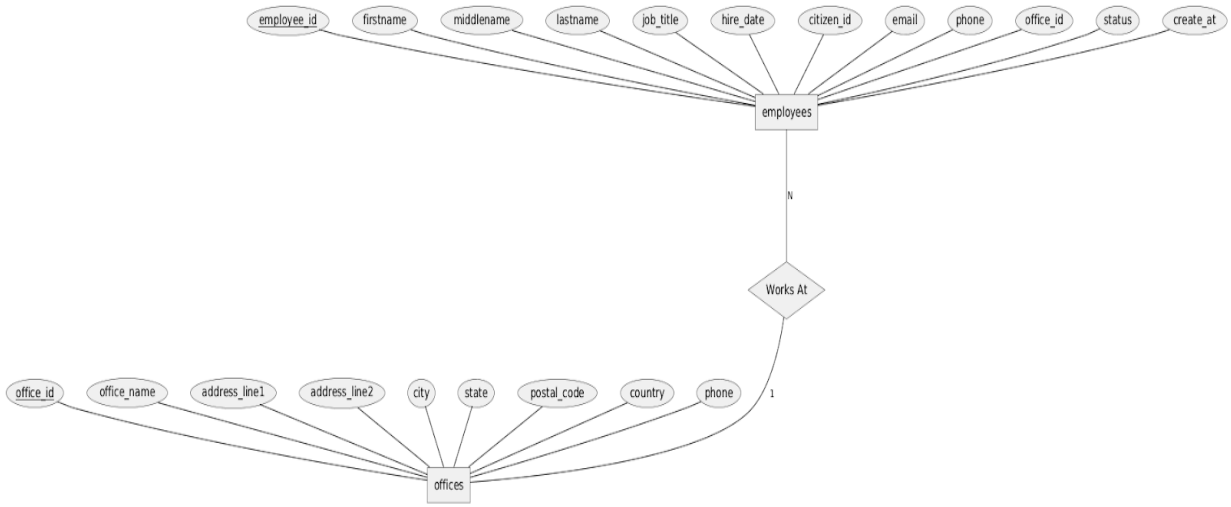
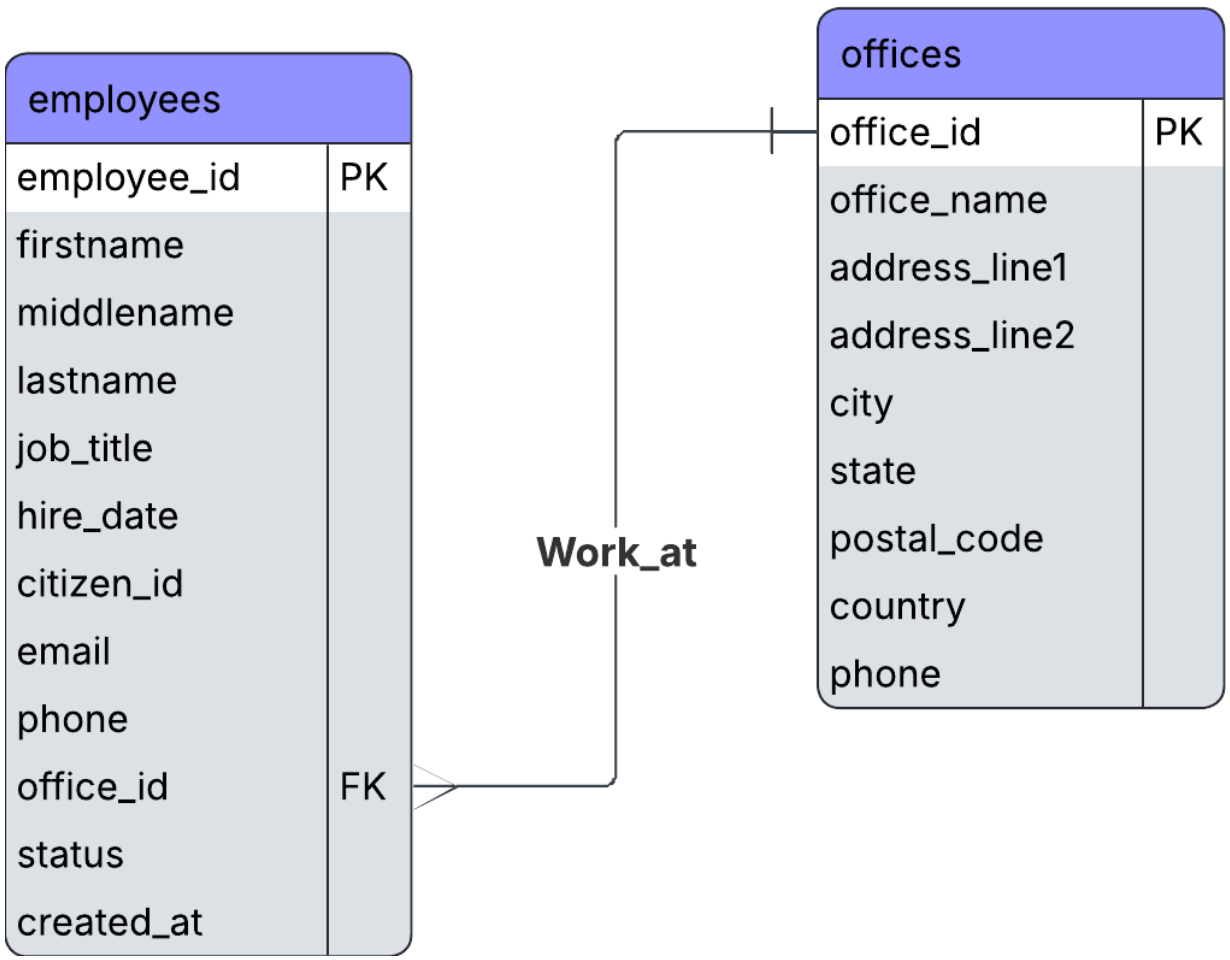
Problem Session #03 ER Diagram Exercise and Modify Existing Database

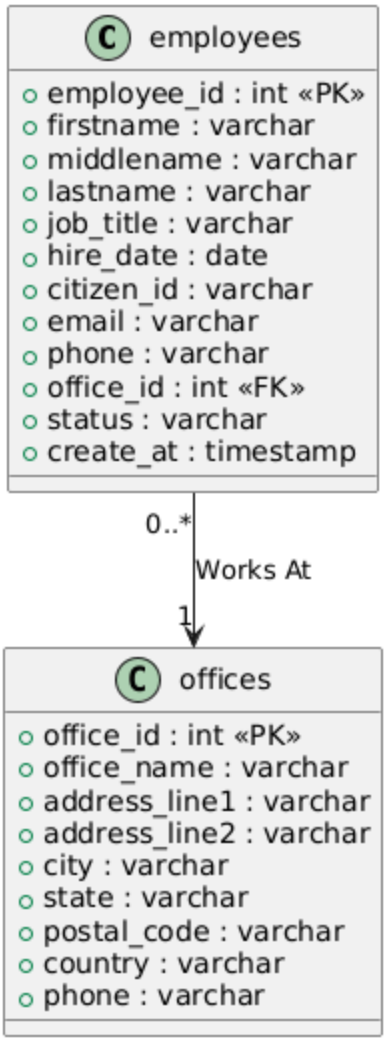
สมาชิกกลุ่ม	
กันต์ธีร์ ดวงมณี	67070501003
นัธทวัฒน์ ปริมสิริคุณาวุฒิ	67070501027
ภานุวัฒน์ บุญศักดิ์	67070501034
รัตนนันทน์ สิริพลวัฒน์	67070501038
วิศิษฐ์ สุวรรณเนาว์	67070501042
ศุภวิชญ์ มารยาท	67070501045
พลวริษฐ์ วัฒนเหมรัตน์	67070501067

Task 1: ER Diagram Exercise

01: A law firm wants to keep employees and their offices.
1. employees(employee_id, firstname, middlename, lastname, job_title, hire_date, citizen_id, email, phone, office_id, status, create_at) 2. offices(office_id, office_name, address_line1, address_line2, city, state, postal_code, country, phone)
   https://drive.google.com/file/d/1dSbx4mv7PEiHi9yLp7nGiUq6Qy-2Ey73/view?usp=sharing รูปภาพแสดง Chen ERD ข้อ 1



รูปภาพแสดง Crow’s Foot ERD ข้อ 1



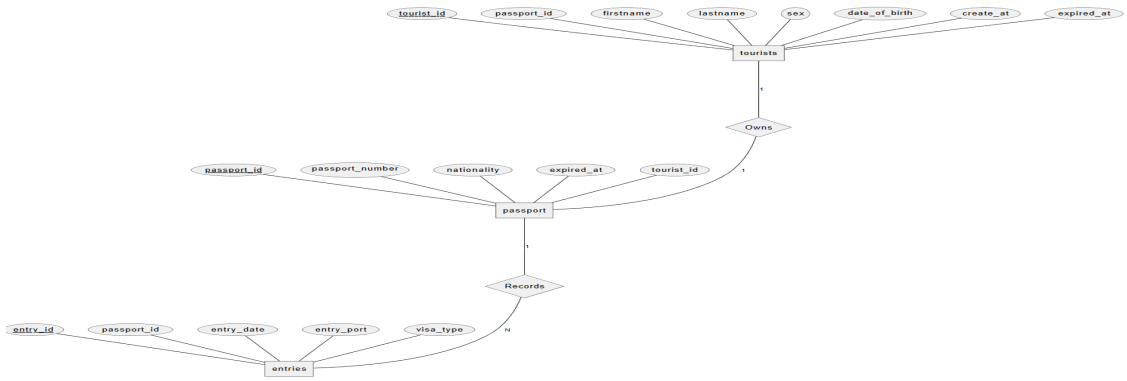
รูปภาพแสดง UML Class Diagram ข้อ 1

```sql
-- Create offices table first (referenced by employees)
CREATE TABLE offices (
    office_id INT NOT NULL,
    office_name VARCHAR(100) NOT NULL,
    address_line1 VARCHAR(200),
    address_line2 VARCHAR(200),
    city VARCHAR(100),
    state VARCHAR(100),
    postal_code VARCHAR(20),
    country VARCHAR(100),
    phone VARCHAR(20),
    PRIMARY KEY (office_id)
);
```

```sql
-- Create employees table with foreign key reference
CREATE TABLE employees (
    employee_id INT NOT NULL,
    firstname VARCHAR(50),
    middlename VARCHAR(50),
    lastname VARCHAR(50),
    job_title VARCHAR(100),
    hire_date DATE,
    citizen_id VARCHAR(50),
    email VARCHAR(100),
    phone VARCHAR(20),
    office_id INT,
    status VARCHAR(50),
    create_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (employee_id),
    FOREIGN KEY (office_id) REFERENCES offices(office_id)
);
```

02: Thai customs want to keep tourists and their passports that they used to enter Thailand.

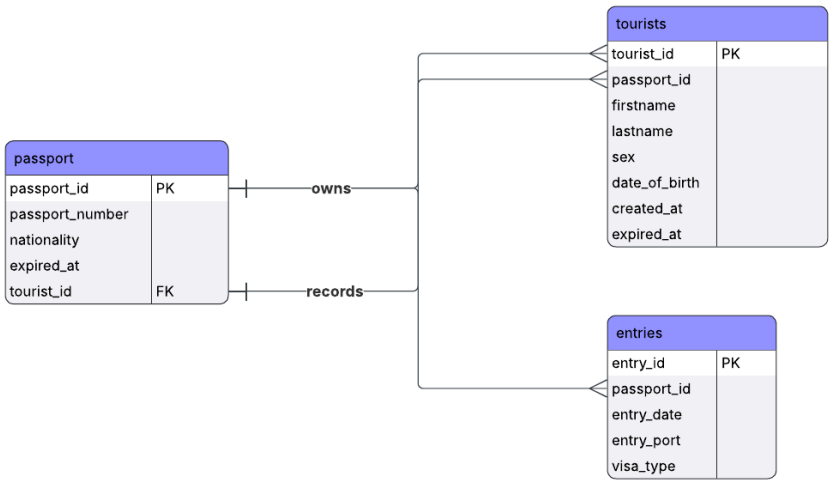
1. tourists(passport_id, firstname, lastname, sex, date_of_birth, create_at, expired_at)
2. passport(passport_id,passport_number, nationality,expired_at,tourist_id)
3. entries(entry_id,passport_id,entry_date,entry_port,visa_type)



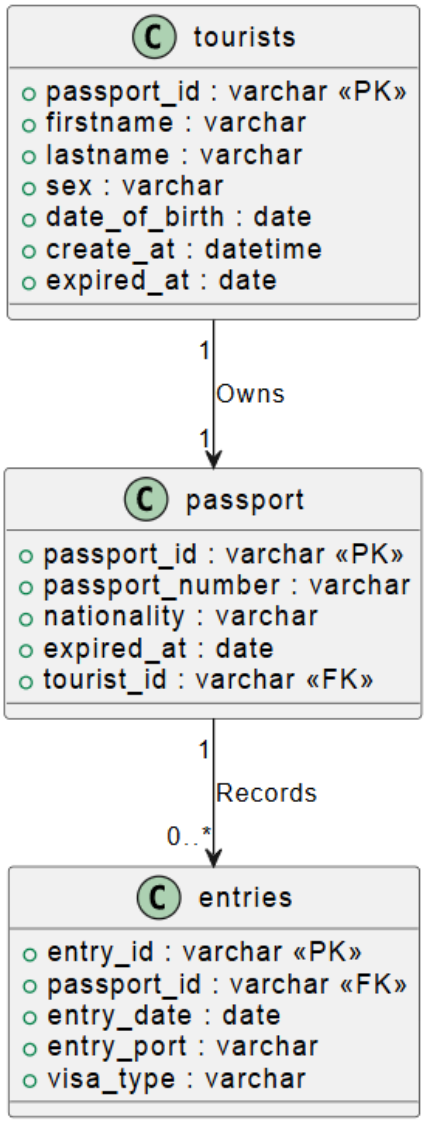
รูปภาพแสดง Chen ERD ข้อ 2

https://drive.google.com/file/d/1G_9vTVSL1rD2jUM61nEu0GsqZq9XxFmn/view?usp=sharing

