor	240.00

Query executed successfully.

6.

```
---list all orders along with the patient's name and the total price of each order
select o.order_id, p.f_name, p.l_name, o.total_price
from patient_orders o join patient_patient pp
on o.patient_id = pp.patient_id join pharmacy_person p
on pp.person_id = p.person_id;
```

100 %

Results | Messages

	order_id	f_name	l_name	total_price
1	401	John	Doe	150.00
2	402	Jane	Smith	200.00
3	403	Michael	Johnson	180.00
4	404	Emily	Williams	220.00
5	405	David	Brown	190.00
6	406	Sarah	Jones	210.00
7	407	Matthew	Davis	160.00
8	408	Jessica	Martinez	230.00
9	409	Christopher	Rodriguez	200.00
10	410	Amanda	Taylor	240.00

Query executed successfully.

7.

```
---prescriptions has each doctor issued
select d.doctor_id, p.f_name, p.l_name, count(pr.p_id) as prescription_count
from pharmacy_doctor d join pharmacy_person p
on d.person_id = p.person_id join patient_prescription pr
on d.doctor_id = pr.doctor_id
group by d.doctor_id, p.f_name, p.l_name;
```

100 %

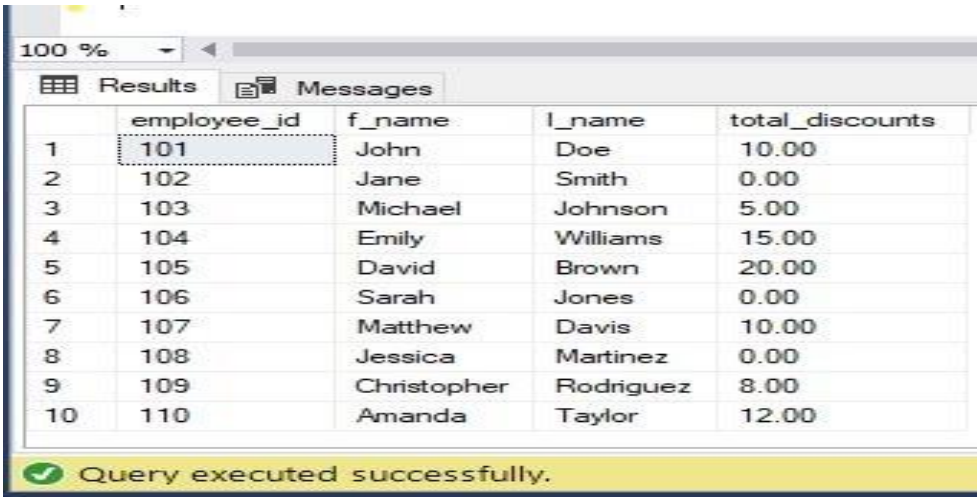
Results | Messages

	doctor_id	f_name	l_name	prescription_count
1	201	Jane	Smith	1
2	202	John	Doe	1
3	203	Emily	Williams	1
4	204	Michael	Johnson	1
5	205	Sarah	Jones	1
6	206	David	Brown	1
7	207	Jessica	Martinez	1
8	208	Matthew	Davis	1
9	209	Amanda	Taylor	1
10	210	Christopher	Rodriguez	1

Query executed successfully.

8.

```
---total amount of discounts given by each employee
select e.employee_id, p.f_name, p.l_name, sum(b.discounts) as total_discounts
from pharmacy_employee e join pharmacy_person p
on e.person_id = p.person_id join pharmacy_employee_order eo
on e.employee_id = eo.employee_id join patient_orders o
on eo.order_id = o.order_id join patient_bill b
on o.order_id = b.order_id
group by e.employee_id, p.f_name, p.l_name;
```



100 %

Results | Messages

	employee_id	f_name	l_name	total_discounts
1	101	John	Doe	10.00
2	102	Jane	Smith	0.00
3	103	Michael	Johnson	5.00
4	104	Emily	Williams	15.00
5	105	David	Brown	20.00
6	106	Sarah	Jones	0.00
7	107	Matthew	Davis	10.00
8	108	Jessica	Martinez	0.00
9	109	Christopher	Rodriguez	8.00
10	110	Amanda	Taylor	12.00

Query executed successfully.

