/\*

**Problem 1:**

Display first and last name, hire\_date, and salary for employees hired before January 1,1994, or having a salary greater

than 10000. Order the resulting data set by salary in descending order and hire\_date in ascending order.

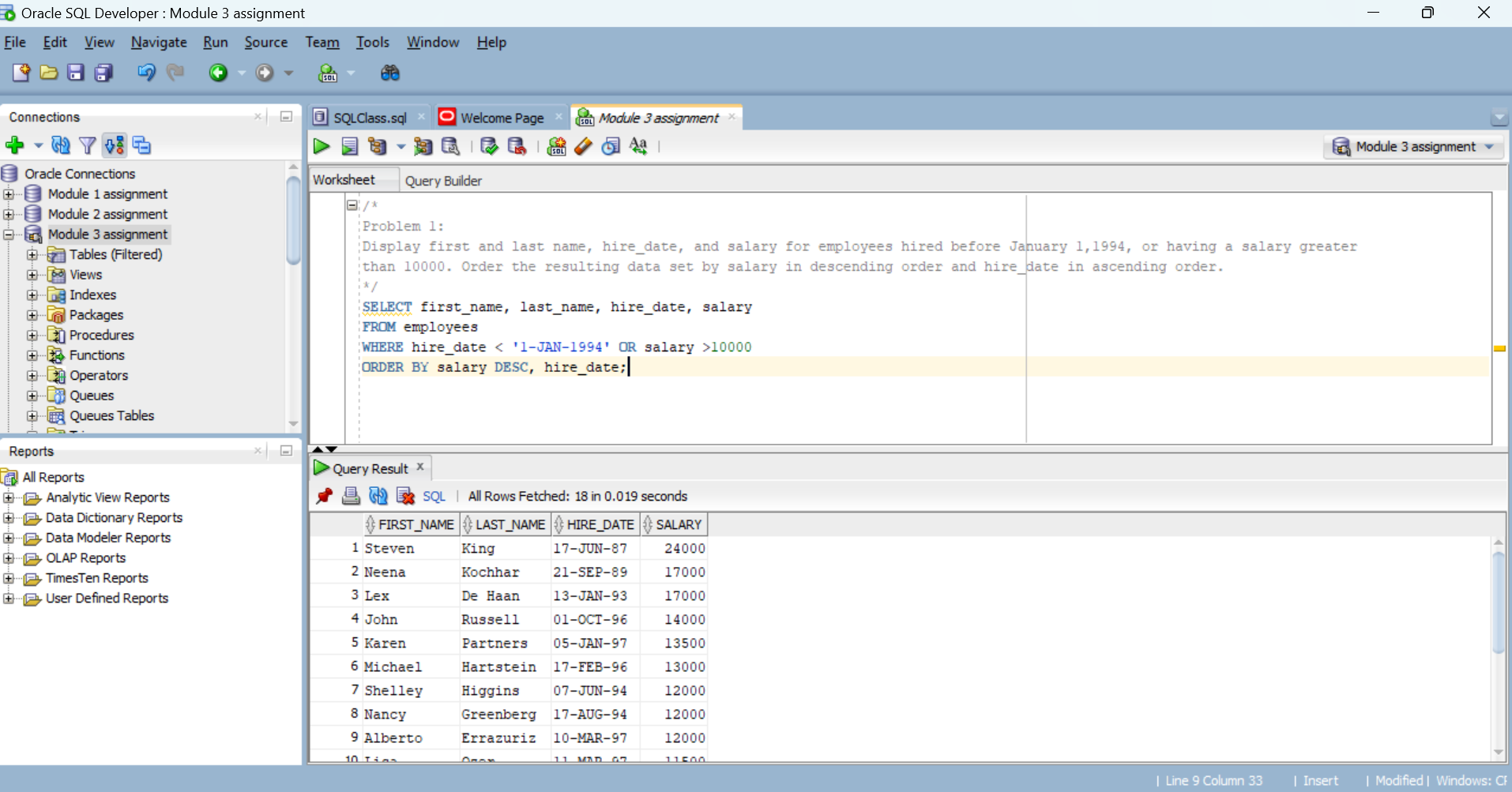
\*/

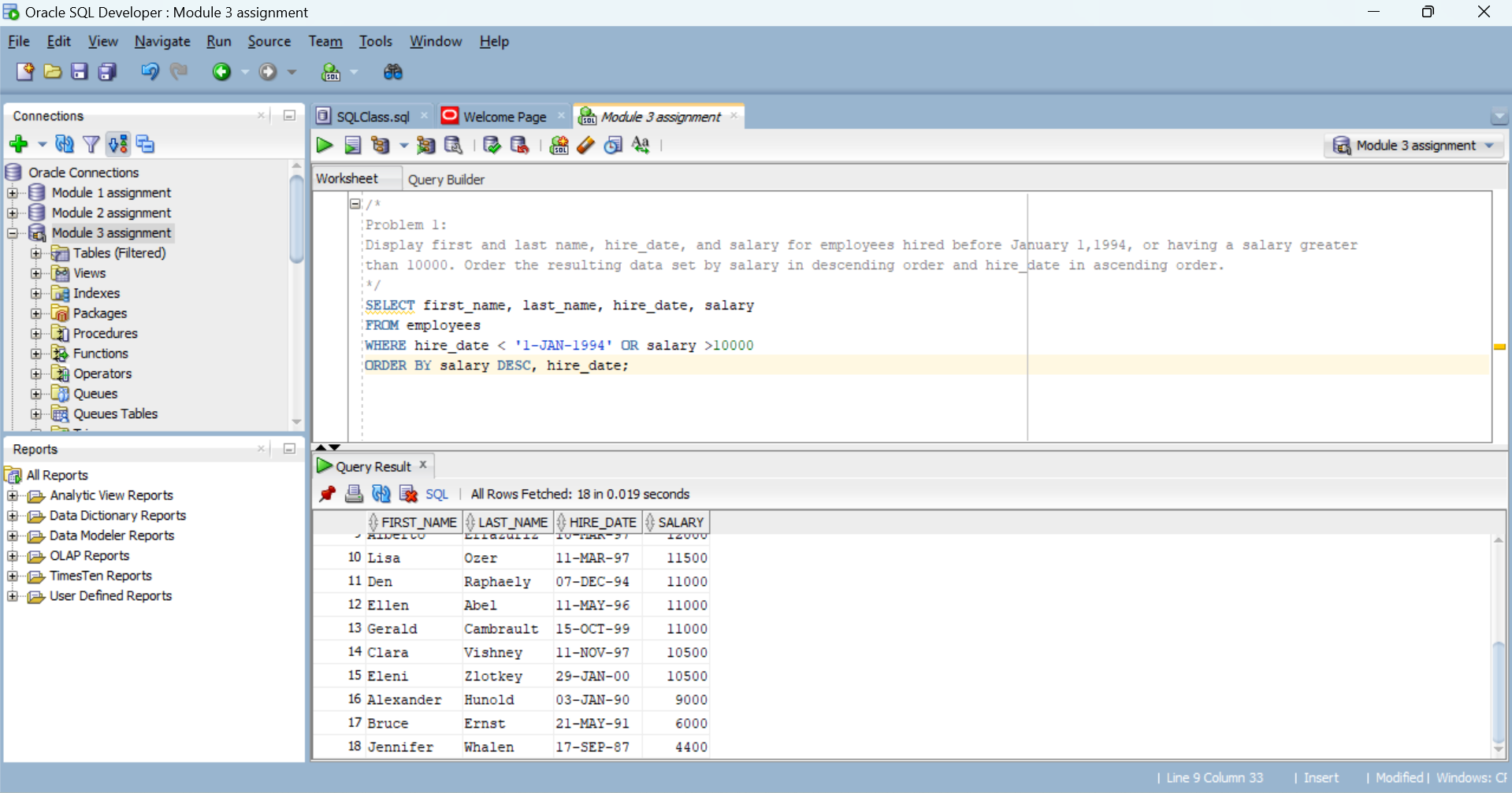
SELECT first\_name, last\_name, hire\_date, salary

