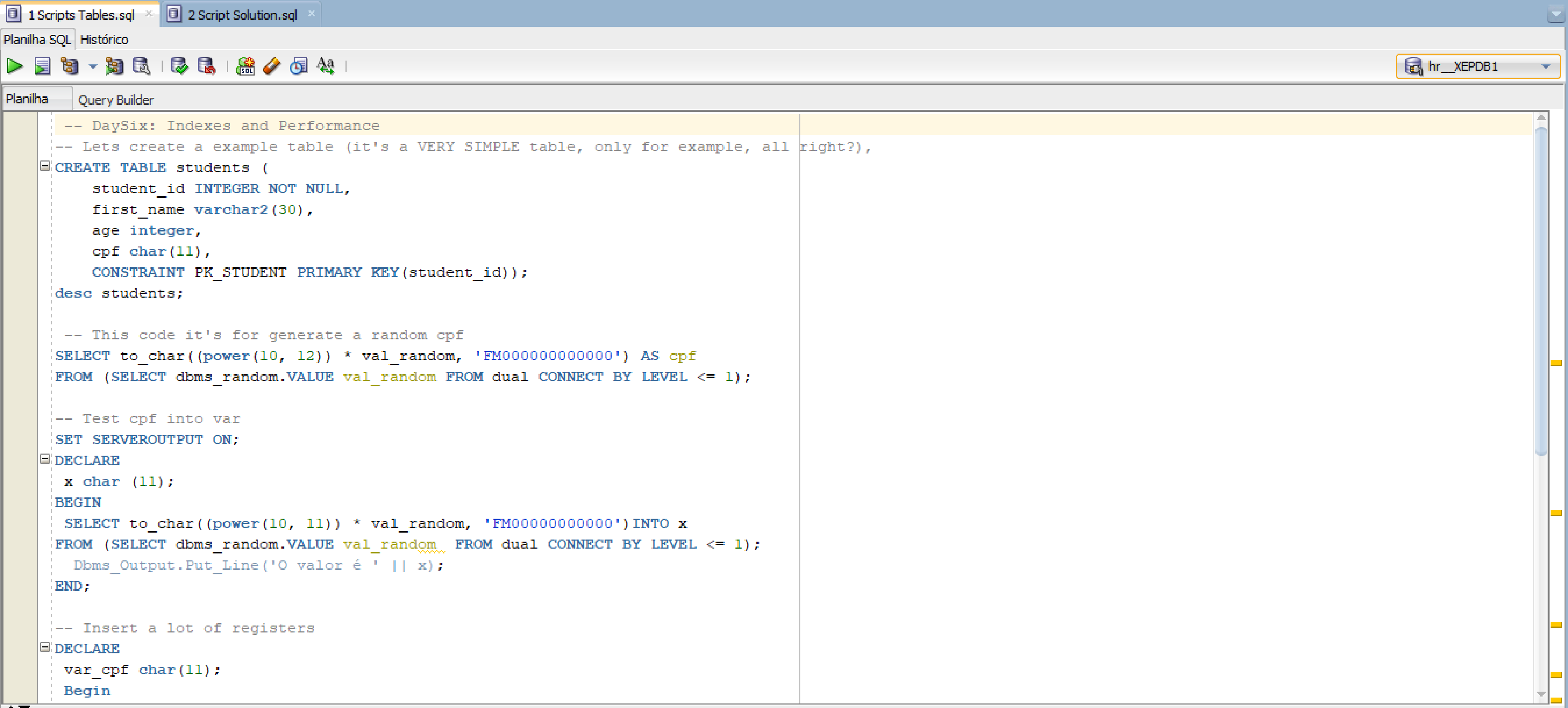
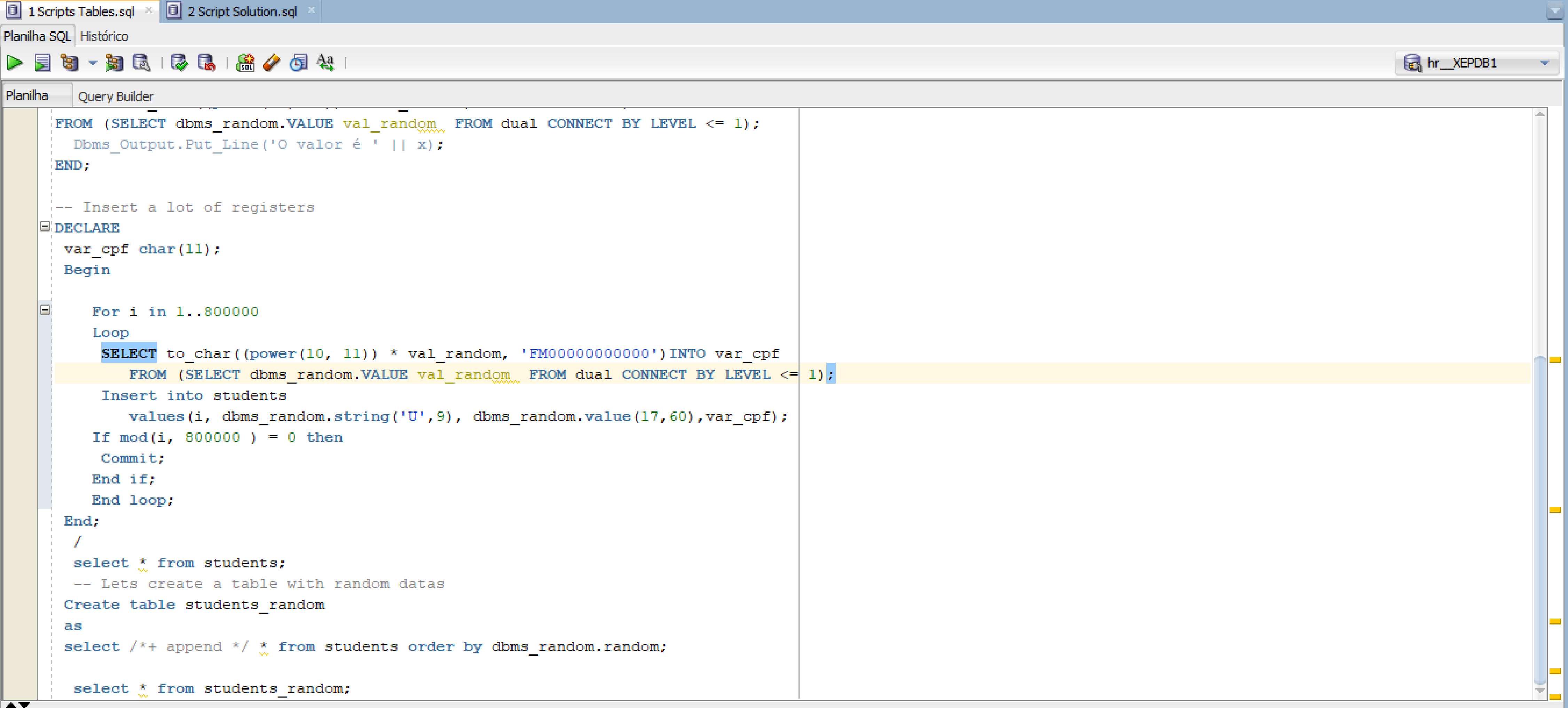


```sql
 -- DaySix: Indexes and Performance
-- Lets create a example table (it's a VERY SIMPLE table, only for example, all right?),
CREATE TABLE students (
    student_id INTEGER NOT NULL,
    first_name varchar2(30),
    age integer,
    cpf char(11),
    CONSTRAINT PK_STUDENT PRIMARY KEY(student_id));
desc students;

 -- This code it's for generate a random cpf
SELECT to_char((power(10, 12)) * val_random, 'FM000000000000') AS cpf
FROM (SELECT dbms_random.VALUE val_random FROM dual CONNECT BY LEVEL <= 1);

-- Test cpf into var
SET SERVEROUTPUT ON;
DECLARE
 x char (11);
BEGIN
 SELECT to_char((power(10, 11)) * val_random, 'FM00000000000')INTO x
