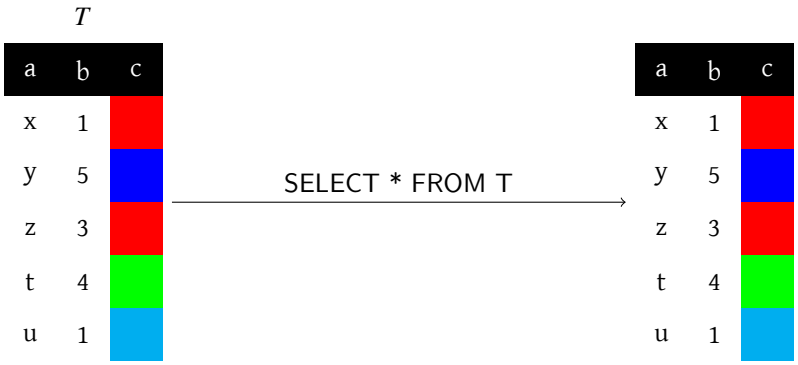
Informatique en MP(2)I

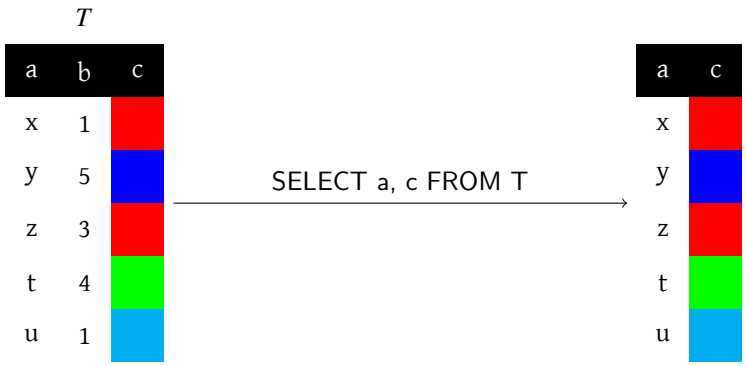
Marc de Falco

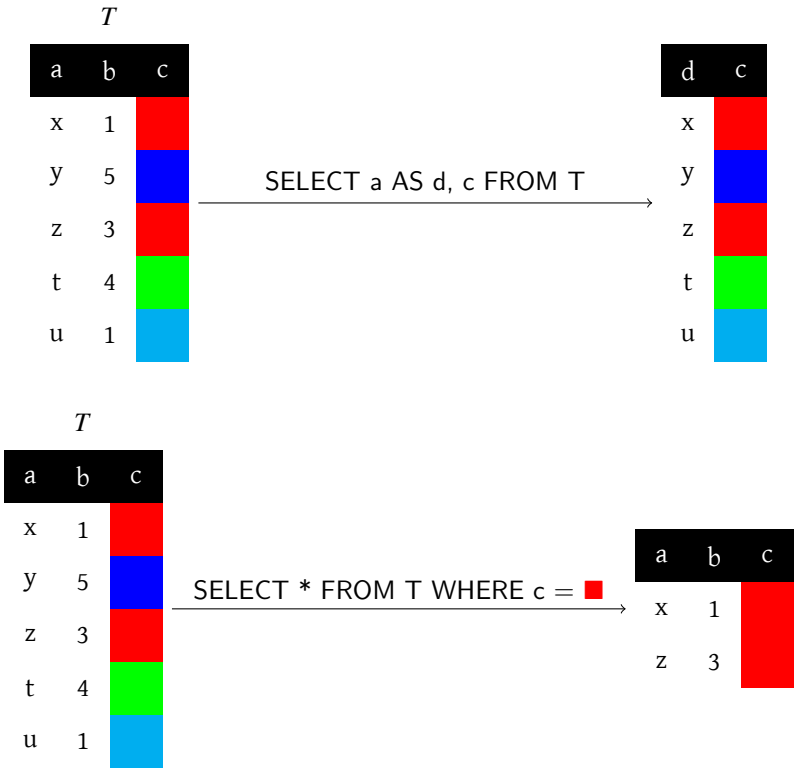
Bases de données

I Vocabulaire

II SELECT

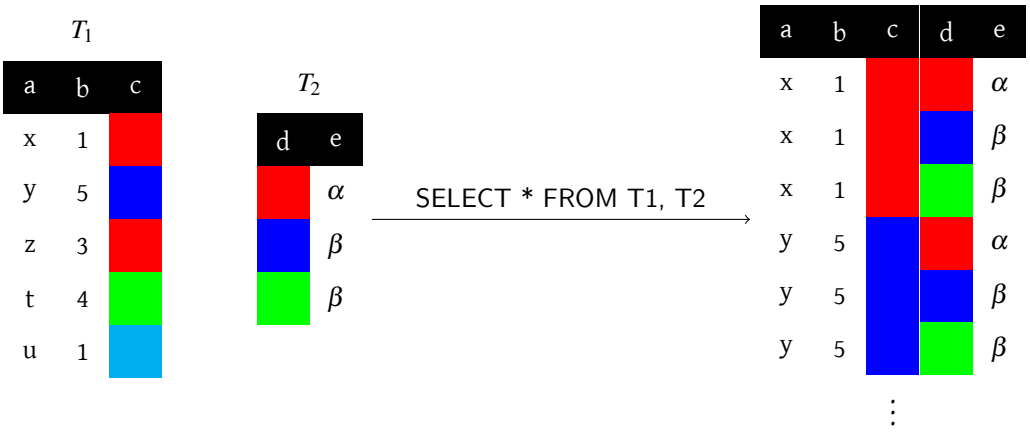


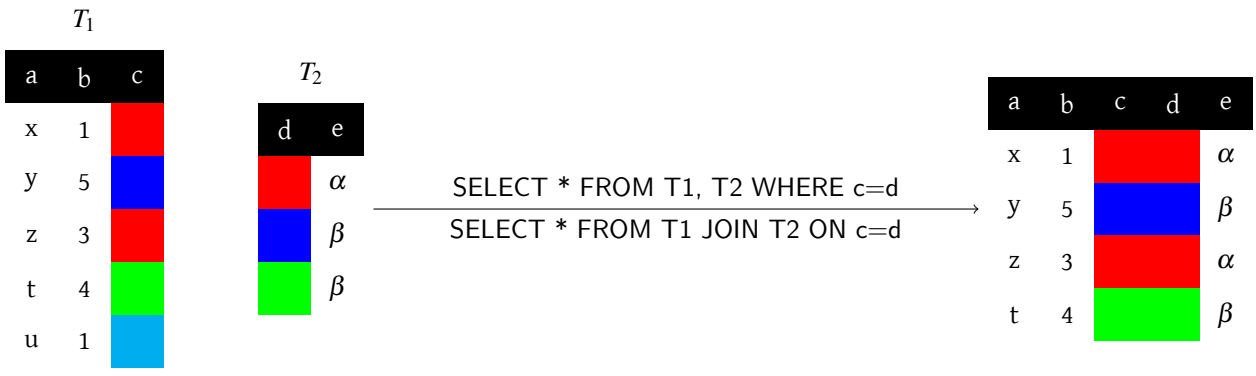




III Opérateurs ensemblistes

IV Produit cartésien et jointures





T_1

a	b	c
x	1	
y	5	
z	3	
t	4	
u	1	

T_2

d	e
	α
	β
	β

SELECT * FROM T1 LEFT JOIN T2 ON c=d

a	b	c	d	e
x	1			α
y	5			β
z	3			α
t	4			β
u	1		NULL	NULL

V Fonctions et requêtes imbriquées

T

a	b	c
x	1	
y	5	
z	3	
t	4	
u	1	

SELECT COUNT(*) FROM T

COUNT(*)
5

T

a	b	c
x	1	
y	5	
z	3	
t	4	
u	1	

SELECT COUNT(*) AS n FROM T

