2.3.- Modelo relacional de datos (aproximación lógica)

Existen dos lenguajes lógicos de manipulación para el modelo relacional:

- El Cálculo Relacional de Tuplas.
- El Cálculo Relacional de Dominios.

La perspectiva lógica del modelo relacional de datos permite realizar consultas y establecer restricciones.

El **Cálculo Relacional de Tuplas** es en el que se basa el lenguaje de manipulación SQL.

2.3.1.- La lógica de 1er orden.

LÓGICA de 1er ORDEN

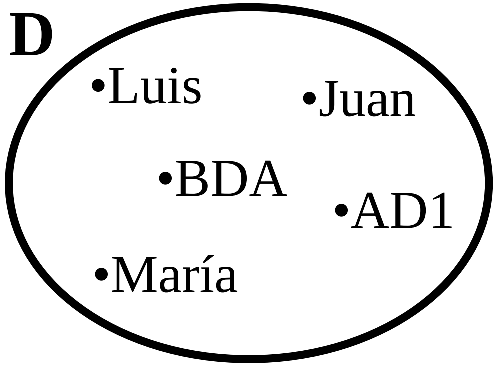
- “Sistema formal que permite razonar sobre un universo de discurso”

- La lógica de 1er orden es un lenguaje formal. Por tanto, posee dos elementos:
  - Un **lenguaje** en el cual se pueden expresar aserciones sobre el universo de interés. (SINTAXIS)
  - Unas **reglas** con las cuales determinar el valor de verdad de las aserciones o afirmaciones realizadas. (SEMÁNTICA)

Ejemplo: “Mortal(Sócrates)” y “Planeta(Sócrates)” son sintácticamente correctas, pero sólo la primera es cierta semánticamente en nuestro mundo.

2.3.1.- La lógica de 1er orden

D



P:

- ser un alumno
- ser una asignatura
- estar matriculado un alumno en una asignatura

ASERCIÓN: “Todos los alumnos están matriculados de alguna asignatura”

¿Es cierta esta aserción?

Se necesita tener más información sobre las propiedades de ***P*** en el dominio ***D***

Si esta información es:

- es-alumno = {Juan, Luis, María}
- es-asignatura = {AD1, BDA}
- está-matriculado = {(Luis,AD1), (Juan,BDA)}

La aserción es <u>falsa</u> para el conocimiento que se tiene de las propiedades ***P*** en el dominio ***D***

2.3.1.- La lógica de 1er orden

FORMALIZACIÓN (SINTAXIS):

- Se debe definir un lenguaje de 1er orden ***L*** para poder referirse a los individuos y a las propiedades del universo de discurso:

  L:

  Constantes = {Luis, María, Juan, AD1, BDA}

  Predicados = {Alumno(.), Asignatura(.), Matrícula(.,.)}

  Variables = {x, y}

  Conectivas = $\{\rightarrow, \neg, \wedge, \vee\}$

  Cuantificadores = $\{\forall, \exists\}$

- Con este lenguaje se podrán realizar afirmaciones y consultas.

2.3.1.- La lógica de 1er orden

FORMALIZACIÓN (SINTAXIS):

FÓRMULAS BIEN FORMADAS DE LA LÓGICA DE 1er ORDEN.

Una **condición** es una expresión que puede ser:

- Comparación: son expresiones de la forma *X* *a* *Y*

  en donde

  » *a* es un operador de comparación

  » *X* e *Y* son constantes o variables.

- Condición de pertenencia: son expresiones de la forma *R(x1, ..., xn)*

  en donde

  » *R* es un predicado n-ario.

  » *V* es una variable-tupla.

2.3.1.- La lógica de 1er orden

Con lo dicho, las **fórmulas bien formadas** (abreviado **fbf**) se pueden definir mediante las siguientes reglas:

- Toda condición es una fbf.
- Si ***F*** es una fbf, entonces ***(F)*** y ***ØF*** son fbf.
- Si ***F*** y ***G*** son fbf, entonces también lo son $\boldsymbol{F \wedge G}$, $\boldsymbol{F \vee G}$ y $\boldsymbol{F \rightarrow G}$
- Si ***F*** es una fbf y ***X*** un símbolo de variable, entonces ***"X F*** y ***$X F*** son fbf.
- Nada más es una fbf.

