UNIVERSIDAD PRIVADA FRANZ TAMAYO
DEFENSA HITO 4- TAREA FINAL Estudiante:

Estudiante: KEVIN JAVIER SANGA ORTIZ

Asignatura: BASE DE DATOS I

Carrera: INGENIERÍA DE SISTEMAS

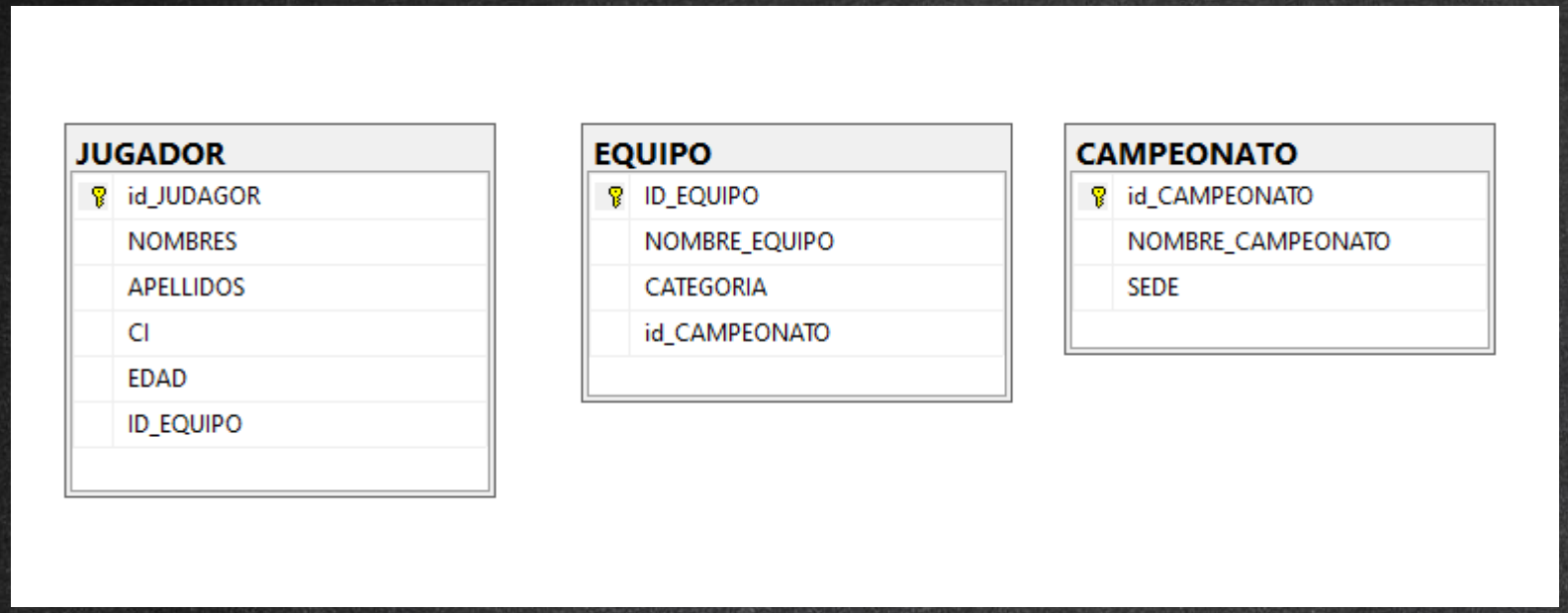
Paralelo: BDA (1)

Docente: Lic. William Barra Paredes

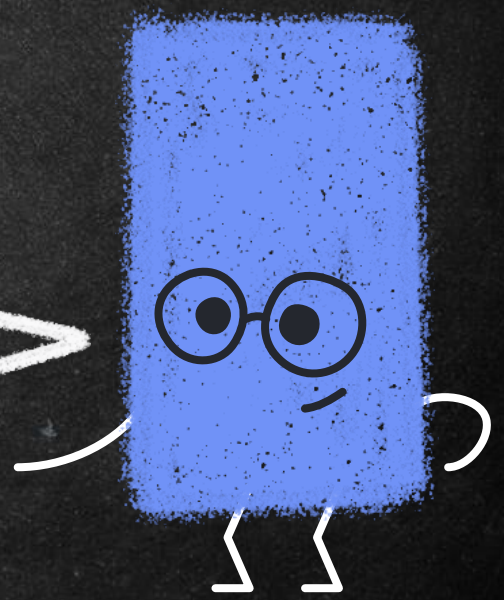
fecha: 10/24/2022

Video explicativo:

https://drive.google.com/drive/folders/1sjx_VDHqpduBp-U-gGZ22l43zR9Ph2aU?usp=sharing



