

ỦY BAN NHÂN DÂN TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN



NGUYỄN ĐÌNH SANG - 3117410210

TRẦN VIỆT THANH HẢI - 3117410063

TÊN ĐỀ TÀI

**HỆ QUẢN TRỊ CSDL QUAN HỆ XÁC SUẤT VỚI GIÁ TRỊ
THUỘC TÍNH KHÔNG CHẮC CHẮN**

KHÓA LUẬN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

TRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC

GV HƯỚNG DẪN: PGS.TS. NGUYỄN HÒA

TP. HỒ CHÍ MINH, THÁNG 4 NĂM 2022

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi, các số liệu và kết quả nghiên cứu nêu trong luận văn là trung thực, được các đồng tác giả cho phép sử dụng và chưa từng được công bố trong bất kì một công trình nào khác.

Thành phố Hồ Chí Minh, ngày 01 tháng 06 năm 2022

Tác giả luận văn

Nguyễn Đình Sang

Trần Viết Thanh Hải

LỜI CẢM ƠN

Chặng đường SGU của chúng em đang dần khép lại, những giây phút này ta mới trân trọng khoảng thời gian được học tập ở trường, những khó khăn, thử thách và đặc biệt là những người bạn, quý thầy cô, những người đã cùng chúng em vượt qua tất cả. Có lẽ đây là lần cuối được học tập, trao đổi và làm bài cùng nhau, được quý thầy cô hướng dẫn. Chúng em mong muốn khép lại chặng đường này của mình một cách trọn vẹn nhất bằng luận văn này. Chúng em Nguyễn Đình Sang và Trần Viết Thanh Hải may mắn khi được học tập tại khoa Công Nghệ Thông Tin, trường Đại học Sài Gòn, và may mắn hơn khi được khoa cho phép và tạo điều kiện làm khóa luận.

Chúng em xin gửi đến PGS. TS. Nguyễn Hòa lời cảm ơn chân thành và sâu sắc nhất, thầy đã tận tình hướng dẫn cũng như cung cấp tài liệu, thông tin và tạo điều kiện thuận lợi nhất cho chúng em trong suốt thời gian thực hiện khóa luận.

Chúng em xin cảm ơn quý thầy, cô của khoa Công Nghệ Thông Tin, trường Đại học Sài Gòn đã tận tình giảng dạy, truyền đạt những kiến thức quý báu trong suốt những năm học, đã cung cấp nền tảng kiến thức vững chắc cho chúng em nghiên cứu và thực hiện khóa luận.

Xin chân thành cảm ơn lãnh đạo, ban giám hiệu, khoa Công Nghệ Thông Tin, cùng toàn thể quý thầy, cô giáo trường Đại học Sài Gòn, đã tạo điều kiện cho chúng em trong quá trình học tập tại trường cũng như qua trình hoàn thành khóa luận này.

Chúng em đã cố gắng để thực hiện khóa luận một cách hoàn chỉnh nhất, vì nhiều lý do khác nhau cũng như hạn chế về kiến thức, kinh nghiệm và thời gian nên khóa luận của chúng em không thể tránh khỏi những hạn chế và thiếu sót. Chúng em rất mong nhận được góp ý của quý thầy, cô để khóa luận được hoàn chỉnh hơn, chúng em sẽ ghi nhận và tiếp tục cố gắng để hoàn thiện sản phẩm.

Chúng em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, ngày 01 tháng 06 năm 2022

Sinh viên

Nguyễn Đình Sang

Trần Viết Thanh Hải

TÓM TẮT

Thực tế đã cho thấy, cơ sở dữ liệu (CSDL) quan hệ truyền thống là rất hiệu quả để mô hình hóa, thiết kế và hiện thực các hệ thống lớn. Tuy nhiên, mô hình CSDL quan hệ truyền thống không biểu diễn và xử lý được thông tin không rõ ràng và không chính xác của các đối tượng trong thực tế. Điều này đã đòi hỏi và thúc đẩy việc nghiên cứu và phát triển các mô hình CSDL quan hệ mờ và xác suất để có thể biểu diễn và xử lý được thông tin không rõ ràng và không chính xác. Tuy nhiên, khó có mô hình nào có thể biểu diễn và xử lý hết mọi khía cạnh không rõ ràng và không chính xác về thông tin của các đối tượng trong thế giới thực. Điều này là do độ phức tạp về lý thuyết khi phát triển mô hình hoặc sự không hiệu quả về ứng dụng nếu có một mô hình như vậy. Do đó, các mô hình CSDL quan hệ mờ và xác suất vẫn được tiếp tục nghiên cứu và phát triển để đáp ứng các mục tiêu ứng dụng khác nhau.

Khóa luận tốt nghiệp này mở rộng mô hình và hệ quản trị CSDL quan hệ truyền thống do Codd đề nghị năm 1970 thành một mô hình và hệ quản trị *CSDL quan hệ xác suất với giá trị thuộc tính không chắc chắn* (Probabilistic Relational Data Base with Uncertain Attribute Values-URDB) cho phép biểu diễn và truy vấn thông tin không chắc chắn và không rõ ràng của các đối tượng trong thực tế với hai đặc tính chính: (1) các quan hệ thể hiện tập các bộ dữ liệu là quan hệ xác suất với giá trị thuộc tính không chắc chắn được biểu diễn bởi các bộ ba xác suất mở rộng; (2) Hệ quản trị CSDL với ngôn ngữ truy vấn tựa SQL thân thiện có thể xử lý và thao tác thông tin không chắc chắn trong thực tế được biểu diễn bởi các quan hệ trong URDB.

Một diễn dịch xác suất các quan hệ hai ngôi trên các tập hợp được giới thiệu dựa trên lý thuyết xác suất làm cơ sở để phát triển mô hình dữ liệu và các phép toán đại số quan hệ trong URDB. Một tập các tính chất của các phép toán đại số quan hệ xác suất cũng được phát biểu và chứng minh như những mở rộng các tính chất của các phép toán đại số quan hệ trong mô hình cơ sở dữ liệu quan hệ truyền thống. Hệ quản trị cho URDB được hiện thực với ngôn ngữ truy vấn tựa SQL dựa trên hệ quản trị mã nguồn mở SQLite, gọi là URDB-SQLite, cho thấy triển vọng và khả năng ứng dụng của URDB để mô hình hóa và thao tác các quan hệ xác suất trong thực tế.

MỤC LỤC

LỜI CAM ĐOAN	2
LỜI CẢM ƠN	3
MỤC LỤC	6
DANH MỤC CÁC BẢNG	9
DANH MỤC CÁC HÌNH	10
DANH SÁCH CÁC CỤM TỪ VIẾT TẮT	11
MỞ ĐẦU	12
Chương 1 TỔNG QUAN VỀ MÔ HÌNH CSDL QUAN HỆ	18
1.1. Giới thiệu	18
1.2. Mô hình dữ liệu	18
1.3. Các phép toán đại số	20
1.4. Tính chất các phép toán đại số	22
1.5. Kết luận	23
Chương 2 CƠ SỞ TOÁN HỌC CỦA MÔ HÌNH URDB	24
2.1. Giới thiệu	24
2.2. Các chiến lược kết hợp các khoảng xác suất	25
2.3. Các hàm phân bố và bộ ba xác suất	28
2.4. Các chiến lược kết hợp các bộ ba xác suất	30
2.5. Diễn dịch xác suất của quan hệ trên các tập hợp	31
2.6. Kết luận	32
Chương 3 LƯỢC ĐỒ VÀ QUAN HỆ CỦA MÔ HÌNH URDB	33
3.1. Giới thiệu	33
3.2. Mô hình ý niệm	33
3.3. Thuộc tính quan hệ	35
3.4. Kiểu và giá trị	36
3.5. Lược đồ và quan hệ	38
3.6. Kết luận	42
Chương 4 CÁC PHÉP TOÁN ĐẠI SỐ TRÊN URDB	43
4.1. Giới thiệu	43
4.2. Phép chọn	43
4.3. Phép chiếu	48
4.4. Phép tích Descartes	49
4.5. Phép kết tự nhiên	50
4.6. Phép giao, hợp và trừ	51
4.7. Tính chất của các phép toán đại số	54

4.8.	Kết luận.....	58
Chương 5	HIỆN THỰC HỆ QUẢN TRỊ CỦA MÔ HÌNH URDB.....	59
5.1.	Giới thiệu	59
5.2.	Các tính năng đặc trưng của SQLite	59
5.2.1.	Tổng quan	59
5.2.2.	Các tính năng đặc trưng của SQLite	60
5.2.3.	Các đối tượng và phương thức chính trong SQLite.Net	61
5.3.	Thiết kế tổng quan hệ quản trị URDB-SQLite.....	62
5.4.	Kiến trúc tổng quan của hệ quản trị URDB-SQLite	64
5.5.	Hiện thực khối biểu diễn mô hình URDB.....	65
5.5.1.	Lớp ProbSchema.....	65
5.5.2.	Lớp ProbRelation.....	66
5.5.3.	Lớp ProbAttribute	67
5.5.4.	Lớp ProbDataType.....	67
5.5.5.	Lớp ProbTuple	68
5.5.6.	Lớp ProbTriple.....	68
5.5.7.	Lớp ValueOfTriple.....	69
5.6.	Hiện thực nhập xuất giá trị cho thuộc tính.....	69
5.6.1.	Cách 1: nhập trực tiếp chuỗi vào bảng theo đúng định dạng	69
5.6.2.	Cách 2: nhập thông qua giao diện.....	70
5.7.	Hiện thực khối xử lý truy vấn tập hợp	71
5.7.1.	Lớp xử lý CompareProbTuple	71
5.7.2.	Các lớp xử lý Union, Intersect, Except	71
5.7.3.	Xử lý phép chọn tập hợp.....	72
5.7.4.	Các lớp hỗ trợ.....	72
5.8.	Hiện thực khối xử lý tập hợp	73
5.9.	Hiện thực khối xử lý truy vấn và tính toán	75
5.9.1.	Xử lý thực thi câu truy vấn	76
5.9.2.	Xử lý điều kiện chọn.....	76
5.9.3.	Xử lý phép chiếu	79
5.9.4.	Xử lý phép Descartes	79
5.9.5.	Xử lý phép kết.....	80
5.9.6.	Xử lý phép hợp.....	80
5.9.7.	Xử lý phép giao.....	80
5.9.8.	Xử lý phép trừ.....	81
5.10.	Giao diện người dùng.....	81
5.10.1.	Giao diện chính	82

5.10.2.	Giao diện Schema	82
5.10.3.	Giao diện Relation	83
5.10.4.	Giao diện truy vấn.....	84
5.10.5.	About ứng dụng	85
5.10.6.	Giao diện truy vấn phép chiếu	86
5.10.7.	Giao diện truy vấn phép chọn	86
5.10.8.	Giao diện truy vấn phép kết	87
5.10.9.	Giao diện phép Tích Descartes	88
5.10.10.	Giao diện truy vấn phép giao	88
5.10.11.	Giao diện truy vấn phép hợp.....	89
5.10.12.	Giao diện truy vấn phép trừ	90
TỔNG KẾT VÀ ĐỀ NGHỊ		91
1.	Tổng kết	91
2.	Đề nghị.....	92
TÀI LIỆU THAM KHẢO		93

DANH MỤC CÁC BẢNG

Bảng 2.2.1 Các tiên đề về chiến lược hội	26
Bảng 2.2.2 Các tiên đề về chiến lược tuyển.....	27
Bảng 2.2.3 Các ví dụ về các chiến lược kết hợp xác suất.....	28
Bảng 3.5.1 Quan hệ PATIENT	40
Bảng 4.2.1 Quan hệ $\sigma_{\phi}(\text{PATIENT})$	48
Bảng 4.3.1 Quan hệ $\prod_{L \oplus \text{in}}(\text{PATIENT})$	49
Bảng 4.5.1 Quan hệ PATIENT_1	51
Bảng 4.5.2 Quan hệ PATIENT_2	51
Bảng 4.5.3 Quan hệ $\text{PATIENT}_1 \bowtie_{\otimes \text{in}} \text{PATIENT}_2$	51
Bảng 4.6.1 Relation DIAGNOSE_1	52
Bảng 4.6.2 Relation DIAGNOSE_2	52
Bảng 4.6.3 $\text{DIAGNOSE}_1 \cap_{\otimes \text{in}} \text{DIAGNOSE}_2$	52
Bảng 4.6.4 $\text{DIAGNOSE}_1 \cup_{\oplus \text{in}} \text{DIAGNOSE}_2$	53
Bảng 4.6.5 $\text{DIAGNOSE}_1 -_{\ominus \text{ig}} \text{DIAGNOSE}_2$	54

DANH MỤC CÁC HÌNH

Hình 3.2.1 Cơ sở dữ liệu khám-chữa bệnh	34
Hình 3.2.2. Kiến trúc của hệ thống URDB	35
Hình 5.4.1 Kiến trúc hệ quản trị URDB	65
Hình 5.6.1 Nhập bộ ba xác suất trực tiếp.....	70
Hình 5.6.2 Nhập các bộ ba xác suất phân bố không đều	70
Hình 5.6.3 Nhập các bộ ba phân bố xác suất đều	71
Hình 5.8.1 Phép chọn với điều kiện là một tập hợp.....	73
Hình 5.8.2 Phép chọn với điều kiện dữ liệu trên hai thuộc tính bằng nhau	73
Hình 5.8.3 Phép chọn với điều kiện chứa một tập dữ liệu.....	74
Hình 5.8.4 Phép chọn với điều kiện nằm trong một tập cha.....	74
Hình 5.8.5 Phép chọn với điều kiện không chứa tập điều kiện	75
Hình 5.10.1 Giao diện chính.....	82
Hình 5.10.2 Giao diện Schema	82
Hình 5.10.3Giao diện Relation	83
Hình 5.10.4 Giao diện truy vấn.....	84
Hình 5.10.5 Giao diện câu truy vấn và kết quả trả về.....	84
Hình 5.10.6 Giao diện các phím chức năng và insert	85
Hình 5.10.7 Giao diện giới thiệu ứng dụng	85
Hình 5.10.8 Giao diện truy vấn phép chiếu	86
Hình 5.10.9 Giao diện truy vấn phép chọn	86
Hình 5.10.10 Giao diện truy vấn phép kết	87
Hình 5.10.11 Giao diện truy vấn phép kết tự nhiên.....	87
Hình 5.10.12 Giao diện truy vấn phép tích Descartes	88
Hình 5.10.13 Giao diện truy vấn phép giao	88
Hình 5.10.14 Giao diện truy vấn phép hợp.....	89
Hình 5.10.15 Giao diện truy vấn phép hợp theo chiến lược tuyển độc lập	89
Hình 5.10.16 Giao diện truy vấn phép trừ	90

DANH SÁCH CÁC CỤM TỪ VIẾT TẮT

CSDL	: Cơ Sở Dữ Liệu
URDB	: Relational DataBase with uncertain multivalued attributes
\subseteq	: quan hệ tập con
$\not\subseteq$: quan hệ không là tập con
\in	: quan hệ liên thuộc
θ	: Quan hệ hai ngôi
\bowtie	: phép kết
Π	: Phép chiếu
\cap	: phép toán giao tập hợp
\cup	: phép toán hợp tập hợp
\leq	: quan hệ nhỏ hơn hoặc bằng trên tập các số thực/khoảng
\geq	: quan hệ lớn hơn hoặc bằng trên tập các số thực/khoảng
\otimes	: phép toán hội xác suất của hai khoảng ứng với hai biến cố
\oplus	: phép toán tuyển xác suất của hai khoảng ứng với hai biến cố
\ominus	: phép toán trừ xác suất của hai khoảng ứng với hai biến cố
Pr	: hàm tính xác suất của một quan hệ/sự kiện
$prob$: hàm tính xác suất của các quan hệ hai ngôi trên các tập hợp
$prob_{R,r,t}$: hàm tính diễn dịch xác suất của các biểu thức chọn xác suất
min	: hàm tính giá trị nhỏ nhất của một tập các số thực
max	: hàm tính giá trị lớn nhất của một tập các số thực

MỞ ĐẦU

1. Phạm vi và mục tiêu

Như chúng ta đã biết, *mô hình quan hệ truyền thống* (conventional relational model), được đề nghị bởi Codd E.F năm 1970 ([1]), đã chứng tỏ nhiều ưu điểm trong các vấn đề mô hình hóa, thiết kế và hiện thực các hệ thống lớn, từ phần mềm cho đến cơ sở dữ liệu (CSDL). Điều đó được thể hiện nhờ khả năng biểu diễn các đối tượng cũng như quan hệ giữa chúng trong mô hình này một cách đúng đắn, phản ánh đặc tính và hành vi của các đối tượng trong thực tế dựa trên việc tích hợp và sử dụng các công cụ toán học cổ điển như quan hệ, đại số quan hệ, ánh xạ (các phụ thuộc hàm), tập hợp, logic mệnh đề, logic vị từ v.v. ([1], [2], [3]). Tuy nhiên, trong mô hình CSDL quan hệ truyền thống các mối quan hệ cũng như trạng thái của các đối tượng luôn luôn được thể hiện một cách chắc chắn và chính xác[4]. Điều này là không hoàn toàn phù hợp với thực tế, như đã được chỉ ra trong ([5], [10], [14], [28]), bởi vì thông tin về các đối tượng trong thế giới thực có thể mơ hồ, không chắc chắn và không đầy đủ.

Hệ quả là các ứng dụng dựa trên các mô hình CSDL truyền thống không biểu diễn được các đối tượng và quan hệ mà thông tin về chúng không được xác định một cách chắc chắn và chính xác. Điều đó làm hạn chế khả năng mô hình hóa và giải quyết các bài toán áp dụng trong thế giới thực. Chẳng hạn, các ứng dụng mô hình CSDL truyền thống không thể trả lời được các câu hỏi, truy vấn trong thực tế kiểu như “tìm tất cả những cầu thủ có 90% khả năng là vua phá lưới giải ngoại hạng Anh mùa giải 2021-2022”; hoặc “tìm các đội bóng ngoại hạng Anh mà có 70-90% khả năng vô địch mùa giải 2021-2022”; hay “tìm tất cả những bệnh nhân mà có khả năng 80 đến 90% bị bệnh viêm túi mật hoặc viêm gan”; hoặc “tìm tất cả các gói bưu kiện được vận chuyển trong thời gian 36 hoặc 48 giờ từ Hà Nội đến Sài Gòn với xác suất ít nhất là 90%”, v.v. Để khắc phục được các hạn chế như vậy, cần phải xây dựng các mô hình dữ liệu có khả năng biểu diễn và xử lý được các đối tượng mà các thông tin về chúng có thể không chắc chắn và không đầy đủ.

Trong những năm 80 của thế kỷ trước đã có những nghiên cứu và xây dựng các mô hình CSDL quan hệ với giá trị NULL để xử lý thông tin không chắc chắn, không đầy

đủ. Tiêu biểu trong số đó là mô hình do Imielinski và Lipski đề nghị trong [4]. Đây là một mô hình CSDL quan hệ được xây dựng như một *hệ thống biểu diễn* (representation system) dựa trên cơ sở toán học chặt chẽ và vững chắc. Trong mô hình này các tác giả đã sử dụng giá trị NULL để biểu diễn các giá trị chưa biết hoặc chưa được định nghĩa của một thuộc tính quan hệ. Cách biểu diễn và ngữ nghĩa của các giá trị NULL như vậy được thừa nhận trong CSDL quan hệ và các hệ quản trị của nó mà ta đã biết hiện nay.

Tuy nhiên, trong những tình huống ở đó chúng ta không biết chính xác giá trị thuộc tính nhưng biết khả năng mà nó nhận một giá trị nào đó thì không thể mô hình hóa thông tin này bởi giá trị NULL. Chẳng hạn, chúng ta không biết chắc chắn bệnh nhân bị bệnh gì, nhưng lại biết khả năng 80-90% bệnh nhân bị bệnh viêm túi mật hoặc viêm gan thì chúng ta không thể dùng CSDL quan hệ truyền thống (có sử dụng giá trị NULL) để biểu diễn bệnh nhân này. Vì nếu chúng ta biểu diễn tình huống này bằng giá trị NULL thay cho bệnh của bệnh nhân, chúng ta đã không thể hiện đúng thông tin thực tế. Như chúng ta đã biết, lý thuyết xác suất có thể mô hình hóa tính không chắc chắn, không đầy đủ của thông tin. Vì vậy, một giải pháp tự nhiên để vượt qua giới hạn của các mô hình CSDL truyền thống trong việc xử lý thông tin không chắc chắn là mở rộng mô hình này bằng cách áp dụng các kết quả của lý thuyết xác suất trong biểu diễn dữ liệu cũng như cách xây dựng các thao tác dữ liệu trên mô hình này.

Theo tinh thần đó, trong những năm qua đã có nhiều mô hình CSDL được nghiên cứu và xây dựng dựa trên sự tích hợp của lý thuyết xác suất vào mô hình CSDL quan hệ nhằm mô hình hóa thông tin của các đối tượng và quan hệ trong thế giới thực sao cho đúng với bản chất không chắc chắn vốn có của chúng. Các mô hình như vậy được gọi là *mô hình cơ sở dữ liệu quan hệ xác suất* (probabilistic relational data base model). Thực chất của việc xây dựng mô hình CSDL quan hệ xác suất là mở rộng mô hình CSDL quan hệ truyền thống bằng cách áp dụng lý thuyết xác suất. Hiện nay có hai cách tiếp cận chính để phát triển mô hình dữ liệu quan hệ xác suất là mở rộng biểu diễn quan hệ của các bộ hoặc mở rộng biểu diễn giá trị thuộc tính bộ của mô hình dữ liệu truyền thống. Trên cơ sở mô hình dữ liệu được mở rộng, mô hình thao tác dữ liệu sẽ được mở rộng tương ứng.

Theo cách tiếp cận thứ nhất, một số mô hình đã mở rộng mỗi quan hệ cổ điển bằng một quan hệ xác suất như trong ([5-15], [18], [24], [26]). Nghĩa là mỗi bộ trong một quan hệ có một mức độ không chắc chắn, được đo bằng xác suất, để nó thuộc về quan hệ. Độ đo xác suất này còn được diễn dịch như là mức độ không chắc chắn mà các thuộc tính có thể nhận các giá trị trong một bộ cụ thể. Trên cơ sở biểu diễn như vậy, các phép toán đại số quan hệ xác suất đã được xây dựng như là một mở rộng của các phép toán đại số quan hệ trong mô hình CSDL quan hệ truyền thống. Kết quả các truy vấn dữ liệu là một quan hệ xác suất với mức độ xác suất cụ thể của từng bộ thỏa mãn yêu cầu của truy vấn.

Trong cách tiếp cận thứ hai, mô hình CSDL quan hệ xác suất cho phép biểu diễn giá trị không chắc chắn thể hiện tình trạng thiếu thông tin về đối tượng. Có một vài khác biệt nhỏ trong cách biểu diễn giá trị thuộc tính bộ trong quan hệ xác suất. Một số mô hình, như trong ([16], [17]), gán một xác suất trong khoảng $[0, 1]$ cho giá trị thuộc tính biểu diễn mức độ không chắc chắn mà thuộc tính có thể nhận giá trị này. Các phép toán đại số quan hệ tương ứng được xây dựng để truy vấn trên các giá trị thuộc tính, xác định các bộ thỏa yêu cầu về xác suất trong một quan hệ của cơ sở dữ liệu.

Một phương pháp khác mềm dẻo hơn, được đề nghị trong [21]. Trong đó, các tác giả đã phát triển một mô hình CSDL quan hệ xác suất cho phép giá trị thuộc tính được kết hợp với một khoảng xác suất biểu diễn mức độ không chắc chắn về cả xác suất và giá trị mà thuộc tính có thể nhận. Các phép toán đại số cũng được xây dựng để thao tác trên các quan hệ và xác định các bộ thỏa một khoảng xác suất được yêu cầu trong truy vấn. Các mô hình trong ([22], [23]) là những mở rộng của mô hình trong [21] bằng cách mở rộng giá trị thuộc tính với hai phân bố xác suất trên một tập, biểu diễn mức độ không chắc chắn để thuộc tính nhận một trong các giá trị của tập với một khoảng xác suất được suy dẫn từ các phân bố xác suất này.

Trong các mô hình CSDL quan hệ xác suất đã được đề nghị như giới thiệu ở trên, bao gồm cả hai hướng tiếp cận, thuộc tính của mỗi bộ hoặc đối tượng chỉ nhận một giá trị đơn, duy nhất trong một tập giá trị với một xác suất tương ứng biểu diễn mức độ không chắc chắn mà thuộc tính có thể nhận giá trị này. Chẳng hạn, giá trị thuộc tính DISEASE (bệnh của bệnh nhân) trong mô hình [22] được biểu diễn bởi DISEASE:

$\langle \{hepatitis, cirrhosis, cholecystitis\}, 0.9u, 1.5u \rangle$ cho biết bệnh của bệnh nhân có thể là *hepatitis* hoặc *cirrhosis* hoặc *cholecystitis* với một xác suất trong khoảng $[0.3, 0.5]$. Tuy nhiên, nếu thực tế bệnh nhân có thể bị đồng thời hai bệnh *hepatitis* và *cirrhosis* hoặc bệnh *cholecystitis* với các xác suất tương ứng nào đó thì mô hình này (và các mô kê trên) không thể biểu diễn được.

Như vậy, mặc dù có nhiều nghiên cứu được đề nghị để xây dựng và phát triển các mô hình CSDL quan hệ xác suất, nhưng không có mô hình nào là hoàn chỉnh, có thể biểu diễn và xử lý mọi khía cạnh không chắc chắn của thông tin của các đối tượng thực tế. Do đó, các mô hình CSDL quan hệ xác suất vẫn được tiếp tục nghiên cứu nhằm đáp ứng các áp dụng khác nhau. Ngoài ra, hầu hết các nghiên cứu về CSDL quan hệ xác suất hiện nay chủ yếu tập trung xây dựng mô hình dữ liệu (biểu diễn thông tin và các phép toán xử lý thông tin) mà ít quan tâm đến việc xây dựng các hệ quản trị CSDL làm cơ sở cho việc áp dụng mô hình vào thực tế. Đề tài khóa luận này nhằm mục tiêu xây dựng một hệ quản trị *CSDL quan hệ xác suất với giá trị thuộc tính không chắc chắn* (Probabilistic Relational Data Base with Uncertain Attribute Values-URDB) mới để biểu diễn và xử lý thông tin không chắc chắn rất phổ biến trong thực tế. Mô hình URDB đã được xây dựng trong [28] như là một mở rộng của các mô hình [22-23] với thuộc tính đa trị không chắc chắn.

Để xây dựng URDB, một diễn dịch xác suất của các quan hệ hai ngôi trên các tập hợp được đề nghị như một độ đo mức độ không chắc chắn của các giá trị thuộc tính quan hệ, các bộ ba xác suất trên một tập hợp trong [23] được mở rộng thành các bộ ba xác suất trên tập các tập hợp để biểu diễn các thuộc tính đa trị và sử dụng các chiến lược kết hợp các khoảng xác suất trong $([19], [20])$ mở rộng các khái niệm trong mô hình CSDL truyền thống như kiểu, giá trị, lược đồ, quan hệ thành các khái niệm tương ứng trong URDB. Các phép toán đại số trong CSDL truyền thống cũng được mở rộng thành các phép toán đại số tương ứng trong URDB để truy vấn các quan hệ với thuộc tính đa trị không chắc chắn.

Mô hình URDB được mở rộng như vậy vẫn đảm bảo tính nhất quán với mô hình CSDL truyền thống. Nói một cách khác, mô hình URDB vẫn hoàn toàn có thể biểu diễn và xử lý được các quan hệ mà mô hình của Codd [1] đã biểu diễn và xử lý. Mô hình

CSDL quan hệ truyền thống có thể được xem như là trường hợp riêng của mô hình URDB. Một tập các tính chất của các phép toán đại số trên URDB cũng được mở rộng trong Chương 4 từ các tính chất của các phép toán đại số quan hệ truyền thống. Các tính chất này được chứng minh đầy đủ, chứng tỏ mô hình URDB được xây dựng là đúng đắn.

Hệ quản trị cho URDB được xây dựng với ngôn ngữ truy vấn thân thiện tựa SQL dựa trên hệ quản trị mã nguồn mở (truyền thống) SQLite, gọi là URDB-SQLite, bước đầu cho thấy triển vọng ứng dụng của URDB để mô hình hóa dữ liệu không chắc chắn và giải quyết các bài toán thực tế.

Một cơ sở dữ liệu các bệnh nhân tại phòng khám của một bệnh viện được dùng làm ví dụ minh họa cho lý thuyết và thử nghiệm chương trình cho thấy rõ hơn bản chất mô hình URDB và cách thức ứng dụng của nó.

2. Những đóng góp chính của đề tài

Sau đây là những đóng góp chính của luận văn này đối với lĩnh vực cơ sở dữ liệu và lĩnh vực *tính toán mềm* (soft computing):

1. Giới thiệu mô hình CSDL quan hệ xác suất URDB (lược đồ, quan hệ, các bộ tương đương giá trị, phụ thuộc hàm xác suất) và các phép toán đại số quan hệ xác suất để xử lý và truy vấn thông tin không chắc chắn trong CSDL được mô hình hóa bởi URDB.
2. Hiện thực một hệ quản trị CSDL với ngôn ngữ truy vấn thân thiện tựa SQL cho mô hình CSDL quan hệ xác suất URDB.

3. Cấu trúc khóa luận

Báo cáo đề tài gồm 5 chương và một phần mở đầu và một phần tổng kết. Phần mở đầu trình bày phạm vi, mục tiêu và ý nghĩa về lý thuyết cũng như ứng dụng của đề tài, giới thiệu cấu trúc, các quy ước ký hiệu và viết tắt trong báo cáo đề tài.

Chương 1 giới thiệu tổng quan về CSDL quan hệ truyền thống như là một mô hình nền tảng, cơ sở để xây dựng và phát triển URDB. Từ mô hình CSDL truyền thống chúng ta thấy được mối liên hệ của những yếu tố mở rộng trong URDB.

Chương 2 trình bày cơ sở toán học để phát triển mô hình URDB. Đó là các khái niệm cơ bản, nền tảng của lý thuyết tập hợp và lý thuyết xác suất làm cơ sở để biểu diễn và xử lý thông tin không chắc chắn của các thuộc tính quan hệ trong URDB.