9.

```
---list of all prescriptions along with the associated doctor's and patient's names
select pr.p_id, d.f_name as doctor_first_name, d.l_name as doctor_last_name, p.f_name as
patient_first_name, p.l_name as patient_last_name
from patient_prescription pr join pharmacy_doctor pd
on pr.doctor_id = pd.doctor_id join pharmacy_person d
on pd.person_id = d.person_id join patient_patient pp
on pr.patient_id = pp.patient_id join pharmacy_person p
on pp.person_id = p.person_id;
```

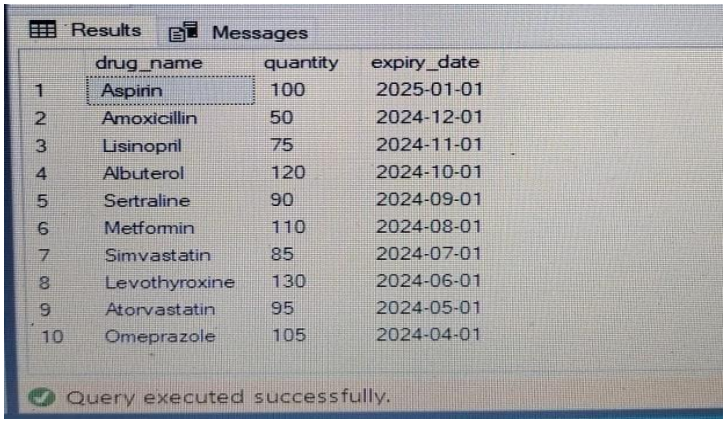
Results | Messages

	p_id	doctor_first_name	doctor_last_name	patient_first_name	patient_last_name
1	601	John	Doe	John	Doe
2	602	Jane	Smith	Jane	Smith
3	603	Michael	Johnson	Michael	Johnson
4	604	Emily	Williams	Emily	Williams
5	605	David	Brown	David	Brown
6	606	Sarah	Jones	Sarah	Jones
7	607	Matthew	Davis	Matthew	Davis
8	608	Jessica	Martinez	Jessica	Martinez
9	609	Christopher	Rodriguez	Christopher	Rodriguez
10	610	Amanda	Taylor	Amanda	Taylor

Query executed successfully. DESKTOP-A4B680R

10.

```
---list all drugs in the inventory along with their quantities and expiry dates?
select d.drug_name, i.quantity, i.expiry_date
from pharmacy_drug d join pharmacy_inventory_drug id
on d.drug_id = id.drug_id join pharmacy_inventory i
on id.inventory_id = i.inventory_id;
```



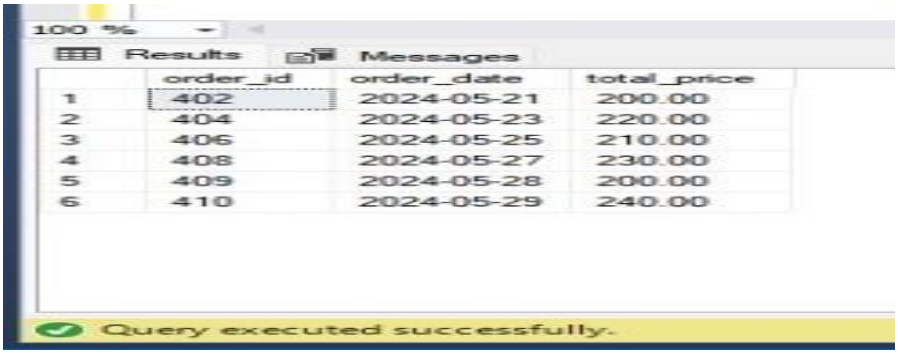
Results | Messages

	drug_name	quantity	expiry_date
1	Aspirin	100	2025-01-01
2	Amoxicillin	50	2024-12-01
3	Lisinopril	75	2024-11-01
4	Albuterol	120	2024-10-01
5	Sertraline	90	2024-09-01
6	Metformin	110	2024-08-01
7	Simvastatin	85	2024-07-01
8	Levothyroxine	130	2024-06-01
9	Atorvastatin	95	2024-05-01
10	Omeprazole	105	2024-04-01

