



HAL
open science

QueryCraft : exécuter SQL en mode "pas à pas"

Emmanuel Desmontils, Laura Monceaux

► To cite this version:

Emmanuel Desmontils, Laura Monceaux. QueryCraft : exécuter SQL en mode "pas à pas". Didapro 11, Mar 2026, Grenoble, France. <hal-05630675>

HAL Id: hal-05630675

<https://hal.science/hal-05630675v1>

Submitted on 22 May 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

QueryCraft : exécuter SQL en mode « pas à pas »

Emmanuel Desmontils
Nantes Université,
LS2N, UMR 6004 / CREN, UR 2661
emmanuel.desmontils@univ-nantes.fr

Laura Monceaux
Nantes Université,
LS2N, UMR 6004
laura.monceaux@univ-nantes.fr

« Pas à pas »

Étapes de construction de la table résultat selon un ordre logique :

- (1) tables → (2) jointure(s) → (3) sélection des tuples*
- (4) regroupement** → (5) sélection des groupes* **
- (6) projection → (7) ordre* → (8) k-premiers*

*: en deux étapes

** : présentation particulière

Exemple

```
SELECT etudiants.noetu, nom, prenom
FROM etudiants Join notes
On etudiants.noetu = notes.noetu
GROUP BY etudiants.noetu
HAVING count(*) > 1 ;
```

(1)a: SELECT * FROM etudiants ;

(1)b: SELECT * FROM notes ;

(2):

```
SELECT *
FROM etudiants Join notes
On etudiants.noetu = notes.noetu ;
```

(3) rien sur cette requête

(4):

```
SELECT etudiants.noetu, etudiants.nom, etudiants.prenom,
notes.noetu, notes.codemat, notes.noteex, notes.notecc
FROM etudiants Join notes
On etudiants.noetu = notes.noetu
ORDER BY etudiants.noetu ;
```

(5):

```
SELECT etudiants.noetu, etudiants.nom, etudiants.prenom,
notes.noetu, notes.codemat, notes.noteex, notes.notecc
FROM etudiants Join notes
On etudiants.noetu = notes.noetu
WHERE (etudiants.noetu) In (
SELECT etudiants.noetu
FROM etudiants Join notes
On etudiants.noetu = notes.noetu
GROUP BY etudiants.noetu
HAVING COUNT(*) > 1
)
GROUP BY etudiants.noetu
ORDER BY etudiants.noetu ;
```

(6):

```
SELECT etudiants.noetu, nom, prenom
FROM etudiants Join notes
On etudiants.noetu = notes.noetu
GROUP BY etudiants.noetu
HAVING COUNT(*) > 1 ;
```

(7) rien sur cette requête

(8) rien sur cette requête

Étape N°4/6

```
SELECT etudiants.noetu, nom, prenom
FROM etudiants Join notes On etudiants.noetu = notes.noetu
GROUP BY etudiants.noetu
HAVING COUNT(*) > 1 ;
```

etudiants.noetu	etudiants.nom	etudiants.prenom	notes.noetu	notes.codemat	notes.noteex	notes.notecc
28936E	Dupont	Franck	28936E	MIAS215	[13.5, 12]	[13.5, No]
46283B	Dupont	Isabelle	46283B	MIAS215	[9.5, 8]	[2, 11]
86719E	Martin	Adrien	86719E	TUP215	[8.5, 12]	[10, 5.5]
99321C	Denou	Michelle	99321C	LIL6	[15, 18]	[14.5, 16]
99322C	Dupont	Isabelle	99322C	MIAS215	[12]	[15.5]
99628C	Robert	Adrien	99628C	MIAS215	[12, 3]	[15.5, 6]

Nombre de tuples : 6

Étape N°5/6

```
SELECT etudiants.noetu, nom, prenom
FROM etudiants Join notes On etudiants.noetu = notes.noetu
GROUP BY etudiants.noetu
HAVING COUNT(*) > 1 ;
```

etudiants.noetu	etudiants.nom	etudiants.prenom	notes.noetu	notes.codemat	notes.noteex	notes.notecc
28936E	Dupont	Franck	28936E	MIAS215	[13.5, 12]	[13.5, No]
46283B	Dupont	Isabelle	46283B	MIAS215	[9.5, 8]	[2, 11]
86719E	Martin	Adrien	86719E	TUP215	[8.5, 12]	[10, 5.5]
99321C	Denou	Michelle	99321C	LIL6	[15, 18]	[14.5, 16]
99322C	Dupont	Isabelle	99322C	MIAS215	[12]	[15.5]
99628C	Robert	Adrien	99628C	MIAS215	[12, 3]	[15.5, 6]