FROM employees

WHERE hire\_date < '1-JAN-1994' OR salary >10000

ORDER BY salary DESC, hire\_date;





/\*

**Problem 2:**

Display location\_id city, state\_province, and the following derived column from table locations:

Extract the street name from the street\_address column (= removing the street number) and alias it as street.

Make sure to remove any leading or trailing blank spaces.

Filter the data for location\_id values 1500, 1700, 1800, and 1900. Order the resulting data set by city in ascending order.

Issue the following command before executing the query to reduce the formatted length of column street:

COLUMN street FORMAT a20

\*/

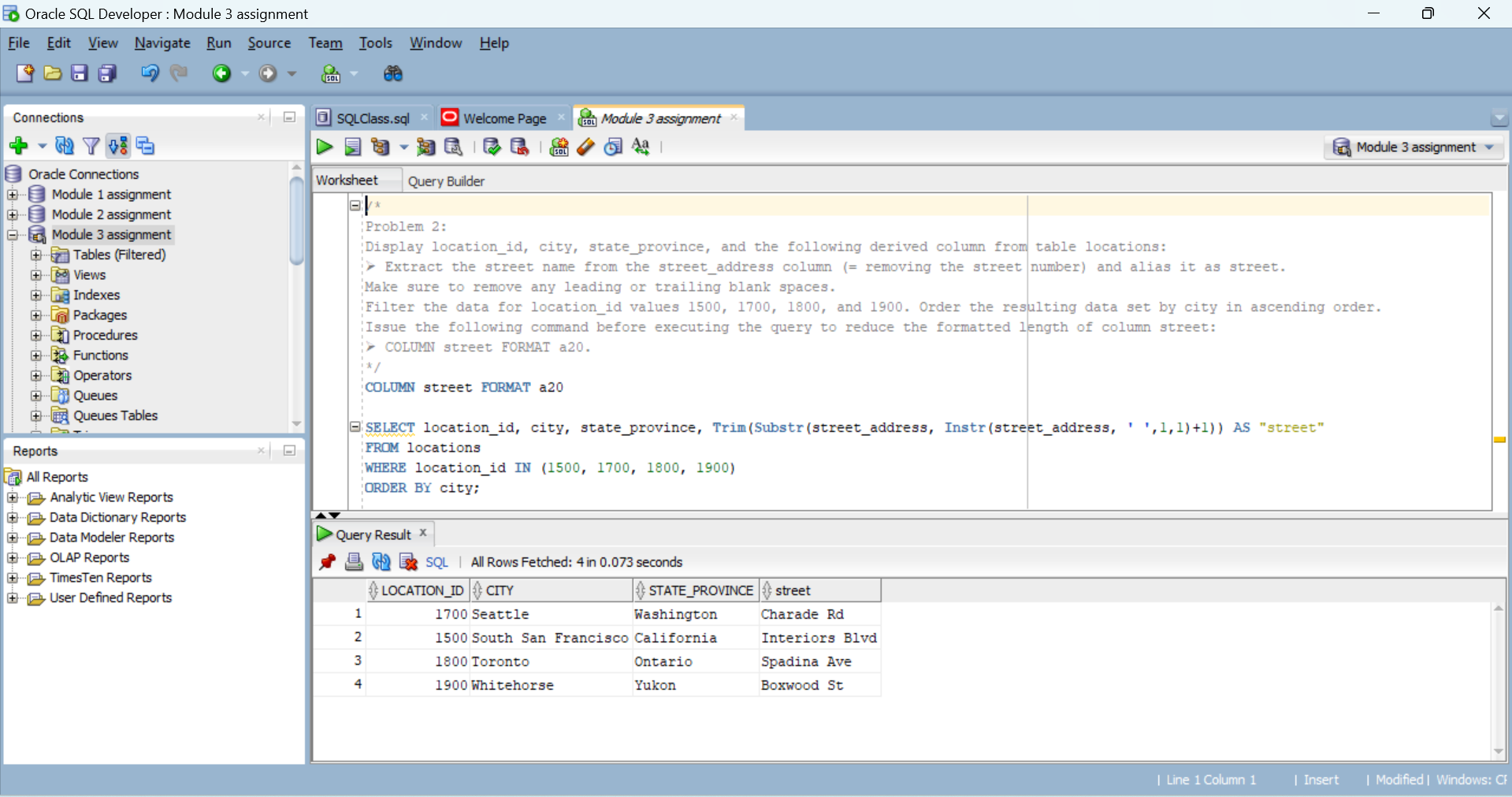
SELECT location\_id, city, state\_province,

Trim(Substr(street\_address,Instr(street\_address, ' ',1)+1)) AS "street"

FROM locations

Where location\_id IN (1500, 1700, 1800, 1900)

ORDER BY city;



/\*

**Problem 3:**

Display first\_name, last\_name, hire\_date, job\_id and salary for job\_id values starting with SA and having a salary equal

to or greater than 10000. Order the resulting data set by salary and hire\_date in ascending order.

