

Bases de Données : valeurs absentes

Stéphane Devismes

Université Grenoble Alpes

26 août 2020

Plan

- 1 mot-clé NULL
- 2 Jointures externes
- 3 NVL
- 4 Vues

Définition

Il arrive qu'une information (un élément d'un n -uplet) manque : cette information n'existe pas, ou on ne la connaît pas, ou on ne la connaît pas encore, ...

il s'agit d'une **valeur absente**.

Lorsqu'il y a des valeurs absentes, on doit utiliser les **jointures externes** à la place des jointures internes.

Remarque. Le traitement des valeurs absentes diffère un peu selon les implantations de SQL.

Attention au comportement de ALL et ANY en présence de valeurs absentes.

Le mot-clé NULL

Valeur absente en SQL : NULL

NULL ne fait partie d'aucun domaine.

Le mot-clé NULL jouant un rôle complètement différent des autres valeurs, il y a des règles spéciales pour le manipuler.

En principe, NULL n'est pas exprimable en algèbre relationnelle. Cependant, nous considérons ici **une algèbre relationnelle enrichie de la valeur NULL**.

Règles spéciales pour NULL (1/2)

- Le mot-clé NULL peut être utilisé pour tous les types de valeurs.
- Pour savoir si une valeur est NULL, on demande « est NULL ? » IS NULL, ou « est non NULL ? » IS NOT NULL, sans utiliser d'égalité.
- Si on teste une égalité entre deux valeurs, lorsque l'une au moins est NULL la réponse n'est ni « vrai » ni « faux », c'est NULL !

Donc, si $x = 4$ et y est NULL, alors la condition $x = y$ n'est pas vérifiée et la condition $x \neq y$ n'est pas vérifiée non plus !

Règles spéciales pour NULL (2/2)

- Les fonctions d'agrégation ignorent les valeurs NULL, à l'exception de COUNT(*) qui compte toutes les lignes, même celles qui contiennent des valeurs absentes.

Si toutes les valeurs de A sont absentes alors COUNT(A) renvoie 0, mais tous les autres, e.g., SUM(A), renvoient NULL dans ce cas.

- On peut remplacer NULL par une valeur (en respectant les types) :
 - NVL($e1, e2$) retourne $e1$ si $e1$ n'est pas NULL et retourne $e2$ sinon.
 - NVL2($e1, e2, e3$) retourne $e2$ si $e1$ pas NULL et retourne $e3$ sinon.

Définition

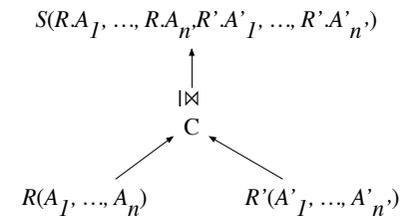
Dans une jointure interne, ou une jointure naturelle, si une ligne de R n'est appariée à aucune ligne de R' alors elle n'apparaît pas dans les jointures $S = R \bowtie_C R'$ ou $S = R \star R'$.

En utilisant une jointure externe, on force cette ligne à apparaître une fois dans S , et les données manquantes sont remplacés par la valeur NULL.

Jointure externe à gauche

Toutes les lignes $(x_1, \dots, x_{n_1}, y_1, \dots, y_{n_2})$ avec (x_1, \dots, x_{n_1}) dans R et (y_1, \dots, y_{n_2}) dans R' telles que $(x_1, \dots, x_{n_1}, y_1, \dots, y_{n_2})$ vérifie la condition C , et si besoin $(x_1, \dots, x_{n_1}, \text{NULL}, \dots, \text{NULL})$ de façon à ce que aucune ligne de R ne manque.

En AR :



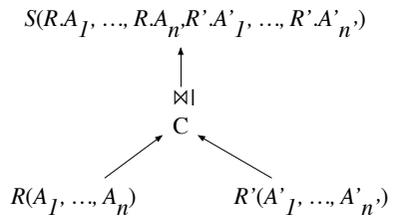
En SQL :

LEFT (OUTER) JOIN... ON/USING

Jointure externe à droite

Toutes les lignes $(x_1, \dots, x_{n_1}, y_1, \dots, y_{n_2})$ avec (x_1, \dots, x_{n_1}) dans R et (y_1, \dots, y_{n_2}) dans R' telles que $(x_1, \dots, x_{n_1}, y_1, \dots, y_{n_2})$ vérifie la condition C , et si besoin $(NULL, \dots, NULL, y_1, \dots, y_{n_2})$ de façon à ce que aucune ligne de R ne manque.

