

## U.T. VI.- BBDD y Valores Nulos.

### 1.- Introducción.

- Entendemos por valor **NULO** o **AUSENTE** una señal o marca utilizada para representar información desconocida por inaplicable o inexistente.
- **Necesidad** de los valores nulos en las BBDD:
  - o Para poder almacenar tuplas de las que se desconozca el valor de algunos de sus atributos.
  - o Para poder utilizar atributos que sean inaplicables a ciertas tuplas.
  - o Para poder añadir un atributo nuevo a una relación. En el momento de añadir dicho atributo, no tendrá ningún valor para las tuplas de la relación.
  - o [Ejemplo, dada la tabla:](#)

[Persona\(DNI, NSS, Nombre, Dirección\)](#)

- [Si queremos añadir datos de una persona desconociendo su dirección \(dato desconocido\).](#)
- [Si queremos añadir datos de una persona que nunca ha trabajado \(dato inaplicable\).](#)
- [Si queremos modificar la tabla para añadirle el atributo Ciudad \(dato nuevo\)](#)

### 2.- Lógica trivaluada.

- Dado que al comparar dos atributos, conteniendo al menos uno de ellos un valor nulo, no se puede deducir si el resultado de la comparación es cierto o falso ( $A > B$ ,  $A = B$ , etc.), es necesario introducir un nuevo valor lógico, el valor **QUIZÁ**.

- Surge así la denominada “**Lógica Trivaluada**”: Mientras que hasta ahora entendíamos por expresión lógica aquella que al evaluarla daba como resultado **CIERTO** o **FALSO**, en la lógica trivaluada una expresión lógica es aquella que al evaluarla puede dar como resultado **CIERTO**, **FALSO** o **QUIZÁ**.
- Dados dos operandos lógicos A y B, las tablas de verdad de los operadores lógicos serán ahora:

A	NOT A
C	F
Q	Q
F	C

A	B	A OR B
C	C	C
C	Q	C
C	F	C
Q	C	C
Q	Q	Q
Q	F	Q
F	C	C
F	Q	Q
F	F	F

A	B	A AND B
C	C	C
C	Q	Q
C	F	F
Q	C	Q
Q	Q	Q
Q	F	F
F	C	F
F	Q	F
F	F	F

- La lógica trivaluada obliga a definir las expresiones de comparación y aritméticas de una manera específica para el caso de que el valor de alguno de los operandos sea nulo:
  - o El resultado de una expresión de comparación es **cierto** o **falso**, excepto si alguno de los operandos, o ambos, son

nulos, en cuyo caso el resultado de la comparación es **quizá**.

- Ejemplo: Si  $A=\text{cierto}$  y  $B=\text{null}$  entonces  $A>B$  es quizá.
- o El resultado de una expresión aritmética (operandos numéricos unidos por los operadores  $+$ ,  $-$ ,  $*$ ,  $/$ ) es nulo cuando alguno de los operandos es nulo.
  - Ejemplo Si  $A=3$  y  $B=\text{null}$  entonces  $a+b=\text{null}$
- También obliga a definir un nuevo operador lógico: **IS NULL**, que toma el valor **cierto** si el operando al que se aplica es nulo y **falso** en caso contrario.
  - Ejemplo:  $A=3$ ,  $B=\text{null}$ , **IS NULL A** es falso, **IS NULL B** es cierto.

### Ejercicio 1:

1. Indica el valor de las siguientes expresiones:

- a)  $\text{NOT}((6>4) \text{ OR } (3<2))$
- b)  $(7>4) \text{ AND } (5<>3)$
- c)  $(7>4) \text{ OR } (5=3)$

2. Siendo  $A = \text{Falso}$ ;  $B = \text{Cierto}$  y  $C = \text{Falso}$ , evaluar:

- a)  $\text{NOT } A \text{ AND } B \text{ OR } C$
- b)  $\text{NOT } A \text{ AND } (B \text{ OR } C)$
- c)  $\text{NOT } (A \text{ AND } B \text{ OR } C)$

3. Siendo  $A = \text{Falso}$ ;  $B = \text{Cierto}$  y  $C = \text{Falso}$ , evaluar:

- a)  $A \text{ AND } \text{Cierto} \text{ OR } \text{NOT } C \text{ AND Falso} \text{ OR } B$
- b)  $A \text{ AND } (\text{Cierto} \text{ OR } \text{NOT } C) \text{ AND } (\text{Falso} \text{ OR } B)$
- c)  $A \text{ AND } \text{Cierto} \text{ OR } \text{NOT } (C \text{ AND Falso} \text{ OR } B)$

### Ejercicio 2:

1. Evaluar las siguientes expresiones lógicas:

- a) NOT((6>4) OR (3<Nulo))
- b) (Nulo>4) AND (5<>3)
- c) (7>4) OR (5=Nulo)