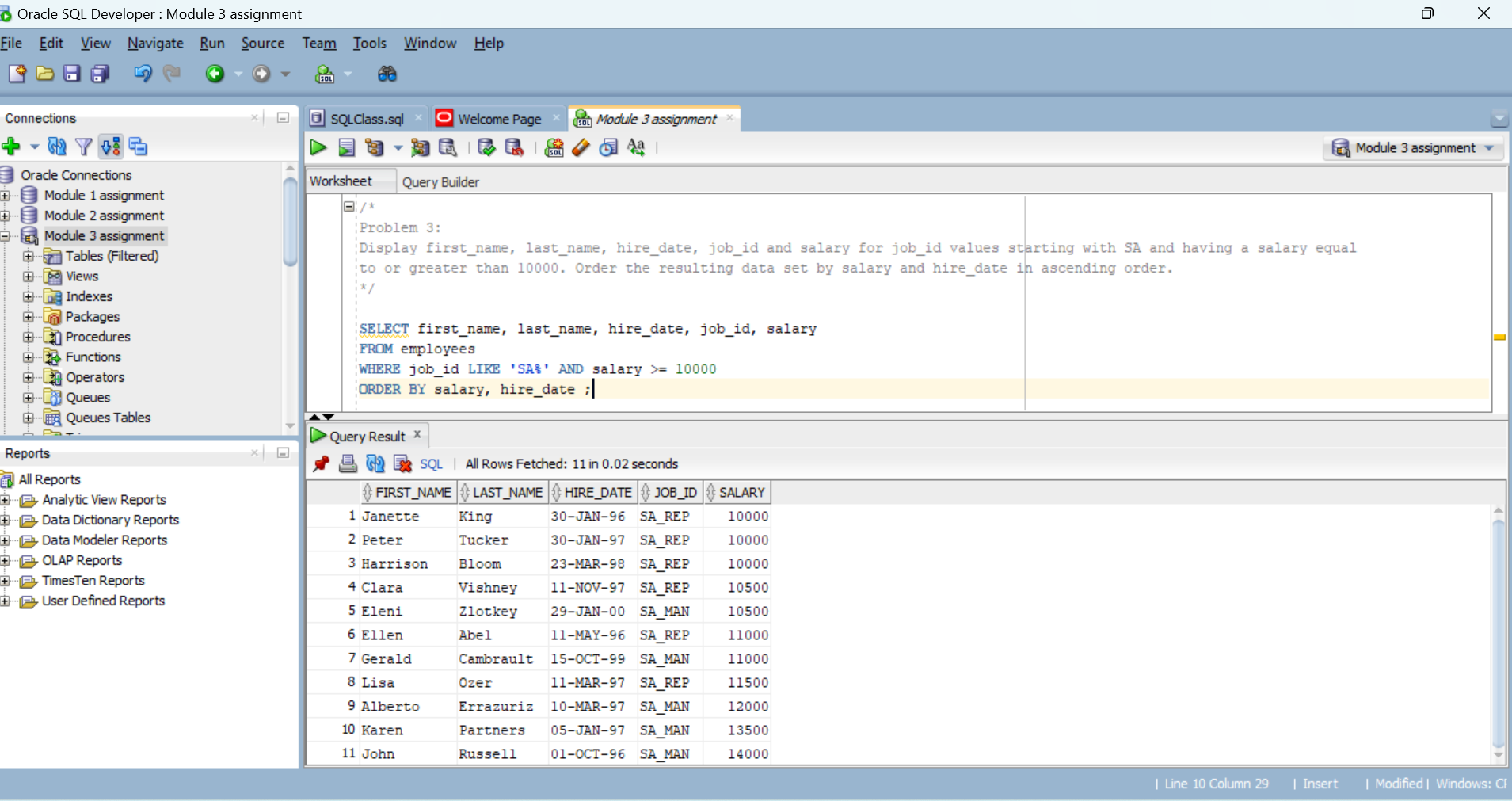
\*/

SELECT first\_name, last\_name, hire\_date, job\_id, salary

FROM employees

Where job\_id LIKE 'SA%' AND salary >= 10000

ORDER BY salary, hire\_date;



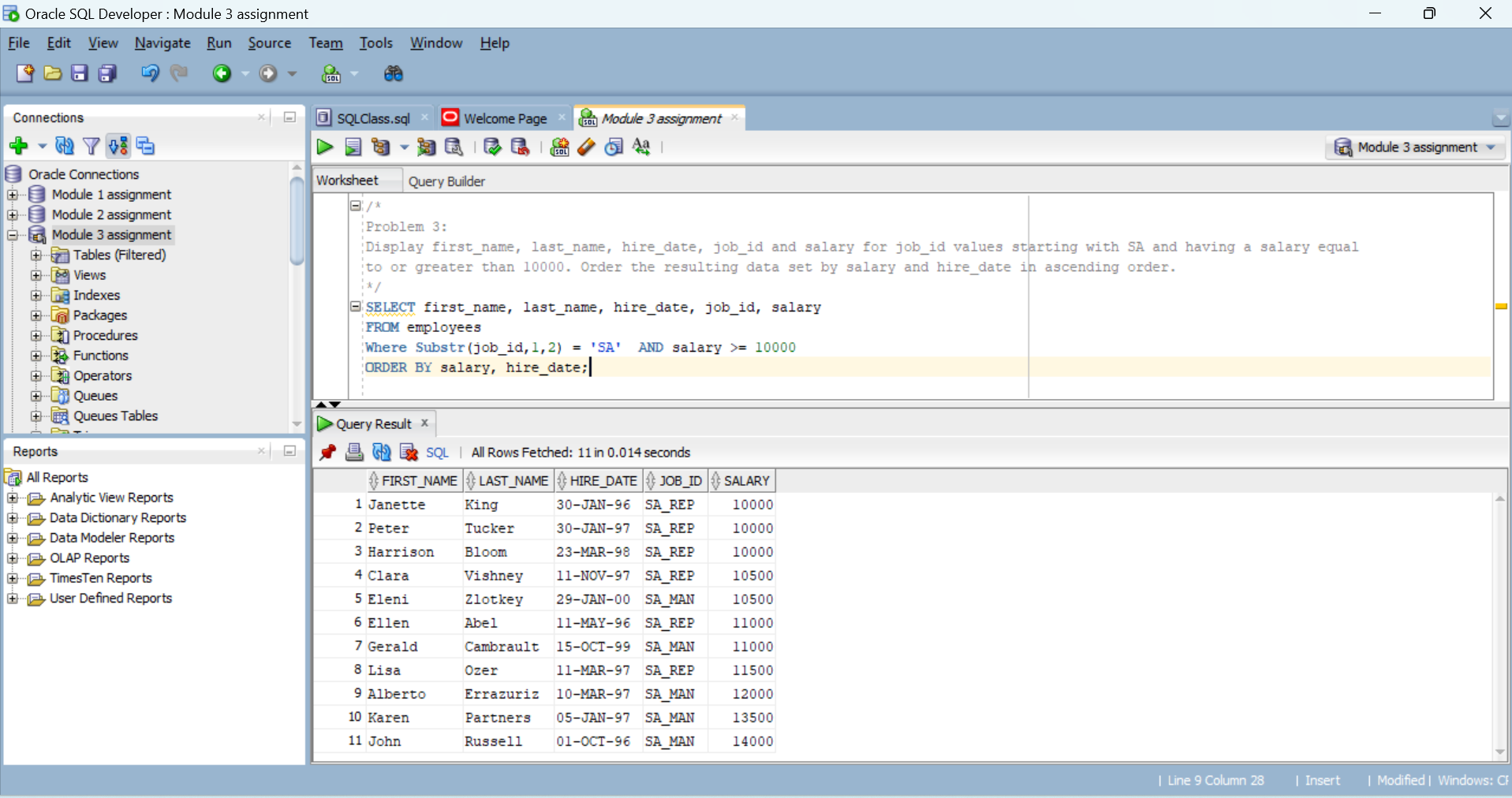
The other solution using Substr function:

SELECT first\_name, last\_name, hire\_date, job\_id, salary

FROM employees

Where Substr(job\_id,1,2) = 'SA' AND salary >= 10000

ORDER BY salary, hire\_date;



/\*

**Problem 4:**

Display first\_name, last\_name, salary and hire\_date for employees having a salary of equal to or greater than 10000.

