

**TECNOLÓGICO DE COSTA RICA**

Ingeniería en Computación

Bases de datos I / Grupo 1

Proyecto 1: INTÉRPRETE DE ÁLGEBRA RELACIONAL.

Gabriel Ramírez Ramírez – 201020244.

Junior Herrera – 2014110993

José Arturo Luna –2014110993

Fecha de entrega: 1 de octubre de 2017.

II Semestre, 2017.

Profesor: William Mata Rodríguez.

Contenido

[Introducción 3](#_Toc495438295)

[Enunciado 4](#_Toc495438296)

[Desarrollo 11](#_Toc495438297)

[Conclusión 12](#_Toc495438298)

# Introducción

El siguiente documento busca detallar de la mejor forma posible el proyecto de Interpretación de Álgebra Relacional.

Dicho proyecto consistió en utilizar los conocimientos obtenidos de álgebra relacional para así poder crear una aplicación de intérprete de la misma, que además de permitirnos realizar algunas operaciones del algebra relacional, también nos ofrece el equivalente a su instrucción en SQL.

Se pretende abarcar características del proceso de elaboración de dicha aplicación conociendo los desafíos que planteaban los requerimientos solicitados por el profesor.

# Enunciado

INTÉRPRETE DE ÁLGEBRA RELACIONAL

Un componente fundamental en los modelos de datos es el lenguaje de manipulación, o lenguaje de consulta, para la extracción y actualización de los datos. Entre estos lenguajes de consulta están el álgebra relacional y el cálculo relacional definidos por Codd (1971). Ambos lenguajes son teóricos y formales. Se han utilizado como base para desarrollar otros lenguajes de manipulación de datos en el modelo relacional como el SQL. Este proyecto consiste en desarrollar un intérprete de algunas operaciones del lenguaje álgebra relacional, ofreciendo también las instrucciones de SQL equivalentes y los datos obtenidos con la operación. Durante el curso estudiamos conceptos de SQL desde el punto de vista del estándar de la ISO. Al usar un manejador de base de datos (DBMS o SABD) de un fabricante particular, podemos encontrar algunas diferencias pues estos han modificado parte del estándar y han agregado sus propias funcionalidades. En tales casos hay que consultar la documentación respectiva para adaptarse a un DBMS específico. El DBMS a utilizar en el proyecto es Microsoft SQL Server o MySQL. Las demás herramientas que necesiten para desarrollar esta aplicación Ustedes las pueden seleccionar. Para interactuar con el usuario la aplicación debe usar una GUI (Graphical User Interface) diseñada por Ustedes que cumpla con los requerimientos solicitados. El interpretador solo hace una operación del álgebra a la vez, no hay operaciones anidadas. El desarrollo del proyecto es en equipos de 3 estudiantes máximo, uno de ellos lo deben nombrar como el coordinador. Importante: las experiencias han demostrado que los proyectos en equipos que no han sido administrados adecuadamente van a fallar, así que en cuanto noten que se presentan problemas al respecto de inmediato trátenlo primeramente con los miembros del equipo, y de no resolver lo comunican al profesor. Cualquier comunicación al profesor que vaya copiada a todos los miembros del equipo. Buenas prácticas de la ingeniería de software: deben usar un software para administrar el desarrollo de proyectos en equipo el cual incluye un control de versiones (Git, otros).

REQUERIMIENTOS DEL PROYECTO

Las operaciones del álgebra relacional que debe manejar el interpretador son:

- Selección

- Proyección generalizada

- Unión

- Diferencia de conjuntos

- Producto cartesiano

- Intersección

- División

- Renombrar un relación y sus atributos

- Concatenación (join)

- Concatenación natural (natural join)

- Agregación

- Agrupación

El usuario debe tener un mecanismo para seleccionar la operación que necesite. El programa debe quedarse pidiendo funciones al usuario hasta que él pida finalizar el programa. En general para cada operación Usted va a encontrar en esta especificación los datos que se deben solicitar al usuario, las validaciones que se deben hacer y los resultados esperados. En las validaciones algunos errores deben tener un manejo de errores personalizado, es decir, se envían al usuario los mensajes tales como se especifican aquí. Cuando no se indiquen mensajes personalizados el manejo de los errores puede hacerlo como Ustedes decidan.

