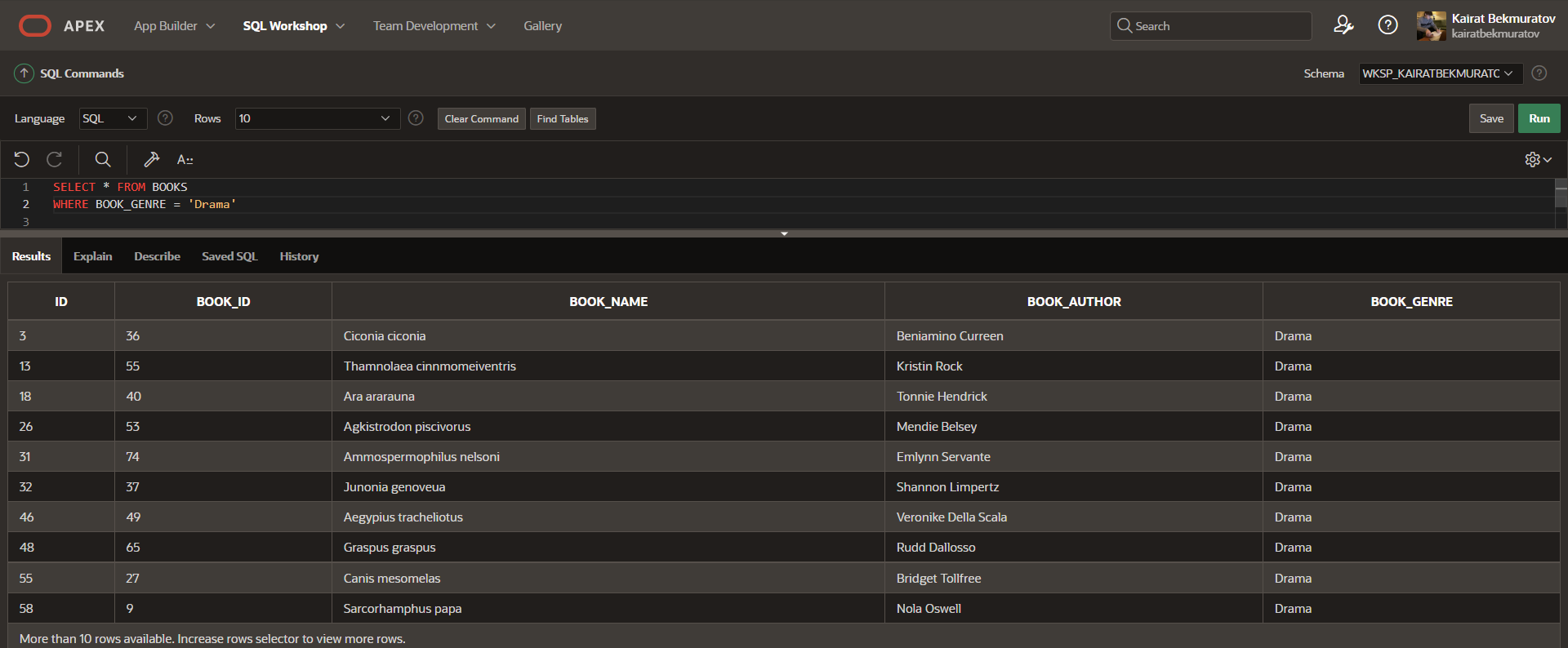
**WHERE**

-- SELECT \* FROM BOOKS

-- WHERE BOOK\_GENRE = 'Drama'

δ

BOOK\_GENRE=‘Drama’ (BOOKS)



**HAVING**

-- SELECT COUNT(DEPARTMENT\_ID)

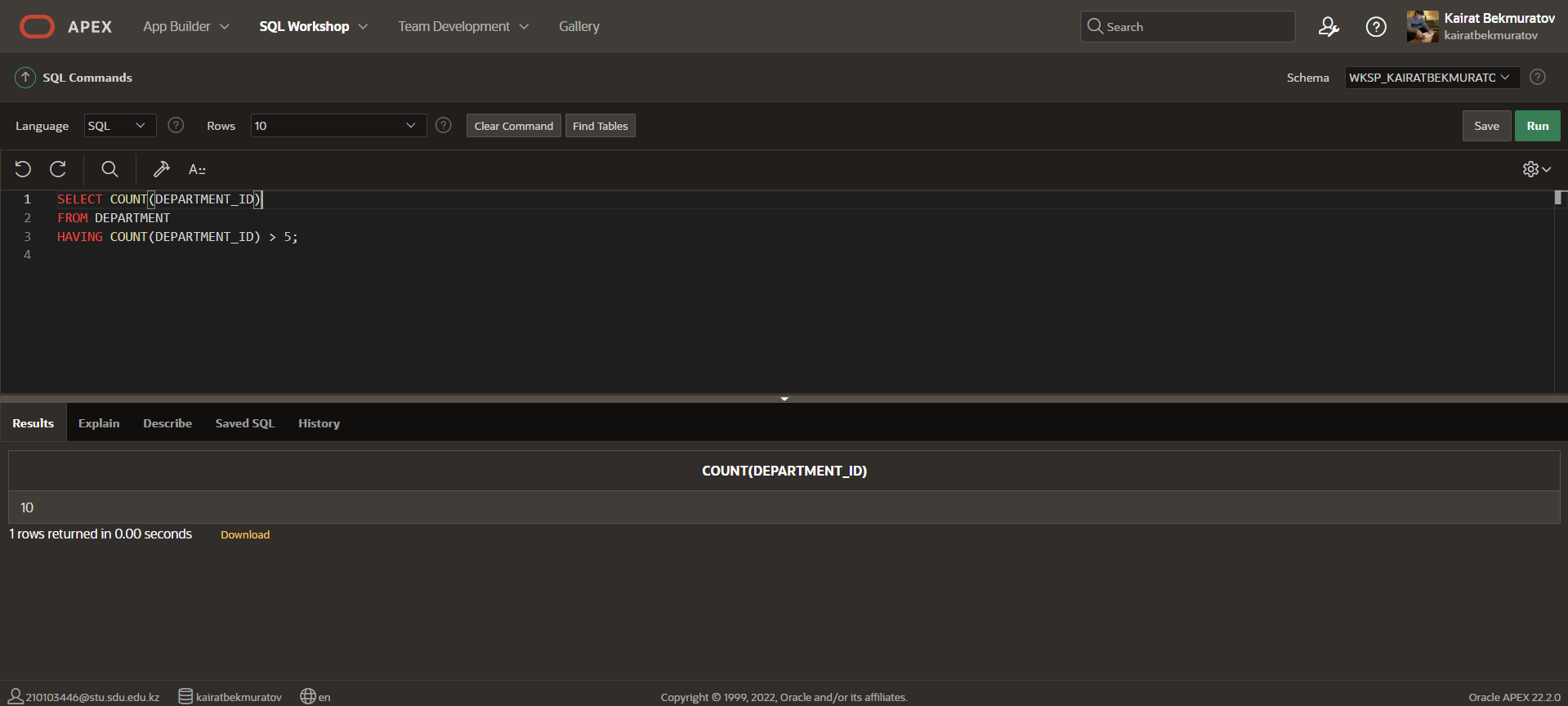
-- FROM DEPARTMENT

-- HAVING COUNT(DEPARTMENT\_ID) > 5;

π COUNT (department\_id)

σ COUNT (department\_id) > 5

γ COUNT (department\_id) (department)