Chương 3 trình bày mô hình ý niệm của URDB, các khái niệm kiểu, thuộc tính, giá trị, lược đồ, phụ thuộc hàm và quan hệ xác suất trên cơ sở mở rộng các khái niệm tương ứng trong CSDL quan hệ truyền thống.

Chương 4 trình bày các phép toán đại số quan hệ trên URDB. Đó là các phép toán như chọn, chiếu, tích Descartes, kết, giao, hợp và trừ trên các quan hệ xác suất trong URDB. Các phép toán này là mở rộng các phép toán đại số trong CSDL quan hệ truyền thống với sự tích hợp đa giá trị không chắc chắn của các thuộc tính.

Chương 5 trình bày hệ quản trị SQLite và cách thức xây dựng hệ quản trị URDB-SQLite với truy vấn chọn cho URDB trên 1 hoặc 2 quan hệ.

Phần tổng kết tóm tắt các kết quả đã đạt được và các hướng nghiên cứu trong tương lai liên quan đến các vấn đề của đề tài.

Chương 1

TỔNG QUAN VỀ MÔ HÌNH CSDL QUAN HỆ

1.1. Giới thiệu

Chương này trình bày một cách khái quát mô hình CSDL quan hệ truyền thống được Codd đề xuất năm 1970 trong [1] và được tiếp tục phát triển sau đó như chúng ta biết hiện nay ([2], [3], [4]). Mô hình CSDL quan hệ truyền thống đã chứng tỏ nhiều ưu điểm trong mô hình hóa các áp dụng thực tế. Tuy nhiên, mô hình quan hệ truyền thống không thể biểu diễn và xử lý được thông tin không chắc chắn ([8], [15], [18]). Hạn chế này thúc đẩy sự nghiên cứu và áp dụng các mô hình CSDL quan hệ xác suất (Probabilistic Relational Data Base). Mô hình CSDL quan hệ xác suất là một mở rộng của mô hình CSDL quan hệ truyền thống. Vì vậy, một giới thiệu có tính tổng quan về mô hình CSDL quan hệ truyền thống cho thấy sự phát triển có tính logic, khái quát từ mô hình này lên mô hình CSDL quan hệ xác suất. Phần 1.2 giới thiệu về mô hình dữ liệu của mô hình CSDL quan hệ bao gồm kiểu, giá trị, lược đồ, quan hệ và phụ thuộc hàm, một khái niệm quan trọng trong CSDL quan hệ. Phần 1.3 trình bày các phép toán đại số quan hệ như là cơ sở cho ngôn ngữ truy vấn thông tin các quan hệ. Phần 1.4 nêu một số tính chất của các phép toán đại số quan hệ như tính giao hoán, kết hợp, v.v. Cuối cùng, Phần 1.5 là một số kết luận đáng lưu ý của chương này.

1.2. Mô hình dữ liệu

Như trong hầu hết các mô hình CSDL đang tồn tại hiện nay, mô hình CSDL quan hệ được dựa trên một tập các khái niệm cơ bản như thuộc tính, kiểu, giá trị, lược đồ và *thể hiện* (instance) của lược đồ (quan hệ trên lược đồ) để xây dựng mô hình dữ liệu. Các khái niệm này lần lượt được định nghĩa sau đây.

Định nghĩa 1.2.1 Giả sử A là một tập các thuộc tính và T là một tập các *kiểu cơ sở* (atomic types). Các *kiểu* (type) được định nghĩa như sau:

1. Mọi kiểu cơ sở trong T là một kiểu.
2. Nếu A_1, A_2, \dots, A_k là các thuộc tính đôi một khác nhau trong A và $\tau_1, \tau_2, \dots, \tau_k$ là các kiểu cơ sở thì $\tau = [A_1: \tau_1, A_2: \tau_2, \dots, A_k: \tau_k]$ là một kiểu, được gọi là *kiểu bộ* (tuple type) trên tập các thuộc tính $\{A_1, A_2, \dots, A_k\}$. Với một kiểu $\tau = [A_1: \tau_1, A_2: \tau_2, \dots, A_k: \tau_k]$, chúng tôi sử dụng $\tau.A_i$ để biểu thị τ_i .

Trong mô hình CSDL quan hệ, kiểu dữ liệu của thuộc tính là miền giá trị mà thuộc tính có thể nhận. Kiểu bộ là miền giá trị của các bộ trong quan hệ có tập thuộc tính tương ứng. Mỗi bộ (v_1, v_2, \dots, v_k) thực chất là một phần tử của tích Descartes $\tau = \tau_1 \times \tau_2 \times \dots \times \tau_k$ trên các miền giá trị tương ứng của các thuộc tính A_1, A_2, \dots, A_k của quan hệ. Và như vậy, một tập các bộ (v_1, v_2, \dots, v_k) là một tập con của tích $\tau = \tau_1 \times \tau_2 \times \dots \times \tau_k$ và đó là một quan hệ k -ngôi trên tập các giá trị các thuộc tính A_1, A_2, \dots, A_k .

Định nghĩa 1.2.2 Mỗi kiểu cơ sở $\tau \in T$ có một miền xác định, được ký hiệu là $dom(\tau)$, kết hợp với nó. *Giá trị* (value) được định nghĩa như sau:

1. Với mọi kiểu cơ sở $\tau \in T$, thì mỗi $v \in dom(\tau)$ là một giá trị kiểu τ .
2. Nếu A_1, \dots, A_k là các thuộc tính đôi một khác nhau trong A và v_1, \dots, v_k là các giá trị tương ứng của các kiểu τ_1, \dots, τ_k thì $[A_1: v_1, \dots, A_k: v_k]$ là một giá trị kiểu $[A_1: \tau_1, \dots, A_k: \tau_k]$, được gọi là *giá trị kiểu bộ* (tuple type value) hay đơn giản là một bộ, trên tập các thuộc tính $\{A_1, A_2, \dots, A_k\}$.

Trong một số trường hợp nếu không quan tâm đến tên thuộc tính, ta có thể viết giá trị kiểu bộ đơn giản là $t = (v_1, v_2, \dots, v_k)$.

Định nghĩa 1.2.3 Giả sử A là một tập các thuộc tính, một tập $R = \{A_1, A_2, \dots, A_k\}$ các thuộc tính đôi một khác nhau trong A được gọi là một *lược đồ quan hệ* (relational schema) trên A_1, A_2, \dots, A_k , ký hiệu là $R(A_1, A_2, \dots, A_k)$.

Ví dụ 1.2.1 Một lược đồ của quan hệ **PATIENT** trong CSDL các bệnh nhân của một bệnh viện có thể là **PATIENT**(PATIENT_ID, PATIENT_NAME, BIRTHDAY, SEX, DISEASE).

Định nghĩa 1.2.4 Một *quan hệ* (relation) r , trên lược đồ $R(A_1, A_2, \dots, A_k)$ là một tập hữu hạn các bộ $\{t_1, t_2, \dots, t_n\}$ trên tập các thuộc tính $\{A_1, A_2, \dots, A_k\}$. Các ký hiệu $t.A$ hoặc $t[A]$ biểu thị giá trị thuộc tính A của bộ t trong r .

Ngoài ra, ký hiệu $t[X]$ được dùng để biểu thị giá trị bộ thu hẹp của t trên tập thuộc tính $X \subseteq \{A_1, A_2, \dots, A_k\}$.

Định nghĩa 1.2.5 Một *cơ sở dữ liệu quan hệ* (relational database) trên một tập các thuộc tính là một tập các quan hệ tương ứng với một tập các lược đồ quan hệ của chúng.

Một khái niệm có ý nghĩa ứng dụng trong mô hình CSDL quan hệ được gọi là khóa của lược đồ quan hệ. Khóa của lược đồ quan hệ là cơ sở để nhận dạng các bộ trong một quan hệ. Khóa của lược đồ quan hệ được định nghĩa như sau.

Định nghĩa 1.2.6 Giả sử $R(A_1, A_2, \dots, A_k)$ là một lược đồ quan hệ trên tập thuộc tính $\{A_1, A_2, \dots, A_k\}$. Một tập thuộc tính $K \subseteq \{A_1, A_2, \dots, A_k\}$ được gọi là *khóa* của R nếu với mọi quan hệ r trên R và hai bộ bất kỳ t_1 và t_2 của r mà $t_1[K] = t_2[K]$ thì $t_1 \equiv t_2$ và không tồn tại bất kỳ tập con nào của K có tính chất này.

Phụ thuộc hàm, một dạng đặc biệt của các quan hệ giữa các thuộc tính như một loại ràng buộc dữ liệu, là một khái niệm quan trọng làm cơ sở cho tối ưu hóa tổ chức dữ liệu trong mô hình CSDL quan hệ. Phụ thuộc hàm được định nghĩa như sau.

Định nghĩa 1.2.7 Cho một lược đồ quan hệ $R(A_1, A_2, \dots, A_k)$, r là một quan hệ bất kỳ trên R , X và Y là hai tập con các thuộc tính của R . Một *phụ thuộc hàm* (function dependence) của Y đối với X trên lược đồ quan hệ R , ký hiệu là $X \rightarrow Y$, nếu:

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$

Phụ thuộc hàm $X \rightarrow Y$ còn được gọi là “ X xác định hàm Y ” hoặc “ Y phụ thuộc hàm vào X ”.

1.3. Các phép toán đại số

Tập các phép toán *đại số quan hệ* (relational algebra) bao gồm phép chọn, chiếu, tích Descartes, kết, giao, hợp và trừ là một ngôn ngữ để thực hiện các truy vấn trong CSDL quan hệ. Đầu vào và đầu ra của các phép toán đại số là các quan hệ. Sau đây,

chúng tôi lần lượt giới thiệu các định nghĩa của các phép toán đại số quan hệ trong mô hình CSDL truyền thống ([2], [3]).

Định nghĩa 1.3.1 Giả sử R là một lược đồ quan hệ, r là một quan hệ trên R và ϕ là một điều kiện chọn. *Phép chọn* (selection) trên r theo ϕ , được ký hiệu $\sigma_\phi(r)$, là một quan hệ trên R , bao gồm tất cả các bộ thỏa mãn điều kiện chọn ϕ

$$\sigma_\phi(r) = \{t \in r \mid \phi(t) = \text{true}\}.$$

Điều kiện chọn ϕ là một mệnh đề biểu diễn các ràng buộc của các giá trị thuộc tính của các bộ t trong quan hệ r .

Định nghĩa 1.3.2 Giả sử R là một lược đồ quan hệ, r là một quan hệ trên R và X là một tập các thuộc tính của R , gọi $\pi_X(R)$ là lược đồ quan hệ trên X . *Phép chiếu* (projection) của r trên X , được ký hiệu là $\pi_X(r)$, là một quan hệ r' trên $\pi_X(R)$ bao gồm các bộ là thu hẹp của r trên X . Nghĩa là $r' = \{t' = t[X] \mid t \in r\}$.

Định nghĩa 1.3.3 Giả sử $\{A_1, A_2, \dots, A_m\}$ và $\{B_1, B_2, \dots, B_n\}$ là hai tập thuộc tính không giao nhau. Gọi r và s là hai quan hệ tương ứng trên hai lược đồ $R(A_1, A_2, \dots, A_m)$ và $S(B_1, B_2, \dots, B_n)$. Phép tích Descartes của hai quan hệ r và s , kí hiệu là $r \times s$, là một quan hệ trên lược đồ $Q(A_1, A_2, \dots, A_m, B_1, B_2, \dots, B_n)$ bao gồm các bộ t được xác định bởi

$$r \times s = \{t = (a_1, \dots, a_m, a_{m+1}, \dots, a_{m+n}) \mid (a_1, \dots, a_m) \in r \text{ và } (a_{m+1}, \dots, a_{m+n}) \in s\}.$$

Định nghĩa 1.3.4 Giả sử A và B là hai tập thuộc tính sao cho nếu chúng có thuộc tính cùng tên thì các thuộc tính đó phải có cùng kiểu giá trị. Gọi r và s tương ứng là các quan hệ trên các lược đồ $R(A)$ và $S(B)$. *Phép kết tự nhiên* (natural join) của r và s , kí hiệu là $r \bowtie s$, là một quan hệ trên lược đồ $Q(A \cup B)$ được xác định theo tích Descartes của r và s bao gồm các bộ t sao cho:

$$t[A] = t_r \in r, \quad t[B] = t_s \in s \text{ và } t_r[C] = t_s[C] \text{ với mọi } C \in A \cap B.$$

Định nghĩa 1.3.5 Giả sử r và s là hai quan hệ trên cùng một lược đồ R . *Phép hợp* (union) của hai quan hệ r và s , kí hiệu là $r \cup s$, là một quan hệ trên R bao gồm các bộ của r hay của s . Nghĩa là $r \cup s = \{t \mid t \in r \text{ hay } t \in s\}$.

Định nghĩa 1.3.6 Giả sử r và s là hai quan hệ trên cùng một lược đồ R . *Phép giao* (intersection) của hai quan hệ r và s , kí hiệu là $r \cap s$, là một quan hệ trên R bao gồm các bộ thuộc đồng thời cả r và s . Nghĩa là $r \cap s = \{t \mid t \in r \text{ và } t \in s\}$.

Định nghĩa 1.3.7 Giả sử r và s là hai quan hệ trên cùng một lược đồ R . *Phép trừ* (difference) của quan hệ r cho s , kí hiệu là $r - s$, là một quan hệ trên R bao gồm các bộ của r không có trong s . Nghĩa là $r - s = \{t \mid t \in r \text{ và } t \notin s\}$.

Các phép toán được trình bày ở trên là những phép toán cơ bản. Trong mô hình CSDL quan hệ còn có *Phép kết θ* (θ -join) được suy dẫn từ phép chọn và tích Descartes, *Phép chia* (division) được suy dẫn từ phép chiếu, tích Descartes và trừ như trong [3].

1.4. Tính chất các phép toán đại số

Từ các định nghĩa của các phép toán đại số trong CSDL quan hệ, dễ dàng nhận thấy chúng có thời gian thực hiện tuyến tính hoặc bậc hai theo kích thước của quan hệ. Chẳng hạn, với quan hệ đầu vào có n bộ thì thời gian tính toán của phép chọn để có kết quả đầu ra là $O(n)$. Các phép toán đại số là hiệu quả. Ngoài ra, cũng dễ dàng nhận thấy chúng có một số tính chất như giao hoán, kết hợp, phân bố v.v. Sau đây là các tính chất của các phép toán đại số quan hệ được phát biểu như những định lý. Việc chứng minh chúng là khá đơn giản nên được bỏ qua. Các chứng minh có thể tìm thấy trong [3].

Định lý 1.4.1 Giả sử r là một quan hệ trên lược đồ R . Gọi ϕ_1 và ϕ_2 là hai điều kiện chọn. Khi đó

$$\sigma_{\phi_1}(\sigma_{\phi_2}(r)) = \sigma_{\phi_2}(\sigma_{\phi_1}(r)) = \sigma_{\phi_1 \wedge \phi_2}(r) \quad (1)$$

Kết quả này chứng tỏ thứ tự của các điều kiện chọn không ảnh hưởng đến kết quả phép chọn.

Định lý 1.4.2 Giả sử r là một quan hệ trên lược đồ R , A và B là các tập thuộc tính trong R , $A \subseteq B$. Khi đó

$$\Pi_A(\Pi_B(r)) = \Pi_A(r) \quad (2)$$

Định lý 1.4.3 Giả sử r_1 , r_2 và r_3 tương ứng là các quan hệ trên R_1 , R_2 và R_3 , sao cho nếu chúng có thuộc tính cùng tên thì cùng kiểu giá trị. Khi đó

$$r_1 \bowtie r_2 = r_2 \bowtie r_1 \quad (3)$$

$$(r_1 \bowtie r_2) \bowtie r_3 = r_1 \bowtie (r_2 \bowtie r_3) \quad (4)$$

Bởi vì phép lấy tích Descartes là trường hợp riêng của phép kết nên một hệ quả trực tiếp của định lý trên được phát biểu như sau.

Hệ quả 1.4.1 Giả sử r_1, r_2 và r_3 tương ứng là các quan hệ trên R_1, R_2 và R_3 , sao cho tập các thuộc tính của chúng không giao nhau. Khi đó

$$r_1 \times r_2 = r_2 \times r_1 \quad (5)$$

$$(r_1 \times r_2) \times r_3 = r_1 \times (r_2 \times r_3) \quad (6)$$

Định lý 1.4.4 Giả sử r_1, r_2 và r_3 là các quan hệ trên cùng một lược đồ R . Khi đó

$$r_1 \cap r_2 = r_2 \cap r_1 \quad (7)$$

$$(r_1 \cap r_2) \cap r_3 = r_1 \cap (r_2 \cap r_3) \quad (8)$$

$$r_1 \cup r_2 = r_2 \cup r_1 \quad (9)$$

$$(r_1 \cup r_2) \cup r_3 = r_1 \cup (r_2 \cup r_3) \quad (10)$$

1.5. Kết luận

Trong chương này, những yếu tố cơ bản nhất của mô hình dữ liệu và mô hình thao tác dữ liệu của CSDL quan hệ truyền thống đã được giới thiệu. Mô hình CSDL quan hệ truyền thống dựa trên cơ sở lý thuyết quan hệ, tuy đơn giản nhưng chặt chẽ và đặc biệt, tập các phép toán đại số là hiệu quả vì có thời gian thực thi là tuyến tính hay bậc hai đối với kích thước quan hệ. Các định lý về các phép toán đại số cho thấy mô hình CSDL quan hệ được xây dựng đúng đắn. Tuy nhiên, như chúng ta đã thấy, mô hình CSDL quan hệ truyền thống không có khả năng xử lý thông tin không chắc chắn vì thông tin về các giá trị thuộc tính quan hệ được biểu diễn là chắc chắn. Đó là cơ sở động lực thúc đẩy chúng tôi xây dựng mô hình URDB bằng cách mở rộng chính CSDL quan hệ truyền thống với giá trị thuộc tính không chắc chắn để xử lý các CSDL trong thế giới thực. Chương 2 tiếp theo giới thiệu cơ sở lý thuyết để xây dựng URDB.

Chương 2

CƠ SỞ TOÁN HỌC CỦA MÔ HÌNH URDB

2.1. Giới thiệu

Chương này giới thiệu một số khái niệm và công cụ cơ bản về toán học làm cơ sở để xây dựng mô hình cơ sở dữ liệu quan hệ xác suất URDB. Đầu tiên là các chiến lược kết hợp xác suất trên các khoảng trong Phần 2.2 do Lakshmanan và các cộng sự đề xuất năm 1997 [14] và được Eiter và các cộng sự bổ sung năm 2001 [19]. Các chiến lược kết hợp xác suất này được xây dựng dựa trên các tính chất cơ bản của lý thuyết xác suất. Đó là công cụ toán học để biểu diễn, tính toán và kết hợp xác suất của các giá trị mà thuộc tính quan hệ có thể nhận trong mô hình CSDL quan hệ xác suất URDB.

Phần 2.3 trình bày khái niệm hàm phân bố xác suất trong [19] và một mở rộng bộ ba xác suất trong [23]. Các hàm phân bố xác suất và bộ ba xác suất là công cụ thích hợp cho phép biểu diễn giá trị không chắc chắn và không đầy đủ của các thuộc tính quan hệ trong URDB.

Phần 2.4 trình bày các chiến lược kết hợp các bộ ba xác suất mở rộng. Các chiến lược kết hợp các bộ ba xác suất là cơ sở để tính toán và kết hợp xác suất của các giá trị thuộc tính của quan hệ khi thực hiện các phép toán đại số như kết, giao, hợp và trừ các quan hệ trong URDB. Phần 2.5 giới thiệu một đề xuất mới cho diễn dịch xác suất của các quan hệ hai ngôi trên các tập hợp. Diễn dịch xác suất của các quan hệ hai ngôi trên các tập hợp là cơ sở để xác định mức độ xác suất mà thuộc tính quan hệ có thể nhận một giá trị trong các thao tác và truy vấn dữ liệu không chắc chắn. Phần 2.6 là một vài lưu ý và kết luận của chương này.

2.2. Các chiến lược kết hợp các khoảng xác suất

Giả sử, chúng ta biết xác suất của các *sự kiện* (event) e_1 và e_2 tương ứng là $\Pr(e_1)$ và $\Pr(e_2)$. Khi đó xác suất $\Pr(e_1 \wedge e_2)$ của sự kiện phức hợp $e_1 \wedge e_2$ có thể được tính toán phụ thuộc vào mối quan hệ của e_1 và e_2 . Chẳng hạn, nếu e_1 và e_2 là độc lập thì $\Pr(e_1 \wedge e_2) = \Pr(e_1) \cdot \Pr(e_2)$, nếu e_1 và e_2 là loại trừ lẫn nhau thì $\Pr(e_1 \wedge e_2) = 0$, v.v. Nếu không biết (hoặc bỏ qua) mối quan hệ của e_1 và e_2 thì $\Pr(e_1 \wedge e_2)$ có thể được ước lượng trong khoảng $[\max(0, \Pr(e_1) + \Pr(e_2) - 1), \min(\Pr(e_1), \Pr(e_2))]$ như đã chỉ ra bởi Eiter và CS. (2001) trong [19].

Như vậy, xác suất của sự kiện $e_1 \wedge e_2$ không chỉ phụ thuộc vào xác suất của các sự kiện e_1 và e_2 mà còn vào cả mối quan hệ giữa chúng. Tương tự, chúng ta cũng có thể tính toán xác suất của sự kiện $e_1 \vee e_2$ tùy thuộc vào thông tin về mối quan hệ giữa chúng. Một cách khái quát, tùy thuộc vào mức độ nắm bắt thông tin về sự phụ thuộc giữa các sự kiện tham gia, có nhiều lựa chọn để tính toán xác suất của một sự kiện phức hợp liên quan đến các sự kiện này. Một sự lựa chọn để tính toán xác suất của các sự kiện phức hợp như vậy gọi là một *chiến lược kết hợp xác suất* (probabilistic combination strategy) của chúng [19].

Cũng như trong Eiter và CS. (2001) và một số mô hình cơ sở dữ liệu xác suất khác ([12], [13], [14], [15],[18], [20], [24]), trong phạm vi nghiên cứu này, các khoảng xác suất được sử dụng thay cho các giá trị xác suất vì hai lý do:

1. Trong nhiều áp dụng, xác suất của một sự kiện thường không được cung cấp một cách chính xác.
2. Khi chúng ta không biết về sự phụ thuộc giữa hai sự kiện, chúng ta chỉ có thể nói xác suất của sự kiện phức hợp của chúng thuộc về một khoảng.

Sự thống kê hay tính toán xác suất trên các khoảng phải đảm bảo các khoảng xác suất của các sự kiện là nhất quán như định nghĩa sau.

Định nghĩa 2.2.1 Giả sử e_1 và e_2 có xác suất tương ứng trong các khoảng $I_1 = [L_1, U_1]$ và $I_2 = [L_2, U_2]$. Một phép gán các khoảng xác suất như vậy được gọi là *nhất quán* (consistency) nếu và chỉ nếu thỏa mãn các điều kiện sau đây:

1. Nếu $e_1 \wedge e_2$ là *mâu thuẫn* (contradictory) thì $L_1 + L_2 \leq 1$.

2. Nếu $e_1 \wedge \neg e_2$ là mâu thuẫn thì $L_1 \leq U_2$.
3. Nếu $\neg e_1 \wedge e_2$ là mâu thuẫn thì $L_2 \leq U_1$.
4. Nếu $\neg e_1 \wedge \neg e_2$ là mâu thuẫn thì $U_1 + U_2 \geq 1$.

Trong định nghĩa này, như đã lưu ý trong [19], khái niệm mâu thuẫn ở đây theo nghĩa của logic mệnh đề, $e_1 \wedge e_2$ mâu thuẫn nghĩa là $e_1 \wedge e_2$ sai. Trong phạm vi nghiên cứu này, chúng tôi giả thiết rằng tất cả các phép gán các khoảng xác suất cho các sự kiện là nhất quán trừ phi phát biểu ngược lại. Với hai khoảng xác suất $I_1 = [L_1, U_1]$ và $I_2 = [L_2, U_2]$, ký hiệu $I_1 \leq I_2$ được sử dụng như một sự viết gọn thay cho $L_1 \leq L_2$ và $U_1 \leq U_2$ còn ký hiệu $I_1 \subseteq I_2$ thay cho $L_2 \leq L_1$ và $U_1 \leq U_2$.

Để thuận tiện cho việc biểu diễn và thực thi các truy vấn trong URDB dựa trên tính toán và suy luận xác suất, chúng tôi sử dụng các *chiến lược hội* (conjunction strategy) và *chiến lược tuyển* (disjunction strategy) xác suất trên các khoảng do Lakshmanan và CS. (1997) đề xuất trong [14], được Eiter và CS. (2001) mở rộng trong [19].

Định nghĩa 2.2.2 Giả sử e_1 và e_2 có xác suất tương ứng trong các khoảng $I_1 = [L_1, U_1]$ và $I_2 = [L_2, U_2]$. Một *chiến lược hội xác suất* của I_1 và I_2 là một phép toán hai ngôi \otimes sử dụng các khoảng xác suất này để tính một khoảng xác suất $I = [L, U]$ cho $e_1 \wedge e_2$, được ký hiệu bởi $I_1 \otimes I_2$, thỏa các tiên đề trong Bảng 2.2.1.

Bảng 2.2.1 Các tiên đề về chiến lược hội

Tên tiên đề (Axiom name)	Chiến lược hội
Bị chặn (Bottomline)	$(I_1 \otimes I_2) \leq [\min(L_1, L_2), \min(U_1, U_2)]$
Bỏ qua (Ignorance)	$(I_1 \otimes I_2) \subseteq [\max(0, L_1 + L_2 - 1), \min(U_1, U_2)]$
Đồng nhất (Identity)	$(I_1 \otimes [1, 1]) = I_1$
Giao hoán (Commutativity)	$(I_1 \otimes I_2) = (I_2 \otimes I_1)$
Kết hợp (Associativity)	$((I_1 \otimes I_2) \otimes I_3) = (I_1 \otimes (I_2 \otimes I_3))$
Đơn điệu (Monotonicity)	$(I_1 \otimes I_2) \leq (I_1 \otimes I_3)$ nếu $I_2 \leq I_3$

Định nghĩa 2.2.3 Giả sử e_1 và e_2 có xác suất tương ứng trong các khoảng $I_1 = [L_1, U_1]$ và $I_2 = [L_2, U_2]$. Một chiến lược tuyển xác suất của I_1 và I_2 là một phép toán hai ngôi \oplus sử dụng các khoảng xác suất này để tính một khoảng xác suất $I = [L, U]$ cho $e_1 \vee e_2$, được ký hiệu bởi $I_1 \oplus I_2$, thỏa các tiên đề trong Bảng 2.2.2.

Bảng 2.2.2 Các tiên đề về chiến lược tuyển

Tên tiên đề	Chiến lược tuyển
Bị chặn	$(I_1 \oplus I_2) \geq [\min(L_1, L_2), \min(U_1, U_2)]$
Bỏ qua	$(I_1 \oplus I_2) \subseteq [\max(L_1, L_2), \min(1, U_1 + U_2)]$
Đồng nhất	$(I_1 \oplus [0, 0]) = I_1$
Giao hoán	$(I_1 \oplus I_2) = (I_2 \oplus I_1)$
Kết hợp	$((I_1 \oplus I_2) \oplus I_3) = (I_1 \oplus (I_2 \oplus I_3))$
Đơn điệu	$(I_1 \oplus I_2) \leq (I_1 \oplus I_3)$ nếu $I_2 \leq I_3$

Ngoài ra, chúng tôi cũng sử dụng các chiến lược hiệu (difference strategy) của các khoảng xác suất do Eiter và CS. (2001) đề xuất như định nghĩa sau.

Định nghĩa 2.2.4 Giả sử e_1 và e_2 có xác suất tương ứng trong các khoảng $I_1 = [L_1, U_1]$ và $I_2 = [L_2, U_2]$. Một chiến lược hiệu xác suất của I_1 và I_2 là một phép toán hai ngôi \ominus sử dụng các khoảng xác suất này để tính một khoảng xác suất $I = [L, U]$ cho sự kiện $e_1 \wedge \neg e_2$, được ký hiệu bởi $I = I_1 \ominus I_2$, thỏa các tiên đề sau:

1. Bị chặn: $(I_1 \ominus I_2) \leq [\min(L_1, 1 - U_2), \min(U_1, 1 - L_2)]$.
2. Bỏ qua: $(I_1 \ominus I_2) \subseteq [\max(0, L_1 - U_2), \min(U_1, 1 - L_2)]$.
3. Đồng nhất: nếu $(\neg e_1 \wedge \neg e_2)$ và $(e_1 \wedge \neg e_2)$ là không mâu thuẫn thì $(I_1 \ominus [0, 0]) = I_1$.

Bảng 2.2.3 là một số ví dụ về các chiến lược kết hợp hội, tuyển và hiệu các khoảng xác suất trong Eiter và CS. (2001).

Bảng 2.2.3 Các ví dụ về các chiến lược kết hợp xác suất

Chiến lược	Phép toán
Bỏ qua (Ignorance)	$([L_1, U_1] \otimes_{ig}[L_2, U_2]) \equiv [\max(0, L_1 + L_2 - 1), \min(U_1, U_2)]$ $([L_1, U_1] \oplus_{ig}[L_2, U_2]) \equiv [\max(L_1, L_2), \min(1, U_1 + U_2)]$ $([L_1, U_1] \ominus_{ig}[L_2, U_2]) \equiv [\max(0, L_1 - U_2), \min(U_1, 1 - L_2)]$
Độc lập (Independence)	$([L_1, U_1] \otimes_{in}[L_2, U_2]) \equiv [L_1.L_2, U_1.U_2]$ $([L_1, U_1] \oplus_{in}[L_2, U_2]) \equiv [L_1 + L_2 - (L_1.L_2), U_1 + U_2 - (U_1.U_2)]$ $([L_1, U_1] \ominus_{in}[L_2, U_2]) \equiv [L_1.(1 - U_2), U_1.(1 - L_2)]$
Tương quan thuận (Positive correlation)	$([L_1, U_1] \otimes_{pc}[L_2, U_2]) \equiv [\min(L_1, L_2), \min(U_1, U_2)]$ $([L_1, U_1] \oplus_{pc}[L_2, U_2]) \equiv [\max(L_1, L_2), \max(U_1, U_2)]$ $([L_1, U_1] \ominus_{pc}[L_2, U_2]) \equiv [\max(0, L_1 - U_2), \max(0, U_1 - L_2)]$
Loại trừ nhau (Mutual Exclusion)	$([L_1, U_1] \otimes_{me}[L_2, U_2]) \equiv [0, 0]$ $([L_1, U_1] \oplus_{me}[L_2, U_2]) \equiv [\min(1, L_1 + L_2), \min(1, U_1 + U_2)]$ $([L_1, U_1] \ominus_{me}[L_2, U_2]) \equiv [L_1, \min(U_1, 1 - L_2)]$

Để đơn giản, kể từ bây giờ chúng tôi gọi các chiến lược kết hợp các khoảng xác suất là các chiến lược kết hợp xác suất.

