



Universidad Don Bosco, El Salvador

## **Datawarehouse y Minería de Datos**

### **Desafío 1**

**Docente: Karens Medrano**

**Estudiante: RENE ALEXIS BARAHONA BONILLA BB241958**

**Fecha de entrega:**

**13/09/2025**

**Enlace repositorio GIT:**

<https://github.com/TheShadowManIncident/Datawarehouse-y-Miner-a-de-Datos-Desafio-1.git>

# Proceso para la creación del diseño estrella incluyendo sus etapas ETL, limpieza, validación, y consolidación de información

## Preparación de scripts

1)Base Chinook.sql	9/12/2025 9:24 PM	Microsoft SQL Ser...	3,835 KB
2)Proceso ETL Optimizado.sql	9/12/2025 10:00 PM	Microsoft SQL Ser...	21 KB
3)Consultas De Prueba.sql	9/12/2025 10:20 PM	Microsoft SQL Ser...	5 KB

1. Procederemos a cargar script de la base de datos Chinook en MSQL Server sin errores

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'Databases' folder expanded, showing the 'Chinook' database. The right pane shows the 'Query Editor' with a script titled '1)Base Chinook.sql'. The script contains SQL code to create the Chinook database and its tables. The script is as follows:

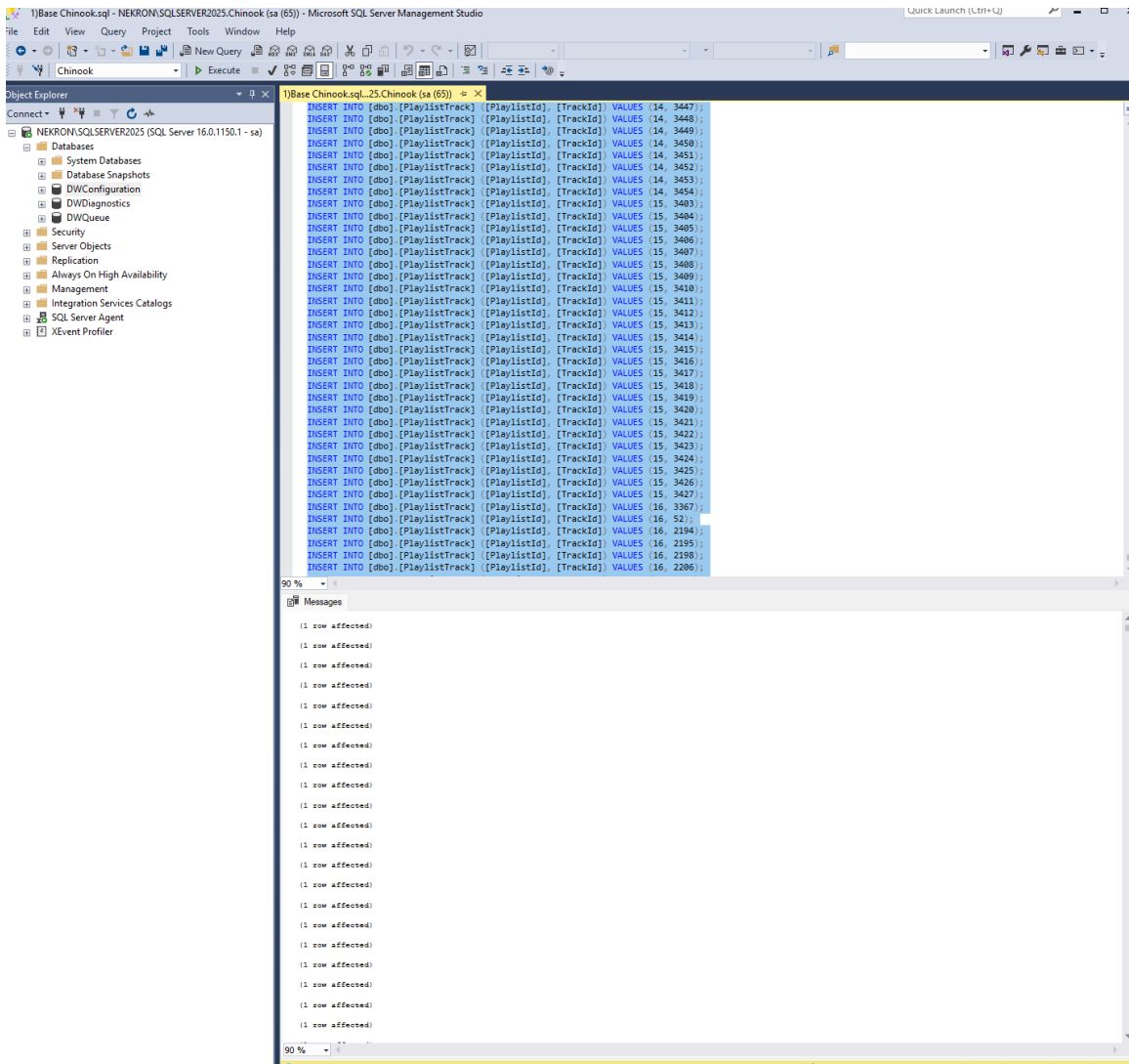
```
/*
Chinook Database - Version 1.4
Script: Chinook_SqlServer.sql
Description: Creates and populates the Chinook database.
DB Server: SqlServer
Author: Luis Rocha
License: https://www.codeplex.com/ChinookDatabase/license
*/

-- Drop database if it exists
IF EXISTS (SELECT name FROM master.dbo.sysdatabases WHERE name = N'Chinook')
BEGIN
    ALTER DATABASE [Chinook] SET OFFLINE WITH ROLLBACK IMMEDIATE;
    ALTER DATABASE [Chinook] SET ONLINE;
    DROP DATABASE [Chinook];
END
GO

-- Create database
CREATE DATABASE [Chinook];
GO

USE [Chinook];
GO

-- Create Tables
CREATE TABLE [dbo].[Album]
(
    [AlbumId] INT NOT NULL,
    [Title] NVARCHAR(160) NOT NULL,
    [ArtistId] INT NOT NULL,
    CONSTRAINT [PK_Album] PRIMARY KEY CLUSTERED ([AlbumId])
);
GO
CREATE TABLE [dbo].[Artist]
(
    [ArtistId] INT NOT NULL,
    [Name] NVARCHAR(120),
    CONSTRAINT [PK_Artist] PRIMARY KEY CLUSTERED ([ArtistId])
);
GO
CREATE TABLE [dbo].[Customer]
(
    [CustomerId] INT NOT NULL,
    [FirstName] NVARCHAR(40) NOT NULL,
    [LastName] NVARCHAR(20) NOT NULL,
    [Company] NVARCHAR(80),
    [Address] NVARCHAR(70),
    [City] NVARCHAR(40),
    [State] NVARCHAR(40),
    [Country] NVARCHAR(40),
    [PostalCode] NVARCHAR(10),
    [Phone] NVARCHAR(24),
    [Fax] NVARCHAR(24),
    [Email] NVARCHAR(60) NOT NULL,
    [SupportRepId] INT,
    CONSTRAINT [PK_Customer] PRIMARY KEY CLUSTERED ([CustomerId])
);
GO
CREATE TABLE [dbo].[Employee]
(
    [EmployeeId] INT NOT NULL,
    [LastName] NVARCHAR(20) NOT NULL,
    [FirstName] NVARCHAR(20) NOT NULL,
    [Title] NVARCHAR(30),
    [ReportsTo] INT,
    [BirthDate] DATETIME,
    [HireDate] DATETIME,
    [Address] NVARCHAR(70),
    [City] NVARCHAR(40),
    [State] NVARCHAR(40),
    [Country] NVARCHAR(40),
    [PostalCode] NVARCHAR(10),
    [Phone] NVARCHAR(24),
    [Fax] NVARCHAR(24),
    [Email] NVARCHAR(60) NOT NULL,
    [SupportRepId] INT,
    CONSTRAINT [PK_Employee] PRIMARY KEY CLUSTERED ([EmployeeId])
);
GO
```



- NEKRON\SQLSERVER2025 (SQL Server 16.0.1)
- Databases
  - System Databases
  - Database Snapshots
  - Chinook
    - Database Diagrams
    - Tables
      - System Tables
      - FileTables
      - External Tables
      - Graph Tables
      - dbo.Album
      - dbo.Artist
      - dbo.Customer
      - dbo.Employee
      - dbo.Genre
      - dbo.Invoice
      - dbo.InvoiceLine
      - dbo.MediaType
      - dbo.Playlist
      - dbo.PlaylistTrack
      - dbo.Track
      - Dropped Ledger Tables

Como demostración adicional algunas tablas que se crearon son las siguientes:

Tabla Track:

SQLQuery27.sql - N...S.Chinook (sa (84))SQLQuery26.sql - N...S.Chinook (sa (83))SQLQuery25.sql - N...S.Chinook (sa (77))SQLQuery24.sql - N...S.Chinook (sa (82))

```
SELECT TOP (1000) [TrackId]
, [Name]
, [AlbumId]
, [MediaTypeId]
, [GenreId]
, [Composer]
, [Milliseconds]
, [Bytes]
, [UnitPrice]
FROM [Chinook].[dbo].[Track]
```

TrackId	Name	AlbumId	MediaTypeId	GenreId	Composer	Milliseconds	Bytes	UnitPrice
1	For Those About To Rock (We Salute You)	1	1	1	Angus Young, Malcolm Young, Brian Johnson	343719	11170334	0.99
2	Balls to the Wall	2	2	1	NULL	342562	5510424	0.99
3	Fast As a Shark	3	2	1	F. Bates, S. Kaufman, U. Dirkschneider & W. Hoff...	230619	3990994	0.99
4	Restless and Wild	3	2	1	F. Bates, R.A. Smith-Diesel, S. Kaufman, U. Dirks...	252051	4331779	0.99
5	Princess of the Dawn	3	2	1	Deafy & R.A. Smith-Diesel	375418	6290521	0.99
6	Put The Finger On You	1	1	1	Angus Young, Malcolm Young, Brian Johnson	205662	6713451	0.99
7	Let's Get It Up	1	1	1	Angus Young, Malcolm Young, Brian Johnson	233526	7636561	0.99
8	Inject The Venom	1	1	1	Angus Young, Malcolm Young, Brian Johnson	210834	6852860	0.99
9	Snowballed	1	1	1	Angus Young, Malcolm Young, Brian Johnson	203102	6599424	0.99
10	Evil Walks	1	1	1	Angus Young, Malcolm Young, Brian Johnson	263497	8611245	0.99
11	C.O.D.	1	1	1	Angus Young, Malcolm Young, Brian Johnson	198336	6566314	0.99
12	Breaking The Rules	1	1	1	Angus Young, Malcolm Young, Brian Johnson	263288	8596840	0.99
13	Night Of The Long Knives	1	1	1	Angus Young, Malcolm Young, Brian Johnson	205688	6706347	0.99
14	Spellbound	1	1	1	Angus Young, Malcolm Young, Brian Johnson	270863	8817038	0.99
15	Go Down	4	1	1	AC/DC	331180	10847611	0.99
16	Dog Eat Dog	4	1	1	AC/DC	215196	7032162	0.99
17	Let There Be Rock	4	1	1	AC/DC	366654	12021261	0.99
18	Bad Boy Boogie	4	1	1	AC/DC	267728	8776140	0.99
19	Problem Child	4	1	1	AC/DC	325041	10617116	0.99
20	Overdose	4	1	1	AC/DC	369319	12066294	0.99
21	Hell Ain't A Bad Place To Be	4	1	1	AC/DC	254380	8331286	0.99
22	Whole Lotta Rosie	4	1	1	AC/DC	323761	10547154	0.99
23	Walk On Water	5	1	1	Steven Tyler, Joe Perry, Jack Blades, Tommy Shaw	295680	9719579	0.99
24	Love In An Elevator	5	1	1	Steven Tyler, Joe Perry	321828	10552051	0.99
25	Rag Doll	5	1	1	Steven Tyler, Joe Perry, Jim Vallance, Holly Knight	264698	8675345	0.99
26	What It Takes	5	1	1	Steven Tyler, Joe Perry, Desmond Child	310622	10144730	0.99
27	Dude (Looks Like A Lady)	5	1	1	Steven Tyler, Joe Perry, Desmond Child	264855	8679940	0.99
28	Janie's Got A Gun	5	1	1	Steven Tyler, Tom Hamilton	330736	10869391	0.99
29	...And	6	1	1	Steven Tyler, Joe Perry, Desmond Child	390363	11066686	0.99

90 %

ResultsMessages

Query executed successfully. NEKRON/SQLSERVER2023 (16.0 ... sa (84)) Chinook 00:00:00 1,000 rows

