

LAB#03 – TASK#01

```
CREATE TABLE PROGRAMS (  
    PROGRAM_ID NUMERIC PRIMARY KEY,  
    PROGRAM_NAME VARCHAR2(30) UNIQUE  
);  
  
ALTER TABLE STUDENTS ADD CONSTRAINT F_KEY FOREIGN KEY (PROGRAM_ID) REFERENCES PROGRAMS (PROGRAM_ID);
```

Script Output x

Task completed in 0.112 seconds

Table PROGRAMS created.

Table STUDENTS altered.

LAB#03 – TASK#02

Welcome Page x homeuser x STUDENTS x

0.054 seconds

Worksheet Query Builder

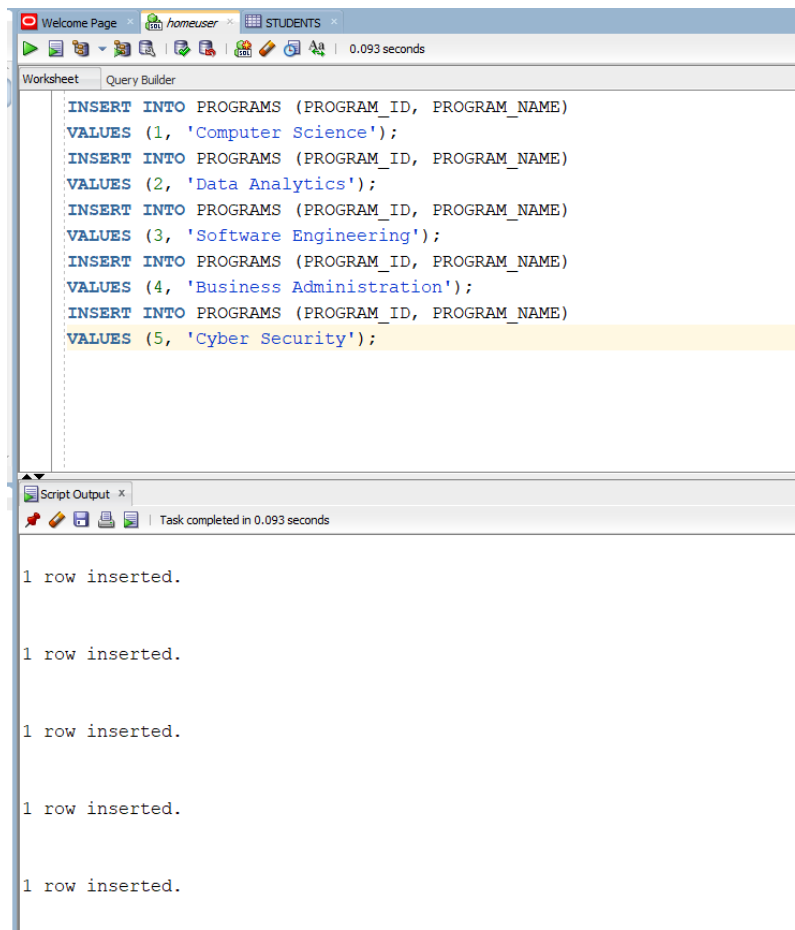
```
CREATE TABLE ENROLLMENT_LOG AS  
SELECT STUDENT_ID, FIRST_NAME, ENROLLMENT_DATE  
FROM STUDENTS  
WHERE TO_CHAR(ENROLLMENT_DATE, 'YYYY') = '2024';
```

Script Output x

Task completed in 0.054 seconds

Table ENROLLMENT_LOG created.