Create an additional column which displays the string ‘Before 2000’ for hire\_date values before January 1st, 2000 and

‘On or after 2000’ for all other values. Use a column alias of Hire\_Date\_Century. Order the resulting data set by the

same column expression (Hire\_Date\_Century) in descending order, last\_name and first\_name both in ascending order.

\*/

SELECT first\_name, last\_name, hire\_date, job\_id, salary,

CASE

When hire\_date < '1-Jan-2000' then 'Befor 2000'

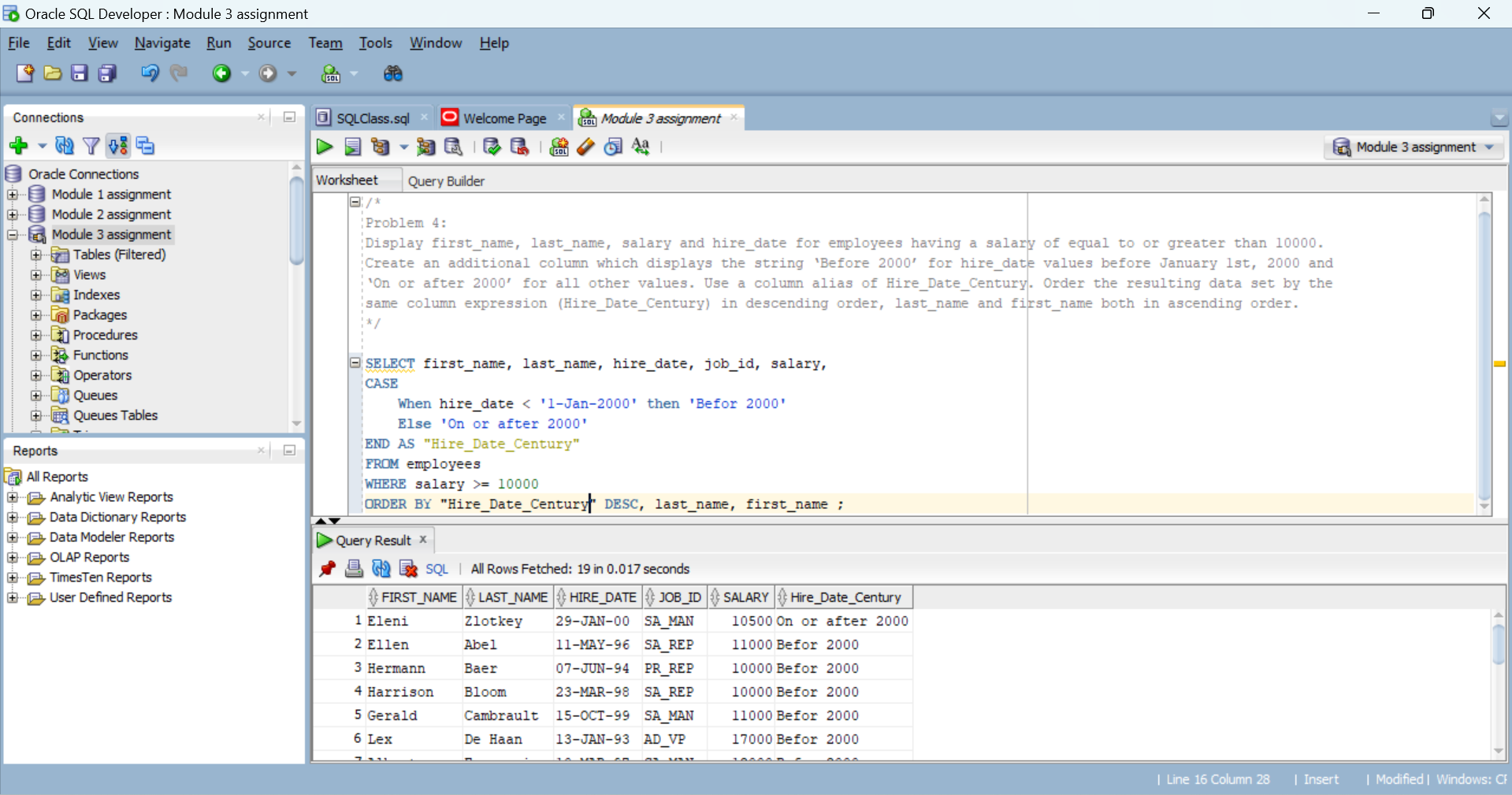
Else 'On or after 2000'