## Tabla Invoice

<pre> SELECT TOP (1000) [InvoiceId] , [CustomerId] , [InvoiceDate] , [BillingAddress] , [BillingCity] , [BillingState] , [BillingCountry] , [BillingPostalCode] , [Total] FROM [Chinook].[dbo].[Invoice] </pre>									
	InvoiceId	CustomerId	InvoiceDate	BillingAddress	BillingCity	BillingState	BillingCountry	BillingPostalCode	Total
1	1	2	2009-01-01 00:00:00.000	Theodor-Heuss-Straße 34	Stuttgart	NULL	Germany	70174	1.98
2	2	4	2009-01-02 00:00:00.000	Ullevålsveien 14	Oslo	NULL	Norway	0171	3.96
3	3	8	2009-01-03 00:00:00.000	Grêtrystraat 63	Brussels	NULL	Belgium	1000	5.94
4	4	14	2009-01-06 00:00:00.000	8210 111 ST NW	Edmonton	AB	Canada	T6G 2C7	8.91
5	5	23	2009-01-11 00:00:00.000	69 Salem Street	Boston	MA	USA	2113	13.86
6	6	37	2009-01-19 00:00:00.000	Berger Straße 10	Frankfurt	NULL	Germany	60316	0.99
7	7	38	2009-02-01 00:00:00.000	Barbarossastraße 19	Berlin	NULL	Germany	10779	1.98
8	8	40	2009-02-01 00:00:00.000	8, Rue Hanovre	Paris	NULL	France	75002	1.98
9	9	42	2009-02-02 00:00:00.000	9, Place Louis Barthou	Bordeaux	NULL	France	33000	3.96
10	10	46	2009-02-03 00:00:00.000	3 Chatham Street	Dublin	Dublin	Ireland	NULL	5.94
11	11	52	2009-02-06 00:00:00.000	202 Hoxton Street	London	NULL	United Kingdom	N1 5LH	8.91
12	12	2	2009-02-11 00:00:00.000	Theodor-Heuss-Straße 34	Stuttgart	NULL	Germany	70174	13.86
13	13	16	2009-02-19 00:00:00.000	1600 Amphitheatre Parkway	Mountain View	CA	USA	94043-1351	0.99
14	14	17	2009-03-04 00:00:00.000	1 Microsoft Way	Redmond	WA	USA	98052-8300	1.98
15	15	19	2009-03-04 00:00:00.000	1 Infinite Loop	Cupertino	CA	USA	95014	1.98
16	16	21	2009-03-05 00:00:00.000	801 W 4th Street	Reno	NV	USA	89503	3.96
17	17	25	2009-03-06 00:00:00.000	319 N. Frances Street	Madison	WI	USA	53703	5.94
18	18	31	2009-03-09 00:00:00.000	194A Chain Lake Drive	Halifax	NS	Canada	B3S 1C5	8.91
19	19	40	2009-03-14 00:00:00.000	8, Rue Hanovre	Paris	NULL	France	75002	13.86
20	20	54	2009-03-22 00:00:00.000	110 Raeburn Pl	Edinburgh	NULL	United Kingdom	EH4 1HH	0.99
21	21	55	2009-04-04 00:00:00.000	421 Bourke Street	Sydney	NSW	Australia	2010	1.98
22	22	57	2009-04-04 00:00:00.000	Calle Lira, 198	Santiago	NULL	Chile	NULL	1.98
23	23	59	2009-04-05 00:00:00.000	3,Raj Bhavan Road	Bangalore	NULL	India	560001	3.96
24	24	4	2009-04-06 00:00:00.000	Ullevålsveien 14	Oslo	NULL	Norway	0171	5.94
25	25	10	2009-04-09 00:00:00.000	Rua Dr. Falcão Filho, 155	São Paulo	SP	Brazil	01007-010	8.91
26	26	19	2009-04-14 00:00:00.000	1 Infinite Loop	Cupertino	CA	USA	95014	13.86
27	27	33	2009-04-22 00:00:00.000	5112 48 Street	Yellowknife	NT	Canada	X1A 1N6	0.99
28	28	34	2009-05-05 00:00:00.000	Rua da Assunção 53	Lisbon	NULL	Portugal	NULL	1.98
29	29	36	2009-05-05 00:00:00.000	Teuscherstrasse 2	Berlin	NULL	Germany	10709	1.98

Querv executed successfully. NEKRON\SOLSERVER2025 (16.0 ... | sa (82) | Chinook | 00:00:00 | 412 rows

## Tabla Costumer

SQLQuery29.sql - N...5.Chinook (sa (91)) SQLQuery28.sql - N...5.Chinook (sa (87)) SQLQuery27.sql - N...5.Chinook (sa (84)) SQLQuery26.sql - N...5.Chinook (sa (83))

```

SELECT TOP (1000) [CustomerId]
, [FirstName]
, [LastName]
, [Company]
, [Address]
, [City]
, [State]
, [Country]
, [PostalCode]
, [Phone]
, [Fax]
, [Email]
, [SupportRepId]
FROM [Chinook].[dbo].[Customer]

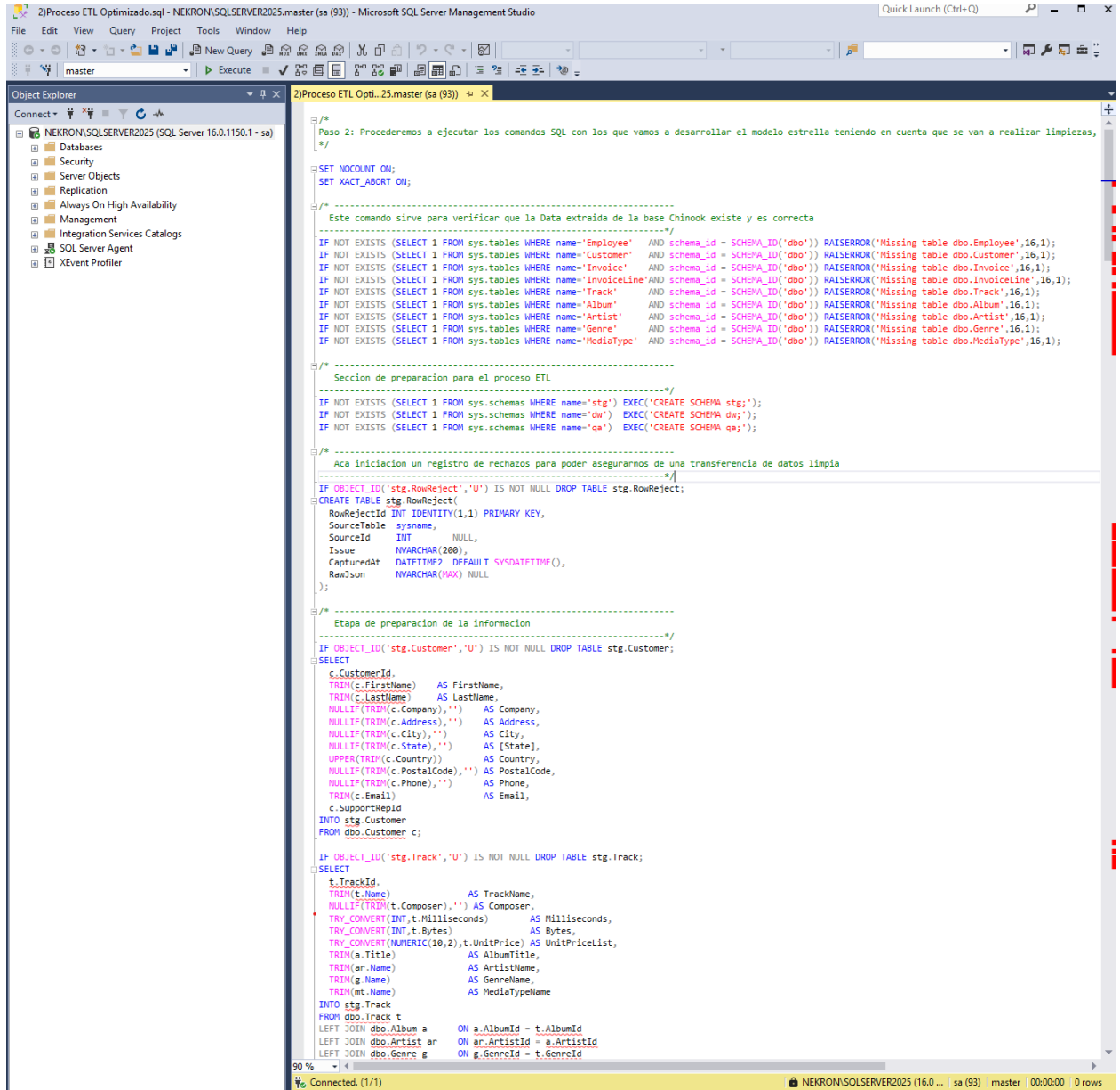
```

90 %

	CustomerId	FirstName	LastName	Company	Address	City	State	Country	PostalCode	Phone
1	1	Luis	Gonçalves	Embraer - Empresa Brasileira de Aeronáutica S.A.	Av. Brigadeiro Faria Lima, 2170	São José dos Campos	SP	Brazil	12227-000	+55 (12)
2	2	Leonie	Köhler	NULL	Theodor-Heuss-Straße 34	Stuttgart	NULL	Germany	70174	+49 071
3	3	François	Tremblay	NULL	1498 rue Bélanger	Montréal	QC	Canada	H2G 1A7	+1 (514)
4	4	Bjørn	Hansen	NULL	Ullevålsveien 14	Oslo	NULL	Norway	0171	+47 22 4
5	5	František	Wichterlová	JetBrains s.r.o.	Klanova 9/506	Prague	NULL	Czech Republic	14700	+420 2 4
6	6	Helena	Holý	NULL	Rileká 3174/6	Prague	NULL	Czech Republic	14300	+420 2 4
7	7	Astrid	Gruber	NULL	Rotentumstraße 4, 1010 Innere Stadt	Vienne	NULL	Austria	1010	+43 01 5
8	8	Daan	Peeters	NULL	Grétrystraat 63	Brussels	NULL	Belgium	1000	+32 02 2
9	9	Kara	Nielsen	NULL	Sønder Boulevard 51	Copenhagen	NULL	Denmark	1720	+453 33
10	10	Eduardo	Martins	Woodstock Discos	Rua Dr. Falcão Filho, 155	São Paulo	SP	Brazil	01007-010	+55 (11)
11	11	Alexandre	Rocha	Banco do Brasil S.A.	Av. Paulista, 2022	São Paulo	SP	Brazil	01310-200	+55 (11)
12	12	Roberto	Almeida	Riotur	Praça Pio X, 119	Rio de Janeiro	RJ	Brazil	20040-020	+55 (21)
13	13	Fernanda	Ramos	NULL	Qe 7 Bloco G	Brasília	DF	Brazil	71020-677	+55 (61)
14	14	Mark	Philips	Telus	8210 111 ST NW	Edmonton	AB	Canada	T6G 2C7	+1 (780)
15	15	Jennifer	Peterson	Rogers Canada	700 W Pender Street	Vancouver	BC	Canada	V6C 1G8	+1 (604)
16	16	Frank	Harris	Google Inc.	1600 Amphitheatre Parkway	Mountain View	CA	USA	94043-1351	+1 (650)
17	17	Jack	Smith	Microsoft Corporation	1 Microsoft Way	Redmond	WA	USA	98052-8300	+1 (425)
18	18	Michelle	Brooks	NULL	627 Broadway	New York	NY	USA	10012-2612	+1 (212)
19	19	Tim	Goyer	Apple Inc.	1 Infinite Loop	Cupertino	CA	USA	95014	+1 (408)
20	20	Dan	Miller	NULL	541 Del Medio Avenue	Mountain View	CA	USA	94040-111	+1 (650)
21	21	Kathy	Chase	NULL	801 W 4th Street	Reno	NV	USA	89503	+1 (775)
22	22	Heather	Leacock	NULL	120 S Orange Ave	Orlando	FL	USA	32801	+1 (407)
23	23	John	Gordon	NULL	69 Salem Street	Boston	MA	USA	2113	+1 (617)
24	24	Frank	Ralston	NULL	162 E Superior Street	Chicago	IL	USA	60611	+1 (312)
25	25	Victor	Stevens	NULL	319 N. Frances Street	Madison	WI	USA	53703	+1 (608)
26	26	Richard	Cunningham	NULL	2211 W Berry Street	Fort Worth	TX	USA	76110	+1 (817)
27	27	Patrick	Gray	NULL	1033 N Park Ave	Tucson	AZ	USA	85719	+1 (520)
28	28	Jillia	Ramett	NULL	307 S 7th F	Salt Lake City	UT	USA	84102	+1 (801)

Query executed successfully. NEKRON\SQLSERVER2025 (16.0 ... sa (91) Chinook: 00:00:00 59 rows

Paso 2: Procederemos a ejecutar los comandos SQL con los que vamos a desarrollar el modelo estrella teniendo en cuenta que se van a realizar limpiezas, validaciones y conversiones