FROM (SELECT dbms_random.VALUE val_random  FROM dual CONNECT BY LEVEL <= 1);
  Dbms_Output.Put_Line('O valor é ' || x);
END;

-- Insert a lot of registers
DECLARE
 var_cpf char(11);
 Begin
```

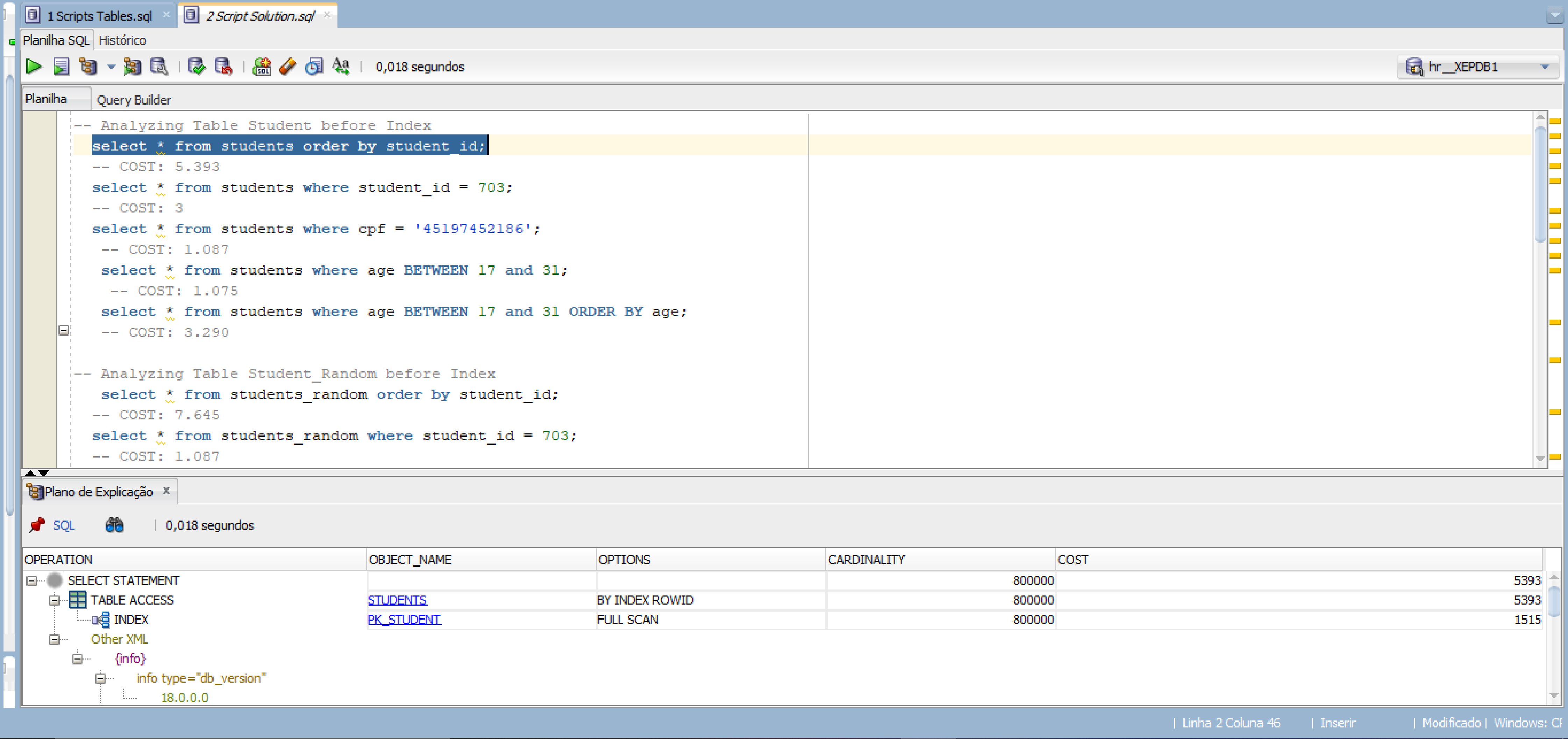


```
FROM (SELECT dbms_random.VALUE val_random  FROM dual CONNECT BY LEVEL <= 1);
  Dbms_Output.Put_Line('O valor é ' || x);
END;

-- Insert a lot of registers
DECLARE
 var_cpf char(11);
 Begin

    For i in 1..800000
    Loop
     SELECT to_char((power(10, 11)) * val_random, 'FM00000000000')INTO var_cpf
        FROM (SELECT dbms_random.VALUE val_random  FROM dual CONNECT BY LEVEL <= 1);
     Insert into students
        values(i, dbms_random.string('U',9), dbms_random.value(17,60),var_cpf);
    If mod(i, 800000 ) = 0 then