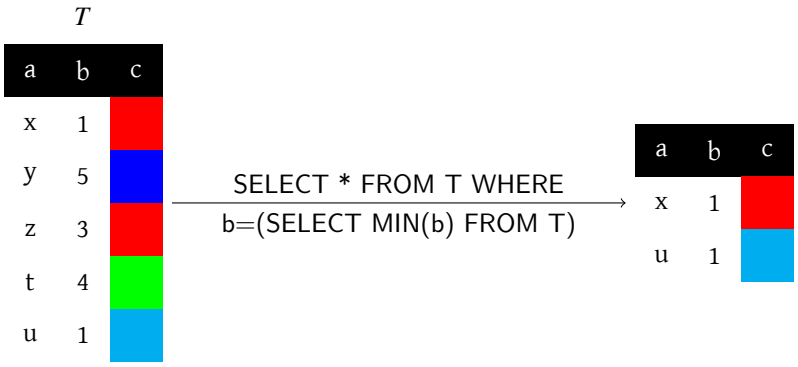
n
5

T

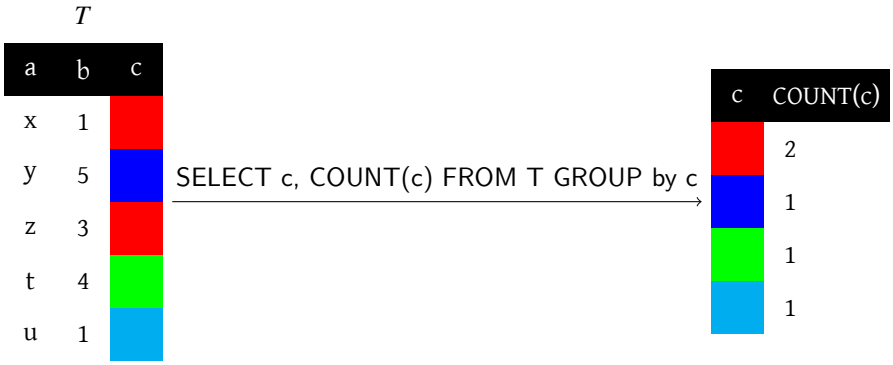
a	b	c
x	1	
y	5	
z	3	
t	4	
u	1	

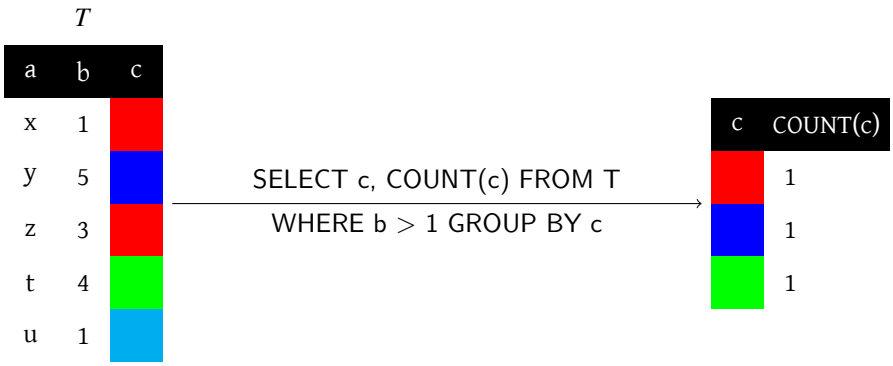
SELECT MAX(a), MIN(b) FROM T

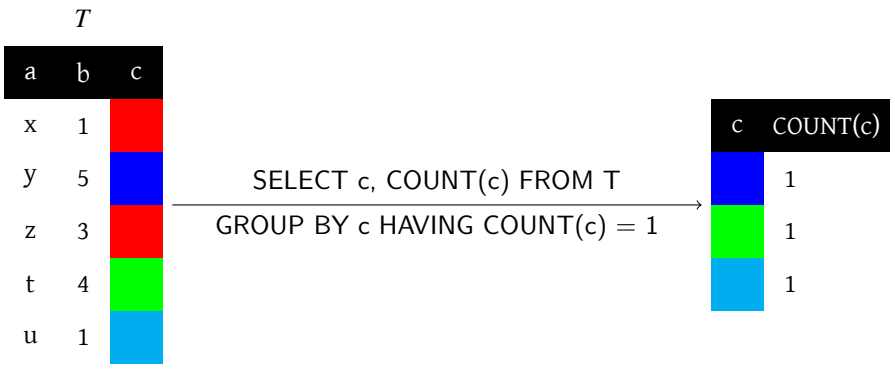
MAX(a)	MIN(b)
u	1



VI Agrégation







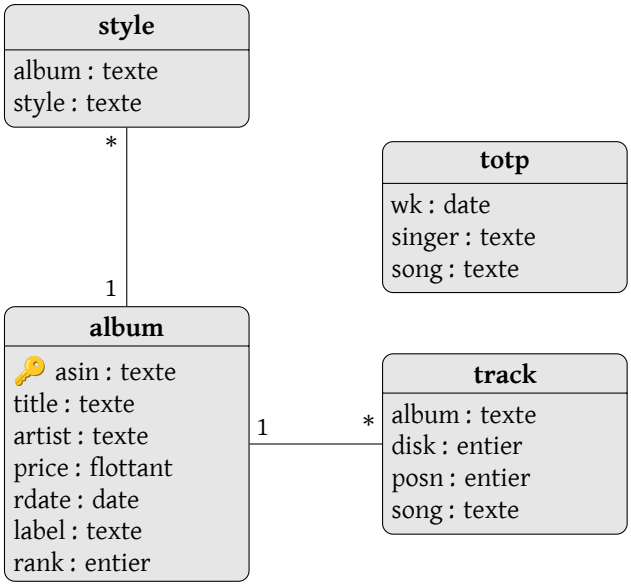
VII TP

Allez sur le site `http://sql.classeprepa.net` et identifiez vous avec le couple utilisateur/mot de passe suivant : mp2i/mp2i.

VII.1 Musique

- Sélectionnez `tp26_musique` dans la barre de navigation à gauche de l'écran.
- Cliquez sur l'onglet `SQL`. On vous demande dans la suite de taper des requêtes et d'appuyer sur `Exécuter` pour les voir s'exécuter. Après la première requête, on peut faire `Éditer en ligne` pour aller plus vite.
