**DDL:**

1. CREATE VIEW V\_ALGUNOS\_PROVEEDORES AS SELECT p.NUMERO, pr.PNRO FROM proveedores AS p, productos AS pr WHERE p.LOCALIDAD != pr.LOCALIDAD;
2. ALTER TABLE productos ADD IMPORTADOR VARCHAR(30);
3. CREATE VIEW V\_PROVEEDORES\_WILDE AS SELECT \* FROM proveedores WHERE LOCALIDAD='Wilde';
4. CREATE TABLE DEPARTAMENTOS (CODIGO INT NOT NULL, NOMBRE VARCHAR(20) NOT NULL, CONSTRAINT pk\_cod\_dep PRIMARY KEY(CODIGO));

CREATE TABLE EMPLEADOS (LEGAJO INT NOT NULL, DEPARTAMENTO INT NOT NULL, NOMBRE VARCHAR(20) NOT NULL, APELLIDO VARCHAR(20) NOT NULL, DNI INT NOT NULL, DOMICILIO VARCHAR(20),

CONSTRAINT pk\_legajo PRIMARY KEY(LEGAJO),

CONSTRAINT pk\_depto PRIMARY KEY(DEPARTAMENTO),

CONSTRAINT fk\_depto FOREIGN KEY(DEPARTAMENTO) REFERENCES DEPARTAMENTOS (CODIGO));

CREATE TABLE PACIENTES (FICHA INT NOT NULL, MEDICAMENTO INT NOT NULL, NOMBRE VARCHAR(20) NOT NULL, APELLIDO VARCHAR(20) NOT NULL, DNI INT NOT NULL, DOMICILIO VARCHAR(20),

CONSTRAINT pk\_ficha PRIMARY KEY(FICHA),

CONSTRAINT pk\_medic PRIMARY KEY(MEDICAMENTO));

CREATE TABLE MEDICAMENTOS (CODIGO INT NOT NULL, NOMBRE VARCHAR(20) NOT NULL, LABORATORIO VARCHAR(20) NOT NULL,

CONSTRAINT pk\_cod\_medic PRIMARY KEY(CODIGO),

CONSTRAINT fk\_cod\_medic FOREIGN KEY(CODIGO) REFERENCES PACIENTES (MEDICAMENTO));

**DML (primera parte):**

1. SELECT \* FROM productos;
2. SELECT \* FROM proveedores WHERE LOCALIDAD='Capital Federal';
3. SELECT \* FROM `prov-prod` WHERE CANTIDAD>=200 AND CANTIDAD<=300;
4. SELECT PNRO FROM `prov-prod` WHERE NUMERO IN (SELECT NUMERO FROM proveedores WHERE LOCALIDAD='Avellaneda');
5. SELECT SUM(CANTIDAD) FROM `prov-prod` WHERE PNRO='1' AND NUMERO='103';
6. SELECT PNRO,LOCALIDAD FROM productos WHERE LOCALIDAD LIKE '\_a%';
7. SELECT PRECIO FROM productos WHERE PNRO IN (SELECT PNRO from `prov-prod` WHERE NUMERO='102');
8. SELECT DISTINCT LOCALIDAD FROM (SELECT LOCALIDAD FROM proveedores UNION ALL SELECT LOCALIDAD FROM productos) a;
9. UPDATE productos SET TAMAÑO='Chico' WHERE TAMAÑO='Mediano';
10. DELETE FROM productos WHERE NOT PNRO IN (SELECT PNRO FROM `prov-prod`);
11. INSERT INTO proveedores(NUMERO, NOMBRE, LOCALIDAD) VALUES ('107','Rosales','Wilde');

