CMPSC 174A/174N Fundamentals of Database System

Relational Calculus

Discussion Session
Friday, 9:00am-9:50am
Zexi Huang

Session 6

Schedule

- General approach
- Example
  - Sailors and Boats
- Exercise
  - Pilots and Airplanes

General Approach

- **General approach to translate statements into relational calculus:**
  - Translate the statements into relational algebra.
    - Hopefully you also prefer relational algebra to relational calculus.
    - Remember the tricks (e.g., finding highest salary, finding exact three boats).
  - Translate the relational algebra into relational calculus.
    - Make sure all variables except those appear on the left of the bar are bound.

$$R \cup S \qquad \{(x_1, \ldots, x_n) \mid R(x_1, \ldots, x_n) \vee S(x_1, \ldots, x_n)\}$$

$$R \cap S \qquad \{(x_1, \ldots, x_n) \mid R(x_1, \ldots, x_n) \wedge S(x_1, \ldots, x_n)\}$$

$$R - S \qquad \{(x_1, \ldots, x_n) \mid R(x_1, \ldots, x_n) \wedge \neg S(x_1, \ldots, x_n)\} \qquad R - S = R \cap \overline{S}$$

$$\sigma_F R \qquad \{(x_1, \ldots, x_n) \mid R(x_1, \ldots, x_n) \wedge F'\}$$

$$\text{where } F' = F|_{x_1}^{A_1} \cdots |_{x_n}^{A_n}$$

$$R \times S \qquad \{(x_1, \ldots, x_n, y_1, \ldots, y_m) \mid R(x_1, \ldots, x_n) \wedge S(y_1, \ldots, y_m)\}$$

$$\pi_{A_{i_1}, \ldots, A_{i_k}} R \quad \{(x_{i_1}, \ldots, x_{i_k}) \mid \exists x_{j_1} \cdots \exists x_{j_{n-k}} R(x_1, \ldots, x_n)\}$$

  - Reduce the translation result into simpler equivalent forms, if you like.

Sailors and Boats

- **Consider the schemas for the sailors and boats example:**
  - Sailors(sid: integer, sname: string)
  - Boats(bid: integer, bname: string, color: string)
  - Reserves(sid: integer, bid: integer, day: date)
- **Q1: Find the names of all boats.**
  - $\pi_{bname}(Boats)$
  - $\{\langle Bname\rangle | \exists Bid \exists Color (\langle Bname, Bid, Color\rangle \in Boats)\}$
  - $\{\langle Bname\rangle | \exists Bid, Color (\langle Bname, Bid, Color\rangle \in Boats)\}$
  - $\{\langle Bname\rangle | \exists Color, Bid (\langle Bname, Bid, Color\rangle \in Boats)\}$
  - $\{\langle Bname\rangle | \exists \langle Bname, Bid, Color\rangle \in Boats\}$, don't suggest.

Sailors and Boats

- **Consider the schemas for the sailors and boats example:**
  - Sailors(<u>sid: integer</u>, sname: string)
  - Boats(<u>bid: integer</u>, bname: string, color: string)
  - Reserves(<u>sid: integer, bid: integer, day: date</u>)
- **Q2: Find the names of the sailors who have reserved at least one boat.**
  - $\pi_{sname}(Sailors \bowtie Reserves)$, or $\pi_{sname}\sigma_{sid1=sid2}(Sailors \times Reserves)$.
  - $\{\langle Sname\rangle | \ldots\}$
  - $\{\langle Sname\rangle | \exists Sid1(\langle Sid1, Sname\rangle \in Sailors \ldots)\}$
  - $\{\langle Sname\rangle | \exists Sid1(\langle Sid1, Sname\rangle \in Sailors \wedge \exists Sid2 \exists Bid \exists Day(\langle Sid2, Bid, Day\rangle \in Reserves \ldots))\}$
  - $\{\langle Sname\rangle | \exists Sid1(\langle Sid1, Sname\rangle \in Sailors \wedge \exists Sid2 \exists Bid \exists Day(\langle Sid2, Bid, Day\rangle \in Reserves \wedge Sid1 = Sid2))\}$

