

GESTIÓN DE BASE DE DATOS

Profesor: Luis Cuevas Parra
Carrera : Ingeniería En Información y Control De Gestión
Código : ICG-032



UST
UNIVERSIDAD SANTO TOMÁS

Creación BD y Tablas

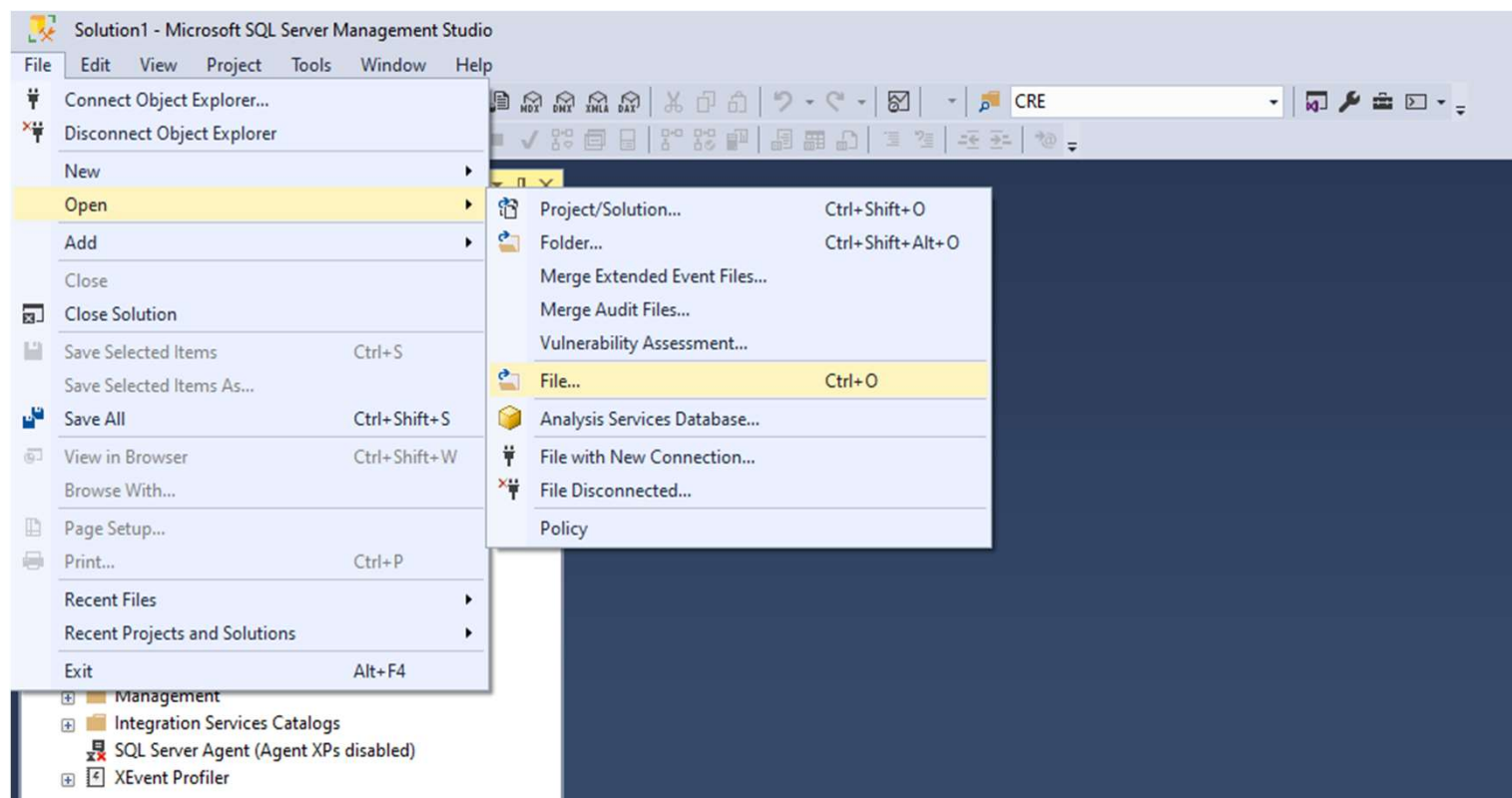
archivo .sql



UST
UNIVERSIDAD SANTO TOMÁS

SQL-Server

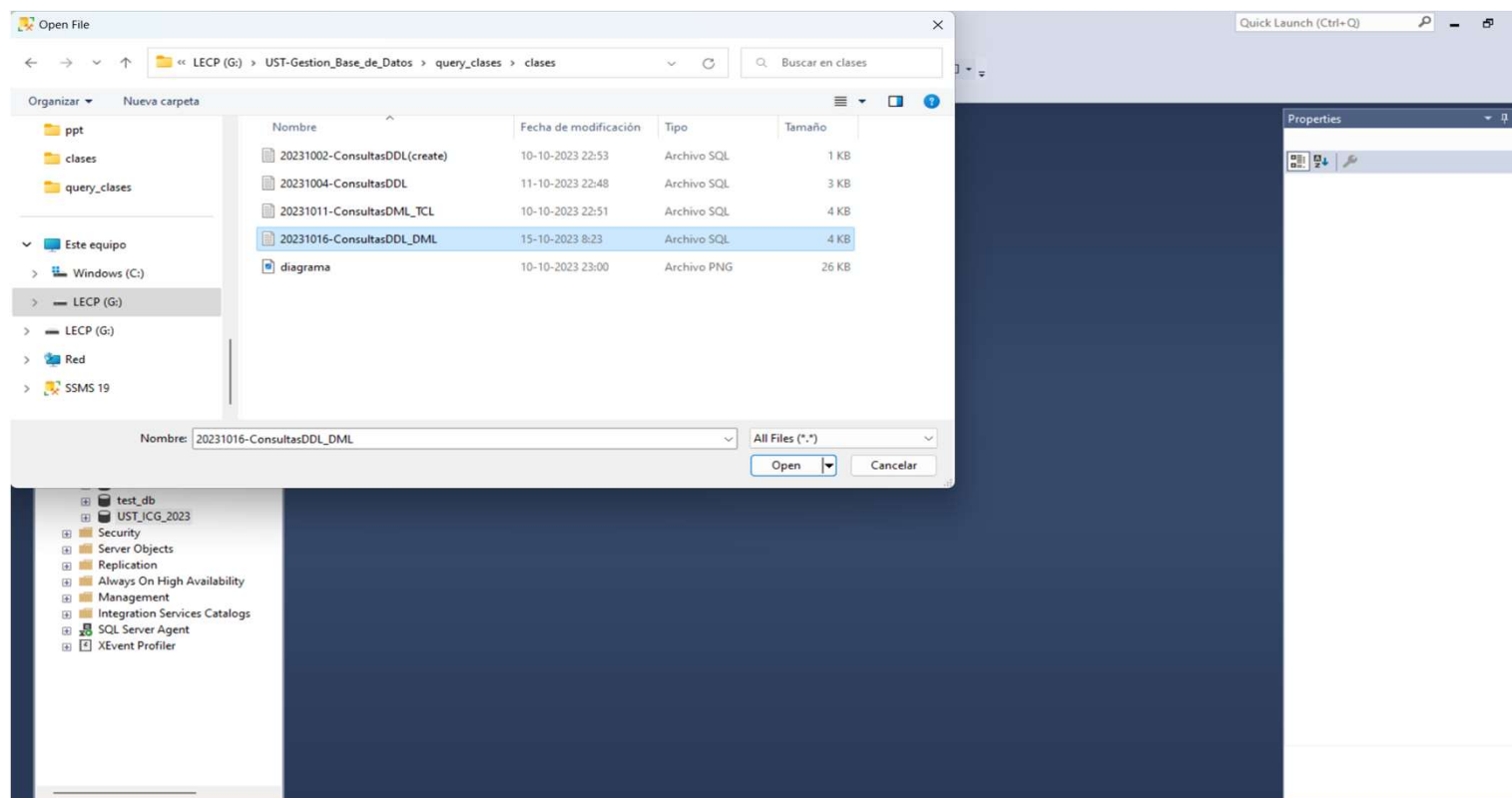
.sql





UST
UNIVERSIDAD SANTO TOMÁS

SQL-Server .sql



SQL-Server .sql



UST
UNIVERSIDAD SANTO TOMÁS

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The main window shows a SQL script for creating a database and tables. The script is as follows:

```
1  -- COMANDOS DDL
2
3  -- 1. CREATE
4
5  -- DATABASE
6
7  CREATE DATABASE UST_ICG_2023OCT;
8
9  GO
10
11  -- USE UST_ICG_2023OCT;
12
13  -- TABLE
14
15  CREATE TABLE ESTUDIANTES(
16      COD_ESTUDIANTES INT NOT NULL,
17      NOMBRE_ESTUDIANTE VARCHAR(30) NOT NULL,
18      APELLIDO_1 VARCHAR(30) NOT NULL,
19      APELLIDO_2 VARCHAR(30) NOT NULL);
20
21  -- CARRERAS
22
23  CREATE TABLE CARRERAS(
24      COD_CARRERA INT NOT NULL,
25      NOMBRE_CARRERA VARCHAR(30) NOT NULL,
26      APELLIDO_1 VARCHAR(30) NOT NULL);
27
28  -- CATEDRAS
29
30  CREATE TABLE CATEDRAS(
31      COD_CATEDRA VARCHAR(10) NOT NULL,
32      NOMBRE_CATEDRA VARCHAR(30) NOT NULL);
