



Evaluación Procesual Hito 2

BASE DE DATOS II

Luis Alvarez Medina



Manejo de conceptos

1. ¿A que se refiere cuando se habla de bases de datos relacionales?

Son aquellas bases de datos que organizan la información en tablas, donde estas están relacionadas entre si mediante un primary key.

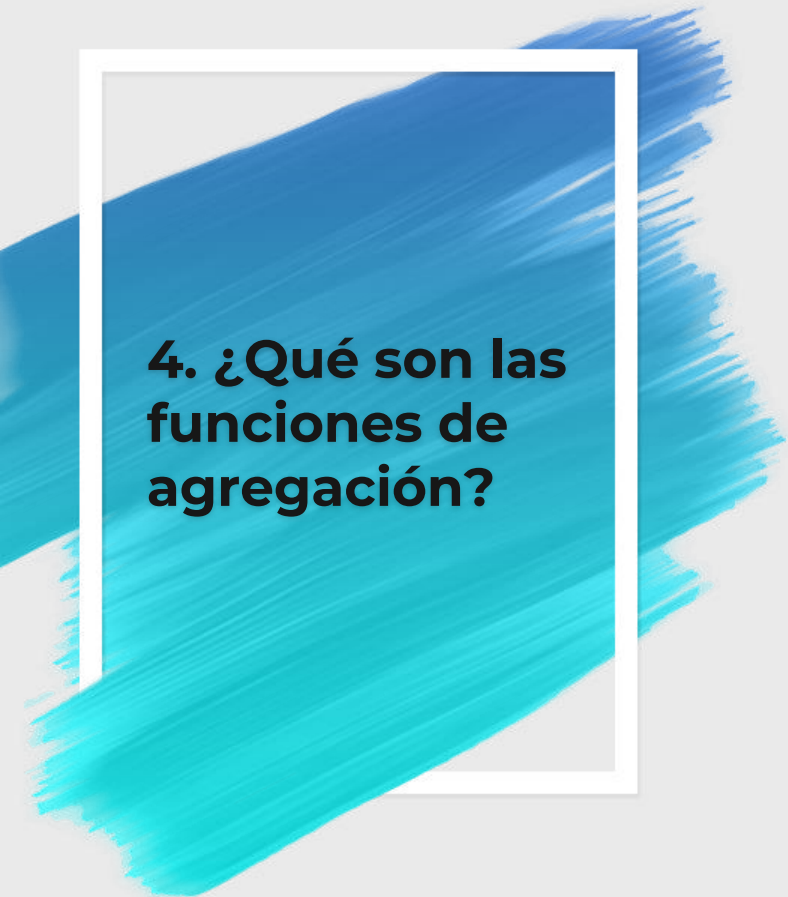
2. ¿A que se refiere cuando se habla de bases de datos no relacionales?

Son aquellas bases de datos que la información lo almacenan en documentos, grafos, etc. donde estas no se encuentran explícitamente relacionadas entre ellas.

3. ¿Qué es MySQL y MariaDB?. Explique si existen diferencias o son iguales.

