SQL SERVER

1.Creamos la base de datos

```
create database BdEjemplo03
     go
     use BdEjemplo03
      -- TABLE: Persona
    □ CREATE TABLE Persona (
      dni char(8) NOT NULL,
      CodigoHospital int not NULL,
      apellidos varchar(50) NOT NULL,
      funcion varchar(50) not null,
      salario decimal(8,2) not null,
      localidad varchar(50) null,
      PRIMARY KEY (dni)
     );
     use bdEjemplo03;
    insert into persona values
      ('98766512', 1, 'Fuentes Bermejo Carlos','Director', 8200, 'Lima'),
      ('98768769', 2, 'Gonzales Marin Alicia', 'Conserje', 1500, 'Lima'),
      ('43258769', 1, 'Castillo Montes Pedro', 'Medico', 5900, 'Huancayo'),
       ('44457769', 2, 'Ruiz Hernandez Caridad', 'Medico', 5700, 'Huancayo'),
    ('42348369', 2, 'Serrano Diaz Alejandro', 'Director', 8000, 'Huancayo'), ('44237669', 1, 'Martinez Molina Andres', 'Medico', 5400, 'Lima'), ('44236889', 1, 'Jimenes Jimenes Dolores', 'Conserje', 1400, 'Lima');
100 % + 4
Messages
```

Commands completed successfully.

Completion time: 2024-10-03T10:17:56.0722979-05:00

2.Usamos la base de datos

Completion time: 2024-10-03T10:18:28.6521658-05:00

```
create database BdEjemplo03
       go
use BdEjemplo03
       go
-- TABLE: Persona
    □CREATE TABLE Persona (
         dni char(8) NOT NULL,
         CodigoHospital int not NULL,
          apellidos varchar(50) NOT NULL,
         funcion varchar(50) not null,
         salario decimal(8,2) not null,
         localidad varchar(50) null,
         PRIMARY KEY (dni)
       use bdEjemplo03;
    insert into persona values
        insert into persona values
('98766512', 1, 'Fuentes Bermejo Carlos','Director', 8200, 'Lima'),
('98768769', 2, 'Gonzales Marin Alicia','Conserje', 1500, 'Lima'),
('43258769', 1, 'Castillo Montes Pedro','Medico', 5900, 'Huancayo'),
('44457769', 2, 'Ruiz Hernandez Caridad','Medico', 5700, 'Huancayo'),
('42348369', 2, 'Serrano Diaz Alejandro','Director', 8000, 'Huancayo'),
('44237669', 1, 'Martinez Molina Andres','Medico', 5400, 'Lima'),
('44236889', 1, 'Jimenes Jimenes Dolores','Conserje', 1400, 'Lima');
00 % ▼ ∢
Messages
   Commands completed successfully.
```

3.Creamos la tabla persona

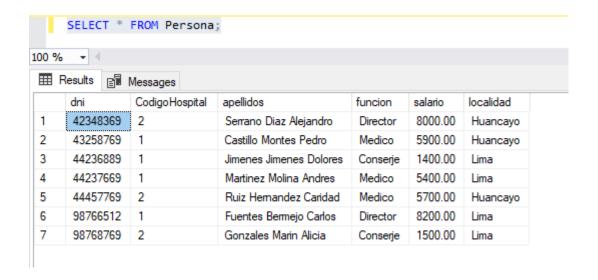
```
create database BdEjemplo03
   use BdEjemplo03
    -- TABLE: Persona
  □ CREATE TABLE Persona (
     dni char(8) NOT NULL,
     CodigoHospital int not NULL,
     apellidos varchar(50) NOT NULL,
     funcion varchar(50) not null,
     salario decimal(8,2) not null,
     localidad varchar(50) null,
     PRIMARY KEY (dni)
    );
   use bdEjemplo03;
  insert into persona values
     ('98766512', 1, 'Fuentes Bermejo Carlos', 'Director', 8200, 'Lima'),
     ('98768769', 2, 'Gonzales Marin Alicia', 'Conserje', 1500, 'Lima'),
     ('43258769', 1, 'Castillo Montes Pedro', 'Medico', 5900, 'Huancayo'), ('44457769', 2, 'Ruiz Hernandez Caridad', 'Medico', 5700, 'Huancayo'),
     ('42348369', 2, 'Serrano Diaz Alejandro','Director', 8000, 'Huancayo'),
     ('44237669', 1, 'Martinez Molina Andres', 'Medico', 5400, 'Lima'),
    ('44236889', 1, 'Jimenes Jimenes Dolores', 'Conserje', 1400, 'Lima');
0 % + <
Messages
 Commands completed successfully.
```

