

**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:** |

**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 categoria 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:** |

**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:** |