En AR :



En SQL :

RIGHT (OUTER) JOIN... ON/USING

Jointures externes naturelles

En AR :

gauche : $| * ,$ droite : $* | ,$ complet : $| * | .$

En SQL :

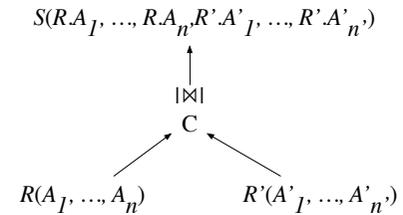
NATURAL LEFT/RIGHT/FULL (OUTER) JOIN

ATTENTION!!! mais pas LEFT/RIGHT/FULL NATURAL JOIN

Jointure externe complète

Toutes les lignes $(x_1, \dots, x_{n_1}, y_1, \dots, y_{n_2})$ avec (x_1, \dots, x_{n_1}) dans R et (y_1, \dots, y_{n_2}) dans R' telles que $(x_1, \dots, x_{n_1}, y_1, \dots, y_{n_2})$ vérifie la condition C , et si besoin $(x_1, \dots, x_{n_1}, NULL, \dots, NULL)$ et $(NULL, \dots, NULL, y_1, \dots, y_{n_2})$ de façon à ce que aucune ligne de R ni de R' ne manque.

En AR :



En SQL :

FULL (OUTER) JOIN... ON/USING

Jointure naturelle gauche : exemple

R ₁		
A	B	C
a	b	c
x	y	z
u	v	w
r	s	t

R ₂		
A	B	D
a	b	w
a	y	z
u	v	z
u	v	i

Jointure naturelle gauche entre R₁ et R₂

R ₃			
A	B	C	D
a	b	c	w
x	y	z	NULL
u	v	w	z
u	v	w	i
r	s	t	NULL

Jointure naturelle droite : exemple

R ₁		
A	B	C
a	b	c
x	y	z
u	v	w
r	s	t

R ₂		
A	B	D
a	b	w
a	y	z
u	v	z
u	v	i

Jointure naturelle droite entre R₁ et R₂

R ₃			
A	B	C	D
a	b	c	w
a	y	NULL	z
u	v	w	z
u	v	w	i

Jointure naturelle complète : exemple

R ₁		
A	B	C
a	b	c
x	y	z
u	v	w
r	s	t

R ₂		
A	B	D
a	b	w
a	y	z
u	v	z
u	v	i

Jointure naturelle complète entre R₁ et R₂

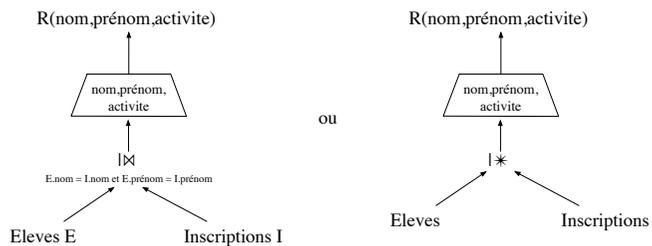
R ₃			
A	B	C	D
a	b	c	w
x	y	z	NULL
u	v	w	z
u	v	w	i
r	s	t	NULL
a	y	NULL	z

Jointure externe en AR : exemples (1/2)

Donner TOUS les élèves avec leurs activités.

- Schéma : R(nom : Chaîne, prénom : Chaîne, activite : Chaîne)
- Spécification : $\langle n, p, a \rangle \in R \iff$ L'élève n, p pratique l'activité a . Les élèves ne pratiquant aucune activité apparaissent avec $a \ll NULL \gg$.

Algèbre relationnelle :



Jointure externe en SQL : exemples (1/2)

Donner TOUS les élèves avec leurs activités.

```
-- avec LEFT JOIN ...ON
SELECT E.nom, E.prenom, activite
FROM Eleves E LEFT JOIN Inscriptions I
ON E.nom = I.nom AND E.prenom = I.prenom;
```

```
-- avec LEFT JOIN ...USING
SELECT nom, prenom, activite
FROM Eleves LEFT JOIN Inscriptions
USING (nom, prenom);
```

```
-- avec NATURAL LEFT JOIN
SELECT nom, prenom, activite
FROM Eleves NATURAL LEFT JOIN Inscriptions;
```

Jointure externe en AR : exemples (2/2)

Quel est le nombre d'activités pratiquées par chaque élève ?

- Schéma : $R(\underline{\text{nom}} : \text{Chaine}, \underline{\text{prénom}} : \text{Chaine}, \text{nbActivites} : \text{Nombre})$
- Spécification : $\langle n, p, na \rangle \in R \iff$ L'élève n, p pratique na activités. Les élèves ne pratiquant aucune activité apparaissent avec $na = 0$.



