ỦY BAN NHÂN DÂN TP HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**A picture containing logo

Description automatically generated**

**NGUYỄN ĐÀO LINH ĐAN - 3121410137**

**TÊN ĐỀ TÀI**

**MỘT HỆ QUẢN TRỊ CSDL QUAN HỆ CHO CÁC THUỘC TÍNH CÓ GIÁ TRỊ KHOẢNG XÁC SUẤT**

**KHÓA LUẬN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**tRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC**

**GV HƯỚNG DẪN: TS. Đỗ Như Tài**

**TP. HỒ CHÍ MINH, THÁNG 5 NĂM 2025**

ỦY BAN NHÂN DÂN TP HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**A picture containing logo

Description automatically generated**

**NGUYỄN ĐÀO LINH ĐAN - 3121410137**

**TÊN ĐỀ TÀI**

**MỘT HỆ QUẢN TRỊ CSDL QUAN HỆ CHO CÁC THUỘC TÍNH CÓ GIÁ TRỊ KHOẢNG XÁC SUẤT**

**KHÓA LUẬN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**tRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC**

**GV HƯỚNG DẪN: TS. Đỗ Như Tài**

**TP. HỒ CHÍ MINH, THÁNG 5 NĂM 2025**

# 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 10 tháng 5 năm 2025

Tác giả luận văn

**Nguyễn Đào Linh Đan**

# LỜI CẢM ƠN

Trước tiên, chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Nguyễn Hòa, người đã tận tình hướng dẫn, truyền đạt kiến thức và kinh nghiệm quý báu, đồng hành cùng chúng em trong suốt quá trình thực hiện khóa luận tốt nghiệp. Sự chỉ bảo và định hướng của thầy là nguồn động lực lớn lao để chúng em hoàn thành tốt bài khóa luận này.

Chúng em cũng xin bày tỏ lòng biết ơn đến quý Thầy/Cô trong Khoa Công nghệ thông tin đã truyền đạt những kiến thức nền tảng, tạo điều kiện thuận lợi cho chúng em trong quá trình học tập và nghiên cứu tại trường.

Bên cạnh đó, chúng em xin chân thành cảm ơn gia đình yêu thương - những người luôn là chỗ dựa vững chắc về tinh thần và vật chất, động viên chúng em không ngừng cố gắng để vượt qua mọi khó khăn trong suốt chặng đường học tập.

Cuối cùng, chúng em xin gửi lời cảm ơn đến các anh/chị, bạn bè và đồng nghiệp, những người đã luôn bên cạnh, hỗ trợ, chia sẻ kinh nghiệm và cùng chúng em vượt qua những thử thách trong suốt quá trình thực hiện khóa luận này.

Dù đã nỗ lực hết mình, nhưng bài khóa luận của chúng em chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự góp ý quý báu từ Thầy/Cô và mọi người để hoàn thiện hơn nữa.

Trân trọng!

Thành phố Hồ Chí Minh, ngày 10 tháng 5 năm 2025

Sinh viên

**Nguyễn Đào Linh Đan**

# 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 các hệ quản trị CSDL quan hệ truyền thống do Codd đề nghị năm 1970 thành một *hệ quản trị CSDL quan hệ mới* của mô hình CSDL quan hệ cho các thuộc tính có giá trị khoảng xác suất(Relational Database model for Probability Interval Valued Attributes-IPRDB) để 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 của các bộ dữ liệu là quan hệ xác suất với giá trị thuộc tính không chắc chắn được biểu diễn bởi một tập giá trị kết hợp với các khoảng xác suất; (2) Hệ quản trị CSDL với ngôn ngữ truy vấn tựa SQL truyền thống có thể xử lý và thao tác thông tin không chắc chắn trong thực tế được biểu diễn bởi các quan hệ xác suất.

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 IPRDB. 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 IPRDB được hiện thực với ngôn ngữ truy vấn tựa SQL dựa trên hệ quản trị mã nguồn mở SQLite, gọi là IPRDB -SQLite, cho thấy triển vọng và khả năng ứng dụng của IPRDB để mô hình hóa và thao tác các quan hệ xác suất trong thực tế.

MỤC LỤC

[LỜI CAM ĐOAN i](#_Toc187405426)

[LỜI CẢM ƠN ii](#_Toc187405427)

[TÓM TẮT iii](#_Toc187405428)

[MỤC LỤC 1](#_Toc187405429)

[DANH MỤC CÁC BẢNG 3](#_Toc187405430)

[DANH MỤC CÁC HÌNH 4](#_Toc187405431)

[DANH SÁCH CÁC CỤM TỪ VIẾT TẮT 5](#_Toc187405432)

[MỞ ĐẦU 6](#_Toc187405433)

[Chương 1 12](#_Toc187405434)

[TỔNG QUAN VỀ MÔ HÌNH CSDL QUAN HỆ 12](#_Toc187405435)

[1.1. Giới thiệu 12](#_Toc187405436)

[1.2. Mô hình dữ liệu 12](#_Toc187405437)

[1.3. Các phép toán đại số 14](#_Toc187405438)

[1.4. Tính chất các phép toán đại số 16](#_Toc187405439)

[1.5. Kết luận 17](#_Toc187405440)

[Chương 2 19](#_Toc187405441)

[CƠ SỞ TOÁN HỌC CỦA MÔ HÌNH IPRDB 19](#_Toc187405442)

[2.1. Giới thiệu 19](#_Toc187405443)

[2.2. Các chiến lược kết hợp các khoảng xác suất 20](#_Toc187405444)

[2.3. Giá trị xác suất 23](#_Toc187405445)

[2.4. Các chiến lược kết hợp các giá trị xác suất 24](#_Toc187405446)

[2.5. Diễn dịch xác suất của quan hệ trên các tập hợp 25](#_Toc187405447)

[2.6. Kết luận 25](#_Toc187405448)

[Chương 3 27](#_Toc187405449)

[LƯỢC ĐỒ VÀ QUAN HỆ CỦA MÔ HÌNH IPRDB 27](#_Toc187405450)

[3.1. Giới thiệu 27](#_Toc187405451)

[3.2. Mô hình ý niệm 27](#_Toc187405452)

[3.3. Thuộc tính quan hệ 29](#_Toc187405453)

[3.4. Kiểu và giá trị 30](#_Toc187405454)

[3.5. Lược đồ và quan hệ 32](#_Toc187405455)

[3.6. Kết luận 35](#_Toc187405456)

[Chương 4 36](#_Toc187405457)

[CÁC PHÉP TOÁN ĐẠI SỐ TRÊN IPRDB 36](#_Toc187405458)

[4.1. Giới thiệu 36](#_Toc187405459)

[4.2. Phép chọn 36](#_Toc187405460)

[4.3. Phép chiếu 41](#_Toc187405461)

[4.4. Phép tích Descartes 42](#_Toc187405462)

[4.5. Phép kết tự nhiên 43](#_Toc187405463)

[4.6. Phép giao, hợp và trừ 44](#_Toc187405464)

[4.7. Tính chất của các phép toán đại số 47](#_Toc187405465)

[4.8. Kết luận 51](#_Toc187405466)

[Chương 5 52](#_Toc187405467)

[HIỆN THỰC HỆ QUẢN TRỊ CỦA MÔ HÌNH IPRDB 52](#_Toc187405468)

[5.1. Giới thiệu 52](#_Toc187405469)

[5.2. Các tính năng đặc trưng của SQLite 53](#_Toc187405470)

[5.2.1. Tổng quan 53](#_Toc187405471)

[5.2.2. Các tính năng đặc trưng của SQLite 53](#_Toc187405472)

[5.2.3. Các đối tượng và phương thức chính trong SQLite.Net 54](#_Toc187405473)

[5.3. Thiết kế tổng quan hệ quản trị IPRDB -SQLite 55](#_Toc187405474)

[5.4. Kiến trúc tổng quan của hệ quản trị IPRDB -SQLite 57](#_Toc187405475)

[5.5. Hiện thực khối biểu diễn mô hình IPRDB 59](#_Toc187405476)

[5.5.1. Biểu diễn các kiểu trong IPRDB 60](#_Toc187405477)

[5.5.2. Biểu diễn mô hình IPRDB 61](#_Toc187405478)

[5.5.2.1. Lớp ProbSchema 61](#_Toc187405479)

[5.6. Hiện thực khối xử lý 66](#_Toc187405480)

[5.6.1. Xử lý truy vấn chọn 66](#_Toc187405481)

[5.6.2. Xử lý điều kiện chọn 67](#_Toc187405482)

[5.7. Giao diện người dùng 69](#_Toc187405483)

[5.7.1. Giao diện chính 70](#_Toc187405484)

[5.7.2. Giao diện lược đồ quan hệ 71](#_Toc187405485)

[5.7.3. Giao diện tạo và nhập dữ liệu cho quan hệ 73](#_Toc187405486)

[5.7.4. Truy vấn dữ liệu trên quan hệ 74](#_Toc187405487)

[5.8. Kết luận 76](#_Toc187405488)

[TỔNG KẾT VÀ ĐỀ NGHỊ 77](#_Toc187405489)

[1. Tổng kết 77](#_Toc187405490)

[2. Đề nghị 78](#_Toc187405491)

[TÀI LIỆU THAM KHẢO 80](#_Toc187405492)

# DANH MỤC CÁC BẢNG

[Bảng 2.2.1 Các tiên đề về chiến lược hội 21](#_Toc185883143)

[Bảng 2.2.2 Các tiên đề về chiến lược tuyển 22](#_Toc185883144)

[Bảng 2.2.3 Các ví dụ về các chiến lược kết hợp xác suất 2](#_Toc185883145)3

[Bảng 3.5.1 Quan hệ PATIENT 34](#_Toc185883143)

[Bảng 4.2.1 Quan hệ σφ(PATIENT) 41](#_Toc185883144)

[Bảng 4.3.1 Quan hệ ∏L⊕in(PATIENT) 4](#_Toc185883145)2

[Bảng 4.5.1 Quan hệPATIENT1 44](#_Toc185883143)

Bảng 4.5.2 Quan hệ PATIENT2 44

[Bảng 4.5.3 Quan hệ PATIENT1 ⋈⊗in PATIENT2 4](#_Toc185883145)4

[Bảng 4.6.1 Relation DIAGNOSE1 46](#_Toc185883143)

[Bảng 4.6.2 Relation DIAGNOSE2 46](#_Toc185883144)

[Bảng 4.6.3 DIAGNOSE1∪⊕in DIAGNOSE2 46](#_Toc185883145)

# 

# DANH MỤC CÁC HÌNH

[Hình 3.2.1 Cơ sở dữ liệu khám-chữa bệnh 28](#_Toc185884403)

[Hình 3.2.2 Kiến trúc của hệ thống IPRDB 29](#_Toc185884404)

[Hình 5.4.1 Kiến trúc hệ quản trị IPRDB 59](#_Toc185884412)

[Hình 5.7.1.1 Giao diện chính của hệ thống IPRDB 70](#_Toc185884421)

[Hình 5.7.1.2 Giao diện chương trình sau khi tạo một cơ sở dữ liệu mới 70](#_Toc185884422)

[Hình 5.7.2.1 Giao diện tạo lược đồ quan hệ 71](#_Toc185884434)

[Hình 5.7.2.2 Màn hình chọn kiểu dữ liệu cho thuộc tính 72](#_Toc185884435)

[Hình 5.7.2.3 Màn hình nhập danh sách giá trị cho kiểu dữ liệu UserDefined 73](#_Toc185884436)

[Hình 5.7.3.1 Màn hình tạo mới một quan hệ 73](#_Toc185884440)

[Hình 5.7.3.2 Màn hình nhập liệu cho quan hệ 74](#_Toc185884441)

[Hình 5.7.4.1 Màn hình tạo mới query 74](#_Toc185884445)

[Hình 5.7.4.2 Màn hình chọn và chèn chiến lược vào câu truy vấn 75](#_Toc185884446)

[Hình 5.7.4.3 Giao diện nhập nội dung câu truy vấn 75](#_Toc185884447)

[Hình 5.7.4.4 Màn hình hiển thị kết quả thực thi câu truy vấn 75](#_Toc185884447)

# DANH SÁCH CÁC CỤM TỪ VIẾT TẮT

CSDL : Cơ Sở Dữ Liệu

URDB : Relational DataBase with uncertain multivalued attributes

⊆ : quan hệ tập con

⊄ : quan hệ không là tập con

∈ : quan hệ liên thuộc

θ : Quan hệ hai ngôi

⋈ : phép kết

∏ : Phép chiếu

∩ : phép toán giao tập hợp

∪ : phép toán hợp tập hợp

≤ : quan hệ nhỏ hơn hoặc bằng trên tập các số thực/khoảng

≥ : quan hệ lớn hơn hoặc bằng trên tập các số thực/khoảng

⊗ : phép toán hội xác suất của hai khoảng ứng với hai biến cố

⊕ : phép toán tuyển xác suất của hai khoảng ứng với hai biến cố

⊖ : 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

*probR,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]). 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 [3]. Điều này là không hoàn toàn phù hợp với thực tế, như đã được chỉ ra trong ([5], [9], [16], [23]), 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 2024-2025”; hoặc “tìm các đội bóng ngoại hạng Anh mà có 70-90% khả năng vô địch mùa giải 2024-2025”; 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 truyền thống bằng một quan hệ xác suất như trong ([5-16]). 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 thuộc tính 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 ([17-18]), 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 [19]. 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 ([20-22]) là những mở rộng của mô hình trong [19] 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. Tuy nhiên, khi quan hệ xác suất, được biểu diễn bởi các mô hình trong ([20-22]), có nhiều thuộc tính, số các hàm phân bố được tạo ra đủ lớn làm giảm hiệu suất tính toán và truy vấn thông tin trên các hệ thống dữ liệu của các mô hình này.

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. Mô hình trong [23] được đề nghị và xây dựng nhằm khắc phục hạn chế của các mô hình [20-22] bằng cách biểu diễn giá trị thuộc tính quan hệ như một phân bố các khoảng xác suất trên một tập giá trị, giúp truy vấn và thao tác dữ liệu hiệu quả hơn. 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 mới* của mô hình CSDL quan hệ cho các thuộc tính có giá trị khoảng xác suất(Relational Database model for Probability Interval Valued Attributes-IPRDB) đã được xây dựng trong [23] để biểu diễn và xử lý thông tin không chắc chắn rất phổ biến trong thực tế.

