
TRABAJO PRÁCTICO 04

Fundamento de bases de datos: Unidad 04

Alumno

González, Matías Ezequiel

Contents

1	Lenguaje DML	3
1.0.1	Creación de tabla	3
1.0.2	Ejercicio 1	3
1.0.3	Ejercicio 2	4
1.0.4	Ejercicio 3	5
1.0.5	Ejercicio 4	5
1.0.6	Ejercicio 5	5
1.0.7	Ejercicio 6	6
1.0.8	Ejercicio 7	7
1.0.9	Ejercicio 8	8

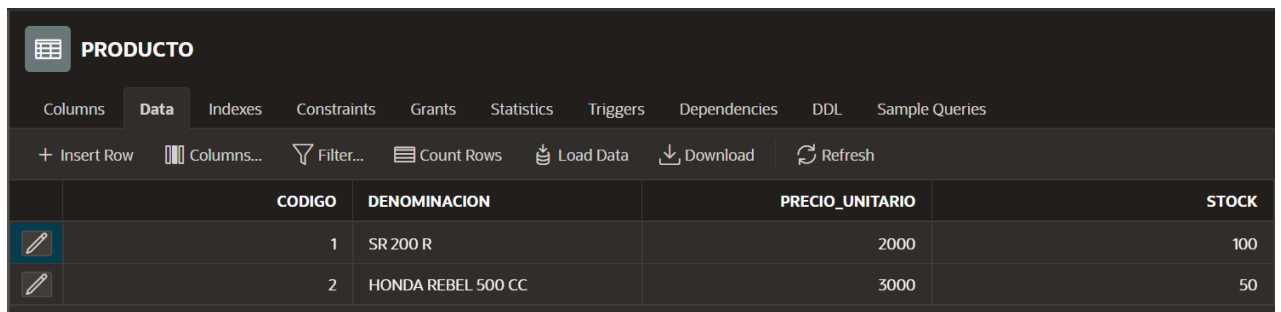
1 Lenguaje DML

1.0.1 Creación de tabla

Creo la tabla con la cual voy a trabajar en todos los ejercicios.

Listing 1: Creo la tabla e inserto columnas.

```
1 create table PRODUCTO (  
2     Codigo number(8),  
3     Denominacion varchar2(100),  
4     Precio_Unitario number(6,2),  
5     Stock number(6)  
6 );  
7 insert into PRODUCTO(Codigo, Denominacion, Precio_Unitario, Stock)  
8 values('00000001', 'SR 200 R', '2000', '100');  
9 insert into PRODUCTO  
10 values('00000002', 'HONDA REBEL 500 CC', '3000', '50');
```



	CODIGO	DENOMINACION	PRECIO_UNITARIO	STOCK
	1	SR 200 R	2000	100
	2	HONDA REBEL 500 CC	3000	50

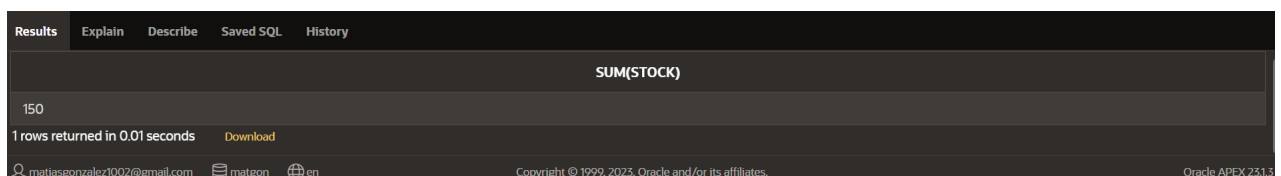
Figure 1: Comprobación de tabla.

1.0.2 Ejercicio 1

Consigna: Escribir una sentencia que contenga una función de grupo.

