Bases de Datos

Clase 11: Nulos y Outer Joins

Información Incompleta

¿Qué hacer cuando no tenemos o perdemos información?

Esta es quizás la parte peor diseñada de SQL (y la más criticada)

Información Incompleta

“A recomendation: avoid nulls”

“Use [nulls] properly and they work for you, but abuse them, and they can ruin everything”

Información Incompleta

Peliculas	titulo	director	actor
	Django sin cadenas	Tarantino	Di Caprio
	Django sin cadenas	Tarantino	Waltz
	null	Tarantino	Thurman
	null	Tarantino	null
	El Hobbit	Jackson	McKellen
	Señor de los Anillos	null	McKellen

¿Qué significan los nulos en este caso?

Nulos

Significado

En el caso anterior puede significar que no se dispone de la información, pero existe!

En general los nulos pueden significar:

- Valor existe, pero no tengo la información
- Valor no existe (si `galardones = null` la información no existe)
- Ni si quiera sé si el valor existe o no

Nulos

Consultando con nulos

Sea la relación **R**(a, b) y las consultas:

- `SELECT * FROM R`
- `SELECT * FROM R WHERE R.b = 3 OR R.b <> 3`

Nulos

Consultando con nulos

Sea la relación **R**(a, b) y las consultas:

- `SELECT * FROM R`
- `SELECT * FROM R WHERE R.b = 3 OR R.b <> 3`

¿Son lo mismo?

Nulos

Consultando con nulos

Sea la relación **R**(a, b) y las consultas:

- `SELECT * FROM R`
- `SELECT * FROM R WHERE R.b = 3 OR R.b <> 3`

¿Son lo mismo?

Si `R.b` es nulo, ni `R.B = 3` o `R.B <> 3` evalúan a verdadero

Nulos

Consultando con nulos

La consulta:

```
SELECT * FROM R
```

Equivale a la unión de:

- `SELECT * FROM R WHERE R.b = 3`
- `SELECT * FROM R WHERE R.b <> 3`
- `SELECT * FROM R WHERE R.b IS NULL`

Nulos

Consultando con nulos

La consulta:

```
SELECT * FROM R
```

Equivale a la unión de:

- `SELECT * FROM R WHERE R.b = 3`
- `SELECT * FROM R WHERE R.b <> 3`
- `SELECT * FROM R WHERE R.b IS NULL`

Para ver si un elemento es nulo usamos `IS NULL`

Nulos

Operaciones con nulos

No podemos usar nulos explícitamente en operaciones:

- `SELECT 5 - NULL`
- `R.a = NULL`

Si algún argumento de una operación aritmética es nulo, el resultado es nulo

Nulos

Operaciones con nulos

Sean las siguientes instancias de **R** y **S**:

R A
1
null

S B
2
3

La consulta `SELECT R.A + S.B FROM R, S` retorna:

Respuesta $R.A + S.B$
3
4
null

Nulos

Operaciones con nulos

Sean las siguientes instancias de **R** y **S**:

R A	S B
1	2
null	3

Tenemos que `R.A = S.B` vale:

- `FALSE` cuando `R.A = 1` y `S.B = 2`
- `UNKNOWN` cuando `R.A = null` y `S.B = 2`

Lógica de tres valores

SQL usa lógica de tres valores:

x	NOT x
true	false
false	true
unknown	unknown

Lógica de tres valores

SQL usa lógica de tres valores:

AND	true	false	unknown
true	true	false	unknown
false	false	false	false
unknown	unknown	false	unknown

OR	true	false	unknown
true	true	true	true
false	true	false	unknown
unknown	true	unknown	unknown

Nulos

Ejemplo

R A	S A
1	1
2	2
	3
	4

```
SELECT S.A FROM S
WHERE S.A NOT IN (SELECT R.A FROM R)
```

Nulos

Ejemplo

R A	S A
1	1
2	2
	3
	4

```
SELECT S.A FROM S
WHERE S.A NOT IN (SELECT R.A FROM R)
```

Resultado: {3, 4}

Nulos

Ejemplo

R A
1
2
null

S A
1
2
3
4

```
SELECT S.A FROM S
WHERE S.A NOT IN (SELECT R.A FROM R)
```

Nulos

Ejemplo

R A
1
2
null

S A
1
2
3
4

```
SELECT S.A FROM S
WHERE S.A NOT IN (SELECT R.A FROM R)
```

Resultado: tabla vacía!

Lógica de tres valores

El resultado puede ser contraintuitivo pero es correcto

Veamos la evaluación de `3 NOT IN (SELECT R.A FROM R)`

$$
\begin{array}{rcl}
3 \texttt{ NOT IN } \{1, 2, \mathit{null}\} & & \\
= & \texttt{NOT } (3 \texttt{ IN } \{1, 2, \mathit{null}\} & \\
= & \texttt{NOT } (3 = 1 \texttt{ OR } 3 = 2 \texttt{ OR } 3 = \mathit{null}) & \\
= & \texttt{NOT (false OR false OR unknown)} & \\
= & \texttt{NOT (unknown)} & \\
= & \texttt{unknown} &
\end{array}
$$

Lógica de tres valores

Además `1 NOT IN (SELECT R.A FROM R)` es `FALSE` (Lo mismo para 2)

Para el valor 4 aplicamos el razonamiento anterior.

