2. Лингвистические переменные

ТАБЛИЦУ С ВЫЧИСЛЕНИЕМ ЛИНГВИСТИЧЕСКОЙ ПЕРЕМЕННОЙ ДЛЯ КАЖДОЙ МАШИНЫ

SELECT \* FROM rental\_price\_membership;

Для физической схемы базы данных библиотеки предложим три лингвистические переменные:

1. Цена проката автомобиля

Поскольку значения цен варьируются от 1000 до 4000, можно разделить их так:

* "Дешевая": от 1000 до 1300, с ядром [1000; 1200] и носителем [1000; 1400].
* "Средняя": от 1300 до 2500, с ядром [1500; 2300] и носителем [1300; 2700].
* "Дорогая": от 2500 до 4000, с ядром [3000; 3500] и носителем [2500; ∞].

2. Длительность аренды

На основе данных о длительности аренды, варьирующейся от 1 до 5 дней:

* "Короткая": от 1 до 2 дней, с ядром [1; 1.5] и носителем [1; 3].
* "Средняя": от 3 до 5 дней, с ядром [3; 4] и носителем [2.5; 5].
* "Долгая": больше 5 дней, с ядром [6; 7] и носителем [5; ∞].

3. Залоговая стоимость

Лингвистическая переменная для залоговой стоимости может содержать следующие значения:

* "Низкая": от 1000 до 1500, с ядром [1000; 1200] и носителем [1000; 1500].
* "Средняя": от 1300 до 2500, с ядром [1450; 2000] и носителем [1500; 3000].
* "Высокая": от 2500 до 4000, с ядром [2950; 3500] и носителем [2500; ∞].

3. SQL-реализация функций принадлежности

Для описания функций принадлежности будем использовать Z-образные, трапециевидные и S-образные функции. Вот реализация на SQL для каждой лингвистической переменной:

-- Z-образная функция принадлежности (например, для "дешевая" цена)

CREATE FUNCTION z\_fig(value DOUBLE PRECISION, a DOUBLE PRECISION, b DOUBLE PRECISION)

RETURNS DOUBLE PRECISION AS $$

BEGIN

IF value <= a THEN

RETURN 1;

ELSIF value >= b THEN

RETURN 0;

ELSE

RETURN (b - value) / (b - a);

END IF;

END;

$$ LANGUAGE plpgsql;

-- S-образная функция принадлежности (например, для "высокая" цена)

CREATE FUNCTION s\_fig(value DOUBLE PRECISION, a DOUBLE PRECISION, b DOUBLE PRECISION)

RETURNS DOUBLE PRECISION AS $$

BEGIN

IF value <= a THEN

RETURN 0;

ELSIF value >= b THEN

RETURN 1;

ELSE

RETURN (value - a) / (b - a);

END IF;

END;

$$ LANGUAGE plpgsql;

-- Трапециевидная функция принадлежности (например, для "средняя" цена)

CREATE FUNCTION trap(value DOUBLE PRECISION, a DOUBLE PRECISION, b DOUBLE PRECISION, c DOUBLE PRECISION, d DOUBLE PRECISION)

RETURNS DOUBLE PRECISION AS $$

BEGIN

IF value <= a OR value >= d THEN

RETURN 0;

ELSIF value >= b AND value <= c THEN

RETURN 1;

ELSIF value > a AND value < b THEN

RETURN (value - a) / (b - a);

ELSE

RETURN (d - value) / (d - c);

END IF;

END;

$$ LANGUAGE plpgsql;

4. Примеры нечетких запросов

Используя функции принадлежности, создадим несколько нечетких запросов:

1. Поиск автомобилей с дешёвой ценой проката:

SELECT \* FROM Cars

WHERE z\_fig(rental\_price, 1000, 1400) > 0.5;

2. Поиск автомобилей с средней ценой проката:

SELECT \* FROM Cars

WHERE trap(rental\_price, 1300, 1500, 2300, 2700) > 0.5;

3. Поиск автомобилей с высокой ценой проката:

SELECT \* FROM Cars

WHERE s\_fig(rental\_price, 2500, 4000) > 0.5;

4. Поиск аренд средней продолжительности:

SELECT r.\*, EXTRACT(DAY FROM AGE(return\_date\_expected, rental\_date)) AS duration

FROM Rentals r

WHERE trap(EXTRACT(DAY FROM AGE(return\_date\_expected, rental\_date)), 2.5, 3, 4, 5) > 0.5;

5. Поиск аренд длинной продолжительности.

