

## Ejercicio de Evaluación Final Modulo2

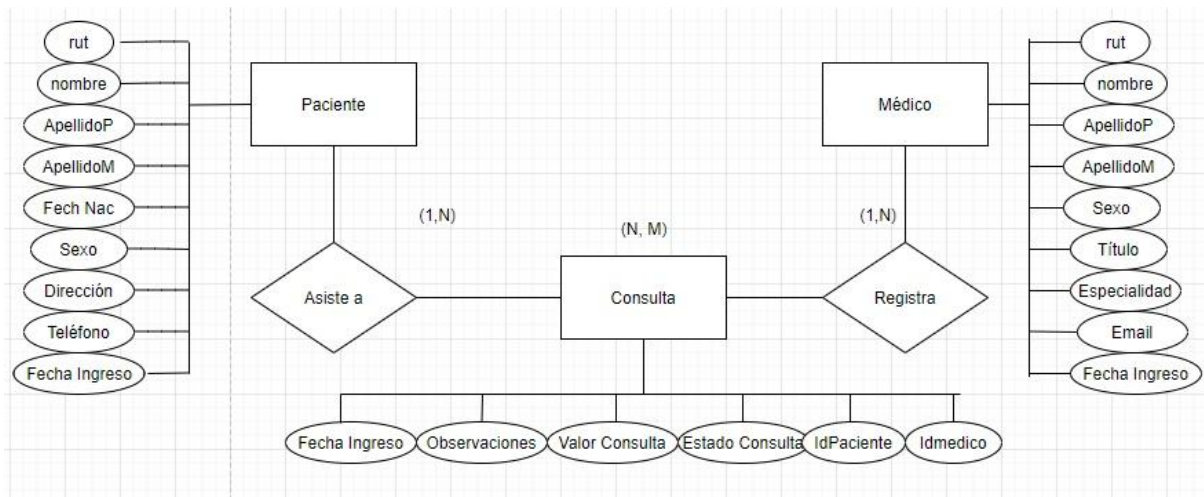
**Nombre:** María Soledad López Guerrero

-Pensar en una problemática para la cual se deba diseñar una base de datos y en base a eso realizar lo siguiente:

### Planteamiento del problema

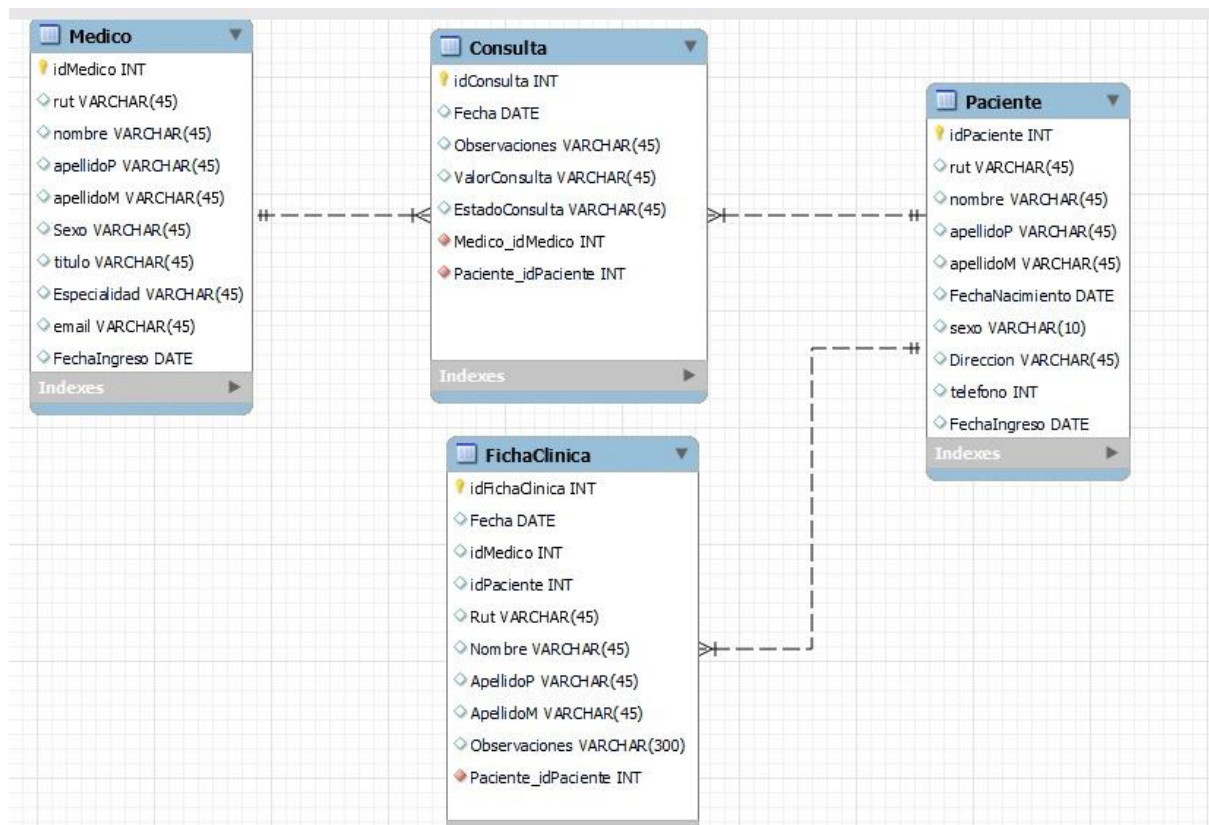
Una clínica necesita crear una base de datos que registre las consultas de cada paciente, para esto se trabajaran con 3 entidades principales: Paciente, Médicos y Consultas.

1)Elaborar el modelo entidad-relacion(Incluir en el documento):



2)Elaborar el modelo en workbench (incluir en el documento):

**Modelo Base de Datos de Registros de consultas médicas.**



3) Copiar el script generado (incluir en el documento):

### Script

-- MySQL Workbench Forward Engineering

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO
_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
  
```

```

-- -----
-- Schema BDregistroConsultas
-- -----
  
```

```

-- Schema BDregistroConsultas
-- -----
  
```

```

CREATE SCHEMA IF NOT EXISTS `BDregistroConsultas` DEFAULT CHARACTER SET utf8 ;
USE `BDregistroConsultas` ;
  
```

```

-- -----
-- Table `BDregistroConsultas`.`Paciente`
-- -----
  
```

```

CREATE TABLE IF NOT EXISTS `BDregistroConsultas`.`Paciente` (
  `idPaciente` INT NOT NULL AUTO_INCREMENT,
  `rut` VARCHAR(45) NULL,
  `nombre` VARCHAR(45) NULL,
  
```

```

`apellidoP` VARCHAR(45) NULL,
`apellidoM` VARCHAR(45) NULL,
`FechaNacimiento` DATE NULL,
`sexo` VARCHAR(10) NULL,
`Direccion` VARCHAR(45) NULL,
`telefono` INT NULL,
`FechaIngreso` DATE NULL,
PRIMARY KEY (`idPaciente`))
ENGINE = InnoDB;

```

```

-----
-- Table `BDregistroConsultas`.`Medico`
-----

```

```

CREATE TABLE IF NOT EXISTS `BDregistroConsultas`.`Medico` (
  `idMedico` INT NOT NULL AUTO_INCREMENT,
  `rut` VARCHAR(45) NULL,
  `nombre` VARCHAR(45) NULL,
  `apellidoP` VARCHAR(45) NULL,
  `apellidoM` VARCHAR(45) NULL,
  `Sexo` VARCHAR(45) NULL,
  `titulo` VARCHAR(45) NULL,
  `Especialidad` VARCHAR(45) NULL,
  `email` VARCHAR(45) NULL,
  `FechaIngreso` DATE NULL,
  PRIMARY KEY (`idMedico`))
ENGINE = InnoDB;

```

```

-----
-- Table `BDregistroConsultas`.`Consulta`
-----

```

```

CREATE TABLE IF NOT EXISTS `BDregistroConsultas`.`Consulta` (
  `idConsulta` INT NOT NULL AUTO_INCREMENT,
  `Fecha` DATE NULL,
  `Observaciones` VARCHAR(45) NULL,
  `ValorConsulta` VARCHAR(45) NULL,
  `EstadoConsulta` VARCHAR(45) NULL,
  `Medico_idMedico` INT NOT NULL,
  `Paciente_idPaciente` INT NOT NULL,
  PRIMARY KEY (`idConsulta`),
  INDEX `fk_Consulta_Medico1_idx` (`Medico_idMedico` ASC),
  INDEX `fk_Consulta_Paciente1_idx` (`Paciente_idPaciente` ASC),
  CONSTRAINT `fk_Consulta_Medico`
    FOREIGN KEY (`Medico_idMedico`)
      REFERENCES `BDregistroConsultas`.`Medico` (`idMedico`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Consulta_Paciente`
    FOREIGN KEY (`Paciente_idPaciente`)
      REFERENCES `BDregistroConsultas`.`Paciente` (`idPaciente`)
    ON DELETE NO ACTION

```

```
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `BDregistroConsultas`.`FichaClinica`
```

```
CREATE TABLE IF NOT EXISTS `BDregistroConsultas`.`FichaClinica` (
  `idFichaClinica` INT NOT NULL AUTO_INCREMENT,
  `Fecha` DATE NULL,
  `idMedico` INT ZEROFILL NULL,
  `idPaciente` INT NULL,
  `Rut` VARCHAR(45) NULL,
  `Nombre` VARCHAR(45) NULL,
  `ApellidoP` VARCHAR(45) NULL,
  `ApellidoM` VARCHAR(45) NULL,
  `Observaciones` VARCHAR(300) NULL,
  `Paciente_idPaciente` INT NOT NULL,
  PRIMARY KEY (`idFichaClinica`),
  INDEX `fk_FichaClinica_Paciente1_idx` (`Paciente_idPaciente` ASC),
  CONSTRAINT `fk_FichaClinica_Paciente1`
    FOREIGN KEY (`Paciente_idPaciente`)
      REFERENCES `BDregistroConsultas`.`Paciente` (`idPaciente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