2.3.1.- La lógica de 1er orden

FORMALIZACIÓN (SINTAXIS):

- Se debe definir un lenguaje de 1er orden ***L*** para poder referirse a los individuos y a las propiedades del universo de discurso:

  L:

  Constantes = {Luis, María, Juan, AD1, BDA}

  Predicados = {Alumno(.), Asignatura(.), Matrícula(.,.)}

  Variables = {x, y}

  Conectivas = {$\rightarrow$, $\neg$, $\wedge$, $\vee$}

  Cuantificadores = {$\forall$, $\exists$ }

- Ejemplo de fórmulas sintácticamente correctas:

  F: $\forall$x (Alumno(x) $\rightarrow$ $\exists$y Matrícula(x, y))

  G: Alumno(x) ^ Matrícula(x,'AD1')

- Ejemplo de fórmula sintácticamente incorrecta:

  F': $\forall$Alumno $\exists$Matrícula(x, y)

2.3.1.- La lógica de 1er orden

FORMALIZACIÓN (SEMÁNTICA):

- La interpretación ***I*** del lenguaje de 1^{er} orden en el dominio ***D*** *correspondiente al ejemplo anterior*:

  D = {Luis, María, Juan, AD1, BDA}

  Alumno = {Juan, Luis, María}

  Asignatura = {AD1, BDA}

  Matrícula = {(Luis, AD1), (Juan, BDA)}

- La evaluación de **F:** $\forall x\ (Alumno(x) \rightarrow \exists y\ Matrícula(x, y))$ en ***I*** se realiza siguiendo unas reglas fijas.

2.3.1.- La lógica de 1er orden

EVALUACIÓN DE FÓRMULAS EN LA LÓGICA DE 1er ORDEN (SEMÁNTICA).

Dada:

- Una fórmula *F*.
- Una interpretación I.
- Un dominio D.
- Una asignación de valores a las variables libres de *F*.

Las reglas para la evaluación de *F* son las siguientes:

2.3.1.- La lógica de 1er orden

1) Sea *F* una comparación:

   - si *F* es de la forma *X* ***a*** *Y* donde *X* y *Y* son constantes o variables, entonces *F* se evalúa al valor de certeza de la comparación.

2) Si *F* es un predicado n-ario de la forma *R(x1, ..., xn)*, entonces *F* se evalúa a cierto si *(x1, ..., xn)* pertenece a la interpretación de R en I; en caso contrario, *F*, se evalúa a falso.

3) Si *F* es de la forma *(G)*, *F* se evalúa al valor de certeza de *G*.

2.3.1.- La lógica de 1er orden

4) Si *F* es de una de las siguientes formas ¬*G*, *G*∧*H*, *G*∨*H* o *G*→*H* donde *G* y *H* son fórmulas bien formadas, entonces *F* se evalúa de acuerdo a las siguientes tablas de verdad:

G	**H**	**F = G Ù H**	**F = G Ú H**	**F = G ® H**
falso	falso	falso	falso	cierto
indefinido	falso	falso	indefinido	indefinido
cierto	falso	falso	cierto	falso
falso	indefinido	falso	indefinido	cierto
indefinido	indefinido	indefinido	indefinido	indefinido
cierto	indefinido	indefinido	cierto	indefinido
falso	cierto	falso	cierto	cierto
indefinido	cierto	indefinido	cierto	cierto
cierto	cierto	cierto	cierto	cierto

G	**F = ¬ G**
falso	cierto
indefinido	indefinido
cierto	falso

2.3.1.- La lógica de 1er orden

5) Si F es de la forma $\$x\ G$, entonces F es cierta si existe una asignación para la variable x que hace cierta la fórmula G.

6) Si F es de la forma $"x\ G$, entonces F es cierta si para toda asignación de la variable x, la fórmula G es cierta.

2.3.1.- La lógica de 1er orden

FORMALIZACIÓN (SEMÁNTICA):

- La interpretación ***I*** del lenguaje de 1er orden en el dominio ***D*** *correspondiente al ejemplo anterior*:

  D = {Luis, María, Juan, AD1, BDA}

  Alumno = {Juan, Luis, María}

  Asignatura = {AD1, BDA}

  Matrícula = {(Luis, AD1), (Juan, BDA)}

- La evaluación de **F:** $\forall x\ (Alumno(x) \rightarrow \exists y\ Matrícula(x, y))$.

2.3.1.- La lógica de 1er orden

EVALUACIÓN DE UNA FÓRMULA ABIERTA (SEMÁNTICA).

Las fórmulas cerradas se utilizan para realizar afirmaciones.

