Relational Algebra:

Problem D. Potpourri: 30 points (5 Points per question):

Please answer the following questions. You must give a YES—NO answer to questions D1–D4 and if your answer is YES you must also write the equivalent RA expression.

D1 Can the intersection of relations R(A,B) and S(A,B) be expressed using only natu- ral joins?

Answer: Yes. $R \cap S = R \bowtie S.$

D2 Can the intersection of relations R(A,B) and S(A,B) be expressed using the set difference operator?

Answer: Yes. R∩S=R−(R−S).

D3 Can the intersection of relations R(A,B) and S(A,B) be expressed using the cartesian product and projection operator

Answer: No.

D4 Can the intersection of relations R(A,B) and S(A,B) be expressed using the carte- sian product, selection and projection operators?

Answer: Yes.

$$R \cap S = \pi_{R.A,R.B}(\sigma_{R.A=S.A \wedge R.B=S.B}(R \times S)).$$

Every condition is evaluated as True, False or Unknown.

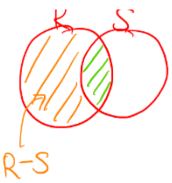
σC (R) -> filters out rows in a relation

πA(R) -> filters out columns in a relation. Select column aka

R X S -> cross product, concatenates tuples from pairs of tuples

Natural Join -> shorthand for σ_student.sid = enroll.sid(R X S) for joining Student and Enroll

ρS(R) -> rename S to S ρS(A1,A2)(R) -> rename R to S(A1, A2) including attribute names



Intersect -> R and S = R – (R – S)

Core relational operators:

- σ, π, $\times$, $\cancel{\bowtie}$, ρ, $\cup$, $\cancel{\cap}$, $-$

A - B

- Q: which ones are "core" and which ones can be expressed with others?

Null handling:

How are NULL values handled for UNION ALL, INTERSECT ALL, and EXCEPT ALL?

Very much like regular values. {NULL } UNION ALL {NULL } = { NULL, NULL } for example.

Arithmetic operators with NULL input returns NULL

SQL only returns only True tuples except for aggregate functions

Aggregates are computed ignoring NULL values except COUNT(*)

When the input to an aggregate is empty, count returns 0; all others return NULL

Coalesce() function:

Returns FIRST NON NULL value in the list (for example if I had a COALESCE(total – average, 0) and the first one was null, then it gets 0.

4. The relation **Company(company-name, valuation)** captures Company-valuation information, where **company-name** is the name of a company and **valuation** is its valuation. Write a relational algebra expression to find the name of the lowest valued companies. (Hint: When a query is difficult, think of its complement.)

$$\Pi_{\text{Company-name}}(\text{Company}) - \Pi_{\text{C1.Company-name}}(\sigma_{\text{C1.valuation}>\text{C2.valuation}}(\rho_{C1}(\text{Company}) \times \rho_{C2}(\text{Company})))$$

→ Find the Max x:

$$\Pi_x(A) - \Pi_{A.x}(\sigma_{A.x < d.x}(A \times \rho_d(A)))$$

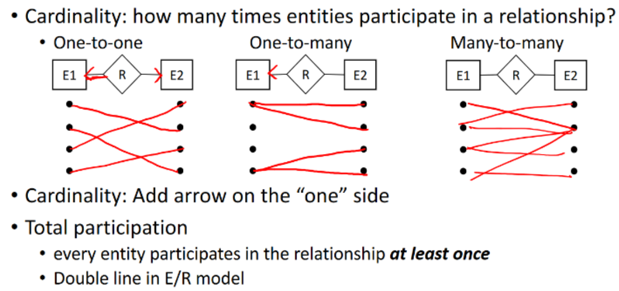
ER Model

The graphical intuitive and informal representation of data, entities relations and attributes

Contains: Entity, entity set, key, relationship, relationship set.

Cardinality of relationships:

1. One to One each entity in E1 is related to at most 1 entity in E2 and vice versa.
2. Many to One: each entity in E1 is related to at most one entity in E2
3. Many to Many: each entity in E1 may be related to 0 or more entities in E2 and vice versa.
4. Total participation: an entity participates in the relationship at least once.