Sailors and Boats

- **Consider the schemas for the sailors and boats example:**
  - Sailors(sid: integer, sname: string)
  - Boats(bid: integer, bname: string, color: string)
  - Reserves(sid: integer, bid: integer, day: date)
- **Q2: Find the names of the sailors who have reserved at least one boat.**
  - $\{\langle Sname\rangle | \exists Sid1(\langle Sid1, Sname\rangle \in Sailors \wedge \exists Sid2 \exists Bid \exists Day(\langle Sid2, Bid, Day\rangle \in Reserves \wedge Sid1 = Sid2))\}$
  - $\{\langle Sname\rangle | \exists Sid1(\langle Sid1, Sname\rangle \in Sailors \wedge \exists Sid2, Bid, Day(\langle Sid2, Bid, Day\rangle \in Reserves \wedge Sid1 = Sid2))\}$
  - $\{\langle Sname\rangle | \exists Sid(\langle Sid, Sname\rangle \in Sailors \wedge \exists Bid, Day(\langle Sid, Bid, Day\rangle \in Reserves))\}$
  - $\{\langle Sname\rangle | \exists Sid, Bid, Day(\langle Sid, Sname\rangle \in Sailors \wedge \langle Sid, Bid, Day\rangle \in Reserves)\}$, don't suggest.
  - $\{\langle Sname\rangle | \exists Sid, Bid, Day(\langle Sid, Sname\rangle \in Sailors) \wedge \langle Sid, Bid, Day\rangle \in Reserves\}$, wrong.

Sailors and Boats

- **Consider the schemas for the sailors and boats example:**
  - Sailors(<u>sid: integer</u>, sname: string)
  - Boats(<u>bid: integer</u>, bname: string, color: string)
  - Reserves(<u>sid: integer, bid: integer, day: date</u>)
- **Q3: Find the names of the sailors who have reserved at least two boats.**
  - $\pi_{sname}\left(Sailors \bowtie \sigma_{(sid1=sid2)\wedge(bid1\neq bid2)}(Reserves1 \times Reserves2)\right)$
  - $\{\langle Sname\rangle | \exists Sid0(\langle Sid0, Sname\rangle \in Sailors \ldots)\}$
  - $\{\langle Sname\rangle\ \exists Sid0\left(\langle Sid0, Sname\rangle \in Sailors \wedge \exists Sid1 \exists Bid1 \exists Day1(\langle Sid1, Bid1, Day1\rangle \in Reserves \ldots)\right)\}$
  - $\{\langle Sname\rangle\ \exists Sid0\left(\langle Sid0, Sname\rangle \in Sailors \wedge \exists Sid1 \exists Bid1 \exists Day1\left(\langle Sid1, Bid1, Day1\rangle \in Reserves \wedge \exists Sid2 \exists Bid2 \exists Day2(\langle Sid2, Bid2, Day2\rangle \in Reserves \ldots)\right)\right)\}$
  - $\{\langle Sname\rangle\ \exists Sid0\left(\langle Sid0, Sname\rangle \in Sailors \wedge \exists Sid1 \exists Bid1 \exists Day1\left(\langle Sid1, Bid1, Day1\rangle \in Reserves \wedge \exists Sid2 \exists Bid2 \exists Day2(\langle Sid2, Bid2, Day2\rangle \in Reserves \wedge Sid1 = Sid2 \wedge Bid1 \neq Bid2 \wedge Sid0 = Sid1)\right)\right)\}$

Sailors and Boats

- **Consider the schemas for the sailors and boats example:**
  - Sailors(sid: integer, sname: string)
  - Boats(bid: integer, bname: string, color: string)
  - Reserves(sid: integer, bid: integer, day: date)
- **Q3: Find the names of the sailors who have reserved at least two boats.**
  - $\{\langle Sname\rangle\ \exists Sid0\left(\langle Sid0, Sname\rangle \in Sailors \wedge \exists Sid1 \exists Bid \exists Day(\langle Sid1, Bid1, Day1\rangle \in Reserves \wedge \exists Sid2 \exists Bid \exists Day(\langle Sid2, Bid2, Day2\rangle \in Reserves \wedge Sid1 = Sid2 \wedge Bid1 \neq Bid2 \wedge Sid0 = Sid1))\right)\}$
  - $\{\langle Sname\rangle\ \exists Sid\left(\langle Sid, Sname\rangle \in Sailors \wedge \exists Bid1, Day1(\langle Sid, Bid1, Day1\rangle \in Reserves \wedge \exists Bid2, Day2(\langle Sid, Bid2, Day2\rangle \in Reserves \wedge Bid1 \neq Bid2))\right)\}$
  - $\{\langle Sname\rangle\ \exists Sid, Bid1, Bid2, Day1, Day2(\langle Sid, Sname\rangle \in Sailors \wedge \langle Sid, Bid1, Day1\rangle \in Reserve \wedge \langle Sid, Bid2, Day2\rangle \in Reserves \wedge Bid1 \neq Bid2)\}$

