Introduction to Databases

(INFR10080)

(Relational Calculus)

Instructor: Yang Cao

(Fall 2025)

THE UNIVERSITY of EDINBURGH

Changelog

v25.0 Initial version

First-order logic

$$
\begin{aligned}
\textbf{term } t &:= x && \text{(variable)}\\
&\mid c && \text{(constant)}
\end{aligned}
$$

$$
\begin{aligned}
\text{formula } \varphi &:= P(t_1, \ldots, t_n)\\
&\mid t_1 \,\mathbf{op}\, t_2 \quad \text{with } \mathbf{op} \in \{=, \neq, >, <, \geqslant, \leqslant\}\\
&\mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \neg\varphi \mid \varphi_1 \rightarrow \varphi_2\\
&\mid \exists x\, \varphi \mid \forall x\, \varphi \qquad \text{if } x \in \mathbf{free}(\varphi)
\end{aligned}
$$

$\mathbf{free}(\varphi)$ = variables that are not in the scope of any quantifier

Notation

We write $\exists x_1 \exists x_2 \cdots \exists x_n\, \varphi$ as $\exists x_1, \ldots, x_n\, \varphi$
and $\forall x_1 \forall x_2 \cdots \forall x_n\, \varphi$ as $\forall x_1, \ldots, x_n\, \varphi$

We assume quantifiers bind till the end of the formula:

Example

$\exists x\, R(x) \wedge S(x)$ stands for $\exists x\big(R(x) \wedge S(x)\big)$
not for $\big(\exists x\, R(x)\big) \wedge S(x)$

Relational calculus

A **relational calculus query** is an expression of the form $\{\bar{x} \mid \varphi\}$ where the set of variables in $\bar{x}$ is **free**(φ)

Examples

- $Q = \{x, y \mid \exists z\, R(x, z) \land S(z, y)\}$
- $Q = \{y, x \mid \exists z\, R(x, z) \land S(z, y)\}$
- $Q = \{x, x \mid \forall y\, R(x, y)\}$

Queries without free variables are called **Boolean queries**

Examples

- $Q = \{() \mid \forall x\, R(x, x)\}$
- $Q = \{() \mid \forall x\, \exists y\, R(x, y)\}$

Data model

Relations (tables) are sets of **tuples** of the same length

Schema

- Set of **relation names**
- **Arity** (i.e., number of columns) of each relation name
  Note that columns are ordered but have no names

Instance

- Each relation name (from the schema) of arity k
  is associated with a k-ary relation

(i.e., a set of tuples that are all of length k)

Examples

Customer : ID, Name, Age

Account : Number, Branch, CustID

Q_1: *Name of customers younger than 33 or older than 50*

$$\{\ y \mid \exists x, z\ \text{Customer}(x, y, z) \land (z < 33 \lor z > 50)\ \}$$

Q_2: *Name and age of customers having an account in London*

$$\{\ y, z \mid \exists x\ \text{Customer}(x, y, z) \land \exists w\ \text{Account}(w, \text{'London'}, x)\ \}$$

Q_3: *ID of customers who have an account in* **every** *branch*

$$\{\ x \mid \exists y, z\ \text{Customer}(x, y, z) \land \big(\forall u, w, v\ \text{Account}(u, w, v) \to \exists u'\ \text{Account}(u', w, x)\big)\ \}$$

Answers to queries

A database instance can be seen as a semantic function in FOL

(mapping relation names to relations of appropriate arity)

$\boxed{\textbf{Recall}}$ An **assignment** ν maps variables to constants

(and each constant to itself)

$\boxed{\textbf{Notation}}$

For $\bar{x} = (x_1, x_2, \ldots, x_n)$, $\quad \nu(\bar{x}) = \big(\nu(x_1), \nu(x_2), \ldots, \nu(x_n)\big)$

The answer to a query $Q = \{\bar{x} \mid \varphi\}$ on a database D is

$$Q(D) = \{\ \nu(\bar{x}) \mid \nu \text{ is an assignment for which } D, \nu \models \varphi\ \}$$

The answer to a **Boolean query** is either $\{()\}$ (true) or $\varnothing$ (false)

Safety

A query is **safe** if it gives a finite answer on **all** databases
and this answer does not depend on the underlying set of constants

Examples of unsafe queries:

- $\{x \mid \neg R(x)\}$
- $\{x, y \mid R(x) \lor R(y)\}$
- $\{x, y \mid x = y\}$

Question: Are Boolean queries safe?

