TP1 BDA

Exercice 1 :

1) création de tables : course, departement, classroom, section

```
create table classroom
    (building       varchar(15),
     room_number        varchar(7),
     capacity       numeric(4,0),
     primary key (building, room_number)
    );

create table department
    (dept_name      varchar(20),
     building       varchar(15),
     budget             numeric(12,2),
     primary key (dept_name)
    );

create table course
    (course_id      varchar(8),
     title          varchar(50),
     dept_name      varchar(20),
     credits        numeric(2,0),
     primary key (course_id),
     foreign key (dept_name) references department
    );
```

```
create table section
    (course_id      varchar(8),
         sec_id         varchar(8),
     semester       varchar(6)
        check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),
     year           numeric(4,0),
     building       varchar(15),
     room_number        varchar(7),
     time_slot_id       varchar(4),
     primary key (course_id, sec_id, semester, year),
     foreign key (course_id) references course,
     foreign key (building, room_number) references classroom
    );
```

2) création des autres tables :

```sql
create table teacher
    (ID         varchar(5),
     name           varchar(20),
     dept_name      varchar(20),
     salary         numeric(8,2),
     primary key (ID),
     foreign key (dept_name) references department
    );
```

```sql
create table teaches
    (ID         varchar(5),
     course_id      varchar(8),
     sec_id         varchar(8),
     semester       varchar(6),
     year           numeric(4,0),
     primary key (ID, course_id, sec_id, semester, year),
     foreign key (course_id,sec_id, semester, year) references section,
     foreign key (ID) references teacher
    );
```

```sql
create table student
    (ID         varchar(5),
     name           varchar(20),
     dept_name      varchar(20),
     tot_cred       numeric(3,0),
     primary key (ID),
     foreign key (dept_name) references department
    );
```

```sql
create table takes
    (ID         varchar(5),
     course_id      varchar(8),
     sec_id         varchar(8),
     semester       varchar(6),
     year           numeric(4,0),
     grade              varchar(2),
     primary key (ID, course_id, sec_id, semester, year),
     foreign key (course_id,sec_id, semester, year) references section,
     foreign key (ID) references student
    );
```

```sql
create table advisor
    (s_ID           varchar(5),
     i_ID           varchar(5),
     primary key (s_ID),
     foreign key (i_ID) references teacher (ID),
     foreign key (s_ID) references student (ID)
    );
```

```sql
create table time_slot
    (time_slot_id   varchar(4),
     day        varchar(1),
     start_hr   numeric(2),
     start_min  numeric(2),
     end_hr     numeric(2),
     end_min    numeric(2),
     primary key (time_slot_id, day, start_hr, start_min)
    );
```

```sql
create table prereq
    (course_id      varchar(8),
     prereq_id      varchar(8),
     primary key (course_id, prereq_id),
     foreign key (course_id) references course,
     foreign key (prereq_id) references course
    );
```

3) Requêtes insert :

	0.01	insert into classroom values ('Packard', '101', '500')	1 row(s) inserted.	1
	0.00	insert into classroom values ('Painter', '514', '10')	1 row(s) inserted.	1
	0.00	insert into classroom values ('Taylor', '3128', '70')	1 row(s) inserted.	1
	0.01	insert into classroom values ('Watson', '100', '30')	1 row(s) inserted.	1
	0.00	insert into classroom values ('Watson', '120', '50')	1 row(s) inserted.	1
	0.00	insert into department values ('Biology', 'Watson', '90000')	1 row(s) inserted.	1
	0.00	insert into department values ('Comp. Sci.', 'Taylor', '1000	1 row(s) inserted.	1
	0.00	insert into department values ('Elec. Eng.', 'Taylor', '8500	1 row(s) inserted.	1
	0.00	insert into department values ('Finance', 'Painter', '120000	1 row(s) inserted.	1

load

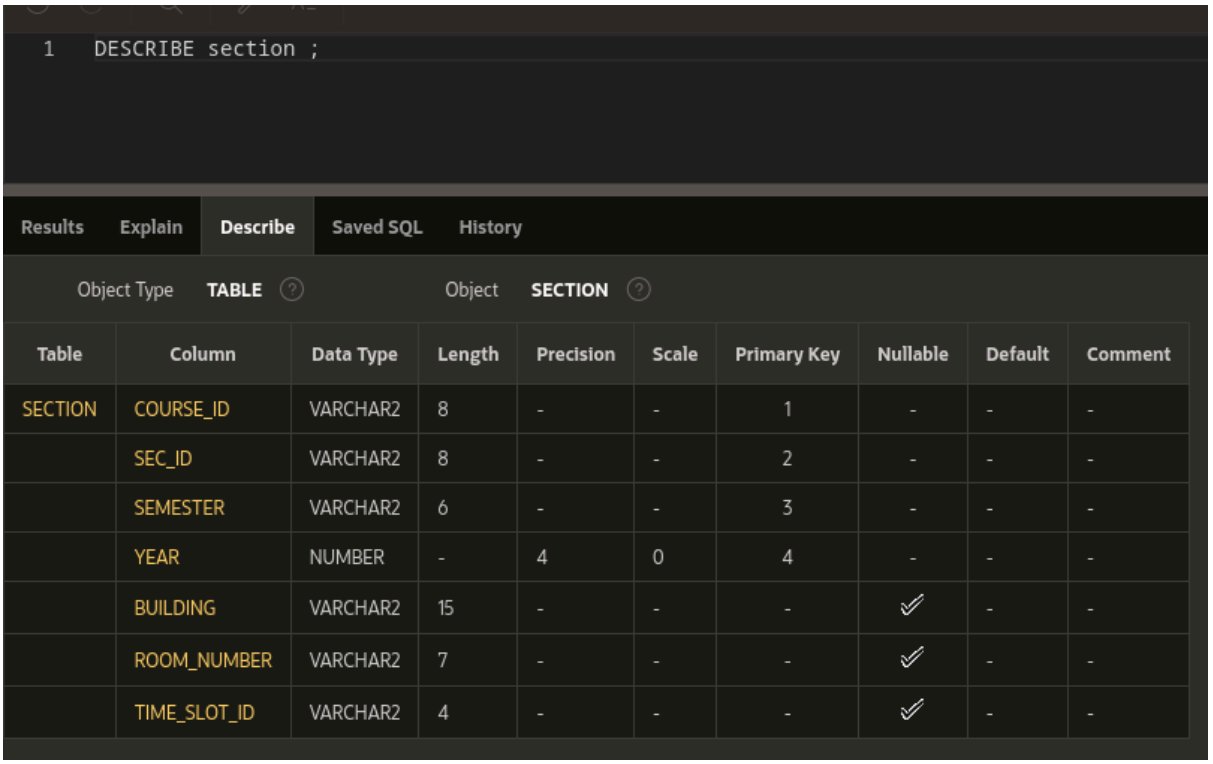
row(s) 1 - 20 of 149 Next ▸

149	149	0
Statements Processed	Successful	With Errors

yrinetlili07@gmail.com bdainfo en Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.2.1

Exercice 2 :

