1. Consultar los nombres completos y la dirección de su vivienda, de los clientes atendidos por el empleado Jacinto durante el mes de febrero del año 2022.

```sql
SELECT DISTINCT cp.person_name, cp.lastname,
ad.street_type, ci.city_name FROM person AS cp
INNER JOIN client_p AS c
ON c.fk_id_person = cp.id_person
INNER JOIN advisor AS a
ON c.fk_id_advisor = a.id_advisor
INNER JOIN person AS ap
ON a.fk_id_person = ap.id_person
INNER JOIN purchase_receipt AS r
ON r.fk_id_client = c.id_client_p
INNER JOIN address AS ad
ON cp.fk_address = ad.id_address
INNER JOIN city as ci
ON ad.fk_code_city = ci.code_city
WHERE ap.person_name = 'Jacinto'
AND YEAR(r.purchase_receipt_date) = 2022
AND MONTH(r.purchase_receipt_date) = 02;
```

person_name		street_type	city_name
Camilo	Amdé	Avenida	Armenia
Clemencia	Mendoza	Avenida	Manizales

2. Consultar el color de carro más vendido de la marca Mazda durante el año 2021.

```
DELIMITER $$
-- retorna el número de ventas de un tipo de vehículo registradas en un mes dado
CREATE PROCEDURE calc_num_vehicles_month_type(IN number_year INT, IN number_month INT,
IN vehicle_brand VARCHAR(30),IN vehicle_type VARCHAR(30), OUT result INT)
BEGIN
    SELECT COUNT(*) INTO result  FROM purchase_receipt AS r
    INNER JOIN vehicle_type AS v
    ON r.fk_id_vehicle_type = v.id_vehicle_type
    INNER JOIN vehicle_brand AS vb
    ON v.fk_id_vehicle_brand = vb.id_vehicle_brand
    WHERE
    vb.vehicle_brand = vehicle_brand
    AND
    MONTH(r.purchase_receipt_date) = number_month
    AND
    YEAR(r.purchase_receipt_date) = number_year
    AND v.vehicle_type = vehicle_type;
END $$
```

```
DELIMITER $$
-- retorna el número de ventas de un typo de vehículo en los primeros n mese
CREATE PROCEDURE calc_num_vehicles_year_type(IN number_year INT, IN limit_month INT,
IN vehicle_brand VARCHAR(30),IN vehicle_type VARCHAR(30), OUT result INT)
BEGIN
    SET @result=0;
    SET @month=1;
    WHILE @month <= limit_month DO
        call calc_num_vehicles_month_type(number_year, @month,vehicle_brand, vehicle_type, @num);
        SET @result := @result+@num;
        SET @month := @month+1;
    END WHILE;
    #SELECT @result;
    SELECT @result INTO result;
END $$
```

```
DELIMITER $$
-- retorna el color del carro más vendido en los primeros n meses de una marca dada
CREATE PROCEDURE color_vehicle_best_seller(IN number_year INT, IN limit_month INT,
IN vehicle_brand VARCHAR(30),IN counter INT, OUT result VARCHAR(30))
BEGIN
    SET counter=0;
    SET @final="nada";
    SET @bandera=0;
    WHILE counter < (SELECT COUNT(*) FROM vehicle_type) DO
        SET @type=(
        SELECT vehicle_type FROM vehicle_type
        limit 1
        OFFSET counter
        );
        CALL calc_num_vehicles_year_type(number_year, limit_month, vehicle_brand, @type, @num);
        IF @num > @bandera THEN
            SET @bandera = @num;
            SET @final=(
                SELECT DISTINCT(vehicle_color_name) FROM vehicle_color AS vc
                INNER JOIN vehicle_type AS vt
                ON vt.fk_id_vehicle_color = vc.id_vehicle_color
                WHERE vt.vehicle_type = @type
                );
        END IF;
        SET counter := counter+1;
    END WHILE;
    SELECT @final INTO result;
END $$
```

```
CALL color_vehicle_best_seller(2022, 12, "Mazda",1, @result);
SELECT @result AS "color vehículo";
```

