

Manejo de Conceptos en Bases de Datos

Nombre: Elio Vásquez Gonzales

• DDL y DML

¿Qué es DDL (Data Definition Language)?
¿Qué es DML (Data Manipulation Language)?
Ejemplo en la base de datos UNIFRANZITOS.

```
create table campeonato
(
  id_campeonato varchar (12) primary key not null,
  nombre_campeonato varchar (30) not null,
  sede varchar (20) not null
)

insert into campeonato (id_campeonato,nombre_campeonato,sede)
values ('camp-111','Campeonato Unifranz','El Alto');
insert into campeonato (id_campeonato,nombre_campeonato,sede)
values ('camp-222','Campeonato Unifranz','Cochabamba');
```

```
id_campeonato varchar(12),
foreign key (id_campeonato) references campeonato(id_campeonato)
```

PRIMARY KEY y FOREIGN KEY

- Definición de PRIMARY KEY.
- Definición de FOREIGN KEY.

```
foreign key (id_equipo) references equipo(id_equipo)
```

Tablas y Uso de IDENTITY

- Definición de una TABLA.
- Uso de IDENTITY en una tabla

```
create table jugador
(
    id_jugador varchar(12) primary key not null,
    nombres varchar(30) not null,
    apellidos varchar(50) not null,
    ci varchar(15) not null,
    edad int not null,

    id_equipo varchar (12) ,
    foreign key (id_equipo) references equipo(id_equipo)
);

insert into jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
values ('jug-111', 'Carlos', 'Villa', '8997811LP', 19, 'equ-222');
insert into jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
values ('jug-222', 'Pedro', 'Salas', '8997822LP', 20, 'equ-222');
insert into jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
values ('jug-333', 'Saul', 'Araj', '8997833LP', 21, 'equ-222');
insert into jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
values ('jug-444', 'Sandra', 'Solis', '8997844LP', 22, 'equ-333');
insert into jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
values ('jug-555', 'Ana', 'Mica', '8997855LP', 23, 'equ-333');
```

Cláusula WHERE

- Uso de la cláusula WHERE en consultas SQL.

```
-----3. Mostrar jugadores mayores o igual a 21 años que sean de la categoría 'VARONES':  
select * from jugador  
inner join equipo on jugador.id_equipo = equipo.id_equipo  
where jugador.edad >= 21 and equipo.categoria = 'varones';
```

INNER JOIN

- Uso de la instrucción INNER JOIN en consultas SQL.
 - Ejemplo de INNER JOIN.

```
-----3. Mostrar jugadores mayores o igual a 21 años que sean de la categoría 'VARONES':  
select * from jugador  
inner join equipo on jugador.id_equipo = equipo.id_equipo  
where jugador.edad >= 21 and equipo.categoria = 'varones';
```

Ejemplo de INNER JOIN con Imagen

Query Builder interface showing an INNER JOIN query:

```
1 select * from estado
2 inner join ciudad on
3 ciudad.idEstado= estado.id
```

Result 1:

id	nombre	id(1)	nombre(1)	idEstado
1	Jalisco	1	Guadalajara	1
1	Jalisco	2	Zapopan	1
1	Jalisco	3	Tlaquepaque	1
2	Michoacan	4	Morelia	2
2	Michoacan	5	Yurécuaro	2
2	Michoacan	6	Uruapan	2

Diagram illustrating the INNER JOIN concept using a Venn diagram with two overlapping circles labeled A and B. The intersection represents the result of the INNER JOIN.

Database structure and data:

estado @pruebas (loc...)

id	nombre
1	Jalisco
2	Michoacan
3	Colima

ciudad @pruebas ...

id	nombre	idEstado
1	Guadalajara	1
2	Zapopan	1
3	Tlaquepaque	1
4	Morelia	2
5	Yurécuaro	2
6	Uruapan	2
7	CDMX	(Null)
8	Cancún	(Null)

LEFT JOIN

Screenshot of a database query tool showing a LEFT JOIN query and its results.

