



MENU

ANALYSIS

CONTACT

DATA ANALYSIS

DEFENSA

HITO 4

BASE DE DATOS 1



0

PRESENTA

EDSON IVER CONDORI CONDORI - SIS10929449

EL ALTO - NOVIEMBRE DE 2022

UNIFRANZ
Internacionalízate

INNOVACIÓN
EN EDUCACIÓN




MENU

ANALYSIS

CONTACT

DATA ANALYSIS



SECCION 1: MANEJO DE CONCEPTOS





1. ¿QUÉ ES DDL Y DML EN SQL SERVER?

DDL

Sirve para crear y modificar la estructura de una Base de Datos, al DDL de la conoce también como Lenguaje de Definición de Datos.

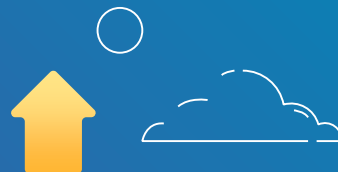
```
create database Unifranzitos;
use Unifranzitos;

--CREACION TABLA Campeonato
create table Campeonato(
id_Campeonato VARCHAR(100) PRIMARY KEY NOT NULL,
nombre_campeonato VARCHAR(100) NOT NULL,
sede VARCHAR(50) NOT NULL
);
```

DML

Es la manipulación de datos a través de Gestores de datos, permitiendo realizar Modificaciones, consultas de los datos que se tienen en la Base de Datos.

```
-- CONSULTAS DE TABLAS
select * from Campeonato;
select * from Equipo;
select * from Jugador;
```

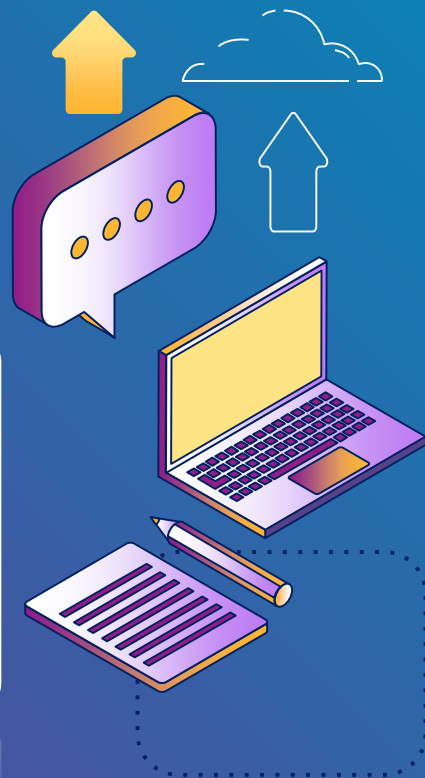




2. ¿PARA QUE NOS SIRVE EL INNER JOIN?

La Instrucción de INNER JOIN de igual manera se utiliza al momento de realizar una consulta, esta nos sirve para relacionar dos o mas tablas entre si, comparando e igualando de esta manera algo que sea característico de las tablas que se están relacionando.

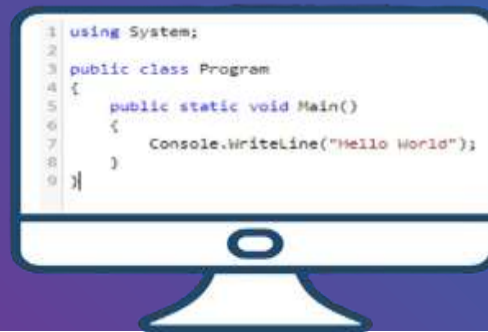
```
select camp.sede, equi.nombre_equipo, jug.nombres, jug.apellidos , jug.edad  
from Campeonato as camp  
inner join Equipo as equi on equi.id_campeonato = camp.id_Campeonato  
inner join Jugador as jug on jug.id_equipo = equi.id_equipo;
```





3. FUNCIONES DE AGREGACIÓN

Cuando se quiera utilizar una función de agregación, esta se debe de usar dentro de la clausula **SELECT** que es aplicado en grupos de registro (una columna de una tabla), y que **SIEMPRE** retornara un único valor (una fila) con los resultados.





4. ALGUNAS FUNCIONES DE AGREGACIÓN SON:

AVG = Calcula el promedio de los valores de un campo determinado.

COUNT = Devuelve el numero de registros de la selección.

SUM = Devuelve la suma de todos los valores de un campo determinado.

MAX = Devolver el valor mas alto de un campo determinado.

MIN = Devolver el valor mas bajo de un campo determinado.





4. ¿PARA QUE NOS SIRVE EL CONCAT?:

Esta Función sirve para poder agregar 2 o varias cadenas juntas es decir que las une.

SQL Statement:

```
SELECT CONCAT('SQL', ' ', 'es', ' ', 'Genial!');
```

Result:

Number of Records: 1

SQL es Genial!





