

ÁLGEBRA RELACIONAL

ÁLGEBRA RELACIONAL

ALGEBRA RELACIONAL

Gómez Urbano Mariana



* Definir cada una de las operaciones del algebra relacional.
* Saber cómo pueden combinarse dichas operaciones para formar expresiones complejas.
* Definir una expresión de recuperación
* Comprender la relación entre las bases de datos utilizando el algebra relacional.

OBJETIVO

OBJETIVO



El algebra relacional permite entender el modelo relacional de base de datos desde la perspectiva matemática. En la presente investigación se mostrará Algebra relacional y se relaciona con la estructura del lenguaje de consulta de base de datos relacionales SQL.

INTRODUCCIÒN

INTRODUCCIÒN



¿QUÉ ES EL ÁLGEBRA RELACIONAL?

¿QUÉ ES EL ÁLGEBRA RELACIONAL?

El "Álgebra Relacional" es un lenguaje, que define una serie de operaciones, que se realizan utilizando "operadores", cada uno de los cuales puede trabajar sobre uno o varios conjuntos de datos produciendo como resultado un nuevo conjunto de datos.

El conjunto de datos resultante de una operación puede a su vez ser utilizado en una nueva operación, en forma anidada, tal como se hace con las operaciones aritméticas. Esta propiedad es conocida como "clausura".

Es un método que consiste básicamente en crear o construir nuevas relaciones a partir de relaciones existentes.



OPERADORES.

OPERADORES.

**Existen 2 tipos de operadores algebraicos:**

* Operadores básicos o primitivos.
* Operadores no básicos o derivados.

**Operadores básicos o primitivos.**

son aquellas operaciones a partir de las cuales podemos definir el resto

Se clasifican en:

1. Proyección (π):

Este operador permite extraer columnas de una relación y de esta manera crea un subconjunto de atributos de la relación, además elimina las filas duplicadas.

Ejemplo de proyección:

SELECT X, Z FROM A WHERE X<50

|  |  |  |
| --- | --- | --- |
| TABLA (A) | | |
| X | Y | Z |
| 1 | 23 | 11 |
| 78 | 32 | 321 |
| 67 | 5 | 33 |
| 15 | 320 | 5 |
| 7 | 5 | 212 |

|  |  |
| --- | --- |
| PROYECCION DE X y Z CON X<50 | |
| X | Z |
| 1 | 11 |
| 15 | 5 |
| 7 | 212 |

2. Selección (σ):

Opera sobre una o más tablas, no siendo necesario que éstas posean la misma estructura, y devolviendo una nueva tabla cuyo contenido es todas las filas de las tablas indicadas que satisfacen una cierta condición, tal como puede verse en el siguiente ejemplo:

|  |  |
| --- | --- |
| TABLA | |
| X | Y |
| 1 | 23 |
| 78 | 32 |
| 67 | 5 |
| 15 | 320 |
| 7 | 5 |

|  |  |
| --- | --- |
| PROYECCION DE X y Z CON X<50 | |
| X | Z |
| 1 | 11 |
| 15 | 5 |
| 7 | 212 |

1. Unión (U):

En algebra relacional la unión de dos relaciones compatibles A y B es:

A UNION B A U B

Produce el conjunto de todas las tuplas que pertenecen ya sea a A o a B o a Ambas.

SELECT \* FROM A   UNION   SELECT \* FROM B

|  |  |
| --- | --- |
| Tabla A | |
| CODIGO | LOCALIDAD |
| 1425 | Buenos Aires |
| 2000 | Rosario |
| 3000 | Santa Fe |
| Tabla B | |
| CODIGO | LOCALIDAD |
| 2000 | Rosario |
| 3000 | Santa Fe |
| 2128 | Arroyo Seco |
| 2121 | Pérez |

|  |  |
| --- | --- |
| A U B | |
| CODIGO | LOCALIDAD |
| 1425 | Buenos Aires |
| 2000 | Rosario |
| 3000 | Santa Fe |
| 2128 | Arroyo Seco |
| 2121 | Pérez |

1. Diferencia (-) :

Se realiza sobre dos tablas, con la misma estructura resultando una table contenido filas en la primer table y no en la otras