2.3. Các hàm phân bố và bộ ba xác suất

Để mở rộng CSDL quan hệ truyền thống thành CSDL quan hệ xác suất URDB, chúng tôi đã sử dụng các hàm phân bố xác suất trong [19] và bộ ba xác suất mở rộng trong [28] với mỗi phần tử có thể là một tập hợp làm cơ sở cho việc biểu diễn giá trị không chắc chắn của các thuộc tính quan hệ. Khái niệm *hàm phân bố xác suất* (probability distribution function) và *bộ ba xác suất* (probabilistic triple) lần lượt được định nghĩa dưới đây.

Định nghĩa 2.3.1 Giả sử \mathcal{D} là một tập hữu hạn, một *hàm phân bố xác suất* α trên \mathcal{D} là một ánh xạ $\alpha : \mathcal{D} \rightarrow [0, 1]$ sao cho $\sum_{x \in \mathcal{D}} \alpha(x) \leq 1$

Một hàm phân bố xác suất quan trọng thường gặp là *hàm phân bố đều* (uniform distribution) $u(x) = 1/|\mathcal{D}|, \forall x \in \mathcal{D}$. Ví dụ, nếu $\mathcal{D} = \{24, 48, 72\}$, thì hàm phân bố đều u trên \mathcal{D} là $u(x) = 1/3, \forall x \in \{24, 48, 72\}$.

Định nghĩa 2.3.2 Giả sử X là một tập hữu hạn, một *bộ ba xác suất* (probabilistic triple) $\langle V, \alpha, \beta \rangle$ trên X bao gồm một tập hữu hạn V các tập con của X (nghĩa là $V \subseteq 2^X$) sao cho hai phần tử (tập hợp) bất kỳ trong V không giao nhau, một hàm phân bố xác suất α trên V và một hàm $\beta: V \rightarrow [0, 1]$ sao cho $\alpha(x) \leq \beta(x), \forall x \in V$.

Như vậy, một bộ ba xác suất $\langle V, \alpha, \beta \rangle$ gán mỗi $x \in V$ cho một khoảng $[\alpha(x), \beta(x)]$ biểu diễn xác suất không chắc chắn của x trong V . Phép gán này là nhất quán theo nghĩa mỗi $x \in V$ được gán một xác suất $p(x) \in [\alpha(x), \beta(x)]$ sao cho $\sum_{x \in V} p(x) = 1$.

Chúng tôi lưu ý rằng mỗi phần tử $v \in X$ cũng có thể được xem như là một tập hợp trên X (tập con của X), nghĩa là cách viết v và $\{v\}$ là như nhau. Vì vậy, một bộ ba xác suất $\langle \{\{v_1\}, \{v_2\}, \dots, \{v_k\}\}, \alpha, \beta \rangle$ cũng có thể được viết đơn giản là $\langle \{v_1, v_2, \dots, v_k\}, \alpha, \beta \rangle$. Bộ ba xác suất trong Định nghĩa 2.3.2 là một mở rộng của bộ ba xác suất trong [19] và [23] vì cho phép các phần tử trong V là các tập hợp trên X . Sự mở rộng này cho phép biểu diễn giá trị không chắc chắn của các thuộc tính quan hệ một cách mềm dẻo và tổng quát hơn. Để đơn giản, “bộ ba xác suất mở rộng” cũng được gọi là “bộ ba xác suất”.

Khái niệm bộ ba xác suất là công cụ để biểu diễn thông tin không chắc chắn về các đối tượng trong thực tế. Chẳng hạn, khi một bác sĩ khám bệnh cho một bệnh nhân, bác sĩ có thể không biết chắc chắn bệnh nhân bị bệnh gì. Tuy nhiên, qua triệu chứng, nếu bác sĩ chẩn đoán rằng 40% đến 60% khả năng bệnh nhân này có thể bị bệnh viêm gan (hepatitis) và xơ gan (cirrhosis) hoặc bị bệnh viêm túi mật (cholecystitis), thì thông tin chẩn đoán bệnh này có thể được biểu diễn bởi bộ ba xác suất $\langle \{\{\text{hepatitis, cirrhosis}\}, \{\text{cholecystitis}\}\}, 0.8u, 1.2u \rangle$. Trong đó, u là hàm phân bố đều trên $\{\{\text{hepatitis, cirrhosis}\}, \{\text{cholecystitis}\}\}, 0.8u$ và $1.2u$ tương ứng là các hàm phân bố xác suất α và β với $\alpha(x) = 0.8u(x) = 0.8(1/2) = 0.4$ và $\beta(x) = 1.2u(x) = 1.2(1/2) = 0.6, \forall x \in \{\{\text{hepatitis, cirrhosis}\}, \{\text{cholecystitis}\}\}$. Lưu ý rằng, theo Định nghĩa 2.3.2 của bộ ba xác suất $\langle V, \alpha, \beta \rangle$, α và β có thể là các hàm phân bố xác suất bất kỳ.

2.4. Các chiến lược kết hợp các bộ ba xác suất

Các phép toán đại số như kết, giao, hợp và trừ của các quan hệ trong cơ sở dữ liệu quan hệ xác suất URDB được xây dựng bằng cách sử dụng các chiến lược kết hợp của các bộ ba xác suất mở rộng trong [28] với tập các tập hợp làm cơ sở cho sự kết hợp xác suất của các giá trị các thuộc tính trong các quan hệ kết quả của các phép toán này. Trước hết là chiến lược hội của các bộ ba xác suất như định nghĩa sau.

Định nghĩa 2.4.1 Giả sử $pt_1 = \langle V_1, \alpha_1, \beta_1 \rangle$ và $pt_2 = \langle V_2, \alpha_2, \beta_2 \rangle$ là các bộ ba xác suất, và \otimes là một chiến lược hội xác suất. Thì một *chiến lược hội của hai bộ ba xác suất* (conjunction strategy of two probabilistic triples) pt_1 và pt_2 theo \otimes , được ký hiệu $pt_1 \otimes pt_2$, là một bộ ba xác suất $pt = \langle V, \alpha, \beta \rangle$, trong đó:

1. $V = \{v = v_1 \cap v_2 \mid v_1 \in V_1, v_2 \in V_2 \text{ và } [\alpha(v), \beta(v)] = [\alpha_1(v_1), \beta_1(v_1)] \otimes [\alpha_2(v_2), \beta_2(v_2)] \neq [0, 0]\}$.
2. $[\alpha(v), \beta(v)] = [\alpha_1(v_1), \beta_1(v_1)] \otimes [\alpha_2(v_2), \beta_2(v_2)]$

Ví dụ 2.4.1 Cho $pt_1 = \langle \{48, 72\}, 0.8u, 1.2u \rangle$, $pt_2 = \langle \{72, 96\}, u, u \rangle$ là các bộ ba xác suất. Khi đó, $pt_1 \otimes_{in} pt_2$ theo chiến lược hội độc lập là bộ ba xác suất $pt = \langle \{72\}, 0.2u, 0.3u \rangle$.

Tiếp theo, các chiến lược tuyển và trừ các bộ ba xác suất lần lượt được định nghĩa như dưới đây.

Định nghĩa 2.4.2 Giả sử $pt_1 = \langle V_1, \alpha_1, \beta_1 \rangle$ và $pt_2 = \langle V_2, \alpha_2, \beta_2 \rangle$ là các bộ ba xác suất, và \oplus là một chiến lược tuyển xác suất. Thì một *chiến lược tuyển của hai bộ ba xác suất* (disjunction strategy of two probabilistic triples) pt_1 và pt_2 theo \oplus , được ký hiệu $pt_1 \oplus pt_2$, là một bộ ba xác suất $pt = \langle V, \alpha, \beta \rangle$, trong đó:

1. $V = P \cup Q \cup R$, trong đó $P = \{v_1 \in V_1 \mid \neg \exists v_2 \in V_2, v_1 \cap v_2 \neq \emptyset\}$, $Q = \{v_2 \in V_2 \mid \neg \exists v_1 \in V_1, v_1 \cap v_2 \neq \emptyset\}$, $R = \{v_1 \cap v_2 \mid v_1 \in V_1, v_2 \in V_2, v_1 \cap v_2 \neq \emptyset\}$.
2. $[\alpha(v), \beta(v)] = \begin{cases} [\alpha_1(v), \beta_1(v)], \forall v \in P \\ [\alpha_2(v), \beta_2(v)], \forall v \in Q \\ [\alpha_1(v_1), \beta_1(v_1)] \oplus [\alpha_2(v_2), \beta_2(v_2)], \forall v = v_1 \cap v_2 \in R. \end{cases}$

Ví dụ 2.4.2 Cho $pt_1 = \langle \{48, 72\}, 0.8u, 1.2u \rangle$, $pt_2 = \langle \{72, 96\}, u, u \rangle$ là các bộ ba xác suất. Khi đó, $pt_1 \oplus_{in} pt_2$ theo chiến lược tuyển độc lập là bộ ba xác suất $pt = \langle \{48, 72, 96\}, \alpha, \beta \rangle$ với $\alpha(48) = 0.4$, $\beta(48) = 0.6$, $\alpha(96) = \beta(96) = 0.5$, $\alpha(72) = 0.7$, $\beta(72) = 0.8$.

Định nghĩa 2.4.3 Giả sử $pt_1 = \langle V_1, \alpha_1, \beta_1 \rangle$ và $pt_2 = \langle V_2, \alpha_2, \beta_2 \rangle$ là các bộ ba xác suất, và \ominus là một chiến lược hiệu xác suất. Thì một *chiến lược hiệu của hai bộ ba xác suất* (difference strategy of two probabilistic triples) pt_1 và pt_2 được ký hiệu $pt_1 \ominus pt_2$, là một bộ ba xác suất $pt = \langle V, \alpha, \beta \rangle$, trong đó:

1. $V = P \cup Q$, trong đó $P = \{v_1 \in V_1 \mid \neg \exists v_2 \in V_2, v_1 \cap v_2 \neq \emptyset\}$, $Q = \{v_1 \cap v_2 \mid v_1 \in V_1, v_2 \in V_2, v_1 \cap v_2 \neq \emptyset \text{ và } [\alpha_1(v_1), \beta_1(v_1)] \ominus [\alpha_2(v_2), \beta_2(v_2)] \neq [0, 0]\}$,
2. $[\alpha(v), \beta(v)] = \begin{cases} [\alpha_1(v), \beta_1(v)], & \forall v \in P \\ [\alpha_1(v_1), \beta_1(v_1)] \ominus [\alpha_2(v_2), \beta_2(v_2)], & \forall v = v_1 \cap v_2 \in Q. \end{cases}$

Ví dụ 2.4.3 Cho $pt_1 = \langle \{48, 72\}, 0.8u, 1.2u \rangle$, $pt_2 = \langle \{72, 96\}, u, u \rangle$ là các bộ ba xác suất. Khi đó, $pt_1 \ominus_{ig} pt_2$ theo chiến lược hiệu bỏ qua là bộ ba xác suất $pt = \langle \{48, 72\}, \alpha, \beta \rangle$ với $\alpha(48) = 0.4$, $\beta(48) = 0.6$, $\alpha(72) = 0.0$, $\beta(72) = 0.5$.

2.5. Diễn dịch xác suất của quan hệ trên các tập hợp

Để thực hiện việc tính toán xác suất của một quan hệ hai ngôi trên các giá trị thuộc tính trong URBD, chúng tôi sử dụng diễn dịch xác suất của quan hệ hai ngôi trên các tập hợp trong [28] như định nghĩa dưới đây.

Định nghĩa 2.5.1 Giả sử A và B là các tập hợp, U và V là các miền giá trị, và θ là một quan hệ hai ngôi trong $\{=, \neq, \leq, \geq, <, >, \subseteq, \supseteq\}$. *Diễn dịch xác suất của quan hệ $A \theta B$* , được ký hiệu $Pr(A \theta B)$, là một giá trị nằm trong khoảng $[0, 1]$ được định nghĩa bởi:

1. $Pr(A \theta B) = p(u \theta v \mid u \in A, v \in B)$, với A là một tập con của U , B là một tập con của V và $\theta \in \{=, \neq, \leq, \geq, <, >\}$ được giả sử là hợp lệ trên $(U \times V)$, $p(u \theta v \mid u \in A, v \in B)$ là xác suất có điều kiện của $u \theta v$ với $u \in A$ và $v \in B$.
2. $Pr(A \theta B) = \begin{cases} p(u \in B \mid u \in A), & \theta \text{ là quan hệ } \subseteq \\ p(u \in A \mid u \in B), & \theta \text{ là quan hệ } \supseteq \end{cases}$

với A và B là hai tập con của U , $p(u \in B | u \in A)$ là xác suất có điều kiện để $u \in B$ khi $u \in A$ và $p(u \in A | u \in B)$ là xác suất có điều kiện để $u \in A$ khi $u \in B$.

Chúng tôi lưu ý rằng, diễn dịch xác suất của các quan hệ hai ngôi trên các tập hợp được định nghĩa ở đây là một mở rộng của định nghĩa tương ứng trong [24] với các quan hệ “ \subseteq ” và “ \supseteq ”, trong khi diễn dịch các quan hệ hai ngôi trên các tập hợp không được định nghĩa trong [19] và [23].

Ví dụ 2.5.1 Giả sử $A = \{3, 4\}$ và $B = \{4, 5\}$ là hai tập hợp trên miền giá trị $\{1, 2, 3, 4, 5, 6\}$. Khi đó:

1. $Pr(A = B) = p(u = v / u \in A, v \in B)$
 $= p(u=v / u \in \{3, 4\}, v \in \{4, 5\}) = 0.25.$
2. $Pr(A \subseteq B) = p(u \in B / u \in A)$
 $= p(u \in \{4, 5\} | u \in \{3, 4\}) = 0.5.$

2.6. Kết luận

Trong Chương 2, các khái niệm cơ bản về lý thuyết xác suất làm cơ sở toán học cho quá trình xây dựng mô hình URDB đã được giới thiệu. Các hàm phân bố và bộ ba xác suất được dùng để xây dựng mô hình dữ liệu của URDB trong chương 3. Các chiến lược kết hợp các khoảng xác suất là mở rộng các chiến lược kết hợp xác suất của các sự kiện trong lý thuyết xác suất cổ điển. Trong khi các chiến lược kết hợp các bộ ba xác suất là sự kết hợp các khoảng xác suất tương ứng với các tập giá trị trong các bộ ba xác suất. Các chiến lược kết hợp xác suất được ứng dụng để xây dựng các phép toán đại số trong chương 4.

Chương 3

LƯỢC ĐỒ VÀ QUAN HỆ CỦA MÔ HÌNH URDB

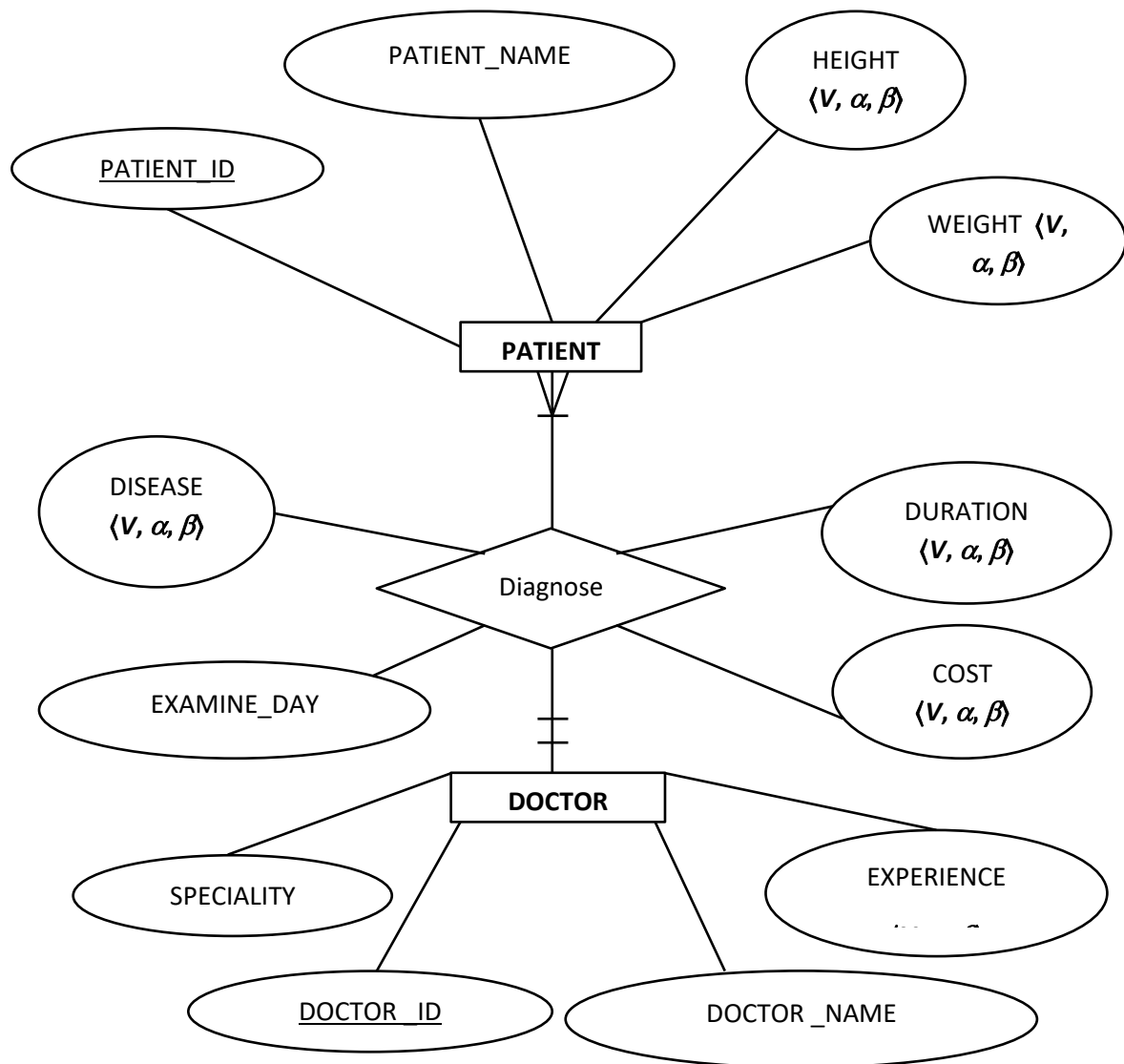
3.1. Giới thiệu

Chương này trình bày những khái niệm nền tảng của mô hình cơ sở dữ liệu quan hệ xác suất URDB. Phần 3.2 giới thiệu khái quát mô hình ý niệm URDB, một cái nhìn có tính trực quan, giản lược và không hình thức về những đặc trưng và khả năng mô hình hóa thông tin không chắc chắn của URDB. Phần 3.3 trình bày khái niệm thuộc tính quan hệ, một khái niệm rất cơ bản trong mô hình cơ sở dữ liệu quan hệ nói chung và quan hệ xác suất URDB nói riêng. Phần 3.4 là mở rộng khái niệm kiểu và giá trị trong mô hình CSDL quan hệ truyền thống với *thuộc tính đa trị không chắc chắn* (uncertain multivalued attribute) thành kiểu và giá trị trong URDB. Kế đến, trong Phần 3.5 và 3.6, lược đồ, quan hệ và phụ thuộc hàm của URDB sẽ được định nghĩa bằng cách mở rộng các khái niệm lược đồ, quan hệ và phụ thuộc hàm trong CSDL quan hệ truyền thống với tập thuộc tính có thể nhận giá trị không chắc chắn. Cuối cùng Phần 3.7 là một vài lưu ý và kết luận của chương này.

3.2. Mô hình ý niệm

Một cách khái quát, URDB là một mở rộng của mô hình cơ sở dữ liệu quan hệ truyền thống để có thể biểu diễn được thông tin không chắc chắn và không đầy đủ của các quan hệ (đối tượng). Một mô hình như vậy sẽ có rất nhiều áp dụng trong thực tế. Chẳng hạn, một cơ sở dữ liệu các bệnh nhân tại một phòng khám của một bệnh viện có thể được mô hình hóa bởi URDB. Trong cơ sở dữ liệu này, căn bệnh của mỗi bệnh nhân không phải luôn luôn được bác sĩ xác định một cách chắc chắn. Tương tự, thời gian điều

trị, chi phí điều trị của mỗi bệnh nhân nói chung cũng không biết rõ là bao nhiêu ngay cả khi bệnh nhân biết được căn bệnh của họ.



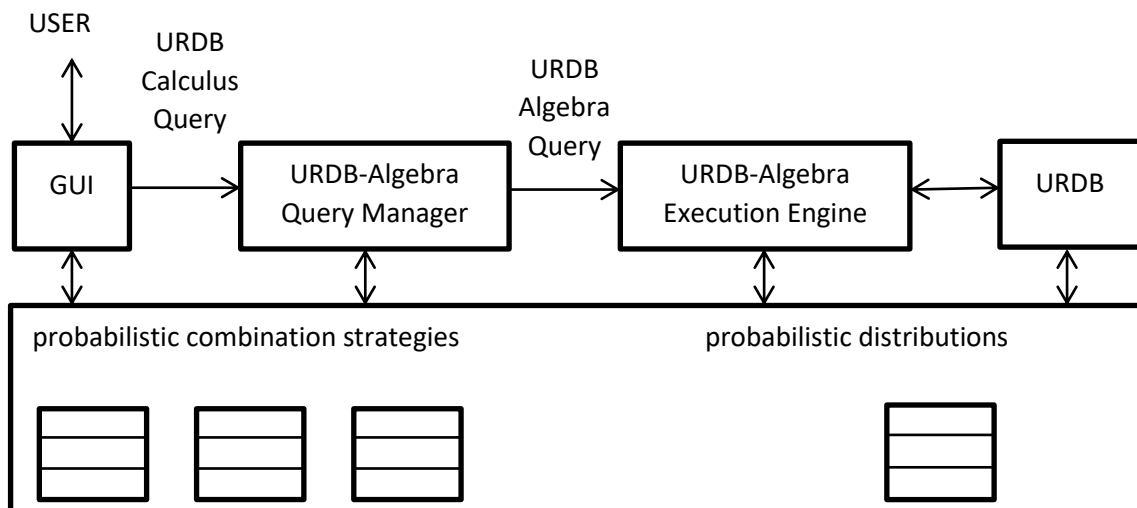
Hình 3.2.1 Cơ sở dữ liệu khám-chữa bệnh

Hình 3.2.1 cho thấy một lược đồ (ý niệm) cơ sở dữ liệu xác suất đơn giản biểu diễn mối quan hệ của bệnh nhân và bác sĩ khám bệnh với các thực thể có các thuộc tính có giá trị không chắc chắn. Chẳng hạn, các thuộc tính được kết hợp với một bộ ba xác suất như DISEASE $\langle v, \alpha, \beta \rangle$, DURATION $\langle v, \alpha, \beta \rangle$ hay COST $\langle v, \alpha, \beta \rangle$ v.v, là các thuộc tính có giá trị không chắc chắn.

Hình 3.2.2 mô tả kiến trúc hệ của thống URDB. Hệ thống này bảo đảm quá trình xử lý các truy vấn URDB, bao gồm một tập các lược đồ và quan hệ tương ứng có thuộc

tính có thể nhận các giá trị không chắc chắn được biểu diễn bởi các bộ ba xác suất. Cụ thể, người sử dụng biểu diễn các truy vấn trong ngôn ngữ URDB thông qua một giao diện đồ họa. Các truy vấn như vậy được nhận dạng và biến đổi bởi bộ quản lý truy vấn đại số URDB thành các truy vấn ngôn ngữ đại số của hệ thống URDB. Sau đó chúng sẽ được bộ thực thi tính toán và xử lý dựa trên dữ liệu trong URDB để trả về kết quả truy vấn. Tất cả các thành phần trên tham khảo một thư viện bao gồm:

1. Một tập các chiến lược kết hợp xác suất cho phép người dùng biểu diễn thông tin về sự phụ thuộc của các sự kiện.
2. Một tập các hàm phân bố xác suất cho phép người dùng biểu diễn xác suất được phân bố trên một không gian các giá trị thuộc tính.



Hình 3.2.2. Kiến trúc của hệ thống URDB

3.3. Thuộc tính quan hệ

Thuộc tính (attribute) quan hệ, là một trong những khái niệm trung tâm của mô hình cơ sở dữ liệu quan hệ truyền thống ([1], [2], [3]). Theo đó các thuộc tính thể hiện thông tin về trạng thái của đối tượng. Một quan hệ được định nghĩa bởi một tập thuộc tính xác định tính chất đặc trưng của một tập đối tượng nào đó. Thông tin về đối tượng hoàn toàn được xác định khi ta biết giá trị thuộc tính của nó. Trong URDB, thuộc tính có thể nhận một giá trị kết hợp với một xác suất biểu diễn tính không chắc chắn về thông tin của đối tượng. Để đơn giản, chúng tôi gọi các thuộc tính như vậy là *thuộc tính không*

chắc chắn (uncertain attribute). Chẳng hạn, trong CSDL các bệnh nhân, các thuộc tính DISEASE và DURATION biểu thị căn bệnh và thời gian điều trị bệnh của bệnh nhân là không chắc chắn. Bởi vì bác sĩ có thể không chắc chắn bệnh nhân bị bệnh gì và cũng không chắc chắn thời gian điều trị là bao lâu. Chúng tôi lưu ý rằng, một thuộc tính nhận một giá trị xác định, chắc chắn (nghĩa là xác suất nhận giá trị đó bằng 1), chẳng hạn như thuộc tính khóa quan hệ, cũng được coi như là không chắc chắn (với mức độ không chắc chắn bằng 0, nghĩa là mức độ chắc chắn bằng 1). Vì vậy, trong URDB, mọi thuộc tính đều có thể coi là không chắc chắn.

3.4. Kiểu và giá trị

Tương tự như mô hình cơ sở dữ liệu quan hệ truyền thống, trong URDB, mỗi quan hệ được đặc trưng bởi một số thuộc tính mà các giá trị của chúng có các kiểu tương ứng nào đó. Đối với URDB, hệ thống kiểu và giá trị của thuộc tính được mở rộng cho cả dữ liệu không chắc chắn như trong các định nghĩa dưới đây.

Định nghĩa 3.4.1 Giả sử A là một tập các thuộc tính mà mỗi $A \in A$ có thể không chắc chắn và T là tập các *kiểu cơ sở* (atomic type). Các kiểu thuộc tính được định nghĩa như sau:

1. Mọi kiểu cơ sở trong T là một kiểu.
2. Nếu τ là một kiểu, thì $\{\tau\}$ là một kiểu được gọi là *kiểu tập hợp* (set type) của τ .
3. Nếu A_1, A_2, \dots, A_k là các thuộc tính đôi một khác nhau trong \mathcal{A} và $\tau_1, \tau_2, \dots, \tau_k$ là các kiểu thì $\tau = [A_1: \tau_1, A_2: \tau_2, \dots, A_k: \tau_k]$ là một kiểu, được gọi là *kiểu bộ* (tuple type) trên tập các thuộc tính $\{A_1, A_2, \dots, A_k\}$. Với một kiểu $\tau = [A_1: \tau_1, A_2: \tau_2, \dots, A_k: \tau_k]$, chúng tôi sử dụng $\tau.A_i$ hoặc $\tau[A_i]$ để biểu thị τ_i .

Ví dụ 3.4.1 Trong CSDL các bệnh nhân trên, một số thuộc tính có thể là PATIENT_NAME, BIRTHDAY, CHECK_DATE, MEDICAL_HISTORY, DISEASE mô tả thông tin về tên, ngày sinh, ngày khám, lịch sử bệnh và loại bệnh của mỗi bệnh nhân. Một số thuộc tính khác có thể là DURATION, COST định nghĩa thời gian điều trị và chi phí điều trị mỗi ngày của các bệnh nhân. Một số kiểu cơ sở có thể là **string**, **datetime** và **integer**. Các kiểu tập hợp và kiểu bộ có thể là $\{\mathbf{string}\}$, $[\mathbf{PATIENT_ID}$:

string, PATIENT_NAME: **string**, BIRTHDAY: **datetime**, SEX: **binary**, HEIGHT: **integer**, WEIGHT: **integer**, MEDICAL_HISTORY: {**string**}].

Tương tự như trong mô hình cơ sở dữ liệu quan hệ truyền thống, trong URDB, mỗi kiểu có một miền giá trị kết hợp với nó như định nghĩa sau đây.

Định nghĩa 3.4.2 Mỗi kiểu cơ bản $\tau \in T$ có một miền xác định $dom(\tau)$ kết hợp với nó. *Giá trị* (value) được định nghĩa như sau:

1. Với mọi kiểu cơ bản $\tau \in T$, thì mọi $v \in dom(\tau)$ là một giá trị kiểu τ .
2. Nếu v_1, v_2, \dots, v_k là các giá trị thuộc kiểu τ , thì $\{v_1, v_2, \dots, v_k\}$ là một giá trị kiểu $\{\tau\}$.
3. Nếu A_1, \dots, A_k là các thuộc tính đôi một khác nhau trong A và v_1, \dots, v_k là các giá trị tương ứng của các kiểu τ_1, \dots, τ_k thì $[A_1: v_1, \dots, A_k: v_k]$ là một giá trị kiểu $[A_1: \tau_1, \dots, A_k: \tau_k]$, được gọi là *giá trị kiểu bộ* (tuple type value) trên tập các thuộc tính $\{A_1, A_2, \dots, A_k\}$. Đối với một giá trị $v = [A_1: v_1, \dots, A_k: v_k]$, chúng tôi sử dụng ký hiệu $v.A_i$ hoặc $v[A_i]$ để biểu thị giá trị của thuộc tính A_i .

Trong một ngữ cảnh nào đó, các thuộc tính có thể được bỏ qua nếu thấy không nhất thiết phải kể đến chúng, thì một giá trị kiểu bộ có thể viết đơn giản $v = (v_1, v_2, \dots, v_k)$.

