

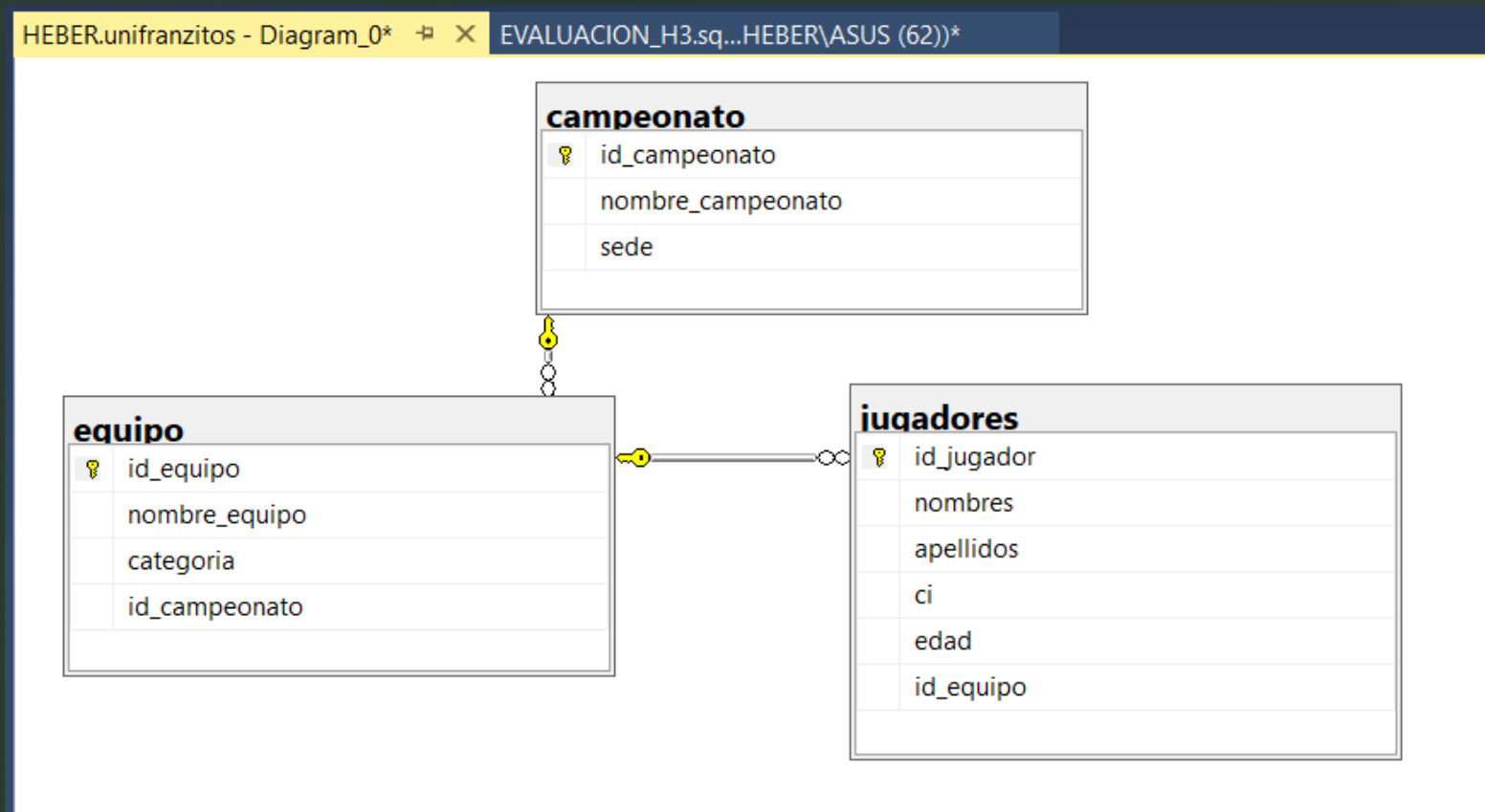
Procesual-H3

Estudiante: Heber Mollericona Miranda

CI:13409189

Manejo de conceptos

- Adjuntar el diagrama E-R GENERADO por su editor



• Que es DDL y DML

Adicionalmente muestra un ejemplo en la base de datos UNIFRANZITOS.

