

FOR TABLES

-- DEPARTMENT TABLE

```
CREATE TABLE DEPARTMENT (  
    DEPARTMENT_ID NUMBER PRIMARY KEY,  
    DEPT_NAME     VARCHAR2(100) NOT NULL  
);
```

-- PROGRAM TABLE

```
CREATE TABLE PROGRAM (  
    PROGRAM_ID    NUMBER PRIMARY KEY,  
    PROGRAM_NAME  VARCHAR2(100) NOT NULL,  
    DEPARTMENT_ID NUMBER,  
    CONSTRAINT FK_PROGRAM_DEPARTMENT  
        FOREIGN KEY (DEPARTMENT_ID) REFERENCES DEPARTMENT(DEPARTMENT_ID)  
);
```

-- STUDENT TABLE

```
CREATE TABLE STUDENT (  
    STUDENT_ID    NUMBER PRIMARY KEY,  
    FNAME         VARCHAR2(50) NOT NULL,  
    LNAME         VARCHAR2(50) NOT NULL,  
    AGE           NUMBER(3) CHECK (AGE > 0),  
    FEE           NUMBER(10,2) CHECK (FEE >= 0),  
    PROGRAM_ID    NUMBER,  
    CONSTRAINT FK_STUDENT_PROGRAM  
        FOREIGN KEY (PROGRAM_ID) REFERENCES PROGRAM(PROGRAM_ID)  
);
```

-- FACULTY TABLE

```
CREATE TABLE FACULTY (  
    FACULTY_ID    NUMBER PRIMARY KEY,  
    NAME          VARCHAR2(100) NOT NULL,  
    JOB_TITLE     VARCHAR2(50),  
    SALARY        NUMBER(10,2) CHECK (SALARY >= 0),  
    DEPARTMENT_ID NUMBER,  
    CONSTRAINT FK_FACULTY_DEPARTMENT  
        FOREIGN KEY (DEPARTMENT_ID) REFERENCES DEPARTMENT(DEPARTMENT_ID)  
);
```

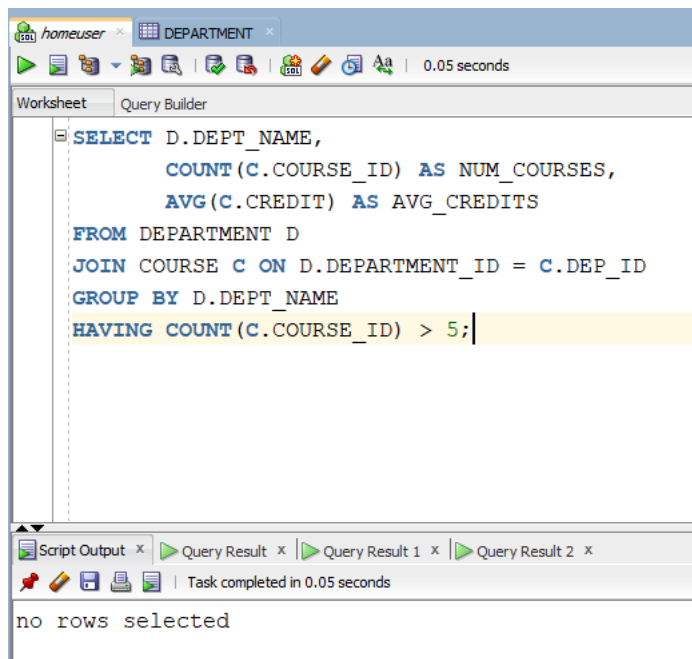
-- COURSE TABLE

```
CREATE TABLE COURSE (  
    COURSE_ID     NUMBER PRIMARY KEY,  
    COURSE_NAME   VARCHAR2(100) NOT NULL,  
    CREDIT        NUMBER(2) CHECK (CREDIT > 0),  
    DEP_ID        NUMBER,  
    FACULTY_ID    NUMBER,
```