33
34  -- NOTAS
35
36  CREATE TABLE NOTAS(
37      COD_ESTUDIANTES INT NOT NULL,
38      COD_CARRERA INT NOT NULL,
39      COD_CATEDRA VARCHAR(10) NOT NULL,
40      NOTAS FLOAT NULL);
41
42  -- 2. RENAME
43
44  EXEC sp_rename 'ESTUDIANTES', 'ESTUDIANTES';
45
46  GO
47
48  -- COLUMNS
49
50  EXEC sp_rename 'ESTUDIANTES.COD_ESTUDIANTES', 'COD_ESTUDIANTE', 'COLUMN';
51
52  GO
```

The left pane shows the Object Explorer with the database structure. The right pane shows the Properties window for the connection PF453BJL (CONTRALORIA\Icuevasp (70)).

Properties

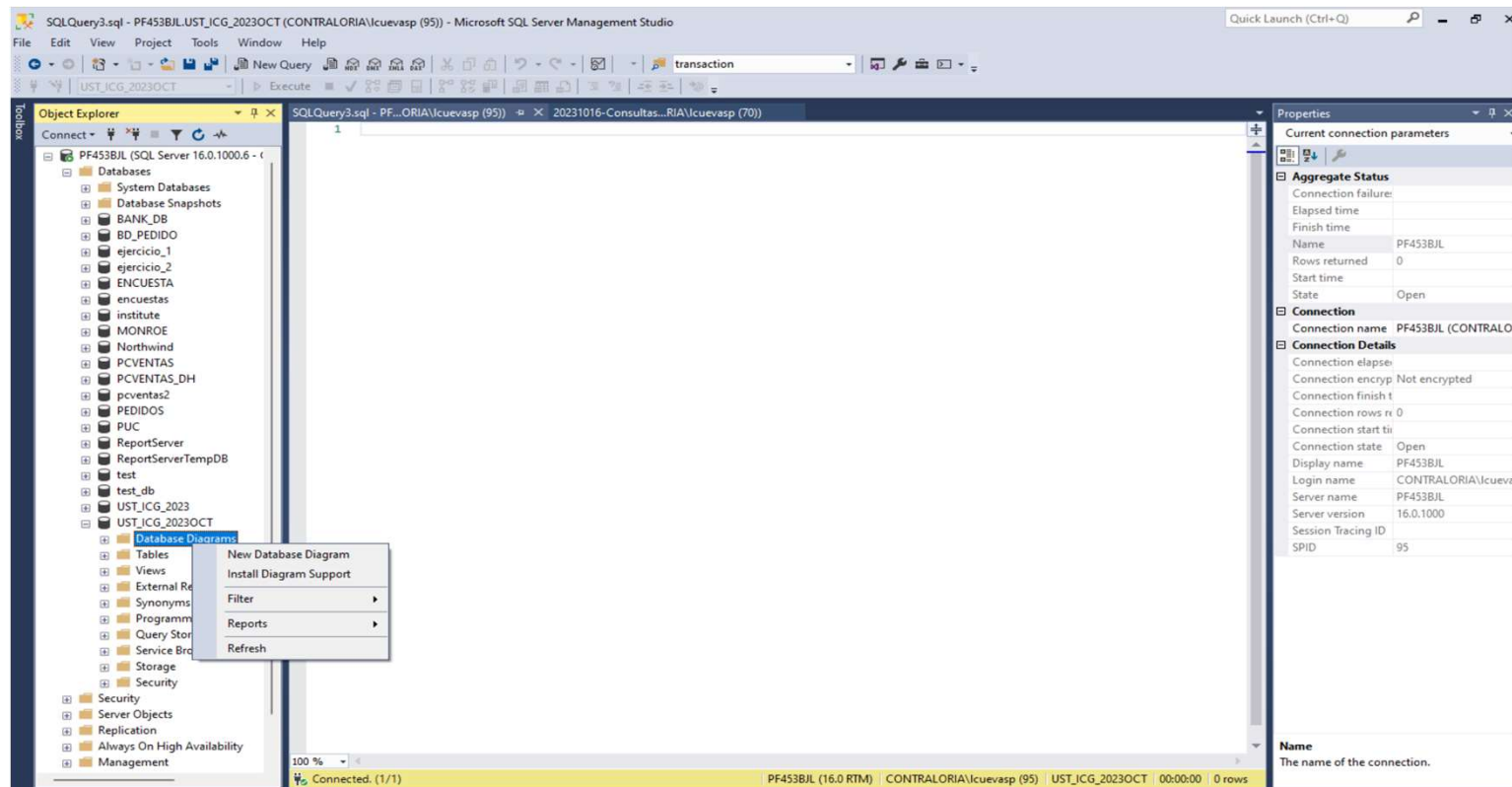
- Current connection parameters
- Aggregate Status
 - Connection failure:
 - Elapsed time
 - Finish time
 - Name: PF453BJL
 - Rows returned: 0
 - Start time
 - State: Open
- Connection
 - Connection name: PF453BJL (CONTRALORIA\Icuevasp (70))
- Connection Details
 - Connection elapsed:
 - Connection encrypt: Not encrypted
 - Connection finish t
 - Connection rows r: 0
 - Connection start t
 - Connection state: Open
 - Display name: PF453BJL
 - Login name: CONTRALORIA\Icuevasp
 - Server name: PF453BJL
 - Server version: 16.0.1000
 - Session Tracing ID
 - SPID: 70
- Name
 - The name of the connection.