LAB#03 – TASK#03



The screenshot shows the SQL Developer interface with a script in the Query Builder. The script consists of five INSERT statements into the PROGRAMS table, each with a unique PROGRAM_ID and a program name. The script is executed, and the Script Output window shows five rows inserted.

```
INSERT INTO PROGRAMS (PROGRAM_ID, PROGRAM_NAME)
VALUES (1, 'Computer Science');
INSERT INTO PROGRAMS (PROGRAM_ID, PROGRAM_NAME)
VALUES (2, 'Data Analytics');
INSERT INTO PROGRAMS (PROGRAM_ID, PROGRAM_NAME)
VALUES (3, 'Software Engineering');
INSERT INTO PROGRAMS (PROGRAM_ID, PROGRAM_NAME)
VALUES (4, 'Business Administration');
INSERT INTO PROGRAMS (PROGRAM_ID, PROGRAM_NAME)
VALUES (5, 'Cyber Security');
```

Script Output x

Task completed in 0.093 seconds

1 row inserted.

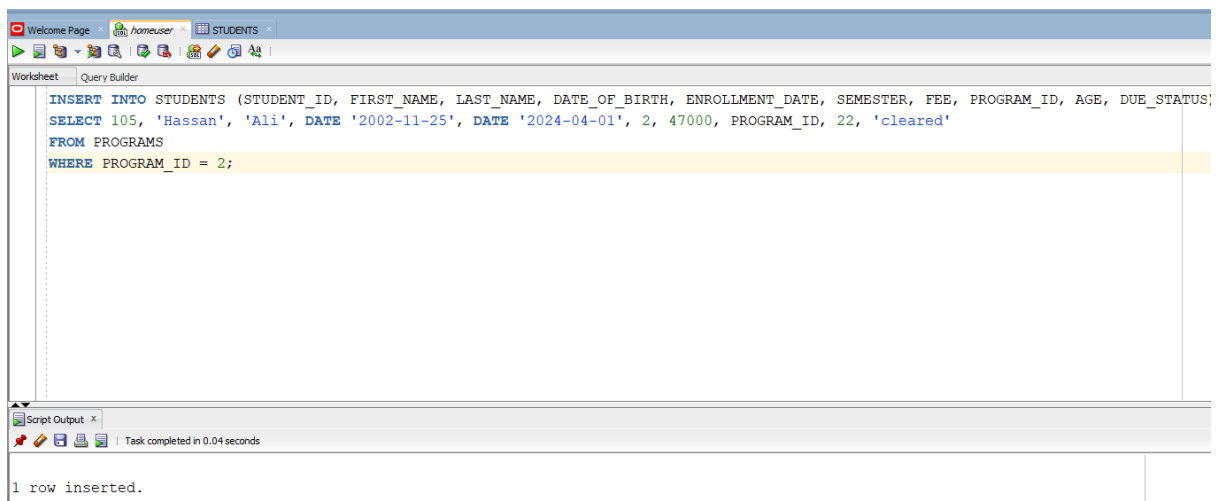
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

LAB#03 – TASK#04



The screenshot shows the SQL Developer interface with a script in the Query Builder. The script consists of an INSERT statement into the STUDENTS table, followed by a SELECT statement to verify the insertion. The script is executed, and the Script Output window shows one row inserted.

```
INSERT INTO STUDENTS (STUDENT_ID, FIRST_NAME, LAST_NAME, DATE_OF_BIRTH, ENROLLMENT_DATE, SEMESTER, FEE, PROGRAM_ID, AGE, DUE_STATUS)
SELECT 105, 'Hassan', 'Ali', DATE '2002-11-25', DATE '2024-04-01', 2, 47000, PROGRAM_ID, 22, 'cleared'
FROM PROGRAMS
WHERE PROGRAM_ID = 2;
```

Script Output x

Task completed in 0.04 seconds

1 row inserted.

