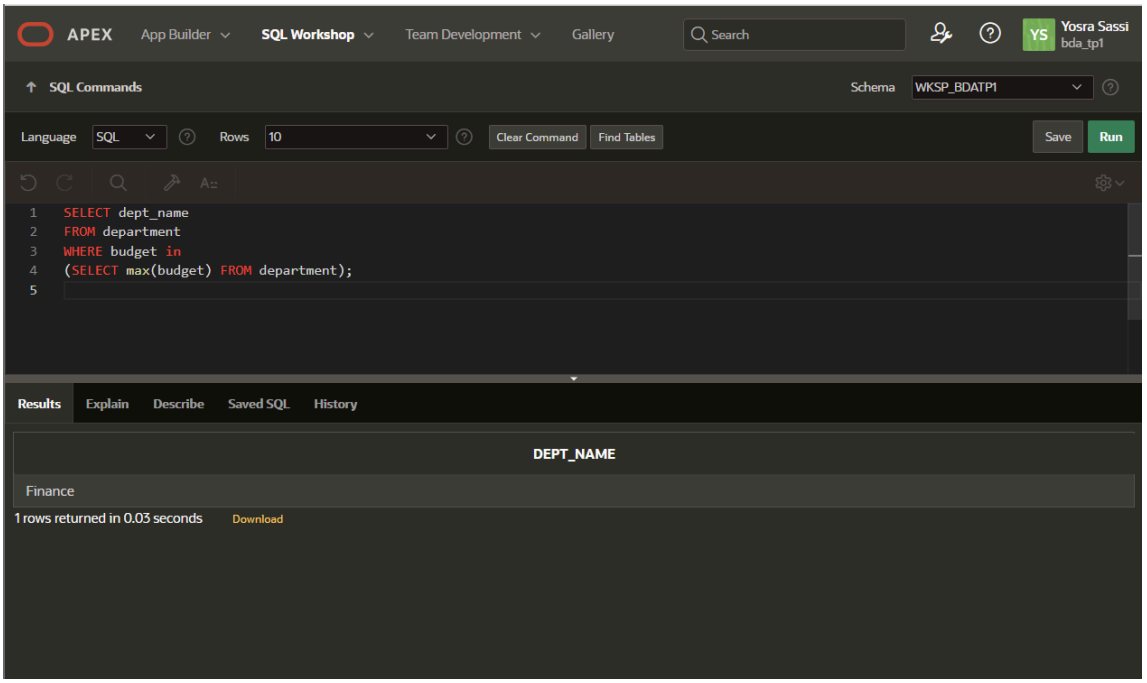
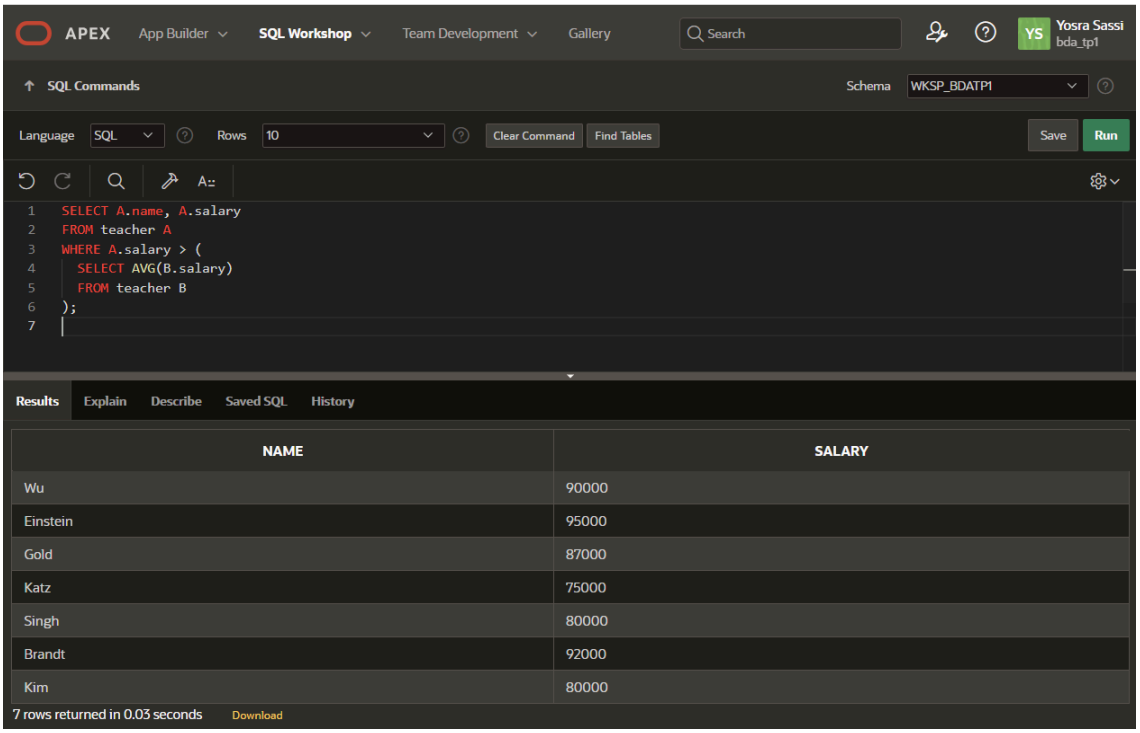
Exercice 1 – Requêtes SQL

Dans cet exercice, nous appliquons des requêtes SQL afin d'extraire des informations pertinentes à partir d'une base de données universitaire.

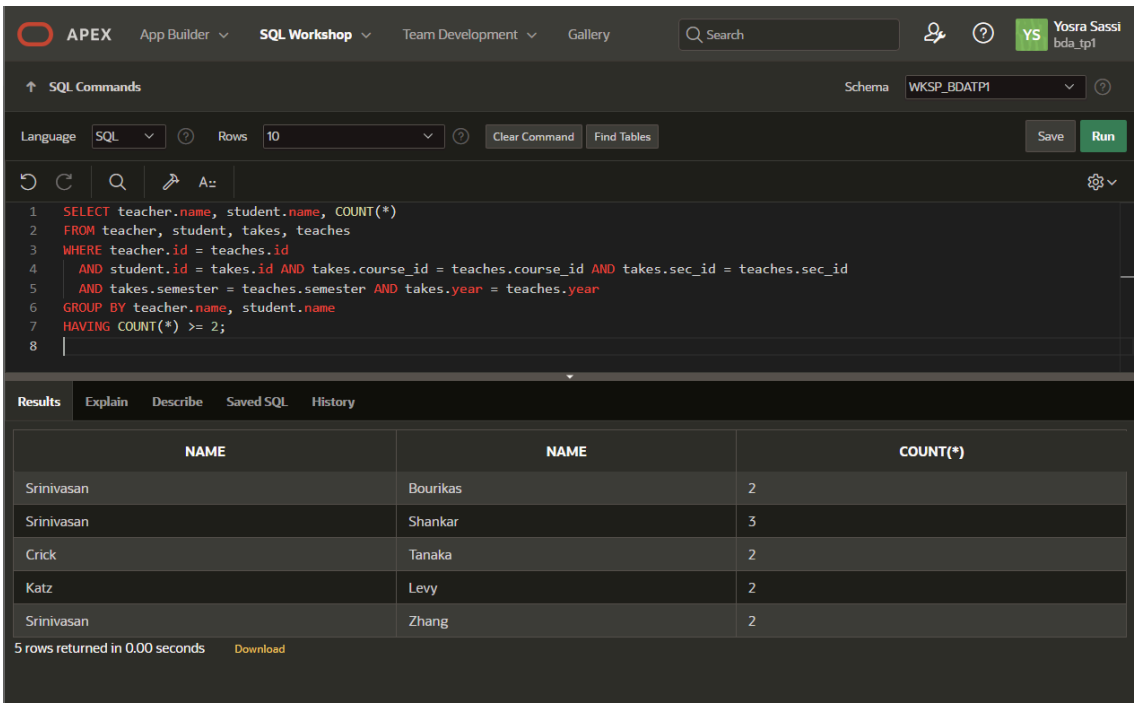
1. Afficher le nom du département qui a le budget le plus élevé.



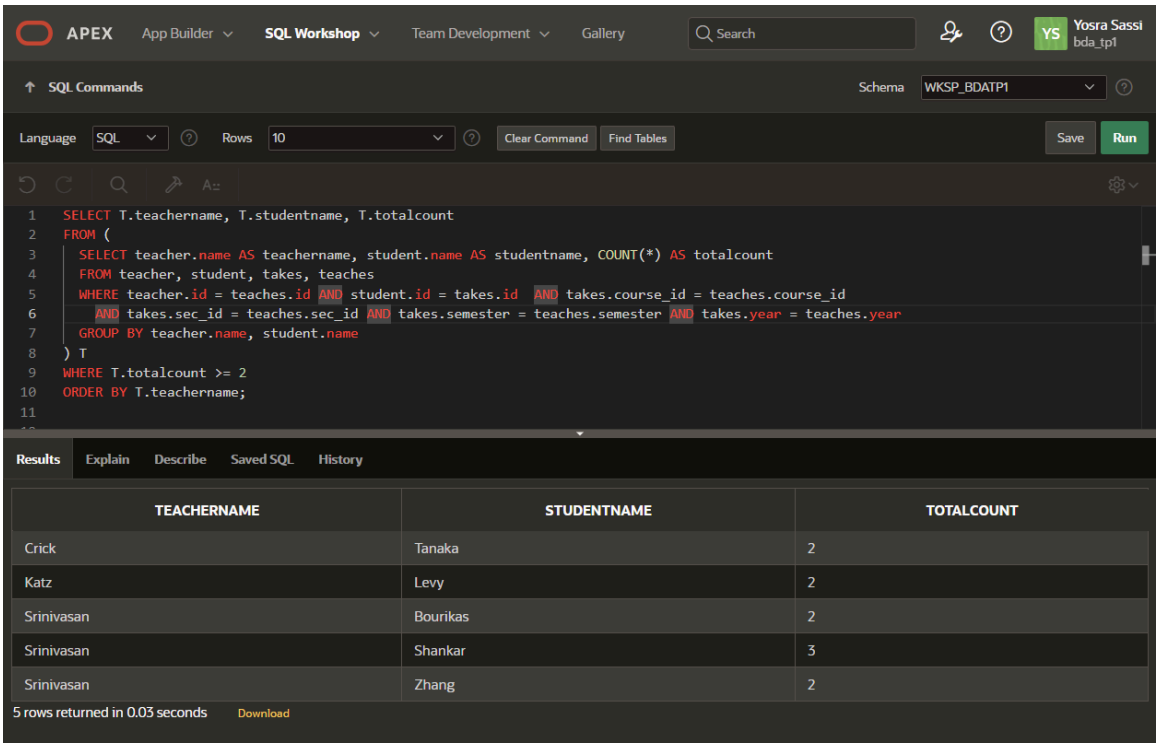
2. Afficher les salaires et les noms des enseignants qui gagnent plus que le salaire moyen.



