Lenguaje SQL: Parte III

Base de Datos

Mónica Caniupán
mcaniupan@ubiobio.cl

Universidad del Bío-Bío

2020

Contenidos

- Valores Nulos
- Reuniones Externas

Valores Nulos

- En la práctica los valores de las columnas pueden ser desconocidos
- Ejemplo: Consideremos la siguiente relación:

PERSONAS			
ID	NOMBRE	EDAD	NOMBRECONYUGE
11	*Pedro*	21	*Maria*
12	*Luis*	22	*Sandra*
13	*Juan*	20	*Paola*

- Se desea insertar una tupla en la relación pero:
  1. No conocemos la edad de una persona. ¿Qué valor le asignamos al atributo EDAD?
  2. ¿Qué valor le asignamos al atributo NOMBRECONYUGE si la nueva persona es soltera?

Valores Nulos

- SQL ofrece un valor especial para las columnas denominado *NULL* (nulo) para emplearlo en estas situaciones
- El valor *NULL* significa *desconocido* o *no aplicable*
- La siguiente operación es válida:
  INSERT INTO PERSONAS VALUES(14,'Enrique',NULL,NULL)
- Sin embargo, los valores nulos en las bases de datos producen un impacto en la evaluación de consultas

Comparaciones que Emplean Valores Nulos

- Consideremos una comparación como $Edad = 20$ evaluada sobre la siguiente relación:

PERSONAS			
ID	NOMBRE	EDAD	NOMBRECONYUGE
11	*Pedro*	21	*Maria*
12	*Luis*	22	*Sandra*
13	*Juan*	20	*Paola*
14	*Enrique*	*NULL*	*NULL*

- ¿Es la condición $Edad = 20$ verdadera o falsa en la fila de *Enrique*?

Comparaciones que Emplean Valores Nulos

- El resultado debería ser *desconocido*
- De hecho éste es el caso para cualquier comparación con operadores $\{<, >, =, \neq\}$, que involucre valores nulos
- Más aún, si comparamos dos valores nulos con $\{<, >, =, \neq\}$ el resultado siempre es *desconocido*
- SQL ofrece el operador de comparación especial IS NULL para verificar si el valor de un atributo es *NULL*
- Por ejemplo, EDAD IS NULL evaluado en la fila de *Enrique* es *verdadero*
- También se puede usar IS NOT NULL, e.g. EDAD IS NOT NULL evaluado en la fila de *Enrique* es *falso*

Operaciones Booleanas con Nulos

- Consideremos la tupla con nulos PERSONA(14,Enrique,NULL,NULL)
  - $EDAD > 20$ OR $NOMBRE =' Enrique'$

    es *verdadero*, porque $NOMBRE =' Enrique'$ es verdad
  - $EDAD > 20$ OR $NOMBRE =' Pedro'$

    es *desconocido* porque la primera comparación es *desconocido* y la segunda es *falsa*
- En la presencia de valores nulos, hay que definir los operadores lógicos AND,OR y NOT mediante una lógica de tres valores en la que las expresiones toman el valor de verdadero, falso o desconocido

Operaciones Booleanas con Nulos

- NOT desconocido es desconocido
- OR de dos argumentos es:
  - verdadero si uno de los argumentos es verdadero
  - desconocido si uno de los argumentos es falso y el otro es desconocido
  - falso si los dos argumentos son falsos
- AND de dos argumentos es:
  - falso si uno de los argumentos es falso
  - desconocido si uno de los argumentos es desconocido y el otro es verdadero o desconocido
  - verdadero si los dos argumentos son verdaderos

Consecuencias para las Estructuras de SQL

- Duplicidad de tuplas: Dos tuplas de una relación se consideran *iguales* si sus atributos tienen el mismo valor o ambas contienen nulos
  - Sin embargo, si comparamos dos valores nulos usando el símbolo de igualdad obtenemos *desconocido* (*NULL* = *NULL* es siempre *desconocido*)
- Los operadores aritméticos $\{+, -, *, /\}$ retornan *NULL* si uno de los argumentos es *NULL*
- Comportamientos inesperados de las operaciones de agregación:
  - COUNT(*) maneja el valor *NULL* como cualquier otro valor
  - Todas las demás operaciones de agregación COUNT, SUM, AVG, MIN MAX y las variaciones usando DISTINCT descartan los valores nulos
  - Como caso especial, si uno de estos operadores, que no sea COUNT, se aplica sólo a valores nulos, el resultado es *NULL*

