GUÍA DE LABORATORIO N°11

Create View y diseño de los modelos conceptual, lógico y físico

1. DATOS INFORMATIVOS:

1.1. Nombre de la asignatura :Modelado de Bases de Datos

1.2. Semestre académico :2025 - I

1.3. Ciclo de estudios :5to

1.4. Área curricular :EBE (Estudios Básicos Específicos)

2. COMPETENCIAS A CONSEGUIR:

2.1. Conoce y aplica el concepto de vistas en Oracle Live SQL y también, en SQL Server.

3. FUNDAMENTOS TEÓRICOS – CONCEPTO:

Las vistas tienen una estructura semejante a las tablas pues contienen información resultante de una consulta a una tabla o más tablas. Las vistas son como fotos de la información de la base de datos en un instante determinado. Sirven para mostrar información rápidamente, en base a consultas ya conocidas, las cuales han unido varias tablas. Las vistas sirven para obtener información de la base de datos sin manipular las tablas.
La sintaxis es:

```
CREATE VIEW vista (campoA, campoB…)
  AS SELECT campoA, campoB,…
        FROM tabla1
```

EXPLICACIÓN:

En esta guía se crearán vistas en la base de datos “baseDiagnosticos”.

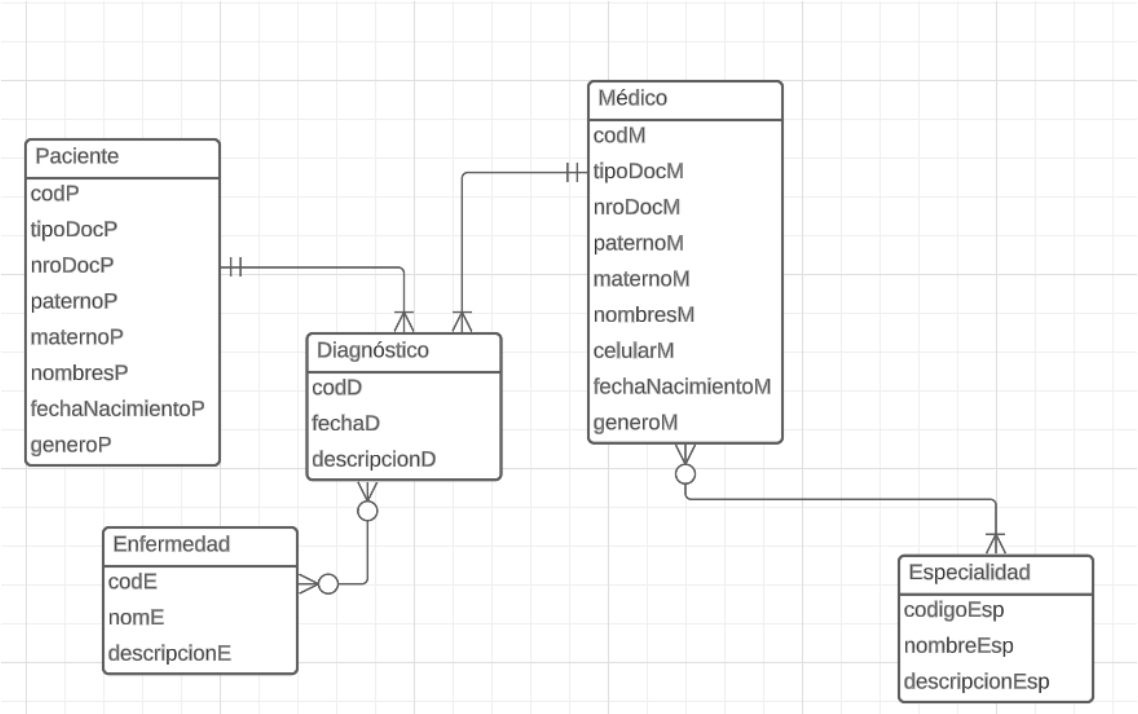
4. ACTIVIDADES - DEMOSTRACIÓN

4.1. Se tiene el siguiente listado de requerimientos:

En la Clínica TodoSalud, se requiere almacenar información sobre los diagnósticos que emiten los médicos. Se consideran los siguientes supuestos:

A. Un paciente debe tener los siguientes atributos: Código del paciente, nombres y otros datos personales.
B. Un médico también tiene código de médico y sus propios atributos, pero el médico podría tener más de una especialidad por lo tanto debe existir una entidad que contenga todas las especialidades que la Clínica brinda a sus clientes. Es importante saber la fecha en la que el médico obtuvo una determinada especialidad.
C. El diagnóstico que emite el médico podría estar relacionado con dos enfermedades.

4.2. A partir del listado de requerimientos, se obtiene el siguiente modelo conceptual:



4.3. A partir del modelo conceptual, se obtiene el siguiente modelo lógico:

1. tPaciente(**codP**, tipoDocP, nroDocP, paternoP, maternoP, nombresP, fechaNacimientoP, generoP)

2. tMedico(**codM**, tipoDocM, nroDocM, paternoM, maternoM, nombresM, celularM, fechaNacimientoM, generoM)

3. tEspecialidad(**codEsp**, nombreEsp, descripcionEsp)
4. tEnfermedad(**codE**, nombreE, descripcionE)
5. tDiagnostico(**codD**, fechaHoraD, descripcionD, codP, codM)
6. tEspecialidadDelMedico(**codEspM**, codEsp, codM, fechaDeObtencionDeLaEspecialidadEspM)
7. tEnfermedadEnElDiagnostico(**codED**, codE, codD)

4.4. Usando el Oracle LiveSQL y a partir del modelo lógico, obtenemos el siguiente script del Modelo Físico:

Primeramente, vamos a eliminar las tablas existentes como resultado de la guía 10. Se usará la orden DROP TABLE <nobre de la tabla> CASCADE CONSTRAINTS; para eliminar todas las tablas existentes sin preocuparnos por las claves foráneas que ligan las tablas de la baseRecursiva.

```
drop table tEscuela cascade constraints;

drop table tplandeestudios cascade constraints;

drop table tasignatura cascade constraints;

drop table TPRERREQUISITO;
```

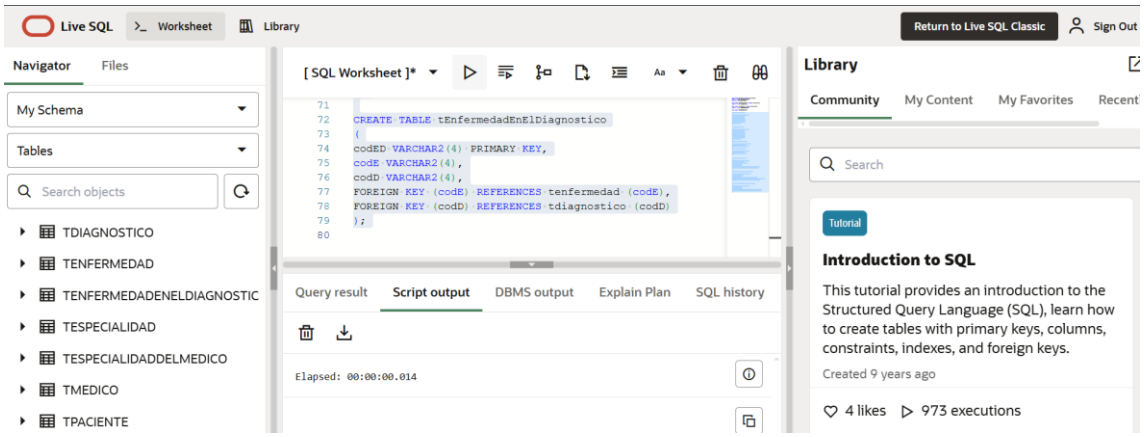
Ahora, recién vamos a crear la baseDiagnosticos:

```
CREATE TABLE tPaciente

(

codP VARCHAR2(4) PRIMARY KEY,

tipoDocP VARCHAR2(15),

nroDocP VARCHAR2(12),
```

```
paternoP VARCHAR2(50),
maternoP VARCHAR2(50),
nombresP VARCHAR2(50),
fechaNacimientoP DATE,
generoP CHAR(1)
);

CREATE TABLE tMedico
(
codM VARCHAR2(4) PRIMARY KEY,
tipoDocM VARCHAR2(15),
nroDocM VARCHAR2(12),
paternoM VARCHAR2(50),
maternoM VARCHAR2(50),
nombresM VARCHAR2(50),
celularM VARCHAR2(12),
fechaNacimientoM DATE,
generoM CHAR(1)
);
CREATE TABLE tEspecialidad
(
codEsp VARCHAR2(4) PRIMARY KEY,
nombreEsp VARCHAR2(70),
descripcionEsp VARCHAR2(100)
```

```
);
CREATE TABLE tEnfermedad
(
codE VARCHAR2(4) PRIMARY KEY,
nombreE VARCHAR2(50),
descricionE VARCHAR2(100)
);


CREATE TABLE tDiagnostico
(
codD VARCHAR2(4) PRIMARY KEY,
fechaHoraD TIMESTAMP,
descripcionD VARCHAR2(100),
codP VARCHAR2(4),
codM VARCHAR2(4),
FOREIGN KEY (codP) REFERENCES tPaciente (codP),
FOREIGN KEY (codM) REFERENCES tMedico (codM)
);

CREATE TABLE tEspecialidadDelMedico
(
codEspM VARCHAR2(4) PRIMARY KEY,
codEsp VARCHAR2(4),
```

```
codM VARCHAR2(4),
fechaDeObtencionDeLaEspecialidadEspM DATE,
FOREIGN KEY (codEsp) REFERENCES tespecialidad (codEsp),
FOREIGN KEY (codM) REFERENCES tMedico (codM)
);

CREATE TABLE tEnfermedadEnElDiagnostico
(
codED VARCHAR2(4) PRIMARY KEY,
codE VARCHAR2(4),
codD VARCHAR2(4),
FOREIGN KEY (codE) REFERENCES tenfermedad (codE),
FOREIGN KEY (codD) REFERENCES tdiagnostico (codD)
);
```