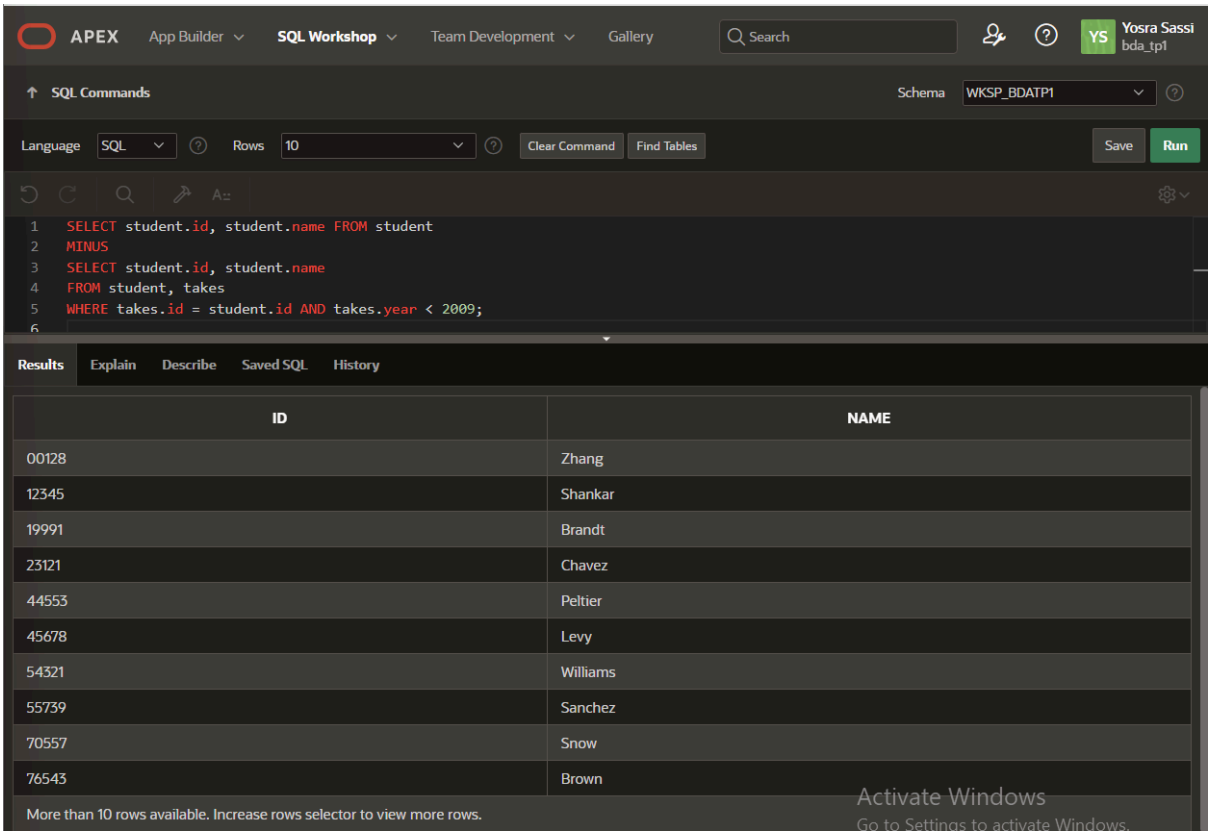
3. Pour chaque enseignant, afficher tous les étudiants qui ont suivi plus de deux cours dispensés par cet enseignent ainsi que le nombre total de cours suivis par chaque étudiant, en utilisant la clause HAVING.



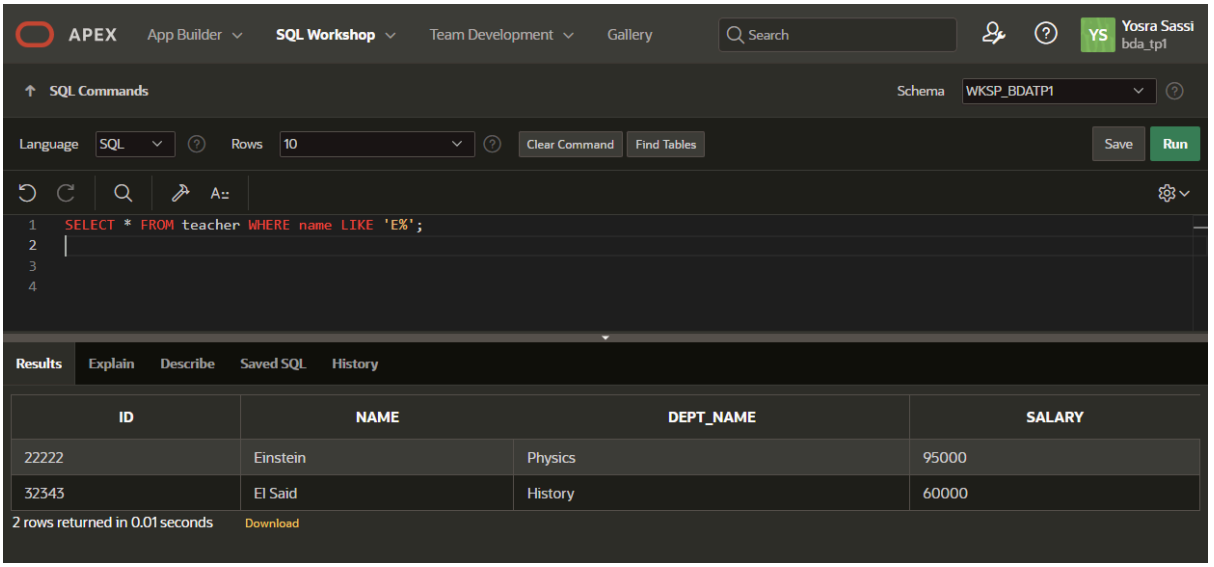
4. Pour chaque enseignant, afficher tous les étudiants qui ont suivi plus de deux cours dispensés par cet enseignent ainsi que le nombre total de cours suivis par chaque étudiant, sans utiliser la clause HAVING.



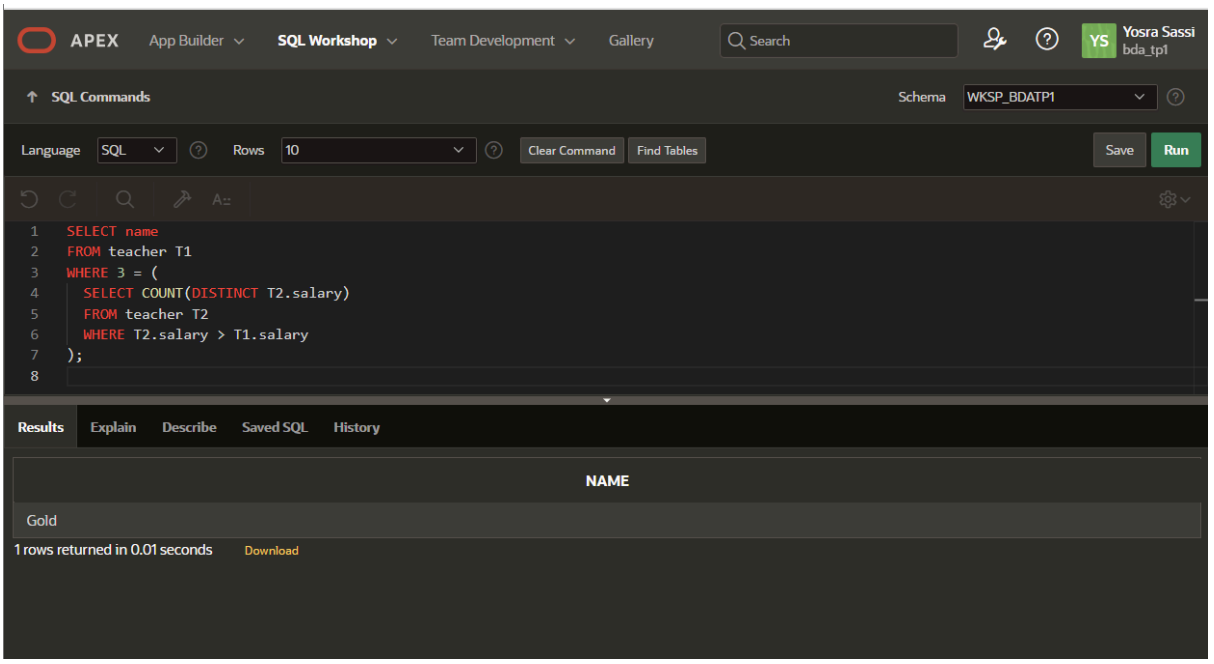
5. Afficher les identifiants et les noms des étudiants qui n'ont pas suivi de cours avant 2010.



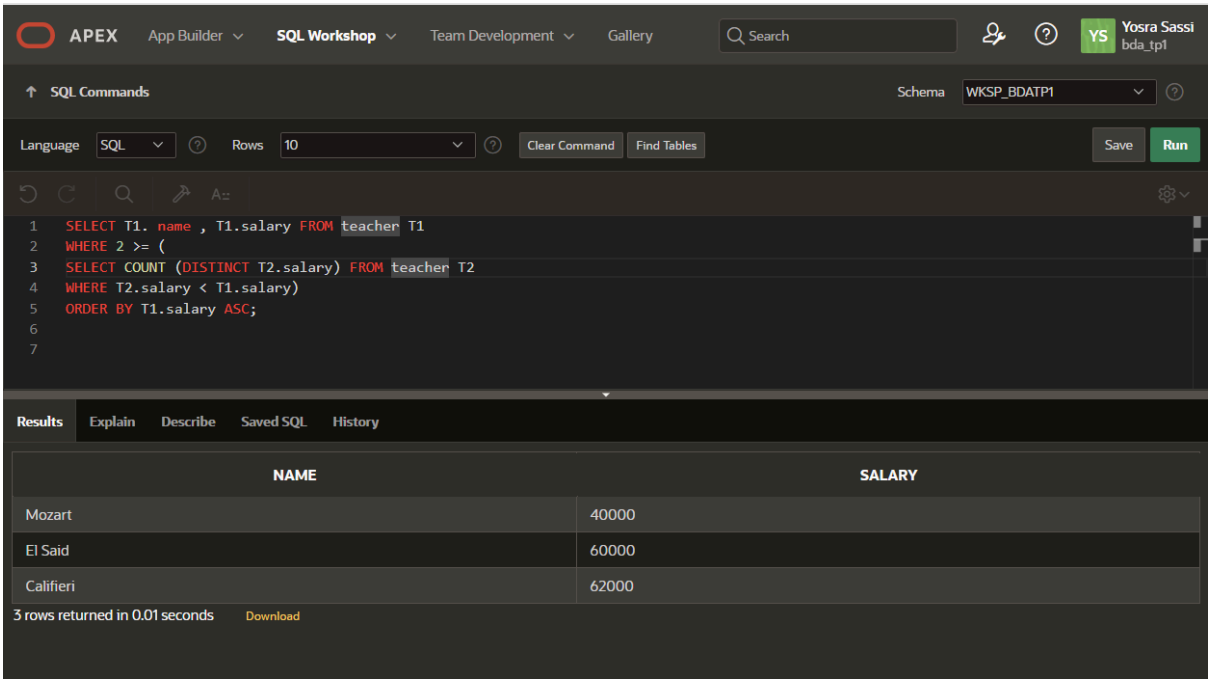
6. Afficher tous les enseignants dont les noms commencent par E.



