

# UAI

Universidad Abierta Interamericana



## SEGUNDO EXAMEN DE LA MATERIA: BASES DE DATOS

FECHA:	03/07/2023		
ALUMNO/A:	Gianluca Carlini		
LEGAJO:	B00045873-T4	DNI:	41.655.321
CURSO:	Base de Datos	TURNO:	Distancia
CARRERA:	Analista Programador		
PROFESOR/A:	Dra. Ing. Roxana Martínez		
MODALIDAD:	Individual – Teórico – Práctico		

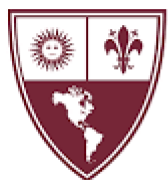
### UNIDADES A EVALUAR DEL PROGRAMA DE LA MATERIA:

- Unidad 4: Diseñar una estructura de datos eficiente
- Unidad 6: Definiendo la Estructura De La Base De Datos
- Unidad 7: Introducción a mejoras en el rendimiento de una base de datos
- Unidad 8: Big Data - NoSQL

### CRITERIOS DE CALIFICACIÓN:

Para aprobar el examen deberá sumar 60 puntos de un total de 100 (Teórico-Práctico), siendo, al menos el 60% de los aspectos conceptuales teóricos y al menos el 60% de los aspectos prácticos.

Duración del Examen: 1 semana aprox. (ver fecha en plataforma)



Universidad Abierta Interamericana



SEGUNDO EXAMEN DE LA MATERIA: BASES DE DATOS			
FECHA:			
ALUMNO/A:			
LEGAJO:		DNI:	
CURSO:		TURNO:	
CARRERA:			
PROFESOR/A:	Dra. Ing. Roxana Martínez		
MODALIDAD:	Individual – Teórico – Práctico		

### PARTE TEÓRICA (10 PTS)

Debe obtener al menos 6/10 para la aprobación de esta parte. Completar en el espacio asignado:

#### UNIDAD 4: DISEÑAR UNA ESTRUCTURA DE DATOS EFICIENTE (2 puntos)

1. Indique 5 características de porque se debe normalizar una base de datos relacional.

Se debe normalizar una base de datos relacional porque:

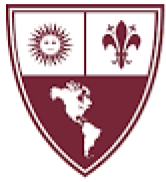
- a) Eliminar toda redundancia de datos.
- b) Reducir o eliminar anomalías de manipulación de datos.
- c) Recuperar la información de forma eficiente.
- d) Facilidad de mantenimiento.
- e) Mantenimiento de la consistencia.

#### UNIDAD 6: DEFINIENDO LA ESTRUCTURA DE LA BASE DE DATOS (2 puntos)

2. ¿Qué se debe tener en cuenta para la definición de la estructura de datos? Desarrolle.

A la hora de la definición de la estructura de datos se debe tener en cuenta los siguientes puntos:

- Requisitos del sistema: Debemos comprender los requerimientos del sistema y las necesidades de los usuarios como por ejemplo que tipo de datos se almacenarán, qué operaciones se realizarán sobre ellos, etc.
- Normalización: Es un proceso para garantizar la integridad y eficiencia de la base de datos. A partir de la división de la información en tablas relacionadas, evitaremos la redundancia y mantendremos la consistencia de los datos.
- Entidades y Relaciones: Es importante que identifiquemos las entidades con sus



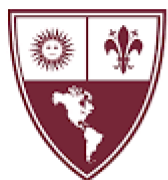
# UAI

**Universidad Abierta Interamericana**



respectivos atributos que se almacenarán en la base de datos para luego establecer relaciones entre ellos.

- Definición de atributos y tipos de datos: Los atributos representan las propiedades de las entidades. Se deben definir los atributos y asignarles un tipo de dato adecuado como texto, número, etc según los requerimientos.
- Claves Primarias y Foráneas: Para establecer las relaciones, se deben definir las PK y FK. Una PK es un atributo que identifica de manera única cada registro en una tabla. En cambio, una FK son atributos que hacen referencia a la clave primaria de otra tabla.
- Seguridad: Se deben aplicar medidas de seguridad como restricciones de acceso, encriptación y auditoría de registros para proteger la información sensible.
- Mantenibilidad: La estructura de datos debe ser fácil de mantener y permitir modificaciones futuras. Debe ser lo suficientemente flexible para adaptarse a cambios en los requisitos del sistema.



