This Lecture

- How the database internally optimizes your query
  - Query trees
  - Algebraic manipulations
  - Heuristic execution
- How you can manually "tune" your database queries
  - The EXPLAIN command
  - Indexes

Query Trees

Consider the following query, using example tables from the book:
```
SELECT p.pnumber,
p.dnum,
e.lname,
e.address,
e.bdate
FROM project p, department d, employee e
WHERE p.dnum = d.dnumber
AND d.mgr_ssn = e.ssn
AND p.plocation = 'Stafford'
```

One of many possible trees!

Canonical tree (query as written)
Query Trees

- There are many possible trees
- Goal: modify some starting tree via algebraic manipulations to reduce query cost
- In general: reduce # of rows and columns working their way up the tree
  - Apply most restrictive conditions as "low" in the tree as possible (reduce rows)
  - Move projections as low as possible (reduce columns)

Algebraic Manipulations

1. A conjunction of selections $\rightarrow$ a sequence of selections
2. Selections are commutative
3. In a sequence of projections, all but the last can be ignored
4. Selection commutes with projection (as long as columns are available to selection)
5. Joins and cross products are both commutative
6. Selections commute with joins/cross products when selection applies to a single table participating in the join
7. Projections commute with joins/cross products (again as long as relevant columns remain available)
8. Set operations union and intersection commute
9. Joins, cross products, set union, and set intersection are all associative

Heuristic to Improve Trees

1. Break up conjunctive selections into a sequence of selections
2. Move each selection as far down the tree as possible:
   - Where selections involve a single table, they can be moved to a leaf
   - Where selection involve 2 tables, move to just above the cross product
3. Rearrange leaf nodes (manipulations 5, 9) so most restrictive selections are executed first (but make sure to avoid cross-products due to lack of join condition)
4. Combine cross products with selections to form joins
5. Move projections as far down the tree as possible
6. Replace entire subtrees with algorithms where possible

Heuristic Example

(On whiteboard)

Query Tuning Example

(On whiteboard)

Next Time

- Transactions and transaction control
- ACID