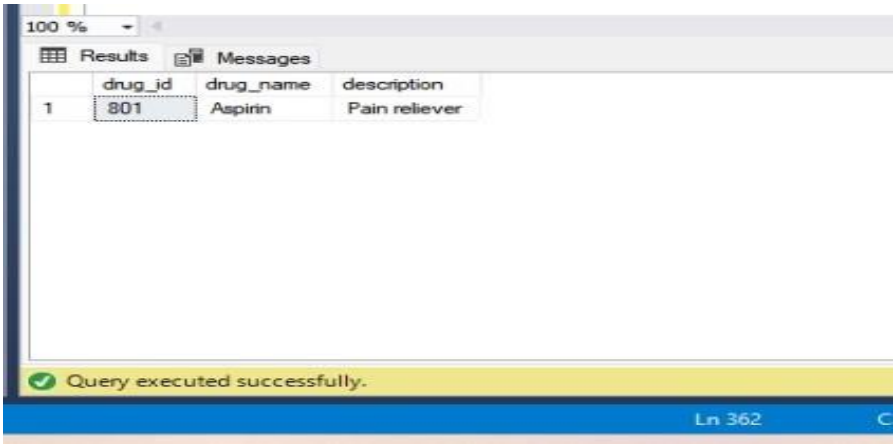
Query executed successfully.

- ## Nested Query

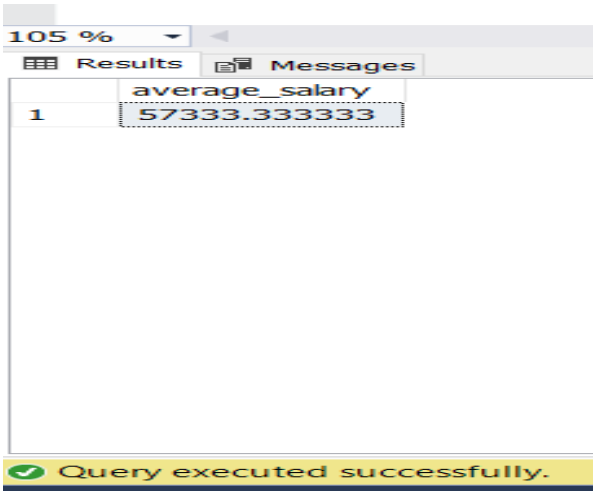
```
--all orders where the total price is above the average total price of all orders
select po.order_id, po.order_date, po.total_price
from patient_orders po
where po.total_price > (select avg (total_price) from patient_orders);
```




```
-- Get details of drugs supplied by 'PharmaCorp'
select pd.drug_id, pd.drug_name, pd.description
from pharmacy_drug pd
where pd.drug_id in (select  drug_id
                     from pharmacy_drug_company
                     where company_name = 'PharmaCorp');
```

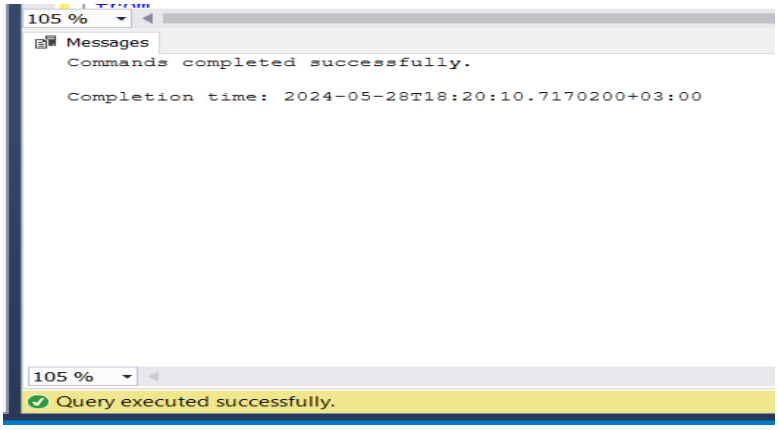


```
-- Get the average salary of employees hired after January 1, 2023
Select AVG(sub.salary) as average_salary
From
    (Select salary
     from pharmacy_employee
                where date_of_hiring > '2023-01-01') as sub;
```

