

Universidad Mariano Gálvez

Ingeniería en Sistemas

Sexto Semestre

Base de datos

Tema:

TAREA 4



Nombre:

Rudy Jaser Samuel

Castellanos López

Fecha:

08/08/2024

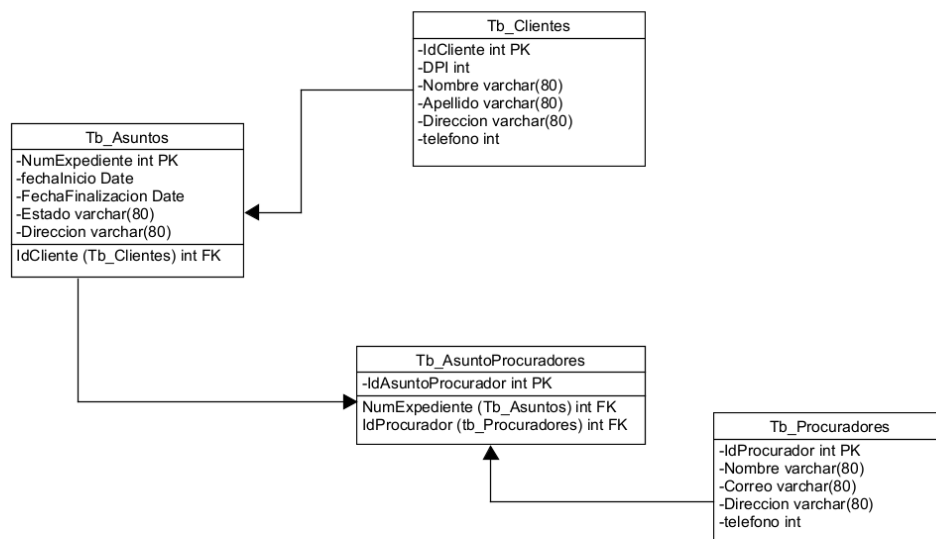
San Benito, Petén

Se pide realizar lo siguiente:

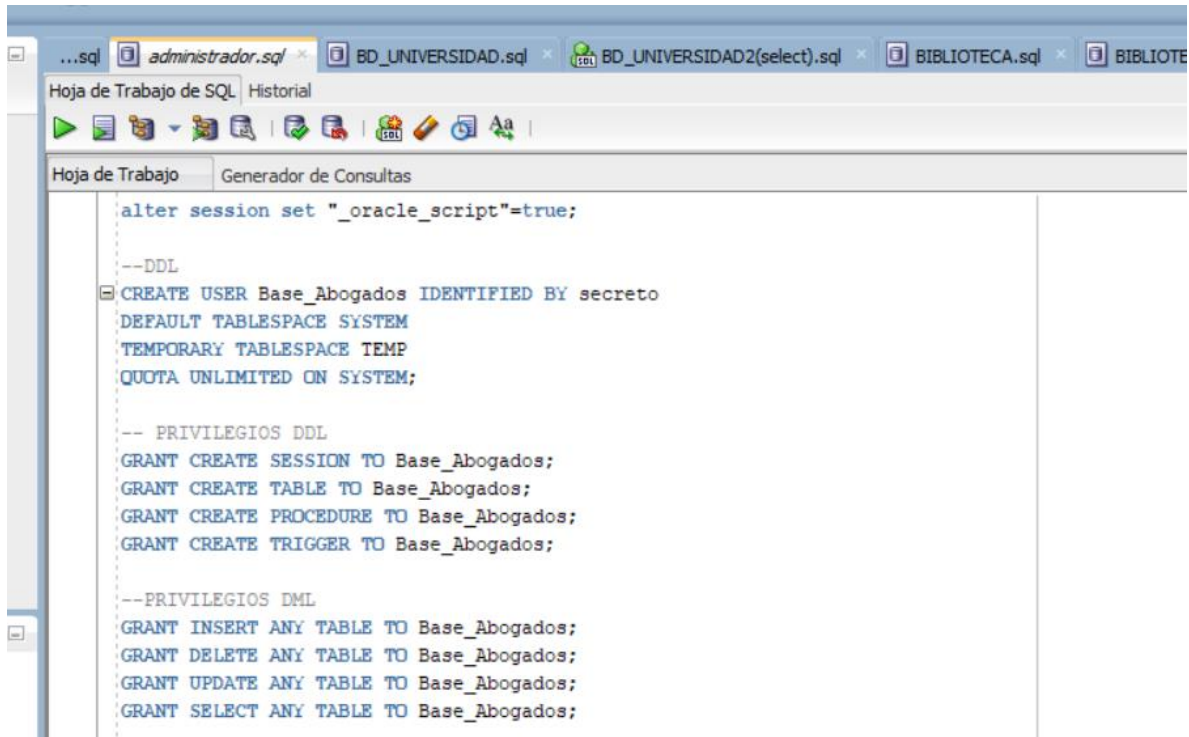
Se quiere diseñar una base de datos relacional para almacenar información sobre los asuntos que lleva un gabinete de abogados. Cada asunto tiene un número de expediente que lo identifica, y corresponde a un solo cliente. Del asunto se debe almacenar el período (fecha de inicio y fecha de archivo o finalización), su estado (en trámite, archivado, etc.), así como los datos personales del cliente al que pertenece (DPI, nombre, dirección, etc.). Algunos asuntos son llevados por uno o varios procuradores, de los que nos interesa también los datos personales.