Completion time: 2024-10-03T10:19:35.3224431-05:00

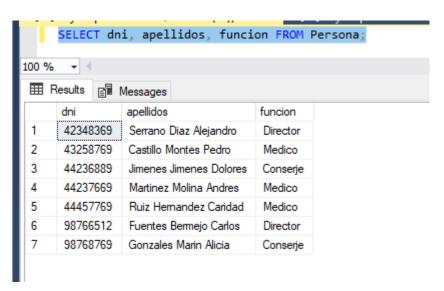
4. Agregamos datos a la BD

```
SQLQuery1.sql - LA...PC11\USER 17 (69))* → ×
      create database BdEjemplo03
      use BdEjemplo03
      -- TABLE: Persona
    □ CREATE TABLE Persona (
       dni char(8) NOT NULL,
       CodigoHospital int not NULL,
       apellidos varchar(50) NOT NULL,
       funcion varchar(50) not null,
       salario decimal(8,2) not null,
       localidad varchar(50) null,
       PRIMARY KEY (dni)
      );
      use bdEjemplo03;
    insert into persona values
       ('98766512', 1, 'Fuentes Bermejo Carlos', 'Director', 8200, 'Lima'),
       ('98768769', 2, 'Gonzales Marin Alicia','Conserje', 1500, 'Lima'),
       ('43258769', 1, 'Castillo Montes Pedro', 'Medico', 5900, 'Huancayo'), ('44457769', 2, 'Ruiz Hernandez Caridad', 'Medico', 5700, 'Huancayo'), ('42348369', 2, 'Serrano Diaz Alejandro', 'Director', 8000, 'Huancayo'), ('44237669', 1, 'Martinez Molina Andres', 'Medico', 5400, 'Lima'),
        ('44236889', 1, 'Jimenes Jimenes Dolores','Conserje', 1400, 'Lima');
100 % ▼ ◀
Messages
    (7 rows affected)
    Completion time: 2024-10-03T10:20:07.7723839-05:00
```

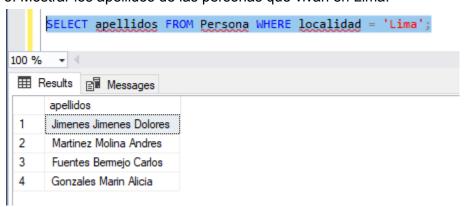
- 5. Hacemos las consultas
- 1. Mostrar todos los datos de todas las personas.



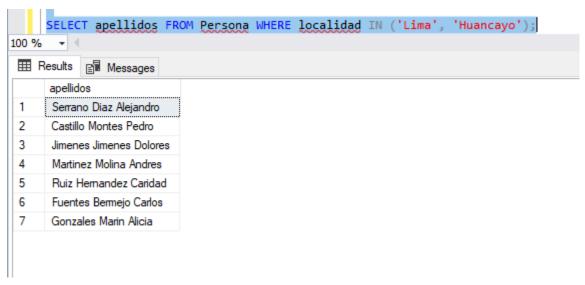
2. Obtén el DNI, apellidos y función de todas las personas



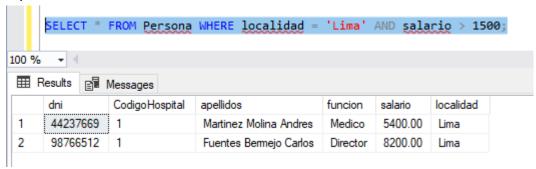
3. Mostrar los apellidos de las personas que vivan en Lima.



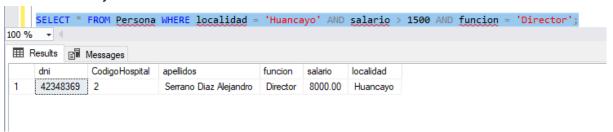
4. Mostrar los apellidos de las personas que vivan en lima o Huancayo



5. Seleccionar los datos de aquellas personas que vivan en Lima y tengan un salario superior a los 1500 soles.



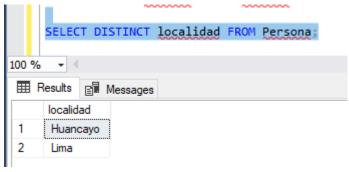
6. Mostrar los datos de las personas que vivan en Huancayo, tengan un salario superior a los 1500 euros y sean directores.



