NYU | TANDON SCHOOL OF ENGINEERING

CS6083: Principles of Database Systems

Writing Queries in SQL, RA, and RC

Topics:

- How to write queries using SQL, RA, RC
- How to move between SQL, RA, and RC
- Using nested subqueries
- Using set operations

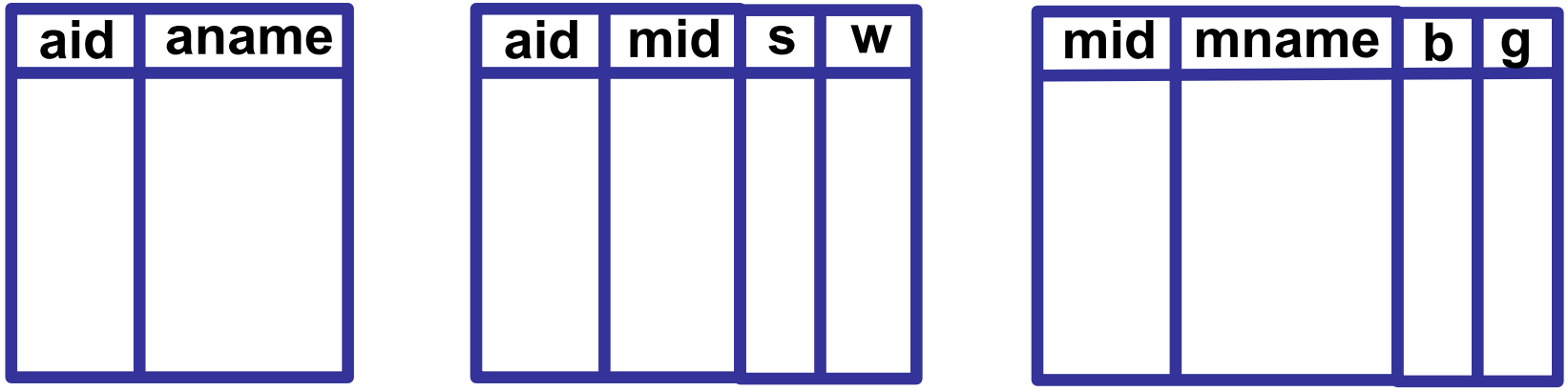
Example Schema: Actor-Movie DB

Actor (aid, aname)

Movie (mid, mname, budget, gross)

ActedIn (aid, mid, starring, wage)

aid refs aid in actor, mid refs mid in movie



Tuple-Relational Calculus:

- Query: ID of the actor with name “K. Bacon”

Tuple-Relational Calculus:

- Query: ID of the actor with name “K. Bacon”

$$\{ t \mid Ex\ a \in Actor (a[aid] = t[aid] \wedge a[aname] = \text{“K. Bacon”}) \}$$

Note: we use Ex “the exists”
Al “for all”
in “element of”

Also, switch from variable t to variable a to output only aid, as this is the only attribute of t that is referenced in the query.

Another Query, in DRC:

- ID and name of actors appearing in movie T2

Another Query, in DRC:

- ID and name of actors appearing in movie T2

- Note: you need to “join” all three tables
- But how to do that in RC?

"ID AND NAME OF ACTORS WHO APPEARED IN T2"

$$\{ \langle aid, an \rangle \mid \langle aid, an \rangle \in ACTOR \wedge \exists mid, s, w (\langle aid, mid, s, w \rangle \in ACTED\_IN \wedge \exists b, g (\langle mid, "T2", b, g \rangle \in MOVIE)) \}$$

Some Variations on this Query:

- ID and name of actors appearing in T2
- How about “and where wage > $500”?
- Or how about actors not appearing in T2?