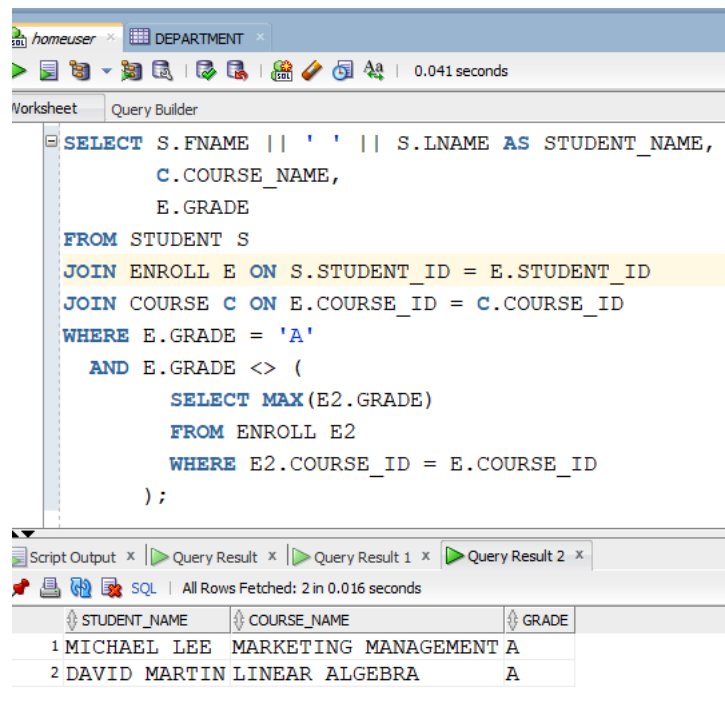
```
CONSTRAINT FK_COURSE_DEPARTMENT
    FOREIGN KEY (DEP_ID) REFERENCES DEPARTMENT(DEPARTMENT_ID),
CONSTRAINT FK_COURSE_FACULTY
    FOREIGN KEY (FACULTY_ID) REFERENCES FACULTY(FACULTY_ID)
);

-- ENROLL TABLE
CREATE TABLE ENROLL (
    ENROLL_ID    NUMBER PRIMARY KEY,
    STUDENT_ID   NUMBER,
    COURSE_ID    NUMBER,
    SEMESTER     VARCHAR2(20),
    GRADE        CHAR(2),
    CONSTRAINT FK_ENROLL_STUDENT
        FOREIGN KEY (STUDENT_ID) REFERENCES STUDENT(STUDENT_ID),
    CONSTRAINT FK_ENROLL_COURSE
        FOREIGN KEY (COURSE_ID) REFERENCES COURSE(COURSE_ID)
);
```

LAB#05 – TASK#01



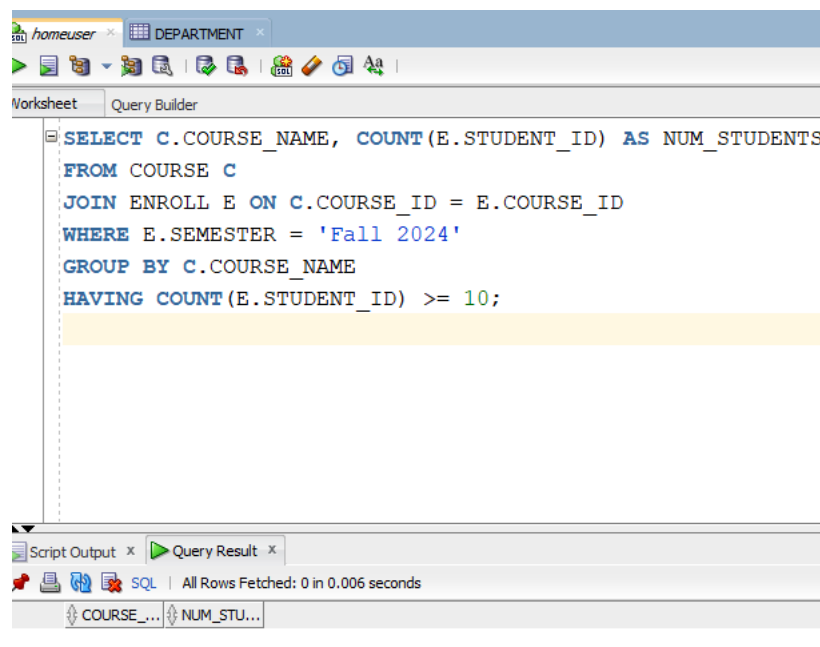
LAB#05 – TASK#02



```
SELECT S.FNAME || ' ' || S.LNAME AS STUDENT_NAME,
       C.COURSE_NAME,
       E.GRADE
FROM STUDENT S
JOIN ENROLL E ON S.STUDENT_ID = E.STUDENT_ID
JOIN COURSE C ON E.COURSE_ID = C.COURSE_ID
WHERE E.GRADE = 'A'
AND E.GRADE <> (
    SELECT MAX(E2.GRADE)
    FROM ENROLL E2
    WHERE E2.COURSE_ID = E.COURSE_ID
);
```