Sailors and Boats

- **Consider the schemas for the sailors and boats example:**
  - Sailors(<u>sid: integer</u>, sname: string)
  - Boats(<u>bid: integer</u>, bname: string, color: string)
  - Reserves(<u>sid: integer, bid: integer, day: date</u>)
- **Q4: Find the names of the sailors who have not reserved boats.**
  - $\pi_{sname}\left(Sailors - \pi_{sid,sname}(Sailors \bowtie Reserves)\right)$ $R - S$ $\{(x_1, \ldots, x_n) \mid R(x_1, \ldots, x_n) \wedge \neg S(x_1, \ldots, x_n)\}$
  - $\{\langle Sname\rangle | \exists Sid(\langle Sid, Sname\rangle \in Sailors \ldots)\}$
  - $\{\langle Sname\rangle | \exists Sid(\langle Sid, Sname\rangle \in Sailors \wedge \neg(\langle Sid, Sname\rangle \in Sailors \ldots))\}$
  - $\{\langle Sname\rangle | \exists Sid(\langle Sid, Sname\rangle \in Sailors \wedge \neg(\langle Sid, Sname\rangle \in Sailors \wedge \exists Bid, Day(\langle Sid, Bid, Day\rangle \in Reserves)))\}$
  - Let $p = \langle Sid, Sname\rangle \in Sailors, q = \exists Bid, Day(\langle Sid, Bid, Day\rangle \in Reserves$, we have $\{\langle Sname\rangle | \exists Sid(p \wedge \neg(p \wedge q))\}$
  - Recall $p \wedge \neg(p \wedge q) = p \wedge (\neg p \vee \neg q) = (p \wedge \neg p) \vee (p \wedge \neg q) = p \wedge \neg q$
  - $\{\langle Sname\rangle | \exists Sid(\langle Sid, Sname\rangle \in Sailors \wedge \neg\exists Bid, Day(\langle Sid, Bid, Day\rangle \in Reserves))\}$
  - Find the names of sailors such that there doesn't exist any corresponding records in reservation table.
  - $\{\langle Sname\rangle | \exists Sid(\langle Sid, Sname\rangle \in Sailors \wedge \forall Bid, Day(\langle Sid, Bid, Day\rangle \notin Reserves))\}$

Sailors and Boats

- **Consider the schemas for the sailors and boats example:**
  - Sailors(sid: integer, sname: string)
  - Boats(bid: integer, bname: string, color: string)
  - Reserves(sid: integer, bid: integer, day: date)
- **Q5: Find the names of the sailors who have reserved all boats.**
  - $\pi_{sname}\left(Sailors \bowtie \pi_{sid,bid}(Reserves) \div \pi_{bid}(Boats)\right)$
  - $\{\langle Sname\rangle | \exists Sid(\langle Sid, Sname\rangle \in Sailors \ldots)\}$
  - $\{\langle Sname\rangle | \ldots \wedge \forall Bid, Bname, Color((Bid, Bname, Color) \in Boats \Rightarrow \exists Day((Bid, Sid, Day) \in Reserves)))\}$
  - Find the name of sailors such that for any boat, there exist a corresponding record between the sailor and the boat in reservation table.
  - $\forall Bid, Bname, Color((Bid, Bname, Color) \in Boats)$ is always false unless there is only one record in Boats.

bid	name	color
102	Interlake	red
103	Clipper	green

- Bid={102,103}, Bname={Interlake, Clipper},Color={red,green}
- All possible combination of variables: (102, Interlake, red), (102, Interlake, green), (102, Clipper, red), (102, Clipper, green), (103, Interlake, red), (103, Interlake, green), (103, Clipper, red), (103, Clipper, red)

Sailors and Boats

- **Consider the schemas for the sailors and boats example:**
  - Sailors(sid: integer, sname: string)
  - Boats(bid: integer, bname: string, color: string)
  - Reserves(sid: integer, bid: integer, day: date)
- **Q5: Find the names of the sailors who have reserved all boats.**
  - $\pi_{sname}\left(Sailors \bowtie \pi_{sid,bid}(Reserves) \div \pi_{bid}(Boats)\right)$
  - $\{\langle Sname \rangle | \ldots \wedge \forall Bid, Bname, Color((Bid, Bname, Color) \in Boats \Rightarrow \exists Day((Bid, Sid, Day) \in Reserves)))\}$
  - Recall $p \Rightarrow q = \neg p \vee q$
  - $\{\langle Sname \rangle | \ldots \wedge \forall Bid, Bname, Color((Bid, Bname, Color) \notin Boats \vee \exists Day((Bid, Sid, Day) \in Reserves)))\}$
  - Find the name of sailors such that for any boat, either the boat doesn't exist in boat table (always false), or there exist a corresponding record between the sailor and the boat in reservation table.
  - $\{\langle Sname \rangle | \ldots \wedge \forall Bid, Bname, Color(\exists Day((Bid, Sid, Day) \in Reserves)))\}$, what are Bname and Color?
  - $\{\langle Sname \rangle | \ldots \wedge \forall Bid\ (\exists Day((Bid, Sid, Day) \in Reserves)))\}$
  - Find the name of sailors such that for any Bid, there exist a corresponding record between the sailor and the Bid in reservation table.