3. This problem is based on an E/R design for a database used in a tech company shown in Figure 1. This database stores information about programmers. Each programmer has a name, which uniquely identifies the programmer. A programmer may in fact be a team leader who in turn leads a team of programmers. For example, Elaine leads a team consisting of Michael and Bryan. Bryan works on project C. Michael works on project A and in turn leads a team consisting of Jane and David who work on project A and B respectively. Each team leader is also associated with the name of the team he leads.

   Convert the E/R diagram to relations. For the translation of subclasses, assume that we generate one table per each subclass, instead of creating one gigantic table for the ISA relationship.

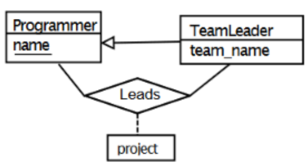
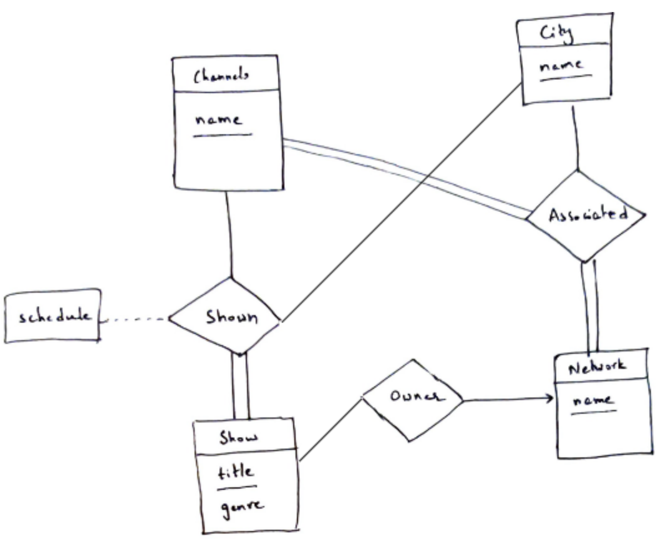


Figure 1: E/R diagram for a tech company

```
Programmer(name)
TeamLeader(name, team)
Leads(TeamLeader.name, Programmer.name, project)
```

Weak Entity Set: entity set without a unique key, double rectangle in ER model.



Many shows in one network, therefore error points to network (as that is the "one").

- A given channel in a given city is associated with one network.
- A given show is either owned by a network (and shown on a channel associated with that network) or is a local show and may be shown on any channel.
- Not all shows are shown in all cities, and the days and times for a given show may differ from city to city.
- You may ignore cable channels, which generally are not city-dependent.

Dashed lines in ER diagrams connect relationship sets to attributes.

SQL

Problem C. SQL — 20 Points

- Given the table: taken(StNo, CourseID, Year, Quarter, Sec, Grade, Remarks):
- write an SQL query to find the students who got a grade less than class average in 7 or more classes they took —a class is identified by (CourseID, Year, Quarter, Sec) and Grade in taken is of type numeric. In your answer, the depth of nesting of sub-queries should not exceed 2.
- ANSWER. We are seeking students who got a grade less than class average in 7 or more classes they took

```
select  StNO from taken as T1
  where  Grade  > (select avg(Grade) from taken as T2
                                          where T2.CourseID=T1.CourseID and
                                         T2.Year= T1.Year and
                                        T1.Quarter= T2.Quarter and
                                      T2.Sec= T1.Sec)
                                      )
group by T1.StNO having count(*) >=7
```

Bag Semantics: a set with duplicate elements and order does not matter. {a, a, b, c} = {a, c, b, a} != {a, b, c}

Under bag semantics, would R U S = S U R (Yes), R ∩ S = S ∩ R? (yes), R ∩ (S U T) = (R ∩ S) U (R ∩ T)? (no)

Joins:

You can have, left outerjoin, right outerjoin, full outerjoin(not available in mysql), R innerjoin s on r.a = s.a, and then r natural join s.

Diff between innerjoin and natural join, innerjoin will return more columns than natural join as it keeps both of the inner join elements (on r.a = s.a) keeps both! But natural join condenses it to just a.

We also do outerjoins if we want null values of a certain table when we combine.

(b) We want to find the movie stars who are not movie executives.

i. Write the query using EXCEPT operator.

