Alan Moss

CSCD 327

Lab #2

1) Write the following queries in relational algebra, using the schema blow, where the primary keys are underlined.
Sailors(<u>sid</u>, sname, rating, age)
Reserves(<u>sid</u>, <u>bid</u>, <u>day</u>)
Boats(<u>bid</u>, bname, color)

   a. Find sids and snames of sailors who have reserved boat 103 (i.e., bid = 103)
   $\Pi_{\text{sailors.sid, sailors.snames}}$ $(\sigma_{\text{bid = 103 ^ sailors.sid = reserves.sid}}(\text{sailors X reserves}))$
   b. Find sids and snames of sailors who have reserved a red boat.
   Natural join is like cartesian product but only common attributes, in other words natural join (r x s) = $\sigma_{\text{r.A = s.A ^ r.B = s.B}}$(r x s). Because we have this structure above, we can write this query in a simpler manner using natural join:
   $\Pi_{\text{sailors.sid, sailors.snames}}(\sigma_{\text{boats.color = 'red'}}(\text{sailors} \bowtie \text{reserves} \bowtie \text{boats}))$
   Another option:
   $\Pi_{\text{sailors.sid, sailors.snames}}$ $(\sigma_{\text{sailors.sid = reserves.sid ^ boats.color = 'red'}}(\text{sailors X reserves}))$
   c. Find sids and snames of sailors who have reserved a red *or* a green boat.
   $\Pi_{\text{sailors.sid, sailors.snames}}(\sigma_{\text{boats.color = 'red'}}(\text{sailors} \bowtie \text{reserves} \bowtie \text{boats})) \cup$
   $\Pi_{\text{sailors.sid, sailors.snames}}(\sigma_{\text{boats.color = 'green'}}(\text{sailors} \bowtie \text{reserves} \bowtie \text{boats}))$
   d. Find sids and snames of sailors who have reserved a red *and* a green boat.
   $\Pi_{\text{sailors.sid, sailors.snames}}(\sigma_{\text{boats.color = 'red'}}(\text{sailors} \bowtie \text{reserves} \bowtie \text{boats})) \cap$
   $\Pi_{\text{sailors.sid, sailors.snames}}(\sigma_{\text{boats.color = 'green'}}(\text{sailors} \bowtie \text{reserves} \bowtie \text{boats}))$

2) Write the following queries in relational algebra, using the university schema I gave you in class.
   a. Find the ids of students who have taken at least one course section taught by 'Dan Li'.
   $\Pi_{\text{takes.ID}}$ $(\sigma_{\text{teaches.course\_id = takes.course\_id ^ instructor.name = 'Dan Li'}}(\text{takes x teaches x instructor}))$
   b. Find the ids of students who have taken all Comp. Sci. courses at 200 level (i.e., course_id like 'CS2%').
   $\Pi_{\text{ID, course\_id}}$ (takes) $\div$ $\Pi_{\text{course\_id}}$ $(\sigma_{\text{course\_id = 'CS\%2'}}(\text{course})$
   c. Find the ids of students who have received an "A" on all the courses they have taken
   $\Pi_{\text{takes.ID}}$ (takes) - $\Pi_{\text{takes.ID}}$ $(\sigma_{\text{grade} \neq \text{'A'}}$ (takes x course))

3) Write the following queries in relational algebra, using the schema below, where the primary keys are underlined

employee (eid, ename, street, city)
company (cid, cname, city)
works (eid, cid, salary)
manages (eid, manager_id) (note: manager_id is a foreign key referencing eid in employee relation. In other words, managers are also employees.)

a. Find the eids, enames, streets, and cities of all employees who work for "EWU" and earn less than \$50,000.

   $\Pi_{\text{eid, ename, street, city}}$ ($\sigma_{\text{employee.eid = works.eid ^ works.cid = company.cid ^ works.salary < 50,000 ^company.cid = 'EWU'}}$(employee ⋈ works))

b. Find the eids and enames of all employees who live in the same city as the company for which they work.

   $\Pi_{\text{eid, ename}}$(employee ⋈ works ⋈ company)

c. Find the ename of Peter Parker's manager.

   Π_{eid}($\sigma_{\text{ename = 'peter parker'}}$ (employee))
   Need to use employee table again to find manager's eid