Relational Algebra II

DSC 301: Lecture 11

March 12, 2021

Lecture Objectives

- Review basic relational algebra
- Advanced (relational) algebra
- Introducing JOINS

Relational Algebra

Recall from Lecture 3 that a relation (a.k.a. table, tuple) is a subset of a Cartesian product and a relational algebra defines a set of operation on relations. An essential property of ALL algebras is the **closure** property, which says that two elements from a set combined (in some way, e.g., $+$) will remain within the set. Symbolically, if $a, b \in R$, then $a \odot b \in R$ where $\odot$ is some operation.

Basic Operations on Relation Algebras

- **Selection**: A subset of rows, $\sigma_p(R)$, where p is a predicate and R is a relation.

```
SELECT * FROM Flights WHERE distance >1000;
```

- **Projection**: A subset of columns, $\Pi_{A_1,A_2,\ldots,A_k}(R)$

```
SELECT carrier, dest FROM Flights;
```

- **Rename**: (alias), $\rho_A(R) \to S$.

```
SELECT dest as City FROM Flights;
```

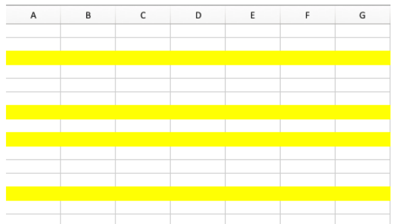

Figure 1: Selection

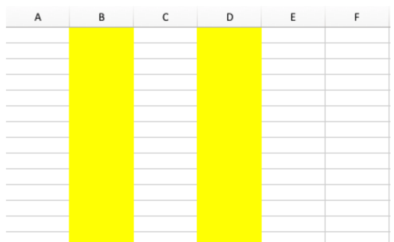

Figure 2: Projection

Join Operations*

A **join** combines two or more tables[1] to retreive data from multiple tables. Tables included in the join are listed in the `FROM` clause. The `JOIN` itself is contained in the `WHERE` clause. The common operators that can be used to join tables include: $=, <, >, \leq, \geq, <>, !=$, `BETWEEN`, `LIKE`, and `NOT`. Probably the most frequent operator used in joins is the $=$. The common joins are:

1. Inner join, $R \bowtie S$
   a. Most common
   b. a.k.a. Equijoin
   c. Returns rows that have matching values in both tables
2. Outer joins
   a. Left, $R \ltimes S$ - Returns all rows from the "left" table and the matched records from the "right" table
   b. Right, $R \rtimes S$ - Returns all rows from the "right" table and the matched records from the "left" table
   c. Full - Returns all records when there is a match in either the left or right table.

[1] Combine one table with itself.

3. Cross join $R \times S$ - Returns EACH records from left table and ALL records from the right table.
    a. WARNING!!!! Can be dangerous!!!! Too many records return, slow query.
4. Self-join
5. Semi-joins
6. Theta joins
7. Natural join
    a. Inner join can be classified as a natural inner join

```
SELECT
        Table1.column1, Table2.column2
FROM
        Table1 INNER JOIN Table2
ON
        Table1.somecol = Table2.samecol;
```

Example: Note the table aliases.

```
SELECT
    O.id, O.date, C.fname, C.lname
FROM
    Orders AS O
        INNER JOIN
    Customers AS C ON O.customer = C.id
ORDER BY O.date;
```

```
SELECT
        Table1.column1, Table2.column2
FROM
        Table1, Table2
WHERE
        Table1.somecol = Table2.samecol;
```

Example: Note the table aliases.

```
SELECT
    O.id, O.date, C.fname, C.lname
FROM
    Orders AS O,
    Customers AS C
WHERE
    O.customer = C.id
ORDER BY O.date;
```