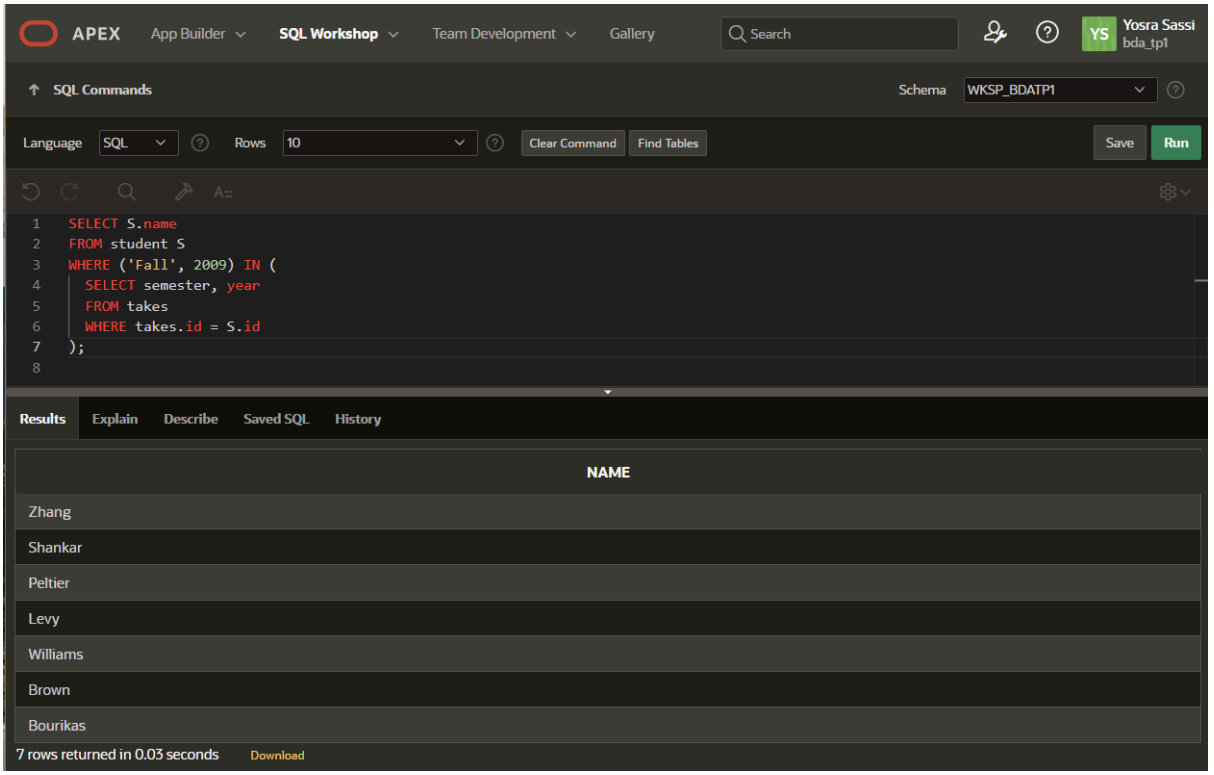
7. Afficher les salaires et les noms des enseignants qui perçoivent le quatrième salaire le plus élevé.



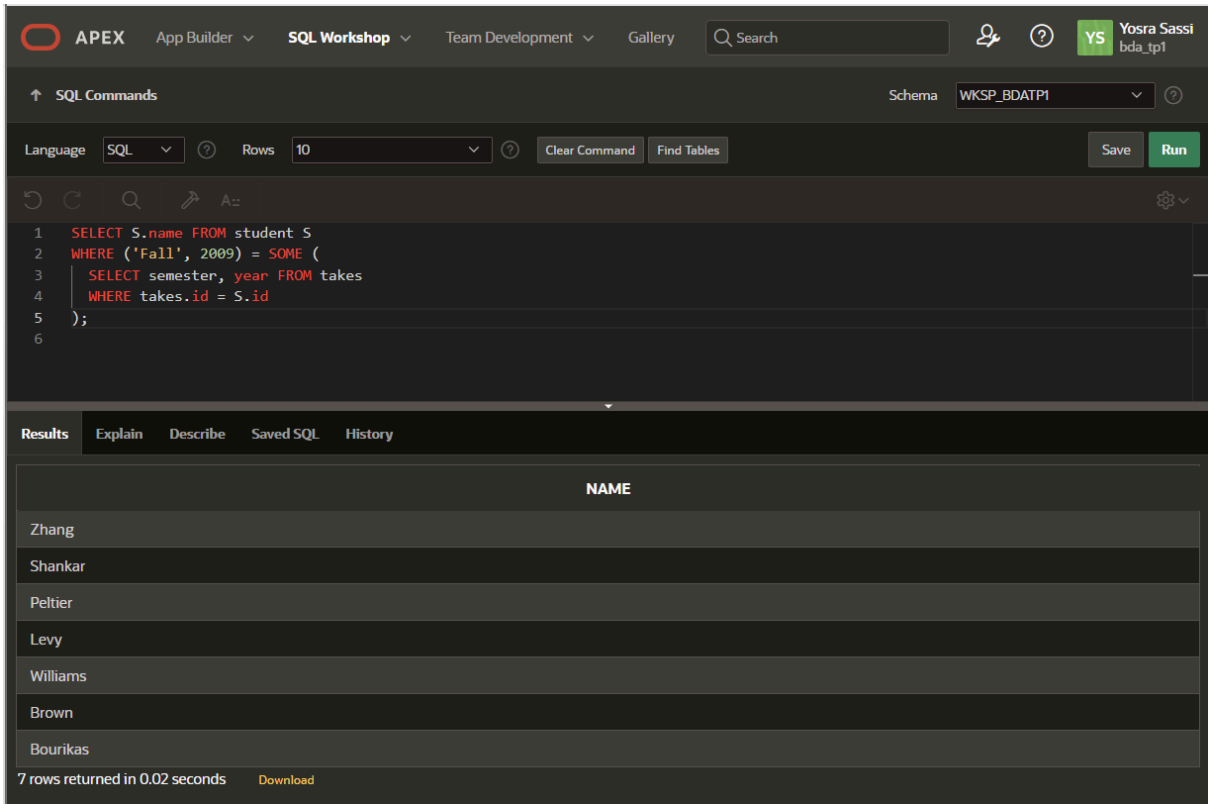
8. Afficher les noms et les salaires des trois enseignants qui perçoivent les salaires les moins élevés. Les afficher par ordre décroissant.



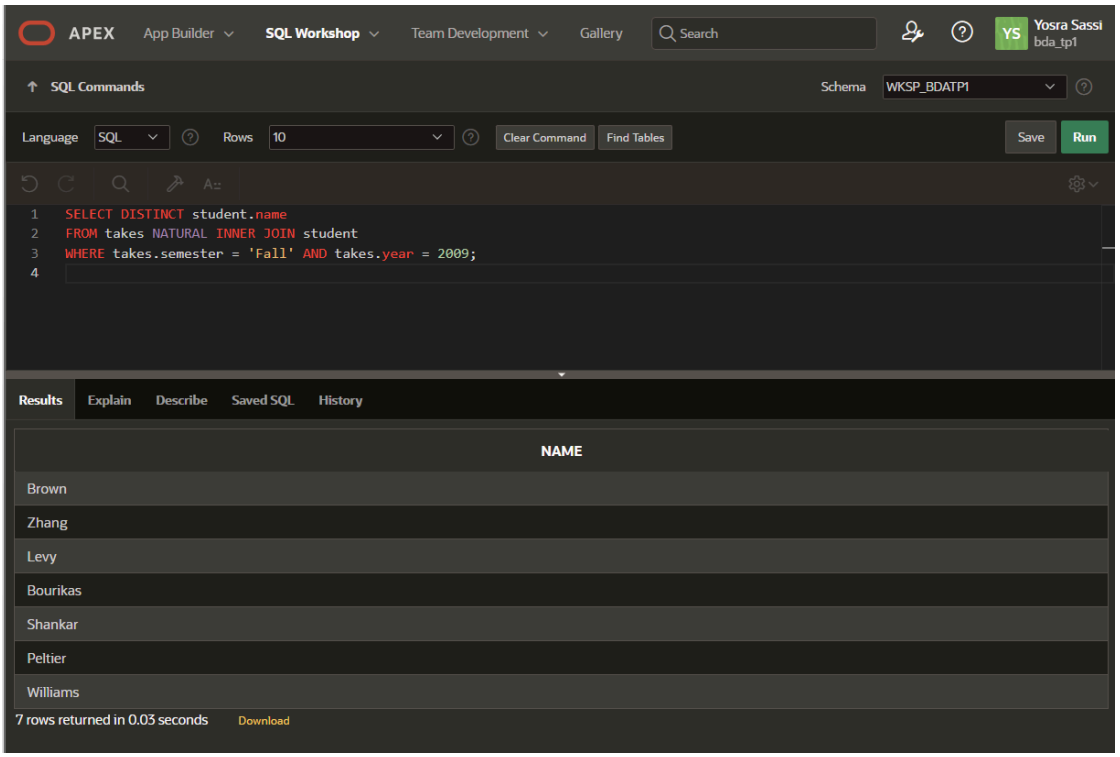
9. Afficher les noms des étudiants qui ont suivi un cours en automne 2009, en utilisant la clause IN.