UST
UNIVERSIDAD SANTO TOMÁS

SQL-Server

Generar Diagrama

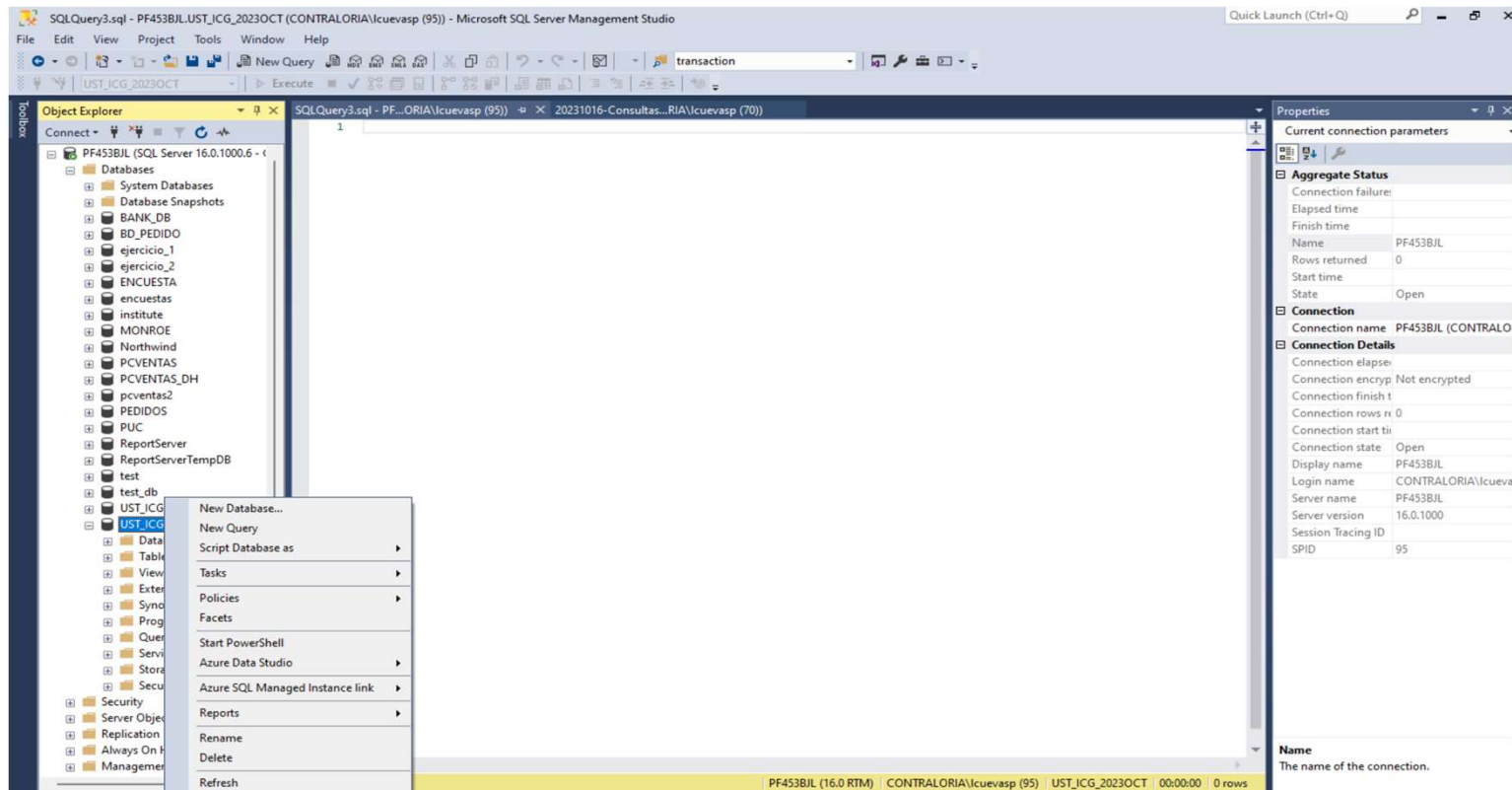




UST
UNIVERSIDAD SANTO TOMÁS

SQL-Server

Generar Diagrama





UST
UNIVERSIDAD SANTO TOMÁS

SQL-Server

Generar Diagrama

SQLQuery3.sql - PF453BJL.UST_JCG_2023OCT (CONTRALORIA\Icuevoasp (95)) - Microsoft SQL Server Management Studio

Object Explorer: PF453BJL (SQL Server 16.0.1000.6) > Databases > UST_JCG_2023OCT > Database Diagrams

Database Properties - UST_JCG_2023OCT

Select a page: General, Files, Filegroups, Options, Change Tracking, Permissions, Extended Properties, Mirroring, Transaction Log Shipping, Query Store

Database name: UST_JCG_2023OCT
Owner: sa

Use full-text indexing

Logical Name	File Type	Filegroup	Size (MB)	Autogrowth / Maxsize	Path
UST_JCG_...	ROWS...	PRIMARY	8	By 64 MB, Unlimited	C:\N...
UST_JCG_...	LOG	Not Applicable	8	By 64 MB, Limited to 2...	C:\N...

Connection: Server: PF453BJL, Connection: CONTRALORIA\Icuevoasp, View connection properties

Progress: Ready

Properties: Current connection parameters, Aggregate Status, Connection, Connection Details

Name: The name of the connection.