Resultado: tabla vacía

Nulos

Agregación

A
1
null

```
SELECT COUNT(*) FROM R vale 2

SELECT COUNT(R.A) FROM R vale 1
```

Nulos

Agregación

`SELECT SUM(R.A) FROM R` retorna 1 si `R.A = {1,null}`

Para funciones de agregación:

- Ignore todos los nulos
- Compute el valor de la agregación

La única excepción es **`COUNT(*)`**

Ejemplo

Ejemplo de Leonid Libkin, U. de Edimburgo

Escenario:

- Sistema de defensa de EEUU, con la base de datos de los misiles dirigidos a las grandes ciudades, y los misiles lanzados para interceptarlos
- ¿Hay un misil dirigido a EEUU que no ha sido interceptado aún?

Ejemplo

Ejemplo de Leonid Libkin, U. de Edimburgo

¿Hay un misil dirigido a EEUU que no ha sido interceptado aún?

```
SELECT M.num, M.objetivo
  FROM Misiles M
  WHERE M.objetivo IN (SELECT nombre
                       FROM CiudadesEEUU) AND
        M.num NOT IN (SELECT I.misil
                       FROM Interceptores I
                       WHERE I.status = ‘activo’)
```

Suponga que nos equivocamos en escribir el nombre del objetivo de un misil para interceptar

Ejemplo

Ejemplo de Leonid Libkin, U. de Edimburgo

Misiles

num	objetivo
666	Springfield
123	Washington
888	New York

Interceptores

id	misil	status
I1	666	activo
I2	null	activo

La consulta retorna vacío a pesar de que hay un misil sin interceptar!

`123 NOT IN {666, null}` y `888 NOT IN {666, null}` evalúan a desconocido

Nulos

NOT NULL

Para olvidarse de los nulos:

```
CREATE TABLE <nombre> (…
                       <atributo> <tipo> NOT NULL,
                       …)
```

Nulos

Solución

Tópico llamado Información Incompleta en Bases de Datos

No está resuelto, pero veamos los conceptos básicos

Tablas de Codd

Reemplace cada nulo por un valor distinto

Peliculas	titulo	director	actor
	Django sin cadenas	Tarantino	Di Caprio
	Django sin cadenas	Tarantino	Waltz
	x	Tarantino	Thurman
	y	Tarantino	z
	El Hobbit	Jackson	McKellen
	Señor de los Anillos	w	McKellen

Tablas de Codd

Semántica:

- Dada la tabla T, POS(T) es el conjunto de todas las tablas sin nulos que puede representar T
- En otras palabras, cada forma de reemplazar nulos por valores

Tablas de Codd

POS(T)

Peliculas	titulo	director	actor
	Django sin cadenas	Tarantino	Di Caprio
	Django sin cadenas	Tarantino	Waltz
	Kill Bill	Tarantino	Thurman
	Kill Bill	Tarantino	Carradine
	El Hobbit	Jackson	McKellen
	Señor de los Anillos	Jackson	McKellen

Tablas de Codd

POS(T)

Peliculas	titulo	director	actor
	Django sin cadenas	Tarantino	Di Caprio
	Django sin cadenas	Tarantino	Waltz
	Titanic	Tarantino	Thurman
	Titanic	Tarantino	Carradine
	El Hobbit	Jackson	McKellen
	Señor de los Anillos	Jackson	McKellen

Tablas de Codd

POS(T)

Peliculas	titulo	director	actor
	Django sin cadenas	Tarantino	Di Caprio
	Django sin cadenas	Tarantino	Waltz
	Gloria	Tarantino	Thurman
	Gloria	Tarantino	Reutter
	El Hobbit	Jackson	McKellen
	Señor de los Anillos	Reutter	McKellen

Tablas de Codd

Para la tabla anterior, ¿Qué nos arroja la consulta `SELECT título FROM Películas`?

Tablas de Codd

Para la tabla anterior, ¿Qué nos arroja la consulta `SELECT título FROM Películas`?

Respuesta	titulo
	Django sin cadenas
	Django sin cadenas
	x
	y
	El Hobbit
	Señor de los Anillos

Tablas de Codd

Suponga que **Q** es una consulta y **T** una tabla de Codd

Definimos:

$$Q'(T) = \{Q(R) | R \in Pos(T)\}$$

En cierta forma, son todas las tablas posibles para cada posible representación. Suponga que hay una tabla de Codd **T**' tal que:

$$Pos(T') = Q'(T)$$

Decimos que **T**' es nuestra respuesta

Tablas de Codd

Peliculas	titulo	director	actor
	Django sin cadenas	Tarantino	Di Caprio
	Django sin cadenas	Tarantino	Waltz
	x	Tarantino	Thurman
	y	Tarantino	z
	El Hobbit	Jackson	McKellen
	Señor de los Anillos	w	McKellen

```
SELECT título FROM Películas
```

$\hat{T}$	titulo
	Django sin cadenas
	Django sin cadenas
	x
	y
	El Hobbit
	Señor de los Anillos

Es posible demostrar que

$$Pos(T') = Q'(T) = \{Q(R)|R \in Pos(T)\}$$

Tablas de Codd

¿Siempre podemos encontrar **T**'?