URL Chen ERD ข้อ 2



รูปภาพแสดง Crow's Foot ERD ข้อ 2



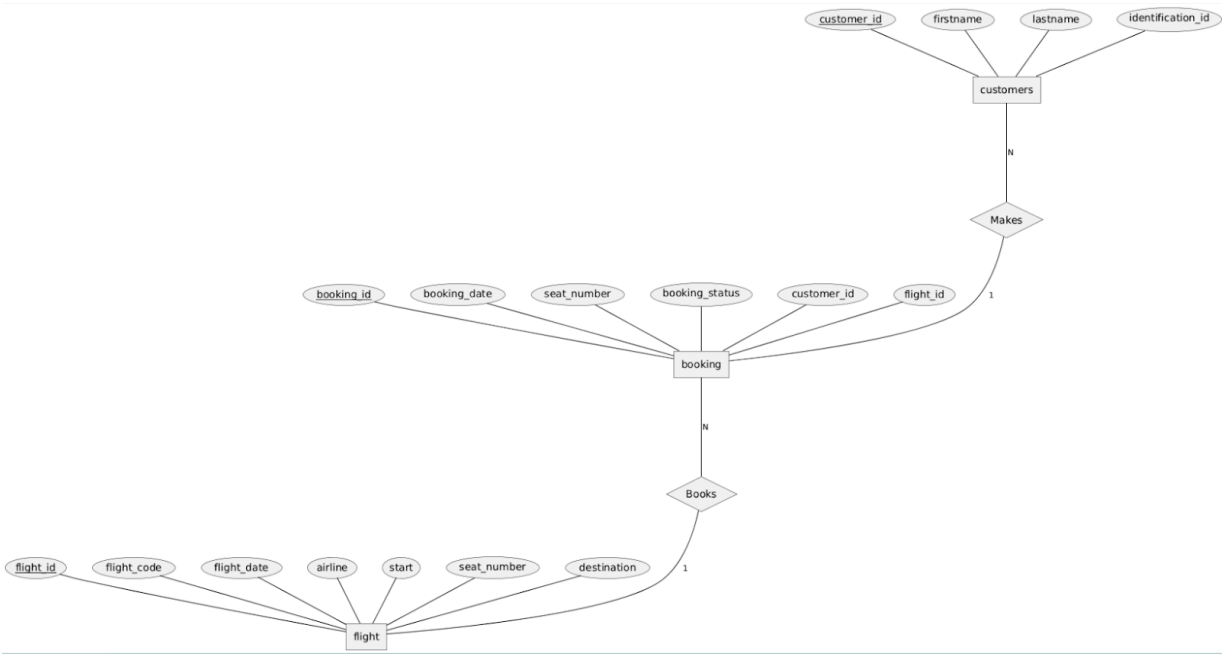
รูปภาพแสดง UML Class Diagram ข้อ 2

```sql
-- Table for storing tourist information
CREATE TABLE tourists (
  tourist_id VARCHAR(20) NOT NULL,
  firstname VARCHAR(100),
  lastname VARCHAR(100),
  sex VARCHAR(10),
  date_of_birth DATE,
```

```
  create_at DATETIME,
  expired_at DATE,
  PRIMARY KEY (tourist_id)
);

-- Table for storing passport information
CREATE TABLE passport (
  passport_id VARCHAR(20) NOT NULL,
  passport_number VARCHAR(50),
  nationality VARCHAR(50),
  expired_at DATE,
  tourist_id VARCHAR(20) NOT NULL,
  PRIMARY KEY (passport_id),
  FOREIGN KEY (tourist_id) REFERENCES tourists(tourist_id)
);

-- Table for storing entry information
CREATE TABLE entries (
  entry_id VARCHAR(20) NOT NULL,
  passport_id VARCHAR(20) NOT NULL,
  entry_date DATE,
  entry_port VARCHAR(100),
  visa_type VARCHAR(50),
  PRIMARY KEY (entry_id),
  FOREIGN KEY (passport_id) REFERENCES passport(passport_id)
);
```

03: The airline wants to retain its customer base and customer's flight booking.

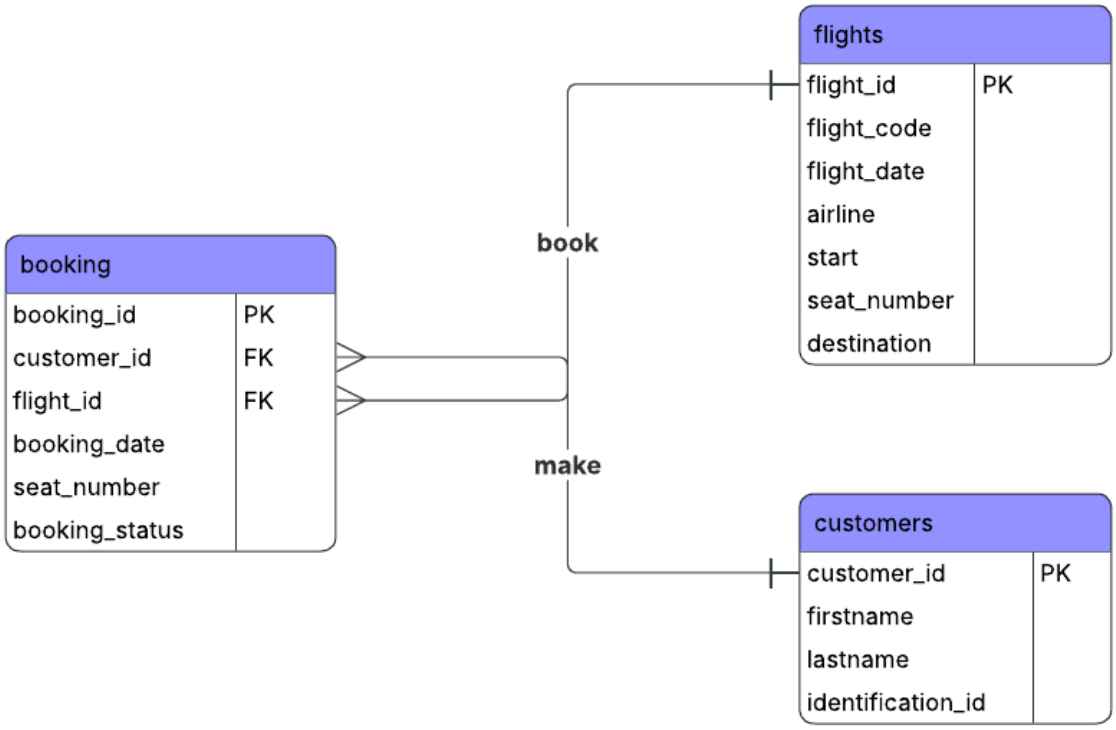
1. customers(customer_id, firstname, lastname, identification_id)
2. flight(flight_id, flight_code,flight_ date, airline, start,seat_number, destination)
3. booking(booking_id,customer_id,flight_id,booking_date,seat_number,booking_status)



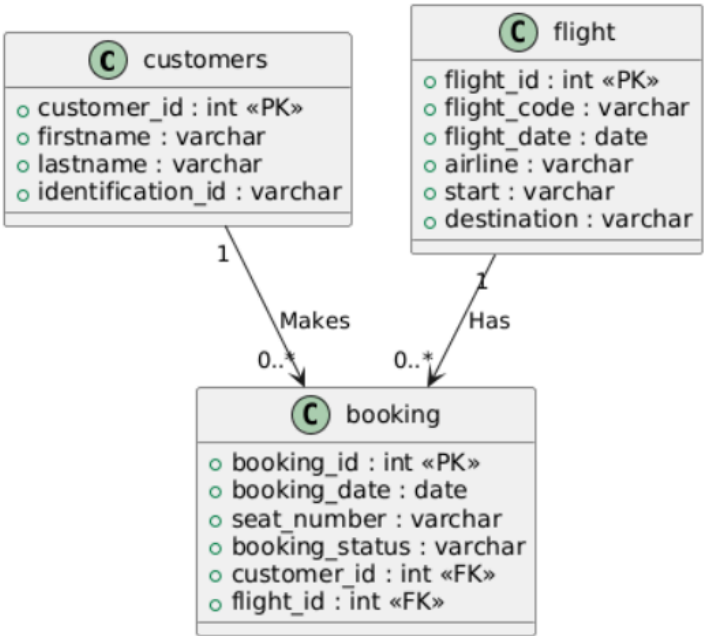
รูปภาพแสดง Chen ERD ข้อ 3

https://drive.google.com/file/d/1ZvdaYzcvM791PwB6gP8aEmCYbGYk4JOR/view?usp=sharing

URL Chen ERD ข้อ 3



รูปภาพแสดง Crow's Foot ERD ข้อ 3

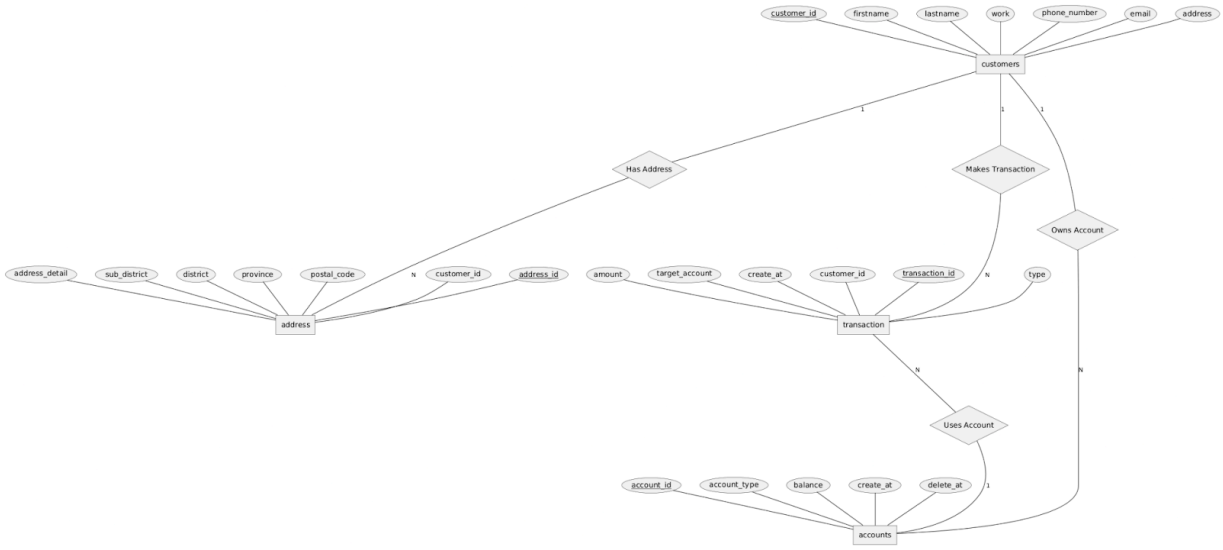


รูปภาพแสดง UML Class Diagram ข้อ 3

```sql
-- Table for storing customer information
CREATE TABLE customers (
  customer_id INT AUTO_INCREMENT PRIMARY KEY,
  firstname VARCHAR(100),
  lastname VARCHAR(100),
  identification_id VARCHAR(50)
);

-- Table for storing flight information
CREATE TABLE flight (
  flight_id INT AUTO_INCREMENT PRIMARY KEY,
  flight_code VARCHAR(20) NOT NULL,
  flight_date DATE NOT NULL,
  airline VARCHAR(100),
  start VARCHAR(100),
  destination VARCHAR(100)
);

-- Table for storing booking information
CREATE TABLE booking (
  booking_id INT AUTO_INCREMENT PRIMARY KEY,
  booking_date DATE NOT NULL,
  seat_number VARCHAR(10),
  booking_status VARCHAR(50),
  customer_id INT NOT NULL,
  flight_id INT NOT NULL,
  FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
  FOREIGN KEY (flight_id) REFERENCES flight(flight_id)
);
```

04: A bank wants to keep customers and their accounts.

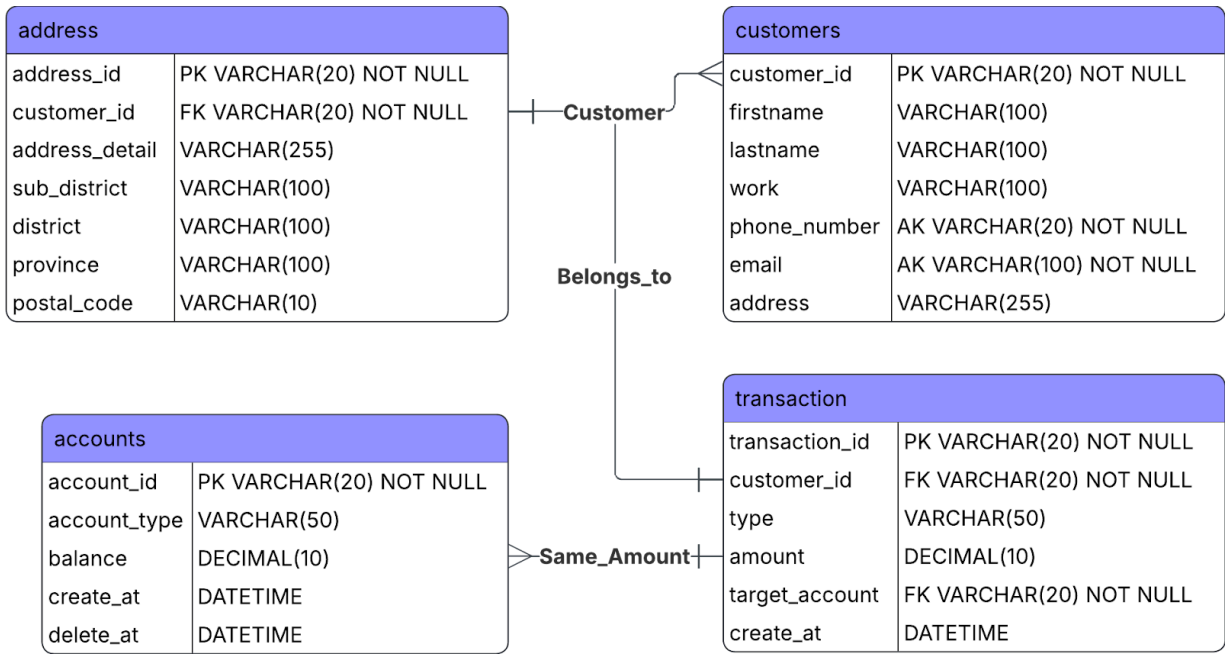
1. customers(customer_id,firstname,lastname,work,phone_number,email,address)
2. accounts(account_id, account_type, balance, create_at, delete_at)
3. address(address_id, customer_id, address_detail, sub_district, district, province, postal_code)
4. transaction(transaction_id, customer_id, type, amount, target_account, create_at)



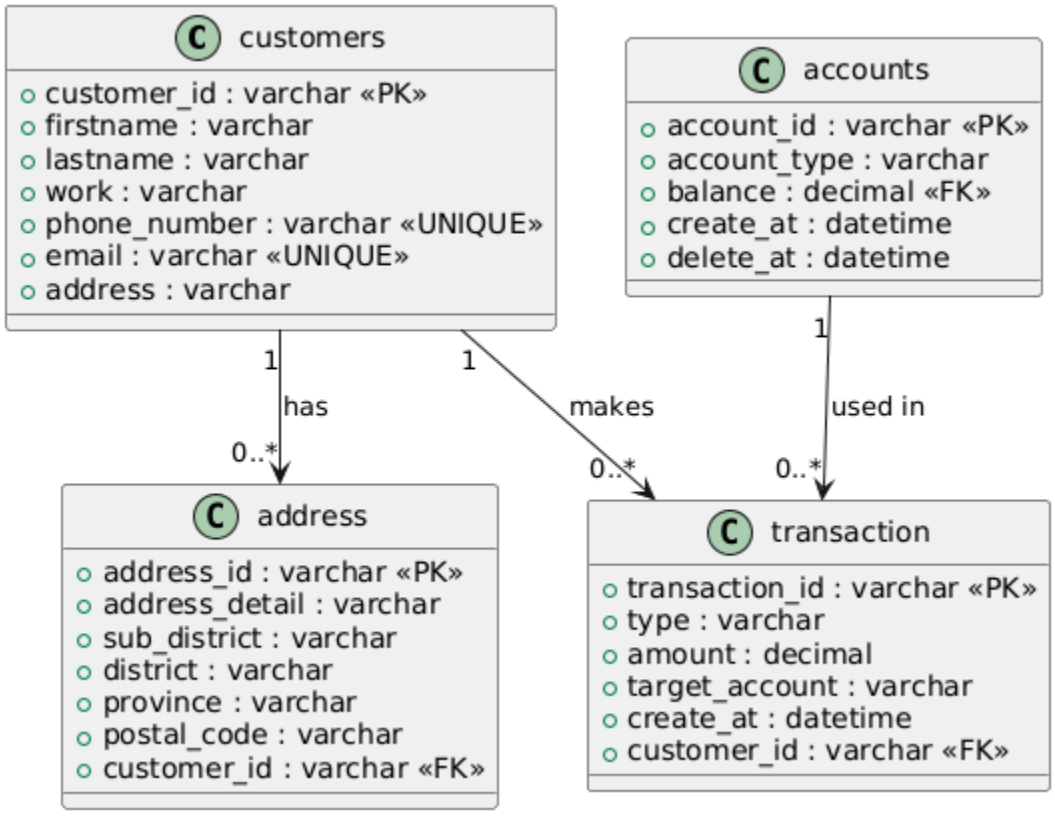
รูปภาพแสดง Chen ERD ข้อ 4

https://drive.google.com/file/d/1ioYrnRaNV47BWde3Ae1thFabZpwqn7co/view?usp=sharing