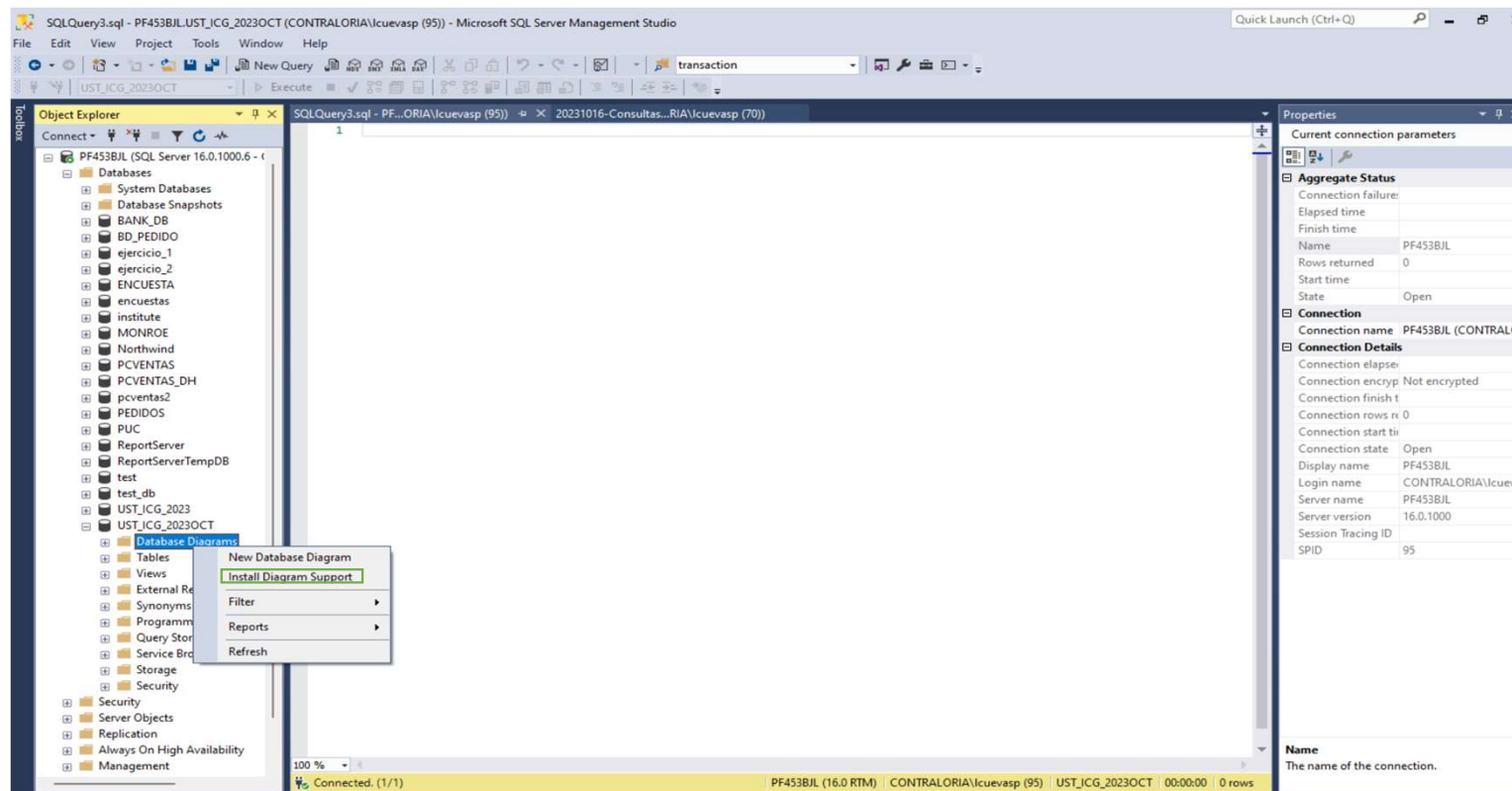
Status bar: 100 % Connected. (1/1) PF453BJL (16.0 RTM) CONTRALORIA\Icuevoasp (95) UST_JCG_2023OCT 00:00:00 0 rows