Để xây dựng mô hình IPRDB, diễn dịch xác suất của các quan hệ hai ngôi trên các tập hợp được sử dụng như một độ đo mức độ không chắc chắn của các giá trị thuộc tính quan hệ, các giá trị xác suất được đề nghị để biểu diễn các thuộc tính có giá trị khoảng xác suất. Các khái niệm trong mô hình CSDL truyền thống như kiểu, giá trị, lược đồ quan hệ, phụ thuộc hàm, khóa được mở rộng thành các khái niệm tương ứng trong IPRDB. Các phép toán đại số trong CSDL truyền thống cũng được mở rộng thành các phép toán đại số tương ứng trong IPRDB để truy vấn các quan hệ với thuộc tính đa trị không chắc chắn.

Mô hình IPRDB được xây dựng và mở rộng như vậy vẫn đảm bảo tính nhất quán với mô hình CSDL truyền thống. Nói một cách khác, mô hình IPRDB 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 IPRDB. Một tập các tính chất của các phép toán đại số trên IPRDB 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 IPRDB được xây dựng là đúng đắn.

Hệ quản trị cho IPRDB 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 IPRDB-SQLite, bước đầu cho thấy triển vọng ứng dụng của IPRDB 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 IPRDB và cách thức ứng dụng của nó.

1. **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 IPRDB (lược đồ, quan hệ, các bộ tương đương giá trị, phụ thuộc hàm, khóa 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 IPRDB.
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 IPRDB.
3. **Cấu trúc khóa luận**

Báo cáo đề tài gồm 5 chương và một phần mở đầu và một phần tổng kế. 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 IPRDB. Từ mô hình CSDL truyền thống chúng ta thấy được mối liên hệ của những yếu tố mở rộng trong IPRDB.

Chương 2 trình bày cơ sở toán học để phát triển mô hình IPRDB. 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 IPRDB.

Chương 3 trình bày mô hình ý niệm của IPRDB, 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 IPRDB. Đó là các phép toán như chọn, chiếu, tích Descartes, kết, giao, hợp và trừ trên các quan hệ xác suất trong IPRDB. 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 IPRDB-SQLite với truy vấn chọn trên các quan hệ xác suất của IPRDB.

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

## 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], [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], [18], [23]). 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.

## 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*,…, Ak* là các thuộc tính đôi một khác nhau trong ***A***và *τ*1, *τ*2,…, *τk*là các kiểu cơ sở thì *τ =* [*A*1: *τ*1, *A*2: *τ*2,…,  *Ak*: *τk*] là một kiểu, được gọi là *kiểu bộ* (tuple type) trên tập các thuộc tính {*A*1, *A*2, …, *Ak*}*.*Với một kiểu *τ =* [*A*1: *τ*1, *A*2: *τ*2 ,…, *Ak*: *τk*]*,* chúng tôi sử dụng *τ.Ai* để biểu thị *τi*.

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, …, *vk*) thực chất là một phần tử của tích Descartes *τ = τ*1× *τ*2 ×…× *τk* trên các miền giá trị tương ứng của các thuộc tính *A*1, *A*2,…, *Ak* của quan hệ. Và như vậy, một tập các bộ (*v*1*, v*2, …, *vk*) là một tập con của tích *τ = τ*1× *τ*2×… × *τk* và đó là một quan hệ *k*- ngôi trên tập các giá trị các thuộc tính *A*1, *A*2,…, *Ak.*

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

1. Với mọi kiểu cơ sở *τ* ∈ ***T*** , thì mỗi *v* ∈ *dom*(*τ*) là một giá trị kiểu *τ*.
2. Nếu *A*1,…, *Ak* là các thuộc tính đôi một khác nhau trong ***A*** và *v*1,…, *vk* là các giá trị tương ứng của các kiểu *τ*1,…, *τk* thì [*A*1: *v1*,…, *Ak*: *vk*] là một giá trị kiểu [*A*1: *τ*1,…, *Ak*: *τk*], 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, …, *Ak*}.

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, …, *vk*).

**Định nghĩa 1.2.3** Giả sử ***A*** là một tập các thuộc tính, một tập *R* = {*A*1, *A*2,…, *Ak*} các thuộc tính đôi một khác nhau trong ***A*** được gọi là một *lược đồ quan hệ* (relational schema) trên *A*1, *A*2, …, *Ak*, ký hiệu là *R*(*A*1, *A*2, …, *Ak*).

**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, …, *Ak*) là một tập hữu hạn các bộ {*t*1, *t*2,…, *tn*} trên tập các thuộc tính {*A*1, *A*2, …, *Ak*}. 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***{*A*1, *A*2, …, *Ak*}.

**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, …, *Ak*) là một lược đồ quan hệ trên tập thuộc tính {*A*1, *A*2, …, *Ak*}. Một tập thuộc tính *K* ⊆{*A*1, *A*2, …, *Ak*} được gọi là *khóa* của *R* nếu với mọi quan hệ *r* trên *R* và hai bộ bất kỳ *t*1và *t2* của *r* mà *t*1[*K*] = *t*2[*K*] thì *t*1 ≡ *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, …, *Ak*), *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* → *Y*, nếu:

∀ *t*1, *t*2 ∈ *r*, *t*1[*X*] = *t*2[*X*] ⇒ *t*1[*Y*] = *t*2[*Y*]

Phụ thuộc hàm *X* → *Y* còn được gọi là “*X* xác định hàm *Y*” hoặc “*Y* phụ thuộc hàm vào *X*”.

## 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]).

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

σφ(*r*) = {*t* ∈ *r* | φ(*t*) = 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 ∏***X***(*R*) là lược đồ quan hệ trên ***X***. *Phép chiếu* (projection) của *r* trên ***X***, được ký hiệu là ∏***X***(*r*), là một quan hệ *r***’** trên ∏***X***(*R*) bao gồm các bộ là thu hẹp của *r* trên ***X***. Nghĩa là *r*’ = {*t’* = *t*[***X***] | *t* ∈*r*}.

**Định nghĩa 1.3.3** Giả sử {*A*1, *A*2,…, *Am*} và {*B*1, *B*2,…, *Bn*} 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,…, *Am*) và *S*(*B*1, *B*2,…, *Bn*). Phép tích Descartes của hai quan hệ *r* và *s*, kí hiệu là *r* × *s*, là một quan hệ trên lược đồ *Q*(*A*1, *A*2,…, *Am, B*1,B2,…, *Bn*) bao gồm các bộ *t* được xác định bởi

*r* × *s* = {*t* = (*a*1,…, *am, am+1*,…, *am+n*) | (*a*1,…, *am*) ∈ *r* và (*am+1*,…, *am+n*) ∈ *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* ⋈ *s*, là một quan hệ trên lược đồ *Q*(***A***∪***B***) được xác định theo tích Descartes của *r* và *s* bao gồm các bộ *t* sao cho:

*t*[***A***] = *tr* ∈ *r*, *t*[***B***] = *ts* ∈ *s* và *tr*[*C*] = *ts*[*C*] với mọi *C* ∈ ***A*** ∩ ***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* ∪ *s*, là một quan hệ trên *R* bao gồm các bộ của *r* hay của *s*. Nghĩa là *r* ∪ *s* = {*t* | *t* ∈ *r* hay *t* ∈ *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* ∩ *s*, là một quan hệ trên *R* bao gồm các bộ thuộc đồng thời cả *r* và *s*. Nghĩa là *r* ∩ *s* ={*t* | *t* ∈ *r* và *t* ∈ *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* | *t* ∈ *r* và *t* ∉ *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].

## 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 [2].

**Định lý 1.4.1** Giả sử *r* là một quan hệ trên lược đồ *R*. Gọi φ1 và φ2 là hai điều kiện chọn. Khi đó

σφ1(σ φ2(*r*)) = σφ2(σφ1(*r*)) = σ φ1∧φ2(*r*) (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*** ⊆ ***B***. Khi đó

Π***A***(Π***B*** (*r*)) = Π***A***(*r*) (2)

**Định lý 1.4.3** Giả sử *r*1, *r*2 và *r*3tương ứng là các quan hệ trên *R1*, *R2* và *R3*, sao cho nếu chúng có thuộc tính cùng tên thì cùng kiểu giá trị. Khi đó

*r*1 ⋈ *r*2 = *r*2 ⋈ *r*1 (3)

(*r*1 ⋈ *r*2) ⋈ *r*3 = *r*1 ⋈ (*r*2 ⋈ *r*3) (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 *× r*2*= r*2 *× r*1 (5)

(*r*1 *× r*2) × *r*3 = *r*1 × (*r*2 *× r*3) (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 ∩ *r*2 = *r*2 ∩ *r*1(7)

(*r*1 ∩ *r*2) ∩ *r*3= *r*1 ∩ (*r*2 ∩ *r*3) (8)

*r*1 ∪ *r*2 = *r*2 ∪ *r*1(9)

(*r*1 ∪ *r*2) ∪ *r*3 = *r*1 ∪ (*r*2 ∪ *r*3) (10)

## 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 để mô hình IPRDB được xây dựng 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 IPRDB.

# Chương 2

# CƠ SỞ TOÁN HỌC CỦA MÔ HÌNH IPRDB

## 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 IPRDB. 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 [13] 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 IPRDB

Phần 2.3 trình bày khái niệm giá trị xác suất trong [23]. Các giá trị 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 IPRDB.

Phần 2.4 trình bày các chiến lược kết hợp các giá trị xác suất. Các chiến lược kết hợp các giá trị 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 IPRDB. Phần 2.5 giới thiệu 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.

## 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 *˄ e*2) của sự kiện phức hợp *e*1 *˄ 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 *˄ e*2) = Pr(*e*1).Pr(*e*2), nếu *e*1và *e*2 là loại trừ lẫn nhau thì Pr(*e*1 *˄ e*2) = 0, v.v. Nếu không biết (hoặc bỏ qua) mối quan hệ của *e*1và *e*2 thì Pr(*e*1 *˄ 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 *˄ e*2không chỉ phụ thuộc vào xác suất của các sự kiện *e*1và *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 *˅ 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 [18].

Cũng như trong Eiter và CS. (2001) và một số mô hình cơ sở dữ liệu xác suất khác ([15], [18], [19], [23]), 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 *˄ e*2 là *mâu thuẫn* (contradictory) thì *L*1 *+ L*2*≤* 1*.*
2. Nếu *e*1 *˄* ¬*e*2 là mâu thuẫn thì *L*1 *≤ U*2*.*
3. Nếu ¬*e*1 *˄ e*2 là mâu thuẫn thì *L*2 *≤ U1.*
4. Nếu ¬*e*1 *˄* ¬*e*2 là mâu thuẫn thì *U*1 *+ U*2 *≥* 1.

Trong định nghĩa này, như đã lưu ý trong [18], khái niệm mâu thuẫn ở đây theo nghĩa của logic mệnh đề, *e*1 *˄ e*2 mâu thuẫn nghĩa là *e*1 *˄ 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*≤ I*2 được sử dụng như một sự viết gọn thay cho *L*1*≤ L*2 và *U*1 *≤ U*2 còn ký hiệu *I*1⊆ *I*2 thay cho *L*2≤ *L*1 và *U*1*≤ 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 [13], được Eiter và CS. (2001) mở rộng trong [18].

**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*1và *I*2là một phép toán hai ngôi ⊗ sử dụng các khoảng xác suất này để tính một khoảng xác suất *I =* [*L, U* ] cho *e*1 *˄ e*2 , được ký hiệu bởi *I*1⊗ *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 ⊗ *I*2) ≤ [*min*(*L*1, *L*2), *min*(*U*1, *U*2)] |
| Bỏ qua (Ignorance) | (*I*1 ⊗ *I*2) ⊆ [*max*(0, *L*1 *+ L*2-1), *min*(*U*1, *U*2)] |
| Đồng nhất (Identity) | (*I*1 ⊗ [1, 1]) = *I*1 |
| Giao hoán (Commutativity) | (*I*1⊗ *I*2) = (*I*2⊗ *I*1) |
| Kết hợp (Associativity) | ((*I*1⊗ *I*2) ⊗ *I*3) = (*I*1 ⊗ (*I*2⊗ *I*3)) |
| Đơn điệu (Monotonicity) | (*I*1⊗ *I*2) ≤ (*I*1⊗ *I*3) nếu *I*2 *≤ I*3 |

**Định nghĩa 2.2.3** Giả sử *e*1và *e*2có 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*1và *I*2 là một phép toán hai ngôi ⊕ sử dụng các khoảng xác suất này để tính một khoảng xác suất *I =* [*L, U*] cho *e*1 *˅ e*2, được ký hiệu bởi *I*1 ⊕ *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 ⊕ *I*2) ≥ [*min*(*L*1, *L*2), *min*(*U*1, *U*2)] |
| Bỏ qua | (*I*1 ⊕ *I*2) ⊆ [*max*(*L*1, *L*2), *min(*1, *U*1 *+U*2)] |
| Đồng nhất | (*I*1 ⊕ [0, 0]) = *I*1 |
| Giao hoán | (*I*1⊕ *I*2) = (*I*2⊕ *I*1) |
| Kết hợp | ((*I*1⊕ *I*2) ⊕ *I*3) = (*I*1 ⊕(*I*2⊕ *I3*)) |
| Đơn điệu | (*I*1⊕ *I*2) ≤ (*I*1⊕ *I*3) nếu *I*2 *≤ 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*2có 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*1và *I*2là một phép toán hai ngôi ⊖ sử dụng các khoảng xác suất này để tính một khoảng xác suất *I =* [*L*, *U* ] cho sự kiện *e*1 *˄¬e*2, được ký hiệu bởi *I = I*1 ⊖ *I*2, thỏa các tiên đề sau:

1. Bị chặn: (*I*1 ⊖ *I*2) ≤ [*min*(*L*1, 1 – *U*2), *min*(*U*1, 1 – *L*2)].
2. Bỏ qua: (*I*1 ⊖ *I*2) ⊆ [*max*(0, *L*1 – *U*2), *min*(*U*1, 1 – *L*2)].
3. Đồng nhất: nếu (*¬e*1 *˄¬e*2) và (*e*1 *˄¬e*2) là không mâu thuẫn thì (*I*1⊖[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] ⊗*ig*[*L*2, *U*2]) ≡ [*max*(0, *L*1 + *L2* – 1), *min*(*U*1, *U*2)]  ([*L*1, *U*1] ⊕*ig*[*L*2, *U*2]) ≡ [*max*(*L*1, *L*2 ), *min*(1, *U1* + *U*2)]  ([*L*1, *U*1] ⊖*ig*[*L*2, *U*2]) ≡ [*max* (0, *L*1 – *U*2 ), *min*(*U*1, 1– *L*2)] |
| Độc lập  (Independence) | ([*L*1, *U*1] ⊗*in*[*L*2, *U*2]) ≡ [*L*1.*L*2, *U*1.*U*2]  ([*L*1, *U*1] ⊕*in*[*L*2, *U*2]) ≡ [*L*1 + *L*2 – (*L*1.*L*2), *U*1+*U*2– (*U*1.*U*2)]  ([*L*1, *U*1] ⊖*in*[*L*2, *U*2]) ≡ [*L*1*.*(1 – *U*2), *U*1.(1– *L*2)] |
| Tương quan thuận  (Positive correlation) | ([*L*1, *U*1] ⊗*pc*[*L*2, *U*2]) ≡ [*min*(*L*1, *L*2), *min*(*U*1, *U*2)]  ([*L*1, *U*1] ⊕*pc*[*L*2, *U*2]) ≡ [*max* (*L*1, *L*2 ), *max*(*U1*, *U*2)]  ([*L*1, *U*1] ⊖*pc*[*L*2, *U*2]) ≡ [*max*(0, *L*1 – *U*2 ), *max*(0, *U*1 – *L*2)] |
| Loại trừ nhau  (Mutual Exclusion) | ([*L*1, *U*1] ⊗*me*[*L*2, *U*2]) ≡ [0, 0]  ([*L*1, *U*1] ⊕*me*[*L*2, *U*2]) ≡ [*min* (1, *L*1+ *L*2 ), *min*(1, *U*1 + *U*2)]  ([*L*1, *U*1] ⊖*me*[*L*2, *U*2]) ≡ [*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.