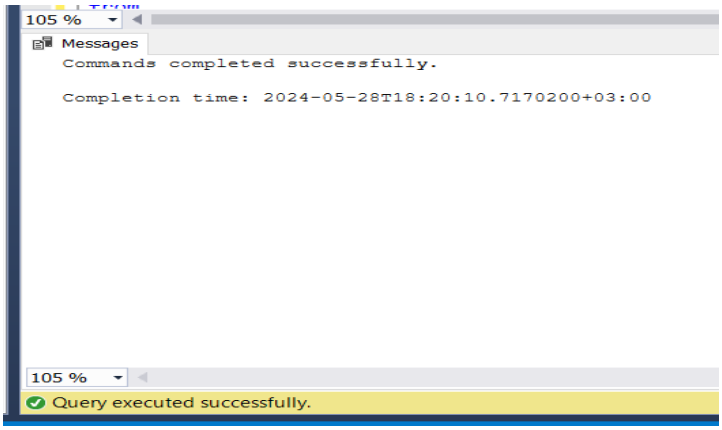


- **View**

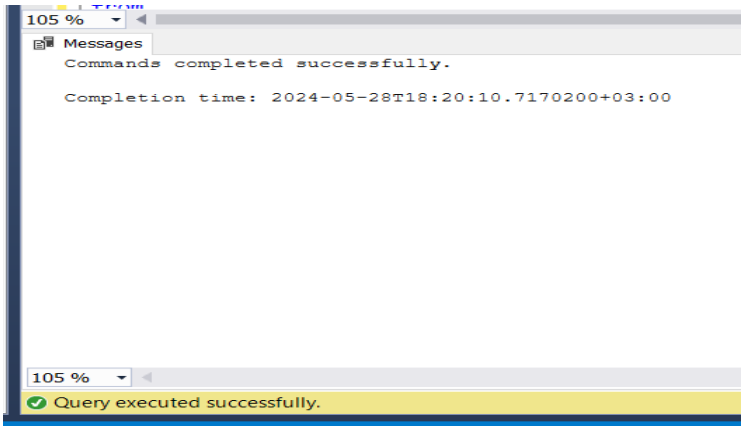
```
---View for drug information with company details
CREATE VIEW pharmacy.drug_info AS
SELECT d.drug_id,d.drug_name,d.description,d.unit_price,c.company_name
FROM [pharmacy_drug] d left join
    pharmacy_drug_company c ON d.drug_id = c.drug_id;
```



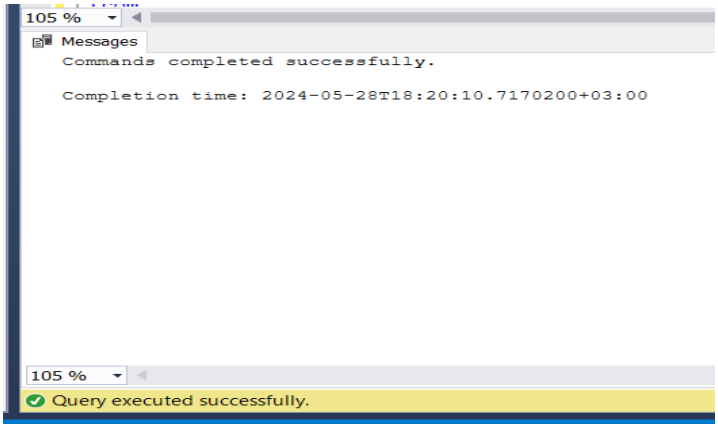
```
---View for prescription details including patient and doctor information
       CREATE VIEW patient.prescription_details AS
SELECT pr.p_id,pr.ssn,pr.p_date,p.f_name AS patient_first_name,p.l_name AS
patient_last_name,d.f_name AS doctor_first_name,d.l_name AS
doctor_last_name,doc.specialization
FROM [dbo].[patient_prescription] pr
JOIN  patient_patient pt ON pr.patient_id = pt.patient_id
JOIN pharmacy_person p ON pt.person_id = p.person_id
JOIN pharmacy_doctor doc ON pr.doctor_id = doc.doctor_id
              JOIN pharmacy_person d ON doc.person_id = d.person_id;
```




```
----View for inventory information including drug details
       CREATE VIEW pharmacy.inventory_info AS
SELECT  i.inventory_id,i.quantity, i.expiry_date, d.drug_name, d.unit_price
FROM  pharmacy_inventory i
JOIN
    pharmacy_inventory_drug id ON i.inventory_id = id.inventory_id
JOIN
                  pharmacy_drug d ON id.drug_id = d.drug_id;
```

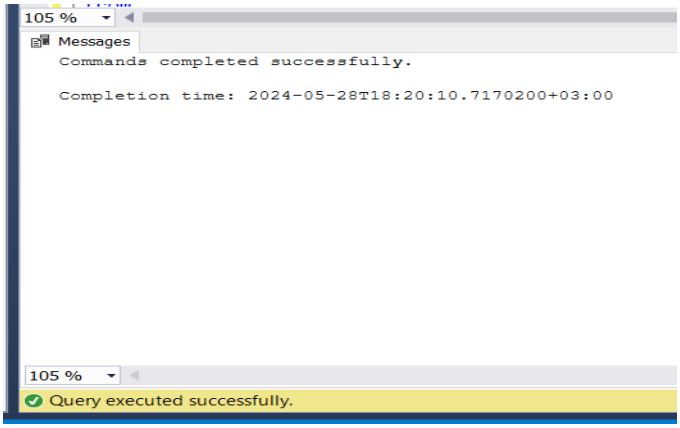


```
----View for supplier details including personal details and product offerings
       CREATE VIEW pharmacy.supplier_info AS
SELECT s.supplier_id,p.f_name,p.l_name,p.email,pa.address,pp.phone, s.product_offerings
FROM
 pharmacy_supplier s JOIN   pharmacy_person p ON s.person_id = p.person_id
LEFT JOIN pharmacy_person_address pa ON p.person_id = pa.person_id
LEFT JOIN  pharmacy_person_phone pp ON p.person_id = pp.person_id;
```