STUDENT_NAME	COURSE_NAME	GRADE
1 MICHAEL LEE	MARKETING MANAGEMENT	A
2 DAVID MARTIN	LINEAR ALGEBRA	A

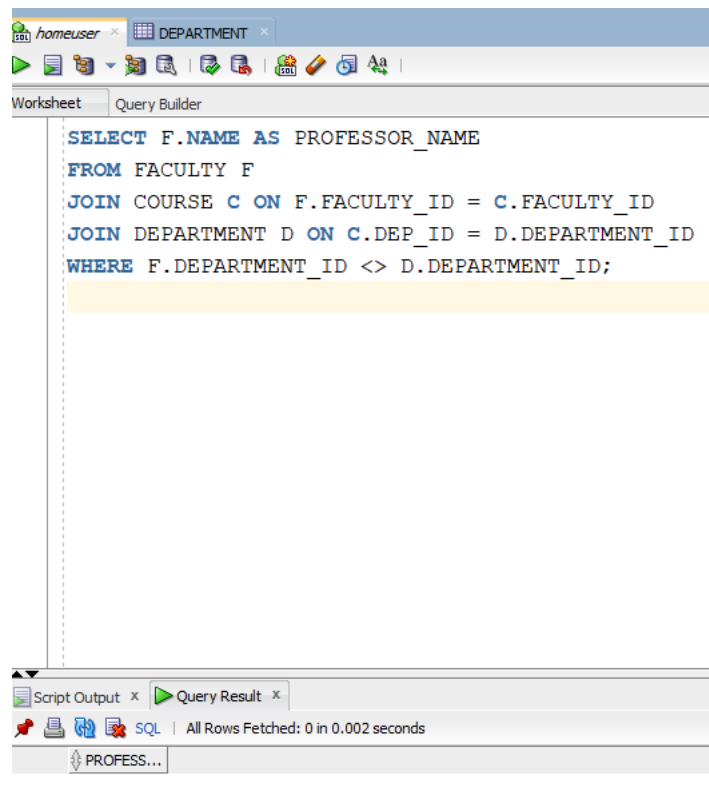
LAB#05 – TASK#03



```
SELECT C.COURSE_NAME, COUNT(E.STUDENT_ID) AS NUM_STUDENTS
FROM COURSE C
JOIN ENROLL E ON C.COURSE_ID = E.COURSE_ID
WHERE E.SEMESTER = 'Fall 2024'
GROUP BY C.COURSE_NAME
HAVING COUNT(E.STUDENT_ID) >= 10;
```