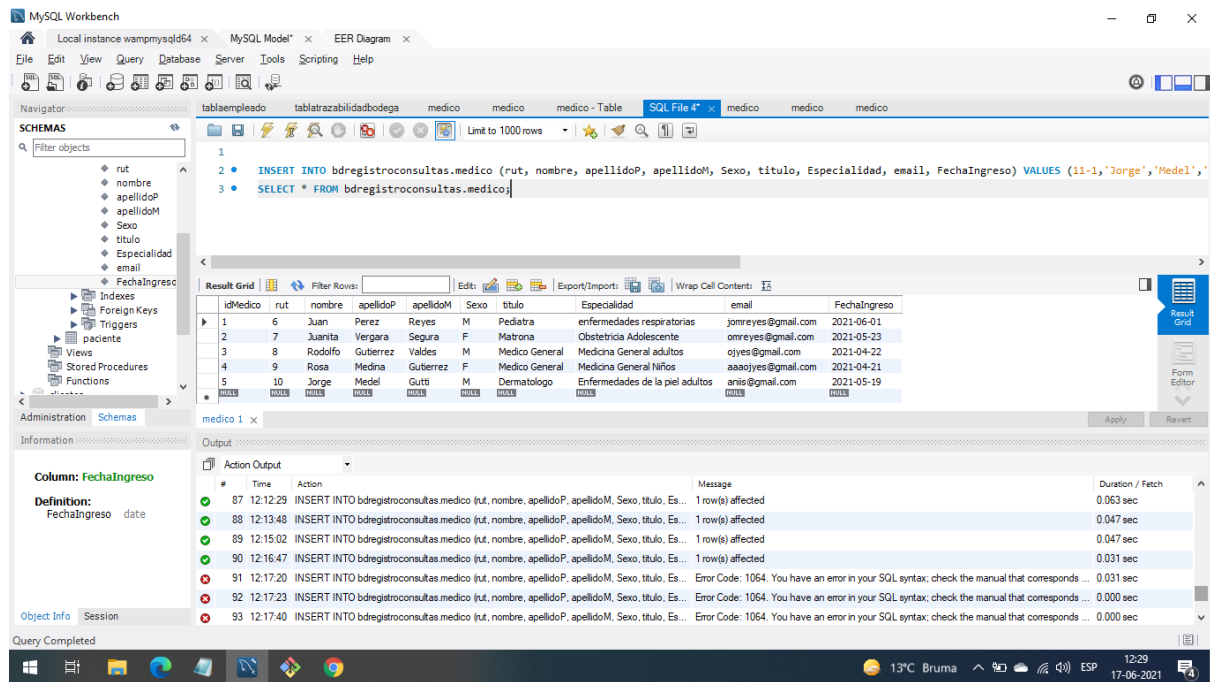
4) Crear registros en la base de datos y hacer consultas utilizando todo lo visto en clases y lo ustedes vieron por su cuenta (mínimo 2 consultas por cada operador, cláusula o comando) dichas consultas deben tener la descripción, query, asociada a ella y el resultado mostrado (incluir en el documento)

### Creación de registros

Query para la creación registros en la base de datos Registro Consulta Médica, para las entidades Médico, Paciente y Consultas.

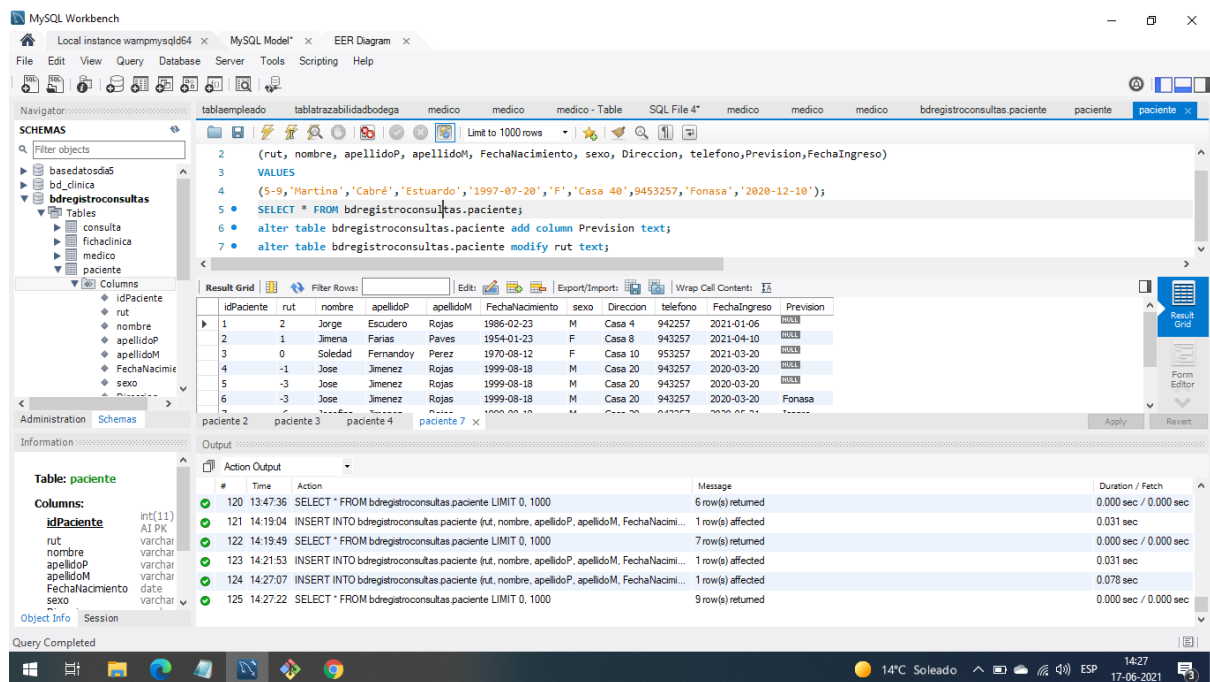
Query Tabla Médicos:

```
INSERT INTO bdregistroconsultas.medico (rut, nombre, apellidoP, apellidoM, Sexo, titulo, Especialidad, email, FechaIngreso) VALUES (11-1,'Jorge','Medel','Gutti','M','Dermatologo','Enfermedades de la piel adultos','aniis@gmail.com','2021-05-19');
```



### Query Tabla Pacientes:

INSERT INTO bdregistroconsultas.paciente (rut, nombre, apellidoP, apellidoM, FechaNacimiento, sexo, Direccion, telefono, Prevision, FechaIngreso) VALUES (5-9,'Martina','Cabré','Estuardo','1997-07-20','F','Casa 40',9453257,'Fonasa','2020-12-10');  
SELECT \* FROM bdregistroconsultas.paciente;



Query Registro de consultas paciente:

```
SELECT * FROM bdregistroconsultas.consulta;
```

```
INSERT INTO bdregistroconsultas.consulta
```

```
(Fecha, Observaciones, ValorConsulta, EstadoConsulta, Medico_idMedico, Paciente_idPaciente)
```

```
VALUES
```

```
('2021-06-04','control',10000,'Pagado',2,2);
```

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
1 SELECT * FROM bdregistroconsultas.consulta;
2 INSERT INTO bdregistroconsultas.consulta
3 (Fecha, Observaciones, ValorConsulta, EstadoConsulta, Medico_idMedico, Paciente_idPaciente)
4 VALUES
5 ('2021-06-04','control',10000,'Pagado',2,2);
6
```

The Results Grid shows the following data:

#	IdConsulta	Fecha	Observaciones	ValorConsulta	EstadoConsulta	Medico_idMedico	Paciente_idPaciente
4		2021-04-03	Hepatitis control	10000	Pagado	1	1
5		2021-02-04	control	10000	Pendiente	1	2
6		2021-03-04	control	10000	Pendiente	1	2
7		2021-06-04	control	10000	Pagado	2	2
8		2021-06-04	control	10000	Pagado	2	2

The Action Output pane shows the following messages:

#	Time	Action	Message	Duration / Fetch
140	14:51:28	INSERT INTO bdregistroconsultas.consulta (Fecha, Observaciones, ValorConsulta, EstadoC...	1 row(s) affected	0.125 sec
141	14:51:39	INSERT INTO bdregistroconsultas.consulta (Fecha, Observaciones, ValorConsulta, EstadoC...	1 row(s) affected	0.031 sec
142	14:52:18	INSERT INTO bdregistroconsultas.consulta (Fecha, Observaciones, ValorConsulta, EstadoC...	1 row(s) affected	0.047 sec
143	14:52:23	SELECT * FROM bdregistroconsultas.medico LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
144	14:52:23	INSERT INTO bdregistroconsultas.consulta (Fecha, Observaciones, ValorConsulta, EstadoC...	1 row(s) affected	0.031 sec
145	14:52:45	SELECT * FROM bdregistroconsultas.consulta LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

## Consultas Cláusula From,

Query para seleccionar datos desde las tablas:

```
SELECT * FROM bdregistroconsultas.paciente;
```

```
select nombre, apellidoP, apellidoM, FechaNacimiento from bdregistroconsultas.paciente;
```

```
select nombre, apellidoP, apellidoM from bdregistroconsultas.paciente where prevision='Fonasa';
```

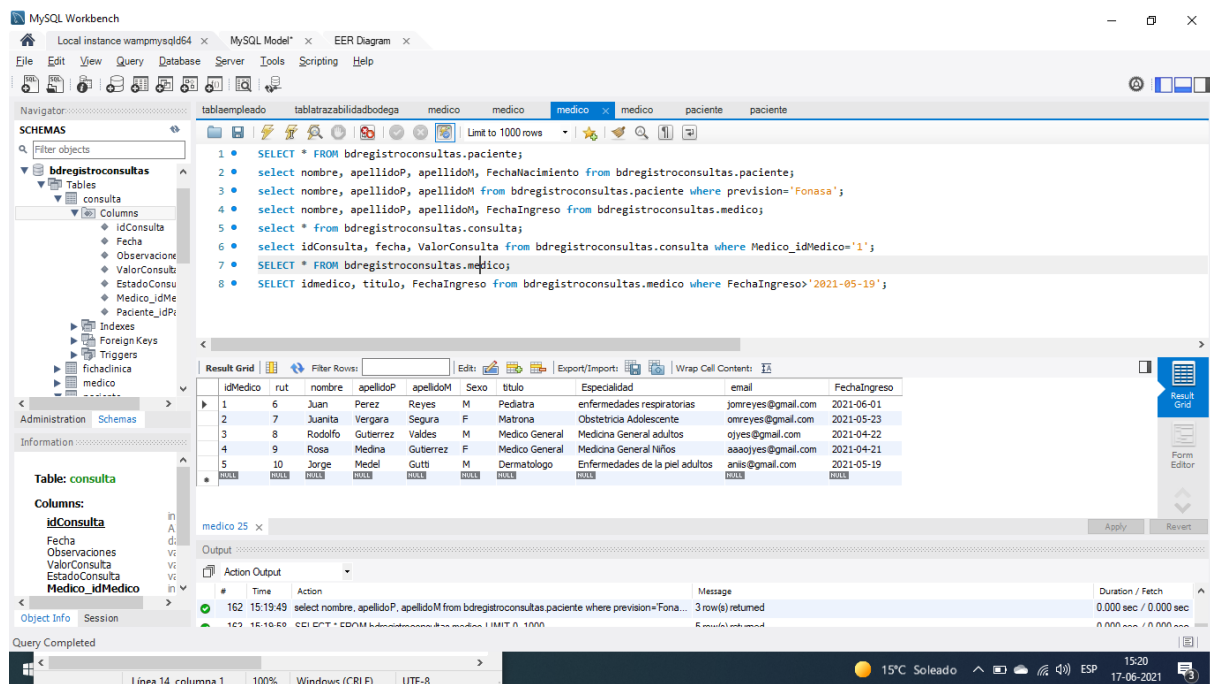
```
select nombre, apellidoP, apellidoM, FechaIngreso from bdregistroconsultas.medico;
```