Bad news
Whether a relational calculus query is safe is **undecidable**

Active domain

$\mathbf{adom}(R) = \{$ all constants occuring in R $\}$

Example

$$\mathbf{adom}\left(\begin{array}{c|cc} R & A & B \\ \hline & a_1 & b_1 \\ & a_1 & b_2 \end{array}\right) = \{a_1, b_1, b_2\}$$

The active domain of a database D is

$$\mathbf{adom}(D) = \bigcup_{R \in D} \mathbf{adom}(R)$$

Active domain semantics

Evaluate queries within **adom**(D) $\Longrightarrow$ **safe relational calculus**

$$Q(D) = \{\ \nu(\bar{x}) \mid \nu \text{ is an assignment of constants in } \mathbf{adom}(D) \text{ for which } D, \nu \models \varphi\ \}$$

Interpretation of formulas under (D, ν)

$$
\begin{array}{lcl}
D, \nu \models P(t_1, \ldots, t_n) & \Longleftrightarrow & \big(\nu(t_1), \ldots, \nu(t_n)\big) \in P^D \\
D, \nu \models \neg\varphi & \Longleftrightarrow & D, \nu \not\models \varphi \\
D, \nu \models \varphi \wedge \psi & \Longleftrightarrow & D, \nu \models \varphi \text{ and } D, \nu \models \psi \\
D, \nu \models \varphi \vee \psi & \Longleftrightarrow & D, \nu \models \varphi \text{ or } D, \nu \models \psi \\
D, \nu \models \varphi \rightarrow \psi & \Longleftrightarrow & \text{if } D, \nu \models \varphi \text{ then } D, \nu \models \psi \\
D, \nu \models \forall x\, \varphi & \Longleftrightarrow & \text{for every } c \in \mathbf{adom}(D): \quad D, \nu[x/c] \models \varphi \\
D, \nu \models \exists x\, \varphi & \Longleftrightarrow & \text{there is } \quad c \in \mathbf{adom}(D) \text{ s.t. } D, \nu[x/c] \models \varphi
\end{array}
$$

Evaluation of quantifiers

$$D, \nu \models \exists x\, \varphi \quad \Longleftrightarrow \quad D, \nu \models \bigvee_{a \in \mathbf{adom}(D)} \varphi[x/a]$$

$$D, \nu \models \forall x\, \varphi \quad \Longleftrightarrow \quad D, \nu \models \bigwedge_{a \in \mathbf{adom}(D)} \varphi[x/a]$$

where $\varphi[x/a]$ denotes the formula obtained from φ
by replacing all **free** occurrences of x with a

Evaluation of quantifiers: Examples

Assume **adom**$(D) = \{1, 2, 3\}$

$$D, \nu \models \exists x\, R(x, y) \wedge S(x)$$

$$\Longleftrightarrow$$

$$D, \nu \models \big(R(1, y) \wedge S(1)\big) \vee \big(R(2, y) \wedge S(2)\big) \vee \big(R(3, y) \wedge S(3)\big)$$

$$D, \nu \models \forall x\, S(x) \rightarrow R(x, y)$$

$$\Longleftrightarrow$$

$$D, \nu \models \big(S(1) \rightarrow R(1, y)\big) \wedge \big(S(2) \rightarrow R(2, y)\big) \wedge \big(S(3) \rightarrow R(3, y)\big)$$