COURSE_NAME	NUM_STUDENTS
MARKETING MANAGEMENT	10
LINEAR ALGEBRA	10

LAB#05 – TASK#04

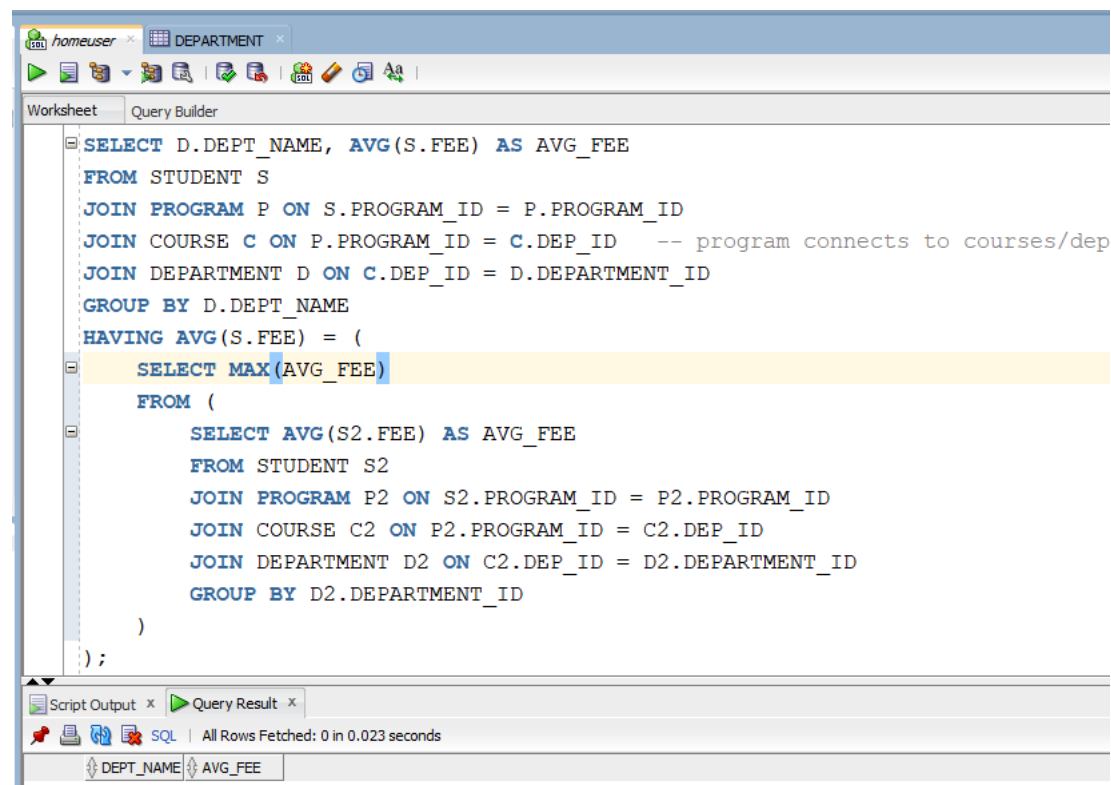


The screenshot shows the SQL Developer interface with a query in the Query Builder. The query is as follows:

```
SELECT F.NAME AS PROFESSOR_NAME
FROM FACULTY F
JOIN COURSE C ON F.FACULTY_ID = C.FACULTY_ID
JOIN DEPARTMENT D ON C.DEP_ID = D.DEPARTMENT_ID
WHERE F.DEPARTMENT_ID <> D.DEPARTMENT_ID;
```

The status bar at the bottom indicates "All Rows Fetched: 0 in 0.002 seconds".

