

Manejo de Datos 2022-1

Tarea-Examen 3

Nombre: José Roberto Torres Bello

No. Cuenta: 305746148

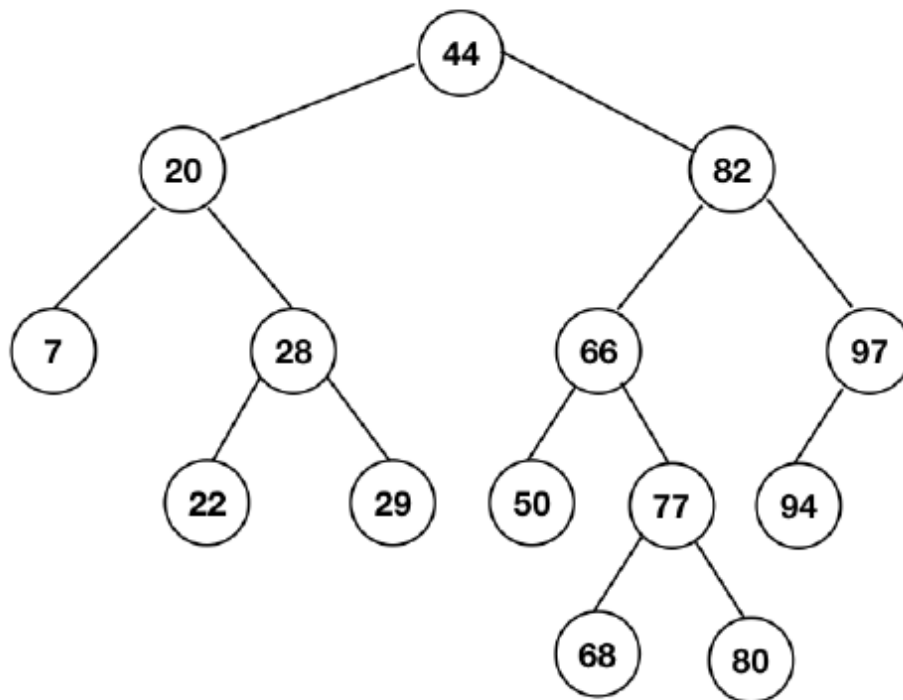


Figura 1: Árbol Binario de Búsqueda.

1. Dado el árbol binario de búsqueda de la Figura 1. Muestra el procedimiento de eliminar los nodos 82, 29 y 44 en ese orden.

Cuando el nodo que se va a eliminar es un nodo de hoja, simplemente se elimina el nodo.

Cuando un nodo que se va a eliminar tiene solo un hijo, copiamos el valor hijo en el valor del nodo y eliminamos el hijo.

Cuando un nodo que se va a eliminar tiene dos hijos, el mínimo del subárbol derecho se puede copiar al nodo y el valor mínimo se puede eliminar del subárbol derecho del nodo (Convertido al Caso 2)

El 82 es un nodo que tiene sus dos hijos por lo tanto iniciamos en el 44 vamos hacia la derecha encontramos el 82 se elimina y el 77 pasa al lugar del 82 así el 77 tiene como nodos hijos al 66 y 97.

Para borrar el 29 no tamos que no tiene nodos hijos (o en este caso nodos hojas) iniciamos en el 44, luego al 20, luego al 28 y al final 29 y se elimina.

El nodo para eliminar 44 tiene dos hijos el 20 y 82

2. Sabemos que si un árbol binario de búsqueda se recorre en inorden obtenemos una lista ordenada de nodos. ¿Cual es la complejidad de ordenar una secuencia de n números creando un árbol binario de búsqueda?

Los árboles binarios de búsqueda (ABB) radican en que su recorrido en in orden proporciona los elementos ordenados de forma ascendente y en que la búsqueda de algún elemento suele ser muy eficiente.

La altura h en el peor de los casos es siempre el mismo tamaño que el número de elementos disponibles. Y en el mejor de los casos viene dada por la expresión $h = \log_2(c+1)$

3. Describe los niveles de abstracción de los datos vistos en clase.

- **TEXT:** se usa para almacenar cadenas de caracteres. Una cadena es una secuencia de caracteres. Se coloca entre comillas (simples); ejemplo: 'Roberto'. El tipo TEXT define una cadena de longitud variable.
- **INTEGER:** se usa para guardar valores numéricos enteros. Definimos campos de este tipo cuando queremos representar, por ejemplo, cantidades.
- **REAL:** se usa para almacenar valores numéricos con decimales. Se utiliza como separador el punto (.). Definimos campos de este tipo para precios, por ejemplo.
- **BLOB:** se usa para almacenar valores en formato binario (imágenes, archivos de sonido etc.)

BLOB
INTEGER
NUMERIC
REAL
TEXT

4. ¿Cuáles son los tipos de dominio básicos de SQL y qué representa cada uno?