1. Afficher la structure de la relation section et son contenu :

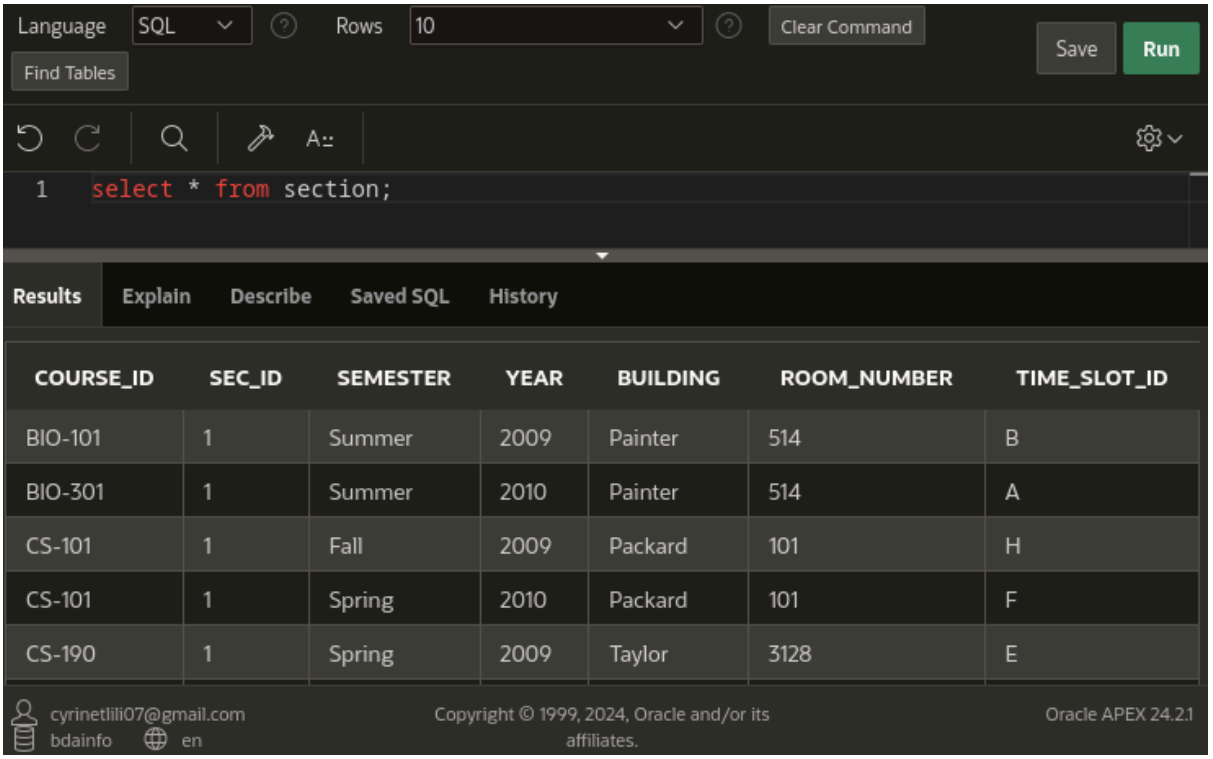


```
1  DESCRIBE section ;
```

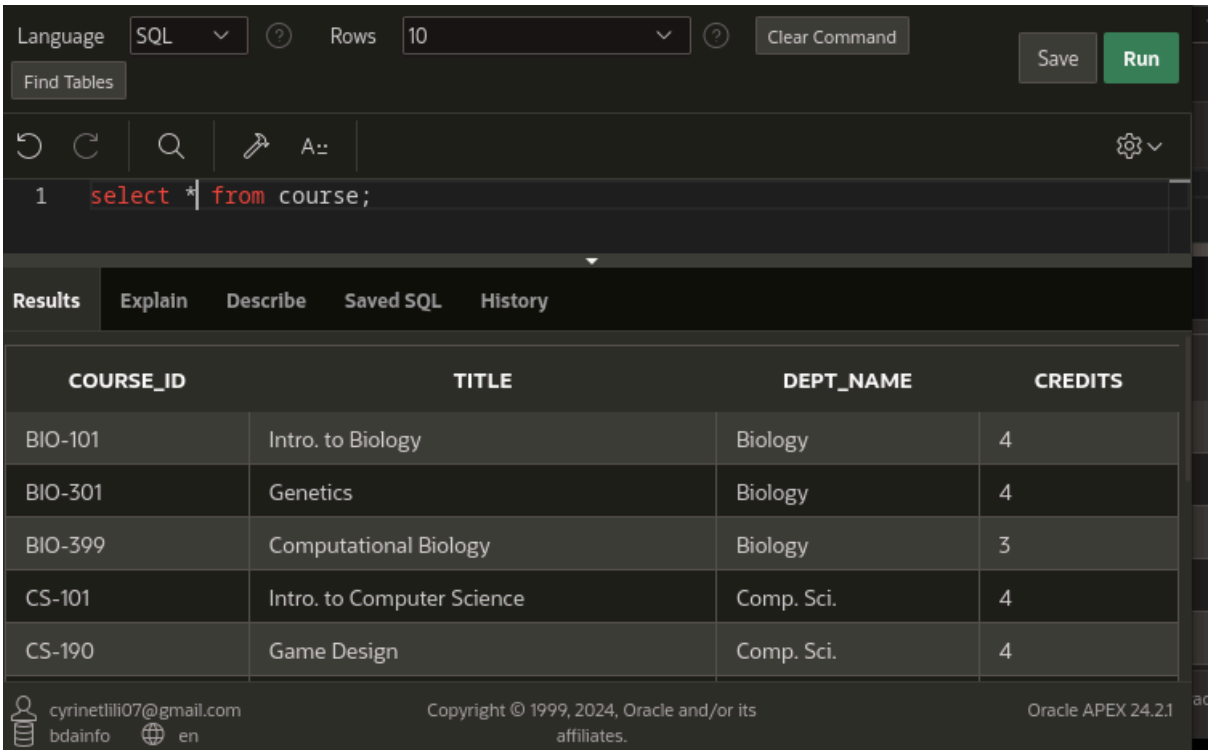
Results Explain **Describe** Saved SQL History

Object Type **TABLE** Object **SECTION**

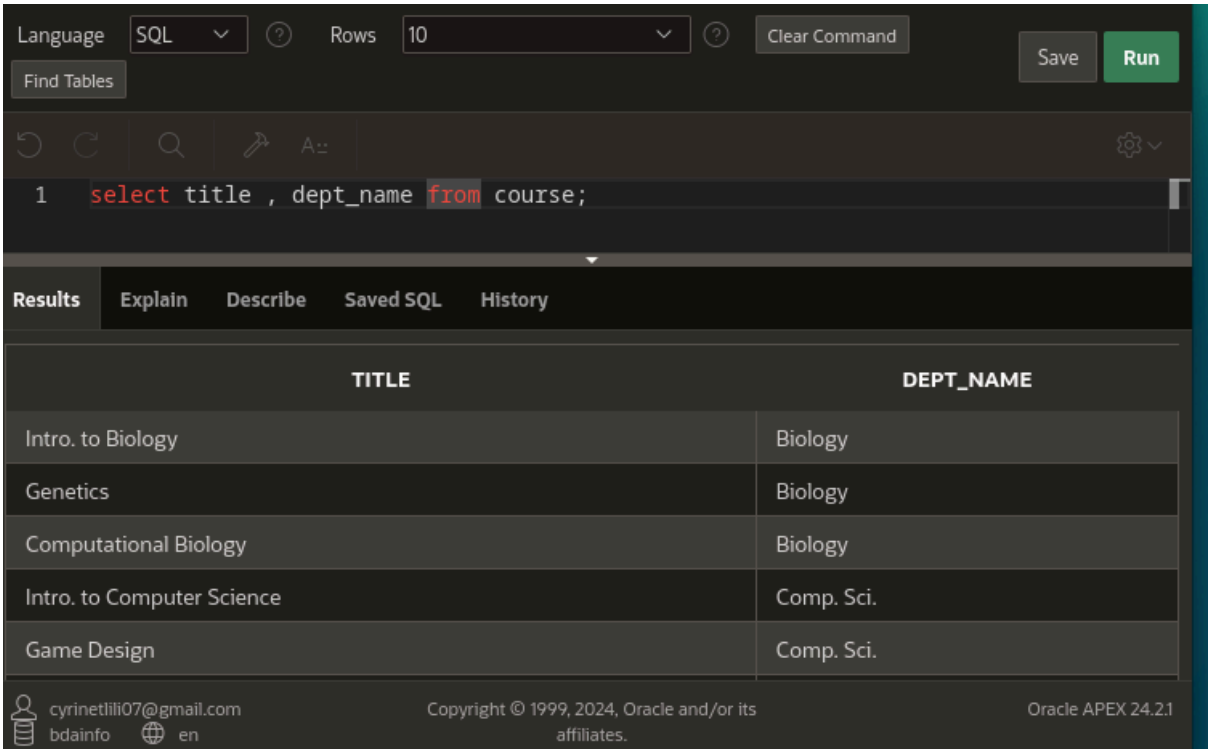
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SECTION	COURSE_ID	VARCHAR2	8	-	-	1	-	-	-
	SEC_ID	VARCHAR2	8	-	-	2	-	-	-
	SEMESTER	VARCHAR2	6	-	-	3	-	-	-
	YEAR	NUMBER	-	4	0	4	-	-	-
	BUILDING	VARCHAR2	15	-	-	-	✓	-	-
	ROOM_NUMBER	VARCHAR2	7	-	-	-	✓	-	-
	TIME_SLOT_ID	VARCHAR2	4	-	-	-	✓	-	-



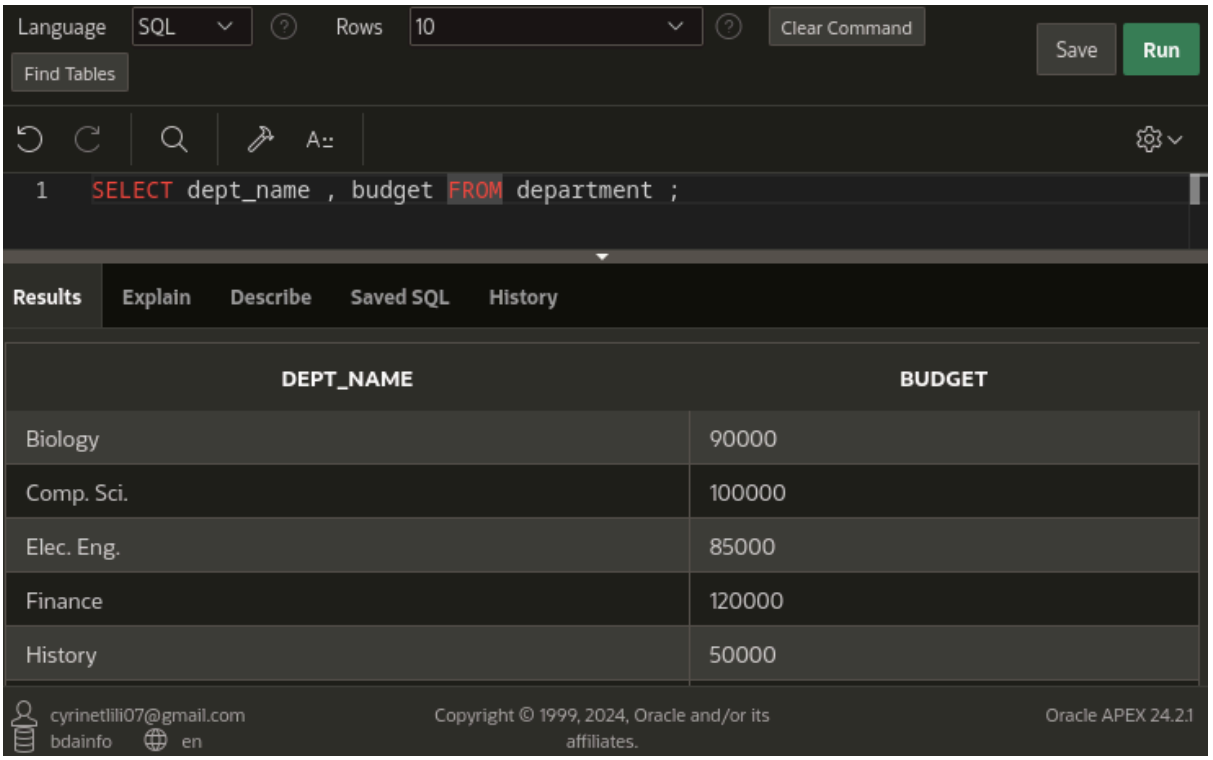
2. Afficher tous les renseignements sur les cours que l'on peut programmer (relation course):