LAB#05 – TASK#05

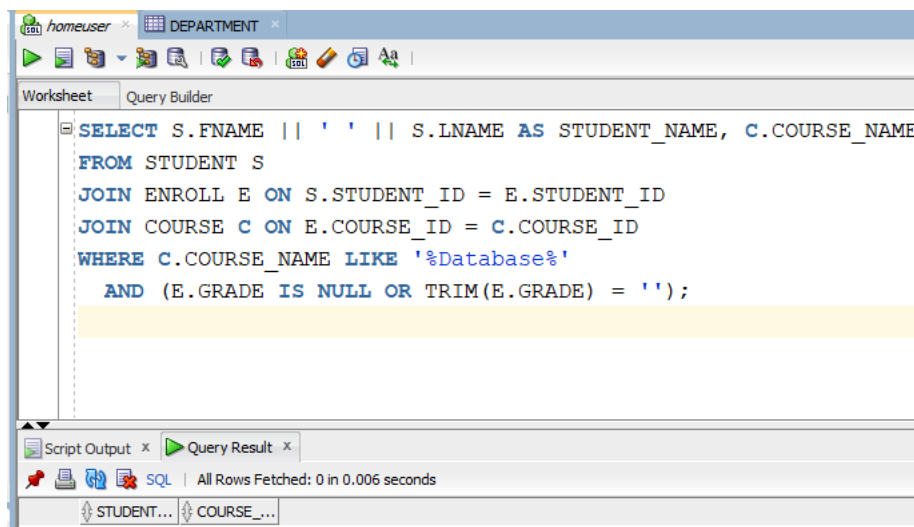


The screenshot shows the SQL Developer interface with a complex query in the Query Builder. The query is as follows:

```
SELECT D.DEPT_NAME, AVG(S.FEE) AS AVG_FEE
FROM STUDENT S
JOIN PROGRAM P ON S.PROGRAM_ID = P.PROGRAM_ID
JOIN COURSE C ON P.PROGRAM_ID = C.DEP_ID -- program connects to courses/dep
JOIN DEPARTMENT D ON C.DEP_ID = D.DEPARTMENT_ID
GROUP BY D.DEPT_NAME
HAVING AVG(S.FEE) = (
    SELECT MAX(AVG_FEE)
    FROM (
        SELECT AVG(S2.FEE) AS AVG_FEE
        FROM STUDENT S2
        JOIN PROGRAM P2 ON S2.PROGRAM_ID = P2.PROGRAM_ID
        JOIN COURSE C2 ON P2.PROGRAM_ID = C2.DEP_ID
        JOIN DEPARTMENT D2 ON C2.DEP_ID = D2.DEPARTMENT_ID
        GROUP BY D2.DEPARTMENT_ID
    )
);
```

The status bar at the bottom indicates "All Rows Fetched: 0 in 0.023 seconds".

LAB#05 – TASK#06

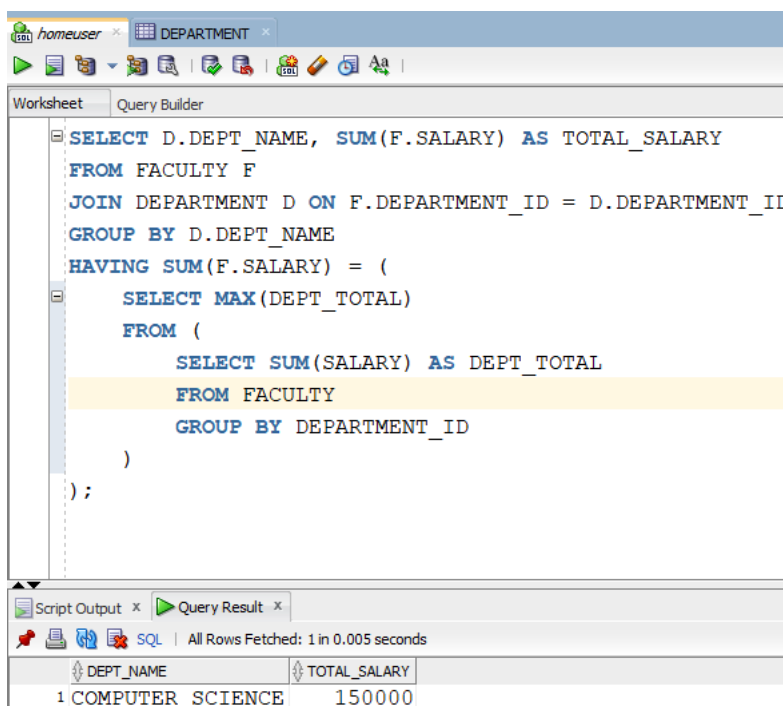


The screenshot shows the SQL Developer interface with the Query Builder tab active. The query is as follows:

```
SELECT S.FNAME || ' ' || S.LNAME AS STUDENT_NAME, C.COURSE_NAME
FROM STUDENT S
JOIN ENROLL E ON S.STUDENT_ID = E.STUDENT_ID
JOIN COURSE C ON E.COURSE_ID = C.COURSE_ID
WHERE C.COURSE_NAME LIKE '%Database%'
AND (E.GRADE IS NULL OR TRIM(E.GRADE) = '');
```

The status bar at the bottom indicates "All Rows Fetched: 0 in 0.006 seconds".