UST
UNIVERSIDAD SANTO TOMÁS

SQL-Server

Generar Diagrama

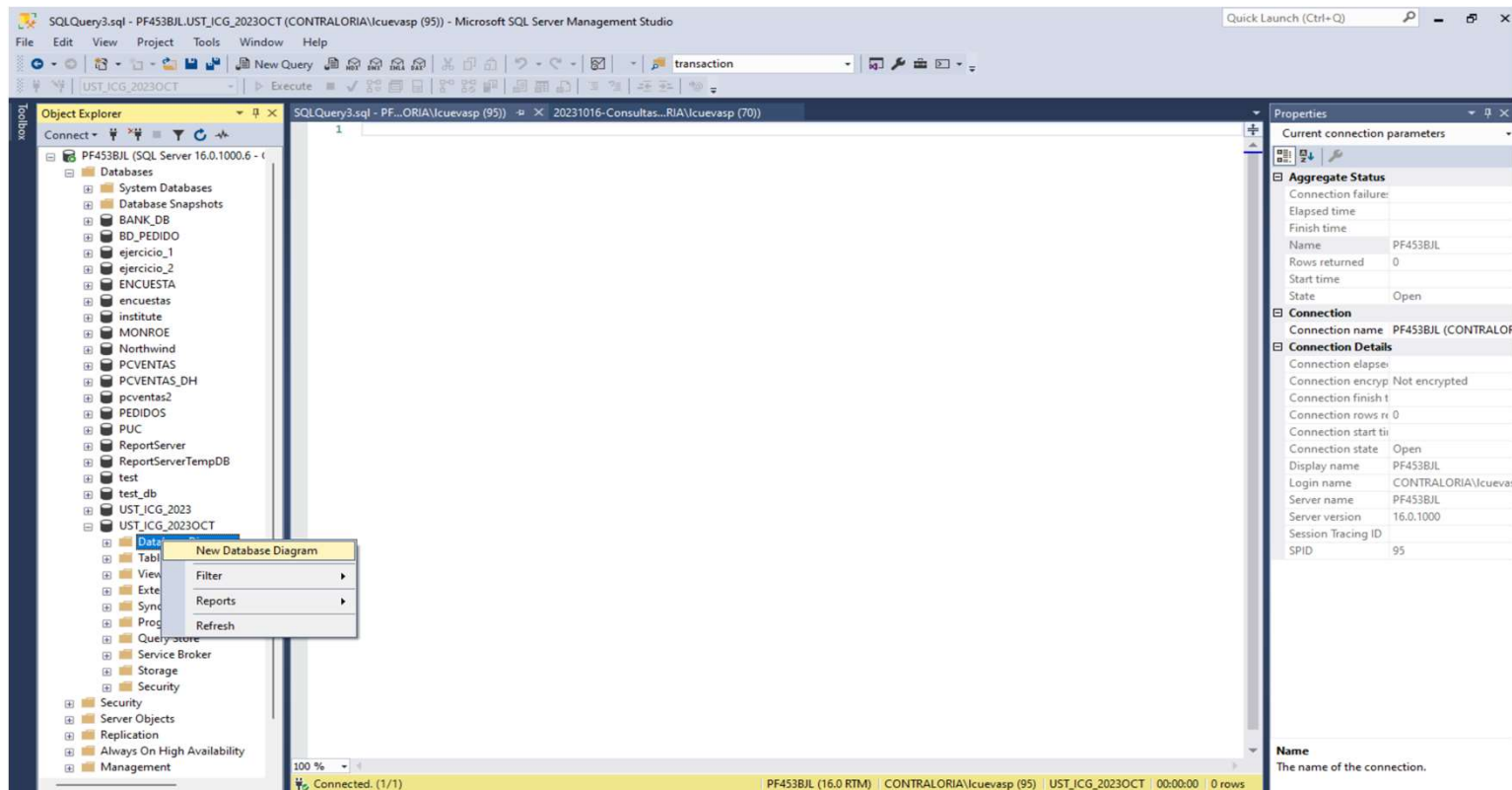




UST
UNIVERSIDAD SANTO TOMÁS

SQL-Server

Generar Diagrama

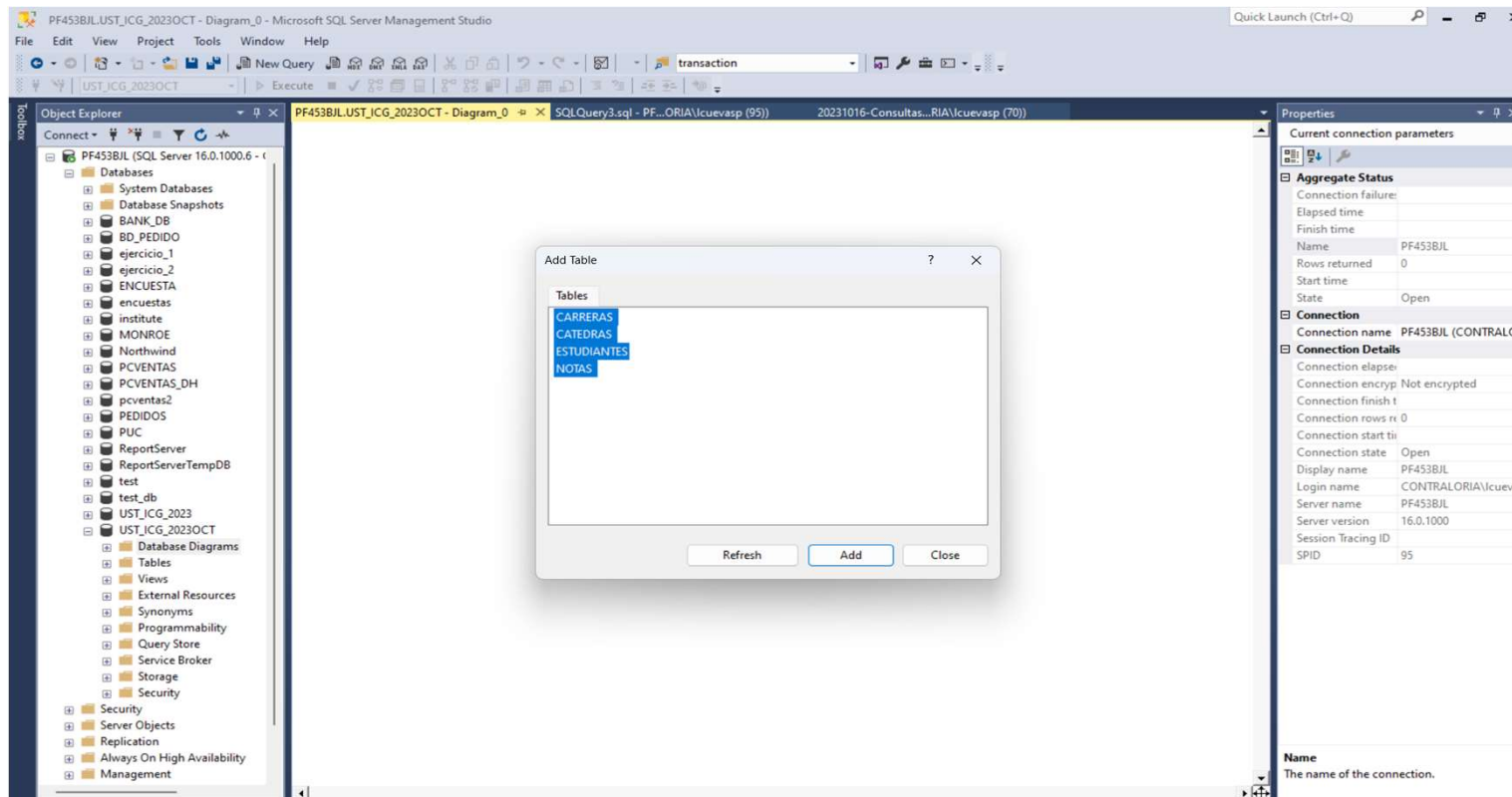




UST
UNIVERSIDAD SANTO TOMÁS

SQL-Server

Generar Diagrama

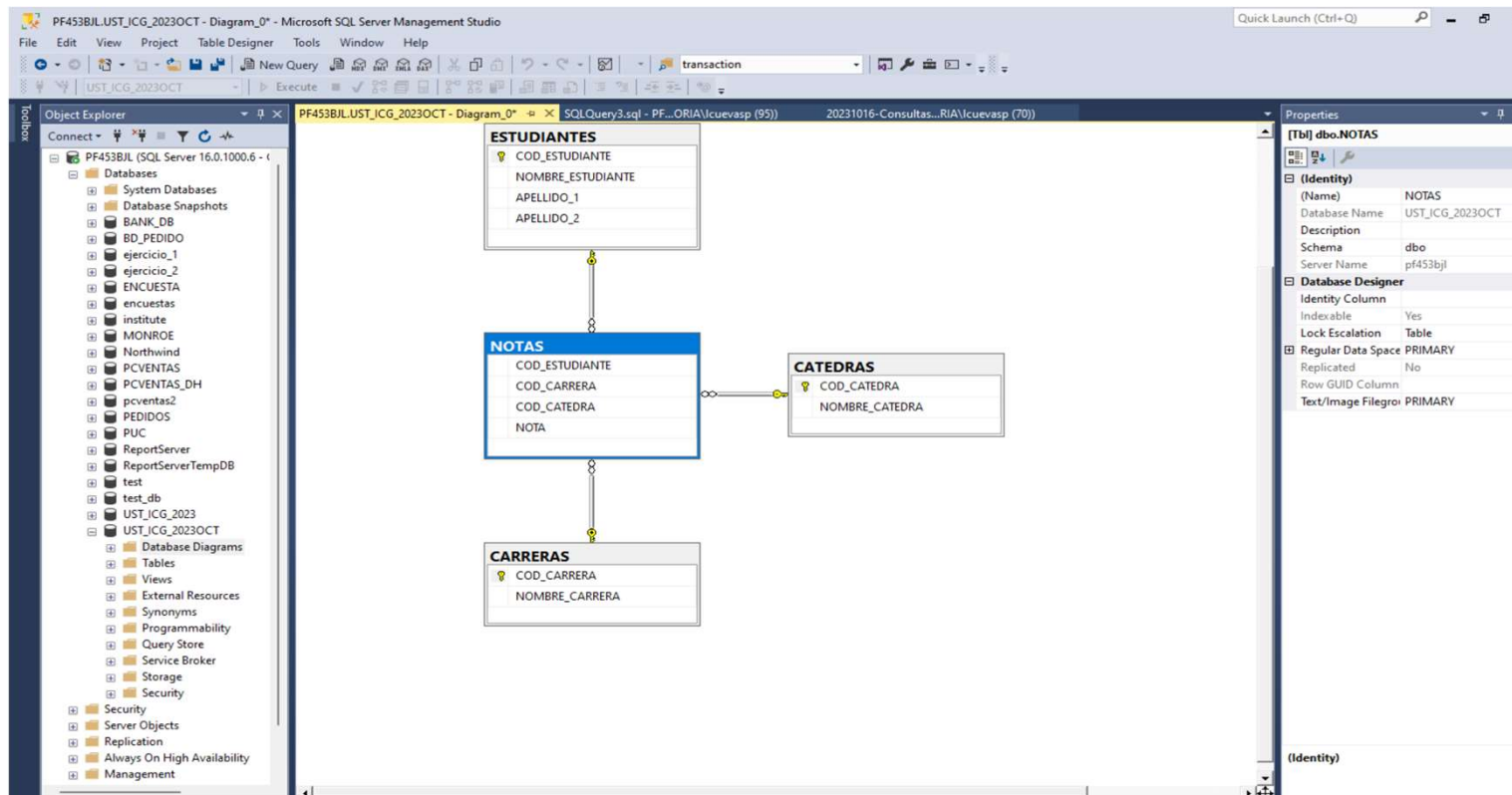




UST
UNIVERSIDAD SANTO TOMÁS

SQL-Server

Generar Diagrama





UST
UNIVERSIDAD SANTO TOMÁS

Consulta básica en SQL

Preguntas claves

- ¿Qué datos nos están pidiendo? → Campo/columna
- ¿Dónde están esos datos? → Tabla
- ¿Qué requisitos deben cumplir los registros? → Condiciones