« Simuler » une jointure externe

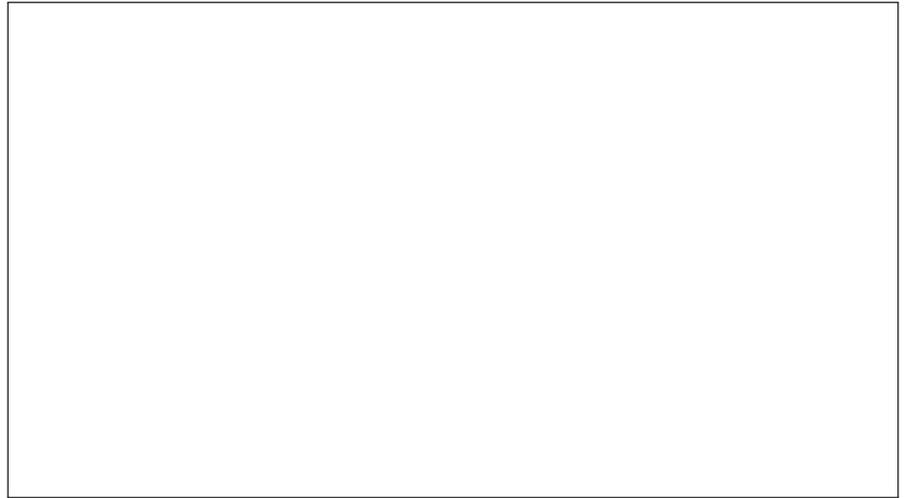
Pour chaque activité proposée par l'école, quel est le nombre d'élèves pratiquant cette activité ?

A priori, c'est symétrique de la requête précédente.

Mais il faudrait utiliser `COUNT(nom, prénom)`, qui est interdit !

Jointure externe en SQL : exemples (2/2)

Quel est le nombre d'activités pratiquées par chaque élève ?



« Simuler » une jointure externe en AR : exemple

Pour chaque activité proposée par l'école, quel est le nombre d'élèves pratiquant cette activité ?

- Schéma : $R(\underline{\text{activite}} : \text{Chaine}, \text{nbEleves} : \text{Nombre})$
- Spécification : $\langle a, ne \rangle \in R \iff$ L'activité a est pratiquée par ne élèves. Les activités qui ne sont pratiquées par aucun élève apparaissent avec $ne = 0$.



« Simuler » une jointure externe en SQL : exemple

Pour chaque activité proposée par l'école, quel est le nombre d'élèves pratiquant cette activité ?

```
SELECT activite, COUNT(*) AS nbActivites
FROM Inscriptions
GROUP BY activite
UNION
SELECT activite, 0 AS nbActivites
FROM Activites
WHERE activite NOT IN (SELECT activite FROM Inscriptions) ;
```

Remplacer une valeur « NULL » par une vraie valeur en SQL, NVL : exemple

Les élèves qui ne sont inscrits à aucune activité vont être inscrits en athlétisme. Quelle sera la nouvelle valeur de la table Inscriptions après cette modification ?

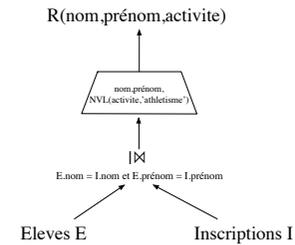
```
SELECT nom, prenom, NVL(activite, 'athletisme') AS activite
FROM Eleves LEFT JOIN Inscriptions
USING (nom, prenom);
```

Remplacer une valeur « NULL » par une vraie valeur en AR, NVL : exemple

Les élèves qui ne sont inscrits à aucune activité vont être inscrits en athlétisme. Quelle sera la nouvelle valeur de la table Inscriptions après cette modification ?

- Schéma : $R(\text{nom} : \text{Chaîne}, \text{prénom} : \text{Chaîne}, \text{activite} : \text{Chaîne})$
- Spécification : $(n, p, a) \in R \iff \text{L'élève } n, p \text{ pratique l'activité } a. \text{ Les élèves qui ne s'étaient inscrits à aucune activité sont inscrits en athlétisme.}$

Algèbre relationnelle :



Opérations d'argument « NULL ».

En général, toute opération avec un argument « NULL » retourne « NULL ».

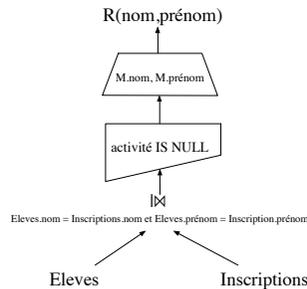
« NULL » n'est ni « vrai », ni « faux », ni le nombre 0, ni la chaîne vide ...

Tester si un attribut a la valeur « NULL » en AR : exemple

Déterminer les élèves qui ne pratiquent aucune activité.

- Schéma : $R(\text{nom} : \text{Chaîne}, \text{prénom} : \text{Chaîne})$
- Spécification : $\langle n, p \rangle \in R \iff \text{L'élève } n, p \text{ ne pratique aucune activité.}$

Algèbre relationnelle :



Tester si un attribut a la valeur « NULL » en SQL, NVL : exemple

Déterminer les élèves qui ne pratiquent aucune activité.

```
SELECT nom, prénom
FROM Eleves LEFT JOIN Inscriptions
USING (nom, prénom)
WHERE activite IS NULL
```

Jointures emboîtées et valeurs absentes en SQL : exemple

Donner, pour chaque élève, la moyenne de ses notes en maths, bio et histoire.