- **char(n).** Longitud fija cadena de caracteres, con una longitud especificada por el usuario n .
- **varchar(n).** Variable de cadenas de caracteres de longitud, con n longitud máxima especificada por el usuario.
- **int.** Número de tipo entero.

- **smallint.** entero pequeño (un subconjunto dependiente de la máquina del tipo de dominio entero).
- **numeric(p,d).** Número de punto fijo, con la precisión especificada por el usuario de p dígitos, con n dígitos a la derecha del punto decimal.
- **float(n).** Número de punto flotante, con la precisión especificada por el usuario de n dígitos al menos.
- **Date.** Se usa para almacenar información de una fecha. Se representa en formato día, mes y año (dd/mm/aaaa).
- **NULL** No se trata de un tipo de datos concreto. Es más bien un valor que indica la ausencia de cualquier valor. Para comparar valores con NULL sólo podremos usar los operadores IS NULL o IS NOT NULL.

5. Describe el concepto de Modelo de Datos y las partes que lo conforman.

Es una colección de herramientas conceptuales para describir los datos, sus relaciones, su semántica y las restricciones de consistencia, se conforma de las siguientes partes:

- Instancia (Los datos)
- Esquema (Descripción de los datos almacenados)
- Lenguaje de consulta (Herramienta para acceder y manipular los datos)

6. Describe el concepto de llave y los tipos de llaves vistos en clase.

La llave no permite que dos registros tengan exactamente los mismos valores, es una forma de distinguir los registros de una relación entre si.

La clave o llave primaria es un campo, o grupo de campos que identifica en forma única un registro. Ningún otro registro puede tener la misma llave primaria.

Superllave: Es un conjunto de uno o varios atributos, que considerados conjuntamente, permiten identificar de manera unívoca un registro de la relación.

Llave candidata: son superllaves mínimas.

6. Describe el concepto de llave y los tipos de llaves vistos en clase.

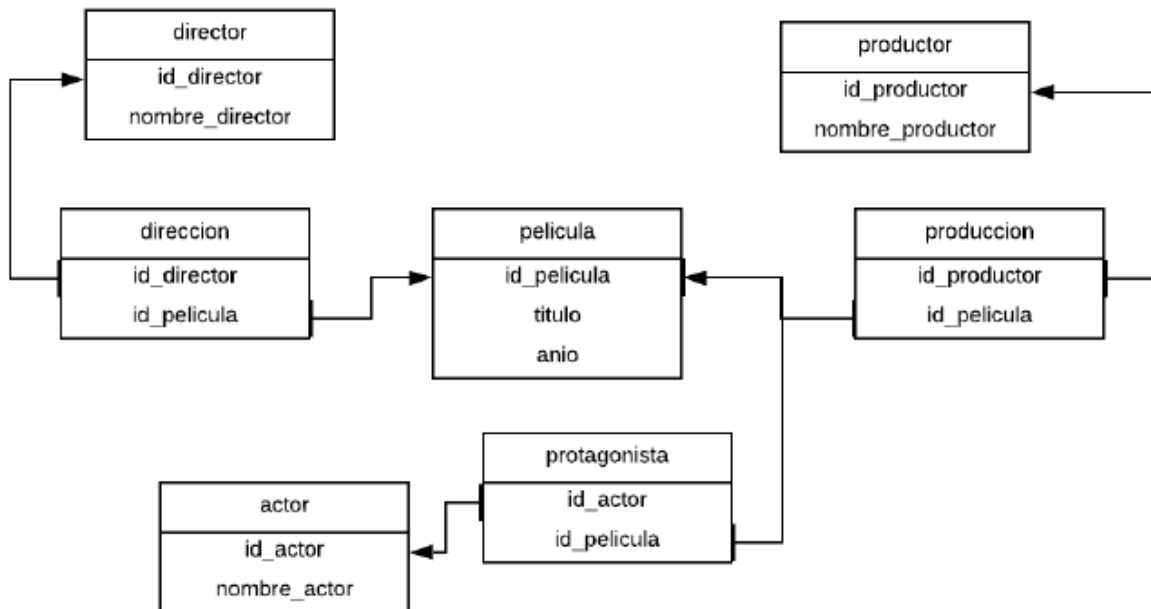


Figura 2: Diagrama de base de datos.