ANSWER:

```
(SELECT name FROM MovieStar) EXCEPT (SELECT name FROM MovieExec)
```

ii. Write the query without using EXCEPT operator.

ANSWER:

```
SELECT name FROM MovieStar WHERE name NOT IN (SELECT name FROM MovieExec)
```

```
(SELECT name, address FROM MovieStar WHERE gender='F') INTERSECT (SELECT name,
address FROM MovieExec WHERE netWorth>1000000)
```

ii. Write the query without using INTERSECT operator.

ANSWER:

```
SELECT name, address FROM MovieStar WHERE gender='F' AND (name, address) IN (SELECT
name, address FROM MovieExec WHERE netWorth>1000000)
```

Expression	minimum #tuples	maximum #tuples
$R \cup \rho_{S(A,B)}(S)$		
$\pi_{A,C}(R \bowtie S)$		
$\pi_B(R) - (\pi_B(R) - \pi_B(S))$		
$(R \bowtie R) \bowtie R$		
$\sigma_{A>B}(R) \cup \sigma_{A<B}(R)$		

1. Min: r (when $S \subseteq R$), Max: $r + s$ (when $R \cap S =$)
2. Min: 0 (when all R.B values are different from S.B values), Max: $r \times s$ (when all R.B = S.B = b)
3. Min: 0, Max: s. This expression is equivalent to $\pi_B(R) \cap \pi_B(S)$
4. Min: r, Max: r. $R \bowtie R$ is always R
5. Min: 0 (when A=B for every tuple in R), Max: r (when A≠B for every tuple in R)

(a) $\pi_{R1.B}(\sigma_{R1.B=R2.B \wedge R1.A \neq R2.A}(\rho_{R1}(R) \times \rho_{R2}(R)))$

True

(b) `SELECT B FROM R GROUP BY B HAVING COUNT(*) > 1`

False

(a) `SELECT B FROM R WHERE NOT EXISTS(SELECT * FROM S WHERE R.B = S.B)`

(b) `(SELECT B FROM R) EXCEPT (SELECT B FROM S)`

Relational Design Theory

1. Suppose that we decompose the schema $R(A,B,C,D,E,F)$ into (A,B,C,F) and (A,D,E). When the following set of functional dependencies hold, is the decomposition lossless?
$A \to BC, CD \to E, B \to D, E \to A$
Explain your answer.

ANSWER:
(A,B,C,F) INTERSECT (A,D,E) $=$ A, and A is a key for (A,D,E), so the decomposition is lossless.

Convert a non BCNF to a BCNF table:

For any R in the schema:

if (nontrivial X -> Y holds on R AND X does not contain a key), then

1. Compute X+ (X+: closure of X)

2. decompose R into R1(X+) and R2(X,Z) X become common attributes Z: all attributes in R except X+

Repeat until no more decomposition

Assume the following set of functional dependencies hold for the relation $R(A,\ B,\ C,\ D,\ E,\ F)$:
$A \to BC,\ C \to E,\ B \to D$
Is it in **BCNF**? Explain your answer. If it is not, normalize it into a set of relations in **BCNF**.
ANSWER:
It is not in BCNF.
The key is AF, so A $\to$ BC, C $\to$ E and B $\to$ D all violate BCNF.
$R(A,B,C,D,E,F) \Longrightarrow R1(A,B,C,D,F) and R2(C,E) using C \to E$
$R1(A,B,C,D,F) \Longrightarrow R3(A,B,C,F) and R4(B,D) using B \to D$
$R3(A,B,C,F) \Longrightarrow R5(A,F) and R6(A,B,C) using A \to BC$

The final BCNF tables are:
$R2(C,E)$
$R4(B,D)$
$R5(A,F)$
$R6(A,B,C)$

FD X -> Y leads to redundancy only if X does not contain a key.

Lossless-join decompositions:

decompositions R(X,Y,Z) -> R1(X,Y), R2(Y,Z) is lossless join if Y->X or Y-> Z shared attributes are the key of one of the decomposed tables

this condition can be checked using FDs

Trivial Functional Dependency

trivial FD: X -> Y is a trivial functional dependency when Y is a subset of X.

nontrivial FD: X -> Y when Y is not a subset of X

Completely non trivial FD: X -> Y where X ∩ Y = empty set