NOW TRYING INVALID PROGRAM ID

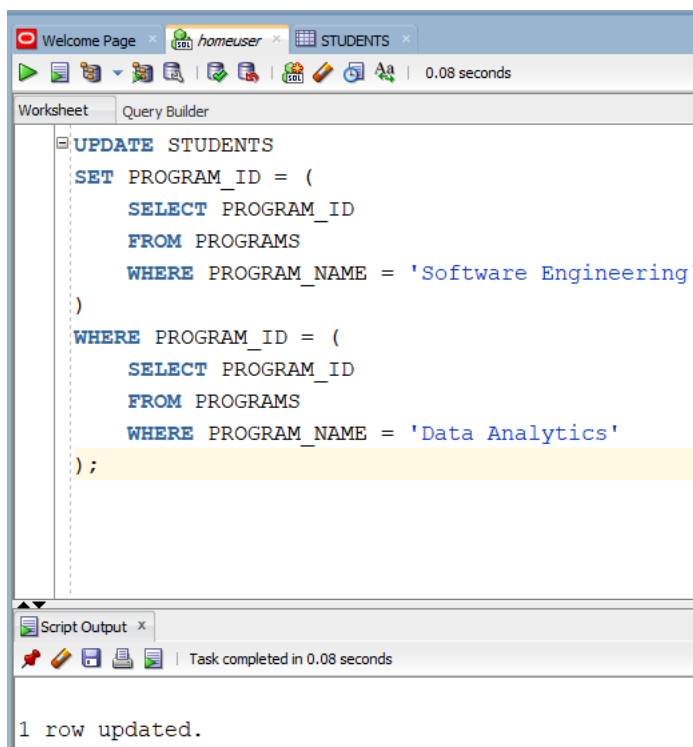


The screenshot shows a database query editor with a 'Query Builder' tab. The query is as follows:

```
INSERT INTO STUDENTS (STUDENT_ID, FIRST_NAME, LAST_NAME, DATE_OF_BIRTH, ENROLLMENT_DATE, SEMESTER, FEE, PROGRAM_ID, AGE, DUE_STATUS)
SELECT 107, 'Rameen', 'Akhtar', DATE '2001-09-10', DATE '2023-09-05', 3, 52000, PROGRAM_ID, 23, 'cleared'
FROM PROGRAMS
WHERE PROGRAM_ID = 8;
```

The 'Script Output' tab at the bottom shows the result: '0 rows inserted.' The task was completed in 0.034 seconds.

LAB#03 – TASK#05

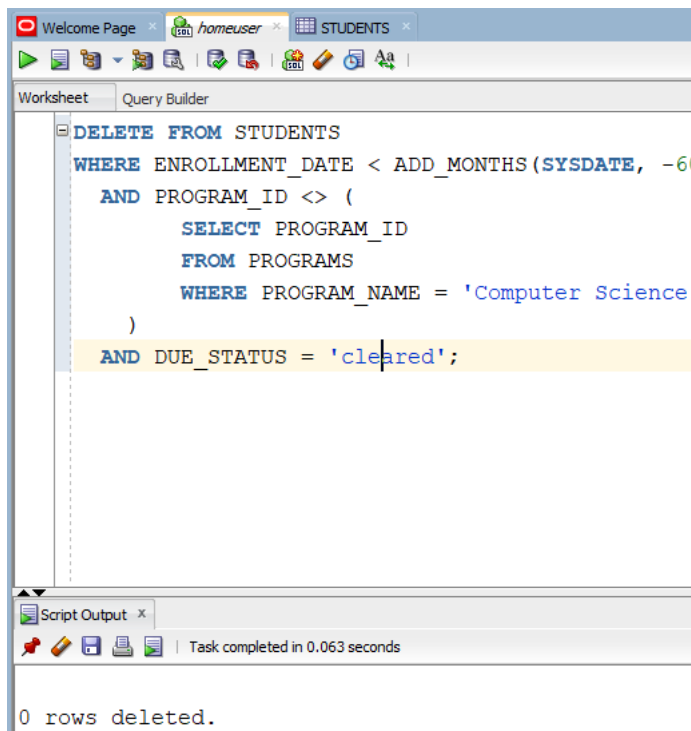


The screenshot shows a database query editor with a 'Query Builder' tab. The query is as follows:

```
UPDATE STUDENTS
SET PROGRAM_ID = (
    SELECT PROGRAM_ID
    FROM PROGRAMS
    WHERE PROGRAM_NAME = 'Software Engineering'
)
WHERE PROGRAM_ID = (
    SELECT PROGRAM_ID
    FROM PROGRAMS
    WHERE PROGRAM_NAME = 'Data Analytics'
);
```

The 'Script Output' tab at the bottom shows the result: '1 row updated.' The task was completed in 0.08 seconds.