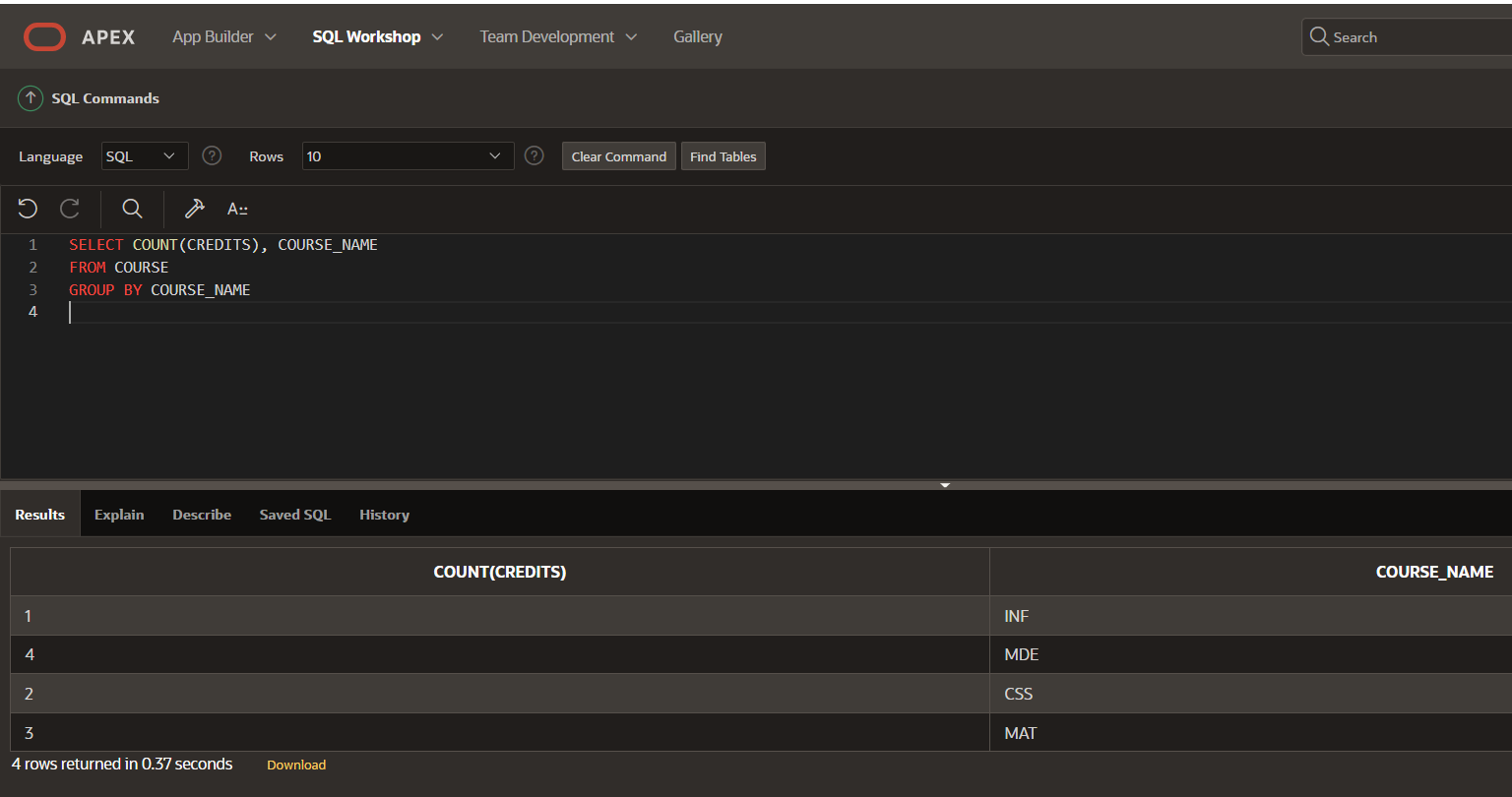
**GROUP BY**

-- SELECT COUNT(CREDITS), COURSE\_NAME

-- FROM COURSE

-- GROUP BY COURSE\_NAME

γ course\_name, COUNT (credits) (course)

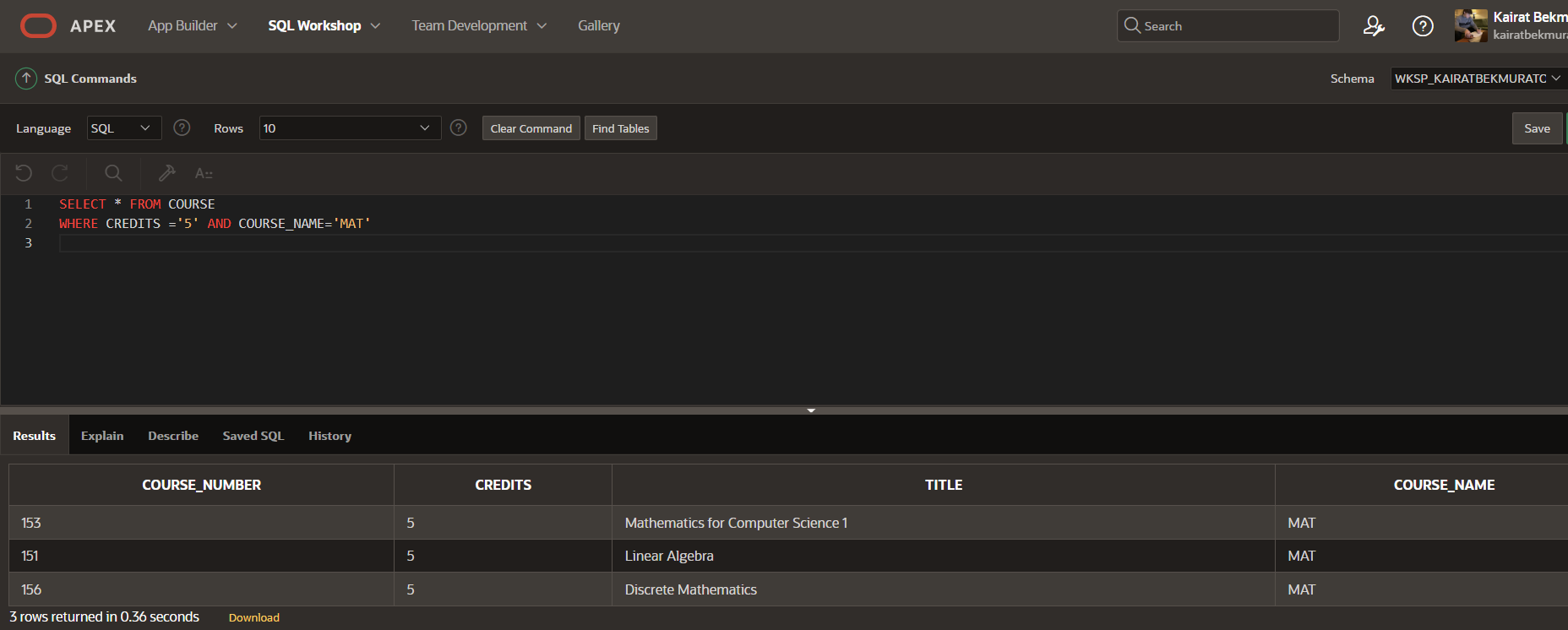


**AND**

-- SELECT \* FROM COURSE

-- WHERE CREDITS ='5' AND COURSE\_NAME='MAT'

σ credits = "5" AND course\_name = "MAT" (course)