DDL se utilizan para describir una base de datos, para definir su estructura, para crear sus objetos y para crear los sub objetos de la tabla.

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

DML se utilizan para controlar la información contenida en la base de datos.

```
insert into campeonato(id_campeonato, nombre_campeonato, sede) values('camp-111',  
delete from campeonato  
select*  
from campeonato
```

• Que significa PRIMARY KEY y FOREIGN KEY

Primary key es una columna o un grupo de columnas que identifica de forma exclusiva cada fila de una tabla, esta clausula restringe que entre valores duplicados.

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

Si se inserta datos duplicado en primary key saltara un mensaje de error

```
109 %  
Messages  
Msg 2627, Level 14, State 1, Line 11  
Infracción de la restricción PRIMARY KEY 'PK_campeona_2AAD398DB28FB97E'. No se puede insertar una clave duplicada en el objeto 'dbo.campeonato'. El valor de la clave duplicada es (camp-111).  
Se terminó la instrucción.  
  
Completion time: 2022-10-20T21:26:59.2472989-04:00
```

FOREIGN KEY es una clave que sirve para relacionar dos tablas, se relaciona o vincula con la **PRIMARY KEY** de otra tabla.

FOREIGN KEY es una restricción que no permite que se agreguen o inserten datos que no válidos en la columna, ya que los valores que se van a insertar deben ser valores que se encuentren en la tabla con la que se quiere relacionar.

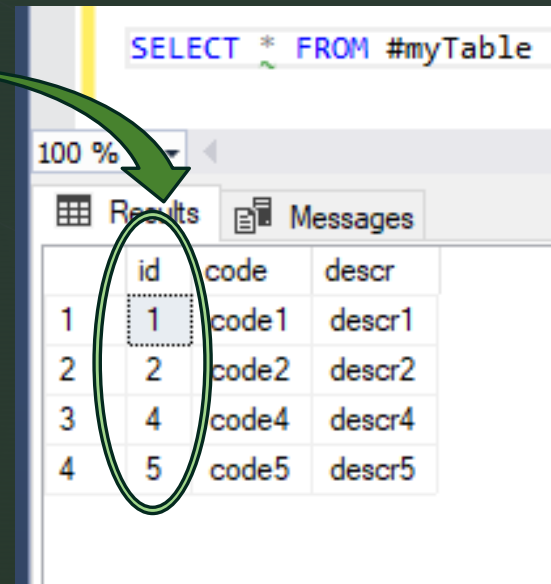
```
create table equipo(  
  id_equipo varchar(20) primary key not null,  
  nombre_equipo varchar(25) not null,  
  categoria varchar(20) not null,  
  id_campeonato varchar(20) not null,  
  foreign key (id_campeonato) references campeonato(id_campeonato)  
);
```

- Defina que es una **TABLA** y el uso de **IDENTITY**.

Son campos que contiene todos los datos de una DATABASE. En las tablas, los datos se organizan con arreglo a un formato de filas y columnas.

IDENTITY es una columna que es auto incrementable cada vez que se inserta un dato.