Ví dụ 3.4.2 Xét cơ sở dữ liệu bệnh nhân tại phòng khám như đã nêu ở trên, một giá trị kiểu bộ có thể là $v = [PATIENT_NAME: Ho\ V.T, BIRTHDAY: 03/15/1969, CHECK_DATE: 20/12/2021, MEDICAL_HISTORY: \{cholecystitis\}]$.

Định nghĩa 3.4.3 Giả sử A_1, A_2, \dots, A_k là các thuộc tính đôi một khác nhau trong A và $\langle V_1, \alpha_1, \beta_1 \rangle, \dots, \langle V_k, \alpha_k, \beta_k \rangle$ là các bộ ba xác suất, trong đó V_1, \dots, V_k là tập các giá trị kiểu τ_1, \dots, τ_k , thì biểu thức $[A_1: \langle V_1, \alpha_1, \beta_1 \rangle, \dots, A_k: \langle V_k, \alpha_k, \beta_k \rangle]$ là một *giá trị bộ xác suất* (probabilistic tuple value) kiểu $[A_1: \tau_1, \dots, A_k: \tau_k]$ trên tập các thuộc tính $\{A_1, \dots, A_k\}$. Với mỗi bộ giá trị xác suất $ptv = [A_1: \langle V_1, \alpha_1, \beta_1 \rangle, \dots, A_k: \langle V_k, \alpha_k, \beta_k \rangle]$, chúng tôi sử dụng ký hiệu $ptv.A_i$ hoặc $ptv[A_i]$ để biểu thị $\langle V_i, \alpha_i, \beta_i \rangle$.

Chúng tôi lưu ý rằng trật tự các $A_i: \langle V_i, \alpha_i, \beta_i \rangle$ trong $ptv = [A_1: \langle V_1, \alpha_1, \beta_1 \rangle, \dots, A_k: \langle V_k, \alpha_k, \beta_k \rangle]$ là không quan trọng. Để đơn giản, chúng ta có thể gọi *bộ* (tuple) (như trong mô hình CSDL quan hệ truyền thống) thay cho giá trị bộ xác suất trong ngữ cảnh chỉ

đề cập đến mô hình URDB. Khái niệm giá trị bộ xác suất cho phép chúng ta biểu diễn một cách thích hợp tính không chắc chắn của các thông tin dữ liệu trong URDB. Ngoài ra, trong một ngữ cảnh nào đó, một giá trị bộ xác suất có thể được viết đơn giản $ptv = (\langle V_1, \alpha_1, \beta_1 \rangle, \dots, \langle V_k, \alpha_k, \beta_k \rangle)$, nếu thấy không nhất thiết phải kể đến các thuộc tính tương ứng của chúng và ký hiệu $[ptv]$ được sử dụng để biểu thị (V_1, \dots, V_k) như là giá trị của bộ ptv .

Ví dụ 3.4.3 Xét tình huống một bệnh nhân có tên là Hồ V.T, ngày sinh 03/15/1969 được bác sĩ khám bệnh tại phòng khám. Bác sĩ chưa biết chắc chắn bệnh nhân bị bệnh gì (điều vẫn thường xảy ra khi chẩn đoán lần đầu). Tuy nhiên, qua các triệu chứng bệnh của bệnh nhân và bằng kinh nghiệm thực tế, ông có thể đưa ra phán đoán là 50% khả năng bệnh nhân này bị bệnh viêm gan (hepatitis) và xơ gan (cirrhosis) hoặc bị bệnh viêm túi mật (cholecystitis). Ngoài ra, ông cũng cho bệnh nhân biết chi phí điều trị mỗi ngày là 60 hoặc 70 nghìn đồng và ước đoán thời gian điều trị là 30 hoặc 32 ngày với một xác suất nằm trong khoảng từ 40 đến 60%, thì thông tin này có thể được biểu diễn bởi một giá trị bộ xác suất $[PATIENT_NAME: \langle \{Ho\ V.T\}, u, u \rangle, BIRTHDAY: \langle \{03/15/1969\}, u, u \rangle, DISEASE: \langle \{hepatitis, cirrhosis\}, \{cholecystitis\} \rangle, u, u \rangle, DURATION: \langle \{30, 32\}, 0.8u, 1.2u \rangle, COST: \langle \{60, 70\}, u, u \rangle]$ Trong đó u là phân bố xác suất đều.

3.5. Lược đồ và quan hệ

Như đã giới thiệu một cách khái quát trong Phần 3.2, một lược đồ URDB mô tả một tập các thuộc tính của một tập các đối tượng nào đó có thể có thông tin không chắc chắn. Lược đồ quan hệ trong URDB ([28]) được mở rộng từ lược đồ quan hệ trong CSDL quan hệ truyền thống và trong ([22], [23]) với các thuộc tính có thể có giá trị tập hợp và không chắc chắn như định nghĩa sau.

Định nghĩa 3.5.1 Một lược đồ quan hệ xác suất (probabilistic relational schema) là một cặp $R = (U, \wp)$, trong đó

1. $U = \{A_1, A_2, \dots, A_k\}$ là một tập các thuộc tính đôi một khác nhau.
2. \wp là một ánh xạ gán mỗi thuộc tính $A \in U$ cho tập tất cả các bộ ba xác suất trên miền giá trị của A (nghĩa là mỗi phần tử của $\wp(A)$ là một bộ ba xác suất có dạng $\langle V, \alpha, \beta \rangle$, trong đó V là một tập hữu hạn các tập con của miền giá trị của A).

Lưu ý rằng, như trong CSDL quan hệ cổ điển, để đơn giản, các ký hiệu $R(U, \wp)$ và R có thể được sử dụng thay cho $R = (U, \wp)$. Miền giá trị của thuộc tính A được ký hiệu là $dom(A)$.

Ví dụ 3.5.1 Một lược đồ của quan hệ xác suất **PATIENT** trong URDB có thể như sau:

PATIENT(P_ID, P_NAME, P_AGE, P_DISEASE, D_COST, \wp).

Ở đây, các thuộc tính P_AGE (tuổi), P_DISEASE (bệnh) và D_COST (chi phí điều trị hàng ngày) có thể có giá trị không chắc chắn, \wp là ánh xạ gán mỗi thuộc tính trong **PATIENT** cho tập tất cả các bộ ba xác suất trên miền giá trị của các thuộc tính này.

Một quan hệ xác suất biểu diễn giá trị (hoặc một phần giá trị) không chắc chắn của một tập đối tượng trong thực tế. Quan hệ xác suất cũng được mở rộng từ quan hệ truyền thống (Định nghĩa 1.2.4) với giá trị bộ xác suất như định nghĩa dưới đây.

Định nghĩa 3.5.2 Giả sử $U = \{A_1, A_2, \dots, A_k\}$ là một tập k thuộc tính đôi một khác nhau. Một *quan hệ xác suất* (probabilistic relation) r trên lược đồ quan hệ xác suất $R(U, \wp)$, là một tập hữu hạn các bộ $\{t \mid t = (\langle V_1, \alpha_1, \beta_1 \rangle, \langle V_2, \alpha_2, \beta_2 \rangle, \dots, \langle V_k, \alpha_k, \beta_k \rangle)\}$ trong đó mỗi $\langle V_i, \alpha_i, \beta_i \rangle$ là một bộ ba xác suất thuộc tập $f_i = \wp(A_i)$ và $V_i \neq \emptyset$, với mọi $i = 1, 2, \dots, k$ (nghĩa là, mỗi t là một giá trị bộ xác suất trên tập các thuộc tính $\{A_1, \dots, A_k\}$).

Mỗi bộ ba xác suất $\langle V_i, \alpha_i, \beta_i \rangle$ biểu diễn giá trị không chắc chắn của thuộc tính A_i của bộ t , các ký hiệu $t.A_i$, $[t.A_i]$ và $[t]$ tương ứng biểu thị $\langle V_i, \alpha_i, \beta_i \rangle$, V_i và (V_1, \dots, V_k) . Nghĩa là $t.A_i = \langle V_i, \alpha_i, \beta_i \rangle$, $[t.A_i] = V_i$ và $[t] = (V_1, \dots, V_k)$. Đối với mỗi tập thuộc tính $X \subseteq \{A_1, A_2, \dots, A_k\}$, ký hiệu $t[X]$ được sử dụng để biểu thị phần còn lại của t sau khi đã loại bỏ các giá trị của các thuộc tính không thuộc X .

Từ định nghĩa 2.3.2, chúng tôi lưu ý rằng, mỗi thuộc tính A_i của bộ t trong quan hệ r trên $R(U, \wp)$ chỉ nhận một giá trị (tập) v trong V_i với một xác suất $p(v) \in [\alpha_i(v), \beta_i(v)]$. Ngoài ra, mỗi một giá trị chính xác $v \in V_i$ cũng được coi là một tập hợp (nghĩa là v và $\{v\}$ được coi như nhau). Vì vậy, mỗi quan hệ xác suất trong [23] có thể được coi như một quan hệ xác suất đặc biệt theo định nghĩa 3.5.2.

Ví dụ 3.5.2 Một quan hệ xác suất đơn giản **PATIENT** trên lược đồ **PATIENT** về các bệnh nhân tại một cơ sở khám chữa bệnh quốc tế (cơ sở khám chữa bệnh của nước ngoài ở Việt nam) có thể bao gồm các bộ như trong Bảng 3.5.1. Trong quan hệ này, các thuộc

tính P_ID , P_NAME , P_AGE , $P_DISEASE$ và D_COST mô tả về mã số, tên, tuổi, bệnh và chi phí điều trị hàng ngày của mỗi bệnh nhân. Thực tế, khi khám bệnh, các bác sĩ không phải luôn luôn xác định được chắc chắn bệnh của mỗi bệnh nhân. Tương tự như vậy, chi phí điều trị của mỗi bệnh nhân cũng không biết được một cách chính xác ngay cả khi các bệnh nhân biết được bệnh của họ. Ở đây, qui ước đơn vị chi phí điều trị là 1 (USD).

Lưu ý rằng, đối với mỗi thuộc tính $A \in U = \{P_ID, P_NAME, P_AGE, P_DISEASE, D_COST\}$ trong lược đồ **PATIENT**(U, \wp), $\wp(A)$ bao gồm tất cả các bộ ba xác suất trên $dom(A)$.

Bảng 3.5.1 Quan hệ PATIENT

P_ID	P_NAME	P_AGE	P_DISEASE	D_COST
PT226	$\langle \{ \text{Oliver} \}, u, u \rangle$	$\langle \{ 65 \}, u, u \rangle$	$\langle \{ \text{lung cancer, tuberculosis} \}, 0.6u, 1.2u \rangle$	$\langle \{ 30, 35 \}, 0.7u, 1.3u \rangle$
PT234	$\langle \{ \text{Blair} \}, u, u \rangle$	$\langle \{ 43, 44 \}, u, u \rangle$	$\langle \{ \{ \text{hepatitis, cirrhosis} \}, \{ \text{cholecystitis} \} \}, 0.9u, 1.3u \rangle$	$\langle \{ 6, 7 \}, 0.8u, 1.4u \rangle$
PT242	$\langle \{ \text{Alice} \}, u, u \rangle$	$\langle \{ 36 \}, u, u \rangle$	$\langle \{ \text{cholecystitis} \}, u, u \rangle$	$\langle \{ 8 \}, u, u \rangle$
PT267	$\langle \{ \text{Anne} \}, u, u \rangle$	$\langle \{ 15 \}, u, u \rangle$	$\langle \{ \{ \text{bronchitis, angina} \} \}, u, u \rangle$	$\langle \{ 7 \}, u, u \rangle$

Để đơn giản, mỗi bộ ba xác suất dạng $\langle V, u, u \rangle$, với $V = \{v\}$, sẽ được biểu diễn như một giá trị đơn v . Bởi vì nếu thuộc tính nhận một bộ ba xác suất như vậy, thì thực sự nó chỉ nhận một giá trị v với xác suất là 1 (Định nghĩa 3.3.2). Nói một cách khác, thuộc tính chắc chắn nhận giá trị v .

Cũng như trong mô hình CSDL quan hệ truyền thống, phụ thuộc hàm mô tả mối liên hệ giữa các thuộc tính và là một trong các khái niệm chính được sử dụng trong quá trình chuẩn hóa CSDL. Để định nghĩa phụ thuộc hàm xác suất trong URDB, trước hết, độ đo xác suất để xác định mức độ bằng nhau của hai giá trị của cùng một thuộc tính của hai bộ khác nhau trong một quan hệ được định nghĩa như dưới đây.

Định nghĩa 3.5.3 Giả sử t_1 và t_2 là hai bộ trong một quan hệ xác suất r , A là một thuộc tính của r và \otimes là một chiến lược hội xác suất. *Khoảng xác suất* cho các giá trị của thuộc tính A của hai bộ t_1 và t_2 bằng nhau theo \otimes , được biểu thị bởi $Prob(t_1.A =_{\otimes} t_2.A)$, là

$$[\sum_{v \in V} \alpha(v).Pr(v_1 = v_2), \min(1, \sum_{v \in V} \beta(v).Pr(v_1 = v_2))].$$

Trong đó, $t_1.A = \langle V_1, \alpha_1, \beta_1 \rangle$, $t_2.A = \langle V_2, \alpha_2, \beta_2 \rangle$ và $[\alpha(v), \beta(v)] = [\alpha_1(v_1), \beta_1(v_1)] \otimes [\alpha_2(v_2), \beta_2(v_2)]$, $\forall v = (v_1, v_2) \in V = V_1 \times V_2$.

Ví dụ 3.5.3 Giả sử t_1 và t_2 là hai bộ như sau:

$$t_1 = [A: \langle \{a, b\}, u, u \rangle, B: \langle \{c, d\}, u, u \rangle],$$

$$t_2 = [A: \langle \{a, c\}, 0.8u, 1.2u \rangle, B: \langle \{c, f\}, u, u \rangle].$$

Thì khoảng xác suất để giá trị của thuộc tính A của hai bộ t_1 và t_2 là bằng nhau theo chiến lược hội độc lập \otimes_{in} là $Prob(t_1.A =_{\otimes_{in}} t_2.A) = [0.2, 0.3]$.

Bây giờ, phụ thuộc hàm xác suất trong URDB ([28]) được mở rộng từ phụ thuộc hàm xác suất trong [23] như sau.

Định nghĩa 3.5.4 Giả $U = \{A_1, A_2, \dots, A_k\}$ là một tập k thuộc tính đôi một khác nhau, $R(U, \wp)$ là một lược đồ quan hệ xác suất, r là một quan hệ bất kỳ trên R , \otimes là một chiến lược hội xác suất, $X = \{A_i, \dots, A_l\}$ và $Y = \{A_j, \dots, A_m\}$ là hai tập con của U . Một *phụ thuộc hàm xác suất của Y đối với X theo \otimes trên R* , được ký hiệu bởi $X \rightsquigarrow_{\otimes} Y$, nếu và chỉ nếu:

$$\forall t_1, t_2 \in r, \mathfrak{I}(t_1[X] =_{\otimes} t_2[X]) \leq \mathfrak{I}(t_1[Y] =_{\otimes} t_2[Y]),$$

Trong đó, $\mathfrak{I}(t_1[X] =_{\otimes} t_2[X]) = Prob(t_1.A_i =_{\otimes} t_2.A_i) \otimes \dots \otimes Prob(t_1.A_l =_{\otimes} t_2.A_l)$ và $\mathfrak{I}(t_1[Y] =_{\otimes} t_2[Y]) = Prob(t_1.A_j =_{\otimes} t_2.A_j) \otimes \dots \otimes Prob(t_1.A_m =_{\otimes} t_2.A_m)$.

Một ví dụ hiển nhiên của phụ thuộc hàm xác suất là mọi thuộc tính A_i phụ thuộc tập $\{A_1, A_2, \dots, A_k\}$ bao gồm tất cả các thuộc tính của lược đồ quan hệ R . Chúng tôi lưu ý rằng, trong CSDL quan hệ cổ điển, vì xác suất cho hai giá trị bằng nhau là bằng 0 hoặc 1, do đó phụ thuộc hàm trong CSDL quan hệ cổ điển là một trường hợp riêng của phụ thuộc hàm xác suất theo định nghĩa này.

Như đối với CSDL quan hệ cổ điển, *khóa* (key) của một lược đồ quan hệ trong URDB là cơ sở để nhận dạng các bộ trong một quan hệ xác suất. Trong mô hình và các hệ quản trị CSDL quan hệ cổ điển, khóa được ràng buộc không nhận giá trị NULL ([1],

[2]). Tương tự, trong URDB, giá trị của mỗi thuộc tính khóa được giả sử là luôn luôn chắc chắn và xác định. Khái niệm khóa của lược đồ quan hệ xác suất được định nghĩa bằng cách sử dụng phụ thuộc hàm xác suất như sau.

Định nghĩa 3.5.5 Giả sử $R(U, \wp)$ là một lược đồ quan hệ xác suất, r là một quan hệ bất kỳ trên R và \otimes là một chiến lược hội xác suất, một tập thuộc tính $K \subseteq U$ được gọi là khóa của R theo \otimes nếu giá trị của mỗi thuộc tính của K là luôn luôn chắc chắn, chính xác trong r và có một phụ thuộc hàm xác suất $K \rightsquigarrow_{\otimes} U$ sao cho không tồn tại tập con thực sự nào của K có tính chất này.

Trong lược đồ và quan hệ xác suất PATIENT ở trên, nếu chúng ta giả sử mỗi bệnh nhân có một mã số duy nhất tương ứng với giá trị của thuộc tính P_ID và mã số này khác với tất cả các mã số của các bệnh nhân khác thì theo định nghĩa 3.5.5, P_ID là một khóa của lược đồ PATIENT theo mọi chiến lược hội xác suất.

Định nghĩa 3.5.6 Một cơ sở dữ liệu quan hệ xác suất (probabilistic relational database) trên một tập các thuộc tính là một tập các quan hệ xác suất tương ứng với một tập các lược đồ quan hệ xác suất của chúng.

Lưu ý rằng, nếu chỉ quan tâm đến một quan hệ duy nhất trên một lược đồ thì có thể đồng nhất ký hiệu tên quan hệ và lược đồ của chúng.

3.6. Kết luận

Trong Chương 4 này các khái niệm thuộc tính, kiểu, giá trị, lược đồ và quan hệ trong mô hình URDB đã được định nghĩa. Các khái niệm này là sự mở rộng các khái niệm tương ứng trong CSDL quan hệ với thuộc tính có giá trị không chắc chắn được biểu diễn bởi các bộ ba xác suất. Bây giờ, thông tin của các đối tượng trong URDB có thể không chắc chắn, không đầy đủ và được biểu diễn bởi các giá trị bộ xác suất. Tập các bộ (có giá trị xác suất) định nghĩa một quan hệ xác suất. Tập các quan hệ xác suất tạo nên một CSDL quan hệ xác suất. Mô hình dữ liệu của URDB đã được xây dựng. Trong Chương 5 tiếp theo, các phép toán đại số trên URDB sẽ được phát triển làm cơ sở cho các truy vấn thông tin không chắc chắn và không đầy đủ về thuộc tính của các quan hệ (các đối tượng).

Chương 4

CÁC PHÉP TOÁN ĐẠI SỐ TRÊN URDB

4.1. Giới thiệu

Các phép toán đại số quan hệ trên URDB như chọn, chiếu, tích Descartes, kết, giao, hợp và trừ là cơ sở để xử lý và thực thi các truy vấn trên các quan hệ trong URDB. Các phép toán đại số trên URDB được xây dựng bằng cách mở rộng tương ứng các phép toán trên CSDL quan hệ truyền thống với sự tích hợp giá trị không chắc chắn cho các thuộc tính quan hệ. Với cách tiếp cận này, trong các Phần 4.2 đến 4.7, các phép toán trên CSDL quan hệ truyền thống được mở rộng thành các phép toán trên URDB sao cho nhất quán với mô hình dữ liệu URDB. Đối số và kết quả của các phép toán đại số trên URDB là các quan hệ URDB. Tính toán xác suất của các giá trị thuộc tính quan hệ được thực hiện bởi các chiến lược kết hợp xác suất. Các phép toán đại số trên URDB không chỉ thao tác trên các giá trị không chắc chắn của các quan hệ trong URDB mà còn cả với các giá trị chắc chắn của quan hệ truyền thống.

Phần 4.8 giới thiệu các tính chất của các phép toán đại số trên URDB như là sự mở rộng các tính chất của các phép toán đại số quan hệ truyền thống (Phần 2.4). Các tính chất này được chứng minh đầy đủ cho thấy các phép toán đại số trong URDB được xây dựng là đúng đắn. Tập các phép toán trên URDB là cơ sở cho một ngôn ngữ truy vấn thông tin không chắc chắn trên các CSDL quan hệ xác suất. Cuối cùng Phần 4.9 là một vài lưu ý và kết luận của chương này.

4.2. Phép chọn

Cũng như trong CSDL quan hệ truyền thống ([1], [2], [3]), phép chọn là một phép toán đại số cơ bản trên URDB. Nói một cách không hình thức, kết quả của một truy vấn

chọn trên một quan hệ r của một lược đồ R là một quan hệ r' trên R sao cho các bộ của r' có các giá trị thuộc tính thỏa mãn điều kiện chọn của truy vấn này.

Trước khi định nghĩa phép chọn, chúng tôi giới thiệu cú pháp và ngữ nghĩa hình thức của các *điều kiện chọn* (selection condition) như là sự mở rộng các định nghĩa tương ứng trong CSDL quan hệ truyền thống với các khoảng xác suất cần thỏa của điều kiện chọn. Chúng tôi bắt đầu với cú pháp của các *biểu thức chọn* (selection expression) như sau.

Định nghĩa 4.2.1 Giả sử R là một lược đồ URDB và X là một tập các biến bộ quan hệ. Các *biểu thức chọn* được định nghĩa một cách đệ quy và có một trong các dạng sau:

1. $x.A \theta c$, trong đó $x \in X$, A là một thuộc tính trong R , θ là một quan hệ hai ngôi thuộc $\{=, \neq, \leq, \geq, <, >, \subseteq, \supseteq\}$ và c là một giá trị.
2. $x.A_1 =_{\otimes} x.A_2$, trong đó $x \in X$, A_1 và A_2 là hai thuộc tính phân biệt trong R và \otimes là một chiến lược hội xác suất kết hợp các xác suất để $x.A_1 = v_1$ và $x.A_2 = v_2$ sao cho $v_1 = v_2$.
3. $E_1 \otimes E_2$, trong đó E_1 và E_2 là các biểu thức chọn trên cùng một biến bộ quan hệ, \otimes là một chiến lược hội xác suất.
4. $E_1 \oplus E_2$, trong đó E_1 và E_2 là các biểu thức chọn trên cùng một biến bộ quan hệ, \oplus là một chiến lược tuyển xác suất.

Hai dạng đầu của biểu thức chọn được gọi là các *biểu thức chọn cơ sở* (atomic selection expression). Các chiến lược hội và tuyển xác suất đã được giới thiệu trong Phần 2.2 của Chương 2.

Ví dụ 4.2.1 Với lược đồ quan hệ **PATIENT** trong Ví dụ 3.5.2, một số biểu thức chọn có thể như sau (x là biến bộ):

1. Tìm tất cả bệnh nhân bị bệnh viêm gan (hepatitis). Yêu cầu này có thể được biểu diễn bởi biểu thức chọn cơ sở $x.P_DISEASE = hepatitis$.
2. Tìm tất cả bệnh nhân bị bệnh viêm gan và chi phí điều trị không ít hơn 7 USD. Yêu cầu này có thể được biểu diễn bởi biểu thức chọn $x.P_DISEASE = hepatitis \otimes x.D_COST \geq 7$.

Trong URDB, mỗi điều kiện chọn là sự kết hợp logic của các biểu thức chọn cùng với các khoảng xác suất cần được thỏa mãn như định nghĩa sau.

Định nghĩa 4.2.2 Giả sử R là một lược đồ quan hệ URDB, các *điều kiện chọn* (selection condition) được định nghĩa một cách đệ quy như sau:

1. Nếu E là một biểu thức chọn, L và U là các số thực trong khoảng $[0, 1]$, $L \leq U$ thì $(E)[L, U]$ là một điều kiện chọn.
2. Nếu ϕ và Ψ là các điều kiện chọn trên cùng một biến đối tượng thì $\neg\phi$, $(\phi \wedge \Psi)$ và $\phi \vee \Psi$ cũng là các điều kiện chọn.

Ví dụ 4.2.2 Với lược đồ quan hệ **PATIENT** trong CSDL các bệnh nhân ở Ví dụ 3.5.2, một số điều kiện chọn có thể như sau (x là biến bộ):

1. Tìm tất cả bệnh nhân tuổi ít hơn 30 với một xác suất ít nhất là 0.4 và có bệnh ung thư phổi với một xác suất ít nhất 0.8. Yêu cầu này có thể được thực hiện bởi điều kiện chọn $(x.P_AGE < 30)[0.4, 1.0] \wedge (x.P_DISEASE = lung\ cancer)[0.8, 1.0]$.
2. Tìm tất cả bệnh nhân bị bệnh viêm gan và chi phí điều trị hàng ngày không ít hơn 7 USD với xác suất từ 0.4 đến 0.6. Yêu cầu này có thể được thực hiện bởi điều kiện chọn $(x.DISEASE = hepatitis \otimes x.D_COST \geq 7)[0.4, 0.6]$.

Đối với URDB, mỗi biểu thức chọn trong một truy vấn là một sự kiện có xác suất. Xác suất của các biểu thức chọn, được kết hợp với các thuộc tính quan hệ, có thể được đo bởi diễn dịch của nó và được định nghĩa như sau.

Định nghĩa 4.2.3 Giả sử R là một lược đồ quan hệ URDB, r là một quan hệ trên R , x là một biến bộ quan hệ và t là một bộ trong r . *Diễn dịch xác suất* (probabilistic interpretation) của các biểu thức chọn theo R , r và t , được biểu thị bởi $prob_{R,r,t}$ là một ánh xạ bộ phận từ tập tất cả các biểu thức chọn đến tập tất cả các khoảng con đóng của khoảng $[0, 1]$ và được định nghĩa đệ quy như sau:

1. $prob_{R,r,t}(x.A \theta c) = [\sum_{v \in V} \alpha(v).Pr(v \theta c), \min(1, \sum_{v \in V} \beta(v).Pr(v \theta c))]$, trong đó $t.A = \langle V, \alpha, \beta \rangle$.

2. $prob_{R,r,t}(x.A_1 =_{\otimes} x.A_2) = [\sum_{v \in V} \alpha(v).Pr(v_1=v_2), \min(1, \sum_{v \in V} \beta(v).Pr(v_1=v_2))]$, trong đó $t.A_1 = \langle V_1, \alpha_1, \beta_1 \rangle$, $t.A_2 = \langle V_2, \alpha_2, \beta_2 \rangle$ và $[\alpha(v), \beta(v)] = [\alpha_1(v_1), \beta_1(v_1)] \otimes [\alpha_2(v_2), \beta_2(v_2)]$, $\forall v = (v_1, v_2) \in V = V_1 \times V_2$.
3. $prob_{R,r,t}(E_1 \otimes E_2) = prob_{R,r,t}(E_1) \otimes prob_{R,r,t}(E_2)$.
4. $prob_{R,r,t}(E_1 \oplus E_2) = prob_{R,r,t}(E_1) \oplus prob_{R,r,t}(E_2)$.

Một cách trực giác, $prob_{R,r,t}(x.A \theta c)$ là khoảng xác suất để thuộc tính A của bộ t có giá trị v sao cho $v \theta c$ và $prob_{R,r,t}(x.A_1 =_{\otimes} x.A_2)$ là khoảng xác suất để các thuộc tính A_1 và A_2 của bộ t tương ứng có giá trị v_1 và v_2 sao cho $v_1 = v_2$.

Ví dụ 4.2.3 Ký hiệu r là quan hệ PATIENT trong ví dụ 3.5.2 và R là lược đồ của PATIENT, xem bộ t_2 (bộ thứ hai) trong r , chúng ta có

$$prob_{R,r,t_2}(x.P_DISEASE = cholecystitis) =$$

$$\begin{aligned} & [0.9u(\{\text{hepatitis, cirrhosis}\}).Pr(\{\text{hepatitis, cirrhosis}\} = \text{cholecystitis}) + \\ & 0.9u(\{\text{cholecystitis}\}).Pr(\{\text{cholecystitis}\} = \{\text{cholecystitis}\}), \\ & \min(1.3u(\{\text{hepatitis, cirrhosis}\}).Pr(\{\text{hepatitis, cirrhosis}\} = \text{cholecystitis}) \\ & + 1.3u(\{\text{cholecystitis}\}).Pr(\{\text{cholecystitis}\} = \{\text{cholecystitis}\}))] \\ & = [0.9 \times 0.5 \times 0.0 + 0.9 \times 0.5 \times 0.0, \min(1, 1.3 \times 0.5 \times 0.0 + 1.3 \times 0.5 \times 1.0)] \\ & = [0.45, 0.65] . \end{aligned}$$

Trong URDB, mỗi điều kiện chọn là sự kết hợp logic của các biểu thức chọn cùng với các khoảng xác suất cần được thỏa mãn. Nói cách khác, sự thỏa mãn về logic đối với một điều kiện chọn là thỏa mãn các cận xác suất được kết hợp với các biểu thức chọn trong điều kiện chọn này. Định nghĩa về sự thỏa mãn các điều kiện chọn trong URDB như sau.

Định nghĩa 4.2.4 Giả sử R là một lược đồ quan hệ URDB, r là một quan hệ trên R và $t \in r$. Sự thỏa mãn (satisfaction) các điều kiện chọn của t theo diễn dịch xác suất $prob_{R,r,t}$ được định nghĩa như sau:

1. $prob_{R,r,t} \models (E)[L, U]$ nếu và chỉ nếu $prob_{R,r,t}(E) \subseteq [L, U]$.
2. $prob_{R,r,t} \models \neg\phi$ nếu và chỉ nếu $prob_{R,r,t} \not\models \phi$ không thỏa.

3. $prob_{R,r,t} \models \phi \wedge \psi$ nếu và chỉ nếu $prob_{R,r,t} \models \phi$ và $prob_{R,r,t} \models \psi$ thỏa.
4. $prob_{R,r,t} \models \phi \vee \psi$ nếu và chỉ nếu $prob_{R,r,t} \models \phi$ thỏa hoặc $prob_{R,r,t} \models \psi$ thỏa.

