Relational Algebra

IDATG2204 Data Modelling and Database Systems

Norwegian University of Science and Technology

Where are We Now?

- W02: Introduction, Relational Algebra
- W03: SQL
- W04: SQL, Conceptual Modelling
- W05: Conceptual Modelling
- W06: Normalisation
- W07: Logical Modelling, NOSQL
- W08: DB Application Development
- W09: DB Security, Project Kick-off
- W10-W14: Project Work with Peer Review
- W15: Indexing, query processing, concurrency
- W16: Recovery
- W17: More SQL and NOSQL
- W18: Review and Wrap-up

Why Relation Algebra

- Help us understand typical relational query operations without worrying about the SQL syntax
- Relational algebra is important in the implementation of RDBMS'es

Sample Relations

car

id	*make*	*model*	*year*	*mileage*	*fuel*	*type*	*price*	*dealer_id*
1	Volkswagen	Passat	2017	97805	diesel	station wagon	425000	Bdf
2	Mazda	CX-3	2019	19777	petrol	suv	378900	Gjvk
3	Volkswagen	UP!	2017	16551	electric	hatchback	125000	Bdf
4	Toyota	RAV4	2019	39661	hybrid	suv	428900	Hmr
5	Mercedes-Benz	C Class	2004	301204	diesel	sedan	31707	Hmr
6	Audi	Q3	2020	18516	diesel	suv	624900	Hrst

dealer

id	*city*	*county_no*
Bdf	Bardufoss	54
Bo	Bodø	18
Elv	Elverum	34

county

no	*name*
30	Viken
3	Oslo
34	Innlandet

Outline

- Relational algebra operations:
  - Unary operations
  - Set and join operations
  - Aggregation and grouping operations
  - (Division operation)

Unary Operations

- A unary operation defines a new relation based on one input relation
  - Projection
  - Selection
  - Rename operation

Projection

- Defines a relation that contains a subset of columns from the input relation
- Notation:
  - $\Pi_{a1, \ldots, an}(R)$
- Examples:
  - Car id, make, model, year only:
    - $\Pi_{id, make, model, year}(car)$
  - Car id, year, mileage, price only:
    - $\Pi_{id, year, milage, price}(car)$
  - Car id, make, dealer_id only:
    - $\Pi_{id, make, dealer\_id}(car)$

Selection

- Defines a relation that contains a subset of rows from the input relation that satisfy the specified condition
- Notation:
  - $\sigma_{\texttt{predicate}}(\texttt{R})$
- Examples:
  - Mazda cars only:
    - $\sigma_{\texttt{make='Mazda'}}(\texttt{car})$
  - No cars older than 2018:
    - $\sigma_{\texttt{year >= 2018}}(\texttt{car})$
  - No cars older than 2018 and mileage larger than 20,000 km:
    - $\sigma_{\texttt{year >= 2018 AND mileage <= 20000}}(\texttt{car})$

Select conditions

- Basic conditions:
  - Comparison: =, <>, <, <=, >, >=
  - Range: `BETWEEN`, `NOT BETWEEN`
    - `year BETWEEN 2013 AND 2015`
  - Set membership: `IN`, `NOT IN`
    - `county_name IN ('Viken', 'Oslo', 'Innlandet')`
  - Pattern match: `LIKE`, `NOT LIKE`
    - `model LIKE 'CX%'`
  - Null: `IS NULL`, `IS NOT NULL`
    - `price IS NULL`
- Logical expression:
  - `NOT`, `AND`, `OR`
  - Combined with parenthesis, if needed

Combining Selections and Projections

- Examples:
  - Car id, make, model, and year for cars older than 2010:
    - $\Pi_{\text{id, make, model, year}}(\sigma_{\text{year < 2010}}(\text{car}))$
  - which can be decomposed to:
    - $R \leftarrow \sigma_{\text{year < 2010}}(\text{car})$
      $\Pi_{\text{id, make, model, year}}(R)$
  - which is equivalent to:
    - $\sigma_{\text{year < 2010}}(\Pi_{\text{id, make, model, year}}(\text{car}))$
  - But:
    - $\Pi_{\text{id, make, model}}(\sigma_{\text{year < 2010}}(\text{car}))$
  - is not equivalent to:
    - $\sigma_{\text{year < 2010}}(\Pi_{\text{id, make, model}}(\text{car}))$

(Rename Operation)

- Provides a new name for a relation - and optionally the attribute names:
- Notation:
  - $\rho_{S(a1, \ldots, an)}(R)$
- Example:
  - Rename the county relation to Norwegian
    - $\rho_{fylke(nr, navn)}(county)$

Outline

- Relational algebra operations:
  - Unary operations
  - Set and join operations
  - Aggregation and grouping operations
  - (Division operation)

Set and Join Operations

- A set and join operation defines a new relation based on two input relations
  - Union
  - (Difference)
  - (Intersection)
  - Cartesian product
  - Equijoin
  - (Theta join)
  - (Natural join)
  - (Semijoin)
  - Outer join