## Giá trị xác suất

Giá trị xác suất (*probabilistic value*) trong [23] là cơ sở để biểu diễn giá trị thuộc tính quan hệ không chắc chắn và không chính xác trong IPRDB và được định nghĩa như sau.

**Định nghĩa 2.3.1** Giả sử *τ* là một kiểu dữ liệu và *D* là miền giá trị của *τ*, một giá trị xác suất trên *D* là một tập hữu hạn các cặp{(*v*1, [*l*1, *u*1]), …, (*vm*, [*lm*, *um*])},trong đó *vi* ∈ *D* và0 ≤ *li* ≤ *ui* ≤ 1, với mọi *i =* 1, 2, …, *m*.

Một giá trị xác suất *pv* = {(*v*1, [*l*1, *u*1]), …, (*vm*, [*lm*, *um*])} biểu diễn khả năng *pv* nhận một trong các giá trị (phần tử) *vi* của tập {*v*1,…, *vm*} với một xác suất trong khoảng [*li*, *ui*]. Một giá trị xác suất *pv* = {(*v*1, [*l*1, *u*1]), …, (*vm*, [*lm*, *um*])} tương ứng với một hàm phân bố xác suất *p* trên tập *V* = {*v*1,…, *vm*} sao cho *p*(*vi*) ∈ [*li*, *ui*] và Σ*vi*∈*V**p*(*vi*) ≤ 1.

**Ví dụ 2.3.1** Giả sử bệnh của một bệnh nhân được chẩn đoán là viêm gan (hepatitis) với một xác suất nằm giữa 0.5 và 0.7hoặcviêm túi mật (cholecystitis) với một xác suất nằm giữa 0.3 và 0.5, thì thông tin này có thể được biểu diễn bởi giá trị xác suất {(hepatitis, [0.5, 0.7]), (cholecystitis, [0.3, 0.5])}.

Một giá trị xác suất có thể được biểu diễn bởi *pv* = {(*v*, *I*)| *v* ∈ *D*, *I* = [*l*, *u*] ⊆ [0, 1]}.

## Các chiến lược kết hợp các giá trị 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 IPRDB được xây dựng bằng cách sử dụng các chiến lược kết hợp của các giá trị xác suất trong [23] 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 giá trị xác suất như định nghĩa sau.

**Định nghĩa 2.4.1** Giả sử *pv*1 và *pv*2 là hai giá trị xác suất và ⊗ là một chiến lược hội xác suất. Thìmột *chiến lược hội của hai giá trị xác suất* *pv*1 và *pv*2 theo ⊗, được ký hiệu bởi *pv*1 ⊗ *pv*2, là giá trị xác suất *pv* được định nghĩa bởi *pv* *=* {(*v*, *I*1 ⊗ *I*2)|(*v*, *I*1)∈ *pv*1,(*v*, *I*2)∈ *pv*2}.

**Ví dụ 2.4.1** Giả sử *pv*1 = {(*hepatitis*, [0.4, 0.6]), (*cholecystitis*, [0.4, 0.6])} và *pv*2 = {(*hepatitis*, [1.0, 1.0])} là các giá trị xác suất, thì *pv*1 ⊗*in pv*2 theo chiến lược hội xác suất độc lập là giá trị xác suất *pv* = {(*hepatitis*, [0.4, 0.6])}.

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

**Định nghĩa 2.4.2** Giả sử *pv*1 và *pv*2 là hai giá trị xác suất và ⊕ là một chiến lược tuyển xác suất. Thìmột *chiến lược* tuyển *của hai giá trị xác suất* *pv*1 và *pv*2 theo ⊕, được ký hiệu bởi *pv*1 ⊕ *pv*2, là giá trị xác suất *pv* được định nghĩa bởi *pv* ={(*v*, *I*1)|(*v*, *I*1)∈ *pv*1 và¬∃ *I*2, (*v*, *I*2) ∈ *pv*2}∪{(*v*, *I*2)|(*v*, *I*2) ∈ *pv*2 và¬∃ *I*1, (*v*, *I*1) ∈ *pv*1}∪{(*v*, *I*1 ⊕ *I*2)|(*v*, *I*1)∈ *pv*1 và(*v*, *I*2)∈ *pv*2}.

**Ví dụ 2.4.2** Giả sử *pv*1 = {(*hepatitis*, [0.2, 0.5]), (*cholecystitis*, [0.3, 0.6])} và *pv*2 = {(*hepatitis*, [0.3, 0.5]), (*pancreatitis*, [0.2, 0.6])} là các giá trị xác suất, thì *pv*1 ⊕*in pv*2 theo chiến lược tuyển xác suất độc lập là giá trị xác suất *pv* = {(*cholecystitis*, [0.3, 0.6]), (*pancreatitis*, [0.2, 0.6]), (*hepatitis*, [0.44, 0.75])}.

**Định nghĩa 2.4.3** Giả sử *pv*1 và *pv*2 là hai giá trị xác suất và ⊖ là một chiến lược hiệu xác suất. Thìmột *chiến lược hiệu của hai giá trị xác suất* *pv*1và *pv*2 theo ⊖, được ký hiệu bởi *pt*1 ⊖ *pt*2, là giá trị xác suất *pv* được định nghĩa bởi *pv* ={(*v*, *I*1)|(*v*, *I*1)∈ *pv*1 và¬∃ *I*2, (*v*, *I*2) ∈ *pv*2} ∪ {(*v*, *I*1⊖*I*2)|(*v*, *I*1)∈ *pv*1 và(*v*, *I*2)∈ *pv*2}.

## 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 IPRDB, diễn dịch xác suất của quan hệ hai ngôi trên các tập hợp trong [23] được sử dụng như định nghĩa dưới đây.

**Định nghĩa 2.5.1** Giả sử *A* và *B* là hai tập hợp, *U* và *V* là các miền giá trị, và θ là một quan hệ hai ngôi trong {=, ≠, ≤, ≥, <, >}*.* *Diễn dịch xác suất của quan hệ A*  *B*, được ký hiệu *Pr*(*A* θ *B*), là một giá trị trong [0, 1] được định nghĩa bởi:

*Pr*(*A B*) *= p*(*u v| uA*, *vB*) = *p*(*u v*)

với *A* là một tập con của *U*, *B* là một tập con của *V* được giả sử là hợp lệ trên (*U* × *V*), *p*(*u*  *v*| *uA*, *vB*) là xác suất có điều kiện của *u*  *v* với *uA* và *vB*.

Từ định nghĩa này chúng ta có

*Pr*(*A B*) *=*|{(*u*, *v*): *u v*, *uA*, *vB*}|/(|*A*|×|*B*|)

Lưu ý rằng diễn dịch xác suất của các quan hệ hai ngôi trong định nghĩa này có thể được áp dụng cho các phần tử của *U* và *V* bởi vì một phần tử *e* của một tập *D* cũng được coi như là một tập con {*e*} của *D*.

**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:

*Pr*(*A = B*) *= p*(*u = v| u*∈*A, v* ∈ *B*) *= p*(*u=v | u* ∈{3, 4}*, v*∈{4, 5}) = 0.25.

*Pr*(*A* < *B*) *= p*(*u* < *v| u*∈*A, v* ∈ *B*) = *p*(*u* < *v | u* ∈{3, 4}*, v*∈{4, 5}) = 0.75.

*Pr*(4 = 4) = 1.0 (được xem như *Pr*({4} = {4}) = 1.0).

*Pr*(3 = 4) = 0.0 (được xem như *Pr*({3} = {4}) = 0.0).

*Pr*(3 ≠ 4) = 1.0 (được xem như *Pr*({3} ≠ {4}) = 1.0).

*Pr*(3 ≥ 4) = 0.0 (được xem như *Pr*({3} ≥ {4}) = 0.0).

*Pr*(3 ≤ 4) =1.0 (được xem như *Pr*({3} ≤ {4}) = 1.0).

## 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 IPRDB đã được giới thiệu. Các giá trị xác suất được dùng để xây dựng mô hình dữ liệu của IPRDB 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 giá trị xác suất là sự kết hợp các khoảng xác suất tương ứng với các tập giá trị trong các giá trị 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 IPRDB

## Giới thiệu

Chương này trình bày những khái niệm nền tảng của mô hình cơ sở dữ liệu quan hệ xác suất IPRDB. Phần 3.2 giới thiệu khái quát mô hình ý niệm IPRDB, 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 IPRDB. 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 IPRDB nói riêng. Phần 3.4 là mở rộng khái niệm kiểu và giá trị trong mô hình CSDL quan hệ truyền thống thành kiểu và giá trị trong IPRDB. Kế đến, trong Phần 3.5 và 3.6, lược đồ, quan hệ và phụ thuộc hàm của IPRDB 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.

## Mô hình ý niệm

Một cách khái quát, IPRDB 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 IPRDB. 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.

**DOCTOR**

Diagnose

**PATIENT**

Hình 3.2.1 Cơ sở dữ liệu khám-chữa bệnh

Hình 3.2.1cho 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 giá trị xác suất như DISEASE: **{(*v*, *I*)}**,DURATION: **{(*v*, *I*)}** hay COST: **{(*v*, *I*)}** v.v, là các thuộc tính có giá trị không chắc chắn.

Hình 3.2.2 mô tả kiến trúc hệ của thống IPRDB. Hệ thống này bảo đảm quá trình xử lý các truy vấn IPRDB, 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 giá trị xác suất. Cụ thể, người sử dụng biểu diễn các truy vấn trong ngôn ngữ IPRDB 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 IPRDB thành các truy vấn ngôn ngữ đại số của hệ thống IPRDB. Sau đó chúng sẽ được bộ thực thi tính toán và xử lý dựa trên dữ liệu trong IPRDB 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 phân bố các khoảng xác suất (giá trị xác suất) cho phép người dùng biểu diễn các khoảng xác suất kết hợp một không gian các giá trị thuộc tính.

probabilistic combination strategies probabilistic values

IPRDB Calculus Query

IPRDB Algebra Query

GUI

IPRDB -Algebra Query Manager

IPRDB

IPRDB -Algebra Execution Engine

USER

Hình 3.2.2 Kiến trúc của hệ thống IPRDB

## 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]). 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 IPRDB, 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 ta 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 chỉ có thể chẩn đoán bệnh với xác suất bao nhiêu và sẽ không chắc chắn 100% thời gian điều trị là bao lâu vì tùy thuộc vào thể trạng bệnh nhân và sự tiến triển. Chúng ta 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 chắn chắn bằng 0, nghĩa là mức độ chắc chắn bằng 1). Vì vậy, trong IPRDB, mọi thuộc tính đều có thể coi là không chắc chắn.

## Kiểu và giá trị

Tương tự như mô hình cơ sở dữ liệu quan hệ truyền thống, trong IPRDB, 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 IPRDB, 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* ∈***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 {*τ*} là một kiểu được gọi là *kiểu tập hợp* (set type) của *τ*.
3. Nếu *A*1*, A*2*,…,Ak* là các thuộc tính đôi một khác nhau trong ***A*** và *τ*1, *τ*2*,*…*τk* là các kiểu thì *τ* = [*A*1: *τ*1, *A*2: *τ*2,…, *Ak*: *τk*] là một kiểu, được gọi là *kiểu bộ* (tuple type) trên tập các thuộc tính{*A*1, *A*2,…, *A*k}. Với một kiểu *τ* = [*A*1: *τ*1, *A*2: *τ*2,…, *Ak*: *τk*], chúng tôi sử dụng *τ*.*Ai*hoặc *τ*[*Ai*] để biểu thị *τi*.

**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 {**string**}, [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 ***T***, 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 *τ* ∈ ***T*** có một miền xác định dom(*τ*) kết hợp với nó. *Giá trị* (value) được định nghĩa như sau:

1. Với mọi kiểu cơ bản *τ* ∈ ***T*** , thì mọi *v* ∈ *dom*(*τ*) là một giá trị kiểu *τ*.
2. Nếu *v*1*, v*2*,…,vk* là các giá trị thuộc kiểu *τ*, thì {*v*1*, v*2*,…, vk*} là một giá trị kiểu {*τ*}.
3. Nếu *A*1,…, *Ak*là các thuộc tính đôi một khác nhau và *v*1*,…, vk* là các giá trị tương ứng của các kiểu *τ*1,…, *τk* thì [*A*1: *v*1, …, *Ak*: *v**k*] là một giá trị kiểu [*A*1: *τ*1,…, *Ak*: *τk*], được gọi là *giá trị kiểu bộ* (tuple type value) trên tập các thuộc tính {*A*1, *A*2…, *Ak*}. Đối với một giá trị *v =* [*A*1: *v*1,…, *Ak*: *vk*], chúng ta sử dụng ký hiệu *v*.*Ai* hoặc *v*[*Ai*] để biểu thị giá trị của thuộc tính *Ai*.