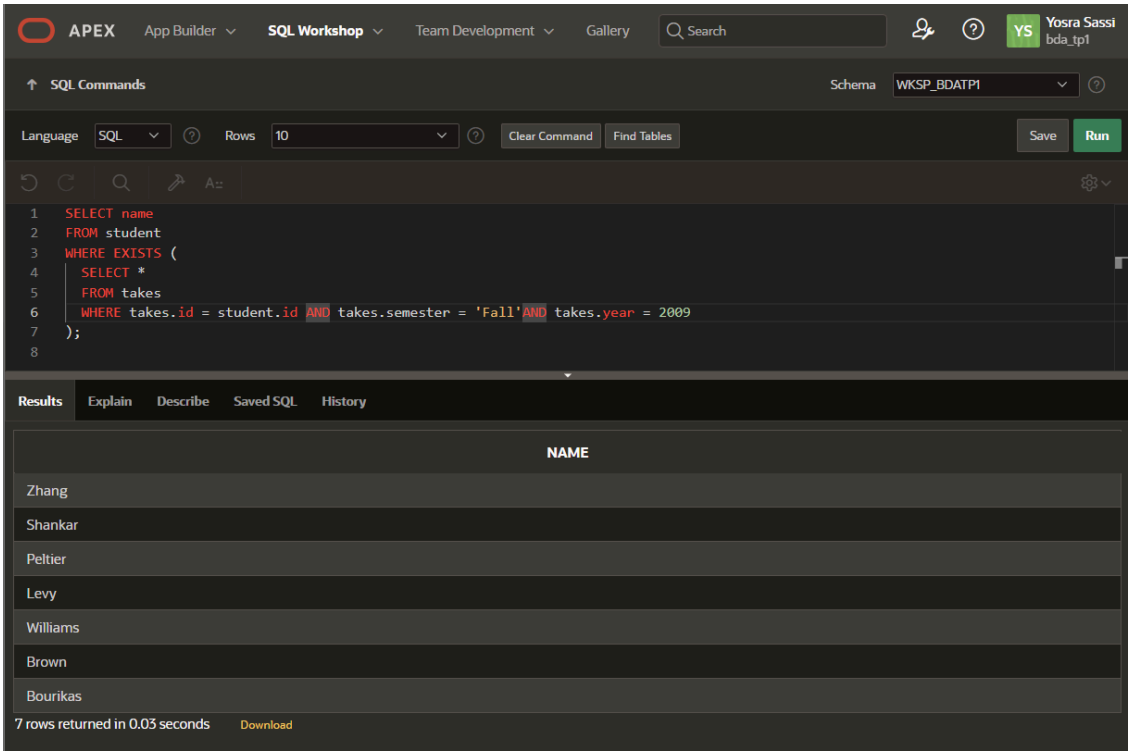
10. Afficher les noms des étudiants qui ont suivi un cours en automne 2009, en utilisant la clause SOME.



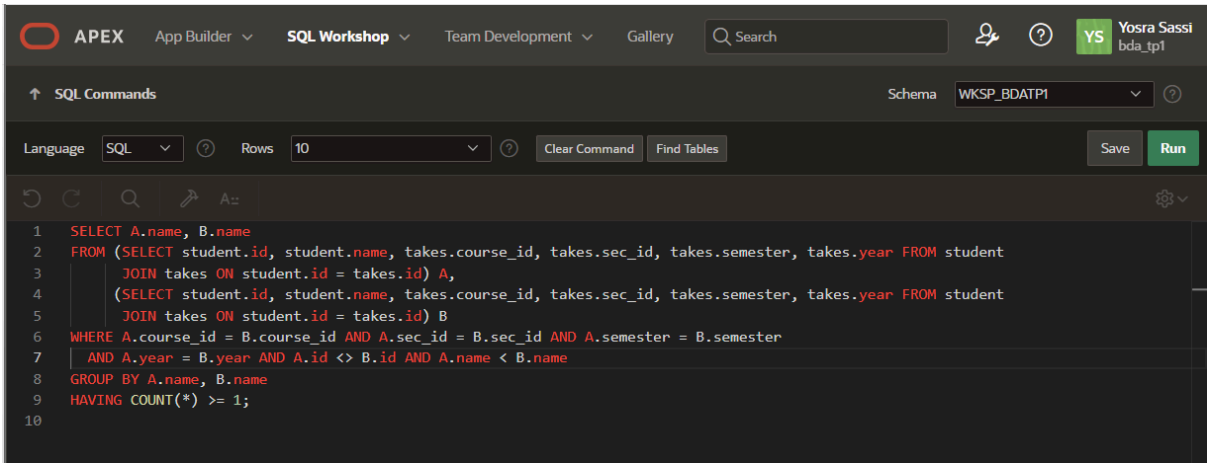
11. Afficher les noms des étudiants qui ont suivi un cours en automne 2009, en utilisant la jointure naturelle (NATURAL INNER JOIN).

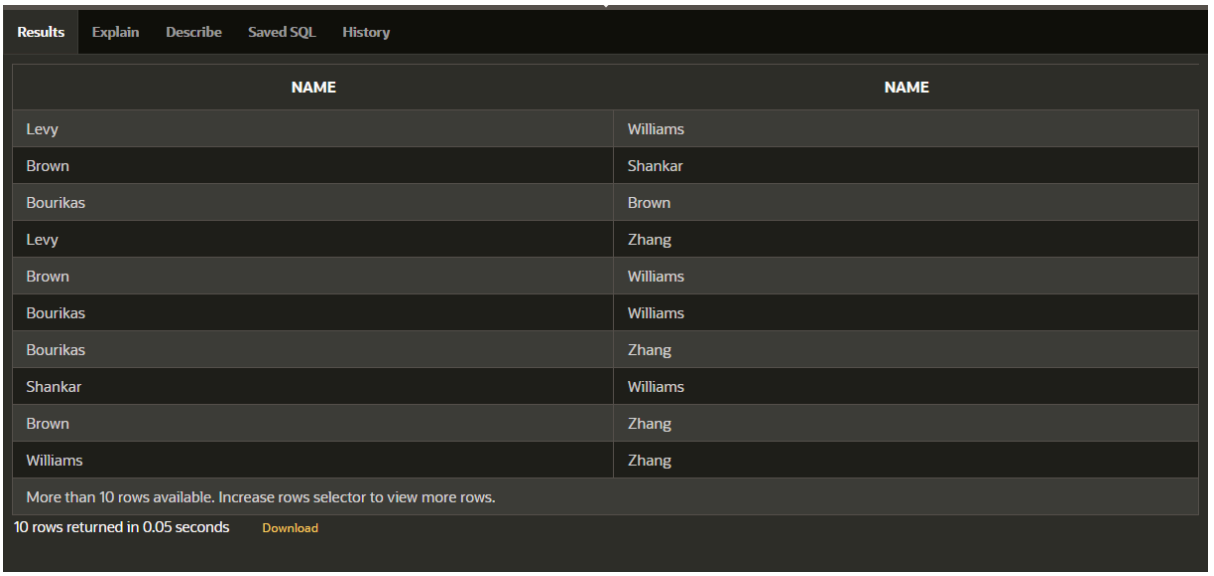


12. Afficher les noms des étudiants qui ont suivi un cours en automne 2009, en utilisant la clause EXISTS.



13. Afficher toutes les paires des étudiants qui ont suivi au moins un cours ensemble.





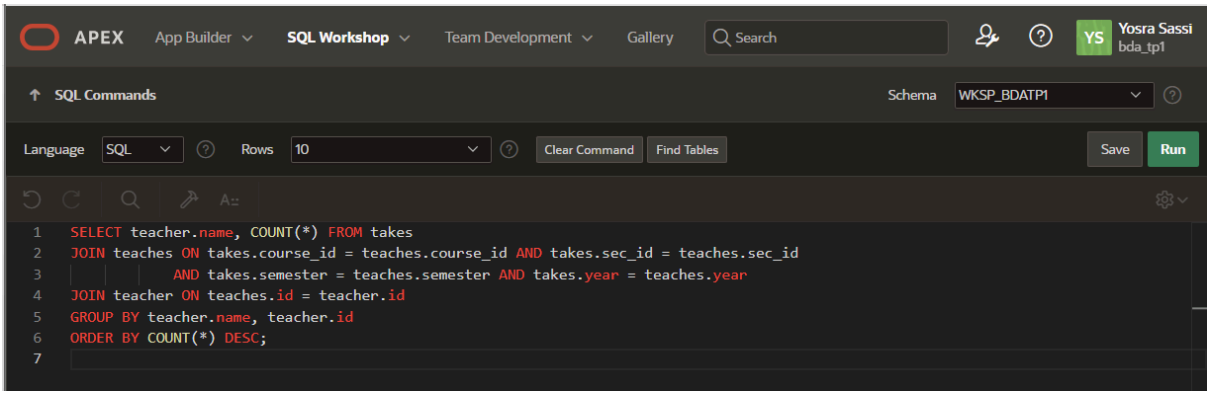
Results | Explain | Describe | Saved SQL | History

NAME	NAME
Levy	Williams
Brown	Shankar
Bourikas	Brown
Levy	Zhang
Brown	Williams
Bourikas	Williams
Bourikas	Zhang
Shankar	Williams
Brown	Zhang
Williams	Zhang

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.05 seconds Download

14. Afficher pour chaque enseignant, qui a effectivement assuré un cours, le nombre total d'étudiant qui ont suivi ses cours. Si un étudiant a suivi deux cours différents avec le même enseignant, on le compte deux fois. Trier le résultat par ordre décroissant.



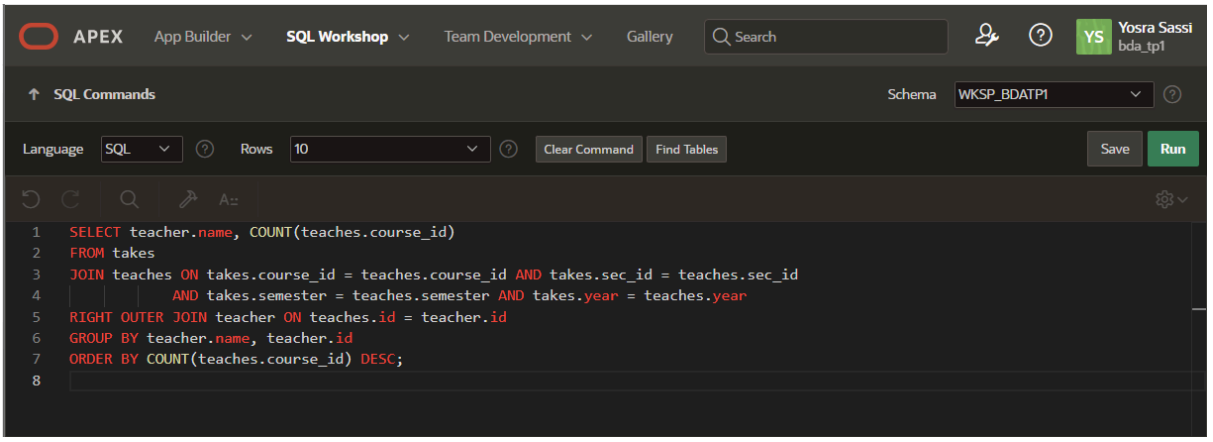
Results | Explain | Describe | Saved SQL | History

NAME	NAME
Levy	Williams
Brown	Shankar
Bourikas	Brown
Levy	Zhang
Brown	Williams
Bourikas	Williams
Bourikas	Zhang
Shankar	Williams
Brown	Zhang
Williams	Zhang

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.05 seconds Download

15. Afficher pour chaque enseignant, même s'il n'a pas assuré de cours, le nombre total d'étudiant qui ont suivi ses cours. Si un étudiant a suivi deux fois un cours avec le même enseignant, on le compte deux fois. Trier le résultat par ordre décroissant.



