

## 4. Módulo Consulta

El módulo Consulta permite construir consultas y obtener información sobre los parámetros que toma un determinado tipo de consulta. Para ello, usa el tipo enumerado TIPOCONSULTA, que incluye los tipos FROM, SELECT, MATCH, PROJ, RENAME, INTER, UNION y PRODUCT. Como las consultas están construidas de manera recursiva, esto posibilita obtener información sobre las consultas previas sobre las cuales está basada la consulta actual hasta llegar a la consulta base, que en todos los casos es de tipo FROM (ya que es la única que no toma otra consulta como parámetro).

Como cada tipo de consulta toma diferentes parámetros, según el tipo de consulta del que se trate es posible obtener distintos tipos de información. En el caso de FROM, solo es posible obtener el nombre de la tabla que se está consultando. Si la consulta es de tipo SELECT, se puede obtener el nombre del campo por el que se aplica el filtro y el valor asociado a ese campo que deben tener los registros que se quieren conservar, mientras que si es de tipo MATCH o RENAME, se puede obtener el nombre de los dos campos que se pasan como parámetro (en el caso de MATCH, los campos que deben coincidir; en el de RENAME, el nombre del campo a renombrar y el nombre nuevo). En el caso de PROJ, se puede obtener información sobre los nombres de los campos que deben estar incluidos en el resultado de la consulta. Finalmente, para todos los tipos de consulta excepto FROM, dado que como hemos mencionado oportunamente toman otra/s consulta/s como parámetro, es posible conocer el tipo de las consultas que reciben.

Vale la pena señalar que el TAD CONSULTA no permite realizar consultas a las tablas de forma directa, ya que para obtener resultados (bajo la forma de un conjunto de registros) es necesario realizar las consultas en el marco de una base de datos (ver TAD BASE DE DATOS). Por este motivo, desde el punto de vista del TAD CONSULTA dos consultas pueden ser iguales observacionalmente incluso si dan resultados distintos al ser aplicadas a diferentes bases de datos, es decir, están definidas como instancias solo a partir de su construcción.

### Interfaz

se explica con: TAD CONSULTA

géneros: consulta.

### Operaciones básicas de Consulta

FROM(in  $n$ : nombre\_tabla)  $\rightarrow res$  : consulta

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res =_{\text{obs}} \text{FROM}(n)\}$

**Complejidad:**  $\Theta(|n|)$

**Descripción:** genera una instancia de consulta FROM a partir del nombre de una tabla.

SELECT(in  $q$ : consulta, in  $c$ : nombre\_campo, in  $v$ : valor)  $\rightarrow res$  : consulta

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res =_{\text{obs}} \text{SELECT}(q, c, v)\}$

**Complejidad:**  $\Theta(|c| + |v|)$

**Descripción:** genera una instancia de consulta SELECT a partir de otra consulta y el nombre de un campo y su valor asociado (que serán el criterio de selección de los registros resultantes).

MATCH(in  $q$ : consulta, in  $c_1$ : nombre\_campo, in  $c_2$ : nombre\_campo)  $\rightarrow res$  : consulta

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res =_{\text{obs}} \text{MATCH}(q, c_1, c_2)\}$

**Complejidad:**  $\Theta(|c_1| + |c_2|)$

**Descripción:** genera una instancia de consulta MATCH a partir de otra consulta y los nombres de los dos campos cuyos valores tienen que coincidir.

PROJ(in  $q$ : consulta, in  $cs$ : conj(nombre\_campo))  $\rightarrow res$  : consulta

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res =_{\text{obs}} \text{PROJ}(q, cs)\}$

**Complejidad:**  $\Theta(|cs|)$

**Descripción:** genera una instancia de consulta PROJ a partir de otra consulta y de un conjunto de nombres de campos (correspondientes a las columnas que se conservarán en el conjunto de registros resultante).

RENAME(in  $q$ : consulta, in  $c_1$ : nombre\_campo, in  $c_2$ : nombre\_campo)  $\rightarrow res$  : consulta

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res =_{\text{obs}} \text{RENAME}(q, c_1, c_2)\}$

**Complejidad:**  $\Theta(|c_1| + |c_2|)$

**Descripción:** genera una instancia de consulta RENAME a partir de una consulta anterior, el nombre del campo a renombrar y el nombre nuevo.