The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'NEKRON\SQLSERVER2025 (SQL Server 16.0.1150.1 - sa)' server selected. The right pane shows a SQL script titled '2)Proceso ETL Opti...25.master (sa (93))'. The script is written in T-SQL and includes comments in Spanish. It starts with a comment: 'Paso 2: Procederemos a ejecutar los comandos SQL con los que vamos a desarrollar el modelo estrella teniendo en cuenta que se van a realizar limpiezas, validaciones y conversiones'. The script then sets 'SET NOCOUNT ON;' and 'SET XACT\_ABORT ON;'. It includes a section for verifying the existence of tables in the 'dbo' schema, with error messages for missing tables like 'Employee', 'Customer', 'Invoice', 'InvoiceLine', 'Track', 'Album', 'Artist', 'Genre', and 'MediaType'. This is followed by a section for preparing the ETL process, which creates schemas 'stg', 'dw', and 'qa' if they don't exist. It then creates a table 'stg.RowReject' with columns for 'RowRejectId', 'SourceTable', 'SourceId', 'Issue', 'CapturedAt', and 'RawJson'. The script continues with a section for preparing the information, starting with a 'SELECT' statement to populate 'stg.Customer' from 'dbo.Customer'. The final part of the script shows a 'SELECT' statement to populate 'stg.Track' from 'dbo.Track', 'dbo.Album', 'dbo.Artist', and 'dbo.Genre'.

```
/*
Paso 2: Procederemos a ejecutar los comandos SQL con los que vamos a desarrollar el modelo estrella teniendo en cuenta que se van a realizar limpiezas,
validaciones y conversiones
*/

SET NOCOUNT ON;
SET XACT_ABORT ON;

/*
-----
Este comando sirve para verificar que la Data extraida de la base Chinook existe y es correcta
-----
*/
IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE name='Employee' AND schema_id = SCHEMA_ID('dbo')) RAISERROR('Missing table dbo.Employee',16,1);
IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE name='Customer' AND schema_id = SCHEMA_ID('dbo')) RAISERROR('Missing table dbo.Customer',16,1);
IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE name='Invoice' AND schema_id = SCHEMA_ID('dbo')) RAISERROR('Missing table dbo.Invoice',16,1);
IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE name='InvoiceLine' AND schema_id = SCHEMA_ID('dbo')) RAISERROR('Missing table dbo.InvoiceLine',16,1);
IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE name='Track' AND schema_id = SCHEMA_ID('dbo')) RAISERROR('Missing table dbo.Track',16,1);
IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE name='Album' AND schema_id = SCHEMA_ID('dbo')) RAISERROR('Missing table dbo.Album',16,1);
IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE name='Artist' AND schema_id = SCHEMA_ID('dbo')) RAISERROR('Missing table dbo.Artist',16,1);
IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE name='Genre' AND schema_id = SCHEMA_ID('dbo')) RAISERROR('Missing table dbo.Genre',16,1);
IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE name='MediaType' AND schema_id = SCHEMA_ID('dbo')) RAISERROR('Missing table dbo.MediaType',16,1);

/*
-----
Seccion de preparacion para el proceso ETL
-----
*/
IF NOT EXISTS (SELECT 1 FROM sys.schemas WHERE name='stg') EXEC('CREATE SCHEMA stg;');
IF NOT EXISTS (SELECT 1 FROM sys.schemas WHERE name='dw') EXEC('CREATE SCHEMA dw;');
IF NOT EXISTS (SELECT 1 FROM sys.schemas WHERE name='qa') EXEC('CREATE SCHEMA qa;');

/*
-----
Aca iniciacion un registro de rechazos para poder asegurarnos de una transferencia de datos limpia
-----
*/
IF OBJECT_ID('stg.RowReject','U') IS NOT NULL DROP TABLE stg.RowReject;
CREATE TABLE stg.RowReject(
    RowRejectId INT IDENTITY(1,1) PRIMARY KEY,
    SourceTable sysname,
    SourceId INT NULL,
    Issue NVARCHAR(200),
    CapturedAt DATETIME2 DEFAULT SYSDATETIME(),
    RawJson NVARCHAR(MAX) NULL
);

/*
-----
Etapa de preparacion de la informacion
-----
*/
IF OBJECT_ID('stg.Customer','U') IS NOT NULL DROP TABLE stg.Customer;
SELECT
    c.CustomerId,
    TRIM(c.FirstName) AS FirstName,
    TRIM(c.LastName) AS LastName,
    NULLIF(TRIM(c.Company),'') AS Company,
    NULLIF(TRIM(c.Address),'') AS Address,
    NULLIF(TRIM(c.City),'') AS City,
    NULLIF(TRIM(c.State),'') AS [State],
    UPPER(TRIM(c.Country)) AS Country,
    NULLIF(TRIM(c.PostalCode),'') AS PostalCode,
    NULLIF(TRIM(c.Phone),'') AS Phone,
    TRIM(c.Email) AS Email,
    c.SupportRepId
INTO stg.Customer
FROM dbo.Customer c;

IF OBJECT_ID('stg.Track','U') IS NOT NULL DROP TABLE stg.Track;
SELECT
    t.TrackId,
    TRIM(t.Name) AS TrackName,
    NULLIF(TRIM(t.Composer),'') AS Composer,
    TRY_CONVERT(INT,t.Milliseconds) AS Milliseconds,
    TRY_CONVERT(INT,t.Bytes) AS Bytes,
    TRY_CONVERT(NUMERIC(10,2),t.UnitPrice) AS UnitPriceList,
    TRIM(a.Title) AS AlbumTitle,
    TRIM(ar.Name) AS ArtistName,
    TRIM(g.Name) AS GenreName,
    TRIM(mt.Name) AS MediaTypeName
INTO stg.Track
FROM dbo.Track t
LEFT JOIN dbo.Album a ON a.AlbumId = t.AlbumId
LEFT JOIN dbo.Artist ar ON ar.ArtistId = t.ArtistId
LEFT JOIN dbo.Genre g ON g.GenreId = t.GenreId
```

```

LEFT JOIN dbo.Artista ar ON ar.ArtistaId = d.ArtistaId
LEFT JOIN dbo.Genre g ON g.GenreId = t.GenreId
LEFT JOIN dbo.MediaType mt ON mt.MediaTypeId = t.MediaTypeId;

IF OBJECT_ID('stg.Invoice','U') IS NOT NULL DROP TABLE stg.Invoice;
SELECT
    i.InvoiceId,
    i.CustomerId,
    CAST(i.InvoiceDate AS DATE) AS InvoiceDate,
    NULLIF(TRIM(i.BillingAddress),'') AS BillingAddress,
    NULLIF(TRIM(i.BillingCity),'') AS BillingCity,
    NULLIF(TRIM(i.BillingState),'') AS BillingState,
    UPPER(TRIM(i.BillingCountry)) AS BillingCountry,
    NULLIF(TRIM(i.BillingPostalCode),'') AS BillingPostalCode,
    TRY_CONVERT(NUMERIC(10,2),i.Total) AS InvoiceTotal
INTO stg.Invoice
FROM dbo.Invoice i;

IF OBJECT_ID('stg.InvoiceLine','U') IS NOT NULL DROP TABLE stg.InvoiceLine;
SELECT
    il.InvoiceLineId,
    il.InvoiceId,
    il.TrackId,
    TRY_CONVERT(NUMERIC(10,2),il.UnitPrice) AS UnitPrice,
    TRY_CONVERT(INT,il.Quantity) AS Quantity
INTO stg.InvoiceLine
FROM dbo.InvoiceLine il;

/*Corroboracion de calidad de informacion */
INSERT INTO stg.RowReject(SourceTable,SourceId,Issue,RawJson)
SELECT 'InvoiceLine', il.InvoiceLineId, 'Quantity<=0 or UnitPrice<0',
    CONCAT('{"Quantity":',il.Quantity,', "UnitPrice":',il.UnitPrice,'}')
FROM stg.InvoiceLine il
WHERE ISNULL(il.Quantity,0) <= 0 OR ISNULL(il.UnitPrice,0) < 0;

DELETE il
FROM stg.InvoiceLine il
WHERE ISNULL(il.Quantity,0) <= 0 OR ISNULL(il.UnitPrice,0) < 0;

/* Integridad referencial: invoice/track */
INSERT INTO stg.RowReject(SourceTable,SourceId,Issue)
SELECT 'InvoiceLine', il.InvoiceLineId, 'Missing parent (Invoice or Track)'
FROM stg.InvoiceLine il
LEFT JOIN stg.Invoice si ON si.InvoiceId=il.InvoiceId
LEFT JOIN stg.Track st ON st.TrackId=il.TrackId
WHERE si.InvoiceId IS NULL OR st.TrackId IS NULL;

DELETE il
FROM stg.InvoiceLine il
LEFT JOIN stg.Invoice si ON si.InvoiceId=il.InvoiceId
LEFT JOIN stg.Track st ON st.TrackId=il.TrackId
WHERE si.InvoiceId IS NULL OR st.TrackId IS NULL;

/* Predeterminados para null money/price */
UPDATE stg.Invoice SET InvoiceTotal = 0 WHERE InvoiceTotal IS NULL;
UPDATE stg.InvoiceLine SET UnitPrice = 0 WHERE UnitPrice IS NULL;

/* -----
4) Creacion de tablas de dimension y hechos
-----*/

IF OBJECT_ID('dw.FactSales','U') IS NOT NULL DROP TABLE dw.FactSales;
IF OBJECT_ID('dw.DimTrack','U') IS NOT NULL DROP TABLE dw.DimTrack;
IF OBJECT_ID('dw.DimBillingLocation','U') IS NOT NULL DROP TABLE dw.DimBillingLocation;
IF OBJECT_ID('dw.DimCustomer','U') IS NOT NULL DROP TABLE dw.DimCustomer;
IF OBJECT_ID('dw.DimEmployee','U') IS NOT NULL DROP TABLE dw.DimEmployee;
IF OBJECT_ID('dw.DimDate','U') IS NOT NULL DROP TABLE dw.DimDate;

CREATE TABLE dw.DimDate(
    DateKey INT NOT NULL PRIMARY KEY, -- yyyyymmdd
    [Date] DATE NOT NULL UNIQUE,
    [Year] SMALLINT NOT NULL,
    [Quarter] TINYINT NOT NULL,
    [Month] TINYINT NOT NULL,
    [Day] TINYINT NOT NULL,
    MonthName NVARCHAR(15),
    DayName NVARCHAR(15),
    WeekOfYear TINYINT

```

0 %

Connected. (1/1)

NEKRON\SQLSERVER2025 (16.0 ... sa (93) master 00:00:00 0 row



```

CREATE TABLE dw.DimDate(
    DateKey INT NOT NULL PRIMARY KEY, -- yyyyymmdd
    [Date] DATE NOT NULL UNIQUE,
    [Year] SMALLINT NOT NULL,
    [Quarter] TINYINT NOT NULL,
    [Month] TINYINT NOT NULL,
    [Day] TINYINT NOT NULL,
    MonthName NVARCHAR(15),
    DayName NVARCHAR(15),
    WeekOfYear TINYINT
);

CREATE TABLE dw.DimEmployee(
    EmployeeKey INT IDENTITY(1,1) PRIMARY KEY,
    EmployeeId INT NOT NULL, -- NK
    FirstName NVARCHAR(20) NOT NULL,
    LastName NVARCHAR(20) NOT NULL,
    Title NVARCHAR(30),
    City NVARCHAR(40),
    State NVARCHAR(40),
    Country NVARCHAR(40),
    Email NVARCHAR(60)
);
CREATE UNIQUE INDEX UX_DimEmployee_NK ON dw.DimEmployee(EmployeeId);

CREATE TABLE dw.DimCustomer(
    CustomerKey INT IDENTITY(1,1) PRIMARY KEY,
    CustomerId INT NOT NULL, -- NK
    FirstName NVARCHAR(40) NOT NULL,
    LastName NVARCHAR(20) NOT NULL,
    Company NVARCHAR(80),
    Address NVARCHAR(70),
    City NVARCHAR(40),
    [State] NVARCHAR(40),
    Country NVARCHAR(40),
    PostalCode NVARCHAR(10),
    Phone NVARCHAR(24),
    Email NVARCHAR(60) NOT NULL,
    SupportRepId INT NULL, -- NK hacia Employee
    SupportRepName NVARCHAR(60) NULL
);
CREATE UNIQUE INDEX UX_DimCustomer_NK ON dw.DimCustomer(CustomerId);

CREATE TABLE dw.DimBillingLocation(
    BillingLocationKey INT IDENTITY(1,1) PRIMARY KEY,
    BillingAddress NVARCHAR(70),
    BillingCity NVARCHAR(40),
    BillingState NVARCHAR(40),
    BillingCountry NVARCHAR(40),
    BillingPostalCode NVARCHAR(10)
);
CREATE UNIQUE INDEX UX_DimBilling_UQ ON dw.DimBillingLocation(BillingAddress, BillingCity, BillingState, BillingCountry, BillingPostalCode);

CREATE TABLE dw.DimTrack(
    TrackKey INT IDENTITY(1,1) PRIMARY KEY,
    TrackId INT NOT NULL, -- NK
    TrackName NVARCHAR(200) NOT NULL,
    Composer NVARCHAR(220),
    Milliseconds INT NOT NULL,
    Bytes INT,
    UnitPriceList NUMERIC(10,2) NOT NULL,
    AlbumTitle NVARCHAR(160),
    ArtistName NVARCHAR(120),
    GenreName NVARCHAR(120),
    MediaTypeNames NVARCHAR(120)
);
CREATE UNIQUE INDEX UX_DimTrack_NK ON dw.DimTrack(TrackId);

CREATE TABLE dw.FactSales(
    FactSalesId BIGINT IDENTITY(1,1) PRIMARY KEY,
    DateKey INT NOT NULL FOREIGN KEY REFERENCES dw.DimDate(DateKey),
    CustomerKey INT NOT NULL FOREIGN KEY REFERENCES dw.DimCustomer(CustomerKey),
    EmployeeKey INT NULL FOREIGN KEY REFERENCES dw.DimEmployee(EmployeeKey),