# UAI

Universidad Abierta Interamericana



SEGUNDO EXAMEN DE LA MATERIA: BASES DE DATOS			
FECHA:			
ALUMNO/A:			
LEGAJO:		DNI:	
CURSO:		TURNO:	
CARRERA:			
PROFESOR/A:	Dra. Ing. Roxana Martínez		
MODALIDAD:	Individual – Teórico – Práctico		

## UNIDAD 6: DEFINIENDO LA ESTRUCTURA DE LA BASE DE DATOS

### 3. Identifique un problema de anomalía de actualización (2 puntos)

Un problema de anomalía de actualización ocurre cuando hay información redundante almacenada en diferentes ubicaciones y una actualización parcial de esos datos resulta inconsistente. Un ejemplo de una anomalía de actualización puede ser el siguiente:

En una BD se registra información de productos y pedidos en dos tablas: “Productos” y “Pedidos”. La tabla “Productos” contiene los detalles de cada producto, incluido su precio. La tabla “Pedidos” registro los pedidos realizados por los clientes y la cantidad de cada producto pedido.

Ahora bien, consideremos actualizar el precio de un producto de la tabla “Productos”, pero no se actualizan los pedidos anteriores que incluyen ese producto en la tabla “Pedidos”. Como resultado, los pedidos anteriores mostrarán el precio anterior en lugar del precio actualizado. Por ende, se genera una anomalía de actualización porque el precio del producto es inconsistente entre las tablas y los registros relacionados.

## UNIDAD 7: INTRODUCCIÓN A MEJORAS EN EL RENDIMIENTO DE UNA BASE DE DATOS (2 puntos)

### 4. Identifique 5 características de mejoras en el rendimiento de una base de datos

- Índices. Son estructuras auxiliares que ayudan a acelerar las consultas en una base de datos. Puede mejorar significativamente el rendimiento, ya que se evita la necesidad de realizar una búsqueda secuencial en toda la tabla.
- Menor número de tablas.
- Redundancia de datos.
- Mejora en la velocidad de consultas.
- Menor integridad de datos.

### **UNIDAD 8: BIG DATA – NOSQL (2 puntos)**

#### **5. Identifique 4 ventajas y desventajas de NoSQL y SQL**

##### **Ventajas SQL:**

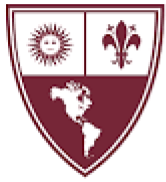
- Procesamiento de consultas más rápido. SQL puede recuperar gran cantidad de datos de forma rápida y eficiente.
- Flexibilidad. SQL permite realizar una amplia gama de operaciones en base de datos como consultas, inserciones, actualizaciones y eliminaciones de datos.
- Independencia de los datos: Permite a los usuarios acceder y manipular datos sin preocuparse por la estructura subyacente de la base de datos.
- Seguridad: Los sistemas de gestión de base de datos suelen tener mecanismos para autenticar usuarios, asignar permisos y controlar el acceso a los datos en función de roles y privilegios previamente definidos.

##### **Desventajas SQL:**

- Dificultades de crecimiento. Cuando estas bases de datos comienzan a crecer en volumen, el almacenamiento y el costo del mantenimiento se convierten en un problema de alto costo.
- Complejidad en la instalación. Algunas bases de datos SQL se ven condicionadas por el SO en el cual van a funcionar y los requisitos mínimos de funcionamiento de los servidores.
- Dificultad en la interfaz. La interfaz de una base de datos SQL son más complejas.
- Costo. SQL puede ser costoso en comparación con otras opciones de bases de datos.

##### **Ventajas NoSQL:**

- Flexibilidad para almacenar datos. No requiere de un esquema fijo predefinido. Además, facilita el alta, baja y modificación de campos en los datos almacenados.
- Rendimiento optimizado. Rendimiento óptimo en el manejo de grandes volúmenes de datos.
- Alta disponibilidad. Diseñadas para mantener servicios en línea sin interrupción, incluso en caso de fallos.
- Escalabilidad. Permite procesar grandes volúmenes de datos distribuyendo la carga de trabajo en múltiples nodos (escalabilidad horizontal).