- Realizar Diagrama Entidad-Relación
- Realizar consultas haciendo uso de al menos 5 funciones con Strings.
- Realizar consultas haciendo uso de al menos 5 funciones matemáticas.

1. Primero crearemos el Diagrama Entidad-Relación para poder tener el concepto de las tablas que vamos a hacer:



2. Lo siguiente será crear la conexión y la base de datos para almacenar los datos de las tablas mostradas en el diagrama:



The screenshot shows the SQL Developer interface with a script editor containing the following SQL code:

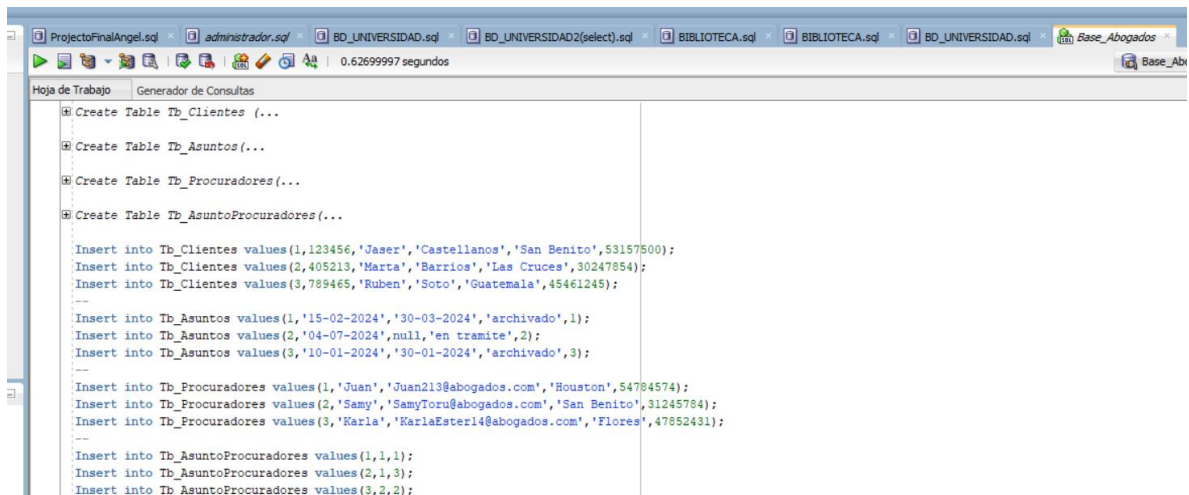
```
alter session set "_oracle_script"=true;

--DDL
CREATE USER Base_Abogados IDENTIFIED BY secreto
DEFAULT TABLESPACE SYSTEM
TEMPORARY TABLESPACE TEMP
QUOTA UNLIMITED ON SYSTEM;

-- PRIVILEGIOS DDL
GRANT CREATE SESSION TO Base_Abogados;
GRANT CREATE TABLE TO Base_Abogados;
GRANT CREATE PROCEDURE TO Base_Abogados;
GRANT CREATE TRIGGER TO Base_Abogados;

--PRIVILEGIOS DML
GRANT INSERT ANY TABLE TO Base_Abogados;
GRANT DELETE ANY TABLE TO Base_Abogados;
GRANT UPDATE ANY TABLE TO Base_Abogados;
GRANT SELECT ANY TABLE TO Base_Abogados;
```