Union

- Defines a relation containing all the tuples or the two input relations; the two input relations must be union-compatible:
  - Same number of attributes
  - Corresponding attributes from the same domain
- Notation:
  - $R \cup S$
- Example:
  - Names of all counties, dealer cities, and car makers
    - $\Pi_{name}(county) \cup \Pi_{city}(dealer) \cup \Pi_{make}(car)$

Set and Join Operations

- A set and join operation defines a new relation based on two input relations
  - Union
  - (Difference)
  - (Intersection)
  - Cartesian product
  - Equijoin
  - (Theta join)
  - (Natural join)
  - (Semijoin)
  - Outer join

Cartesian Product

- Defines a relation that is the concatenation of every single tuple in one input relation with every single tuple of the other
- Notation:
  - $R \times S$
- Example:
  - All combinations of county name and city for county number 18 and 30:
    - $\Pi_{name}(\sigma_{no \text{ IN } (18,30)}(county)) \times \Pi_{city}(\sigma_{county\_no \text{ IN } (18,30)}(dealer))$
  - Decomposed:
    - $R \leftarrow \Pi_{name}(\sigma_{no \text{ IN } (18,30)}(county))$
      $S \leftarrow \Pi_{city}(\sigma_{county\_no \text{ IN } (18,30)}(dealer))$
      $R \times S$

Equijoin

- Defines a relation that contains tuples satisfying the equality predicate F from the Cartesian product of the two input relations
- Notation:
  - $R \bowtie_F S$
- Example:
  - Car and corresponding dealer information:
    - $\texttt{car} \bowtie_{\texttt{dealer\_id = dealer.id}} \texttt{dealer}$
  - Dealer and corresponding county information:
    - $\texttt{dealer} \bowtie_{\texttt{county\_no = no}} \texttt{county}$
  - Car, dealer, and county information joined:
    - $\texttt{car} \bowtie_{\texttt{dealer\_id = dealer.id}} \texttt{dealer} \bowtie_{\texttt{county\_no = no}} \texttt{county}$

Set and Join Operations

- A set and join operation defines a new relation based on two input relations
  - Union
  - (Difference)
  - (Intersection)
  - Cartesian product
  - Equijoin
  - (Theta join)
  - (Natural join)
  - (Semijoin)
  - Outer join

Outer join

- Defines a join where tupples in one input relation that do not match tupples in the other are also included
- Notation:
  - $R \overset{\leftarrow}{\bowtie}_F S$ (left outer join)
  - $R \overset{\rightarrow}{\bowtie}_F S$ (right outer join)
- Example:
  - All counties including dealer cities where these exist:
    - $\text{county} \overset{\leftarrow}{\bowtie}_{\text{county\_no = no}} \text{dealer}$
  - All cars and dealers, including dealers that have no cars:
    - $\text{car} \overset{\rightarrow}{\bowtie}_{\text{dealer\_id = dealer.id}} \text{dealer}$
  - All counties, cities, and corresponding cars:
    - $\text{county} \overset{\leftarrow}{\bowtie}_{\text{county\_no = no}} \text{dealer} \overset{\rightarrow}{\bowtie}_{\text{dealer\_id = dealer.id}} \text{car}$

Outline

- Relational algebra operations:
  - Unary operations
  - Set and join operations
  - Aggregation and grouping operations
  - (Division operation)

Aggregation

- Defines a relation that contains one tuple holding the value(s) of the listed aggregate function(s)
- Notation:
  - $_{AL}(R)$
- Example:
  - Number of dealers in the company:
    - $_{COUNT\ id}(dealer)$
  - Result column renamed to `dealerCount`:
    - $\rho_{R(dealerCount)}(_{COUNT\ id}(dealer))$
  - Min, max, and average price of a Volkswagen car:
    - $_{MIN\ price,\ MAX\ price,\ AVG\ price}(\sigma_{make='Volkswagen'}(car))$

Aggregation Functions

- Common ones are:
  - `COUNT` -- Count number of values
  - `SUM` -- Sum of the values
  - `AVG` -- Average of the values
  - `MIN` -- Smallest value
  - `MAX` -- Largest value

Grouping

- Groups the tuples of a relation by one or more attributes and applies aggregate function(s) to each group.
- Notation:
  - $_{GA}\,\mathcal{A}_{L}(R)$
- Example:
  - Number of cars of each make:
    - $_{make}\,\mathcal{COUNT}_{id}(car)$
  - Min, max, and average price per car make and model:
    - $_{make,\ model}\,\mathcal{MIN}_{price},\ \mathcal{MAX}_{price},\ \mathcal{AVG}_{price}(car)$
  - Number of dealers per county - for ALL counties:
    - $_{name}\,\mathcal{COUNT}_{id}\ (county \bowtie_{county\_no\ =\ no} dealer)$