Relational Algebra On Sets

Hui Chen [a]

[a]CUNY Brooklyn College, Brooklyn, NY, USA

March 3, 2022

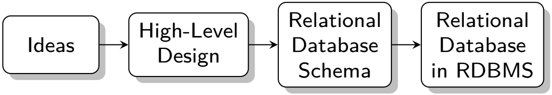
Outline

1. Introduction to SQL

2. Relational Algebra
   - Selection
   - Projection
   - Product
   - θ-Join
   - Natural Join
   - Renaming
   - Queries

3. Assignment

Overview


Ideas
High-Level Design
Relational Database Schema
Relational Database in RDBMS

Outline

1. Introduction to SQL

2. Relational Algebra
   - Selection
   - Projection
   - Product
   - θ-Join
   - Natural Join
   - Renaming
   - Queries

3. Assignment

Introduction to SQL

SQL (pronounced as "sequel") is the principal language used to describe and manipulate relational database, and has several aspects:

- Data definition language (DDL).
  - SQL includes commands to create database objects such as tables, indexes, and views, as well as commands to define access rights to those database objects.
  - Topics of this lecture: SQL commands to create database tables (relations)
- Data manipulation language (DML).
  - SQL includes commands to insert, update, delete, and retrieve data within the database tables.
- Transaction control language (TCL).
  - The DML commands in SQL are executed within the context of a transaction.
- Data control language (DCL).
  - Data control commands are used to control access to data objects.

To Discuss Subset of DML

- Relational Algebra
- Query Databases

Outline

1. Introduction to SQL

2. Relational Algebra
   - Selection
   - Projection
   - Product
   - θ-Join
   - Natural Join
   - Renaming
   - Queries

3. Assignment

Why Relational Algebra

- SQL, incorporates relational algebra at its center, and many SQL programs are really "syntactically sugared" relational algebra expressions
- When a DBMS processes queries, it first the first thing translate queries into relational algebra or a very similar internal representation.
- Help us understand queries.

What's Algebra?

A mathematical system consisting of

- Operands – variables or values from which new values can be constructed.
- Operators – symbols denoting procedures that construct new values from given values.

Relational Algebra

An algebra whose operands are relations or variables that represent relations.

- designed to do the most common tasks with relations in a relational database
- can be used as a query language for relations

Core Relational Algebra Operations

- Union, intersection, and difference – usual set operations.
  - both operands (relations) must have the same relation schema
- Selection. picking certain tuples (rows)
- Projection. picking certain columns
- Products and joins: compositions of relations
- Renaming of relations and attributes

Selection

$$R_1 := \sigma_C(R_2) \tag{1}$$

where C is a condition, R_1 and R_2 are two relations, and R_1 is all those tuples of R_2 that satisfy condition C.

Selection: Example

Given a relation `Items`

Code	Desc	Price
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	19.99
213345	9v Battery	1.99
254467	100W bulb	1.99
311452	Power Drill	59.99

What is $\sigma_{Price \geq 10}(Items)$?

Selection: Example

What is $\sigma_{Price \geq 10}(Items)$?

Code	Desc	Price
123457	Lamp	25.15
123458	Box Fan	19.99
311452	Power Drill	59.99

Projection

$$R_1 := \pi_L(R_2) \tag{2}$$

where R_1 and R_2 are two relations, L is a list of attributes from the schema of R2.

- R_1 is constructed by looking at each tuple of R_2, extracting the attributes on list L, in the order specified, and creating from those components a tuple for R_1.
- Eliminate duplicate tuples, if any.

Projection: Example

Given a relation `Items`

Code	Desc	Price
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	19.99
213345	9v Battery	1.99
254467	100W bulb	1.99
311452	Power Drill	59.99

What is $\pi_{Price}(Items)$?

Projection: Example

What is $\pi_{Price}(Items)$?

Price
5.26
25.15
19.99
1.99
59.99

Extended Projection

Using π_L operator, we allow the list L to contain arbitrary expressions involving attributes:

- Arithmetic on attributes, e.g., $A + B \rightarrow C$.
- Duplicate occurrences of the same attribute.