URL Chen ERD ข้อ 4



รูปภาพแสดง Crow's Foot ERD ข้อ 4



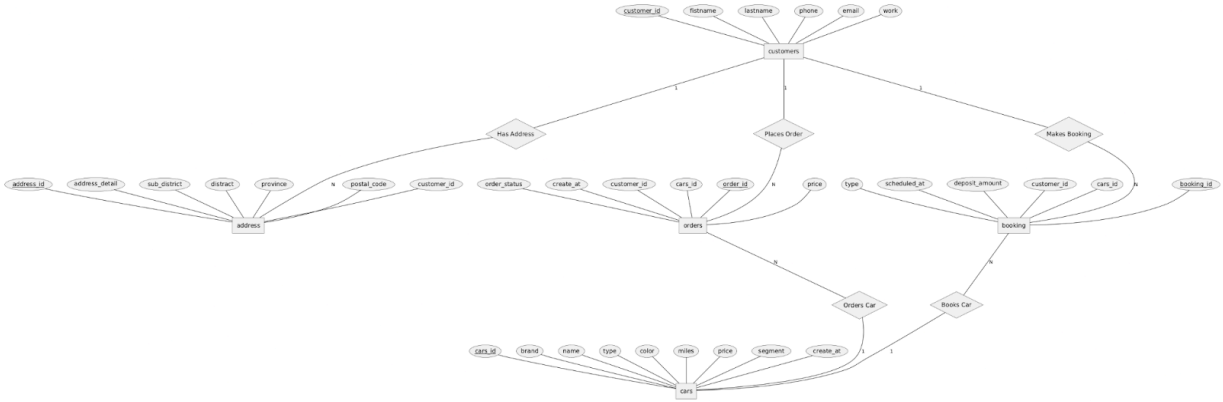
รูปภาพแสดง UML Class Diagram ข้อ 4

```sql
-- Table for storing address information
CREATE TABLE address (
  address_id VARCHAR(20) NOT NULL,
  customer_id VARCHAR(20) NOT NULL,
  address_detail VARCHAR(255),
  sub_district VARCHAR(100),
  district VARCHAR(100),
  province VARCHAR(100),
  postal_code VARCHAR(10),
  PRIMARY KEY (address_id)
);

-- Table for storing transaction information
CREATE TABLE transaction (
  transaction_id VARCHAR(20) NOT NULL,
  customer_id VARCHAR(20) NOT NULL,
  type VARCHAR(50),
  amount DECIMAL(10),
  target_account VARCHAR(20) NOT NULL,
  create_at DATETIME,
  PRIMARY KEY (transaction_id)
);

-- Table for storing account information
CREATE TABLE accounts (
  account_id VARCHAR(20) NOT NULL,
  account_type VARCHAR(50),
  balance DECIMAL(10),
  create_at DATETIME,
  delete_at DATETIME,
  PRIMARY KEY (account_id),
  FOREIGN KEY (balance) REFERENCES transaction(amount)
);

-- Table for storing customer information
CREATE TABLE customers (
  customer_id VARCHAR(20) NOT NULL,
  firstname VARCHAR(100),
  lastname VARCHAR(100),
```

```
  work VARCHAR(100),
  phone_number VARCHAR(20) NOT NULL,
  email VARCHAR(100) NOT NULL,
  address VARCHAR(255),
  PRIMARY KEY (customer_id),
  UNIQUE (phone_number),
  UNIQUE (email),
  FOREIGN KEY (customer_id) REFERENCES address(customer_id)
);
```

05: A car dealer that wants to store cars, customers, and orders.

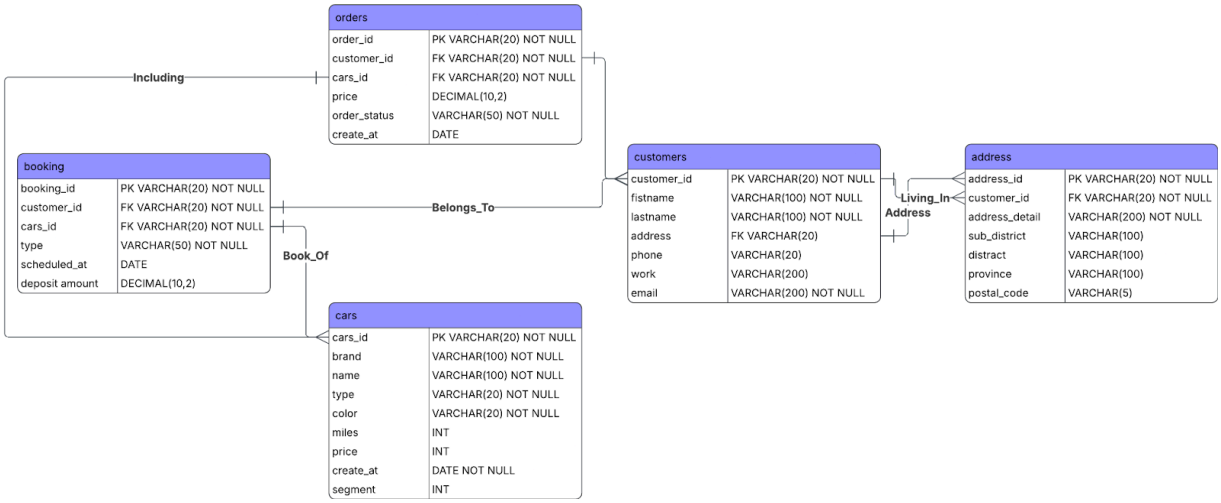
1. cars(cars_id, brand, name, type, color, miles, price, created_at, segment)
2. customers(customers_id, firstname, lastname, address, phone, work, email)
3. orders(order_id, customer_name, cars_name, price, order_status, created_at)
4. address(address_id, customer_id, address_detail, sub_district, district, province, postal_code)
5. booking(booking_id, type, scheduled_at, deposit_amount)



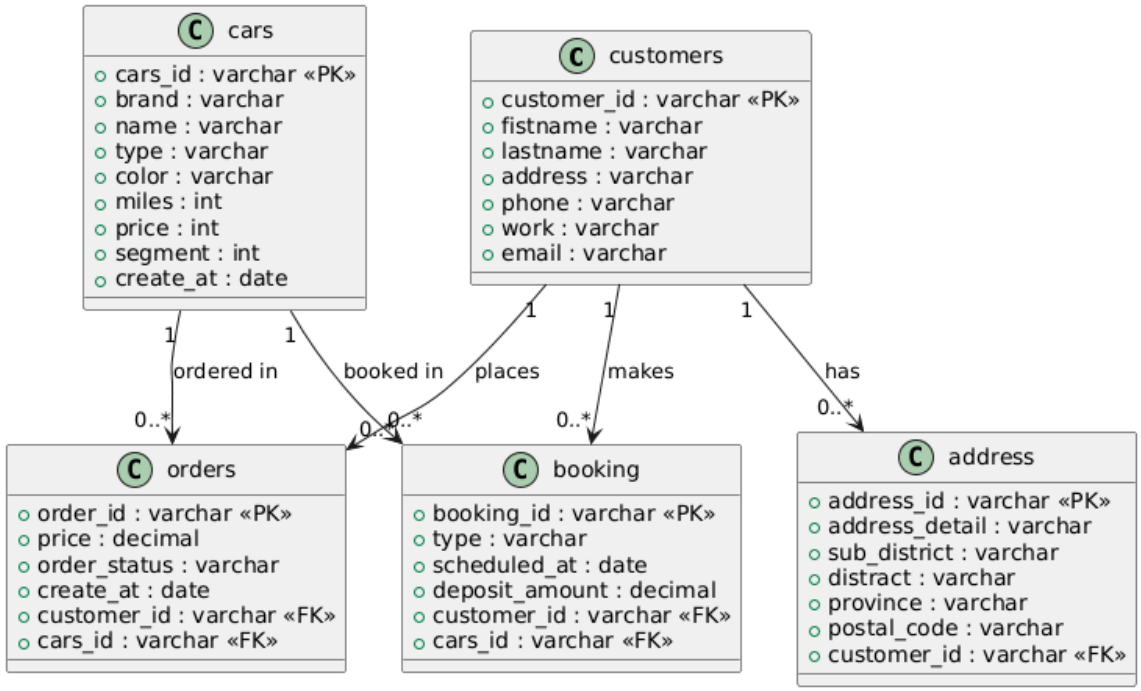
รูปภาพแสดง Chen ERD ข้อ 5

https://drive.google.com/file/d/114LNmM38zceftkndd8K1r7YWnEIagrk-/view?usp=sharing

URL Chen ERD ข้อ 5



รูปภาพแสดง Crow’s Foot ERD ข้อ 5



รูปภาพแสดง UML Class Diagram ข้อ 5

```sql
-- Table for storing booking information
CREATE TABLE booking (
  booking_id VARCHAR(20) NOT NULL,
  customer_id VARCHAR(20) NOT NULL,
  cars_id VARCHAR(20) NOT NULL,
  type VARCHAR(50) NOT NULL,
  scheduled_at DATE,
  deposit_amount DECIMAL(10,2),
  PRIMARY KEY (booking_id)
);

-- Table for storing order information
CREATE TABLE orders (
  order_id VARCHAR(20) NOT NULL,
  customer_id VARCHAR(20) NOT NULL,
  cars_id VARCHAR(20) NOT NULL,
  price DECIMAL(10,2),
  order_status VARCHAR(50) NOT NULL,
  create_at DATE,
  PRIMARY KEY (order_id)
);
```

```sql
-- Table for storing customer information
CREATE TABLE customers (
  customer_id VARCHAR(20) NOT NULL,
  fistname VARCHAR(100) NOT NULL,
  lastname VARCHAR(100) NOT NULL,
  address VARCHAR(20),
  phone VARCHAR(20),
  work VARCHAR(200),
  email VARCHAR(200) NOT NULL,
  PRIMARY KEY (customer_id),
  FOREIGN KEY (customer_id) REFERENCES booking(customer_id),
  FOREIGN KEY (customer_id) REFERENCES orders(customer_id)
);

-- Table for storing address information
CREATE TABLE address (
  address_id VARCHAR(20) NOT NULL,
  customer_id VARCHAR(20) NOT NULL,
  address_detail VARCHAR(200) NOT NULL,
  sub_district VARCHAR(100),
  distract VARCHAR(100),
  province VARCHAR(100),
  postal_code VARCHAR(5),
  PRIMARY KEY (address_id),
  FOREIGN KEY (address_id, customer_id)
    REFERENCES customers(address, customer_id)
);

-- Table for storing car information
CREATE TABLE cars (
  cars_id VARCHAR(20) NOT NULL,
  brand VARCHAR(100) NOT NULL,
  name VARCHAR(100) NOT NULL,
  type VARCHAR(20) NOT NULL,
  color VARCHAR(20) NOT NULL,
  miles INT,
  price INT,
  create_at DATE NOT NULL,
```

```
  segment INT,
  PRIMARY KEY (cars_id),
  FOREIGN KEY (cars_id) REFERENCES orders(cars_id),
  FOREIGN KEY (cars_id) REFERENCES booking(cars_id)
);
```

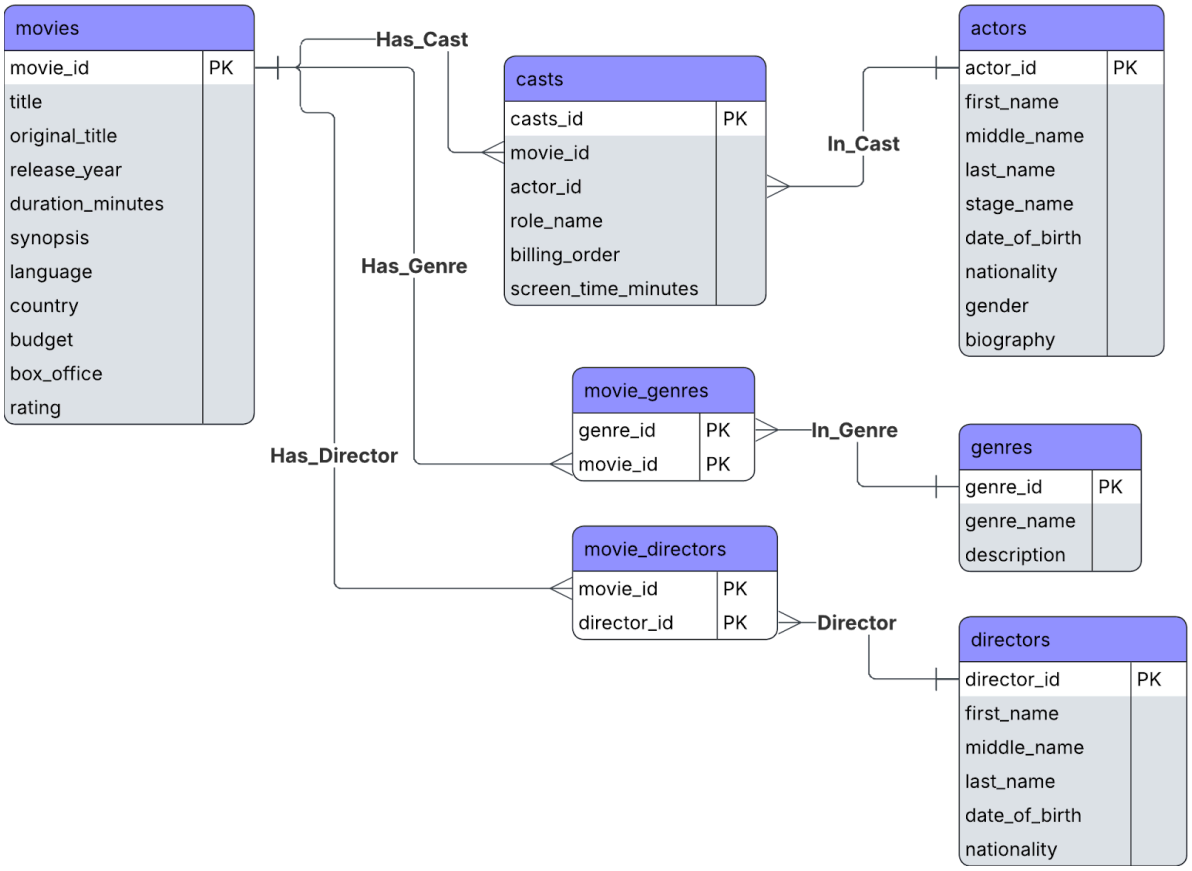
06: A film archive that wants to store movies, actors, and casts.