Query:

```
1 select * from estado
2 left join ciudad on
3 ciudad.idEstado= estado.id
```

Result 1:

id	nombre	id(1)	nombre(1)	idEstado
1	Jalisco	1	Guadalajara	1
1	Jalisco	2	Zapopan	1
1	Jalisco	3	Tlaquepaque	1
2	Michoacan	4	Morelia	2
2	Michoacan	5	Yurécuaro	2
2	Michoacan	6	Uruapan	2
3	Colima	(Null)	(Null)	(Null)

Diagram: A Venn diagram illustrating a LEFT JOIN. It shows two overlapping circles, A (yellow) and B (orange). The intersection is shaded orange. The left circle is labeled A, and the right circle is labeled B.

Tool Interface: The screenshot shows a database query tool with a menu bar (File, Edit, View, Window, Help) and a toolbar. The query is executed against a database named 'pruebas'. The results are displayed in a table with columns 'id', 'nombre', 'id(1)', 'nombre(1)', and 'idEstado'. The table shows data for Jalisco, Michoacan, and Colima. The 'id(1)', 'nombre(1)', and 'idEstado' columns are highlighted in blue. The 'id' column is highlighted in blue. The 'nombre' column is highlighted in blue. The 'idEstado' column is highlighted in blue. The 'id' column is highlighted in blue. The 'nombre' column is highlighted in blue. The 'idEstado' column is highlighted in blue.

RIGHT JOIN

localhost pruebas Run Stop Explain

```
1 select * from estado
2 right join ciudad on
3 ciudad.idEstado= estado.id
```

Message Result 1 Profile Status

id	nombre	id(1)	nombre(1)	idEstado
1	Jalisco	1	Guadalajara	1
1	Jalisco	2	Zapopan	1
1	Jalisco	3	Tlaquepaque	1
2	Michoacan	4	Morelia	2
2	Michoacan	5	Yurécuaro	2
2	Michoacan	6	Uruapan	2
(Null)	(Null)	7	CDMX	(Null)
(Null)	(Null)	8	Cancún	(Null)

estado @pruebas (loc...)

id	nombre
1	Jalisco
2	Michoacan
3	Colima

ciudad @pruebas ...

id	nombre	idEstado
1	Guadalajara	1
2	Zapopan	1
3	Tlaquepaque	1
4	Morelia	2
5	Yurécuaro	2
6	Uruapan	2
7	CDMX	(Null)
8	Cancún	(Null)

INSERT Syntax DML
Insert new rows into an existing table

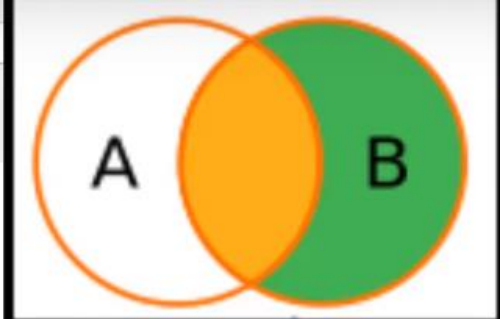
LOOP Flow Control
Create a simple loop construct