SELECT \* FROM A   EXCEPT   SELECT \* FROM B

|  |  |
| --- | --- |
| Tabla A | |
| CODIGO | LOCALIDAD |
| 1425 | Buenos Aires |
| 2000 | Rosario |
| 3000 | Santa Fe |
| Tabla B | |
| CODIGO | LOCALIDAD |
| 2000 | Rosario |
| 3000 | Santa Fe |
| 2128 | Arroyo Seco |
| 2121 | Pérez |

|  |  |
| --- | --- |
| A - B | |
| CODIGO | LOCALIDAD |
| 1425 | Buenos Aires |
|  | |

|  |  |
| --- | --- |
| Tabla A | |
| W | X |
| 1 | 23 |
| 78 | 32 |
| 67 | 5 |

1. Producto cartesiano (\*):

Efectúa un producto del contenido de estas, no siendo necesario que ambas tengan la misma estructura resultando una tercera tabla su contenido son las combinaciones de las filas de una de ambas:

|  |  |
| --- | --- |
| Tabla B | |
| Y | Z |
| 15 | 320 |
| 7 | 5 |
|  | |

|  |  |  |  |
| --- | --- | --- | --- |
| A\*B | | | |
| W | X | Y | Z |
| 1 | 23 | 15 | 320 |
| 1 | 23 | 7 | 5 |
| 78 | 32 | 15 | 320 |
| 78 | 32 | 7 | 5 |
| 67 | 5 | 15 | 320 |
| 67 | 5 | 7 | 5 |



IMPLEMENTACIÓN EN SQL

Y EJEMPLO

IMPLEMENTACIÓN EN SQL

Y EJEMPLO

El lenguaje de manipulación de datos (SQL-Data Manipulation Language) se compone de instrucciones que permiten la gestión, visualización y extracción de las filas de las tablas y de las vistas.

En el lenguaje SQL se pueden rede nominar atributos mediante la cláusula AS. Por ejemplo:

SELECT "grado-experimentalidad", nombre as denominacion

FROM DEPARTAMENTOS

#### **Operaciones específicamente relacionales**

##### La proyección permite elegir algunos atributos

La operación de proyección permite **seleccionar algunos atributos de una relación** (sea ésta una relación o el resultado de otras operaciones). Formalmente, dada una relación R(A1, A2, ... , An), y un subconjunto de atributos de esa relación (Ai, Aj, ..., Ak), la proyección sobre esos atributos se denota como R(Ai, Aj, ..., Ak), y contiene las partes de las tuplas con los valores de los atributos proyectados.

Es importante notar lo siguiente en las proyecciones:

* Como el resultado de cualquier operación de álgebra relacional es una relación, las tuplas repetidas en el resultado de la proyección se eliminarán.
* El orden de los atributos en la proyección no tiene que ser el mismo que en la relación original.

La operación de proyección en el lenguaje de consulta SQL se realiza simplemente especificando el subconjunto de atributos en la cláusula SELECT.

SELECT Ai, Aj, ..., Ak FROM R

Por ejemplo, si tuviésemos la siguiente extensión de la tabla PROFESORES:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dni | apellidos | nombre | fecha-nacimiento | es-doctor | teléfono-móvil | D\_nombre |
| 32323 | Pérez | Juan | 30/10/1965 | true | 65454545 | Matemáticas |
| 4434343 | Díez | José | 30/10/1970 | true | 65789767 | Historia |
| 23423432 | Sánchez | Juan | 30/10/1980 | false | 66789899 | Matemáticas |

La siguiente sentencia SELECT hace una proyección del nombre y los apellidos.

SELECT nombre, apellidos FROM PROFESORES

Es importante notar que la siguiente sentencia no devolvería una relación, ya que tendría dos tuplas repetidas.

SELECT nombre FROM profesores

Para asegurar que el resultado es una relación en el sentido matemático, hay que utilizar el modificador DISTINCT.

SELECT DISTINCT nombre FROM profesores

##### La selección permite elegir algunas tuplas

La selección es una operación que elige algunas tuplas de una relación y elimina el resto. La nueva relación contiene por tanto solo tuplas seleccionadas que cumplen una determinada **condición de selección C** La condición de selección es una condición lógica que permita decidir qué incluir y qué no.

Los atributos de la selección son los mismos que los de la relación original, y todas las tuplas de la selección cumplen la condición C. Por ejemplo:

R := DEPARTAMENTOS(grado-experimentalidad >1.0)

Q := PROFESORES(no es-doctor y teléfono-móvil <> null)

En SQL, la selección se especifica mediante la cláusula WHERE. Por ejemplo:

SELECT \* FROM DEPARTAMENTOS

WHERE grado-experimentalidad > 1.0

SELECT \* FROM PROFESORES

WHERE not es doctor and teléfono-móvil IS NOT NULL.