Ejemplo: Operaciones con NULL

- Consideremos la siguiente relación:

PERSONAS			
ID	NOMBRE	EDAD	NOMBRECONYUGE
11	*Pedro*	21	*Maria*
12	*Luis*	22	*Sandra*
13	*Juan*	20	*Paola*
14	*NULL*	*NULL*	*NULL*

- La respuesta a:
  SELECT COUNT (*)
  FROM PERSONAS
  es 4
- La respuesta a:
  SELECT SUM(EDAD)
  FROM PERSONAS
  es 63

Evitando los Valores Nulos

- SQL nos permite prohibir que ciertos atributos tomen valores nulos
- Esta restricción se especifica en la definición de atributos:

```
CREATE TABLE .....
EDAD INTEGER NOT NULL
...
```

- Para cada atributo clave existe una restricción NOT NULL implícita

Contenidos

- √ Valores Nulos
- Reuniones Externas

Variantes de Joins

- SQL soporta algunas variedades interesantes de la operación join que aprovechan los valores nulos, las que se denominan Outer Joins
- Considere la siguiente operación: *Navegantes* $\bowtie_{idn=idn}$ *Reservas*

Navegantes			
idn	nombre	edad	categoria
22	*Pedro*	45	4
23	*Andres*	35	6
33	*Loreto*	31	6
29	*Natalia*	40	7
30	*Esteban*	50	8

Reservas		
idn	idb	fecha
23	102	10.11.00
22	102	10.11.00
33	101	05.01.02

- El resultado es:

idn	nombre	edad	categoria	idn	idb	fecha
22	*Pedro*	45	4	22	102	10.11.00
23	*andres*	35	6	23	102	10.11.00
33	*loreto*	31	6	33	101	05.01.02

Variantes de Joins

- Sin embargo, podría ser interesante mantener las tuplas de *Navegantes* que no tienen reservas en el resultado. Para esto usamos el Outer Join
- Con Outer Join, las tuplas que no tienen reservas aparecen en el resultado del join y los atributos correspondientes a reservas toman valores nulos
- Existen tres variantes de Outer Join:
  1. Left outer join
  2. Right outer join
  3. Full outer join

LEFT OUTER JOIN

- Consideremos la consulta:

  SELECT *
  FROM Navegantes NATURAL LEFT OUTER JOIN RESERVAS

- El resultado es:

idn	nombre	edad	categoria	idn	idb	fecha
22	*Pedro*	45	4	22	102	10.11.00
23	*andres*	35	6	23	102	10.11.00
33	*loreto*	31	6	33	101	05.01.02
29	*natalia*	40	7	NULL	NULL	NULL
30	*esteban*	50	8	NULL	NULL	NULL

RIGHT OUTER JOIN

- Consideremos las siguientes relaciones:

Reservas		
idn	idb	fecha
23	102	10.11.00
22	102	10.11.00
33	101	05.01.02

Botes		
idb	nombreb	color
101	*marino*	azul
102	*inter-lagos*	rojo
103	*clipper*	verde
104	*inter-lagos*	rojo

- SELECT *
  FROM Reservas NATURAL RIGHT OUTER JOIN Botes

idn	idb	fecha	idb	nombreb	color
23	102	10.11.00	102	*inter-lagos*	rojo
22	102	10.11.00	102	*inter-lagos*	rojo
33	101	05.01.02	101	*marino*	azul
NULL	NULL	NULL	103	*clipper*	verde
NULL	NULL	NULL	104	*inter-lagos*	rojo

FULL OUTER JOIN

- Consideremos las siguientes relaciones:

Reservas		
idn	idb	fecha
23	102	10.11.00
22	102	10.11.00
33	101	05.01.02
33	106	06.01.02

Botes		
101	*marino*	azul
102	*inter-lagos*	rojo
103	*clipper*	verde
104	*inter-lagos*	rojo

- SELECT *
  FROM Reservas NATURAL FULL OUTER JOIN Botes

idn	idb	fecha	idb	nombreb	color
23	102	10.11.00	102	*inter-lagos*	rojo
22	102	10.11.00	102	*inter-lagos*	rojo
33	101	05.01.02	101	*marino*	azul
NULL	NULL	NULL	103	*clipper*	verde
NULL	NULL	NULL	104	*inter-lagos*	rojo
33	106	06.01.02	NULL	NULL	NULL