IDENTITY

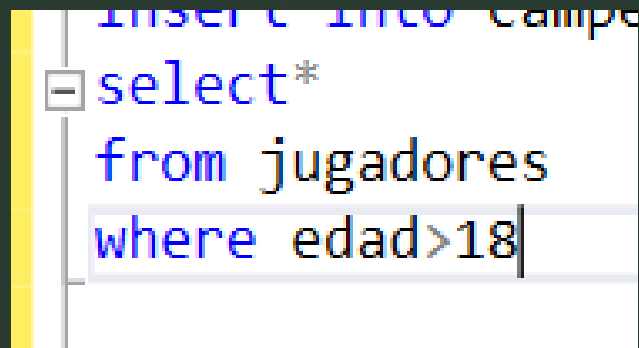


A screenshot of a SQL query result window. The query is `SELECT * FROM #myTable`. The results are displayed in a table with columns `id`, `code`, and `descr`. The `id` column contains values 1, 2, 4, and 5, which are circled in green. A green arrow points from the word ***IDENTITY*** to the `id` column. The `code` column contains values `code1`, `code2`, `code4`, and `code5`. The `descr` column contains values `descr1`, `descr2`, `descr4`, and `descr5`.

	id	code	descr
1	1	code1	descr1
2	2	code2	descr2
3	4	code4	descr4
4	5	code5	descr5

- Para que se utiliza la cláusula **WHERE**.

La clausula WHERE se usa mas que todo para hacer consultas en una DATABASE, se debe dar las instrucciones para que realice la búsqueda que estamos realizando

A screenshot of a code editor window showing a SQL query. The query is: `select*
from jugadores
where edad>18|`. The text is color-coded: `select*` is blue, `from jugadores` is blue, and `where edad>18|` is blue. A yellow vertical bar is on the left side of the code editor. The cursor is at the end of the line `where edad>18|`.

```
Insert Into campe  
select*  
from jugadores  
where edad>18|
```


- Para que se utiliza la instrucción
INNER JOIN

Inner Join se utiliza para poder combina los registros de dos tablas, si hay valores iguales en las tablas seleccionadas se mostrará los registros que son iguales de ambas tablas.

```
select jgd.nombres, jgd.apellidos, jgd.edad  
from equipo as equi  
inner join jugadores as jgd on equi.id_equipo = jgd.id_equipo  
where equi.nombre_equipo = '484 Not found' and jgd.edad <=20
```



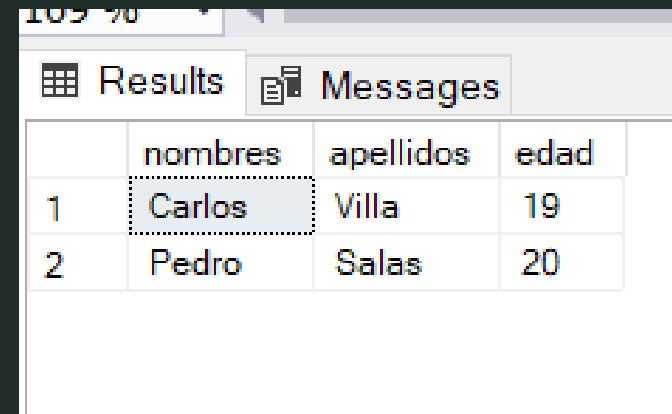
EJEMPLOS

DE LOS CONCEPTOS ANTERIORES:

INNER JOIN

USANDO BASE DE
DATOS DEL EJEMPLO

```
--menores a 20 años
select jgd.nombres, jgd.apellidos, jgd.edad
from equipo as equi
inner join jugadores as jgd on equi.id_equipo = jgd.id_equipo
where equi.nombre_equipo = '484 Not found' and jgd.edad <=20
```



The screenshot shows a database application window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with four columns: an index, 'nombres', 'apellidos', and 'edad'. There are two rows of data. The first row has index 1, name 'Carlos', surname 'Villa', and age 19. The second row has index 2, name 'Pedro', surname 'Salas', and age 20. The 'Messages' tab is empty.