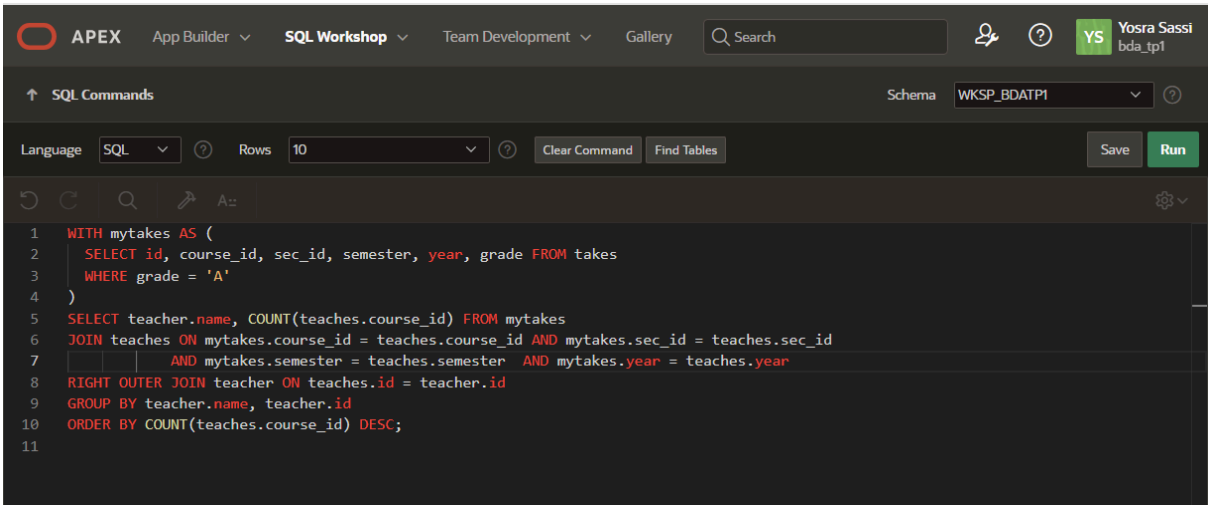
Results | Explain | Describe | Saved SQL | History

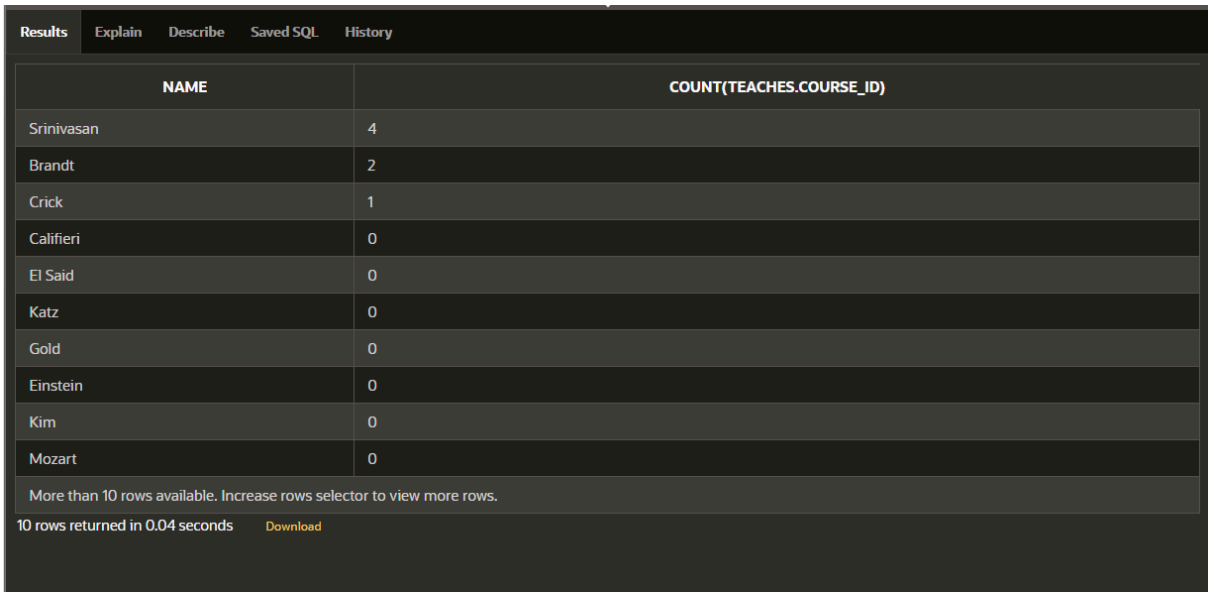
NAME	COUNT(TEACHES.COURSE_ID)
Srinivasan	10
Brandt	3
Katz	2
Crick	2
El Said	1
Einstein	1
Kim	1
Wu	1
Mozart	1
Gold	0

More than 10 rows available. Increase rows selector to view more rows.

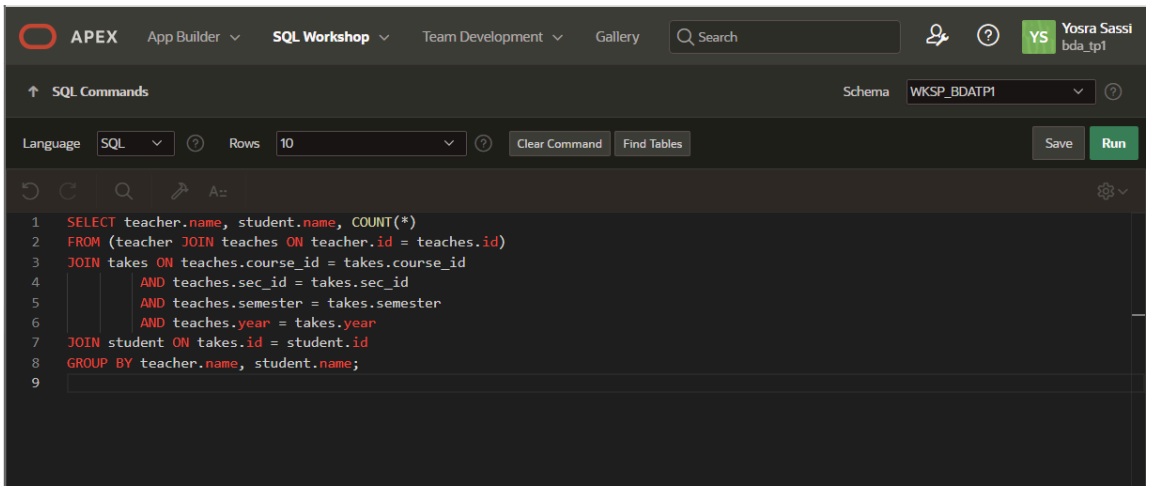
10 rows returned in 0.08 seconds Download

16. Pour chaque enseignant, afficher le nombre total de grades A qu'il a attribué.





17. Afficher toutes les paires enseignant-élèves où un élève a suivi le cours de l'enseignant, ainsi que le nombre de fois que cet élève a suivi un cours dispensé par cet enseignant.



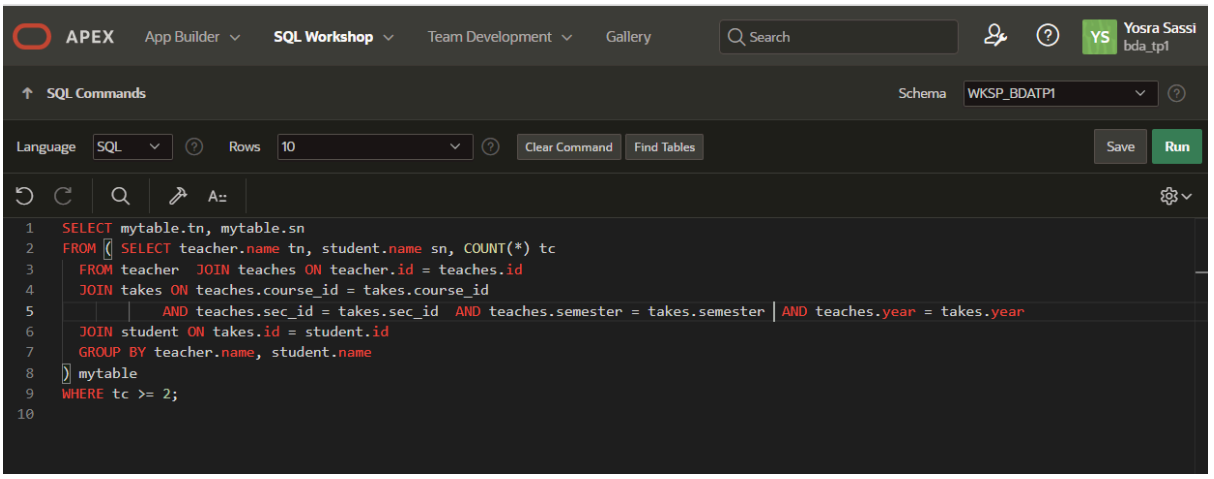
Results | Explain | Describe | Saved SQL | History

NAME	NAME	COUNT(*)
Brandt	Shankar	1
Wu	Chavez	1
El Said	Brandt	1
Einstein	Peltier	1
Brandt	Brown	1
Srinivasan	Bourikas	2
Mozart	Sanchez	1
Kim	Aoi	1
Srinivasan	Shankar	3
Brandt	Williams	1

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.01 seconds Download

18. Afficher toutes les paires enseignant-élève où un élève a suivi au moins deux cours dispensé par l'enseignant en question.



