 



Homework

9-3: Set Operators

Practice Activities

# Vocabulary

# UNION = operator that returns all rows from both tables and eliminates duplicates

# TO\_CHAR(NULL) sau TO\_DATE(NULL) sau TO\_NUMBER(NULL) = columns that were made up to match queries in another table that are not in both tables

# UNION ALL = operator that returns all rows from both tables, including duplicates

# SET OPERATIONS = used to combine results into one single result from multiple SELECT statements

# MINUS = operator that returns rows that are unique to each table

# INTERSECT = operator that returns rows common to both tables

# Try It / Solve It

1. Name the different Set operators?

UNION( si UNION ALL), INTERSECT, MINUS

1. Write one query to return the employee\_id, job\_id, hire\_date, and department\_id of all employees and a second query listing employee\_id, job\_id, start\_date, and department\_id from the job\_history table and combine the results as one single output. Make sure you suppress duplicates in the output.

SELECT employee\_id, job\_id, hire\_date, department\_id

FROM employees

UNION

SELECT employee\_id, job\_id, start\_date, department\_id

FROM job\_history

29 de randuri returnate

1. Amend the previous statement to not suppress duplicates and examine the output. How many extra rows did you get returned and which were they? Sort the output by employee\_id to make it easier to spot.

SELECT employee\_id, job\_id, hire\_date, department\_id

FROM employees

UNION ALL

SELECT employee\_id, job\_id, start\_date, department\_id

FROM job\_history

ORDER BY employee\_id

30 de randuri returnate

1. List all employees who have not changed jobs even once. (Such employees are not found in the job\_history table)

SELECT employee\_id, job\_id, hire\_date, department\_id

FROM employees

MINUS

SELECT employee\_id, job\_id, start\_date, department\_id

FROM job\_history

ORDER BY employee\_id

1. List the employees that HAVE changed their jobs at least once.

SELECT employee\_id, job\_id, hire\_date, department\_id

FROM employees

INTERSECT

SELECT employee\_id, job\_id, start\_date, department\_id

FROM job\_history

ORDER BY employee\_id

1. Using the UNION operator, write a query that displays the employee\_id, job\_id, and salary of ALL present and past employees. If a salary is not found, then just display a 0 (zero) in its place.

SELECT employee\_id, job\_id, hire\_date, department\_id, salary

FROM employees

UNION

SELECT employee\_id, job\_id, start\_date, department\_id, NVL(TO\_NUMBER(null), 0)

FROM job\_history

ORDER BY employee\_id