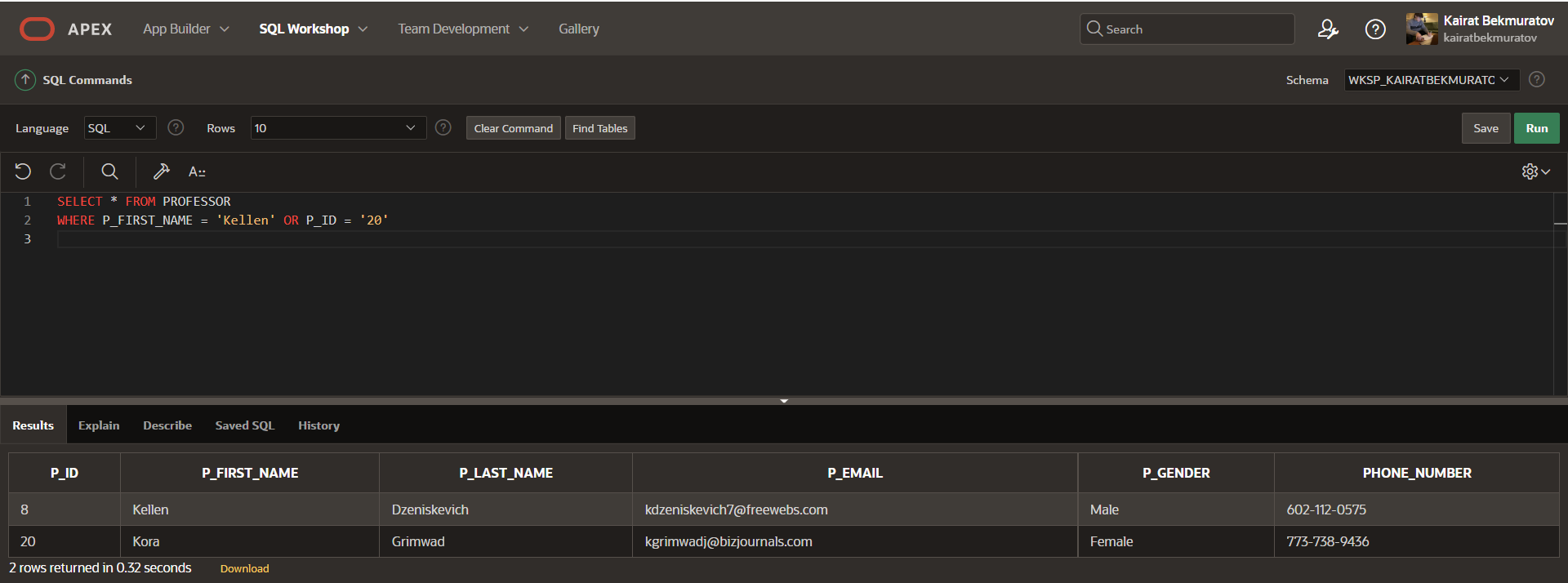


**OR**

-- SELECT \* FROM PROFESSOR

-- WHERE P\_FIRST\_NAME = 'Kellen' OR P\_ID = '20'

σ p\_first\_name = "Kellen" OR p\_id = "20" (professor)

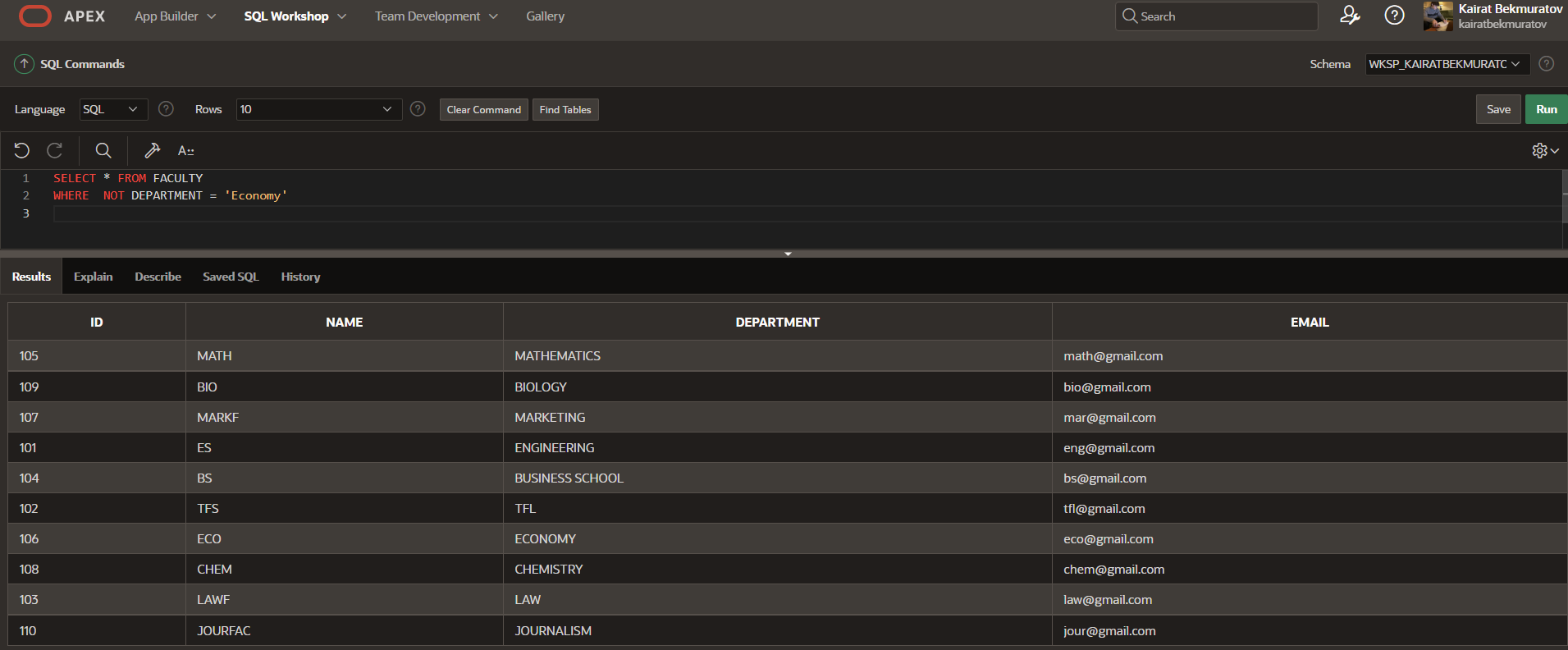


**NOT**

-- SELECT \* FROM FACULTY

-- WHERE NOT DEPARTMENT = 'Economy'

σ NOT (department = "Economy") (faculty)



