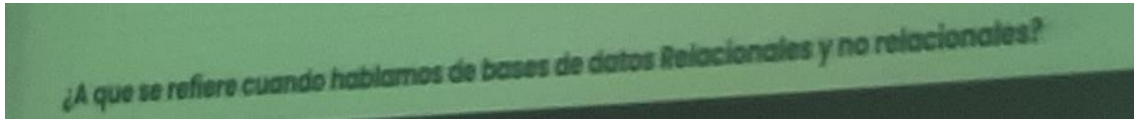


# Procesual



Base de datos relacionales

Una base de datos relacional es una recopilación de elementos de datos con relaciones predefinidas entre ellos

Base de datos no relacionales

Están diseñadas para varios patrones de acceso a **datos que** incluyen aplicaciones de baja latencia.

```
create database UNI_Hito2
```

```
use UNI_Hito2;
```

```
create table estudiante(
```

```
    estudiante_id INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
```

```
    nombres VARCHAR(20),
```

```
    apellidos VARCHAR(40),
```

```
    edad INT,
```

```
    genero VARCHAR(20)
```

```
)
```

```
create table materia(
```

```
    materia_id INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
```

```
    nombre VARCHAR(20),
```

```
    codigo VARCHAR(20),
```

```
    aula VARCHAR(20)
```

```
)
```

```
create table inscribe(
```

```
    inscripcion_id INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
```

```
    estudiante_id INT NOT NULL,
```

```

        materia_id INT,
        semestre VARCHAR(30),
        gestion VARCHAR(30),
        FOREIGN KEY (estudiante_id) REFERENCES estudiante(estudiante_id),
        FOREIGN KEY (materia_id) REFERENCES materia(materia_id)
    )

```

```

INSERT INTO estudiante(nombres,apellidos,edad,genero) VALUES('Miguel',
'Marino Condori', 20, 'Masculino');

```

```

INSERT INTO estudiante(nombres,apellidos,edad,genero) VALUES('Jessica',
'Mixa Suxo', 25, 'Femenino');

```

```

INSERT INTO materia(nombre,codigo,aula) VALUES('Fisica', 'FIS-101', 'A01');

```

```

INSERT INTO materia(nombre,codigo,aula) VALUES('Calculo', 'CAL-101',
'B01');

```

```

INSERT INTO inscribe(estudiante_id, materia_id, semestre, gestion) VALUES(1,
2, 'Primero', 2022);

```

```

INSERT INTO inscribe(estudiante_id, materia_id, semestre, gestion) VALUES(2,
1, 'Primero', 2022);

```

*\*/ejercicio 5\*/*

```

create function menor_est(gen VARCHAR(50))

```

```

    returns INTEGER

```

```

    begin

```

```

        return (

```

```

            select MIN(edad)

```

```

            from estudiante

```

```

            where genero = gen

```

```

        );

```

```

    end;

```

```

select menor_est('femenino') as ID_EST

```

```
select MIN(edad) min from estudiante
```

```
*/ejercicio 7 */
```

```
select CONCAT(e.nombres,' ',e.apellidos) nombre, m.nombre materia, i.gestion
```

```
from inscribe i
```

```
left join estudiante e on i.estudiante_id = e.estudiante_id
```

```
left join materia m on i.materia_id = m.materia_id
```

```
where i.gestion = 2022
```

```
/* ejercicio 8 */
```

```
create function buscar_val(mat VARCHAR(50), val VARCHAR(50))
```

```
returns BOOLEAN
```

```
begin
```

```
return (select if(mat = val, TRUE, FALSE));
```

```
end;
```

```
select CONCAT(e.nombres,' ',e.apellidos) nombre, m.nombre materia
```

```
from inscribe i
```

```
left join estudiante e on i.estudiante_id = e.estudiante_id
```

```
left join materia m on i.materia_id = m.materia_id
```

```
where buscar_val(m.codigo,'FIS-101')
```

```
create function verf(nom VARCHAR(20), ap VARCHAR(40))
```

```
returns BOOLEAN
```

```
begin
```

```
return (select if(nom != " AND ap != "", TRUE, FALSE));
```

```
end;
```

```
select *
```

```
from estudiante
```