Sailors and Boats

- **Consider the schemas for the sailors and boats example:**
  - Sailors(sid: integer, sname: string)
  - Boats(bid: integer, bname: string, color: string)
  - Reserves(sid: integer, bid: integer, day: date)
- **Q5: Find the names of the sailors who have reserved all boats.**
  - All red boats?
  - $\pi_{sname}\left(Sailors \bowtie \pi_{sid,bid}(Reserves) \div \pi_{bid}(\sigma_{color='red'} Boats)\right)$
  - $\{\langle Sname\rangle \mid \ldots \wedge \forall Bid, Bname, Color(((Bid, Bname, Color) \in Boats \wedge Color =' red') \Rightarrow \exists Day \ldots))\}$
  - $\{\langle Sname\rangle \mid \ldots \wedge \forall\ Boats(Bid, Bname, Color)(Color =' red' \Rightarrow \exists Day((Bid, Sid, Day) \in Reserves)))\}$
  - $\{\langle Sname\rangle \mid \ldots \wedge \forall Bid, Bname((Bid, Bname, 'red') \in Boats \Rightarrow \exists Day((Bid, Sid, Day) \in Reserves)))\}$
  - Find the name of sailors such that for any red boat, there exist a corresponding record between the sailor and the boat in reservation table.

Pilots and Airplanes

- **Ex 4.5: Consider the schemas for the pilots and airplanes:**
  - Flights(flno: integer, from: string, to: string, distance: real, departs: time, arrives: time)
  - Aircraft(aid: integer, aname: string, cruisingrange: real)
  - Employees(eid: integer, ename: string, salary: real)
  - Certified(eid: integer, aid: integer)
- **Q1: Find the names of pilots certified for some Boeing aircraft.**
  - $\pi_{ename}(Employees \bowtie Certified \bowtie (\sigma_{aname='Boeing'}(Aircraft))$
  - $\{\langle Ename\rangle \mid \exists Eid1, Salary\left(\langle Eid1, Ename, Salary\rangle \in Employees \land \exists Eid2, Aid2\left(\langle Eid2, Aid2\rangle \in Certified \land \exists Aid1, Aname, Cruisingrange(\langle Aid1, Aname, Cruisingrange\rangle \in Aircraft \land Eid1 = Eid2 \land Aid1 = Aid2 \land Aname = ' Boeing')\right)\right)\}$
  - $\{\langle Ename\rangle \mid \exists Eid, Salary\left(\langle Eid, Ename, Salary\rangle \in Employees \land \exists Aid\left(\langle Eid, Aid\rangle \in Certified \land \exists Cruisingrange(\langle Aid, ' Boeing', Cruisingrange\rangle \in Aircraft)\right)\right)\}$

Pilots and Airplanes

- **Ex 4.5: Consider the schemas for the pilots and airplanes:**
  - Flights(<u>flno: integer</u>, from: string, to: string, distance: real, departs: time, arrives: time)
  - Aircraft(<u>aid: integer</u>, aname: string, cruisingrange: real)
  - Employees(<u>eid: integer</u>, ename: string, salary: real)
  - Certified(<u>eid: integer, aid: integer</u>)

- **Q2: Find the names of all aircraft that can be used on non-stop flights from Los Angeles to Tokyo.**
  - $\pi_{aname}(\sigma_{cruisingrange>distance}(Aircraft \times \sigma_{(from='Los\ Angeles')\wedge(to='Tokyo')}Flights))$
  - $\{\langle Aname\rangle \mid \exists Aid, Cruisingrange(\langle Aid, Aname, Cruisingrange\rangle \in Aircraft \wedge \exists Flno, From, To, Distance, Departs, Arrives(\langle Flno, From, To, Distance, Departs, Arrives\rangle \in Flights \wedge \exists Cruisingrange > Distance \wedge From =' Los\ Angeles' \wedge To =' Tokyo'))\}$
  - $\{\langle Aname\rangle \mid \exists Aid, Cruisingrange(\langle Aid, Aname, Cruisingrange\rangle \in Aircraft \wedge \exists Flno, Distance, Departs, Arrives(\langle Flno,' Los\ Angeles',' Tokyo', Distance, Departs, Arrives\rangle \in Flights \wedge \exists Cruisingrange > Distance))\}$

