Relational Calculus

Relational Calculus

- More "declarative" than relational algebra
  - Foundation for query languages (such as SQL)
  - Relational algebra used more for physical operators
- Comes in two flavours: Tuple relational calculus (TRC) and Domain relational calculus (DRC)
  - TRC: Variables range over (i.e., get bound to) *tuples*
  - DRC: Variables range over *domain elements* (= field values)
  - Both TRC and DRC are simple subsets of first-order logic
- We will study DRC

Domain Relational Calculus

- *Query* has the form:
  $$\{\langle x1, x2, ..., xn\rangle \mid p(\langle x1, x2, ..., xn\rangle)\}$$
- *Answer* includes all tuples $\langle x1, x2, ..., xn\rangle$ that make the *formula* $p(\langle x1, x2, ..., xn\rangle)$ be *true*.
- Formula is recursively defined
  - Starting with simple *atomic formulas*
  - Logical connectives
  - Quantification

DRC Formulas

- *Atomic formula:*
  - $\langle x1, x2, ..., xn\rangle \in Rname$, or X *op* Y, or X *op* constant
  - *op* is one of $<, >, =, \leq, \geq, \neq$
- *Formula:*
  - an atomic formula, or … (more later)
- Example of query with atomic formula
  $$\{< I, N, T, A > \mid < I, N, T, A > \in Sailors\}$$
  - Equivalent relational algebra query?

DRC Formulas

- *Formula:*
  - an atomic formula, or
  - $\neg p, p \wedge q, p \vee q$, where p and q are formulas, or
  - (more later)
- Query using logical connectives
  $$\{< I, N, T, A > \mid < I, N, T, A > \in Sailors \wedge T > 7\}$$
  - Equivalent relational algebra query?
  - Find sailors who are older than 18 or have a rating under 9, and are called 'Joe'

DRC Formulas

- *Formula:*
  - an atomic formula, or
  - $\neg p, p \wedge q, p \vee q$, where p and q are formulas, or
  - $\exists X\,(p(X))$, where variable X is *free* in p(X), or
  - $\forall X\,(p(X))$, where variable X is *free* in p(X)
- The use of quantifiers $\exists X$ and $\forall X$ is said to bind X.
  - A variable that is not bound is free
- For query: $\{\langle x1, x2, ..., xn\rangle \mid p(\langle x1, x2, ..., xn\rangle)\}$
  - The variables x1, ..., xn that appear to the left of `|` must be the *only* free variables in the formula p(...).

Find sailors rated > 7 who've reserved boat #103

$$\{\langle I,N,T,A\rangle \mid \langle I,N,T,A\rangle \in Sailors \wedge T>7 \wedge$$
$$\exists\, Ir,Br,D\ (\langle Ir,Br,D\rangle \in \mathrm{Re}serves \wedge Ir=I \wedge Br=103)\}$$

- We have used $\exists\, Ir,\, Br,\, D\ (\ldots)$ as a shorthand for $\exists\, Ir\,(\exists\, Br\,(\exists\, D\,(\ldots)))$
- Note the use of $\exists$ to find a tuple in Reserves that `joins with' the Sailors tuple under consideration.

Find sailors rated > 7 who've reserved a red boat

$$\{\langle I,N,T,A\rangle \mid \langle I,N,T,A\rangle \in Sailors \wedge T>7 \wedge$$
$$\exists\, Ir,Br,D\ (\langle Ir,Br,D\rangle \in \mathrm{Re}serves \wedge Ir=I \wedge$$
$$\exists\, B,BN,C\,(\langle B,BN,C\rangle \in Boats \wedge B=Br \wedge C='red'))\}$$

- Observe how the parentheses control the scope of each quantifier's binding.
- This may look cumbersome, but with a good user interface, it is very intuitive.

Find sailors who've reserved all boats

$$\{\langle I,N,T,A\rangle \mid \langle I,N,T,A\rangle \in Sailors \wedge$$
$$\forall B,BN,C(\langle B,BN,C\rangle \in Boats \Rightarrow$$
$$(\exists\, Ir,Br,D\ (\langle Ir,Br,D\rangle \in \mathrm{Re}serves \wedge I=Ir \wedge Br=B)))\}$$

- Find all sailors *I* such that for each 3-tuple $\langle B,BN,C\rangle$ there is a tuple in Reserves showing that sailor *I* has reserved it.

Find sailors who've reserved all boats

$$\{\langle I,N,T,A\rangle \mid \langle I,N,T,A\rangle \in Sailors \wedge$$
$$\forall B,BN,C(\langle B,BN,C\rangle \in Boats \wedge$$
$$(\exists\, Ir,Br,D\ (\langle Ir,Br,D\rangle \in \mathrm{Re}serves \wedge I=Ir \wedge Br=B)))\}$$

- What is wrong with this?

Find sailors who've reserved all boats

$$\{\langle I,N,T,A\rangle \mid \langle I,N,T,A\rangle \in Sailors \wedge$$
$$\forall\, B,BN,C\,(\neg(\langle B,BN,C\rangle \in Boats) \vee$$
$$(\exists\, Ir,Br,D\ (\langle Ir,Br,D\rangle \in \mathrm{Re}serves \wedge I=Ir \wedge Br=B)))\}$$

- Find all sailors *I* such that for each 3-tuple $\langle B,BN,C\rangle$ either it is not a tuple in Boats or there is a tuple in Reserves showing that sailor *I* has reserved it.

Find sailors who've reserved all boats

$$\{\langle I,N,T,A\rangle \mid \langle I,N,T,A\rangle \in Sailors \wedge$$
$$\neg\exists B,BN,C(\langle B,BN,C\rangle \in Boats \wedge$$
$$\neg(\exists Ir,Br,D(\langle Ir,Br,D\rangle \in \mathrm{Re}serves \wedge I=Ir \wedge Br=B)))\}$$

- Find all sailors *I* such that there does not exist a 3-tuple $\langle B,BN,C\rangle$ in Boats for which there does not exist a tuple in Reserves showing that sailor *I* has reserved it.

Find sailors who've reserved all red boats

$$\{\langle I,N,T,A\rangle \mid \langle I,N,T,A\rangle \in Sailors \wedge$$
$$\forall B,BN,C(\langle B,BN,C\rangle \in Boats \wedge C = \text{'Red'} \Rightarrow$$
$$(\exists Ir,Br,D\ (\langle Ir,Br,D\rangle \in Reserves \wedge I = Ir \wedge Br = B)))\}$$

- Find all sailors I such that for each 3-tuple $\langle B,BN,C\rangle$ there is a tuple in Reserves showing that sailor I has reserved it.

Unsafe Queries, Expressive Power

- Syntactically correct calculus queries that have an infinite number of answers
  - Such queries are called *unsafe*.
  - e.g., $\{\langle I,N,T,A\rangle \mid \neg(\langle I,N,T,A\rangle \in Sailors)\}$
- Every relational algebra query can be expressed as a safe query in DRC/TRC
  - And vice versa!
- *Relational Completeness*: Query language (e.g., SQL) can express every query that is expressible in relational algebra/calculus.

Summary

- Relational calculus is "declarative" or non-operational
  - Users define queries in terms of what they want, not in terms of how to compute it
- Algebra and safe calculus have same expressive power
  - Relational "completeness"