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*,.., vk*).

**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: 20/9/2024, MEDICAL\_HISTORY: {*cholecystitis*}].

**Định nghĩa 3.4.3** Giả sử *A*1, *A*2, …, *Ak* là các thuộc tính đôi một khác nhau, có kiểu giá trị tương ứng là *τ*1,…, *τk* và *pv*1,…, *pvk* là các giá trị xác suất với *pvi =* {(*vi*, [*li*, *ui*])| *vi* ∈ *dom*(*τi*), [*li*, *ui*] ⊆ [0, 1]}, thì biểu thức [*A*1: *pv*1, *…*, *Ak* : *pvk*] là một *giá trị bộ xác suất* (probabilistic tuple value) kiểu [*A*1: *τ*1,…, *Ak*: *τk*] trên tập các thuộc tính {*A*1,…, *Ak*}. Với mỗi bộ giá trị xác suất *ptv =* [*A*1: *pv*1, *…*, *Ak* : *pvk*], chúng ta sử dụng ký hiệu *ptv.Ai*hoặc *ptv*[*Ai*] để biểu thị *pvi*. Nghĩa là *ptv.Ai* = *ptv*[*Ai*] = {(*vi*, [*li*, *ui*])| *vi* ∈ *dom*(*τi*), [*li*, *ui*] ⊆ [0, 1]}.

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 IPRDB. 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 IPRDB. Ngoài ra, trong một ngữ cảnh nào đó, một giá trị bộ xác suất có thể được viết đơn giản *ptv=*({(*v*1, [*l*1, *u*1])}, *…*,{(*vk*, [*lk*, *uk*])}), nếu thấy không nhất thiết phải kể đến các thuộc tính tương ứng của chúng.

**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) hoặc xơ gan (cirrhosis). 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: {(Ho V.T, [1, 1])}, BIRTHDAY: {(03/15/1969, [1, 1])}, DISEASE: {(hepatitis, [0.5, 0.5]), (cirrhosis, [0.5, 0.5])}, DURATION: {(30, [0.4, 0.6]), (32, [0.4, 0.6])}, COST: {60, [0.5, 0.5]), (70, [0.5, 0.5])}.

## Lược đồ và quan hệ

Như đã giới thiệu một cách khái quát trong Phần 3.2, một lược đồ IPRDB 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 IPRDB [23] được mở rộng từ lược đồ quan hệ trong CSDL quan hệ truyền thống với các thuộc tính có thể có giá trị 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***, ℘), trong đó

1. ***U*** = {*A*1, *A*2, …, *Ak*} là một tập các thuộc tính đôi một khác nhau.
2. ℘ là một ánh xạ gán mỗi thuộc tính *A* ∈ ***U*** cho tập tất cả các giá trị xác suất trên miền giá trị của *A* (nghĩa là mỗi phần tử của ℘(*A*) là một giá trị xác suất có dạng {(*v*, *I*)| *v* ∈ *D*, *I* = [*l*, *u*] ⊆ [0, 1]}, trong đó *D* là miền giá trị của *A*).

Lưu ý rằng, như trong CSDL quan hệ cổ điển, để đơn giản, các ký hiệu *R*(***U***,℘) và *R* có thể được sử dụng thay cho *R* = (***U***, ℘). 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, ℘).

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, ℘ là ánh xạ gán mỗi thuộc tính trong **PATIENT** cho tập tất cả các giá trị 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, …, *Ak*} 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***, ℘), là một tập hữu hạn các phần tử (bộ) {*t*1, *t*2,…, *tn*}, trong đó mỗi *ti =* (*pvi*1, *pvi*2, …, *pvik*)là một danh sách *k* giá trị xác suất *pvij =* {(*vij*, [*lij*, *uij*])| *vij* ∈ *dom*(*Aj*), [*lij*, *uij*] ⊆ [0, 1]}, *j* =1, 2,…, *k*  sao cho *pvij* ∈℘(*Aj*)với mọi *i =* 1, 2,…, *n* (nghĩa là, mỗi *ti* là một giá trị bộ xác suất trên tập các thuộc tính {*A*1*,…, Ak*}).

Mỗi giá trị xác suất *pvij* biểu diễn giá trị không chắc chắn của thuộc tính *Aj* của bộ *ti*. Các ký hiệu *ti*.*Aj* hoặc *ti*[*Aj*] biểu thị *pvij* và [*ti*] biểu thị (*Vi*1, *Vi*2, …, *Vik*), trong đó *Vij* = {*vij* | (*vij*, [*lij*, *uij*]) ∈ *pvij*}. Đối với mỗi tập thuộc tính ***H*** ⊆ {*A*1*, A*2*, …, Ak*}, ký hiệu *t*[***H***] 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 ***H***.

Từ định nghĩa 2.3.1, chúng tôi lưu ý rằng, mỗi thuộc tính *Aj* của bộ *ti* trong quan hệ *r* trên *R*(***U***, ℘) chỉ nhận một giá trị *vij* trong *Vij* với một xác suất *p*(*vij*) ∈ [*lij*, *uij*].

**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).

Bảng 3.5.1 Quan hệ PATIENT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P\_ID | P\_NAME | P\_AGE | P\_DISEASE | D\_COST |
| P104 | John | {(65, [1, 1])} | {(lung cancer, [0.5, 0.5]), (tuberculosis, [0.5, 0.5])} | {($30, [0.3, 0.6]), ($35, [0.4, 0.7])} |
| P218 | Paul | {(43, [0.5, 0.5]), (44, [0.5, 0.5])} | {(hepatitis, [0.3, 0.5]), (cirrhosis, [0.5, 0.7])} | {($6, [0.4, 0.6]), ($7, [0.4, 0.6])} |
| P325 | Helen | {(36, [1, 1])} | {(duodenitis, [0.5, 0.5]), (gastritis, [0.5, 0.5])} | {($8, [0.5, 0.5]), ($9, [0.5, 0.5])} |
| P412 | Anne | {(15, [1, 1])} | {(bronchitis, [1, 1])} | {($7, [1, 1])} |
| P426 | George | {(36, [1, 1])} | {(duodenitis, [0.4, 0.5]), (gastritis, [0.5, 0.6])} | {($8, [0.3, 0.5]), ($9, [0.5, 0.7])} |

Lưu ý rằng, đối với mỗi thuộc tính *A* ∈***U*** ={P\_ID, P\_NAME, P\_AGE, P\_DISEASE, D\_COST} trong lược đồ **PATIENT**(***U*,** ℘), ℘(*A*) bao gồm tất cả các giá trị xác suất trên *dom*(*A*).

Để đơn giản, mỗi giá trị xác suất dạng {(*v*, [1, 1])}, sẽ được biểu diễn như một giá trị đơn *v*. Bởi vì nếu thuộc tính nhận một giá trị xác suất như vậy, thì thực sự nó chỉ nhận một giá trị *v* với xác suất là 1 (Định nghĩa 3.5.2). 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 IPRDB, 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 ⊗ 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 ⊗, được biểu thị bởi *p*(*t*1.*A* =⊗ *t*2.*A*), là

*p*(*t*1.*A* =⊗ *t*2.*A*) = (([*l*1*i*, *u*1*i*]⊗ [*l*2*j*, *u*2*j*]).*Pr*(*v*1*i* = *v*2*j*)).

Trong đó *t*1*.A =* {(*v*11, [*l*11, *u*11]), …, (*v*1*m*, [*l*1*m*, *u*1*m*])}, *t*2.*A =* {(*v*21, [*l*21, *u*21]), …, (*v*2*n*, [*l*2*n*, *u*2*n*])} và ⊕ là phép tuyển xác suất loại trừ nhau

Bây giờ, phụ thuộc hàm xác suất trong IPRDB được mở rộng từ phụ thuộc hàm xác suất trong mô hình CSDL truyền thống như sau.

**Định nghĩa 3.5.4** Giả ***U*** = {*A*1, *A*2,…, *Ak*} là một tập *k* thuộc tính đôi một khác nhau, *R*(***U***, ℘) là một lược đồ quan hệ xác suất, *r* là một quan hệ bất kỳ trên *R*, ⊗ là một chiến lược hội xác suất, *X* và *Y* là hai tập con của ***U***. Một *phụ thuộc hàm xác suất của* *Y* đối với *X* theo ⊗ trên *R*, được ký hiệu bởi *X* ⇝⊗ *Y*, nếu và chỉ nếu:

∀*t*1, *t*2 ∈ *r*: ⊗*A*∈***X*** *p*(*t*1.*A* =⊗ *t*2.*A*) ≤ ⊗*A*∈***Y*** *p*(*t*1.*A* =⊗ *t*2.*A*).

Một ví dụ hiển nhiên của phụ thuộc hàm xác suất là mọi thuộc tính *Ai* phụ thuộc tập {*A*1,*A*2,*…*, *Ak*} bao gồm tất cả các thuộc tính của lược đồ quan hệ *R*. Chúng ta lưu ý rằng, trong CSDL quan hệ truyền thống, vì xác suất cho hai giá trị bằng nhau là bằng 0 hoặc 1, do đó phụ thuộc hàm trong CSDL quan hệ truyền thống là một trường hợp riêng của phụ thuộc hàm xác suất theo định nghĩa này.

Như đối với CSDL quan hệ truyền thống, *khóa* (key) của một lược đồ quan hệ trong IPRDB 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 truyền thống, khóa được ràng buộc không nhận giá trị NULL ([1], [2]). Tương tự, trong IPRDB, 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***, ℘) là một lược đồ quan hệ xác suất, *r* là một quan hệ bất kỳ trên *R* và ⊗ là một chiến lược hội xác suất, một tập thuộc tính *K* ⊆ ***U*** được gọi là khóa của *R* theo ⊗ 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* ⇝⊗ ***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.

## Kết luận

Trong Chương 3 này các khái niệm thuộc tính, kiểu, giá trị, lược đồ và quan hệ trong mô hình IPRDB 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 giá trị xác suất. Bây giờ, thông tin của các đối tượng trong IPRDB 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 IPRDB đã được xây dựng. Trong Chương 4 tiếp theo, các phép toán đại số trên IPRDB 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 IPRDB

## Giới thiệu

Các phép toán đại số quan hệ trên IPRDB 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 IPRDB. Các phép toán đại số trên IPRDB 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 IPRDB sao cho nhất quán với mô hình dữ liệu IPRDB. Đối số và kết quả của các phép toán đại số trên IPRDB là các quan hệ IPRDB. 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 IPRDB không chỉ thao tác trên các giá trị không chắc chắn của các quan hệ trong IPRDB 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 IPRDB 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 IPRDB được xây dựng là đúng đắn. Tập các phép toán trên IPRDB 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.

## Phép chọn

Cũng như trong CSDL quan hệ truyền thống ([1], [2]), phép chọn là một phép toán đại số cơ bản trên IPRDB. 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, cú pháp và ngữ nghĩa hình thức của các *biểu thức chọn* (selection expression) và *điều kiện chọn* (selection condition) được mở rộng từ cú pháp và ngữ nghĩa của các biểu thức chọn và điều kiện chọn trong CSDL quan hệ truyền thống với giá trị xác suất như sau.

**Định nghĩa 4.2.1** Giả sử *R* là một lược đồ IPRDB 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* θ *c*, trong đó *x* ∈ ***X***, *A* là một thuộc tính trong *R*, θ là một quan hệ hai ngôi thuộc {=, ≠, ≤, ≥, <, >} và *c* ∈ *dom*(*A*).
2. *x.A*1 =⊗ *x.A*2, trong đó *x* ∈ ***X***, *A*1 và *A*2là hai thuộc tính phân biệt trong *R* và ⊗ 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 ⊗ *E*2, trong đó *E*1và *E*2 là các biểu thức chọn trên cùng một biến bộ quan hệ, ⊗ là một chiến lược hội xác suất.
4. *E*1 ⊕ *E*2, trong đó *E*1và *E*2 là các biểu thức chọn trên cùng một biến bộ quan hệ, ⊕ 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* ⊗ *x*.D\_COST ≥ 7.

Trong IPRDB, 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 IPRDB, 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* ≤ *U* thì (*E*)[*L, U*] là một điều kiện chọn.
2. Nếu ɸ và Ψ là các điều kiện chọn trên cùng một biến đối tượng thì ¬φ, (φ∧ψ) và φ∨ψ) 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] ∧ (*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* ⊗ *x*.D\_COST ≥ 7)[0.4, 0.6].

Đối với IPRDB, 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 IPRDB, *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 *probR,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. *probR*,*r*,*t*(*x*.*A* θ *c*) = ([*li*, *ui*].*Pr*(*vi* θ *c*)),trong đó *t.A =* {(*v*1, [*l*1, *u*1]), …, (*vk*, [*lk*, *uk*])} và ⊕ là phép tuyển xác suất loại trừ.
2. *probR*,*r*,*t*(*x*.*A*1=⊗*x*.*A*2) = (([*l*1*i*, *u*1*i*]⊗ [*l*2*j*, *u*2*j*]).*Pr*(*v*1*i* = *v*2*j*)), trong đó *t.A*1 *=* {(*v*11, [*l*11, *u*11]), …, (*v*1*m*, [*l*1*m*, *u*1*m*])}, *t*.*A*2 *=* {(*v*21, [*l*21, *u*21]), …, (*v*2*n*, [*l*2*n*, *u*2*n*])} và ⊕ là phép tuyển xác suất loại trừ.
3. *probR*,*r*,*t*(*E*1 ⊗ *E*2) = *probR*,*r*,*t*(*E*1) ⊗ *probR*,*r*,*t*(*E*2).
4. *probR*,*r*,*t*(*E*1 ⊕ *E*2) = *probR*,*r*,*t*(*E*1) ⊕ *probR*,*r*,*t*(*E*2).

Lưu ý rằng phép tuyển xác suất loại trừ ⊕*me* được sử dụng ở mục 1 và 2 của Định nghĩa này bởi vì các khoảng [*l*1, *u*1], …, [*lk*, *uk*] biểu diễn một hàm phân bố trên {*v*1,…, *vk*}, tương tự cho [*l*11, *u*11], …, [*l*1*m*, *u*1*m*] và [*l*21, *u*21], …, [*l*2*n*, *u*2*n*]. Một cách trực giác, *probR,****r****,t*(*x*.*A* θ *c*) là khoảng xác suất để thuộc tính *A* của bộ *t* có giá trị *vi* sao cho *vi* θ *c* và *probR*,*r*,*t*(*x*.*A*1 =⊗ *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*i* và *v*2*j* sao cho *v*1*i* = *v*2*j*.