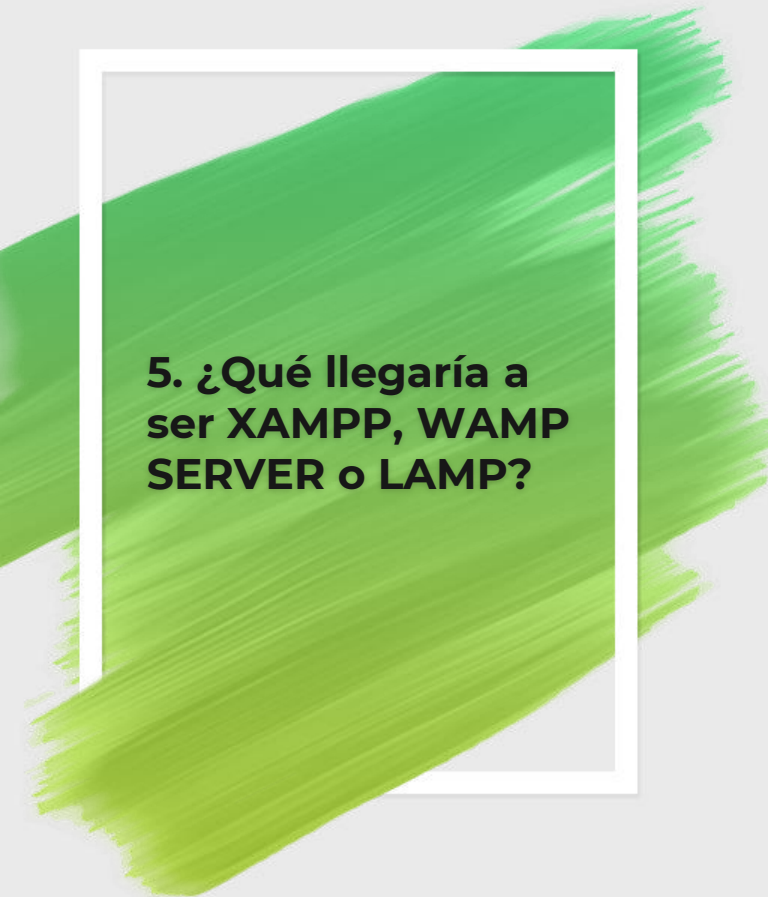
Tanto MySQL como MariaDB son gestores de datos relacionales, pero MariaDB es una bifurcación o copia de MySQL.

Estos gestores de datos son muy similares pero tienen algunas diferencias por ejemplo MariaDB contiene motores de almacenamiento adicionales, mejoras en seguridad, MariaDB saca versiones más frecuentes que MySQL.



4. ¿Qué son las funciones de agregación?

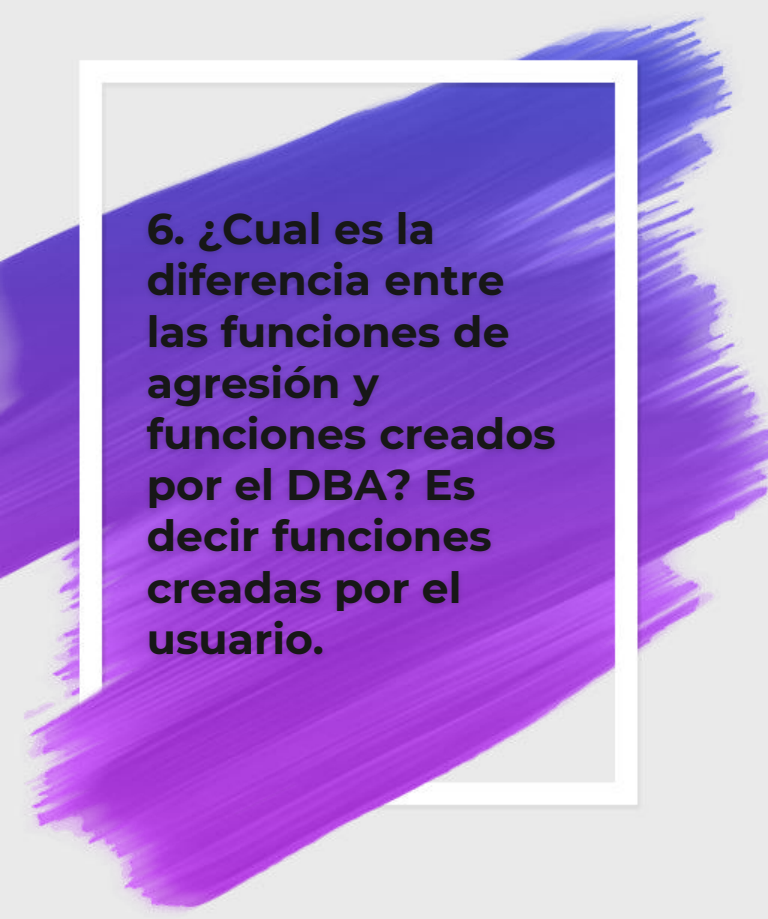
Son funciones que se utiliza en la base de datos para realizar cálculos en algún conjunto de datos el cual retorna un único valor el cual es el resultado de la función.