1. movies(movie_id, title, original_title, release_year, release_date, duration_minutes, synopsis, language, country, budget, box_office, rating)
2. actors(actor_id, first_name, last_name, stage_name, date_of_birth, nationality, gender, biography)
3. directors(director_id, first_name, last_name, date_of_birth, nationality
4. genres(genre_id, genre_name, description)
5. movie_genres(movie_id, genre_id)
6. casts(cast_id, movie_id, actor_id, role_name, billing_order, screen_time_minutes)
7. movie_directors(movie_id, director_id)

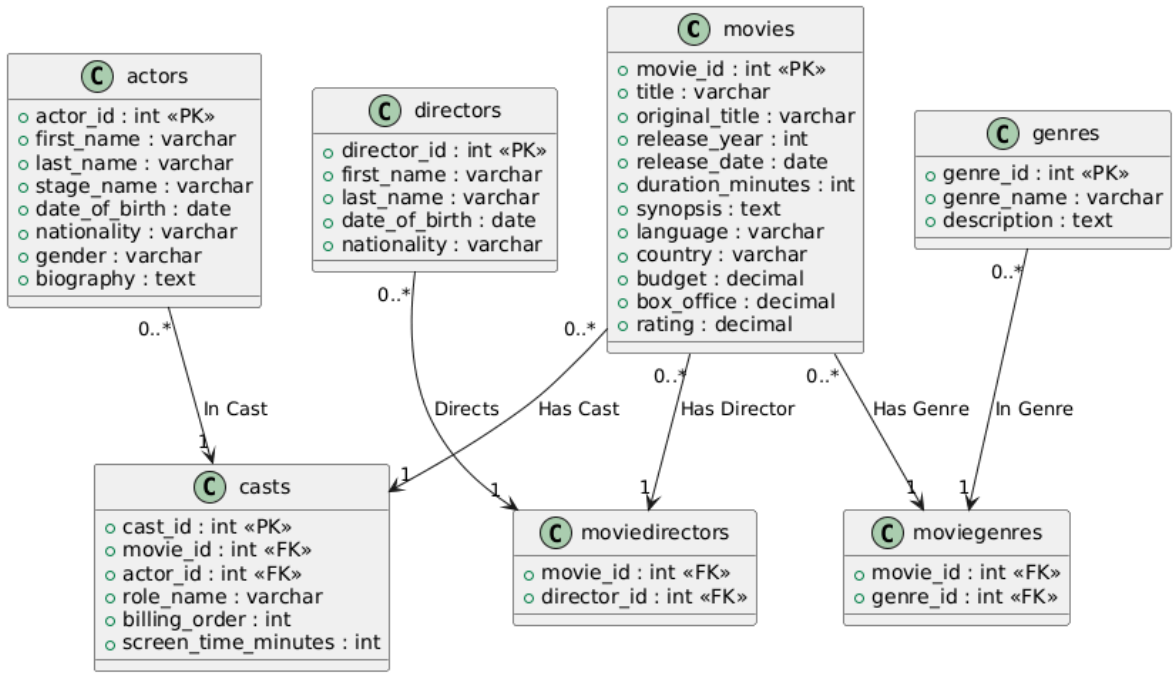
รูปภาพแสดง Chen ERD ข้อ 6

https://drive.google.com/file/d/1BdafTdnIehNmReh35XflDLpd-6YvvP8v/view?usp=sharing

URL Chen ERD ข้อ 6



รูปภาพแสดง Crow's Foot ERD ข้อ 6



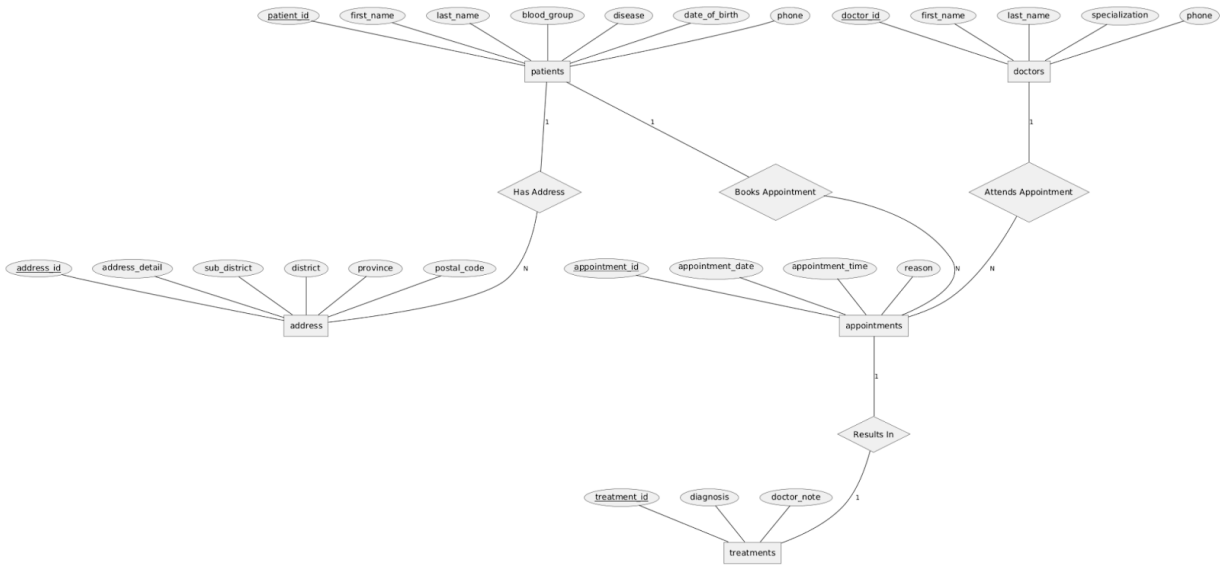
รูปภาพแสดง UML Class Diagram ข้อ 6

```sql
-- Create genres table first
CREATE TABLE genres (
    genre_id INT NOT NULL,
    genre_name VARCHAR(100) NOT NULL,
    description TEXT,
    PRIMARY KEY (genre_id)
);

-- Create actors table
CREATE TABLE actors (
    actor_id INT NOT NULL,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    stage_name VARCHAR(100),
    date_of_birth DATE,
    nationality VARCHAR(100),
    gender VARCHAR(20),
    biography TEXT,
    PRIMARY KEY (actor_id)
);

-- Create directors table
CREATE TABLE directors (
    director_id INT NOT NULL,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    date_of_birth DATE,
    nationality VARCHAR(100),
    PRIMARY KEY (director_id)
);

-- Create movies table
CREATE TABLE movies (
    movie_id INT NOT NULL,
    title VARCHAR(255) NOT NULL,
    original_title VARCHAR(255),
    release_year INT,
    release_date DATE,
    duration_minutes INT,
```

```
    synopsis TEXT,
    language VARCHAR(50),
    country VARCHAR(100),
    budget DECIMAL(15,2),
    box_office DECIMAL(15,2),
    rating DECIMAL(3,1),
    PRIMARY KEY (movie_id)
);

-- Create movie_genres junction table
CREATE TABLE movie_genres (
    movie_id INT NOT NULL,
    genre_id INT NOT NULL,
    PRIMARY KEY (movie_id, genre_id),
    FOREIGN KEY (movie_id) REFERENCES movies(movie_id),
    FOREIGN KEY (genre_id) REFERENCES genres(genre_id)
);

-- Create casts junction table
CREATE TABLE casts (
    cast_id INT NOT NULL,
    movie_id INT NOT NULL,
    actor_id INT NOT NULL,
    role_name VARCHAR(255),
    billing_order INT,
    screen_time_minutes INT,
    PRIMARY KEY (cast_id),
    FOREIGN KEY (movie_id) REFERENCES movies(movie_id),
    FOREIGN KEY (actor_id) REFERENCES actors(actor_id)
);

-- Create movie_directors junction table
CREATE TABLE movie_directors (
    movie_id INT NOT NULL,
    director_id INT NOT NULL,
    PRIMARY KEY (movie_id, director_id),
    FOREIGN KEY (movie_id) REFERENCES movies(movie_id),
    FOREIGN KEY (director_id) REFERENCES directors(director_id)
);
```

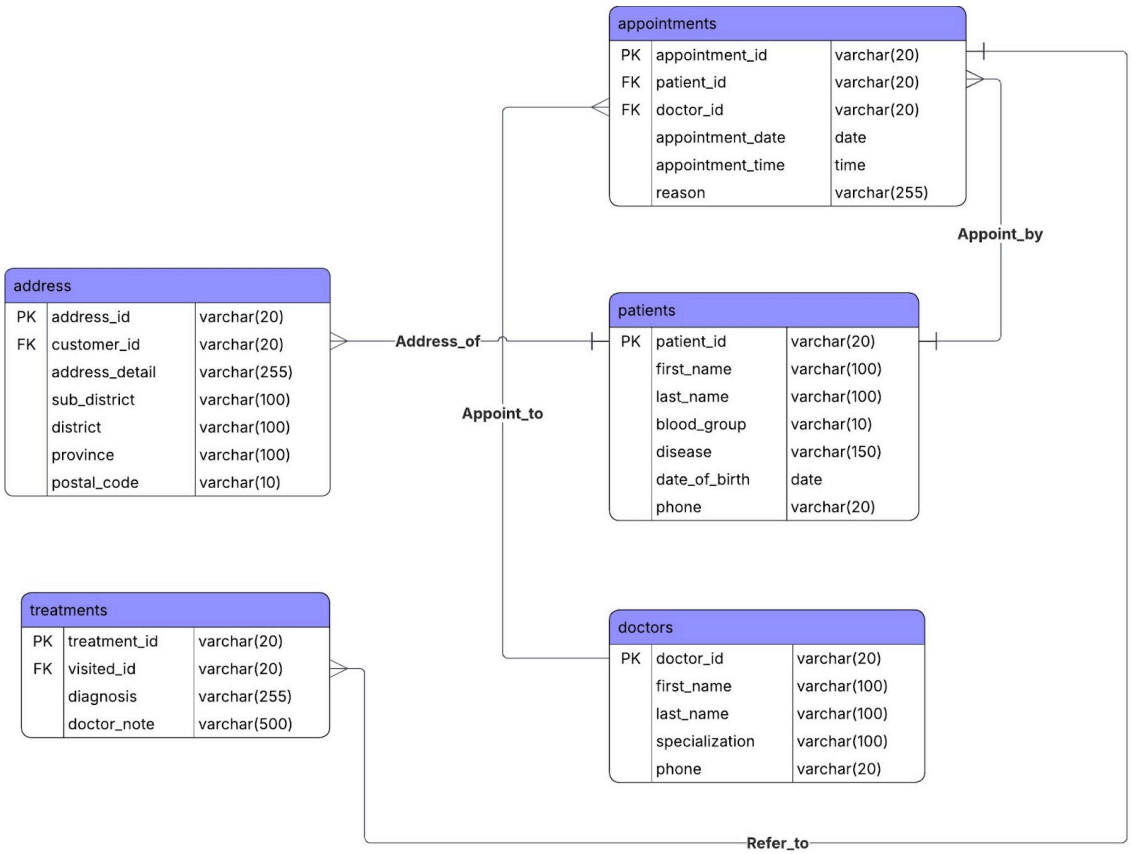
07: A hospital that wants to store patients, doctors, and appointments.

1. patients(patient_id, first_name, last_name,, blood_group, disease, date_of_birth, phone)
2. doctors(doctor_id, first_name, last_name, specialization, phone)
3. appointments(appointment_id, patient_id, doctor_id, appointment_date, appointment_time, reason)
4. address(address_id, customer_id, address_detail, sub_district, district, province, postal_code)
5. treatments(treatment_id, visited_id, diagnosis, doctor_note)

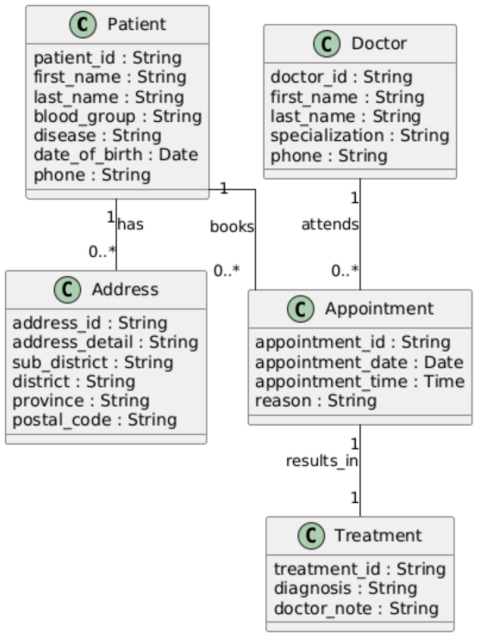


https://drive.google.com/file/d/16RsCwBD2TgcJIMdgh5NjGWhGBVj8XkQ6/view?usp=drive_link