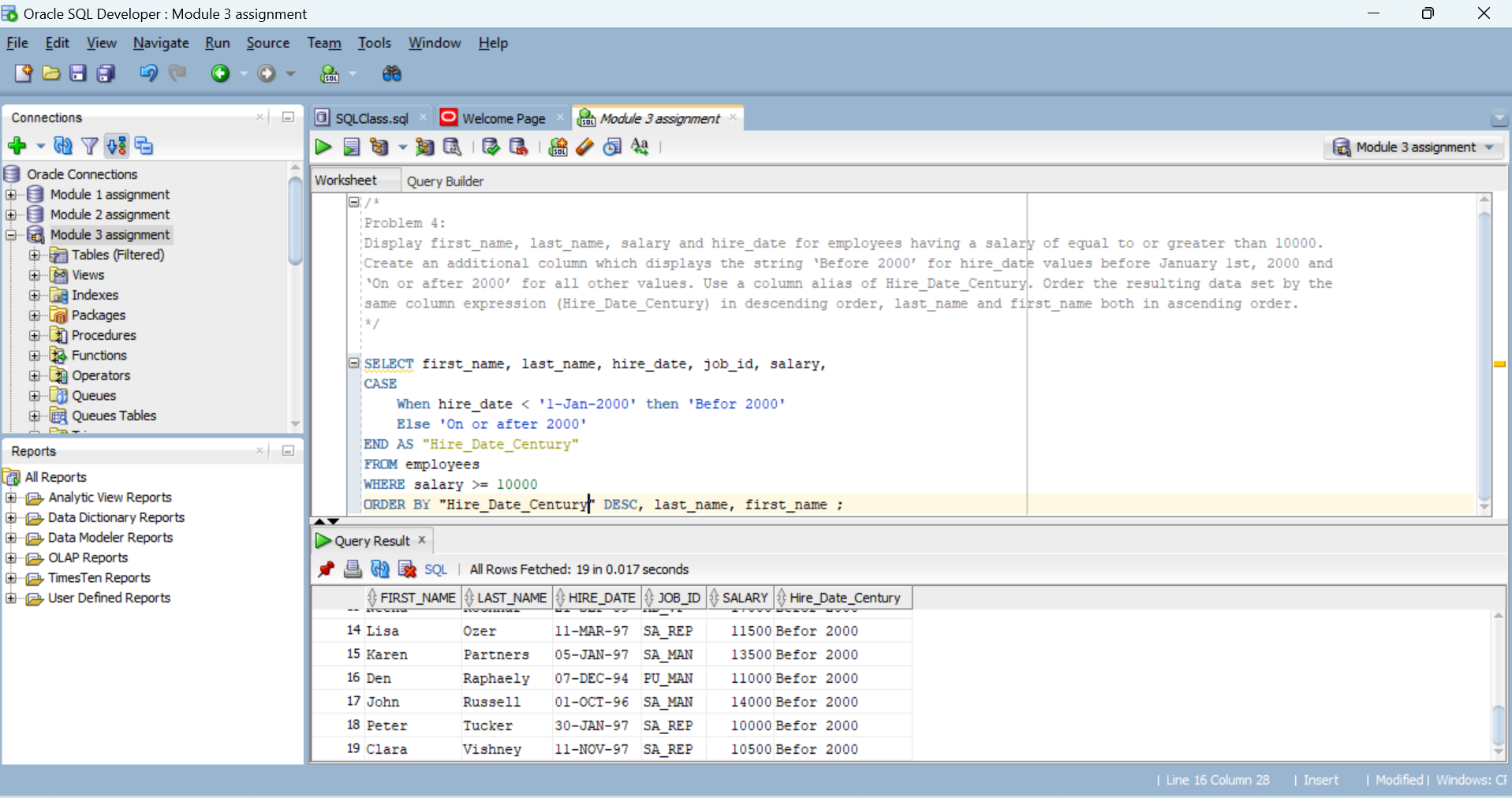
END AS "Hire\_Date\_Century"

FROM employees

WHERE salary >= 10000

ORDER BY "Hire\_Date\_Century" DESC, last\_name, first\_name ;





/\*

**Problem 5:**

Display last\_name, salary, department\_id and commission\_pct and order the resulting data set by department\_id and

last\_name. Create an additional column for giving employees an additional bonus: Employees will receive a certain

percentage of their salary as a bonus: This percentage is 10% of their commission\_pct value, for those employees having

no commission\_pct use 1%. Alias this column Addtl\_Bonus. Display the resulting data set only for department\_id values

of 20 and 80 and for last\_name starting with A or F or H or S.

\*/

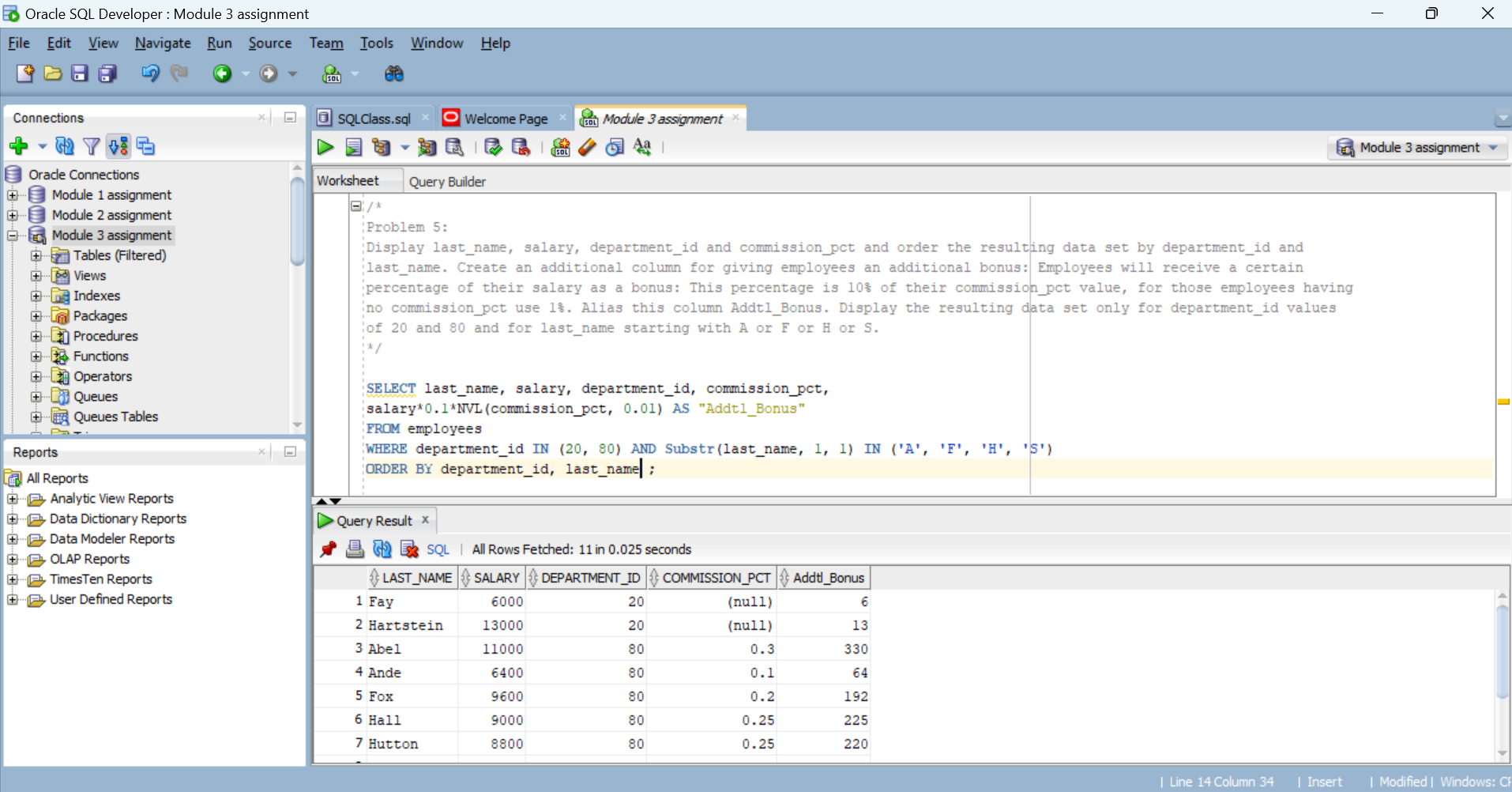
SELECT last\_name, salary, department\_id, commission\_pct,