**La combinación o reunión (join) permite cruzar los valores de tablas relacionadas**

La combinación toma dos relaciones y devuelve una relación con las tuplas que resultan de concatenar tuplas de la primera con tuplas de la segunda y después seleccionar las que cumplen una condición de combinación C. Una combinación entre las relaciones R y Q mediante la condición C se denota como R[C]Q.

Por ejemplo, si queremos obtener la información de profesores y departamentos combinada, utilizaremos como combinación la igualdad de la clave ajena en PROFESORES con la clave primaria en DEPARTAMENTOS:

DPT:=DEPARTAMENTOS

PROFESORES[D\_nombre = DPT.nombre]DPT

En SQL, la combinación se hace incluyendo más de una relación en la cláusula FROM, y la condición C se coloca como cláusula WHERE.

SELECT \*

FROM PROFESORES AS P, DEPARTAMENTOS AS D

WHERE P.D\_nombre = D.nombre

**La unión de relaciones es la unión de conjuntos**

La unión de dos relaciones devuelve una nueva relación que contiene todas las tuplas que aparecían en cualquiera de las relaciones originales, o en ambas.

Así, la operación de unión de dos relaciones R y T, que denominaremos UNION(R, T) sólo se puede aplicar el operador unión a dos relaciones que tengan los mismos atributos, o que sean compatibles (es decir, que se pueda establecer una biyección entre los atributos de las dos relaciones).

Es importante tener en cuenta lo siguiente:

* Se eliminarán las tuplas repetidas. Se entiende que en las dos relaciones no debe haber tuplas con la misma clave primaria y el resto de la información diferente.

Si tenemos dividida la información de los profesores antiguos y los nuevos en dos relaciones (por ejemplo, por motivos de rendimiento), pero queremos obtener un listado histórico de todos ellos, podemos utilizar el operador UNION de SQL.

SELECT \* FROM PROFESORES

UNION

SELECT \* FROM EX-PROFESORES

El operador UNION requiere que los atributos de las relaciones devueltas por las dos SELECT sean del mismo tipo según el orden de aparición. Si en una de las tablas este orden fuese diferente, habría que utilizar una proyección que alterase el orden. Los nombres de atributos de la nueva relación serán los de la primera SELECT.

##### **La intersección de relaciones es la intersección de conjuntos**

La intersección de relaciones toma dos relaciones y devuelve una relación con las tuplas que aparecían en ambas relaciones originales. Como la unión, la intersección solo puede aplicarse a relaciones con atributos compatibles.

La siguiente consulta en SQL por tanto debería devolver una relación vacía:

SELECT \* FROM PROFESORES

INTERSECT

SELECT \* FROM EX-PROFESORES

##### El producto cartesiano combina todas las tuplas de dos relaciones

El producto cartesiano es una operación que toma dos relaciones y obtiene una nueva relación cuyas tuplas son la concatenación de las tuplas de la primera relación con las tuplas de la segunda. En general, si la primera relación de entrada tenía N tuplas y la segunda M tuplas, el producto cartesiano produce N×M tuplas, aunque en ocasiones serán menos dado que las tuplas repetidas habrán de eliminarse.

El producto cartesiano de dos relaciones en SQL se obtiene simplemente colocando más de una tabla en la cláusula FROM.

SELECT \* FROM PROFESORES, DEPARTAMENTOS

El producto cartesiano raramente se utiliza en la práctica, pero es interesante conocerlo para diferenciarlo de la operación de combinación (join).



CONCLUSIONES

CONCLUSIONES

Fuentes:

Sicilia, M. A. (n.d.). *OpenStax CNX*. Openatx CNX. Retrieved February 17, 2021, from https://cnx.org/contents/74gU77-S@1/Introducci%C3%B3n-al-%C3%A1lgebra-relacional-Con-ejemplos-en-SQL

*4 Algebra relacional*. (n.d.). Algebra Relacional. Retrieved February 16, 2021, from http://ict.udlap.mx/people/carlos/is341/bases04.html

UNIVERSIDA AUTONOMA DEL ESTADO DE HIDALGO. (n.d.). *3.3 Algebra Relacional*. Centro de Innovación Para El Desarrollo y La Capacitación En Materiales Educativos. Retrieved February 16, 2021, from http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/33\_algebra\_relacional.html

Confirme en base de este trabajo las operaciones álgebra relacional resultan importantes para hacer eficiente las consultas de una base de datos. Es fundamental conocer estas operaciones ya que optimiza la consulta y al ser incorporado a la lengua de consulta SQL este último utilizado mundialmente por distintas organizaciones. Por lo tanto, el áálgebra relacional es el pionero de todo este sistema.