Lưu ý rằng, trong CSDL truyền thống, khái niệm biểu thức chọn và điều kiện chọn là một và có thể xem các khoảng xác suất $[L, U]$ trong các điều kiện chọn luôn luôn bằng $[1.0, 1.0]$. Điều này cũng có nghĩa là khái niệm sự thỏa mãn các điều kiện chọn trong mô hình CSDL truyền thống là trường hợp riêng của khái niệm sự thỏa mãn các điều kiện chọn trong URDB.

Ví dụ 4.2.4 Ký hiệu r là quan hệ PATIENT trong ví dụ 3.5.2 và R là lược đồ của PATIENT, từ Ví dụ 4.2.3 ta có $prob_{R,r,t2} \models (x.P_DISEASE = cholecystitis)[0.4, 1.0]$ và $prob_{R,r,t2} \not\models (x.P_DISEASE = cholecystitis)[0.5, 1.0]$, vì $prob_{R,r,t2}(x.P_DISEASE = cholecystitis) = [0.45, 0.65] \subseteq [0.4, 1.0]$ và $prob_{R,r,t2}(x.P_DISEASE = cholecystitis) = [0.45, 0.65] \not\subseteq [0.5, 1.0]$

Bây giờ, trên cơ sở các khái niệm đã được giới thiệu, phép chọn các quan hệ trong URDB được định nghĩa như sau.

Định nghĩa 4.2.5 Giả sử R là một lược đồ quan hệ URDB, r là một quan hệ trên R và ϕ là một điều kiện chọn trên biến bộ x . *Phép chọn* trên r theo ϕ , được ký hiệu $\sigma_\phi(r)$, là một quan hệ r^* trên R , bao gồm tất cả các bộ thỏa mãn điều kiện chọn ϕ .

$$r^* = \{t \in r \mid prob_{R,r,t} \models \phi\}.$$

Ví dụ 4.2.5 Xét quan hệ $r = \text{PATIENT}$ trong Ví dụ 3.5.2. Truy vấn “Tìm tất cả các bệnh nhân hơn 40 tuổi với một xác suất ít nhất là 0.9, có cả hai bệnh viêm gan và xơ gan và trả một chi phí điều trị hàng ngày không ít hơn 6 USD với một xác suất giữa 0.3 và 0.7” có thể được thực hiện bởi phép chọn $\sigma_\phi(\text{PATIENT})$, trong đó $\phi = (x.AGE > 40)[0.9, 1] \wedge (x.DISEASE \supseteq \{\text{hepatitis, cirrhosis}\} \otimes_{in} x.D_COST \geq 6)[0.3, 0.7]$.

Chỉ có bộ thứ hai t_2 của quan hệ PATIENT trong ví dụ 3.5.2 thỏa mãn ϕ , bởi vì:

$$prob_{R,r,t2}(x.AGE > 40) = [u(43) \times Pr(43 > 40) + u(44) \times Pr(44 > 40), \min(1, u(43) \times Pr(43 > 40) + u(44) \times Pr(44 > 40))] = [0.5 \times 1 + 0.5 \times 1, \min(1, 0.5 \times 1 + 0.5 \times 1)] = [1, 1] \subseteq [0.9, 1],$$

$$prob_{R,r,t2}(x.D_COST \geq 6)$$

$$= [0.8u \times Pr(6 \geq 6) + 0.8u \times Pr(7 \geq 6), \min(1, 1.4u \times Pr(6 \geq 6) + 1.4u \times Pr(7 \geq 6))]$$

$$= [0.8 \times 0.5 \times 1 + 0.8 \times 0.5 \times 1, \min(1, 1.4 \times 0.5 \times 1 + 1.4 \times 0.5 \times 1)] = [0.8, 1].$$

$$prob_{R,r,t_2}(x.DISEASE \supseteq \{\text{hepatitis, cirrhosis}\})$$

$$= [0.9u(\{\text{hepatitis, cirrhosis}\}).Pr(\{\text{hepatitis, cirrhosis}\} \supseteq \{\text{hepatitis, cirrhosis}\}) + 0.9u(\{\text{cholecystitis}\}).Pr(\{\text{cholecystitis}\} \supseteq \{\text{hepatitis, cirrhosis}\}),$$

$$\min(1, 1.3u(\{\text{hepatitis, cirrhosis}\}).Pr(\{\text{hepatitis, cirrhosis}\} \supseteq \{\text{hepatitis, cirrhosis}\}) + 1.3u(\{\text{cholecystitis}\}).Pr(\{\text{cholecystitis}\} \supseteq \{\text{hepatitis, cirrhosis}\}))]$$

$$= [0.9 \times 0.5 \times 1.0 + 0.9 \times 0.5 \times 0.0, \min(1, 1.3 \times 0.5 \times 1.0 + 1.3 \times 0.5 \times 0.0)]$$

$$= [0.45, 0.65].$$

$$prob_{R,r,t_2}(x.DISEASE \supseteq \{\text{hepatitis, cirrhosis}\} \otimes_{in} x.D_COST \geq 6) = [0.45, 0.65] \otimes_{in} [0.8, 1] = [0.36, 0.65] \subseteq [0.3, 0.7].$$

Với các bộ khác, chúng ta có $prob_{R,r,t_i}(x.DISEASE \supseteq \{\text{hepatitis, cirrhosis}\} \otimes_{in} x.D_COST \geq 6) = [0, 0] \not\subseteq [0.3, 0.7], \forall i \neq 4$. Vì vậy, kết quả của truy vấn chọn là quan hệ như trong Bảng 4.2.1.

Bảng 4.2.1 Quan hệ $\sigma_{\emptyset}(PATIENT)$

P_ID	P_NAME	P_AGE	P_DISEASE	D_COST
PT234	$\langle \{\text{Blair}\}, u, u \rangle$	$\langle \{43, 44\}, u, u \rangle$	$\langle \{\{\text{hepatitis, cirrhosis}\}, \{\text{cholecystitis}\}\}, 0.9u, 1.3u \rangle$	$\langle \{6, 7\}, 0.8u, 1.4u \rangle$

4.3. Phép chiếu

Cũng như trong các cơ sở dữ liệu truyền thống, ý nghĩa của *phép chiếu* (projection) các quan hệ URDB trên một tập thuộc tính là phục vụ cho việc trích một phần thông tin của các đối tượng theo nhu cầu áp dụng. Phép chiếu các quan hệ xác suất như là một mở rộng của phép chiếu quan hệ trong CSDL quan hệ truyền thống trong đó các bộ có cùng giá trị thuộc tính phải được kết hợp thành một bộ trong quan hệ kết quả bằng một chiến lược kết hợp xác suất như dưới đây.

Định nghĩa 4.3.1 Giả sử $R = (\mathbf{U}, \wp)$ là một lược đồ quan hệ xác suất, r là quan hệ trên R và \mathbf{L} là tập con của tập thuộc tính \mathbf{U} , \oplus là một chiến lược tuyển xác suất. *Phép chiếu*

của r trên L theo \oplus , ký hiệu là $\Pi_{L\oplus}(r)$, là quan hệ xác suất r^* trên lược đồ R^* được xác định bởi:

1. $R^* = (L, \wp^*)$ và $\wp^*(A) = \wp(A), \forall A \in L$.
2. $r^* = \{t^* \mid t^*.A = u.A \oplus \dots \oplus w.A, \forall A \in L, t \in r \text{ sao cho } \exists u, \dots, w \in r \text{ và } [u[L]] = \dots = [w[L]] = [t[L]]\}$.

Ví dụ 4.3.1 Kết quả phép chiếu quan hệ **PATIENT** trong Ví dụ 3.5.2 trên tập thuộc tính $L = \{P_NAME, P_AGE, P_DISEASE\}$ theo \oplus_{in} được tính như trong bảng 4.3.1.

Bảng 4.3.1 Quan hệ $\Pi_{L\oplus in}(PATIENT)$

P_NAME	P_AGE	P_DISEASE
$\langle \{\text{Oliver}\}, u, u \rangle$	$\langle \{65\}, u, u \rangle$	$\langle \{\text{lung cancer, tuberculosis}\}, 0.6u, 1.2u \rangle$
$\langle \{\text{Blair}\}, u, u \rangle$	$\langle \{43, 44\}, u, u \rangle$	$\langle \{\{\text{hepatitis, cirrhosis}\}, \{\text{cholecystitis}\}\}, 0.9u, 1.3u \rangle$
$\langle \{\text{Alice}\}, u, u \rangle$	$\langle \{36\}, u, u \rangle$	$\langle \{\text{cholecystitis}\}, u, u \rangle$
$\langle \{\text{Anne}\}, u, u \rangle$	$\langle \{15\}, u, u \rangle$	$\langle \{\{\text{bronchitis, angina}\}\}, u, u \rangle$

4.4. Phép tích Descartes

Tích Descartes (Descartes product) của hai quan hệ trong URDB cũng tương tự như tích Descartes của hai quan hệ trong CSDL quan hệ truyền thống. Nghĩa là, tích của hai quan hệ đạt được bằng cách nối danh sách thuộc tính và giá trị của một bộ bất kỳ trong quan hệ thứ nhất với danh sách thuộc tính của một bộ bất kỳ trong quan hệ thứ hai. Ý nghĩa thực tế của phép toán là nhằm xem xét thông tin về tất cả các cặp đối tượng (quan hệ) tương ứng trong hai quan hệ của CSDL. Cũng như trong CSDL quan hệ truyền thống, tích Descartes trong URDB chỉ được định nghĩa trên hai quan hệ có tập thuộc tính rời nhau. Phép tích Descartes của hai quan hệ URDB được mở rộng từ tích Descartes của hai quan hệ truyền thống với giá trị tập hợp có xác suất của các thuộc tính các bộ như sau.

Định nghĩa 4.4.1 Giả sử U_1, U_2 là hai tập hợp các thuộc tính không có phần tử chung nào, $R_1 = (U_1, \wp_1), R_2 = (U_2, \wp_2)$ là hai lược đồ URDB, r_1, r_2 lần lượt là hai quan hệ

trên R_1 và R_2 . *Phép tích Descartes* của r_1 và r_2 , ký hiệu là $r_1 \times r_2$, là quan hệ xác suất r trên R , được xác định bởi:

1. $R = (U, \wp)$, trong đó $U = U_1 \cup U_2$, $\wp(A) = \wp_1(A)$ nếu $A \in U_1$ và $\wp(A) = \wp_2(A)$ nếu $A \in U_2$.
2. $r = \{t \mid t.A = t_1.A \text{ nếu } A \in U_1, t.A = t_2.A \text{ nếu } A \in U_2, t_1 \in r_1, t_2 \in r_2\}$.

4.5. Phép kết tự nhiên

Phép kết tự nhiên hai quan hệ xác suất trong URDB được định nghĩa bằng cách kết nối các bộ tương ứng của hai quan hệ, đồng thời hợp nhất các giá trị và thuộc tính cùng tên của chúng trong quan hệ kết quả bằng một chiến lược hội các bộ ba xác suất (là giá trị các thuộc tính cùng tên). Đó là một sự mở rộng phép kết trong CSDL quan hệ truyền thống (Định nghĩa 1.3.4).

Ý nghĩa thực tế của phép kết trong URDB là tìm các cặp đối tượng (bộ) mà thông tin về chúng có một đặc tính chung nào đó. Chẳng hạn, “tìm các cặp gói bưu kiện có cùng thời gian vận chuyển”, hay “tìm các cặp bệnh nhân có cùng quê và cùng loại bệnh” v.v., trong đó thông tin về thời gian hay loại bệnh có thể không chắc chắn và không đầy đủ. Phép kết của hai quan hệ URDB được mở rộng từ phép kết tự nhiên của hai quan hệ truyền thống với xác suất và các giá trị tập hợp như định nghĩa sau.

Định nghĩa 4.5.1 Giả sử U_1 và U_2 là hai tập thuộc tính sao cho nếu chúng có thuộc tính cùng tên tương ứng trong hai tập đó thì các thuộc tính như vậy có cùng miền giá trị. Giả sử $R_1 = (U_1, \wp_1)$ và $R_2 = (U_2, \wp_2)$ là hai lược đồ URDB, r_1, r_2 lần lượt là hai quan hệ trên R_1 và R_2 , và \otimes là một chiến lược hội xác suất. *Phép kết tự nhiên* của r_1 và r_2 theo \otimes , ký hiệu là $r_1 \bowtie_{\otimes} r_2$, là quan hệ xác suất r trên lược đồ R , được xác định bởi:

1. $R = (U, \wp)$ trong đó $U = U_1 \cup U_2$, $\wp(A) = \wp_1(A)$ nếu $A \in U_1 - U_2$, $\wp(A) = \wp_2(A)$ nếu $A \in U_2 - U_1$ và $\wp(A) = \wp_1(A) = \wp_2(A)$ nếu $A \in U_1 \cap U_2$.
2. $r = \{t \mid t.A = t_1.A \text{ nếu } A \in U_1 - U_2, t.A = t_2.A \text{ nếu } A \in U_2 - U_1, t.A = t_1.A \otimes t_2.A \text{ nếu } A \in U_1 \cap U_2 \text{ và } t_1.A \otimes t_2.A \neq \langle \emptyset, \alpha, \beta \rangle, t_1 \in r_1, t_2 \in r_2\}$.

Ví dụ 4.5.1 Giả sử hai quan hệ **PATIENT**₁ và **PATIENT**₂ được cho như trong các Bảng 4.5.1 và 4.5.2, thì kết quả phép kết của chúng theo chiến lược hội độc lập là quan hệ **PATIENT** được tính như bảng 4.5.3.

Bảng 4.5.1 Quan hệ PATIENT₁

P_ID	P_DISEASE
PT0421	$\langle \{ \text{bronchitis} \}, u, u \rangle$
PT3829	$\langle \{ \text{cholecystitis}, \text{gall-stone} \}, 0.8u, 1.2u \rangle$

Bảng 4.5.2 Quan hệ PATIENT₂

P_NAME	P_DISEASE
$\langle \{ \text{Peter} \}, u, u \rangle$	$\langle \{ \text{bronchitis} \}, u, u \rangle$
$\langle \{ \text{George} \}, u, u \rangle$	$\langle \{ \text{cholecystitis}, \text{cirrhosis} \}, 0.8u, 1.4u \rangle$

Bảng 4.5.3 Quan hệ PATIENT₁ \bowtie_{in} PATIENT₂

P_ID	P_NAME	P_DISEASE
PT0421	$\langle \{ \text{Peter} \}, u, u \rangle$	$\langle \{ \text{bronchitis} \}, u, u \rangle$
PT3829	$\langle \{ \text{George} \}, u, u \rangle$	$\langle \{ \text{cholecystitis} \}, 0.16u, 0.42u \rangle$

4.6. Phép giao, hợp và trừ

Một cách không hình thức, các phép toán *giao* (intersection), *hợp* (union) và *trừ* (difference) của hai quan hệ URDB trên cùng một lược đồ là sự mở rộng các phép toán tương ứng trong cơ sở dữ liệu quan hệ truyền thống. Cụ thể, giao của hai quan hệ URDB là tập các bộ chung của hai quan hệ đó. Tuy nhiên, do giá trị của mỗi thuộc tính bộ trên hai quan hệ là có xác suất nên giá trị thuộc tính của các bộ chung phải được kết hợp theo một chiến lược hội xác suất. Tương tự, đối với phép hợp và trừ, giá trị thuộc tính của các bộ chung cũng cần được kết hợp bởi một chiến lược tuyền và hiệu xác suất tương ứng. Lưu ý là trong cơ sở dữ liệu quan hệ truyền thống giá trị của mỗi thuộc tính trong hai quan hệ trên cùng một lược đồ là chắc chắn, nghĩa là có xác suất bằng 1, nên phép kết hợp xác suất của các bộ chung là tầm thường. Nói một cách khác các phép giao, hợp và trừ trên URDB là mở rộng chính các phép toán đó trên mô hình cơ sở dữ

liệu quan hệ truyền thống ([1], [3]). Phép giao, hợp và trừ của hai quan hệ URDB lần lượt được định nghĩa như dưới đây.

Định nghĩa 4.6.1 Giả sử $R = (U, \wp)$ là một lược đồ URDB, r_1 và r_2 là hai quan hệ trên R , \otimes là một chiến lược hội xác suất. *Phép giao* của r_1 và r_2 theo \otimes , ký hiệu là $r_1 \cap_{\otimes} r_2$, là quan hệ xác suất trên R được xác định bởi $r = \{t \mid t.A = t_1.A \otimes t_2.A, t_1 \in r_1, t_2 \in r_2, \text{ sao cho } [t_1] = [t_2] \text{ và } t_1.A \otimes t_2.A \neq \langle V, 0, 0 \rangle, A \in U\}$.

Chúng tôi lưu ý rằng, giao của hai quan hệ trong CSDL quan hệ truyền thống là tập các bộ chung của hai quan hệ đó. Các bộ chung của hai quan hệ trên cùng một lược đồ là các bộ cùng giá trị thuộc tính tương ứng thuộc về hai quan hệ đó. Khái niệm “các bộ chung” trong URDB cũng tương tự như trong CSDL truyền thống, tuy nhiên giá trị thuộc tính được mở rộng với tập hợp có xác suất. Như vậy, định nghĩa giao của hai quan hệ xác suất là mở rộng giao của hai quan hệ truyền thống.

Ví dụ 4.6.1 Giả sử hai quan hệ DIAGNOSE_1 và DIAGNOSE_2 trên cùng một lược đồ $\text{DIAGNOSE}(\text{P_ID}, \text{D_ID}, \text{P_DISEASE}, \text{D_COST})$ được cho như trong các Bảng 4.6.1 và 4.6.2, thì giao của chúng theo chiến lược hội xác suất độc lập \otimes_{in} là quan hệ $\text{DIAGNOSE} = \text{DIAGNOSE}_1 \cap_{\otimes_{in}} \text{DIAGNOSE}_2$ được tính toán như trong Bảng 4.6.3.

Bảng 4.6.1 Relation DIAGNOSE_1

P_ID	D_ID	P_DISEASE	D_COST
PT0421	D101	$\langle \{\text{lung cancer, tuberculosis}\}, 0.8u, 1.2u \rangle$	$\langle \{30, 35\}, u, u \rangle$
PT3829	D102	$\langle \{\text{hepatitis, gall-stone}\}, u, u \rangle$	$\langle \{6, 7\}, u, u \rangle$

Bảng 4.6.2 Relation DIAGNOSE_2

P_ID	D_ID	P_DISEASE	D_COST
PT3830	D101	$\langle \{\text{lung cancers}\}, u, u \rangle$	$\langle \{35, 40\}, u, u \rangle$
PT3829	D102	$\langle \{\text{hepatitis, gall-stone}\}, 0.8u, 1.2u \rangle$	$\langle \{6, 7\}, u, u \rangle$
PT2938	D025	$\langle \{\text{hepatitis}\}, u, u \rangle$	$\langle \{6\}, u, u \rangle$

Bảng 4.6.3 $\text{DIAGNOSE}_1 \cap_{\otimes_{in}} \text{DIAGNOSE}_2$

P_ID	D_ID	P_DISEASE	D_COST
PT3829	D102	$\langle \{\text{hepatitis, gall-stone}\}, 0.4u, 0.6u \rangle$	$\langle \{6, 7\}, 0.5u, 0.5u \rangle$

Tiếp theo, phép hợp của hai quan hệ URDB trên cùng một lược đồ sẽ được định nghĩa như sau.

Định nghĩa 4.6.2 Giả sử $R = (U, \wp)$ là một lược đồ URDB, r_1 và r_2 là hai quan hệ trên R , \oplus là một chiến lược tuyến xác suất. *Phép hợp* của r_1 và r_2 theo \oplus , ký hiệu là $r_1 \cup_{\oplus} r_2$, là quan hệ xác suất r trên R được xác định bởi $r = \{t_1 \in r_1 \mid \text{không có bộ } t_2 \in r_2 \text{ sao cho } [t_1] = [t_2]\} \cup \{t_2 \in r_2 \mid \text{không có bộ } t_1 \in r_1 \text{ sao cho } [t_2] = [t_1]\} \cup \{t \mid t.A = t_1.A \oplus t_2.A, t_1 \in r_1, t_2 \in r_2, \forall A \in U \text{ sao cho } [t_1] = [t_2]\}$.

Ví dụ 4.6.2 Xét hai quan hệ DIAGNOSE_1 và DIAGNOSE_2 trên cùng lược đồ $\text{DIAGNOSE}(U, \wp)$ như đã cho trong các Bảng 4.6.1 và 4.6.2 của Ví dụ 4.6.1. Khi đó, hợp của DIAGNOSE_1 và DIAGNOSE_2 theo chiến lược tuyến xác suất độc lập \oplus_{in} là quan hệ $\text{DIAGNOSE} \cup_{\oplus_{in}} \text{DIAGNOSE}_2$ được tính như trong Bảng 4.6.4.

Bảng 4.6.4 $\text{DIAGNOSE}_1 \cup_{\oplus_{in}} \text{DIAGNOSE}_2$

P_ID	D_ID	P_DISEASE	D_COST
PT0421	D101	$\langle \{\text{lung cancer, tuberculosis}\}, 0.8u, 1.2u \rangle$	$\langle \{30, 35\}, u, u \rangle$
PT3829	D102	$\langle \{\text{hepatitis, gall-stone}\}, 1.4u, 1.6u \rangle$	$\langle \{6, 7\}, 1.5u, 1.5u \rangle$
PT3830	D101	$\langle \{\text{lung cancers}\}, u, u \rangle$	$\langle \{35, 40\}, u, u \rangle$
PT2938	D025	$\langle \{\text{hepatitis}\}, u, u \rangle$	$\langle \{6\}, u, u \rangle$

Cuối cùng, phép trừ của hai quan hệ xác suất trên cùng một lược đồ URDB được định nghĩa như sau.

Định nghĩa 4.6.3 Giả sử $R = (U, \wp)$ là lược đồ URDB, r_1 và r_2 là hai quan hệ trên R , \ominus là một chiến lược hiệu xác suất. *Phép trừ* của r_1 và r_2 theo \ominus , ký hiệu là $r_1 -_{\ominus} r_2$, là quan hệ xác suất r trên R được xác định bởi $r = \{t_1 \in r_1 \mid \text{không có bất kỳ bộ } t_2 \in r_2 \text{ sao cho } [t_1] = [t_2]\} \cup \{t \mid t.A = t_1.A \ominus t_2.A, t_1 \in r_1, t_2 \in r_2 \text{ sao cho } [t_1] = [t_2] \text{ và } t_1.A \ominus t_2.A \neq \langle V, 0, 0 \rangle, A \in U\}$.

Ví dụ 4.6.3 Xét hai quan hệ DIAGNOSE_1 và DIAGNOSE_2 trên cùng lược đồ $\text{DIAGNOSE}(U, \wp)$ như đã cho trong các Bảng 4.6.1 và 4.6.2 của Ví dụ 4.6.1. Khi đó,

hiệu của $DIAGNOSE_1$ và $DIAGNOSE_2$ theo chiến lược hiệu xác suất bỏ qua \ominus_{ig} là quan hệ $DIAGNOSE - \ominus_{ig} DIAGNOSE_2$ được tính như trong Bảng 4.6.5.

Bảng 4.6.5 $DIAGNOSE_1 - \ominus_{ig} DIAGNOSE_2$

P_ID	D_ID	P_DISEASE	D_COST
PT0421	D101	$\langle \{\text{lung cancer, tuberculosis}\}, 0.8u, 1.2u \rangle$	$\langle \{30, 35\}, u, u \rangle$

4.7. Tính chất của các phép toán đại số

Các phép toán đại số trong URDB không chỉ được định nghĩa dựa trên các cơ sở toán học vững chắc mà còn nhất quán với mô hình dữ liệu URDB. Thông qua các ví dụ cụ thể, các phép toán đại số URDB đã chứng tỏ khả năng thao tác, tính toán, truy vấn dữ liệu không chắc chắn, được biểu diễn bởi mô hình URDB. Tương tự như các phép toán đại số trên mô hình CSDL quan hệ truyền thống, các phép toán trên URDB cũng có những tính chất của chúng. Như sẽ thấy dưới đây, mặc dù được mở rộng xác suất từ mô hình CSDL quan hệ truyền thống, nhưng các phép toán trên URDB vẫn duy trì được các tính chất của các phép toán trên CSDL quan hệ truyền thống (Phần 1.4). Điều này cho thấy URDB là một mở rộng hợp logic, có tính kế thừa và phát triển của CSDL quan hệ truyền thống.

Tương tự như trong CSDL quan hệ truyền thống, phép chọn không phụ thuộc vào thứ tự thực hiện đối với các điều kiện chọn, phép tích Descartes, phép kết, phép giao và phép hợp có tính giao hoán và kết hợp. Hay phép chiếu không phụ thuộc vào số các tập thuộc tính được chiếu theo thứ tự bao hàm của các tập con các thuộc tính của lược đồ quan hệ. Các tính chất của các phép toán đại số không chỉ cho thấy đặc trưng thao tác dữ liệu của chúng mà còn cho thấy mô hình dữ liệu cũng như tập các phép toán được xây dựng là đúng đắn. Sau đây, các tính chất của các phép toán đại số trong CSDL truyền thống lần lượt được mở rộng cho URDB. Chứng minh của các tính chất này được dựa trên các định nghĩa của các chiến lược kết hợp xác suất và các định nghĩa các phép toán đại số.

Định lý 4.7.1 Giả sử r là một quan hệ xác suất trên lược đồ R trong URDB. Gọi ϕ_1 và ϕ_2 là hai điều kiện chọn. Khi đó

$$\sigma_{\phi_1}(\sigma_{\phi_2}(r)) = \sigma_{\phi_2}(\sigma_{\phi_1}(r)) = \sigma_{\phi_1 \wedge \phi_2}(r) \quad (1)$$

Với giả thiết trong phép chọn $\sigma_{\phi_1 \wedge \phi_2}(r)$ các điều kiện chọn ϕ_1 và ϕ_2 là có cùng một biến bộ. Kết quả này chứng tỏ thứ tự của các điều kiện chọn không ảnh hưởng đến kết quả phép chọn.

Chứng minh: Giả sử $r_2 = \sigma_{\phi_2}(r)$. Khi đó với mọi $t \in r$ ta có:

$$\begin{aligned} \sigma_{\phi_1}(\sigma_{\phi_2}(r)) &= \{t \in r_2 \mid \text{prob}_{R,r_2,t} \models \phi_1\} \\ &= \{t \in r \mid (\text{prob}_{R,r,t} \models \phi_2) \wedge (\text{prob}_{R,r_2,t} \models \phi_1)\} \\ &= \{t \in r \mid (\text{prob}_{R,r,t} \models \phi_2) \wedge (\text{prob}_{R,r,t} \models \phi_1)\} \text{ (do } r_2 \subseteq r) \\ &= \{t \in r \mid (\text{prob}_{R,r,t} \models \phi_1 \wedge \phi_2)\} \text{ (Định nghĩa 4.2.4)} \\ &= \sigma_{\phi_1 \wedge \phi_2}(r) \end{aligned}$$

Từ đó hệ thức $\sigma_{\phi_1}(\sigma_{\phi_2}(r)) = \sigma_{\phi_1 \wedge \phi_2}(r)$ được chứng minh. Hệ thức $\sigma_{\phi_2}(\sigma_{\phi_1}(r)) = \sigma_{\phi_2 \wedge \phi_1}(r)$ được chứng minh tương tự. Vì $\phi_1 \wedge \phi_2 \Leftrightarrow \phi_2 \wedge \phi_1$ (phép hội trên tập các điều kiện chọn xác suất cũng như trên mệnh đề có tính giao hoán), nên $\sigma_{\phi_1 \wedge \phi_2}(r) = \sigma_{\phi_2 \wedge \phi_1}(r)$. Từ đó suy ra hệ thức $\sigma_{\phi_1}(\sigma_{\phi_2}(r)) = \sigma_{\phi_2}(\sigma_{\phi_1}(r))$ và do đó $\sigma_{\phi_1}(\sigma_{\phi_2}(r)) = \sigma_{\phi_2}(\sigma_{\phi_1}(r)) = \sigma_{\phi_1 \wedge \phi_2}(r)$. Ngoài ra, có thể chứng minh trực tiếp hệ thức $\sigma_{\phi_1}(\sigma_{\phi_2}(r)) = \sigma_{\phi_2}(\sigma_{\phi_1}(r))$ không mấy khó khăn. Như vậy, định lý 5.7.1 đã được chứng minh.

Định lý 4.7.2 Giả sử R là một lược đồ URDB, r là quan hệ trên R , \oplus là một chiến lược tuyển xác suất, A và B là hai tập hợp con của các thuộc tính của R , $A \subseteq B$. Khi đó

$$\Pi_{A \oplus}(\Pi_{B \oplus}(r)) = \Pi_{A \oplus}(r) \quad (2)$$

Chứng minh: Vì $A \subseteq B$, nên $A \cap B = A$ và các vế của (2) là các quan hệ trên cùng một lược đồ (Định nghĩa 4.3.1) có cùng một tập các bộ cùng giá trị. Từ đó, chúng ta dễ dàng thấy rằng $\Pi_{A \oplus}(\Pi_{B \oplus}(r)) = \Pi_{A \cap B \oplus}(r) = \Pi_{A \oplus}(r)$ theo \oplus . Do đó, phương trình (2) được chứng minh.

Định lý 4.7.3 Giả sử R_1, R_2 và R_3 là các lược đồ URDB sao cho nếu chúng có các thuộc tính cùng tên thì các thuộc tính đó có cùng miền giá trị, r_1, r_2 và r_3 lần lượt là các quan hệ trên R_1, R_2 và R_3 , \otimes là một chiến lược hội xác suất. Khi đó

$$r_1 \bowtie_{\otimes} r_2 = r_2 \bowtie_{\otimes} r_1 \quad (3)$$

$$(r_1 \bowtie_{\otimes} r_2) \bowtie_{\otimes} r_3 = r_1 \bowtie_{\otimes} (r_2 \bowtie_{\otimes} r_3) \quad (4)$$

Phương trình (3) và (4) chứng tỏ phép kết của các quan hệ URDB có tính giao hoán và kết hợp.

Chứng minh: Các hệ thức trong định lý này lần lượt được chứng minh như sau:

Hệ thức (3): Thứ nhất, theo quy ước CSDL truyền thống (cũng như trong URDB) các cặp bộ (t_1, t_2) và (t_2, t_1) được coi là như nhau (cùng một phần tử của tích Descartes). Thứ hai, các chiến lược hội các bộ ba xác suất là giao hoán theo Định nghĩa 2.4.1 (nhờ tính giao hoán của phép giao các tập hợp và các chiến lược hội xác suất). Vì vậy, suy ra hệ thức $r_1 \bowtie_{\otimes} r_2 = r_2 \bowtie_{\otimes} r_1$. Tính giao hoán của phép kết được chứng minh.