# UAI

Universidad Abierta Interamericana



## SEGUNDO EXAMEN DE LA MATERIA: BASES DE DATOS

FECHA:			
ALUMNO/A:			
LEGAJO:		DNI:	
CURSO:		TURNO:	
CARRERA:			
PROFESOR/A:	Dra. Ing. Roxana Martínez		
MODALIDAD:	Individual – Teórico – Práctico		

### PARTE PRÁCTICA (10 PTS)

Debe obtener al menos 6/10 para la aprobación de esta parte.

En base a la siguiente tabla desnormalizada, realizar las siguientes consignas:

- 1) Identificar la primera forma normal (2 punto).
- 2) Identificar la segunda forma normal (1 punto).
- 3) Identificar la tercera forma normal (1 punto).
- 4) Identificar el DER Lógico del esquema final (1 punto).
- 5) Realizar un procedimiento almacenado con parámetros de entrada con una query con Group by (2 punto)
- 6) Realizar un procedimiento almacenado con parámetros de entrada y salida con una query con Having e Inner join (1 punto)
- 7) Realizar una vista con una subconsulta (1 punto)
- 8) Realizar dos consultas con (1 punto):
  - 1) DDL y DML

*ordenes* (id\_orden, fecha, id\_cliente, nom\_cliente, estado, num\_art, nom\_art, cant, precio)

### Ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Estado	Num_art	nom_art	cant	Precio
2301	23/02/11	101	Martin	Caracas	3786	Red	3	35,00
2301	23/02/11	101	Martin	Caracas	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Caracas	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Coro	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Maracay	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Maracay	3141	Funda	2	10,00

## PARTE PRÁCTICA – BASE DE DATOS

### 1) Identificar la PRIMERA FORMA NORMAL

	A	B	C	D	E	F
1						
2		<b>ORDEN</b>				
3		<u>ID_ORDEN</u>	<u>FECHA</u>	<u>ID_CLIENTE</u>	<u>Nom_Cliente</u>	<u>Estado</u>
4		2301	23/02/2011	101	Martin	Caracas
5		2301	23/02/2011	101	Martin	Caracas
6		2301	23/02/2011	101	Martin	Caracas
7		2302	25/2/2011	107	Hernan	Coro
8		2303	27/2/2011	110	Pedro	Maracay
9		2303	27/2/2011	110	Pedro	Maracay
10						
11						
12		<b>DETALLE_ORDEN</b>				
13		<u>ID_ORDEN*</u>	<u>Num_art</u>	<u>nom_art</u>	<u>cant</u>	<u>Precio</u>
14		2301	3786	Red	3	35,00
15		2301	4011	Raqueta	6	65,00
16		2301	9132	Paq-3	8	4,75
17		2302	5794	Paq-6	4	5,00
18		2303	4011	Raqueta	2	65,00
19		2303	3141	Funda	2	10,00
20						
21						
22		*ID_ORDEN es PK y FK				
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
48						
49						
50						
51						
52						
53						
54						
55						
56						
57						
58						
59						
60						
61						
62						
63						
64						
65						
66						
67						
68						
69						
70						
71						
72						
73						
74						
75						
76						
77						
78						
79						
80						
81						
82						
83						
84						
85						
86						
87						
88						
89						
90						
91						
92						
93						
94						
95						
96						
97						
98						
99						
100						