Ahora insertamos información:

APC-MNP-HEH

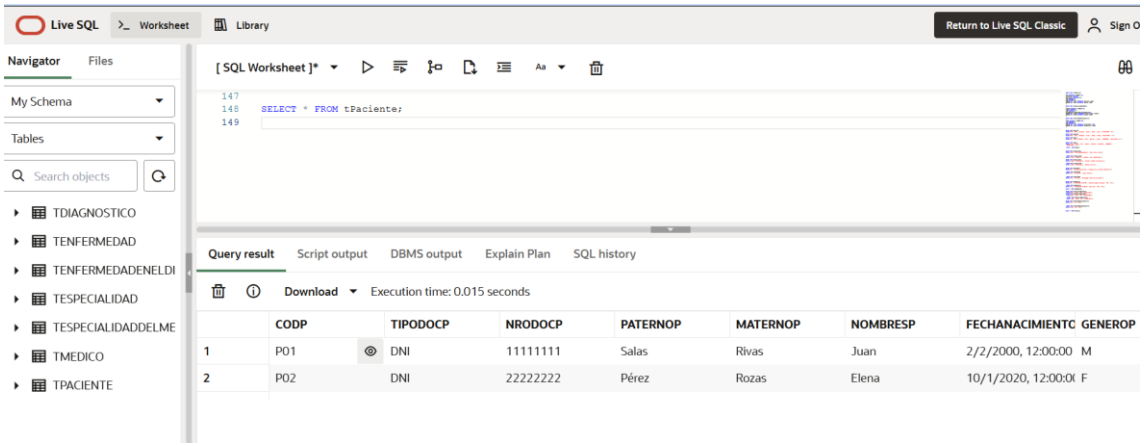
Insertamos dos médicos: Javier y Alejandra; dos pacientes: Juan y Elena; cuatro especialidades: Otorrinolaringología, Pediatría, Cardiología y Odontología; tres enfermedades: Alergia respiratoria, sinusitis y artritis; dos diagnósticos: "Sinusitis aguda y alergia" y "niña sana". El médico Javier obtuvo su especialidad de Otorrinolaringología el 12 de diciembre del 2010, Alejandra se graduó como Pediatra el 02 de febrero del 2009 y también se graduó como Cardióloga el 21 de noviembre del 2019. Asimismo, es necesario insertar información que registre que el diagnóstico de "Sinusitis aguda y alergia" está relacionado con las enfermedades: "Sinusitis" y "Alergia respiratoria".

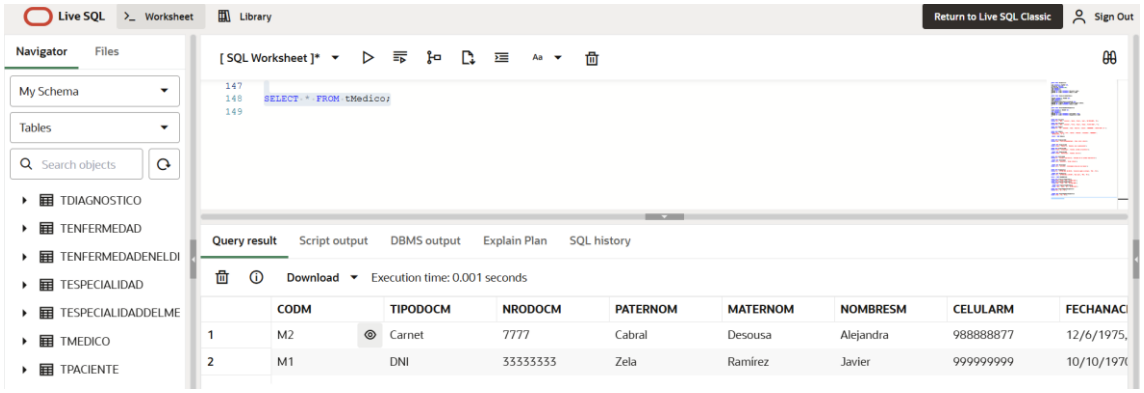
```
INSERT INTO tpaciente
VALUES('P01', 'DNI', '11111111', 'Salas', 'Rivas', 'Juan', '02-FEB-2000', 'M');

INSERT INTO tpaciente
VALUES('P02', 'DNI', '22222222', 'Pérez', 'Rozas', 'Elena', '01-OCT-2020', 'F');

INSERT INTO tmedico
VALUES('M1', 'DNI','33333333', 'Zela', 'Ramírez', 'Javier', '999999999', '10-OCT-1970','M' );

INSERT INTO tmedico
VALUES ('M2', 'Carnet','7777', 'Cabral', 'Desousa', 'Alejandra', '988888877',
 '06-DEC-1975', 'F' );

INSERT INTO tespecialidad
VALUES('Esp1', 'Otorrinolagingología', 'Ojos, nariz y boca');
```

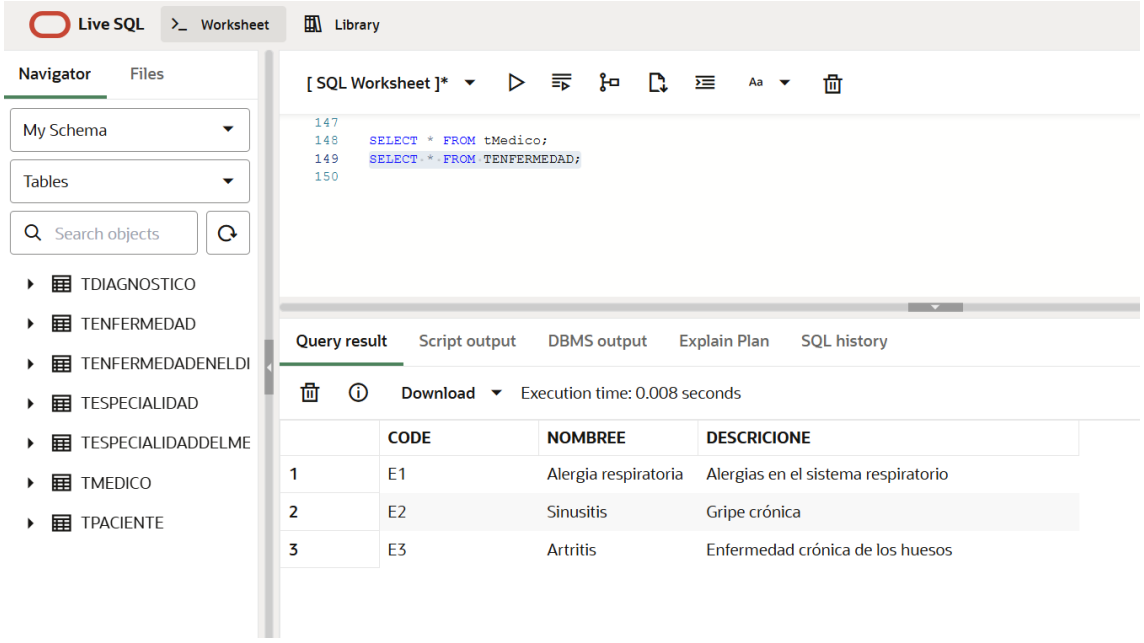
```
INSERT INTO tespecialidad
VALUES ('Esp2', 'Pediatría', 'Neonato, niño y adolescente');

INSERT INTO tespecialidad
VALUES ('Esp3', 'Cardiología', 'Corazón, sistema circulatorio');

INSERT INTO tespecialidad
VALUES ('Esp4', 'Odontología', 'Dientes, encías');

INSERT INTO tEnfermedad
VALUES('E1', 'Alergia respiratoria', 'Alergias en el sistema respiratorio');

INSERT INTO tEnfermedad
VALUES ('E2', 'Sinusitis', 'Gripe crónica');

INSERT INTO tEnfermedad
VALUES ('E3', 'Artritis', 'Enfermedad crónica de los huesos');

INSERT INTO tdiagnostico
VALUES('D1', '25-FEB-2022 10:00:00', 'Sinusitis aguda y alergia', 'P01', 'M1');
```