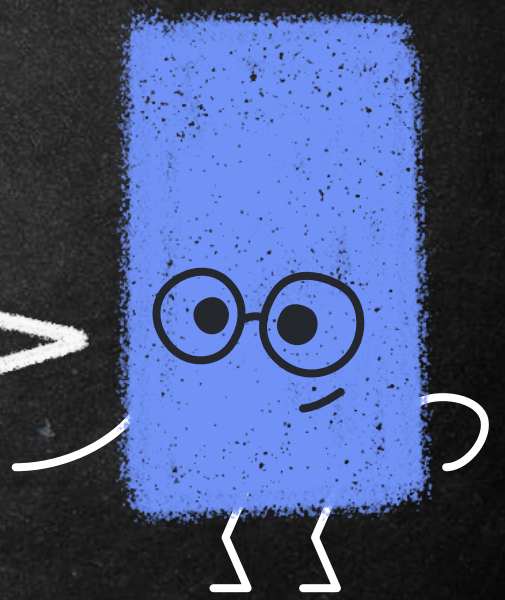
Tablas principales



	id_CAMPEONATO	NOMBRE_CAMPEONATO	SEDE
1	camp-111	Campeonato Unifranz	El Alto
2	camp-222	Campeonato Unifranz	Cochabamba

	ID_EQUIPO	NOMBRE_EQUIPO	CATEGORIA	id_CAMPEONATO
1	equ-111	Google	varones	camp-111
2	equ-222	404 Not found	varones	camp-111
3	equ-333	girls unifranz	mujeres	camp-111

	id_JUDAGOR	NOMBRES	APELLIDOS	CI	EDAD	ID_EQUIPO
1	jug-111	Carlos	Villa	8997811LP	19	equ-222
2	jug-222	Pedro	Salas	8997822LP	20	equ-222
3	jug-333	Saul	Araj	8997833LP	21	equ-222
4	jug-444	Sandra	Solis	8997844LP	20	equ-333
5	jug-555	Ana	Mica	8997855LP	23	equ-333



Datos de las tablas

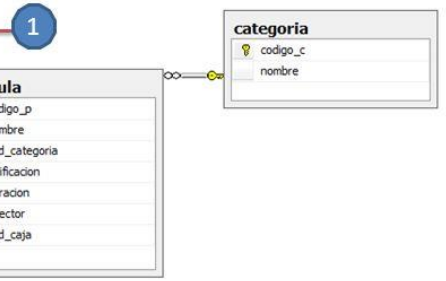
Que es DDL

- **La sentencia DDL** se utiliza para describir una base de datos, para definir su estructura, para crear sus objetos y para crear los sub objetos de la tabla

Ejemplo Create

```
use biblioteca
create table categoria(
codigo_c int identity (1,1),
nombre varchar(50),
primary key(codigo_c)
);

create table pelicula(
codigo_p varchar(20),
nombre varchar(50),
cod_categoria int,
calificacion int default '3',
duracion time,
director varchar(50),
cod_caja varchar(50),
primary key(codigo_p),
unique(cod_caja),
foreign key(cod_categoria) references  categoria(codigo_c)
on delete cascade
on update cascade
);
```

1 2 3 4 5



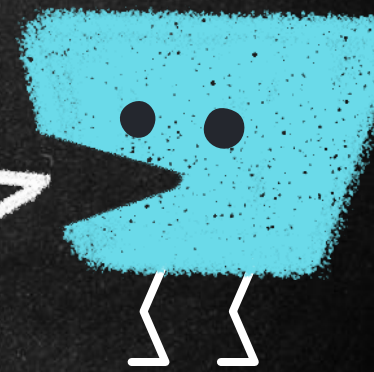
JUAN BENITEZ

Que es DML

→ **La sentencia DML** es el conjunto de instrucciones de lenguaje SQL que se utiliza para administrar datos en tablas.