Hệ thức (4): Do các lược đồ R_1, R_2 và R_3 được giả sử nếu có thuộc tính chung thì các thuộc tính đó cùng miền giá trị nên hai vế của (4) là các quan hệ trên cùng một lược đồ. Giả sử A là một thuộc tính chung trong các tập thuộc tính của R_1, R_2 và R_3 , do giao các tập hợp có tính kết hợp, từ Định nghĩa 2.4.1 suy ra hội các bộ ba xác suất là giá trị thuộc tính A của các bộ trên r_1, r_2 , và r_3 là kết hợp. Vì vậy, phép kết các giá trị bộ xác suất có tính kết hợp. Từ các lập luận trên suy ra tính kết hợp của phép kết các quan hệ URDB. Vì vậy, hệ thức $(r_1 \bowtie_{\otimes} r_2) \bowtie_{\otimes} r_3 = r_1 \bowtie_{\otimes} (r_2 \bowtie_{\otimes} r_3)$ được chứng minh.

Bởi vì phép lấy tích Descartes là trường hợp riêng của phép kết nên một hệ quả trực tiếp của định lý 4.7.3 được phát biểu như sau.

Hệ quả 4.7.1 Giả sử R_1, R_2 và R_3 là các lược đồ URDB đôi một không có thuộc tính chung và r_1, r_2 , và r_3 tương ứng là các quan hệ trên R_1, R_2 và R_3 . Khi đó

$$r_1 \times r_2 = r_2 \times r_1 \quad (5)$$

$$(r_1 \times r_2) \times r_3 = r_1 \times (r_2 \times r_3) \quad (6)$$

Các tính chất của các phép toán giao, hợp và trừ các quan hệ URDB được phát biểu bởi Định lý 4.7.4 như dưới đây.

Định lý 4.7.4 Giả sử R là một lược đồ URDB, r_1, r_2 và r_3 là các quan hệ xác suất trên R . Gọi \otimes và \oplus tương ứng là các chiến lược hội và tuyến xác suất. Khi đó

$$r_1 \cap_{\otimes} r_2 = r_2 \cap_{\otimes} r_1 \quad (7)$$

$$(r_1 \cap_{\otimes} r_2) \cap_{\otimes} r_3 = r_1 \cap_{\otimes} (r_2 \cap_{\otimes} r_3) \quad (8)$$

$$r_1 \cup_{\oplus} r_2 = r_2 \cup_{\oplus} r_1 \quad (9)$$

$$(r_1 \cup_{\oplus} r_2) \cup_{\oplus} r_3 = r_1 \cup_{\oplus} (r_2 \cup_{\oplus} r_3) \quad (10)$$

Các hệ thức (7), (8), (9) và (10) cho thấy các phép toán giao và hợp các quan hệ URDB tương ứng có tính giao hoán và kết hợp.

Chứng minh: Các hệ thức trong định lý này được chứng minh như sau:

Hệ thức (7) và (8): Từ tính giao hoán và kết hợp của phép giao các tập hợp suy ra các chiến lược hội các bộ ba xác suất có tính giao hoán và kết hợp. Do vậy, từ Định nghĩa 4.6.1 suy ra tập các bộ chung (cùng giá trị tập hợp trong các bộ ba xác suất) trong vế trái và vế phải của mỗi hệ thức (7) và (8) là như nhau. Từ đó suy ra các hệ thức (7) và (8).

Hệ thức (9) và (10): Từ tính giao hoán của phép hợp, phép giao các tập hợp, phép tuyến các bộ ba xác suất (Định nghĩa 2.4.2) suy ra hệ thức (9). Đối với hệ thức (10), không mất tính tổng quát, giả sử rằng mỗi bộ t thuộc về một trong các quan hệ r_1, r_2 và r_3 thì cũng sẽ xuất hiện trong hai quan hệ còn lại; hơn nữa, tập V các giá trị của một thuộc tính A của t trong cả ba quan hệ này luôn luôn như nhau. Điều này có thể thực hiện được bằng cách thêm t vào các quan hệ mà nó còn thiếu và thêm giá trị v vào V trong r_i và đặt $\alpha(v) = \beta(v) = 0$ sao cho V là như nhau trong cả ba quan hệ. Theo giả thiết này, mọi bộ t trong mỗi quan hệ đều thuộc giao của r_1, r_2 , và r_3 . Bây giờ, áp dụng Định nghĩa 4.6.2 với tính kết hợp của các chiến lược tuyến các bộ ba xác suất suy ra hệ thức (10).

Để kết thúc phần này, chúng tôi nêu ước lượng về độ phức tạp tính toán của các phép toán đại số trên các quan hệ URDB, để thấy tính khả áp dụng của URDB trong thực tế.

Từ các định nghĩa của các phép toán, có thể thấy chúng được tính toán trong thời gian đa thức theo kích thước của các quan hệ URDB. Chẳng hạn, với phép chọn, thời gian tính toán để xác định một bộ có thỏa mãn một điều kiện chọn hay không theo Định nghĩa 4.2.3 và 4.2.4 là bị chặn trên bởi một hằng số, theo $\max\{|V|\}$ với mọi tập giá trị V của các thuộc tính bộ trong một quan hệ URDB (lưu ý số phần tử trong V là hữu hạn). Từ đó, nếu số bộ trong một quan hệ là n thì thời gian thực hiện một phép chọn trên quan hệ này là $O(n)$. Có thể nhận xét tương tự như vậy cho các phép toán đại số khác như tích Descartes, kết, giao, v.v.. Ví dụ, đối với phép kết, từ Định nghĩa 4.5.1, có thể thấy thời gian kết hợp các giá trị của các thuộc tính chung của một cặp bộ tương ứng trên hai quan hệ bị chặn trên bởi một hằng số, theo $\max\{|V|\}$ với mọi tập giá trị V của các thuộc tính chung của các bộ trong cả hai quan hệ. Do vậy, từ Định nghĩa 4.5.1, suy ra phép kết hai quan hệ được tính toán trong thời gian là $O(m.n)$ nếu kích thước các quan hệ lần lượt là m và n .

4.8. Kết luận

Trong Chương 4 này, tất cả các phép toán đại số quan hệ xác suất cơ bản trên mô hình URDB đã được xây dựng. Trong đó đặc biệt chú ý các phép toán như chọn, chiếu, kết, giao, hợp và trừ vì chúng là các phép toán thể hiện bản chất về truy vấn và xử lý thông tin không chắc chắn (thông tin xác suất). Trong các phép toán đại số, phép chọn được dùng cho các truy vấn tìm kiếm các thông tin không chắc chắn của các thuộc tính quan hệ (cũng xem như là thuộc tính đối tượng). Một bộ trong quan hệ thỏa mãn một truy vấn chọn nếu nó thỏa một ngưỡng xác suất nào đó do người sử dụng yêu cầu (có thể coi mô hình CSDL quan hệ truyền thống là mô hình xác suất khi ngưỡng xác suất yêu cầu truy vấn là 1). Một tập các tính chất của các phép toán đại số đã được đề nghị và chứng minh chặt chẽ, chứng tỏ mô hình CSDL xác suất đã xây dựng là đúng.

Chương 5

HIỆN THỰC HỆ QUẢN TRỊ CỦA MÔ HÌNH URDB

5.1. Giới thiệu

Chương này trình bày ý tưởng và phương pháp thực hiện cho mô hình URDB như là một phần mềm hệ quản trị CSDL quan hệ xác suất. Phần mềm được xây dựng dựa trên hệ quản trị cơ sở dữ liệu SQLite như một nền tảng để xử lý và lưu trữ dữ liệu ở mức thấp. Phần mềm hiện thực mô hình URDB bao gồm một tập các lớp xử lý dữ liệu có xác suất bên trên SQLite có thể được coi như một hệ quản trị CSDL xác suất và được gọi là URDB-SQLite. Phần 5.2 giới thiệu tổng quan về hệ quản trị CSDL SQLite cùng với các hàm và thư viện được sử dụng trong chương trình. Phần 5.3 trình bày các bước thiết kế cho hệ quản trị URDB-SQLite bao gồm yêu cầu chung đối với hệ thống và mô hình kiến trúc tổng quan của hệ quản trị. Phần 5.4 trình bày các bước hiện thực cho khối biểu diễn mô hình URDB bao gồm lược đồ, quan hệ, thuộc tính, kiểu dữ liệu, bộ ba xác suất và giá trị bộ xác suất. Phần 5.5 trình bày các bước hiện thực cho khối xử lý truy vấn, cụ thể là truy vấn chọn bao gồm việc phân tích truy vấn, xử lý biểu thức chọn, xử lý điều kiện chọn và diễn dịch xác suất. Phần 5.6 giới thiệu giao diện người dùng và cách sử dụng hệ thống URDB-SQLite để tạo lược đồ, tạo quan hệ, truy vấn trên các quan hệ và lưu trữ cơ sở dữ liệu. Cuối cùng, Phần 5.7 là các kết luận và lưu ý của chương này.

5.2. Các tính năng đặc trưng của SQLite

5.2.1. Tổng quan

Với khả năng ứng dụng ngày càng mạnh mẽ của mô hình CSDL quan hệ truyền thống, đã có rất nhiều hệ quản trị CSDL quan hệ xuất hiện, trong đó có SQLite.

SQLite là một hệ quản trị CSDL mã nguồn mở hiện đang rất phổ biến. SQLite ra đời vào năm 2000 và có khả năng cung cấp đầy đủ các tiện ích của một hệ quản trị CSDL. Ngoài ra, SQLite có kích thước nhỏ, không cần cài đặt, dễ sử dụng và đáng tin cậy. Đó là lý do vì sao SQLite được lựa chọn cho việc lưu trữ và xử lý dữ liệu ở mức thấp của hệ thống PRDB-SQLite.

5.2.2. Các tính năng đặc trưng của SQLite

SQLite là phần mềm nhúng mã nguồn mở. SQLite hoạt động như một tiến trình độc lập bên trong ứng dụng và thực hiện được hầu hết các chức năng của một hệ quản trị CSDL thông thường. Mã nguồn của SQLite được nhúng vào mã nguồn của chương trình và đóng vai trò là một engine dùng để truy xuất dữ liệu.

Ưu điểm của việc tích hợp SQLite vào trong chương trình là người dùng không cần phải cấu hình mạng hay cấu hình chức năng quản lý truy cập khi cần sử dụng. Cả máy khách và máy chủ đều chạy trên cùng một tiến trình. Điều này giúp làm giảm thời gian giao tiếp giữa chương trình ứng dụng và cơ sở dữ liệu, đơn giản hóa việc quản lý dữ liệu và làm cho ứng dụng dễ dàng triển khai hơn. Mọi thứ cần thiết đều được biên dịch ngay trong chương trình.

Một ưu điểm khác của SQLite là được hỗ trợ bởi nhiều nhà cung ứng sản phẩm trong việc tích hợp SQLite vào các ngôn ngữ lập trình khác nhau. Ngày nay, các ngôn ngữ lập trình có thể được sử dụng với SQLite là Perl, Ruby, Python, Java, Tcl, PHP, C, C++, C#, v.v. Với việc mở rộng phiên bản SQLite dùng cho công nghệ lập trình ADO.NET trên môi trường .NET framework của Visual Studio, việc phát triển ứng dụng của các lập trình viên .NET cũng trở nên dễ dàng và thuận tiện hơn.

Phiên bản SQLite.NET trong là một phiên bản mã nguồn mở của SQLite3 và được viết bằng ngôn ngữ C# 3.0. Phiên bản này đầu tiên được thiết kế để làm việc với MonoTouch trên thiết bị iPhone, và sau đó được mở rộng để làm việc với bất cứ môi trường CLI (command-line interface) nào khác. Phiên bản này có các đặc điểm như:

1. Dễ dàng tích hợp vào các dự án (project)
2. Chạy nhanh và hiệu quả

3. Có nhiều phương thức cho việc thực thi các truy vấn một cách an toàn (có sử dụng các tham số) và cho việc nhận kết quả trả về đúng kiểu dữ liệu.
4. Hỗ trợ truy vấn với Linq
5. Có khả năng làm việc với các mô hình dữ liệu mà không phải thay đổi cấu trúc lớp.

Không phụ thuộc vào việc biên dịch thư viện sqlite3. Các lớp thư viện chủ yếu được sử dụng trong phiên bản này là: (1) Lớp SQLite.dll: chứa các phương thức tĩnh cho phép kết nối và truy xuất dữ liệu trên cơ sở dữ liệu sqlite; (2) Lớp SQLite.NET.dll: chứa các phương thức và đối tượng để làm việc với cơ sở dữ liệu sqlite3 trên môi trường .NET framework; (3) Lớp SQLite3.dll: chứa các phương thức tĩnh cho phép kết nối và truy xuất dữ liệu trên cơ sở dữ liệu sqlite3.

5.2.3. Các đối tượng và phương thức chính trong SQLite.Net

Phiên bản SQLite.NET chính là phiên bản được chúng tôi lựa chọn làm nền tảng để xây dựng hệ quản trị URDB-SQLite. Sau đây, chúng tôi xin giới thiệu một số đối tượng và phương thức chính được sử dụng trong quá trình truy cập cơ sở dữ liệu:

1. SQLiteConnection: đối tượng thể hiện việc kết nối đến cơ sở dữ liệu SQLite.
2. SQLiteDataAdapter: đối tượng thực hiện các lệnh truy vấn và kết nối với CSDL. Kết quả của truy vấn có thể được chứa vào một DataSet hoặc DataTable.
3. SQLiteCommand: đối tượng thể hiện câu truy vấn SQL để thực thi cho cơ sở dữ liệu SQLite.
4. SQLiteDataReader: đọc các bộ dữ liệu từ cơ sở dữ liệu SQLite.
5. SQLiteDataSet: đối tượng thể hiện tập các dữ liệu và quan hệ được lưu trữ trong bộ nhớ.
6. SQLiteDataTable: đối tượng thể hiện tập các quan hệ được lưu trữ trong bộ nhớ.
7. SQLiteParameter: đối tượng thể hiện tập các tham số được truyền vào câu truy vấn.

8. `ExecuteNonQuery()`: phương thức của đối tượng `SQLiteCommand` được dùng cho việc cập nhật cơ sở dữ liệu và trả về số dòng thực thi.
9. `ExecuteScalar()`: phương thức thực thi các hàm tính toán hoặc thống kê trong cơ sở dữ liệu và trả về cột đầu tiên trong dòng đầu tiên của tập kết quả.

5.3. Thiết kế tổng quan hệ quản trị URDB-SQLite

Hệ quản trị URDB-SQLite được hiện thực dựa trên nền tảng lý thuyết của mô hình URDB. Để định ra những chức năng của hệ quản trị URDB-SQLite, trước hết, ta cần xem xét lại những vấn đề cốt lõi được đưa ra từ mô hình URDB.

Các yêu cầu của hệ thống

Mỗi lược đồ URDB bao gồm một tập các thuộc tính đôi một khác nhau. Mỗi lược đồ URDB thể hiện thông tin về các thuộc tính của các đối tượng trong CSDL. Để thể hiện lược đồ URDB ta cần xây dựng một lớp `ProbScheme` có chứa danh sách các tập thuộc tính. Ngoài ra, trong lớp này còn có một thuộc tính lưu lại tên lược đồ để phân biệt với các lược đồ khác trong cùng một cơ sở dữ liệu.

Mỗi quan hệ URDB trên một lược đồ quan hệ R là một tập hữu hạn các bộ trên tập các thuộc tính của R . Mỗi quan hệ URDB cũng chính là một thể hiện của lược đồ URDB tương ứng. Để hiện thực các quan hệ URDB ta cần xây dựng một lớp `ProbRelation`. Trong lớp `ProbRelation` này có chứa một thuộc tính kiểu `ProbScheme` để cho biết quan hệ thuộc về lược đồ URDB tương ứng và có chứa một thuộc tính kiểu danh sách dùng để lưu trữ tập các bộ trong một quan hệ. Ngoài ra, trong lớp này còn có một thuộc tính lưu lại tên quan hệ để phân biệt với các quan hệ khác trong cùng một cơ sở dữ liệu.

Thuộc tính quan hệ trong URDB được dùng để biểu diễn thông tin trạng thái về tập các đối tượng trong CSDL. Mỗi thuộc tính của quan hệ được gắn liền với một kiểu dữ liệu xác định. Để hiện thực các thuộc tính trong các quan hệ URDB, ta cần xây dựng một *lớp mẫu* (template class) có tên là `ProbAttribute` chứa các kiểu dữ liệu kết hợp với tên các thuộc tính tương ứng.

Trong mô hình URDB có ba kiểu giá trị: kiểu nguyên tố, kiểu dữ liệu liệt kê và kiểu bộ. Kiểu nguyên tố là những kiểu cơ bản như: integer, string, bool,... Ngoài các kiểu nguyên tố cơ bản, người dùng còn có thể tự định nghĩa ra một tập giá trị đại diện một kiểu dữ liệu thông thường được gọi là kiểu dữ liệu liệt kê. Kiểu bộ bao gồm một tập hữu hạn các kiểu cơ bản hoặc kiểu do người dùng tự định nghĩa của các thuộc tính kết hợp với nhau biểu diễn các bộ của các quan hệ. Kiểu giá trị nói chung trong URDB có thể được hiện thực bằng việc xây dựng một lớp ProbDataType. Trong đó, các thuộc tính như TypeName dùng để lưu tên của kiểu dữ liệu và thuộc tính Domain dùng để lưu trường giá trị của kiểu dữ liệu. Nếu kiểu dữ liệu là kiểu do người dùng định nghĩa thì Domain sẽ lưu lại một tập giá trị tương ứng với kiểu dữ liệu đó.

Ngoài các thuộc tính thuộc ProbAttribute, một thuộc tính quan hệ quan trọng và cũng là cốt lõi của toàn bộ khóa luận này là thuộc tính xác suất (Ps). Tất cả các quan hệ yêu cầu phải có thuộc tính Ps và nằm trong danh sách tập thuộc tính của một quan hệ. Thuộc tính xác suất Ps được xây dựng bởi lớp ElemProb bao gồm hai thuộc tính nhỏ kiểu float: upperBound và lowerBound. UpperBound và lowerBound tương ứng với cận trên và cận dưới của một xác suất và có giá trị nằm trong khoảng $[0,1]$ ($\text{upperBound} \geq \text{lowerBound}$).

Giá trị kiểu bộ trong URDB là một danh sách các tập các giá trị tương ứng với từng thuộc tính trên một bộ dữ liệu của quan hệ và thuộc tính xác suất Ps. Giá trị kiểu bộ được biểu diễn bằng một lớp PTuple có chứa một danh sách các tập giá trị dùng để lưu lại các giá trị của các thuộc tính PAttribute và một thuộc tính ElemProb dùng để lưu trữ khoảng xác suất trên một bộ của quan hệ.

Các phép toán đại số là các tác vụ cơ bản trong mô hình URDB. chúng cho phép thực hiện các truy vấn và thao tác trên các quan hệ của URDB. Tuy nhiên, vì đây là mô hình dữ liệu quan hệ xác suất nên cú pháp và ngữ nghĩa của câu truy vấn và tính toán (thao tác) sẽ khác với cú pháp truy vấn và thao tác trên mô hình CSDL quan hệ truyền thống. Theo như định nghĩa trong mô hình URDB, một phép truy vấn chọn định ra một điều kiện chọn, trong điều kiện chọn có các biểu thức chọn được kết hợp bằng các phép toán \otimes , \oplus , \ominus theo một chiến lược xác suất nhất định. Do

đó, để hiện thực được truy vấn chọn này, ta cần xây dựng các lớp SelectionCondition, QueryExcutor để xử lý câu điều kiện và biểu thức chọn. Để hiện thực được các phép toán khác (chiều, tích Descartes, kết, giao, hợp và trừ) ta cần các hàm và thủ tục tính toán và kết hợp xác suất của các bộ trong quan hệ.

Tóm lại, các yêu cầu chung đối với việc hiện thực hệ quản trị URDB gồm:

1. Biểu diễn và lưu trữ các phần tử được định nghĩa từ các khái niệm trong mô hình URDB, bao gồm lược đồ, quan hệ URDB, kiểu bộ cũng như phân bố xác suất trên tập các giá trị tương ứng với tập các thuộc tính.
2. Thực hiện truy vấn và thao tác trên các quan hệ URDB. Công việc này đòi hỏi cần có một khối thực hiện *phân tích* (parse) và thực thi truy vấn, thao tác cùng với một thư viện các hàm thực hiện diễn dịch xác suất và thực hiện các chiến lược kết hợp xác suất.

Cung cấp một giao diện thân thiện cho người sử dụng. Các lớp giao diện này chính là tầng trung gian chuyển đổi qua lại giữa thông tin người dùng nhập vào và quá trình xử lý bên trong của hệ thống.

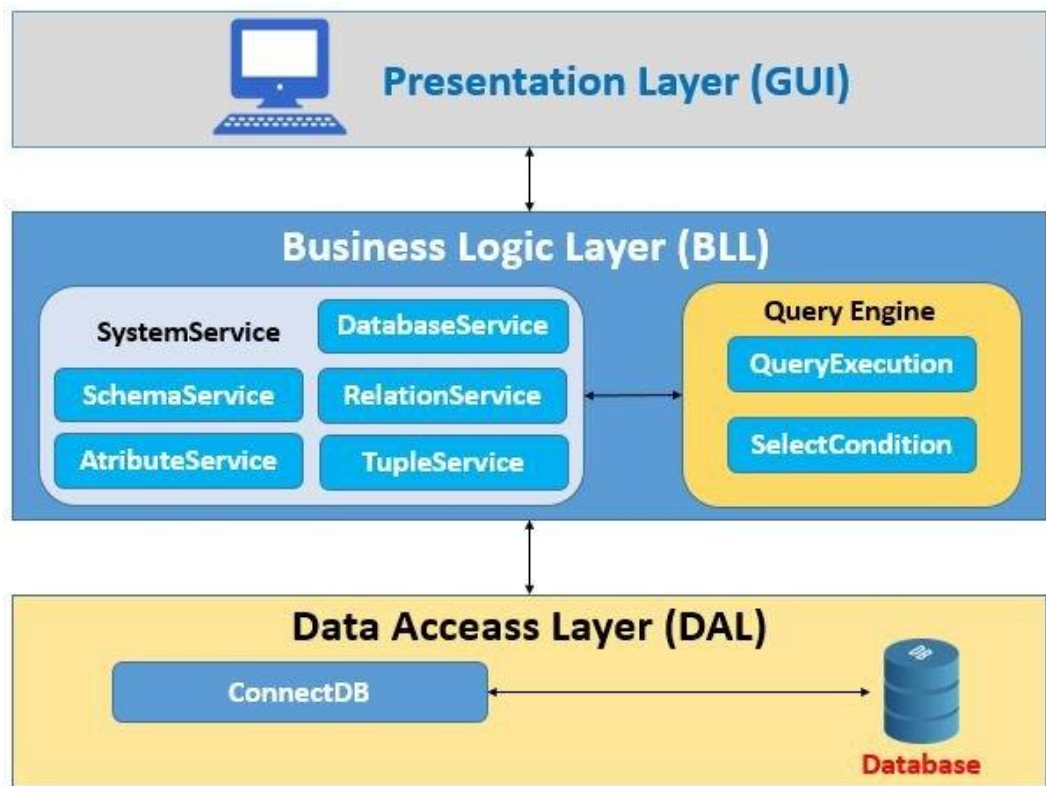
5.4. Kiến trúc tổng quan của hệ quản trị URDB-SQLite

Từ những yêu cầu đặc tả bên trên, hệ quản trị URDB được chúng tôi phát triển có kiến trúc như Hình 5.4.1. Kiến trúc này cho một cái nhìn tổng quan về toàn bộ hệ thống URDB-SQLite.

Kiến trúc hệ quản trị URDB được chia làm ba tầng:

1. Tầng giao diện (Presentation Layer): Cung cấp một giao diện đồ họa người dùng, cho phép người sử dụng tạo ra tập các lược đồ cùng với các thuộc tính dữ liệu tương ứng (Schema Creation) và thực hiện các truy vấn và thao tác trên các quan hệ URDB (QueryEditor).
2. Tầng dịch vụ (Business Logic Layer): Tầng này còn cung cấp các chức năng lưu trữ và truy xuất dữ liệu. Là cầu nối tương tác giữa người dùng và cơ sở dữ liệu. Ngoài ra, tầng này còn cung cấp các loại hình truy cập hệ quản trị cơ sở dữ liệu khác nhau. Đáp ứng nhu cầu mở rộng URDB_SQLite sau này.

3. Tầng truy cập dữ liệu (Data Access Layer): tầng giao tiếp trực tiếp với cơ sở dữ liệu với các hoạt động liên quan đến kết nối và thực hiện các truy vấn trực tiếp đến dữ liệu trên cơ sở dữ liệu.



Hình 5.4.1 Kiến trúc hệ quản trị URDB

Phần tiếp theo, chúng tôi sẽ mô tả các bước hiện thực cho từng thành phần chức năng chính của hệ quản trị URDB.

5.5. Hiện thực khôi biểu diễn mô hình URDB

Trước hết hệ thống cần có một giao diện cho phép người dùng tạo lược đồ bằng cách nhập các thuộc tính và kiểu dữ liệu tương ứng. Để hiển thị thông tin về tập các thuộc tính do người dùng nhập vào, chúng tôi sử dụng điều khiển **DataGridView**. Đây là một điều khiển có dạng lưới do đó cho phép người dùng nhập cùng lúc tên thuộc tính, kiểu dữ liệu thuộc tính và các mô tả về thuộc tính trên cùng một dòng. Các thông tin đặc tả cho lược đồ, quan hệ và thuộc tính tạo ra được lưu lại dưới dạng text bởi phần mềm quản trị **SQLite**. Quá trình xây dựng lược đồ URDB, quan hệ URDB và các thuộc tính cần có sự thể hiện của các lớp như sau:

5.5.1. Lớp **ProbSchema**

Đây là lớp hỗ trợ lưu trữ các lược đồ quan hệ (Schema) trong URDB. Lớp **Schema** là một lớp cơ sở của hệ quản trị URDB-**SQLite**. Mỗi đối tượng của lớp

Schema gồm có tên lược đồ và danh sách các thuộc tính của lược đồ. Mỗi phần tử trong danh sách thuộc tính này là một đối tượng thuộc kiểu dữ liệu Attribute được định nghĩa trong lớp ProbAttribute. Mỗi lược đồ trên giao diện của URDB-SQLite sẽ được ánh xạ thành một đối tượng của lớp ProbSchema và được lưu trữ xuống CSDL thông qua đối tượng này.

Các thuộc tính chính:

1. public int IDScheme { get; set; }: định danh của lược đồ quan hệ.
2. public string SchemeName { get; set; }: tên lược đồ quan hệ.

Các phương thức chính:

1. internal List<string> ListOfAttributeNameToUpper(): hàm biến đổi tên các thuộc tính thành chữ hoa
2. internal List<int> ListIndexPrimaryKey(): hàm thêm số thứ tự.
3. internal List<string> ListOfAttributeNameToLower(): hàm biến đổi tên các thuộc tính thành chữ thường.

5.5.2. Lớp ProbRelation

Đây là lớp hỗ trợ lưu trữ các quan hệ. Mỗi đối tượng của lớp ProbRelation gồm có tên quan hệ, tập các thuộc tính của quan hệ, một danh sách các PTuple để lưu lại tập các bộ của quan hệ và một thuộc tính kiểu ProbScheme để cho biết quan hệ này thuộc về lược đồ quan hệ nào.

Các thuộc tính chính:

1. public int IDRelation { get; set; }: định danh của lược đồ quan hệ trong cơ sở dữ liệu.
2. public string RelationName { get; set; }: tên lược đồ quan hệ.
3. public ProbScheme Scheme { get; set; }: lược đồ quan hệ tương ứng.

Các phương thức chính:

1. internal List<ProbRelation> getAllRelation(): lấy danh sách các quan hệ.
2. internal void DropTableByTableName(): xóa table theo tên.
3. internal void DeleteAllRelation(): xóa tất cả các quan hệ.
4. internal void InsertSystemRelation(): thêm các quan hệ.

5.5.3. Lớp ProbAttribute

Đây là lớp hỗ trợ lưu trữ các thuộc tính của lược đồ hoặc quan hệ. Lớp ProbAttribute là một lớp cơ sở của hệ quản trị URDB-SQLite. Mỗi đối tượng của lớp ProbAttribute gồm có tên thuộc tính và kiểu dữ liệu của thuộc tính. Kiểu dữ liệu thuộc tính là kiểu ProbDataType được định nghĩa trong lớp ProbDataType.

Các thuộc tính chính:

1. public int IDAttribute { get; set; }: định danh thuộc tính.
2. public string AttributeName { get; set; }: Tên của các thuộc tính
3. public ProbDataType Type { get; set; }: loại thuộc tính

Các phương thức chính:

1. internal void DeleteAllAttribute(): xóa tất cả các thuộc tính.
2. internal void Insert(): Thêm thuộc tính.
3. internal List<ProbAttribute> getListAttributeByIDScheme(int IDScheme): lấy danh sách các thuộc tính theo định danh của lược đồ quan hệ.

5.5.4. Lớp ProbDataType

Như đã trình bày ở mục 5.3.1, cơ sở dữ liệu quan hệ xác suất có 3 dạng kiểu dữ liệu đó là: kiểu nguyên tố, kiểu dữ liệu liệt kê và kiểu bộ. Kiểu nguyên tố là những kiểu cơ bản như: integer, string, bool,... Kiểu bộ bao gồm một tập hữu hạn các kiểu cơ bản của các thuộc tính kết hợp với nhau biểu diễn các bộ của các quan hệ. Ngoài ra, trong cơ sở dữ liệu quan hệ nói chung, kiểu bộ đơn giản chỉ là một tập hợp các kiểu của các thuộc tính trên bộ và trong các tác vụ liên quan đến các bộ của quan hệ, ta thường chỉ ra cụ thể giá trị thuộc tính trên bộ để đưa vào các phép so sánh hoặc tính toán. Do vậy, việc hiện thực các kiểu dữ liệu cơ bản cho từng thuộc tính trong các quan hệ đã đủ đáp ứng tất cả các yêu cầu về kiểu dữ liệu của hệ quản trị URDB-SQLite.