Pilots and Airplanes

- **Ex 4.5: Consider the schemas for the pilots and airplanes:**
  - Flights(<u>flno: integer</u>, from: string, to: string, distance: real, departs: time, arrives: time)
  - Aircraft(<u>aid: integer</u>, aname: string, cruisingrange: real)
  - Employees(<u>eid: integer</u>, ename: string, salary: real)
  - Certified(<u>eid: integer, aid: integer</u>)

- **Q3: Find eids of employees with the highest salary.**
  - $\pi_{ename}\left(\text{Employees} - \pi_{eid1,ename1,salary1}\left(\sigma_{salary1<salary2} Employees1 \times Employees2\right)\right)$
  - $\{\langle Eid\rangle \mid \exists Ename0, Salary0\left(\langle Eid, Ename0, Salary0\rangle \in Employees \wedge \neg\exists Eid1, Ename1, Salary1\left(\langle Eid1, Ename1, Salary1\rangle \in Employees \wedge \exists Eid2, Ename2, Salary2(\langle Eid2, Ename2, Salary2\rangle \in Employees \wedge Eid = Eid1 \wedge Salary1 < Salary2)\right)\right)\}$
  - $\{\langle Eid\rangle \mid \exists Ename1, Salary1\left(\langle Eid, Ename1, Salary1\rangle \in Employees \wedge \neg\exists Eid2, Ename2, Salary2(\langle Eid2, Ename2, Salary2\rangle \in Employees \wedge Salary1 < Salary2)\right)\}$

    $p \wedge \neg(p \wedge q) = p \wedge \neg q$
  - $\{\langle Eid\rangle \mid \exists Ename1, Salary1\left(\langle Eid, Ename1, Salary1\rangle \in Employees \wedge \forall Eid2, Ename2, Salary2(\langle Eid2, Ename2, Salary2\rangle \in Employees \Rightarrow Salary1 \geq Salary2)\right)\}$

    $\neg(p \wedge q) = \neg p \vee \neg q = p \Rightarrow \neg q$

Pilots and Airplanes

- **Ex 4.5: Consider the schemas for the pilots and airplanes:**
  - Flights(<u>flno: integer</u>, from: string, to: string, distance: real, departs: time, arrives: time)
  - Aircraft(<u>aid: integer</u>, aname: string, cruisingrange: real)
  - Employees(<u>eid: integer</u>, ename: string, salary: real)
  - Certified(<u>eid: integer, aid: integer</u>)

- **Q4: Find the names of pilots who can operate planes with a range greater than 3,000 miles but are not certified on any Boeing aircraft.**
  - $\pi_{ename}\big(\text{Employees} \bowtie (\text{Certified} \bowtie (\sigma_{cruisingrange>3000} Aircraft) \cap (\text{Certified} \bowtie (\sigma_{aname\neq'Boeing'} Aircraft)))\big)$
  - Wrong! Certified $\bowtie (\sigma_{aname\neq'Boeing'} Aircraft)$ means pilots who are certified on some non-Boeing aircraft.
  - $\pi_{ename}\big(\text{Employees} \bowtie ((\text{Certified} \bowtie (\sigma_{cruisingrange>3000} Aircraft) - (\text{Certified} \bowtie (\sigma_{aname='Boeing'} Aircraft)))\big)$
  - $\{\langle Ename\rangle \mid \exists Eid, Salary\big(\langle Eid, Ename, Salary\rangle \in Employees \wedge \exists Aid1\big(\langle Eid, Aid1\rangle \in Certified \wedge \exists Aname1, Cruisingrange1(\langle Aid1, Aname1, Cruisingrange1\rangle \in Aircraft \wedge Cruisingrange > 3000\ )\big) \wedge \neg\exists Aid2\big(\langle Eid, Aid2\rangle \in Certified \wedge \exists Aname2, Cruisingrange2(\langle Aid2, Aname2, Cruisingrange2\rangle \in Aircraft \wedge Aname2 =' Boeing')\big)\big)\}$

Pilots and Airplanes

- **Ex 4.5: Consider the schemas for the pilots and airplanes:**
  - Flights(flno: integer, from: string, to: string, distance: real, departs: time, arrives: time)
  - Aircraft(aid: integer, aname: string, cruisingrange: real)
  - Employees(eid: integer, ename: string, salary: real)
  - Certified(eid: integer, aid: integer)
- **Q5: Find the eids of employees who are certified for the largest number of aircraft.**
  - We can't do it with relational calculus.
- **Q6: Find total amount paid to employees as salaries.**
  - We can't do it with relational calculus.