A large, abstract green brushstroke graphic that starts from the left edge and extends towards the center of the slide. It has a white rectangular frame around its left and bottom portions.

5. ¿Qué llegaría a ser XAMPP, WAMP SERVER o LAMP?

XAMPP - WAMP SERVER – LAMP

Son herramientas que se utilizan para probar o depurar aplicaciones antes de publicarse, esto les sirve a los desarrolladores para concentrarse en la creación de aplicaciones web sin tener que preocuparse por la configuración del servidor. Al proporcionar un conjunto completo de componentes en un solo paquete.



6. ¿Cual es la diferencia entre las funciones de agresión y funciones creados por el DBA? Es decir funciones creadas por el usuario.

Funciones de agregación

Estas funciones vienen integradas en el sistema de gestión de base de datos en donde se los utiliza para hacer cálculos básicos.

Funciones creados por el DBA

Estas son creadas por el usuario donde son personalizadas y permiten a los usuarios realizar operaciones mas complejas.

7. ¿Para qué sirve el comando USE?

El comando USE se lo utiliza para escoger una base de datos en específico y usarla y así poder trabajar en esa base de datos en específico.

8. Que es DML y DDL?

DML: este se lo utiliza para la manipulación de la base de datos: **SELECT – INSERT – UPDATE – DELETE**

DDL: este se lo utiliza para la definir la estructura y características de al base de datos en si:
CREATE – ALTER – DROP

9. ¿Qué cosas características debe de tener una función? Explique sobre el nombre, el return, parametros, etc.

```
create function nombre(datos de entrada)
returns tipo variable que retornara retorno
begin
    proceso de la duncion
    return retornar el resultado ;
end;
```

Nombre de la función: esta debe ser el nombre que el usuario decidido darle con respecto al funcionamiento de la función.

Entrada de datos o parámetros: son aquellos datos que ingresan a la función desde el exterior para que la función se active.

Returns: es el tipo de dato que se retornara ya sea una cadena de caracteres, numéricos, bool.

Proceso: en ese parte se debe poner todo lo que ara la función para calcular un resultado final.

Return: en esta parte se retornara el resultado obtenido en el proceso.

10.¿Cómo crear, modificar y cómo eliminar una función?

para crear una función se debe utilizar el siguiente

```
create function nombre(datos de entrada)
returns tipo variable que retornara retorno
begin
    proceso de la duncion
    return retornar el resultado ;
end;
```

Para modificar una función

```
alter function nombre(datos de entrada)
returns tipo variable que retornara retorno
begin
    proceso de la duncion
    return retornar el resultado ;
end;
```

Para Eliminar una función

```
drop function nombre de la funcion ;
```

The background features two broad, horizontal brushstrokes. The top stroke is a vibrant blue, and the bottom stroke is a rich purple. They overlap, with the purple stroke appearing in front of the blue one. The edges of the strokes are rough and textured, resembling paint applied with a brush. The entire composition is set against a light gray background.