รูปภาพแสดง Chen ERD ข้อ 7



รูปภาพแสดง Crow's Foot ERD ข้อ 7



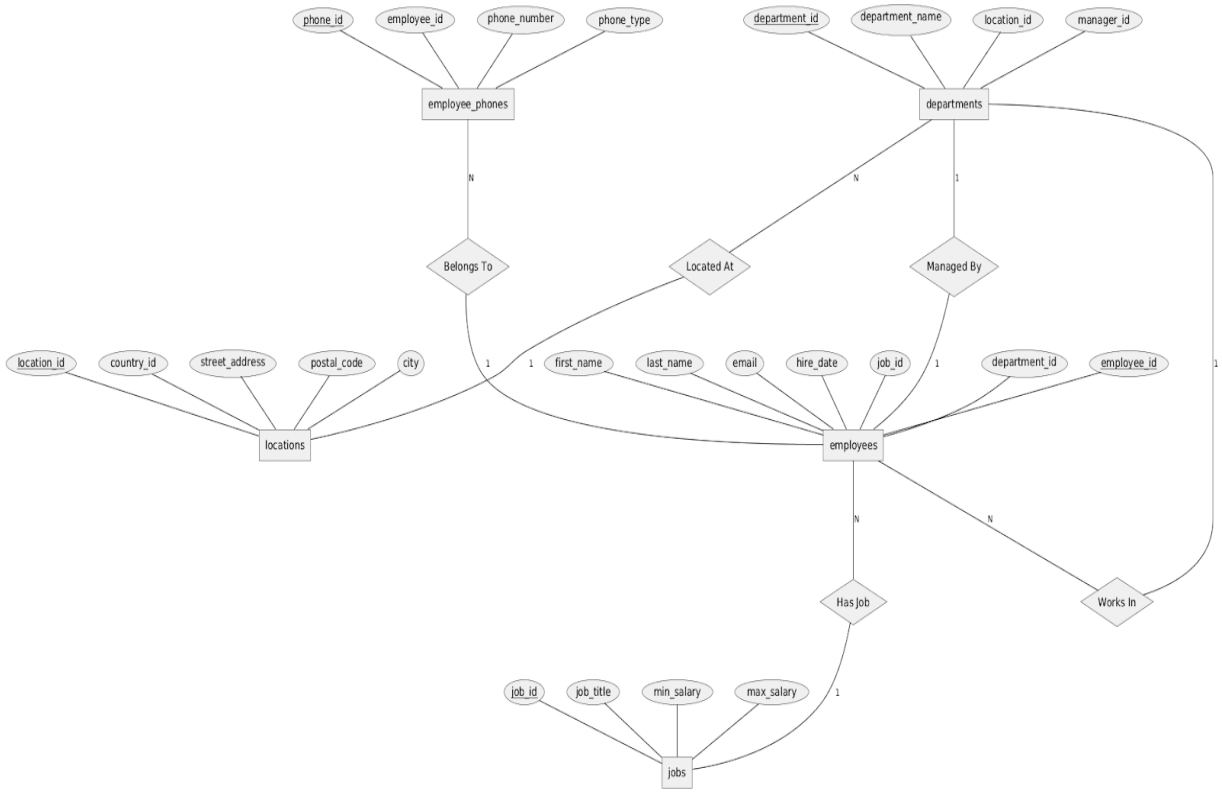
รูปภาพแสดง UML Class Diagram ข้อ 7

```sql
CREATE TABLE patients (
  patient_id VARCHAR(20) PRIMARY KEY,
  first_name VARCHAR(100),
  last_name VARCHAR(100),
  blood_group VARCHAR(10),
  disease VARCHAR(150),
  date_of_birth DATE,
  phone VARCHAR(20)
);

CREATE TABLE doctors (
  doctor_id VARCHAR(20) PRIMARY KEY,
  first_name VARCHAR(100),
  last_name VARCHAR(100),
  specialization VARCHAR(100),
  phone VARCHAR(20)
);

CREATE TABLE appointments (
  appointment_id VARCHAR(20) PRIMARY KEY,
  patient_id VARCHAR(20),
  doctor_id VARCHAR(20),
  appointment_date DATE,
  appointment_time TIME,
  reason VARCHAR(255)
);

CREATE TABLE treatments (
  treatment_id VARCHAR(20) PRIMARY KEY,
  visited_id VARCHAR(20),
  diagnosis VARCHAR(255),
  doctor_note VARCHAR(500)
);

CREATE TABLE address (
  address_id VARCHAR(20) PRIMARY KEY,
  customer_id VARCHAR(20),
  address_detail VARCHAR(255),
  sub_district VARCHAR(100),
```

```
  district VARCHAR(100),
  province VARCHAR(100),
  postal_code VARCHAR(10)
);
```

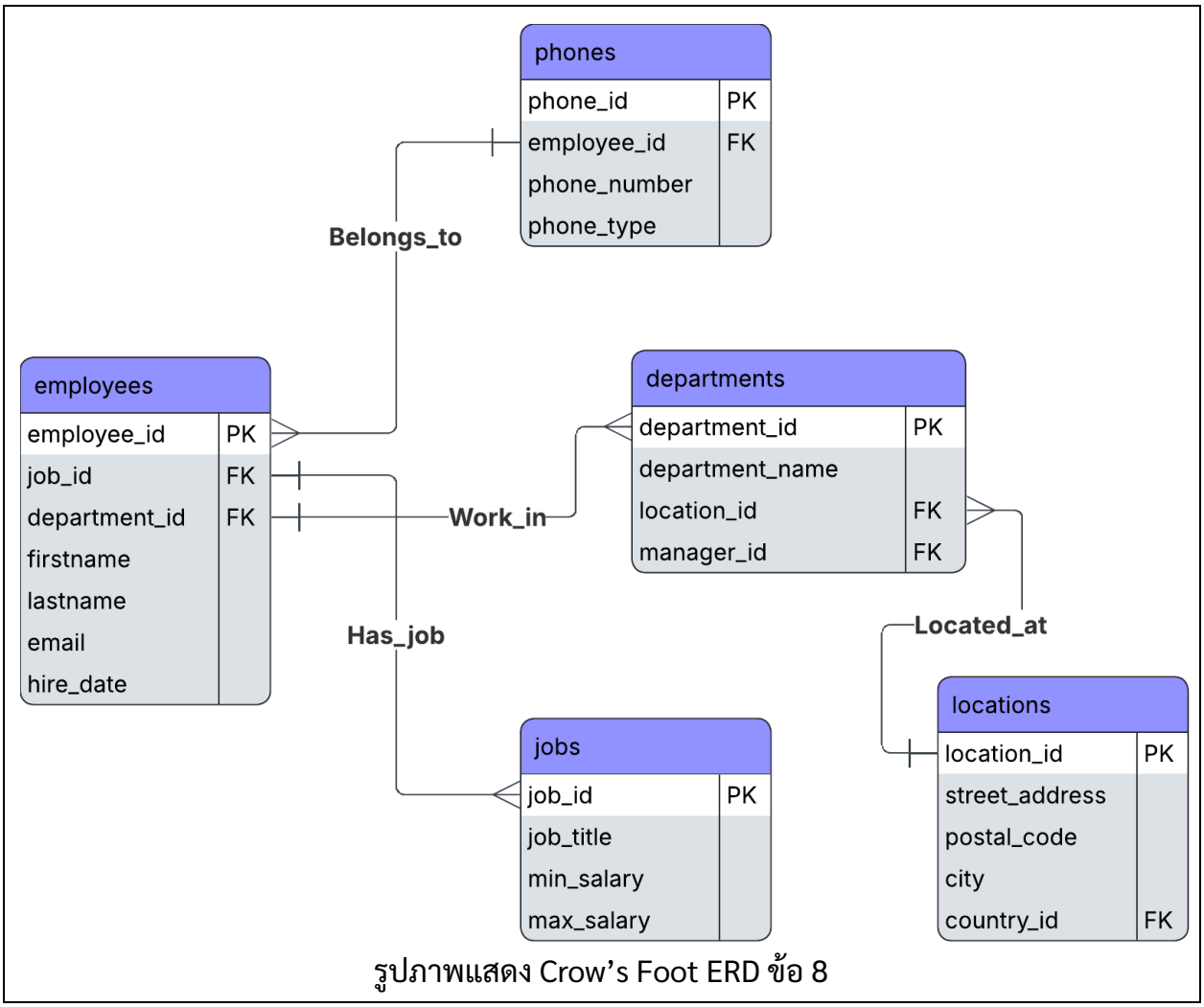
08: A company that wants to store departments, employees, employee phone numbers

6. locations (location_id, street_address, postal_code, city, country_id)
7. departments (department_id, department_name, location_id, manager_id)
8. jobs (job_id, job_title, min_salary, max_salary)
9. employees (employee_id, first_name, last_name, email, hire_date, job_id, department_id)
10. phones (phone_id, employee_id, phone_number, phone_type)

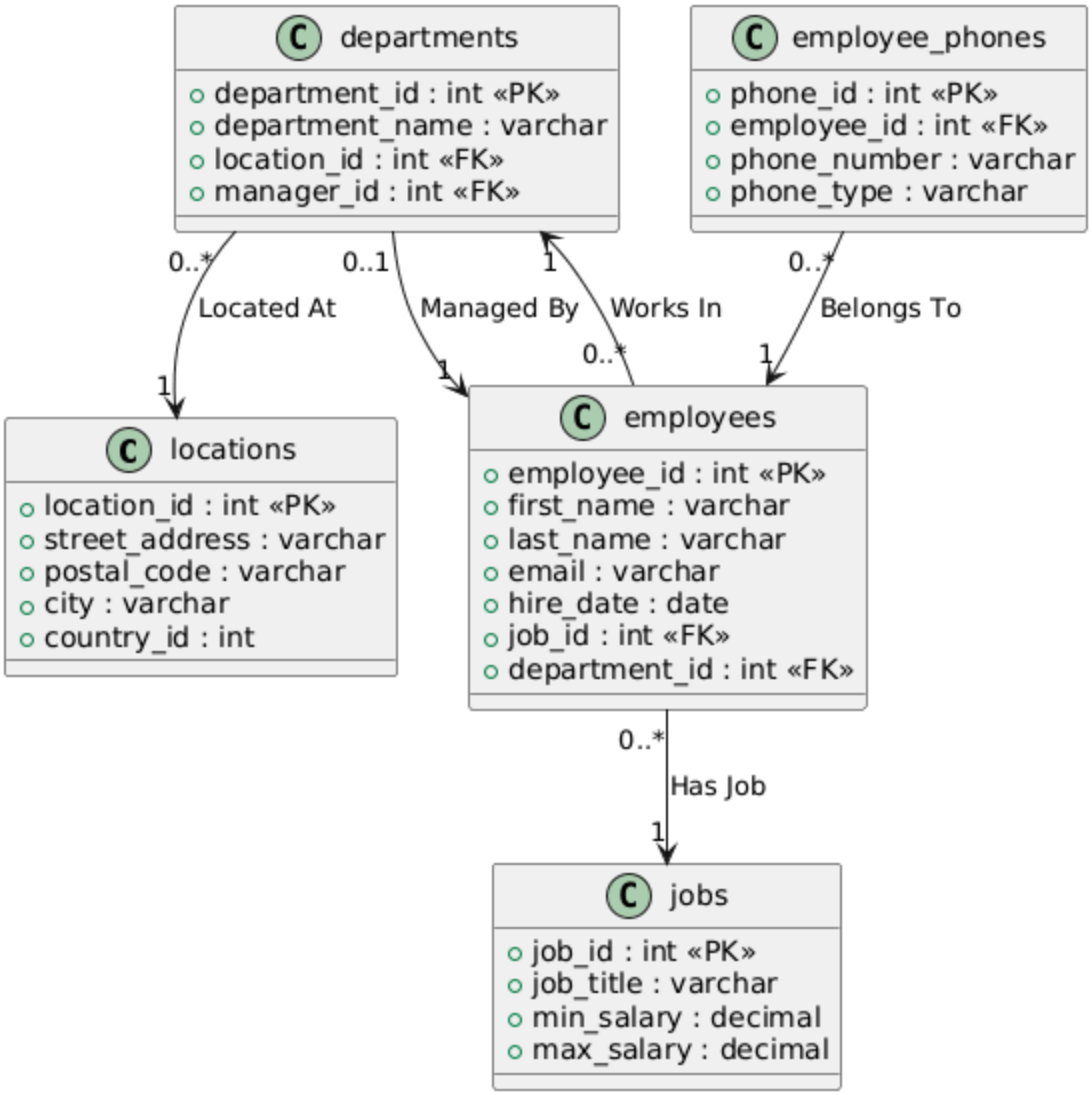


https://drive.google.com/file/d/10xwczegam-ECgkOAsFw0CYoLTB6SuFq4/view?usp=sharing

รูปภาพแสดง Chen ERD ข้อ 8



รูปภาพแสดง Crow's Foot ERD ข้อ 8



รูปภาพแสดง UML Class Diagram ข้อ 8

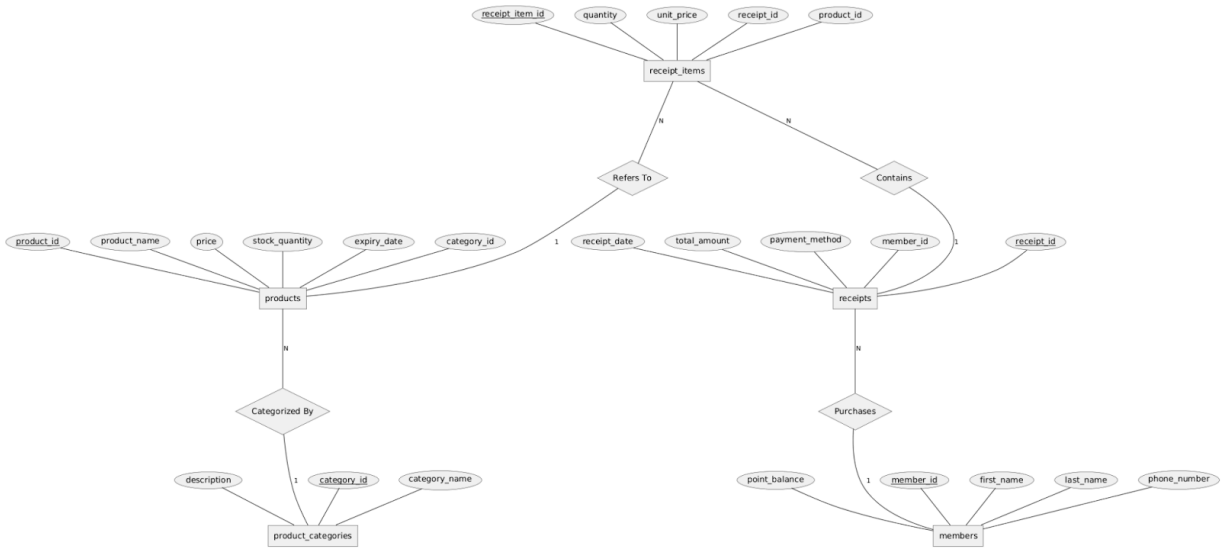
```sql
-- Create locations table first (referenced by departments)
CREATE TABLE locations (
    location_id INT NOT NULL,
    street_address VARCHAR(200),
    postal_code VARCHAR(20),
    city VARCHAR(100),
    country_id INT,
    PRIMARY KEY (location_id)
);
```

```sql
-- Create jobs table (referenced by employees)
CREATE TABLE jobs (
    job_id INT NOT NULL,
    job_title VARCHAR(100),
    min_salary DECIMAL(10,2),
    max_salary DECIMAL(10,2),
    PRIMARY KEY (job_id)
);

-- Create employees table (referenced by departments and
employee_phones)
CREATE TABLE employees (
    employee_id INT NOT NULL,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    hire_date DATE,
    job_id INT,
    department_id INT,
    PRIMARY KEY (employee_id),
    FOREIGN KEY (job_id) REFERENCES jobs(job_id)
);

-- Create departments table with foreign key references
CREATE TABLE departments (
    department_id INT NOT NULL,
    department_name VARCHAR(100),
    location_id INT,
    manager_id INT,
    PRIMARY KEY (department_id),
    FOREIGN KEY (location_id) REFERENCES locations(location_id),
    FOREIGN KEY (manager_id) REFERENCES employees(employee_id)
);

-- Add foreign key constraint for employees.department_id after
departments table is created
ALTER TABLE employees
ADD FOREIGN KEY (department_id) REFERENCES
departments(department_id);
```

```sql
-- Create employee_phones table with foreign key reference
CREATE TABLE employee_phones (
    phone_id INT NOT NULL,
    employee_id INT NOT NULL,
    phone_number VARCHAR(20),
    phone_type VARCHAR(20),
    PRIMARY KEY (phone_id),
    FOREIGN KEY (employee_id) REFERENCES employees(employee_id)
);
```

09: A supermarket that wants to store products, receipts, and reviews.