3. Creamos las tablas e insertamos datos para poder hacer los Selects correspondientes:



The screenshot shows the SQL Developer interface with a script editor containing the following SQL code:

```
Create Table Tb_Clientes (...
Create Table Tb_Asuntos(...)
Create Table Tb_Procuradores(...)
Create Table Tb_AsuntoProcuradores(...)

Insert into Tb_Clientes values(1,123456,'Jaser','Castellanos','San Benito',53157500);
Insert into Tb_Clientes values(2,405213,'Marta','Barrios','Las Cruces',30247854);
Insert into Tb_Clientes values(3,789465,'Ruben','Soto','Guatemala',45461245);
--
Insert into Tb_Asuntos values(1,'15-02-2024','30-03-2024','archivado',1);
Insert into Tb_Asuntos values(2,'04-07-2024',null,'en tramite',2);
Insert into Tb_Asuntos values(3,'10-01-2024','30-01-2024','archivado',3);
--
Insert into Tb_Procuradores values(1,'Juan','Juan213@abogados.com','Houston',54784574);
Insert into Tb_Procuradores values(2,'Samy','SamyToru@abogados.com','San Benito',31245784);
Insert into Tb_Procuradores values(3,'Karla','KarlaEster14@abogados.com','Flores',47852431);
--
Insert into Tb_AsuntoProcuradores values(1,1,1);
Insert into Tb_AsuntoProcuradores values(2,1,3);
Insert into Tb_AsuntoProcuradores values(3,2,2);
```

4. Creamos las 5 Consultas de tipo String que nos pide:

4.1. Consulta “rpad” llama a los datos y si no tiene 15 rellena con "x"

```
--Funciones con String
select rpad(apellido,15,'x') as Apellido from Tb_Clientes;
```

Salida de Script x		Resultado de la Consulta x	
SQL Todas las Filas Recuperadas: 3 en 0.068 segundos			
APELLIDO			
1	Castellanosxxxxx		
2	Barriosxxxxxxxxx		
3	Sotoxxxxxxxxxxxxx		

4.2. Consulta “initcap” Cada letra inicial del String la pasa a mayúscula

```
select initcap(Estado) from Tb_Asuntos;
```

Salida de Script x		Resultado de la Consulta x	
SQL Todas las Filas Recuperadas: 3 en 0.008 segundos			
INITCAP(ESTADO)			
1	Archivado		
2	En Tramite		
3	Archivado		

4.3. Consulta “Concat” Concatena la columna nombre, con la de apellido

```
select concat(Nombre,apellido) from Tb_clientes;
```

Salida de Script x		Resultado de la Consulta x	
SQL Todas las Filas Recuperadas: 3 en 0.003 segundos			
CONCAT(NOMBRE,APELLIDO)			
1	JaserCastellanos		
2	MartaBarrios		
3	RubenSoto		

4.4. Consulta “replace” reemplaza el dato que se le indica por otro, sin afectar a la tabla

```
select replace (IdProcurador,1,'Abogado Juan') as IdProcurador from Tb_AsuntoProcuradores;
```

Resultado de la Consulta	
Todas las Filas Recuperadas: 3 en 0.004 segundos	
IDPROCURADOR	
1	Abogado Juan
2	3
3	2

4.5. Consulta “upper” convierte en mayúsculas toda los datos de la columna

```
select upper (correo) from Tb_procuradores;
```

Resultado de la Consulta	
Todas las Filas Recuperadas: 3 en 0.013 segundos	
UPPER(CORREO)	
1	JUAN213@ABOGADOS.COM
2	SAMYTORU@ABOGADOS.COM
3	KARLAESTER14@ABOGADOS.COM

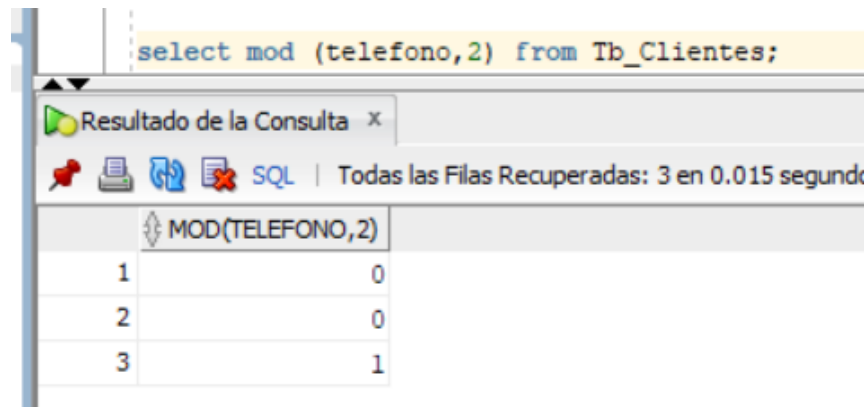
5. Creamos las 5 consultas con funciones matemáticas