PARTE PRACTICA

11. Crear las tablas y 2 registros para cada tabla para el siguiente modelo ER

```
create database POLLOS_COPA;  
use POLLOS_COPA;
```

```
create table cliente(  
  
    id_cliente integer auto_increment primary key not null ,  
    fullname varchar(50) not null,  
    lastname varchar(50) not null,  
    edad integer not null ,  
    domicilio varchar(100) not null
```

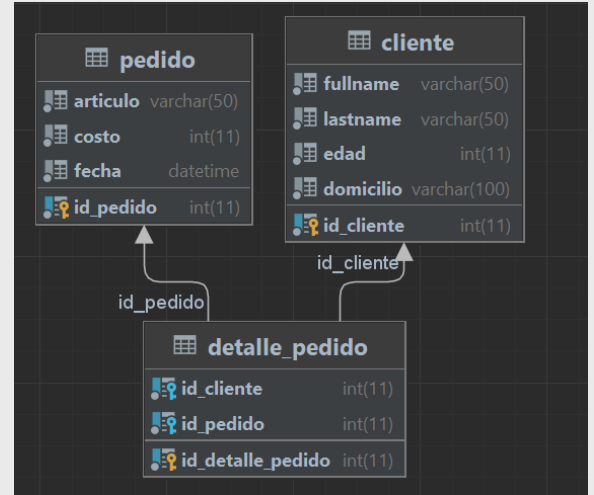
```
);
```

```
create table pedido(  
  
    id_pedido integer auto_increment primary key not null ,  
    articulo varchar (50) not null ,  
    costo integer not null,  
    fecha datetime not null
```

```
);
```

```
create table detalle_pedido(  
  
    id_detalle_pedido integer auto_increment primary key not null ,  
    id_cliente integer not null ,  
    id_pedido integer not null ,
```

```
foreign key (id_cliente) references cliente(id_cliente),  
foreign key (id_pedido) references pedido(id_pedido)  
);
```



```
insert into  
    cliente (fullname, lastname, edad, domicilio)  
values ('Juan De Dios', 'Apaza Arcalia', 20, 'El Alto-Extranca-  
calle 1-702'),  
        ('jhonny', 'Usquiano mamani', 19, 'El Alto-16 de julio -  
call 3-456');
```

```
insert into  
    pedido (articulo, costo, fecha)  
values ('Pollo broaster', 25, '03/10/2023'),  
        ('hamburguesa', 15, '03/10/2023');
```

```
insert into  
    detalle_pedido(id_cliente, id_pedido)  
values (1, 2),  
        (2, 1);
```

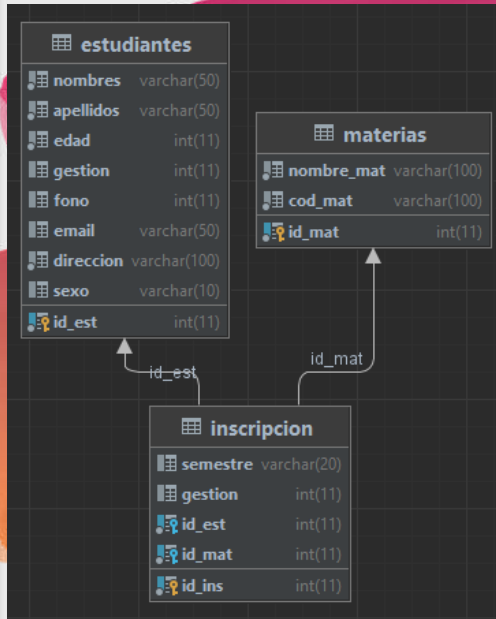
12.Crear una consulta SQL en base al ejercicio anterior

La consulta nos muestra el nombre, apellido, domicilio del cliente también nos muestra el articulo que compro y su precio.

```
select c.fullname,c.lastname,c.domicilio,p.articulo,p.costo
      from detalle_pedido dp
      join cliente c on c.id_cliente = dp.id_cliente
      join pedido p on p.id_pedido = dp.id_pedido;
```

	fullname	lastname	domicilio	articulo	costo
1	jhonny	Usquiano mamani	El Alto-16 de julio -call 3-456	Pollo broaster	25
2	Juan De Dios	Apaza Arcalla	El Alto-Extranca-calle 1-782	hamburguesa	15