```
select * from bdregistroconsultas.consulta;
```

```
select idConsulta, fecha, ValorConsulta from bdregistroconsultas.consulta where Medico_idMedico='1';
```

```
SELECT * FROM bdregistroconsultas.medico;
```

```
SELECT idmedico, titulo, FechaIngreso from bdregistroconsultas.medico where FechaIngreso>'2021-05-19';
```



## Clausulas where con operador lógico AND

Query para saber los datos de atención de un paciente que haya asistido a control y su estado de pago esté pendiente:

-Select fecha, Paciente\_idPaciente, observaciones, EstadoConsulta fecha from bdregistroconsultas.consulta where Observaciones='control' and EstadoConsulta='Pendiente';

Query para saber todos los pacientes registrados de sexo masculino y con prevision fonasa:

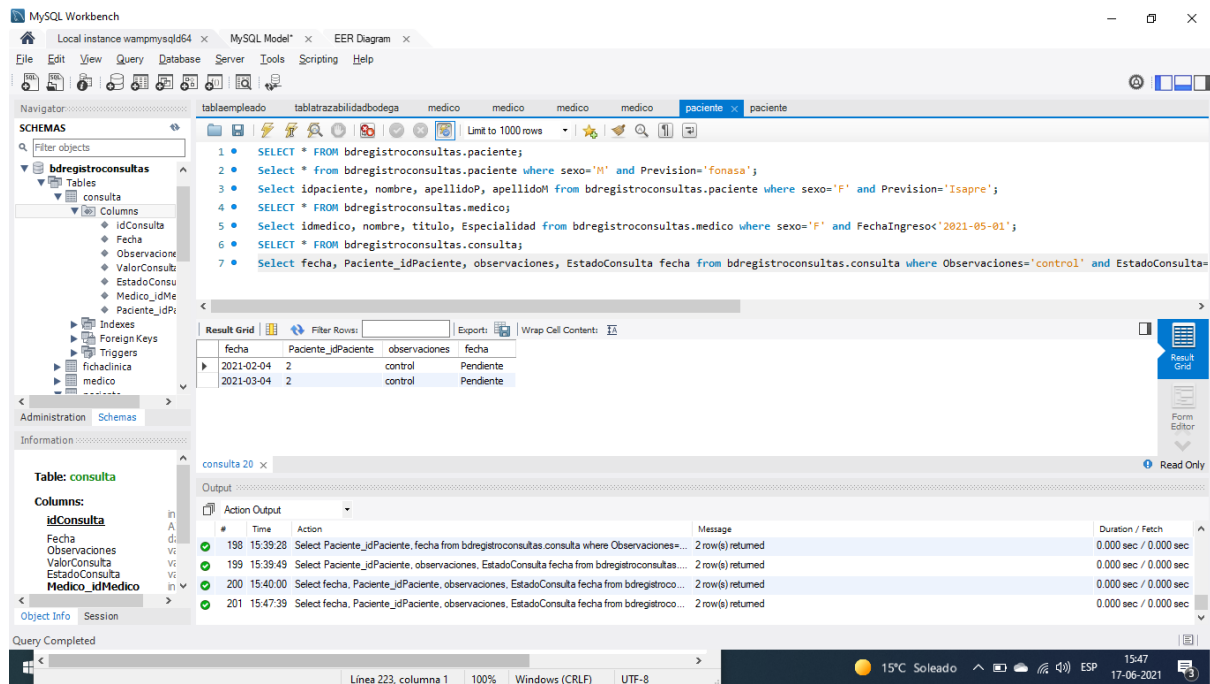
-Select \* from bdregistroconsultas.paciente where sexo='M' and Prevision='fonasa';

Query para traer un informe de las pacientes de sexo femenino que incluyen su Nombre, ApellidoP, ApellidoM y que pertenecen a Isapre:

-Select idpaciente, nombre, apellidoP, apellidoM from bdregistroconsultas.paciente where sexo='F' and Prevision='Isapre';

Query para consultar los datos de los medicos de sexo femenino y cuyo ingreso haya sido menor a la fecha 01-mayo-2021.

-Select idmedico, nombre, titulo, Especialidad from bdregistroconsultas.medico where sexo='F' and FechaIngreso<'2021-05-01';



## Operador lógico OR

Query para consultar los pacientes de sexo masculino o con prevision Fonasa.

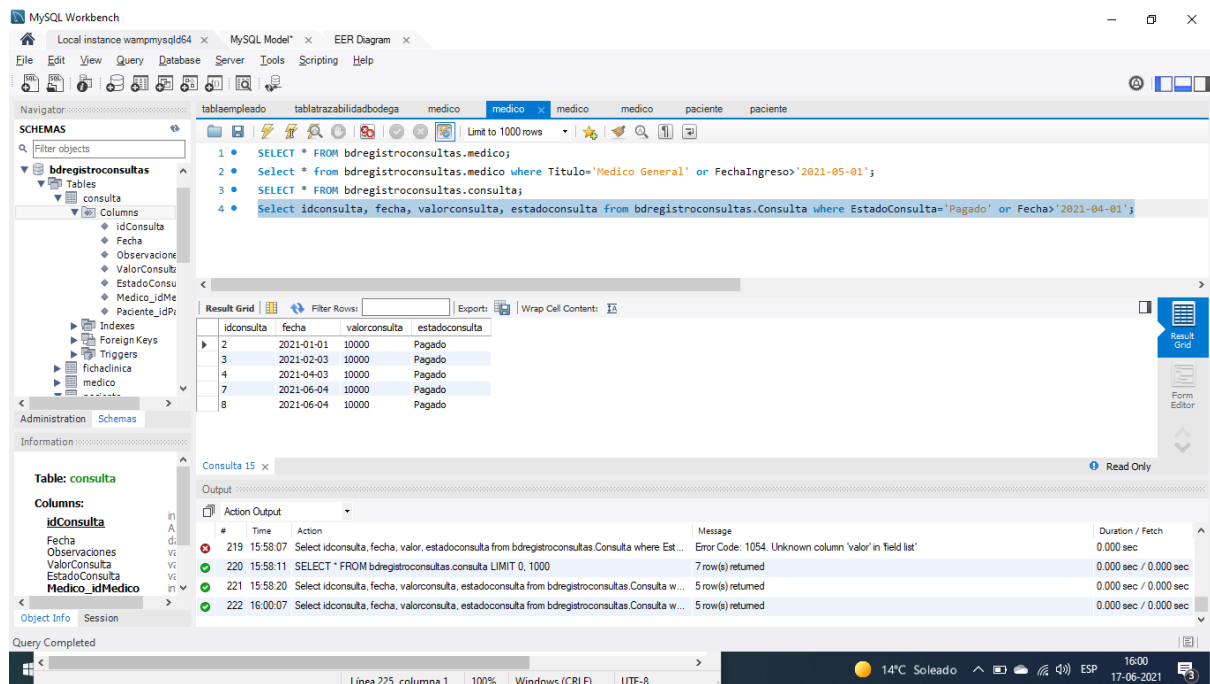
-Select \* from bdregistroconsultas.paciente where sexo='M' or Prevision='Fonasa';

Query para consultar los médicos generales o aquellos que hayan ingresado posterior a la fecha 01-05-2021

-Select \* from bdregistroconsultas.medico where Titulo='Medico General' or FechaIngreso>'2021-05-01';

Query para seleccionar las consultas que estén pagadas o con fecha mayor al 01-Abril-2021.

-Select idconsulta, fecha, valorconsulta, estadoconsulta from bdregistroconsultas.Consulta where EstadoConsulta='Pagado' or Fecha>'2021-04-01';



## Cláusula where con IN



Query para seleccionar los datos de los médicos que su título sea pediatra, médico general y su fecha de ingreso sea posterior al 20-mayo-2021:

-Select \* from bdregistroconsultas.medico where titulo in ('Pediatra','Medico General') and FechaIngreso>2021-05-20;

Query para traer los datos de idconsulta, fecha, cuya estadoConsulta sea pendiente, pagado y observaciones iguales a control:

-Select idconsulta, fecha, EstadoConsulta, Observaciones from bdregistroconsultas.consulta where EstadoConsulta in ('Pagado','Pendiente') and Observaciones='control';

Query para traer los datos de los pacientes que tengan prestacion isapre, particular y sexo masculino:

-select \* from bdregistroconsultas.paciente where prevision in ('Particular','Isapre') and sexo='M';

The screenshot shows the MySQL Workbench interface. The query editor contains the following queries:

1. SELECT \* FROM bdregistroconsultas.medico;
2. select \* from bdregistroconsultas.medico where titulo in ('Pediatra','Medico General') and FechaIngreso>2021-05-20;
3. SELECT \* FROM bdregistroconsultas.consulta;
4. select idconsulta, fecha, EstadoConsulta, Observaciones from bdregistroconsultas.consulta where EstadoConsulta in ('Pagado','Pendiente') and Observaciones='control';
5. SELECT \* FROM bdregistroconsultas.paciente;
6. select \* from bdregistroconsultas.paciente where prevision in ('Particular','Isapre') and sexo='M';

The result grid shows the output of the last query, displaying patient data with columns: idPaciente, rut, nombre, apellidoP, apellidoM, FechaNacimiento, sexo, Direccion, telefono, FechaIngreso, and Prevision. The data includes 6 rows of patient information.

idPaciente	rut	nombre	apellidoP	apellidoM	FechaNacimiento	sexo	Direccion	telefono	FechaIngreso	Prevision
7	-6	Josefina	Jimenez	Rojas	1999-08-18	M	Casa 20	943257	2020-05-21	Isapre
13	5	Mariano	Bravo	Herrera	1994-09-21	M	Casa 41	9453237	2021-11-10	Isapre