SELECT r.\*, EXTRACT(DAY FROM AGE(return\_date\_expected, rental\_date)) AS duration

FROM Rentals r

WHERE s\_fig(EXTRACT(DAY FROM AGE(return\_date\_expected, rental\_date)), 5, 7) > 0.5;

5. Примеры нечетких запросов с модификацией лингвистических переменных

Для модификации лингвистических переменных, можно использовать такие операции, как возведение в степень (POWER) для усиления значения принадлежности (“очень”) или извлечение квадратного корня (SQRT) для ослабления значения (“более или менее”).

Вот несколько примеров нечетких запросов на основе таких модификаций:

1. Нечеткий запрос на аренду с "немного дешевая" стоимостью

SELECT \* FROM Cars

WHERE s\_fig(rental\_price, 1100, 1500) > 0.5;

2. Нечеткий запрос на "короткую" аренду с немного расширенным диапазоном

SELECT r.\*, EXTRACT(DAY FROM AGE(return\_date\_expected, rental\_date)) AS rental\_days FROM Rentals r

WHERE s\_fig(EXTRACT(DAY FROM AGE(return\_date\_expected, rental\_date)), 1, 3) < 0.6;

3. Нечеткий запрос на аренды с "средней и длинной" длительностью

SELECT r.\*, EXTRACT(DAY FROM AGE(return\_date\_expected, rental\_date)) AS rental\_days FROM Rentals r

WHERE trap(EXTRACT(DAY FROM AGE(return\_date\_expected, rental\_date)), 4, 5, 6, 8) > 0.5;

4. Нечеткий запрос на аренды с "чуть выше средней" стоимостью

SELECT \* FROM Cars

WHERE trap(rental\_price, 2000, 2300, 2500, 2800) > 0.5;

5. Нечеткий запрос на аренды с "весьма длинной" продолжительностью

SELECT r.\*, EXTRACT(DAY FROM AGE(return\_date\_expected, rental\_date)) AS rental\_days

FROM Rentals r

WHERE z\_fig(EXTRACT(DAY FROM AGE(return\_date\_expected, rental\_date)), 7, 13) > 0.4

AND EXTRACT(DAY FROM AGE(return\_date\_expected, rental\_date)) >= 5;

6. Выбор параметра для индекса соответствия (CI) и нечеткие запросы

Для создания нечетких запросов с учетом индекса соответствия (CI), выберем параметр, который рассчитывает минимальную степень соответствия между несколькими нечеткими критериями с помощью функции LEAST:

1. Создание функции CI:

CREATE FUNCTION calculate\_ci(degree1 DOUBLE PRECISION, degree2 DOUBLE PRECISION)

RETURNS DOUBLE PRECISION AS $$

BEGIN

RETURN LEAST(degree1, degree2);

END;

$$ LANGUAGE plpgsql;

2. Поиск автомобилей с низкой ценой проката:

SELECT c.\*, r.\* FROM Cars c

JOIN Rentals r ON c.car\_id = r.car\_id

WHERE z\_fig(c.rental\_price, 1000, 1300) > 0.6;

3. Поиск автомобилей средней ценовой категории с длительной арендой:

SELECT c.\*, r.\* FROM Cars c

JOIN Rentals r ON c.car\_id = r.car\_id

WHERE z\_fig(c.rental\_price, 1300, 2500) > 0.4

AND z\_fig(EXTRACT(DAY FROM AGE(r.return\_date\_expected, r.rental\_date)), 5, 10) > 0.6;

4. Поиск дорогих автомобилей с короткой арендой:

SELECT c.\*, r.\* FROM Cars c

JOIN Rentals r ON c.car\_id = r.car\_id

WHERE z\_fig(c.rental\_price, 2500, 4000) > 0.4

AND z\_fig(EXTRACT(DAY FROM AGE(r.return\_date\_expected, r.rental\_date)), 1, 3) > 0.4;

5. Поиск средних автомобилей с длинной арендой

SELECT c.\*, r.\* FROM Cars c

JOIN Rentals r ON c.car\_id = r.car\_id

WHERE z\_fig(c.rental\_price, 1300, 2500) > 0.4

AND z\_fig(EXTRACT(DAY FROM AGE(r.return\_date\_expected, r.rental\_date)), 5, 10) > 0.4;

6. Поиск дорогих автомобилей с краткосрочной арендой

SELECT c.\*, r.\* FROM Cars c

JOIN Rentals r ON c.car\_id = r.car\_id

WHERE z\_fig(c.rental\_price, 2500, 4000) > 0.4

AND z\_fig(EXTRACT(DAY FROM AGE(r.return\_date\_expected, r.rental\_date)), 1, 3) > 0.4;