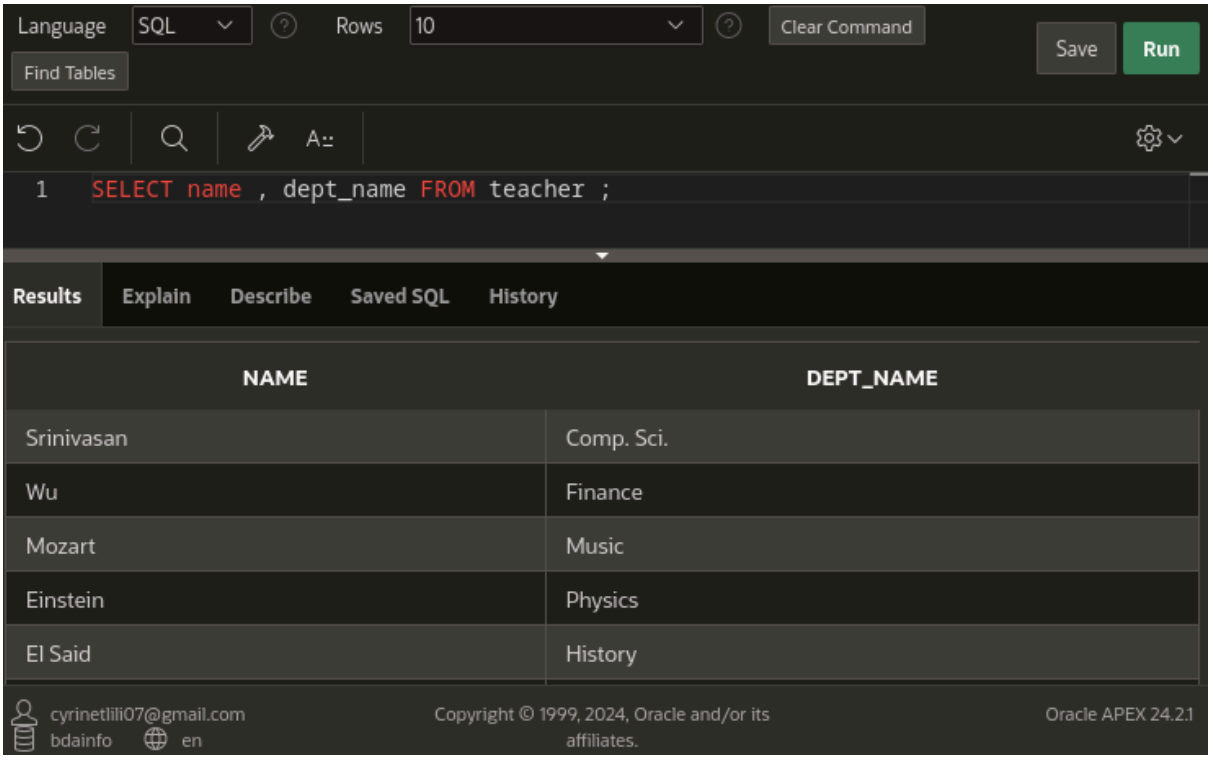
3. Afficher les titres des cours et les départements qui les proposent :



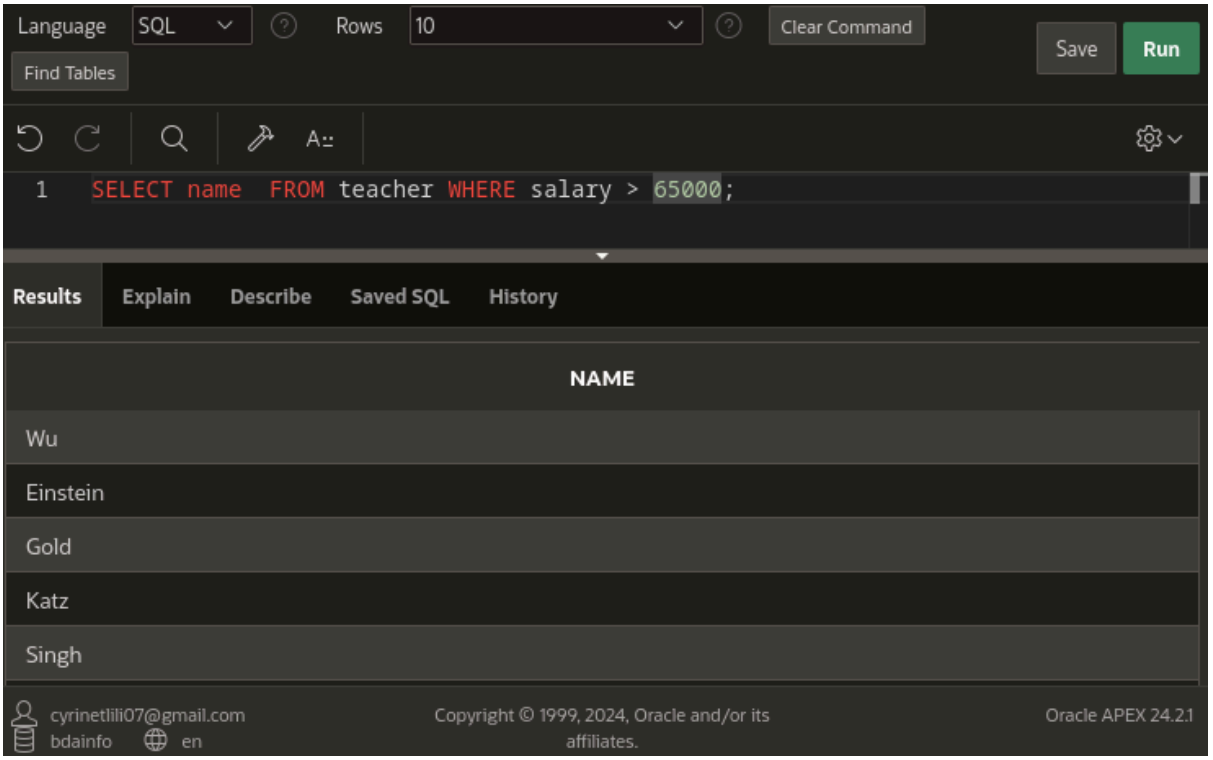
4. Afficher les noms des départements ainsi que leur budget.



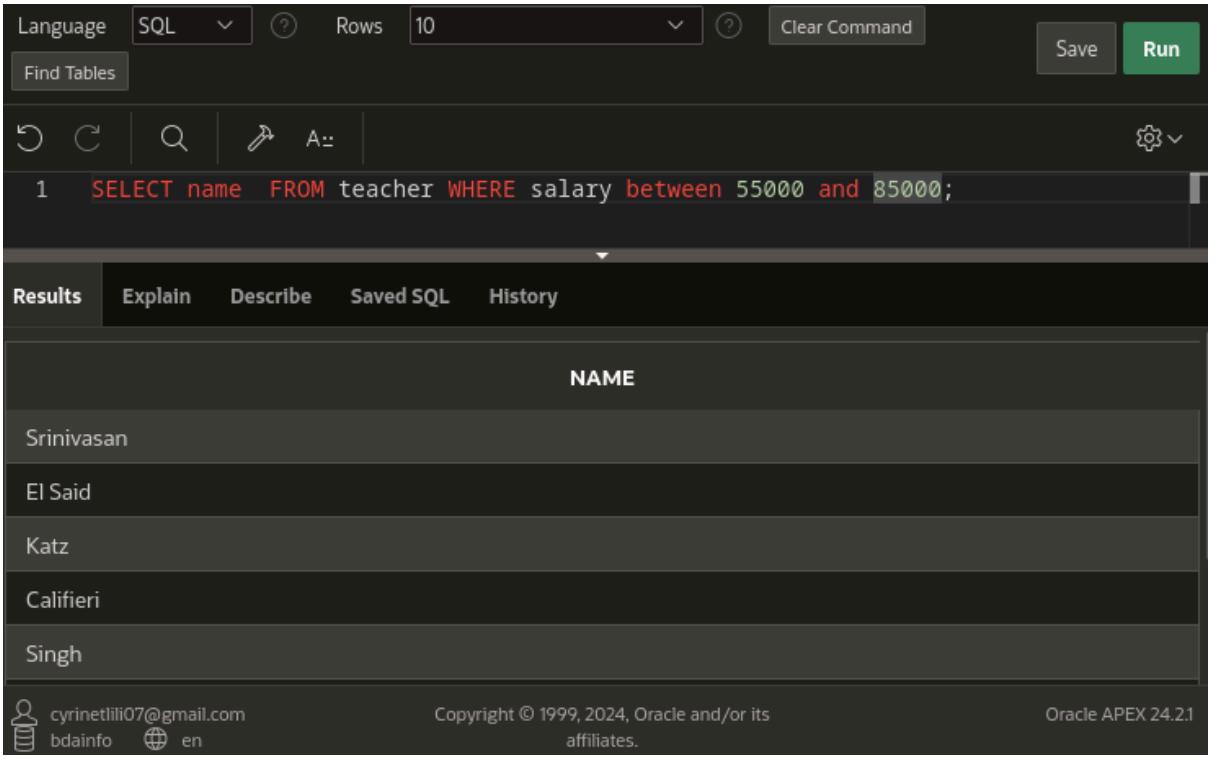
5. Afficher tous les noms des enseignants et leur département.