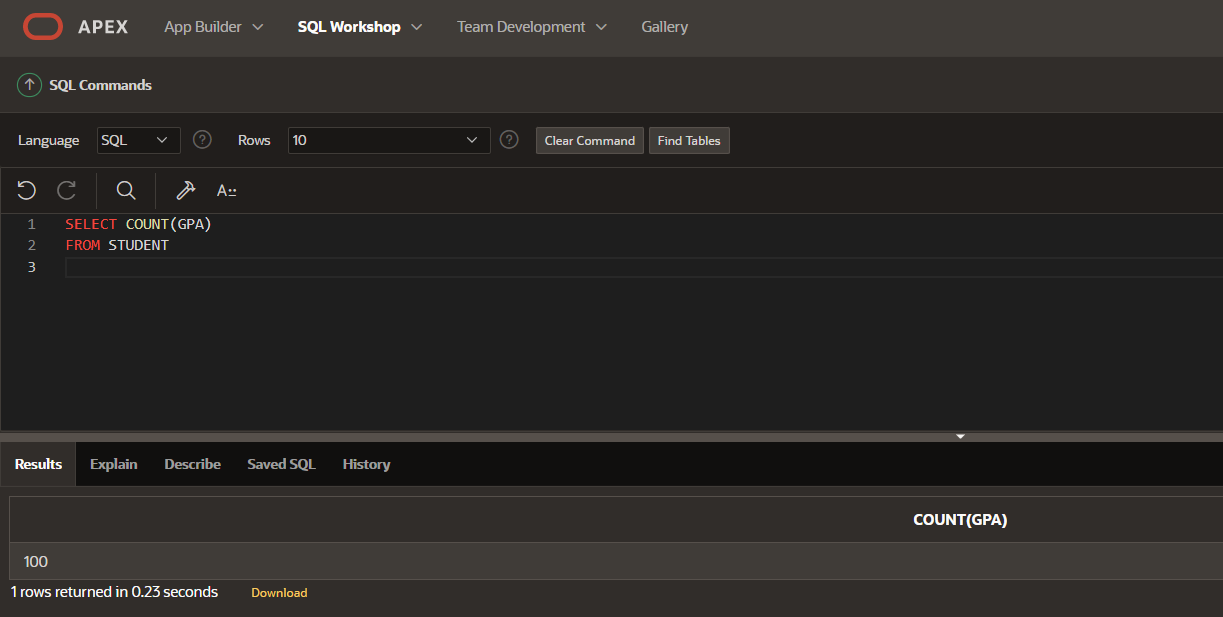
**COUNT**

-- SELECT COUNT(GPA)

-- FROM STUDENT

π COUNT (gpa)

γ COUNT (gpa) (student)



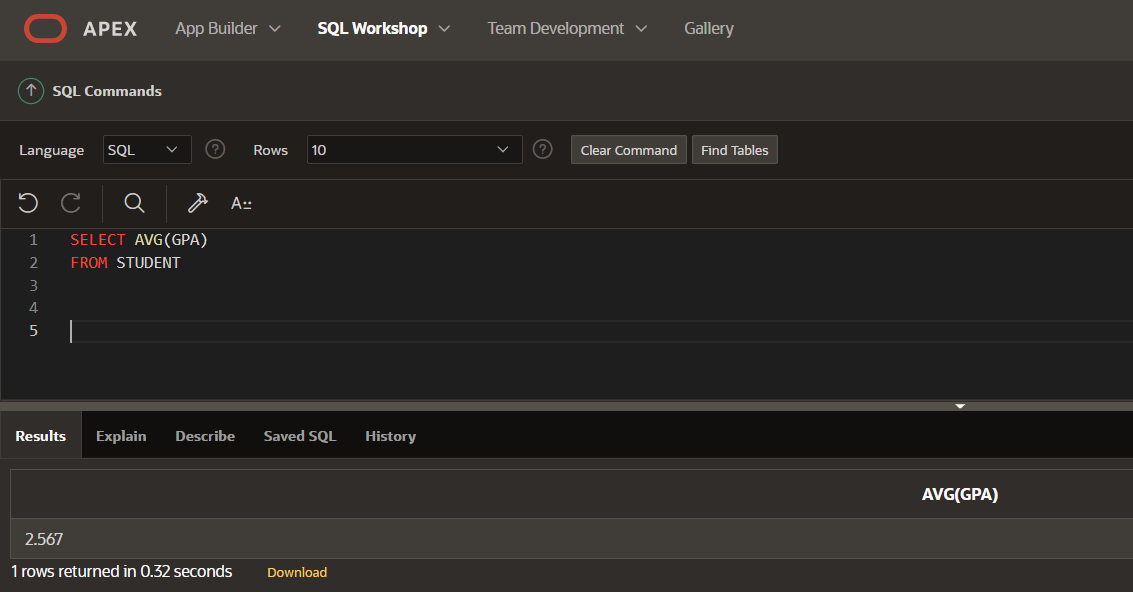
**AVG**

-- SELECT AVG(GPA)

-- FROM STUDENT

π AVG (gpa)

γ AVG (gpa) (student)



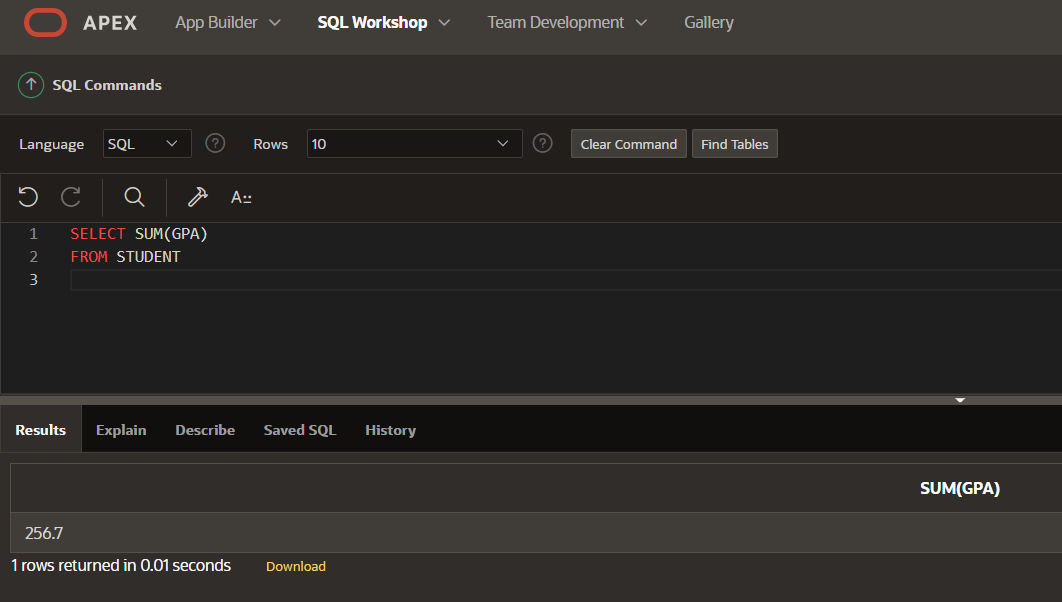
**SUM**

-- SELECT SUM(GPA)

-- FROM STUDENT

π SUM (gpa)

γ SUM (gpa) (student)