Listing 2: Sentencia con función grupo.

```
1 select sum(STOCK)  
2 from PRODUCTO  
3 where DENOMINACION is not null;
```



SUM(STOCK)
150

1 rows returned in 0.01 seconds [Download](#)

matiasgonzalez1002@gmail.com matgon en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.1.3

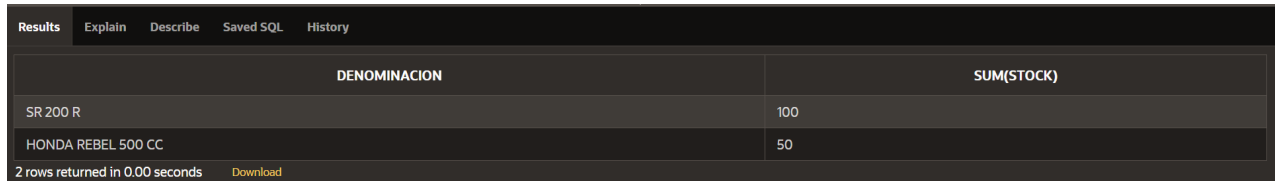
Figure 2: Resultado de sentencia de grupo.

1.0.3 Ejercicio 2

Consigna: Escribir una sentencia que contenga una función de grupo y la cláusula GROUP BY.

Listing 3: Sentencia para agrupar en denominaciones la sumatorias de stock.

```
1 select DENOMINACION , sum(STOCK)
2   from PRODUCTO
3  group by DENOMINACION;
```



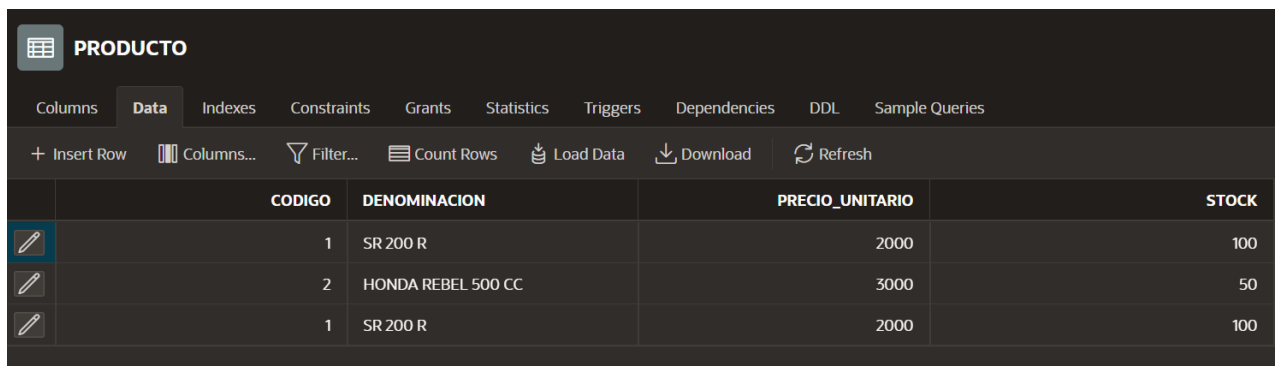
DENOMINACION	SUM(STOCK)
SR 200 R	100
HONDA REBEL 500 CC	50

Figure 3: Resultado de última sentencia.

Con la poca información que tengo en mi tabla no tiene mucha gracia. Sumo una fila y repito la sentencia.

Listing 4: Añado otras fila.

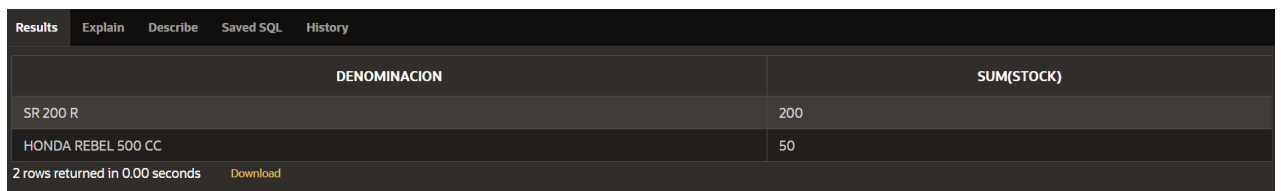
```
1 insert into PRODUCTO(Codigo, Denominacion, Precio_Unitario, Stock)
2   values('00000001', 'SR 200 R', '2000', '100');
```



	CODIGO	DENOMINACION	PRECIO_UNITARIO	STOCK
	1	SR 200 R	2000	100
	2	HONDA REBEL 500 CC	3000	50
	1	SR 200 R	2000	100

Figure 4: Compruebo mi nueva tabla.

Vuelvo a aplicar la sentencia de grupo.



DENOMINACION	SUM(STOCK)
SR 200 R	200
HONDA REBEL 500 CC	50

Figure 5: Resultado de la sentencia grupo.