     Commit;
    End if;
    End loop;
 End;
  /
  select * from students;
  -- Lets create a table with random datas
 Create table students_random
 as
 select /*+ append */ * from students order by dbms_random.random;

  select * from students_random;
```

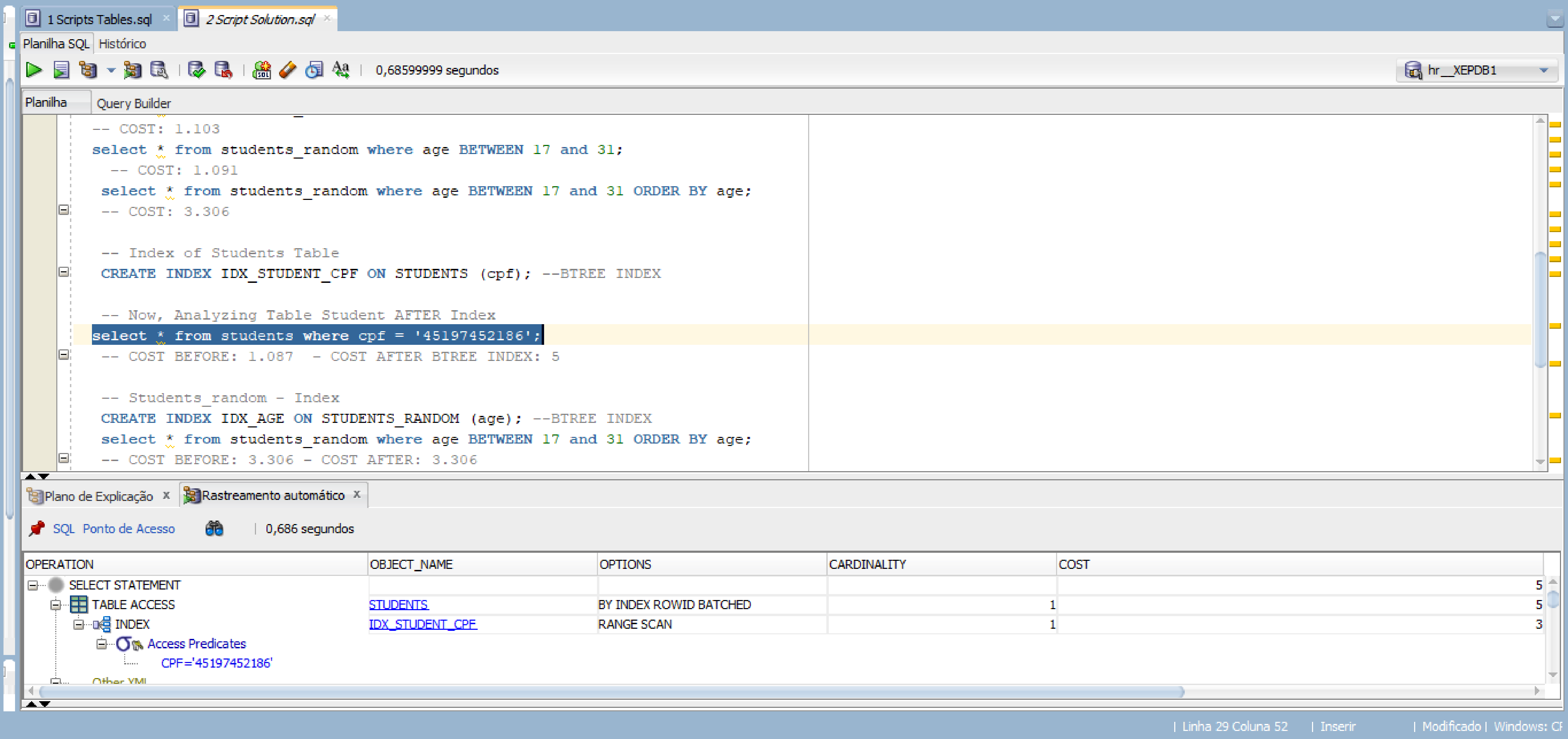


```
-- Analyzing Table Student before Index
  select * from students order by student_id;
  -- COST: 5.393
  select * from students where student_id = 703;
  -- COST: 3
  select * from students where cpf = '45197452186';
   -- COST: 1.087
   select * from students where age BETWEEN 17 and 31;
    -- COST: 1.075
   select * from students where age BETWEEN 17 and 31 ORDER BY age;
   -- COST: 3.290

-- Analyzing Table Student_Random before Index