**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

*probR,r,t*2(*x*.P\_DISEASE = cirrhosis)

= [0.3, 0.5].*Pr*(hepatitis = cirrhosis) ⊕*me* [0.5, 0.7].*Pr*(cirrhosis = cirrhosis)

= [0.3, 0.5]×0.0 ⊕*me* [0.5, 0.7]×1.0

= [0, 0] ⊕*me*[0.5, 0.7]

= [0.5, 0.7]

Trong IPRDB, 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 IPRDB như sau.

**Định nghĩa 4.2.4** Giả sử *R* là một lược đồ quan hệ IPRDB, *r* là một quan hệ trên *R* và *t* ∈ *r*. *Sự thỏa mãn* (satisfaction) các điều kiện chọn của *t* theo diễn dịch xác suất *probR,r,t* được định nghĩa như sau:

1. *probR,r,t* ⊨ (*E*)[*L*, *U*] nếu và chỉ nếu *probR,r,t*(*E*) ⊆ [*L*, *U*].
2. *probR,r,t* ⊨ ¬φ nếu và chỉ nếu *probR,r,t* ⊨φ không thỏa.
3. *probR,r,t* ⊨ φ∧ψ nếu và chỉ nếu *probR,r,t* ⊨ φ và *probR,r,t* ⊨ ψ thỏa.
4. *probR,r,t* ⊨ φ∨ψ nếu và chỉ nếu *probR,r,t* ⊨ φ thỏa hoặc *probR,r,t* ⊨ ψ 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 IPRDB.

**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 *probR,r,t*2⊨ (*x*.P\_DISEASE = cirrhosis)[0.4, 1.0] và *probR,r,t*2⊭ (*x*.P\_DISEASE = cirrhosis)[0.6, 1.0], vì *probR,r,t*2(*x*.P\_DISEASE = cirrhosis) = [0.5, 0.7] ⊆ [0.4, 1.0] và *probR,r,t*2(*x*.P\_DISEASE = cirrhosis) = [0.5, 0.7] ⊄ [0.6, 1.0].

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

**Định nghĩa 4.2.5** Giả sử *R* là một lược đồ quan hệ IPRDB, *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 σφ(*r*), là một quan hệ *r\** trên *R*, bao gồm tất cả các bộ thỏa mãn điều kiện chọn φ.

*r\** = {*t* ∈*r* | *probR,r,t*⊨ φ}.

**Ví dụ 4.2.5** Xét quan hệ *r* = 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 bệnh xơ gan (cirrhosis) 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 σφ(PATIENT), trong đó φ =(*x*.P\_AGE > 40)[0.9, 1] ∧ (*x*.P\_DISEASE = cirrhosis ⊗*in* *x*.D\_COST ≥ 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:

*probR,r,t*2(*x*.P\_AGE > 40) = [0.5, 0.5]×*Pr*(43 > 40) ⊕*me* [0.5, 0.5]×*Pr*(44 > 40)

= [0.5, 0.5]×1.0 ⊕*me* [0.5, 0.5]×1.0

= [1.0, 1.0] ⊆ [0.9, 1].

*probR,r,t*2(*x*.D\_COST ≥ 6) = [0.4, 0.6]×*Pr*(6 ≥ 6) ⊕*me* [0.4, 0.6]×*Pr*(7 ≥ 6)

= [0.4, 0.6] ×1.0 ⊕*me* [0.4, 0.6]×1.0

= [0.4, 0.6] ⊕*me* [0.4, 0.6]

= [0.8, 1].

Từ kết quả tính toán trong Ví dụ 4.2.3, chúng ta có

*probR,r,t*2(*x*.P\_DISEASE=cirrhosis ⊗*in* *x*.D\_COST≥ 6) = [0.5, 0.7] ⊗*in* [0.8, 1]

= [0.4, 0.7] ⊆ [0.3, 0.7].

Với các bộ khác ta có p*robR,r,ti*(*x*.P\_DISEASE = cirrhosis ⊗*in* *x*.D\_COST ≥ 6) = [0, 0] ⊄ [0.3, 0.7], ∀*i* ≠ 2. Vì vậy, kết quả của truy vấn chọn là quan hệ như trong Bảng 4.2.1.

Bảng 4.2.1 Quan hệ σφ(PATIENT)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P\_ID | P\_NAME | P\_AGE | P\_DISEASE | D\_COST |
| P218 | Paul | {(43, [0.5, 0.5]), (44, [0.5, 0.5])} | {(hepatitis, [0.3, 0.5]), (cirrhosis, [0.5, 0.7])} | {($6, [0.4, 0.6]), ($7, [0.4, 0.6])} |

## Phép chiếu

Cũng như trong các cơ sở dữ liệu truyền thống, ý nghĩa của *phép chiếu* (projection) các quan hệ IPRDB 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* = (***U***,) là một lược đồ quan hệ xác suất, *r* là quan hệ trên *R* và ***L*** là tập con của tập thuộc tính ***U***, ⊕ là một chiến lược tuyển xác suất. *Phép chiếu của r trên* ***L*** theo ⊕, ký hiệu là Π***L***⊕(*r*),là quan hệ xác suất *r\** trên lược đồ *R\** được xác định bởi:

1. *R*\* = (***L***, ℘\*) và ℘\*(*A*) = ℘(*A*), ∀*A* ∈ ***L***.
2. *r*\* = {*t*\* | *t*\*.*A* = *u*.*A* ⊕…⊕ *w*.*A*, ∀*A* ∈ ***L***,*t*∈*r* sao cho ∃ *u*, …, *w* ∈*r* và [*u*[***L***]] = …=[*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\_AGE, P\_DISEASE, D\_COST} theo ⊕*in* được tính như trong bảng 4.3.1.

Bảng 4.3.1 Quan hệ ∏L⊕in(PATIENT)

|  |  |  |
| --- | --- | --- |
| P\_AGE | P\_DISEASE | D\_COST |
| {(65, [1, 1])} | {(lung cancer, [0.5, 0.5]), (tuberculosis, [0.5, 0.5])} | {($30, [0.3, 0.6]), ($35, [0.4, 0.7])} |
| {(43, [0.5, 0.5]), (44, [0.5, 0.5])} | {(hepatitis, [0.3, 0.5]), (cirrhosis, [0.5, 0.7])} | {($6, [0.4, 0.6]), ($7, [0.4, 0.6])} |
| {(15, [1, 1])} | {(bronchitis, [1, 1])} | {($7, [1, 1)} |
| {(36, [1, 1])} | {(duodenitis, [0.7, 0.75]), (gastritis, [0.75, 0.8])} | {($8, [0.65, 0.75]), ($9, [0.75, 0.85])} |

Lưu ý rằng trong quan hệ PATIENT, chúng ta có [*t*3[***L***]] = [*t*5[***L***]], vì vậy hai bộ *t*3 và *t*5, được chiếu trên ***L*** và được kết hợp thành bộ *t*4 theo chiến lược tuyển xác suất độc lập ⊕*in* trong bảng 4.3.1.

## Phép tích Descartes

*Tích Descartes* (Descartes product) của hai quan hệ trong IPRDB 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 IPRDB chỉ được định nghĩa trên hai quan hệ có tập thuộc tính rời nhau. Phép tích Descartes của hai quan hệ IPRDB 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**, ℘1)*, R*2 = (***U*2**, ℘2) là hai lược đồ IPRDB, *r*1, *r*2 lần lượt là hai quan hệ trên *R*1 và *R*2. *Phép tích Descartes* của *r*1và *r*2, ký hiệu là *r*1× *r*2, là quan hệ xác suất *r* trên *R,* được xác định bởi:

1. *R* = (***U***,℘), trong đó ***U*** = ***U­*1** ∪ ***U*2**, ℘(*A*) = ℘1(*A*) nếu *A* ∈***U***1 và ℘(*A*) = ℘2(*A*) nếu *A*∈***U***2.
2. *r* = {*t* | *t*.*A* = *t*1.*A* nếu *A* ∈ ***U*1**, *t*.*A* = *t*2.*A* nếu *A* ∈ ***U*2**, *t*1 ∈ *r*1, *t*2 ∈ *r*2}.

## Phép kết tự nhiên

Phép kết tự nhiên hai quan hệ xác suất trong IPRDB 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 giá trị 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 IPRDB 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 IPRDB được mở rộng từ phép kết tự nhiên của hai quan hệ truyền thống với xác suất và giá trị thuộc tính không chắc chắn 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***,* ℘1) và *R*2 = (***U*2**, ℘2) là hai lược đồ IPRDB, *r*1*, r*2lần lượt là hai quan hệ trên *R*1và *R*2, và ⊗ là một chiến lược hội xác suất. *Phép kết tự nhiên* của *r*1và *r*2 theo ⊗, ký hiệu là *r*1⋈⊗ *r*2, là quan hệ xác suất *r* trên lược đồ *R*, được xác định bởi:

1. *R* = (***U***,℘) trong đó ***U*** = ***U­*1** ∪ ***U­*2**, ℘(*A*) = ℘1(*A*) nếu *A* ∈***U***1- ***U­*2**, ℘(*A*) = ℘2(*A*) nếu *A* ∈ ***U***2 - ***U­*1** và ℘(*A*) = ℘1(*A*) =℘2(*A*) nếu *A*∈ ***U­*1**∩***U­*2**.
2. *r* = {*t* | *t*.*A* = *t*1.*A* nếu *A* ∈ ***U*1** − ***U*2**, *t*.*A* = *t*2.*A* nếu *A* ∈ ***U*2** − ***U*1**, *t*.*A* = *t*1.*A* ⊗ *t*2.*A* nếu *A* ∈ ***U*1** ∩ ***U*2**, *t*1 ∈ *r*1, *t*2 ∈ *r*2}.

**Ví dụ 4.5.1** Giả sử hai quan hệ PATIENT1 và PATIENT2 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 hệPATIENT1

|  |  |
| --- | --- |
| P\_ID | P\_DISEASE |
| P421 | {(bronchitis, [0.35, 0.45]), (bronchiectasis, [0.55, 0.65)} |
| P829 | {(pancreatitis, [1, 1])} |

Bảng 4.5.2 Quan hệ PATIENT2

|  |  |
| --- | --- |
| P\_NAME | P\_DISEASE |
| Peter | {(bronchiectasis, [1, 1])} |
| **Selena** | {(pancreatitis, [0.4, 0.5]), (cirrhosis, [0.5, 0.6])} |

Bảng 4.5.3 Quan hệ PATIENT1 ⋈⊗in PATIENT2

|  |  |  |
| --- | --- | --- |
| P\_ID | P\_NAME | P\_DISEASE |
| P421 | Peter | {(bronchiectasis, [0.55, 0.65)} |
| P829 | **Selena** | {(pancreatitis, [0.4, 0.5])} |

## Phép giao, hợp và trừ

Một cách không hình thức, các phép toán *giao* (intersection), *hợp* (union) và *trừ* (difference) của hai quan hệ IPRDB 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 IPRDB 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 tuyển 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 IPRDB là mở rộng chính các phép toán đó trên mô hình cơ sở dữ liệu quan hệ truyền thống ([1], [2]). Phép giao, hợp và trừ của hai quan hệ IPRDB lần lượt được định nghĩa như dưới đây.

**Định nghĩa 4.6.1** Giả sử *R =* (***U***,℘) là một lược đồ IPRDB, *r*1 *và r*2 là hai quan hệ trên *R*, *K* là một khóa của *R* và ⊗ là một chiến lược hội xác suất. *Phép giao* của *r*1 và *r*2 theo ⊗, ký hiệu là *r*1 ∩⊗ *r*2, là quan hệ xác suất trên *R* được xác định bởi *r* = {*t* | *t*.*A* = *t*1.*A* ⊗ *t*2.*A*, *t*1 ∈ *r*1, *t*2 ∈ *r*2, *A* ∈ ***U***,sao cho *t*1[*K*] = *t*2[*K*]}.

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 khóa tương ứng thuộc về hai quan hệ đó. Trong CSDL quan hệ xác suất, do tính không chắc chắn và không chính xác của các giá trị thuộc tính, bộ chung là không hoàn toàn trùng nhau. Khái niệm “các bộ chung” được mở rộng thành khái niệm “các bộ cùng khóa” (các bộ cùng khóa thực chất biểu diễn một phần giá trị của cùng một đối tượng). Như vậy, định nghĩa giao của hai quan hệ xác suất là mở rộng giao của hai quan hệ truyền thống. Khái niệm các bộ cùng khóa cũng được sử dụng cho các định nghĩa hợp và hiệu của hai quan hệ xác suất

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

**Định nghĩa 4.6.2** Giả sử *R* = (***U***, ℘) là một lược đồ IPRDB, *r*1và *r*2 là hai quan hệ trên *R*, *K* là một khóa của *R* và ⊕ là một chiến lược tuyển xác suất. *Phép hợp* của *r*1 và *r*2 theo ⊕, ký hiệu là *r*1∪⊕ *r*2, là quan hệ xác suất *r* trên *R* được xác định bởi *r* = {*t*1 ∈ *r*1 | ∀*t*2 ∈ *r*2, *t*1[*K*] ≠ *t*2[*K*]}∪{*t*2 ∈ *r*2 | ∀*t*1 ∈ *r*1, *t*2[*K*] ≠ *t*1[*K*]}∪{*t* | *t*.*A* = *t*1.*A* ⊕ *t*2.*A*, *t*1 ∈ *r*1, *t*2 ∈ *r*2, *A* ∈ ***U*** sao cho *t*1[*K*] = *t*2[*K*]}.

**Ví dụ 4.6.1** Giả sử hai quan hệ DIAGNOSE1 và DIAGNOSE2 trên cùng một lược đồ **DIAGNOSE**({P\_ID, D\_ ID, P\_DISEASE, D\_COST}, ℘) với khóa là {P\_ID, D\_ID} được cho như trong các Bảng 4.6.1 và 4.6.2, thì hợp của chúng theo chiến lược tuyển xác suất độc lập ⊕*in* là quan hệ DIAGNOSE = DIAGNOSE1 ∪⊕*in* DIAGNOSE2 được tính toán như trong Bảng 4.6.3.

