

TRABAJO OBLIGATORIO

SIMULADOR DE UN MANEJADOR DE BASE DE DATOS

Introducción

Continuando con el trabajo realizado para el obligatorio uno, se pide en esta instancia la finalización de la implementación del manejador agregando la gestión de las tablas.

Se pide que la representación utilizada para el conjunto de las tablas sea un Árbol Binario de Búsqueda (ABB) ordenado de acuerdo al nombre de la tabla. Se pide la implementación de las siguientes operaciones:

Operaciones entre Tablas

SELECCIÓN – `selectWhere (nombreTabla1, condicion, nombreTabla2)`

Descripción: Dado un nombre de tabla `nombreTabla1` y una condición, genera una nueva tabla en la base de datos de nombre `nombreTabla2`, si `nombreTabla1` existe y `nombreTabla2` no existe, con las tuplas de la tabla `nombreTabla1` que cumplan la condición. En caso de que la condición sea “”, la operación tiene aplicación sobre todas las tuplas de la tabla1. La condición respeta el formato descrito anteriormente para condiciones (ver operación delete).

Ejemplo: Copiar la tabla `Personas` a una nueva tabla llamada `Personas2`.

```
selectWhere (Personas, , Personas2)
```

Tabla *Personas2*

Nombre	CI
Telma	3333111
Juan	4232323
Pepe	1555000

TipoRet `selectWhere (char * nombreTabla1, char * condicion, char * nombreTabla2)`

Retornos posibles:	
OK	<ul style="list-style-type: none">• Si se pudo ejecutar exitosamente el comando.
ERROR	<ul style="list-style-type: none">• Si <i>nombreTabla1</i> no existe o no se especifica.• Si <i>nombreTabla2</i> existe.• Si la columna dentro de <i>condicion</i> no pertenece a la tabla <i>nombreTabla1</i> o no se especifica. <p>No se considera error si ninguna tupla cumple la condición. Además se asume que la condición <i>condicion</i> respeta el formato establecido.</p>
NO_IMPLEMENTADA	<ul style="list-style-type: none">• Cuando aún no se implementó. Es el tipo de retorno por defecto.

PROYECCIÓN – `Select (nombreTabla1, columnas, nombreTabla2)`

Descripción: Dado un nombre de tabla `nombreTabla1` y uno o más nombres de columnas de dicha tabla especificados en el parámetro `columnas`, genera una nueva tabla en la base de datos de nombre

nombreTabla2, si nombreTabla1 existe y nombreTabla2 no existe. La tabla nombreTabla2 tendrá las tuplas de la tabla nombreTabla1 sólo con las columnas especificadas, manteniendo el orden establecido en el parámetro columnas. Los nombres de las columnas se separan con el uso del caracter dos puntos (:).

Ejemplo: Copiar la columna CI de la tabla Personas a una nueva tabla llamada Cedula.

```
select (Personas, CI, Cedula)
```

Tabla Cedula

CI
3333111
4232323
1555000

TipoRet select(char * nombreTabla1, char * columnas, char * nombreTabla2)

Retornos posibles:	
OK	<ul style="list-style-type: none"> • Si se pudo ejecutar exitosamente el comando.
ERROR	<ul style="list-style-type: none"> • Si <i>nombreTabla1</i> no existe o no se especifica. • Si <i>nombreTabla2</i> existe. • Si la/s columna/s no existe/n en nombreTabla1 o no se especifica.
NO_IMPLEMENTADA	<ul style="list-style-type: none"> • Cuando aún no se implementó. Es el tipo de retorno por defecto.

JOIN NATURAL – Join (nombreTabla1, nombreTabla2, nombreTabla3)

Descripción: Dadas dos tablas de nombres nombreTabla1 y nombreTabla2, tales que ambas tengan *exactamente una única* columna en común con el mismo nombre y tipo de datos, calificada como PRIMARY KEY, genera una nueva tabla en la base de datos de nombre nombreTabla3, si nombreTabla1 y nombreTabla2 existen y nombreTabla3 no existe. La nueva tabla nombreTabla3 tendrá todas las columnas de la tabla nombreTabla1 y las columnas de nombreTabla2 excepto la que esté calificada como PRIMARY KEY (de modo de no repetirla), en ese orden. Asimismo, nombreTabla3 tendrá exclusivamente las tuplas que satisfagan la igualdad de las claves de ambas tablas.