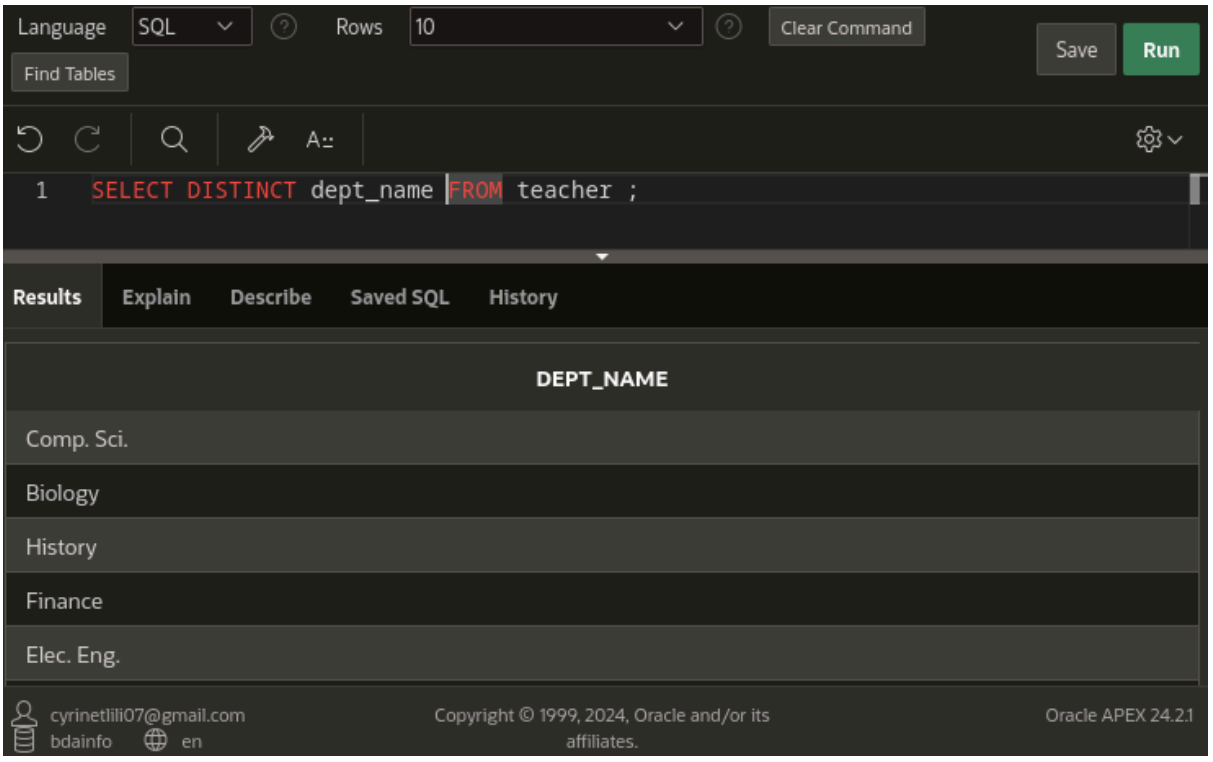
6. Afficher tous les noms des enseignants ayant un salaire supérieur strictement à 65.000 $.



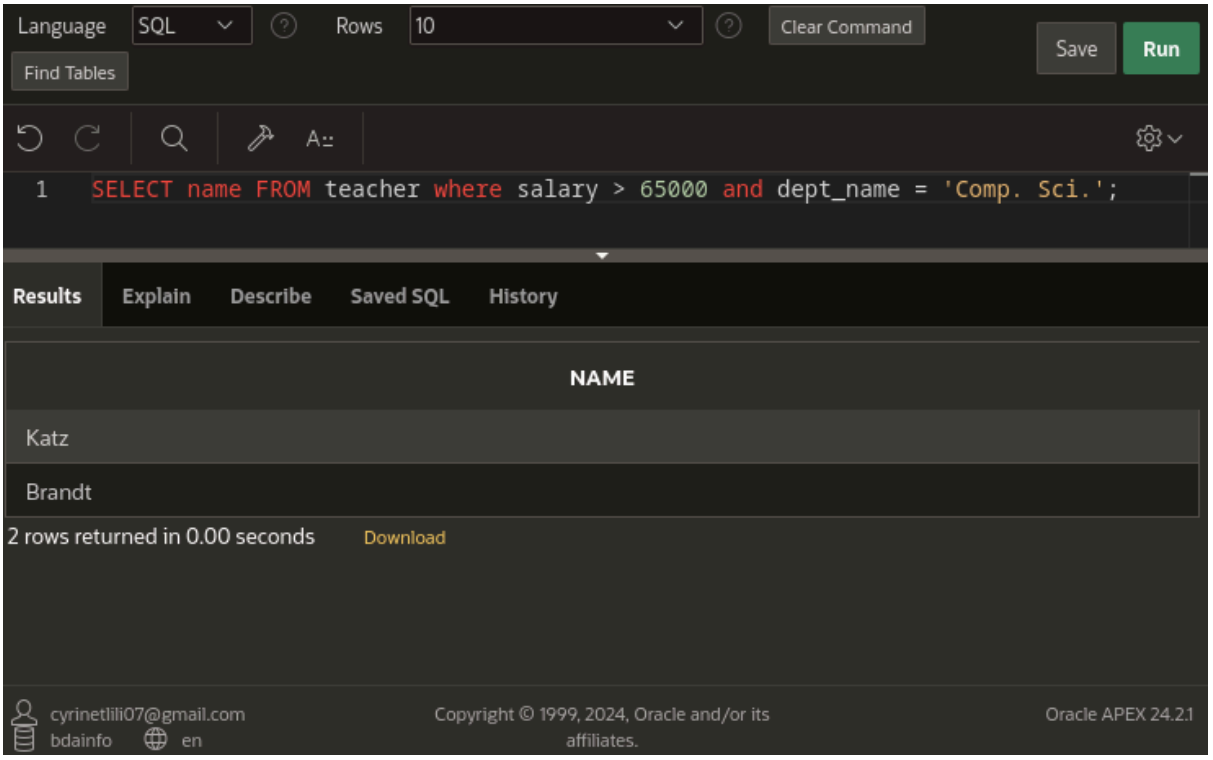
7. Afficher les noms des enseignants ayant un salaire compris entre 55.000 $ et 85.000 $.



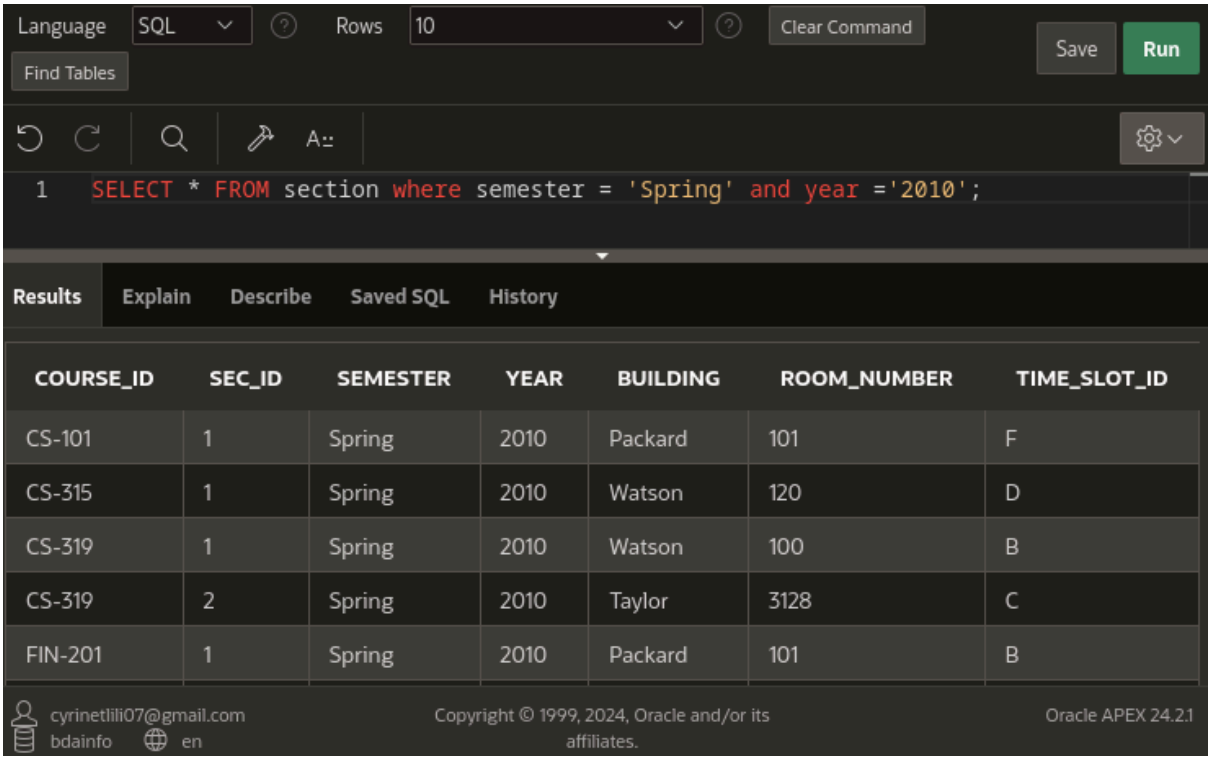
8. Afficher les noms des départements, en utilisant la relation teacher et éliminer les doublons :



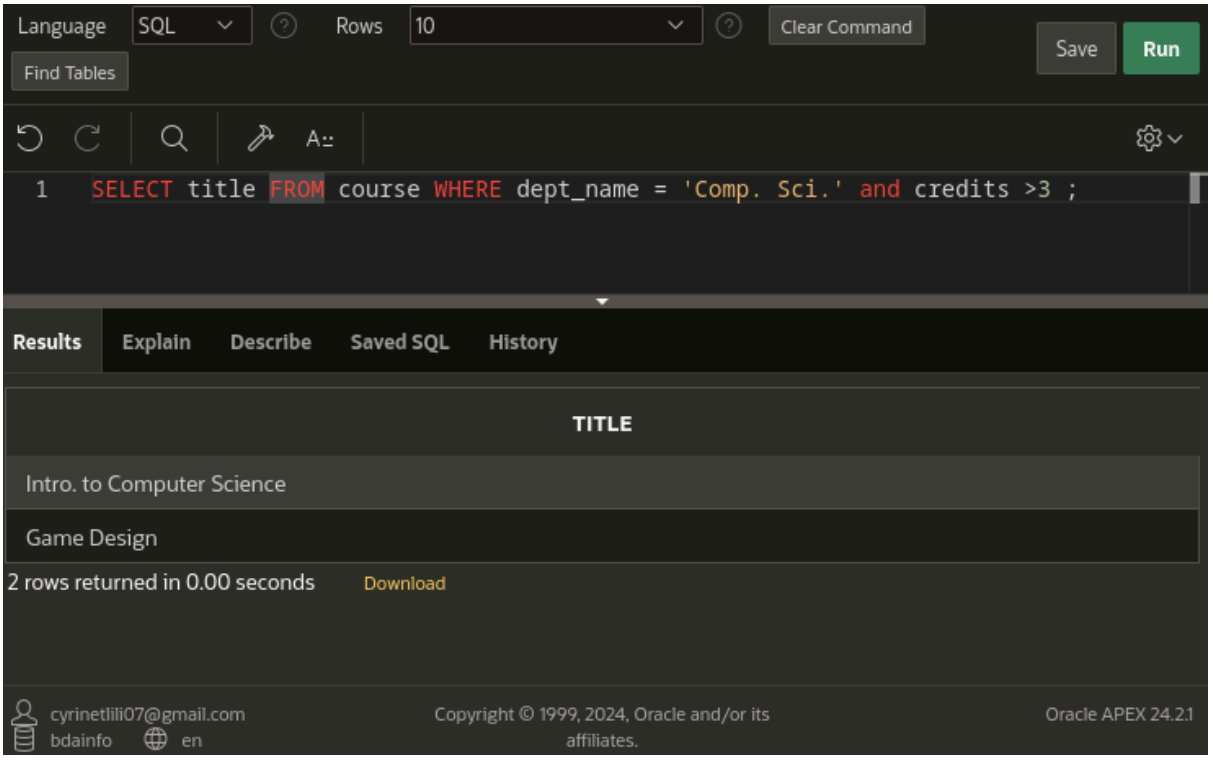
9. Afficher tous les noms des enseignants du département informatique ayant un salaire supérieur strictement à 65.000 $.