COMPARE TO RELATIONAL ALGEBRA:

$$\pi_{AID,\,ANAME}\left(\leftarrow_{NNAME = "T2"}\left(ACTOR \bowtie \left(ACTED\_IN \bowtie MOVIE\right)\right)\right)$$

JOIN vs. $\exists x,y\,(\langle x,y,\ldots\rangle \in \ldots)$

"ID OF ACTOR WHO HAS APPEARED IN EVERY MOVIE"

"ID OF ACTOR WHO HAS APPEARED IN EVERY MOVIE"

$$\{\langle aid \rangle \mid \nexists\, mid, mn, b, g \left(\langle mid, mn, b, g \rangle \in MOVIE \right.$$

$$\left. \wedge \nexists\, s, w \left(\langle aid, mid, s, w \rangle \in ACTED\_IN \right) \right)\}$$

OR

$$\{\langle aid \rangle \mid \forall\, mid, mn, b, g \left(\langle mid, mn, b, g \rangle \in MOVIE \right.$$

$$\left. \Rightarrow \exists\, s, w \left(\langle aid, mid, s, w \rangle \in ACTED\_IN \right) \right)\}$$

RELATIONAL ALGEBRA :

RELATIONAL ALGEBRA :

$$\left(\Pi_{aid,mid}(ACTED\_IN)\right) \div \left(\Pi_{MID}(MOVIE)\right)$$

Set Operations in SQL:

- Union, Intersect, Except
- In, not in, some, all, exists, unique

  (used in nested queries)
- "Actors who appeared in Star Wars I and II"

Set Operations in SQL:

- "Actors who appeared in Star Wars I and II"

```
( SELECT A.name
  FROM Actor A, ActedIn AI, Movie M
  WHERE  A.aid = AI.aid  AND  AI.mid = M.mid AND
                M.title = "Star Wars I" )
INTERSECT
( SELECT A.name
  FROM Actor A, ActedIn AI, Movie M
  WHERE  A.aid = AI.aid  AND  AI.mid = M.mid AND
                M.title = "Star Wars II" )
```

Set Operations in SQL:

- "Actors who appeared in Star Wars I and II"

```
( SELECT A.name
  FROM Actor A, ActedIn AI, Movie M
  WHERE  A.aid = AI.aid  AND  AI.mid = M.mid AND
                M.title = "Star Wars I" )
INTERSECT
( SELECT A.name
  FROM Actor A, ActedIn AI, Movie M
  WHERE  A.aid = AI.aid  AND  AI.mid = M.mid AND
                M.title = "Star Wars II" )
```


How does result change if use UNION or EXCEPT?

Set Operations in SQL:

- Another solution:

```
SELECT A.name
FROM Actor A, ActedIn AI, Movie M
WHERE  A.aid = AI.aid  AND  AI.mid = M.mid AND
          M.title = "Star Wars I" AND A.aid IN

 ( SELECT A.aid
   FROM Actor A, ActedIn AI, Movie M
   WHERE  A.aid = AI.aid  AND  AI.mid = M.mid AND
             M.title = "Star Wars II" )
```

Set Operations in SQL:

- Another solution:

```
SELECT A.name
FROM Actor A, ActedIn AI, Movie M
WHERE  A.aid = AI.aid  AND  AI.mid = M.mid AND
          M.title = "Star Wars I" AND A.aid IN

 ( SELECT A.aid
   FROM Actor A, ActedIn AI, Movie M
   WHERE  A.aid = AI.aid  AND  AI.mid = M.mid AND
             M.title = "Star Wars II" )
```

What if we use "NOT IN"?

Using Nested Query:

- "Actors who received the highest wage ever"

```
SELECT A.name
FROM  Actor A, ActedIn AI
WHERE A.aid = AI.aid AND
          AI.wage >= ALL  ( SELECT AI.wage
                            FROM ActedIn AI )
```

How about if we use "SOME" ?