4. EJEMPLO DE COUNT:

COUNT = Devuelve el numero de registros de la selección.

```
SELECT COUNT(*) Numero_de_Jugadores_Inscritos  
FROM Campeonato AS camp  
INNER JOIN Equipo AS equi on camp.id_Campeonato = equi.id_campeonato  
INNER JOIN Jugador as jug on equi.id_equipo = jug.id_equipo
```

Results		Messages	
		Numero_de_Jugadores_Inscritos	
1	5		





4. EJEMPLO DE AVG:

AVG = Calcula el promedio de los valores de un campo determinado.

```
Select  AVG (jug.edad) El_Promedio_de_edad_de_la_categoria_es  
from Equipo AS eq  
INNER JOIN Jugador as jug on eq.id_equipo = jug.id_equipo  
WHERE eq.categoria = 'Mujeres' and jug.edad >= 18;
```

Results		Messages
El_Promedio_de_edad_de_la_categoria_es		
1	21	





MENU

ANALYSIS

CONTACT

DATA ANALYSIS



SECCION 2: PARTE PRACTICA



3.1

3.1. Mostrar que jugadores que formen parte del equipo **equ-333**

```
-- 3.1 Mostrar que jugadores que son del equipo equ-333
Select jug.nombres , jug.apellidos, equ.id_equipo
from Jugador as jug
inner join Equipo as equ on equ.id_equipo = jug.id_equipo
where equ.id_equipo = 'equ-333';
```

Results		Messages	
	nombres	apellidos	id_equipo
1	Sandra	Solis	equ-333
2	Ana	Mica	equ-333

3.2

3.2. Crear una función que permita saber cuántos jugadores están inscritos.

- La función debe llamarse Crear una función que permita saber cuántos jugadores están inscritos.
- La función debe llamarse **F1_CantidadJugadores()**

```
CREATE FUNCTION F1_CantidadJugadores() -- CREANDO LA FUNCION
RETURNS INTEGER AS
BEGIN
    DECLARE @response INTEGER
    SELECT @response = COUNT(*)
    FROM Campeonato AS camp
    INNER JOIN Equipo AS equi on camp.id_Campeonato = equi.id_campeonato
    INNER JOIN Jugador as jug on equi.id_equipo = jug.id_equipo
    RETURN @response
END; -- FINALIZA LA FUNCION

-- REALIZANDO LA CONSULTA A LA FUNCION F1_CantidadJugadores
select dbo.F1_CantidadJugadores() Cantidad_de_Jugadores_Inscritos;
```

	Cantidad_de_Jugadores_Inscritos
1	5





3.3

3.3. Crear una función que permita saber cuántos jugadores están inscritos y que sean de la categoría varones o mujeres.

- La función debe llamarse **F2_CantidadJugadoresParam()**
- La función debe recibir un parámetro "Varones" o "Mujeres"

```
CREATE FUNCTION F2_CantidadJugadoresParam(@genero varchar(30)) -- CREANDO LA FUNCION
RETURNS INTEGER AS
BEGIN
DECLARE @response INTEGER
SELECT @response = COUNT(*)
FROM Campeonato AS camp
INNER JOIN Equipo AS equi on camp.id_Campeonato = equi.id_campeonato
INNER JOIN Jugador as jug on equi.id_equipo = jug.id_equipo
WHERE equi.categoria = @genero
RETURN @response
END; -- FINALIZA LA FUNCION

-- REALIZANDO LA CONSULTA A LA FUNCION F1_CantidadJugadores
select dbo.F2_CantidadJugadoresParam('Mujeres') Cantidad_de_Jugadores
```



Results

Messages

	Cantidad_de_Jugadores
1	2





3.4

3.4. Crear una función que obtenga el promedio de las edades mayores a una cierta edad.

- La función debe llamarse **F3_PromedioEdades()**
- La función debe recibir como parámetro 2 valores.
- La categoría. (Varones o Mujeres)
- La edad con la que se comparara (21 años ejemplo)
- Es decir mostrar el promedio de edades que sean de una categoría y que sean mayores a 21 años.

```
CREATE FUNCTION F3_PromedioEdades(@genero VARCHAR(20), @edad INTEGER)
RETURNS INTEGER AS
BEGIN
    DECLARE @response INTEGER

    Select @response = avg (jug.edad)
    from Equipo AS eq
    INNER JOIN Jugador as jug on eq.id_equipo = jug.id_equipo
    WHERE eq.categoria = @genero and jug.edad >= @edad;

    RETURN @response
END; -- FINALIZA LA FUNCION

-- REALIZANDO LA CONSULTA A LA FUNCION F3_PromedioEdades
SELECT dbo.F3_PromedioEdades('mujeres',21) El_promedio_de_edad_de_cate
```

Results		Messages
El_promedio_de_edad_de_categoria_ingresada_es		
1	23	





3.5

3.5. Crear una función que permita concatenar 3 parámetros.

- La función debe llamarse F4_ConcatItems()
- La función debe de recibir 3 parámetros.
- La función debe de concatenar los 3 valores.
- Para verificar la correcta creación de la función debe mostrar lo siguiente.
- Mostrar los nombres de los jugadores, el nombre del equipo y la sede concatenada, utilizando la función que acaba de crear.

```
ALTER FUNCTION F4_ConcatItems(@nombre VARCHAR (20),@nom_equipo VARCHAR(50),@sede VARCHAR(30))
RETURNS VARCHAR (100) AS
BEGIN
DECLARE @response VARCHAR(100)

SELECT @response = CONCAT (jug.nombres, ' ',eq.nombre_equipo, ' ',cam.sede)
FROM Campeonato AS cam
INNER JOIN Equipo as eq on cam.id_Campeonato = eq.id_campeonato
INNER JOIN Jugador as jug on eq.id_equipo = jug.id_equipo
WHERE jug.nombres = @nombre AND eq.nombre_equipo = @nom_equipo AND cam.sede = @sede

RETURN @response
END -- finalizando

-- REALIZANDO LA CONSULTA A LA FUNCION CREADA
SELECT dbo.F4_ConcatItems('Carlos','404 Not Found','El Alto');
```

Results		Messages	
		(No column name)	
1	Carlos 404 Not found El Alto		





MENU

ANALYSIS

CONTACT

DATA ANALYSIS

GRACIAS POR SU ATENCION !!!



EL ALTO - NOVIEMBRE DE 2022

UNIFRANZ
Internacionalízate

INNOVACIÓN
EN EDUCACIÓN