- On vous recommande de copier les réponses dans un fichier `tp26.sql` pour en garder une trace.

La base de donnée `tp26_musique` comporte quatre tables dont le schéma est donné dans la figure suivante :



VII.2 Premières requêtes

L'attribut `asin` de `album` est la clé primaire qui sert de clé étrangère `album` dans `track` et `style`.

Question VII.2.1 Cherchez tous les albums de `'U2'`.

Ici, les dates sont au format texte `'yyyy-mm-dd'` qui permet d'effectuer des comparaisons directement avec `<, <=, >, >=`.

Par exemple, '1981-08-05' est le 5 août 1981.

Question VII.2.2 Cherchez tous les albums parus pendant l'année 1991.

Les données sont issues d'un célebre vendeur en ligne, à un instant précis dans le passé, c'est en ce sens qu'il faut interpréter l'attribut `price` (c'est le prix en dollars) et l'attribut `rank` (c'est le rang dans les ventes de disques).

L'attribut `price` n'est pas toujours renseigné. Dans ce cas là, il prend la valeur spéciale `NULL`. Pour tester si une valeur est `NULL` on utilise l'opérateur spécial `IS NULL`.

Question VII.2.3 Cherchez tous les albums dont le prix est renseigné.

Question VII.2.4 Trouvez le titre du 5 ème album le plus cher **et uniquement de celui-ci**.

Question VII.2.5 Affichez une table contenant pour chaque chanson, son nom et le nom de l'album qui la contient.

