

Trabajo Practico Final
Lenguaje interactivo para el álgebra relacional

Facundo Emmanuel Messulam

18 de febrero de 2022

1. Introducción

El álgebra relacional permite operar de forma completa con tablas, es decir, permite hacer todas las operaciones que sean necesarias hacer para generar una nueva tabla con la información original organizada diferentemente. Esto es, cualquier cambio en la representación que sea necesario hacer se puede hacer. Esto lo hace muy útil para modelar las operaciones tanto de lenguajes SQL como NoSQL.

El presente trabajo establece una serie de comandos que permite de forma simple interactuar con tablas prearmadas en archivos y permite al usuario realizar una cantidad arbitraria de operaciones en el orden que se desee.

2. Compilacion

Se requiere stack. Luego se puede hacer:

```
stack build --exec TPFinal-exe
```

Para compilar y ejecutar.

3. Relaciones

Las relaciones constan de:

- **Nombre** El nombre es un alfanumérico empezando con mayúscula, ejemplo "Relation"
- **Encabezado** Consta de un nombre alfanumérico y un tipo entre paréntesis (String S, Integer I, Floating F) por cada atributo.
- **Lista de filas** Consta de un valor coherente con el tipo del atributo por cada atributo en el encabezado. También admite N.

3.1. Ejemplo

```
RelationC
AA(S)   BB(I) CC(F) DD(I) EE(I)
"No"    1    2.0  3    1000
"Nop"   1    2.0  3     N
"Not"   1    2.0  3    1000
```

4. Operaciones

4.1. Comandos

4.1.1. Asignación

Primero tenemos la asignación:

```
X = Relation
```

```
X = project(A, B)(Relation)
```

Esta permite darle un nuevo nombre a una relación, particularmente útil para relaciones resultado de una operación, que no tienen nombre.

4.1.2. Salida

```
exit
```

Sale del programa

4.1.3. Impresión

```
print X  
print rename(Y)(X)
```

Imprime una relación, formateada.

4.2. Operaciones básicas

4.2.1. Proyección

```
print (project(x1, x2)(A))
```

Dada una relación A con atributos x_1, \dots, x_n , genera una nueva relación que solo tiene estos atributos.

4.2.2. Selección

```
print (select(x1=3)(A))
```

Dada una relación A con atributos x_1, \dots, x_n , devuelve todas las filas de A para las cuales el predicado es verdadero (ver sección de operaciones booleanas).

4.2.3. Renombre

```
print (rename(B)(A))
```

Dada una relación A, copia A con el nuevo nombre "B", esto no se refleja de forma permanente, y no tiene en cuenta si ya existe B (ver sección sobre la representación interna de relaciones). Esto es muy diferente a $B = A$, este último comando reescribe el estado y guarda de forma permanente el renombre.

4.2.4. Unión

```
print (A + B)
```

Dadas dos relaciones A y B que tienen los mismos atributos (mismo tipo y mismo nombre), une las filas, es decir todas las filas de A y todas las filas de B pertenecen a la unión de A y B.

4.2.5. Producto Cartesiano

```
print (A * B)
```

Dadas dos relaciones A y B, los atributos del producto cartesiano son los atributos de A y los de B; por cada fila de A se generan tantas filas como B tenga, y cada concatenación de filas pasa a ser una fila del producto cartesiano entre A y B.

4.2.6. Diferencia

```
print (A - B)
```

Dadas dos relaciones A y B que tienen los mismos atributos (mismo tipo y mismo nombre), se remueven todas la filas de A que estén en B y se devuelve el resultado.

4.3. Operaciones de extensión

Estas operaciones son azúcar sintáctica, se pueden escribir con las operaciones básicas, pero toma bastante tiempo.

4.3.1. Intersección

```
print (A & B)
```

Dadas dos relaciones A y B que tienen atributos en común (mismo tipo y mismo nombre), primero proyecta sobre los atributos en común, luego devuelve las filas que están en las dos. si no hay atributos en común, devuelve una relación vacía.

4.3.2. Producto natural

```
print (A |*| B)
```

Dadas dos relaciones A y B que tienen atributos $A_1...A_n$ en común (mismo tipo y mismo nombre), se comparan los atributos en común y se unen las filas que tienen los valores $A_1...A_n$ idénticos y se devuelven en una relación.

4.3.3. Division

```
print (A / B)
```

Dadas dos relaciones A y B, donde los atributos de B son un subconjunto de los atributos de A, la división elimina todas la filas de A que tienen intersección vacía con las filas en B (teniendo en cuenta solo los atributos que pertenecen a B), tener en cuenta que se devuelven solo los atributos que no pertenecen a B. La operacion equivalente es proyectar todos los atributos de A de $A \mid * \mid B$.

4.3.4. Producto natural

```
print (A |A.x=3| B)
```

Dadas dos relaciones A y B, por cada fila de $A * B$ se chequea el predicado (ver seccion de operaciones booleanas) entre las barras verticales, y se devuelven las filas que dan verdadero.

4.4. Operaciones booleanas

4.4.1. Igualdad

```
print (A |x=3| B)
```

Compara contra un valor literal o contra otro atributo.

4.4.2. Desigualdad

```
print (A |A.x<>N| B)
```

Compara contra un valor literal o contra otro atributo.

4.4.3. Menor y Menor o igual

```
print (A |x<3| B)
print (A |x<=3| B)
```

Compara contra un valor literal o contra otro atributo.

4.4.4. Mayor y Mayor o igual

```
print (A |B.x>3| B)
print (A |x>=3| B)
```

Compara contra un valor literal o contra otro atributo.

4.4.5. And logico

```
print (A |x>3 && y<>N| B)
```

Une predicados, todos los predicados deben ser verdaderos.

4.4.6. Or logico

```
print (A |x>3 || y<>N| B)
```

Une predicados, al menos uno de los predicados debe ser verdadero.

5. Errores

Se crearon una serie de errores que el usuario puede usar para entender porque una operacion no se puede hacer.

5.1. Error de operacion

Quando se intenta hacer una operacion que no tiene sentido o las relaciones no se prestan a realizar. Estos errores estan explicados claramente el momento en el que ocurren.

5.2. Error de lectura de archivo

Errores en la lectura de las relaciones. Estos errores son similares a errores de parseo, pero tienen arriba a la derecha el nombre del archivo.

5.3. Error de parseo

Esto es una categoria muy grande, pero todos los errores tienen la misma forma, en el caso de cometer un error en la consola:

```
(line 1, column 21):  
unexpected "\""  
expecting letter or digit or ")"
```

En el caso de cometer un error en un archivo, también nos dice el nombre:

```
"relations/empty.r" (line 2, column 8):  
unexpected "a"  
expecting "("
```

La lectura es simple, nos dice la linea y el numero de carácter en la linea, luego nos dice el carácter no esperado, y todas las posibles continuaciones.

6. Representación interna

6.1. Tipos

Los tres tipos son:

- String o S
- Integral o I
- Floating o F

Todos los tipos tienen un valor extra llamado nulo o N, este N tiene tipo, que puede ser cualquiera de los 3.

6.2. Relaciones

La relaciones tienen:

- un nombre
- un encabezado que son una serie de atributos con tipos
- y una serie de filas, que siguen la misma forma que el encabezado en numero y tipo