

Relational Algebra

CISC437/637, Lecture #6

Ben Carterette

Copyright © Ben Carterette

1

Relational Query Languages

- A query language allows manipulation and retrieval of data from a database
- The relational model supports simple but powerful query languages
 - Formal foundation based on logic
 - Allows for a great deal of optimization
- A query language is not a programming language
 - Not Turing-complete, not intended for complex calculations
 - Supports easy, efficient access to data

Copyright © Ben Carterette

2

Formal Query Languages

- Two mathematical formalisms:
 - Relational algebra, based on operators over relations
 - Relational calculus, based on declarative statements about data
- The algebra more directly supports computation
 - A relational algebra query implies a sequence of steps that can be taken to execute it
- SQL is an implementation of relational algebra

Copyright © Ben Carter@Co

3

Preliminaries

- A query is applied to *relation instances*, and the result of a query is a relation instance
 - Schemas of input relations are fixed
 - Schemas of output relations are also fixed, though may be different from input relation schemas
- Positional versus named-field notation:
 - Positional notation is easier for formal definitions; named-field easier to read
 - SQL supports both

Copyright © Ben Carter@Co

4

Relational Algebra

- **Basic operators:**
 - **Selection:** $\sigma(R)$ – select a subset of records from relation R
 - **Projection:** $\pi(R)$ – drop unwanted fields from relation R
 - **Cross-product:** $R1 \times R2$ – concatenate each record in R1 with each record in R2
 - **Set-difference:** $R1 - R2$ – return records in R1 that are not in R2
 - **Union:** $R1 \cup R2$ – return records in either R1 or R2
- Algebra is all about **composing** operators
 - Every operator takes relations as input and returns relations
 - Algebra is **closed** under these operators

Copyright © Ben Carter@Ge

5

Relational Algebra

- **More operators:**
 - **Intersection:** $R1 \cap R2$ – return records in both R1 and R2
 - **Join:** $R1 \bowtie R2$ – combine information from relations R1 and R2
 - **Division:** $R1/R2$ – return records in R1 that “match” every record in R2 in a subset of fields
 - **Renaming:** $\rho(R(F), E)$ – the relation returned by expression E is named R and its fields are renamed according to mapping F
 - **Aggregate functions:** $G_f(R)$ – calculate aggregating function f on relation R

Copyright © Ben Carter@Ge

6