##########################################################

##########################################################

-- Guided Project: The SQL SELECT Statement

##########################################################

##########################################################

#############################

-- Task One: Introduction

-- In this task, retrieve data from the four tables in the database

#############################

-- Retrieve all data from the employees table

SELECT \* FROM employees;

-- Retrieve all data from the departments table

SELECT \* FROM departments;

-- Retrieve all data from the departments employees table

SELECT \* FROM dept\_emp;

-- Retrieve all data from the salaries table

SELECT \* FROM salaries;

#############################

-- Task Two: SELECT - FROM AND WHERE

-- In this task, you will retrieve data from tables as in task one. In addition,

-- you will learn how to set conditions on the result set of a query using the WHERE clause

#############################

##########

-- SELECT - FROM

-- 2.1: Select the first name and last name of all employees

SELECT first\_name, last\_name

FROM employees;

-- 2.2 Select all records from the employees table

SELECT \* FROM employees;

-- 2.3 (Ex.) Select the department number from the departments table

SELECT dept\_no FROM departments;

-- 2.4 (Ex.) Select all records from the departments table

SELECT \* FROM departments;

############

-- SELECT FROM - WHERE

-- 2.5 Select all employees whose first name is 'Elvis'

SELECT \*

FROM employees

WHERE first\_name = 'Elvis';

-- 2.6 (Ex.) Select all employees whose first name is 'Hilari'

SELECT \*

FROM employees

WHERE first\_name ='Hilari';

#############################

-- Task Three: SQL Operators

-- In this task, you will retrieve data from tables in the employees database,

-- using different SQL operators together with the WHERE clause

#############################

###########

-- AND

-- 3.1 Select all male employees whose first name is 'Denis'

SELECT \*

FROM employees

WHERE gender = 'M' and first\_name = 'Denis';

-- 3.2 (Ex.) Select all female employees whose last name is 'Reistad'

SELECT \*

FROM employees

WHERE gender = 'F' AND last\_name = 'Reistad';

###########

-- OR

-- Recall we have done this in 3.1 above

SELECT \*

FROM employees

WHERE first\_name = 'Denis' AND gender = 'M';

-- 3.3 Select all employees whose first name is 'Denis' or 'Elvis'

SELECT \*

FROM employees

WHERE first\_name = 'Denis' OR first\_name = 'Elvis';

-- Let us compare with this

-- Select all employees whose first name is 'Denis' and 'Elvis'

SELECT \*

FROM employees

WHERE first\_name = 'Denis' AND first\_name = 'Elvis';

-- 3.4 (Ex.) Retrieve a list of all employees whose first name is either

'Kellie' or 'Aruna'

SELECT \*

FROM employees

WHERE first\_name = 'Kellie' OR first\_name = 'Aruna';

###########

-- Operator Precedence

-- 3.5 Retrieve a list of all male or female employees whose last name is 'Denis'

SELECT \*

FROM employees

WHERE last\_name = 'Denis' AND gender = 'M' OR gender = 'F';

-- Correct Syntax

SELECT \*

FROM employees

WHERE last\_name = 'Denis' AND (gender = 'M' OR gender = 'F');

-- 3.6 (Ex.) Retrieve a list of all female employees whose first name is either

'Kellie' or 'Aruna'

SELECT \*

FROM employees

WHERE gender = 'F' AND (first\_name = 'Kellie' OR first\_name = 'Aruna');

###########

-- IN / NOT IN

-- 3.7: Retrieve a list of all employees whose first name is either 'Cathie', 'Mark' or 'Nathan'

SELECT \*

FROM employees

WHERE first\_name = 'Cathie' OR first\_name = 'Mark' OR first\_name = 'Nathan';

-- Alternative Solution

SELECT \*

FROM employees

WHERE first\_name IN ('Cathie', 'Mark', 'Nathan');

-- NOT IN

SELECT \*

FROM employees

WHERE first\_name NOT IN ('Cathie', 'Mark', 'Nathan')

#############################

-- Task Four: Wildcard Characters

-- In this task, you will retrieve data from tables in the employees database,

-- using different wildcard characters together with the WHERE clause

#############################

###########

-- LIKE / NOT LIKE

-- 4.1: Extract a list of all employees whose first name starts with 'Mar'

SELECT \*

FROM employees

WHERE first\_name LIKE ('Mar%');

-- 4.2: Extract a list of all employees whose first name starts with 'Ar'

SELECT \*

FROM employees

WHERE first\_name LIKE ('Ar%');

-- 4.3: Extract a list of all employees whose first name ends with 'Ar'

SELECT \*

FROM employees

WHERE first\_name LIKE ('%ar');

-- 4.4: What do you think the result of this query will be?

SELECT \*

FROM employees

WHERE first\_name LIKE('%ar%');

\*\*\*This query will populate employee first names that have 'ar' in the middle of their first name\*\*

-- 4.5: What do you think the output will be here?

SELECT \*

FROM employees

WHERE first\_name LIKE ('Mar\_');

\*\*\*This query will populate employee first names that start with 'Mar' that only has a single character after the 'Mar'\*\*

-- 4.6: What about this??

