SQL and Relational Algebra

Introduction to Data Science - University of Washington

April 24, 2015

1 Union

Does not include duplicates:

```
SELECT * FROM R1
UNION
SELECT * FROM R2
```

Includes duplicates:

```
SELECT * FROM R1
UNION ALL
SELECT * FROM R2
```

2 Difference

Removes rows in R1 that also exist in R2.

```
SELECT * FROM R1
EXCEPT
SELECT * FROM R2
```

Note - Intersection can be found using Union and Difference, or using Join.

3 Selection

Returns all tuples which satisfy a certain condition. This can involve any sort of function that returns a boolean value.

```
SELECT * FROM R1
WHERE salary > 40000 AND Name = "Smith"
```

4 Project

Eliminates columns. Note that duplicates will remain unless you explicitly get rid of them.

```
SELECT name, lastname FROM R1
```

4.1 Join

Here are two ways to join two tables that share a certain ID variable (equi-join):

```
SELECT * FROM R1, R2
WHERE R1.A = R2.B
```

```
SELECT * FROM R1 JOIN R2
ON R1.A = R2.B
```

In practice, the queries are equivalent because the program will optimize the relational algebra anyway.

A theta-join is a join with a condition. Assume that we've written a distance function that takes two locations.

```
SELECT DISTINCT h.name
FROM Hospitals h, Schools s
WHERE distance(h.location, s.location) < 5
```

Left outer join takes all of the tuples from R1, takes R2 if it matches but is set to NULL otherwise. Right outer join is the opposite. Full outer join takes all tuples, pads out the missing values wtih NULLs.

```
SELECT * FROM R1 OUTER JOIN R2
ON R1.A = R2.B
```

5 Nesting

We can select from another query:

```
FROM (
SELECT * FROM R1
) x
```

If we want to add in a constant variable column, we can include this in our select statement:

```
SELECT 5.0 as binsize
```

We can also use aggregated functions such as *avg*() with the GROUP BY clause.

```
SELECT binid, avg(height) as height FROM R1
GROUP BY binid
ORDER BY binid asc
```

6 Case

```
CASE WHEN (x.day > 2)
THEN ...
WHEN (...)
THEN ...
END AS myFunction
```

7 User defined functions

- Scalar functions - Can appear anywhere.
- Aggregate functions - In SELECT clause, requires a GROUP BY clause.
- Table functions - Comes after FROM.

Note that Microsoft SQL Server has support for user-defined functions written in SQL, but SQLite does not! PostgreSQL and Greenplum have support for Python and R functions.

8 Physical optimization

We can use the keyword EXPLAIN to understand what is going on behind the scenes. Generally we don't need to worry about optimization because the database program will do this for us.