BASE DE DATOS QUE SE UTILIZARA PARA LOS SIGUIENTES EJERCICIOS



```
INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion, sexo)
VALUES
```

```
('Miguel', 'Gonzales Veliz', 20, 2832115, 'miguel@gmail.com', 'Av. 6 de Agosto', 'masculino'),
('Sandra', 'Mavir Uria', 25, 2832116, 'sandra@gmail.com', 'Av. 6 de Agosto', 'femenino'),
('Joel', 'Adubiri Mondar', 30, 2832117, 'joel@gmail.com', 'Av. 6 de Agosto', 'masculino'),
('Andrea', 'Arias Ballesteros', 21, 2832118, 'andrea@gmail.com', 'Av. 6 de Agosto', 'femenino'),
('Santos', 'Montes Valenzuela', 24, 2832119, 'santos@gmail.com', 'Av. 6 de Agosto', 'masculino');
```

```
INSERT INTO materias (nombre_mat, cod_mat)
VALUES
```

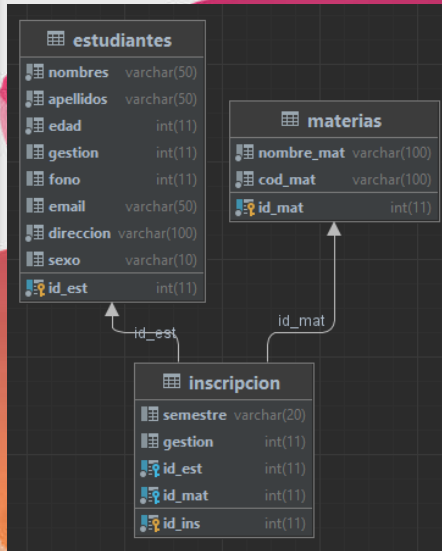
```
('Introduccion a la Arquitectura', 'ARQ-101'),
('Urbanismo y Diseno', 'ARQ-102'),
('Dibujo y Pintura Arquitectonico', 'ARQ-103'),
('Matematica discreta', 'ARQ-104'),
('Fisica Basica', 'ARQ-105');
```

```
INSERT INTO inscripcion (id_est, id_mat,
semestre, gestion)
```

```
VALUES
```

```
(1, 1, '1er Semestre', 2018),
(1, 2, '2do Semestre', 2018),
(2, 4, '1er Semestre', 2019),
(2, 3, '2do Semestre', 2019),
(3, 3, '2do Semestre', 2020),
(3, 1, '3er Semestre', 2020),
(4, 4, '4to Semestre', 2021),
(5, 5, '5to Semestre', 2021);
```

13. Crear un función que compare dos códigos de materia.



```
select est.nombres, est.apellidos, mat.nombre_mat
from inscripcion ins
join estudiantes est on est.id_est = ins.id_est
join materias mat on mat.id_mat = ins.id_mat
where mat.cod_mat = 'ARQ-105';
```

```
create or replace function comparaMaterias( comp varchar(100), entr varchar(100))
returns bool
begin
    declare resp bool default false;
    if comp = entr then
        set resp = true;
    else
        set resp = false;
    end if;
    return resp;
end;
```

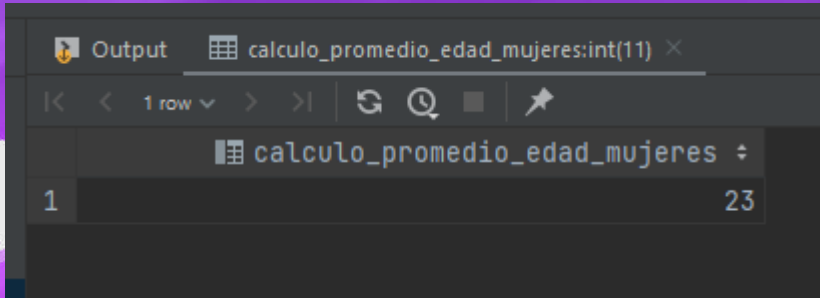
```
select est.id_est, est.nombres, est.apellidos, mat.nombre_mat, est.sexo, est.edad
from inscripcion ins
join estudiantes est on est.id_est = ins.id_est
join materias mat on mat.id_mat = ins.id_mat
where comparaMaterias(mat.cod_mat, 'ARQ-104');
```