color vehículo
Azul Mar

3. Consultar la cantidad de vehículos que vendió Mazda en la sucursal situada en la ciudad de Manizales durante el año 2021.

```
DELIMITER $$
-- retorna el número de ventas de una marca registradas en una ciudad en un mes
CREATE PROCEDURE calc_num_vehicles_month(IN number_year INT,
IN number_month INT, IN city VARCHAR(30), IN vehicle_brand VARCHAR(30),
OUT result INT)
BEGIN
    SELECT COUNT(*) INTO result  FROM purchase_receipt AS r
    INNER JOIN vehicle_type AS v
    ON r.fk_id_vehicle_type = v.id_vehicle_type
    INNER JOIN vehicle_brand AS vb
    ON v.fk_id_vehicle_brand = vb.id_vehicle_brand
    INNER JOIN dealership AS d
    ON vb.fk_nit_dealership = d.nit_dealership
    INNER JOIN branch AS b
    ON b.fk_nit_dealership = d.nit_dealership
    INNER JOIN address AS ad
    ON b.fk_id_address = ad.id_address
    INNER JOIN city as ci
    ON ad.fk_code_city = ci.code_city
    WHERE ci.city_name = city
    AND
    vb.vehicle_brand = vehicle_brand
    AND
    MONTH(r.purchase_receipt_date) = number_month
    AND YEAR(r.purchase_receipt_date) = number_year;
END $$
```

```
DELIMITER $$
-- retorna el número de ventas de una marca en los primeros n meses en determinada ciudad
CREATE PROCEDURE calc_num_vehicles_year(IN number_year INT, IN limit_month INT,
IN city VARCHAR(30), IN vehicle_brand VARCHAR(30), OUT result INT)
BEGIN
    SET @result=0;
    SET @month=1;
    WHILE @month <= limit_month DO
        call calc_num_vehicles_month(number_year, @month, city, vehicle_brand, @num);
        SET @result := @result+@num;
        SET @month := @month+1;
    END WHILE;
    #SELECT @result;
    SELECT @result INTO result;
END $$
```

```
CALL calc_num_vehicles_year(2021, 12, "Manizales", "Mazda", @num);
SELECT @num AS "Número de vehículos";
```

Número de vehículos
4

4. Consultar cuál fue el mes en el que Jacinto vendió más vehículos durante el segundo semestre del año 2021. ***Recomendación***: consultar sobre procedimientos almacenados y ciclo while para recorrer los meses.

```
DELIMITER $$
-- retorna el número de ventas hechas por un supervisor en un mes dado y año dado
CREATE PROCEDURE sales_advisor_month(IN advisor_name VARCHAR(50),
IN number_month INT, IN number_year INT, OUT result INT)
BEGIN
    SELECT COUNT(*) INTO result FROM purchase_receipt AS r
    INNER JOIN advisor AS a
    ON r.fk_id_advisor = a.id_advisor
    INNER JOIN person AS p
    ON a.fk_id_person = p.id_person
    WHERE
    p.person_name = advisor_name
    AND
    MONTH(r.purchase_receipt_date) = number_month
    AND
    YEAR(r.purchase_receipt_date) = number_year;
END $$
```

```
DELIMITER $$
-- retorna el mes del segundo semestre en el que el supervisor dado vendió más
CREATE PROCEDURE month_more_sales_second_semster_advisor(IN advisor_name VARCHAR(50), IN number_year INT,
OUT result INT)
BEGIN
    SET @counter=6;
    SET @greate=0;
    WHILE @counter <= 12 DO
        CALL sales_advisor_month(advisor_name, @counter, number_year, @sales);
        IF @sales > @greate THEN
            SET @greate = @counter;
        END IF;
        SET @counter := @counter+1;
    END WHILE;
    SELECT @greate INTO result;
END $$
```

```
CALL month_more_sales_second_semster_advisor("Jacinto", 2021, @mes);
SELECT @mes AS "número de més en el que mas vendió JACINTO";
```