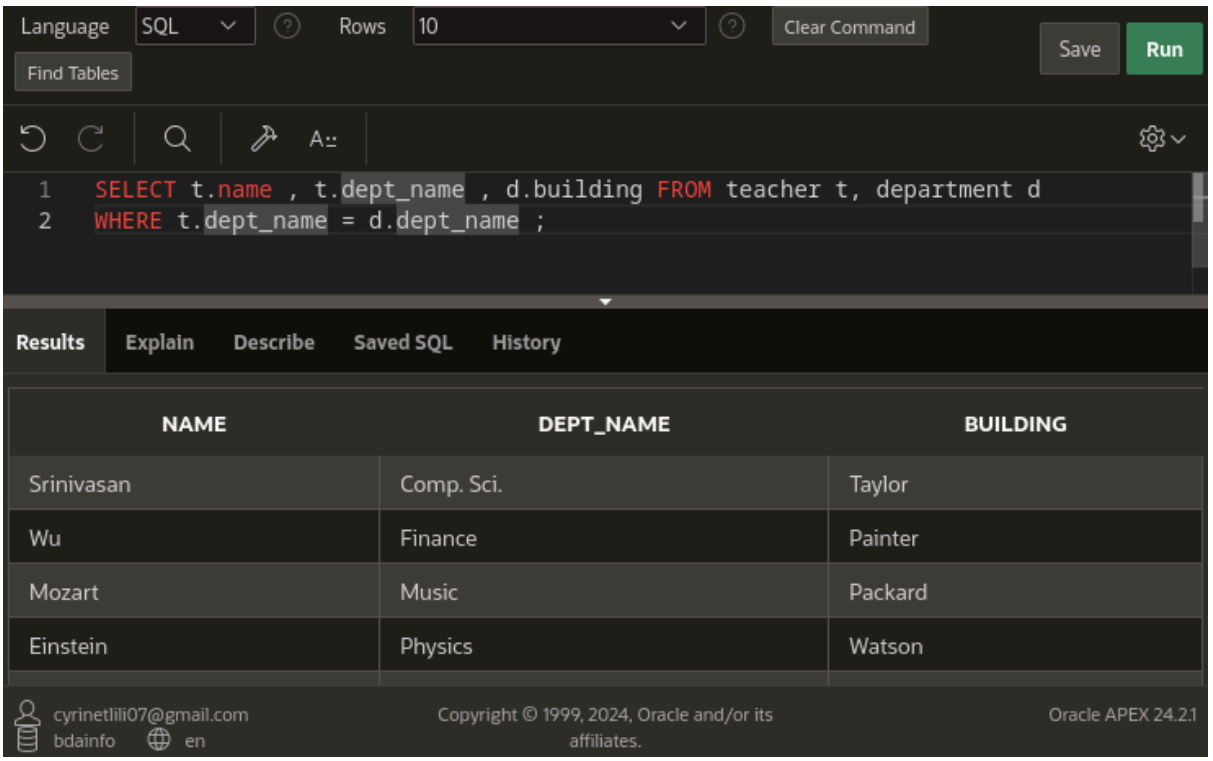
10. Afficher tous les renseignements sur les cours proposés au printemps 2010 :



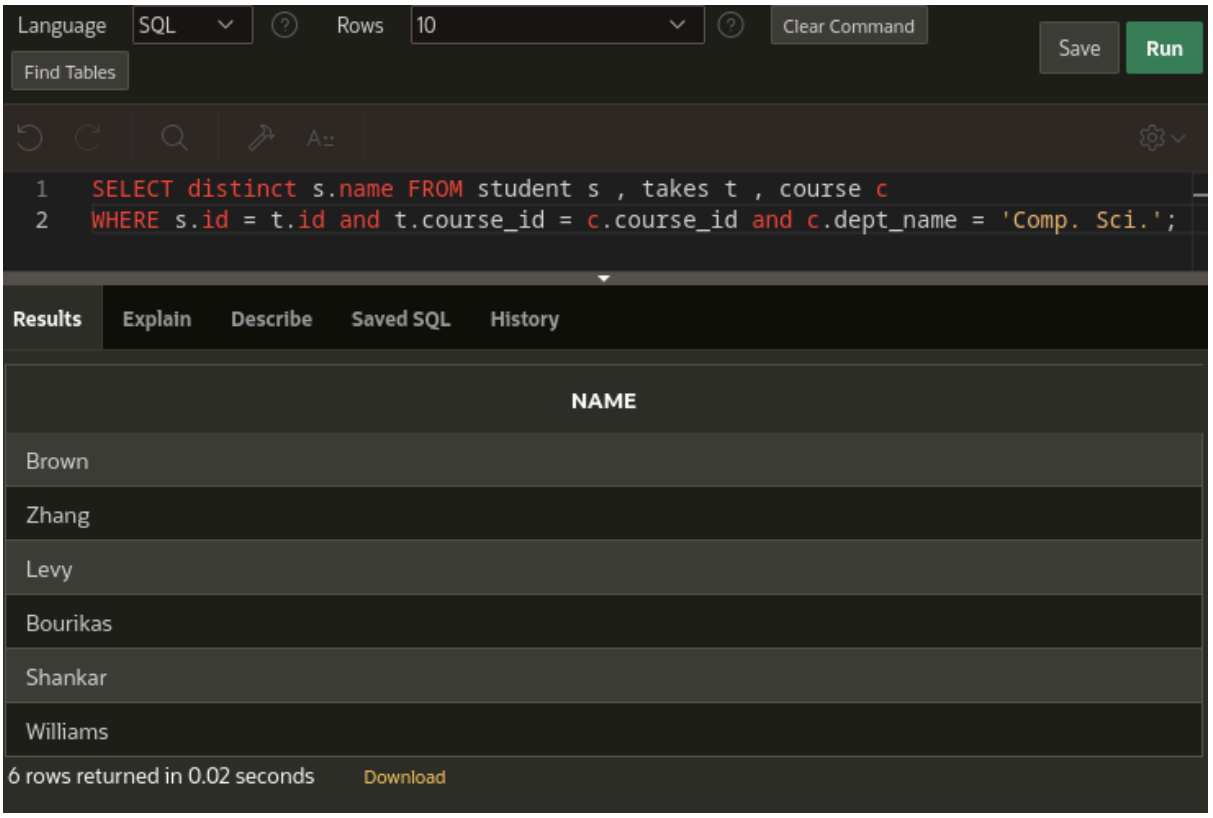
11. Afficher tous les titres des cours dispensés par le département informatique qui ont plus de trois crédits.



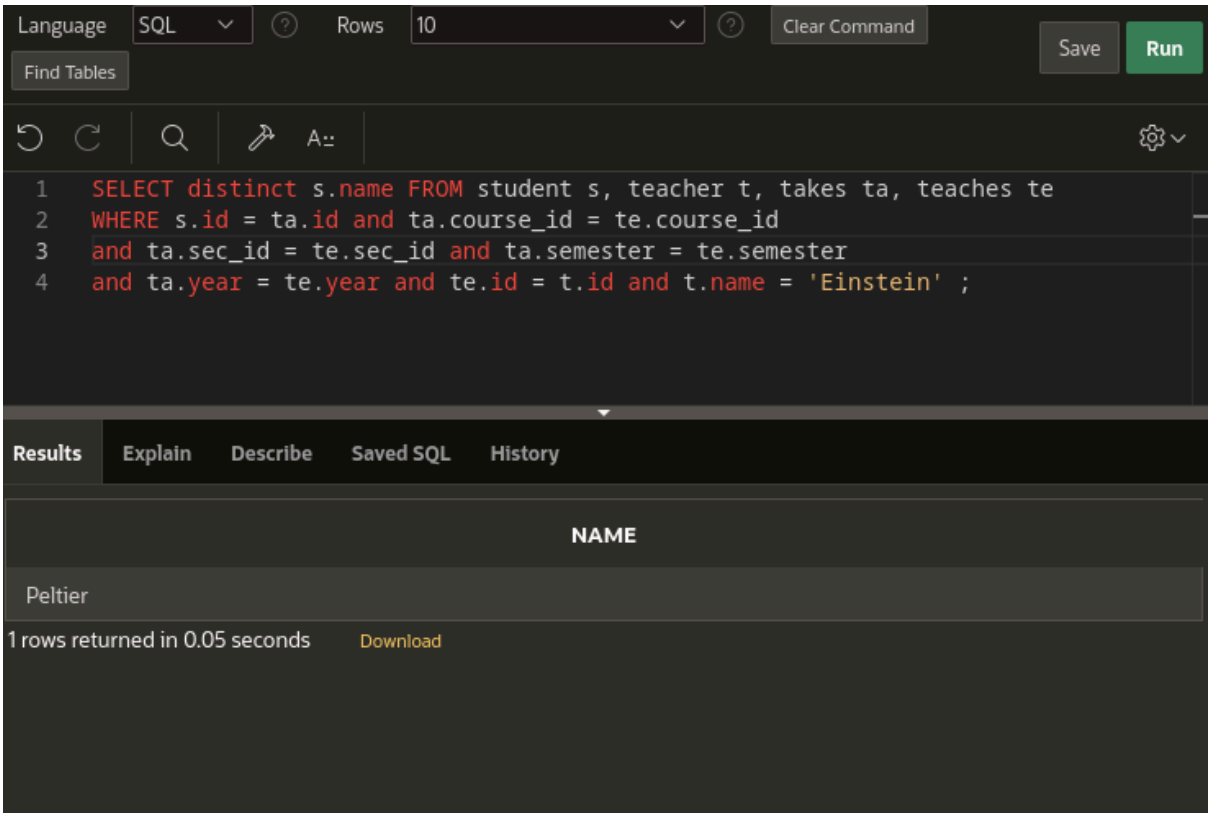
12. Afficher tous les noms des enseignants ainsi que le nom de leur département et les noms des bâtiments qui les hébergent