Results | Explain | Describe | Saved SQL | History

TN	SN
Srinivasan	Bourikas
Srinivasan	Shankar
Crick	Tanaka
Katz	Levy
Srinivasan	Zhang

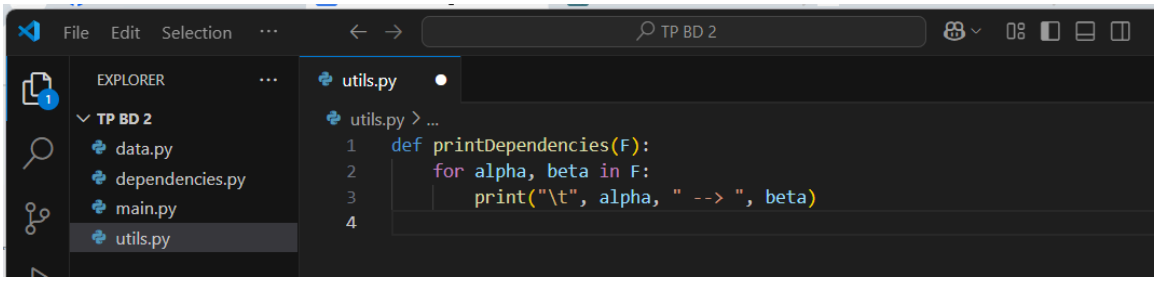
5 rows returned in 0.04 seconds Download

Exercice 2 – Requêtes SQL

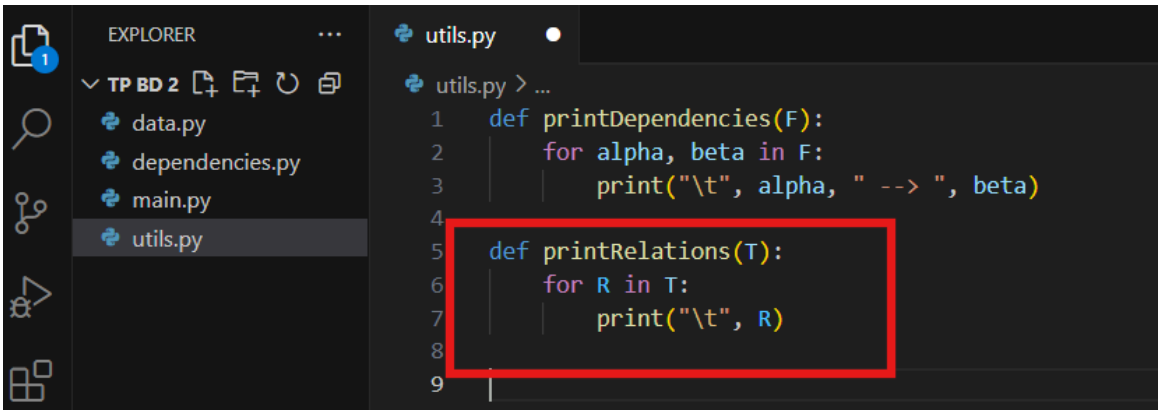
Cet exercice consiste à manipuler des dépendances fonctionnelles, tester des propriétés comme super-clé et clé candidate, vérifier si une relation est en **BCNF**, et appliquer une **décomposition** en BCNF en Python.

Fonctions disponibles :

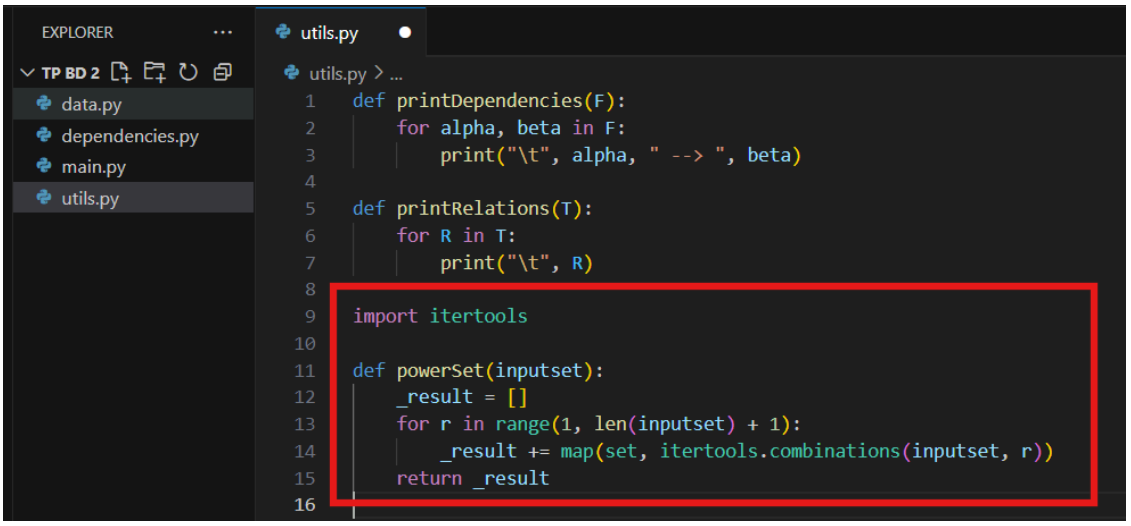
1. Affichage des dépendances et relations :



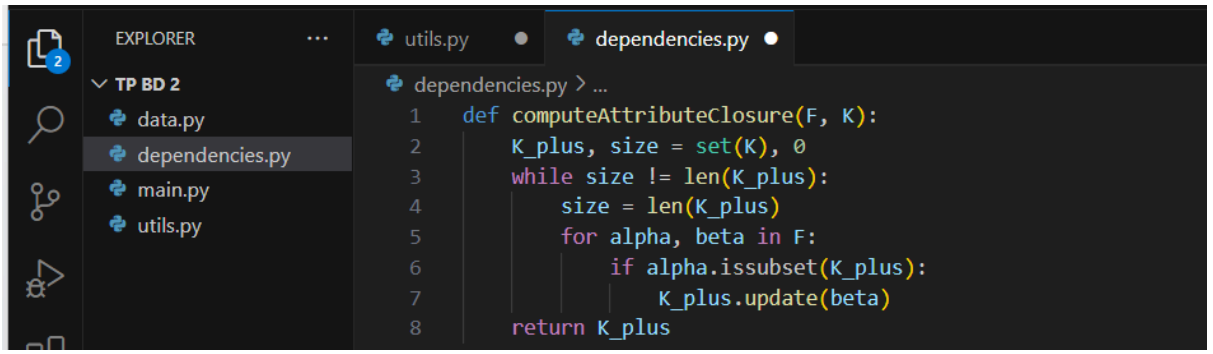
2. Affichage des relations :



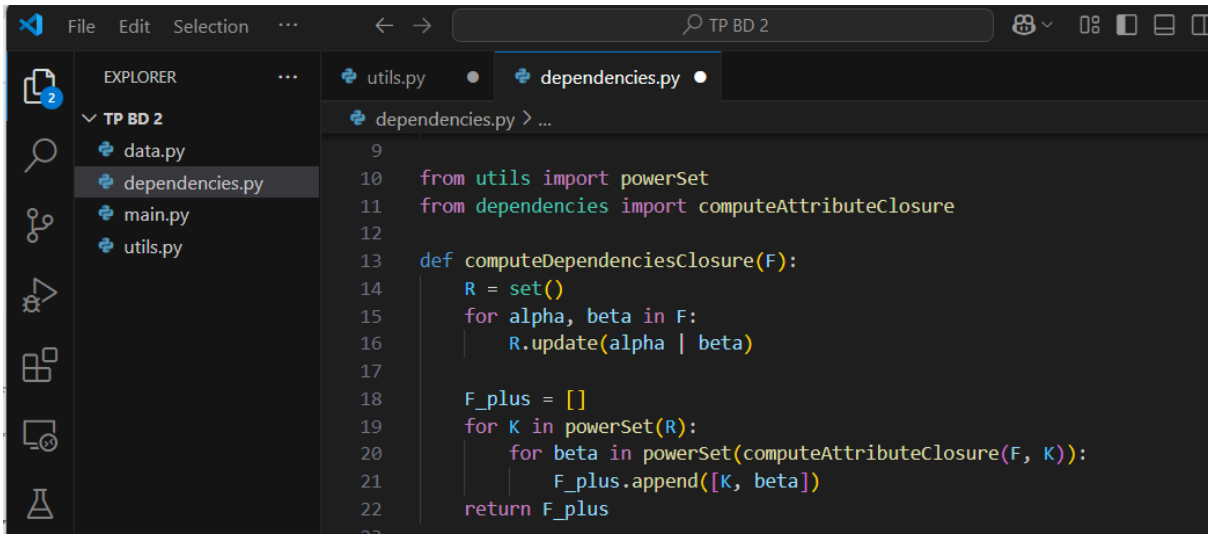
3. Génération de tous les sous-ensembles (power set) :



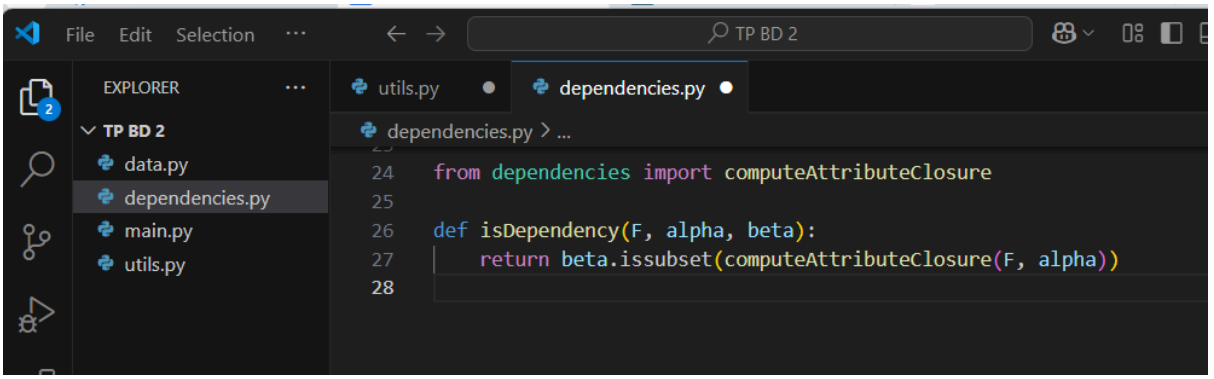
4. Calcul de la fermeture d'un ensemble d'attributs :



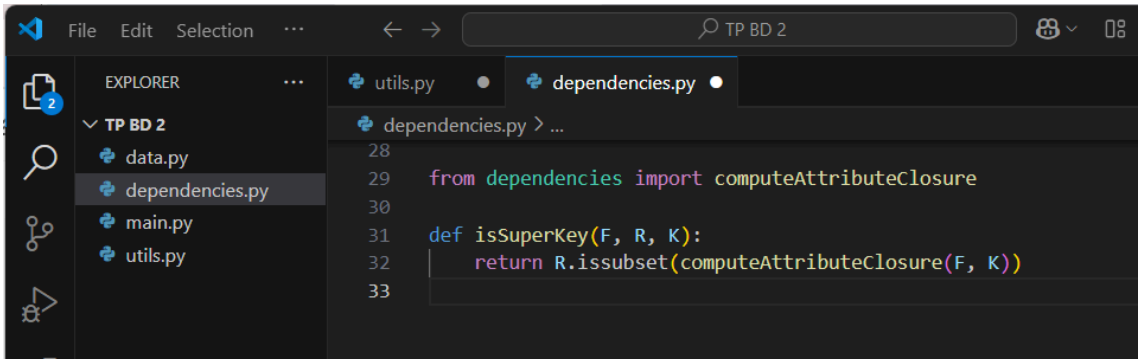
5. Calcul de la fermeture d'un ensemble de dépendances :



