



ДЕДУКТИВНЫЕ СИСТЕМЫ БАЗ ДАнных

Логика – это искусство приходить к непредсказуемому выводу.

С. Джонсон

Пост-реляционные системы баз данных

Содержание

- Модельно-теоретический и доказательно-теоретический подход к технологиям баз данных
- Исчисление высказываний и исчисление предикатов
- Дедуктивные базы данных и СУБД
- Язык запросов Datalog
- Обработка рекурсивных запросов

Системы баз данных: логика

3

- Модельно-теоретическое (традиционное) восприятие систем баз данных
 - База данных – множество явно заданных отношений, каждое из которых содержит множество явно заданных кортежей.
 - Выполнение запроса – вычисление некоторой заданной формулы с этими явно заданными отношениями и кортежами.
- Доказательно-теоретическое восприятие систем баз данных
 - База данных – множество аксиом.
 - Основные аксиомы: кортежи отношений.
 - Дедуктивные аксиомы: правила вывода.
 - Выполнение запроса – доказательство того, что некоторая формула является логическим следствием аксиом (доказательство того, что она является теоремой).

Системы баз данных: логика

4

- Модельно-теоретическое (традиционное) восприятие систем баз данных
 - ▣ `select * from SP where SP.Qty>20;`
 - Сканировать отношение SP.
 - Вычислить для каждого кортежа формулу $Qty > 20$.
 - Записать в результат те кортежи, для которых формула истинна.
- Доказательно-теоретическое восприятие систем баз данных
 - ▣ `exists SPX where SPX.Qty>20`
 - Рассмотреть кортежи SP как аксиомы некоторой логической теории
 - Применить методы доказательства теорем и выяснить, для каких возможных значений переменной области значений SPX формула $SPX.QTY > 20$ является логическим следствием этих аксиом в теории.
 - Записать в результат эти конкретные значения SPX.

Основные и дедуктивные аксиомы

5

□ Основные аксиомы

▣ Мать-Дочь

Мать	Дочь
Анна	Галина
Галина	Марина

□ Дедуктивные аксиомы

▣ если $\text{Мать}(x,y)$ и $\text{Мать}(y,z)$ то $\text{Бабушка}(x,z)$

▣ если $\text{Мать}(x,y)$ и $\text{Мать}(y,z)$ то $\text{Внучка}(z,x)$

▣ create view Бабушка-Внучка as
select A.Мать as Бабушка, B.Дочь as Внучка
from Мать-Дочь as A, Мать-Дочь as B
where A.Дочь=B.Мать

Исчисление высказываний

6

- Распределительные законы.
 - ▣ $f \text{ AND } (g \text{ OR } h) \equiv (f \text{ AND } g) \text{ OR } (f \text{ AND } h)$
 - ▣ $f \text{ OR } (g \text{ AND } h) \equiv (f \text{ OR } g) \text{ AND } (f \text{ OR } h)$
- Законы де Моргана.
 - ▣ $\text{NOT } (f \text{ AND } g) \equiv (\text{NOT } f) \text{ OR } (\text{NOT } g)$
 - ▣ $\text{NOT } (f \text{ OR } g) \equiv (\text{NOT } f) \text{ AND } (\text{NOT } g)$

Термы

7

- Терм – утверждение, которое включает литералы и:
 - ▣ либо не включает никаких логических операторов, либо заключено в круглые скобки;
 - ▣ в результате вычисления принимает значение либо TRUE, либо FALSE.
- Примеры термов:
 - ▣ Поставщик S1 находится в Лондоне.
 - ▣ Поставщик S1 поставляет деталь P1.

Формулы

8

- Формула ::=
 - терм |
 - NOT терм |
 - NOT терм AND формула |
 - терм \Rightarrow формула
- Терм ::=
 - атомарная формула |
 - (формула)
- $f \Rightarrow g ::= \text{NOT } a \text{ OR } g$

Правила вывода

9

- $\models (f \text{ AND } g) \Rightarrow f$
- $\models f \Rightarrow (f \text{ OR } g)$
- $\models ((f \Rightarrow g) \text{ AND } (g \Rightarrow h)) \Rightarrow (f \Rightarrow)$
- Modus ponens
 $\models (f \text{ AND } (f \Rightarrow g)) \Rightarrow g$
- $\models (f \Rightarrow (g \Rightarrow h)) \Rightarrow ((f \text{ AND } g) \Rightarrow h)$
- Правило резолюции
 $\models ((f \text{ OR } g) \text{ AND } (\text{NOT } g \text{ OR } h)) \Rightarrow (f \text{ OR } h)$