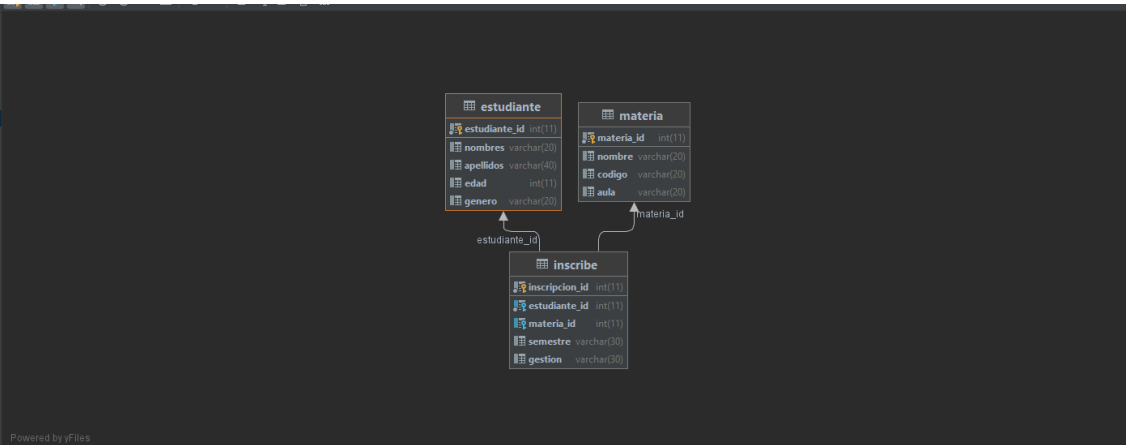
```
where verf(nombres, apellidos)
```

```
36 create function men_est(gen VARCHAR(50))
37 returns INTEGER
38 begin
39     return (
40         select estudiante_id
41         from estudiante
42         where genero = gen
43         ORDER BY edad asc
44         LIMIT 1
45     );
46 end;
47
48 select men_est( gen: 'femenino') as ID_EST
```

ID\_EST

Output × ID\_EST: int(11) ×

ID_EST
1



```
create function menor_est(gen VARCHAR(50))
returns INTEGER
begin
    return (
        select MIN(edad)
        from estudiante
        where genero = gen
    );
end;

select menor_est( gen: 'femenino') as ID_EST

select MIN(edad) min from estudiante
```

menor\_est()

Output × ID\_EST: int(11) ×

ID_EST
1

```

50 select MIN(edad) min from estudiante;
51 select CONCAT(e.nombres, ' ', e.apellidos) nombre, m.nombre materia, i.gestion
52 from inscribe i
53 left join estudiante e on i.estudiante_id = e.estudiante_id
54 left join materia m on i.materia_id = m.materia_id
55 where i.gestion = 2022

```

Output × Result 21 ×

	nombre	materia	gestion
1	Miguel Marino Condori	Calculo	2022
2	Jessica Mixa Suxo	Fisica	2022

```

50 select MIN(edad) min from estudiante;
51 select CONCAT(e.nombres, ' ', e.apellidos) nombre, m.nombre materia, i.gestion
52 from inscribe i
53 left join estudiante e on i.estudiante_id = e.estudiante_id
54 left join materia m on i.materia_id = m.materia_id
55 where i.gestion = 2022

```

Output × Result 21 ×

	nombre	materia	gestion
1	Miguel Marino Condori	Calculo	2022
2	Jessica Mixa Suxo	Fisica	2022

```

1 create function buscar_val(mat VARCHAR(50), val VARCHAR(50))
2 returns BOOLEAN
3 begin
4 return (select if(mat = val, TRUE, FALSE));
5 end;
6
7 select CONCAT(e.nombres, ' ', e.apellidos) nombre, m.nombre materia
8 from inscribe i
9 left join estudiante e on i.estudiante_id = e.estudiante_id
10 left join materia m on i.materia_id = m.materia_id
11 where buscar_val(m.codigo, 'FIS-101')

```

Output × Result 23 ×

```

RETURN BOOLEAN
begin
return (select if(mat = val, TRUE, FALSE));
end
[2022-03-30 22:51:59] completed in 169 ms
uni_hira2> select CONCAT(e.nombres, ' ', e.apellidos) nombre, m.nombre materia
from inscribe i
left join estudiante e on i.estudiante_id = e.estudiante_id
left join materia m on i.materia_id = m.materia_id
where buscar_val(m.codigo, 'FIS-101')
[2022-03-30 22:52:58] 1 row retrieved starting from 1 in 42 ms (execution: 7 ms, fetching: 35 ms)

```

```
/* ejercicio 8 */
create function buscar_val(mat VARCHAR(50), val VARCHAR(50))
returns BOOLEAN
begin
return (select if(mat = val, TRUE, FALSE));
end;

select CONCAT(e.nombres, ' ', e.apellidos) nombre, m.nombre materia
from inscribe i
left join estudiante e on i.estudiante_id = e.estudiante_id
left join materia m on i.materia_id = m.materia_id
where buscar_val(mati: m.codigo, val: 'FIS-101')
```

Output × Result 23 ×

	nombre	materia
1	Jessica Mixa Suxo	Fisica

```
67 where buscar_val(mati: m.codigo, val: 'FIS-101')
68
69 create function verf(nom VARCHAR(20), ap VARCHAR(40))
70 returns BOOLEAN
71 begin
72 return (select if(nom != '' AND ap != '', TRUE, FALSE));
73 end;
74
75 select *
76 from estudiante
77 where verf(nom: nombres, ap: apellidos)
```

Output × Result 25 ×

	estudiante_id	nombres	apellidos	edad	genero
1	1	Miguel	Marino Condori	20	Masculino
2	2	Jessica	Mixa Suxo	25	Femenino