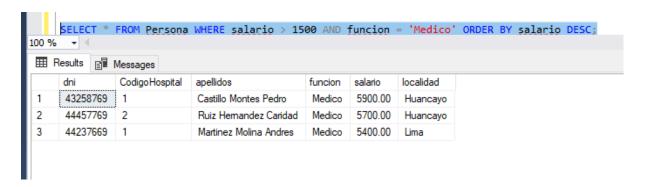
7. Mostrar los datos de las personas cuya función sea Médico ordenados por apellidos descendentemente.



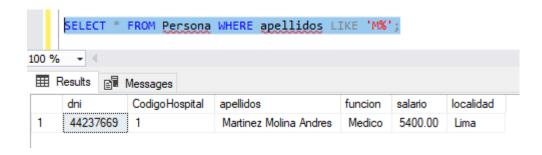
8. Mostrar los datos de todas las localidades que hay en la tabla personas sin repeticiones (debes emplear la cláusula DISTINCT)



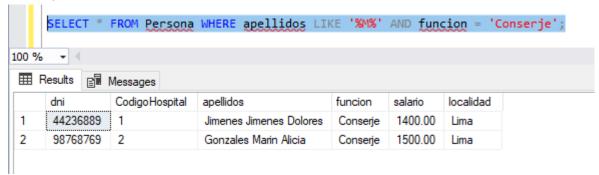
9. Mostrar los datos de las personas que tengan un salario superior a 1500 soles y sean médicos. Ordenar la salida por salario descendentemente.



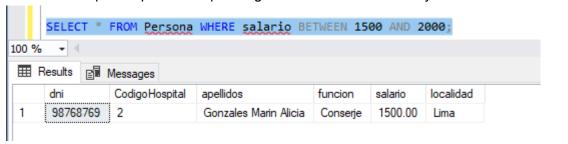
10. Seleccionar aquellas personas cuyo apellido comience por M.



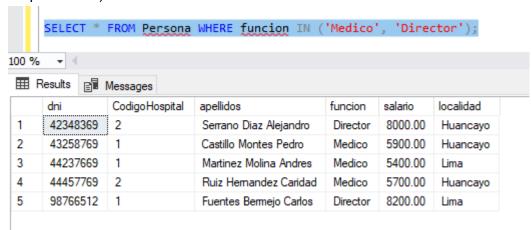
11. Mostrar los datos de las personas que tengan una M en el apellido y cuya función sea Conserje



12. Mostrar aquellas personas que tengan un salario entre 1500 y 200 soles.



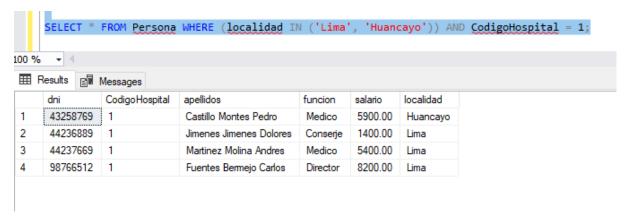
13. Seleccionar los datos de aquellas personas cuya función sea Médico o director (utilizar el operador IN)



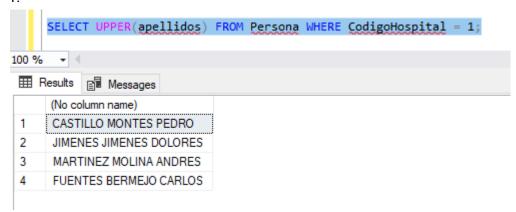
14. Obtener los datos de aquellas personas cuya función no sea Conserje (utilizar el operador NOT IN) y tengan un salario superior a los 1500 soles, ordenados por apellido descendentemente.



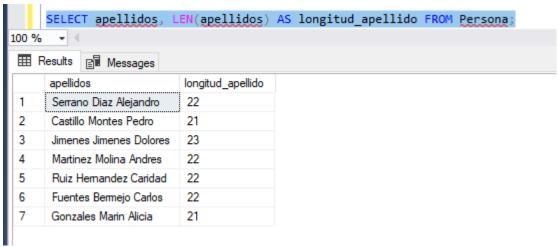
15. Mostrar los datos de las personas que sean de Lima o Huancayo y que pertenezcan al hospital número 1.



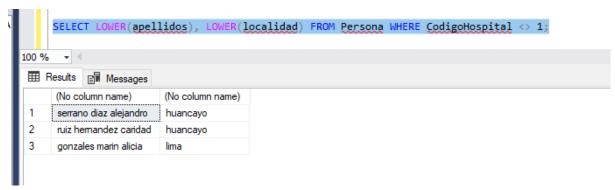
16. Obtén los apellidos en mayúsculas de las personas que trabajen en el hospital número 1.