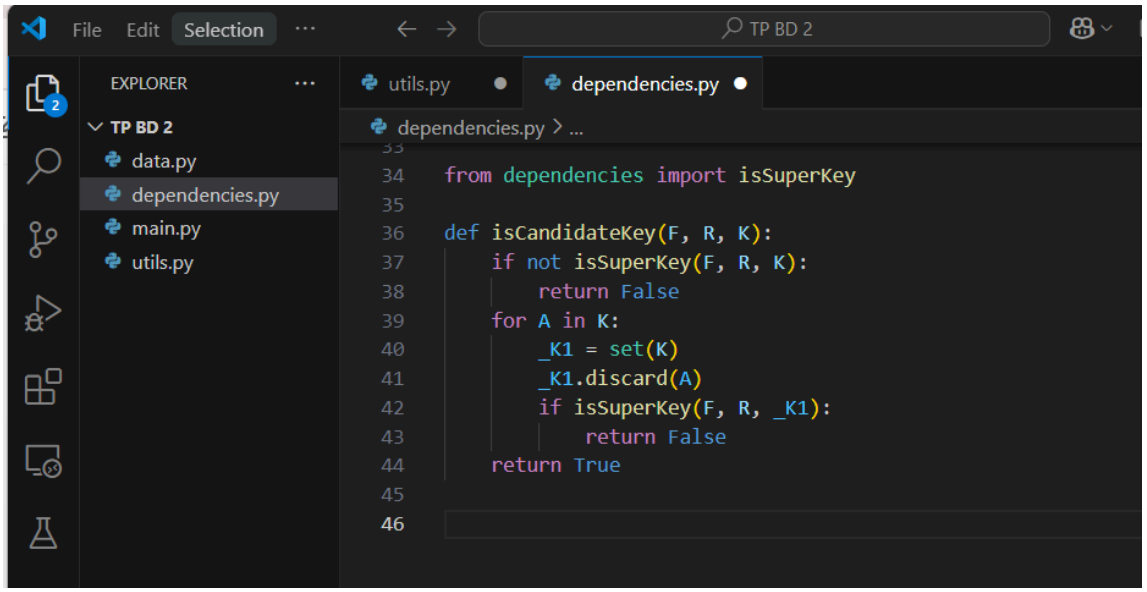
6. Vérification d'une dépendance déduite ($\alpha \rightarrow \beta$?) :



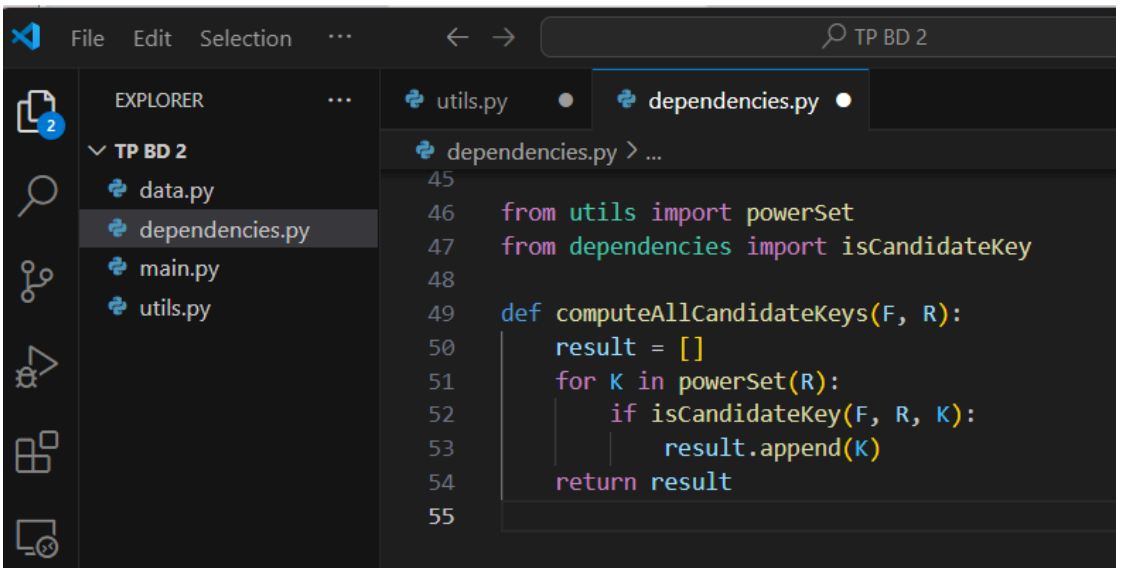
7. Vérification si un ensemble est une super-clé :



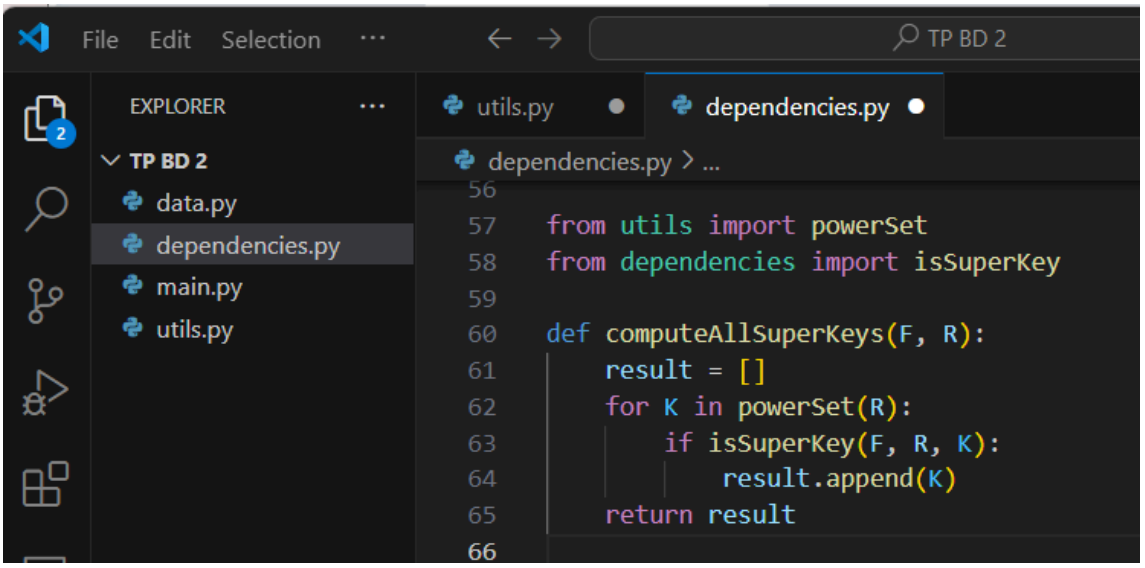
8. Vérification si un ensemble est une clé candidate :



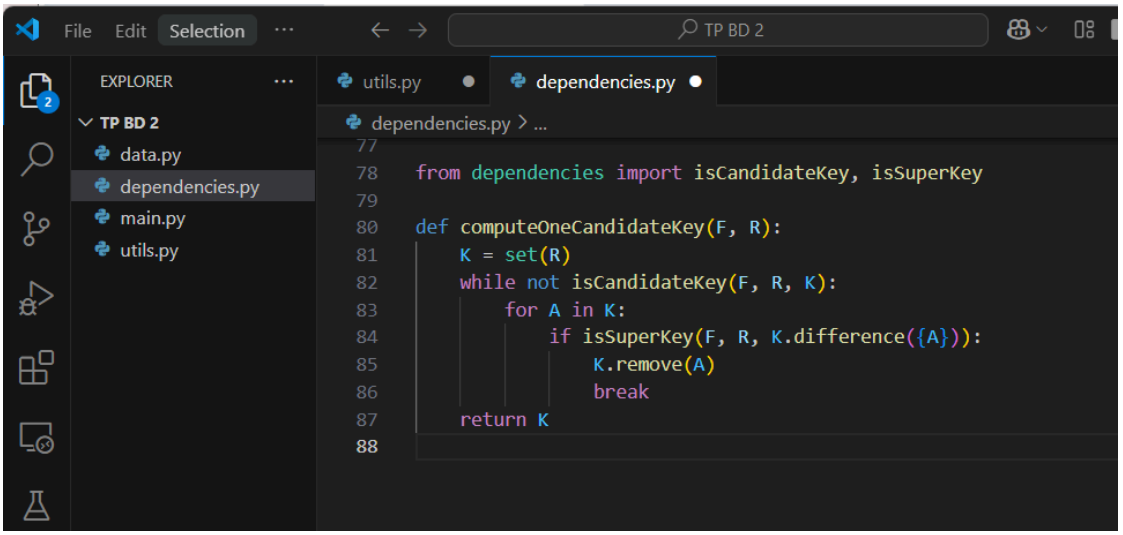
9. Calcul de toutes les clés candidates :



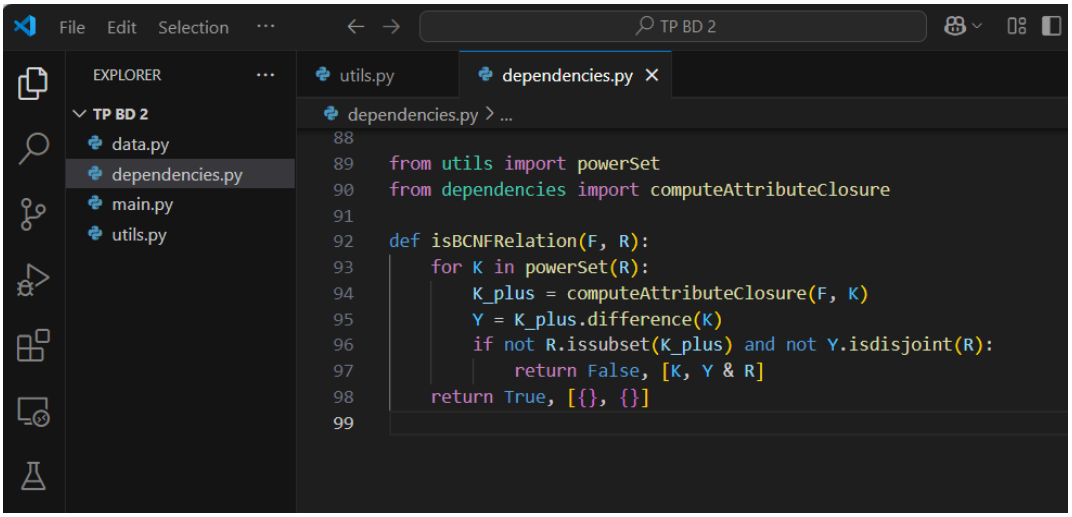
10. Calcul de toutes les super-clés :