7. Dado el Diagrama de la Figura 2, escribe las consultas SQL siguientes:

- Elimina todos los actores que protagonizaron la película *Iron Man*.
- Obtener el título y año de estreno de las películas dirigidas por Joss Whedon.
- Obtener el título y productor(es) de las películas protagonizadas por Robert Downey Jr. y Tom Holland (ambos actores en una misma película).

a)

```
select nombre_actor
from actor A
left join protagonista P
on A.id_actor = P.id_actor
left join pelicula PE
on PE.id_pelicula = P.id_pelicula
where PE.titulo = "Iron Man"
```

b)

```
select A.id_actor, A.nombre_actor
from actor A
left join protagonista P
on A.id_actor = P.id_actor
left join pelicula PE
on PE.id_pelicula = P.id_pelicula
where PE.titulo = "Joss Whedon"
```

c)

```
select P.titulo, PROD.nombre_productor
from actor A
left join protagonista P
on A.id_actor = P.id_actor
left join pelicula PE
on PE.id_pelicula = P.id_pelicula
left join produccion PR
on PR.id_pelicula = PE.id_pelicula
left join productor PROD
on PROD.id_prodcutor=PR.id_prodcutor
where A.nombre_actor = "Downey Jr" and A.nombre_actor = "Tom Holland"
```

8. Supon que tienes un archivo de texto `movies.txt` que contiene información de películas, la información que se tiene es: título, año de estreno, género, protagonista(s), director(es).

```
titulo|anio|genero(s)|protagonista(s)|director(es)
The Godfather|1972|Drama, Crimen|Marlon Brando, Al Pacino|Francis Ford Coppola
The Dark Knight|2008|Accion, Crimen, Drama|Christian Bale, Heath Ledger, Aaron Eckhart|Christopher Nolan
Pulp Fiction|1994|Crimen, Drama|John Travolta, Uma Thurman, Samuel L. Jackson|Quentin Tarantino
Fight Club|1999|Drama|Brad Pitt, Edward Norton, Meat Loaf
```

Describe una función que lea los datos del archivo y cree un archivo JSON que contenga la información de cada renglón en un objeto JSON. ¿Como se ve el archivo JSON con los datos de las películas mostradas en el cuadro anterior?



Figura 3: Árbol XML.

Paso 1: Leer el fichero en formato txt

Paso 2: Separar los pipes “|”

Paso 3: Crear una lista de Python donde cada elemento sea un libro que se compone de un diccionario don de las llaves sean titulo, año, genero(s), protagonista(s) y directores(es), cada llave con su respectivo valor por ejemplo:

```
"titulo": "The Godfather",  
"anio": 1972,  
"genero(s)": "Drama, Crimen",  
"protagonista(s)": "Marlon Brando, Al Pacino",  
"director(es)": "Francis Ford Coppola"
```

Paso 4: Utilizar la librería de json para guardar el archivo con dumps.

9. Dado el árbol de la Figura 3. Crea un archivo XML que conserve la estructura y contenga los datos del árbol.

```
<?xml version="1.0" encoding="UTF-8" ?>  
  
<root>  
  
  <row>  
  
    <titulo>The Godfather</titulo>  
  
    <anio>1972</anio>  
  
    <genero(s)>Drama, Crimen</genero(s)>  
  
    <protagonista(s)>Marlon Brando, Al Pacino</protagonista(s)>  
  
    <director(es)>Francis Ford Coppola</director(es)>  
  
  </row>  
  
  <row>  
  
    <titulo>The Dark Knight</titulo>  
  
    <anio>2008</anio>  
  
    <genero(s)>Accion, Crimen, Drama</genero(s)>  
  
    <protagonista(s)>Christian Bale, Heath Ledger, Aaron Eckhart</protagonista(s)>  
  
    <director(es)>Christopher Nolan</director(es)>  
  
  </row>  
  
  <row>  
  
    <titulo>Pulp Fiction</titulo>  
  
    <anio>1994</anio>
```

```

<genero(s)>Crimen, Drama</genero(s)>
<protagonista(s)>John Travolta, Uma Thurman, Samuel L. Jackson</protagonista(s)>
<director(es)>Quentin Tarantino</director(es)>
</row>
<row>
<titulo>Fight Club</titulo>
<anio>1999</anio>
<genero(s)>Drama</genero(s)>
<protagonista(s)>Brad Pitt, Edward Norton, Meat Loaf</protagonista(s)>
<director(es)/>
</row>
</root>

```

10. Dado el árbol de la Figura 3. Crea un archivo JSON que conserve la estructura y contenga los datos del árbol.

```

[
  {
    "titulo": "The Godfather",
    "anio": 1972,
    "genero(s)": "Drama, Crimen",
    "protagonista(s)": "Marlon Brando, Al Pacino",
    "director(es)": "Francis Ford Coppola"
  },
  {
    "titulo": "The Dark Knight",
    "anio": 2008,
    "genero(s)": "Accion, Crimen, Drama",
    "protagonista(s)": "Christian Bale, Heath Ledger, Aaron Eckhart",

```



```
    "director(es)": "Christopher Nolan"
  },
  {
    "titulo": "Pulp Fiction",
    "anio": 1994,
    "genero(s)": "Crimen, Drama",
    "protagonista(s)": "John Travolta, Uma Thurman, Samuel L. Jackson",
    "director(es)": "Quentin Tarantino"
  },
  {
    "titulo": "Fight Club",
    "anio": 1999,
    "genero(s)": "Drama",
    "protagonista(s)": "Brad Pitt, Edward Norton, Meat Loaf",
    "director(es)": null
  }
]
```