número de més en el que mas vendió JACINTO
7

5. Cuál aseguradora de las que tiene convenio con la concesionaria Mazda, ofrecio el seguro más costoso durante el primer trimestre del año 2022. La consulta debe mostrar el nombre de la aseguradora, el costo del seguro contra todo riesgo, la placa del vehículo y el nombre del cliente que adquirió el seguro.

```
DELIMITER $$
-- retorna el ID del insurance_type más vendido hasta el mes n asociado a un vehicle_insurer asociado a
-- su vez a un dealership
CREATE PROCEDURE insurer_cost_n_months(IN limit_month INT, IN number_year INT,
IN dealership VARCHAR(30), OUT result INT)
BEGIN
    SET @counter=1;
    SET @final=0;
    SET @bandera=0;
    WHILE @counter <= limit_month DO
        SET @id_insurance=(
        SELECT it.id_insurance_types FROM dealership AS d
        INNER JOIN dealership_insurer AS d_i
        ON d_i.fk_nit_dealership = d.nit_dealership
        INNER JOIN vehicle_insurer AS vi
        ON d_i.fk_id_vehicle_insurer = vi.id_vehicle_insurer
        INNER JOIN insurance_types AS it
        ON it.fk_id_vehicle_insurer = vi.id_vehicle_insurer
        WHERE MONTH(it.inception_date) = @counter
        AND
        YEAR(it.inception_date) = number_year
        AND d.dealership_name = dealership
        ORDER BY it.insurance_value DESC
        LIMIT 1
        );

        SET @value_insurance_month = (
            SELECT insurance_value FROM insurance_types
            WHERE id_insurance_types = @id_insurance
        );
        IF @value_insurance_month > @bandera THEN
        SET @final = @id_insurance;
        SET @bandera = @value_insurance_month;
        END IF;
        SET @counter := @counter+1;
    END WHILE;
    SELECT @final INTO result;
END $$;
```

```sql
DELIMITER $$
-- hace la consulta de la concesionaria, cliente, placas, valor de seguro, del seguro más vendidio
-- en el primer trimestre del 2022
CREATE PROCEDURE info_insurance(IN limit_month INT, IN number_year INT, IN dealership VARCHAR(30))
BEGIN
    CALL insurer_cost_n_months(limit_month, number_year, dealership, @result);
    SELECT DISTINCT it.insurance_types AS aseguradora, it.insurance_value AS "valor seguro",
    r.tuition_value AS placas, p.person_name AS cliente FROM insurance_types AS it
    INNER JOIN vehicle_insurer AS vi
    ON it.fk_id_vehicle_insurer = vi.id_vehicle_insurer
    INNER JOIN dealership_insurer AS d_i
    ON d_i.fk_id_vehicle_insurer = vi.id_vehicle_insurer
    INNER JOIN dealership AS d
    ON d_i.fk_nit_dealership = d.nit_dealership
    INNER JOIN vehicle_type AS v
    ON it.fk_id_vehicle_type = v.id_vehicle_type
    INNER JOIN purchase_receipt AS r
    ON r.fk_id_vehicle_type = v.id_vehicle_type
    INNER JOIN client_p AS c
    ON r.fk_id_client = c.id_client_p
    INNER JOIN person AS p
    ON c.fk_id_person = p.id_person
    WHERE it.id_insurance_types = @result;
END $$
```

```sql
CALL info_insurance(3, 2022, 'Mazda');
```

aseguradora	valor seguro	placas	cliente
Ordinario	5000000	4566	Martín
Ordinario	5000000	4345	Andrea
Ordinario	5000000	3234	Angi
Ordinario	5000000	9564	Helen