Another Solution:

- "Actors who received the highest wage ever"

```
SELECT A.name
FROM  Actor A, ActedIn AI
WHERE A.aid = AI.aid AND
          AI.wage = ( SELECT MAX(AI.wage)
                      FROM ActedIn AI )
```

Yet Another Solution:

- "Actors who received the highest wage ever"

```
SELECT A.name
FROM  Actor A, ActedIn AI
WHERE A.aid = AI.aid AND
        NOT EXIST ( SELECT AI2.wage
                    FROM ActedIn AI2
                    WHERE AI2.wage > AI.wage )
```

Why is this not a good way to write this query?

Yet Another Solution:

- "Actors who received the highest wage ever"

```
SELECT A.name
FROM  Actor A, ActedIn AI
WHERE A.aid = AI.aid AND
        NOT EXIST ( SELECT AI2.wage
                    FROM ActedIn AI2
                    WHERE AI2.wage > AI.wage )
```

Note: this is an RC approach to writing a query

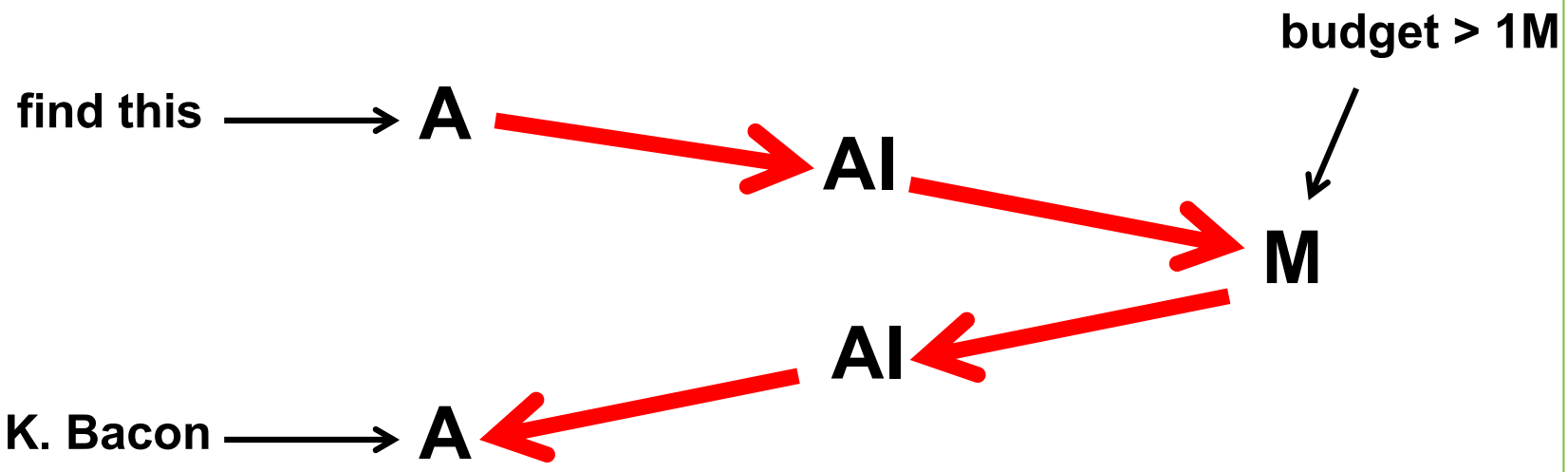
Query Graphs, RC, RA, and SQL:

- "Actors who acted in a movie with Kevin Bacon where budget was > 1 million"

