**INT206 Lab 3**

**Query Optimization**

The query optimization is divided into two types: **Heuristic** (sometimes called Rule-based) **Optimization** and **Systematic** (sometimes called Cost-based) **Optimization**.

**Part 1: Understanding Cost-based Optimization**

### The Optimizer Cost Model

To generate execution plans, the optimizer uses a cost model that is based on estimates of the cost of various operations that occur during query execution. The optimizer has a set of compiled-in default “cost constants” available to it to make decisions regarding execution plans.

The optimizer also has a database of cost estimates to use during execution plan construction. These estimates are stored in the server\_cost and engine\_cost tables in the mysql system database and are configurable at any time. The intent of these tables is to make it possible to easily adjust the cost estimates that the optimizer uses when it attempts to arrive at query execution plans.

#### The Cost Model Database

The optimizer cost model database consists of two tables in the mysql system database that contain cost estimate information for operations that occur during query execution:

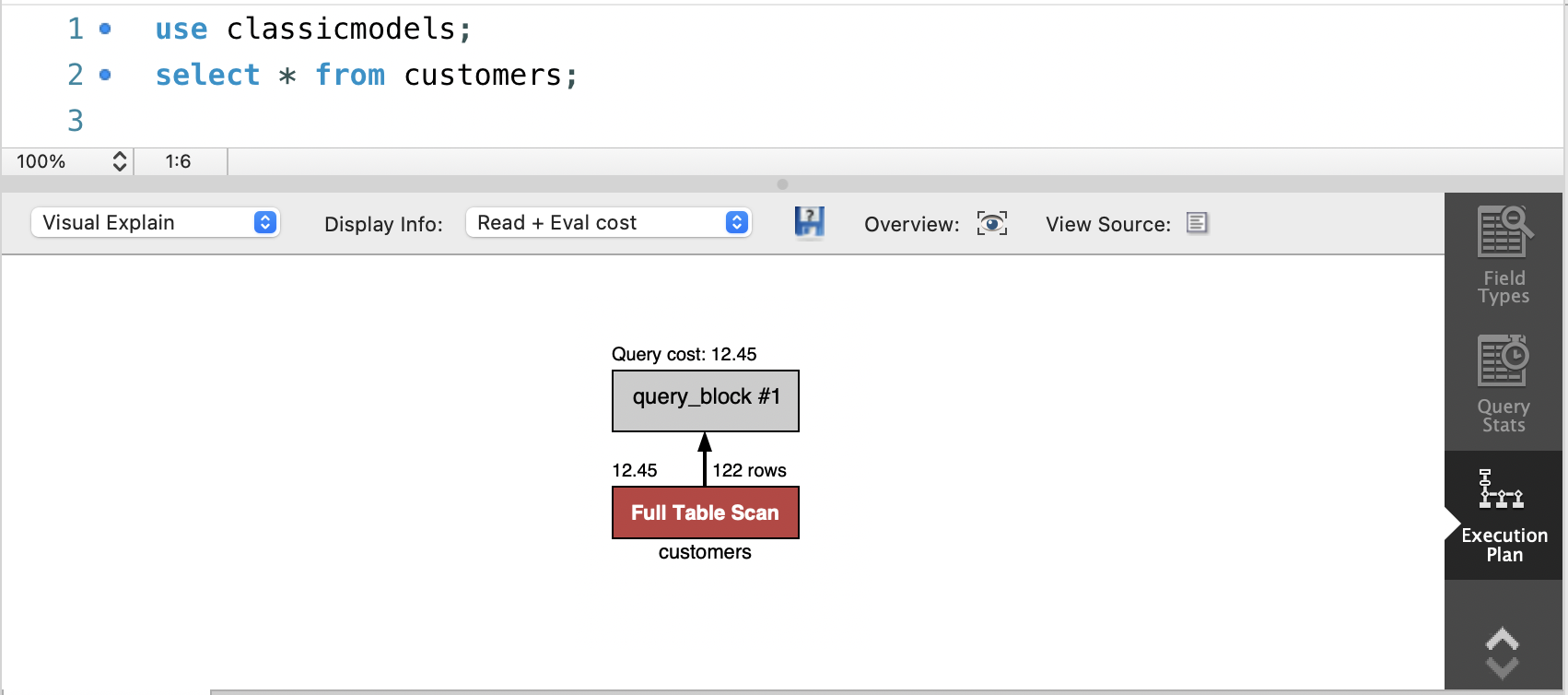
* server\_cost: Optimizer cost estimates for general server operations
* engine\_cost: Optimizer cost estimates for operations specific to particular storage engines

## Visual Explain Plan

The visual explain feature generates and displays a visual representation of the MySQL [EXPLAIN](https://dev.mysql.com/doc/refman/8.0/en/explain.html) statement by using extended information available in the extended JSON format. MySQL Workbench provides all of the EXPLAIN formats for executed queries including the raw extended JSON, traditional format, and visual query plan.

### Visual Explain Usage

To view a visual explain execution plan, execute your query from the SQL editor and then select **Execution Plan** within the query results tab. The execution plan defaults to Visual Explain, but it also includes a Tabular Explain view that is similar to what you see when executing [EXPLAIN](https://dev.mysql.com/doc/refman/8.0/en/explain.html) in the MySQL client. For information about how MySQL executes statements, see [Optimizing Queries with EXPLAIN](https://dev.mysql.com/doc/refman/8.0/en/using-explain.html).