	id_est	nombres	apellidos	nombre_mat	sexo	edad
1	5	Santos	Montes Valenzuela	Fisica Basica	masculino	24

14. Crear una función que permita obtener el promedio de las edades del género masculino o femenino de los estudiantes inscritos en la asignatura ARQ-104.

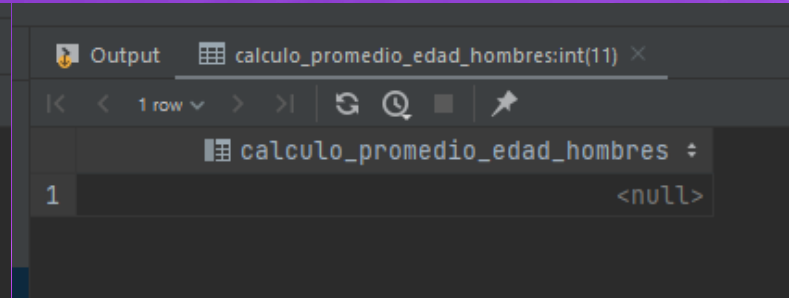
```
create or replace function calculopromedio( genero varchar(10), cod_materia varchar(100))
returns integer
begin
    return (select avg(est.edad)
            from estudiantes est
            join inscripcion i on est.id_est = i.id_est
            join materias m on m.id_mat = i.id_mat
            where est.sexo= genero and m.cod_mat= cod_materia) ;
end;

select calculopromedio('femenino', 'ARQ-104') as calculo_promedio_edad_mujeres;
select calculopromedio('masculino', 'ARQ-104') as calculo_promedio_edad_hombres
```



Output calculo_promedio_edad_mujeres:int(11) ×

1	23
---	----



Output calculo_promedio_edad_hombres:int(11) ×

1	<null>
---	--------

15. Crear una función que permita concatenar 3 cadenas.

```
create or replace function nombresdeestudaintes(nombre varchar(100), apellido varchar(100), edad integer)
returns varchar (100)
begin
    return (select concat('(',est.nombres,'),(',est.apellidos,'),(', est.edad,')') as si
            from estudiantes est where est.nombres = nombre and est.apellidos = apellido and est.edad = edad);
end;

select nombresdeestudaintes(est.nombres,est.apellidos,est.edad) from estudiantes est;
```

```
1 `nombresdeestudaintes(est.nombres,est.apellidos,est.edad)`
1 (Miguel),(Gonzales Veliz),(20)
2 (Sandra),(Mavir Uria),(25)
3 (Joel),(Adubiri Mondar),(30)
4 (Andrea),(Arias Ballesteros),(21)
5 (Santos),(Montes Valenzuela),(24)
```