17. Con una consulta devuelve los apellidos de todas las personas. Al lado debe aparecer la longitud de cada apellido.



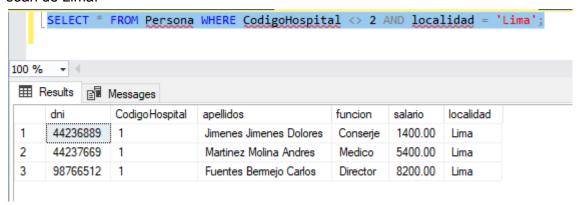
18. Obtener los apellidos y localidad en minúscula de todas aquellas personas que no trabajen en el hospital número 1.



19. Obtener los datos de las personas que trabajen en los hospitales 1 ó 2 y tengan un salario superior a 1500 soles.



20. Visualizar los datos de aquellas personas que no trabajen en el hospital número 2 y que sean de Lima.



POSTGRESQL

1.Creamos la base de datos

```
-- Create database
CREATE DATABASE BdEjemplo03;
```

2.Creamos la tabla persona

```
Create Table: Persona
CREATE TABLE Persona (
   dni CHAR(8) NOT NULL,
   CodigoHospital INT NOT NULL,
   apellidos VARCHAR(50) NOT NULL,
   funcion VARCHAR(50) NOT NULL,
   salario DECIMAL(8,2) NOT NULL,
   localidad VARCHAR(50),
   PRIMARY KEY (dni)
);
```

3.Insertamos datos a la tabla Persona

```
-- Insert data into Persona

INSERT INTO Persona (dni, CodigoHospital, apellidos, funcion, salario, localidad) VALUES

('98766512', 1, 'Fuentes Bermejo Carlos', 'Director', 8200, 'Lima'),

('98768769', 2, 'Gonzales Marin Alicia', 'Conserje', 1500, 'Lima'),

('43258769', 1, 'Castillo Montes Pedro', 'Medico', 5900, 'Huancayo'),

('44457769', 2, 'Ruiz Hernandez Caridad', 'Medico', 5700, 'Huancayo'),

('42348369', 2, 'Serrano Diaz Alejandro', 'Director', 8000, 'Huancayo'),

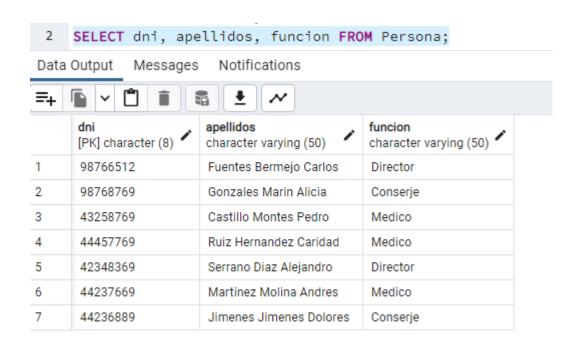
('44237669', 1, 'Martinez Molina Andres', 'Medico', 5400, 'Lima'),

('44236889', 1, 'Jimenes Jimenes Dolores', 'Conserje', 1400, 'Lima');
```

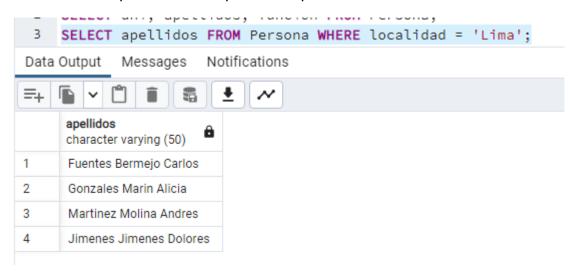
- 4. Realizamos las consultas
- 1. Mostrar todos los datos de todas las personas.



2. Obtén el DNI, apellidos y función de todas las personas.



3. Mostrar los apellidos de las personas que vivan en Lima.



4. Mostrar los apellidos de las personas que vivan en lima o Huancayo



5. Seleccionar los datos de aquellas personas que vivan en Lima y tengan un salario superior a los 1500 soles.



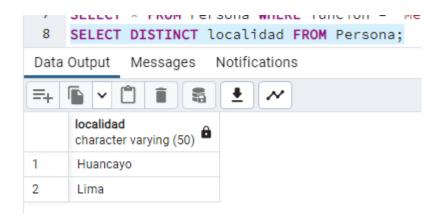
6. Mostrar los datos de las personas que vivan en Huancayo, tengan un salario superior a los 1500 euros y sean directores.