Nombre de tuples : 10

Étape N°6/6 (Résultat)

```
SELECT etudiants.noetu, nom, prenom
FROM etudiants Join notes On etudiants.noetu = notes.noetu
GROUP BY etudiants.noetu
HAVING COUNT(*) > 1 ;
```

etudiants.noetu	etudiants.nom	etudiants.prenom
28936E	Dupont	Franck
46283B	Dupont	Isabelle
86719E	Martin	Adrien
99321C	Denou	Michelle
99628C	Robert	Adrien

Nombre de tuples : 5

© E. Desmontils, Nantes Université, 2024-2026 | QueryCraft v1.5.5 — Crédits et informations

IA (optionnel)

- Reformulation - développement
 - ▶ de la requête
 - ▶ des erreurs du SQGB
 - ▶ d'erreurs de contrôle (faiblesse des SGBD, sécurité...)
 - ▶ en complément de la correction d'exercice
- Approche contrôlée et éthique
 - ▶ Mode optionnel
 - ▶ (S/L)LM locaux ou distants
 - ▶ Cache
 - ▶ Prompt sécurisé
 - ▶ Transparence : reconnaissable, origine et message de prudence

Exercice et Question (optionnel)

Exercice : Didapro11 Question : q2 - Les étudiants dont le nom commence par D.

Intention : Les étudiants dont le nom commence par D.

Requête SQL :
SELECT * FROM etudiants WHERE nom in ("Dupont","Durant") ;

Mode pas à pas Aide de l'IA

Requête étudiée
SELECT * FROM etudiants WHERE nom in ("Dupont","Durant") ;

Explication de la requête par IA générative
Objectif fonctionnel : récupérer toutes les lignes de la table etudiants dont le nom est soit "Dupont" soit "Durant".

Source : POE (https://poe.com/) avec gpt-5-nano. Le 2026-02-23 à 16:42:46.867241. Attention, POE/gpt-5-nano ne garantit pas la validité de l'aide. Veuillez vérifier la réponse et vous rapprocher de vos enseignants si nécessaire.

Exercices (optionnel)

- Choix possible d'un exercice et d'une question (requête) avec choix du type : < I→R > ou < IT→R > [2].
- Correction partielle (évaluation de la table et aide)

Correction

✗ La requête proposée n'est pas correcte. Elle ne renvoie pas les mêmes résultats que la requête attendue.

Regarder la première lettre du nom.

Correction de la requête par IA générative

1) Non, sa requête n'est pas équivalente à la requête correcte. Sa condition nom in ("Dupont","Durant") ne sélectionne que les étudiants dont le nom est exactement l'un des éléments de la liste, alors que la requête correcte sélectionne tous les étudiants dont le nom commence par la lettre D (c'est-à-dire dont la première lettre est D), ce qui peut inclure d'autres noms comme Delarue, Deschamps, etc.

2) Feedback:
- Pourquoi ça ne fonctionne pas comme attendu: ta condition filtre sur des valeurs précises de nom, pas sur la première lettre. Regarde l'indice donné par l'enseignant: "Regarder la première lettre du nom."
- Pistes pour corriger:
- Comment vérifier la première lettre d'un champ texte dans SQL sans connaître tous les noms possibles ?
- Pense à une opération qui permet de tester si une chaîne commence par une lettre donnée.
- Si tu avais un exemple de noms dans la table, pourrais-tu déduire une condition qui englobe tous ceux qui commencent par D sans nommer chacun séparément?
- Question guiding: quel mécanisme pourrait te permettre d'utiliser uniquement la première lettre du champ nom pour filtrer les résultats, plutôt qu'une liste précise de valeurs?

Source : POE (https://poe.com/) avec gpt-5-nano. Le 2026-02-23 à 16:42:56.150870. Attention, POE/gpt-5-nano ne garantit pas la validité de l'aide. Veuillez vérifier la réponse et vous rapprocher de vos enseignants si nécessaire.

Références

- [1] Desmontils, E. (2025). QueryCraft. <https://gitlab.univ-nantes.fr/ls2n-didactique/querycraft> <https://pypi.org/project/querycraft/>
- [2] Desmontils, E. & Monceaux, L. (2023). Enseigner SQL en NSI. In Atelier « Apprendre la Pensée Informatique de la Maternelle à l'Université », dans le cadre de la conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH), pages 17-24. (hal-04108258v2)