INTER(in  $q_1$ : consulta, in  $q_2$ : consulta)  $\rightarrow res$ : consulta

Pre  $\equiv \{\text{true}\}$

Post  $\equiv \{res =_{\text{obs}} \text{INTER}(q_1, q_2)\}$

Complejidad:  $\Theta(|q_1| + |q_2|)$

**Descripción:** genera una instancia de consulta INTER a partir de otras dos consultas.

UNION(in  $q_1$ : consulta, in  $q_2$ : consulta)  $\rightarrow res$ : consulta

Pre  $\equiv \{\text{true}\}$

Post  $\equiv \{res =_{\text{obs}} \text{UNION}(q_1, q_2)\}$

Complejidad:  $\Theta(|q_1| + |q_2|)$

**Descripción:** genera una instancia de consulta UNION a partir de otras dos consultas.

PRODUCT(in  $q_1$ : consulta, in  $q_2$ : consulta)  $\rightarrow res$ : consulta

Pre  $\equiv \{\text{true}\}$

Post  $\equiv \{res =_{\text{obs}} \text{PRODUCT}(q_1, q_2)\}$

Complejidad:  $\Theta(|q_1| + |q_2|)$

**Descripción:** genera una instancia de consulta PRODUCT.

TIPO\_CONSULTA(in  $q$ : consulta)  $\rightarrow res$ : tipo\_consulta

Pre  $\equiv \{\text{true}\}$

Post  $\equiv \{res =_{\text{obs}} \text{tipo\_consulta}(q)\}$

Complejidad:  $\Theta(1)$

**Descripción:** devuelve el tipo de consulta de la consulta  $q$ .

**Aliasing:**  $res$  no es modificable y se devuelve por referencia.

NOMBRE\_TABLA(in  $q$ : consulta)  $\rightarrow res$ : nombre\_tabla

Pre  $\equiv \{\text{tipo\_consulta}(q) \in \{\text{FROM}\}\}$

Post  $\equiv \{res =_{\text{obs}} \text{nombre\_tabla}(q)\}$

Complejidad:  $\Theta(1)$

**Descripción:** devuelve el nombre de la tabla sobre la cual se realiza la consulta  $q$ , que debe ser de tipo FROM.

**Aliasing:**  $res$  no es modificable y se devuelve por referencia.

CAMPO<sub>1</sub>(in  $q$ : consulta)  $\rightarrow res$ : nombre\_campo

Pre  $\equiv \{\text{tipo\_consulta}(q) \in \{\text{SELECT}, \text{MATCH}, \text{RENAME}\}\}$

Post  $\equiv \{res =_{\text{obs}} \text{campo}_1(q)\}$

Complejidad:  $\Theta(1)$

**Descripción:** devuelve el nombre del primer (o único) campo que la consulta  $q$ , que debe ser de tipo SELECT, MATCH o RENAME, toma como parámetro.

**Aliasing:**  $res$  no es modificable y se devuelve por referencia.

CAMPO<sub>2</sub>(in  $q$ : consulta)  $\rightarrow res$ : nombre\_campo

Pre  $\equiv \{\text{tipo\_consulta}(q) \in \{\text{MATCH}, \text{RENAME}\}\}$

Post  $\equiv \{res =_{\text{obs}} \text{campo}_2(q)\}$

Complejidad:  $\Theta(1)$

**Descripción:** devuelve el nombre del segundo campo que la consulta  $q$ , que debe ser de tipo MATCH o RENAME, toma como parámetro.

**Aliasing:**  $res$  no es modificable y se devuelve por referencia.

CONJ\_CAMPOS(in  $q$ : consulta)  $\rightarrow res$ : conj(nombre\_campo)

Pre  $\equiv \{\text{tipo\_consulta}(q) \in \{\text{PROJ}\}\}$

Post  $\equiv \{res =_{\text{obs}} \text{conj\_campos}(q)\}$

Complejidad:  $\Theta(1)$

**Descripción:** devuelve el conjunto de nombres de campo que la consulta  $q$ , que debe ser de tipo PROJ, toma como parámetro.

**Aliasing:**  $res$  no es modificable y se devuelve por referencia.

SUBCONSULTA<sub>1</sub>(in  $q$ : consulta)  $\rightarrow res$ : conj(nombre\_campo)

Pre  $\equiv \{\text{tipo\_consulta}(q) \in \{\text{SELECT}, \text{MATCH}, \text{PROJ}, \text{RENAME}, \text{INTER}, \text{UNION}, \text{PRODUCT}\}\}$

**Post**  $\equiv \{res =_{\text{obs}} \text{subconsulta}_1(q)\}$

**Complejidad:**  $\Theta(1)$

**Descripción:** devuelve la primera (o única) consulta que la consulta  $q$ , que puede ser de cualquier tipo excepto FROM, toma como parámetro.