Las fórmulas abiertas se utilizan para realizar consultas.

EJEMPLOS:

Alumno = {Juan, Luis, María}

Asignatura = {AD1, BDA}

Matrícula = {(Luis, AD1), (Juan, BDA)}

Una fórmula cerrada:

$\forall x$ (Alumno(x) $\rightarrow \exists y$ Matrícula(x, y)). ➠

Una fórmula abierta:

Alumno(x) ^ Matrícula(x, 'AD1')). ➠

2.3.1.- La lógica de 1er orden

EVALUACIÓN DE UNA FÓRMULA ABIERTA (SEMÁNTICA).

EJEMPLOS:

Alumno = {Juan, Luis, María}

Asignatura = {AD1, BDA}

Matrícula = {(Luis, AD1), (Juan, BDA)}

Una fórmula abierta:

Alumno(x) ^ Matrícula(x, ‘AD1’)).

EVALUACIÓN:

Consiste en buscar los valores del dominio que, asignados a las variables libres (en este caso *x*) hacen cierta la fórmula.

2.3.2.- Interpretación lógica de una base de datos relacional

Una **interpretación** consiste en asociar a cada predicado n-ario del lenguaje una relación n-aria definida sobre el dominio ***D***:

Alumno

Juan
Luis
María

Asignatura

AD1
BDA

Matrícula

Luis	AD1
Juan	BDA

Por tanto, una interpretación de un lenguaje de 1er orden puede verse como una base de datos relacional en la que:

- Los nombres de relación coinciden con los predicados.
- Los dominios de los atributos coinciden con las constantes.

2.3.2.- Interpretación lógica de una base de datos relacional

Provincia

pcod	nomprov
44	Teruel
46	Valencia
16	Cuenca
12	Castellón

Río

rcod	nombre
r1	Sénia
r2	Túria
r3	Xúquer

Pasa_por

pcod	rcod
44	r1
46	r2
30	r2
20	r1
44	r3
12	r1

Predicados: {Provincia(.,.) Río(.,.) Pasa_por(.,.)}

Interpretación: Las extensiones de las relaciones de la BDA

F1: Río(x,y) ^ Pasa_por(44,x) ⇒

F2: Río(x,y) ^ ¬∃z Pasa_por(z,x) ⇒

2.3.2.- Interpretación lógica de una base de datos relacional

Variable tupla:

- Son variables que se declaran sobre las relaciones de la base de datos *Variable_tupla:Nombre_relación*.
- Sus posibles valores se restringen a las tuplas de la extensión de la relación sobre la que se definieron.
- Sus componentes pueden referirse *Variable_tupla.Atributo_relación*.

EJEMPLO:

Río(rcod:tira(6), nombre:tira(20))

RX : Río

posibles valores para ***RX***:

rcod	**nombre**
r1	Sénia
r2	Túria
r3	Xúquer

{(rcod: “r1”), (nombre: “Sénia”)}
{(rcod: “r3”), (nombre: “Xúquer”)}
~~{(rcod: “xx”), (nombre: “xftrfsdh”)}~~
~~{(rcod: “r2”), (nombre: “Tajo”)}~~
~~{(rcod: “r3”), (nombre: “Túria”)}~~

2.3.2.- Interpretación lógica de una base de datos relacional

Consultas con variables tupla:

- Las consultas con variables tupla tienen la siguiente forma:

  {*Declaración_variables_libres*|*Fórmula_bien_formada*}

- Los ejemplos escritos en lógica de 1er orden se pueden rescribir con variables tupla de la siguiente manera:

Lógica de 1er orden:

F1: Río(x,y) ^ Pasa_por(44,x)

F2: Río(x,y) ^ ¬∃z Pasa_por(z,x)

Lógica de 1er orden con variables tupla:

F1: RX:Río | ∃PPX:Pasa_por (PPX.rcod = RX.rcod ^ PPX.pcod = 44)

F2: RX:Río | ¬∃PPX:Pasa_por (RX.rcod = PPX.rcod)

2.3.2.- Interpretación lógica de una base de datos relacional

Consultas con variables tupla:

EJEMPLOS:

Río(rcod:dom_rcod, nombre:dom_nom, longitud: dom_lon)

Provincia(pcod:dom_pcod, nombre:dom_nom)

Pasa_por(pcod:dom_pcod, rcod:dom_rcod)

¿Qué ríos pasan por al menos dos provincias?

¿Qué ríos pasan por todas las provincias?