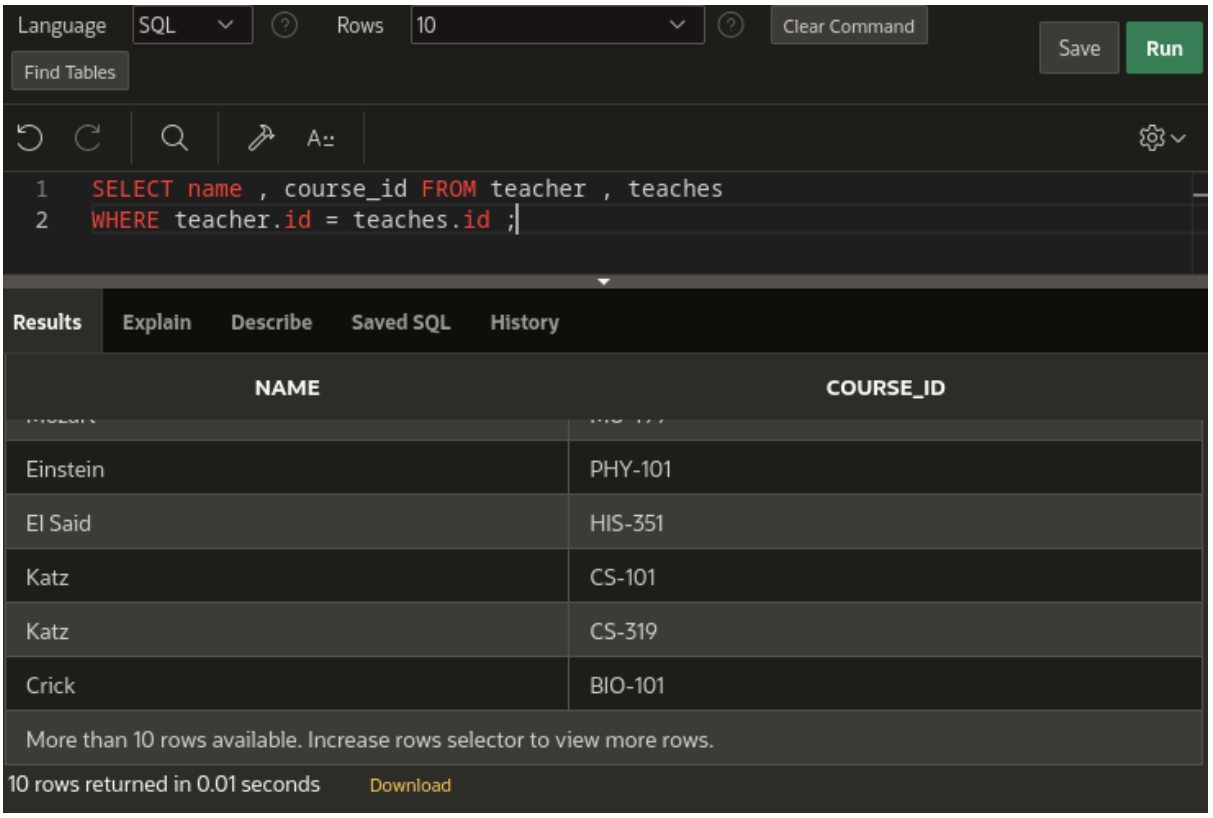
13. Afficher tous les étudiants ayant suivi au moins un cours en informatique.



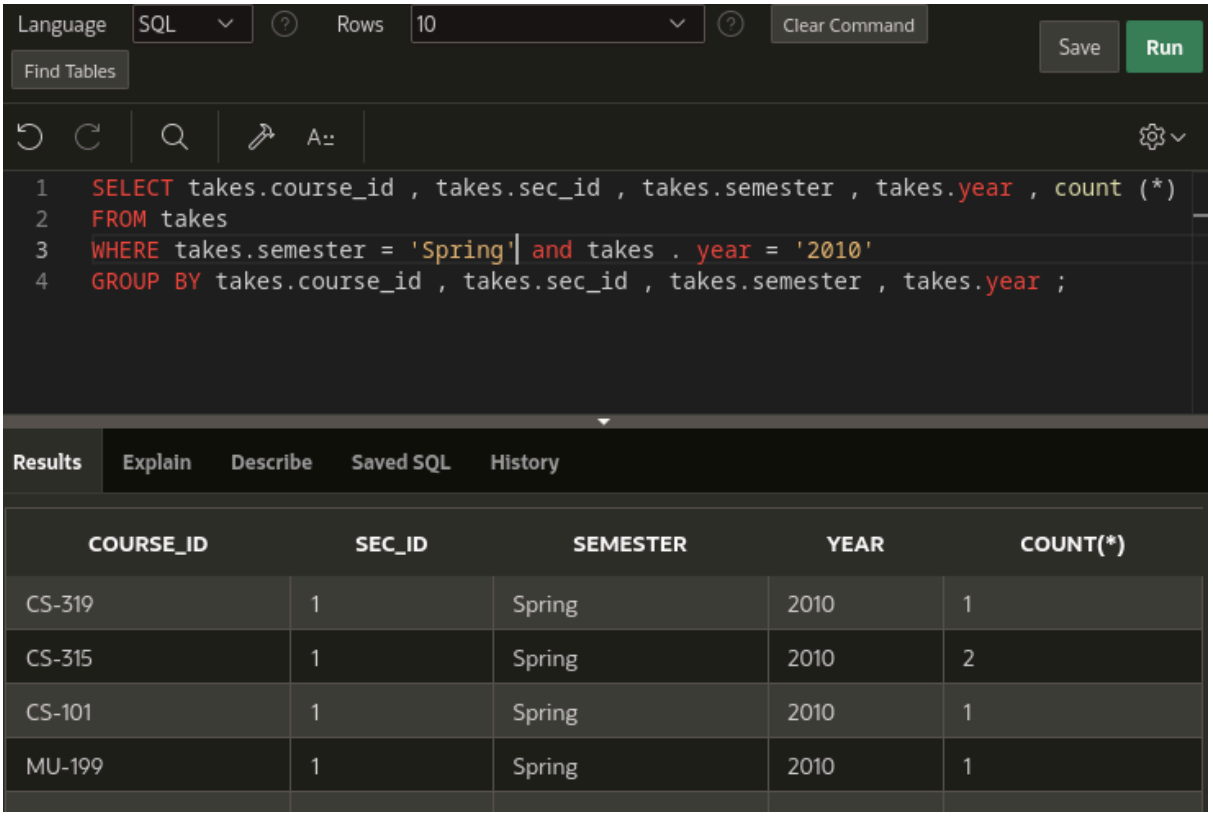
14. Afficher les noms des étudiants ayant suivi un cours dispensé par un enseignant nommé Einstein (éliminer les doublons).



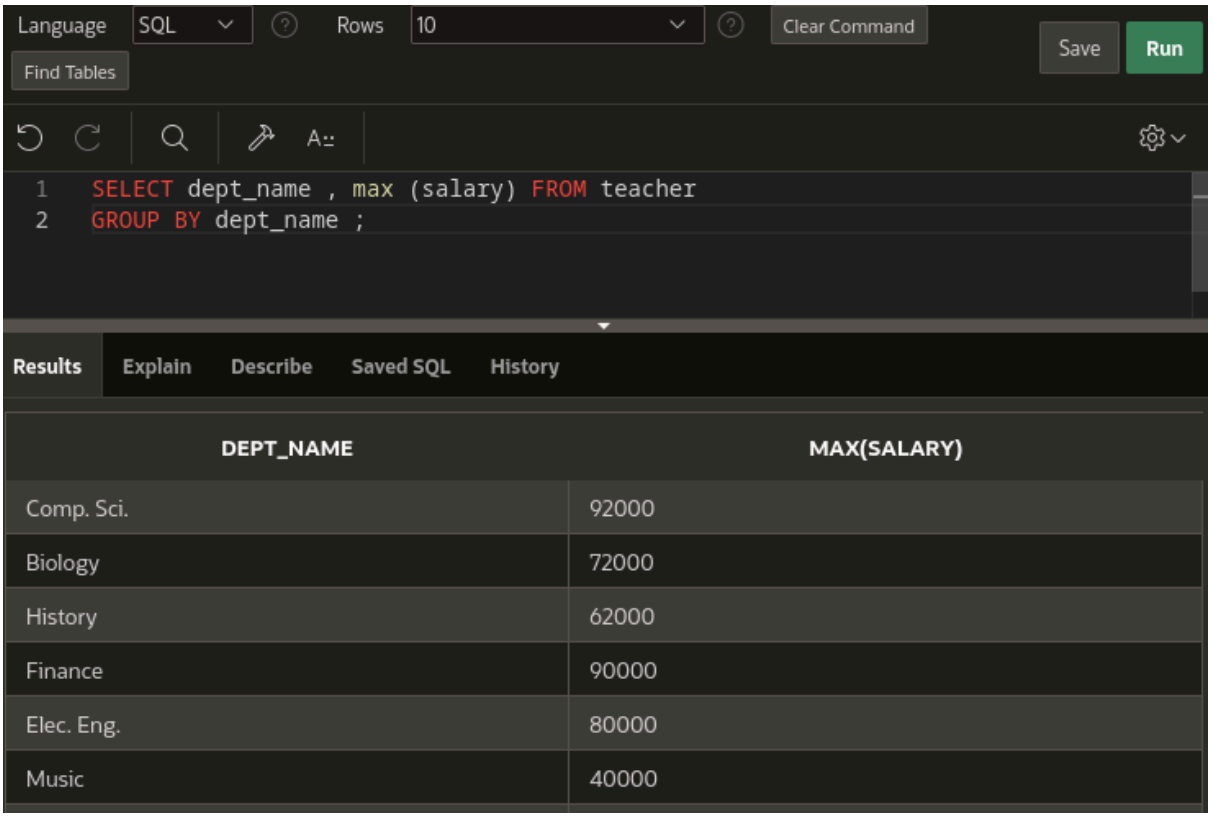
15. Afficher tous les identifiants des cours et les enseignants qui les ont assurés.