L'attribut `disk` indique le numéro du disque sur l'album, la plupart contiennent un disque, et le numéro `posn` des pistes sur l'album.

Question VII.2.6 Affichez une table contenant les titres des albums ayant au moins deux disques.
On fera en sorte qu'il n'y ait pas de titres repétés.

La table `totp` contient des chansons ayant apparu dans le classement mensuel des meilleurs ventes. Apparement, les tables `track` et `totp` ne sont pas associées.

Question VII.2.7 A l'aide de `LEFT JOIN` déterminez s'il y a des éléments de `totp` dont l'attribut `song` n'est pas présent dans `track`. Même question en inversant les tables.
Est-il pour autant possible de considérer que `song` est une clé primaire d'une de ces deux tables et que l'autre est une clé étrangère ?

VII.2.i Fonctions

On peut appliquer des fonctions sur un ou plusieurs attributs, ou expressions, pour obtenir des valeurs. Se faisant, on va déduire une table à une ligne contenant les résultats.
Par exemple :

```
SELECT COUNT(*) FROM album
```

va renvoyer le nombre de lignes de la table `album`.
Les fonctions au programme sont :

- `COUNT` qui permet de trouver la taille des résultats. Attention, `COUNT(attr)` ne compte pas les valeurs distinctes. On utilisera `COUNT(*)`.
- `MAX` et `MIN` qui renvoie le maximum ou le minimum d'un attribut ou d'une expression
- `SUM` qui fait la somme des valeurs d'un attribut ou d'une expression sur les lignes
- `AVG` qui calcule la moyenne

Question VII.2.8 En une seule requête, calculer le prix moyen d'un album et le plus petit rang.

VII.2.ii Requêtes imbriquées

Attention, la requête

```
SELECT title, MIN(rank) FROM album
```

n'a aucune raison de renvoyer le titre de l'album de plus petit rang. En fait, avec un gestionnaire peu permissif, la requête précédente est invalide.
Une manière de réaliser cela est d'utiliser une requête imbriquée comme :

```
SELECT title, rank FROM album WHERE rank = (SELECT MIN(rank) FROM album)
```

Ici, comme la sous-requête `SELECT MIN(rank) FROM album` renvoie un nombre, on peut tester cela dans la première requête.
On peut noter qu'il est possible de réaliser une telle requête avec une jointure externe à gauche comme

```
SELECT A1.title, A1.rank
FROM album AS A1 LEFT JOIN album AS A2
on A2.rank < A1.rank
WHERE A2.rank IS NULL
```

mais cette requête n'est pas très naturelle.

Question VII.2.9 Expliquez cette requête et pourquoi elle répond à la question.

Question VII.2.10 Déterminer en une requête les albums ayant le plus de pistes sur un de leur disque.

VII.2.iii Agrégation

Pour grouper les chansons dans la table track selon l'album qui les contient on peut utiliser la requête :

```
SELECT album FROM track GROUP BY album
```

On obtient alors une ligne pour chaque valeur de `album` présente dans la table. On parle d'agrégat ou de paquet. Cela équivaut ici à

```
SELECT DISTINCT album FROM track
```

Une règle importante en SQL est qu'il est interdit de projeter des attributs qui ne sont pas compris dans le `GROUP BY`. En effet, il y a ambiguité sur la valeur qu'on pourrait choisir pour ces attributs.

Là où les `GROUP BY` sont très utiles c'est qu'on peut calculer des fonctions sur chaque agrégat qui elles-sont non ambigues.

Ici la piste de numéro maximum sur un album indique le nombre de pistes de l'album :

```
SELECT album, MAX(posn) FROM track GROUP BY album
```

Question VII.2.11 Déterminer le numéro moyen d'une piste d'un album, tout disque compris.