7. Mostrar los datos de las personas cuya función sea Médico ordenados por apellidos descendentemente.



8. Mostrar los datos de todas las localidades que hay en la tabla personas sin repeticiones (debes emplear la cláusula DISTINCT)



9. Mostrar los datos de las personas que tengan un salario superior a 1500 soles y sean médicos. Ordenar la salida por salario descendentemente.



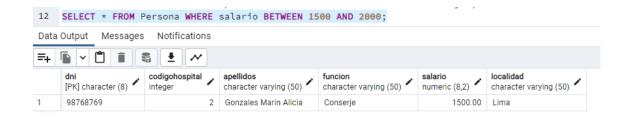
10. Seleccionar aquellas personas cuyo apellido comience por M.



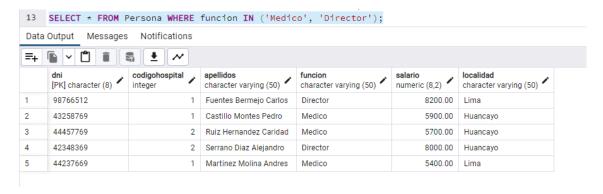
11. Mostrar los datos de las personas que tengan una M en el apellido y cuya función sea Conserje



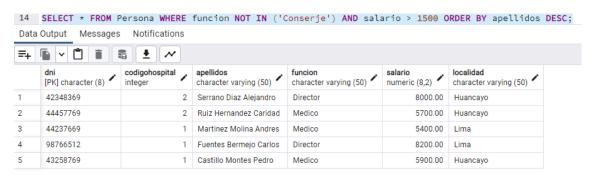
12. Mostrar aquellas personas que tengan un salario entre 1500 y 200 soles.



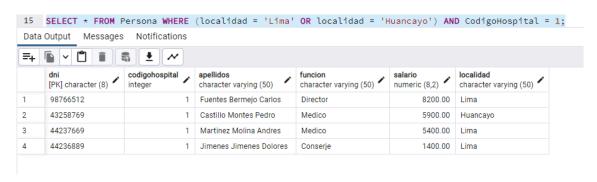
13. Seleccionar los datos de aquellas personas cuya función sea Médico o director (utilizar el operador IN)



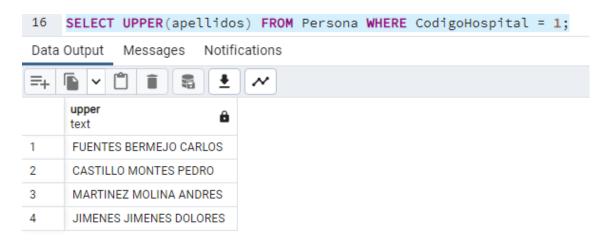
14. Obtener los datos de aquellas personas cuya función no sea Conserje (utilizar el operador NOT IN) y tengan un salario superior a los 1500 soles, ordenados por apellido descendentemente.



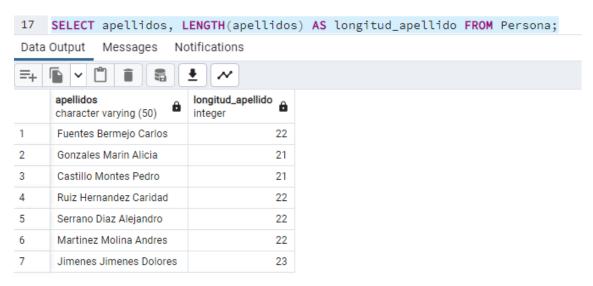
15. Mostrar los datos de las personas que sean de Lima o Huancayo y que pertenezcan al hospital número 1.



16. Obtén los apellidos en mayúsculas de las personas que trabajen en el hospital número 1.



17. Con una consulta devuelve los apellidos de todas las personas. Al lado debe aparecer la longitud de cada apellido.



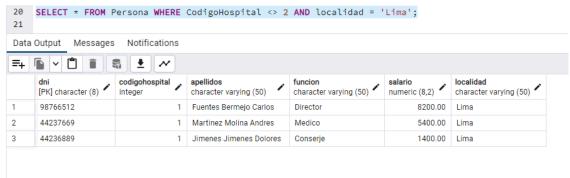
18. Obtener los apellidos y localidad en minúscula de todas aquellas personas que no trabajen en el hospital número 1.



19. Obtener los datos de las personas que trabajen en los hospitales 1 ó 2 y tengan un salario superior a 1500 soles.



20. Visualizar los datos de aquellas personas que no trabajen en el hospital número 2 y que sean de Lima.



El SQLSERVER y PostgreSQl tienen sintaxis en común y diferentes lo cual lo hace intuitivo para crear una base de datos.