**Aliasing:**  $res$  no es modificable y se devuelve por referencia.

**SUBCONSULTA<sub>2</sub>**(in  $q$ : consulta)  $\rightarrow res$  : conj(nombre\_campo)

**Pre**  $\equiv \{\text{tipo\_consulta}(q) \in \{\text{INTER}, \text{UNION}, \text{PRODUCT}\}\}$

**Post**  $\equiv \{res =_{\text{obs}} \text{subconsulta}_2(q)\}$

**Complejidad:**  $\Theta(1)$

**Descripción:** devuelve la segunda consulta que la consulta  $q$ , que debe ser de tipo INTER, UNION o PRODUCT, toma como parámetro.

**Aliasing:**  $res$  no es modificable y se devuelve por referencia.

## Representación

### Representación de consulta

La consultas se representan con una estructura que incluye:

- un string que indica el tipo de consulta.
- cuatro *strings* correspondientes a los parámetros de tipo *string* que pueden tomar las consultas.
- un arreglo estático de dos punteros a las subconsultas. Si el tipo de consulta definido en *tipoConsulta* toma dos subconsultas, ambas posiciones tendrán punteros a subconsultas. Si toma solo una, la segunda posición del arreglo contendrá un puntero nulo. Si no toma ninguna (solo en el caso de FROM), ambos punteros deberán ser nulos.

consulta se representa con cons

donde cons es `tupla(tipoConsulta: string, nombreTabla: string, valor: string, campo1: string, campo2: string, conjCampos: conj(string), subconsultas: arreglo[2] de puntero(consulta))`

### Invariante de representación

1. El tipo de consulta debe ser uno de los strings 'FROM', 'SELECT', 'MATCH', 'PROJ', 'RENAME', 'INTER', 'UNION', 'PRODUCT'.
2. *nombreTabla* será no vacío si y solo si *tipoConsulta* es igual a 'FROM'.
3. *valor* será no vacío si y solo si *tipoConsulta* es 'SELECT'.
4. *campo1* será no vacío si y solo si *tipoConsulta* es 'SELECT', 'MATCH' o 'RENAME'.
5. *campo2* será no vacío si y solo si *tipoConsulta* es 'MATCH' o 'RENAME'.
6. *conjCampos* será no vacío si y solo si *tipoConsulta* es 'PROJ'.
7. *subconsultas* tendrá punteros nulos en ambas posiciones si y solo si *tipoConsulta* es 'FROM', tendrá un puntero válido en la primera posición y uno nulo en la segunda si y solo si *tipoConsulta* es 'SELECT', 'MATCH', 'PROJ' o 'RENAME' y tendrá punteros válidos en ambas posiciones si *tipoConsulta* es 'INTER', 'UNION' o 'PRODUCT'.

### Función de abstracción

1. La consulta representada por la estructura será del tipo que se encuentra en el elemento *tipoConsulta*.
2. Si la consulta es de tipo FROM, el nombre de la tabla que toma como parámetro será igual al *string* de *nombreTabla*.
3. Si la consulta es de tipo SELECT, el valor que toma como parámetro será igual al *string* de *valor*.
4. Si la consulta es de tipo SELECT, el campo que toma como parámetro será igual al *string* de *campo1*. Si la consulta es de tipo MATCH o RENAME, el primer campo que toma como parámetro será igual al *string* de *campo1*.

5. Si la consulta es de tipo MATCH o RENAME, el segundo campo que toma como parámetro será igual al *string* de *campo2*.
6. Si la consulta es de tipo PROJ, el conjunto de campos que toma como parámetro será igual al conjunto de campos de *conjCampos*.
7. Si la consulta es de tipo SELECT, MATCH, PROJ o RENAME, la subconsulta que toman como parámetro será igual a la subconsulta a la que apunta el puntero de la primera posición del arreglo *subconsultas*. Si es de tipo INTER, UNION o PRODUCT, esta será igual a la primera subconsulta que toman como parámetro.
8. Si la consulta es de tipo INTER, UNION o PRODUCT, la segunda subconsulta que toman como parámetro será igual a la subconsulta a la que apunta el puntero de la segunda posición del arreglo *subconsultas*.