   select * from students_random order by student_id;
  -- COST: 7.645
  select * from students_random where student_id = 703;
  -- COST: 1.087
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			800000	5393
TABLE ACCESS	STUDENTS	BY INDEX ROWID	800000	5393
INDEX	PK_STUDENT	FULL SCAN	800000	1515
Other XML				
{info}				
info type="db_version"				
18.0.0.0				

| Linha 2 Coluna 46 | Inserir | Modificado | Windows: CR



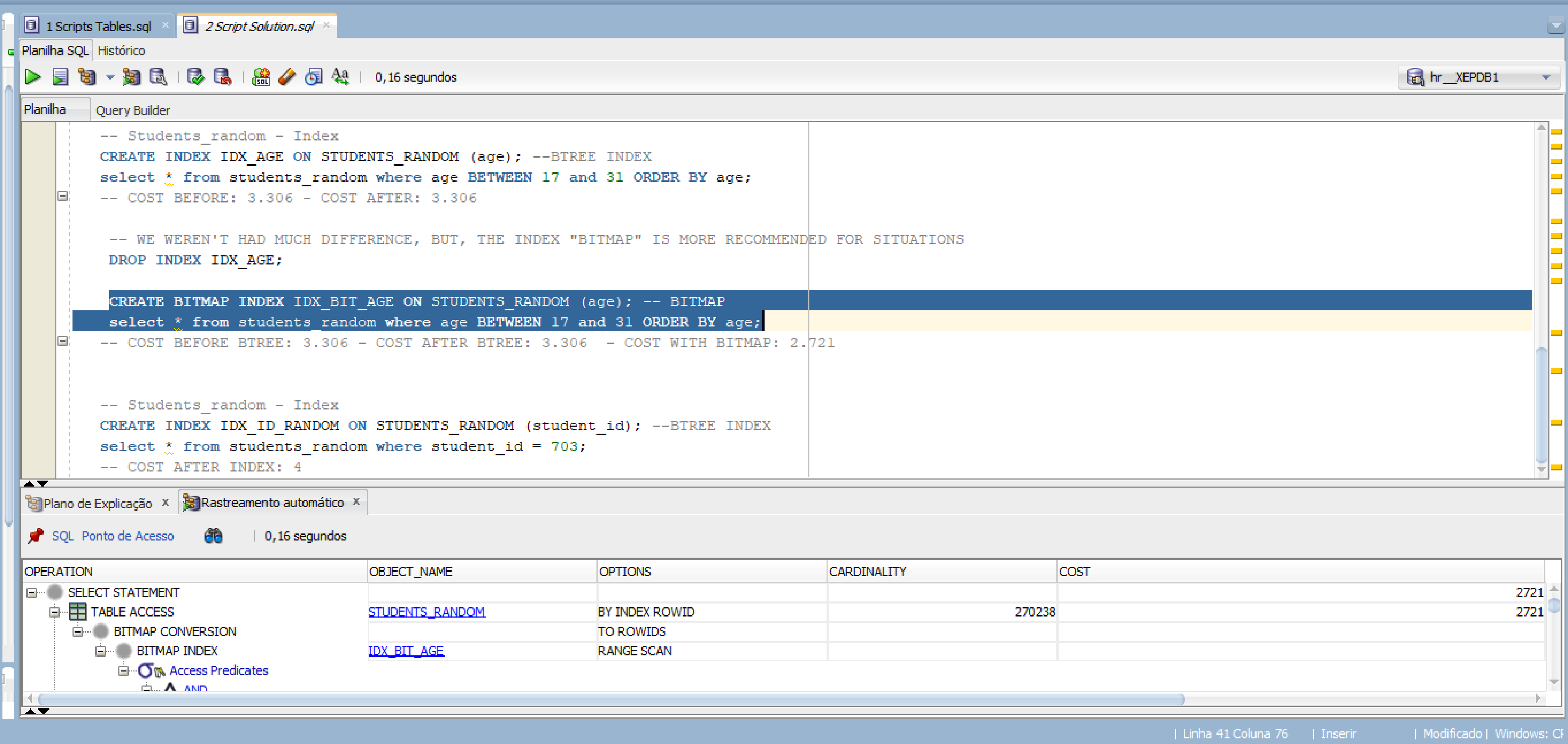
```
-- COST: 1.103
select * from students_random where age BETWEEN 17 and 31;
  -- COST: 1.091
 select * from students_random where age BETWEEN 17 and 31 ORDER BY age;
 -- COST: 3.306

 -- Index of Students Table
 CREATE INDEX IDX_STUDENT_CPF ON STUDENTS (cpf); --BTREE INDEX