Extended Projection: Example

Given a relation `Items`

Code	Desc	Price
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	19.99
213345	9v Battery	1.99
254467	100W bulb	1.99
311452	Power Drill	59.99

What is $\pi_{Code,Desc,Price,Price*0.9\rightarrow DiscountPrice}(Items)$?

Extended Projection: Example

What is $\pi_{Code,Desc,Price,Price*0.9 \rightarrow DiscountPrice}(Items)$?

Code	Desc	Price	DiscountPrice
123456	Flashlight	5.26	4.73
123457	Lamp	25.15	22.64
123458	Box Fan	19.99	17.99
213345	9v Battery	1.99	1.79
254467	100W bulb	1.99	1.79
311452	Power Drill	59.99	53.99

Product

$$R_3 := R_1 \times R_2 \tag{3}$$

1. Pair each tuple $t_{R_1,i}$ of R_1 with each tuple $t_{R_2,j}$ of R_2.
2. Concatenate $t_{R_1,i}t_{R_2,j}$, a tuple of R_3.
3. Schema of R_3 is the attributes of R_1 and then R_2, in order.
4. However, if attribute A of the same name in R_1 and R_2, use $R_1.A$ and $R_2.A$.

Product: Example

Given relation `Items` and `Location`

Code	Desc	Price
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	19.99
213345	9v Battery	1.99
254467	100W bulb	1.99
311452	Power Drill	59.99

Store	Aisle	Shelf	Code
23	W	5	123457
24	K	9	213345
25	Z	6	311452

What is $Items \times Location$?

θ-Join

$$R_3 := R_1 \bowtie_C R_2 \tag{4}$$

1. Take the product $R_1 \times R_2$.
2. Then apply σ_C to the result, where C can be any boolean-valued condition.

θ-Join: Example

Given relation `Items` and `Location`

Code	Desc	Price
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	19.99
213345	9v Battery	1.99
254467	100W bulb	1.99
311452	Power Drill	59.99

Store	Aisle	Shelf	Code
23	W	5	123457
24	K	9	213345
25	Z	6	311452

What is $Items \bowtie_{Items.Code=Loations.Code} Location$?

Natural Join

$$R_3 := R_1 \bowtie R_2. \tag{5}$$

A variant of join where,

- Equating attributes of the same name, and
- Projecting out one copy of each pair of equated attributes.

Natural Join: Example

Given relation `Items` and `Location`

Code	Desc	Price
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	19.99
213345	9v Battery	1.99
254467	100W bulb	1.99
311452	Power Drill	59.99

Store	Aisle	Shelf	Code
23	W	5	123457
24	K	9	213345
25	Z	6	311452

What is $Items \bowtie Location$?

Renaming

$$R_1 := \rho R_1(A_1, \ldots, A_n)(R_2) \tag{6}$$

or the simplified notation

$$R_1(A_1, \ldots, A_n) := R_2 \tag{7}$$

which makes R_1 be a relation with attributes $A_1, \ldots, A_n$ and the same tuples as R_2.

Combining Operations to Form Queries

Combine operators with parentheses and precedence rules to construction expressions

Precedence Rules

Precedence of relational operators:

1. σ, π, ρ (highest).
2. $\times$, $\bowtie$
3. $\cap$
4. $\cup$, $-$ (lowest)

Queries: Examples

Query: What are the name and student ID of the students who major in Computer Science and has a GPA 3.0 or higher?

Solution 1

1. Select those Students tuples that have GPA ≥ 3.0
2. Select those Students tuples that has `major = 'CS'`
3. Compute the intersection of the above two
4. Project the relation from the above onto attributes name and ID

$$\pi_{name,id}(\sigma_{gpa\geq 3.0}(Students) \cap \sigma_{major='CS'}(Students)) \tag{8}$$

Solution 2 – can we answer the query with a single selection operation?

Outline

1. Introduction to SQL

2. Relational Algebra
   - Selection
   - Projection
   - Product
   - θ-Join
   - Natural Join
   - Renaming
   - Queries

3. Assignment

Assignment

Let's work on an assignment using paper and pencil/pen ...