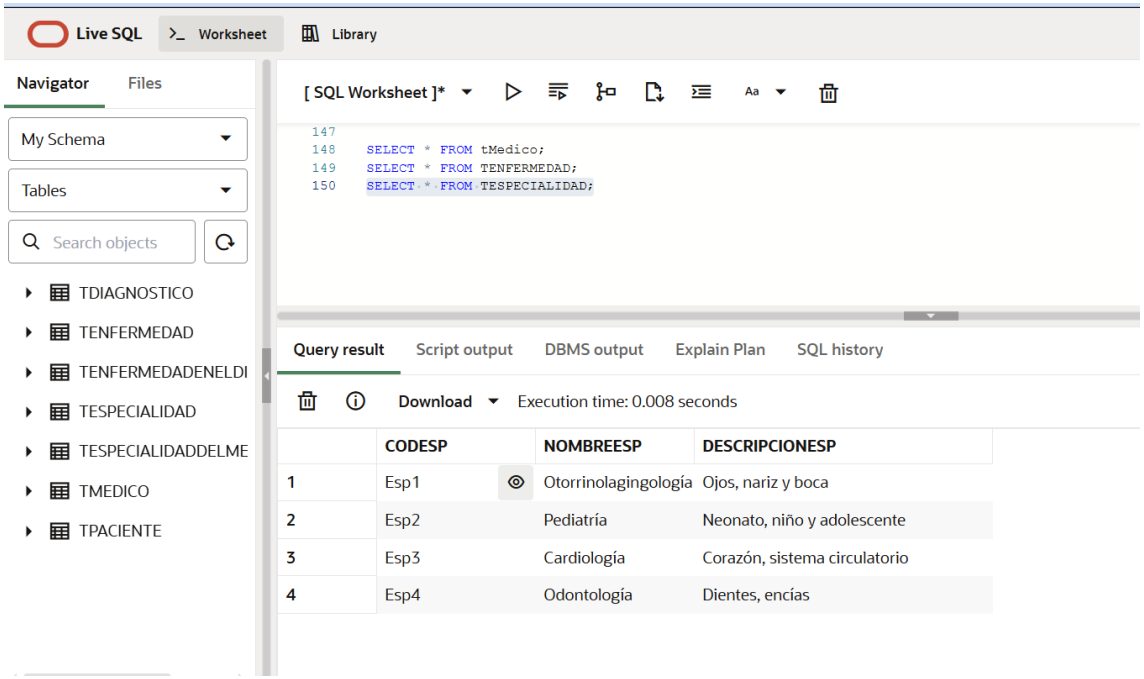
```
INSERT INTO tdiagnostico
VALUES ('D2', '28-FEB-2022 12:00:00', 'Niña sana', 'P02', 'M2');

INSERT INTO tespecialidaddelmedico
VALUES('EM1', 'Esp1', 'M1', '12-DEC-2010');
INSERT INTO tespecialidaddelmedico
VALUES('EM2', 'Esp2', 'M2', '02-FEB-2009');

INSERT INTO tespecialidaddelmedico
VALUES ('EM3', 'Esp3', 'M2', '21-NOV-2019');

INSERT INTO tenfermedadeneldiagnostico
VALUES('ED1', 'E1', 'D1');

INSERT INTO tenfermedadeneldiagnostico
VALUES ('ED2', 'E2', 'D1');
```

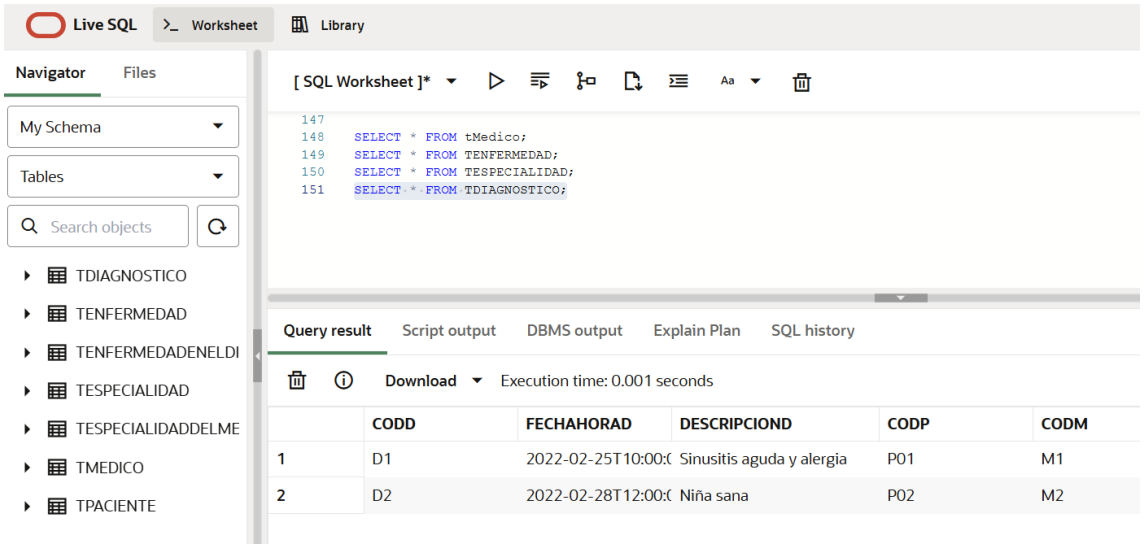
Verificamos el contenido de las tablas:

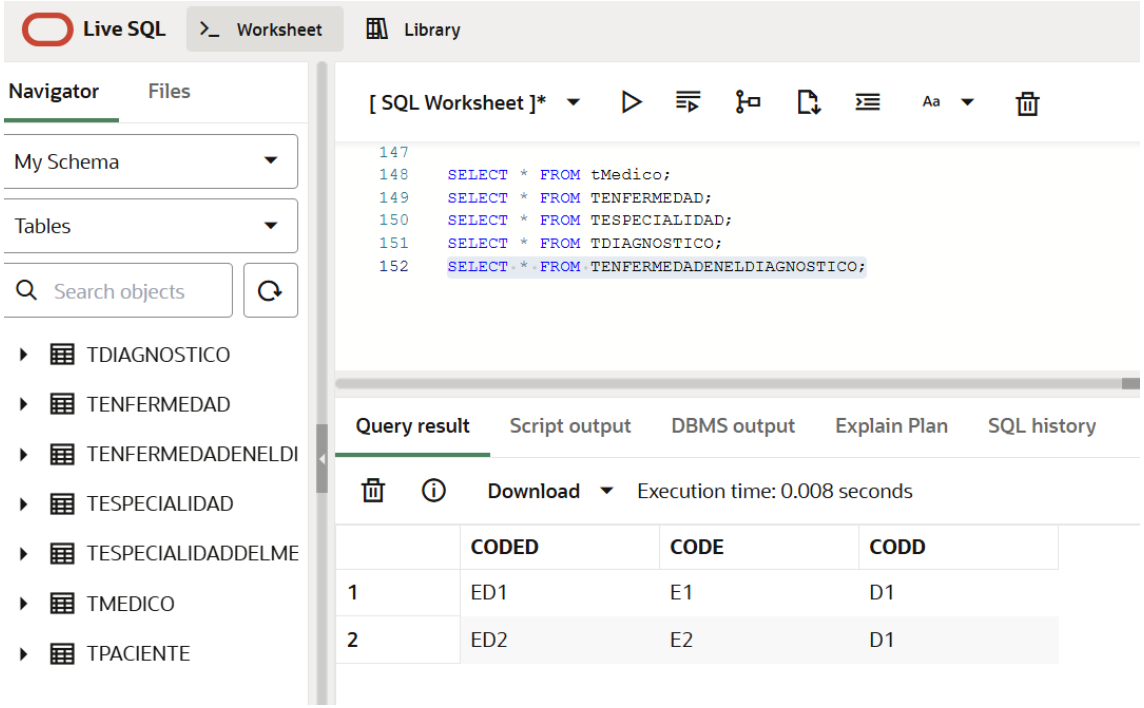

Live SQL
Worksheet
Library
Return to Live SQL Classic
Navigator
Files
My Schema
Tables
Search objects
TDIAGNOSTICO
TENFERMEDAD
TENFERMEDADENELDI
TESPECIALIDAD
TESPECIALIDADDELME
TMEDICO
TPACIENTE
[SQL Worksheet]*
147
148 SELECT * FROM tPaciente;
149
Query result
Script output
DBMS output
Explain Plan
SQL history
Download
Execution time: 0.015 seconds
CODP TIPODOCP NRODOCP PATERNOP MATERNOP NOMBRESP FECHANACIMIENTO GENEROP
1 P01 DNI 11111111 Salas Rivas Juan 2/2/2000, 12:00:00 M
2 P02 DNI 22222222 Pérez Rozas Elena 10/1/2020, 12:00:0 F

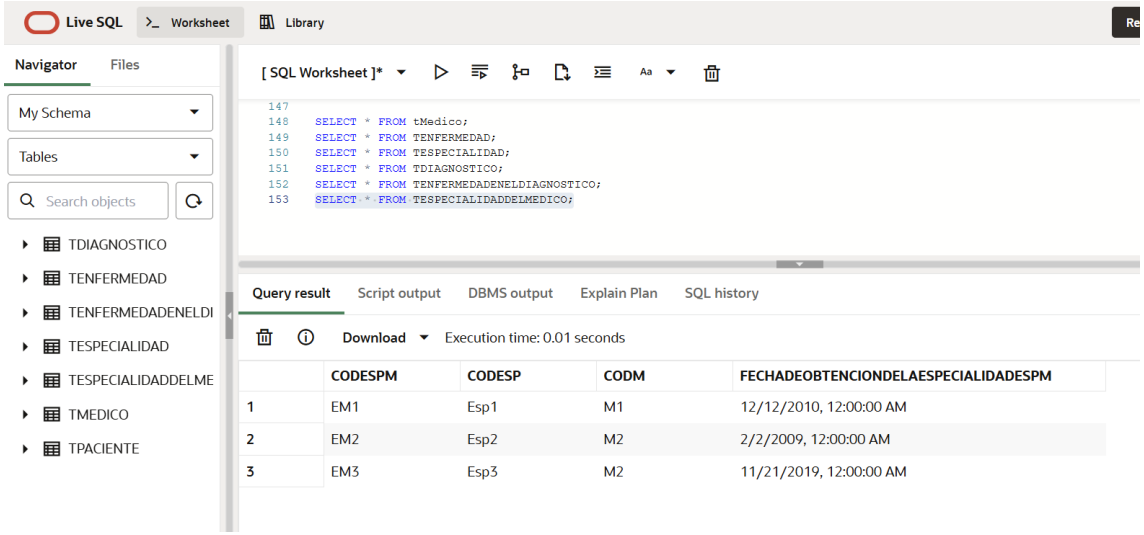

Live SQL
Worksheet
Library
Return to Live SQL Classic
Sign Out
Navigator
Files
My Schema
Tables
Search objects
TDIAGNOSTICO
TENFERMEDAD
TENFERMEDADENELDI
TESPECIALIDAD
TESPECIALIDADDELME
TMEDICO
TPACIENTE
[SQL Worksheet]*
147
148 SELECT * FROM tMedico;
149
Query result
Script output
DBMS output
Explain Plan
SQL history
Download
Execution time: 0.001 seconds
CODM TIPODOCM NRODOCM PATERNOM MATERNOM NOMBRESM CELULARM FECHANAC
1 M2 Carnet 7777 Cabral Desousa Alejandra 988888877 12/6/1975,
2 M1 DNI 33333333 Zela Ramírez Javier 999999999 10/10/197


Live SQL
Worksheet
Library
Navigator
Files
My Schema
Tables
Search objects
TDIAGNOSTICO
TENFERMEDAD
TENFERMEDADENELDI
TESPECIALIDAD
TESPECIALIDADDELME
TMEDICO
TPACIENTE
[SQL Worksheet]*
147
148 SELECT * FROM tMedico;
149 SELECT * FROM TENFERMEDAD;
150
Query result
Script output
DBMS output
Explain Plan
SQL history
Download
Execution time: 0.008 seconds
CODE
NOMBREE
DESCRICIONE
1 E1 Alergia respiratoria Alergias en el sistema respiratorio
2 E2 Sinusitis Gripe crónica
3 E3 Artritis Enfermedad crónica de los huesos


Live SQL
Worksheet
Library
Navigator
Files
My Schema
Tables
Search objects
TDIAGNOSTICO
TENFERMEDAD
TENFERMEDADENELDI
TESPECIALIDAD
TESPECIALIDADDELME
TMEDICO
TPACIENTE
[SQL Worksheet]*
147
148 SELECT * FROM tMedico;
149 SELECT * FROM TENFERMEDAD;
150 SELECT * FROM TESPECIALIDAD;
Query result
Script output
DBMS output
Explain Plan
SQL history
Download
Execution time: 0.008 seconds
CODESP
NOMBREESP
DESCRIPCIONESP
1 Esp1 Otorrinolagingología Ojos, nariz y boca
2 Esp2 Pediatría Neonato, niño y adolescente
3 Esp3 Cardiología Corazón, sistema circulatorio
4 Esp4 Odontología Dientes, encías


Live SQL
Worksheet
Library
Navigator
Files
My Schema
Tables
Search objects
TDIAGNOSTICO
TENFERMEDAD
TENFERMEDADENELDI
TESPECIALIDAD
TESPECIALIDADDELME
TMEDICO
TPACIENTE
[SQL Worksheet]*
147
148 SELECT * FROM tMedico;
149 SELECT * FROM TENFERMEDAD;
150 SELECT * FROM TESPECIALIDAD;
151 SELECT * FROM TDIAGNOSTICO;
Query result
Script output
DBMS output
Explain Plan
SQL history
Download
Execution time: 0.001 seconds
CODD
FECHAHORAD
DESCRIPCIOND
CODP
CODM
1 D1 2022-02-25T10:00:(Sinusitis aguda y alergia P01 M1
2 D2 2022-02-28T12:00:(Niña sana P02 M2


Live SQL
Worksheet
Library
Navigator
Files
My Schema
Tables
Search objects
TDIAGNOSTICO
TENFERMEDAD
TENFERMEDADENELDI
TESPECIALIDAD
TESPECIALIDADDELME
TMEDICO
TPACIENTE
[SQL Worksheet]*
147
148 SELECT * FROM tMedico;
149 SELECT * FROM TENFERMEDAD;
150 SELECT * FROM TESPECIALIDAD;
151 SELECT * FROM TDIAGNOSTICO;
152 SELECT * FROM TENFERMEDADENELDIAGNOSTICO;
Query result
Script output
DBMS output
Explain Plan
SQL history
Download
Execution time: 0.008 seconds
CODED
CODE
CODD
1 ED1 E1 D1
2 ED2 E2 D1



El modelo físico En SQL Server:

-- Primero, crear la baseDiagnosticos:

CREATE DATABASE baseDiagnosticos;

--Luego, activarla y crear la estructura de tablas:

USE baseDiagnosticos;