UST
UNIVERSIDAD SANTO TOMÁS

Operaciones DML

Operadores Básicos

-- ESTRUCTURA BÁSICA EN CONSULTAS SQL

SELECT	→ 2° CAMPOS A SELECCIONAR
FROM	→ 1° TABLA A SELECCIONAR
WHERE	→ 3° CONDICIÓN DE FILTRADO



UST
UNIVERSIDAD SANTO TOMÁS

Operaciones DML

SELECT

```
-- CONOCER TABLAS DE LA BASE DE DATOS
```

```
SELECT *
```

```
FROM sys.tables
```

```
SELECT *
```

```
FROM sys.schemas
```

```
SELECT t.name
```

```
FROM sys.tables t
```

```
INNER JOIN sys.schemas s
```

```
ON t.schema_id = s.schema_id
```

```
ORDER BY t.name;
```



UST
UNIVERSIDAD SANTO TOMÁS

Operaciones DML

SELECT

-- CONSULTAR TABLE

```
SELECT COL1,...,COLN } PARA SELECCIONAR TODAS LOS CAMPOS SE COLOCA *  
FROM NOMBRE_TABLA;  } INDICAR ORIGEN DE DATOS
```

-- CONSULTAR TABLE

```
SELECT *  
FROM NOTAS;
```




UST
UNIVERSIDAD SANTO TOMÁS

Operaciones DML

SELECT

-- CONSULTAR TABLA

```
SELECT COL1,...,COLN } PARA SELECCIONAR TODAS LOS CAMPOS SE COLOCA *  
FROM NOMBRE_TABLA;  } INDICAR ORIGEN DE DATOS  
WHERE CONDICION_1;   } INDICAR CONDICIÓN
```

-- CONSULTAR TABLA NOTAS MAYORES A 6

```
SELECT *  
FROM NOTAS  
WHERE NOTA >= 6;
```



UST
UNIVERSIDAD SANTO TOMÁS

Operadores

Booleanos

- Permite establecer condiciones (expresiones) por medio de la cláusula **WHERE**. Sirve para **filtrar** registros de las tablas.
- El motor de SQL evalúa la expresión BOOLEANA de la cláusula **WHERE** para cada registro de la tabla
- El **resultado** de la expresión puede ser **VERDADERO** o **FALSO**.
- Si el resultado es **VERDADERO** la consulta **arrojará resultados**. En caso contrario, ignorará el resultado



UST
UNIVERSIDAD SANTO TOMÁS

Operadores Booleanos

Expresión	Interpretación
>	$A > B$ Devuelve cierto si A es estrictamente mayor que B (muestra el resultado), de lo contrario devuelve falso (no muestra el resultado)
<	$A < B$ Devuelve cierto si A es estrictamente menor que B (muestra el resultado), de lo contrario devuelve falso (no muestra el resultado)
=	$A = B$ Devuelve cierto si A es igual que B (muestra el resultado), de lo contrario devuelve falso (no muestra el resultado)
>=	$A \geq B$ Devuelve cierto si A es mayor o igual que B (muestra el resultado), de lo contrario devuelve falso (no muestra el resultado)
<=	$A \leq B$ Devuelve cierto si A es menor o igual que B (muestra el resultado), de lo contrario devuelve falso (no muestra el resultado)
!=	$A \neq B$ Devuelve cierto si A es distinto que B (muestra el resultado), de lo contrario devuelve falso (no muestra el resultado). Puede usar la notación $A < > B$



UST
UNIVERSIDAD SANTO TOMÁS

Operadores Booleanos

-- CONSULTAR TABLE

SELECT COL1,...,COLN	}	PARA SELECCIONAR TODAS LOS CAMPOS SE COLOCA *
FROM NOMBRE_TABLA;	}	INDICAR ORIGEN DE DATOS
WHERE COND_BOOLEANA;	}	INDICAR CONDICIÓN

-- CONSULTAR TABLA NOTAS MAYORES A 6

```
SELECT *  
FROM NOTAS  
WHERE NOTA >= 6;
```



UST
UNIVERSIDAD SANTO TOMÁS

Operadores Booleanos

```
-- CONSULTAR TABLA NOTAS DISTINTAS A 6  
SELECT *  
FROM NOTAS  
WHERE NOTA != 6;
```

```
-- CONSULTAR TABLA NOTAS IGUALES A 6.5  
SELECT *  
FROM NOTAS  
WHERE NOTA = 6.5;
```



Operadores Lógicos

- Ayudan a formar expresiones booleanas tomando como operando otras expresiones booleanas.

Operator	Significado
<u>ALL</u>	VERDADERO si el conjunto completo de comparaciones es VERDADERO.
<u>AND</u>	VERDADERO si ambas expresiones booleanas son VERDADERO.
<u>ANY</u>	VERDADERO si cualquier miembro del conjunto de comparaciones es VERDADERO.
<u>BETWEEN</u>	VERDADERO si el operando está dentro de un intervalo.
<u>EXISTS</u>	VERDADERO si una subconsulta contiene cualquiera de las filas.
<u>IN</u>	VERDADERO si el operando es igual a uno de la lista de expresiones.
<u>LIKE</u>	VERDADERO si el operando coincide con un patrón.
<u>NOT</u>	Invierte el valor de cualquier otro operador booleano.
<u>OR</u>	VERDADERO si cualquiera de las dos expresiones booleanas es VERDADERO.
<u>SOME</u>	VERDADERO si alguna de las comparaciones de un conjunto es VERDADERO.



UST
UNIVERSIDAD SANTO TOMÁS

Operadores Lógicos

-- CONSULTAR TABLE

SELECT COL1,...,COLN	}	PARA SELECCIONAR TODAS LOS CAMPOS SE COLOCA *
FROM NOMBRE_TABLA;	}	INDICAR ORIGEN DE DATOS
WHERE COND_BOOLEANA	}	INDICAR CONDICIÓN
OPERADOR_LÓGICO	}	AND/OR/NOT
COND_BOOLEANA_2;	}	INDICAR CONDICIÓN

-- AND: CONSULTAR TABLA NOTAS DE ESTUDIANTE 100 CON NOTAS MAYORES A 6

```
SELECT *  
FROM NOTAS  
WHERE COD_ESTUDIANTE = 100;  
AND NOTA > 6;
```



UST
UNIVERSIDAD SANTO TOMÁS

Operadores Lógicos

```
-- OR: CONSULTAR TABLA NOTAS.NOTAS MAYORES A 6 O MENORES A 5.5  
SELECT *  
FROM NOTAS  
WHERE NOTA > 6  
OR NOTA < 5.5;
```

```
-- NOT: NOTAS DE COD_ESTUDIANTE = 100 Y QUE NO SEAN DE ING-103  
SELECT *  
FROM NOTAS  
WHERE COD_ESTUDIANTE = 100  
AND NOT COD_CATEDRA = 'ING-103';
```




UST
UNIVERSIDAD SANTO TOMÁS

Operadores Lógicos

- **BETWEEN**: Especifica un intervalo que se va a probar.

```
-- BETWEEN: COD_ESTUDIANTE ENTRE ENTRE 100 Y 101;  
SELECT *  
FROM NOTAS  
WHERE COD_ESTUDIANTE BETWEEN 100 AND 101;
```



Operadores Lógicos

- **LIKE**: Determina si una cadena de caracteres coincide con un patrón especificado

Carácter comodín	Descripción
%	Cualquier cadena de cero o más caracteres.
_ (guió bajo)	Cualquier carácter individual.
[]	Cualquier carácter individual que se encuentre en el intervalo [a-f] o el conjunto [abcdef] que se haya especificado.
[^]	Cualquier carácter individual que no se encuentre en el intervalo [^a-f] o el conjunto [^abcdef] que se haya especificado.