	nombres	apellidos	edad
1	Carlos	Villa	19
2	Pedro	Salas	20

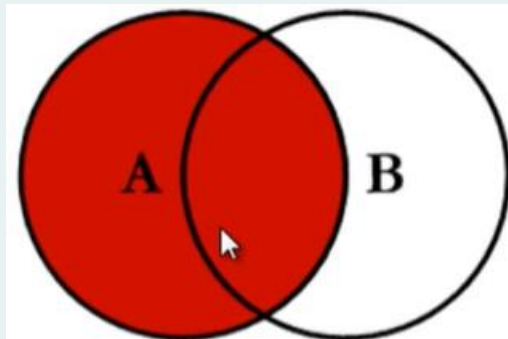
```
select*  
from jugadores as j  
left join equipo as q on j.id_equipo = q.id_equipo;
```

```
select*  
from jugadores as j  
left join equipo as q on j.id_equipo = q.id_equipo;
```

09 %

Results Messages

	id_jugador	nombres	apellidos	ci	edad	id_equipo	id_equipo	nombre_equipo	categoria	id_campeonato
1	jug-111	Carlos	Villa	8997811LP	19	equ-222	equ-222	484 Not found	VARONES	camp-111
2	jug-222	Pedro	Salas	8997822LP	20	equ-222	equ-222	484 Not found	VARONES	camp-111
3	jug-333	Saul	Araj	8997833LP	21	equ-222	equ-222	484 Not found	VARONES	camp-111
4	jug-444	Sandra	Solis	8997844LP	20	equ-333	equ-333	girls unifranz	MUJERES	camp-111
5	jug-555	Ana	Mica	8997855LP	23	equ-333	equ-333	girls unifranz	MUJERES	camp-111



LEFT JOIN

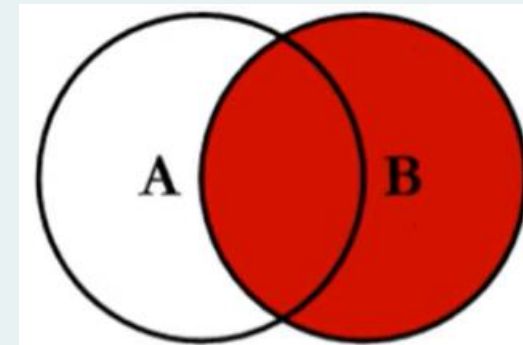
USANDO BASE DE
DATOS DEL EJEMPLO

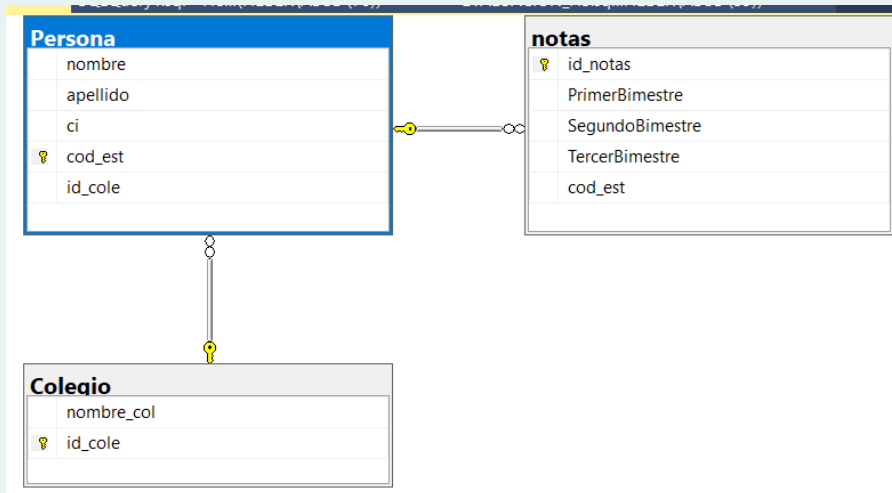
RIGHT JOIN

USANDO BASE DE
DATOS DEL EJEMPLO

```
select*  
from jugadores as j  
right join equipo as q on j.id_equipo = q.id_equipo;
```