14	6	Mariano	Bravo	Herrera	1993-09-20	M	Casa 41	9453237	2021-09-10	Isapre
15	7	Paola	Fernandez	Chavez	1993-09-20	M	Casa 41	9453237	2021-09-10	Isapre
16	91	Carolina	Fernandez	Chavez	1993-09-20	M	Casa 41	9453237	2021-09-10	Isapre
19	81	Fernando	Farias	Mesa	1991-09-20	M	Casa 41	9453237	2020-09-10	Particular

The bottom panel shows the 'Action Output' tab with a log of executed queries and their results, including row counts and execution times.

## Cláusula where con NOT IN

Query para consultar los datos de los pacientes cuya previsión no sea Fonasa ni Particular.

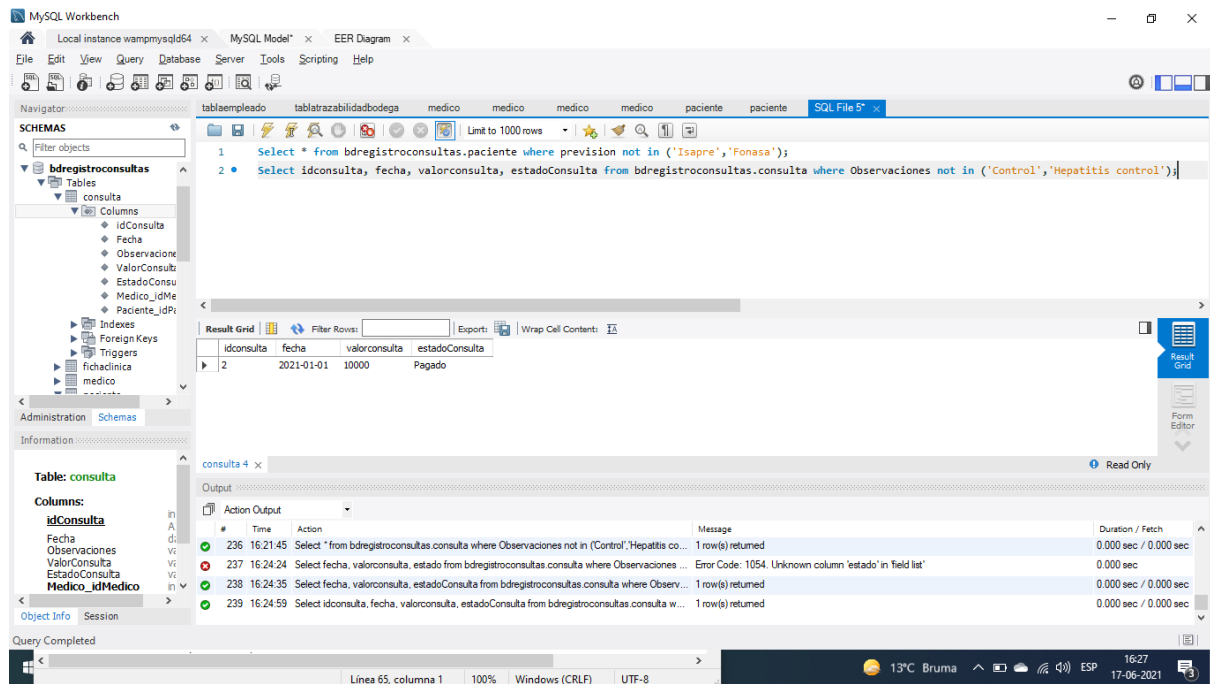
-Select \* from bdregistroconsultas.paciente where prevision not in ('Isapre','Fonasa');

Query para consultar la fecha de los pacientes que no asistieron a control, ni control por Hepatitis.

-Select \* from bdregistroconsultas.consulta where Observaciones not in ('Control','Hepatitis control');

Query para consultar los datos de fecha, valor y estado de pago de consultas que no hayan sido por control ni control de hepatitis.

-Select idconsulta, fecha, valorconsulta, estadoConsulta from bdregistroconsultas.consulta where Observaciones not in ('Control','Hepatitis control');



## Clausula where con <> (distinto)

Query para consultar los datos de los médicos cuyo título sea distinto al de Médico General.

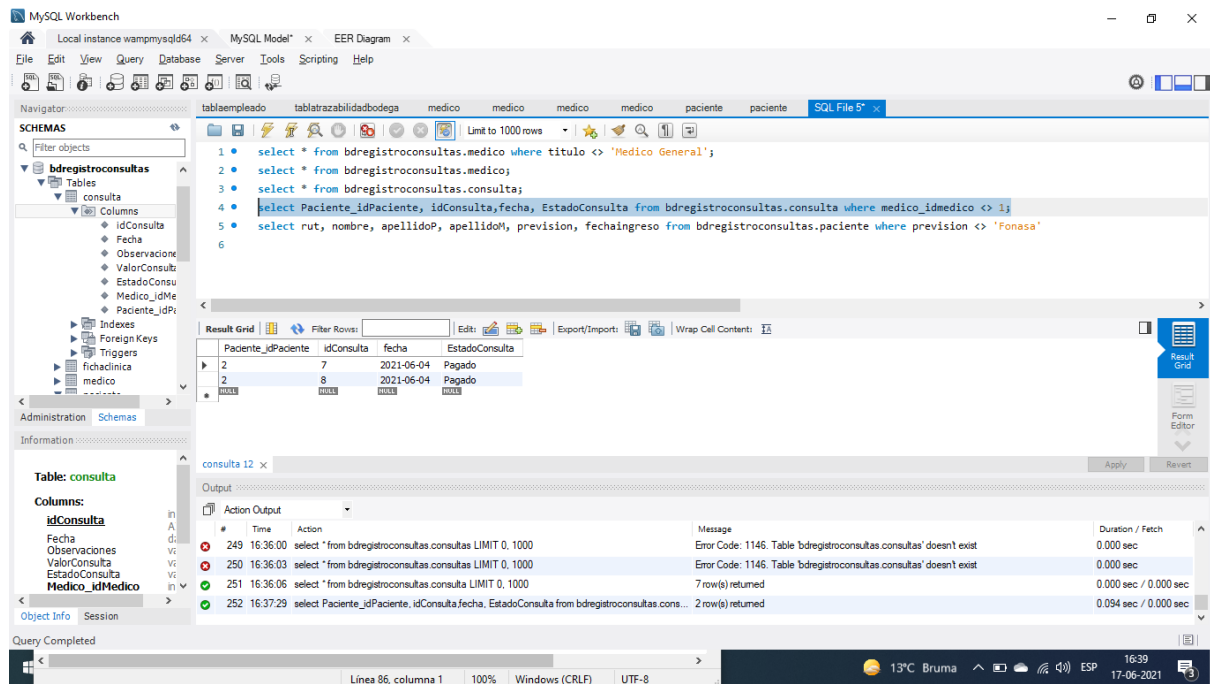
-select \* from bdregistroconsultas.medico where titulo <> 'Medico General';

Query para generar un informe con los datos personales (rut, Nombre, ApellidoP, ApellidoM, Prevision, FechaIngreso) de los pacientes cuya prevision sea distinta a Fonasa.

-select rut, nombre, apellidoP, apellidoM, prevision, fechaIngreso from bdregistroconsultas.paciente where prevision <> 'Fonasa'.

Query para solicitar los id de Paciente, id de medico, id de consulta la fecha y el estado del pago de aquellos pacientes atendidos por idmedico distintos de idmedico 1.

-select Paciente\_idPaciente, idConsulta, fecha, EstadoConsulta from bdregistroconsultas.consulta where medico\_idmedico <> 1;



## Cláusula Between

Query para consultar los datos de aquellas consultas cuyo valor se encuentren entre \$15.000 y \$30.000.

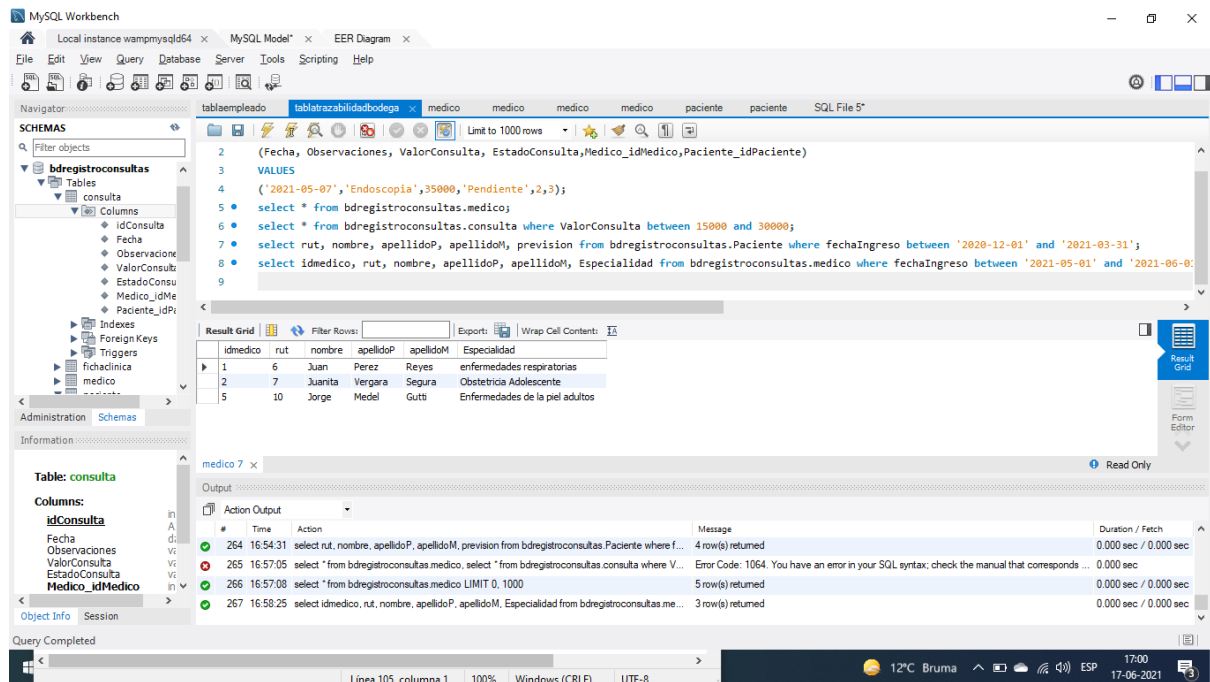
-Select \* from bdregistroconsultas.consulta where ValorConsulta between 15000 and 30000;

Query para consultar los datos personales (rut, Nombre, Apellidos) y la prevision de aquellos pacientes que hayan ingresado entre el 01-12-2020 y el 31-03-2021.

-select rut, nombre, apellidoP, apellidoM, prevision from bdregistroconsultas.Paciente where fechaIngreso between '2020-12-01' and '2021-03-31';

Query para traer el rut, Nombre, Apellidos y especialidad de los médicos que hayan ingresado entre el 01-05-2021 y el 01-06-2021.

-select idmedico, rut, nombre, apellidoP, apellidoM, Especialidad from bdregistroconsultas.medico where fechaIngreso between '2021-05-01' and '2021-06-01';



## Operador Like

Query para seleccionar los datos de los pacientes cuyo nombre comience por la letra 'C':

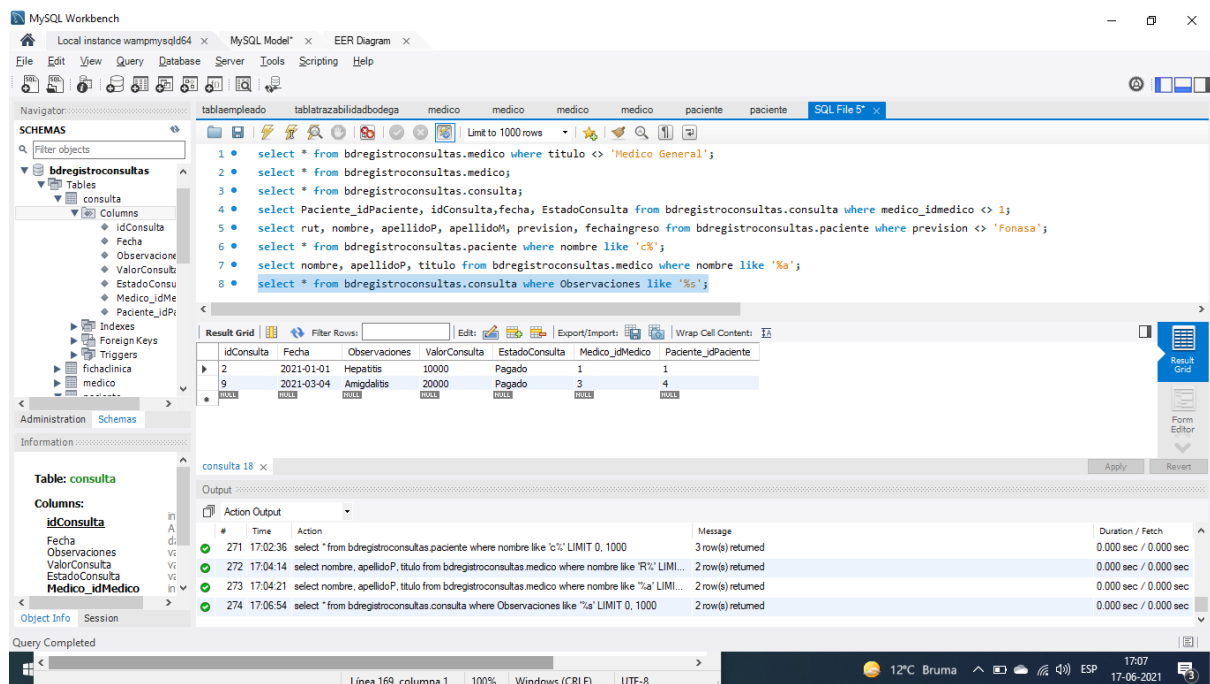
-select \* from bdregistroconsultas.paciente where nombre like 'c%';

Query para seleccionar los datos nombre, apellidoP y título de los médicos cuyo nombre termine con la letra 'A'.

-select nombre, apellidoP, titulo from bdregistroconsultas.medico where nombre like '%a';

Query para seleccionar los datos de las consultas cuya observacion termine en la letra 'S'

-select \* from bdregistroconsultas.consulta where Observaciones me like '%S';



## Clausula Order By

Query para traer los datos de la paciente ordenados por su Nombre de forma ascendente.

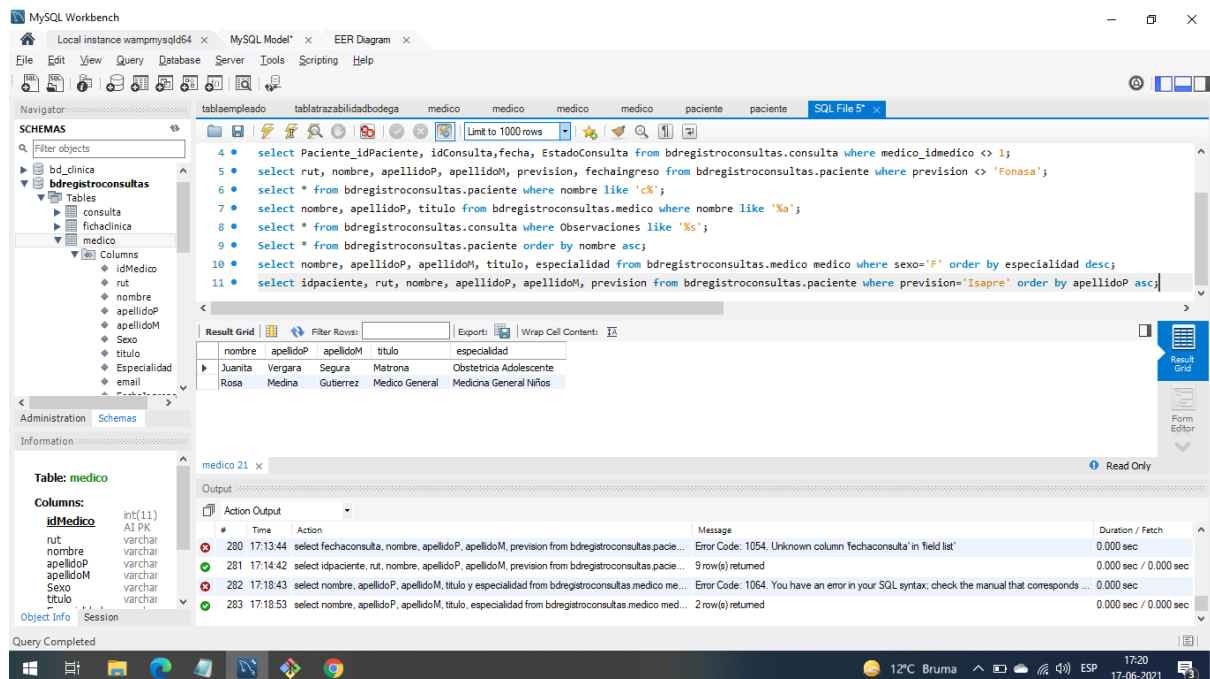
-Select \* from bdregistroconsultas.paciente order by nombre asc

Query para traer los datos (idpaciente, rut, nombre y apellidos) de los pacientes que tengan previsión isapre y los ordene de forma alfabéticamente ascendente de acuerdo a su apellido paterno.

-select idpaciente, rut, nombre, apellidoP, apellidoM, prevision from bdregistroconsultas.paciente where prevision='Isapre' order by apellidoP asc;