Trong số các kiểu dữ liệu cơ bản, kiểu dữ liệu liệt kê là một dạng kiểu dữ liệu đặc biệt. Nó cho phép người dùng tự định nghĩa kiểu và tập giá trị tương ứng với kiểu dữ liệu đó. Do vậy, trong quá trình hiện thực, chúng tôi đã xây dựng một form

nhập liệu có chức năng cho phép người dùng tự định nghĩa kiểu dữ liệu riêng với tên gọi và trường giá trị tương ứng của kiểu. Ngoài ra, chúng tôi cũng xây dựng một lớp ProbDataType có chứa một danh sách giá trị kiểu chuỗi để đảm bảo ghi nhận được các giá trị mà người dùng nhập vào. Sau đây, thuộc tính và phương thức chính của lớp ProbDataType sẽ được trình bày.

Các thuộc tính chính:

1. public string TypeName { get; set; }: Tên loại dữ liệu.
2. public string DataType { get; set; }: kiểu dữ liệu.

Các phương thức chính:

1. public bool CheckDataTypeOfVariables(string value): kiểm tra giá trị của kiểu dữ liệu.
2. public void GetDataType(): lấy kiểu dữ liệu.

5.5.5. Lớp ProbTuple

Giá trị bộ dữ liệu trong URDB chính là một danh sách các tập giá trị tương ứng của các thuộc tính và một thuộc tính có kiểu là ElemProb đại diện có xác suất của bộ . Giá trị bộ dữ liệu trong URDB-SQLite được hiện thực bằng một lớp ProbTuple có chứa danh sách các tập giá trị và một khoảng xác suất.

Các thuộc tính chính:

1. public List<ProbTriple> Triples { get; set; }: Tập các giá trị bộ ba xác suất trên một tuple.

Các phương thức chính:

1. internal List<ProbTuple> getAllTypeByRelationName(string relationname, int nTriples): lấy tất cả bộ theo tên lược đồ quan hệ.
2. internal void DeleteTypeById(): Xóa bộ theo id.

5.5.6. Lớp ProbTriple

Lớp lưu trữ dữ liệu của thuộc tính trong một bộ, lưu trữ tập hợp các bộ ba xác suất.

Các thuộc tính chính:

1. public List<double> MinProb { get; set; }: tập các giá trị cận dưới của các bộ ba xác suất.
2. public List<double> MaxProb { get; set; }: tập các giá trị cận dưới của các bộ ba xác suất.
3. public List<ValueOfTriple> Value2 { get; set; }: tập các tập giá trị của bộ ba xác suất.

Các phương thức chính:

1. public string GetStrValue(): xuất tập các bộ ba xác suất ra chuỗi giá trị.
2. public ProbTriple(string V): tạo một tập các bộ ba xác suất từ chuỗi giá trị.

5.5.7. Lớp ValueOfTriple

Lưu trữ và xử lý tập giá trị của một bộ ba xác suất

Các thuộc tính chính:

1. public List<Object> Value { get; set; }: tập hợp các giá trị của bộ ba xác suất.

Các phương thức chính:

1. public ValueOfTriple(String multiValue): tạo tập giá trị của bộ ba từ một chuỗi giá trị.
2. public string GetStrValue(): xuất tập giá trị ra chuỗi giá trị.

5.6. Hiện thực nhập xuất giá trị cho thuộc tính

5.6.1. Cách 1: nhập trực tiếp chuỗi vào bảng theo đúng định dạng

- Định dạng: {các giá trị của bộ ba} [cận trên và cận dưới của bộ ba]
- Nếu có nhiều hơn 1 bộ ba thì ngăn cách nhau bởi dấu ‘||’

Relation patient				
Query				
p_id	p_name	p_age	disease	p_cost
{ PT226 }[1, 1]	{ Oliver }[1, 1]	{ 65 }[1, 1]	{ lung cancer, tuberculosis }[0.3, 0.6]	{ 30, 35 }[0.35, ...
{ PT234 }[1, 1]	{ Blair }[1, 1]	{ 43, 44 }[1, 1]	{ hepatitis, cirrhosis }[0.22, 0.32] { cholecystitis }[0.22, 0.32]	{ 6, 7 }[0.4, 0.7]
{ PT242 }[1, 1]	{ Alice }[1, 1]	{ 36 }[1, 1]	{ cholecystitis }[1, 1]	{ 8 }[1, 1]
{ PT267 }[1, 1]	{ Anne }[1, 1]	{ 15 }[1, 1]	{ bronchitis, angina }[1, 1]	{ 7 }[1, 1]
{ PT345 }[1, 1]	{ Blair }[1, 1]	{ 25 }[1, 1]	{ hepatitis, cirrhosis }[0.3, 0.5] { cholecystitis }[0.5, 0.7]	{ 10 }[1, 1]

Hình 5.6.1 Nhập bộ ba xác suất trực tiếp

5.6.2. Cách 2: nhập thông qua giao diện

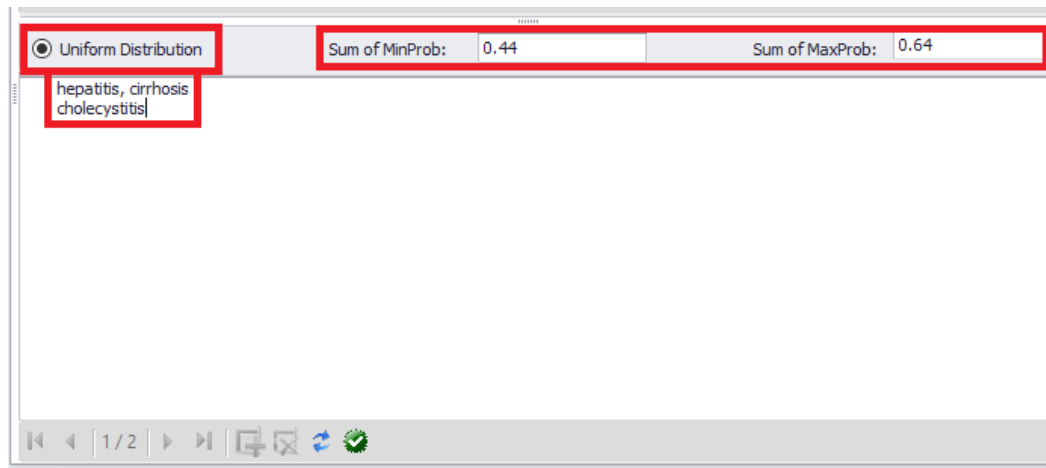
- ❖ Các bộ ba phân bố xác suất không đều:
 - Chọn UnUniform Distribution.
 - Nhập các bộ ba và phân bố xác suất vào theo từng hàng, mỗi hàng là một bộ 3 xác suất.

<input checked="" type="radio"/> UnUniform Distribution		<input type="radio"/> Uniform Distribution	
	Value	MinProb	MaxProb
▶	hepatitis, cirrhosis	0.3	0.5
	cholecystitis	0.5	0.7

Hình 5.6.2 Nhập các bộ ba xác suất phân bố không đều

- ❖ Các bộ ba phân bố xác suất đều:
 - Chọn Uniform Distribution.
 - Nhập tổng xác suất của các bộ ba vào Sum of MinProb và Sum of MaxProb.
 - Nhập các giá trị của từng bộ ba:

Mỗi bộ ba cùng một dòng và các giá trị cách nhau bằng dấu “,”.
 - Hệ thống sẽ tính toán và phân bố đều xác suất cho các bộ ba.



Hình 5.6.3 Nhập các bộ ba phân bố xác suất đều

5.7. Hiện thực khối xử lý truy vấn tập hợp

5.7.1. Lớp xử lý CompareProbTuple

Mục đích: So sánh 2 bộ dữ liệu, thực hiện tuyến xác suất cho phép chiếu.

Các thuộc tính chính:

1. private ProbRelation relationData: dữ liệu quan hệ cần được xử lý.
2. private List<Double> flags: mảng đánh dấu so sánh các bộ dữ liệu.
3. private ProbRelation relationResult: dữ liệu quan hệ trả về sau xử lý.

Các phương thức chính:

1. public CompareProbTuple(ProbRelation probRelation): nhập tập dữ liệu cần xử lý.
2. public ProbRelation equal(String rule): trả về tập dữ liệu đã xử lý theo rule là chiến lược kết hợp xác suất.

5.7.2. Các lớp xử lý Union, Intersect, Except

Bao gồm các lớp: Intersect, Except, CompareProbTuple

Các thuộc tính chính:

1. private ProbRelation relationData1: tập dữ liệu đầu vào của truy vấn con thứ nhất.

2. private ProbRelation relationData2: tập dữ liệu đầu vào của truy vấn con thứ hai.
3. private List<Double> flags: mảng đánh dấu khi so sánh các bộ dữ liệu.
4. private ProbRelation relationResult: dữ liệu quan hệ trả về sau xử lý.

Các phương thức chính:

1. public Constructor (ProbRelation probRelation1, ProbRelation probRelation2): nhận vào 2 tập dữ liệu của 2 truy vấn con.
2. public ProbRelation equal(String rule): trả về tập dữ liệu đã xử lý theo rule là chiến lược kết hợp xác suất.

5.7.3. Xử lý phép chọn tập hợp

Lớp CompareTriple

Mục đích: Xử lý các phép so sánh tập hợp.

Các phương thức chính:

1. public bool equal(ValueOfTriple triple, List<Object> condition): kiểm tra tập hợp bằng tập hợp điều kiện.
2. public bool difference(ValueOfTriple triple, List<Object> condition): kiểm tra tập hợp khác tập hợp điều kiện.
3. public bool greaterThan(ValueOfTriple triple, List<Object> condition): kiểm tra tập hợp chứa tập hợp điều kiện.
4. public bool lessThan(ValueOfTriple triple, List<Object> condition): kiểm tra tập hợp bị tập hợp điều kiện chứa.

5.7.4. Các lớp hỗ trợ

- CompareTwoTriple: so sánh 2 tập hợp dữ liệu trong bộ ba xác suất.
- HandleEqual: hỗ trợ tính toán tổ hợp xác suất cho điều kiện chọn giá trị hai thuộc tính bằng nhau.
- HandleValue: chuẩn hóa dữ liệu.
- ReplaceString: chuẩn hóa chuỗi điều kiện hỗ trợ cho phép chọn tập hợp.

5.8. Hiện thực khối xử lý tập hợp

- ❖ Điều kiện chọn bằng một tập hợp

Scheme	Relation patient	Query chọn7
--------	------------------	-------------


```

select p_id, p_name, disease
from patient
where (patient.disease = {hepatitis,cirrhosis})[0.2, 1]

```


Query Result	Message
--------------	---------

	patient.p_id	patient.p_name	patient.disease
▶	{ PT234 }[1, 1]	{ Blair }[1, 1]	{ hepatitis, cirrhosis }[0.22, 0.32] { cholecystitis }[0.22, 0.32]
	{ PT345 }[1, 1]	{ Blair }[1, 1]	{ hepatitis, cirrhosis }[0.3, 0.5] { cholecystitis }[0.5, 0.7]

Hình 5.8.1 Phép chọn với điều kiện là một tập hợp

- ❖ Điều kiện chọn hai thuộc tính có giá trị bằng nhau

Scheme

Relation patient

Query chọn11

```

select d_name, d_identity, d_phone
from doctor
where (doctor.d_phone =  $\otimes$ _ig doctor.d_identity)[0.2 , 1]

```

Query Result

Message

	Number	doctor.d_name	doctor.d_identity	doctor.d_phone
▶	1	{ Andrew }[1, 1]	{ 285765 }[0.5, 0.5]	{ 285765 }[0.5, 0.5] { 255816 }[0.5, 0.5]
	2	{ Leon }[1, 1]	{ 012345 }[0.2, 0.5]	{ 012345 }[0.2, 0.5]
	3	{ Oliver }[1, 1]	{ 999999 }[1, 1]	{ 999999 }[1, 1]

Hình 5.8.2 Phép chọn với điều kiện dữ liệu trên hai thuộc tính bằng nhau

❖ Điều kiện chọn chứa một tập dữ liệu

Scheme	Relation patient	Query chọn8
--------	------------------	-------------

```

select p_id, p_name, disease
from patient
where (patient.disease  $\supseteq$  {cholecystitis})[0.2, 1]

```


Query Result	Message
--------------	---------

	patient.p_id	patient.p_name	patient.disease
▶	{PT234}[1, 1]	{Blair}[1, 1]	{ hepatitis, cirrhosis }[0.22, 0.32] { cholecystitis }[0.22, 0.32]
	{PT242}[1, 1]	{Alice}[1, 1]	{ cholecystitis }[1, 1]
	{PT345}[1, 1]	{Blair}[1, 1]	{ hepatitis, cirrhosis }[0.3, 0.5] { cholecystitis }[0.5, 0.7]

Hình 5.8.3 Phép chọn với điều kiện chứa một tập dữ liệu

❖ Điều kiện chọn dữ liệu nằm trong một tập cha

Scheme	Relation patient	Query chọn9
--------	------------------	-------------

```

select p_id, p_name, disease, p_age
from patient
where (patient.disease  $\subseteq$  {hepatitis,cirrhosis,cholecystitis})[0.2, 1]

```


Query Result	Message
--------------	---------

	patient.p_id	patient.p_name	patient.disease	patient.p_age
▶	{PT234}[1, 1]	{Blair}[1, 1]	{ hepatitis, cirrhosis }[0.22, 0.32] { cholecystitis }[0.22, 0.32]	{ 43, 44 }[1, 1]
	{PT242}[1, 1]	{Alice}[1, 1]	{ cholecystitis }[1, 1]	{ 36 }[1, 1]
	{PT345}[1, 1]	{Blair}[1, 1]	{ hepatitis, cirrhosis }[0.3, 0.5] { cholecystitis }[0.5, 0.7]	{ 25 }[1, 1]

Hình 5.8.4 Phép chọn với điều kiện nằm trong một tập cha

- ❖ Điều kiện chọn dữ liệu không chứa tập điều kiện

Scheme	Relation patient	Query chọn10
--------	------------------	--------------

```

select p_id, p_name, disease
from patient
where (patient.disease  $\not\supseteq$  {cholecystitis})[0.2, 1]

```

Query Result	Message
--------------	---------

	patient.p_id	patient.p_name	patient.disease
▶	{ PT226 }[1, 1]	{ Oliver }[1, 1]	{ lung cancer, tuberculosis }[0.3, 0.6]
	{ PT234 }[1, 1]	{ Blair }[1, 1]	{ hepatitis, cirrhosis }[0.22, 0.32] { cholecystitis }[0.22, 0.32]
	{ PT267 }[1, 1]	{ Anne }[1, 1]	{ bronchitis, angina }[1, 1]
	{ PT345 }[1, 1]	{ Blair }[1, 1]	{ hepatitis, cirrhosis }[0.3, 0.5] { cholecystitis }[0.5, 0.7]

Hình 5.8.5 Phép chọn với điều kiện không chứa tập điều kiện

5.9. Hiện thực khối xử lý truy vấn và tính toán

Khối xử lý truy vấn trong hệ quản trị URDB-SQLite được xây dựng trên lớp SelectionCondition và QueryExcutor. Trong phạm vi khóa luận này, chúng tôi sẽ trình bày cú pháp cũng như cách xử lý câu truy vấn chọn. Đối với kiểu truy vấn chọn chúng tôi đặt ra một cú pháp chung cho câu truy vấn:

Select [tập các thuộc tính]

From [quan hệ được dùng để truy vấn]

Where [điều kiện chọn]

Cú pháp truy vấn này rất quen thuộc với người sử dụng vì nó giống với cú pháp truy vấn của ngôn ngữ SQL. Như đã thấy trong Chương 4, bản chất của truy vấn chọn trên URDB là truy vấn kiểu ngôn ngữ đại số. Tuy nhiên để có giao diện thân thiện hơn với người sử dụng, chúng tôi đã tái sử dụng câu truy vấn SQL của SQLite và tích hợp thêm vào đó điều kiện chọn của URDB. Như vậy, một truy vấn với cú pháp dạng SQL mới trong URDB-SQLite tương đương với một truy vấn chọn với điều kiện chọn tương ứng trong URDB. Điều khác biệt của truy vấn SQL mới trong URDB-SQLite so với truy vấn SQL thông thường là trong mỗi truy vấn chọn trên URDB chỉ có tối đa hai quan hệ xác suất được tham chiếu với các điều kiện chọn có dạng $(E)[\alpha, \beta]$. Trong một

điều kiện chọn phải có ít nhất từ một biểu thức chọn trở lên và nếu có nhiều biểu thức chọn thì giữa các biểu thức chọn phải có các phép toán kết hợp như phép and, or hoặc not. Sau khi nhận biết được truy vấn chọn của người dùng, chương trình sẽ tiến hành xử lý truy vấn chọn đó.

5.9.1. Xử lý thực thi câu truy vấn

Tác vụ này được tiến hành bởi các thuộc tính và phương thức được xây dựng trong lớp QueryExcutor, bao gồm:

Các thuộc tính chính:

1. public ProbRelation relations { get; set; }: tập các quan hệ
2. public ProbTuple tuple { get; set; }: tập các giá trị thuộc tính
3. public string conditionString { get; set; }: điều kiện chuỗi truy vấn

Các phương thức chính:

1. public bool CheckConditionString(): kiểm tra điều kiện của chuỗi truy vấn.
2. private static string converConditionStringToExpression(string conditionString): chuyển đổi chuỗi điều kiện sang biểu thức.
3. private bool IsSelectionExpression(string conditionString): kiểm tra điều kiện biểu thức.

static public string[] Operator = new string[21] { "_<", "_>", "<=", ">=", "_=", "!=", "⊗_in", "⊗_ig", "⊗_me", "⊕_in", "⊕_ig", "⊕_me", "equal_in", "equal_ig", "equal_me", "⊖_ig", "⊖_in", "⊖_me", "⊆", "⊇", "⊄" };: Xử lý thêm các phép toán cần thiết.

5.9.2. Xử lý điều kiện chọn

Sau khi phương thức ExecuteQuery() của lớp QueryExcutor được gọi nếu câu truy vấn chọn có chứa biểu thức chọn thì chương trình sẽ tạo ra một đối tượng thuộc lớp SelectionCondition và truyền hai tham số selectedRelations và conditionString vào cho phương thức khởi tạo của lớp này. Các phương thức khởi tạo sẽ gán giá trị của hai tham số trên vào các thuộc tính thành viên của lớp và xây dựng một tập attribute để lưu trữ các thuộc tính của quan hệ. Sau đó, phương thức ExecuteQuery()

sẽ lấy ra từng bộ trong tập các bộ của quan hệ được chọn và gọi phương thức Satisfied(tuple) để kiểm tra xem mỗi bộ trên có thỏa mãn điều kiện chọn hay không. Nếu bộ thỏa mãn điều kiện chọn thì nó sẽ được thêm vào quan hệ relationResult của lớp QueryExcutor. Đây cũng chính là tập kết quả của truy vấn.

Trong lớp SelectionCondition, phương thức Satisfied() sẽ phân tích chuỗi để lấy ra từng biểu thức chọn. Với mỗi biểu thức chọn tìm được, nó sẽ gọi phương thức ExpressionValue để tính toán giá trị của biểu thức chọn này. Giá trị của biểu thức chọn được trả về có kiểu là boolean sau đó sẽ được thêm trở lại vào chuỗi điều kiện chọn và thay thế cho chuỗi biểu thức chọn ban đầu. Sau khi tính toán hết giá trị của các biểu thức chọn có trong chuỗi, phương thức này tạo ra một danh sách kiểu chuỗi chứa các toán hạng và toán tử được sinh ra từ phép chuyển đổi biểu thức trung tố thành biểu thức hậu tố trong phương thức SC_PostfixNotation() của nó. Giá trị sau cùng của điều kiện chọn được tính toán bằng thuật toán nghịch đảo kí pháp Ba Lan và trả về một kết quả kiểu boolean.

Trong phần tiếp theo, chúng tôi sẽ trình bày về các phương thức chính để xử lý đối với biểu thức chọn.

Trong quá trình xử lý biểu thức chọn, để tránh xảy ra sự nhầm lẫn giữa các phép toán so sánh, chúng tôi đã chuyển đổi một số kí hiệu phép toán thông thường sang một dạng khác theo quy ước dưới đây:

“>”	→	“_>”
“<”	→	“_<”
“=”	→	“_=”
“>=”	→	“_>=”
“<=”	→	“_<=”
“!=”	→	“!=”
“=⊗in”	→	“EQUAL(in)”

$"=\otimes_{ig}"$	\rightarrow	$"EQUAL(ig)"$
$"=\otimes_{me}"$	\rightarrow	$"EQUAL(me)"$
$"\otimes_{in}"$	\rightarrow	$"\otimes_{in}"$
$"\otimes_{ig}"$	\rightarrow	$"\otimes_{ig}"$
$"\otimes_{me}"$	\rightarrow	$"\otimes_{me}"$
$"\oplus_{in}"$	\rightarrow	$"\oplus_{in}"$
$"\oplus_{ig}"$	\rightarrow	$"\oplus_{ig}"$
$"\oplus_{me}"$	\rightarrow	$"\oplus_{me}"$
$"\ominus_{in}"$	\rightarrow	$"\ominus_{in}"$
$"\ominus_{ig}"$	\rightarrow	$"\ominus_{ig}"$
$"\ominus_{me}"$	\rightarrow	$"\ominus_{me}"$

Quá trình xử lý chung của biểu thức chọn sau khi đã chuyển đổi thành dạng biểu thức hậu tố là:

Chương trình sẽ đọc các phân tử của biểu thức hậu tố từ trái sang phải

1. Nếu gặp các toán hạng thì thêm vào stack.
2. Nếu gặp toán tử thì lấy hai toán hạng trên cùng của stack ra và xét điều kiện của toán tử:
 - Toán tử so sánh giữa một thuộc tính với một giá trị \rightarrow thực hiện phép so sánh này và lấy kết quả là khoảng xác suất của các giá trị thỏa mãn điều kiện.
 - Toán tử so sánh bằng giữa hai thuộc tính trên cùng một bộ \rightarrow thực hiện phép so sánh bằng cách lấy từng phần tử giá trị của hai thuộc tính so sánh với nhau và thực hiện kết hợp các khoảng xác suất của các giá trị thỏa mãn điều kiện theo chiến lược hội đã lựa chọn.

- Toán tử kết hợp giữa hai khoảng xác suất \rightarrow tính toán giá trị khoảng xác suất kết hợp của hai khoảng trên dựa theo chiến lược xác suất đã chọn.
3. So sánh khoảng xác suất sau cùng của biểu thức chọn với khoảng xác suất điều kiện. Nếu khoảng xác suất này nằm trong khoảng xác suất điều kiện thì kết quả biểu thức chọn là true, ngược lại là false.

Phần cuối cùng, chúng tôi sẽ trình bày về giao diện người dùng của hệ quản trị và cách tạo ra các lược đồ, quan hệ cũng như truy vấn dữ liệu trên quan hệ.

5.9.3. Xử lý phép chiếu

Sau khi phương thức `ExecuteQuery()` của lớp `QueryExcutor` được gọi nếu câu truy vấn có chứa biểu thức chiếu thì chương trình sẽ kiểm tra điều kiện câu truy vấn viết có hợp lệ hay không ở lớp `SelectionCondition` và lớp `QueryExcutor` cụ thể ở phương thức `GetAttribute`. Nếu câu truy vấn không hợp lệ sẽ xuất ra thông báo lỗi. Ngược lại thì nó sẽ truyền attribute và gọi phương thức `getIntersect` để lấy dữ liệu. Sau đó, phương thức `ExecuteQuery()` sẽ lấy ra từng bộ trong tập các bộ của quan hệ, nó sẽ được thêm vào quan hệ `relationResult` của lớp `QueryExcutor`. Đây cũng chính là tập kết quả của truy vấn.

Đối với các bộ có cùng giá trị thì sẽ được tính lại cột xác suất tùy theo các chiến lược xác suất “ \otimes_{in} , \otimes_{me} , \otimes_{ig} , \oplus_{in} , \oplus_{ig} , \oplus_{me} , \ominus_{me} , \ominus_{ig} , \ominus_{in} ” mà bạn chọn.

5.9.4. Xử lý phép Descartes

Sau khi phương thức `ExecuteQuery()` của lớp `QueryExcutor` được gọi nếu câu truy vấn có chứa biểu thức Descartes thì chương trình sẽ kiểm tra điều kiện câu truy vấn viết có hợp lệ hay không ở lớp `SelectionCondition` và lớp `QueryExcutor` cụ thể ở phương thức `GetAttribute`. Nếu câu truy vấn không hợp lệ sẽ xuất ra thông báo lỗi. Ngược lại thì nó sẽ truyền attribute của hai bảng và gọi phương thức `Descartes()` để xử lý dữ liệu. Sau đó, phương thức `ExecuteQuery()` sẽ lấy ra từng bộ trong tập các bộ của quan hệ, nó sẽ được thêm vào quan hệ `relationResult` của lớp `QueryExcutor`. Đây cũng chính là tập kết quả của truy vấn.

5.9.5. Xử lý phép kết

Sau khi phương thức `ExecuteQuery()` của lớp `QueryExcutor` được gọi nếu câu truy vấn có chứa biểu thức của phép kết ***natural join in*** thì chương trình sẽ kiểm tra điều kiện câu truy vấn viết có hợp lệ hay không ở lớp `SelectionCondition` và lớp `QueryExcutor` cụ thể ở phương thức `GetAllRelation`. Nếu câu truy vấn không hợp lệ sẽ xuất ra thông báo lỗi. Ngược lại thì nó sẽ truyền attribute của hai bảng và gọi phương thức `NaturalJoin()`, `calProp` để xử lý dữ liệu. Sau đó, phương thức `ExecuteQuery()` sẽ lấy ra từng bộ trong tập các bộ của quan hệ, nó sẽ được thêm vào quan hệ `relationResult` của lớp `QueryExcutor`. Đây cũng chính là tập kết quả của truy vấn.

Đối với các bộ có cùng giá trị thì sẽ được tính lại cột xác suất tùy theo các chiến lược xác suất “ \otimes in, \otimes me, \otimes ig” mà bạn chọn.

5.9.6. Xử lý phép hợp

Sau khi phương thức `ExecuteQuery()` của lớp `QueryExcutor` được gọi nếu câu truy vấn có chứa biểu thức của phép hợp ***union*** thì chương trình sẽ kiểm tra điều kiện câu truy vấn viết có hợp lệ hay không ở lớp `SelectionCondition` và lớp `QueryExcutor` cụ thể ở phương thức `GetAllRelation`, `check_e_Val_eq`. Nếu câu truy vấn không hợp lệ sẽ xuất ra thông báo lỗi. Ngược lại thì nó sẽ truyền `pRelation1`, `pRelation2` gọi phương thức `unionOperator`, `calProp` để xử lý dữ liệu. Sau đó, phương thức `ExecuteQuery()` sẽ lấy ra từng bộ trong tập các bộ của quan hệ, nó sẽ được thêm vào quan hệ `relationResult` của lớp `QueryExcutor`. Đây cũng chính là tập kết quả của truy vấn.

5.9.7. Xử lý phép giao

Sau khi phương thức `ExecuteQuery()` của lớp `QueryExcutor` được gọi nếu câu truy vấn có chứa biểu thức của phép giao ***intersert*** thì chương trình sẽ kiểm tra điều kiện câu truy vấn viết có hợp lệ hay không ở lớp `SelectionCondition` và lớp `QueryExcutor` cụ thể ở phương thức `GetAllRelation`, `check_e_Val_eq`. Nếu câu truy vấn không hợp lệ sẽ xuất ra thông báo lỗi. Ngược lại thì nó sẽ truyền `pRelation1`, `pRelation2` gọi phương thức `intersertOperator`, `calProp` để xử lý dữ liệu. Sau đó,

phương thức `ExecuteQuery()` sẽ lấy ra từng bộ trong tập các bộ của quan hệ, nó sẽ được thêm vào quan hệ `relationResult` của lớp `QueryExcutor`. Đây cũng chính là tập kết quả của truy vấn.

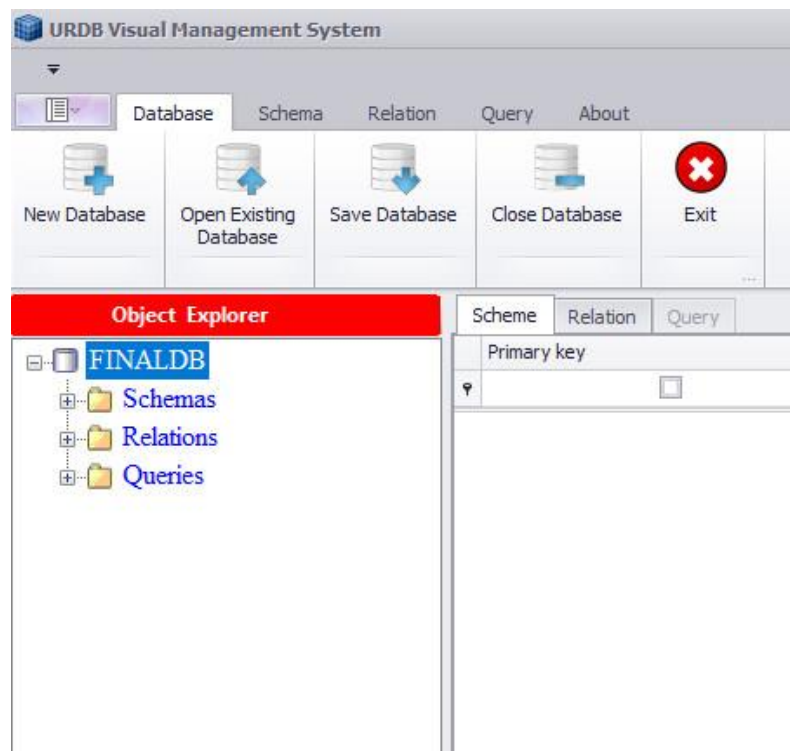
5.9.8. Xử lý phép trừ

Sau khi phương thức `ExecuteQuery()` của lớp `QueryExcutor` được gọi nếu câu truy vấn có chứa biểu thức của phép trừ **except** thì chương trình sẽ kiểm tra điều kiện câu truy vấn viết có hợp lệ hay không ở lớp `SelectionCondition` và lớp `QueryExcutor` cụ thể ở phương thức `GetAllRelation`, `check_e_Val_eq`. Nếu câu truy vấn không hợp lệ sẽ xuất ra thông báo lỗi. Ngược lại thì nó sẽ truyền `pRelation1`, `pRelation2` gọi phương thức `exceptOperator`, `calProp` để xử lý dữ liệu. Sau đó, phương thức `ExecuteQuery()` sẽ lấy ra từng bộ trong tập các bộ của quan hệ, nó sẽ được thêm vào quan hệ `relationResult` của lớp `QueryExcutor`. Đây cũng chính là tập kết quả của truy vấn.