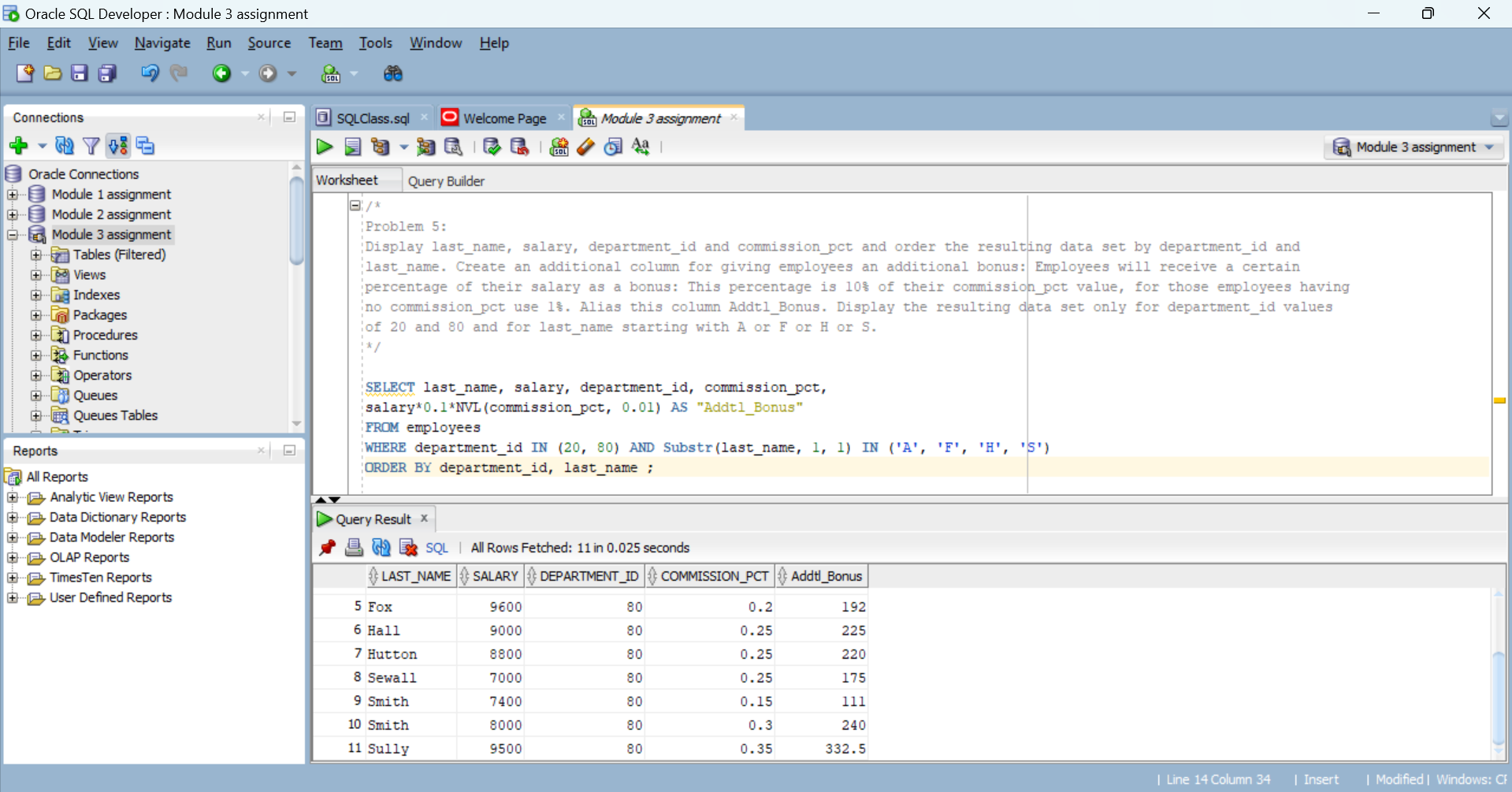
salary\*0.1\*NVL(commission\_pct, 0.01) AS "Addtl\_Bonus"

FROM employees

WHERE department\_id IN (20, 80) AND Substr(last\_name, 1, 1) IN ('A', 'F', 'H', 'S')

ORDER BY department\_id, last\_name ;





/\*

**Problem 6:**

Create the following two SQL statements from table countries displaying only the country\_name column:

Set 1: Countries starting with A, N, or I

Set 2: Countries from Europe (region\_id=1)

Then create 3 queries performing the following SET operations (must use SET operators):