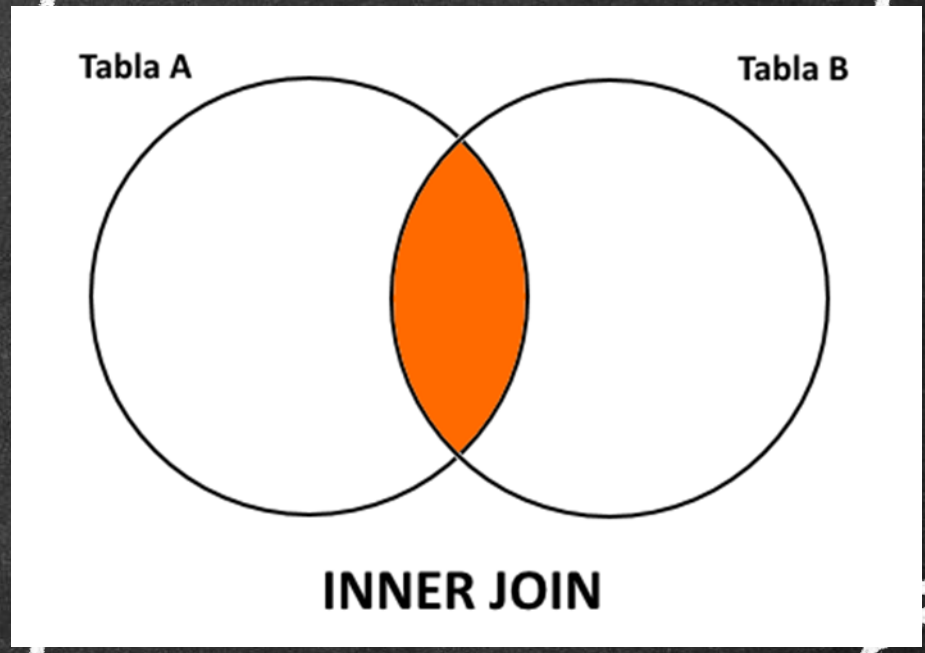
- **Las instrucciones DML se utilizan para cambiar datos o recuperar información**
  - SELECT
  - INSERT
  - UPDATE
  - DELETE
- **Deben tener los permisos adecuados**

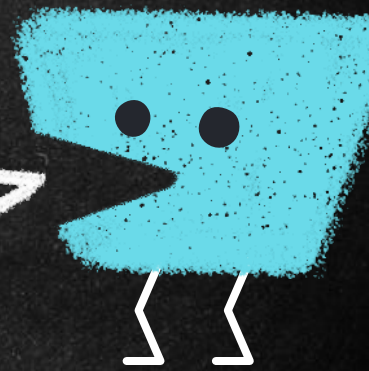
```
DELIMITER $$

DROP PROCEDURE IF EXISTS procedimiento1 $$
CREATE PROCEDURE procedimiento1 ()
BEGIN
    DECLARE i TINYINT UNSIGNED DEFAULT 1;
    DROP TABLE IF EXISTS alumnos ;
    CREATE TABLE alumnos
         (id         INT PRIMARY KEY,
          alumno VARCHAR(30))
     ENGINE=innodb;
--
    WHILE (i<=5) DO
         INSERT INTO alumnos VALUES(i,CONCAT("alumno ",i));
         SET i=i+1;
    END WHILE;

END $$

DELIMITER ;
```



Para que drive INNER JOIN.

- Combina los registros de dos tablas si hay valores coincidentes en un campo común.





Defina que es una función de agregación.

Las funciones de agregación en SQL nos permiten efectuar operaciones sobre un conjunto de resultados, pero devolviendo un único valor agregado para todos ellos.

SQLQuery2.sql - (localdb)\...\...n (53))*

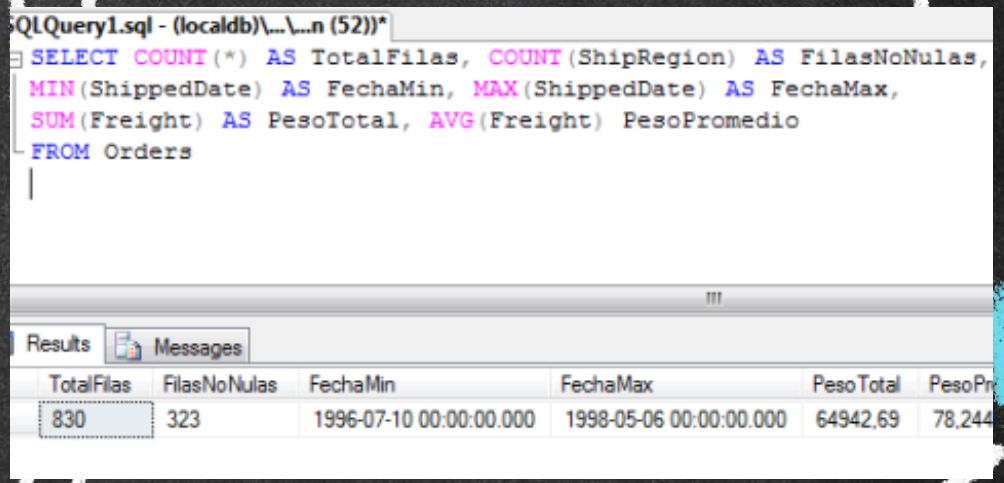
```
SELECT Employees.FirstName + ' ' + Employees.LastName AS Empleado, COUNT(*) AS TotalPedidos,
COUNT(ShipRegion) AS FilasNoNulas,
MIN(ShippedDate) AS FechaMin, MAX(ShippedDate) AS FechaMax,
SUM(Freight) PesoTotal, AVG(Freight) PesoPromedio
FROM Orders INNER JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
GROUP BY Employees.FirstName + ' ' + Employees.LastName
```

