

TAD Base de Datos

TAD BASE DE DATOS

géneros base_de_datos

usa BOOL, CONJUNTO, DICCIONARIO, TABLA, REGISTRO, CONSULTA

exporta generadores, observadores, eliminarTabla, eliminarRegistro, realizarConsulta

igualdad observacional

$$(\forall b_1, b_2 : \text{base_de_datos}) (b_1 =_{\text{obs}} b_2 \iff (\text{tablas}(b_1) =_{\text{obs}} \text{tablas}(b_2)))$$

generadores

crear : $\longrightarrow \text{base_de_datos}$
 agregarTabla : nombre_tabla $n \times$ nombre_campo $k \times$ $\longrightarrow \text{base_de_datos}$
 conj(nombre_campo) $cs \times \text{base_de_datos } b$ $\longrightarrow \text{base_de_datos}$
 $\{ \neg \text{def?}(n, \text{tablas}(b)) \wedge k \in cs \}$
 agregarRegistro : registro $r \times$ nombre_tabla $n \times$ base_de_datos $b \longrightarrow \text{base_de_datos}$
 $\left\{ \begin{array}{l} \text{def?}(n, \text{tablas}(b)) \wedge_L \text{campos}(\text{obtener}(n, b)) =_{\text{obs}} \text{campos}(r) \wedge (\forall \text{reg} : \text{registro})(\text{reg} \in \text{registros}(\text{obtener}(n, b)) \Rightarrow_L \text{reg}[\text{clave}(\text{obtener}(n, b))] \neq r[\text{clave}(\text{obtener}(n, b))]) \end{array} \right\}$

observadores básicos

tablas : base_de_datos $\longrightarrow \text{dicc}(\text{nombre_tabla}, \text{tabla})$

otras operaciones

eliminarTabla : nombre_tabla $n \times$ base_de_datos $b \longrightarrow \text{base_de_datos}$
 $\{ \text{def?}(n, \text{tablas}(b)) \}$
 eliminarRegistro : valor $v \times$ nombre_tabla $n \times$ base_de_datos $b \longrightarrow \text{base_de_datos}$
 $\left\{ \begin{array}{l} \text{def?}(n, \text{tablas}(b)) \wedge_L (\exists r : \text{registro})(r \in \text{registros}(\text{obtener}(n, b))) \\ \wedge_L r[\text{clave}(\text{obtener}(n, b))] = v \end{array} \right\}$
 realizarConsulta : consulta \times base_de_datos $\longrightarrow \text{conj}(\text{registro})$
 from : nombre_tabla \times base_de_datos $\longrightarrow \text{conj}(\text{registro})$
 select : conj(registro) \times nombre_campo \times valor $\longrightarrow \text{conj}(\text{registro})$
 match : conj(registro) \times nombre_campo \times nombre_campo $\longrightarrow \text{conj}(\text{registro})$
 proj : conj(registro) \times conj(nombre_campo) $\longrightarrow \text{conj}(\text{registro})$
 filtrarCampos : registro \times registro \times conj(nombre_campo) $\longrightarrow \text{registro}$
 rename : conj(registro) \times nombre_campo \times nombre_campo \times base_de_datos $\longrightarrow \text{conj}(\text{registro})$
 renombrarCampo : nombre_campo \times nombre_campo \times conj(nombre_campo) $\longrightarrow \text{registro}$
 $\times \text{registro} \times \text{registro}$
 inter : conj(registro) \times conj(registro) $\longrightarrow \text{conj}(\text{registro})$
 union : conj(registro) \times conj(registro) $\longrightarrow \text{conj}(\text{registro})$
 product : conj(registro) \times conj(registro) \times base_de_datos $\longrightarrow \text{conj}(\text{registro})$
 emparejarUnoAMuchos : registro \times conj(registro) $\longrightarrow \text{conj}(\text{registro})$
 emparejarUnoAUno : registro \times registro $\longrightarrow \text{registro}$
 poblarRegistro : registro \times registro \times conj(nombre_campo) $\longrightarrow \text{registro}$

axiomas $\forall b : \text{base de datos}, t : \text{tabla}, r : \text{registro}, n : \text{nombre_tabla}, c, k : \text{nombre_campo}, q : \text{consulta}, v : \text{valor}, cs : \text{conj}(\text{nombre_campo}), cr : \text{conj}(\text{registro})$