Show the countries that both sets have in common

Show the countries that exists in Set 1 but not in Set 2

Show all countries from both sets including duplicates

Deliverables: 5 queries

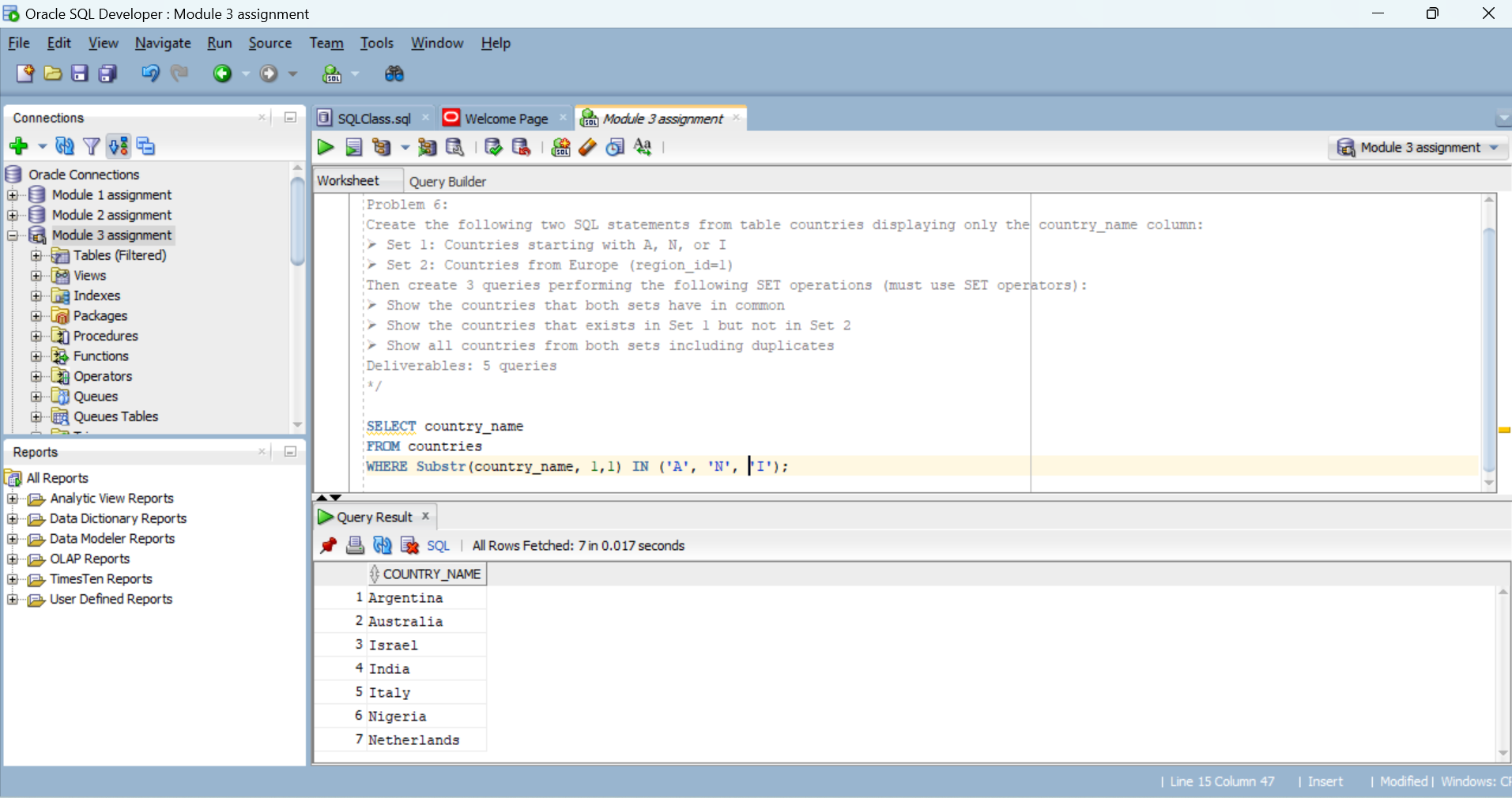
\*/

1.

SELECT country\_name

FROM countries

WHERE Substr(country\_name, 1,1) IN ('A', 'N', 'I');

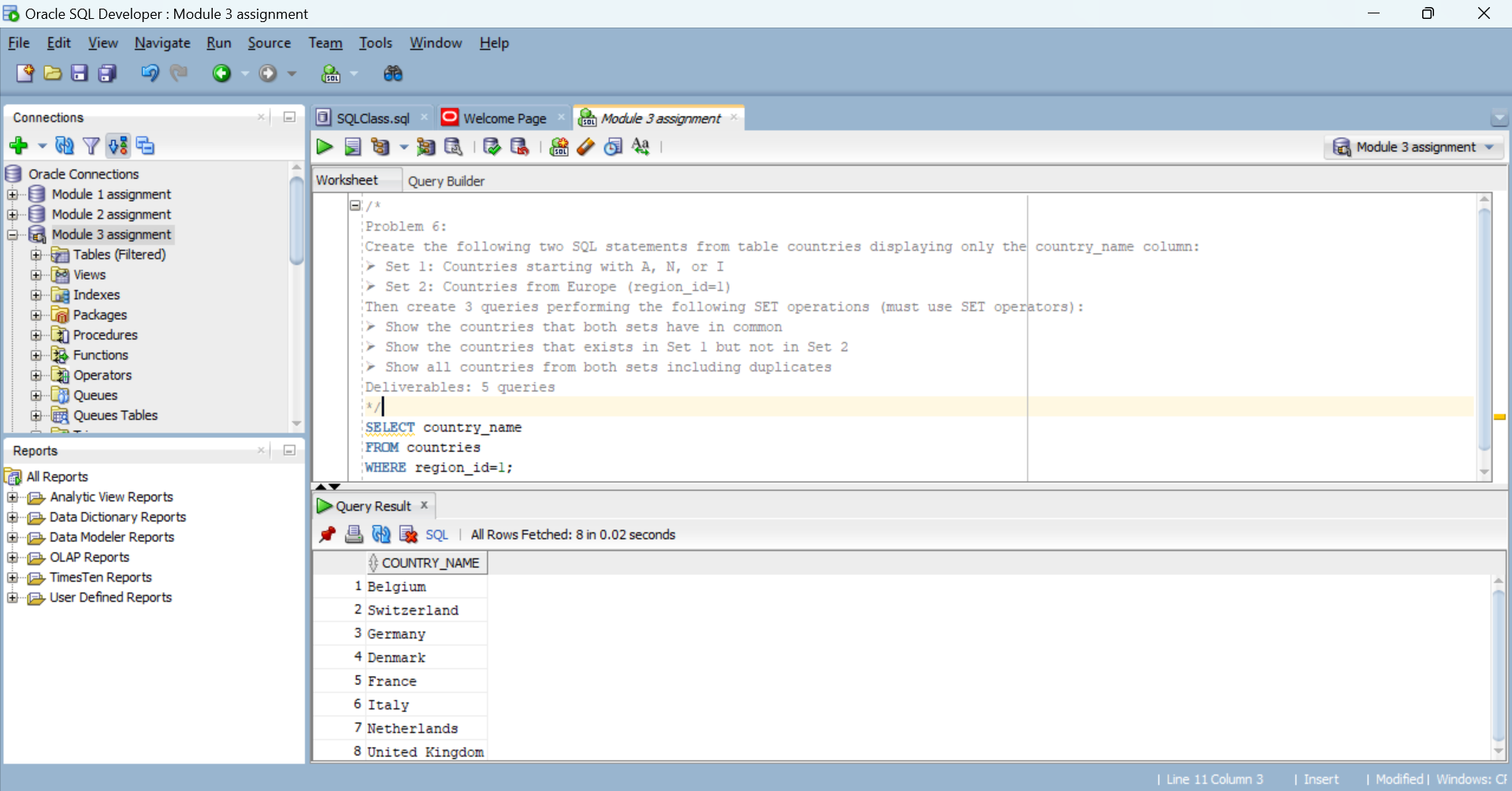


2.

SELECT country\_name

FROM countries

WHERE region\_id=1;



3.