UST
UNIVERSIDAD SANTO TOMÁS

Operadores Lógicos

-- que inicie con la letra A

SELECT *

FROM ESTUDIANTES

WHERE NOMBRE_ESTUDIANTE like 'A%';

-- que inicie con la letra A,B,C,D

SELECT *

FROM ESTUDIANTES

WHERE NOMBRE_ESTUDIANTE like '[A-D]%';



UST
UNIVERSIDAD SANTO TOMÁS

Operadores Lógicos

```
-- que contenga la palabra GE  
SELECT *  
FROM ESTUDIANTES  
WHERE NOMBRE_ESTUDIANTE like '%GE%'
```

```
-- que termine con la letra A  
SELECT *  
FROM ESTUDIANTES  
WHERE NOMBRE_ESTUDIANTE like '%A'
```



UST
UNIVERSIDAD SANTO TOMÁS

Operadores Lógicos

```
-- que termine con la letra A,B,C,D  
SELECT *  
FROM ESTUDIANTES  
WHERE NOMBRE_ESTUDIANTE like '%[A-D]';
```

```
-- que termine con la palabra ID  
SELECT *  
FROM ESTUDIANTES  
WHERE NOMBRE_ESTUDIANTE like '%ID';
```



UST
UNIVERSIDAD SANTO TOMÁS

Operadores Lógicos

-- que comience con la palabra AN y continúe con cualquier carácter

```
SELECT *  
FROM ESTUDIANTES  
WHERE NOMBRE_ESTUDIANTE like 'AN_'
```

-- que comience con la palabra AN y que la letra siguiente no sea A (Ej: ANA)

```
SELECT *  
FROM ESTUDIANTES  
WHERE NOMBRE_ESTUDIANTE like 'AN[^A]'
```



UST
UNIVERSIDAD SANTO TOMÁS

Operadores Lógicos

- **IN**: Determina si un valor especificado coincide con algún valor de una subconsulta o una lista.

```
-- REGISTROS CUYAS NOTAS SEAN 5.6 Y 6.1
```

```
SELECT *
```

```
FROM NOTAS
```

```
WHERE NOTA IN (5.6, 6.1)
```

```
-- REGISTROS CUYAS NOTAS NO SEAN 5.6 Y 6.1
```

```
SELECT *
```

```
FROM NOTAS
```

```
WHERE NOT NOTA IN (5.6, 6.1)
```



UST
UNIVERSIDAD SANTO TOMÁS

Operadores Lógicos

-- que comience con la palabra AN y continúe con cualquier carácter

```
SELECT *  
FROM ESTUDIANTES  
WHERE NOMBRE_ESTUDIANTE like 'AN_'
```

-- que comience con la palabra AN y que la letra siguiente no sea A (Ej: ANA)

```
SELECT *  
FROM ESTUDIANTES  
WHERE NOMBRE_ESTUDIANTE like 'AN[^A]'
```




UST
UNIVERSIDAD SANTO TOMÁS

Funciones de Agregación

Definición

- Una función de agregado realiza un cálculo sobre un conjunto de valores y devuelve un solo valor.
- Las funciones de agregado ignoran los valores NULL (Con la excepción de COUNT(*)).
- Las funciones de agregado se suelen usar con la cláusula GROUP BY de la instrucción SELECT.



UST
UNIVERSIDAD SANTO TOMÁS

Funciones de Agregación

Transact-SQL: funciones de agregado

- APPROX COUNT DISTINCT
- AVG
- CHECKSUM AGG
- COUNT
- COUNT BIG
- GROUPING
- GROUPING_ID
- MAX
- MIN
- STDEV
- STDEVP
- STRING AGG
- SUM
- VAR
- VARP



UST
UNIVERSIDAD SANTO TOMÁS

Funciones de Agregación

Funciones más utilizadas

- **AVG**: Devuelve el **promedio** de los valores en un campo (columna).
- **COUNT**: Cuenta la **cantidad de valores presentes** (valores no nulos) en un campo (columna).
- **SUM**: Retorna la **suma** de los valores de un determinado campo.
- **MAX**: Retorna el valor **máximo** de un determinado campo.
- **MIN**: Retorna el valor **mínimo** de un determinado campo.
- **STDEV**: Retorna la **desviación standard** de un determinado campo numérico ($N > 1$).



UST
UNIVERSIDAD SANTO TOMÁS

Funciones de Agregación

Sin agrupación por campo

-- AVG

```
SELECT AVG(NOTA) AS NOTAS  
FROM NOTAS
```

```
SELECT ROUND(AVG(NOTA),2) AS NOTAS  
FROM NOTAS
```

-- COUNT

```
SELECT COUNT(NOTA) AS CANTIDAD_NOTAS  
FROM NOTAS
```



UST
UNIVERSIDAD SANTO TOMÁS

Funciones de Agregación

Sin agrupación por campo

```
-- SUM  
SELECT SUM(NOTA) AS SUMATORIA_NOTAS  
FROM NOTAS;
```

```
-- MAX  
SELECT MAX(NOTA) AS NOTA_MAXIMA  
FROM NOTAS;
```



UST
UNIVERSIDAD SANTO TOMÁS

Funciones de Agregación

Sin agrupación por campo

```
-- MIN
SELECT MIN(NOTA) AS NOTA_MINIMA
FROM NOTAS;

-- SD
SELECT STDEV(NOTA) AS DEV_STANDARD
FROM NOTAS;
```



UST
UNIVERSIDAD SANTO TOMÁS

Agrupación de Datos

SELECT: GROUP BY con Funciones de Agregación

- Función de agregado
- ¿Cómo debemos agrupar los datos?
- Divide el resultado de la consulta en grupos de filas, normalmente realizando una o varias agregaciones en cada grupo. La instrucción SELECT devuelve una fila por grupo.
- Sirve para obtener totales usando las funciones de agregación, tales como SUM, MAX, MIN, ect.
- Se puede omitir en la cláusula SELECT el campo a agrupar, sin embargo, es útil indicarlo para saber a qué valor corresponde la agregación realizada.