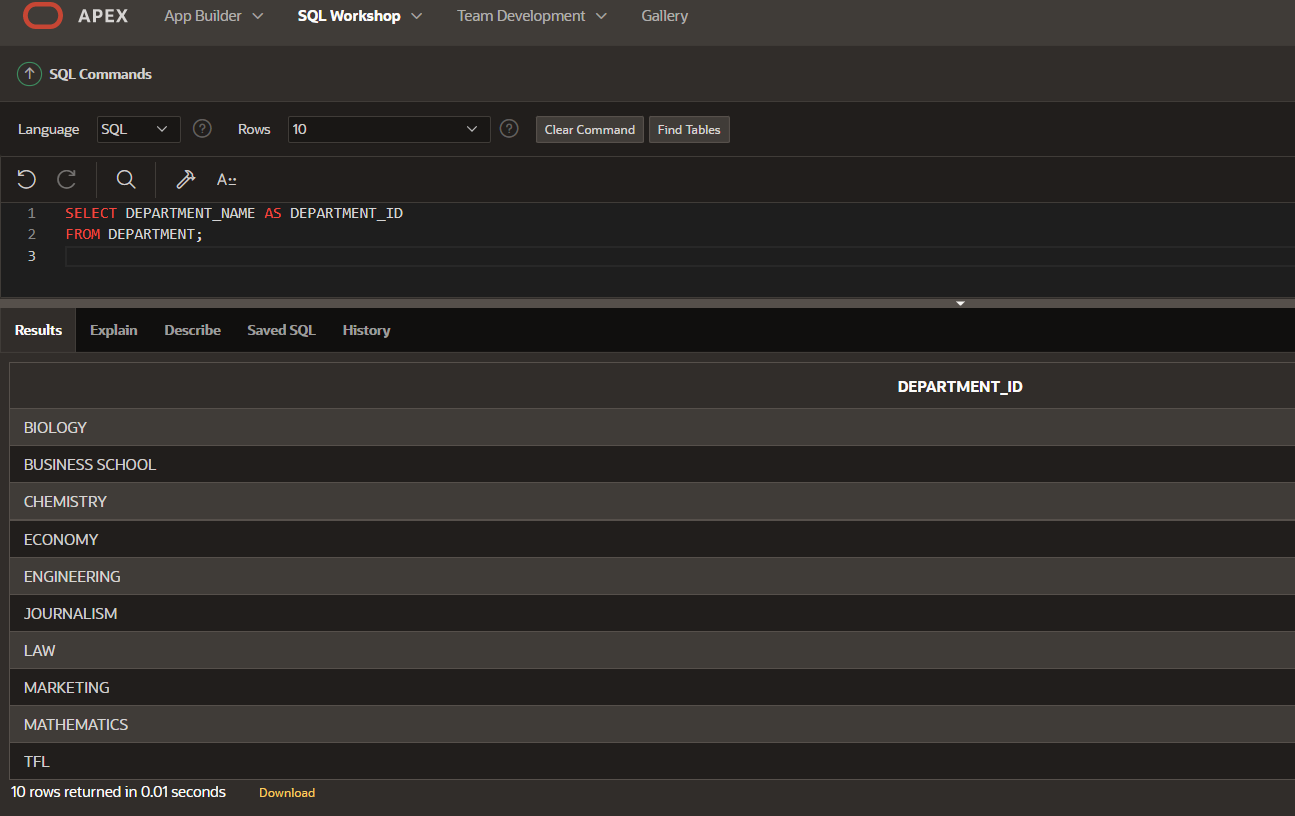


**AS(ALIASES)**

-- SELECT DEPARTMENT\_NAME AS DEPARTMENT\_ID

-- FROM DEPARTMENT;

π department\_name → department\_id (department)



**FULL JOIN**

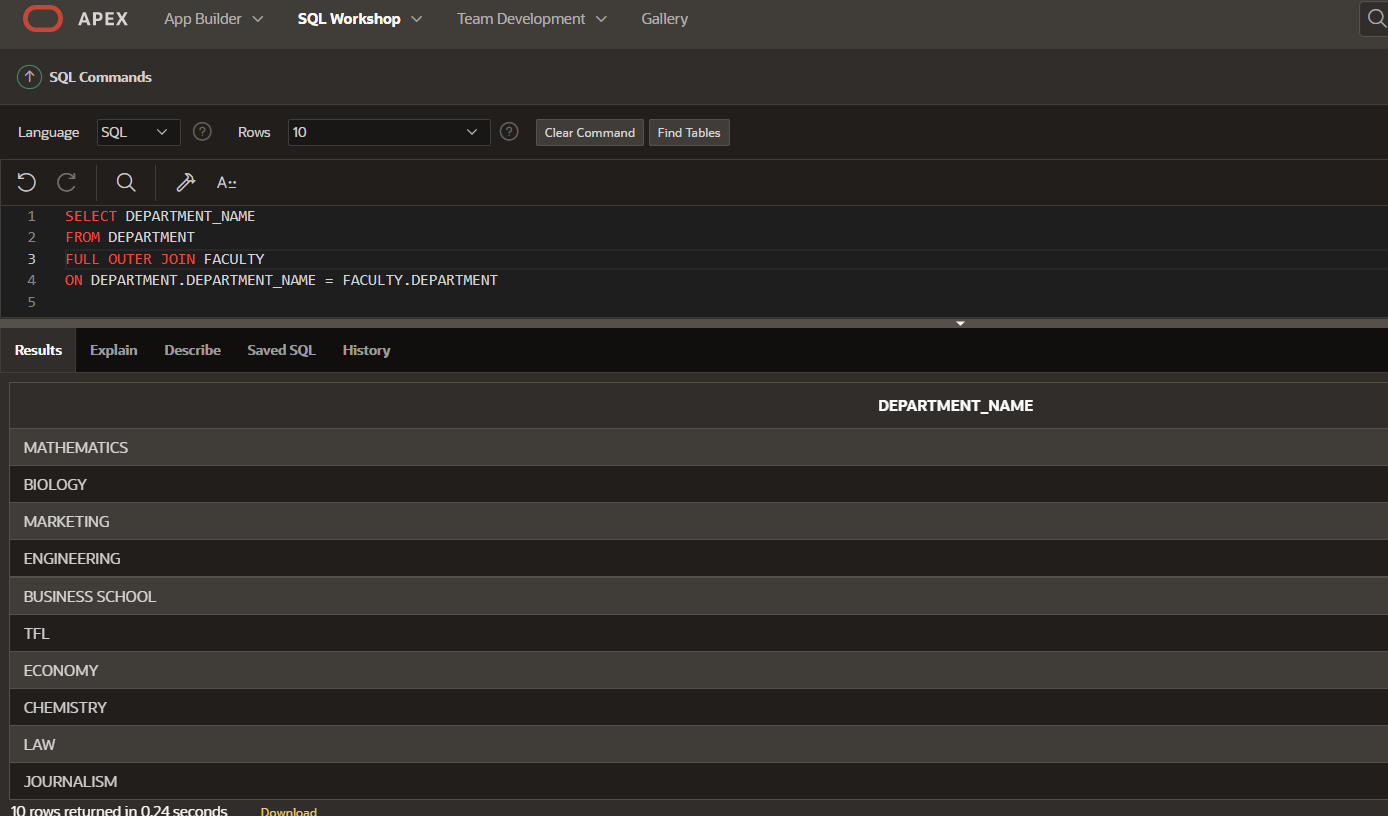
-- SELECT DEPARTMENT\_NAME

-- FROM DEPARTMENT

-- FULL OUTER JOIN FACULTY

-- ON DEPARTMENT.DEPARTMENT\_NAME = FACULTY.DEPARTMENT

π department\_name ((department) ⋈o department.department\_name = faculty.department (faculty))



**ORDER BY**

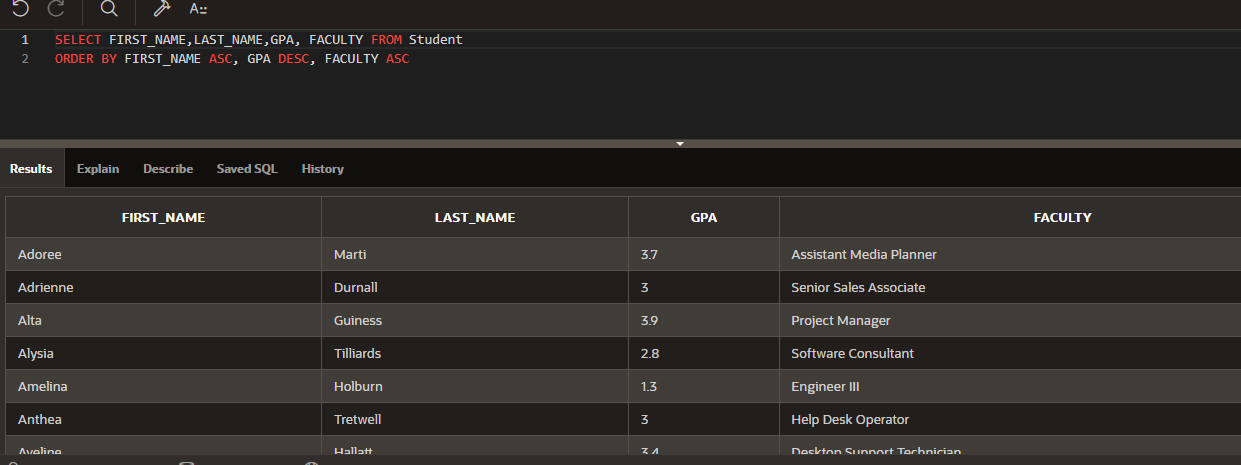
SELECT FIRST\_NAME,LAST\_NAME,GPA, FACULTY FROM Student

