

**Discussed assignment with:**

In this assignment, you will be asked to write sentences and queries in Predicate Logic, SQL and Python. You will also be required to test your SQL and Python code on some data.

You will need to use PostgreSQL to implement your SQL code. And, you will need a Python programming language environment to implement your Python code.

Along with this assignment, there are two files

`'assignment4Script.sql'` and `'assignment4Script.py'`

that contain the data that you will need to use for this assignment.

You will need to submit 3 files:

1. A file with name `'assignment4.pdf'` that contains the solutions for problems indicated with a red bullet ●. This file should be generated using Latex;
2. A file with name `'assignment4.sql'` that contains the solutions for the problems requiring SQL code; and
3. A file with name `'assignment4.py'` that contains solutions for the problems requiring Python code.

Each problem or subproblem is worth 10 points.

1. • The domain of the following predicates is the set of all plants. Consider the following predicates:

$$\begin{aligned} P(x) &= \text{"}x \text{ is poisonous."} \\ Q(x) &= \text{"Jeff has eaten } x\text{."} \end{aligned}$$

Express the following sentences into Predicate Logic.

- (a) "Some plants are not poisonous."

$$\neg(\exists x P(x))$$

- (a) "Jeff has eaten a poisonous plant as well as a non-poisonous plant."

$$\exists x \exists y (P(x) \wedge \neg P(y) \wedge Q(x) \wedge Q(y))$$

- (a) "There are some non-poisonous plants that Jeff has not eaten."

$$\exists x (\neg P(x) \wedge \neg Q(x))$$

2. • The domain of the following predicates is the set of all books. Consider the following predicates:

$$\begin{aligned} H(x) &= \text{"}x \text{ is heavy."} \\ C(x) &= \text{"}x \text{ is confusing."} \end{aligned}$$

Express the following Predicate Logic sentences into ordinary English.

- (a)  $\forall x (H(x) \rightarrow C(x))$ .

Sol: if x is heavy, then x is confusing.

- (a)  $\neg \exists x (H(x) \wedge \neg C(x))$ .

Sol: No heavy object is not confusing

- (a)  $\forall x (H(x) \rightarrow (\exists x C(x)))$ .

Sol: Every heavy object is associated with at least one confusing object.

3. • Consider the predicates  $P(x)$ ,  $M(x)$ , and  $A(x, y)$  in a domain of people. The predicate  $P(x)$  states of a person that he or she is a philosopher, the predicate  $M(x)$  states of a person that he or she is a mathematician, and the predicate  $A(x, y)$  states that  $x$  is an admirer of  $y$ . Write a sentence in Predicate Logic for the following natural languages sentences.

- (a) Each philosopher admires a mathematician.

$$\forall x \forall y (P(x) \wedge M(y) \rightarrow A(x, y))$$

- (a) Not everyone admires a philosopher who is also a mathematician.

$$\neg \exists x (P(x) \wedge M(x) \wedge \forall y A(y, x))$$

- (a) Some person admires all philosophers but only some mathematicians.

$$\exists x \forall y (P(y) \rightarrow A(x, y)) \wedge \exists z \exists w (M(w) \wedge A(z, w) \wedge \neg \forall y (P(y) \rightarrow A(z, y)))$$

- (a) Some person admires a philosopher or a mathematician, but not both.

$$\exists x \exists y ((P(x) \wedge \neg M(y) \wedge A(x, y)) \vee (\neg P(x) \wedge M(y) \wedge A(x, y)))$$

4. • Reconsider the predicates of question 3.

Express the following queries in the Predicate Logic.

- (a) Find each person who admires a philosopher.

$$\exists x \exists y (P(y) \wedge A(x, y))$$

- (a) Find each philosopher who admires a mathematician who admires all philosophers.

$$\exists x \exists y \exists z (P(x) \wedge M(y) \wedge \forall w (P(w) \rightarrow A(y, w)) \wedge A(x, y) \wedge A(y, z))$$

- (a) Find each pair of persons  $(x, y)$  such that if  $x$  admires a philosopher then  $y$  also admires that philosopher.

$$\forall x \forall y ((\exists z (P(z) \wedge A(x, z))) \rightarrow (\exists w (P(w) \wedge A(y, w) \wedge \forall z (P(z) \rightarrow (A(x, z) \rightarrow A(y, z)))))$$

## Expressing sentences and queries in SQL and Python

Consider three domains ‘**P**’ (the domain of persons), ‘**C**’ (the domain of courses), and ‘**M**’ (the domain of majors).