LAB#03 – TASK#06

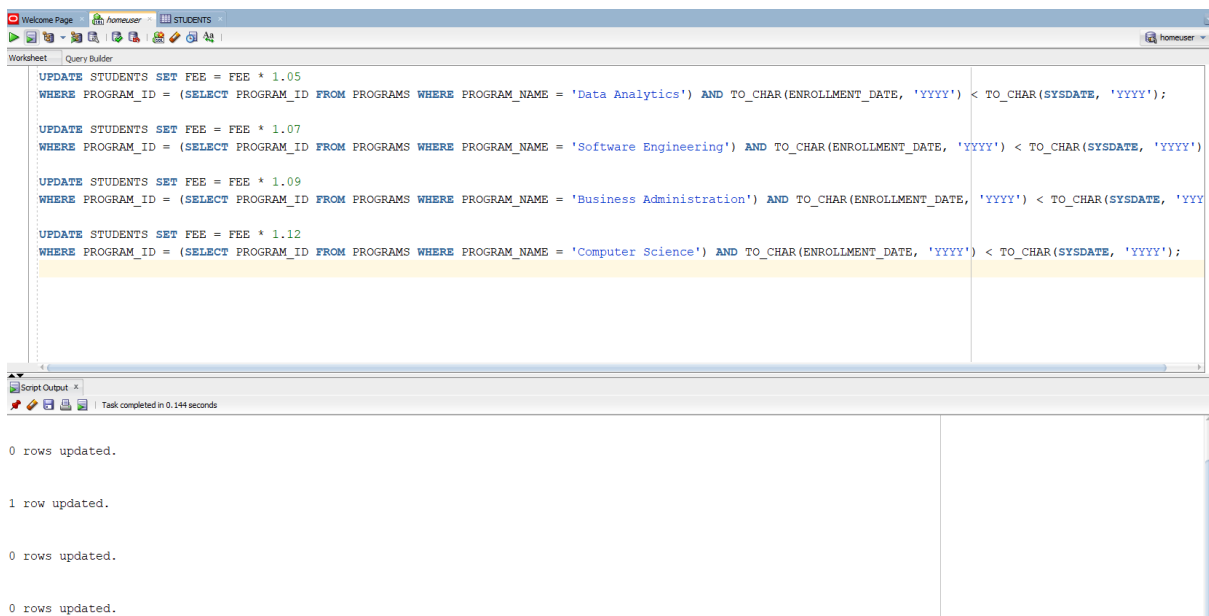


The screenshot shows the SQL Developer interface with a 'Query Builder' tab. The query editor contains the following SQL statement:

```
DELETE FROM STUDENTS
WHERE ENROLLMENT_DATE < ADD_MONTHS (SYSDATE, -60)
AND PROGRAM_ID <> (
    SELECT PROGRAM_ID
    FROM PROGRAMS
    WHERE PROGRAM_NAME = 'Computer Science'
)
AND DUE_STATUS = 'cleared';
```

The 'Script Output' tab at the bottom shows the result: '0 rows deleted.' and 'Task completed in 0.063 seconds'.