**DML (segunda parte):**

1. SELECT SUM(cantidad) AS CANTIDAD FROM item\_ventas;
2. SELECT SUM(cantidad) AS CANTIDAD FROM item\_ventas WHERE codigo\_producto='c';
3. SELECT iv.codigo\_producto,p.nombre\_producto,SUM(iv.cantidad) AS cantidad FROM item\_ventas iv,productos p WHERE iv.codigo\_producto=p.codigo\_producto GROUP BY codigo\_producto ORDER BY cantidad DESC;
4. SELECT iv.codigo\_producto,p.nombre\_producto,SUM(iv.cantidad) AS cantidad FROM item\_ventas iv,productos p WHERE iv.codigo\_producto=p.codigo\_producto GROUP BY codigo\_producto HAVING SUM(iv.cantidad) > 30 ORDER BY p.nombre\_producto;
5. SELECT v.codigo\_cliente,c.nombre,COUNT(v.numero\_factura) AS compras FROM ventas v,clientes c WHERE v.codigo\_cliente=c.codigo\_cliente GROUP BY codigo\_cliente ORDER BY compras DESC;
6. SELECT codigo\_producto, avg(cantidad) as promedio FROM Ventas as v, item\_ventas as iv WHERE iv.numero\_factura = v.numero\_factura AND codigo\_cliente=1 GROUP BY codigo\_producto;

**DML (tercera parte consultas simples):**

1. SELECT cod\_oficina, descripcion AS nombre FROM oficinas;
2. SELECT cod\_producto, descripcion, precio, precio\*1.21 AS iva FROM productos;
3. SELECT cod\_empleado,apellido,nombre,fecha\_nacimiento AS 'fecha de cumpleaños',YEAR(NOW())-YEAR(fecha\_nacimiento) AS edad FROM empleados;
4. SELECT \* FROM empleados WHERE cod\_jefe IS NOT NULL;
5. SELECT \* FROM empleados WHERE nombre='Maria' ORDER BY apellido;
6. SELECT \* FROM clientes WHERE razón\_social LIKE 'L%' ORDER BY cod\_cliente;
7. SELECT \* FROM pedidos WHERE fecha\_pedido LIKE '\_\_/03/\_\_\_\_' ORDER BY fecha\_pedido;
8. SELECT \* FROM oficinas WHERE codigo\_director IS NULL;
9. SELECT cod\_producto,descripcion,precio FROM productos ORDER BY precio LIMIT 4;
10. SELECT cod\_empleado,cuota FROM datos\_contratos ORDER BY cuota DESC LIMIT 3;