Results | Messages

	Empleado	TotalPedidos	FilasNoNulas	FechaMin	FechaMax	PesoTotal	PesoPromedio
1	Andrew Fuller	96	31	1996-08-12 00:00:00.000	1998-05-04 00:00:00.000	8696,41	90,5876
2	Anne Dodsworth	43	14	1996-07-15 00:00:00.000	1998-05-04 00:00:00.000	3326,26	77,3548
3	Janet Leverling	127	56	1996-07-15 00:00:00.000	1998-05-06 00:00:00.000	10884,74	85,7066
4	Laura Callahan	104	42	1996-07-25 00:00:00.000	1998-05-05 00:00:00.000	7487,88	71,9988
5	Margaret Peacock	156	62	1996-07-11 00:00:00.000	1998-05-01 00:00:00.000	11346,14	72,7316
6	Michael Suyama	67	32	1996-07-10 00:00:00.000	1998-04-24 00:00:00.000	3780,47	56,4249
7	Nancy Davolio	123	50	1996-07-23 00:00:00.000	1998-05-06 00:00:00.000	8836,64	71,8426
8	Robert King	72	22	1996-08-28 00:00:00.000	1998-05-05 00:00:00.000	6665,44	92,5755
9	Steven Buchanan	42	14	1996-07-16 00:00:00.000	1998-04-29 00:00:00.000	3918,71	93,3026

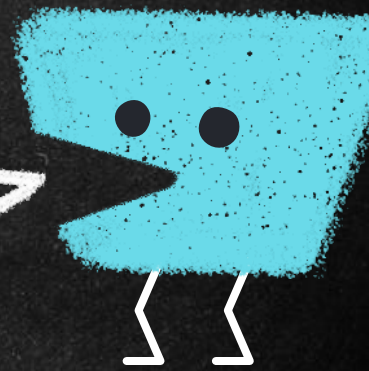
funciones de agregación que conozca.

- AVG. Devuelve el promedio de los valores en una columna.
- COUNT. Cuenta el número de filas que contienen un valor que no sea nulo en una columna.
- MAX. Devuelve el valor máximo que las filas tienen en una determinada columna.
- MIN. Devuelve el valor mínimo que las filas tienen en una determinada columna. Se puede seleccionar el calificador



funciones propias de SQL-Server.

- **Datename()**. Para obtener la fecha, utiliza parámetros como @Date part o @Expression.
- **SUM()**. Suma de distintos valores.
- **MID()**. Extrae valores de un campo tipo texto.
- **AVG()**. Proporciona la media de una serie de valores.



Para qué sirve la función CONCAT en SQL-Server

- **CONCAT()** toma dos o hasta 255 cadenas de entrada y las une en una. Requiere al menos dos cadenas de entrada. Si pasa una cadena de entrada, la función CONCAT() generará un error. Si pasa valores de cadena sin caracteres, la función CONCAT() convertirá implícitamente esos valores en cadenas antes de concatenar.