5.1. Consulta “power” genera una potencia, el primer dato es el numero a potenciar y el segundo es el nivel de potencia

```
--Funciones Matematicas  
select power(DPI,2) from Tb_Clientes;
```

Resultado de la Consulta	
Todas las Filas Recuperadas: 3 en 0.036 segundos	
POWER(DPI,2)	
1	15241383936
2	164197575369
3	623254986225

- 5.2. La consulta “Mod” sirve para mostrar el residuo de la división en este caso de los datos de teléfono dividido 2



The screenshot shows a SQL query window with the query: `select mod (telefono,2) from Tb_Clientes;`. Below the query, the results are displayed in a table with the column header `MOD(TELEFONO,2)`. The table contains three rows with IDs 1, 2, and 3, and their corresponding modulo results: 0, 0, and 1.

	MOD(TELEFONO,2)
1	0
2	0
3	1

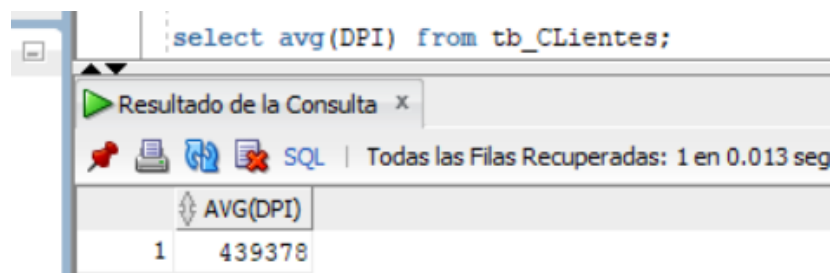
- 5.3. La consulta “Sqrt” nos da la raíz cuadrada de los datos que le demos



The screenshot shows a SQL query window with the query: `select sqrt(telefono) from Tb_procuradores;`. Below the query, the results are displayed in a table with the column header `SQRT(TELEFONO)`. The table contains three rows with IDs 1, 2, and 3, and their corresponding square root values.

	SQRT(TELEFONO)
1	7401.66021916704847060016703588093639897
2	5589.792840526382154774354913941238926868
3	6917.545157062583281449194255898303125317

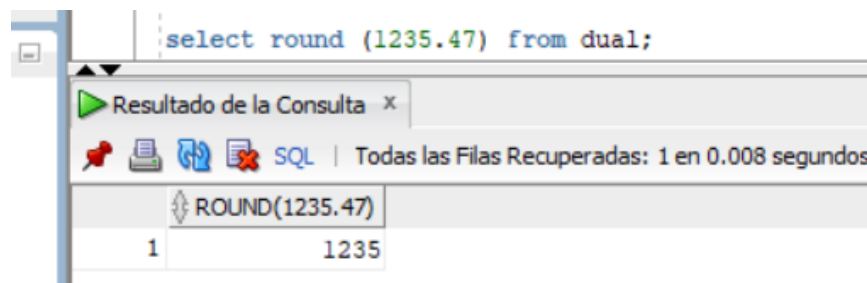
- 5.4. Consulta “avg” devuelve un promedio de todos los datos ingresados



The screenshot shows a SQL query window with the query: `select avg(DPI) from tb_Clientes;`. Below the query, the results are displayed in a table with the column header `AVG(DPI)`. The table contains one row with ID 1 and the average value 439378.

	AVG(DPI)
1	439378

- 5.5. Consulta “round” sirve para aproximar, si el numero después del punto es mayor a 5 aproxima, sino no.



The screenshot shows a SQL query window with the query: `select round (1235.47) from dual;`. Below the query, the results are displayed in a table with the column header `ROUND(1235.47)`. The table contains one row with ID 1 and the rounded value 1235.

	ROUND(1235.47)
1	1235