Доказательства

10

- Определение того, является ли формула g (заключение) логическим следствием из некоторого заданного множества формул f_1, f_2, \dots, f_n (предпосылок).
 - $f_1, f_2, \dots, f_n \vdash g$
- Подходы к доказательству
 - Принятие предпосылки.
 - Если g имеет форму $p \Rightarrow q$, принять p в качестве дополнительной предпосылки и показать, что q может быть дедуктивно выведена из заданных предпосылок и p .
 - Обратный логический вывод.
 - Вместо попытки доказать, что $p \Rightarrow q$, доказать противоположное утверждение, $\text{NOT } q \Rightarrow \text{NOT } p$.
 - Приведение к абсурду.
 - Вместо того, чтобы доказывать непосредственно, что $p \Rightarrow q$, принять предположение, что p и $\text{NOT } q$ одновременно являются истинными, и вывести из этого противоречие.
 - Резолюция.
 - Используется правило логического вывода на основе резолюции:
 $\vdash ((f \text{ OR } g) \text{ AND } (\text{NOT } g \text{ OR } h)) \Rightarrow (f \text{ OR } h)$

Пример резолюции

11

- Определить является ли допустимым следующее доказательство:
 - ▣ $A \Rightarrow (B \Rightarrow C)$, $\text{NOT } D \text{ OR } A, B \vdash D \Rightarrow C$
- Примем отрицание заключения в качестве дополнительной предпосылки и запишем каждую предпосылку на отдельной строке:
 - ▣ $A \Rightarrow (B \Rightarrow C)$
 - ▣ $\text{NOT } D \text{ OR } A$
 - ▣ B
 - ▣ $\text{NOT } (D \Rightarrow C)$
- Преобразуем каждую отдельную строку в КНФ
 - ▣ $\text{NOT } A \text{ OR } \text{NOT } B \text{ OR } C$
 - ▣ $\text{NOT } D \text{ OR } A$
 - ▣ B
 - ▣ $D \text{ AND } \text{NOT } C$

Пример резолюции

12

- Каждая строка, включающая операторы AND, заменяется множеством отдельных строк (по одной для каждой формулы), соединенных оператором AND (в ходе этого процесса операторы AND уничтожаются).
 - NOT A OR NOT B OR C
 - NOT D OR A
 - B
 - D
 - NOT C

Пример резолюции

13

- Применение резолюции для пар строк, одна из которых содержит формулу, а другая ее отрицание.
 - ▣ NOT D OR NOT B OR C
 - ▣ B
 - ▣ D
 - ▣ NOT C
- Применение резолюции.
 - ▣ NOT D OR C
 - ▣ D
 - ▣ NOT C
- Применение резолюции.
 - ▣ C
 - ▣ NOT C
- Приведением к абсурду показан результат ложности исходного доказательства.

Исчисление предикатов

14

- Предикат – булева функция с параметрами, содержащая имена переменных и кванторы существования и всеобщности.
 - $S(S\#, Name, Rating, City)$ – 4-местный предикат, $S(S1, Bond, 7, London)$ – вызов предиката.
 - $\exists P(P\#, Name, Price) \exists S(S1, Name, Rating, City) \exists SP(P\#, S1, Qty)$

Правильные формулы

15

- $\langle \text{wff-формула} \rangle ::=$
 - терм
 - $| \text{NOT} (\langle \text{wff-формула} \rangle)$
 - $| (\langle \text{wff-формула} \rangle) \text{AND} (\langle \text{wff-формула} \rangle)$
 - $| (\langle \text{wff-формула} \rangle) \text{OR} (\langle \text{wff-формула} \rangle)$
 - $| (\langle \text{wff-формула} \rangle) \Rightarrow (\langle \text{wff-формула} \rangle)$
 - $| \text{EXISTS} \langle \text{переменная} \rangle (\langle \text{wff-формула} \rangle)$
 - $| \text{FORALL} \langle \text{переменная} \rangle (\langle \text{wff-формула} \rangle)$
- $\langle \text{терм} \rangle ::= [\text{NOT}] \text{ предикат} [(\text{список параметров})]$

Обработка логических запросов

16

□ Исходные аксиомы

1. Мать(Анна, Галина)
2. Мать(Галина, Марина)
3. Мать(x, y) AND Мать(y, z) \Rightarrow Бабушка(x, z)
4. NOT Мать(x, y) OR NOT Мать(y, z) OR Бабушка(x, z)

□ Докажем, что Анна – бабушка Марины. Начнем с отрицания доказываемого заключения:

5. NOT Бабушка(Анна, Марина).

□ Чтобы применить правило резолюции, необходимо систематически подставлять вместо переменных конкретные значения таким образом, чтобы можно было найти, соответственно, правильно построенную формулу и ее отрицание. Подставим в 4. Анна вместо x и Марина вместо z и выполним резолюцию:

6. NOT Мать(Анна, y) OR NOT Мать(y , Марина)

Обработка логических запросов

17

□ АКСИОМЫ

1. Мать(Анна, Галина)
 2. Мать(Галина, Марина)
 3. Мать(х,у) AND Мать(у,з) \Rightarrow Бабушка(х,з)
 4. NOT Мать(х,у) OR NOT Мать(у,з) OR Бабушка(х,з)
 5. NOT Бабушка(Анна, Марина).
 6. NOT Мать(Анна,у) OR NOT Мать(у,Марина)
- Подставим в 6. вместо у Галина из 2. и выполним резолюцию:
7. NOT Мать(Анна, Галина)
- Противоречие с 1.

Обработка логических запросов

18

- Кто является внучкой Анны (чьей бабушкой является Анна)?

Аксиомы

1. Мать(Анна, Галина)
 2. Мать(Галина, Марина)
 3. NOT Мать(x,y) OR NOT Мать(y,z) OR Бабушка(x,z)
 4. NOT Бабушка(Анна, r) OR Результат(r).
- Подставим в 3. Анна вместо x и r вместо z и выполним резолюцию 3. и 4.:
 5. NOT Мать(Анна,y) OR NOT Мать(y,z) OR Результат(z)
 - Подставим в 3. Анна вместо y и выполним резолюцию 5. и 1.:
 6. NOT Мать(Галина, z) OR Результат(z)
 - Подставим в 6. Марина вместо z и выполним резолюцию 6. и 2.:
 7. Результат(Марина)

Дедуктивные СУБД

19

- Дедуктивная СУБД – СУБД, поддерживающая доказательно-теоретический подход к базам данных.
- Дедуктивная база данных состоит из
 - экстенциональной базы данных – основных аксиом, соответствующих элементам домена и кортежам отношений;
 - интенциональной базы данных – ограничений целостности и дедуктивных аксиом.

Экстенсиональная база данных

20

□ Аксиомы доменов

- ▣ $S\# (S1), S\# (S2), S\# (S3), \dots$
- ▣ $Name (Adams), Name (Bond), Name (Clark), \dots$
- ▣ $Rating (5), Rating (10), Rating (15), \dots$
- ▣ $City (London), City (New York), City (Moscow), \dots$

□ Аксиомы отношений

- ▣ $S (S1, Adams, 10, New York), S (S7, Bond, 7, London), \dots$
- ▣ $P (P1, \dots), P (P2, \dots), \dots$
- ▣ $SP (S1, P1, 200), SP (S7, P2, 7), \dots$

Интенсиональная база данных

21

- Ограничения домена
 - $S(s, name, rating, city) \Rightarrow S\#(s) \text{ AND Name}(name) \text{ AND Rating}(rating) \text{ AND City}(city)$
 - $P(p, name, price) \Rightarrow P\#(p) \text{ AND Name}(name) \text{ AND Price}(price)$
- Ограничения потенциального ключа
 - $P(p1, name1, price1) \text{ AND } P(p2, name2, price2) \Rightarrow P\#(p1=p2) \text{ AND } (name1=name2) \text{ AND } (price1=price2)$
- Ограничения внешнего ключа
 - $SP(s, p, qty) \Rightarrow S(s, name, rating, city) \text{ AND } P(p, name, price)$
- Дедуктивные аксиомы
 - $S(s, name, rating, city) \text{ AND } rating > 15 \Rightarrow \text{GoodSupplier}(s, name, city)$
 - $S(s1, name1, rating1, city) \text{ AND } S(s2, name2, rating2, city) \Rightarrow \text{CoLocated}(s1, s2)$

Язык запросов Datalog

22

- Получить список надежных поставщиков
 - ▣ ? \Leftarrow GoodSupplier (s, name, city)
- Получить список надежных поставщиков из Лондона
 - ▣ ? \Leftarrow GoodSupplier (s, name, London)
- Является ли поставщик S1 надежным?
 - ▣ ? \Leftarrow GoodSupplier (s1, name, city)

Обработка рекурсивных запросов

23

- Описание структуры детали: какие детали pX включают другие детали pY в качестве непосредственных компонентов.
 - ▣ $PS(pX, pY) \Rightarrow$
 $P(pX, nameX, priceX) \text{ AND } P(pY, nameY, priceY)$
 - ▣ $PS(P1, P2), PART_STRUCT(P1, P3), PS(P2, P3), PS(P2, P4).$
- Описание структуры детали: какие детали включают другие детали в качестве компонентов на любом уровне.
 - ▣ $PS(pX, pY) \Rightarrow COMP(pX, pY)$
 - ▣ $PS(pX, pZ) \text{ AND } COMP(pZ, pY) \Rightarrow COMP(pX, pY)$