6. Consulta cual de las tres aseguradoras que tienen convenio con Ford para vehículos de uso particular, ofrece un menor precio para el Bronco Sport 4x4 que adquirió Helen Chufe y cual de las tres aseguradoras ofrece un mayor precio.

```
DELIMITER $$
-- retorna el id del seguro menos costoso para un tipo de vehículo de asociado a una concesionaria
CREATE PROCEDURE more_expensive_for_type_vehicle(IN type_vehicle VARCHAR(45), IN dealership VARCHAR(30),
IN index_insurer_type INT,OUT result INT)
BEGIN
    SELECT it.id_insurance_types INTO result FROM dealership AS d
    INNER JOIN dealership_insurer AS d_i
    ON d_i.fk_nit_dealership = d.nit_dealership
    INNER JOIN vehicle_insurer AS vi
    ON d_i.fk_id_vehicle_insurer  = vi.id_vehicle_insurer
    INNER JOIN insurance_types AS it
    ON vi.id_vehicle_insurer = it.fk_id_vehicle_insurer
    INNER JOIN vehicle_type AS v
    ON it.fk_id_vehicle_type = v.id_vehicle_type
    WHERE v.vehicle_type = type_vehicle
    AND d.dealership_name = dealership
    ORDER BY it.insurance_value DESC
    LIMIT 1
    OFFSET index_insurer_type;
END $$
```

```
DELIMITER $$
-- retorna el id del seguro más costoso para un tipo de vehículo de asociado a una concesionaria
CREATE PROCEDURE less_expensive_for_type_vehicle(IN type_vehicle VARCHAR(45), IN dealership VARCHAR(30),
IN index_insurer_type INT,OUT result INT)
BEGIN
    SELECT it.id_insurance_types INTO result FROM dealership AS d
    INNER JOIN dealership_insurer AS d_i
    ON d_i.fk_nit_dealership = d.nit_dealership
    INNER JOIN vehicle_insurer AS vi
    ON d_i.fk_id_vehicle_insurer  = vi.id_vehicle_insurer
    INNER JOIN insurance_types AS it
    ON vi.id_vehicle_insurer = it.fk_id_vehicle_insurer
    INNER JOIN vehicle_type AS v
    ON it.fk_id_vehicle_type = v.id_vehicle_type
    WHERE v.vehicle_type = type_vehicle
    AND d.dealership_name = dealership
    ORDER BY it.insurance_value
    LIMIT 1
    OFFSET index_insurer_type;
END $$
```

```
DELIMITER $$
-- consulta la aseguradora que ofrece el seguro más económico para un carro
-- consulta la aseguradora que ofrece el seguro más costoso para un carro
CREATE PROCEDURE insurer_more_less(IN type_vehicle VARCHAR(45), IN dealership VARCHAR(30))
BEGIN
    CALL less_expensive_for_type_vehicle(type_vehicle, dealership, 0, @less_price);
    CALL more_expensive_for_type_vehicle(type_vehicle, dealership, 0, @more_price);

    SELECT vi.vehicle_insurer, insurance_value AS "Aseguradora menor ofrece valor" FROM vehicle_insurer AS vi
    INNER JOIN insurance_types AS it
    ON it.fk_id_vehicle_insurer = vi.id_vehicle_insurer
    WHERE it.id_insurance_types = @less_price;

    SELECT vi.vehicle_insurer, insurance_value AS "Aseguradora mayor ofrece valor" FROM vehicle_insurer AS vi
    INNER JOIN insurance_types AS it
    ON it.fk_id_vehicle_insurer = vi.id_vehicle_insurer
    WHERE it.id_insurance_types = @more_price;
END $$
```

```
CALL insurer_more_less("Bronco Spor 4x4", "Mazda");
```