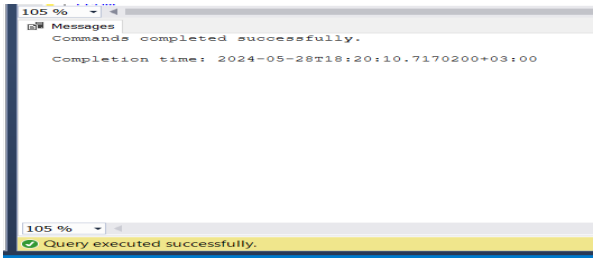


- **Stored procedures**

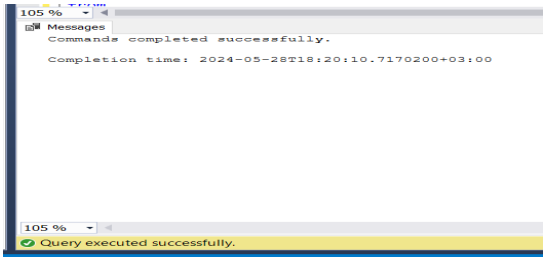
```
---Add New Pharmacy Person

    CREATE PROCEDURE AddPharmacyPerson
        @FirstName VARCHAR(50),
        @LastName VARCHAR(50),
        @Email VARCHAR(100)
    AS
    BEGIN
        INSERT INTO pharmacy.pharmacy_person (f_name, l_name, email)
    VALUES (@FirstName, @LastName, @Email);
    END;
```



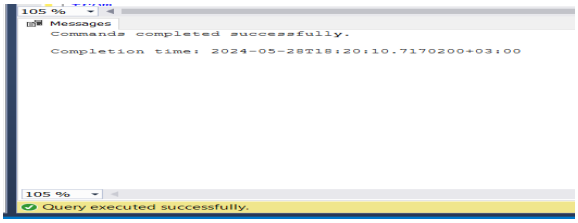
```
---Retrieve Doctor Details by Specialization
 CREATE PROCEDURE uspGetDoctorsBySpecialization
    @specialization VARCHAR(100)
AS
BEGIN
    SELECT d.doctor_id, p.f_name, p.l_name, d.salary
    FROM pharmacy.pharmacy_doctor d JOIN pharmacy.pharmacy_person p ON d.person_id =
p.person_id
    WHERE   d.specialization = @specialization;

END
```




```
--Update Employee Salary
CREATE PROCEDURE uspUpdateEmployeeSalary
    @employee_id INT,
    @new_salary DECIMAL(10, 2)
AS
BEGIN
    UPDATE pharmacy.pharmacy_employee
    SET salary = @new_salary
    WHERE employee_id = @employee_id;
END;
```



```
--Retrieve Orders by Patient ID
CREATE PROCEDURE uspGetOrdersByPatient
    @patient_id INT
AS
BEGIN
    SELECT  o.order_id, o.order_date, o.total_price, o.status
    FROM patient.patient_orders o
    WHERE o.patient_id = @patient_id;
END;
```



- **Trigger** `--i need to auditing any add or delete  any drugs`

1.

```
create table auditing_drugs (
auditing_ID int identity(1,1) primary key ,
drug_name varchar(100),
change_date datetime default getdate() )

create trigger add_drugs
on [dbo].[pharmacy_drug]
after insert
as
begin
insert into auditing_drugs(drug_name, change_date)
select
    drug_name, getdate()
from
    inserted
end;
```

2.

```
create trigger delete_drugs
on [dbo].[pharmacy_drug]
after delete
as
```

```
begin
insert into auditing_drugs(drug_name,change_date)
select
drug_name,getdate()
from
deleted
end;

INSERT INTO pharmacy_drug (drug_id, drug_name, description, unit_price)
VALUES
 (850, 'Aspirin', 'hosna', 111.00)
```