16. Afficher le nombre d'inscrits pour chaque enseignement proposé au printemps 2010 :

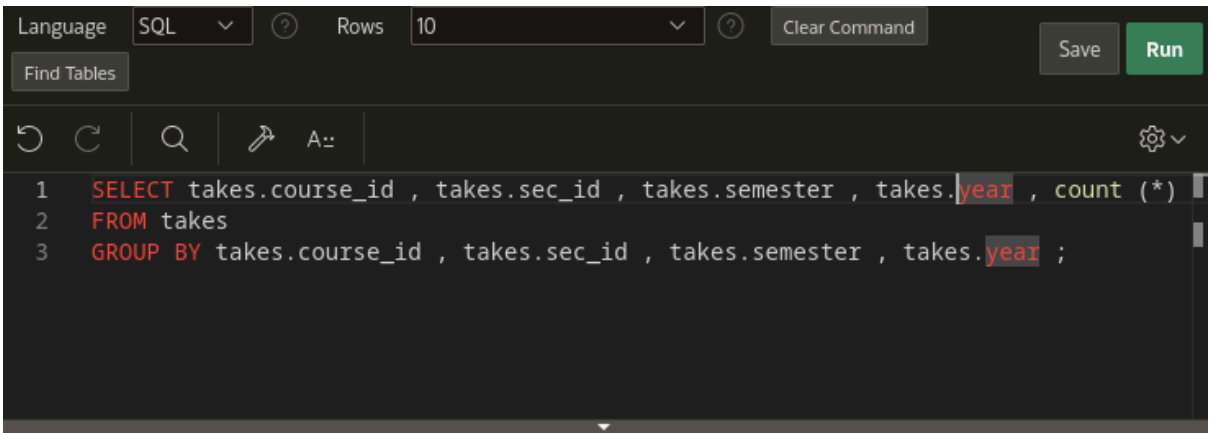


17. Afficher les noms des départements et les salaires maximum de leurs enseignants.



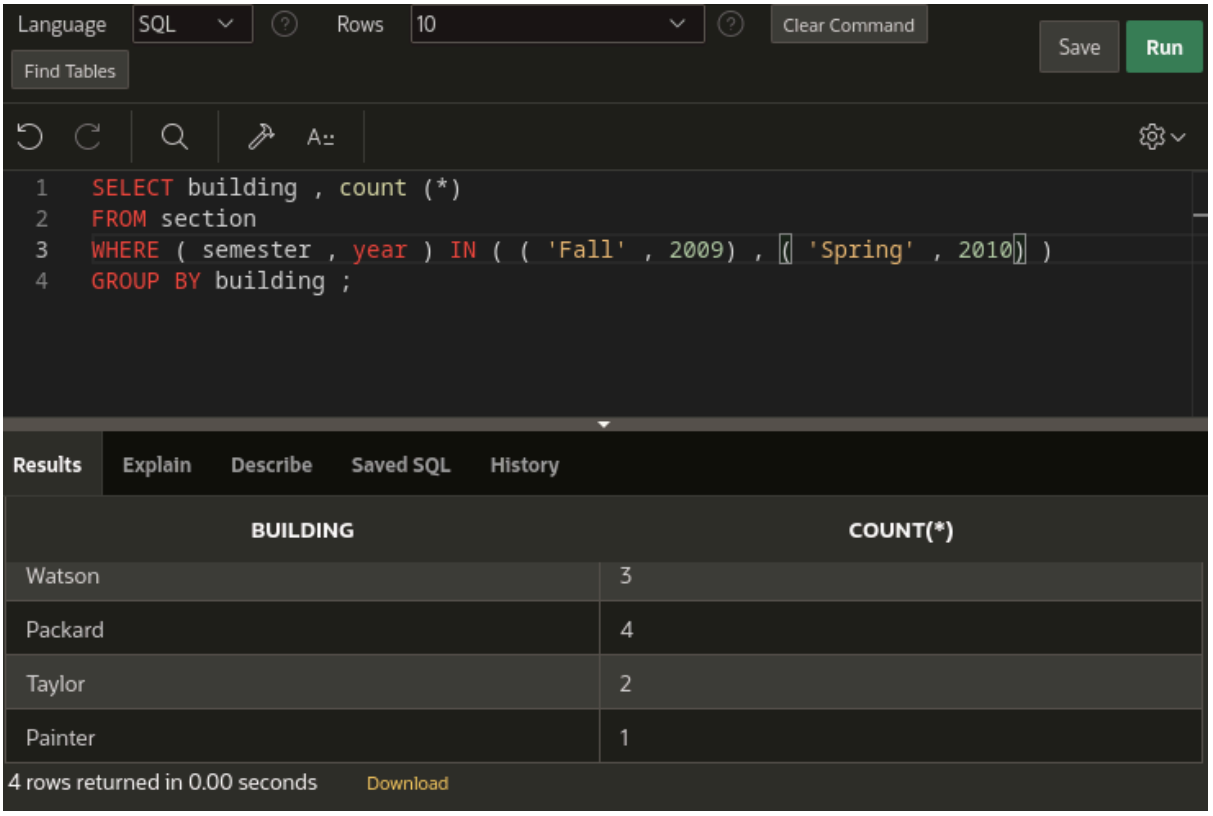
DEPT_NAME	MAX(SALARY)
Comp. Sci.	92000
Biology	72000
History	62000
Finance	90000
Elec. Eng.	80000
Music	40000

18. Afficher le nombre d'inscrits pour chaque enseignement proposé.

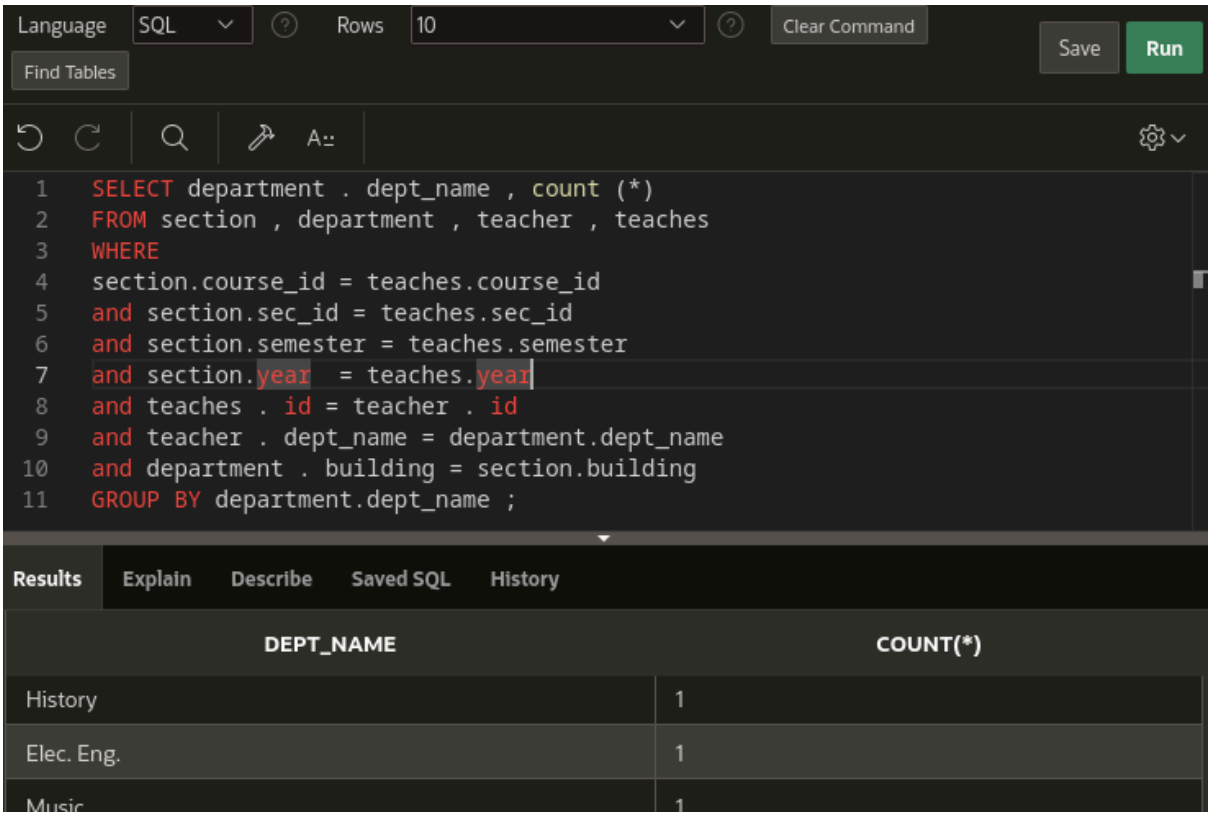


COURSE_ID	SEC_ID	SEMESTER	YEAR	COUNT(*)
CS-190	2	Spring	2009	2
CS-319	1	Spring	2010	1
CS-347	1	Fall	2009	2
MU-199	1	Spring	2010	1