Bảng 4.6.1 Relation DIAGNOSE1

|  |  |  |  |
| --- | --- | --- | --- |
| P\_ID | D\_ ID | P\_DISEASE | D\_COST |
| P226 | D014 | {(lung cancer, [0.3, 0.6]), (tuberculosis, [0.4, 0.7])} | {($30, [0.3, 0.4]), ($35, [0.6, 0.7])} |
| P255 | D020 | {(hepatitis, [0.3, 0.8]), (pancreatitis, [0.2, 0.7])} | {($8, [0.6, 1])} |

Bảng 4.6.2 Relation DIAGNOSE2

|  |  |  |  |
| --- | --- | --- | --- |
| P\_ID | D\_ ID | P\_DISEASE | D\_COST |
| P228 | D016 | {(lung cancer, [1, 1])} | {($30, [1, 1])} |
| P255 | D020 | {(hepatitis, [0.4, 0.8]), (cholecystitis, [0.2, 0.6])} | {($7, [0.2, 0.4]), ($8, [0.4, 0.8])} |
| P262 | D022 | {(dyspepsia, [1, 1])} | {($5, [1, 1])} |

Bảng 4.6.3 DIAGNOSE1∪⊕in DIAGNOSE2

|  |  |  |  |
| --- | --- | --- | --- |
| P\_ID | D\_ ID | P\_DISEASE | D\_COST |
| P226 | D014 | {(lung cancer, [0.3, 0.6]), (tuberculosis, [0.4, 0.7])} | {($30, [0.3, 0.4]), ($35, [0.6, 0.7])} |
| P228 | D016 | {(lung cancer, [1, 1])} | {($30, [1, 1])} |
| P262 | D022 | {(dyspepsia, [1, 1])} | {($5, [1, 1])} |
| P255 | D020 | {(hepatitis, [0.58, 0.96]), (pancreatitis, [0.2, 0.7]),  (cholecystitis, [0.2, 0.6])} | {($7, [0.2, 0.4]), ($8, [0.76, 1.0])} |

Chúng ta lưu ý rằng bộ *t*2 trong Bảng 4.6.1 và bộ *t*2 trong Bảng 4.6.2 có cùng giá trị khóa nên được kết hợp thành bộ *t*4 theo chiến lược tuyển xác suất độc lập ⊕*in* trong bảng 4.6.2.

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

**Định nghĩa 4.6.3** Giả sử *R* = (***U***, ℘) là lược đồ IPRDB, *r*1 và *r*2 là hai quan hệ trên *R*, *K* là một khóa của *R* và ⊖ là một chiến lược hiệu xác suất. *Phép trừ* của *r*1 và *r*2 theo ⊖, ký hiệu là *r*1 −⊖ *r*2, là quan hệ xác suất *r* trên *R* được xác định bởi *r* = {*t*1 ∈ *r*1 | ∀*t*2 ∈ *r*2, *t*1[*K*] ≠ *t*2[*K*]}∪{*t* | *t*.*A* = *t*1.*A* ⊖ *t*2.*A*, *t*1 ∈ *r*1, *t*2 ∈ *r*2, *A* ∈ ***U*** sao cho *t*1[*K*] = *t*2[*K*]}.

## Tính chất của các phép toán đại số

Các phép toán đại số trong IPRDB không chỉ được định nghĩa dựa trên các cơ sở toán học vững chắc mà còn nhất quán với mô hình dữ liệu IPRDB. Thông qua các ví dụ cụ thể, các phép toán đại số IPRDB đã chứng tỏ khả năng thao tác, tính toán, truy vấn dữ liệu không chắc chắn, được biểu diễn bởi mô hình IPRDB. Tương tự như các phép toán đại số trên mô hình CSDL quan hệ truyền thống, các phép toán trên IPRDB 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 IPRDB vẫn duy trì được các tính chất của các phép toán trên CSDL quan hệ truyền thống (Phần 1.4). Điều này cho thấy IPRDB 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 IPRDB. 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 IPRDB. Gọi φ1 và φ2 là hai điều kiện chọn. Khi đó

σφ1(σφ2(*r*)) = σφ2(σφ1(*r*)) = σφ1∧ φ2(*r*) (1)

Với giả thiết trong phép chọn σφ1∧ φ2(*r*) các điều kiện chọn φ1 và φ2 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 = σφ2(*r*). Khi đó với mọi *t* ∈ *r* ta có:

σφ1(σφ2(*r*)) = {*t*∈*r*2 | *probR*,*r*2,*t* ⊨ φ1}

= {*t* ∈ *r* | (*probR,r ,t* ⊨ φ2)˄ (*probR*,*r*2,*t* ⊨ φ1)}

= {*t* ∈ *r* | (*probR,r ,t* ⊨ φ2)˄ (*probR,r,t* ⊨ φ1 )} (do *r*2 ⊆ *r* )

= {*t* ∈ *r* | (*probR,r ,t* ⊨ φ1 ˄ φ2} (Định nghĩa 4.2.4)

= σφ1∧ φ2(*r*)

Từ đó hệ thức σφ1(σφ2(*r*)) = σφ1∧φ2(*r*) được chứng minh. Hệ thức σφ2(σφ1(*r*))= σφ2∧ φ1(*r*) được chứng minh tương tự. Vì φ1˄φ2 ⟺ φ2˄φ1 (phép hội trên tập các điều kiện chọn xác suất cũng như trên mệnh đề có tính giao hoán), nên σφ1∧ φ2(*r*) = σφ2∧ φ1(*r*). Từ đó suy ra hệ thức σφ1(σφ2(*r*)) = σφ2(σφ1(*r*)) và do đó σφ1(σφ2(*r*)) = σφ2(σφ1(*r*)) = σφ1 ∧φ2(*r*). Ngoài ra, có thể chứng minh trực tiếp hệ thức σφ1(σφ2(*r*)) = σφ2(σφ1(*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 IPRDB, *r* là quan hệ trên *R*, ⊕ là một chiến lược tuyển xác suất, ***A*** và ***B*** là hai tập hợp con của các thuộc tính của *R*, ***A*** ⊆ ***B***. Khi đó

Π***A***⊕(Π***B***⊕(*r*)) = Π***A***⊕(*r*)(2)

**Chứng minh:** Vì ***A*** ⊆ ***B***, nên ***A***∩***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 Π***A***⊕(Π***B***⊕(*r*)) = Π***A***∩***B***⊕(*r*) = Π***A***⊕(*r*) theo ⊕. 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 IPRDB 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, ⊗là một chiến lược hội xác suất. Khi đó

*r*1 ⋈⊗ *r*2 = *r*2 ⋈⊗ *r*1 (3)

(*r*1 ⋈⊗ *r*2) ⋈⊗ *r*3 = *r*1 ⋈⊗ (*r*2 ⋈⊗ *r*3) (4)

Phương trình (3) và (4) chứng tỏ phép kết của các quan hệ IPRDB 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 IPRDB) 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 giá trị 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⋈⊗ *r*2 *= r*2⋈⊗ *r*1. Tính giao hoán của phép kết được chứng minh.

Hệ thức (4): Do các lược đồ *R*1*, R*2 và *R*3được 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 giá trị 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 IPRDB. Vì vậy, hệ thức (*r*1⋈⊗ *r*2) ⋈⊗ *r*3 *= r*1⋈⊗(*r*2⋈⊗ *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 *R3* là các lược đồ IPRDB 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× *r*2 *= r*2× *r*1 (5)

(*r*1× *r*2)× *r*3 *= r*1×(*r*2× *r*3) (6)

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

**Định lý 4.7.4** Giả sử *R* là một lược đồ IPRDB, *r*1, *r*2và *r*3 là các quan hệ xác suất trên *R*. Gọi ⊗ và ⊕ tương ứng là các chiến lược hội và tuyển xác suất. Khi đó

*r*1∩⊗ *r*2 = *r*2∩⊗ *r*1(7)

(*r*1 ∩⊗ *r*2)∩ ⊗ *r*3*= r*1 ∩⊗(*r*2 ∩⊗*r*3) (8)

*r*1 ∪⊕ *r*2 = *r*2∪⊕ *r*1(9)

(*r*1∪⊕ *r*2) ∪⊕ *r*3*= r*1∪⊕ (*r*2∪⊕ *r*3) (10)

Các hệ thức (7), (8), (9) và (10) cho thấy các phép toán giao và hợp các quan hệ IPRDB 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 giá trị xác suất có tính giao hoán và kết hợp (Định nghĩa2.4.1). Do vậy, ta có giao của các quan hệ relations *r*1, *r*2, and *r*3 theo chiến lược hội xác suất ⊗ và mọi khóa được chọn có tính giao hoán và kết hợp. Từ đó, theo định nghĩa 4.6.1 suy ra các hệ thức (7) và (8).

Hệ thức (9) và (10): Từ tính giao hoán và kết hợp của phép hợp các tập hợp suy ra các chiến lược hợp các giá trị xác suất có tính giao hoán và kết hợp (Định nghĩa2.4.2). Do vậy, ta có hợp của các quan hệ relations *r*1, *r*2, and *r*3 theo chiến lược tuyển xác suất ⊕ và mọi khóa được chọn có tính giao hoán và kết hợp. Từ đọ, theo định nghĩa 4.6.2, chúng ta có các hệ thức (9) và (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 IPRDB, để thấy tính khả áp dụng của IPRDB 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 IPRDB. 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 *V* là tập giá trị của các thuộc tính bộ trong một quan hệ IPRDB (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 *V* là tập giá trị 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*.

## Kết luận

Trong Chương 4 này, tất cả các phép toán đại số quan hệ xác suất cơ bản trên mô hình IPRDB 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 IPRDB

## Giới thiệu

Chương này trình bày ý tưởng và phương pháp thực hiện cho mô hình IPRDB 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 IPRDB 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 IPRDB -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 IPRDB -SQLite bao gồm yêu cầu chung đối với hệ thống và mô hình kiến trúc tổng quan của hệ quản trị. Phần 6.4 trình bày các bước hiện thực cho khối biểu diễn mô hình IPRDB bao gồm lược đồ, quan hệ, thuộc tính, kiểu dữ liệu, giá trị 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 IPRDB -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.

## Các tính năng đặc trưng của SQLite

### 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 IPRDB -SQLite.

### 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.

### 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 IPRDB -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.

## Thiết kế tổng quan hệ quản trị IPRDB -SQLite

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

**Các yêu cầu của hệ thống**

Mỗi lược đồ IPRDB bao gồm một tập các thuộc tính đôi một khác nhau. Mỗi lược đồ IPRDB thể hiện thông tin về các thuộc tính của các đối tượng trong CSDL. Để thể hiện lược đồ IPRDB 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 IPRDB trên một lược đồ quan hệ *R* là một tập hữu hạn các bộ trên tập các thuộc tính của *R*. Mỗi quan hệ IPRDB cũng chính là một thể hiện của lược đồ IPRDB tương ứng. Để hiện thực các quan hệ IPRDB 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 IPRDB 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 IPRDB 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 IPRDB, 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 IPRDB 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 liểu 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 nghỉa của các thuộc tính kết hợp với nhau biểu diễn các bộ của các quan hệ . Kiểu giá trị nói chung trong IPRDB 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] (upperBound >= lowerBound).

Giá trị kiểu bộ trong IPRDB 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 IPRDB. chúng cho phép thực hiện các truy vấn và thao tác trên các quan hệ của IPRDB. 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 IPRDB, 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 ⊗, ⊕, ⊖ 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 (chiếu, 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 IPRDB gồm:

1. Biểu diễn và lưu trữ các phần tử được định nghĩa từ các khái niệm trong mô hình IPRDB, bao gồm lược đồ, quan hệ IPRDB, 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.

## Kiến trúc tổng quan của hệ quản trị IPRDB -SQLite

Từ những yêu cầu đặc tả bên trên, hệ quản trị IPRDB 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 IPRDB-SQLite.

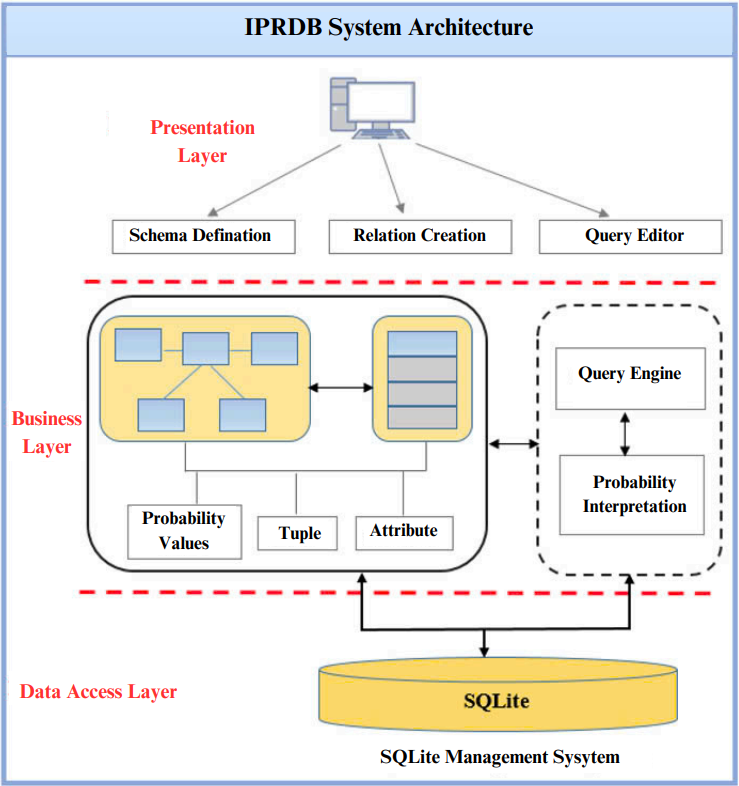
Kiến trúc hệ quản trị IPRDB được chia làm ba tầng:

1. *Tầng giao diện* (GUI 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 trên một quan hệ IPRDB (Query Editor).
2. *Tầng xử lý logic* (Business Layer): gồm hai khối chính, một khối chịu trách nhiệm về biểu diễn cho mô hình IPRDB, và một khối chịu trách nhiệm thực hiện phép truy vấn. Cụ thể, khối thứ nhất sẽ đóng vai trò xây dựng “cơ sở hạ tầng”, định ra cấu trúc dữ liệu của lược đồ, của các kiểu thuộc tính và của sự phân bố xác suất trên giá trị thuộc tính. Khối thứ hai sẽ đóng vai trò như một “trình biên dịch”, thực hiện biên dịch và thực thi câu truy vấn, cũng như thực hiện các phép diễn dịch xác suất phục vụ cho việc truy vấn.
3. *Tầng truy cập dữ liệu* (Data Access Layer): chính là hệ quản trị SQLite. Tầng này cung cấp các chức năng lưu trữ và truy xuất dữ liệu khi cần thiết. Nói chung, tầng này chỉ thao tác với dữ liệu bên dưới ngoài ra nó không làm gì khác.