Query para seleccionar los datos personales de los medicos de sexo femenino y que sean ordenados por su especialidad de forma descendente.

-select nombre, apellidoP, apellidoM, titulo, especialidad from bdregistroconsultas.medico medico where sexo='F' order by especialidad desc;



## Clausula Group By

Query para consultar la cantidad de pacientes y agruparlos por sexo y cuyo ingreso haya sido entre el 01-10-2021 y el 17-06-2021.

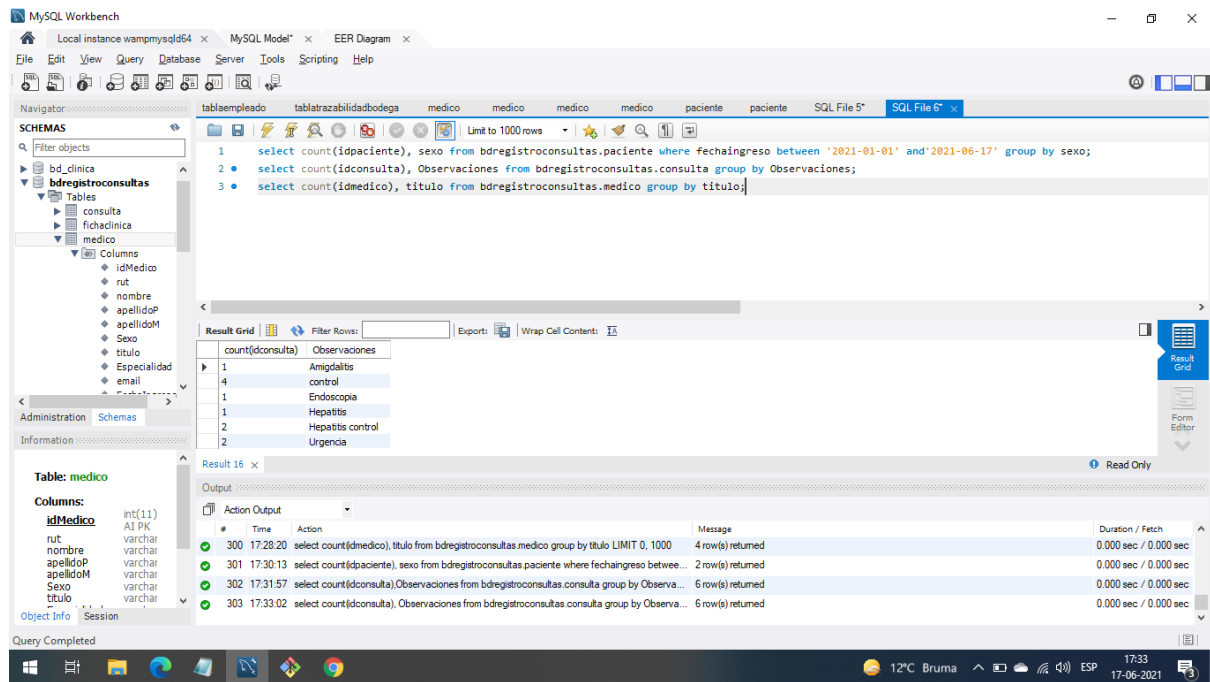
-select count(idpaciente), sexo from bdregistroconsultas.paciente where fechaingreso between '2021-01-01' and '2021-06-17' group by sexo;

Query para consultar la cantidad de consultas totales y agruparlas por enfermedad.

- select count(idconsulta), Observaciones, fecha as cantidad from bdregistroconsultas.consulta group by Observaciones;

Query para consultar la cantidad de médicos y agruparlos por título.

-select count(idmedico), titulo from bdregistroconsultas.medico group by titulo;



## Función COUNT

Query para consultar la cantidad de pacientes totales:

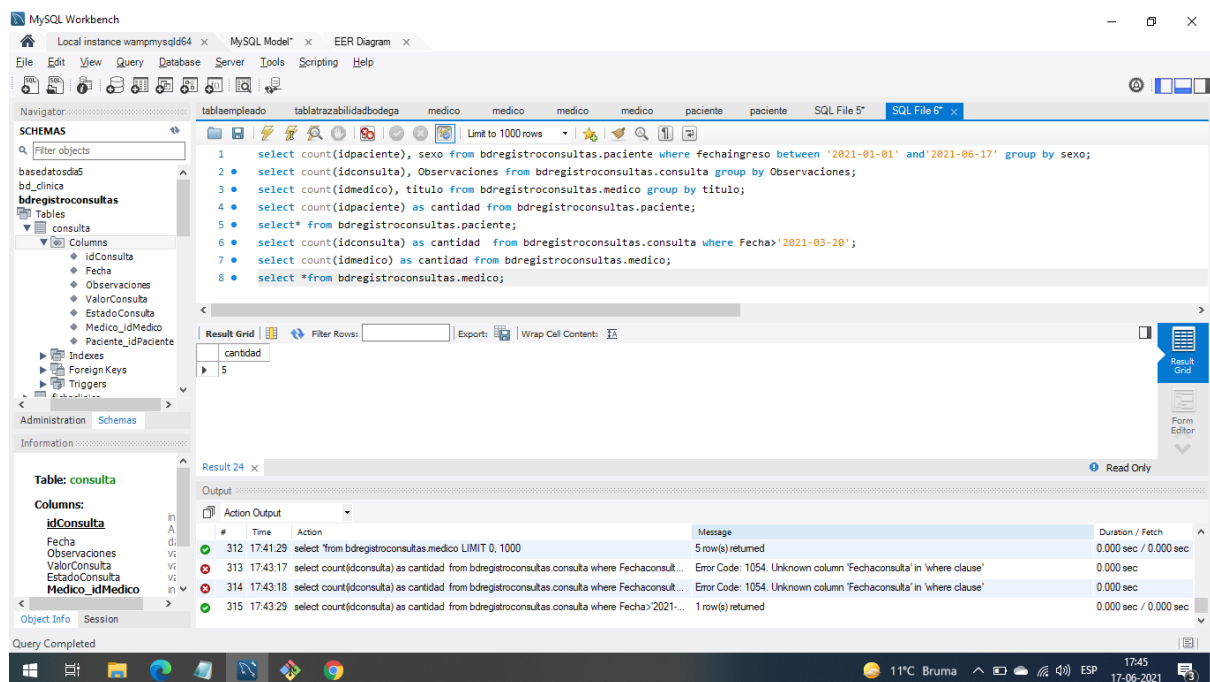
-select count(idpaciente) as cantidad from bdregistroconsultas.paciente;

Query para consultar la cantidad de Consultas ingresadas con fecha posterior al 20-03-2021:

-select count(idconsulta) as cantidad from bdregistroconsultas.consulta where Fecha>'2021-03-20';

Query para consultar la cantidad de médicos ingresados.

-select count(idmedico) as cantidad from bdregistroconsultas.medico;



## Función MAX

Query para seleccionar fecha más reciente en la que cual ingresó un médico.

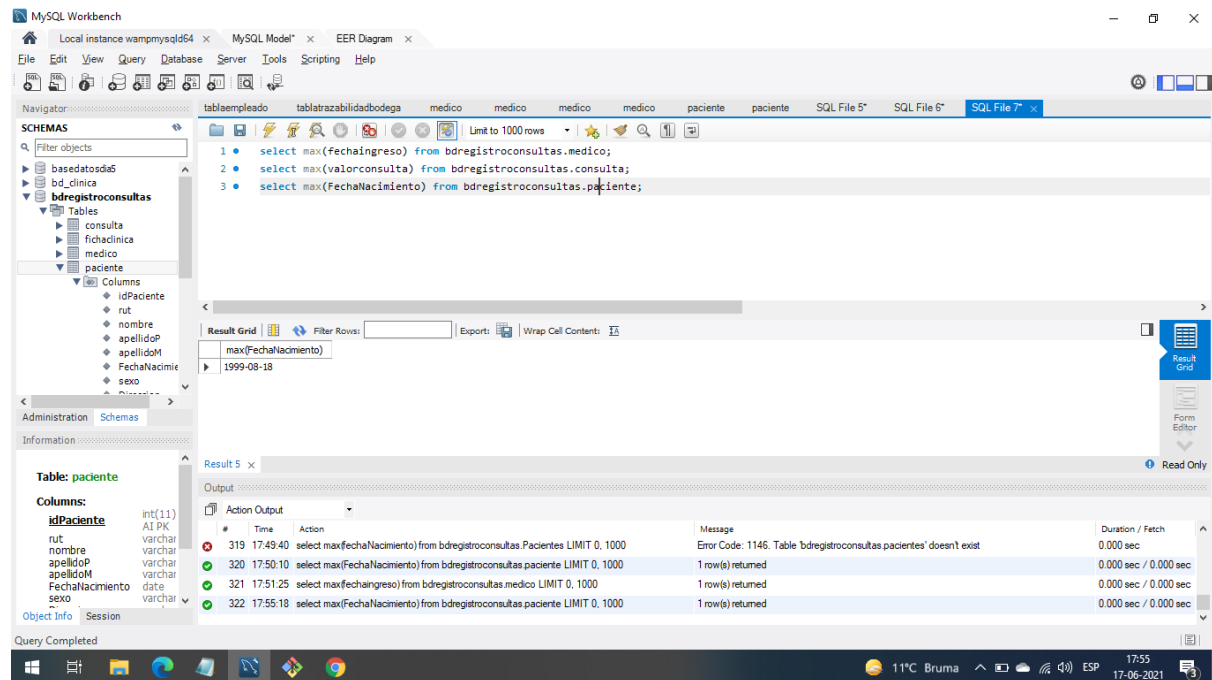
-select max(fechaingreso) from bdregistroconsultas.medico;

Query para seleccionar el arancel más alto de consulta.

-select max(valorconsulta) from bdregistroconsultas.consulta;

Query para consultar la fecha de nacimiento del paciente más reciente.

-select max(FechaNacimiento) from bdregistroconsultas.paciente;



## Función MIN:

Query para consultar por la fecha de nacimiento más antigua de los pacientes:

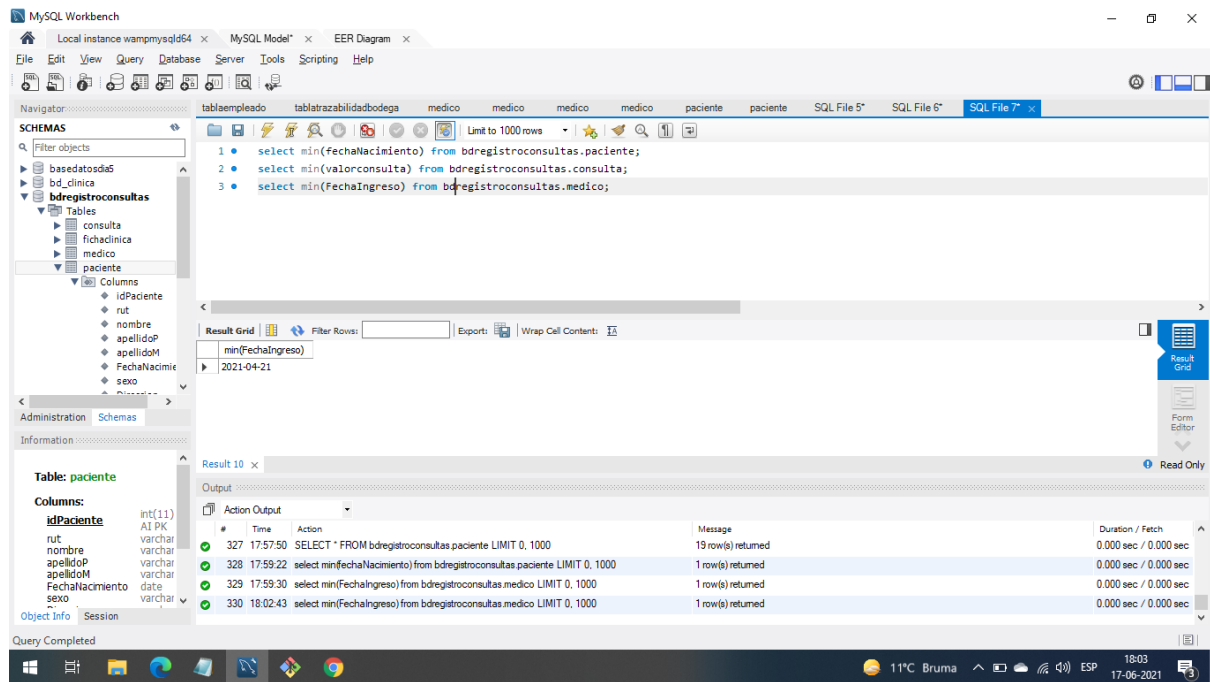
-select min(fechaNacimiento) from bdregistroconsultas.paciente;

Query para consultar el valor más económico de las consulta:

-select min(valorconsulta) from bdregistroconsultas.consulta;

Query para consultar por la fecha de ingreso más antigua de un médico:

-select min(FechaIngreso) from bdregistroconsultas.medico;



## Función SUM:

Query para consultar la suma total de los pagos de todas las consultas registradas.

-select sum(ValorConsulta) from bdregistroconsultas.consulta;

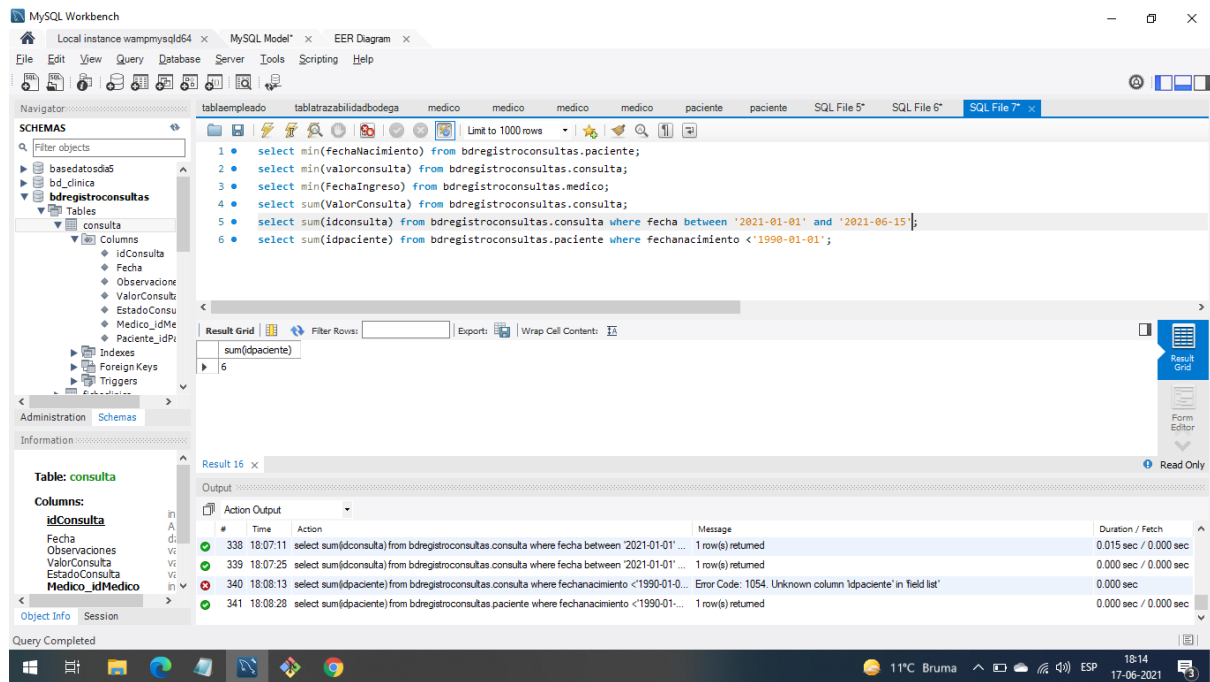
Query para consultar el total de las consultas realizadas entre los días 01-01-2021 y el 15-06-2021.

-select sum(idconsulta) from bdregistroconsultas.consulta where fecha between '2021-01-01' and '2021-06-15';

Query para consultar el total de pacientes con fecha de nacimiento anterior al 01-01-1999.

-select sum(idpaciente) from bdregistroconsultas.paciente where fechaNacimiento <'1990-01-01';





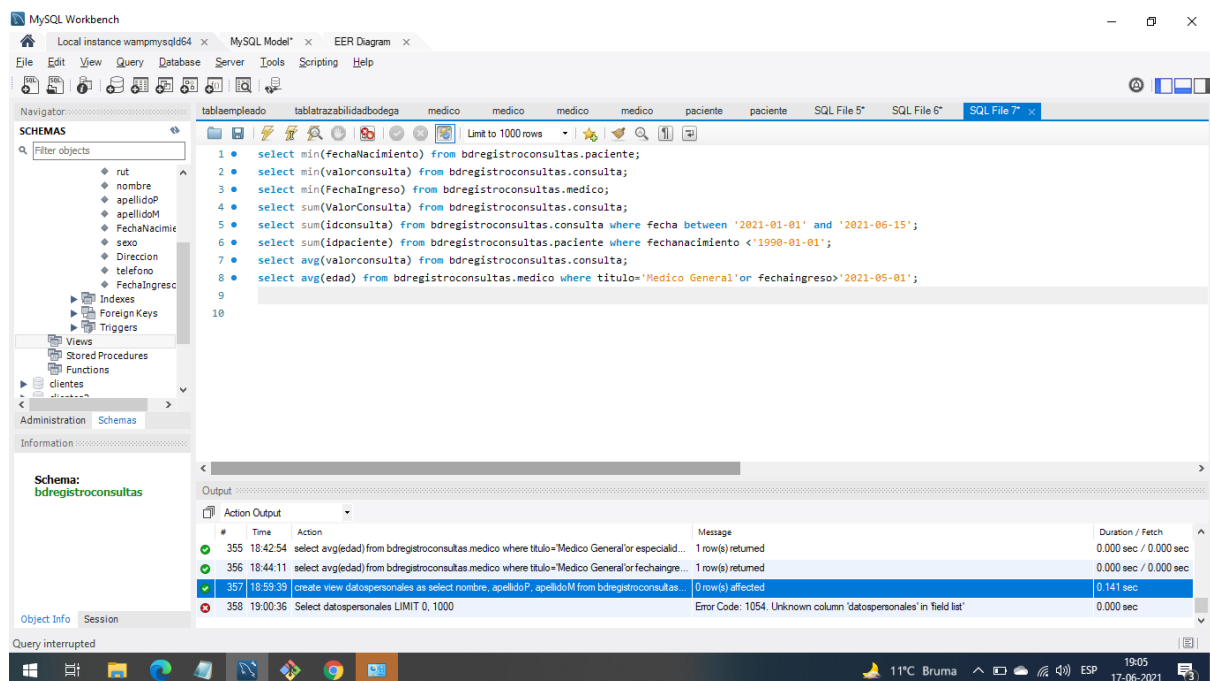
## Función AVG

Query para consultar el valor promedio de arancel por cada consulta.

-select avg(valorconsulta) from bdregistroconsultas.consulta;

Query para consultar el promedio de edad de los médicos cuyo título sea médico general.

-select avg(edad) from bdregistroconsultas.medico where titulo='Medico General';



## Clausula Having

Query para seleccionar las observaciones con los valores más altos y que sean mayor a \$15.000.

-Select observaciones, valorconsulta from bdregistroconsultas.consulta  
group by Observaciones  
having max(valorconsulta)>15000;

Query para consultar las especialidades de aquellos médicos cuya edad sea menor a 60 años.

-Select count(idmedico), especialidad from bdregistroconsultas.medico group by titulo  
having max(edad)<60;

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'bdregistroconsultas' schema selected. The main editor window contains the following SQL queries:

```
7 • select avg(valorconsulta) from bdregistroconsultas.consulta;  
8 • select avg(edad) from bdregistroconsultas.medico where titulo='Medico General' or fechaingreso>'2021-05-01';  
9 • Select observaciones, valorconsulta from bdregistroconsultas.consulta  
10 group by Observaciones  
11 having max(valorconsulta)>15000;  
12 • select * from bdregistroconsultas.medico;  
13 • Select count(idmedico), especialidad from bdregistroconsultas.medico group by titulo  
14 having max(edad)<60;  
15  
16
```

The 'Result Grid' shows the results of the last query (line 13):

count(idmedico)	especialidad
5	Medicina General adultos
2	enfermedades respiratorias
1	Traumas Adolescentes

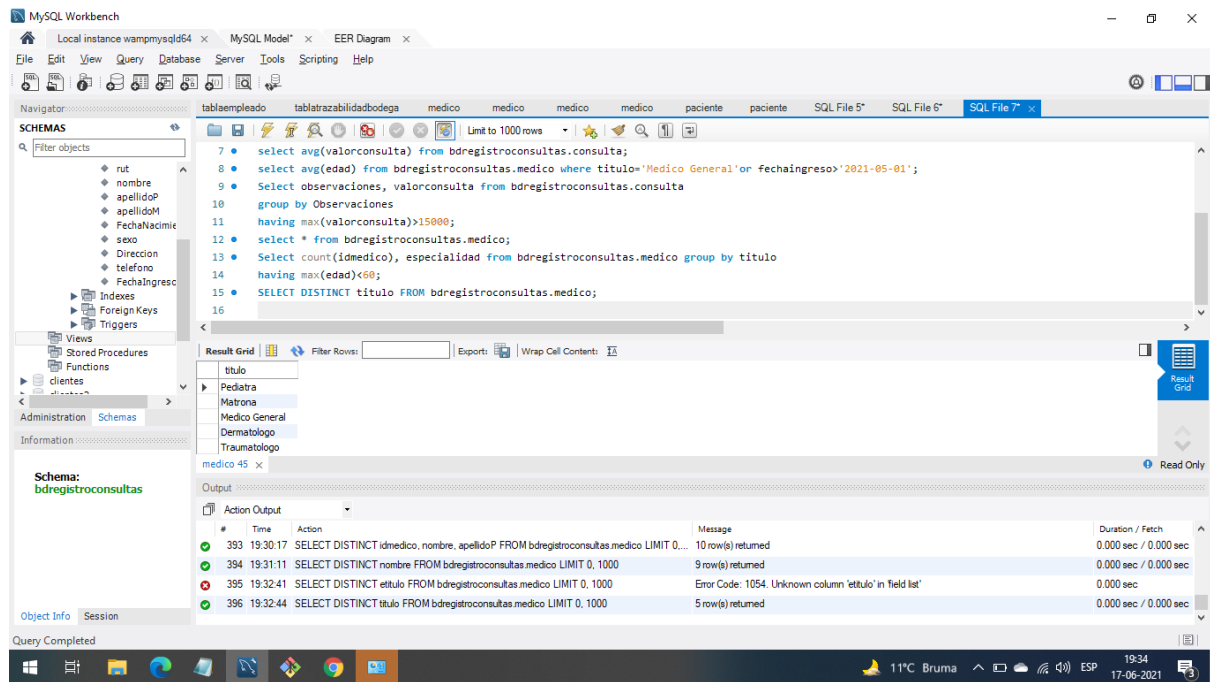
The 'Output' tab at the bottom shows the execution log with the following messages:

#	Time	Action	Message	Duration / Fetch
388	19:21:36	Select count(idconsulta), especialidad from bdregistroconsultas.medico group by titulo having ...	Error Code: 1054. Unknown column 'idconsulta' in field list'	0.000 sec
389	19:21:58	Select count(idmedico), especialidad from bdregistroconsultas.medico group by titulo having ...	3 row(s) returned	0.000 sec / 0.000 sec
390	19:23:31	Select count(idmedico), especialidad from bdregistroconsultas.medico group by titulo having ...	Error Code: 1054. Unknown column 'edad' in having clause'	0.000 sec
391	19:23:38	Select count(idmedico), especialidad from bdregistroconsultas.medico group by titulo having ...	3 row(s) returned	0.015 sec / 0.000 sec

## Función DISTINCT

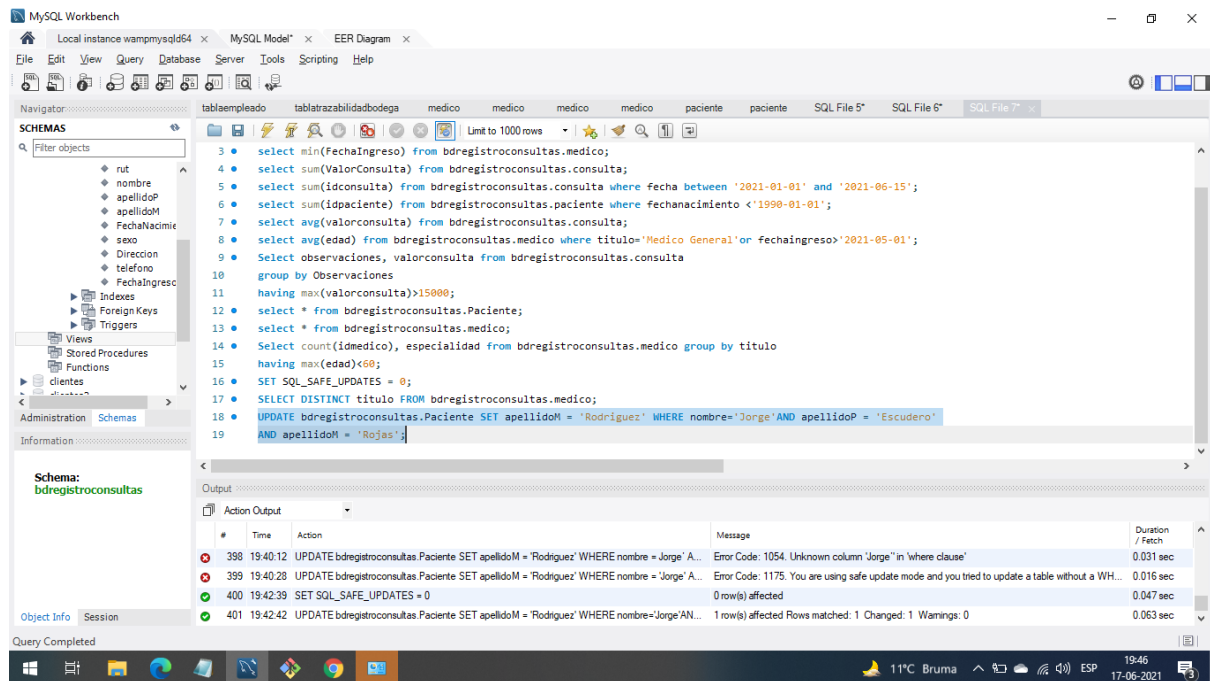
Query para seleccionar todas las especialidades sin que se repitan y que se atienden en la consulta.

SELECT DISTINCT titulo FROM bdregistroconsultas.medico;



## UPDATE

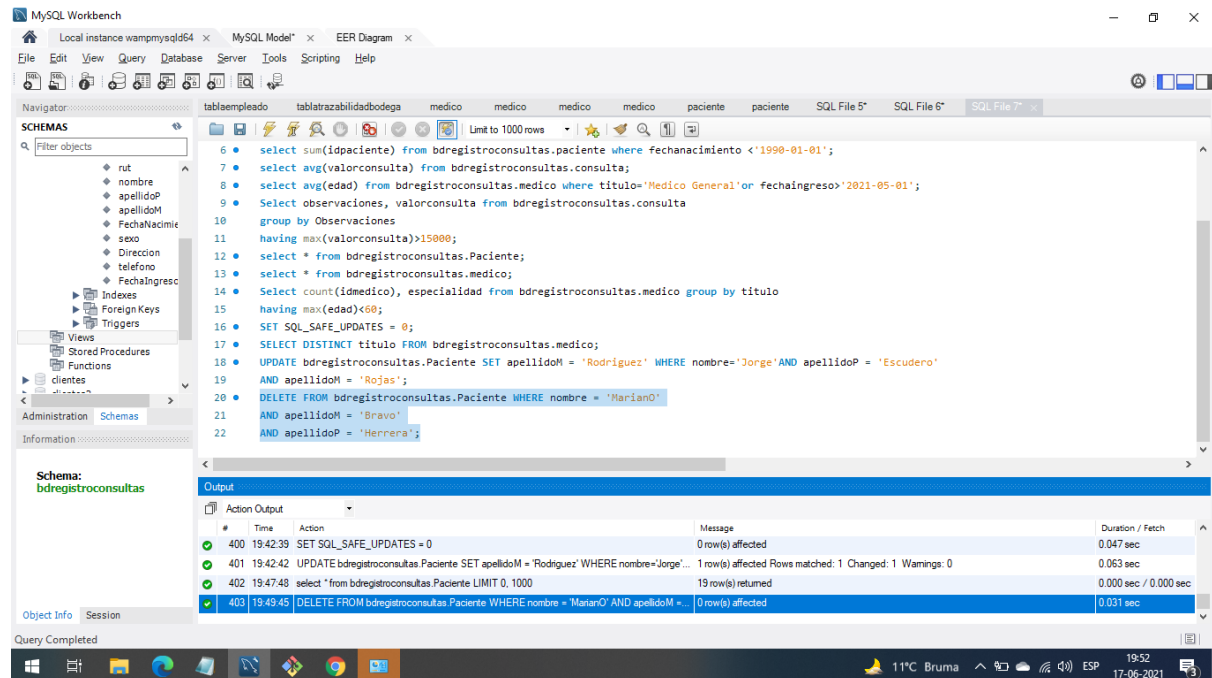
Query para modificar el apellidoM de un paciente que por error se ingresó como 'Rojas' en vez de 'Rodriguez'.  
 Update bdregistroconsultas.Paciente SET apellidoM = 'Rodriguez' WHERE nombre='Jorge' AND apellidoP = 'Escudero' AND apellidoM = 'Rojas';



## Delete

Query para eliminar un registro mal ingresado de la tabla Pacientes.