2. Siendo A = Falso; B = Cierto y C = Quizá, evaluar:

- a) NOT A AND B OR C
- b) NOT A AND (B OR C)
- c) NOT (A AND B OR C)

3. Siendo A = Quizá; B = Cierto y C = Falso, evaluar:

- a) A AND Cierto OR NOT C AND Quizá OR B
- b) A AND (Cierto OR NOT C) AND ( falso OR B)
- c) A AND Quizá OR NOT (C AND falso OR B)

- **Por otro lado**, la utilización de valores nulos también plantea problemas a la hora de aplicar determinadas funciones, como las estadísticas, a los atributos de una relación.

- o Ejemplo, dada la tabla

Ref	Artículo	Unid
12	Mesa	10
25	Silla	15
26	Mesa	
29	Mesa	3
34	Silla	12
37	Sillón	

- o Si queremos hallar la media aritmética de las unidades, no es lo mismo no tener en cuenta los valores nulos que considerarlos 0:
  - En el primer caso:  $(10+15+3+12)/4 = 10$
  - En el segundo caso:  $(10+15+0+3+12+0)/6 = 6,67$
- **Como consecuencia** de todo lo anterior, es aconsejable diseñar BD evitando, en la medida de lo posible, los valores nulos, especificando siempre que se pueda en todos los atributos la restricción **NOT NULL**.
- En el caso de tener que aceptar valores nulos, habrá que prestar atención al resultado de las operaciones en las que estén implicados, así como estudiar el tratamiento que hace de los valores nulos el SGBD que se esté utilizando, que no está estandarizado.

### 3.- Regla de Integridad de Tabla en MySQL.

- El estándar de las BBDD relacionales exige que todas sus relaciones o tablas cumplan la denominada Regla de Integridad:
  - o "Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo, es decir, un valor desconocido o inexistente".
- Debido a ello, si se intenta insertar en una tabla un registro con un valor nulo en alguno de los atributos que forman su clave principal, el SGBD deberá emitir un mensaje de error y no realizar la inserción.
- Si utilizamos el SGBD MySQL debemos tener en cuenta que se salta esta regla del modo siguiente:
  - o Si se intenta insertar un registro en una tabla sin asignar un valor a algún campo sobre el que se haya

definido una restricción PRIMARY KEY o NOT NULL, MySQL sustituye los nulos por ceros si el/los campos son de tipo numérico o por blancos si son de tipo texto.

### 3.- Operadores relacionales con valores nulos.

- **select from where:** Únicamente se obtienen resultados para aquellas tuplas que hacen cierta la cláusula **where**, desechando aquéllas para las que es falsa o quizá.

- o Ejemplo:

```
use compraventa;  
select idpedido,fechaenvío from pedidos  
where fechaenvío>'1998-05-01';
```

### Ejercicio 3.

Obtener el identificador de todos los pedidos que no tienen fecha de envío.

- Combinación interna (**inner join**): Únicamente se obtienen las tuplas para las que la expresión lógica es cierta, desechando aquéllas para las que es falsa o quizá.

- o Ejemplo:

```
select e1.apellido empleado,e2.apellido jefe  
from empleados e1 join empleados e2 on  
e1.jefe=e2.idempleado
```

- No aparece Fuller como empleado dado que para esa tupla e1.jefe es null, por lo que el resultado de la comparación es quizá.

### Ejercicio 4:

¿Qué ocurre con la combinación externa izquierda (**left join**)?

¿Y con la combinación externa derecha (**right join**)?

Nota: Mysql no soporta full join.

- Funciones de Agregado (**group by**).
  - o Los nulos en un campo objeto de agrupación son tratados como si tuvieran el mismo valor.
    - Esto es incoherente con la lógica trivaluada, que nos dice que el resultado de null=null es quizá.
  - o Los valores null no son tenidos en cuenta en las funciones de agregado count(),sum(),avg(),max() y min().
  - o Si todos los valores son null, las funciones de agregado devuelven null excepto count(), que devuelve 0.

### Ejercicio 5.

Crear una BD con la siguiente tabla:

Ref	Artículo	Unid
12	Mesa	10
25	Silla	15
26	Mesa	
29	Mesa	3
34	Silla	12
37	Sillón	
39		10
42		20

Obtener el resultado de las siguientes sentencias y explicarlo:

- `select articulo,count(unid),sum(unid),avg(unid) from Muebles where ref in(39,42) group by articulo;`
- `select articulo,count(unid),sum(unid),avg(unid) from Muebles group by articulo;`
- `select count(unid),max(unid),min(unid) from muebles;`

- `select count(unid),max(unid),min(unid) from muebles  
where ref in(26,37);`