vehicle_insurer	Aseguradora menor ofrece valor
Zura	900000

vehicle_insurer	Aseguradora mayor ofrece valor
Zura	5000000

7. Cuál sería tu propuesta de base de datos no relacional para la base de datos de concesionario?

    a. Adjunta la estructura del BSON resultante

```
// address
{
    "_id" : "String",
    "street_type": "String",
    "city" : {
        "code_city" : "Number",
        "city_name" : "String"
    }
}

//advisor
{
    "_id" : "String",
    "advisor_name" : "String",
    "advisor_lastname" : "String",
    "document_type" : "String",
    "document" : "String",
    "age" : "Number",
    "email" : "String",
    "contract_type" : "String",
    "branch" : "String(_idBrand)"
}

//client_p
{
    "_id" : "String",
    "client_name" : "String",
    "client_lastname" : "String",
    "document_type" : "String",
    "document" : "String",
    "age" : "Number",
    "email" : "String",
    "driver_license" : "Number",
    "advisor" : "String(_idAdvisor)"

}
```

```
// dealership
{
    "_id" : "String",
    "dealership_name": "String",
    "insurance" : [{
    "id_insrance" : "String",
    "vehicle_insurer" : "String"
    }]
}

//branch
{
    "_id" : "String",
    "branch_name" : "String",
    "address" : "String(_idAdress)",
    "dealership" : "String(_isDealership)"
}

//vehicle_brand
{
    "_id" : "String",
    "vehicle_brand" : "String",
    "original_country" : "String",
    "dealership" : "String(_idDealership)"
}

//vehicle_color
{
    "_id" : "String",
    "vehicle_color" : "String"
}
```

```
//vehicle_type
{
    "_id" : "String",
    "vehicle_type" : "String",
    "model" : "Number",
    "chasis_vehicle" : "String",
    "vehicle_brand" : "String(_idVehicle_brand)",
    "vehicle_color" : "String(_idVehicle_color)"
}

//purchase_receipt
{
    "_id" : "String",
    "purchase_receipt_date" : "Date",
    "unit_price" : "Number",
    "tuition_value" : "Number",
    "total" : "Number",
    "payment_methods" : "String",
    "vehicle_type" : "String(_idVehicle_type)",
    "advisor" : "String(_idAdvisor)",
    "cliet_p" : "String(_idClient_p)"
}



//insurance_types
{
    "_id" : "Stirng",
    "insurance_types" : "String",
    "inception_date" : "Date",
    "vehicle_insurer" : "String(_idVehicle_insurer",
    "vehicle_type" : "String(_idVehicle_type)"
}
```

b. Realiza las siguientes consultas:

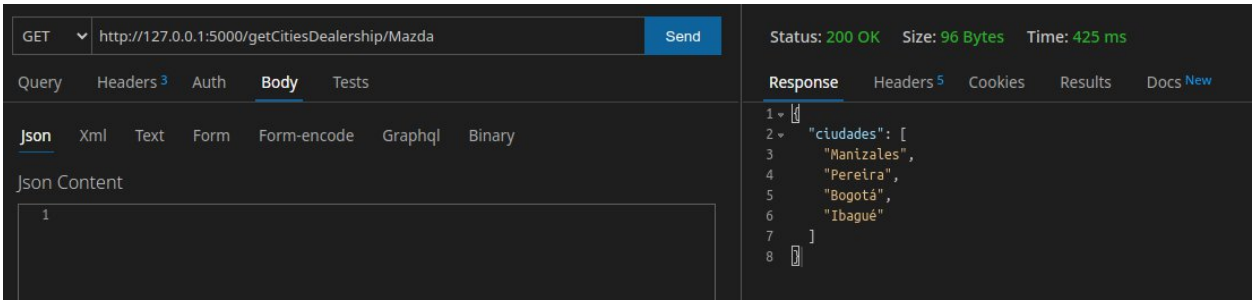
i. ¿En qué ciudades hay sucursales de la concesionaria Mazda?

```python
# retorna las ciudades en las que está precente la concesionaria dada
@app.route('/getCitiesDealership/<dealership>', methods=['GET'])
def getCitiesDealership(dealership):
    id = dealership_name('Mazda')
    filtradas = branch_in_dealership(id)
    cities = city_branch(filtradas)
    return { 'ciudades' : cities}

#retorna el id de la concecionaria dado su nombre
def dealership_name(name_dealership):
    res = mongo.db.dealership.find()
    response_string = json_util.dumps(res)
    response_json = json.loads(response_string)
    for n in response_json:
        if n['dealership_name'] == name_dealership:
            return n['_id']['$oid']

#retorna las sedes de una concecionaria dado su id
def branch_in_dealership(id_dealership):
    res = mongo.db.branch.find()
    response_string = json_util.dumps(res)
    response_json = json.loads(response_string)
    final = []
    for n in response_json:
        if (n['dealership'] == id_dealership):
            final.append(n)
            None
    return final
```