Việc xây dựng hệ quản trị IPRDB sẽ nằm ở hai tầng trên trong kiến trúc này. Hệ thống IPRDB-SQLite phân theo chức nằng gồm hai khối: khối biểu diễn mô hình IPRDB và khối thực thi truy vấn IPRDB. Mỗi khối sẽ thực hiện xuyên suốt từ việc trình bày giao diện đến cấu trúc lưu trữ và xử lý bên dưới:

Khối biểu diễn mô hình IPRDB: gồm các lớp biểu diễn các phần tử lưu trữ bên dưới hệ thống cũng như các phần tử tương tác bên trên giao diện cho các lược đồ, các quan hệ, các thuộc tính, kiểu dữ liệu, giá trị xác suất,…

Khối thực hiện truy vấn IPRDB: khối này cung cấp lớp cho phép nhân phát biểu truy vấn, phân tích cú pháp và thực thi nó. Việc thực thi truy vấn cần một hệ thống các hàm diễn dịch xác suất, các hàm thực hiện các chiến lược kết hợp xác suất. Với giao diện thân thiện, dễ sử dụng, hệ thống IPRDB có thể biểu diễn và xử lý dữ liệu của mô hình IPRDB.



Hình 5.4.1 Kiến trúc hệ quản trị IPRDB

Các ý tưởng và kỹ thuật chủ yếu để thiết kế các khối biểu diễn mô hình và thực hiện truy vấn IPRDB trong hai tầng bên trên của kiến trúc hệ quản trị IPRDB lần lượt được trình bày dưới đây.

## Hiện thực khối biểu diễn mô hình IPRDB

Đối với hệ quản trị IPRDB, tất cả các thành phần tạo nên cơ sở dữ liệu, bao gồm lược đồ, các thuộc tính, kiểu dữ liệu, kiểu bộ, hoặc các phân bố xác suất, đều được người dùng nhập vào từ môi trường trực quan. Vì vậy, cần chuyển đổi các thông tin này thành các đối tượng có ý nghĩa trong hệ thống. Điều này đòi hỏi hệ thống phải chuẩn bị sẵn các lớp biểu diễn cho các kiểu dữ liệu phức tạp, các giá trị xác suất, khoảng xác suất, v.v., để biên dịch thành các lớp đối tượng phục vụ việc quản lý dữ liệu IPRDB cũng như các kiểu thuộc tính.

Đầu tiên, hệ thống cần cung cấp một giao diện cho phép người dùng tạo lược đồ bằng cách nhập vào các thuộc tính và kiểu dữ liệu tương ứng. Để hiển thị tập các thuộc tính mà người dùng nhập vào, chúng tôi sử dụng điều khiển DataGridView. Đây là một công cụ dạng lưới, cho phép người dùng nhập đồng thời tên thuộc tính, kiểu dữ liệu của thuộc tính và các thông tin mô tả khác trên cùng một dòng.

Các thông tin đặc tả liên quan đến lược đồ, quan hệ, và thuộc tính sau khi được tạo ra sẽ được lưu lại dưới dạng tệp văn bản nhờ hệ quản trị cơ sở dữ liệu SQLite. Việc xây dựng lược đồ IPRDB, quan hệ IPRDB, và các thuộc tính tương ứng đòi hỏi sự thể hiện của các lớp được mô tả trong các phần tiếp theo.

### Biểu diễn các kiểu trong IPRDB

Trong IPRDB, có hai kiểu chính được định nghĩa: kiểu cơ sở và kiểu bộ.

Kiểu cơ sở:

Kiểu cơ sở trong mô hình IPRDB là các kiểu cơ bản, không có cấu trúc, bao gồm các kiểu như integer, string hoặc kiểu liệt kê do người dùng tự định nghĩa. Trong quá trình hiện thực, các kiểu này có thể được hiểu là các kiểu có sẵn do ngôn ngữ lập trình cung cấp hoặc các kiểu do người dùng định nghĩa.

Hệ quản trị IPRDB hỗ trợ các kiểu có sẵn như Integer, Double, Boolean, String, DateTime, Bit, Decimal, v.v. Đối với các kiểu do người dùng định nghĩa, hệ thống cung cấp cơ chế để người dùng đặt tên và xác định miền giá trị tương ứng của kiểu. Sau khi được định nghĩa, các kiểu này sẽ được chuyển đổi và lưu vào hệ thống, có thể sử dụng tương tự như các kiểu Integer, String, v.v.

Trong quá trình hiện thực, một giao diện nhập liệu đã được xây dựng, cho phép người dùng định nghĩa kiểu dữ liệu riêng với tên gọi và miền giá trị tương ứng. Bên cạnh đó, một lớp **ProbDataType** được phát triển để chứa danh sách giá trị kiểu chuỗi, đảm bảo hệ thống ghi nhận đầy đủ các giá trị được người dùng nhập vào.

Kiểu bộ:

Kiểu bộ trong IPRDB là tập hợp các kiểu thuộc tính liên quan đến bộ. Trong các tác vụ liên quan đến các bộ trong quan hệ, kiểu bộ thường được sử dụng để chỉ định giá trị cụ thể của các thuộc tính trên bộ nhằm thực hiện các phép so sánh hoặc tính toán.

Ngoài ra, kiểu tập hợp có dạng {τ} cũng được xem là một kiểu bộ, trong đó các thuộc tính bên trong có cùng kiểu dữ liệu.

### Biểu diễn mô hình IPRDB

Như đã đề cập ở trên, chúng ta sẽ xây dưng các lớp để lưu trữ và thể hiện các đối tượng thuộc tính, quan hệ, … của hệ thống như sau:

### Lớp ProbSchema

Đây là lớp hỗ trợ lưu trữ các lược đồ quan hệ trong IPRDB. Lớp Schema là một lớp cơ sở của hệ quản trị IPRDB-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 IPRDB-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 trong cơ sở dữ liệu.
2. public string SchemeName { get; set; } : tên lược đồ quan hệ.
3. public List<ProbAttribute> Attributes { get; set; } danh sách các thuộc tính của lược đồ.

Các phương thức chính:

* + - 1. public ProbScheme(int IDScheme,string schemename, List<ProbAttribute> Attributes): phương thức khởi tạo đối tượng với các tham số được truyền vào.
      2. internal List<int> ListIndexPrimaryKey(): lấy danh sách chỉ số các thuộc tính là khóa chính trong lược đồ quan hệ.
      3. internal bool IsInherited(List<ProbRelation> Relations): kiểm tra lược đồ quan hệ đã có quan hệ nào thừa kế hay chưa.
      4. internal List<string> ListOfAttributeNameToLower(): lấy danh sách tên của thuộc tính theo trường hợp chữ thường.
      5. internal List<ProbScheme> GetAllScheme(): lấy tất cả lược đồ quan hệ từ cơ sở dữ liệu.
      6. internal ProbScheme GetSchemeById(): lấy lược đồ quan hệ từ SQLite theo IDScheme.
      7. internal void DeleteAllScheme(): xóa tất cả các lược đồ quan hệ trong cơ sở dữ liệu.
      8. internal void Insert(): thêm lược đồ quan hệ vào trong cơ sở dữ liệu.
      9. internal void Update(): cập nhật lại dữ liệu của lược đồ quan hệ trong cơ sở dữ liệu.
      10. internal void DeleteSchemeById(): xóa lược đồ quan hệ theo IDScheme có trong cơ sở dữ liệu.
      11. **Lớp ProbRelation**

Đây là lớp hỗ trợ lưu trữ các quan hệ. Nhiệm vụ chủ yếu của lớp này là cho phép chương trình ánh xạ các dữ liệu được nhập trên điều khiển DataGridView của giao diện vào các bộ dữ liệu của quan hệ và cập nhật các quan hệ xuống cơ sở dữ liệu. Mỗi đối tượng của lớp ProbRelation gồm có tên quan hệ, tập các thuộc tính của quan hệ, một danh sách các bộ 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 quan hệ trong trong cơ sở dữ liệu.
2. public string RelationName { get; set; } : tên quan hệ.
3. public ProbScheme Scheme { get; set; }: lược đồ quan hệ tương ứng.
4. public List<ProbTuple> tuples { get; set; }: danh sách các bộ của quan hệ.

Các phương thức chính:

* + - 1. public ProbRelation(int IDRelation,string relationName, List<ProbTuple> tuples,ProbScheme scheme): phương thức khởi tạo đối tượng với các tham số được truyền vào.
      2. internal List<ProbRelation> GetAllRelation(): lấy danh sách quan hệ trong cơ sở dữ liệu.
      3. internal void DeleteAllRelation(): xóa tất cả các quan hệ trong bảng SystemRelation trong cơ sở dữ liệu.
      4. internal void DropTableByRelationName(): xóa quan hệ theo tên của quan hệ trong cơ sở dữ liệu.
      5. internal void CreateTableRelation(): tạo mới quan hệ trong cơ sở dữ liệu.
      6. internal void InsertTupleIntoTableRelation(): thêm danh sách bộ quan hệ xuống cơ sở dữ liệu.
      7. internal void DeleteRelationById(): xóa quan hệ trong cơ sở dữ liệu theo IDRelation.
      8. **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 IPRDB-SQLite. Mỗi đối tượng của lớp ProbAtribute 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 của thuộc tính trong cơ sở dữ liệu.
2. public string AttributeName { get; set; }: tên thuộc tính.
3. public string Description { get; set; }: miêu tả về thuộc tính.
4. public ProbScheme probScheme { get; set; }: miêu tả thuộc tính thuộc về lược đồ quan hệ nào.
5. public ProbDataType Type { get; set; }: kiểu dữ liệu thuộc tính.

Các phương thức chính:

1. public ProbAttribute(ProbAttribute attr): phương thức khởi tạo đối tượng với tham số được truyền vào là đối tượng ProbAttribute .
2. internal List<ProbAttribute> getListAttributeByIDScheme(int IDScheme): lấy danh sách các Attribute theo IDScheme trong cơ sở dữ liệu.
3. internal void DeleteAllAttribute(): xóa tất cả các Attribute trong cơ sở dữ liệu.
4. internal void Insert(): thêm Attribute vào cơ sở dữ liệu.
5. internal void DeleteAllAttributeByIdScheme: xóa tất cả các Attribute theo IdScheme trong cơ sở dữ liệu.
   * + 1. **Lớp ProbDataType**

Như đã trình bày ở chương 3, cơ sở dữ liệu quan hệ khoảng xác suất có 3 dạng kiểu dữ liệu đó là: kiểu cơ bản, kiểu tập hợp và kiểu bộ. Tuy nhiên, một kiểu tập hợp có dạng {τ} cũng chính là một kiểu bộ với các thuộc tính bên trong có kiểu dữ liệu giống nhau. 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 IPRDB-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 kiểu dữ liệu (kiểu dữ liệu do người dùng tự định nghĩa).
2. public string DataType { get; set; }: kiểu dữ liệu cơ bản.
3. public List<string> Domain { get; set; }: miền giá trị cho kiểu dữ liệu mà người dùng tự định nghĩa.

Các phương thức chính:

1. public ProbDataType(ProbDataType type): phương thức khởi tạo đối tượng với tham số được truyền vào là đối tượng ProbDataType.
2. public bool CheckDomain(string value): phương thức kiểm tra tham số truyền vào có thuộc miền giá trị hay không.
3. public void GetDataType():phương thức trả về kiểu dữ liệu cơ bản của thuộc tính dựa vào tên kiểu dữ liệu.
   * + 1. **Lớp ProTriple**

Các giá trị xác suất là một kiểu giá trị đặc trưng trong mô hình IPRDB. Để hiện thực cho kiểu giá trị này thì trước hết ta cần xem xét lại thành phần cấu tạo của nó. Theo định nghĩa đã được trình bày ở chương 3 thì một giá trị xác suất có dạng {(*vi*, [*li*, *ui*]) trong đó *vi* là tập giá trị của thuộc tính, *li* và *ui* là xác suất cận dưới và xác suất cận trên của tập giá trị thuộc tính.

Để biểu diễn giá trị xác suất cho hệ quản trị IPRDB-SQLite chúng tôi đã xây dựng lớp ProbTriple với các thuộc tính và phương thức như sau:

Các thuộc tính chính:

1. public List<object> Values { get; set; }:danh sách các giá trị của thuộc tính.
2. List<double> MinProbs { get; set; }: danh sách các giá trị xác suất cận dưới, tương ứng với các giá trị trong danh sách Values.
3. public List<double> MaxProbs { get; set; }: danh sách các giá trị xác suất cận trên, tương ứng với các giá trị trong danh sách Values.

Các phương thức chính:

1. public ProbTriple(string value): phương thức khởi tạo đối tượng từ tham số truyền vào.
2. public string GetStrValue(): lấy chuỗi giá trị (kết hợp tập các giá trị với tập các khoảng xác suất).
3. public bool IsProbTripleValue(string Value): kiểm tra tham số truyền vào có phải là các giá trị xác suất hay không.
   * + 1. **Lớp ProbTuple**

Giá trị bộ xác suất trong IPRDB chính là một tập các giá trị xác suất tương ứng của các thuộc tính. Giá trị bộ xác suất trong IPRDB-SQLite được hiện thực bằng một lớp ProbTuple có chứa danh sách các giá trị xác suất:

Các thuộc tính chính:

1. public List<ProbTriple> Triples { get; set; }: danh sách các giá trị xác suất trên một bộ của quan hệ.

Các phương thức chính:

1. public ProbTuple(ProbTuple tuple): phương thức khởi tạo đối tượng từ tham số truyền vào.
2. internal List<ProbTuple> getAllTypleByRelationName(string relationname, int nTriples): lấy tất cả các bộ xác suất trong cơ sở dữ liệu theo tên quan hệ.

## Hiện thực khối xử lý

Khối xử lý truy vấn trong hệ quản trị IPRDB-SQLite được xây dựng trên lớp SelectionExpression. 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 IPRDB 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 IPRDB. Như vậy, một truy vấn với cú pháp dạng SQL mới này trong IPRDB-SQLite tương đương với một truy vấn chọn với điều kiện chọn tương ứng trong IPRDB. Điều khác biệt của truy vấn SQL mới trong IPRDB-SQLite so với truy vấn SQL thông thường là trong mỗi truy vấn chọn trên IPRDB cho phép truy vấn trên nhiều quan hệ xác suất được tham chiếu với các điều kiện chọn có dạng {(*vi*, [*li*, *ui*]). 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.