tablas(crear)	\equiv vacío
tablas(agregarTabla(n, k, cs, b))	\equiv definir($n, nueva(cs, k)$)
tablas(agregarRegistro(r, n, b))	\equiv definir($n, insertar(obtener(n, b), r), tablas(b)$)
eliminarTabla($n_1, crear$)	\equiv vacío
eliminarTabla($n_1, agregarTabla(n_2, k, cs, b)$)	\equiv if $n_1 = n_2$ then b else eliminarTabla(n_1, b) fi
eliminarTabla($n_1, agregarRegistro(r, n_2, b)$)	\equiv if $n_1 = n_2$ then eliminarTabla(n_1, b) else eliminarTabla($n_1, agregarRegistro(r, n_2, b)$) fi
eliminarRegistro($v, n_1, crear$)	\equiv vacío
eliminarRegistro($v, n_1, agregarTabla(n_2, k, cs, b)$)	\equiv eliminarRegistro(v, n_1, b)
eliminarRegistro($v, n_1, agregarRegistro(r, n_2, b)$)	\equiv if $n_1 = n_2 \wedge_L r[clave(obtener(n_1, b))] = v$ then b else eliminarRegistro(v, n_1, b) fi


```

realizarConsulta( $q, b$ )  $\equiv$  if tipo_consulta( $q$ ) = FROM then
    from(nombre_tabla( $q$ ),  $b$ )
else
    if tipo_consulta( $q$ ) = SELECT then
        select(realizarConsulta(subconsulta1( $q$ ),  $b$ ), campo1( $q$ ), valor( $q$ ))
    else
        if tipo_consulta( $q$ ) = MATCH then
            match(realizarConsulta(subconsulta1( $q$ ),  $b$ ), campo1( $q$ ),
                campo2( $q$ ))
        else
            if tipo_consulta( $q$ ) = PROJ then
                proj(realizarConsulta(subconsulta1( $q$ ),  $b$ ), conj_campos( $q$ ))
            else
                if tipo_consulta( $q$ ) = RENAME then
                    rename(realizarConsulta(subconsulta1( $q$ ),  $b$ ), campo1( $q$ ),
                        campo2( $q$ ))
                else
                    if tipo_consulta( $q$ ) = INTER then
                        inter(realizarConsulta(subconsulta1( $q$ ),  $b$ ),
                            realizarConsulta(subconsulta2( $q$ ),  $b$ ))
                    else
                        if tipo_consulta( $q$ ) = UNION then
                            union(realizarConsulta(subconsulta1( $q$ ),  $b$ ),
                                realizarConsulta(subconsulta2( $q$ ),  $b$ ))
                        else
                            product(realizarConsulta(subconsulta1( $q$ ),  $b$ ),
                                realizarConsulta(subconsulta2( $q$ ),  $b$ ))
                    fi
                fi
            fi
        fi
    fi

```

from(n, b)	\equiv	if $\neg \text{def?}(n, \text{tablas}(b))$ then \emptyset else registros(obtener(n, b)) fi
select(cr, c, v)	\equiv	if $\emptyset?(cr)$ then \emptyset else if $c \in \text{campos}(\text{dameUno}(cr)) \wedge_{\text{L}} \text{dameUno}(cr)[c] = v$ then Ag($\text{dameUno}(cr), \text{select}(\text{sinUno}(cr), c, v)$) else select($\text{sinUno}(cr), c, v$) fi fi
match(cr, c_1, c_2)	\equiv	if $\emptyset?(cr)$ then \emptyset else if $(c_1 \in \text{campos}[\text{dameUno}(cr)] \wedge c_2 \in \text{campos}[\text{dameUno}(cr)])$ $\wedge_{\text{L}} \text{dameUno}(cr)[c_1] = \text{dameUno}(cr)[c_2]$ then Ag($\text{dameUno}(cr), \text{match}(\text{sinUno}(cr), c_1, c_2)$) else match($\text{sinUno}(cr), c_1, c_2$) fi fi
filtrarCampos(r_1, r_2, cs)	\equiv	if $\emptyset?(cs)$ then r_2 else if $\text{dameUno}(C) \in \text{campos}(r_1)$ then definir(filtrarCampos($r_1, r_2, \text{sinUno}(cs)$), $\text{dameUno}(cs)$, $r_1[\text{dameUno}(cs)]$) else filtrarCampos($r_1, \text{sinUno}(cs), r_2$) fi fi
proj(cr, cs)	\equiv	if $\emptyset?(cr)$ then \emptyset else Ag(filtrarCampos($\text{dameUno}(cr)$, nuevo, C), proj($\text{sinUno}(cr), cs$)) fi
rename(cr, c_1, c_2)	\equiv	if $\emptyset?(cr)$ then \emptyset else if $c_1 \notin \text{campos}(\text{dameUno}(cr)) \vee c_2 \in \text{campos}(\text{dameUno}(cr))$ then Ag($\text{dameUno}(cr), \text{rename}(\text{sinUno}(cr), c_1, c_2)$) else Ag($\text{renombrarCampo}(c_1, c_2, \text{campos}(\text{dameUno}(cr)),$ $\text{dameUno}(cr), \text{nuevo}, \text{rename}(\text{sinUno}(cr), c_1, c_2))$) fi fi

renombrarCampo(c_1, c_2, cs, r_1, r_2)	\equiv if $\emptyset?(cs)$ then r_2 else if $\text{dameUno}(cs) = c_1$ then renombrarCampo($c_1, c_2, \text{sinUno}(cs), r_1,$ definir($r_2, c_2, r_1[c_1]$)) else renombrarCampo($c_1, c_2, \text{sinUno}(cs), r_1,$ definir($res, \text{dameUno}(cs), r_1[\text{dameUno}(cs)]$)) fi fi
inter(cr_1, cr_2)	$\equiv cr_1 \cap cr_2$
union(cr_1, cr_2)	$\equiv cr_1 \cup cr_2$
product(cr_1, cr_2)	\equiv if $\emptyset?(cr_1) \vee \emptyset?(cr_2)$ then \emptyset else emparejarUnoAMuchos($\text{dameUno}(cr_1), cr_2$) \cup product($\text{sinUno}(cr_1), cr_2$) fi
emparejarUnoAMuchos(r, cr)	\equiv if $\emptyset?(cr)$ then \emptyset else Ag($\text{emparejarUnoAUno}(r, \text{dameUno}(cr)),$ $\text{emparejarUnoAMuchos}(r, \text{sinUno}(cr))$) fi
emparejarUnoAUno(r_1, r_2)	$\equiv \text{poblarRegistro}(r_1, r_2, \text{campos}(r_2))$
poblarRegistro(r_1, r_2, cs)	\equiv if $\emptyset?(cs)$ then r_1 else poblarRegistro($\text{definir}(r_1, \text{dameUno}(cs),$ $r_2[\text{dameUno}(cs)], r_2, \text{sinUno}(cs))$) fi

Fin TAD