```
-DELETE FROM bdregistroconsultas.Paciente WHERE nombre = 'Mariano'  
AND apellidoM = 'Bravo'  
AND apellidoP = 'Herrera';
```



## SubConsultas

Query para traer los datos personales (nombre, ApellidoP, ApellidoM y edad) del médico más joven.

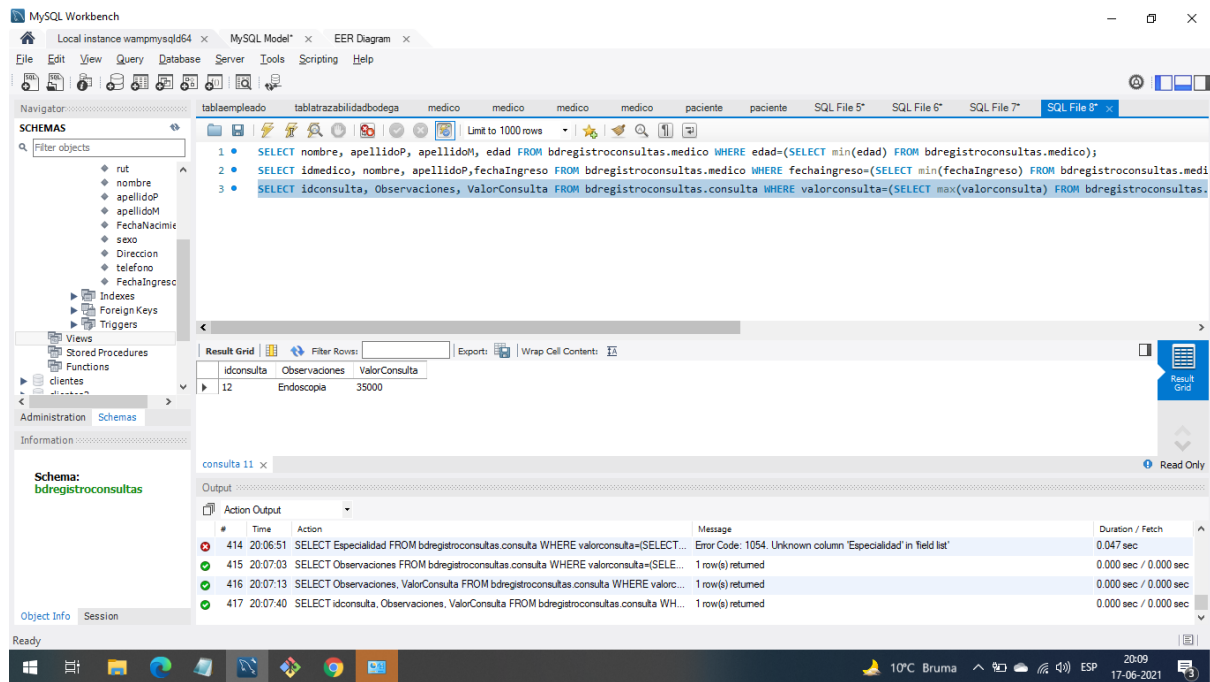
```
-SELECT nombre, apellidoP, apellidoM, edad FROM bdregistroconsultas.medico WHERE edad=(SELECT  
min(edad) FROM bdregistroconsultas.medico);
```

Query para consultar los datos personales (idmedico, nombre y apellido paterno) y fecha de ingreso del médico antiguo que haya sido ingresado a la bd.