LAB#03 – TASK#07



The screenshot shows the SQL Developer interface with a 'Query Builder' tab. The query editor contains the following SQL statements:

```
UPDATE STUDENTS SET FEE = FEE * 1.05
WHERE PROGRAM_ID = (SELECT PROGRAM_ID FROM PROGRAMS WHERE PROGRAM_NAME = 'Data Analytics') AND TO_CHAR(ENROLLMENT_DATE, 'YYYY') < TO_CHAR(SYSDATE, 'YYYY');

UPDATE STUDENTS SET FEE = FEE * 1.07
WHERE PROGRAM_ID = (SELECT PROGRAM_ID FROM PROGRAMS WHERE PROGRAM_NAME = 'Software Engineering') AND TO_CHAR(ENROLLMENT_DATE, 'YYYY') < TO_CHAR(SYSDATE, 'YYYY');

UPDATE STUDENTS SET FEE = FEE * 1.09
WHERE PROGRAM_ID = (SELECT PROGRAM_ID FROM PROGRAMS WHERE PROGRAM_NAME = 'Business Administration') AND TO_CHAR(ENROLLMENT_DATE, 'YYYY') < TO_CHAR(SYSDATE, 'YYYY');

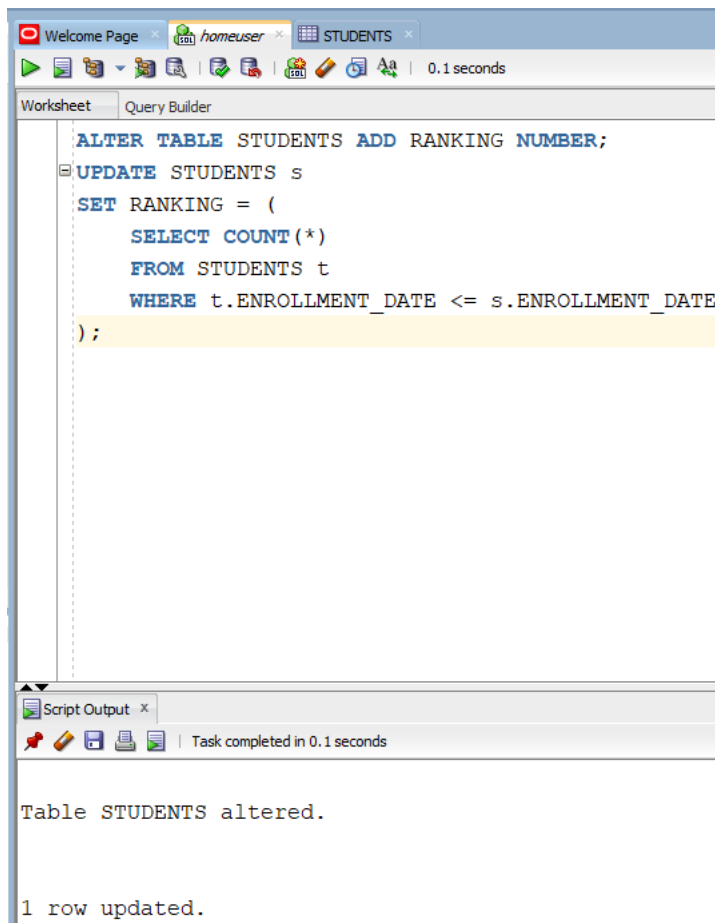
UPDATE STUDENTS SET FEE = FEE * 1.12
WHERE PROGRAM_ID = (SELECT PROGRAM_ID FROM PROGRAMS WHERE PROGRAM_NAME = 'Computer Science') AND TO_CHAR(ENROLLMENT_DATE, 'YYYY') < TO_CHAR(SYSDATE, 'YYYY');
```

The 'Script Output' tab at the bottom shows the results for each statement:

- 0 rows updated.
- 1 row updated.
- 0 rows updated.
- 0 rows updated.

The total execution time is 'Task completed in 0.144 seconds'.

LAB#03 – TASK#08



The screenshot shows a database application interface. At the top, there are three tabs: 'Welcome Page', 'homeuser', and 'STUDENTS'. Below the tabs is a toolbar with various icons and a timer showing '0.1 seconds'. The main area is divided into two sections: 'Worksheet' and 'Query Builder'. The 'Query Builder' section contains the following SQL code:

```
ALTER TABLE STUDENTS ADD RANKING NUMBER;  
UPDATE STUDENTS s  
SET RANKING = (  
    SELECT COUNT(*)  
    FROM STUDENTS t  
    WHERE t.ENROLLMENT_DATE <= s.ENROLLMENT_DATE  
);
```

Below the query editor is a 'Script Output' window. It shows the results of the executed queries:

```
Table STUDENTS altered.  
  
1 row updated.
```