Tablas de Codd

$$T = \begin{array}{cc} \text{A} & \text{B} \\ \hline 0 & 1 \\ x & 2 \end{array} \qquad \text{y} \qquad Q = \sigma_{A=3}(T)$$

Suponga que existe **T'** tal que $Pos(T') = \{Q(R) | R \in Pos(T)\}$

Tablas de Codd

$$R_1 = \begin{array}{cc} \text{A} & \text{B} \\ \hline 0 & 1 \\ 2 & 2 \end{array} \qquad \text{y} \qquad R_2 = \begin{array}{cc} \text{A} & \text{B} \\ \hline 0 & 1 \\ 3 & 2 \end{array}$$

Q(R1) es vacío y Q(R2) no lo es, pero

$$\emptyset \in POS(T') \; ssi \; T = \emptyset$$

Concluimos que **T**’ no puede existir!

Posibles soluciones

¿Cuál debería ser la respuesta?

- Respuesta certera: una tupla **t** es una respuesta certera de **Q** sobre una tabla de Codd **T** si t pertenece a la respuesta de **Q** en todas las relaciones $\mathbf{T}' \in$ POS(**T**)

- Extender tablas de Codd agregando restricciones para los valores nulos (ej. estos nulos son el mismo, o estos nulos representan distinta información, etc...)

Posibles soluciones

El problema es que los algoritmos de evaluación se vuelven demasiado complejos y los resultados de las consultas difíciles de comprender

¿Entonces, los nulos nos ayudan en algo?

Outer Joins

Motivación: calcular el ingreso neto de los estudios de Hollywood

Estudios

Nombre	Titulo
Warner	Argo
Warner	El Origen
MGM	El Hobbit

Peliculas

Titulo	Ingreso
Argo	136
El Origen	292
El Artista	44

Outer Joins

```
SELECT Estudio.nombre, SUM(Pelicula.ingreso)
FROM Estudio, Pelicula
WHERE Estudio.titulo = Pelicula.titulo
GROUP BY Estudio.nombre
```

- MGM se pierde porque El Hobbit no tiene con quien hacer Join
- ¿Qué pasa si no nos queremos deshacer de MGM?
- Por ahora no hay películas, pero sabemos que MGM es un estudio

Outer Joins

```
SELECT Estudio.nombre, SUM(Pelicula.ingreso)
FROM Estudio LEFT OUTER JOIN Pelicula
ON Estudio.titulo = Pelicula.titulo
GROUP BY Estudio.nombre
```

- Resultado: (Warner, 428), (MGM, **`null`**)

Left Outer Join

```
SELECT *
FROM Estudio LEFT OUTER JOIN Pelicula
ON Estudio.titulo = Pelicula.titulo
```

nombre	titulo	titulo	ingreso
Warner	Argo	Argo	136
Warner	El Origen	El Origen	292
MGM	El Hobbit	El Hobbit	null

Left Outer Join

```
SELECT * FROM Estudio NATURAL LEFT OUTER JOIN Pelicula
```

nombre	titulo	ingreso
Warner	Argo	136
Warner	El Origen	292
MGM	El Hobbit	null

NATURAL JOIN trabaja sobre los atributos con nombre en común

Right Outer Join

```
SELECT * FROM Estudio NATURAL RIGHT OUTER JOIN Pelicula
```

nombre	titulo	ingreso
Warner	Argo	136
Warner	El Origen	292
null	El Artista	44

Full Outer Join

```
SELECT * FROM Estudio NATURAL FULL OUTER JOIN Pelicula
```

nombre	titulo	ingreso
Warner	Argo	136
Warner	El Origen	292
MGM	El Hobbit	null
null	El Artista	44

Hacemos el Join y a las tuplas sin correspondencia le agregamos nulos

Outer Joins

- `R LEFT OUTER JOIN S`: mantenemos las tuplas de **R** que no tienen correspondencia
- `R RIGHT OUTER JOIN S`: mantenemos las tuplas de **S** que no tienen correspondencia
- `R FULL OUTER JOIN S`: mantenemos las tuplas de **R** y **S** que no tienen correspondencia

Outer Joins

Ojo! como resultados tienen nulos, hay problemas con usar agregación

```
SELECT Estudio.nombre, SUM(Pelicula.ingreso)
FROM Estudio LEFT OUTER JOIN Pelicula
ON Estudio.titulo = Pelicula.titulo
GROUP BY Estudio.nombre
```

- Resultado: (Warner, 428), (MGM, `null`)

Outer Joins

Ojo! como resultados tienen nulos, hay problemas con usar agregación

```
SELECT Estudio.nombre, COUNT(Pelicula.ingreso)
FROM Estudio LEFT OUTER JOIN Pelicula
ON Estudio.titulo = Pelicula.titulo
GROUP BY Estudio.nombre
```

- Resultado: (Warner, 2), (MGM, 0)