1.0.4 Ejercicio 3

Consigna: Escribir una sentencia que contenga una función de grupo y la cláusula GROUP BY y HAVING.

Listing 5: Sentencia con cláusulas Group by y Having.

```
1 select DENOMINACION , sum(STOCK)
2   from PRODUCTO
3  group by DENOMINACION
4   having sum(STOCK) > 100;
```

Results

Explain

Describe

Saved SQL

History

DENOMINACION		SUM(STOCK)
SR 200 R		200

1 rows returned in 0.00 seconds

Download

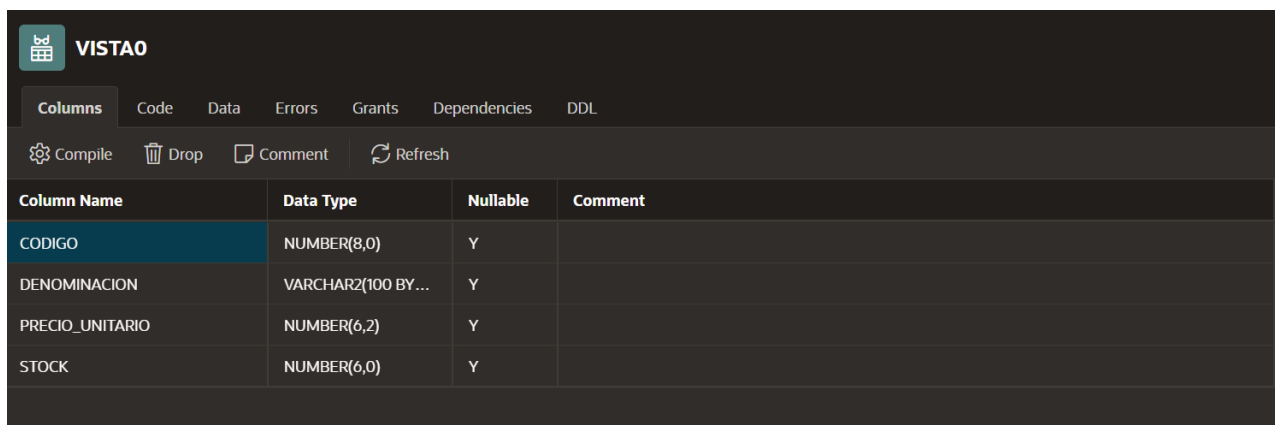
Figure 6: Resultado de sentencia Group by y Having.

1.0.5 Ejercicio 4

Consigna: Escribir una sentencia que cree una vista “simple” de la base de datos.

Listing 6: Creo vista con características de vista simple.

```
1 create view VISTA0 as
2 select CODIGO ,
3        DENOMINACION ,
4        PRECIO_UNITARIO ,
5        STOCK
6 from PRODUCTO
7 where STOCK > 0;
```



VISTA0			
Columns	Code	Data	Errors
Compile Drop Comment Refresh			
Column Name	Data Type	Nullable	Comment
CODIGO	NUMBER(8,0)	Y	
DENOMINACION	VARCHAR2(100 BY ...)	Y	
PRECIO_UNITARIO	NUMBER(6,2)	Y	
STOCK	NUMBER(6,0)	Y	

Figure 7: Comprobación e la anterior vista.

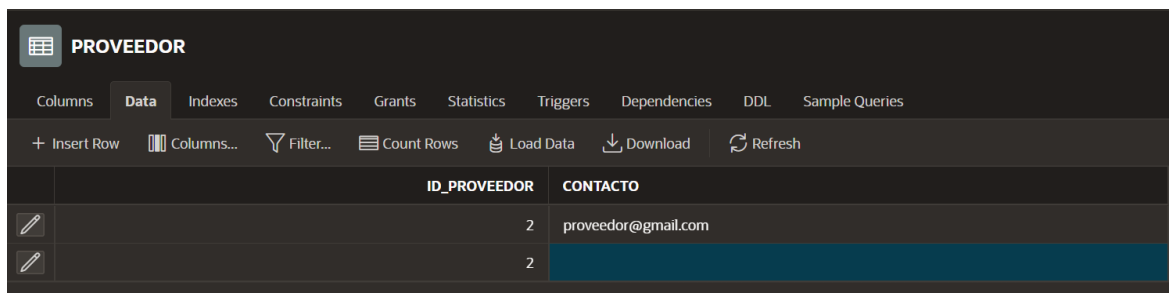
1.0.6 Ejercicio 5

Consigna: Escribir una sentencia que cree una vista “doble” de la base de datos.