Si on souhaite obtenir avec un `GROUP BY` les albums ayant au moins 10 pistes, on pourrait être tenté d'écrire

```
SELECT album FROM track GROUP BY album WHERE MAX(posn) >= 10
```

le probléme est que le `WHERE` s'effectue avant le `GROUP BY` même s'il apparait après dans la requête. `SQL` fournit une syntaxe permettant de filtrer sur des conditions dans les agrégats : `HAVING`. Ainsi, on écrira :

```
SELECT album FROM track GROUP BY album HAVING MAX(posn) >= 10
```

On aimerait aussi avoir le nom de l'album, pour cela, il y a plusieurs solutions, une des solutions les plus simples et d'utiliser `MAX(title)` sur un agrégat.

```
SELECT MAX(title) FROM track JOIN album
ON track.album = asin
GROUP BY track.album
HAVING MAX(posn) >= 10
```

On peut le faire avec une requête imbriquée dans un `JOIN` interne qui ne va donc sélectionner que les albums présents dans les deux tables. Ici, le renommage est obligatoire pour que la requête soit valide.

```
SELECT title
FROM album
JOIN (SELECT album FROM track
    GROUP BY album HAVING MAX(posn) >= 10) as T
ON asin = T.album
```

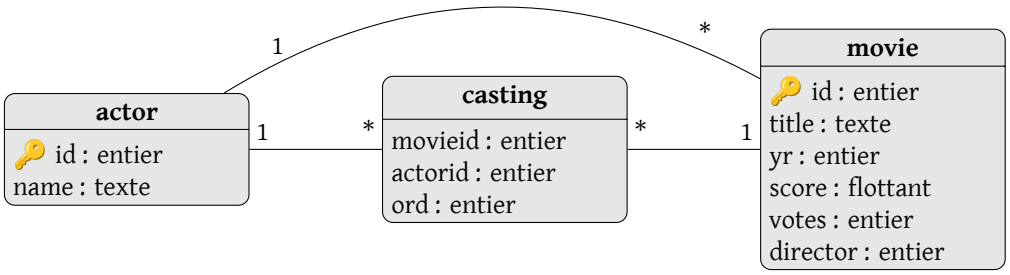
On pourrait aussi utilser une requête imbriquée et l'opérateur `IN` hors programme pour tester la présence d'une valeur dans une table de résultat.

```
SELECT title FROM album
WHERE asin IN (SELECT album FROM track
    GROUP BY album HAVING MAX(posn) >= 10)
```

Question VII.2.12 Trouver les noms des albums ayant au moins deux disques.

VII.3 Films

On considère maintenant la base de données `tp26_film`. Ses tables sont décrites dans le diagramme :



Quelques précisions :

- Les clés `id` permettent d'identifier uniquement un film (`movie`) ou un acteur (`actor`).
- L'attribut `director` fait référence à un id d'`actor` qui est également réalisateur de ce film (un pur réalisateur peut ne pas être un vrai acteur mais être présent dans la table `actor` quand même).
- L'entier `ord` indique l'ordre dans lequel sont énumérés les acteurs dans le casting d'un film : du premier nom de valeur 1 (le plus important) au dernier.
- `score` est le score des internautes sur le film.
- `votes` est le nombre de votes
- `yr` est un entier à quatre chiffres comme 1977.

On va rajouter un opérateur qui est pratique pour le TP mais pas au programme : `LIKE` qui permet de comparer des textes avec un joker le caractère `%` qui signifie n'importe quoi.

Ainsi,

```
SELECT title FROM movie WHERE title LIKE '%, The'
```

va sélectionner les films dont le nom finit par `, The`.

Question VII.3.1 Déterminer les noms des acteurs ayant joué dans des films dont le nom commence par Star.

Question VII.3.2 Déterminer le film qui a le plus d'acteurs au casting.

Question VII.3.3 Déterminer les noms des acteurs ayant tournés dans au moins 5 films tout en étant toujours premier au casting.

Question VII.3.4 Déterminer les noms des acteurs qui n'ont tourné qu'avec un seul réalisateur.

Question VII.3.5 Déterminer les noms des acteurs qui n'ont tourné que dans des films qu'ils ont réalisés.