CREATE TABLE tPaciente

(

codP VARCHAR(4) PRIMARY KEY,

tipoDocP VARCHAR(15),

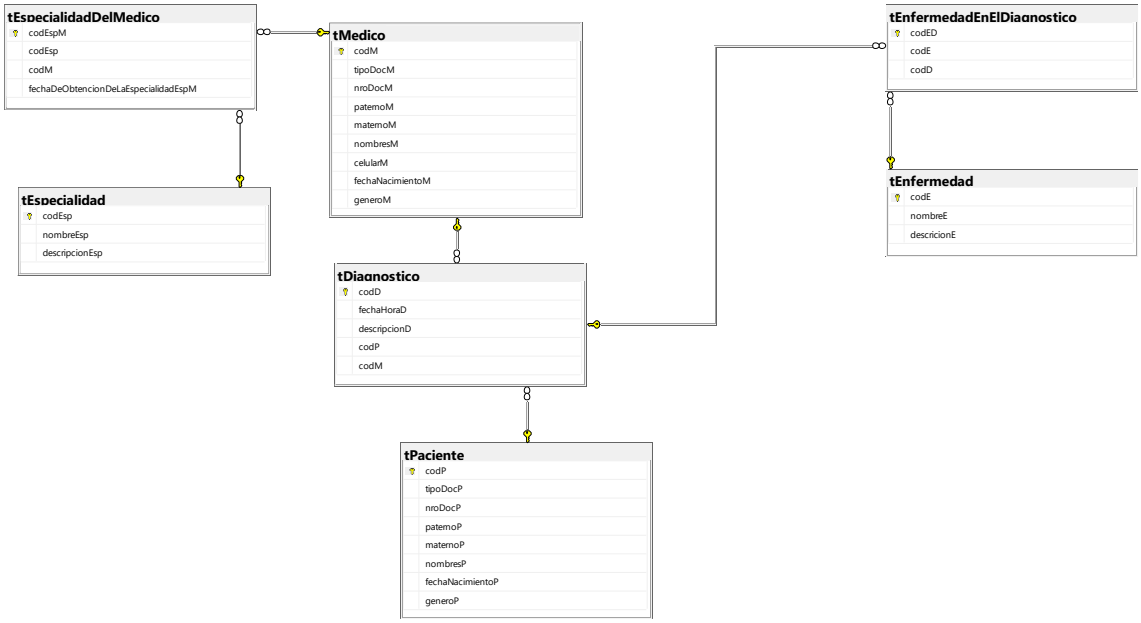
nroDocP VARCHAR(12),

```
paternoP VARCHAR(50),
maternoP VARCHAR(50),
nombresP VARCHAR(50),
fechaNacimientoP DATE,
generoP CHAR(1)
);

CREATE TABLE tMedico
(
codM VARCHAR(4) PRIMARY KEY,
tipoDocM VARCHAR(15),
nroDocM VARCHAR(12),
paternoM VARCHAR(50),
maternoM VARCHAR(50),
nombresM VARCHAR(50),
celularM VARCHAR(12),
fechaNacimientoM DATE,
generoM CHAR(1)
);
CREATE TABLE tEspecialidad
(
codEsp VARCHAR(4) PRIMARY KEY,
nombreEsp VARCHAR(70),
descripcionEsp VARCHAR(100)
```

```
);
CREATE TABLE tEnfermedad
(
codE VARCHAR(4) PRIMARY KEY,
nombreE VARCHAR(50),
descricionE VARCHAR(100)
);


CREATE TABLE tDiagnostico
(
codD VARCHAR(4) PRIMARY KEY,
fechaHoraD DATETIME,
descripcionD VARCHAR(100),
codP VARCHAR(4),
codM VARCHAR(4),
FOREIGN KEY (codP) REFERENCES tPaciente (codP),
FOREIGN KEY (codM) REFERENCES tMedico (codM)
);

CREATE TABLE tEspecialidadDelMedico
(
codEspM VARCHAR(4) PRIMARY KEY,
codEsp VARCHAR(4),
```

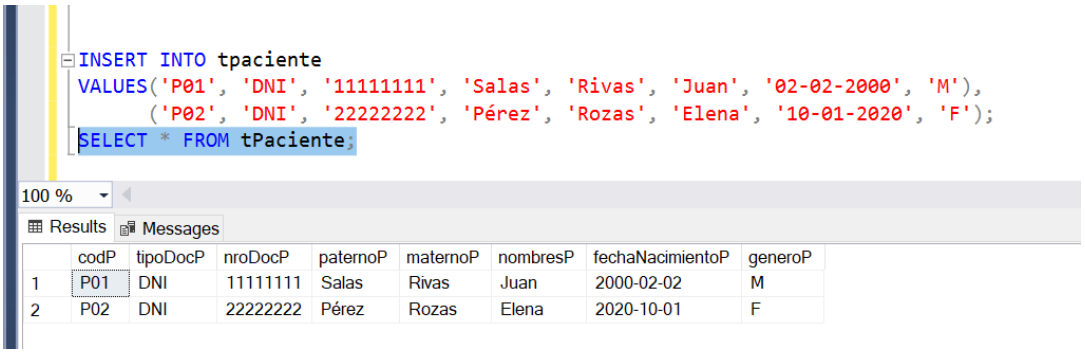
```
codM VARCHAR(4),
fechaDeObtencionDeLaEspecialidadEspM DATE,
FOREIGN KEY (codEsp) REFERENCES tespecialidad (codEsp),
FOREIGN KEY (codM) REFERENCES tMedico (codM)
);

CREATE TABLE tEnfermedadEnElDiagnostico
(
codED VARCHAR(4) PRIMARY KEY,
codE VARCHAR(4),
codD VARCHAR(4),
FOREIGN KEY (codE) REFERENCES tenfermedad (codE),
FOREIGN KEY (codD) REFERENCES tdiagnostico (codD)
);
```

Diagramar en SQL Server:



Insertamos información:

```
INSERT INTO tpaciente
VALUES('P01', 'DNI', '11111111', 'Salas', 'Rivas', 'Juan', '02-02-2000', 'M'),
    ('P02', 'DNI', '22222222', 'Pérez', 'Rozas', 'Elena', '10-01-2020', 'F');
```



```
INSERT INTO tpaciente
VALUES('P01', 'DNI', '11111111', 'Salas', 'Rivas', 'Juan', '02-02-2000', 'M'),
      ('P02', 'DNI', '22222222', 'Pérez', 'Rozas', 'Elena', '10-01-2020', 'F');
SELECT * FROM tPaciente;
```

100 %

Results | Messages

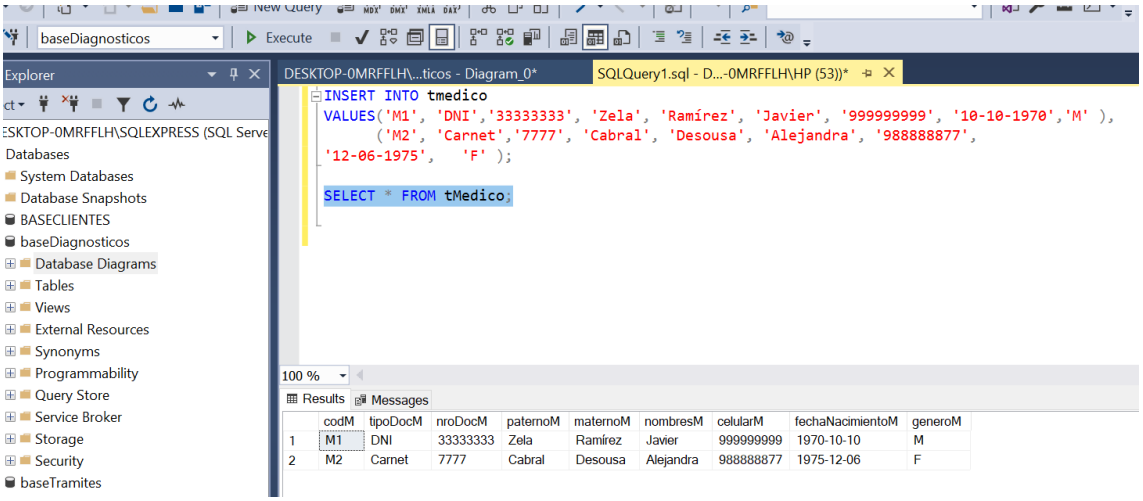
	codP	tipoDocP	nroDocP	paternoP	maternoP	nombresP	fechaNacimientoP	generoP
1	P01	DNI	11111111	Salas	Rivas	Juan	2000-02-02	M
2	P02	DNI	22222222	Pérez	Rozas	Elena	2020-10-01	F

INSERT INTO tmedico

VALUES('M1', 'DNI','33333333', 'Zela', 'Ramírez', 'Javier', '999999999', '10-10-1970','M'),

('M2', 'Carnet','7777', 'Cabral', 'Desousa', 'Alejandra', '988888877',

'12-06-1975', 'F');