```

30 %

Connected. (1/1)

NEKRON\SQLSERVER2025 (16.0 ... | sa (93) | master

```

CREATE TABLE dw.FactSales(
    FactSalesId BIGINT IDENTITY(1,1) PRIMARY KEY,
    DateKey INT NOT NULL FOREIGN KEY REFERENCES dw.DimDate(DateKey),
    CustomerKey INT NOT NULL FOREIGN KEY REFERENCES dw.DimCustomer(CustomerKey),
    EmployeeKey INT NOT NULL FOREIGN KEY REFERENCES dw.DimEmployee(EmployeeKey),
    BillingLocationKey INT NOT NULL FOREIGN KEY REFERENCES dw.DimBillingLocation(BillingLocationKey),
    TrackKey INT NOT NULL FOREIGN KEY REFERENCES dw.DimTrack(TrackKey),
    InvoiceId INT NOT NULL, -- degenerate
    InvoiceLineId INT NOT NULL, -- degenerate
    Quantity INT NOT NULL,
    UnitPrice NUMERIC(10,2) NOT NULL,
    LineTotal AS CAST(Quantity * UnitPrice AS NUMERIC(12,2)) PERSISTED
);
CREATE INDEX IX_FactSales_Date ON dw.FactSales(DateKey);
CREATE INDEX IX_FactSales_Customer ON dw.FactSales(CustomerKey);
CREATE INDEX IX_FactSales_Track ON dw.FactSales(TrackKey);

```

### Paso 3: Cargaremos datos a las tablas de dimensiones

```

MERGE dw.DimEmployee AS T
USING dbo.Employee AS S
ON T.EmployeeId=S.EmployeeId
WHEN MATCHED THEN UPDATE SET
    T.FirstName=S.FirstName, T.LastName=S.LastName, T.Title=S.Title,
    T.City=S.City, T.State=S.State, T.Country=S.Country, T.Email=S.Email
WHEN NOT MATCHED BY TARGET THEN
    INSERT(EmployeeId,FirstName,LastName,Title,City,State,Country,Email)
    VALUES(S.EmployeeId,S.FirstName,S.LastName,S.Title,S.City,S.State,S.Country,S.Email);

MERGE dw.DimCustomer AS T
USING (
    SELECT c.*, CONCAT(e.FirstName, ' ', e.LastName) AS SupportRepName
    FROM stg.Customer c
    LEFT JOIN dbo.Employee e ON e.EmployeeId=c.SupportRepId
) AS S
ON T.CustomerId=S.CustomerId
WHEN MATCHED THEN UPDATE SET
    T.FirstName=S.FirstName, T.LastName=S.LastName, T.Company=S.Company,
    T.Address=S.Address, T.City=S.City, T.[State]=S.[State], T.Country=S.Country,
    T.PostalCode=S.PostalCode, T.Phone=S.Phone, T.Email=S.Email,
    T.SupportRepId=S.SupportRepId, T.SupportRepName=S.SupportRepName
WHEN NOT MATCHED BY TARGET THEN
    INSERT(CustomerId,FirstName,LastName,Company,Address,City,[State],Country,PostalCode,Phone,Email,SupportRepId,SupportRepName)
    VALUES(S.CustomerId,S.FirstName,S.LastName,S.Company,S.Address,S.City,S.[State],S.Country,S.PostalCode,S.Phone,S.Email,S.SupportRepId,S
);

MERGE dw.DimBillingLocation AS T
USING (
    SELECT DISTINCT BillingAddress,BillingCity,BillingState,BillingCountry,BillingPostalCode
    FROM stg.Invoice
) AS S
ON T.BillingAddress=S.BillingAddress
AND ISNULL(T.BillingCity,'')=ISNULL(S.BillingCity,'')
AND ISNULL(T.BillingState,'')=ISNULL(S.BillingState,'')
AND T.BillingCountry=S.BillingCountry
AND ISNULL(T.BillingPostalCode,'')=ISNULL(S.BillingPostalCode,'')
WHEN NOT MATCHED BY TARGET THEN
    INSERT(BillingAddress,BillingCity,BillingState,BillingCountry,BillingPostalCode)
    VALUES(S.BillingAddress,S.BillingCity,S.BillingState,S.BillingCountry,S.BillingPostalCode);

MERGE dw.DimTrack AS T
USING stg.Track AS S
ON T.TrackId=S.TrackId
WHEN MATCHED THEN UPDATE SET
    T.TrackName=S.TrackName, T.Composer=S.Composer, T.Milliseconds=ISNULL(S.Milliseconds,0),
    T.Bytes=S.Bytes, T.UnitPriceList=ISNULL(S.UnitPriceList,0),
    T.AlbumTitle=S.AlbumTitle, T.ArtistName=S.ArtistName,
    T.GenreName=S.GenreName, T.MediaTypeName=S.MediaTypeName
WHEN NOT MATCHED BY TARGET THEN
    INSERT(TrackId,TrackName,Composer,Milliseconds,Bytes,UnitPriceList,AlbumTitle,ArtistName,GenreName,MediaType)
    VALUES(S.TrackId,S.TrackName,S.Composer,ISNULL(S.Milliseconds,0),S.Bytes,ISNULL(S.UnitPriceList,0),S.AlbumTitle,S.ArtistName,S.GenreName,S.MediaType);

/* DimDate from stg.Invoice range */
IF NOT EXISTS (SELECT 1 FROM dw.DimDate)
BEGIN
    WITH D AS (
        SELECT MIN(InvoiceDate) d0, MAX(InvoiceDate) d1 FROM stg.Invoice
    ),
    Cal AS (
        SELECT d0 AS [Date], d1 FROM D
        UNION ALL SELECT DATEADD(DAY,1,[Date]), d1 FROM Cal WHERE [Date] < d1
    )
    INSERT INTO dw.DimDate(DateKey,[Date],[Year],[Quarter],[Month],[Day],MonthName,DayName,WeekOfYear)
    SELECT CONVERT(INT,FORMAT([Date],'yyyyMMdd')),[Date],
        DATEPART(YEAR,[Date]),DATEPART(QUARTER,[Date]),DATEPART(MONTH,[Date]),DATEPART(DAY,[Date]),
        DATENAME(MONTH,[Date]),DATENAME(WEEKDAY,[Date]), DATEPART(WEEK,[Date])
    FROM Cal OPTION (MAXRECURSION 0);
END;

/* -----
6) Carga de FactSales
-----*/
INSERT INTO dw.FactSales
(DateKey,CustomerKey,EmployeeKey,BillingLocationKey,TrackKey,InvoiceId,InvoiceLineId,Quantity,UnitPrice)
SELECT

```

#### Paso 4: Cargaremos datos a la tabla de hechos

```
/* -----
6) Carga de FactSales
-----*/
INSERT INTO dw.FactSales
(DateKey, CustomerKey, EmployeeKey, BillingLocationKey, TrackKey, InvoiceId, InvoiceLineId, Quantity, UnitPrice)
SELECT
    CONVERT(INT, FORMAT(i.InvoiceDate, 'yyyyMMdd')) AS DateKey,
    dc.CustomerKey,
    de.EmployeeKey,
    db1.BillingLocationKey,
    dt.TrackKey,
    i.InvoiceId,
    il.InvoiceLineId,
    il.Quantity,
    il.UnitPrice
FROM stg.InvoiceLine il
JOIN stg.Invoice i ON i.InvoiceId=il.InvoiceId
JOIN dw.DimCustomer dc ON dc.CustomerId=i.CustomerId
LEFT JOIN dbo.Employee e ON e.EmployeeId=dc.SupportRepId
LEFT JOIN dw.DimEmployee de ON de.EmployeeId=e.EmployeeId
LEFT JOIN dw.DimBillingLocation db1
    ON db1.BillingAddress = i.BillingAddress
    AND ISNULL(db1.BillingCity, '') = ISNULL(i.BillingCity, '')
    AND ISNULL(db1.BillingState, '') = ISNULL(i.BillingState, '')
    AND db1.BillingCountry = i.BillingCountry
    AND ISNULL(db1.BillingPostalCode, '')=ISNULL(i.BillingPostalCode, '')
JOIN dw.DimTrack dt ON dt.TrackId=il.TrackId;
```

## Paso 5: Implementamos agregaciones para propósitos de limpieza de datos

```
SQLQuery1.sql - NEK...25.master (sa (65)) 2)Proceso ETL Opti...25.master (sa (93)) -p X
FROM stg.InvoiceLine il
JOIN stg.Invoice i ON i.InvoiceId=il.InvoiceId
JOIN dw.DimCustomer dc ON dc.CustomerId=i.CustomerId
LEFT JOIN dbo.Employee e ON e.EmployeeId=dc.SupportRepId
LEFT JOIN dw.DimEmployee de ON de.EmployeeId=e.EmployeeId
LEFT JOIN dw.DimBillingLocation dbl
ON dbl.BillingAddress = i.BillingAddress
AND ISNULL(dbl.BillingCity,'') = ISNULL(i.BillingCity,'')
AND ISNULL(dbl.BillingState,'') = ISNULL(i.BillingState,'')
AND dbl.BillingCountry = i.BillingCountry
AND ISNULL(dbl.BillingPostalCode,'')=ISNULL(i.BillingPostalCode,'')
JOIN dw.DimTrack dt ON dt.TrackId=il.TrackId;

/*
7) Agregados (materialized)
*/
IF OBJECT_ID('dw.AggregCustomerSales','U') IS NOT NULL DROP TABLE dw.AggregCustomerSales;
CREATE TABLE dw.AggregCustomerSales(
    CustomerKey INT NOT NULL PRIMARY KEY,          -- FK a DimCustomer
    CustomerId INT NOT NULL,                        -- NK (for convenience)
    FirstInvoice DATE NULL,
    LastInvoice DATE NULL,
    InvoiceCount INT NOT NULL,
    LineCount INT NOT NULL,
    Units INT NOT NULL,
    Amount NUMERIC(14,2) NOT NULL,                  -- total spent by customer
    AvgTicket NUMERIC(14,2) NULL
);

IF OBJECT_ID('dw.AggregSalesByGenre','U') IS NOT NULL DROP TABLE dw.AggregSalesByGenre;
CREATE TABLE dw.AggregSalesByGenre(
    GenreName NVARCHAR(120) NOT NULL PRIMARY KEY,
    Units INT NOT NULL,
    Amount NUMERIC(14,2) NOT NULL
);

IF OBJECT_ID('dw.AggregSalesByArtist','U') IS NOT NULL DROP TABLE dw.AggregSalesByArtist;
CREATE TABLE dw.AggregSalesByArtist(
    ArtistName NVARCHAR(120) NOT NULL PRIMARY KEY,
    Units INT NOT NULL,
    Amount NUMERIC(14,2) NOT NULL
);

IF OBJECT_ID('dw.AggregSalesByCustomerCountry','U') IS NOT NULL DROP TABLE dw.AggregSalesByCustomerCountry;
CREATE TABLE dw.AggregSalesByCustomerCountry(
    Country NVARCHAR(40) NOT NULL PRIMARY KEY,
    Units INT NOT NULL,
    Amount NUMERIC(14,2) NOT NULL
);

WITH PerInvoice AS (
    SELECT
        fs.CustomerKey, fs.InvoiceId,
        CAST(MIN(d.[Date]) AS DATE) AS InvoiceDate,
        SUM(fs.Quantity) AS UnitsByInvoice,
        SUM(fs.LineTotal) AS AmountByInvoice
    FROM dw.FactSales fs
    JOIN dw.DimDate d ON d.DateKey = fs.DateKey
    GROUP BY fs.CustomerKey, fs.InvoiceId
)
INSERT INTO dw.AggregCustomerSales(CustomerKey, CustomerId, FirstInvoice, LastInvoice, InvoiceCount, LineCount, Units, Amount, AvgTicket)
SELECT
    dc.CustomerKey,
    dc.CustomerId,
    MIN(pi.InvoiceDate) AS FirstInvoice,
    MAX(pi.InvoiceDate) AS LastInvoice,
    COUNT(*) AS InvoiceCount,
    SUM(pi.UnitsByInvoice) AS LineCount,
    SUM(pi.UnitsByInvoice) AS Units,
    CAST(SUM(pi.AmountByInvoice) AS NUMERIC(14,2)) AS Amount,
    CAST(AVG(pi.AmountByInvoice) AS NUMERIC(14,2)) AS AvgTicket
FROM PerInvoice pi
JOIN dw.DimCustomer dc ON dc.CustomerKey = pi.CustomerKey
GROUP BY dc.CustomerKey, dc.CustomerId;

INSERT INTO dw.AggregSalesByGenre(GenreName, Units, Amount)
SELECT
    dt.GenreName,
    SUM(fs.Quantity) AS Units,
```