ORDER BY FIRST\_NAME ASC, GPA DESC, FACULTY ASC

τ first\_name, gpa↓, faculty

(Student)

π first\_name, last\_name, gpa, faculty

****

**LIKE**

SELECT \* FROM PROFESSOR

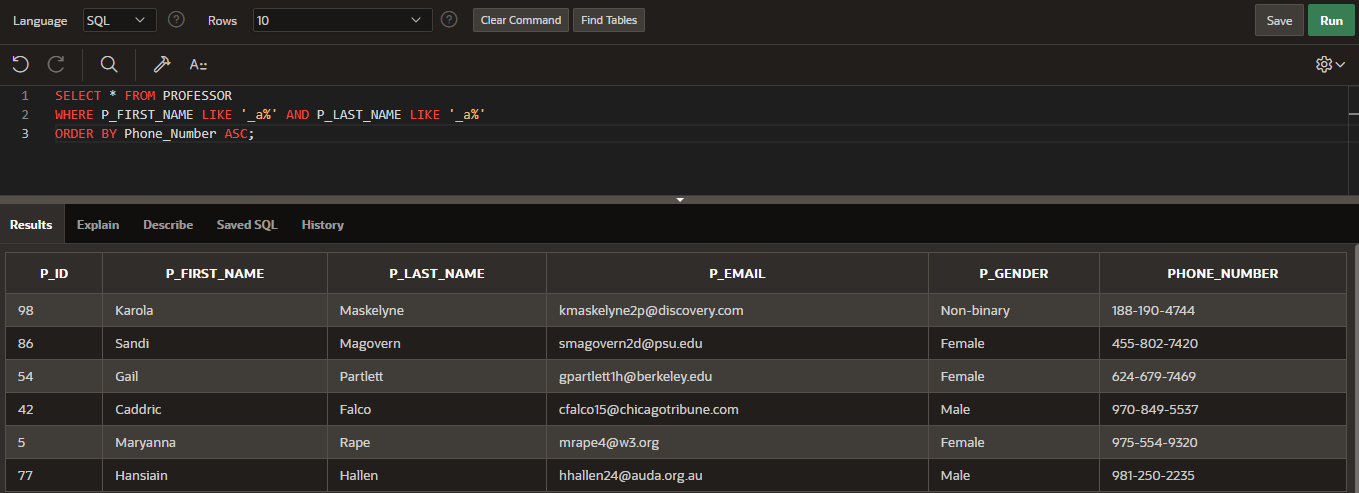
WHERE P\_FIRST\_NAME LIKE '\_a%' AND P\_LAST\_NAME LIKE '\_a%'

ORDER BY Phone\_Number ASC;

τ phone\_number

(professor)

σ p\_first\_name LIKE "\_a%" AND p\_last\_name LIKE "\_a%"



**IN**

SELECT FIRST\_NAME, LAST\_NAME from Hostel

WHERE AGE IN ('20', '21', '22') AND FIRST\_NAME LIKE '%a'

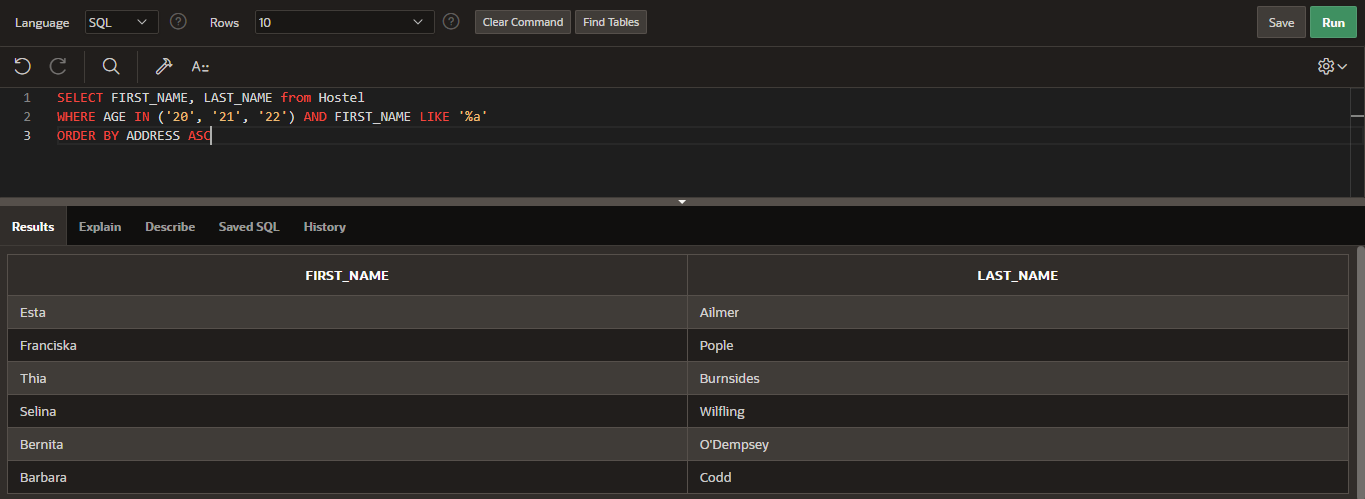
ORDER BY ADDRESS ASC

τ address

π first\_name, last\_name

(hostel)

σ age = "20" OR age = "21" OR age = "22" AND first\_name LIKE "%a"

****

**INNER JOIN**

SELECT OFFICE\_NAME, EXAMS.ROOM, EXAM\_CODE

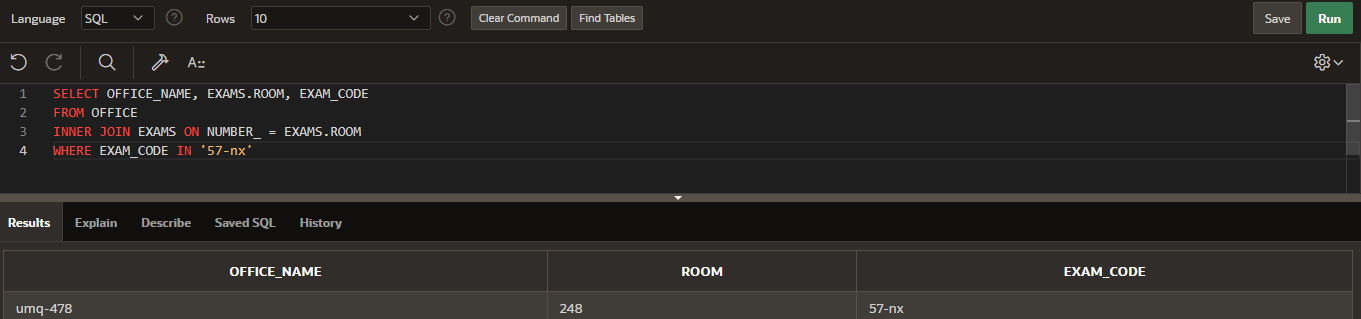
FROM OFFICE

INNER JOIN EXAMS ON NUMBER\_ = EXAMS.ROOM

WHERE EXAM\_CODE IN '57-nx'