Ejemplo: El join natural de las tabla1 y tabla2 genera la tabla3:

Nombre	CI
Telma	3333111
Juan	8232323
Pepe	1555000

CI	Cargo
8232323	Ingeniero
7777777	Escribano
3333111	Dentista

CI es clave (PRIMARY KEY) en las tres tablas.

Nombre	CI	Cargo
Telma	3333111	Dentista
Juan	8232323	Ingeniero

TipoRet join(char * nombreTabla1, char * nombreTabla2, char * nombreTabla3)

Retornos posibles:	
OK	<ul style="list-style-type: none"> • Si se pudo ejecutar exitosamente el comando.
ERROR	<ul style="list-style-type: none"> • Si <i>nombreTabla1</i> no existe o no se especifica. • Si <i>nombreTabla2</i> no existe o no se especifica. • Si <i>nombreTabla3</i> existe. • Si no existe una columna en común entre los <i>nombreTabla1</i> y <i>nombreTabla2</i>. • Si existe más de una columna en común entre <i>nombreTabla1</i> y <i>nombreTabla2</i> <p>No se considera error si ninguna tupla cumple la condición.</p>
NO_IMPLEMENTADA	<ul style="list-style-type: none"> • Cuando aún no se implementó. Es el tipo de retorno por defecto.

Operaciones para la Impresión de Información

LISTAR TABLAS – printTables()

Descripción: Imprime los nombres de las tablas de la base de datos del sistema, ordenados alfabéticamente de menor a mayor.

Ejemplo:

printTables()

TipoRet printTables()

Retornos posibles:	
OK	<ul style="list-style-type: none"> • Si se pudo ejecutar exitosamente el comando.
ERROR	<ul style="list-style-type: none"> • Si no hay tablas para imprimir.
NO_IMPLEMENTADA	<ul style="list-style-type: none"> • Cuando aún no se implementó. Es el tipo de retorno por defecto.

COMANDOS EJECUTADOS ERRÓNEAMENTE

Aunque lo siguiente fue expresado al comienzo de este documento, decidimos sintetizar aquí la política de manejo de errores frente a la ejecución de las operaciones del sistema. Cada operación tiene determinadas precondiciones para que funcione correctamente. En caso de que alguna de estas precondiciones no se cumpla la operación debería retornar ERROR (tal cual se señala para cada operación), imprimiendo además por pantalla un texto breve apropiado a cada caso para destacar el tipo de error ocurrido. En cualquier caso que la ejecución de una operación no sea satisfactoria (retorne ERROR), el estado del sistema deberá permanecer inalterado.

SOBRE LOS CHEQUEOS

Se permite considerar que la sintaxis de la entrada que recibirá el programa es válida. Esto quiere decir que no se requiere la realización de chequeos como los siguientes:

- cadenas de nombres de columnas y valores de tuplas que no respeten el formato establecido (usando el (:) como separador).
- nombres de tablas y columnas que tengan caracteres no permitidos.
- condición en la operación deleteFrom que no respete el formato establecido.

Esto no quiere decir que no se deban chequear las condiciones establecidas para las operaciones en la letra del obligatorio. Por ejemplo, creación de una tabla ya existente, supresión de una columna inexistente, entre otros.

SEGUNDA ENTREGA:

Operaciones obligatorias

selectWhere select join printTables
--

Notas:

- Se debe respetar la modularidad como fue vista en el curso.

FORMAS Y PLAZOS DE ENTREGA:

La tarea deberá realizarse en los **mismos grupos** de la primera parte.

La tarea deberá ser entregada en la plataforma EVA y deberá constar de TODOS los archivos utilizados en el proyecto, comprimidos en una carpeta, dicha carpeta tendrá los apellidos de los integrantes del equipo. **El trabajo deberá ser realizado en ZinjaI versión 20191006.** La fecha límite de entrega de la segunda parte del obligatorio es el lunes 7 de noviembre a las 19:59 horas. No se admitirán trabajos entregados luego de esa fecha y hora salvo expresa y previa autorización de la docente del curso.