LAB#05 – TASK#07



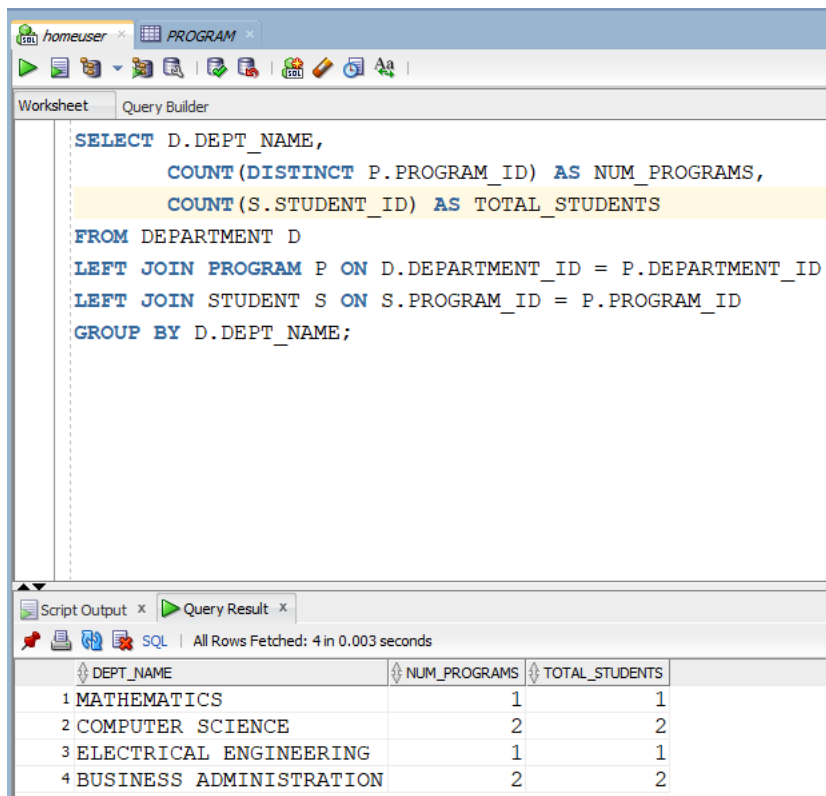
The screenshot shows the SQL Developer interface with the Query Builder tab active. The query is as follows:

```
SELECT D.DEPT_NAME, SUM(F.SALARY) AS TOTAL_SALARY
FROM FACULTY F
JOIN DEPARTMENT D ON F.DEPARTMENT_ID = D.DEPARTMENT_ID
GROUP BY D.DEPT_NAME
HAVING SUM(F.SALARY) = (
    SELECT MAX(DEPT_TOTAL)
    FROM (
        SELECT SUM(SALARY) AS DEPT_TOTAL
        FROM FACULTY
        GROUP BY DEPARTMENT_ID
    )
);
```

The status bar at the bottom indicates "All Rows Fetched: 1 in 0.005 seconds".