SQLQuery13.sql - E...QUINERO\javi (60))* | SQLQuery12.sql - E...QUINERO\javi (74))*

```
10  (Nombre, apellidoP, ApellidoM, Edad)
11  VALUES
12  ('Nombre1','ApellidoP1','ApellidoM1','Edad1'),
13  ('Nombre2','ApellidoP2','ApellidoM2','Edad2'),
14  ('Nombre3','ApellidoP3','ApellidoM3','Edad3');
15  GO
16  [Nombre], [apellidoP], [ApellidoM], [Edad]
17  SELECT
18      CONCAT(Nombre,',',apellidoP,',',ApellidoM,',',Edad)
19      FROM dbo.nombres
20  go
21  INSERT INTO dbo.nombres
22  (Nombre, apellidoP, ApellidoM, Edad)
23  VALUES
24  ('Nombre4',NULL,'ApellidoM1','Edad1');
25  GO
```

Muestra un ejemplo del uso de COUNT

SQLQuery1.sql - DI...EGO-PC\diego (56))*

```
insert into libros
  values('Aprenda PHP','Mario Molina','Nuevo siglo',null);
insert into libros
  values('Uno','Richard Bach','Planeta',20);

-- Averiguemos la cantidad de libros usando la función "count()":
select count(*)
  from libros;

-- Contamos los libros de editorial "Planeta":
select count(*)
  from libros
  where editorial='Planeta';

-- Contamos los registros que tienen precio (sin tener en cuenta
-- los que tienen valor nulo),
select count(precio)
  from libros;
```

100 %

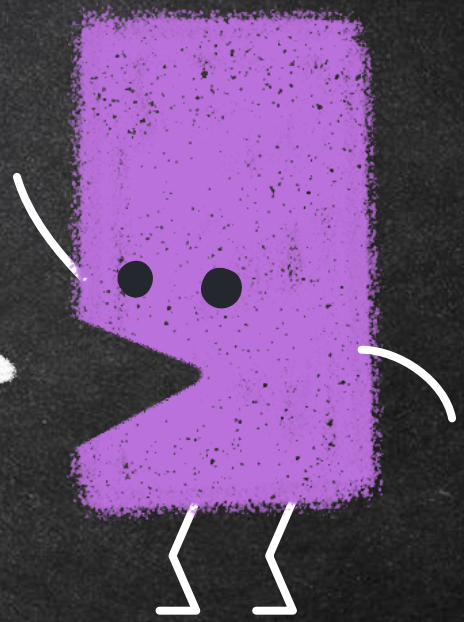
Results | Messages

	(No column name)
1	7

	(No column name)
1	2

	(No column name)
1	5

Query executed successfully.



Muestra un ejemplo del usos de AVG

SQLQuery5.sql - SQ...Administrador (53))* | SQL.Ejemplo - dbo....emplo - dbo.Ventas

```
Use Ejemplo
Go

SELECT Sum(Ventas)AS 'Total Ventas:'from Ventas group by Mes
SELECT Avg(Ventas)AS 'Media Ventas:'from Ventas group by Mes
SELECT Count(Ventas)AS 'Ventas Realizadas:'from Ventas group by Mes
```

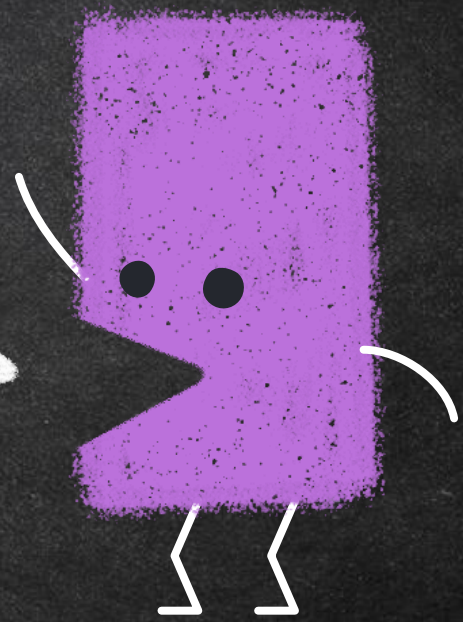
100 %

Resultados | Mensajes

	Total Ventas:
1	2357
2	2700
3	9757

	Media Ventas:
1	785
2	900
3	3252

	Ventas Realizadas:
1	3
2	3
3	3



Muestra un ejemplo del uso de MIN-MAX

SQLQuery5.sql - SQ...Administrador (53))* | SQL.Ejemplo - dbo.Ventas

```
Use Ejemplo
Go

SELECT MAX(Ventas)AS 'Mayor:' from Ventas
SELECT Min(Ventas)AS 'Menor' from Ventas
SELECT Sum(Ventas)AS 'Total Ventas' from Ventas
SELECT Avg(Ventas)AS 'Media Ventas:' from Ventas
```

100 %

Resultados | Mensajes

	Mayor:
1	5432

	Menor
1	123

	Total Ventas
1	14814

	Media Ventas:
1	1646