< >

TABLA 0

1FN

2FN

3FN

+

2) Identificar la SEGUNDA FORMA NORMAL

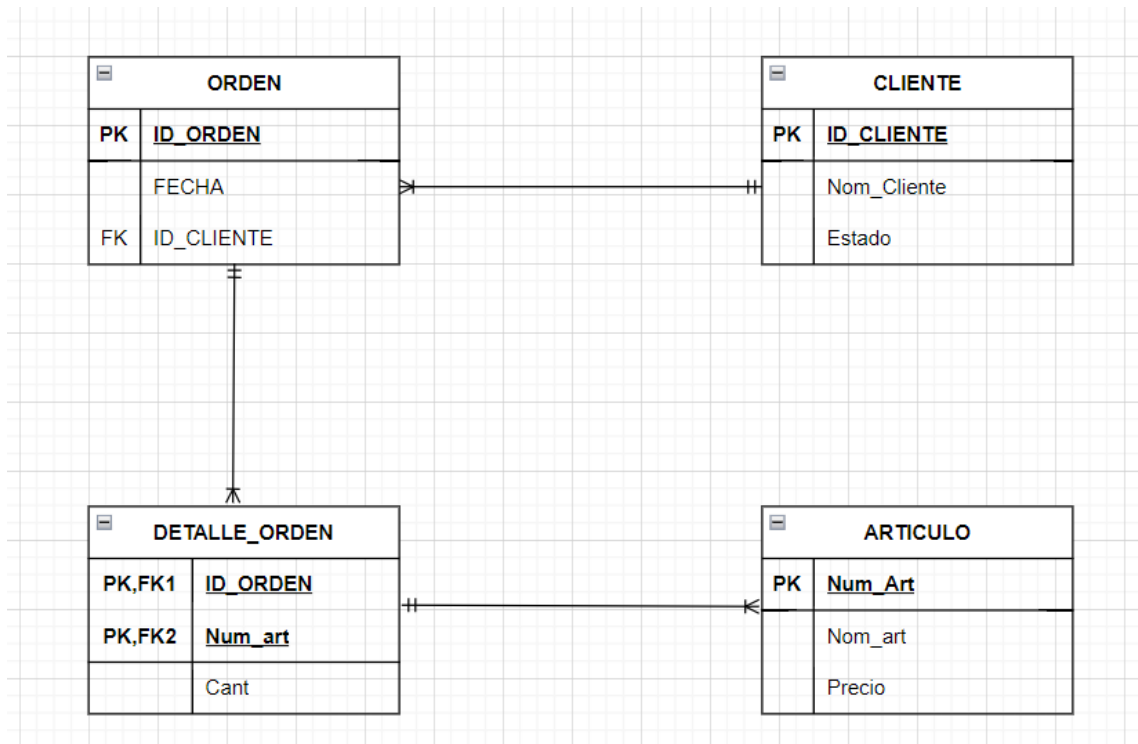
	A	B	C	D	E	F	G	H
1								
2								
3			ORDEN					
4			ID_ORDEN	FECHA	ID_CLIENTE	Nom_Cliente	Estado	
5			2301	23/02/2011	101	Martin	Caracas	
6			2301	23/02/2011	101	Martin	Caracas	
7			2301	23/02/2011	101	Martin	Caracas	
8			2302	25/2/2011	107	Hernan	Coro	
9			2303	27/2/2011	110	Pedro	Maracay	
10			2303	27/2/2011	110	Pedro	Maracay	
11								
12			DETALLE_ORDEN			ARTICULO		
13			ID_ORDEN	Num_art	cant	Num_art	nom_art	Precio
14			2301	3786	3	3786	Red	35,00
15			2301	4011	6	4011	Raqueta	65,00
16			2301	9132	8	9132	Paq-3	4,75
17			2302	5794	4	5794	Paq-6	5,00
18			2303	4011	2	4011	Raqueta	65,00
19			2303	3141	2	3141	Funda	10,00
20								
21								
22								
23								
24								
25								
26								
27								
	<	>	TABLA 0	1FN	2FN	3FN	+	

3) Identificar la SEGUNDA FORMA NORMAL



	A	B	C	D	E	F	G	H
1								
2		ORDEN				CLIENTE		
3		ID_ORDEN	FECHA	ID_CLIENTE*		ID_CLIENTE	Nom_Cliente	Estado
4		2301	23/02/2011	101		101	Martin	Caracas
5		2302	25/2/2011	107		107	Hernan	Coro
6		2303	27/2/2011	110		110	Pedro	Maracay
7								
8								
9								
10		DETALLE_ORDEN				ARTICULO		
11		ID_ORDEN	Num_art	cant		Num_art	nom_art	Precio
12		2301	3786	3		3786	Red	35,00
13		2301	4011	6		4011	Raqueta	65,00
14		2301	9132	8		9132	Paq-3	4,75
15		2302	5794	4		5794	Paq-6	5,00
16		2303	4011	2		3141	Funda	10,00
17		2303	3141	2				
18		*ID_CLIENTE es FK. (Si fuese compuesta permitira que pertenezca a varios)						
19								
20								
21								
22								
23								
24								
25								
26								
27								
	<	>	TABLA 0	1FN	2FN	3FN	+	

