Relational Calculus
Chapter 4, Part B

Domain Relational Calculus
- Query has the form:
  \[ p(x_1, x_2, ..., x_n) \]
- Answer includes all tuples \( (x_1, x_2, ..., x_n) \) that make the formula \( p(x_1, x_2, ..., x_n) \) be true.
- Formula is recursively defined, starting with simple atomic formulas (getting tuples from relations or making comparisons of values), and building bigger and better formulas using the logical connectives.

DRC Formulas
- Atomic formula:
  - \( (x_1, x_2, ..., x_n) \in Rname \) or \( x \in op Y \) or \( x \in op constant \)
  - \( op \) is one of \( <, >, =, \neq, \leq, \geq \)
- Formula:
  - an atomic formula, or
  - \( t \land p \land q \), \( p \lor q \), where \( p \) and \( q \) are formulas, or
  - \( \exists X (p(X)) \), where variable \( X \) is free in \( p(X) \)
  - \( \forall X (p(X)) \), 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 said to be free.

Free and Bound Variables
- The use of quantifiers \( \exists X \) and \( \forall X \) in a formula is said to bind \( X \).
  - A variable that is not bound is free.
- Let us revisit the definition of a query:
  \[ p(x_1, x_2, ..., x_n) \]
- There is an important restriction: the variables \( x_1, ..., x_n \) that appear to the left of `\( \)` must be the only free variables in the formula \( p(\cdot) \).

Find all sailors with a rating above 7
\[ \{ I, N, T, A \} \mid \{ I, N, T, A \} \in \text{Sailors} \land T > 7 \]
- The condition \( \{ I, N, T, A \} \in \text{Sailors} \) ensures that the domain variables \( I, N, T \) and \( A \) are bound to fields of the same Sailors tuple.
- The term \( \{ I, N, T, A \} \) to the left of `\( \)` (which should be read as `such that`) says that every tuple \( \{ I, N, T, A \} \) that satisfies \( T > 7 \) is in the answer.
- Modify this query to answer:
  - Find sailors who are older than 18 or have a rating under 9, and are called `jock`.
Find sailors rated > 7 who've reserved boat #103

\[ (I, N, T, A) \in \text{Sailors} \land T > 7 \]

\[ \exists \ I_r, B_r, D_r \ [ (I_r, B_r, D_r) \in \text{Reserves} \land I_r = 1 \land B_r = 103 ] \]

- We have used \( \exists \ I_r, B_r, D_r \) (…) as a shorthand for \( \exists \ I_r \ (\exists \ B_r \ (\exists \ D_r \ (\ldots))) \)
- Note the use of \( \exists \) to find a tuple in Reserves that 'joins with' the Sailors tuple under consideration.

Find sailors who've reserved all boats

\[ (I, N, T, A) \in \text{Sailors} \land \forall \ B, B, N, C \ [ (B, B, N, C) \in \text{Boats} ] \]

\[ \exists \ I_r, B_r, D_r \ [ (I_r, B_r, D_r) \in \text{Reserves} \land I_r = 1 \land B_r = B ] \]

- Find all sailors \( I \) such that for each 3-tuple \( B, B, N, C \) either it is not a tuple in Boats or there is a tuple in Reserves showing that sailor \( I \) has reserved it.

unsafe Queries, Expressive Power

- It is possible to write syntactically correct calculus queries that have an infinite number of answers! Such queries are called unsafe.
- For example: \[ S \subseteq \text{Sailors} \]

- It is known that every query that can be expressed in relational algebra can be expressed as a safe query in DRC / TRC; the converse is also true.
- Relational Completeness: Query language (e.g., SQL) can express every query that is expressible in relational algebra/calculus.

Summary

- Relational calculus is non-operational, and users define queries in terms of what they want, not in terms of how to compute it. (Declarativeness.)
- Algebra and safe calculus have same expressive power, leading to the notion of relational completeness.