**Operación: Selección**

Datos solicitados: Tabla

Predicado (string)

Tabla resultante (se explica más adelante)

Validaciones:

- Tabla debe existir. Mensaje personalizado cuando no exista: ERROR: NO EXISTE LA TABLA nombre\_de\_tabla

- Otros errores Usted los maneja.

**Operación: proyección generalizada**

Datos solicitados: Tabla

Expresión de la proyección generalizada (string)

Tabla resultante

Validaciones:

- Tabla debe existir. Mensaje personalizado cuando no exista: ERROR: NO EXISTE LA TABLA nombre\_de\_tabla

- Otros errores Usted los maneja.

**Operación: Unión**

Datos solicitados: Tabla 1

Tabla 2

Tabla resultante

Validaciones:

- Tablas deben existir. Mensaje personalizado cuando no existan: ERROR: NO EXISTE LA TABLA nombre\_de\_tabla

- Las tablas deben tener igual aridad. Mensaje personalizado: ERROR: TABLAS CON DIFERENTE ARIDAD. LA TABLA 1 TIENE ARIDAD “X” Y LA TABLA 2 TIENE ARIDAD “Y”

- Cada atributo respectivo debe tener el mismo dominio. Mensaje personalizado: ERROR: DOMINIOS DIFERENTES. EL ATRIBUTO “X” TIENE “DOMINIO X” Y EL ATRIBUTO “Y” TIENE “DOMINIO “Y”.

- Otros errores Usted los maneja.

**Operación: Diferencia de conjuntos**

Datos solicitados: Tabla 1

Tabla 2

Tabla resultante

Validaciones:

- Igual a la operación de unión.

**Operación: Producto cartesiano**

Datos solicitados: Tabla 1

Tabla 2

Tabla resultante

Validaciones:

- Tablas deben existir. Mensaje personalizado cuando no existan: ERROR: NO EXISTE LA TABLA nombre\_de\_tabla

- Otros errores Usted los maneja.

**Operación: Intersección**

Datos solicitados: Tabla 1

Tabla 2

Tabla resultante

Validaciones:

- Igual a la operación de unión.

**Operación: División**

Datos solicitados: Tabla 1

Tabla 2

Tabla resultante

Validaciones:

- Tablas deben existir. Mensaje personalizado cuando no existan: ERROR: NO EXISTE LA TABLA nombre\_de\_tabla

- Todos los atributos de Tabla 2 deben estar en Tabla 1. Mensaje personalizado: ERROR: EL ATRIBUTO “X” DE LA TABLA nombre\_de\_tabla1 NO ESTA EN LA TABLA nombre\_de\_tabla2.

- Otros errores Usted los maneja.

**Operación: Renombrar una relación y sus atributos**

Físicamente en la base de datos tanto la tabla como los atributos siguen manteniendo el nombre de origen. El renombramiento es solo para efectos de conocer esos objetos con otro nombre en la corrida actual de esta aplicación.

Datos solicitados: Tabla

Nombres de atributos (string): Cada uno de los nombres nuevos de los atributos: el i-ésimo atributo dado aquí corresponde al i-ésimo atributo de la tabla.

Tabla resultante.

Validaciones:

- Tabla debe existir. Mensaje personalizado cuando no exista: ERROR: NO EXISTE LA TABLA nombre\_de\_tabla

- Por cada atributo en la tabla debe existir un nombre de atributo. Mensaje de error personalizado: ERROR: NO HAY CORRESPONDENCIA EN LA CANTIDAD DE ATRIBUTOS. LA TABLA nombre\_de\_tabla TIENE “X” ATRIBUTOS Y SE ESTAN DANDO “Y” ATRIBUTOS.

- Otros errores Usted los maneja.

**Operación: concatenación (join)**

Datos solicitados: Tabla 1.

Tabla 2.

Predicado (string): para unir las tablas.

Tabla resultante.

Validaciones:

- Tablas deben existir. Mensaje personalizado cuando no existan: ERROR: NO EXISTE LA TABLA nombre\_de\_tabla

- Otros errores Usted los maneja.