 -- Now, Analyzing Table Student AFTER Index
select * from students where cpf = '45197452186';
 -- COST BEFORE: 1.087  - COST AFTER BTREE INDEX: 5

 -- Students_random - Index
 CREATE INDEX IDX_AGE ON STUDENTS_RANDOM (age); --BTREE INDEX
 select * from students_random where age BETWEEN 17 and 31 ORDER BY age;
 -- COST BEFORE: 3.306 - COST AFTER: 3.306
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				5
TABLE ACCESS	STUDENTS	BY INDEX ROWID BATCHED	1	5
INDEX	IDX_STUDENT_CPF	RANGE SCAN	1	3
Access Predicates				
CPF='45197452186'				
Other XML				



```
-- Students_random - Index
CREATE INDEX IDX_AGE ON STUDENTS_RANDOM (age); --BTREE INDEX
select * from students_random where age BETWEEN 17 and 31 ORDER BY age;
-- COST BEFORE: 3.306 - COST AFTER: 3.306

 -- WE WEREN'T HAD MUCH DIFFERENCE, BUT, THE INDEX "BITMAP" IS MORE RECOMMENDED FOR SITUATIONS
 DROP INDEX IDX_AGE;

 CREATE BITMAP INDEX IDX_BIT_AGE ON STUDENTS_RANDOM (age); -- BITMAP
 select * from students_random where age BETWEEN 17 and 31 ORDER BY age;
-- COST BEFORE BTREE: 3.306 - COST AFTER BTREE: 3.306  - COST WITH BITMAP: 2.721


-- Students_random - Index
CREATE INDEX IDX_ID_RANDOM ON STUDENTS_RANDOM (student_id); --BTREE INDEX
select * from students_random where student_id = 703;
-- COST AFTER INDEX: 4
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				2721
TABLE ACCESS	STUDENTS_RANDOM	BY INDEX ROWID	270238	2721
BITMAP CONVERSION		TO ROWIDS		
BITMAP INDEX	IDX_BIT_AGE	RANGE SCAN		
Access Predicates				
AND				