CMPSC 174A/174N Fundamentals of Database System

Relational Algebra

Discussion Session

Friday, 9:00am-9:50am

Zexi Huang

Schedule

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

Sailors and Boats

- **Consider the schemas for the sailors and boats example:**
  - Sailors(<u>sid: integer</u>, sname: string, rating: integer, age: real)
  - Boats(<u>bid: integer</u>, bname: string, color: string)
  - Reserves(<u>sid: integer, bid: integer, day: date</u>)

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25.5
95	Bob	3	63.5

Sailors

bid	name	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

Boats

sid	bid	day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Reserves

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>)
- **Q1: Find the names of all sailors.**
  - $\pi_{sname}(Sailors)$

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.**
  - $\pi_{sname}(Sailors \bowtie Reserves)$

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.**
  - $\rho(ReservationPairs, Reserves \times Reserves)$
  - $\rho((1 \rightarrow sid1, 2 \rightarrow bid1, 4 \rightarrow sid2, 5 \rightarrow bid2), ReservationPairs))$
  - $\rho\left(RequestedSailors, \pi_{sid1}\sigma_{(sid1=sid2)\wedge(bid1\neq bid2)}(ReservationPairs)\right)$
  - $\rho((1 \rightarrow sid), RequestedSailors)$
  - $\pi_{sname}(RequestedSailors \bowtie Sailors)$

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.**
  - $\rho(R1, Reserves)$
  - $\rho(R2, Reserves)$
  - $\rho(RequestedSailors, \pi_{R1.sid}(\sigma_{(R1.sid=R2.sid)\wedge(R1.bid\neq R2.bid)} R1 \times R2))$
  - $\pi_{sname}(RequestedSailors \bowtie Sailors)$
  - Sailors reserving at least three boats?

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 reserved all boats.**
  - $\rho(RequestedSailors, \pi_{sid,bid}(Reserves) \div \pi_{bid}(Boats))$
  - $\rho(RequestedSailors, \pi_{sid}(Reserves) - \pi_{sid}\left(\pi_{sid}(Reserves) \times \pi_{bid}(Boats) - \pi_{sid,bid}(Reserves)\right))$
  - $\pi_{sname}(RequestedSailors \bowtie Sailors)$
  - All red boats?
  - $\pi_{sname}((\pi_{sid,bid}(Reserves) \div \pi_{bid}\sigma_{color='red'}(Boats)) \bowtie Sailors)$

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 not reserved boats.**
  - $\rho(RequestedSailors, \pi_{sid}(Sailors) - \pi_{sid}(Sailors \bowtie Reserves))$
  - $\pi_{sname}(RequestedSailors \bowtie Sailors)$
  - Not reserved boats named 'Marine'?
  - $\rho(RequestedSailors, \pi_{sid}(Sailors) - \pi_{sid}(Sailors \bowtie Reserves \bowtie (\sigma_{bname='Marine'}(Boats)))$
  - $\pi_{sname}(RequestedSailors \bowtie Sailors)$

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.**
  - $\pi_{ename}(Employees \bowtie Certified \bowtie (\sigma_{aname='Boeing'}(Aircraft))$
  - $\pi_{ename}(\sigma_{aname='Boeing'}(Employees \bowtie Certified \bowtie Aircraft))$

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.**
  - $\rho(LAtoTokyo, \sigma_{(from='Los\ Angeles')\wedge(to='Tokyo')} Flights)$
  - $\pi_{aname}(\sigma_{cruisingrange>distance}(Aircraft \times LAtoTokyo))$
  - Can we do a natural join here?
  - $\pi_{aname}(Aircraft \bowtie_{cruisingrange>distance} LAtoTokyo)$

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 the names of pilots who can operate planes with a range greater than 3,000 miles but are not certified on any Boeing aircraft.**
  - $\rho(Pilots3000, \pi_{eid}(\text{Certified} \bowtie (\sigma_{cruisingrange>3000} Aircraft)))$
  - $\rho(PilotsBoeing, \pi_{eid}(\text{Certified} \bowtie (\sigma_{aname='Boeing'} Aircraft)))$
  - $\pi_{aname}((Pilots3000 - PilotsBoeing) \bowtie \text{Employees})$
  - Any other solutions?

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 eids of employees with the highest salary.**
  - $\rho(EmployeePairs(1 \to eid1, 3 \to salary1, 4 \to eid2, 6 \to salary2), \text{Employees} \times \text{Employees})$
  - $\rho(EmployeeLowSalary(1 \to eid), \pi_{eid1}(\sigma_{salary1 < salary2} EmployeePairs))$
  - $\pi_{eid}(Employees) - EmployeeLowSalary$
  - Employees with second highest salary?
  - In SQL we will use aggregation operator **MAX**.

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 eids of employees who are certified for exactly three aircraft.**
  - $\rho(Employee3(1 \rightarrow eid1, 2 \rightarrow aid1, 3 \rightarrow eid2, 4 \rightarrow aid2, 5 \rightarrow eid3, 6 \rightarrow aid3), \text{Certified} \times \text{Certified} \times \text{Certified})$
  - $\rho(EmployeeG3(1 \rightarrow eid), \pi_{eid1}(\sigma_{(eid1=eid2)\wedge(eid2=eid3)\wedge(aid1\neq aid2)\wedge(aid2\neq aid3)\wedge(aid1\neq aid3)} Employee3))$
  - Similarly, we can find $Employee4$ and $EmployeeG4$.
  - $EmployeeG3 - EmployeeG4$

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>)
- **Q6: Find the eids of employees who are certified for the largest number of aircraft.**
  - We can't do it with relational algebra. In SQL we will use aggregation operator **COUNT**.
- **Q7: Find total amount paid to employees as salaries.**
  - We can't do it with relational algebra. In SQL we will use aggregation operator **SUM**.