π office\_name, exams . room, exam\_code

σ exam\_code = "57-nx" ((office) ⋈ number\_ = exams . room (exams))



**LEFT JOIN**

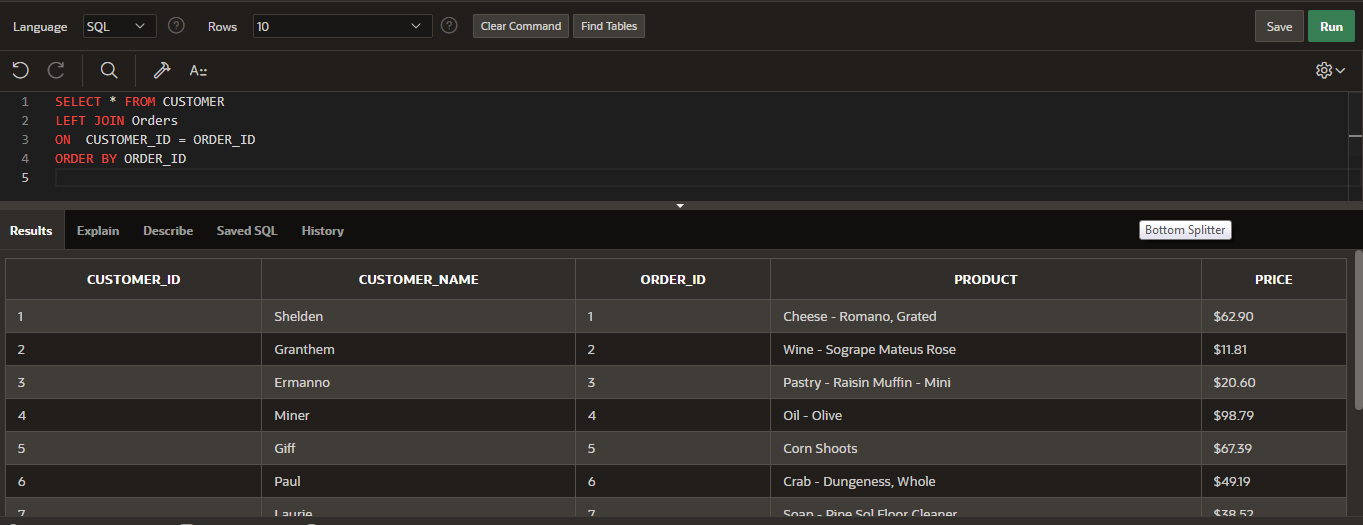
SELECT \* FROM CUSTOMER

LEFT JOIN Orders

ON CUSTOMER\_ID = ORDER\_ID

ORDER BY ORDER\_ID

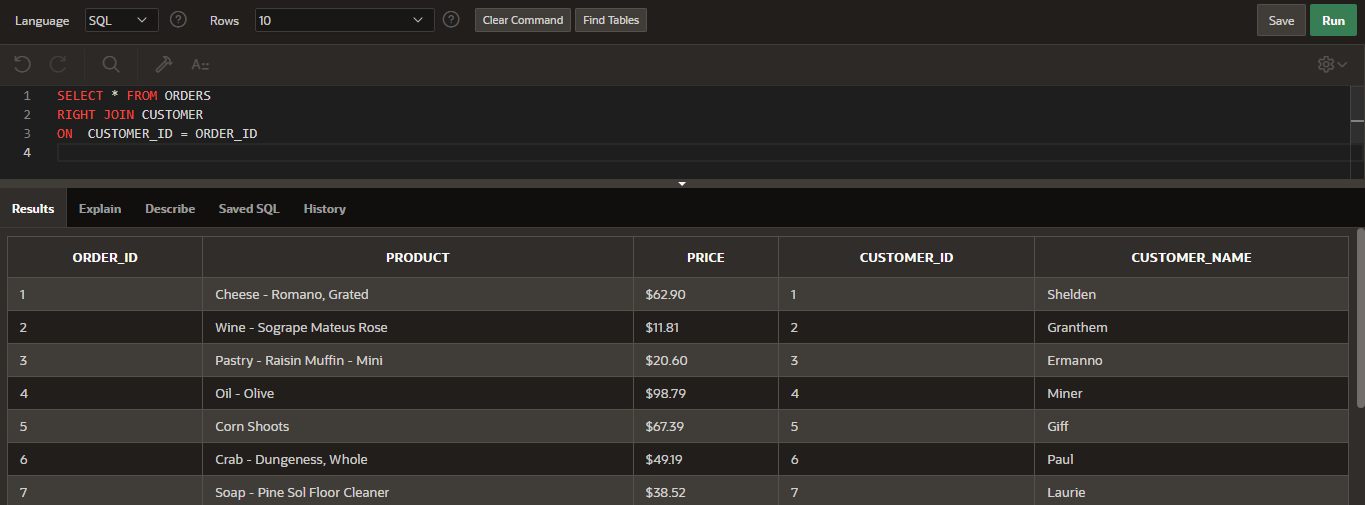
τ order\_id ((customer) ⟕ customers\_id = order\_id (orders))



**RIGHT JOIN**

SELECT \* FROM ORDERS  
RIGHT JOIN CUSTOMER  
ON CUSTOMER\_ID = ORDER\_ID

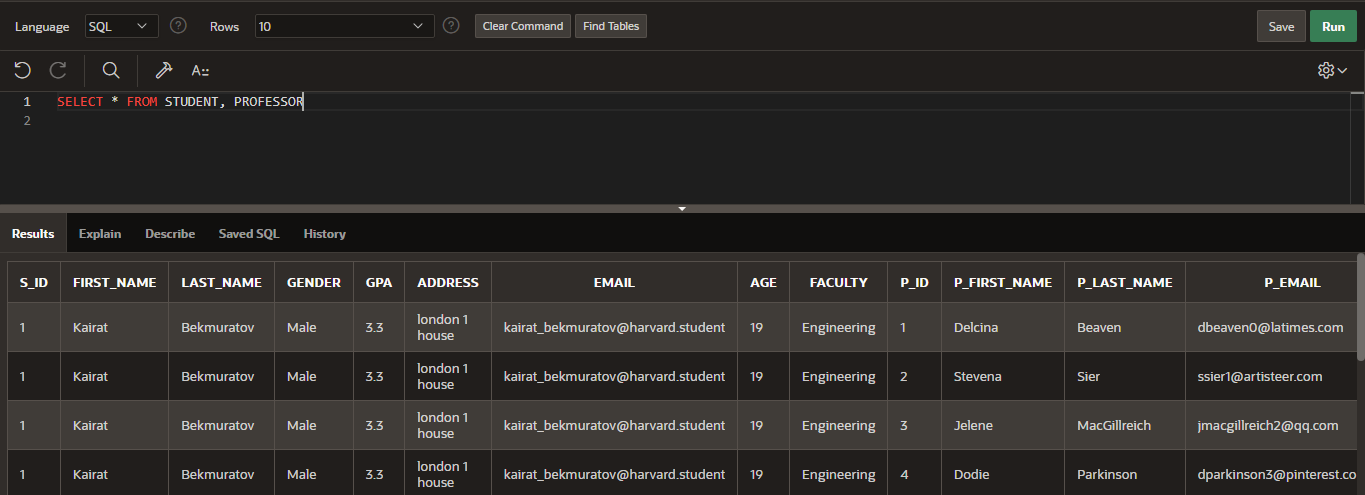
(orders) ⟖ customer\_id = order\_id (customer)



**SELECT**

SELECT \* FROM STUDENT, PROFESSOR

π (student × professor)