Обработка рекурсивных запросов

24

- Дедуктивные аксиомы на Datalog
 - ▣ $\text{COMP} (p_x, p_y) \Leftarrow \text{PS} (p_x, p_y)$
 - ▣ $\text{COMP} (p_x, p_y) \Leftarrow \text{PS} (p_x, p_z) \text{ AND } \text{COMP} (p_z, p_y)$
- Запрос на Datalog: Показать компоненты детали P1.
 - ▣ $? \Leftarrow \text{COMP} (P1, p_y)$

Унификация и резолюция

25

1. NOT PS (px, py) OR COMP (px, py)
 2. NOT PS (px, pz) OR NOT COMP (pz, py)
OR COMP (px, py)
 3. NOT COMP (P1, py) OR RESULT (py)
 4. PS (P1, P2)
 5. После подстановки P1 вместо px и P2 вместо py в строке 1 можно применить правило резолюции к строкам 1 и 4, получив COMP (P1, P2)
 6. После подстановки P2 вместо py в строке 3 и применения правила резолюции к строкам 3 и 5 получим RESULT (P2)
- Т.е. деталь P2 является компонентом детали P1.
 - Аналогично можно показать, что деталь P3 также является компонентом детали P1. Таким образом, получены две дополнительные аксиомы COMP (P1, P2) и COMP (P1, P3).
 - Теперь можно рекурсивно применить описанный выше процесс для определения всего состава компонентов.

<u>PS</u>	PX	PY
	P1	P2
	P1	P3
	P2	P3
	P2	P4
	P3	P5
	P4	P5
	P5	P6

Статическое фильтрование

26

□ Алгоритм:

```
NEW := PS where PX = 'P1';
```

```
COMP := NEW;
```

```
while not isEmpty(NEW) do begin
```

```
  // COMP ⋈ PS ::=
```

```
  // (COMP rename PY as Attr)
```

```
  // join
```

```
  // (PS rename PX as Attr) [COMP.PX, PS.PY]
```

```
  NEW := (COMP ⋈ PS) minus COMP;
```

```
  COMP := COMP union NEW;
```

```
end;
```

```
Result := COMP;
```

<u>PS</u>	PX	PY
	P1	P2
	P1	P3
	P2	P3
	P2	P4
	P3	P5
	P4	P5
	P5	P6

Статическое фильтрование

27

□ Алгоритм:

```
NEW := PS where PX = 'P1';
```

```
COMP := NEW;
```

```
while not isEmpty(NEW) do begin
```

```
  // COMP ⋈ PS ::=
```

```
  // (COMP rename PY as Attr)
```

```
  // join
```

```
  // (PS rename PX as Attr) [COMP.PX, PS.PY]
```

```
    NEW := (COMP ⋈ PS) minus COMP;
```

```
    COMP := COMP union NEW;
```

```
end;
```

```
Result := COMP;
```

<u>NEW</u>	PX	PY	<u>COMP</u>	PX	PY
	P1	P2		P1	P2
	P1	P3		P1	P3

<u>NEW</u>	PX	PY
	P1	P4
	P1	P5

<u>COMP</u>	PX	PY
	P1	P2
	P1	P3
	P1	P4
	P1	P5

Статическое фильтрование

28

□ Алгоритм:

`NEW := PS where PX = 'P1';`

`COMP := NEW;`

`while not isEmpty(NEW) do begin`

`// COMP ⋈ PS ::=`

`// (COMP rename PY as Attr)`

`// join`

`// (PS rename PX as Attr) [COMP.PX, PS.PY]`

`NEW := (COMP ⋈ PS) minus COMP;`

`COMP := COMP union NEW;`

`end;`

`Result := COMP;`

<u>NEW</u>	PX	PY	<u>COMP</u>	PX	PY
	P1	P4		P1	P2
	P1	P5		P1	P3
				P1	P4
				P1	P5

<u>NEW</u>	PX	PY
	P1	P6

<u>COMP</u>	PX	PY
	P1	P2
	P1	P3
	P1	P4
	P1	P5
	P1	P6

Заключение

- Доказательно-теоретическое восприятие систем баз данных:
 - База данных – множество основных и дедуктивных аксиом.
 - Выполнение запроса – доказательство того, что некоторая формула является логическим следствием аксиом.
- Дедуктивная СУБД – СУБД, поддерживающая доказательно-теоретический подход к базам данных.
- Дедуктивная база данных состоит из экстенциональной и интенциональной частей.
- Datalog – язык запросов к дедуктивным базам данных.
- Обработка рекурсивных запросов может выполняться на базе техники резолюции (менее эффективно) и статического фильтрования (более эффективно).