	id_jugador	nombres	apellidos	ci	edad	id_equipo	id_equipo	nombre_equipo	categoria	id_campeonato
1	NULL	NULL	NULL	NULL	NULL	NULL	equ-111	Google	VARONES	camp-111
2	jug-111	Carlos	Villa	8997811LP	19	equ-222	equ-222	484 Not found	VARONES	camp-111
3	jug-222	Pedro	Salas	8997822LP	20	equ-222	equ-222	484 Not found	VARONES	camp-111
4	jug-333	Saul	Araj	8997833LP	21	equ-222	equ-222	484 Not found	VARONES	camp-111
5	jug-444	Sandra	Solis	8997844LP	20	equ-333	equ-333	girls unifranz	MUJERES	camp-111
6	jug-555	Ana	Mica	8997855LP	23	equ-333	equ-333	girls unifranz	MUJERES	camp-111





```
select col.nombre_col, per.nombre, not1.PrimerBimestre
from Persona as per
inner join notas as not1 on per.cod_est = not1.cod_est
inner join Colegio as col on col.id_cole = per.id_cole
where not1.PrimerBimestre > 51
```


Results			
	nombre_col	nombre	PrimerBimestre
1	San Vicente De Paul	Heber	56
2	San Vicente De Paul	Juan	86

Crear 3 tablas
y crear una
consulta SQL
que muestra el
uso de INNER
JOIN.

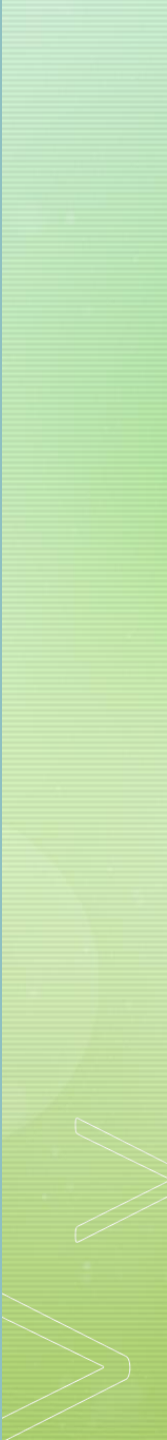


MANEJO DE CONSULTAS


Mostrar los
jugadores que son
del equipo
equ-222



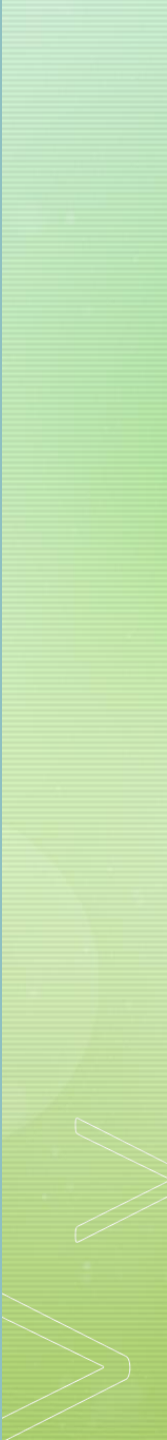
Mostrar que
jugadores (nombres,
apellidos) que juegan
en la sede de **El Alto**



Mostrar aquellos
jugadores mayores o
igual a **21** años que sean
de la categoría
VARONES.



Mostrar a todos los
estudiantes en donde su
apellido empieza con la
letra S.





Mostrar que equipos forman
parte del campeonato
camp-111 y además sean de
la categoría **MUJERES**

Mostrar el nombre del
equipo del jugador con
id_jugador igual a
jug-333

Mostrar el nombre del
campeonato del
jugador con
id_jugador igual a
jug-333

■ Crear una
consulta SQL que
maneje las 3
tablas de la base
de datos

¿Qué estrategia
utilizaría para
determinar
**cuántos equipos
inscritos hay?**

¿Qué estrategia
utilizaría para
determinar cuántos
jugadores pertenecen a
la categoría **VARONES**
o Categoría
MUJERES.