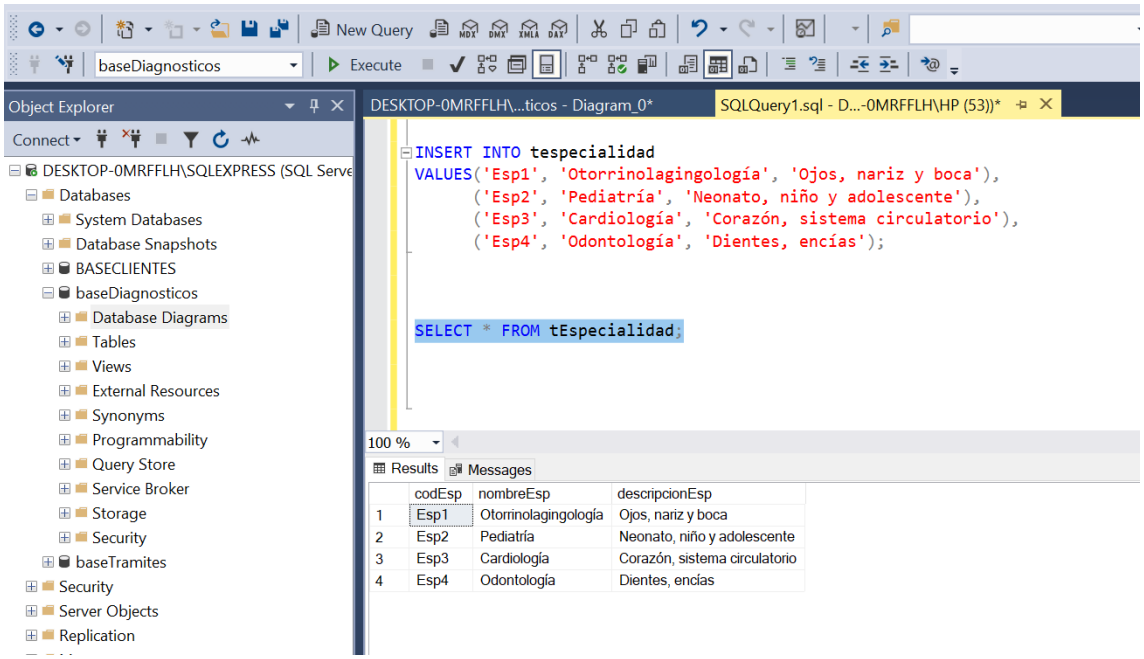
INSERT INTO tespecialidad

VALUES('Esp1', 'Otorrinolagingología', 'Ojos, nariz y boca'),

('Esp2', 'Pediatría', 'Neonato, niño y adolescente'),

('Esp3', 'Cardiología', 'Corazón, sistema circulatorio'),

('Esp4', 'Odontología', 'Dientes, encías');

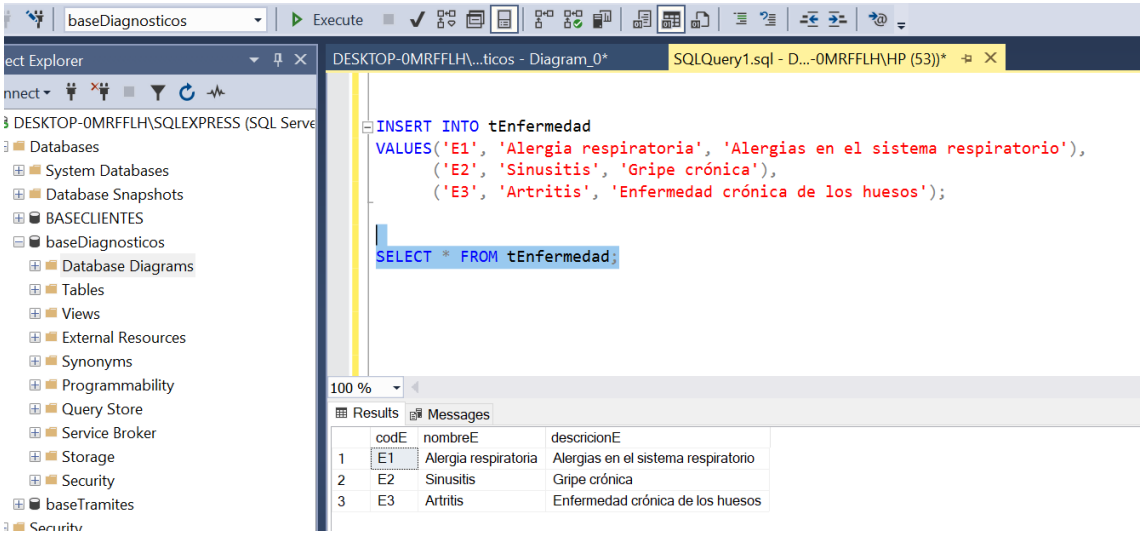


INSERT INTO tEnfermedad

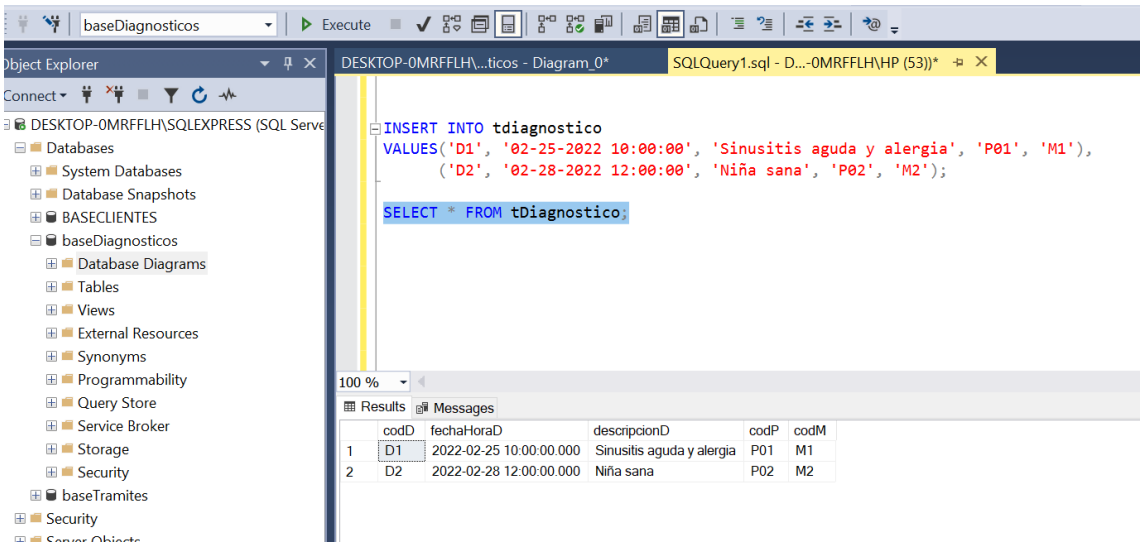
VALUES('E1', 'Alergia respiratoria', 'Alergias en el sistema respiratorio'),

('E2', 'Sinusitis', 'Gripe crónica'),

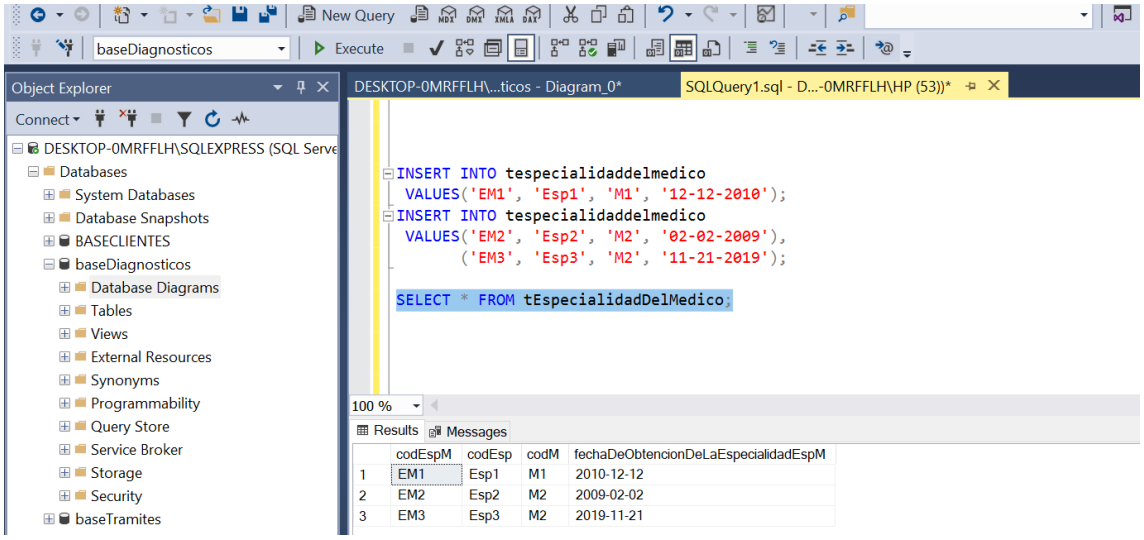
('E3', 'Artritis', 'Enfermedad crónica de los huesos');



```
INSERT INTO tdiagnostico
VALUES('D1', '02-25-2022 10:00:00', 'Sinusitis aguda y alergia', 'P01', 'M1'),
    ('D2', '02-28-2022 12:00:00', 'Niña sana', 'P02', 'M2');
```



