

1. Módulo Base de Datos

El módulo Base de Datos permite crear una base de datos, agregarle información y realizar distintas consultas. Una base de datos consiste en un conjunto de tablas (cada una de las cuales representa un concepto), identificadas con un nombre único (equivalente a un TAD `DICCIONARIO(NOMBRE_TABLA, TABLA)`). Una vez creada una base de datos, es posible agregarle tablas, indicando los nombres de sus campos y cuál es el campo clave. A cada tabla, a su vez, es posible agregarle registros, indicando el nombre de la tabla y el valor asociado a los campos de la tabla a la que se le está agregando (es decir, un conjunto de asociaciones campo-valor; ver TAD `REGISTRO`). También se pueden eliminar registros de una tabla, identificándolos por el valor del campo clave, y eliminar tablas completas. Por último, la base de datos dispone de ocho tipos de consultas, que dan como resultado un conjunto de registros.

En ningún caso se imponen restricciones a las entradas de las consultas, sino que se devuelve un conjunto de registro al que no se le aplicó la consulta (por ejemplo, si se quiere usar `RENAME` para renombrar un campo que no existe, la consulta no tendrá efecto) o un conjunto de registros vacío (por ejemplo, si se hace la consulta `FROM` con el nombre de una tabla que no existe en la base).

Interfaz

se explica con: TAD BASE DE DATOS

géneros: `base_de_datos`.

Operaciones básicas de Base de Datos

`CREAR()` $\rightarrow res : \text{base_de_datos}$

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

Post $\equiv \{res =_{\text{obs}} \text{crear}()\}$

Descripción: crea una base de datos vacía.

`AGREGARTABLA(in n : nombre_tabla, in k : nombre_campo, in cs : conj(nombre_campo), in/out b : base_de_datos)`

Pre $\equiv \{b = b_0 \wedge \neg \text{def?}(n, \text{tablas}(b)) \wedge k \in cs\}$

Post $\equiv \{b =_{\text{obs}} \text{agregarTabla}(n, k, cs, b_0)\}$

Descripción: crea una tabla (vacía) con el nombre n y los campos incluidos en cs en la base de datos b e indica cuál es el campo clave de la tabla (k).

`AGREGARREGISTRO(in r : registro, in n : nombre_tabla, in/out b : base_de_datos)`

Pre $\equiv \{b = b_0 \wedge \text{def?}(n, \text{tablas}(b)) \wedge_{\text{L}} \text{campos}(\text{obtener}(n, b)) =_{\text{obs}} \text{campos}(r) \wedge (\forall \text{reg} : \text{registro})$

$(\text{reg} \in \text{registros}(\text{obtener}(n, b)) \Rightarrow_{\text{L}} \text{reg}[\text{clave}(\text{obtener}(n, b))] \neq r[\text{clave}(\text{obtener}(n, b))])\}$

Post $\equiv \{res =_{\text{obs}} \text{agregarRegistro}(r, n, b)\}$

Descripción: agrega un registro r a la tabla de nombre n definida en la base de datos b .

`TABLAS(in b : base_de_datos) $\rightarrow res : \text{dicc}(\text{nombre_tabla}, \text{tabla})$`

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

Post $\equiv \{res =_{\text{obs}} \text{tablas}(b)\}$

Descripción: a partir de una base de datos b , devuelve un diccionario cuyas claves son los nombres de las tablas que la conforman y sus significados, las instancias tipo tabla en la base de datos que corresponden a ese nombre.

`ELIMINARTABLA(in n : nombre_tabla, in/out b : base_de_datos)`

Pre $\equiv \{b = b_0 \wedge \text{def?}(n, \text{tablas}(b))\}$

Post $\equiv \{b =_{\text{obs}} \text{eliminarTabla}(n, b_0)\}$

Descripción: elimina la tabla de nombre n de la base de datos b .

`ELIMINARREGISTRO(in v : valor, in n : nombre_tabla, in/out b : base_de_datos)`

Pre $\equiv \{b = b_0 \wedge \text{def?}(n, \text{tablas}(b)) \wedge_{\text{L}} (\exists r : \text{registro})(r \in \text{registros}(\text{obtener}(n, b)) \wedge_{\text{L}} r[\text{clave}(\text{obtener}(n, b)) = v])\}$

Post $\equiv \{b =_{\text{obs}} \text{eliminarRegistro}(v, n, b_0)\}$

Descripción: elimina el registro de la tabla de nombre n que tenga el valor v en el campo clave.

`REALIZARCONSULTA(in q : consulta, in b : base_de_datos) $\rightarrow res : \text{conj}(\text{registros})$`

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

Post $\equiv \{res =_{\text{obs}} \text{realizarConsulta}(t, b)\}$

Descripción: realiza una consulta q sobre la base de datos b y devuelve un conjunto de registros que contiene los resultados de la misma. A continuación se detallan los tipos de consultas disponibles y las operaciones que realiza

cada uno:

- FROM: a partir del nombre n de una tabla, devuelve el conjunto de registros de dicha tabla. Si la tabla no existe en la base de datos, devuelve un conjunto vacío.
- SELECT: a partir de los resultados de una consulta previa q , el nombre de un campo c y un valor v , devuelve el conjunto de registros que tienen ese valor asociado al campo elegido. Si hay registros que no tienen este campo, no se incluyen en los resultados.
- MATCH: a partir de los resultados de una consulta previa q y dos nombres de campo c_1 y c_2 , devuelve un conjunto que contiene solo los registros que tienen el mismo valor v en ambos campos. Si hay registros que no tienen alguno de esos dos, no se incluyen en el conjunto final.
- PROJ: a partir de los resultados de una consulta previa q y un conjunto de nombres de campo cs , devuelve un conjunto con los mismos registros que q pero dejando solo las columnas correspondientes a los campos de nombre perteneciente al conjunto cs .
- RENAME: toma los resultados de una consulta previa y cambia el nombre de uno de los campos por otro pasado como parámetro. Si hay registros que no tienen definido el campo c_1 , no se efectúa ningún renombre y se devuelve el registro tal como está. Si ya tienen definido un campo con el nombre c_2 , la operación de renombre queda sin efecto para ese registro.
- INTER: toma los resultados de dos consultas previas q_1 , q_2 y devuelve el conjunto de registros que se encuentran tanto en el conjunto resultante de la primera consulta como en el de la segunda.
- UNION: toma los resultados de dos consultas previas q_1 , q_2 y devuelve el conjunto de registros que se encuentran en el conjunto resultante de la primera consulta o en el de la segunda.
- PRODUCT: toma los resultados de dos consultas previas q_1 , q_2 y devuelve un conjunto de registros que es el producto cartesiano de esos dos conjuntos, es decir, combina cada uno de los registros del primer conjunto con cada uno de los del segundo (formando nuevos registros con ambos campos). Si hay registros provenientes de la primera consulta q_1 que tienen campos en común con registros provenientes de la segunda, se conserva el valor del campo correspondiente a la segunda consulta.