REPEAT Flow Control
Create a loop until the condition is met

SELECT Flow Control
Retrieve data from the database

UPDATE Flow Control
Update new values in the database

WHILE Flow Control
Create a loop until the condition is met



EJERCICIO DEL CAMPEONATO

```
create database procesual_h3;
use procesual_h3;

create table campeonato
(
  id_campeonato varchar (12) primary key not null,
  nombre_campeonato varchar (30) not null,
  sede varchar (20) not null
)

insert into campeonato (id_campeonato,nombre_campeonato,sede)
values ('camp-111','Campeonato Unifranz','El Alto');

insert into campeonato (id_campeonato,nombre_campeonato,sede)
values ('camp-222','Campeonato Unifranz','Cochabamba');
```

```
create table equipo
(
    id_equipo varchar(12) primary key not null,
    nombre_equipo varchar(30) not null,
    categoria varchar(8) not null check (categoria in ('varones', 'mujeres')),

    id_campeonato varchar(12),
    foreign key (id_campeonato) references campeonato(id_campeonato)
);

INSERT INTO equipo (id_equipo, nombre_equipo, categoria, id_campeonato)
VALUES ('equ-111', 'Google', 'VARONES', 'camp-111');
INSERT INTO equipo (id_equipo, nombre_equipo, categoria, id_campeonato)
VALUES ('equ-222', '404 Not found', 'VARONES', 'camp-111');
INSERT INTO equipo (id_equipo, nombre_equipo, categoria, id_campeonato)
VALUES ('equ-333', 'Girls Unifranz', 'MUJERES', 'camp-111');
```

```
create table jugador
(
    id_jugador varchar(12) primary key not null,
    nombres varchar(30) not null,
    apellidos varchar(50) not null,
    ci varchar(15) not null,
    edad int not null,

    id_equipo varchar (12) ,
    foreign key (id_equipo) references equipo(id_equipo)
);

insert into jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
values ('jug-111', 'Carlos', 'Villa', '8997811LP', 19, 'equ-222');
insert into jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
values ('jug-222', 'Pedro', 'Salas', '8997822LP', 20, 'equ-222');
insert into jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
values ('jug-333', 'Saul', 'Araj', '8997833LP', 21, 'equ-222');
insert into jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
values ('jug-444', 'Sandra', 'Solis', '8997844LP', 22, 'equ-333');
insert into jugador (id_jugador, nombres, apellidos, ci, edad, id_equipo)
values ('jug-555', 'Ana', 'Mica', '8997855LP', 23, 'equ-333');
```

```
---- 1.Mostrar jugadores que son del equipo 'equ-222':
select * from jugador
where id_equipo = 'equ-222';

---- 2.Mostrar jugadores (nombres, apellidos) que juegan en la sede de El Alto:
select nombres, apellidos from jugador
inner join equipo on jugador.id_equipo = equipo.id_equipo
inner join campeonato on equipo.id_campeonato = campeonato.id_campeonato
where campeonato.sede = 'El alto';

-----3. Mostrar jugadores mayores o igual a 21 años que sean de la categoría 'VARONES':
select * from jugador
inner join equipo on jugador.id_equipo = equipo.id_equipo
where jugador.edad >= 21 and equipo.categoria = 'varones';

-----4.Mostrar todos los jugadores en donde su apellido empiece con la letra 'S':
select * from jugador
where apellidos like 's%';

-----5.Mostrar los equipos que forman parte del campeonato 'camp-111' y sean de la categoría 'MUJERES':
select * from equipo
```

-----6. Mostrar el nombre del equipo del jugador con id_jugador igual a 'jug-333':

```
= select equipo.nombre_equipo from jugador
  inner join equipo on jugador.id_equipo = equipo.id_equipo
 where jugador.id_jugador = 'jug-333';
```

-----7. Mostrar el nombre del campeonato del jugador con id_jugador igual a 'jug-333':

```
= select campeonato.nombre_campeonato from jugador
  inner join equipo on jugador.id_equipo = equipo.id_equipo
  inner join campeonato on equipo.id_campeonato = campeonato.id_campeonato
 where jugador.id_jugador = 'jug-333';
```

-----8. Consulta SQL que maneje las 3 tablas (jugador, equipo, campeonato):

```
= select jugador.id_jugador, jugador.nombres, jugador.apellidos, jugador.ci, jugador.edad,
        equipo.nombre_equipo, equipo.categoria,
        campeonato.nombre_campeonato, campeonato.sede
 from jugador
  inner join equipo on jugador.id_equipo = equipo.id_equipo
  inner join campeonato on equipo.id_campeonato = campeonato.id_campeonato;
```

-----9. Para determinar cuántos equipos inscritos hay:

```
select count(distinct id_equipo) as cantidad_equipos from jugador;
```

-----10. Para determinar cuántos jugadores pertenecen a la categoría 'VARONES' o 'MUJERES':

```
= select categoria, count(*) as cantidad_jugadores
 from equipo
  inner join jugador on equipo.id_equipo = jugador.id_equipo
 group by categoria;
```