SELECT \*

FROM employees

WHERE first\_name NOT LIKE ('%Mar%');

\*\*\*This query will populate employee first names that do not have 'Mar' in the middle\*\*

-- 4.7 (Ex.): Extract all individuals from the 'employees' table whose first name

-- starts with 'Mark'

SELECT \*

FROM employees

WHERE first\_name LIKE ('Mark%');

-- 4.8 (Ex.): Extract all individuals from the employees table whose first name

-- does not contain 'Jack'

SELECT \*

FROM employees

WHERE first\_name NOT LIKE ('%Jack%');

#############################

-- Task Five: SQL Operators - Part 2

-- In this task, you will retrieve data from tables in the employees database,

-- using different SQL operators together with the WHERE clause

#############################

###########

-- BETWEEN - AND

-- 5.1: Extract a list of all employees who were employed between

-- 1st of January, 1990 and 1st January, 2000

SELECT \*

FROM employees

WHERE hire\_date BETWEEN '1990-01-01' AND '2000-01-01';

-- 5.2: Extract a list of all employees who were not employed between

-- 1st of January, 1990 and 1st January, 2000

SELECT \*

FROM employees

WHERE hire\_date NOT BETWEEN '1990-01-01' AND '2000-01-01';

-- 5.3 (Ex.): Select all the records from the salaries table, regarding contracts

-- from 66000 to 70000 dollars/year

-- Selecting all records from the salaries table

SELECT \* FROM salaries;

-- Solution to 5.3

SELECT \*

FROM salaries

WHERE salary BETWEEN '66000' AND '70000';

-- 5.4: Retrieve a list of individuals whose emp\_no is not between

-- 10004 and 10012.

SELECT \*

FROM employees

WHERE emp\_no NOT BETWEEN '10004' AND '10012';

###########

-- IS NOT NULL / IS NULL

-- 5.5: Extract a list of employees whose first name is not null

-- Let us count how many first names are in the employees table

SELECT COUNT(first\_name)

FROM employees;

-- Solution to 5.5

SELECT \*

FROM employees

WHERE first\_name IS NOT NULL;

-- 5.6: Extract a list of employees whose first name is null

SELECT \*

FROM employees

WHERE first\_name IS NULL;

-- 5.7 (Ex.): Select the names of all departments whose department number value is not null

SELECT dept\_name

FROM departments

WHERE dept\_no IS NOT NULL;

#############################

-- Task Six: Other Comparison Operators

-- In this task, you will retrieve data from tables in the employees database,

-- using other comparison operators together with the WHERE clause

#############################

###########

-- Equal to (=) & Not Equal to (<> or !=)

-- 6.1: Retrieve a list of all employees whose first name is 'Mark'

SELECT \*

FROM employees

WHERE first\_name = 'Mark';

-- 6.2: Retrieve a list of all employees whose first name is not 'Mark'

SELECT \*

FROM employees

WHERE first\_name <> 'Mark';

-- 6.3: Retrieve a list of all employees whose first name is not 'Mark'

SELECT \*

FROM employees

WHERE first\_name != 'Mark';

#########

-- Greater than & Less than

-- 6.4: Retrieve a list of all employees who were employed after 1st of January, 1997.

SELECT \*

FROM employees

WHERE hire\_date > '1997-01-01';

-- 6.5: Retrieve a list of all employees who were employed on or after 1st of January, 1997.

SELECT \*

FROM employees

WHERE hire\_date >= '1997-01-01';

-- 6.6: Retrieve a list of all employees who were employed before 1st of February, 1990

SELECT \*

FROM employees

WHERE hire\_date < '1990-02-01';

-- 6.7: What do you think the result will be??

SELECT \*

FROM employees

WHERE hire\_date <= '1990-02-01';

\*\*\*This query will populate employees hired before or on February 1, 1990\*\*

-- 6.8 (Ex.): Retrieve a list of all female employees who were hired on or after 1st January, 1999.

SELECT \*

FROM employees

WHERE gender = 'F' AND hire\_date >= '1999-01-01';

-- 6.9: Extract a list with employees' salaries higher than 50,000 dollars per annum

SELECT \*

FROM salaries

WHERE salary > 50000;

#############################

-- Task Seven: Select Distinct, Order BY, Limit, and SQL Aliases.

-- In this task, you will learn how to retrieve distinct data from tables in the employees database.

-- In addition, you will learn how to sort the result set, how to limit the result set and how to

-- give a temporary name to a column of a table.

#############################

###########

-- SELECT DISTINCT

-- 7.1: Select distinct gender from the employees table

-- Let us retrieve data from the gender column from the employees table

SELECT gender

FROM employees;

-- Solution to 7.1

SELECT DISTINCT gender

FROM employees;

-- 7.2: Retrieve a list of the different first names in the employees database

SELECT DISTINCT first\_name

FROM employees;

###########

-- ORDER BY

-- 7.3: Extract a list with employees' salaries higher than 50,000 dollars per annum

-- Sort the list by the most paid employee

SELECT \*

FROM salaries

WHERE salary > 50000

ORDER BY salary DESC;

-- 7.4: Select all records in the employees table and order by first name

SELECT \*

FROM employees

ORDER BY first\_name;

-- 7.5: Select all records and order by first name in descending order

SELECT \*

FROM employees

ORDER BY first\_name DESC;

-- 7.6: Select all records and order by first name and last name

-- in ascending order

SELECT \*

FROM employees

ORDER BY first\_name, last\_name ASC;

##########

-- Limit and SQL Aliases

-- 7.7: Retrieve a list of the first 100 records in the employees table

-- and order by emp\_no in ascending order

SELECT \*

FROM employees

ORDER BY emp\_no ASC

LIMIT 100;

-- From 5.5: We counted how many first names are in the employees table

SELECT COUNT(first\_name)

FROM employees;

-- 7.8: How many firstname are in the employees table. Rename the result as

-- FirstName\_Count

SELECT COUNT(first\_name) AS Firstname\_count

FROM employees;

-- Let's count how many distinct first names we have in the employees table

SELECT DISTINCT first\_name, COUNT(\*) AS Names\_Counts

FROM employees

GROUP BY first\_name

ORDER BY first\_name;