SELECT country\_name

FROM countries

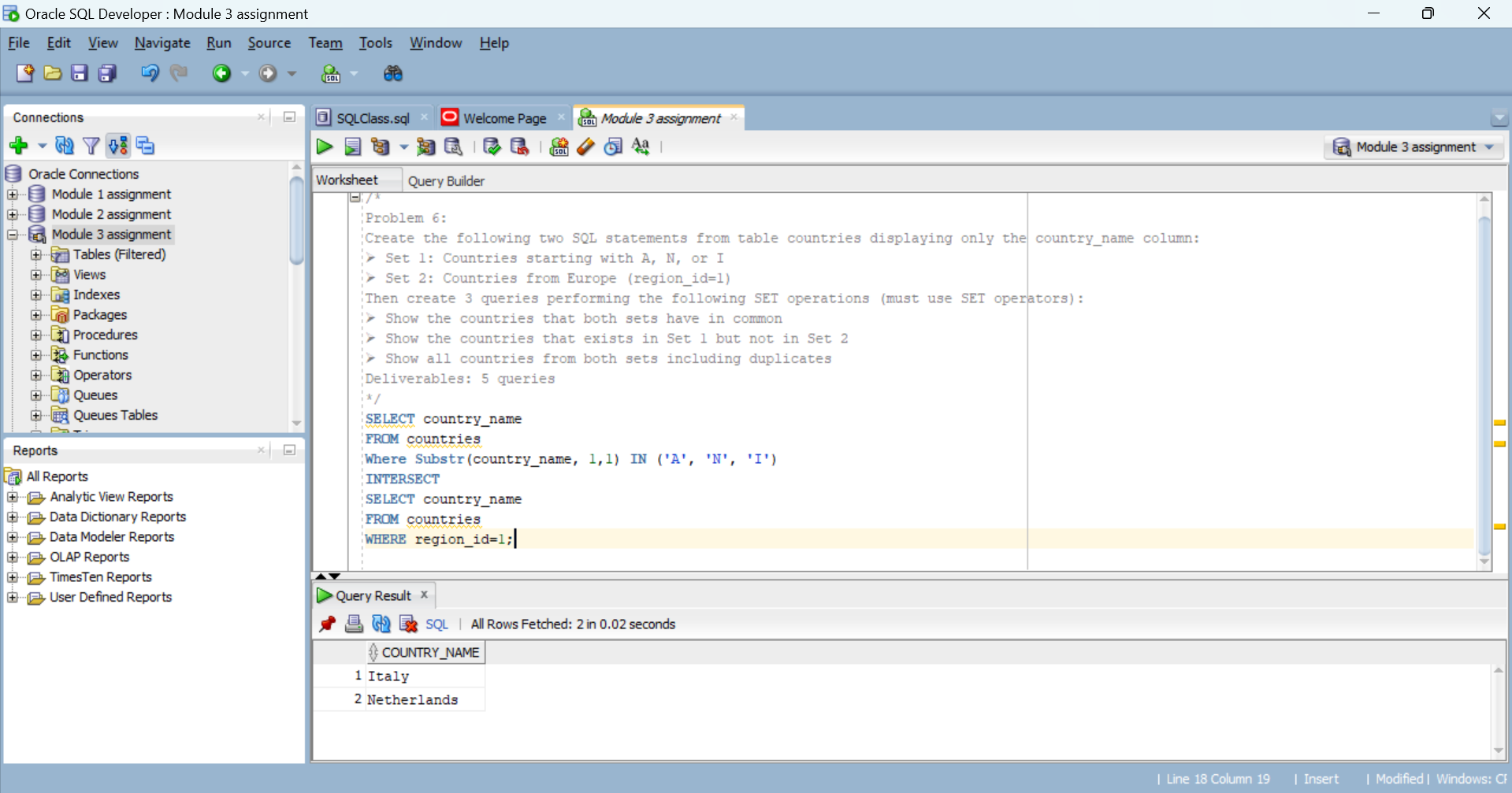
Where Substr(country\_name, 1,1) IN ('A', 'N', 'I')

INTERSECT

SELECT country\_name

FROM countries

WHERE region\_id=1;



4.

SELECT country\_name

FROM countries

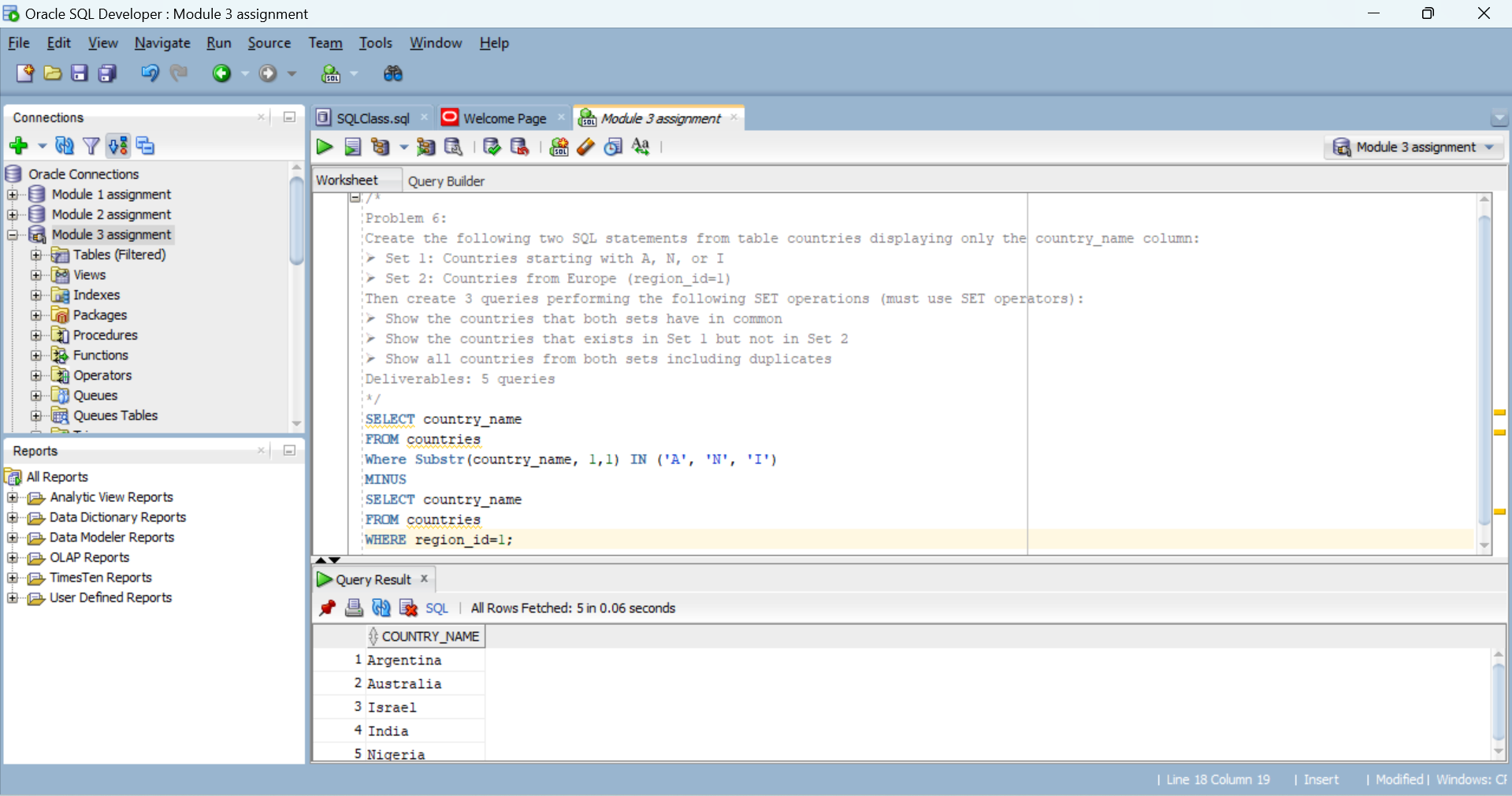
Where Substr(country\_name, 1,1) IN ('A', 'N', 'I')

MINUS

SELECT country\_name

FROM countries

WHERE region\_id=1;



5.

SELECT country\_name, 'table set 1' AS "Table\_Name"

FROM countries

Where Substr(country\_name, 1,1) IN ('A', 'N', 'I')

UNION ALL

SELECT country\_name, 'table set 2'

FROM countries

WHERE region\_id=1;