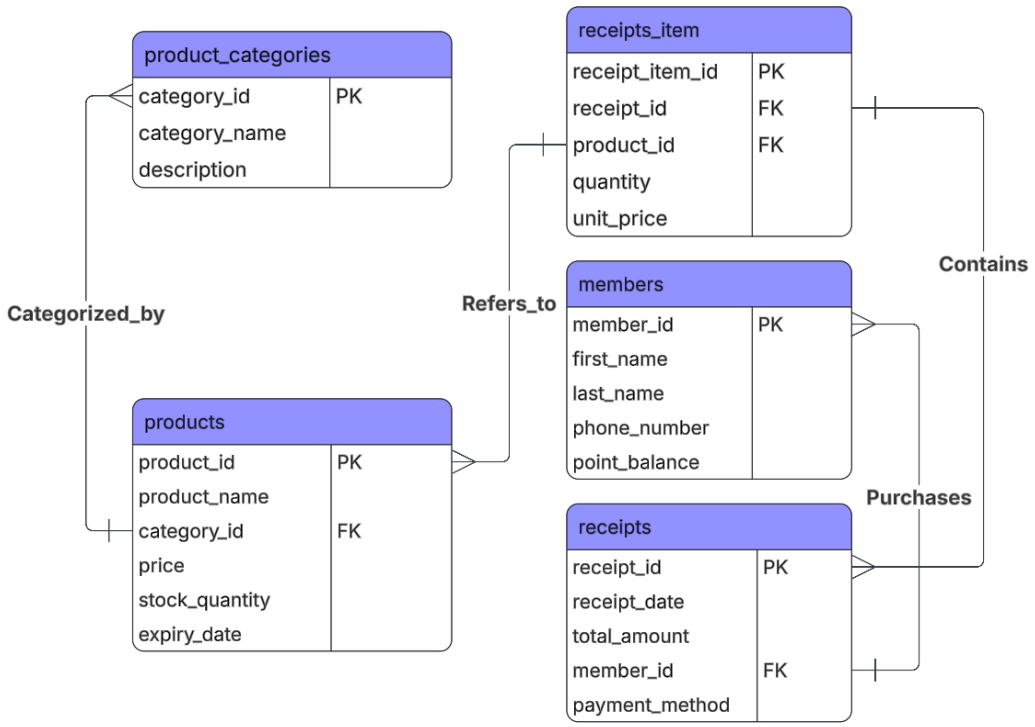
1. product_categories (category_id, category_name, description)
2. products (product_id, product_name, category_id, price, stock_quantity, expiry_date)
3. members (member_id, first_name, last_name, phone_number, point_balance)
4. receipts (receipt_id, receipt_date, total_amount, member_id, payment_method)
5. receipt_items (receipt_item_id, receipt_id, product_id, quantity, unit_price)



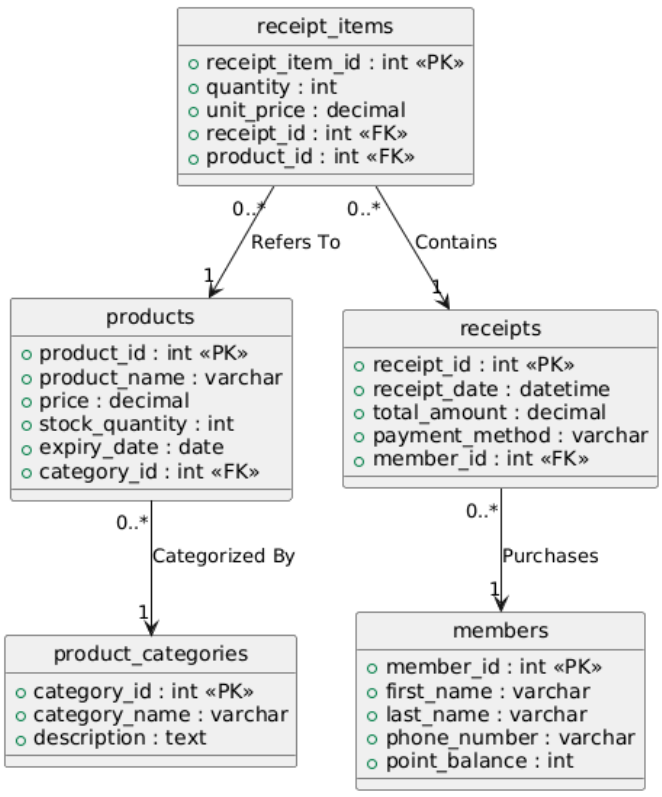
รูปภาพแสดง Chen ERD ข้อ 9

https://drive.google.com/file/d/1beGONWvrRF9kY4mt0osH2bALpBBABoDp/view?usp=sharing

URL Chen ERD ข้อ 9



รูปภาพแสดง Crow's Foot ERD ข้อ 9



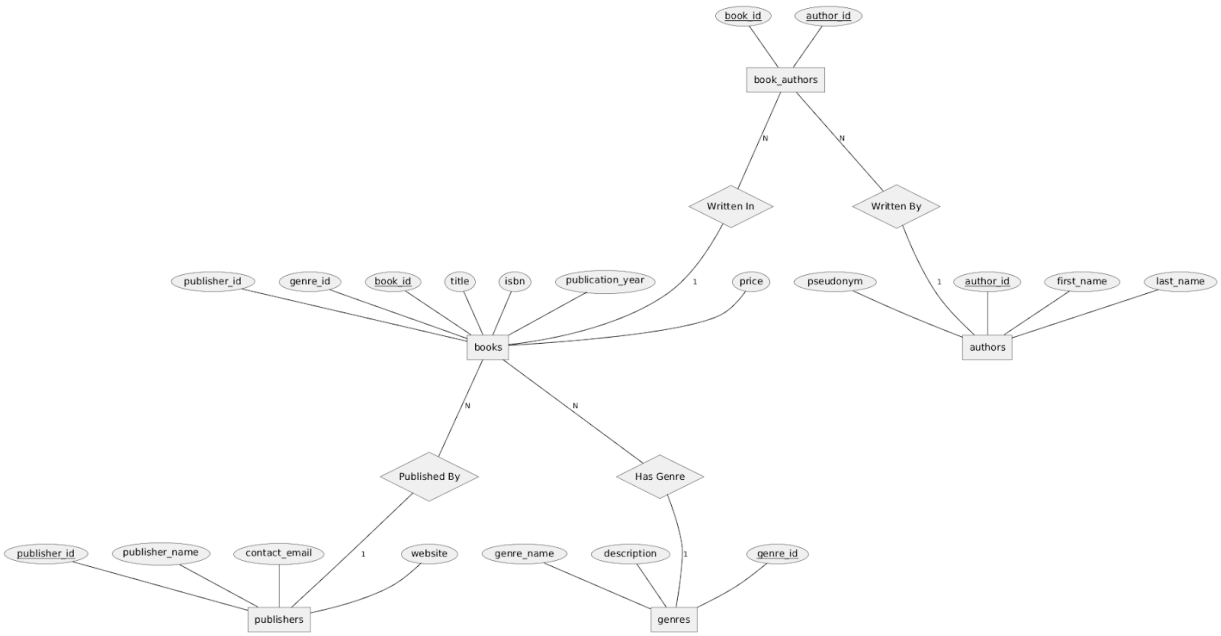
รูปภาพแสดง UML Class Diagram ข้อ 9

```sql
-- Create product_categories table first
CREATE TABLE product_categories (
    category_id INT NOT NULL,
    category_name VARCHAR(100) NOT NULL,
    description VARCHAR(500),
    PRIMARY KEY (category_id)
);

-- Create products table with foreign key reference
CREATE TABLE products (
    product_id INT NOT NULL,
    product_name VARCHAR(150) NOT NULL,
    price DECIMAL(10, 2),
    stock_quantity INT,
    expiry_date DATE,
    category_id INT,
    PRIMARY KEY (product_id),
    FOREIGN KEY (category_id) REFERENCES
product_categories(category_id)
);

-- Create members table
CREATE TABLE members (
    member_id INT NOT NULL,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    phone_number VARCHAR(20),
    point_balance INT DEFAULT 0,
    PRIMARY KEY (member_id)
);

-- Create receipts table with foreign key reference
CREATE TABLE receipts (
    receipt_id INT NOT NULL,
    receipt_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    total_amount DECIMAL(10, 2),
    payment_method VARCHAR(50),
    member_id INT,
    PRIMARY KEY (receipt_id),
```

```
    FOREIGN KEY (member_id) REFERENCES members(member_id)
);

-- Create receipt_items table with foreign keys
CREATE TABLE receipt_items (
    receipt_item_id INT NOT NULL,
    quantity INT,
    unit_price DECIMAL(10, 2),
    receipt_id INT,
    product_id INT,
    PRIMARY KEY (receipt_item_id),
    FOREIGN KEY (receipt_id) REFERENCES receipts(receipt_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

10: A bookstore wants to keep books and authors of each book.

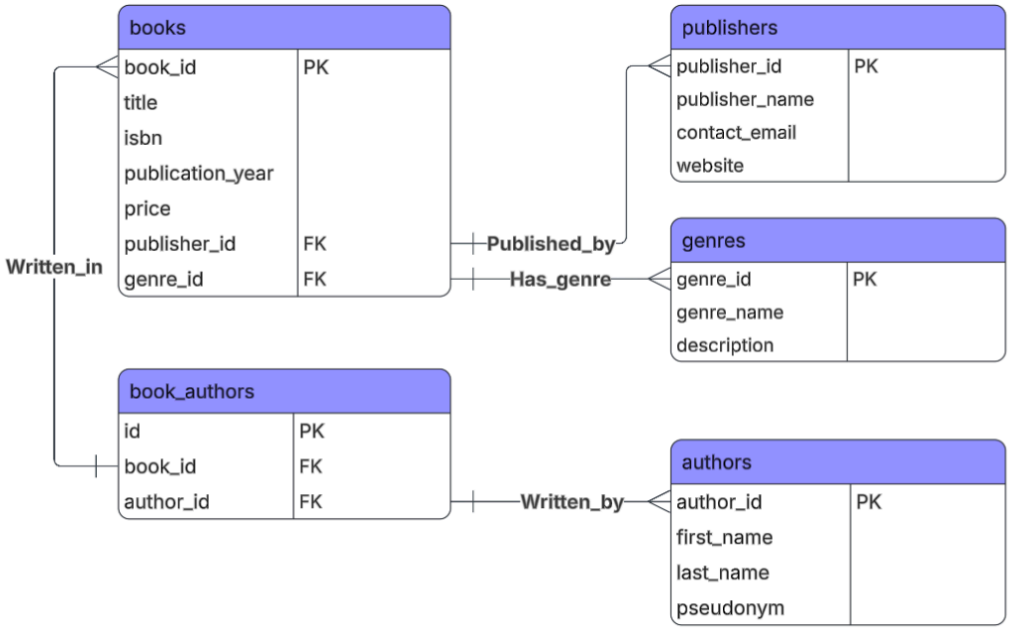
1. publishers (publisher_id, publisher_name, contact_email, website)
2. genres (genre_id, genre_name, description)
3. books (book_id, title, isbn, publication_year, price, publisher_id, genre_id)
4. authors (author_id, first_name, last_name, pseudonym)
5. book_authors (id,book_id, author_id)



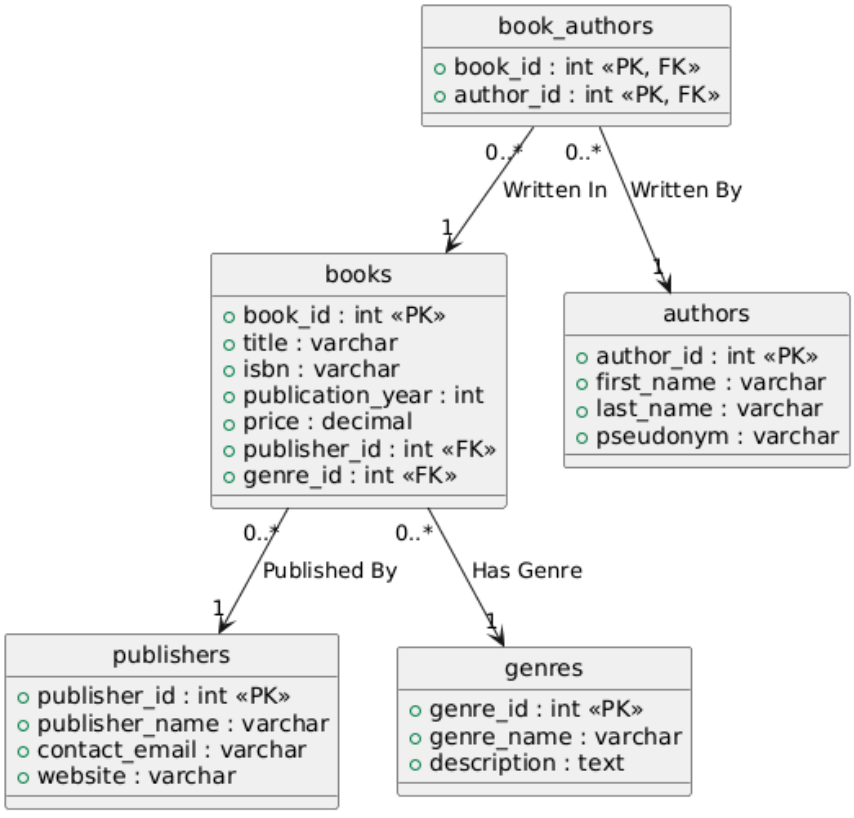
รูปภาพแสดง Chen ERD ข้อ 10

https://drive.google.com/file/d/14Sj3vfHhl6sCz_M53lB7w1oEI7AvWHdY/view?usp=sharing

URL Chen ERD ข้อ 10



รูปภาพแสดง Crow's Foot ERD ข้อ 10



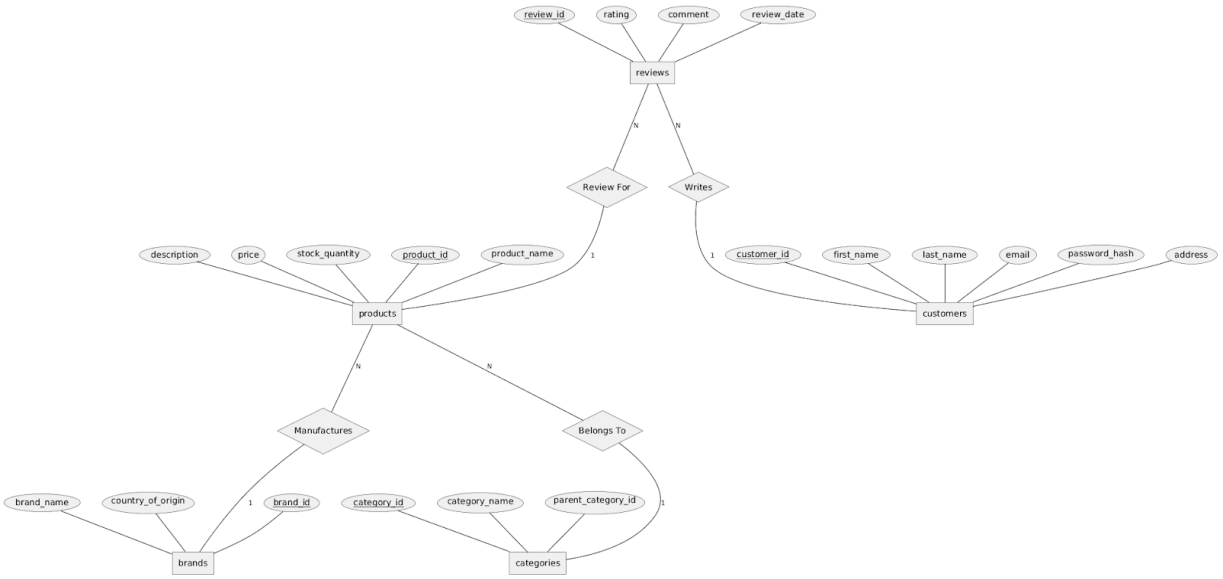
รูปภาพแสดง UML Class Diagram ข้อ 10