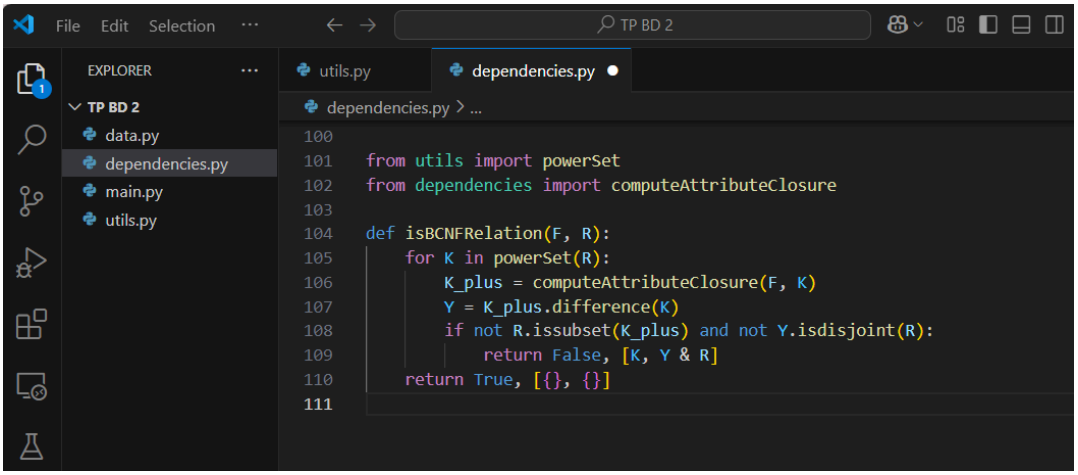
11. Calcul d'une seule clé candidate :



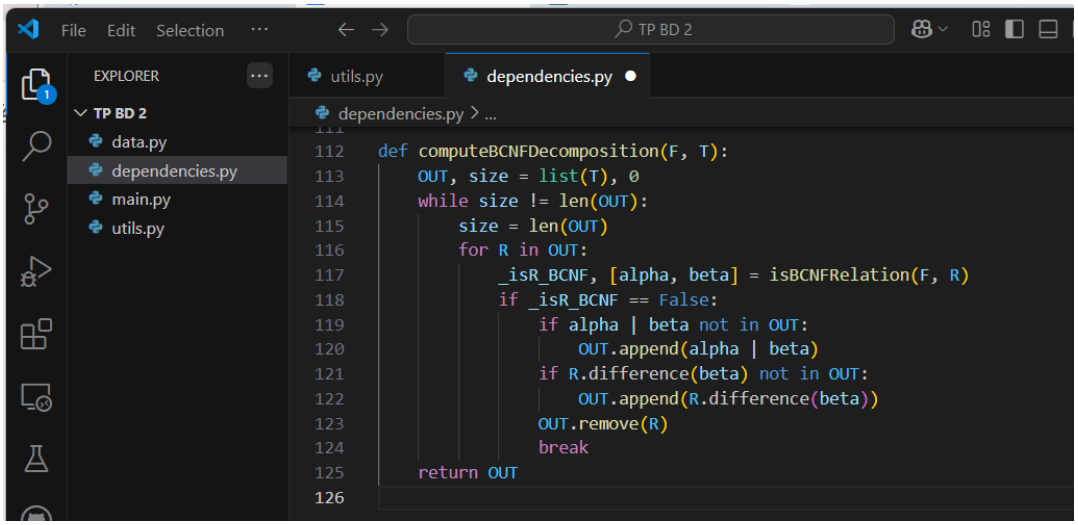
12. Vérification si une relation est en BCNF :



13. Vérification si un schéma complet est en BCNF :

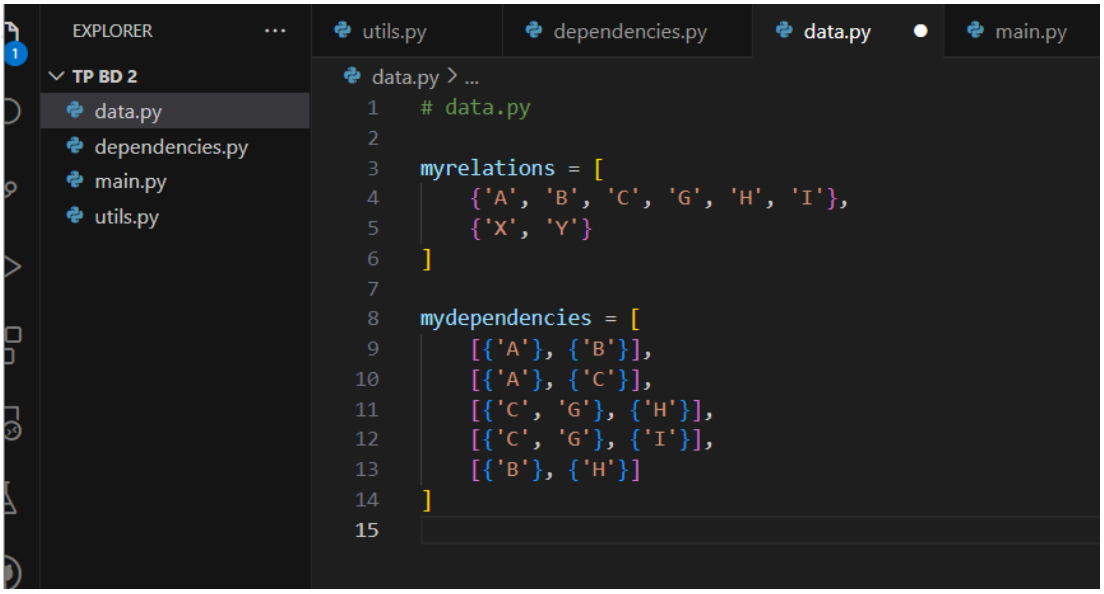


14. Algorithme de décomposition d'un schéma en BCNF :

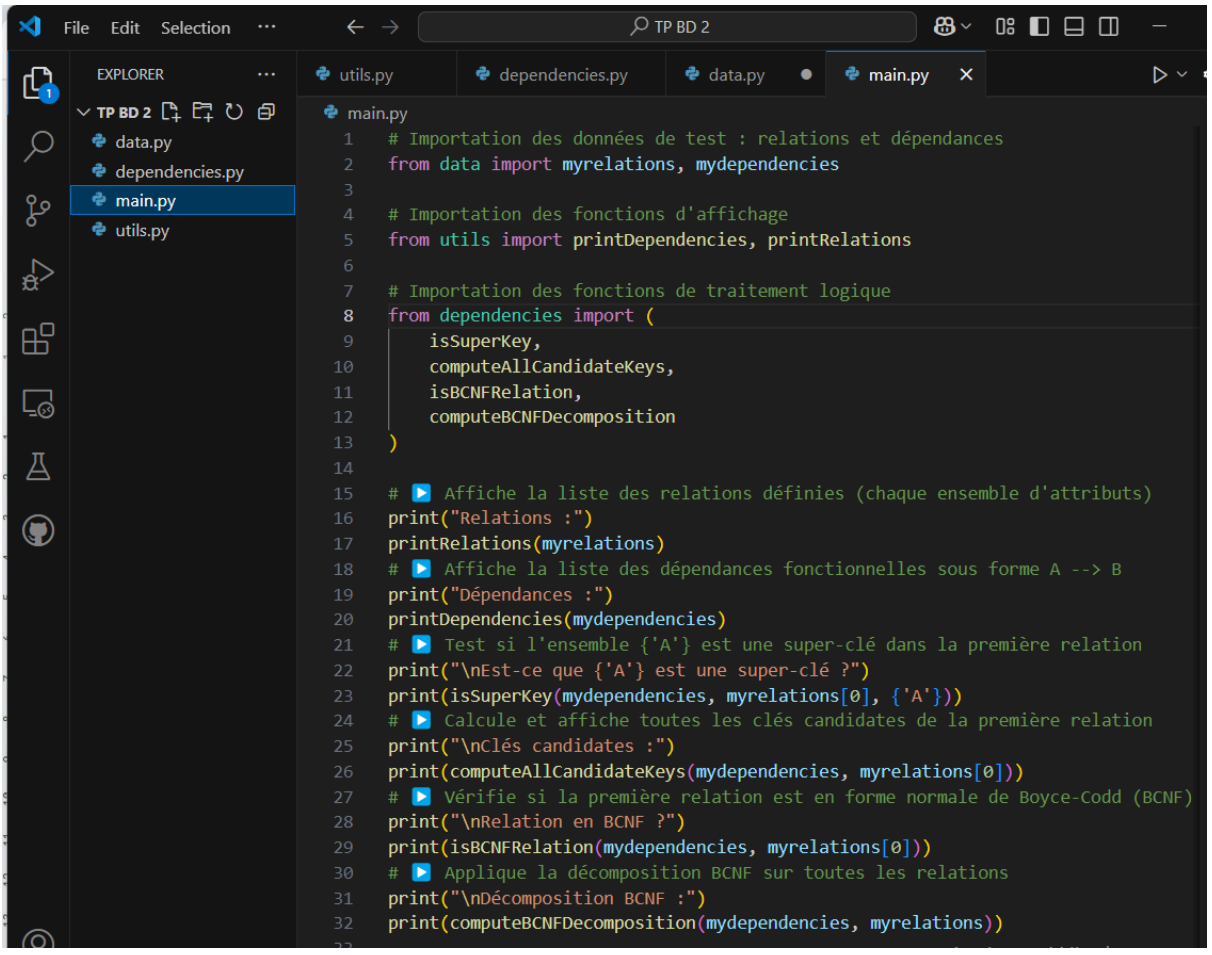


Script de test

Fichier : data.py : **Ce fichier sert de base de données locale** pour toutes les fonctions.



Fichier : main.py : Ce fichier aide à **afficher** et **manipuler** facilement les structures de données.



Étape finale : Lancer le test

Dans ton terminal, exécute le fichier : **python main.py**

```
PS C:\Users\Lenovo\Desktop\Sup Galilée - Cours\Base de données\TP BD 2> python main.py
Relations :
        {'A', 'H', 'C', 'I', 'B', 'G'}
        {'Y', 'X'}
Dépendances :
        {'A'}  -->  {'B'}
        {'A'}  -->  {'C'}
        {'C', 'G'}  -->  {'H'}
        {'C', 'G'}  -->  {'I'}
        {'B'}  -->  {'H'}

Est-ce que {'A'} est une super-clé ?
False

Clés candidates :
[{'A', 'G'}]

Relation en BCNF ?
(False, [{'A'}, {'H', 'C', 'B'}])

Décomposition BCNF :
[{'Y', 'X'}, {'I', 'A', 'G'}, {'H', 'B'}, {'A', 'C', 'B'}]
PS C:\Users\Lenovo\Desktop\Sup Galilée - Cours\Base de données\TP BD 2>
```

→ Les deux relations affichées représentent des schémas relationnels sur lesquels s'appliquent les dépendances fonctionnelles listées.

→ Le test effectué montre que l'attribut `{A}` n'est pas une super-clé, car il ne permet pas à lui seul de déterminer tous les attributs de la relation.