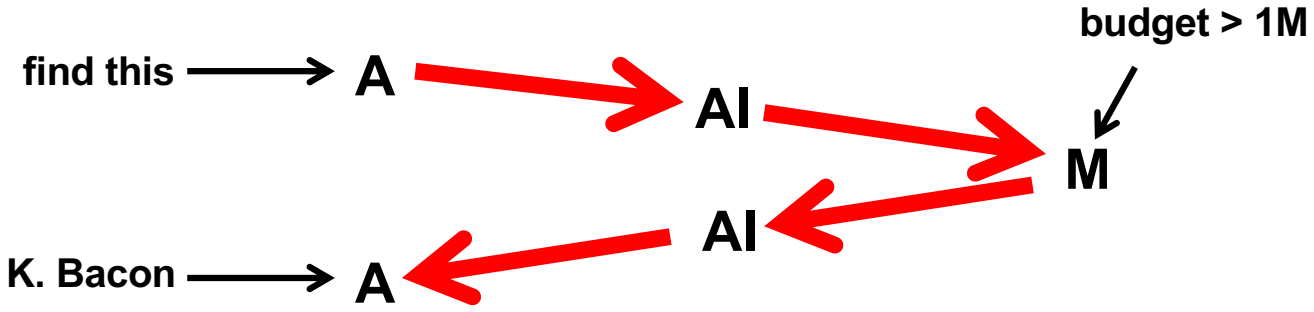
Query Graphs, RC, RA, and SQL:

- "Actors who acted in a movie with Kevin Bacon where budget was > 1 million"

High level view of query:

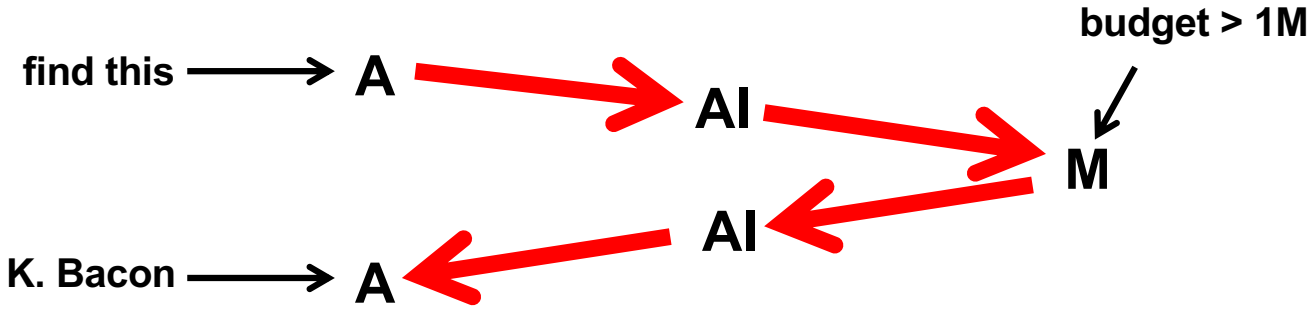


TRC:



$$\{ t \mid \text{Ex } a1 \text{ in Actor } (a1[aname] = t[aname] \wedge$$
$$\text{Ex } ai1 \text{ in ActedIn } (ai1[aid] = a1[aid] \wedge$$
$$\text{Ex } m \text{ in Movie } (m[mid] = ai1[mid] \wedge$$
$$m[budget\} > 1000000 \wedge$$
$$\text{Ex } ai2 \text{ in ActedIn } (ai2[mid] = m[mid] \wedge$$
$$\text{Ex } a2 \text{ in Actor } (a2[aid] = ai2[aid] \wedge$$
$$a2[name] = \text{“K. Bacon”}))))) \}$$