**Operación: concatenación natural(natural join)**

Datos solicitados: Tabla 1.

Tabla 2.

Tabla resultante.

Validaciones:

- Tablas deben existir. Mensaje personalizado cuando no existan: ERROR: NO EXISTE LA TABLA nombre\_de\_tabla

- Para hacer este tipo de join deben existir al menos dos nombres de atributos comunes (y dominios comunes) entre las dos tablas. Mensaje personalizado de error: ERROR: NO HAY ATRIBUTOS COMUNES

- Otros errores Usted los maneja.

**Operación: Agregación**

Datos solicitados: Tabla.

Lista de operaciones de agregación (string).

Tabla resultante.

Se permiten al menos las funciones básicas: sum, count, mix, max, avg.

Validaciones:

- Tablas deben existir. Mensaje personalizado cuando no existan: ERROR: NO EXISTE LA TABLA nombre\_de\_tabla

- Otros errores Usted los maneja.

**Operación: Agrupación**

Datos solicitados: Tabla.

Lista de atributos que indican las agrupaciones a realizar (string).

Lista de operaciones de agregación que se van a aplicar a los grupos (string).

Tabla resultante.

Se permiten al menos las funciones básicas: sum, count, mix, max, avg.

Validaciones:

- Tablas deben existir. Mensaje personalizado cuando no existan: ERROR: NO EXISTE LA TABLA nombre\_de\_tabla

- Otros errores Usted los maneja.

**Tabla resultante.**

El usuario puede dejar en blanco este dato solicitado o dar un nombre de tabla. En caso de dar un nombre significa que necesita guardar los resultados en una tabla temporal para usos posteriores en otras operaciones. Note que este guardado es la operación de asignación.

Estas tablas donde se guardan los resultados son temporales, van a existir solamente cuando esté corriendo la aplicación. Cuando la aplicación termina las tablas temporales deben eliminarse de la base de datos.

No se permite dar el nombre de una tabla que sea permanente de la base de datos. Validar esta condición, mensaje personalizado de error:

ERROR: NO SE PUEDE DEJAR EL RESULTADO EN UNA TABLA PERMANENTE DE LA BASE DE DATOS.

**Resultados.**

Después de validar los datos solicitados en cada operación se deben mostrar tres resultados:

- La instrucción del álgebra relacional: siempre debe estar visible al usuario.

- La instrucción equivalente de SQL: siempre debe estar visible al usuario.

- La tabla resultante: el programa debe ofrecer una forma para que el usuario pueda desplazarse por los resultados: se despliega una línea por cada tupla.

**Funciones adicionales del intérprete:**

**Ver tablas de la base de datos**

Nombre de la tabla1

Atributo1 Dominio Propiedades (PK, FK, etc.) (un atributo por línea)

…

AtributoN Dominio Propiedades (PK, FK, etc.)

…

Nombre de la tablaN

Atributo1 Dominio Propiedades (PK, FK, etc.)

…

AtributoN Dominio Propiedades (PK, FK, etc.)

**Ver tablas temporales**

Nombre de la tabla1

Atributo1 Propiedades (dominio, PK, FK, etc.) (un atributo por línea)

…

AtributoN Propiedades (dominio, PK, FK, etc.)

…

Nombre de la tablaN

Atributo1 Propiedades (dominio, PK, FK, etc.)

…

AtributoN Propiedades (dominio, PK, FK, etc.)

**Ver referencia cruzada atributos / tablas**

Esta consulta permite ver las tablas de la base de datos en donde aparece cada atributo según el diccionario de datos.

Atributo1 Propiedades (dominio, PK, FK, etc.)

Tabla1

…

TablaN

…

AtributoN Propiedades (dominio, PK, FK, etc.)

Tabla1

…

TablaN

**Ayuda**

Esta opción la usaremos para que el usuario pueda ver el Manual de Usuario (explicado más adelante) preparado en la documentación.

**Acerca de**

Esta opción la usaremos para desplegar una ventana con información “Acerca de la aplicación” donde colocaremos al menos los datos del nombre de la aplicación, la versión, la fecha de creación y los autores.

**Salir**

Esta opción se usa para salir del programa.