## Algoritmos

---

**iFROM**(in  $n$ : nombre\_tabla)  $\rightarrow res$ : consulta

1:  $res \leftarrow \langle \text{'FROM', 'n', ', ', ', ', Vacío(), [NULL, NULL]} \rangle$

Complejidad:  $\Theta(|n|)$

---



---

**iSELECT**(in  $q$ : consulta, in  $c$ : nombre\_campo, in  $v$ : valor)  $\rightarrow res$ : consulta

1:  $res \leftarrow \langle \text{'SELECT', ', ', 'v', 'c', ', ', Vacío(), [&q, NULL]} \rangle$

Complejidad:  $\Theta(|c| + |v|)$

---



---

**iMATCH**(in  $q$ : consulta, in  $c_1$ : nombre\_campo, in  $c_2$ : nombre\_campo)  $\rightarrow res$ : consulta

1:  $res \leftarrow \langle \text{'MATCH', ', ', ', 'c_1', 'c_2', Vacío(), [&q, NULL]} \rangle$

Complejidad:  $\Theta(|c_1| + |c_2|)$

---



---

**iPROJ**(in  $q$ : consulta, in  $cs$ : conj(nombre\_campo))  $\rightarrow res$ : consulta

1:  $res \leftarrow \langle \text{'PROJ', ', ', ', ', cs, [&q, NULL]} \rangle$

Complejidad:  $\Theta(|cs|)$

---



---

**iRENAME**(in  $q$ : consulta, in  $c_1$ : nombre\_campo, in  $c_2$ : nombre\_campo)  $\rightarrow res$ : consulta

1:  $res \leftarrow \langle \text{'RENAME', ', ', ', 'c_1', 'c_2', Vacío(), [&q, NULL]} \rangle$

Complejidad:  $\Theta(|c_1| + |c_2|)$

---



---

**iINTER**(in  $q_1$ : consulta, in  $q_2$ : consulta)  $\rightarrow res$ : consulta

1:  $res \leftarrow \langle \text{'INTER', ', ', ', ', Vacío(), [&q_1, &q_2]} \rangle$

Complejidad:  $\Theta(|q_1| + |q_2|)$

---



---

**iUNION**(in  $q_1$ : consulta, in  $q_2$ : consulta)  $\rightarrow res$ : consulta

1:  $res \leftarrow \langle \text{'UNION', ', ', ', ', Vacío(), [&q_1, &q_2]} \rangle$

Complejidad:  $\Theta(|q_1| + |q_2|)$

---

---

---

**iPRODUCT**(in  $q_1 : \text{consulta}$ , in  $q_2 : \text{consulta}$ )  $\rightarrow res : \text{consulta}$

1:  $res \leftarrow \langle \text{'PRODUCT'}, ", ", ", ", \text{Vacío}(), [\&q_1, \&q_2] \rangle$

Complejidad:  $\Theta(|q_1| + |q_2|)$

---

---

---

**iTipo\_consulta**(in  $q : \text{consulta}$ )  $\rightarrow res : \text{tipo\_consulta}$

1:  $res \leftarrow q.\text{tipoConsulta}$

Complejidad:  $\Theta(1)$

Justificación:  $res$  se devuelve por referencia.

---

---

---

**iNombre\_tabla**(in  $q : \text{consulta}$ )  $\rightarrow res : \text{nombre\_tabla}$

1:  $res \leftarrow q.\text{nombreTabla}$

Complejidad:  $\Theta(1)$

Justificación:  $res$  se devuelve por referencia.

---

---

---

**iCampo<sub>1</sub>**(in  $q : \text{consulta}$ )  $\rightarrow res : \text{nombre\_campo}$

1:  $res \leftarrow q.\text{campo}_1$

Complejidad:  $\Theta(1)$

Justificación:  $res$  se devuelve por referencia.

---

---

---

**iCampo<sub>2</sub>**(in  $q : \text{consulta}$ )  $\rightarrow res : \text{nombre\_campo}$

1:  $res \leftarrow q.\text{campo}_2$

Complejidad:  $\Theta(1)$

Justificación:  $res$  se devuelve por referencia.

---

---

---

**iConj\_campos**(in  $q : \text{consulta}$ )  $\rightarrow res : \text{conj}(\text{nombre\_campo})$

1:  $res \leftarrow q.\text{conjCampos}$

Complejidad:  $\Theta(1)$

Justificación:  $res$  se devuelve por referencia.

---

---

---

**iSubconsulta<sub>1</sub>**(in  $q : \text{consulta}$ )  $\rightarrow res : \text{consulta}$

1:  $res \leftarrow q.\text{subconsultas}[0]$

Complejidad:  $\Theta(1)$

Justificación:  $res$  se devuelve por referencia.

---

---

---

**iSubconsulta<sub>2</sub>**(in  $q : \text{consulta}$ )  $\rightarrow res : \text{consulta}$

1:  $res \leftarrow q.\text{subconsultas}[1]$

Complejidad:  $\Theta(1)$

Justificación:  $res$  se devuelve por referencia.

---