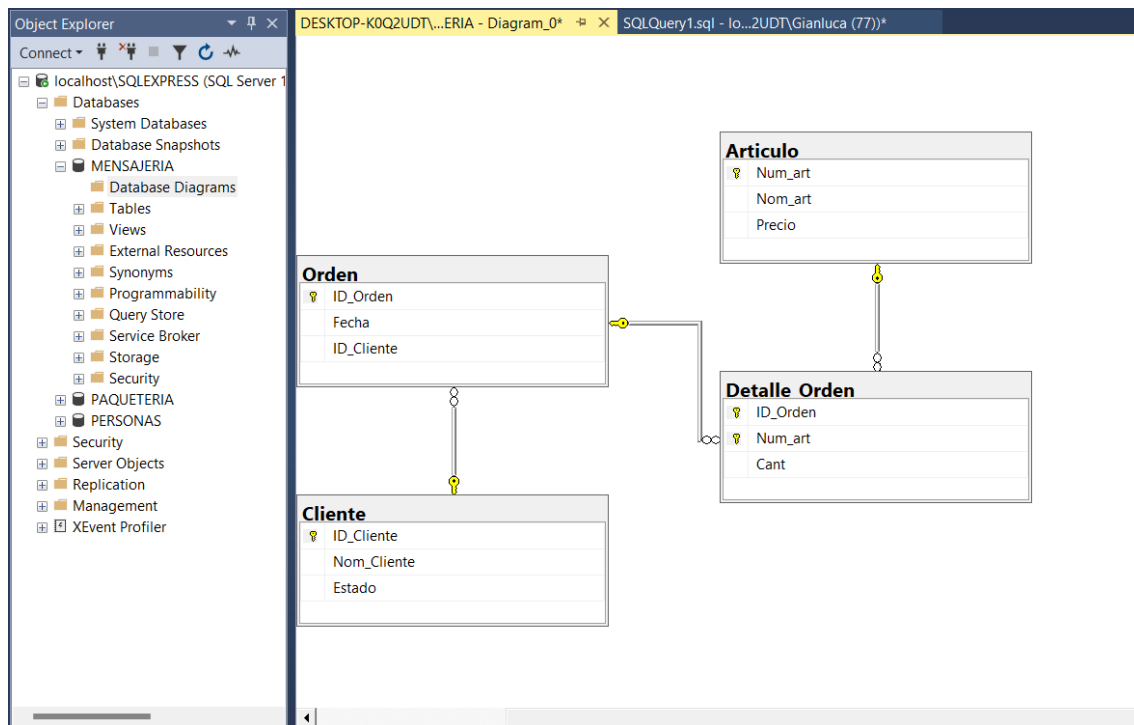
#### 4) Identificar el DER Lógico del esquema final.



#### CREACIÓN DE TABLAS EN SQL MANAGEMENT STUDIO

```
SQLQuery1.sql - localhost\SQLEXPRESS:master (DESKTOP-KOQZUDT\Gianluca (77)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query Execute
master
Object Explorer
Connect
localhost\SQLEXPRESS (SQL Server 10.0.5512.1)
Databases
Security
Server Objects
Replication
Management
XEvent Profiler
SQLQuery1.sql - localhost\SQLEXPRESS:master (DESKTOP-KOQZUDT\Gianluca (77))
--CREATE DATABASE MENSAJERIA
GO
USE MENSAJERIA
GO
-- Crear tabla Cliente
CREATE TABLE Cliente (
    ID_Cliente INT PRIMARY KEY,
    Nom_Cliente VARCHAR(50),
    Estado VARCHAR(50)
);
-- Crear tabla Articulo
CREATE TABLE Articulo (
    Num_art INT PRIMARY KEY,
    Nom_art VARCHAR(50),
    Precio DECIMAL(10, 2)
);
-- Crear tabla Orden
CREATE TABLE Orden (
    ID_Orden INT PRIMARY KEY,
    Fecha DATETIME,
    ID_Cliente INT,
    FOREIGN KEY (ID_Cliente) REFERENCES Cliente(ID_Cliente)
);
-- Crear tabla Detalle_Orden
CREATE TABLE Detalle_Orden (
    ID_Orden INT,
    Num_art INT,
    Cant INT,
    PRIMARY KEY (ID_Orden, Num_art),
    FOREIGN KEY (ID_Orden) REFERENCES Orden(ID_Orden),
    FOREIGN KEY (Num_art) REFERENCES Articulo(Num_art)
);
```