### Xử lý truy vấn chọn

Tác vụ này được tiến hành bởi các phương thức được xây dựng trong lớp SelectionExpression bao gồm:

1. private List<ProbAttribute> GetAttribute(string valueString): lấy ra tập các thuộc tính được chọn trong truy vấn. Nếu người dùng đặt kí tự “\*” trong truy vấn thì chương trình sẽ lấy toàn bộ các thuộc tính trong quan hệ được chọn.
2. private List<ProbRelation> GetAllRelation(string valueString): lấy ra các quan hệ được chọn trong truy vấn.
3. private static string GetCondition(string valueString): trả về chuỗi điều kiện chọn trong truy vấn
4. private ProbRelation Descartes(): phương thức thực hiện tích Descartes trên hai quan hệ.
5. private ProbRelation NaturalJoin(): phương thức thực hiện việc kết tự nhiên trên hai quan hệ.
6. private static bool CheckStringAttribute(string stringAttribute): phương thức kiểm tra sự hợp lệ của thuộc tính trong câu truy vấn.
7. private bool CheckStringQuery(string stringQuery): phương thức kiểm tra sự hợp lệ của câu lệnh truy vấn.
8. private static bool CheckCondition(string conditionString): phương thức kiểm tra sự hợp lệ của điều kiện chọn trong câu truy vấn.
9. internal bool ExecuteQuery(): thực thi câu truy vấn, kết quả trả ra là một danh sách các bộ thỏa mãn điều kiện chọn trong quan hệ được chọn.

Trong phần tiếp theo chúng tôi sẽ trình bày cách thức xử lý các biểu thức chọn của chương trình.

### Xử lý điều kiện chọn

Sau khi phương thức ExecuteQuery() của lớp QueryExecution 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) của lớp SelectionCondition 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 QueryExecution. 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 CalculateTotalCondition của lớp SelectCondition 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 CalculationCon() 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” → “⊗\_in”

“⊗ig” → “⊗\_ig”

“⊗me” → “⊗\_me”

“⊕in” → “⊕\_in”

“⊕ig” → “⊕\_ig”

“⊕me” → “⊕\_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 phần 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 🡺 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 🡺 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 🡺 tính toán giá trị khoảng xác suất kết hợp của hai khoảng trên dựa theo chiến lược xác suất đã chọn.

1. 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.

## Giao diện người dùng

Trong phần này, chúng tôi sẽ trình bày về giao diện của hệ quản trị IPRDB-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.

### Giao diện chính

A screenshot of a computer

Description automatically generated

Hình 5.7.1.1 Giao diện chính của hệ thống IPRDB

Giao diện chính của chương trình được thiết kế như Hình 5.7.1.1 cho phép người sử dụng có thể tạo mới quan hệ IPRDB hoặc tạo mới lược đồ IPRDB. Một quan hệ IPRDB có thể được tạo mới từ một cơ sở dữ liệu IPRDB có sẵn hoặc từ một cơ sở dữ liệu IPRDB mới hoàn toàn.Để tạo mới một cơ sở dữ liệu mới thì người dùng chọn chức năng “New Database” trên thanh trình đơn Database để tạo mới. Nếu quan hệ IPRDB được tạo từ một cơ sở dữ liệu có sẵn, thì người dùng phải sử dụng chức năng “Open Existing Database” trên thanh trình đơn Database để mở cơ sở dữ liệu này. Ứng với lược đồ Diagnose đã được xây dựng, ta có thể tạo nhiều quan hệ khác nhau thông qua giao diện.

A screenshot of a computer

Description automatically generated

Hình 5.7.1.2 Giao diện chương trình sau khi tạo một cơ sở dữ liệu mới

### Giao diện lược đồ quan hệ

Sau khi đã mở ra một cơ sở dữ liệu trong hệ quản trị, ta có thể tạo mới lược đồ quan hệ. Để tạo mới lược đồ quan hệ, trước tiên ta chọn tab Schema và chọn chức năng “New Schema” và nhập tên cho lược đồ và tên thuộc tính, kiểu dữ liệu thuộc tính, mô tả về thuộc tính và chọn khóa chính cho quan hệ. Giao diện tạo lược đồ quan hệ như Hình 5.7.2.1.

Để nhập được kiểu dữ liệu cho thuộc tính thì ta phải nhấp đúp vào ô kiểu dữ liệu tương ứng của field DataType. Một Form chọn kiểu dữ liệu sẽ hiện ra. Nếu người dùng chọn những kiểu dữ liệu thông thường như Int16, Int32, string, double,…thì chương trình sẽ tự động điền tên dữ liệu và trường giá trị của dữ liệu vào bảng. Còn nếu người dùng chọn kiểu dữ liệu UserDefined thì phải nhập thêm tên kiểu dữ liệu và trường giá trị cho kiểu dữ liệu đó. Hình 5.7.2.2 chỉ ra cách chọn kiểu dữ liệu cho các thuộc tính đã nhập.

A screenshot of a computer

Description automatically generated

Hình 5.7.2.1 Giao diện tạo lược đồ quan hệ

A screenshot of a computer

Description automatically generated

Hình 5.7.2.2 Màn hình chọn kiểu dữ liệu cho thuộc tính

Cách nhập trường giá trị cho kiểu dữ liệu UserDefined như Hình 5.7.2.3 là người dùng sẽ nhập mỗi giá trị trên một dòng trong MemoEdit, chương trình sẽ tự động kiểm tra lỗi cho mỗi kí tự nhập vào để tránh trường hợp người dùng nhập các kí tự đặc biệt. Lưu ý rằng, trường giá trị cho kiểu dữ liệu UserDefined không phân biệt chữ hoa hay chữ thường.

A screenshot of a computer

Description automatically generated

Hình 5.7.2.3 Màn hình nhập danh sách giá trị cho kiểu dữ liệu UserDefined

### Giao diện tạo và nhập dữ liệu cho quan hệ

Sau khi đã mở ra một cơ sở dữ liệu trong hệ quản trị và tạo mới một lược đồ quan hệ ta có thể tạo mới quan hệ. Để tạo mới quan hệ, trước tiên ta chọn tab Relation và chọn chức năng “New Relation” và nhập tên cho quan hệ và chọn lược đồ quan hệ cho quan hệ muốn tạo mới. Màn hình tạo quan hệ như Hình 5.7.3.1.

A screenshot of a computer

Description automatically generated

Hình 5.7.3.1 Màn hình tạo mới một quan hệ

Sau khi đã thiết kế lược đồ quan hệ cho cơ sở dữ liệu, ta có thể tiến hành nhập giá trị dữ liệu cho quan hệ đó. Giao diện nhập giá trị dữ liệu cho các quan hệ được thiết kế như Hình 5.6.3.2. Để nhập giá trị dữ liệu, ta nhấp chọn quan hệ cần nhập dữ liệu. Giá trị dữ liệu của quan hệ sẽ được nhập vào các ô của GridViewData. Đối với mỗi ô dữ liệu, giá trị có định dạng {(*vi*, [*li*, *ui*]), (*vi+1*, [*li+1*, *ui+1*]), …} cho từng thuộc tính.

A screenshot of a computer

Description automatically generated

Hình 5.7.3.2 Màn hình nhập liệu cho quan hệ

### Truy vấn dữ liệu trên quan hệ

Sau khi đã tạo được quan hệ và lược đồ quan hệ trên cơ sở dữ liệu, ta có thể thực hiện các phép truy vấn trên quan hệ đó. Để thực hiện việc truy vấn, trước hết ta chuyển sang cửa sổ truy vấn bằng cách chọn vào nút Tab Query trên của giao diện chương trình. Để có thể truy vấn thì trước tiên ta phải tạo mới 1 câu lệnh query bằng cách sử dụng chức năng “New Query”. Giao diện tạo mới 1 query như Hình 5.7.4.1.

**A screenshot of a computer

Description automatically generated**

Hình 5.7.4.1 Màn hình tạo mới query

Trong quá trình truy vấn, nếu gặp biểu thức chọn có các phép toán kết hợp như ⊗, ⊕, ⊖ với các chiến lược kết hợp xác suất thì ta có thể chèn các phép toán này vào câu truy vấn bằng cách chọn các thành phần này trên thanh chức năng như Hình 5.7.4.4.

A screenshot of a computer

Description automatically generated

Hình 5.7.4.2 Màn hình chọn và chèn chiến lược vào câu truy vấn

**A close-up of a computer screen

Description automatically generated**

Hình 5.7.4.3 Giao diện nhập nội dung câu truy vấn

Câu lệnh truy vấn có thể được thực thi bằng việc sử dụng phím tắt “F5” hoặc nút “Exute Query”. Kết quả thực thi của câu lệnh truy vấn sẽ được hiển thị bên dưới cửa sổ “Query result” như trong hình 5.7.4.3.

A screenshot of a computer screen

Description automatically generated

Hình 5.7.4.4 Màn hình hiển thị kết quả thực thi câu truy vấn

Trong trường hợp, truy vấn không có kết quả, cửa sổ “Messages…” sẽ hiện ra với thông báo “No tuple satisfies the condition”. Trường hợp, câu lệnh truy vấn do người dùng nhập không đúng cú pháp thì cửa sổ “Messages…” sẽ hiện ra với thông báo lỗi để báo cho người dùng nhận ra câu lệnh truy vấn đã nhập sai ở đâu.

## Kết luận

Trong chương 5 này, chúng tôi đã trình bày tất cả các bước để thiết kế, xây dựng và hiện thực hệ quản trị IPRDB-SQLite. Trong đó, nội dung của quá trình hiện thực được chia làm hai phần: hiện thực khối biểu diễn dữ liệu và hiện thực khối xử lý truy vấn chọn dựa trên nền tảng lý thuyết của mô hình IPRDB trong các chương trước. Trong phần cuối của chương, chúng tôi cũng đã trình bày cách thức sử dụng đối với hệ quản trị IPRDB-SQLite, trong đó nhấn mạnh đến việc thực thi các câu truy vấn đối với các quan hệ trong CSDL xác suất. Việc hiện thực thành công hệ quản trị IPRDB-SQLite cho mô hình IPRDB chứng tỏ triển vọng và tính khả áp dụng vào thực tế của mô hình này.

# TỔNG KẾT VÀ ĐỀ NGHỊ

## 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 IPRDB được trình bày trong đề tài khóa luận này là một đóng góp mới bằng cách tích hợp giá trị xác suất (vào mô hình CSDL truyền thống) như là một cách biểu diễn mới của giá trị thuộc tính quan hệ, cho phép thuộc tính có giá trị không chắc chắn trong mô hình IPRDB. 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 IPRDB 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 giá trị 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 có giá trị không chắc chắn và giá trị bộ xác suất.
3. Kế đến, các phép toán đại số trên IPRDB 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. Các tính chất của các phép toán đại số trên IPRDB đã được chứng minh cho thấy quá trình mở rộng CSDL quan hệ thành IPRDB là đúng đắn.
5. Một phần mềm IPRDB -SQLite như là một hệ quản trị CSDL của IPRDB với ngôn ngữ truy vấn tựa SQL đã được phát triển. Bước đầu cho thấy hệ thống này có thể biểu diễn và thực thi một cách hiệu quả các truy vấn chọn trên các quan hệ có giá trị thuộc tính không chắc chắn.

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, các phép toán đại số quan hệ xác suất và hệ quản trị IPRDB -SQLite cho thấy rõ hơn bản chất và cách thức ứng dụng của IPRDB.

## 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. IPRDB -SQLitechưa được hiện thực đầy đủ như một hệ quản trị CSDL để cho phép tiến tới ứng dụng mô hình IPRDB vào thực tế. Do đó đề nghị đầu tiên là xây dựng một hệ quản trị CSDL cho IPRDB 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 IPRDB 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 IPRDB.

# 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. R. Elmasri and S.B. Navathe, *Fundamentals of Database Systems*, *6th edition, Addison-Wesley*, 2015.
3. D. Dey and S. Sarkar, “A probabilistic relational model and algebra,” *ACM Transactions on Database Systems*, vol.21, no.3, pp.339-369, 1996.
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. 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.
6. V.V. Kheradkar and S. K. Shirgave, “Query processing over relational cross model in uncertain and probabilistic databases,” *Proceedings of 3Th International Conference on Artificial Intelligence and Smart Energy*, Coimbatore, India, pp.763-769, 2023.
7. 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.
8. 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.
9. Z. Ma and L. Yan, *Advances in probabilistic databases for uncertain information management*, Springer-Verlag Berlin Heidelberg, 2013.
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. 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.
12. H. Debbi, “Explaining query answers in probabilistic databases,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol.8, no.4, pp.140-152, 2023.
13. 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.
14. H. Nguyen, “Extending relational database model for uncertain information,” *Journal of Computer Science and Cybernetics*, vol.35, no.4, pp.355-372, 2019.
15. W. Zhao, A. Dekhtyar, and J. Goldsmith, “Databases for interval probabilities,” *International Journal of Intelligent Systems*, vol.19, no.9, pp.789-815, 2004.
16. R. Ross and V.S. Subrahmanian, “Aggregate operators in probabilistic databases,” *Journal of the ACM*, vol.52, no.1, pp.54-101, 2005.
17. 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.
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. S.K. Lee, “An extended relational database model for uncertain and imprecise information,” *Proceedings of 18th Conference on Very Large Data Bases*, Vancouver, Canada, pp.211-220, 1992.
20. H. Nguyen, “A probabilistic relational database model and algebra,” *Journal of Computer Science and Cybernetics*, vol.31, no.4, pp.305-321, 2015.
21. H. Nguyen, T.N. Nguyen, and T.T.N. Tran, “A probabilistic relational database model with uncertain multivalued attributes,” *ICIC Express Letters*, vol. 16, no.3, pp.241-248, 2022.
22. H. Nguyen, “Extending probabilistic relational database model with uncertain multivalued attributes,” *International Journal of Innovative Computing, Information and Control*, vol.18, no.5, pp.1477–1492, 2022.
23. H. Nguyen, D.N. Le, 2024, “A Relational Database Model with Interval Probability Valued Attributes for Uncertain and Imprecise Information,” *ECTI Transactions on Computer and Information Technology,* vol.18, no.3, pp.307- 318.
24. SQLite, “SQLite Home Page,” http://www.sqlite.org, accessed Jan. 10, 2025.