```
-SELECT idmedico, nombre, apellidoP, fechaIngreso FROM bdregistroconsultas.medico WHERE  
fechaIngreso=(SELECT min(fechaIngreso) FROM bdregistroconsultas.medico);
```

Query para consultar el idConsulta y la observaciones de la consulta con el valor más alto:

```
-SELECT idconsulta, Observaciones, ValorConsulta FROM bdregistroconsultas.consulta WHERE  
valorconsulta=(SELECT max(valorconsulta) FROM bdregistroconsultas.consulta);
```



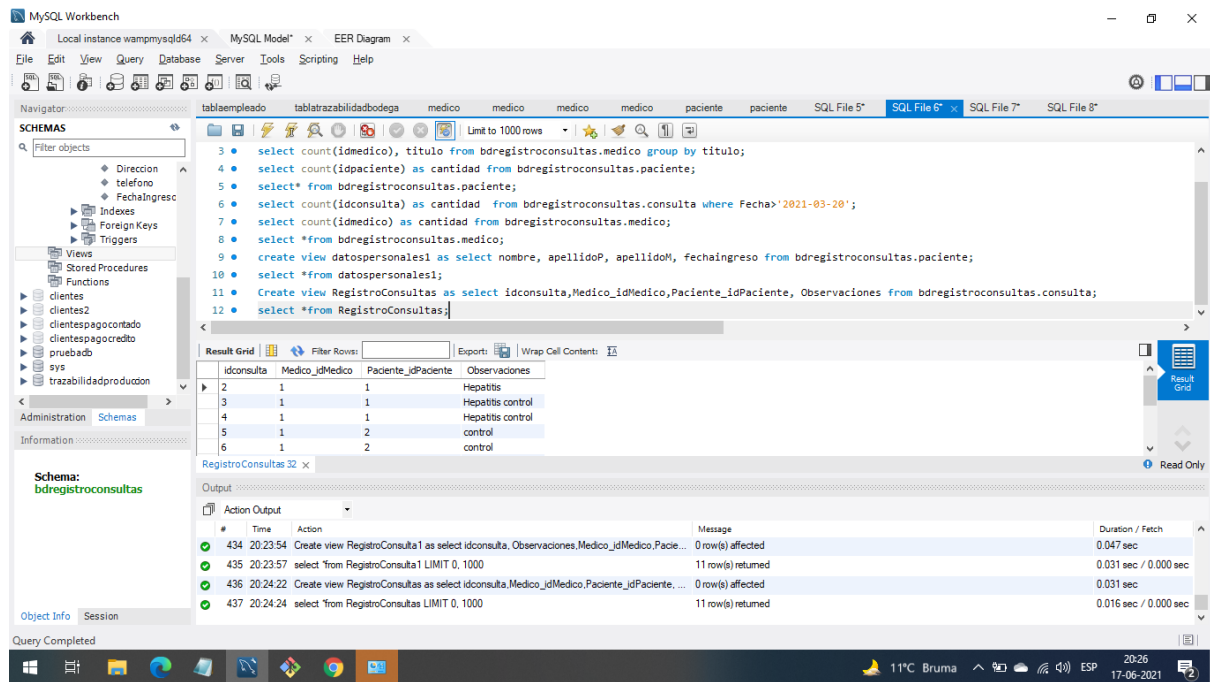
## Creación de Vistas

Query para la creación de una nueva tabla que nos muestre los datos personales de todas los pacientes ingresados;

-Create view datospersonales as select nombre, apellidoP, apellidoM, fechaingreso from bdregistroconsultas.paciente;  
select \*from datospersonales;

Query para la creación de una tabla de registro por consulta que nos traiga el id de la consulta, el id del médico, el id del paciente y las observaciones.

-Create view RegistroConsultas as select idconsulta,Medico\_idMedico,Paciente\_idPaciente, Observaciones from bdregistroconsultas.consulta;  
select \*from RegistroConsultas;



## Joins

Query para extraer un informe con los datos de las consultas registradas, que nos traiga los datos personales de los pacientes que asistieron a la consulta, las observaciones y el médico que los atendió.

```

-select consulta.idconsulta, paciente.idPaciente, paciente.nombre, Paciente.apellidoP,
Observaciones, Medico_idMedico from bdregistroconsultas.consulta inner join
bdregistroconsultas.paciente on consulta.Paciente_idPaciente=paciente.idPaciente;

```

Query para consultar el detalle de pago de la consulta por cada id de médico, nombre y apellidoP

```

-consulta.Paciente_idPaciente=paciente.idPaciente;
select idmedico, medico.nombre, medico.apellidoP, consulta.idConsulta, consulta.ValorConsulta,
consulta.EstadoConsulta from bdregistroconsultas.medico inner join bdregistroconsultas.consulta on
medico.idmedico=consulta.Medico_idMedico;

```