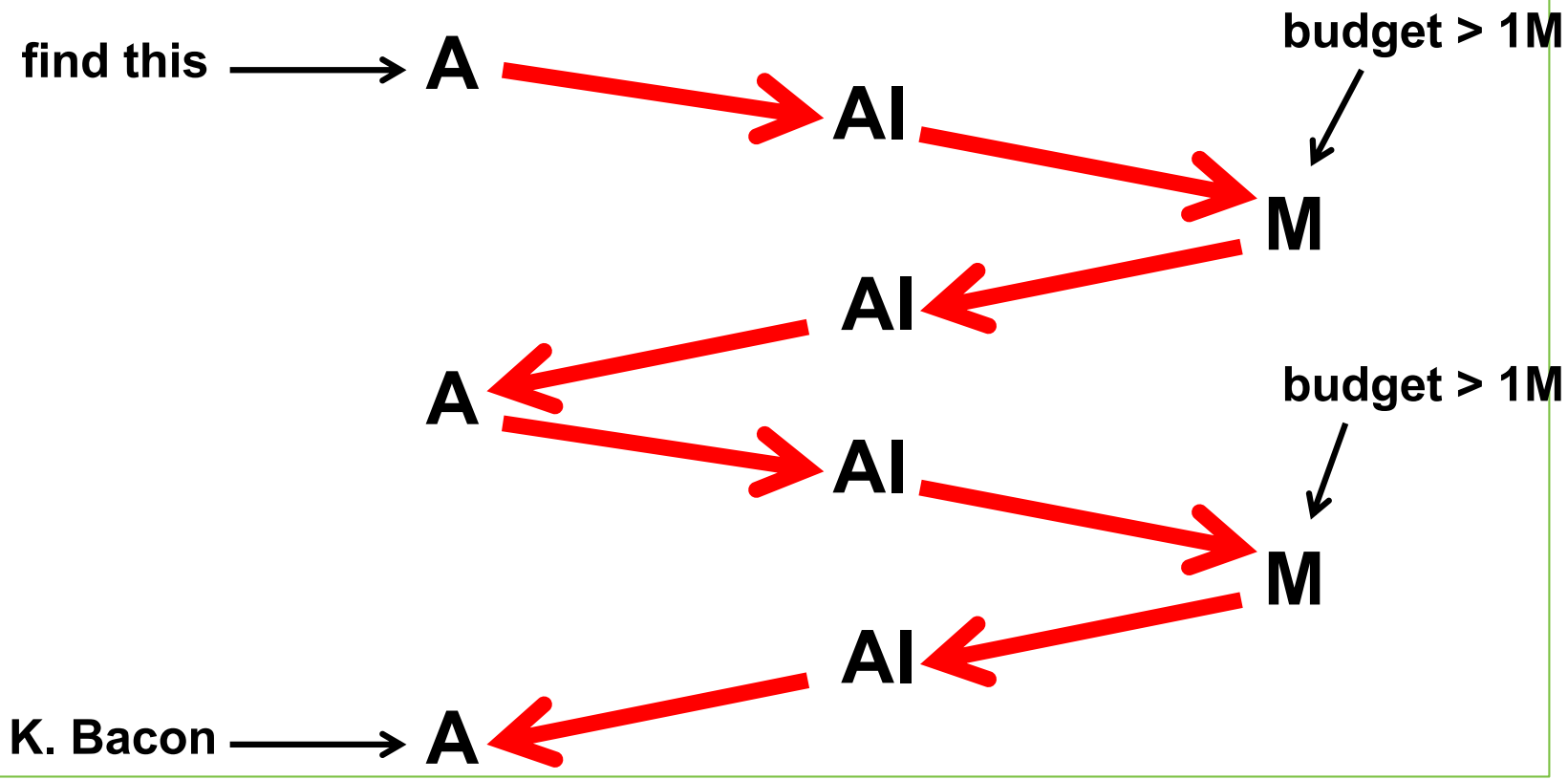
SQL:




```
SELECT A1.aname
FROM Actor A1, Actor A2, ActedIn AI1, ActedIn AI2,
        Movie M
WHERE <join conditions> AND M.budget > 1000000
        AND A2.aname = "K. Bacon"
```

- **“Actors who acted in a movie > 1M with someone who acted in a movie > 1M with Kevin Bacon”**

- **"Actors who acted in a movie > 1M with someone who acted in a movie > 1M with Kevin Bacon"**



Query Graphs, RC, RA, and SQL:

- "Actors who acted in a movie with Kevin Bacon where budget was > 1 million"

- how about actors who acted with an actor who has acted with Kevin Bacon?

- how about distance 3, 4, infinity?

- transitive closure limitation