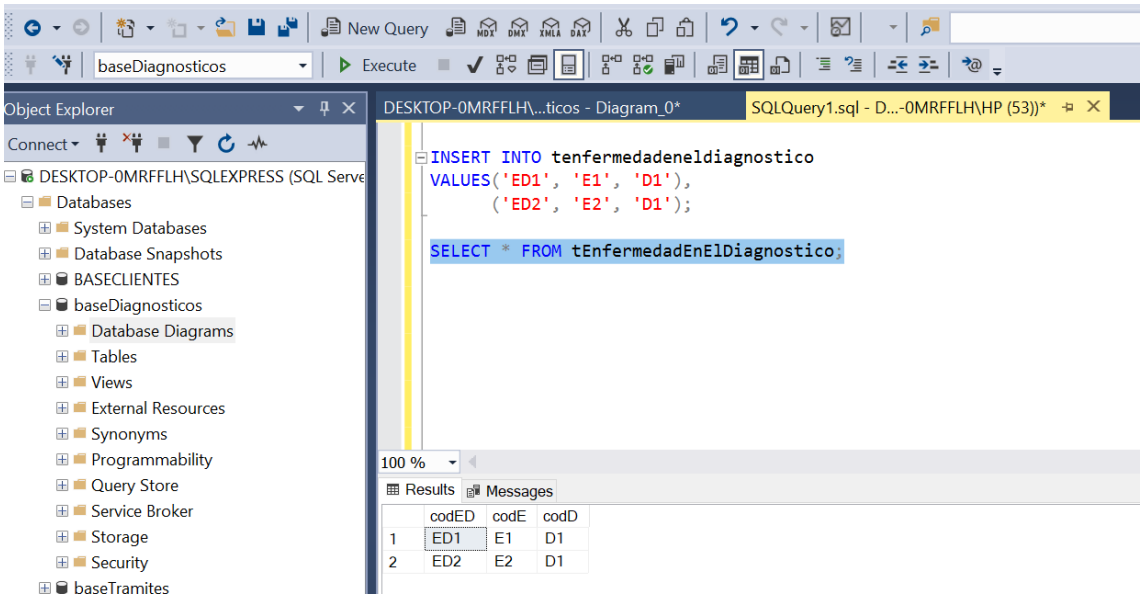

```
INSERT INTO tespecialidaddelmedico
 VALUES('EM1', 'Esp1', 'M1', '12-12-2010');
INSERT INTO tespecialidaddelmedico
 VALUES('EM2', 'Esp2', 'M2', '02-02-2009'),
     ('EM3', 'Esp3', 'M2', '11-21-2019');
```



INSERT INTO tenfermedadeneldiagnostico

VALUES('ED1', 'E1', 'D1'),

('ED2', 'E2', 'D1');



4.5. Según el script, las tablas tienen la siguiente información:

SELECT * FROM `tpaciente`

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

	codP	tipoDocP	nroDocP	paternoP	maternoP	nombresP	fechaNacimientoP	generoP
Editar Copiar Borrar	P01	DNI	11111111	Salas	Rivas	Juan	2000-02-02	M
Editar Copiar Borrar	P02	DNI	22222222	Pèrez	Rozas	Elena	2020-10-01	F

SELECT * FROM `tmedico`

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

	codM	tipoDocM	nroDocM	paternoM	maternoM	nombresM	celularM	fechaNacimientoM	generoM
Editar Copiar Borrar	M1	DNI	33333333	Zela	Ramírez	Javier	999999999	1970-10-10	M
Editar Copiar Borrar	M2	Carnet de extra	7777	Cabral	Desousa	Alejandra	988888877	1975-12-06	F

```
SELECT * FROM `tdiagnostico`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave:

Opciones extra

←T→	codD	fechaHoraD	descripcionD	codP	codM
Editar Copiar Borrar	D1	2022-02-25 10:00:00	Sinusitis aguda y alergia	P01	M1
Editar Copiar Borrar	D2	2022-02-28 12:00:00	Niña sana	P02	M2

```
SELECT * FROM `tenfermedad`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave:

Opciones extra

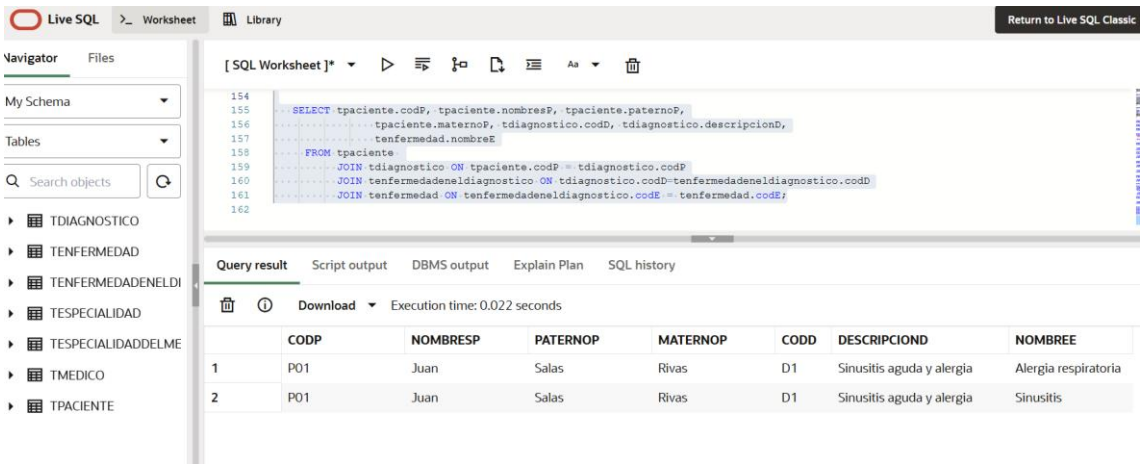
←T→	codE	nombreE	descricionE
Editar Copiar Borrar	E1	Alergia respiratoria	Alergias en el sistema respiratorio
Editar Copiar Borrar	E2	Sinusitis	Gripe crónica
Editar Copiar Borrar	E3	Artritis	Enfermedad crónica de los huesos

SELECT * FROM `tespecialidad`
Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]
Mostrar todo
Número de filas: 25
Filtrar filas: Buscar en esta tabla
Ordenar seg
Opciones extra
codEsp nombreEsp descripcionEsp
Editar Copiar Borrar Esp1 Otorrinolagingología Ojos, nariz y boca
Editar Copiar Borrar Esp2 Pediatría Neonato, niño y adolescente
Editar Copiar Borrar Esp3 Cardiología Corazón, sistema circulatorio
Editar Copiar Borrar Esp4 Odontología Dientes, encías

SELECT * FROM `tespecialidaddelmedico`
Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]
Mostrar todo
Número de filas: 25
Filtrar filas: Buscar en esta tabla
Opciones extra
codEspM codEsp codM fechaDeObtencionDeLaEspecialid
Editar Copiar Borrar EM1 Esp1 M1 2010-12-12
Editar Copiar Borrar EM2 Esp2 M2 2009-02-02
Editar Copiar Borrar EM3 Esp3 M2 2019-11-21