UST
UNIVERSIDAD SANTO TOMÁS

Agrupación de Datos

SELECT: GROUP BY con Funciones de Agregación

-- AVG

```
SELECT COD_ESTUDIANTE, AVG(NOTA) AS NOTAS  
FROM NOTAS  
GROUP BY COD_ESTUDIANTE;
```

```
SELECT COD_ESTUDIANTE, ROUND(AVG(NOTA),2) AS NOTAS  
FROM NOTAS  
GROUP BY COD_ESTUDIANTE;
```

-- COUNT

```
SELECT COD_ESTUDIANTE, COUNT(NOTA) AS CANTIDAD_NOTAS  
FROM NOTAS  
GROUP BY COD_ESTUDIANTE;
```




UST
UNIVERSIDAD SANTO TOMÁS

Agrupación de Datos

SELECT: GROUP BY con Funciones de Agregación

```
-- SUM
SELECT COD_ESTUDIANTE, SUM(NOTA) AS SUMATORIA_NOTAS
FROM NOTAS
GROUP BY COD_ESTUDIANTE;
```

```
-- MAX
SELECT COD_ESTUDIANTE, MAX(NOTA) AS NOTA_MAXIMA
FROM NOTAS
GROUP BY COD_ESTUDIANTE;
```



UST
UNIVERSIDAD SANTO TOMÁS

Agrupación de Datos

SELECT: GROUP BY con Funciones de Agregación

-- MIN

```
SELECT COD_ESTUDIANTE, MIN(NOTA) AS NOTA_MINIMA  
FROM NOTAS  
GROUP BY COD_ESTUDIANTE;
```

-- SD

```
SELECT COD_ESTUDIANTE, STDEV(NOTA) AS DEV_STANDARD  
FROM NOTAS  
GROUP BY COD_ESTUDIANTE;
```



UST
UNIVERSIDAD SANTO TOMÁS

Agrupación de Datos

SELECT: GROUP BY con Funciones de Agregación

-- MIN

```
SELECT COD_ESTUDIANTE, MIN(NOTA) AS NOTA_MINIMA  
FROM NOTAS  
GROUP BY COD_ESTUDIANTE;
```

-- SD

```
SELECT COD_ESTUDIANTE, STDEV(NOTA) AS DEV_STANDARD  
FROM NOTAS  
GROUP BY COD_ESTUDIANTE;
```



UST
UNIVERSIDAD SANTO TOMÁS

Filtrado Cálculos de Totalización

SELECT: HAVING

- Función de filtrado a datos totalizados.
- ¿Qué requisitos debe cumplir los datos totalizados?
- HAVING solo se puede utilizar con la instrucción SELECT. Normalmente, HAVING se usa con una cláusula GROUP BY. Cuando no se usa GROUP BY, hay un solo grupo implícito agregado.
- Permite establecer filtros sobre los cálculos de una consulta SQL que realizan las funciones de agregado (SUM, COUNT)



UST
UNIVERSIDAD SANTO TOMÁS

Filtrado Cálculos de Totalización

SELECT: HAVING

```
-- SUM
SELECT COD_ESTUDIANTE, SUM(NOTA) AS SUMATORIA_NOTAS
FROM NOTAS
GROUP BY COD_ESTUDIANTE
HAVING SUM(NOTA) > 10 ;

-- COUNT
SELECT COD_ESTUDIANTE, COUNT(NOTA) AS CANTIDAD_NOTAS
FROM NOTAS
GROUP BY COD_ESTUDIANTE
HAVING COUNT(NOTA) < 2;
```



UST
UNIVERSIDAD SANTO TOMÁS

Funciones de Cadena

Transact-SQL

Las siguientes funciones escalares realizan una operación sobre un valor de cadena de entrada y devuelven un valor de cadena o un valor numérico:

ASCII

LEN

REVERSE

STRING_SPLIT

CHAR

LOWER

RIGHT

STUFF

CHARINDEX

LTRIM

RTRIM

SUBSTRING

CONCAT

NCHAR

SOUNDEX

TRANSLATE

CONCAT_WS

PATINDEX

SPACE

TRIM

DIFFERENCE

QUOTENAME

STR

UNICODE

FORMAT

REPLACE

STRING_AGG

UPPER

LEFT

REPLICATE

STRING_ESCAPE



UST
UNIVERSIDAD SANTO TOMÁS

Funciones de Cadena

Transact-SQL

```
-- LEFT: 3 primeros caracteres campo COD_CATEDRA  
SELECT COD_CATEDRA, LEFT(COD_CATEDRA, 3) AS LEFT_3  
FROM CATEDRAS;
```

```
-- RIGHT: 3 últimos caracteres campo COD_CATEDRA  
SELECT COD_CATEDRA, RIGHT(COD_CATEDRA, 3) AS RIGHT_3  
FROM CATEDRAS;
```

```
--LEN: Cantidad Caracteres campo COD_CATEDRA  
SELECT COD_CATEDRA, RIGHT(COD_CATEDRA, 3) AS RIGHT_3  
FROM CATEDRAS;
```



UST
UNIVERSIDAD SANTO TOMÁS

Funciones de Cadena

Transact-SQL

```
-- LOWER: Devuelve a minúscula campo COD_CATEDRA  
SELECT COD_CATEDRA, LOWER(COD_CATEDRA) AS COD_CATEDRA_LOWER  
FROM CATEDRAS;
```

```
-- UPPER: Devuelve a mayúscula campo COD_CATEDRA  
DECLARE @NOMBRE VARCHAR(10)  
SET @NOMBRE = 'Luis'  
SELECT @NOMBRE, UPPER(@NOMBRE) AS NOMBRE_UPPER;  
GO
```




UST
UNIVERSIDAD SANTO TOMÁS

Funciones de Cadena Transact-SQL

```
-- REPLACE: Reemplazar caracteres '-' de camp COD_CATEDRA  
SELECT COD_CATEDRA, REPLACE(COD_CATEDRA, '-', '' ) AS COD_CATEDRA_NEW  
FROM CATEDRAS;
```

```
-- LTRIM: Elimina los espacios en blanco lado izquierdo  
DECLARE @NOMBRE VARCHAR(10)  
SET @NOMBRE = 'Luis'  
SELECT @NOMBRE, UPPER(@NOMBRE) AS NOMBRE_UPPER;  
GO
```



UST
UNIVERSIDAD SANTO TOMÁS

Funciones de Cadena

Transact-SQL

-- RTRIM: Elimina los espacios en blanco lado derecho

```
DECLARE @NOMBRE VARCHAR(10)
```

```
SET @NOMBRE = 'Luis    '
```

```
SELECT @NOMBRE, RTRIM(@NOMBRE) AS NOMBRE_RTRIM;
```

```
GO
```

-- TRIM: Elimina los espacios en blanco lado derecho e izquierdo

```
DECLARE @NOMBRE VARCHAR(10)
```

```
SET @NOMBRE = '    Luis    '
```

```
SELECT @NOMBRE, TRIM(@NOMBRE) AS NOMBRE_TRIM;
```

```
GO
```



UST
UNIVERSIDAD SANTO TOMÁS

Funciones de Cadena

Transact-SQL

-- CONCAT: Une los valores de distintos campos

```
SELECT *, CONCAT(NOMBRE_ESTUDIANTE, ' ', APELLIDO_1, ' ', APELLIDO_2) AS  
[NOMBRE COMPLETO]  
FROM ESTUDIANTES;
```

-- SUBSTRING: Extrae caracteres campo NOMBRE_ESTUDIANTE, indicando inicio y largo a extraer

```
SELECT NOMBRE_ESTUDIANTE, SUBSTRING(NOMBRE_ESTUDIANTE, 2, 3)  
FROM ESTUDIANTES;
```

-- SPACE: Añade espacios a los valores de campo

```
SELECT *, CONCAT(NOMBRE_ESTUDIANTE, SPACE(1), APELLIDO_1, SPACE(1),  
, APELLIDO_2) AS [NOMBRE COMPLETO]  
FROM ESTUDIANTES;
```