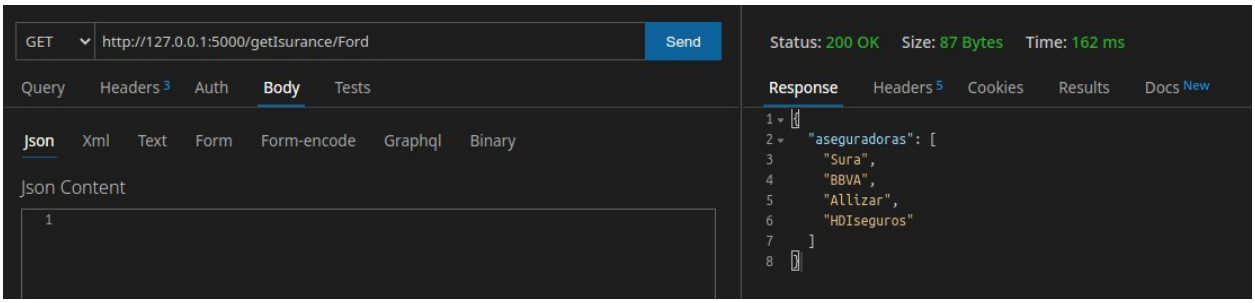
```python
#retorna lac ciudades de las sedes dadas
def city_branch(branch):
    res = mongo.db.address.find()
    response_string = json_util.dumps(res)
    response_json = json.loads(response_string)
    final = []
    for address in response_json:
        for bran in branch:
            if bran['address'] == address['_id']['$oid'] :
                final.append(address['city']['city_name'])
    return final
```



ii. ¿Cuáles son las aseguradoras que tienen convenio con Ford?

```python
# retorna las ciudades en las que está precente la concesionaria dada
@app.route('/getIsurance/<namedealership>', methods=['GET'])
def getIsurance(namedealership):
    insures = ids_insurance(namedealership)
    return { 'aseguradoras' : insures}

#retorna aseguradoras asociadas a una concesionaria
def ids_insurance(dealership_name):
    res = mongo.db.dealership.find()
    response_string = json_util.dumps(res)
    response_json = json.loads(response_string)
    final = []
    for dealership in response_json:
        if dealership['dealership_name'] == dealership_name :
            for insurance in dealership['insurance']:
                final.append(insurance['vehicle_insurer'])
    return final
```



iii. ¿Cuáles son los clientes que atiende Jacinto, asesor de la concesionaria de Mazda de la sucursal Colautos ubicada en la ciudad de Manizales?

```python
# retorna las los clientes de un advisor dado
@app.route('/getClients/<nameAdvisor>', methods=['GET'])
def getClients(nameAdvisor):
    insures = clientsAdvisor(nameAdvisor)
    return { 'clientes' : insures}


#retorna el id del asesor dado su nombre
def id_advisor(advisor_name):
    res = mongo.db.advisor.find()
    response_string = json_util.dumps(res)
    response_json = json.loads(response_string)
    for n in response_json:
        if n['advisor_name'] == advisor_name:
            return n['_id']['$oid']

#retorna los cliente de un asesor dado
def clientsAdvisor(advisor):
    advisor = id_advisor(advisor)
    res = mongo.db.client_p.find()
    response_string = json_util.dumps(res)
    response_json = json.loads(response_string)
    final = []
    for n in response_json:
        if (n['advisor'] == advisor):
            final.append(n)
            None
    return final
```