

1. Manejo de consultas.

Mostrar el título del libro, los nombres, CI de los usuarios que se prestaron libros donde la categoría sea COMEDIA o MANGA.

Resultado esperado:

```
select us.fullname,us.ci,b.title
from prestamos as p
  inner join users us on p.id_user = us.id_user
  inner join book b on p.id_book = b.id_book
  inner join category c on b.id_book = c.id_book
where c.type = 'COMEDIA' OR C.type = 'MANGA';
```

2. Manejo de Funciones.

Se desea saber cuántos usuarios se prestaron libros que correspondan a la editorial IBRANI y que la cantidad de sus páginas sea mayor a 90.

Deberá de crear una función que retorne esa cantidad.

La función debe de recibir 2 parámetros.

Editorial y la cantidad de páginas.

```
CREATE FUNCTION UsserIbrani(EditorialBook varchar(100), pagesbook int)
RETURNS integer
BEGIN
  DECLARE usuario integer;
  select count(p.id_prestamo) INTO usuario
  from prestamos as p
    inner join book b on p.id_book = b.id_book
    inner join users u on p.id_user = u.id_user
    where editorial = EditorialBook AND b.pages > pagesbook;
  return usuario;
END;

select UsserIbrani('IBRANI',90);
select Mostrar_cantidad( 'IBRANI' , 90) as IBRANI_90;
```

3. Manejo de Funciones

Se desea saber qué libros tienen una cantidad de páginas par de la categoría MANGA.

- Deberá de crear una función que verifique si es par o no y retornar un TEXT que indique PAR o IMPAR adicionalmente concatenado el numero de paginas.
- Esta función recibe como parámetro la cantidad de páginas.
- La función deberá ser usada en la cláusula SELECT.
- Deberá de crear una función que concatena cadenas.
- Se tiene el siguiente comportamiento esperado. (Debe de usar los mismos alias que de la imagen)

```

create function verificaPar(pagesBook Integer) returns TEXT
begin
    declare respuesta TEXT DEFAULT '';

    IF pagesBook % 2=0
    then
        set respuesta = CONCAT('PAR : ',pagesBook);
    else
        set respuesta = CONCAT('impar : ', pagesBook);
    end if;
    return respuesta;
end;

select CONCAT('EDITORIAL : ',bk.editorial,'Categoria : ',cat.type) as
DESCRIPCION,
        verificaPar(bk.pages)
from book as bk
INNER JOIN category as cat on bk.id_book = cat.id_book
where cat.type = 'MANGA';

```

4. Manejo de Funciones

Crear una función que pueda eliminar, modificar o insertar un registro de la tabla AUTOR.

La función recibe 4 parámetros. deleteOrModifyOrInsertTableAutor('delete', 6, 'autor_name_6', 'NICARAGUA')

- El primer parámetro es la acción a realizar (puede ser delete - insert - update).
- El segundo parámetro es el ID del autor
- El tercer parámetro es el nombre del autor.
- El cuarto parámetro es la nacionalidad del autor.
- Si la acción es delete, debe de eliminar el registro con el ID que llega en el 2do parámetro.
- Si la acción es update deberá de modificar el registro. (En este caso se utilizarán todos los parámetros)
- Si la acción es insert deberá de crear un nuevo registro en la tabla autor.
- Comportamiento esperado

```

create function deleteOrModifyOrInsertTableAutor
(action TEXT, idAutor INTEGER, nameAutor TEXT,nacionalidadAutor TEXT)
returns TEXT
BEGIN
    if action = 'delete'
    then
        delete from autor where id_autor= idAutor;
    end if;
    if action = 'update'
    then
        update autor SET name = nameAutor;
    end if;
end;

```

```

end if;
if action ='insert'
then
    Insert INTO autor(name,nacionality) values ( nameAutor,nacionality);
end if;
return CONCAT(action, ' action realizado exitosamente' );
end;

```

5. Manejo de Funciones

Se desea saber cuántos libros fueron prestados en la gestión 2018.

- Crear una función que permita saber esa cantidad.
- Comportamiento esperado.

```

select COUNT(PRE.id_prestamo) AS CANTIDAD_2018
FROM prestamos as pre
where YEAR(pre.fec_prestamo)= 2018

```