- avec des valeurs absentes :

```
SELECT nom, prénom, (M.note+B.note+H.note)/3 AS moyenne
FROM Maths M JOIN Bio B USING (nom, prénom)
LEFT JOIN Histoire H USING (nom, prénom);
```

-- 7 rows selected.
 -- (l'attribut moyenne a 3 valeurs absentes, ex. Arcila « NULL »)
- sans valeur absente, moyenne sur le nombre de cours suivis :

```
SELECT nom, prénom, NVL((M.note+B.note+H.note)/3, (M.note+B.note)/2) AS moyenne
FROM Maths M JOIN Bio B USING (nom, prénom)
LEFT JOIN Histoire H USING (nom, prénom);
```

-- 7 rows selected. (aucune valeur absente, ex. Arcila 18)
- sans valeur absente, avec NVL (les valeurs absentes valent 0) :

```
SELECT nom, prénom, (M.note+B.note+NVL(H.note,0))/3 AS moyenne
FROM Maths M JOIN Bio B USING (nom, prénom)
LEFT JOIN Histoire H USING (nom, prénom);
```

-- 7 rows selected. (aucune valeur absente, ex. Arcila 12)

Agrégation et valeurs absentes en SQL : exemple

Donner la moyenne de l'école en maths, bio et histoire.

- AVG sans NVL, en histoire on ne compte que les élèves inscrits :

```
SELECT AVG(M.note) AS maths, AVG(B.note) AS bio, AVG(H.note) AS histoire
FROM Maths M JOIN Bio B USING (nom, prénom)
LEFT JOIN Histoire H USING (nom, prénom);
```

-- HISTOIRE 15.6666667: moyenne sur les 3 notes existantes
- AVG avec NVL, en histoire on met 0 aux élèves non inscrits :

```
SELECT AVG(M.note) AS maths, AVG(B.note) AS bio,
AVG(NVL(H.note,0)) AS histoire
FROM Maths M JOIN Bio B USING (nom, prénom)
LEFT JOIN Histoire H USING (nom, prénom);
```

-- HISTOIRE 6.71428571: moyenne sur les 7 élèves

Opérations par ligne et par colonne (1/3)

Une opération par ligne avec au moins un argument « NULL » retourne « NULL ».

Mais une opération d'agrégation (par colonne) (sauf COUNT(*)) ignore les valeurs « NULL »

Soit une table de schéma : T(id:Chaîne,c1:Nombre,c2:Nombre).

$$sommeHV = SUM(c1 + c2) \text{ vs. } sommeVH = SUM(c1) + SUM(c2)$$

Opérations par ligne et par colonne (3/3)

id	c1	c2	
/1	1	2	3
/2	3	4	7
/3	—	—	—
	4	6	10

sommeHV=10, sommeVH=10

id	c1	c2	
/1	1	—	—
/2	3	—	—
/3	5	—	—
	9	—	—

sommeHV=NULL, sommeVH=NULL

Opérations par ligne et par colonne (2/3)

id	c1	c2	
/1	1	2	3
/2	3	4	7
/3	5	6	11
	9	12	21

sommeHV=21, sommeVH=21

id	c1	c2	
/1	1	2	3
/2	—	4	—
/3	5	—	—
	6	6	??

sommeHV=3, sommeVH=12

Définition

On peut nommer la relation qui est retournée par une requête : cela s'appelle une **vue**.

En ce qui concerne les requêtes, les vues se comportent exactement comme les relations.

Contrairement aux relations, en général les vues ne sont pas stockées sur le disque, donc elles se comportent différemment des relations pour les mises à jour.

Syntaxe en SQL

- `CREATE VIEW maVue AS (SELECT... FROM... WHERE...)` pour créer une vue.
- `CREATE VIEW maVue (B1,...) AS (SELECT A1,... FROM... WHERE...)` pour créer une vue en renommant les attributs.
- `DROP VIEW maVue` pour supprimer une vue : c'est sans danger, car cela n'efface aucune donnée (contrairement au fait d'effacer une table, qui est très dangereux!).

Vues en SQL : exemple

Créer une vue `Option1` contenant les nom, prénom et date de naissance de tous les élèves de l'option 1.

```
CREATE VIEW Option1 (nom, prenom, naissance) AS
SELECT nom, prenom, naissance FROM Eleves WHERE opt = 1;
-- View created
-- Pour obtenir la valeur de la vue:
SELECT * FROM Option1;
-- 4 rows selected.
-- Pour détruire la vue:
DROP VIEW Option1;
-- View dropped
```