En este momento las tablas temporales se deben borrar.

Puede agregar cualquier funcionalidad que Usted considere va a mejorar el producto.

# Desarrollo

Para poder desarrollar el Proyecto lo primero que se hizo fue buscar si ya existía algún software que realizara dicha tarea, y realizando esa búsqueda nos encontramos con un proyecto realizado por unos estudiantes llamado JITRAX.

**Temas Investigados**

**Expresiones Regulares:** Estas son utilizadas en la parte de validación ya que con estas es más fácil detectar que digito el usuario en los campos y así poder mandar un error en caso de ser necesario.

**Java Database Connectivity (JDBC):** Se investigó acerca del driver jdbc para poder conectar la base de datos con la aplicación independientemente del sistema operativo el cual se este utilizando, esta parte era de suma importancia para la funcionalidad de la aplicación.

**Git:** Se investigó sobre Git como herramienta para controlar de versiones y como el repositorio en el cual todos subíamos los cambios realizados en el proyecto, debido a que es una muy buena herramienta para poder implementar trabajo en equipo.

**Software de manejo de Versiones:** Para el desarrollo del proyecto decidimos usar la plataforma de GitHub además de que el lenguaje de manipulación de la base de datos usada fue NetBeans.

# Conclusión

Aprendimos que al utilizar Github se facilitaba el trabajo debido a que varias personas podían avanzar diferentes partes para luego solo subir los cambios, además de que se podía manejar y administrar mejor el proyecto que se estaba desarrollando.

Se tuvieron problemas con el Pull y el Push de Github, hubieron varias veces que duraba en cargar y pero lo mas difícil fue que nos funcionara a todos los integrantes del Grupo.

En java la conexión con la base de datos al principio fue un problema, pero luego de investigar un poco se pudo resolver fácilmente, además java al ser un lenguaje muy utilizado se encontró con facilidad la información que buscábamos.

**Sección 5: RUBRICAS DE EVALUACIÓN.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Concepto** |  |  |  |  | | **Puntos** | **Puntos obtenidos** | **% de Avance**  **100/x/0** | **Análisis de resultados** |
| Resultado de selección | 2 |  | 75 |  |
| Validación | 1 |  |  |  |
| Resultado de proyección generalizada | 2 |  | 75 |  |
| Validación | 1 |  |  |  |
| Resultado de unión | 2 |  | 75 |  |
| Validación | 4 |  |  |  |
| Resultado de diferencia de conjuntos | 2 |  | 75 |  |
| Validación | 4 |  |  |  |
| Resultado de producto cartesiano | 2 |  | 75 |  |
| Validación | 1 |  |  |  |
| Resultado de intersección | 2 |  | 75 |  |
| Validación | 4 |  |  |  |
| Resultado de división | 2 |  |  |  |
| Validación | 4 |  |  |  |
| Resultado de renombrar una relación y atributos | 7 |  |  |  |
| Validación | 7 |  |  |  |
| Resultado de concatenación (join) | 2 |  | 75 |  |
| Validación | 1 |  |  |  |
| Resultado de concatenación natural (natural join) | 5 |  | 75 |  |
| Validación | 5 |  |  |  |
| Resultado de agregación | 2 |  | 75 |  |
| Validación | 1 |  |  |  |
| Resultado de agrupación | 2 |  |  |  |
| Validación | 1 |  |  |  |
| Resultado de mostrar expresión de algebra relacional | 2 |  | 100 |  |
| Resultado de mostrar instrucción SQL | 3 |  | 100 |  |
| Manejo de la tabla de datos resultante (incluye scroll) | 5 |  | 100 |  |
| Ver tablas de la base de datos | 5 |  | 100 |  |
| Ver tablas temporales | 5 |  | 75 |  |
| Ver referencia cruzada atributos/tablas | 5 |  |  |  |
| Borrar tablas temporales | 2 |  |  |  |
| Creación de usuarios (BDA, usuario) | 2 |  | 100 |  |
| Ayuda (manual de usuario) | 5 |  | 100 |  |
| Acerca de / Salir | 0 |  | 100 |  |
| TOTAL | 100 |  |  |  |
| Partes desarrolladas adicionalmente |  |  |  |  |