Paso 6: Realizaremos unos chequeos aleatorios durante la ejecución del código para ver que todo esta bien

```
/* -----
8) (Inpeccion calidad) QA checks (non-fatal; inspect outputs)
-----*/
-- a) counts
SELECT (SELECT COUNT(*) FROM dbo.InvoiceLine) AS SrcLines,
       (SELECT COUNT(*) FROM dw.FactSales) AS FactLines;

-- b) global reconciliation (detail sums)
SELECT
  (SELECT SUM(CAST(Quantity*UnitPrice AS NUMERIC(12,2))) FROM dbo.InvoiceLine) AS SrcDetailTotal,
  (SELECT SUM(LineTotal) FROM dw.FactSales) AS FactDetailTotal;

-- c) per-invoice reconciliation (header vs detail) - any rows returned indicate differences > 1 cent
SELECT i.InvoiceId, i.Total AS HeaderTotal, SUM(fs.LineTotal) AS DetailTotal,
       i.Total - SUM(fs.LineTotal) AS Diff
FROM dbo.Invoice i
JOIN dw.FactSales fs ON fs.InvoiceId=i.InvoiceId
GROUP BY i.InvoiceId, i.Total
HAVING ABS(i.Total - SUM(fs.LineTotal)) > 0.01;

-- d) orphan checks
SELECT TOP(1) 'Missing DateKey' AS Issue
FROM dw.FactSales fs LEFT JOIN dw.DimDate dd ON dd.DateKey=fs.DateKey
WHERE dd.DateKey IS NULL
UNION ALL
SELECT TOP(1) 'Missing CustomerKey'
FROM dw.FactSales fs LEFT JOIN dw.DimCustomer dc ON dc.CustomerKey=fs.CustomerKey
WHERE dc.CustomerKey IS NULL
UNION ALL
SELECT TOP(1) 'Missing TrackKey'
FROM dw.FactSales fs LEFT JOIN dw.DimTrack dt ON dt.TrackKey=fs.TrackKey
WHERE dt.TrackKey IS NULL;

-- e) negative/zero measures
SELECT COUNT(*) AS BadMeasureRows
FROM dw.FactSales
WHERE Quantity <= 0 OR UnitPrice < 0;

-- f) quick views
SELECT TOP (10) * FROM dw.AggsCustomerSales ORDER BY Amount DESC;
SELECT TOP (10) * FROM dw.AggsSalesByGenre ORDER BY Amount DESC;
SELECT TOP (10) * FROM dw.AggsSalesByArtist ORDER BY Amount DESC;
SELECT TOP (10) * FROM dw.AggsSalesByCustomerCountry ORDER BY Amount DESC;

PRINT('DW ETL completed.');
```

## Diagrama creado con éxito incluyendo pruebas rápidas dentro del mismo

137)) - Microsoft SQL Server Management Studio

Help

SQLQuery1.sql - NE....Chinook (sa (137)) \* 1)Base Chinook.sql...25.Chinook (sa (66))

```
INSERT INTO dw.AggsalesByCustomerCountry(Country, Units, Amount)
SELECT dc.Country,
       SUM(fs.Quantity) AS Units,
       CAST(SUM(fs.LineTotal) AS NUMERIC(14,2)) AS Amount
FROM dw.FactSales fs
JOIN dw.DimCustomer dc ON dc.CustomerKey = fs.CustomerKey
GROUP BY dc.Country;

/*
8) (Inpeccion calidad) QA checks (non-fatal; inspect outputs)
-----*/
-- a) counts
SELECT (SELECT COUNT(*) FROM dbo.InvoiceLine) AS SrcLines,
       (SELECT COUNT(*) FROM dw.FactSales) AS FactLines;

-- b) global reconciliation (detail sums)
SELECT
  (SELECT SUM(CAST(Quantity*UnitPrice AS NUMERIC(12,2))) FROM dbo.InvoiceLine) AS SrcDetailTotal,
  (SELECT SUM(LineTotal) FROM dw.FactSales) AS FactDetailTotal;

-- c) per-invoice reconciliation (header vs detail) - any rows returned indicate differences > 1 cent
SELECT i.InvoiceId, i.Total AS HeaderTotal, SUM(fs.LineTotal) AS DetailTotal,
       i.Total - SUM(fs.LineTotal) AS Diff
FROM dbo.Invoice i
JOIN dw.FactSales fs ON fs.InvoiceId=i.InvoiceId
GROUP BY i.InvoiceId, i.Total
HAVING ABS(i.Total - SUM(fs.LineTotal)) > 0.01;
```

90 %

Results Messages

	SrcLines	FactLines
1	2240	2240

	SrcDetailTotal	FactDetailTotal
1	2328.60	2328.60

	InvoiceId	HeaderTotal	DetailTotal	Diff
--	-----------	-------------	-------------	------

	Issue
--	-------

	BadMeasureRows
1	0

	CustomerKey	CustomerId	FirstInvoice	LastInvoice	InvoiceCount	LineCount	Units	Amount	AvgTicket
1	6	6	2009-07-11	2013-11-13	7	38	38	49.62	7.09
2	26	26	2009-11-07	2013-04-05	7	38	38	47.62	6.80
3	57	57	2009-04-04	2012-10-14	7	38	38	46.62	6.66
4	45	45	2010-01-08	2013-07-20	7	38	38	45.62	6.52
5	46	46	2009-02-03	2013-11-04	7	38	38	45.62	6.52
6	24	24	2010-02-08	2013-08-20	7	38	38	43.62	6.23
7	28	28	2009-11-07	2013-05-19	7	38	38	43.62	6.23
8	27	27	2009-01-10	2013-06-03	7	38	38	43.62	6.23

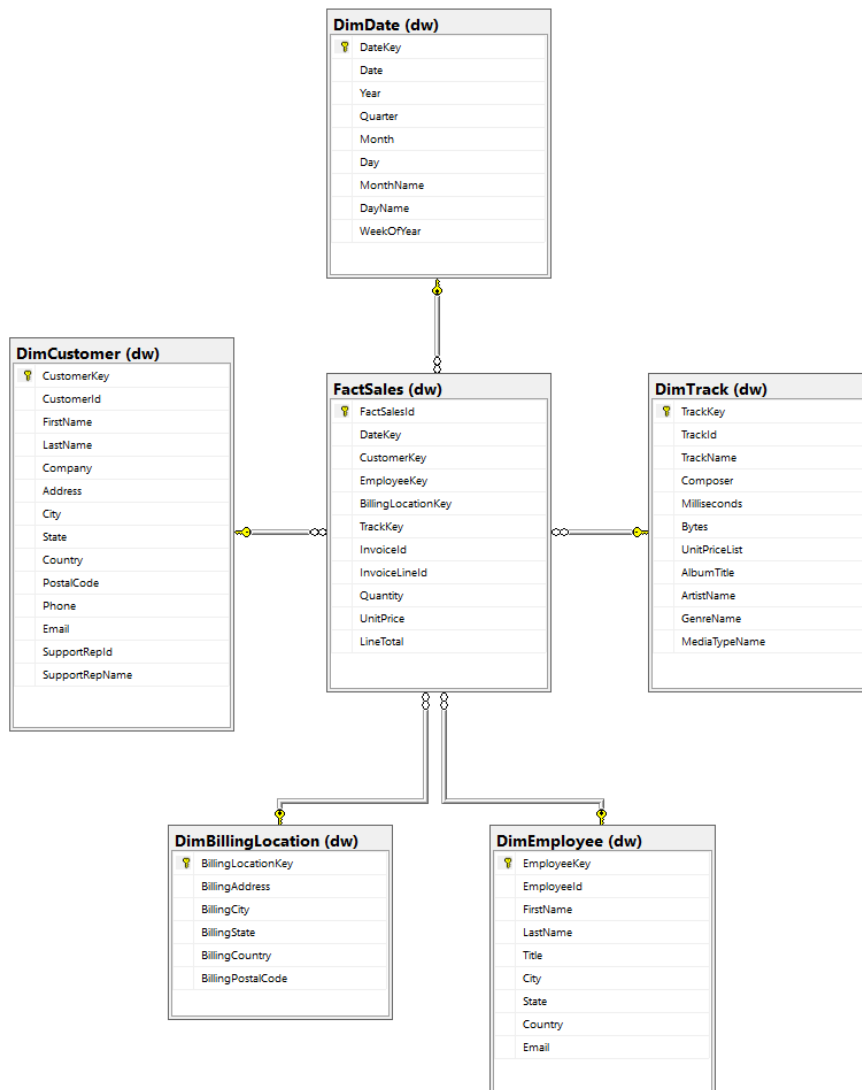
Query executed successfully.