```sql
-- Create publishers table first
CREATE TABLE publishers (
    publisher_id INT NOT NULL,
    publisher_name VARCHAR(100) NOT NULL,
    contact_email VARCHAR(150),
    website VARCHAR(150),
    PRIMARY KEY (publisher_id)
);

-- Create genres table
CREATE TABLE genres (
    genre_id INT NOT NULL,
    genre_name VARCHAR(100) NOT NULL,
    description VARCHAR(200),
    PRIMARY KEY (genre_id)
);

-- Create authors table
CREATE TABLE authors (
    author_id INT NOT NULL,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    pseudonym VARCHAR(100),
    PRIMARY KEY (author_id)
);

-- Create books table with foreign key references
CREATE TABLE books (
    book_id INT NOT NULL,
    title VARCHAR(150) NOT NULL,
    isbn VARCHAR(20),
    publication_year INT,
    price DECIMAL(10, 2),
    publisher_id INT,
    genre_id INT,
    PRIMARY KEY (book_id),
    FOREIGN KEY (publisher_id) REFERENCES publishers(publisher_id),
    FOREIGN KEY (genre_id) REFERENCES genres(genre_id)
);
```

```sql
-- Create book_authors table (Junction Table)
CREATE TABLE book_authors (
    book_id INT NOT NULL,
    author_id INT NOT NULL,
    PRIMARY KEY (book_id, author_id),
    FOREIGN KEY (book_id) REFERENCES books(book_id),
    FOREIGN KEY (author_id) REFERENCES authors(author_id)
);
```

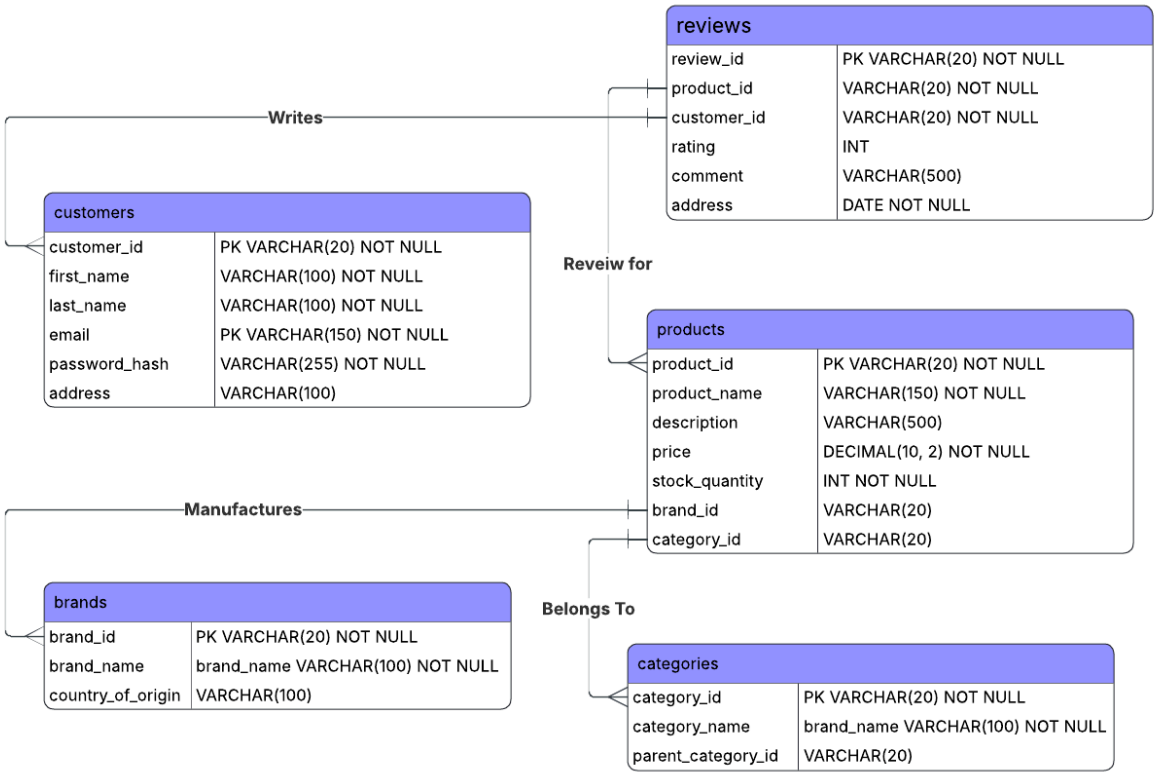
11: An online store that wants to store customers, products, and reviews.

1. brands (brand_id, brand_name, country_of_origin)
2. categories (category_id, category_name, parent_category_id)
3. products (product_id, product_name, description, price, stock_quantity, brand_id, category_id)
4. customers (customer_id, first_name, last_name, email, password_hash, address)
5. reviews (review_id, product_id, customer_id, rating, comment, review_date)

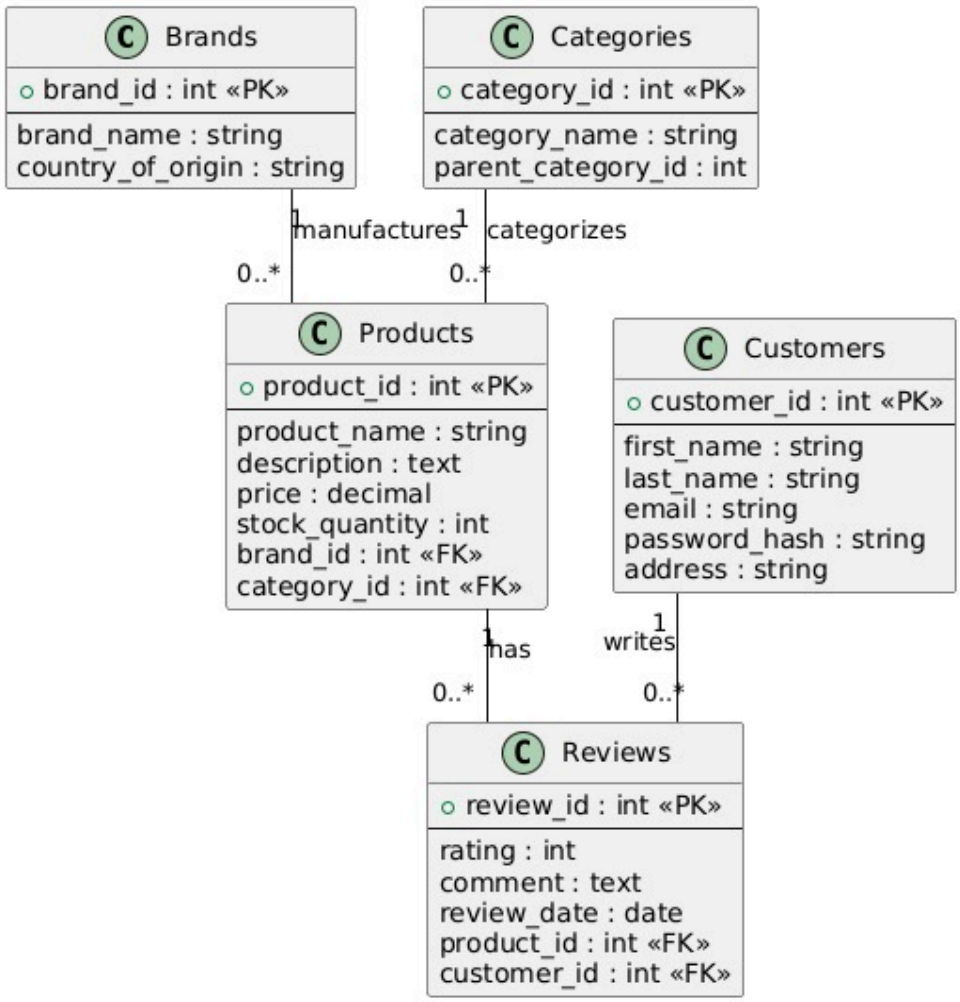


https://drive.google.com/file/d/1uSh8n74DrS0MaWB9tMEhDbo9LGrkK2zu/view?usp=sharing

รูปภาพแสดง Chen ERD ข้อ 11



รูปภาพแสดง Crow’s Foot ERD ข้อ 11



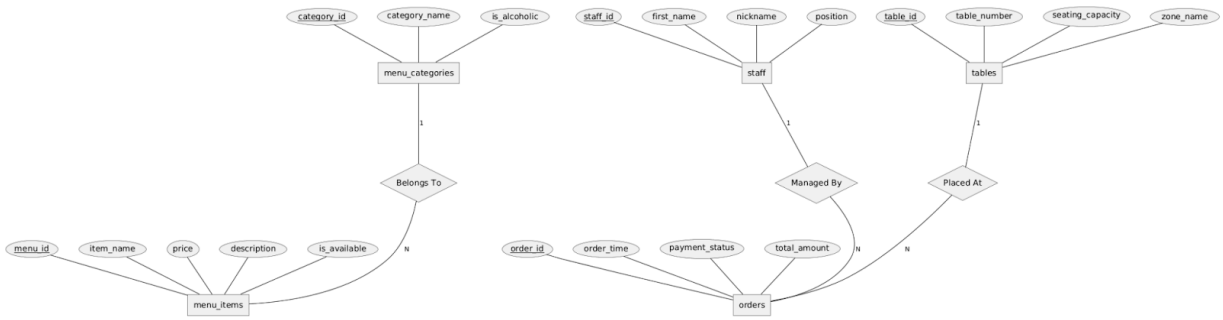
รูปภาพแสดง UML Class Diagram ข้อ 11

```sql
--สร้างตาราง Brands
CREATE TABLE brands (
    brand_id INT PRIMARY KEY AUTO_INCREMENT,
    brand_name VARCHAR(100) NOT NULL,
    country_of_origin VARCHAR(50)
);

--สร้างตาราง Categories (รองรับ Hierarchy)
CREATE TABLE categories (
    category_id INT PRIMARY KEY AUTO_INCREMENT,
    category_name VARCHAR(100) NOT NULL,
    parent_category_id INT,
    FOREIGN KEY (parent_category_id) REFERENCES
```

```
categories(category_id)
);

--สร้างตาราง Products
CREATE TABLE products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    product_name VARCHAR(255) NOT NULL,
    description TEXT,
    price DECIMAL(10, 2) NOT NULL,
    stock_quantity INT DEFAULT 0,
    brand_id INT,
    category_id INT,
    FOREIGN KEY (brand_id) REFERENCES brands(brand_id),
    FOREIGN KEY (category_id) REFERENCES categories(category_id)
);

--สร้างตาราง Customers
CREATE TABLE customers (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    address TEXT
);

--สร้างตาราง Reviews
CREATE TABLE reviews (
    review_id INT PRIMARY KEY AUTO_INCREMENT,
    product_id INT,
    customer_id INT,
    rating INT CHECK (rating BETWEEN 1 AND 5),
    comment TEXT,
    review_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (product_id) REFERENCES products(product_id),
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);
```

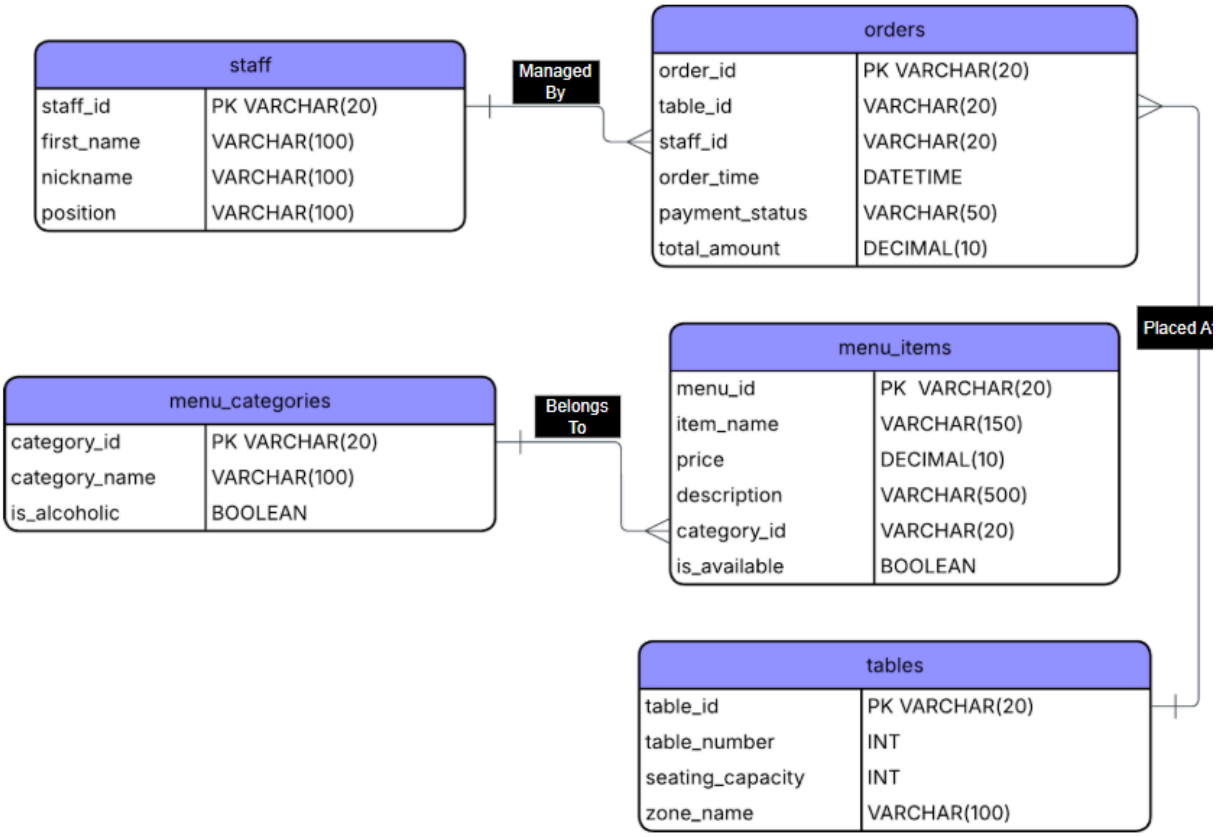
12: A restaurant wants to keep tables, menus, and order of each table.

1. menu_categories (category_id, category_name, is_alcoholic)
2. menu_items (menu_id, item_name, price, description, category_id, is_available)
3. tables (table_id, table_number, seating_capacity, zone_name)
4. staff (staff_id, first_name, nickname, position)
5. orders (order_id, table_id, staff_id, order_time, payment_status, total_amount)

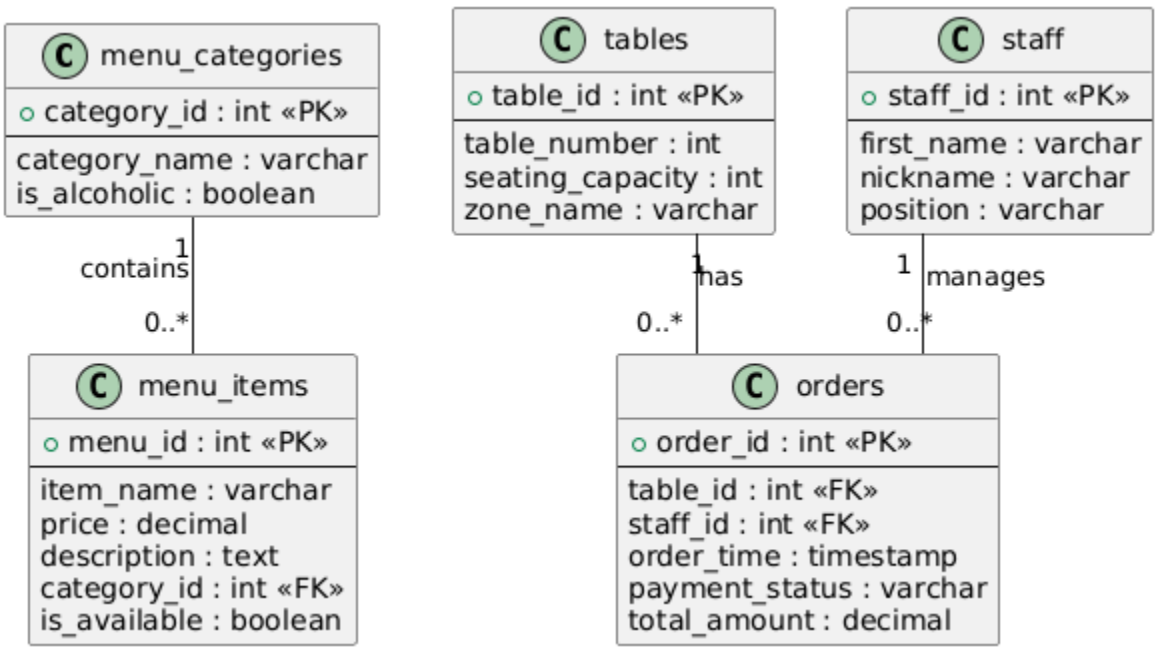


https://drive.google.com/file/d/1BEfhEGeqCAcsQspxfZj0yslVw9psUYNf/view?usp=sharing

รูปภาพแสดง Chen ERD ข้อ 12



รูปภาพแสดง Crow's Foot ERD ข้อ 12



รูปภาพแสดง UML Class Diagram ข้อ 12

