



Examen Resuelto Hito 2

- **Nombre Completo:** Diego Emiliano Rivera Tapia
- **Asignatura:** BASE DE DATOS II
- **Carrera:** INGENIERÍA DE SISTEMAS
- **Paralelo:** BDA (1)
- **Docente:** Lic. William R. Barra Paredes
- **Fecha:** 10/09/2019

COCHABAMBA-BOLIVIA

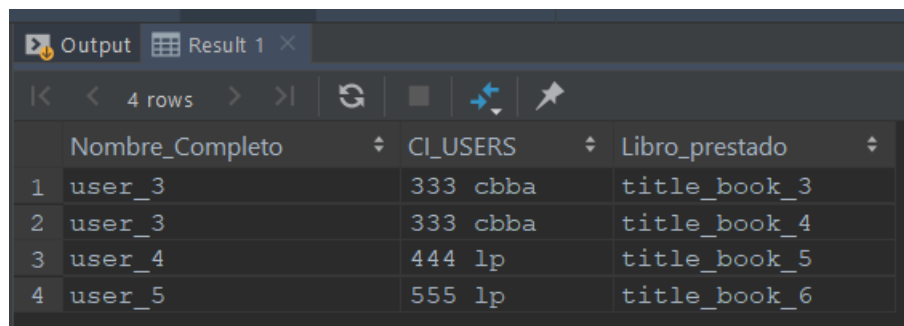
2019

Ejercicio 1: Mostrar el título del libro, los nombres, CI de los usuarios que se prestaron libros donde la categoría sea **COMEDIA** o **MANGA**.

```
Select fullname as Nombre_Completo,ci as CI_USERS, bok.title as Libro_prestado
from users
INNER JOIN prestamos as pre on users.id_user = pre.id_user
INNER JOIN book as bok on pre.id_book = bok.id_book
INNER JOIN category as cat on cat.id_book = bok.id_book
where cat.type = 'manga' or cat.type = 'comedia';
```

drop function *Mostrar_cantidad*;

Solución:



The screenshot shows a database interface with a table titled 'Result 1'. The table has three columns: 'Nombre_Completo', 'CI_USERS', and 'Libro_prestado'. There are 4 rows of data displayed.

	Nombre_Completo	CI_USERS	Libro_prestado
1	user_3	333 cbba	title_book_3
2	user_3	333 cbba	title_book_4
3	user_4	444 lp	title_book_5
4	user_5	555 lp	title_book_6

Ejercicio 2: 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 funcion que retorne esa cantidad.
- La función debe de recibir 2 parámetros.
 - Editorial y la cantidad de páginas.

```
CREATE FUNCTION Mostrar_cantidad( editorialbook text,pagebook integer) RETURNS integer
BEGIN
declare respuesta INTEGER default 0;

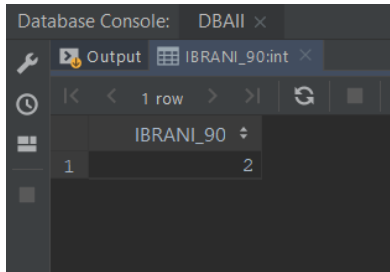
select COUNT(pre.id_prestamo) into respuesta
from prestamos as pre
inner join book as bk on pre.id_book = bk.id_book
where bk.editorial= editorialbook and bk.pages > pagebook;

return respuesta;

end;

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

Solución:



Ejercicio 3: Se desea saber qué libros tienen una cantidad de páginas par de la categoría MANGA.

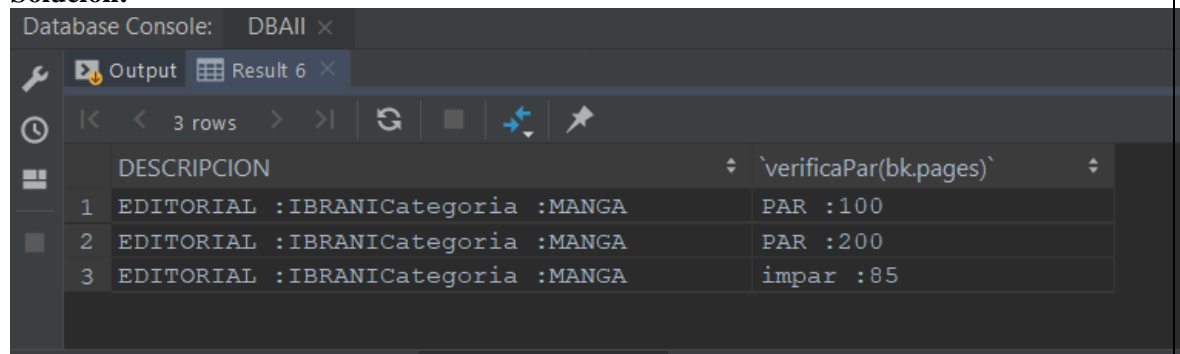
- Deberá de crear una función que verifique si es par o no y retornar un TEXT que indique PAR o IMPAR adicionalmente concatenado el numero de paginas.
 - Esta función recibe como parámetro la cantidad de páginas.
- La función deberá ser usada en la cláusula SELECT.
- Deberá de crear una función que concatena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

```
CREATE FUNCTION Mostrar_cantidad( editorialbook text,pagebook integer) RETURNS
integer
BEGIN
  declare respuesta INTEGER default 0;

  select COUNT(pre.id_prestamo) into respuesta
  from prestamos as pre
  inner join book as bk on pre.id_book = bk.id_book
  where bk.editorial= editorialbook and bk.pages > pagebook;

  return respuesta;
end;
```

Solución:



Ejercicio 4: 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 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';
```

Ejercicio 5: 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;
  if action ='insert'
  then
    Insert INTO autor(name,nacionality) values( nameAutor,nacionality);
  end if;
  return CONCAT(action, ' action realizado exitosamente' );
end;
```

SOLUCION:

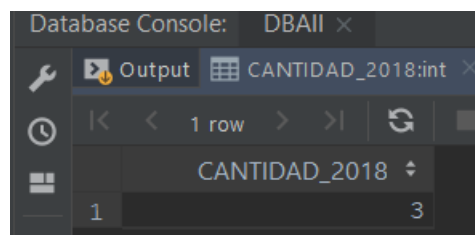
```
1 delete action realizado exitosamente
```

Ejercicio 6: 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;
```

Solución:



CANTIDAD_2018	
1	3