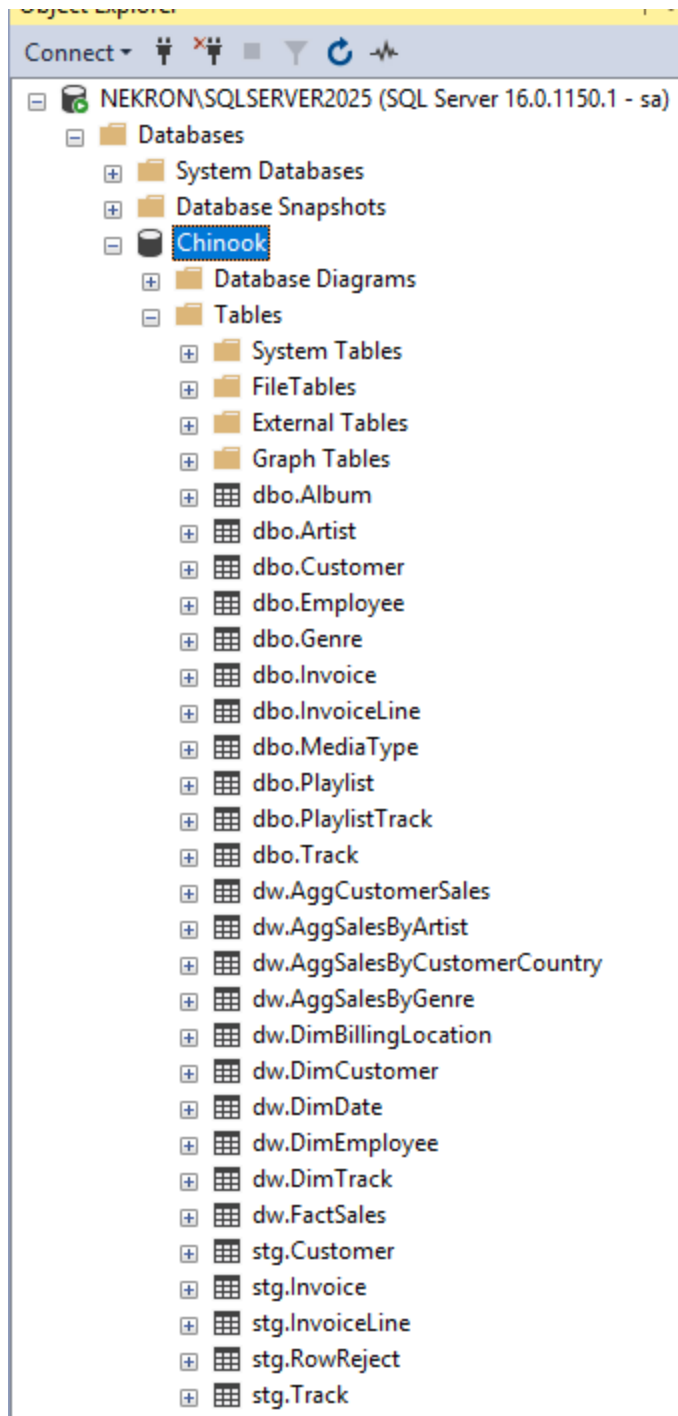
Ln 460 Col 1 Ch 1

**Tablas de dimensiones:** DimDate, DimCustomer, DimBillingLocation, DimEmployee, DimTrack.

**Tabla de Hechos:** FactSales

### Diagrama multidimensional de estrella

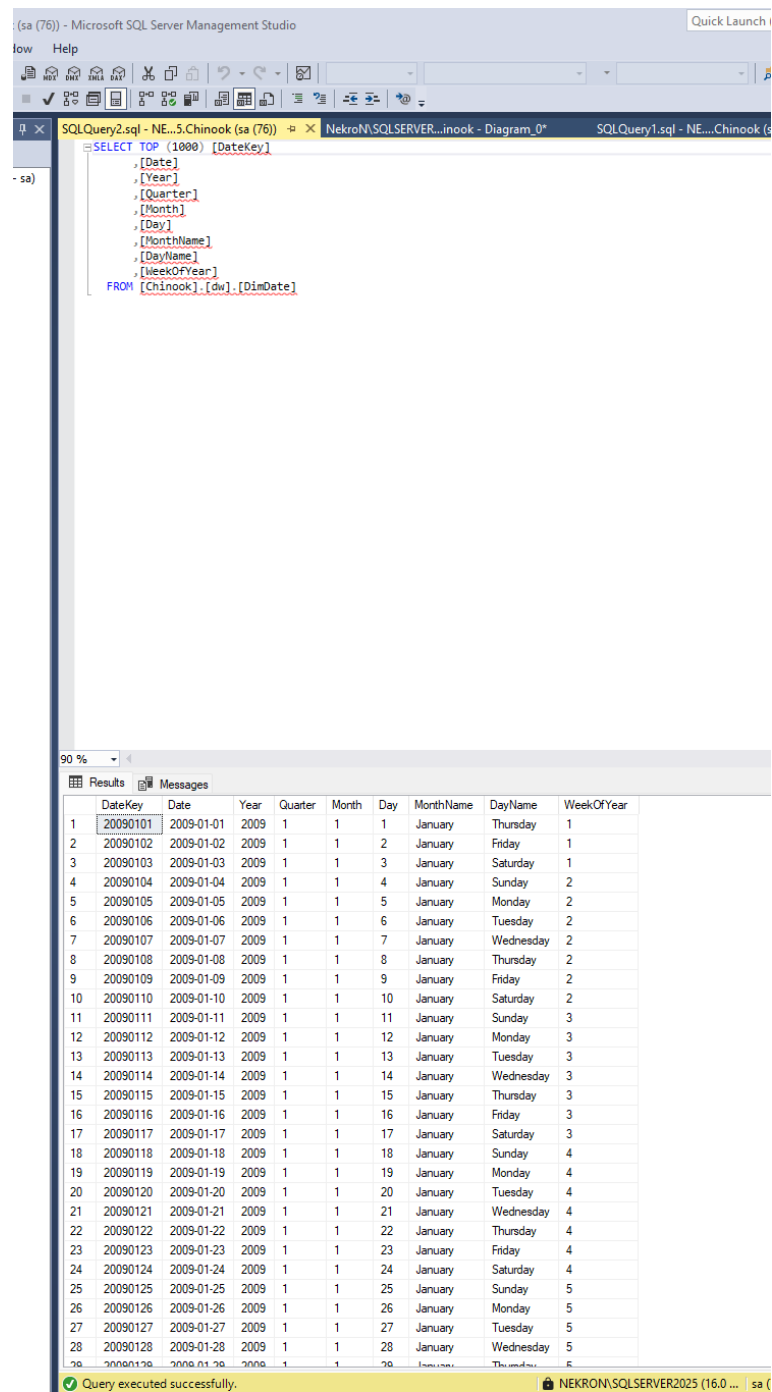






Mostraremos las tablas de dimensiones creadas con éxito:

DimDate:



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays a SQL query that selects the top 1000 rows from the DimDate table in the Chinook database. The query includes columns for DateKey, Date, Year, Quarter, Month, Day, MonthName, DayName, and WeekOfYear. The results pane shows the first 28 rows of the query results, which are dates from 2009-01-01 to 2009-01-28. The status bar at the bottom indicates that the query was executed successfully.

```
SELECT TOP (1000) [DateKey]
, [Date]
, [Year]
, [Quarter]
, [Month]
, [Day]
, [MonthName]
, [DayName]
, [WeekOfYear]
FROM [Chinook].[dw].[DimDate]
```

	DateKey	Date	Year	Quarter	Month	Day	MonthName	DayName	WeekOfYear
1	20090101	2009-01-01	2009	1	1	1	January	Thursday	1
2	20090102	2009-01-02	2009	1	1	2	January	Friday	1
3	20090103	2009-01-03	2009	1	1	3	January	Saturday	1
4	20090104	2009-01-04	2009	1	1	4	January	Sunday	2
5	20090105	2009-01-05	2009	1	1	5	January	Monday	2
6	20090106	2009-01-06	2009	1	1	6	January	Tuesday	2
7	20090107	2009-01-07	2009	1	1	7	January	Wednesday	2
8	20090108	2009-01-08	2009	1	1	8	January	Thursday	2
9	20090109	2009-01-09	2009	1	1	9	January	Friday	2
10	20090110	2009-01-10	2009	1	1	10	January	Saturday	2
11	20090111	2009-01-11	2009	1	1	11	January	Sunday	3
12	20090112	2009-01-12	2009	1	1	12	January	Monday	3
13	20090113	2009-01-13	2009	1	1	13	January	Tuesday	3
14	20090114	2009-01-14	2009	1	1	14	January	Wednesday	3
15	20090115	2009-01-15	2009	1	1	15	January	Thursday	3
16	20090116	2009-01-16	2009	1	1	16	January	Friday	3
17	20090117	2009-01-17	2009	1	1	17	January	Saturday	3
18	20090118	2009-01-18	2009	1	1	18	January	Sunday	4
19	20090119	2009-01-19	2009	1	1	19	January	Monday	4
20	20090120	2009-01-20	2009	1	1	20	January	Tuesday	4
21	20090121	2009-01-21	2009	1	1	21	January	Wednesday	4
22	20090122	2009-01-22	2009	1	1	22	January	Thursday	4
23	20090123	2009-01-23	2009	1	1	23	January	Friday	4
24	20090124	2009-01-24	2009	1	1	24	January	Saturday	4
25	20090125	2009-01-25	2009	1	1	25	January	Sunday	5
26	20090126	2009-01-26	2009	1	1	26	January	Monday	5
27	20090127	2009-01-27	2009	1	1	27	January	Tuesday	5
28	20090128	2009-01-28	2009	1	1	28	January	Wednesday	5

Query executed successfully.

## DimTrack

SQLQuery3.sql - NE...S.Chinook (sa (78))

SQLQuery2.sql - NE...S.Chinook (sa (76))

Nekron\SQLSERVER...inook - Diagram...\*

SQLQuery1.sql - NE...Chinook (sa (137))

Base Chinook.sql...25.Chinook (sa (66))

```

SELECT TOP (1000) [TrackKey]
, [TrackId]
, [TrackName]
, [Composer]
, [Milliseconds]
, [Bytes]
, [UnitPriceList]
, [AlbumTitle]
, [ArtistName]
, [GenreName]
, [MediaType]
FROM [Chinook].[dbo].[DimTrack]

```

90 %

Results Messages

	TrackKey	TrackId	TrackName	Composer	Milliseconds	Bytes	UnitPriceList	Album Title	ArtistName	GenreName	MediaType
1	1	1	For Those About To Rock (We Salute You)	Angus Young, Malcolm Young, Brian Johnson	343719	11170334	0.99	For Those About To Rock We Salute You	AC/DC	Rock	MPEG audio file
2	2	2	Balls to the Wall	NULL	342562	5510424	0.99	Balls to the Wall	Accept	Rock	Protected AAC audio file
3	3	3	Fast As a Shark	F. Baltes, S. Kaufman, U. Dirksneider & W. Hoff...	230619	3990994	0.99	Restless and Wild	Accept	Rock	Protected AAC audio file
4	4	4	Restless and Wild	F. Baltes, R.A. Smith-Diesel, S. Kaufman, U. Dirks...	252051	4331779	0.99	Restless and Wild	Accept	Rock	Protected AAC audio file
5	5	5	Princess of the Dawn	Deafy & R.A. Smith-Diesel	375418	6290521	0.99	Restless and Wild	Accept	Rock	Protected AAC audio file
6	6	6	Put The Finger On You	Angus Young, Malcolm Young, Brian Johnson	205662	6713451	0.99	For Those About To Rock We Salute You	AC/DC	Rock	MPEG audio file
7	7	7	Let's Get It Up	Angus Young, Malcolm Young, Brian Johnson	233926	7636561	0.99	For Those About To Rock We Salute You	AC/DC	Rock	MPEG audio file
8	8	8	Inject The Venom	Angus Young, Malcolm Young, Brian Johnson	210834	6852860	0.99	For Those About To Rock We Salute You	AC/DC	Rock	MPEG audio file
9	9	9	Snowballed	Angus Young, Malcolm Young, Brian Johnson	203102	6599424	0.99	For Those About To Rock We Salute You	AC/DC	Rock	MPEG audio file
10	10	10	Evil Walks	Angus Young, Malcolm Young, Brian Johnson	263497	8611245	0.99	For Those About To Rock We Salute You	AC/DC	Rock	MPEG audio file
11	11	11	C.O.D.	Angus Young, Malcolm Young, Brian Johnson	199836	6566314	0.99	For Those About To Rock We Salute You	AC/DC	Rock	MPEG audio file
12	12	12	Breaking The Rules	Angus Young, Malcolm Young, Brian Johnson	263288	8596840	0.99	For Those About To Rock We Salute You	AC/DC	Rock	MPEG audio file
13	13	13	Night Of The Long Knives	Angus Young, Malcolm Young, Brian Johnson	205688	6706347	0.99	For Those About To Rock We Salute You	AC/DC	Rock	MPEG audio file
14	14	14	Spellbound	Angus Young, Malcolm Young, Brian Johnson	270863	8817038	0.99	For Those About To Rock We Salute You	AC/DC	Rock	MPEG audio file
15	15	15	Go Down	AC/DC	331180	10847611	0.99	Let There Be Rock	AC/DC	Rock	MPEG audio file
16	16	16	Dog Eat Dog	AC/DC	215196	7032162	0.99	Let There Be Rock	AC/DC	Rock	MPEG audio file
17	17	17	Let There Be Rock	AC/DC	366654	10221261	0.99	Let There Be Rock	AC/DC	Rock	MPEG audio file
18	18	18	Bad Boy Boogie	AC/DC	267728	8776140	0.99	Let There Be Rock	AC/DC	Rock	MPEG audio file
19	19	19	Problem Child	AC/DC	325041	10617116	0.99	Let There Be Rock	AC/DC	Rock	MPEG audio file
20	20	20	Overdose	AC/DC	369319	12066294	0.99	Let There Be Rock	AC/DC	Rock	MPEG audio file
21	21	21	Hell Ain't A Bad Place To Be	AC/DC	254380	8331286	0.99	Let There Be Rock	AC/DC	Rock	MPEG audio file
22	22	22	Whole Lotta Rosie	AC/DC	323761	10547154	0.99	Let There Be Rock	AC/DC	Rock	MPEG audio file
23	23	23	Walk On Water	Steven Tyler, Joe Perry, Jack Blades, Tommy Shaw	295680	9719579	0.99	Big Ones	Aerosmith	Rock	MPEG audio file
24	24	24	Love In An Elevator	Steven Tyler, Joe Perry	321828	10552051	0.99	Big Ones	Aerosmith	Rock	MPEG audio file
25	25	25	Rag Doll	Steven Tyler, Joe Perry, Jim Vallance, Holly Knight	264698	8675345	0.99	Big Ones	Aerosmith	Rock	MPEG audio file
26	26	26	What It Takes	Steven Tyler, Joe Perry, Desmond Child	310622	10144730	0.99	Big Ones	Aerosmith	Rock	MPEG audio file
27	27	27	Dude Looks Like A Lady	Steven Tyler, Joe Perry, Desmond Child	264855	8679940	0.99	Big Ones	Aerosmith	Rock	MPEG audio file
28	28	28	Janie's Got A Gun	Steven Tyler, Tom Hamilton	330736	10869391	0.99	Big Ones	Aerosmith	Rock	MPEG audio file
29	29	29	Crimin...	Steven Tyler, Joe Perry, Desmond Child	296793	10066606	0.99	Big Ones	Aerosmith	Rock	MPEG audio file

Query executed successfully.