## RELACIONES ENTRE TABLAS



## INSERCIÓN DE DATOS TABLA [ARTICULO]

```
SQLQuery2.sql - Io...2UDT\Gianluca (63))* X
-- Inserción Datos Articulo

--INSERT INTO Articulo(Num_art, Nom_art, Precio)
--VALUES (3786, 'Red', 35.00)

INSERT INTO Articulo(Num_art, Nom_art, Precio)
VALUES (4011, 'Raqueta', 65.00)

INSERT INTO Articulo(Num_art, Nom_art, Precio)
VALUES (9132, 'Paq-3', 04.75)

INSERT INTO Articulo(Num_art, Nom_art, Precio)
VALUES (5794, 'Paq-6', 05.00)

INSERT INTO Articulo(Num_art, Nom_art, Precio)
VALUES (3141, 'Funda', 10.00)
```

SQLQuery5.sql - lo...2UDT\Gianluca (70) X

```
SELECT TOP (1000) [Num_art]
, [Nom_art]
, [Precio]
FROM [MENSAJERIA].[dbo].[Articulo]
```

100 %

Results Messages

	Num_art	Nom_art	Precio
1	3141	Funda	10.00
2	3786	Red	35.00
3	4011	Raqueta	65.00
4	5794	Paq-6	5.00
5	9132	Paq-3	4.75

**INSERCIÓN DE DATOS TABLA [CLIENTE]**

```
-- Inserción Datos Cliente
```

```
INSERT INTO Cliente(ID_Cliente, Nom_Cliente, Estado)  
VALUES (101, 'Martin', 'Caracas')
```

```
INSERT INTO Cliente(ID_Cliente, Nom_Cliente, Estado)  
VALUES (107, 'Hernan', 'Coro')
```

```
INSERT INTO Cliente(ID_Cliente, Nom_Cliente, Estado)  
VALUES (110, 'Martin', 'Caracas')
```

```
SELECT TOP (1000) [ID_Cliente]  
    ,[Nom_Cliente]  
    ,[Estado]  
FROM [MENSAJERIA].[dbo].[Cliente]
```

100 %

Results Messages

	ID_Cliente	Nom_Cliente	Estado
1	101	Martin	Caracas
2	107	Hernan	Coro
3	110	Martin	Caracas

### INSERCIÓN DE DATOS TABLA [ORDEN]

SQLQuery7.sql - lo...2UDT\Gianluca (74))

SQLQuery2.sql - lo...2UDT\Gianluca (63))\*

-- Inserción Datos Orden

```
INSERT INTO Orden
VALUES(2301, '2011/02/23', 101)

INSERT INTO Orden
VALUES(2302, '2011/02/25', 107)

INSERT INTO Orden(ID_Orden, Fecha, ID_Cliente)
VALUES(2303, '2011/02/27', 110)
```

SQLQuery7.sql - lo...2UDT\Gianluca (74))

SQLQuery2.sql - lo...2UDT\Gianluca (63))\*

```
SELECT TOP (1000) [ID_Orden]
, [Fecha]
, [ID_Cliente]
FROM [MENSAJERIA].[dbo].[Orden]
```

100 %

Results Messages

	ID_Orden	Fecha	ID_Cliente
1	2301	2011-02-23	101
2	2302	2011-02-25	107
3	2303	2011-02-27	110

**INSERCIÓN DE DATOS TABLA [DETALLE\_ORDEN]**

SQLQuery8.sql - lo...2UDT\Gianluca (63))\* ✕

-- Inserción Datos Detalle\_Orden

INSERT INTO Detalle\_Orden  
VALUES (2301,3786,3)

INSERT INTO Detalle\_Orden  
VALUES (2301,4011,6)

INSERT INTO Detalle\_Orden  
VALUES (2301,9132,8)

INSERT INTO Detalle\_Orden  
VALUES (2302,5794,4)

INSERT INTO Detalle\_Orden  
VALUES (2303,4011,2)

INSERT INTO Detalle\_Orden  
VALUES (2303,3141,2)



SQLQuery9.sql - lo...2UDT\Gianluca (81))

```

SELECT TOP (1000) [ID_Orden]
      , [Num_art]
      , [Cant]
FROM [MENSAJERIA].[dbo].[Detalle_Orden]

```

100 %

Results Messages

	ID_Orden	Num_art	Cant
1	2301	3786	3
2	2301	4011	6
3	2301	9132	8
4	2302	5794	4
5	2303	3141	2
6	2303	4011	2