DEPT_NAME	TOTAL_SALARY
1 COMPUTER SCIENCE	150000

LAB#05 – TASK#08



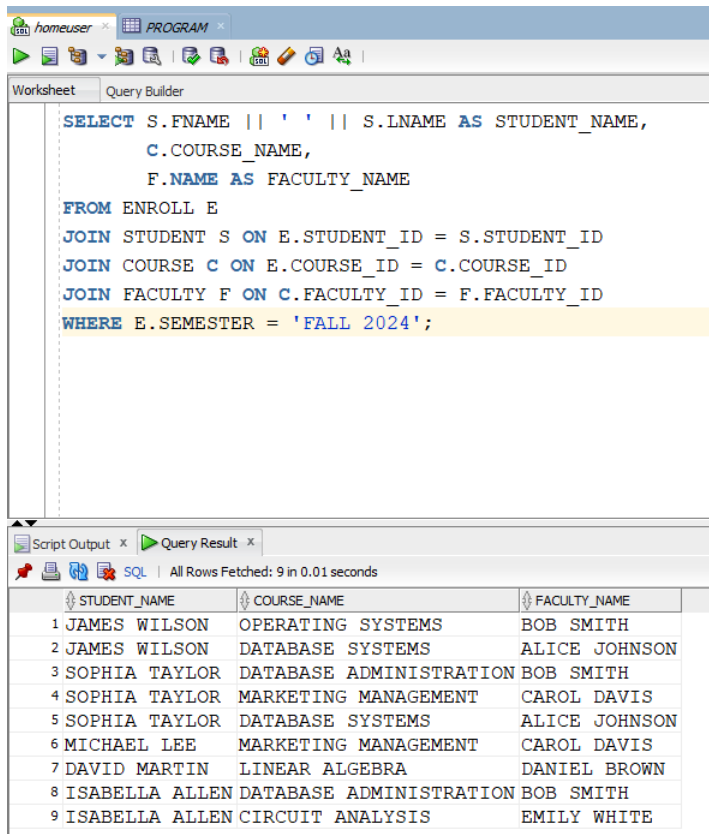
The screenshot shows the SQL Developer interface. The top toolbar includes icons for running queries, saving, and other standard database operations. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT D.DEPT_NAME,  
       COUNT(DISTINCT P.PROGRAM_ID) AS NUM_PROGRAMS,  
       COUNT(S.STUDENT_ID) AS TOTAL_STUDENTS  
FROM DEPARTMENT D  
LEFT JOIN PROGRAM P ON D.DEPARTMENT_ID = P.DEPARTMENT_ID  
LEFT JOIN STUDENT S ON S.PROGRAM_ID = P.PROGRAM_ID  
GROUP BY D.DEPT_NAME;
```

Below the query editor, the 'Query Result' tab shows the output of the query. It indicates that all rows were fetched in 0.003 seconds. The results are displayed in a table with three columns: DEPT_NAME, NUM_PROGRAMS, and TOTAL_STUDENTS.

DEPT_NAME	NUM_PROGRAMS	TOTAL_STUDENTS
1 MATHEMATICS	1	1
2 COMPUTER SCIENCE	2	2
3 ELECTRICAL ENGINEERING	1	1
4 BUSINESS ADMINISTRATION	2	2

LAB#05 – TASK#09



The screenshot shows the SQL Developer interface. The top toolbar includes icons for running queries, saving, and other standard database operations. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT S.FNAME || ' ' || S.LNAME AS STUDENT_NAME,  
       C.COURSE_NAME,  
       F.NAME AS FACULTY_NAME  
FROM ENROLL E  
JOIN STUDENT S ON E.STUDENT_ID = S.STUDENT_ID  
JOIN COURSE C ON E.COURSE_ID = C.COURSE_ID  
JOIN FACULTY F ON C.FACULTY_ID = F.FACULTY_ID  
WHERE E.SEMESTER = 'FALL 2024';
```

Below the query editor, the 'Query Result' tab shows the output of the query. It indicates that all rows were fetched in 0.01 seconds. The results are displayed in a table with three columns: STUDENT_NAME, COURSE_NAME, and FACULTY_NAME.

STUDENT_NAME	COURSE_NAME	FACULTY_NAME
1 JAMES WILSON	OPERATING SYSTEMS	BOB SMITH
2 JAMES WILSON	DATABASE SYSTEMS	ALICE JOHNSON
3 SOPHIA TAYLOR	DATABASE ADMINISTRATION	BOB SMITH
4 SOPHIA TAYLOR	MARKETING MANAGEMENT	CAROL DAVIS
5 SOPHIA TAYLOR	DATABASE SYSTEMS	ALICE JOHNSON
6 MICHAEL LEE	MARKETING MANAGEMENT	CAROL DAVIS
7 DAVID MARTIN	LINEAR ALGEBRA	DANIEL BROWN
8 ISABELLA ALLEN	DATABASE ADMINISTRATION	BOB SMITH
9 ISABELLA ALLEN	CIRCUIT ANALYSIS	EMILY WHITE

LAB#05 – TASK#10