NEKRON\SQLSERVER2025 (16.0... sa (78)) Chinook 00:00:00 1,000 rows

## DimEmployee

SQLQuery4.sql - NE...5.Chinook (sa (65)) SQLQuery3.sql - NE...5.Chinook (sa (78)) SQLQuery2.sql - NE...5.Chinook (sa (76))

```
SELECT TOP (1000) [EmployeeKey]
, [EmployeeId]
, [FirstName]
, [LastName]
, [Title]
, [City]
, [State]
, [Country]
, [Email]
FROM [Chinook].[dw].[DimEmployee]
```

90 %

Results Messages

	EmployeeKey	EmployeeId	FirstName	LastName	Title	City	State	Country	Email
1	1	1	Andrew	Adams	General Manager	Edmonton	AB	Canada	andrew@chinookcorp.com
2	2	2	Nancy	Edwards	Sales Manager	Calgary	AB	Canada	nancy@chinookcorp.com
3	3	3	Jane	Peacock	Sales Support Agent	Calgary	AB	Canada	jane@chinookcorp.com
4	4	4	Margaret	Park	Sales Support Agent	Calgary	AB	Canada	margaret@chinookcorp.com
5	5	5	Steve	Johnson	Sales Support Agent	Calgary	AB	Canada	steve@chinookcorp.com
6	6	6	Michael	Mitchell	IT Manager	Calgary	AB	Canada	michael@chinookcorp.com
7	7	7	Robert	King	IT Staff	Lethbridge	AB	Canada	robert@chinookcorp.com
8	8	8	Laura	Callahan	IT Staff	Lethbridge	AB	Canada	laura@chinookcorp.com

## DimBillingLocation

SQLQuery5.sql - NE...5.Chinook (sa (66)) - X NekoN\SQLSERVER...inook - Diagram\_0\* SQLQuery1.sql - NE...Chinook (sa (137))\*

```
SELECT TOP (1000) [BillingLocationKey]
, [BillingAddress]
, [BillingCity]
, [BillingState]
, [BillingCountry]
, [BillingPostalCode]
FROM [Chinook].[dw].[DimBillingLocation]
```

10 %

Results Messages

	BillingLocationKey	BillingAddress	BillingCity	BillingState	BillingCountry	BillingPostalCode
1	1	1 Infinite Loop	Cupertino	CA	USA	95014
2	2	1 Microsoft Way	Redmond	WA	USA	98052-8300
3	3	1033 N Park Ave	Tucson	AZ	USA	85719
4	4	11, Place Bellecour	Lyon	NULL	FRANCE	69002
5	5	110 Raeburn Pl	Edinburgh	NULL	UNITED KINGDOM	EH4 1HH
6	6	113 Lupus St	London	NULL	UNITED KINGDOM	SW1V 3EN
7	7	12,Community Centre	Delhi	NULL	INDIA	110017
8	8	120 S Orange Ave	Orlando	FL	USA	32801
9	9	1498 rue Bélanger	Montréal	QC	CANADA	H2G 1A7
10	10	1600 Amphitheatre Parkway	Mountain View	CA	USA	94043-1351
11	11	162 E Superior Street	Chicago	IL	USA	60611
12	12	194A Chain Lake Drive	Halifax	NS	CANADA	B3S 1C5
13	13	202 Hoxton Street	London	NULL	UNITED KINGDOM	N1 5LH
14	14	2211 W Berry Street	Fort Worth	TX	USA	76110
15	15	230 Elgin Street	Ottawa	ON	CANADA	K2P 1L7
16	16	3 Chatham Street	Dublin	Dublin	IRELAND	NULL
17	17	3,Raj Bhavan Road	Bangalore	NULL	INDIA	560001
18	18	302 S 700 E	Salt Lake City	UT	USA	84102
19	19	307 Macacha Güemes	Buenos Aires	NULL	ARGENTINA	1106
20	20	319 N. Frances Street	Madison	WI	USA	53703
21	21	4, Rue Milton	Paris	NULL	FRANCE	75009
22	22	421 Bourke Street	Sidney	NSW	AUSTRALIA	2010
23	23	5112 48 Street	Yellowknife	NT	CANADA	X1A 1N6
24	24	541 Del Medio Avenue	Mountain View	CA	USA	94040-1111
25	25	627 Broadway	New York	NY	USA	10012-2612
26	26	68, Rue Jouvence	Dijon	NULL	FRANCE	21000
27	27	69 Salem Street	Boston	MA	USA	2113
28	28	696 Osborne Street	Winnipeg	MB	CANADA	R3L 2B9
29	29	700 W. Beaver Street	Vancouver	BC	CANADA	V6C 1C8

Query executed successfully. NEKRON\SQLSERVER2025 (16.0 ... sa (66)) Chinook 00:00:00 59 rows

# DimCustomer

65) - Microsoft SQL Server Management Studio

SQLQuery6.sql - NE...S.Chinook (sa (65))

SQLQuery5.sql - NE...S.Chinook (sa (66))

Nekron\SQLSERVER...inook - Diagram\_0\*

SQLQuery1.sql - NE...Chinook (sa (137))\*

```
--SELECT TOP (1000) [CustomerKey]
,[CustomerId]
,[FirstName]
,[LastName]
,[Company]
,[Address]
,[City]
,[State]
,[Country]
,[PostalCode]
,[Phone]
,[Email]
,[SupportRepId]
,[SupportRepName]
FROM [Chinook].[dw].[DimCustomer]
```

90 %

	CustomerKey	CustomerId	FirstName	LastName	Company	Address	City	State	Country	PostalCode	Phone	Email	SupportRepId	SupportRepName
1	1	1	Luis	Gongalves	Embraer - Empresa Brasileira de Aeronáutica S.A.	Av. Brigadeiro Faria Lima, 2170	São José dos Campos	SP	BRAZIL	12227-000	+55 (12) 3923-5555	luisg@embraer.com.br	3	Jane Peacock
2	2	2	Leonie	Köhler	NULL	Theodor-Heuss-Straße 34	Stuttgart	NULL	GERMANY	70174	+49 0711 2842222	leoniekohler@surfeu.de	5	Steve Johnson
3	3	3	François	Tremblay	NULL	1490 rue Bélanger	Montréal	QC	CANADA	H2G 1A7	+1 (514) 721-4711	ftremblay@gmail.com	3	Jane Peacock
4	4	4	Bjorn	Hansen	NULL	Lilleveivveien 14	Oslo	NULL	NORWAY	0171	+47 22 44 22 22	bjorn.hansen@yahoo.no	4	Margaret Park
5	5	5	František	Wichterlová	JarBrains s.r.o.	Klanova 5/506	Prague	NULL	CZECH REPUBLIC	14700	+420 2 4172 5555	frantisek.w@jarbrains.com	4	Margaret Park
6	6	6	Helena	Hajó	NULL	Rákó 3174-6	Prague	NULL	CZECH REPUBLIC	14300	+420 2 4177 0449	hholy@gmail.com	5	Steve Johnson
7	7	7	Astid	Gruher	NULL	Rotentumstraße 4, 1010 Innere Stadt	Vienne	NULL	AUSTRIA	1010	+43 01 5134505	astid.gruber@apple.at	5	Steve Johnson
8	8	8	Daan	Peeters	NULL	Geltrykstraat 63	Brussels	NULL	BELGIUM	1000	+32 02 219 03 03	daan.peeters@apple.be	4	Margaret Park
9	9	9	Kara	Nielsen	NULL	Sander Boulevard 51	Copenhagen	NULL	DENMARK	1720	+453 3331 9991	kara.nielsen@tubi.dk	4	Margaret Park
10	10	10	Eduardo	Martins	Woodstock Discos	Rua Dr. Falcão Filho, 155	São Paulo	SP	BRAZIL	01007-010	+55 (11) 3033-5446	eduardo@woodstock.com.br	4	Margaret Park
11	11	11	Alexandre	Rocha	Banco do Brasil S.A.	Av. Paulista, 2022	São Paulo	SP	BRAZIL	01310-200	+55 (11) 3055-3278	alero@uol.com.br	5	Steve Johnson
12	12	12	Roberto	Almeida	Riotur	Prça Pio X, 119	Rio de Janeiro	RJ	BRAZIL	20040-020	+55 (21) 2271-7000	roberto.almeida@riotur.gov.br	3	Jane Peacock
13	13	13	Fernanda	Ramos	NULL	Ge 7 Bloco G	Brasília	DF	BRAZIL	71020-677	+55 (61) 3363-5547	fernandaramos4@uol.com.br	4	Margaret Park
14	14	14	Mark	Philips	Telus	8210 111 ST NW	Edmonton	AB	CANADA	T6G 2C7	+1 (780) 434-4554	mphilips12@shaw.ca	5	Steve Johnson
15	15	15	Jennifer	Peterson	Rogers Canada	700 W Pender Street	Vancouver	BC	CANADA	V6C 1G8	+1 (604) 688-2255	jennifep@rogers.ca	3	Jane Peacock
16	16	16	Frank	Harris	Google Inc.	1600 Amphitheatre Parkway	Mountain View	CA	USA	94043-1351	+1 (650) 253-0000	fharris@google.com	4	Margaret Park
17	17	17	Jack	Smith	Microsoft Corporation	1 Microsoft Way	Redmond	WA	USA	98052-8300	+1 (425) 882-8080	jacksmith@microsoft.com	5	Steve Johnson
18	18	18	Michelle	Brooks	NULL	627 Broadway	New York	NY	USA	10012-2612	+1 (212) 221-3546	micheleb@aol.com	3	Jane Peacock
19	19	19	Tim	Goyer	Apple Inc.	1 Infinite Loop	Cupertino	CA	USA	95014	+1 (408) 996-1010	tgoyer@apple.com	3	Jane Peacock
20	20	20	Dan	Miller	NULL	541 Del Medio Avenue	Mountain View	CA	USA	94040-1111	+1 (650) 544-3358	dmiller@comcast.com	4	Margaret Park
21	21	21	Kathy	Chase	NULL	801 W 4th Street	Reno	NV	USA	89503	+1 (775) 223-7665	kchase@hotmail.com	5	Steve Johnson
22	22	22	Heather	Leacock	NULL	120 S Orange Ave	Orlando	FL	USA	32801	+1 (407) 999-7788	hleacock@gmail.com	4	Margaret Park
23	23	23	John	Gordon	NULL	69 Salem Street	Boston	MA	USA	2113	+1 (617) 522-1333	johngordon22@yahoo.com	4	Margaret Park
24	24	24	Frank	Ralston	NULL	162 E Superior Street	Chicago	IL	USA	60611	+1 (312) 332-3232	fralston@gmail.com	3	Jane Peacock
25	25	25	Victor	Stevens	NULL	319 N. Frances Street	Madison	WI	USA	53703	+1 (608) 257-0597	vstevens@yahoo.com	5	Steve Johnson
26	26	26	Richard	Cunningham	NULL	2211 W Berry Street	Fort Worth	TX	USA	76110	+1 (817) 524-7272	rcunningham@hotmail.com	4	Margaret Park
27	27	27	Patrick	Gray	NULL	1033 N Park Ave	Tucson	AZ	USA	85719	+1 (520) 622-4200	patrick.gray@aol.com	4	Margaret Park
28	28	28	Julia	Barnett	NULL	302 S 700 E	Salt Lake City	UT	USA	84102	+1 (801) 531-7272	jubarnett@gmail.com	5	Steve Johnson
29	29	29	Robert	Brown	NULL	700 Dandridge Street West	Toronto	ON	CANADA	M6L 1A1	+1 (416) 563-8888	robbrown@robbrow.ca	3	Jane Peacock

Query executed successfully.