- 5) REALIZAR UN PROCEDIMIENTO ALMACENADO CON PARAMETROS DE ENTRADA CON UNA QUERY CON GROUP BY

SQLQuery10.sql - I...2UDT\Gianluca (61))\*

```
SELECT Precio  
FROM Articulo  
WHERE Precio > 10  
GROUP BY Precio
```

100 %

Results Messages

	Precio
1	35.00
2	65.00

```
CREATE PROCEDURE TOTAL_CLIENTE_X_PRODUCTO_PRECIO
    @Precio decimal(18,2)
AS
BEGIN
    SET NOCOUNT ON;

    SELECT C.Nom_Cliente, ord.Fecha, SUM(det_ord.Cant
FROM Orden ord
JOIN Detalle_Orden det_ord
    ON ord.Id_Orden = det_ord.Id_orden
JOIN Cliente C
    ON C.Id_Cliente = ord.Id_Cliente
JOIN Articulo art
    ON art.Num_Art = det_ord.Num_Art
WHERE art.Precio > @Precio
GROUP BY C.Nom_Cliente, ord.Fecha
END
GO
```

100 %

## Messages

Msg 2714, Level 16, State 3, Procedure TOTAL\_CLIENTE\_X\_PRODUCTO\_PRECIO,  
There is already an object named 'TOTAL\_CLIENTE\_X\_PRODUCTO\_PRECIO' in the

Completion time: 2023-07-04T20:59:43.3290838+02:00

- 6) REALIZAR UN PROCEDIMIENTO ALMACENADO CON PARÁMETROS DE ENTRADA Y SALIDA CON UNA QUERY CON HAVING E INNER JOIN

SQLQuery11.sql - I...2UDT\Gianluca (61))\* ✕

```
SELECT art.Num_art, art.Nom_art, SUM(OD.Cant) as Total_Cantidad
FROM Articulo art
INNER JOIN Detalle_Orden OD ON art.Num_art = OD.Num_art
GROUP BY art.Num_art, art.Nom_art
HAVING SUM(OD.Cant) > 5
```

100 %

Results Messages

	Num_art	Nom_art	Total_Cantidad
1	4011	Raqueta	8
2	9132	Paq-3	8

## 7) REALIZAR UNA VISTA CON UNA SUBCONSULTA

The screenshot shows a SQL query window titled 'SQLQuery13.sql - I...2UDT\Gianluca (52))\*'. The query is as follows:

```
-- CREATE VIEW VistaOrdenes AS
SELECT art.Num_art, art.Nom_art,
       (SELECT SUM(Cant) FROM Detalle_Orden WHERE Num_art = art.Num_art) AS TotalOrdenes
FROM Articulo art;

SELECT * FROM VistaOrdenes
```

Below the query editor, the 'Results' tab is active, displaying a table with 5 rows and 4 columns: Num\_art, Nom\_art, and TotalOrdenes. The data is as follows:

	Num_art	Nom_art	TotalOrdenes
1	3141	Funda	2
2	3786	Red	3
3	4011	Raqueta	8
4	5794	Paq-6	4
5	9132	Paq-3	8

## 8) REALIZAR DOS CONSULTAS CON:

### a) DDL

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane shows the database structure for 'MENSAJERIA'. The 'Tables' folder is expanded, showing a list of tables: System Tables, FileTables, External Tables, Graph Tables, dbo.Articulo, dbo.Cliente, dbo.Detalle\_Orden, dbo.Empleado, and dbo.Orden. The 'dbo.Empleado' table is highlighted.

On the right, the 'SQLQuery14.sql - I...2UDT\Gianluca (52))\*' window shows the following DDL query:

```
-- Consulta con DDL

CREATE TABLE Empleado (
    ID_EMPLEADO INT PRIMARY KEY,
    Nombre VARCHAR(50),
    Edad INT
);
```

b) DML

SQLQuery14.sql - I...2UDT\Gianluca (52))\*

```
-- Consulta con DML

UPDATE Cliente
SET Nom_Cliente = 'Herman'
WHERE ID_Cliente = 107

SELECT *
FROM Cliente
```

100 %

Results Messages

	ID_Cliente	Nom_Cliente	Estado
1	101	Martin	Caracas
2	107	Herman	Coro
3	110	Martin	Caracas