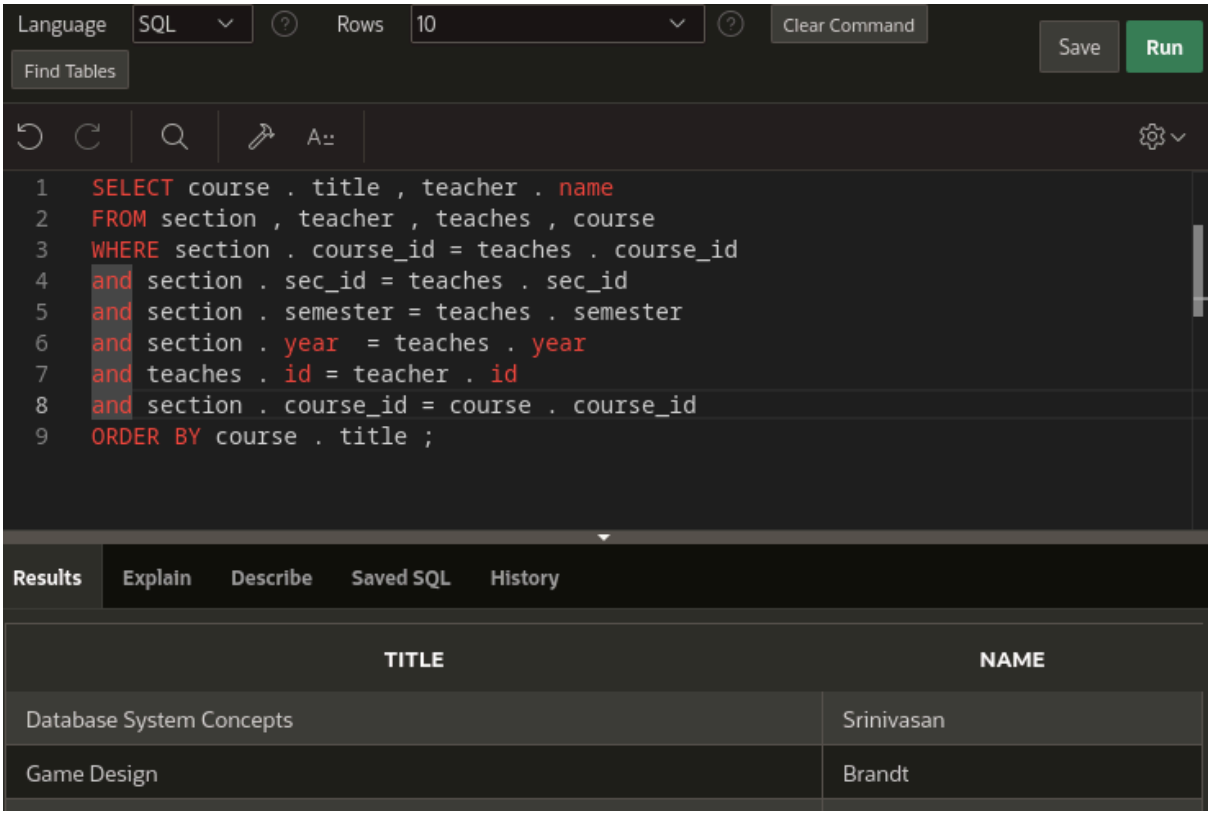
19. Afficher le nombre total de cours qui ont eu lieu dans chaque bâtiment, pendant l'automne 2009 et le printemps 2010



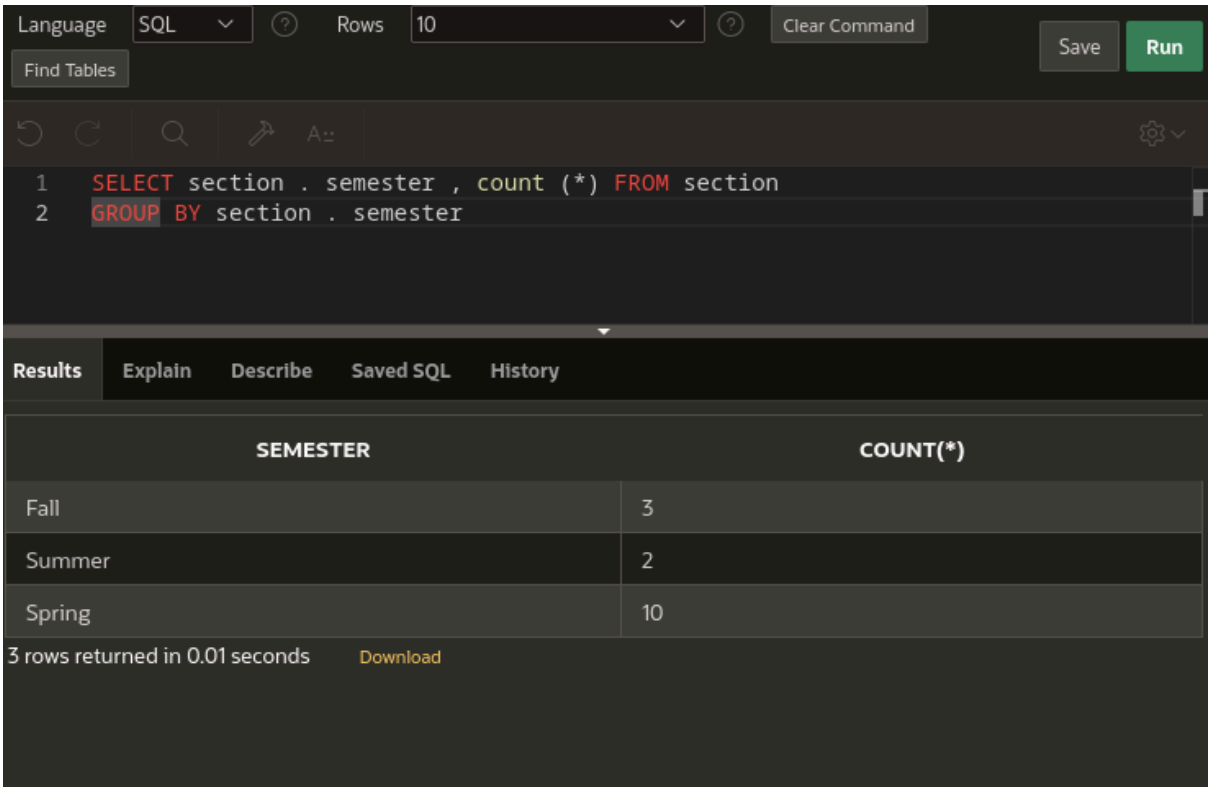
20. Afficher le nombre total de cours dispensés par chaque département et qui ont eu dans le même bâtiment qui l'abrite



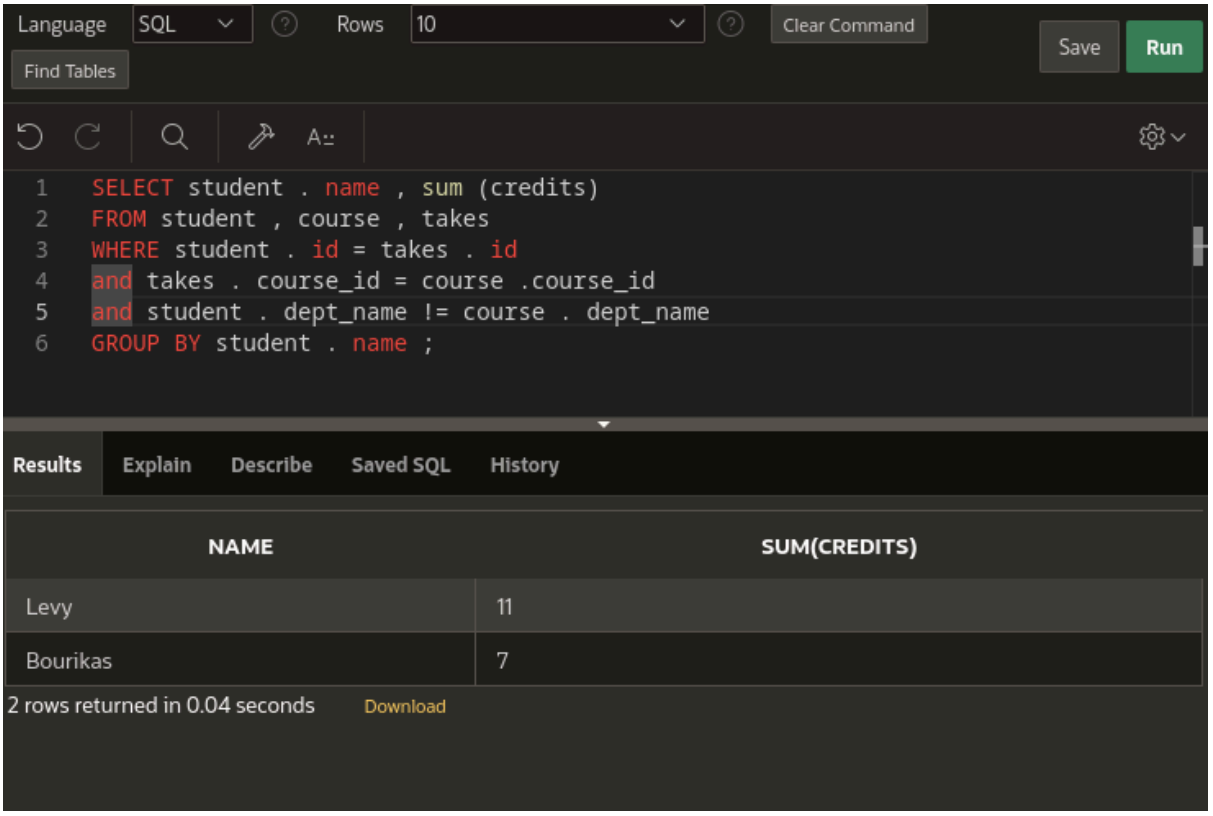
21. Afficher les titres des cours proposés et qui ont eu lieu et les enseignants qui les ont assurés.



22. Afficher le nombre total de cours qui ont eu lieu pour chacune des périodes Summer, Fall et Spring.



23. Afficher pour chaque étudiant le nombre total de crédits qu'il a obtenu, en suivant des cours qui n'ont pas été proposés par son département.



24. Pour chaque département, afficher le nombre total de crédits des cours qui ont eu lieu dans ce département.