5.10. Giao diện người dùng

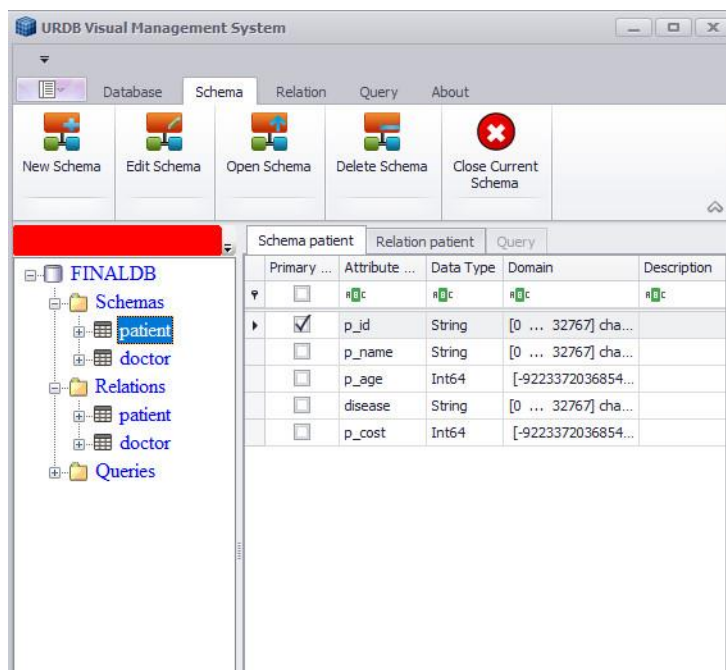
Trong phần này, chúng tôi sẽ trình bày về giao diện của hệ quản trị URDB-SQLite thông qua một ví dụ áp dụng cho các quan hệ Diagnose trên cơ sở dữ liệu khám-chữa bệnh như đã trình bày trong báo cáo.

5.10.1. Giao diện chính



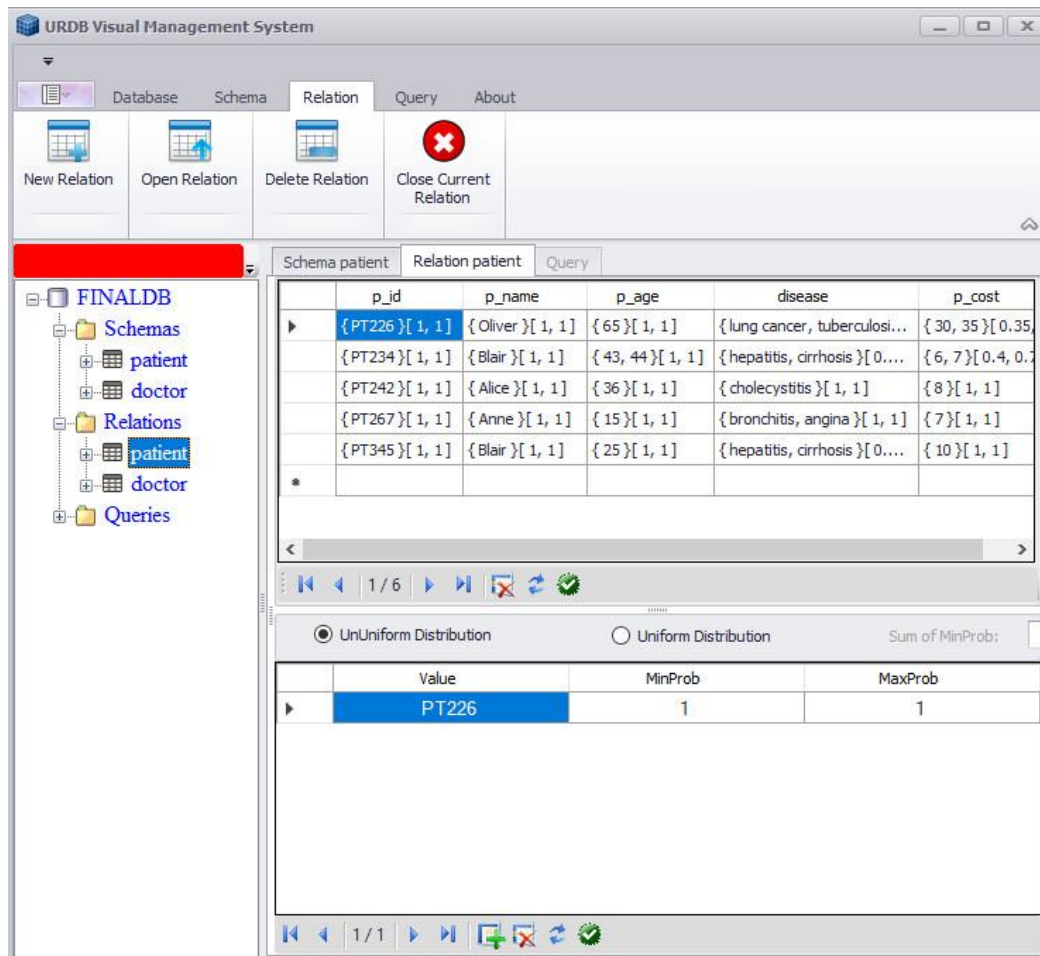
Hình 5.10.1 Giao diện chính

5.10.2. Giao diện Schema



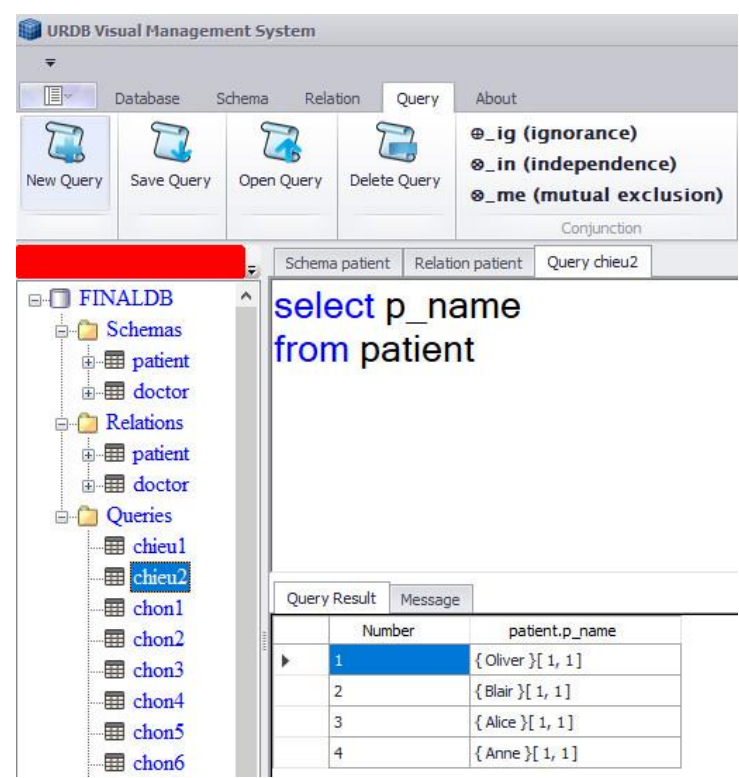
Hình 5.10.2 Giao diện Schema

5.10.3. Giao diện Relation

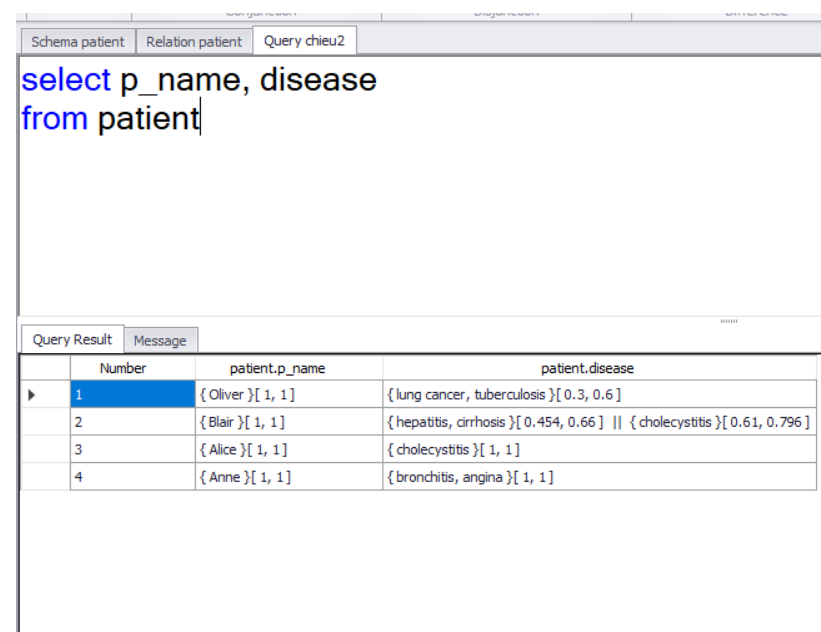


Hình 5.10.3 Giao diện Relation

5.10.4. Giao diện truy vấn

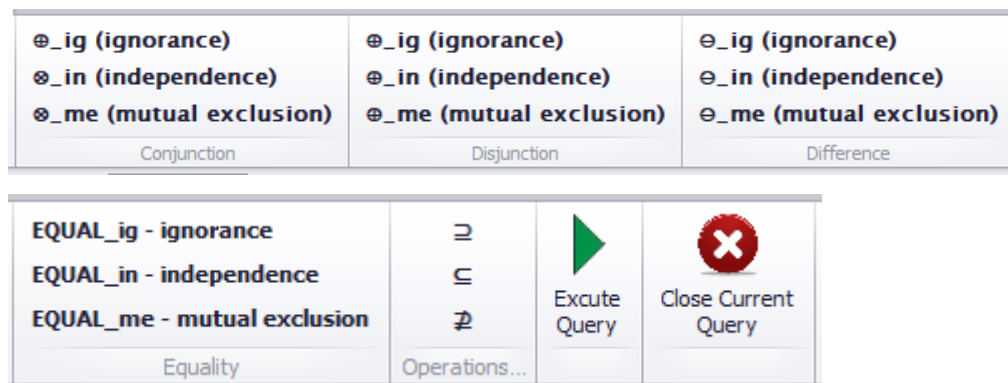


Hình 5.10.4 Giao diện truy vấn



Hình 5.10.5 Giao diện câu truy vấn và kết quả trả về

Các phím chức năng insert các chiến lược kết hợp xác suất.



Hình 5.10.6 Giao diện các phím chức năng và insert

5.10.5. About ứng dụng



Hình 5.10.7 Giao diện giới thiệu ứng dụng

5.10.6. Giao diện truy vấn phép chiếu

Schema patient

Relation patient

Query chieu2

select p_name, disease
from patient

Query Result

Message

	Number	patient.p_name	patient.disease
▶	1	{ Oliver }[1, 1]	{ lung cancer, tuberculosis }[0.3, 0.6]
	2	{ Blair }[1, 1]	{ hepatitis, cirrhosis }[0.454, 0.66] { cholecystitis }[0.61, 0.796]
	3	{ Alice }[1, 1]	{ cholecystitis }[1, 1]
	4	{ Anne }[1, 1]	{ bronchitis, angina }[1, 1]

Hình 5.10.8 Giao diện truy vấn phép chiếu

5.10.7. Giao diện truy vấn phép chọn

Schema patient

Relation patient

Query chon6

```

select *
from patient
where (patient.disease = 'cholecystitis' ⊗_in patient.p_name = 'Blair')[0.5, 0.7]

```

Query Result

Message

	Number	patient.p_id	patient.p_name	patient.p_age	patient.disease	patient.p_cost
▶	1	{ PT345 }[1, 1]	{ Blair }[1, 1]	{ 25 }[1, 1]	{ hepatitis, cirrhosis }[0.3, 0.5] { cholecystitis }[0.5, 0.7]	{ 10 }[1, 1]

Hình 5.10.9 Giao diện truy vấn phép chọn

5.10.8. Giao diện truy vấn phép kết

❖ Phép kết qua thuộc tính chung

Schema patient | Relation patient | Query ket2

```
select p.p_id, p.p_name, d.d_id, d.d_name
from patient p, doctor d
where (p.p_id EQUAL_in d.d_id)[1, 1] and (d.d_name = 'Leon')[1, 1]
```

Query Result | Message

	Number	p.p_id	p.p_name	d.d_id	d.d_name
▶	1	{PT267}[1, 1]	{Anne}[1, 1]	{PT267}[1, 1]	{Leon}[1, 1]

Hình 5.10.10 Giao diện truy vấn phép kết

❖ Phép kết tự nhiên

Ví dụ: 2 bảng kết với nhau qua thuộc tính Project theo chiến lược tuyển độc lập.

Schema patient | Relation patient | Query ket3

select *
from patient natural join in doctor

Query Result | Message

	patient.p_name	patient.p_age	patient.disease	patient.p_cost	doctor.d_id	doctor.d_name
▶	{ Oliver }[1, 1]	{ 65 }[1, 1]	{ lung cancer, tuberculosis }[0.3, 0.6]	{ 30, 35 }[0.35, 0.65]	{ D165 }[1, 1]	{ Oliver }[1, 1]
	{ Blair }[1, 1]	{ 43, 44 }[1, 1]	{ hepatitis, cirrhosis }[0.22, 0.32]	{ 6, 7 }[0.4, 0.7]	{ D123 }[1, 1]	{ Andrew }[1, 1]
	{ Blair }[1, 1]	{ 43, 44 }[1, 1]	{ hepatitis, cirrhosis }[0.11, 0.32]	{ 6, 7 }[0.4, 0.7]	{ D152 }[1, 1]	{ Louis }[1, 1]
	{ Blair }[1, 1]	{ 43, 44 }[1, 1]	{ hepatitis, cirrhosis }[0.11, 0.224]	{ 6, 7 }[0.4, 0.7]	{ PT345 }[1, 1]	{ Alice }[1, 1]
	{ Blair }[1, 1]	{ 25 }[1, 1]	{ hepatitis, cirrhosis }[0.3, 0.5]	{ 10 }[1, 1]	{ D123 }[1, 1]	{ Andrew }[1, 1]
	{ Blair }[1, 1]	{ 25 }[1, 1]	{ hepatitis, cirrhosis }[0.15, 0.5]	{ 10 }[1, 1]	{ D152 }[1, 1]	{ Louis }[1, 1]
	{ Blair }[1, 1]	{ 25 }[1, 1]	{ hepatitis, cirrhosis }[0.15, 0.35]	{ 10 }[1, 1]	{ PT345 }[1, 1]	{ Alice }[1, 1]

Hình 5.10.11 Giao diện truy vấn phép kết tự nhiên

5.10.9. Giao diện phép Tích Descartes

Schema patient

Relation patient

Query tích2

```

select p.p_id, p.p_name, p.p_age, d.d_id, d.d_name
from patient p, doctor d
where (p_age > 20 ⊗ _in d_name = 'Oliver')[0.2, 1]

```

Query Result

Message

	p.p_id	p.p_name	p.p_age	d.d_id	d.d_name
▶	{ PT226 }[1, 1]	{ Oliver }[1, 1]	{ 65 }[1, 1]	{ D165 }[1, 1]	{ Oliver }[1, 1]
	{ PT234 }[1, 1]	{ Blair }[1, 1]	{ 43, 44 }[1, 1]	{ D165 }[1, 1]	{ Oliver }[1, 1]
	{ PT242 }[1, 1]	{ Alice }[1, 1]	{ 36 }[1, 1]	{ D165 }[1, 1]	{ Oliver }[1, 1]
	{ PT345 }[1, 1]	{ Blair }[1, 1]	{ 25 }[1, 1]	{ D165 }[1, 1]	{ Oliver }[1, 1]

Hình 5.10.12 Giao diện truy vấn phép tích Descartes

5.10.10. Giao diện truy vấn phép giao

Schema patient	Relation patient	Query giao1
<pre>select p_name from patient intersect ⊗ _ig select d_name from doctor</pre>		
Query Result	Message	
Number	doctor.d_name	
1	{ Oliver }[1, 1]	
2	{ Alice }[1, 1]	

Hình 5.10.13 Giao diện truy vấn phép giao

5.10.11. Giao diện truy vấn phép hợp

Schema patient	Relation patient	Query hop1																														
<pre>select p_name from patient union all select d_name from doctor</pre>																																
Query Result	Message																															
<table><thead><tr><th></th><th>Number</th><th>doctor.d_name</th></tr></thead><tbody><tr><td>▶</td><td>1</td><td>{ Oliver }[1, 1]</td></tr><tr><td></td><td>2</td><td>{ Blair }[1, 1]</td></tr><tr><td></td><td>3</td><td>{ Alice }[1, 1]</td></tr><tr><td></td><td>4</td><td>{ Anne }[1, 1]</td></tr><tr><td></td><td>5</td><td>{ Andrew }[1, 1]</td></tr><tr><td></td><td>6</td><td>{ Leon }[1, 1]</td></tr><tr><td></td><td>7</td><td>{ Louis }[1, 1]</td></tr><tr><td></td><td>8</td><td>{ Oliver }[1, 1]</td></tr><tr><td></td><td>9</td><td>{ Alice }[1, 1]</td></tr></tbody></table>				Number	doctor.d_name	▶	1	{ Oliver }[1, 1]		2	{ Blair }[1, 1]		3	{ Alice }[1, 1]		4	{ Anne }[1, 1]		5	{ Andrew }[1, 1]		6	{ Leon }[1, 1]		7	{ Louis }[1, 1]		8	{ Oliver }[1, 1]		9	{ Alice }[1, 1]
	Number	doctor.d_name																														
▶	1	{ Oliver }[1, 1]																														
	2	{ Blair }[1, 1]																														
	3	{ Alice }[1, 1]																														
	4	{ Anne }[1, 1]																														
	5	{ Andrew }[1, 1]																														
	6	{ Leon }[1, 1]																														
	7	{ Louis }[1, 1]																														
	8	{ Oliver }[1, 1]																														
	9	{ Alice }[1, 1]																														

Hình 5.10.14 Giao diện truy vấn phép hợp

Schema patient	Relation patient	Query hop3																								
<pre>select p_name from patient union ⊕_in select d_name from doctor</pre>																										
Query Result	Message																									
<table><thead><tr><th></th><th>Number</th><th>doctor.d_name</th></tr></thead><tbody><tr><td>▶</td><td>1</td><td>{ Oliver }[1, 1]</td></tr><tr><td></td><td>2</td><td>{ Blair }[1, 1]</td></tr><tr><td></td><td>3</td><td>{ Alice }[1, 1]</td></tr><tr><td></td><td>4</td><td>{ Anne }[1, 1]</td></tr><tr><td></td><td>5</td><td>{ Andrew }[1, 1]</td></tr><tr><td></td><td>6</td><td>{ Leon }[1, 1]</td></tr><tr><td></td><td>7</td><td>{ Louis }[1, 1]</td></tr></tbody></table>				Number	doctor.d_name	▶	1	{ Oliver }[1, 1]		2	{ Blair }[1, 1]		3	{ Alice }[1, 1]		4	{ Anne }[1, 1]		5	{ Andrew }[1, 1]		6	{ Leon }[1, 1]		7	{ Louis }[1, 1]
	Number	doctor.d_name																								
▶	1	{ Oliver }[1, 1]																								
	2	{ Blair }[1, 1]																								
	3	{ Alice }[1, 1]																								
	4	{ Anne }[1, 1]																								
	5	{ Andrew }[1, 1]																								
	6	{ Leon }[1, 1]																								
	7	{ Louis }[1, 1]																								

Hình 5.10.15 Giao diện truy vấn phép hợp theo chiến lược tuyển độc lập

5.10.12. Giao diện truy vấn phép trừ

Schema patient Relation patient Query tru1		
<pre>select p_name from patient except select d_name from doctor</pre>		
Query Result Message		
	Number	doctor.d_name
▶	1	{ Blair }[1, 1]
	2	{ Anne }[1, 1]

Hình 5.10.16 Giao diện truy vấn phép trừ

TỔNG KẾT VÀ ĐỀ NGHỊ

1. Tổng kết

Như đã trình bày trong phần mở đầu, thông tin về các đối tượng trong thế giới thực thường là không chắc chắn, không đầy đủ và thiếu chính xác. Tuy nhiên, do hạn chế về cơ sở toán học, các mô hình CSDL truyền thống nói chung và CSDL quan hệ truyền thống nói riêng hầu như không thể biểu diễn, thao tác và xử lý được thông tin không chắc chắn và không đầy đủ. Đối với mô hình CSDL quan hệ, mặc dù giá trị NULL đã được sử dụng như một giải pháp, nhưng khả năng đáp ứng của mô hình này để biểu diễn thông tin không chắc chắn trong thực tế là rất hạn chế. Hệ quả là lý thuyết xác suất đã được ứng dụng để xây dựng các mô hình CSDL (quan hệ) xác suất nhằm đáp ứng nhu cầu giải quyết các bài toán trong thế giới thực.

Nhiều mô hình CSDL quan hệ xác suất đã được đề nghị. Các mô hình này đã sử dụng các cách thức và phương pháp vận dụng lý thuyết xác suất khác nhau để nâng cao khả năng mô hình hóa và xử lý thông tin dữ liệu. Tuy nhiên, không có mô hình nào có khả năng mô hình hóa mọi khía cạnh của thông tin không chắc chắn trong thực tế. Vì vậy, các mô hình CSDL xác suất vẫn được tiếp tục nghiên cứu và phát triển. Mô hình URDB được trình bày trong đề tài khóa luận này là một đóng góp mới bằng cách tích hợp bộ ba xác suất mở rộng (vào mô hình CSDL truyền thống) như là một cách biểu diễn mới của giá trị thuộc tính quan hệ, cho phép thuộc tính đa trị không chắc chắn trong mô hình URDB. Từ đó, hệ thống các phép toán đại số quan hệ xác suất được xây dựng như là một ngôn ngữ truy vấn dữ liệu tương ứng. Một tập các tính chất của các phép toán đại số quan hệ xác suất được đề nghị và được chứng minh chặt chẽ chứng tỏ mô hình được xây dựng là đúng đắn. Như đã trình bày trong phần cuối của Chương 4, các phép toán đại số là hiệu quả (với độ phức tạp đa thức). Mô hình được xây dựng là một đóng góp cho quá trình nghiên cứu và phát triển các hệ thống CSDL nói chung và CSDL xác suất nói riêng. Quá trình phát triển URDB có thể được tóm lược như sau:

1. Đầu tiên, các khái niệm thuộc tính không chắc chắn, kiểu, giá trị và giá trị bộ xác suất được đề nghị dựa trên khái niệm bộ ba xác suất mở rộng.
2. Sau đó, khái quát mở rộng các định nghĩa lược đồ và quan hệ CSDL truyền thống thành lược đồ và quan hệ xác suất dựa trên thuộc tính đa trị không chắc chắn và giá trị bộ xác suất.
3. Kế đến, các phép toán đại số trên URDB được xây dựng bằng cách mở rộng một cách logic và nhất quán các phép toán đại số trên CSDL quan hệ truyền thống dựa trên cơ sở toán học trong Chương 2.
4. Cuối cùng, các tính chất của các phép toán đại số trên URDB đã được chứng minh cho thấy quá trình mở rộng CSDL quan hệ thành URDB là đúng đắn.

Một cơ sở dữ liệu các bệnh nhân tại phòng khám của một bệnh viện được dùng làm ví dụ minh họa cho tiến trình mở rộng và phát triển các khái niệm lý thuyết nền tảng mô hình và các phép toán đại số quan hệ xác suất cho thấy rõ hơn bản chất và cách thức ứng dụng của URDB.

2. Đề nghị

Từ các nghiên cứu liên quan đã được đề cập và từ các kết quả của đề tài này, chúng tôi đề nghị một số vấn đề và hướng nghiên cứu tiếp theo như sau:

1. *URDB chưa được hiện thực như một hệ quản trị CSDL* để cho phép tiến tới ứng dụng mô hình vào thực tế. Do đó đề nghị đầu tiên là xây dựng một hệ quản trị CSDL cho URDB với ngôn ngữ truy vấn thân thiện tựa SQL làm cơ sở cho các ứng dụng thao tác và xử lý thông tin không chắc chắn và không chính xác trong thực tế.
2. Phát triển một tập các hàm kết gộp xác suất (probabilistic aggregate function) như min, max, average (trung bình) v.v. trên các thuộc tính quan hệ xác suất để hỗ trợ xây dựng các phép toán kết gộp và gom nhóm (aggregate and grouping operation) trên URDB cho các ứng dụng tính toán trong CSDL.
3. Như trong CSDL truyền thống, tối ưu hóa truy vấn cũng là một bài toán cần nghiên cứu trong URDB.

TÀI LIỆU THAM KHẢO

- [1] E.F. Codd, A relational model of data for large shared data banks, *Communications of the ACM*, vol.13, no.6, pp.377-387, 1970.
- [2] C.J. Date, An introduction to database systems, *Addison-Wesley, Eighth Edition*, 2004.
- [3] R. Elmasri and S.B. Navathe, Fundamentals of Database Systems, *6th edition, Addison-Wesley*, 2011.
- [4] T. Imielinski, W. J.R. Lipski, Incomplete information in relational databases, *Journal of the Association for Computing Machinery*, vol.31, no.4, pp.761-791, 1984.
- [5] A. Ali, S. Talpur and S. Narejo, Detecting faulty sensors by analyzing the uncertain data using probabilistic database, *Proc. of 3rd International Conference on Computing, Mathematics and Engineering Technologies*, Sukkur, Pakistan, January 29-30, pp.143-150, 2020.
- [6] I.I. Ceylan, S. Borgwardt and T. Lukasiewicz, Most probable explanations for probabilistic database queries, *Proc. of 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia, August 19-25, pp.950-956, 2017.
- [7] I.I. Ceylan, A. Darwiche and G.V.D Broeck, Open-world probabilistic databases: Semantics, algorithms, complexity, *Journal of Artificial Intelligence*, vol.295, no.11, pp.103474-103513, 2021.
- [8] D. Dey and S. Sarkar, A probabilistic relational model and algebra, *ACM Transactions on Database Systems*, vol.21, no.3, pp.339-369, 1996.
- [9] N. Fuhr and T. Rolleke, A probabilistic relational algebra for the integration of information retrieval and database systems, *ACM Transactions on Information Systems*, vol.15, no.1, pp.32-66, 1997.
- [10] Y. Li, J. Chen and L. Feng, Dealing with uncertainty: a survey of theories and practices, *IEEE Transactions on Knowledge and Data Engineering*, vol.25, no.11, pp.2463-2482, 2013.

- [11] S. Zhang and C. Zhang, A probabilistic data model and its semantics, *Journal of Research and Practice in Information Technology*, vol.35, no.4, pp.237-256, 2003.
- [12] A. Dekhtyar, R. Ross and V.S. Subrahmanian, Probabilistic temporal databases, I: algebra, *ACM Transactions on Database Systems*, vol.26, no.1, pp.41-95, 2001.
- [13] W. Zhao, A. Dekhtyar and J. Goldsmith, Databases for interval probabilities, *International Journal of Intelligent Systems*, vol.19, no.9, pp.789-815, 2004.
- [14] L.V.S. Lakshmanan, N. Leone, R. Ross and V.S. Subrahmanian, Probview: A flexible probabilistic database system, *ACM Transactions on Database Systems*, vol.22, no.3, pp.419-469, 1997.
- [15] R. Ross and V.S. Subrahmanian, Aggregate operators in probabilistic databases, *Journal of the ACM*, vol.52, no.1, pp.54-101, 2005.
- [16] D. Dey and S. Sarkar, Generalized normal forms for probabilistic relational data, *IEEE Transactions on Knowledge and Data Engineering*, vol.14, no.3, pp.485-497, 1992.
- [17] D. Barbara, H. Garcia-Molina and D. Porter, The management of probabilistic data, *IEEE Transactions on Knowledge and Data Engineering*, vol.4, no.5, pp.487-502, 1992.
- [18] T. Eiter, T. Lukasiewicz and M. Walter, A data model and algebra for probabilistic complex values, *Annals of Mathematics and Artificial Intelligence*, vol.33, pp.205-252, 2001.
- [19] T. Eiter, J.J. Lu, T. Lukasiewicz and V.S. Subrahmanian, Probabilistic object bases, *ACM Transactions on Database Systems*, vol.26, no.3, pp.264-312, 2001.
- [20] Y. Kornatzky and S.E. Shimony, A probabilistic object-oriented data model, *Data and Knowledge Engineering*, vol.12, pp.143-166, 1994.
- [21] S.K. Lee, An extended relational database model for uncertain and imprecise information, *Proc. of 18th Conference on Very Large Data Bases*, Vancouver, British Columbia, Canada, August 23-27, pp.211-220, 1992.
- [22] H. Nguyen and D.H. Tran, A probabilistic relational data model for uncertain information, *Proc. of 3rd IEEE International Conference on Information Science and Technology*, Yangzhou, China, March 23-25, pp.607-613, 2013.

- [23] H. Nguyen, A probabilistic relational database model and algebra, *Journal of Computer Science and Cybernetics*, vol. 31, no.4, pp.305-321, 2015.
- [24] H. Nguyen, Extending relational database model for uncertain information, *Journal of Computer Science and Cybernetics*, vol.35, no.4, pp.355-372, 2019.
- [25] A.V. Vitianingsih, I. Wisnubhadra, S.S.K. Baharin, R. Marco and A.L. Maukar, Classification of pertussis vulnerable area with location analytics using multiple attribute decision making, *International Journal of Innovative Computing, Information and Control*, vol.16, no.6, pp.1943-1957, 2020.
- [26] T. Friedman, G. Broeck, Symbolic querying of vector spaces: probabilistic databases meets relational embeddings, *Proc. of 36th Conference on Uncertainty in Artificial Intelligence*, Toronto, Canada, August 3-6, vol.124, pp.1268-1277, 2020.
- [27] J. Bernad, C. Bobed, E. Mena, Uncertain probabilistic range queries on multidimensional data, *Information Sciences*, vol. 537, pp.334-367, 2020.
- [28] H. Nguyen, N.T. Nguyen and N.T.T. Tran, “A probabilistic relational database model with uncertain multivalued attributes”, *International Journal of Innovative Computing, Information and Control (ICIC Express Letters)*, Accepted, 2021.