- Variables that range over ‘**P**’ should be named  $p, p_1, p_2$ , etc.
- Variables that range over ‘**C**’ should be named  $c, c_1, c_2$ , etc.
- Variables that range over ‘**M**’ should be named  $m, m_1, m_2$ , etc.

Consider the unary predicates

$$\begin{aligned}\text{student}(p) &:= \text{“person } p \text{ is a student”} \\ \text{professor}(p) &:= \text{“person } p \text{ is a professor”}\end{aligned}$$

and the binary predicates

$$\begin{aligned}\text{hasMajor}(p, m) &\quad \text{“person } p \text{ has major } m\text{”} \\ \text{enroll}(p, c) &\quad \text{“person } p \text{ is enrolled in (i.e., takes) course } c\text{”} \\ \text{teaches}(p, c) &\quad \text{“person } p \text{ takes course } c\text{”} \\ \text{knows}(p_1, p_2) &\quad \text{“person } p_1 \text{ knows person } p_2\text{”}\end{aligned}$$

5. Consider the sentence “Some student knows a professor who teaches the ‘Databases’ course.” This sentence can be expressed in Predicate Logic as follows:

$$\exists p_1(\text{student}(p_1) \wedge \exists p_2(\text{professor}(p_2) \wedge \text{knows}(p_1, p_2) \wedge \text{teaches}(p_2, \text{Databases}))).$$

- (a) Express this Predicate Logic sentence as an SQL statement. Place your answer in the ‘assignment4.sql’ file.
- (b) Express this Predicate Logic sentence as a Python statement. Place your answer in the ‘assignment4.py’ file.
6. Consider the sentence “Each course taught by professor ‘Anna’ is taken by at least two students.” This sentence can be expressed in Predicate Logic as follows:

$$\begin{aligned}\forall c((\text{teaches}(c, \text{Anna}) \wedge \text{professor}(\text{Anna}) \rightarrow \\ \exists p_1 \exists p_2 (p_1 \neq p_2 \wedge \text{student}(p_1) \wedge \text{student}(p_2) \wedge \text{enroll}(p_1, c) \wedge \text{enroll}(p_2, c))).\end{aligned}$$

- (a) Express this Predicate Logic sentence as an SQL statement. Place your answer in the ‘assignment4.sql’ file.
- (b) Express this Predicate Logic sentence as a Python statement. Place your answer in the ‘assignment4.py’ file.
- (c) Rewrite the Predicate Logic sentence above into a logically equivalent Predicate Logic sentence in which only existential quantifiers ( $\exists$ ) appear. In other words, rewrite all the universal quantifiers. (Hint: use Replacement Laws of Predicate Logic for quantifiers and Predicate Logic sentences.)

- (d) Express the resulting Predicate Logic sentence as an SQL statement. Place your answer in the 'assignment4.sql' file.
  - (e) Express the resulting Predicate Logic sentence as a Python statement. Place your answer in the 'assignment4.py' file.
7. Consider the query "Find the majors of students who are enrolled in the course 'Algorithms'."
- (a) Express this query as an SQL query. Place your answer in the 'assignment4.sql' file.
  - (b) Express this query as a Python query (comprehension). Place your answer in the 'assignment4.py' file.
8. Consider the query "Find each student who knows a student who takes a taught by Professor 'Emma' or a course taught by professor 'Arif' or a course taught by professor 'Anna'."
- (a) Express this query as an SQL query. Place your answer in the 'assignment4.sql' file.
  - (b) Express this query as a Python query (comprehension). Place your answer in the 'assignment4.py' file.
9. Consider the query "Find each pair of different students who both know a same professor who teaches the course 'Databases'."
- (a) Express this query as an SQL query. Place your answer in the 'assignment4.sql' file.
  - (b) Express this query as a Python query (comprehension). Place your answer in the 'assignment4.py' file.
10. Consider the query "Find each professor who only teaches courses taken by all students who major in 'DataScience'."
- (a) Express this query as an SQL query. Place your answer in the 'assignment4.sql' file.
  - (b) Express this query as a Python query (comprehension). Place your answer in the 'assignment4.py' file.
11. Consider the query "Find each professor who does not know any student who majors in both 'DataScience' and in 'Chemistry'."
- (a) Express this query as an SQL query. Place your answer in the 'assignment4.sql' file.
  - (b) Express this query as a Python query (comprehension). Place your answer in the 'assignment4.py' file.
12. Consider the sentence "Find each pair of different students who have a common major and who take none of the courses taught by professor 'Pedro'."

- (a) Express the resulting sentence as an SQL statement. Place your answer in the 'assignment4.sql' file.
- (b) Express the resulting sentence as a Python statement. Place your answer in the 'assignment4.py' file.