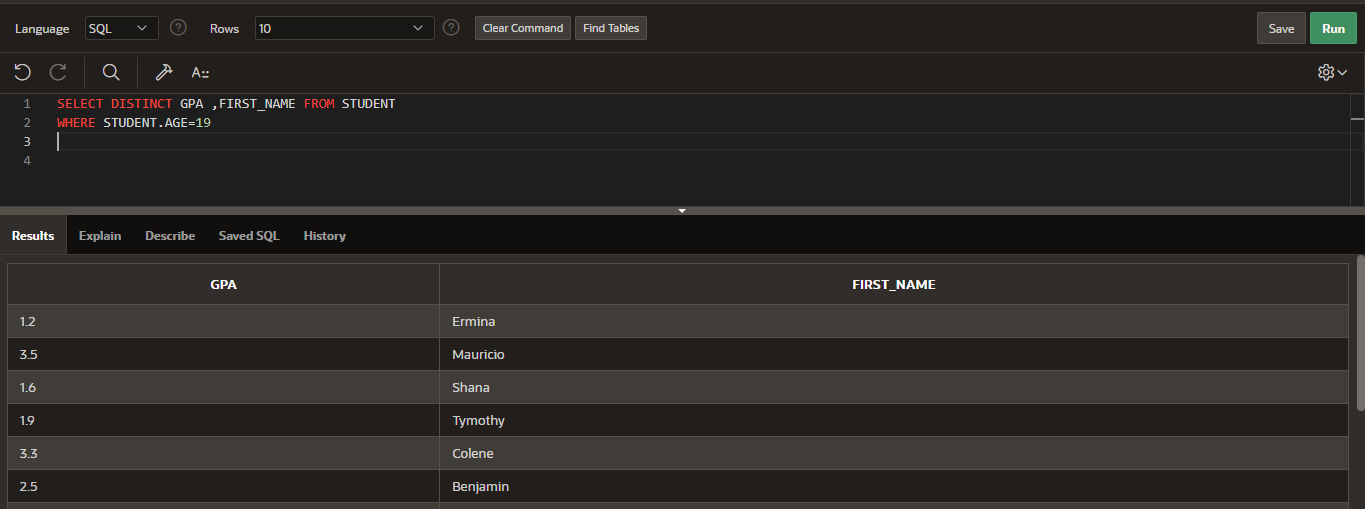
**DISTINCT**

SELECT DISTINCT GPA ,FIRST\_NAME FROM STUDENT  
WHERE STUDENT.AGE=19

δ

π gpa, first\_name

σ student.age = 19 (student)

****

**UNION**

SELECT First\_name, Last\_name

FROM Student

Where First\_name like 'a%'

UNION

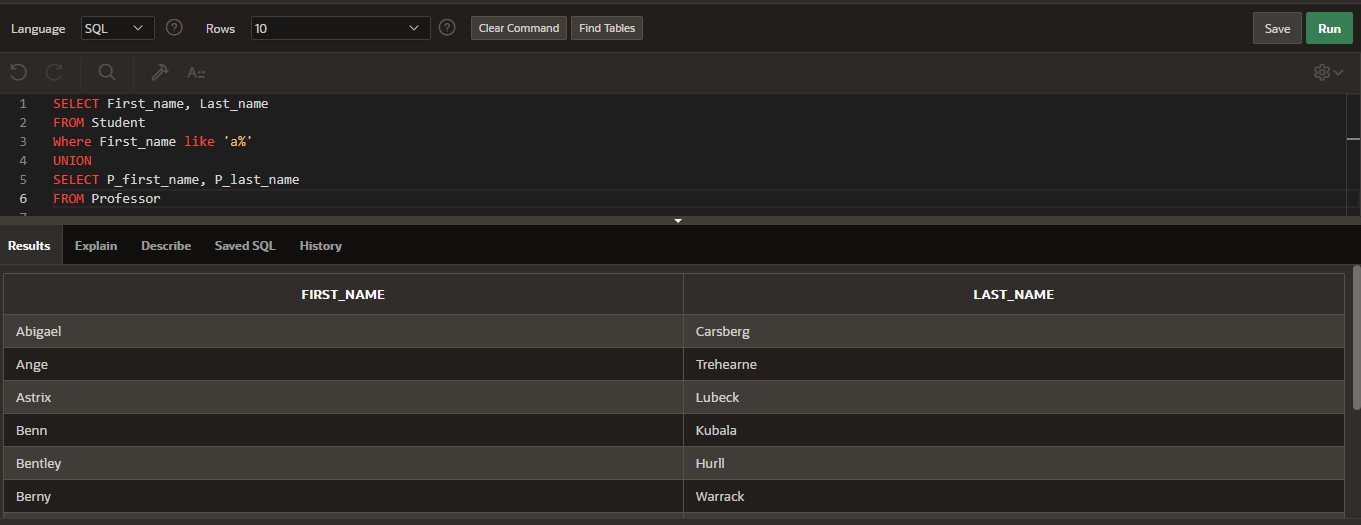
SELECT P\_first\_name, P\_last\_name

FROM Professor

π first\_name, last\_name

σ first\_name LIKE "a%" student ∪

π p\_first\_name, p\_last\_name (professor)

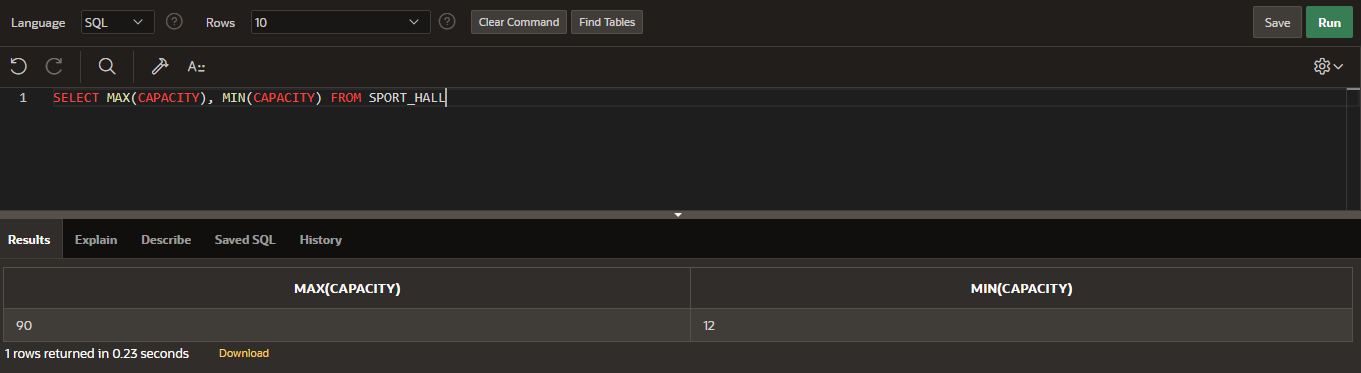
**

**MAX, MIN**

SELECT MAX(CAPACITY), MIN(CAPACITY) FROM SPORT\_HALL

π MAX (capacity), MIN (capacity)

γ MAX (capacity), MIN (capacity) (sport\_hall)



**TRIGGER 1**

create or replace trigger "FORKITCHEN"  
AFTER  
insert or update on "KITCHEN"  
for each row   
declare  
pragma autonomous\_transaction;  
begin  
if :new.product\_name = 'Dr.Pepper' then   
insert into kitchen(id,product\_code,product\_quantity,product\_price,product\_name,product\_type)   
values(:[new.id](http://new.id/),:new.product\_code,:new.product\_quantity,:new.product\_price-10;'Dr.Pepper',:new.product\_type);  
update kitchen  
set product\_price = :new.product\_price/2 where product\_name = 'Pepsi';  
commit;  
end if;  
end;