Creo otra tabla para hacer una vista doble.

Listing 7: Creo tabla y le añado información a sus filas.

```
1 create table PROVEEDOR(  
2     ID_PROVEEDOR number(8),  
3     CONTACTO varchar2(100),  
4 );  
5  
6 insert into PROVEEDOR  
7     values('00000002', 'proveedor@gmail.com');  
8  
9 insert into PROVEEDOR  
10    values('00000002', '');
```



	ID_PROVEEDOR	CONTACTO
	2	proveedor@gmail.com
	2	

Figure 8: Comprobación de nueva tabla.

Creo la requerida vista doble.

Listing 8: Creo la vista doble.

```
1 create view VISTAD2 as  
2 select CODIGO, DENOMINACION, PRECIO_UNITARIO, STOCK, P.CONTACTO  
3     from PRODUCTO A, PROVEEDOR P  
4     where A.CODIGO = P.ID_PROVEEDOR and P.CONTACTO is not null;
```



CODIGO	DENOMINACION	PRECIO_UNITARIO	STOCK	CONTACTO
2	HONDA REBEL 500 CC	3000	50	proveedor@gmail.com

Figure 9: Comprobación de nueva tabla doble.

1.0.7 Ejercicio 6

Consigna: . Escribir una sentencia que contenga la cláusula ORDER BY.

Listing 9: Pido la información de denominacion y stock de mi tabla producto en orden ascendente en el stock.

```
1 select DENOMINACION ,
2     STOCK
3     from PRODUCTO
4 order by STOCK asc;
```

Results

Explain

Describe

Saved SQL

History

DENOMINACION	STOCK
HONDA REBEL 500 CC	50
SR 200 R	100
SR 200 R	100

3 rows returned in 0.00 seconds

Download

Figure 10: Resultado obtenido.

1.0.8 Ejercicio 7

Consigna: Escribir una sentencia que contenga la cláusula ROWNUM.

Listing 10: Setencia con cláusula Rownum.

```
1 select DENOMINACION ,
2     STOCK
3     from PRODUCTO
4 where rownum <= 2;
```

Results

Explain

Describe

Saved SQL

History

DENOMINACION	STOCK
SR 200 R	100
HONDA REBEL 500 CC	50

2 rows returned in 0.01 seconds

Download

Figure 11: Resultado de sentencia con rownum.

Recuerdo mi tabla para comprobar el sentido de la sentencia y el resultado.

PRODUCTO

Columns

Data

Indexes

Constraints

Grants

Statistics

Triggers

Dependencies

DDL

Sample Queries

+ Insert Row

Columns...

Filter...

Count Rows

Load Data

Download

Refresh

	CODIGO	DENOMINACION	PRECIO_UNITARIO	STOCK
<div></div>	1	SR 200 R	2000	100
<div></div>	2	HONDA REBEL 500 CC	3000	50
<div></div>	1	SR 200 R	2000	100

Figure 12: Mi tabla a la cual le apliqué la sentencia rownum.

1.0.9 Ejercicio 8

Consigna: Escribir una sentencia que contenga la cláusula DISTINCT.

Listing 11: Sentencia que entrega las denominaciones de mi tabla producto sin repetirse.

```
1 select distinct(DENOMINACION)
2   from(PRODUCTO);
```

DENOMINACION	
SR 200 R	
HONDA REBEL 500 CC	
2 rows returned in 0.01 seconds Download	

Figure 13: Comprobación de la sentencia distinct.

PRODUCTO					
Columns	Data	Indexes	Constraints	Grants	Statistics
Triggers	Dependencies	DDL	Sample Queries		
+ Insert Row	Columns...	Filter...	Count Rows	Load Data	Download
Refresh					
	CODIGO	DENOMINACION	PRECIO_UNITARIO	STOCK	
	1	SR 200 R	2000	100	
	2	HONDA REBEL 500 CC	3000	50	
	1	SR 200 R	2000	100	

Figure 14: Reviso mi tabla original para ver si tiene sentido mi resultado.

Tiene sentido.