16. Crear la siguiente VISTA:





- La vista deberá llamarse ARQUITECTURA_DIA_LIBRE
- El día viernes tendrán libre los estudiantes de la carrera de ARQUITECTURA debido a su aniversario
- Este permiso es solo para aquellos estudiantes inscritos en el año 2021.
- La vista deberá tener los siguientes campos.
 1. Nombres y apellidos concatenados = FULLNAME
 2. La edad del estudiante = EDAD
 3. El año de inscripción = GESTION
 4. Generar una columna de nombre DIA_LIBRE
 - a. Si tiene libre mostrar LIBRE
 - b. Caso contrario mostrar NO LIBRE

```
create or replace view ARQUITECTURA_DIA_LIBRE as

select
    concat(est.nombres,' ',est.apellidos) as FULLNAME,
    est.edad as EDAD,
    ins.gestion as GESTION,
    ( case
        when ins.gestion = 2021 then 'LIBRE'
        else 'NO LIBRE'
      end
    ) as DIA_LIBRE
from estudiantes est
join inscripcion ins on ins.id_est = est.id_est;

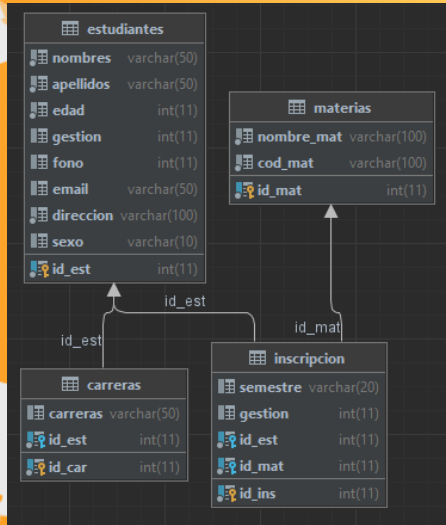
select * from arquitectura_dia_libre;
```

arquitectura_dia_libre	
 FULLNAME	varchar(101)
 EDAD	int(11)
 GESTION	int(11)
 DIA_LIBRE	varchar(9)

	 FULLNAME	 EDAD	 GESTION	 DIA_LIBRE
1	Miguel-Gonzales Veliz	20	2018	NO LIBRE
2	Miguel-Gonzales Veliz	20	2018	NO LIBRE
3	Sandra-Mavir Uria	25	2019	NO LIBRE
4	Sandra-Mavir Uria	25	2019	NO LIBRE
5	Joel-Adubiri Mondar	30	2020	NO LIBRE
6	Joel-Adubiri Mondar	30	2020	NO LIBRE
7	Andrea-Arias Ballesteros	21	2021	LIBRE
8	Santos-Montes Valenzuela	24	2021	LIBRE

17. Crear la siguiente VISTA:

- Agregar una tabla cualquiera al modelo de base de datos.
- Después generar una vista que maneje las 4 tablas
- La vista deberá llamarse **PARALELO_DBA_I**



```
create table carreras (  
    id_car int auto_increment primary key not null ,  
    carreras varchar(50),  
    id_est int not null,  
  
    foreign key (id_est) references estudiantes (id_est)  
);
```

```
insert into carreras (carreras, id_est)  
VALUES ('Arquitectura',1),  
       ('Arquitectura',2),  
       ('Arquitectura',3),  
       ('Matematica',4),  
       ('Ingeniero Civil',5);
```

La siguiente vista muestra datos del estudiante(Nombre,apellido,edad) concatenados, muestra las materias que cursan, los semestres del mismo y a la carrera que pertenecen.

```
create or replace view Paralelo_DBA_I as
```

```
select nombresdeestudaintes(e.nombres,e.apellidos,e.edad) AS  
DATOS_DEL_ESTUDIANTE, i.semestre, m.nombre_mat,car.carreras  
FROM carreras car  
join estudiantes e on e.id_est = car.id_est  
join inscripcion i on i.id_est = e.id_est  
join materias m on m.id_mat = i.id_mat  
where e.id_est= 2 or e.id_est=4;
```

```
select * from Paralelo_DBA_I;
```

DATOS_DEL_ESTUDIANTE	semestre	nombre_mat	carreras
1 (Sandra),(Mavir Uria),(25)	1er Semestre	Matematica discreta	Arquitectura
2 (Sandra),(Mavir Uria),(25)	2do Semestre	Dibujo y Pintura Arquitectonico	Arquitectura
3 (Andrea),(Arias Ballesteros),(21)	4to Semestre	Matematica discreta	Matematica



GRACIAS POR SU ATENCION!!