4.6. Mostrar los nombres de los pacientes con las enfermedades que tengan.

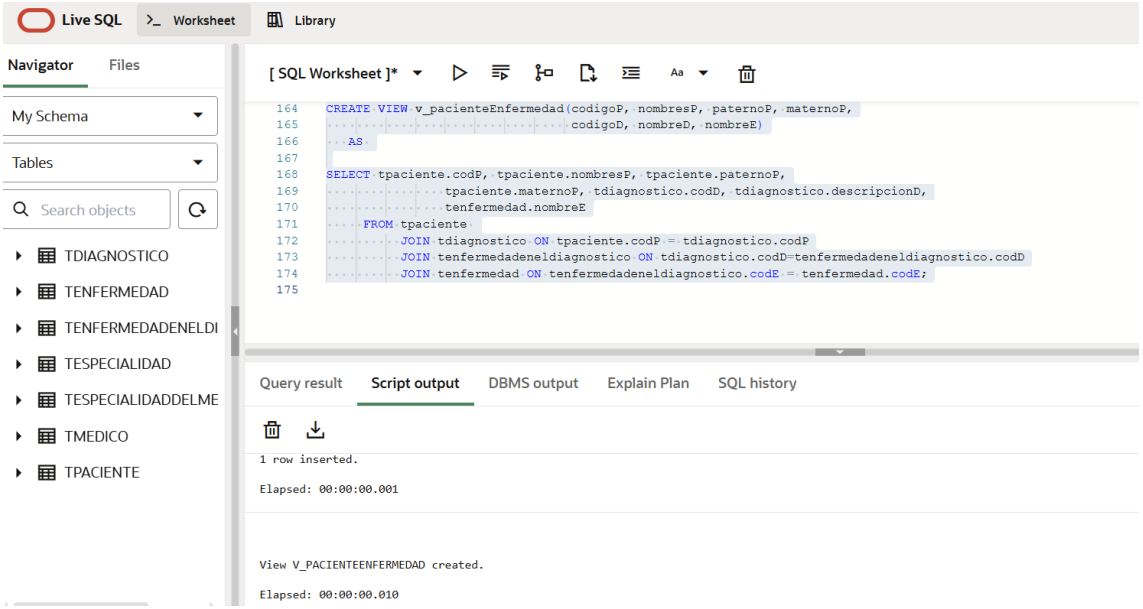
```
SELECT tpaciente.codP, tpaciente.nombresP, tpaciente.paternoP,
        tpaciente.maternoP, tdiagnostico.codD, tdiagnostico.descripcionD,
        tenfermedad.nombreE
 FROM tpaciente
    JOIN tdiagnostico ON tpaciente.codP = tdiagnostico.codP
    JOIN tenfermedadeneldiagnostico ON tdiagnostico.codD=tenfermedadeneldiagnostico.codD
    JOIN tenfermedad ON tenfermedadeneldiagnostico.codE = tenfermedad.codE;
```



4.7. Como se puede ver en la consulta 4.6. para obtener la información solicitada, se requiere unir varias tablas. Entonces, podríamos almacenar este código de consulta en una vista, a la cual, sea más fácil acceder. Entonces, creamos la vista "v_PacienteEnfermedad"

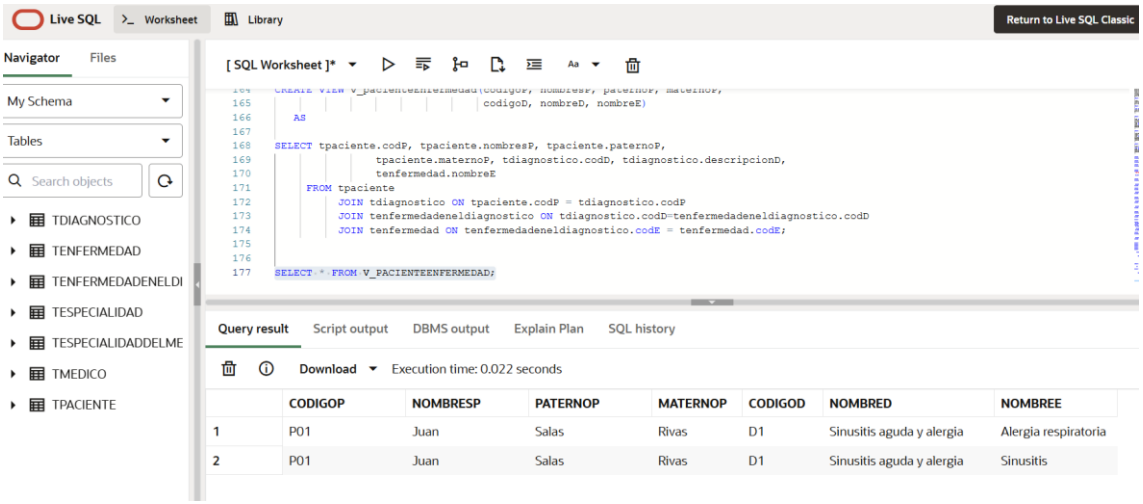
```
CREATE VIEW v_pacienteEnfermedad(codigoP, nombresP, paternoP, maternoP,
                                 codigoD, nombreD, nombreE)
  AS

SELECT tpaciente.codP, tpaciente.nombresP, tpaciente.paternoP,
       tpaciente.maternoP, tdiagnostico.codD, tdiagnostico.descripcionD,
       tenfermedad.nombreE
   FROM tpaciente
      JOIN tdiagnostico ON tpaciente.codP = tdiagnostico.codP
      JOIN tenfermedadeneldiagnostico ON
tdiagnostico.codD=tenfermedadeneldiagnostico.codD
```

JOIN tenfermedad ON tenfermedadeneldiagnostico.codE = tenfermedad.codE;



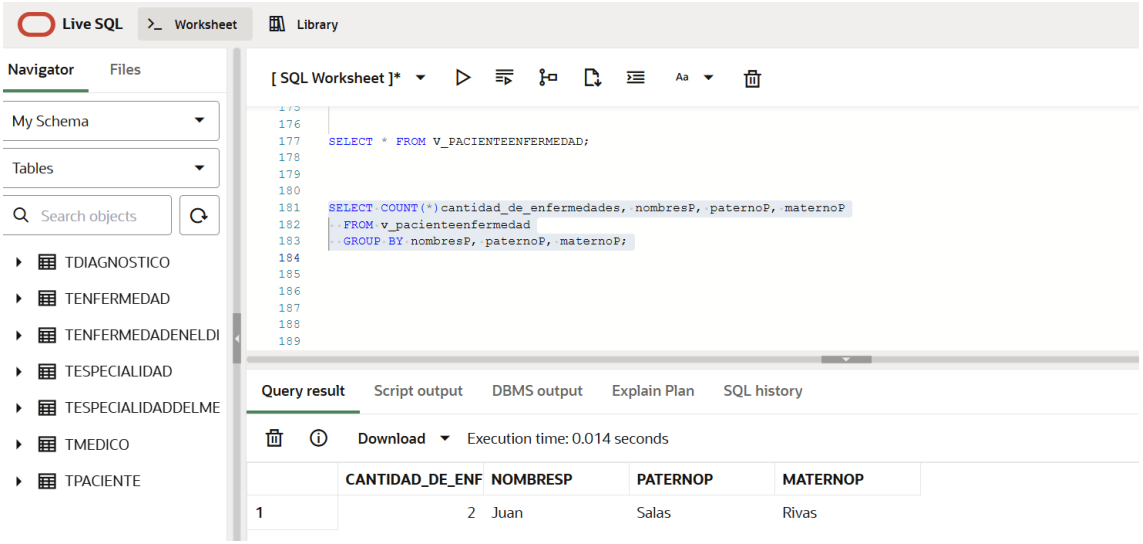
4.8. Ahora, podemos acceder a la vista, como si se tratara de una tabla.

```
SELECT * FROM V_PACIENTEENFERMEDAD;
```



4.9. Podemos ver cuántas enfermedades ha tenido un paciente con una consulta sencilla.

```
SELECT COUNT(*)cantidad_de_enfermedades, nombresP, paternoP, maternoP
 FROM v_pacienteenfermedad
 GROUP BY nombresP, paternoP, maternoP;
```



Las consultas para los enunciados del 4.6. al 4.9. son exactamente las mismas para SQL Server.

5. EJERCICIOS EN CLASE: (CONSTRUCCIÓN)

5.1. Crear una vista para mostrar todos los datos de los médicos y sus especialidades.

5.2. A partir de la vista anterior, mostrar cuántas especialidades tiene cada médico.

5.3. A partir de la vista anterior, mostrar solamente a los médicos que tengan 2 a más especialidades.

5.4. Crear una vista que muestre solamente a las enfermedades que aún no se hayan detectado en los diagnósticos de la Clínica.

5.5. Crear una vista que muestre a todas las especialidades para las cuales aún no se tengan médicos en la Clínica.

5.6. A partir de la vista anterior, mostrar cuántas especialidades aún no tienen médico en la Clínica.

5.7. Crear una vista que muestre los nombres de los pacientes cuyos diagnósticos no estén relacionados con ninguna enfermedad.