

Relational Algebra

Definitions

- ↳ attribute \Rightarrow cols. in a relation
- ↳ tuples \Rightarrow rows in a relation
- ↳ domain \Rightarrow type of an attr.
- ↳ schema \Rightarrow the structure of a relation
- ↳ instance \Rightarrow content of a relation
- ↳ key \Rightarrow attr. that uniquely identifies a tuple

Uses SET SEMANTICS

- ↳ No dups + no ordering

Operators

- ↳ σ_R
- ↳ $\pi_R \Rightarrow \pi_A(\pi_B R)$
- ↳ $R \times S$
 - ↳ output per pair of input tuples
- ↳ SET SEMANTICS
- ↳ $R \bowtie S$
 - ↳ Not comm., generalizes into \times
 - ↳ $R \bowtie R = R$
- ↳ $\rho_R \Rightarrow$ rename
- ↳ $R \cup S, R - S, R \cap S$
 - ↳ Should have same schema
 - ↳ \cap not comm. $\Rightarrow R \cap S = R - (R - S)$

Data Modification

- ↳ INSERT INTO <schema>
VALUES (<val> | <relations>)
- ↳ DELETE FROM <schema>
WHERE <cond>
- ↳ UPDATE <schema>
SET A = An...
WHERE <cond>

Operators

- ↳ SELECT
↳ + DISTINCT
- ↳ FROM
- ↳ WHERE
- ↳ UNION, INTERSECT, EXCEPT
↳ SET SEMANTICS
↳ + ALL for bag semantics
- ↳ NOT IN, IN
- ↳ ALL, SOME
↳ Ex: $a > \text{ALL } R$
- Subqueries
 - ↳ Nested SELECT
↳ $1 \times 1 \Rightarrow$ scalar
 - Correlated subquery
 - ↳ References a table in the outer clause
 - ↳ EXISTS ()
 - Common Table Expr.
 - ↳ WITH alias > AS <subquery>

ORDER BY / FETCH FIRST

- ↳ ORDER BY <attr> ASC / DESC
- ↳ OFFSET > FETCH FIRST <count> ROWS ONLY
- ↳ change bag semantics
- ↳ LIMIT <count> OFFSET <offset>
- General SELECT
 - ↳ SELECT...
 - FROM...
 - WHERE...
 - GROUP BY...
 - HAVING...
 - ORDER BY...
 - FETCH FIRST...
- NULLS
 - ↳ null + arithmetic ops = null
 - ↳ DIV by 0 = null
 - ↳ null \neq null
 - ↳ logical ops w/ null = Unknown
 - ↳ Aggregates ignore tuples w/ null
 - ↳ NOT COUNT(*)
 - ↳ Agg(NULL) = NULL
 - ↳ COUNT() = 0
 - ↳ Setups treat null normally
 - ↳ IS NULL or IS NOT NULL
 - ↳ COALESCE() \Rightarrow 1st non-NULL

Outer join

- ↳ preserves dangling tuples
- ↳ LEFT or RIGHT

Recursion

- ↳ With RECURSIVE
- ancestor (<id>, <number>)
- AS (SELECT *
FROM Parent)
UNION
(SELECT P.<id>, A.<number>
FROM Parent P, ancestor A
WHERE P.parent = A.<id>))
- SELECT ancestor
FROM Ancestor
WHERE <id> = 'Succ.'

Basic SQL


- Types
 - ↳ CHAR(N)
 - ↳ VARCHAR(N)
 - ↳ INT
 - ↳ DECIMAL(N,M)
 - ↳ N: digits, M: after dec.
 - ↳ REAL
 - ↳ DOUBLE
 - ↳ TIMES TAMP
- TABLE creation
 - ↳ CREATE TABLE <name domain, ...>
 - ↳ PRIMARY KEY (attr) \Rightarrow can't be NULL
 - ↳ UNIQUE (attr)
 - ↳ DEFAULT <val>
 - ↳ DROP TABLE <schema>


Advanced SQL

- Aggregate func.
 - ↳ AVG, SUM, COUNT, MIN, MAX
 - ↳ results \Rightarrow single tuple
 - ↳ shouldn't appear in WHERE
 - ↳ GROUP BY: forms groups, where a single tuple exists for each unique attr
 - ↳ SELECT has agg. funcs on unique attr. only
 - ↳ HAVING: checks aggregate cond.
- Window Func.
 - ↳ FUNCTION (attr) OVER()
 - ↳ Applies agg. over all inputs
 - ↳ PARTITION BY: agg. over partition
- Case Expr.
 - ↳ CASE <attr> WHEN <cond> THEN <expr> ELSE <expr> END

E/R Model

↳ 2 sets: entity and relationship

↳ entity: 

↳ relationship: 

↳ Cardinality

↳ # entities participate

↳ 1-to-1, 1-to-many, many-to-many

↳ minimum "1" side

↳ total participation =

↳ General notation

↳ a..b, * = infinite

Role: designated to edges from an entity set

to label the role of that entity

Superclass and Subclass

↳ Specialization and Generalization

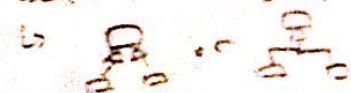
↳ Subclass inherits all attr. of superclass

↳ Subclass participates in the relationship of the superclass

↳ Subclass may participate in its own relationship

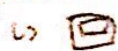
↳ Disjoint specialization vs. overlapping specialization

↳ Exclusive vs. multiple specialization



Weak Entity Set

↳ Not enough attributes



↳ Key comes from linked entity sets

↳ Identifying relationship: weak entity set and owner



Normalization

FD

↳ $U[S] \Rightarrow U[SID, name] \Rightarrow (100, James)$

↳ $X \rightarrow Y \Rightarrow$ no tuple can have x_1, x_2 but $y_1 \neq y_2$

↳ Trivial: $(SID) \rightarrow SID$

Logical implication

↳ $A \rightarrow B, B \rightarrow C \Rightarrow A \rightarrow C$

Closure

↳ F^+ is the set of FDs logically implied by F

↳ A^+ is set of all attributes that are functionally determined by X

↳ start with $X^+ = X$ and repeat

↳ X^+ is a key if $X^+ = \{ \text{all attr} \}$

Lossless Decomp. is when the shared attr. are keys for one of the tables

BCNF

↳ For every non-trivial FD, $(X \rightarrow Y) \in F^+$, X contains a key

↳ For any R in the schema

If (non-trivial $X \rightarrow Y$ holds and X does not contain a key)

Compute X^+

Decompose R into $R_1(X^+)$ and $R_2(R - X^+)$

↳ X is common

↳ Z is all attr. in R except X

Repeat

↳ lossless, not unique