```sql
--สร้างตารางหมวดหมู่เมนู
CREATE TABLE menu_categories (
    category_id SERIAL PRIMARY KEY,
    category_name VARCHAR(100) NOT NULL,
    is_alcoholic BOOLEAN DEFAULT FALSE
);

--สร้างตารางรายการเมนู
CREATE TABLE menu_items (
    menu_id SERIAL PRIMARY KEY,
    item_name VARCHAR(150) NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    description TEXT,
    category_id INT,
    is_available BOOLEAN DEFAULT TRUE,
    FOREIGN KEY (category_id) REFERENCES menu_categories(category_id)
);


--สร้างตารางโต๊ะ
CREATE TABLE tables (
    table_id SERIAL PRIMARY KEY,
    table_number INT NOT NULL UNIQUE,
    seating_capacity INT,
    zone_name VARCHAR(50)
);

--สร้างตารางพนักงาน
CREATE TABLE staff (
    staff_id SERIAL PRIMARY KEY,
    first_name VARCHAR(100) NOT NULL,
    nickname VARCHAR(50),
    position VARCHAR(100)
);

--สร้างตารางคำสั่งซื้อ
CREATE TABLE orders (
    order_id SERIAL PRIMARY KEY,
    table_id INT,
```

```
    staff_id INT,
    order_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    payment_status VARCHAR(20),
    total_amount DECIMAL(10, 2) DEFAULT 0.00,
    FOREIGN KEY (table_id) REFERENCES tables(table_id),
    FOREIGN KEY (staff_id) REFERENCES staff(staff_id)
);
```

Task 2: Modify Modified Existing Database

Step 01: Insert Duplicate and Conflicting Data

Insert Anomalies:

- A duplicate studentID but with a different name.

```
INSERT INTO students (studentID, firstname, lastname) VALUES (101, 'Mark', 'Taylor');
```

- A duplicate courseID but with a different course name.

```
INSERT INTO courses (courseID, courseName) VALUES (201, 'Advanced Databases');
```

- The same combination of studentID and courseID but with a different grade.

```
INSERT INTO grades (studentID, courseID, grade) VALUES (101, 201, 'B');
```

students table

	studentid integer	firstname character varying (50)	lastname character varying (50)
1	101	Max	Naja
2	102	Beam	LnwZa
3	103	Up	Malaew
4	101	Mark	Taylor

courses table

	courseid integer	coursename character varying (100)
1	201	Database Systems
2	202	Data Model
3	203	Computer Architecture
4	201	Advanced

grades table

	studentid integer	courseid integer	grade character varying (2)
1	101	201	B
2	102	201	A
3	103	202	B
4	101	201	B

Step 02: Resolve the Anomalies

Delete the duplicate and conflicting rows:

- students table

```
DELETE FROM students WHERE studentID = 101 AND firstname = 'Mark';
```

- courses table

```
DELETE FROM courses WHERE courseID = 201 AND courseName = 'Advanced Databases';
```

- grades table

```
DELETE FROM grades WHERE studentID = 101 AND courseID = 201 AND grade = 'B';
```

students table

	studentid integer	firstname character varying (50)	lastname character varying (50)
1	101	Max	Naja
2	102	Beam	LnwZa
3	103	Up	Malaew

courses table

	courseid integer	coursename character varying (100)
1	201	Database Systems
2	202	Data Model
3	203	Computer Architecture

grades table

	studentid integer	courseid integer	grade character varying (2)
1	101	201	B
2	102	201	A
3	103	202	B

Step 03: Add Primary Key and Foreign Key Constraints

Add primary key(PK) to each table:

- students table

```
ALTER TABLE students ADD CONSTRAINT pk_students PRIMARY KEY (studentID);
```

- courses table

```
ALTER TABLE courses ADD CONSTRAINT pk_courses PRIMARY KEY (courseID);
```

- grades table

```
ALTER TABLE grades ADD CONSTRAINT pk_grades PRIMARY KEY (studentID, courseID);
```

Add foreign key(FK):

```
ALTER TABLE grades ADD CONSTRAINT fk_grades_students FOREIGN KEY (studentID)
REFERENCES students(studentID);
ALTER TABLE grades ADD CONSTRAINT fk_grades_courses FOREIGN KEY (courseID)
REFERENCES courses(courseID);
```

students table	ERROR: duplicate key value violates unique constraint "pk_students" Key (studentid)=(101) already exists. SQL state: 23505 Detail: Key (studentid)=(101) already exists.
courses table	ERROR: duplicate key value violates unique constraint "pk_courses" Key (courseid)=(201) already exists. SQL state: 23505 Detail: Key (courseid)=(201) already exists.
grades table	ERROR: duplicate key value violates unique constraint "pk_grades" Key (studentid, courseid)=(101, 201) already exists. SQL state: 23505 Detail: Key (studentid, courseid)=(101, 201) already exists.

Step 04: Add the departments Entity and more tables

departments table

	departmentid [PK] integer	departmentname character varying (50)
1	1	Computer Science
2	2	Information Systems

students table

	studentid [PK] integer	firstname character varying (50)	lastname character varying (50)	departmentid integer
1	101	Max	Naja	1
2	102	Beam	LnwZa	1
3	103	Up	Malaew	1

courses table

	courseid [PK] integer	coursename character varying (100)	departmentid integer
1	201	Database Systems	2
2	202	Data Model	2
3	203	Computer Architecture	2

Step 04-01: A law firm wants to keep employees and their offices.

	employee_id [PK] integer	firstname character varying (50)	middlename character varying (50)	lastname character varying (50)	job_title character varying (100)	hire_date date	citizen_id character varying (50)	email character varying (100)	phone character varying (20)	office_id integer	status character varying (50)	create_at timestamp without time zone
1	1001	Guntee	Piang	Duamanee	Senior Partner	2015-01-15	1234567890123	somchai.w@lawfirm.com	081-111-2222	1	Active	2026-01-27 14:39:53.337475
2	1002	Nutthawat	Up	Primsirikunawut	Associate Lawyer	2018-06-20	2345678901234	nuttapong.s@lawfirm.c...	082-222-3333	1	Active	2026-01-27 14:39:53.337475
3	1003	Panuwat	Poom	Boonsak	Legal Assistant	2020-03-10	3456789012345	apinya.t@lawfirm.com	083-333-4444	2	Active	2026-01-27 14:39:53.337475
4	1004	Ratananan	Yok	Siriponwat	Junior Lawyer	2021-09-01	4567890123456	wanchai.p@lawfirm.com	084-444-5555	2	Active	2026-01-27 14:39:53.337475

employees table

	office_id [PK] integer	office_name character varying (100)	address_line1 character varying (200)	address_line2 character varying (200)	city character varying (100)	state character varying (100)	postal_code character varying (20)	country character varying (100)	phone character varying (20)
1	1	Downtown Office	123 Main Street	Suite 500	Bangkok	Bangkok	10500	Thailand	02-123-4567
2	2	Sukhumvit Branch	456 Sukhumvit Road	Floor 12	Bangkok	Bangkok	10110	Thailand	02-234-5678
3	3	Silom Office	789 Silom Road	[null]	Bangkok	Bangkok	10500	Thailand	02-345-6789

offices table

Step 04-06: A film archive that wants to store movies, actors, and casts.

genres table

	genre_id [PK] integer	genre_name character varying (100)	description text
1	1	Action	High energy films with physical stunts and chases
2	2	Drama	Serious narrative focusing on character developm...
3	3	Comedy	Light-hearted films designed to amuse
4	4	Sci-Fi	Science fiction exploring futuristic concepts
5	5	Romance	Love stories and relationships

actors table

	actor_id [PK] integer	first_name character varying (100)	last_name character varying (100)	stage_name character varying (100)
1	101	Tom	Cruise	Tom Cruise
2	102	Scarlett	Johansson	Scarlett Johansson
3	103	Leonardo	DiCaprio	Leo DiCaprio
4	104	Emma	Stone	Emma Stone
5	105	Denzel	Washington	Denzel Washington

date_of_birth date	nationality character varying (100)	gender character varying (20)	biography text
1962-07-03	American	Male	Award-winning action star
1984-11-22	American	Female	Versatile actress and sin...
1974-11-11	American	Male	Academy Award winner
1988-11-06	American	Female	Oscar-winning actress
1954-12-28	American	Male	Legendary dramatic actor

directors table

	director_id [PK] integer	first_name character varying (100)	last_name character varying (100)	date_of_birth date	nationality character varying (100)
1	201	Christopher	Nolan	1970-07-30	British
2	202	Steven	Spielberg	1946-12-18	American
3	203	Greta	Gerwig	1983-08-04	American
4	204	James	Cameron	1954-08-16	Canadian
5	205	Martin	Scorsese	1942-11-17	American

movies table

	movie_id [PK] integer	title character varying (255)	original_title character varying (255)	release_year integer	release_date date	duration_minutes integer
1	301	Inception	Inception	2010	2010-07-16	148
2	302	The Shawshank Redempti...	The Shawshank Redempti...	1994	1994-09-23	142
3	303	Titanic	Titanic	1997	1997-12-19	195
4	304	La La Land	La La Land	2016	2016-12-09	128

	synopsis text	language character varying (50)
1	A thief who steals corporate secrets through dream-sharing technol...	English
2	Two imprisoned men bond over years finding redemption	English
3	A love story aboard the ill-fated maiden voyage	English
4	A jazz musician and aspiring actress fall in love in LA	English

	country character varying (100)	budget numeric (15,2)	box_office numeric (15,2)	rating numeric (3,1)
1	USA	160000000.00	829895144.00	8.8
2	USA	25000000.00	28341469.00	9.3
3	USA	200000000.00	2187463944.00	7.9
4	USA	30000000.00	446092357.00	8.0

movie_genres table

	movie_id [PK] integer	genre_id [PK] integer
1	301	1
2	301	4
3	302	2
4	303	2
5	303	5
6	304	2
7	304	5

casts table

	cast_id [PK] integer	movie_id integer	actor_id integer	role_name character varying (255)	billing_order integer	screen_time_minutes integer
1	1	301	103	Dom Cobb	1	120
2	2	301	102	Natasha	2	80
3	3	303	103	Jack Dawson	1	150
4	4	304	104	Mia Dolan	1	100

movie_directors table

	movie_id [PK] integer	director_id [PK] integer
1	301	201
2	303	204
3	304	203

Step 04-08: A company that wants to store departments, employees, employee phone numbers

locations table

	location_id [PK] integer	street_address character varying (200)	postal_code character varying (20)	city character varying (100)	country_id integer
1	1	1600 Amphitheatre Parkw...	94043	Mountain View	1
2	2	One Microsoft Way	98052	Redmond	1
3	3	1 Infinite Loop	95014	Cupertino	1
4	4	410 Terry Avenue North	98109	Seattle	1

jobs table

	job_id [PK] integer	job_title character varying (100)	min_salary numeric (10,2)	max_salary numeric (10,2)
1	1	Product Manager	100000.00	180000.00
2	2	Senior Software Engine...	120000.00	200000.00
3	3	Software Engineer	80000.00	150000.00

employees table

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	hire_date date	job_id integer	department_id integer
1	5001	Wisit	Suwannao	pluem@company.com	2018-03-15	2	10
2	5002	Supawit	Maryat	beam@company.com	2019-07-22	1	10
3	5003	Polwarit	Watthanahemmart	max@company.com	2020-01-10	3	10

departments table

	department_id [PK] integer	department_name character varying (100)	location_id integer	manager_id integer
1	10	Engineering	1	5001
2	20	Product	2	5003
3	30	Human Resources	3	5002

employee_phones table

	phone_id [PK] integer	employee_id integer	phone_number character varying (20)	phone_type character varying (20)
1	1	5001	555-0101	Mobile
2	2	5001	555-0102	Office
3	3	5002	555-0201	Mobile
4	4	5003	555-0301	Mobile
5	5	5003	555-0302	Home

Summary

จากแลปนี้ได้เรียนรู้เรื่องของการสร้าง Primary Key และ Foreign Key เพื่อป้องกันการซ้ำของข้อมูล จากการทำแลปนี้ทำให้รู้ว่า constraints มีความสำคัญในการรักษาความถูกต้องและความสมบูรณ์ของข้อมูล โดย Primary Key ป้องกันข้อมูลซ้ำ ขณะที่ Foreign Key รับรองความสัมพันธ์ระหว่างตาราง

และจากการขยายฐานข้อมูลด้วยการเพิ่ม departments table และเชื่อมกับตาราง students และ courses ทำให้เข้าใจถึงการออกแบบฐานข้อมูลแบบ normalized ที่ช่วยลดความซ้ำซ้อนและจัดระเบียบข้อมูลได้ดีขึ้น ความท้าทายที่พบคือการลบข้อมูลเกินจำเป็น จากความผิดพลาดทางเทคนิค