**DML (tercera parte consultas multitabla):**

1. SELECT p.cod\_producto,p.descripcion,f.razón\_social,p.cantidad\_stock FROM productos p,fabricantes f WHERE p.cod\_fabricante=f.cod\_fabricante ORDER BY f.razón\_social, p.descripcion;
2. SELECT p.cod\_pedido,p.fecha\_pedido,e.apellido,c.razón\_social FROM pedidos p,empleados e,clientes c WHERE p.cod\_empleado=e.cod\_empleado AND p.cod\_cliente=c.cod\_cliente;
3. SELECT e.cod\_empleado,e.apellido,d.cuota,o.descripcion AS oficina FROM empleados e,datos\_contratos d,oficinas o WHERE e.cod\_empleado=d.cod\_empleado AND e.cod\_oficina=o.cod\_oficina ORDER BY d.cuota DESC;
4. SELECT DISTINCT c.razón\_social FROM clientes c,pedidos p WHERE c.cod\_cliente=p.cod\_cliente AND p.fecha\_pedido LIKE '\_\_/04/\_\_\_\_';
5. SELECT DISTINCT p.cod\_producto,p.descripcion,p.precio FROM productos p,detalle\_pedidos d, pedidos pe WHERE p.cod\_producto=d.cod\_producto AND d.cod\_pedido=pe.cod\_pedido AND pe.fecha\_pedido LIKE '\_\_/03/\_\_\_\_';
6. SELECT e.cod\_empleado,e.apellido,YEAR(NOW())-YEAR(d.fecha\_contrato) AS antiguedad FROM empleados e,datos\_contratos d WHERE e.cod\_empleado=d.cod\_empleado AND YEAR(NOW())-YEAR(d.fecha\_contrato) > 10 ORDER BY antiguedad DESC;
7. SELECT c.cod\_cliente,c.razón\_social,COUNT(p.cod\_pedido) AS compras FROM clientes c,pedidos p WHERE c.cod\_cliente=p.cod\_cliente GROUP BY cod\_cliente HAVING COUNT(p.cod\_pedido) > 5 ORDER BY c.razón\_social;
8. SELECT DISTINCT c.cod\_cliente,c.razón\_social,p.descripcion AS producto FROM clientes c,productos p,pedidos pe, detalle\_pedidos d WHERE c.cod\_cliente=pe.cod\_cliente AND pe.cod\_pedido=d.cod\_pedido AND d.cod\_producto=p.cod\_producto ORDER BY c.razón\_social,p.descripcion;
9. SELECT p.cod\_producto, p.descripcion, p.cantidad\_stock, p.punto\_reposición, (p.punto\_reposición - p.cantidad\_stock)+10 AS 'cantidad a comprar', f.razón\_social FROM productos p,fabricantes f WHERE p.punto\_reposición - p.cantidad\_stock >= 0 AND p.cod\_fabricante=f.cod\_fabricante ORDER BY f.razón\_social,p.descripcion;
10. SELECT e.cod\_empleado,e.apellido,e.nombre,d.cuota FROM empleados e,datos\_contratos d WHERE e.cod\_empleado=d.cod\_empleado AND (d.cuota > 100000 OR d.cuota < 50000);

**Catalogo del sistema:**

1. SELECT TBNAME FROM SYSCOLUMNS WHERE NAME='LOCALIDAD';
2. SELECT COLCOUNT FROM SYSTABLES WHERE NAME=PRODUCTOS;
3. SELECT DISTINCT CREATOR, COUNT(CREATOR) AS TABLES FROM SYSTABLES GROUP BY CREATOR;
4. SELECT DISTINCT TBNAME FROM SYSINDEXES;

**Constraints:**

1. PACIENTES (CODIGO-PAC, APELLIDO-PAC, EDAD)

MEDICAMENTOS (CODIGO-MED, PRECIO-UNITARIO)

GASTOS (CODIGO-PAC, CODIGO-MED)

\* \*

1. CURSOS (NUMCURSO, TITULO)

OFRECIMIENTOS (NUMCURSO, NUMOFR, FECHA, AULA)

\*

PROFESORES (NUMCURSO, NUMOFR, NUMEMP)

\* \* \*

ESTUDIANTES (NUMCURSO, NUMOFR, NUMEMP, CALIFICACION)

\* \* \*

EMPLEADOS (NUMEMP, EMPLE-NOMBRE, PUESTO)

1. CREATE TABLE Empleados (cod\_emp INT NOT NULL, nombre VARCHAR(30) NOT NULL, apellido VARCHAR(30) NOT NULL, tipo\_doc INT NOT NULL, num\_doc INT NOT NULL, categoria VARCHAR(30) NOT NULL, cod\_ofic INT NOT NULL,

CONSTRAINT pk\_cod\_emp PRIMARY KEY(cod\_emp),

CONSTRAINT uk\_tipo\_num\_doc UNIQUE(tipo\_doc, num\_doc),

CONSTRAINT fk\_cod\_ofic FOREIGN KEY(cod\_ofic) REFERENCES oficinas(cod\_ofic),

CONSTRAINT ck\_cod\_emp CHECK(cod\_emp > 100 AND cod\_emp < 1000),

CONSTRAINT ck\_categoria CHECK(categoria in ('Senior', 'Semi Senior', 'Junior')));

CREATE TABLE Oficinas (cod\_ofic INT NOT NULL, descripcion VARCHAR(30) NOT NULL, CONSTRAINT pk\_cod\_ofic PRIMARY KEY(cod\_ofic));