NEKRON\SQLSERVER2025 (16.0 ... sa (65) : Chinook : 001

## Tabla de hechos FactSales

SQLQuery7.sql - NE...5.Chinook (sa (75))											
SQLQuery6.sql - NE...5.Chinook (sa (65))											
SQLQuery5.sql - NE...5.Chinook (sa (66))											
NekroN\SQLSERVER...inook - Diagram_0*											
<pre> SELECT TOP (1000) [FactSalesId] , [DateKey] , [CustomerKey] , [EmployeeKey] , [BillingLocationKey] , [TrackKey] , [InvoiceId] , [InvoiceLineId] , [Quantity] , [UnitPrice] , [LineTotal] FROM [Chinook].[dw].[FactSales] </pre>											
90 %											
Results Messages											
FactSalesId	DateKey	CustomerKey	EmployeeKey	BillingLocationKey	TrackKey	InvoiceId	InvoiceLineId	Quantity	UnitPrice	LineTotal	
1	20100413	47	5	59	1	108	579	1	0.99	0.99	
2	20090101	2	5	57	2	1	1	1	0.99	0.99	
3	20110725	33	3	23	2	214	1154	1	0.99	0.99	
4	20121101	13	4	49	3	319	1728	1	0.99	0.99	
5	20090101	2	5	57	4	1	2	1	0.99	0.99	
6	20100413	47	5	59	5	108	580	1	0.99	0.99	
7	20090102	4	4	58	6	2	3	1	0.99	0.99	
8	20090102	4	4	58	8	2	4	1	0.99	0.99	
9	20110725	33	3	23	8	214	1155	1	0.99	0.99	
10	20121101	13	4	49	9	319	1729	1	0.99	0.99	
11	20100413	47	5	59	9	108	581	1	0.99	0.99	
12	20090102	4	4	58	10	2	5	1	0.99	0.99	
13	20090102	4	4	58	12	2	6	1	0.99	0.99	
14	20100413	47	5	59	13	108	582	1	0.99	0.99	
15	20110725	33	3	23	14	214	1156	1	0.99	0.99	
16	20121101	13	4	49	15	319	1730	1	0.99	0.99	
17	20090103	8	4	43	16	3	7	1	0.99	0.99	
18	20100416	53	3	6	19	109	583	1	0.99	0.99	
19	20090103	8	4	43	20	3	8	1	0.99	0.99	
20	20110725	33	3	23	20	214	1157	1	0.99	0.99	
21	20121101	13	4	49	21	319	1731	1	0.99	0.99	
22	20090103	8	4	43	24	3	9	1	0.99	0.99	
23	20100416	53	3	6	25	109	584	1	0.99	0.99	
24	20110725	33	3	23	26	214	1158	1	0.99	0.99	
25	20090103	8	4	43	28	3	10	1	0.99	0.99	
26	20121106	22	4	8	30	320	1732	1	0.99	0.99	
27	20100416	53	3	6	31	109	585	1	0.99	0.99	
28	20090103	8	4	43	32	3	11	1	0.99	0.99	
29	20110725	33	3	23	33	214	1159	1	0.99	0.99	

Query executed successfully.

NEKRON\SQLSERVER2025 (16.0 ... | sa (75) | Chinook | 00:00:00 | 1,0

Paso 7: Consultas de pruebas:

Ventas por cliente:

```
/*
Consultas de pruebas rapidas
*/

SET NOCOUNT ON;

/* -----
1) Total de ventas por CLIENTE
----- */
DECLARE @From INT
DECLARE @To INT
SELECT
    dc.CustomerId,
    dc.FirstName,
    dc.LastName,
    dc.Country,
    COUNT(DISTINCT fs.InvoiceId) AS Invoices,
    SUM(fs.Quantity) AS Units,
    CAST(SUM(fs.LineTotal) AS NUMERIC(14,2)) AS Amount
FROM dw.FactSales fs
JOIN dw.DimCustomer dc ON dc.CustomerKey = fs.CustomerKey
GROUP BY dc.CustomerId, dc.FirstName, dc.LastName, dc.Country;

/* Uso:
-- SELECT * FROM dw.vw_TotalVentasPorCliente ORDER BY Amount DESC;
-- Con filtro de fechas:
-- SELECT c.*
-- FROM dw.vw_TotalVentasPorCliente c
-- JOIN dw.FactSales fs ON fs.CustomerKey = (SELECT CustomerKey FROM dw.DimCustomer WHERE CustomerId = c.CustomerId)
-- JOIN dw.DimDate d ON d.DateKey = fs.DateKey
-- WHERE d.[Date] BETWEEN '2009-01-01' AND '2010-12-31'
-- GROUP BY c.CustomerId, c.FirstName, c.LastName, c.Country
-- ORDER BY SUM(fs.LineTotal) DESC;
*/

/* -----
2) Total de ventas por GÉNERO MUSICAL
----- */
```

90 %

Results Messages

	CustomerId	FirstName	LastName	Country	Invoices	Units	Amount
1	1	Luis	Gonçalves	BRAZIL	7	38	39.62
2	2	Leonie	Köhler	GERMANY	7	38	37.62
3	3	François	Tremblay	CANADA	7	38	39.62
4	4	Bjorn	Hansen	NORWAY	7	38	39.62
5	5	František	Wichterlová	CZECH REPUBLIC	7	38	40.62
6	6	Helena	Holý	CZECH REPUBLIC	7	38	49.62
7	7	Astrid	Gruber	AUSTRIA	7	38	42.62
8	8	Daan	Peeters	BELGIUM	7	38	37.62
9	9	Kara	Nielsen	DENMARK	7	38	37.62
10	10	Eduardo	Martins	BRAZIL	7	38	37.62
11	11	Alexandre	Rocha	BRAZIL	7	38	37.62
12	12	Roberto	Almeida	BRAZIL	7	38	37.62
13	13	Fernanda	Ramos	BRAZIL	7	38	37.62
14	14	Mark	Philips	CANADA	7	38	37.62
15	15	Jennifer	Peterson	CANADA	7	38	38.62
16	16	Frank	Harris	USA	7	38	37.62
17	17	Jack	Smith	USA	7	38	39.62
18	18	Michelle	Brooks	USA	7	38	37.62
19	19	Tim	Goyer	USA	7	38	38.62
20	20	Dan	Miller	USA	7	38	39.62
21	21	Kathy	Chase	USA	7	38	37.62
22	22	Heather	Leacock	USA	7	38	39.62
23	23	John	Gordon	USA	7	38	37.62
24	24	Frank	Ralston	USA	7	38	43.62
25	25	Victor	Stevens	USA	7	38	42.62
26	26	Richard	Cunningham	USA	7	38	47.62
27	27	Patrick	Gray	USA	7	38	37.62
28	28	Julia	Barnett	USA	7	38	43.62
29	29	Robert	Brown	CANADA	7	38	37.62

Query executed successfully. NFKRON\SQL SERVER2025 (16.0 ... sa (79) Chinook 00:00

## Ventas por genero musical:

```
~/
/* -----
2) Total de ventas por GÉNERO MUSICAL
----- */
DECLARE @From INT
DECLARE @To INT
SELECT
    dt.GenreName,
    SUM(fs.Quantity) AS Units,
    CAST(SUM(fs.LineTotal) AS NUMERIC(14,2)) AS Amount
FROM dw.FactSales fs
JOIN dw.DimTrack dt ON dt.TrackKey = fs.TrackKey
GROUP BY dt.GenreName;

/* Uso:
-- SELECT * FROM dw.vw_TotalVentasPorGenero ORDER BY Amount DESC;
-- Con filtro de periodo:
-- SELECT dt.GenreName, SUM(fs.LineTotal) AS Amount
-- FROM dw.FactSales fs
-- JOIN dw.DimTrack dt ON dt.TrackKey = fs.TrackKey
-- JOIN dw.DimDate d ON d.DateKey = fs.DateKey
-- WHERE d.[Year] = 2010
```

0 %

Results Messages

	GenreName	Units	Amount
1	Pop	28	27.72
2	Hip Hop/Rap	17	16.83
3	Bossa Nova	15	14.85
4	World	13	12.87
5	Comedy	9	17.91
6	Rock	835	826.65
7	Drama	29	57.71
8	Easy Listening	10	9.90
9	R&B/Soul	41	40.59
10	Alternative	14	13.86
11	Alternative & Punk	244	241.56
12	Heavy Metal	12	11.88
13	Jazz	80	79.20
14	Science Fiction	6	11.94
15	TV Shows	47	93.53
16	Sci Fi & Fantasy	20	39.80
17	Rock And Roll	6	5.94
18	Reggae	30	29.70
19	Latin	386	382.14
20	Electronica/Dance	12	11.88
21	Classical	41	40.59
22	Metal	264	261.36
23	Soundtrack	20	19.80
24	Blues	61	60.39



## Ventas por artista:

```
-- GROUP BY dt.genrename
-- ORDER BY Amount DESC;
*/

/* -----
3) Total de ventas por ARTISTA
----- */
DECLARE @From INT
DECLARE @To INT
SELECT
    dt.ArtistName,
    SUM(fs.Quantity) AS Units,
    CAST(SUM(fs.LineTotal) AS NUMERIC(14,2)) AS Amount
FROM dw.FactSales fs
JOIN dw.DimTrack dt ON dt.TrackKey = fs.TrackKey
GROUP BY dt.ArtistName;

/* Uso:
-- SELECT * FROM dw.vw_TotalVentasPorArtista ORDER BY Amount DESC;
-- Por año:
-- SELECT dt.ArtistName, SUM(fs.LineTotal) AS Amount
-- FROM dw.FactSales fs
-- JOIN dw.DimTrack dt ON dt.TrackKey = fs.TrackKey
-- JOIN dw.DimDate d ON d.DateKey = fs.DateKey
-- WHERE d.[Year] = 2010
-- GROUP BY dt.ArtistName
-- ORDER BY Amount DESC;
*/

/* -----
4) Total de ventas por PAÍS
(a) País del Cliente
(b) País de Facturación (dirección en la factura)
----- */
DECLARE @From INT
DECLARE @To INT
SELECT
    dc.Country,
    SUM(fs.Quantity) AS Units,
```

90 %

Results Messages

	ArtistName	Units	Amount
1	Calexico	1	0.99
2	Marillion	5	4.95
3	Raul Seixas	10	9.90
4	Marcos Valle	7	6.93
5	Stone Temple Pilots	10	9.90
6	English Concert & Trevor Pinnock	1	0.99
7	Aerosmith	10	9.90
8	João Suplicy	6	5.94
9	Aquaman	2	3.98
10	Alanis Morissette	8	7.92
11	Lost	41	81.59
12	Motörhead	6	5.94
13	Alice In Chains	7	6.93
14	BackBeat	6	5.94
15	David Coverdale	8	7.92
16	Scholars Baroque Ensemble	1	0.99
17	Jota Quest	6	5.94
18	Falamansa	8	7.92
19	Page & Plant	8	7.92
20	James Brown	12	11.88
21	Os Paralamas Do Sucesso	45	44.55
22	The Police	7	6.93
23	Battlestar Galactica	12	23.88
24	The Tea Party	17	16.83
25	Gonzaguinha	9	8.91
26	Guns N' Roses	36	35.64
27	Yo-Yo Ma	1	0.99
28	Dread Zeppelin	1	0.99
29	Pepper Marinton - London Classical Blusom	1	0.99

Query executed successfully. NEKRON\SQLSERVER2025 (16)

## Ventas por país:

3) Consultas De Prue...5.Chinook (sa (79))			
<pre> /* ----- 4) Total de ventas por PAÍS (a) País del Cliente (b) País de Facturación (dirección en la factura) -----*/  DECLARE @From INT DECLARE @To INT SELECT     dc.Country,     SUM(fs.Quantity) AS Units,     CAST(SUM(fs.LineTotal) AS NUMERIC(14,2)) AS Amount FROM dw.FactSales fs JOIN dw.DimCustomer dc ON dc.CustomerKey = fs.CustomerKey GROUP BY dc.Country;  DECLARE @From INT DECLARE @To INT SELECT     db1.BillingCountry AS Country,     SUM(fs.Quantity) AS Units,     CAST(SUM(fs.LineTotal) AS NUMERIC(14,2)) AS Amount FROM dw.FactSales fs JOIN dw.DimBillingLocation db1 ON db1.BillingLocationKey = fs.BillingLocationKey GROUP BY db1.BillingCountry;  /* Uso: -- SELECT * FROM dw.vw_TotalVentasPorPais_Cliente ORDER BY Amount DESC; -- SELECT * FROM dw.vw_TotalVentasPorPais_Facturacion ORDER BY Amount DESC; -- Filtro por rango de fechas: -- SELECT Country, SUM(fs.LineTotal) AS Amount -- FROM dw.FactSales fs -- JOIN dw.DimBillingLocation db1 ON db1.BillingLocationKey = fs.BillingLocationKey -- JOIN dw.DimDate d ON d.DateKey = fs.DateKey -- WHERE d.[Date] BETWEEN '2009-01-01' AND '2010-12-31' -- GROUP BY Country -- ORDER BY Amount DESC; */ </pre>			
90 %			
Results Messages			
	Country	Units	Amount
1	ARGENTINA	38	37.62
2	AUSTRALIA	38	37.62
3	AUSTRIA	38	42.62
4	BELGIUM	38	37.62
5	BRAZIL	190	190.10
6	CANADA	304	303.96
7	CHILE	38	46.62
8	CZECH REPUBLIC	76	90.24
9	DENMARK	38	37.62
10	FINLAND	38	41.62
11	FRANCE	190	195.10
12	GERMANY	152	156.48
13	HUNGARY	38	45.62
14	INDIA	74	75.26
15	IRELAND	38	45.62
16	ITALY	38	37.62
17	NETHERLANDS	38	40.62
18	NORWAY	38	39.62
19	POLAND	38	37.62
20	PORTUGAL	76	77.24
21	SPAIN	38	37.62
22	SWEDEN	38	38.62
23	UNITED KINGDOM	114	112.86
24	USA	494	523.06