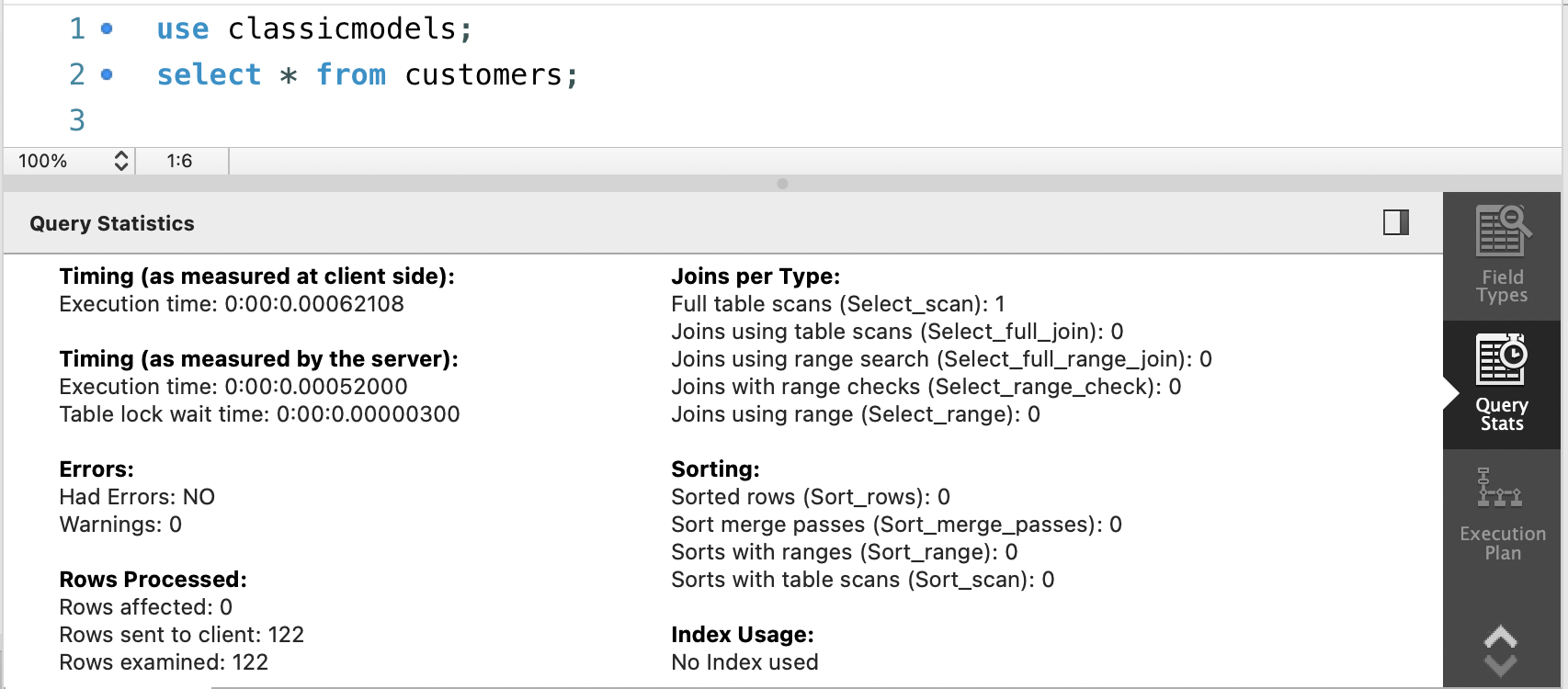


### Visual Explain Conventions

The order of execution in a visual explain diagram is bottom to top and left to right. The diagram examples that follow provide an overview of the graphic, textual, and informational conventions used to represent aspects of the visual explain plans. For specific information, see:

* [Graphic Conventions](https://dev.mysql.com/doc/workbench/en/wb-performance-explain.html%23wb-performance-visual-explain-graphics)
* [Textual and Informational Conventions](https://dev.mysql.com/doc/workbench/en/wb-performance-explain.html%23wb-performance-visual-explain-text-info)

The **Query Stats** SQL editor results tab uses Performance Schema data to gather key statistics collected for executed query, such as timing, temporary tables, indexes, joins, and more.



References:

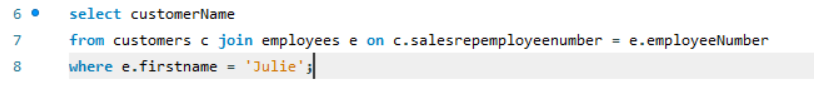
* <https://dev.mysql.com/doc/refman/8.0/en/cost-model.html>
* <https://www.oreilly.com/library/view/high-performance-mysql/9780596101718/ch04.html>

Graphical user interface, diagram

Description automatically generated

**Practice:**

1. Write a statement to display customer name of all customers who has sales employee first named “Julie”. Capture the screen and paste the information below.



Graphical user interface, text, application, chat or text message

Description automatically generated

1. Write a statement or click on the results tab to display the query cost and query stats of the statement in Question 1. Capture two screens and paste the information below.

* Query Cost

Diagram

Description automatically generated

* Query Stats

Graphical user interface, text, application

Description automatically generated

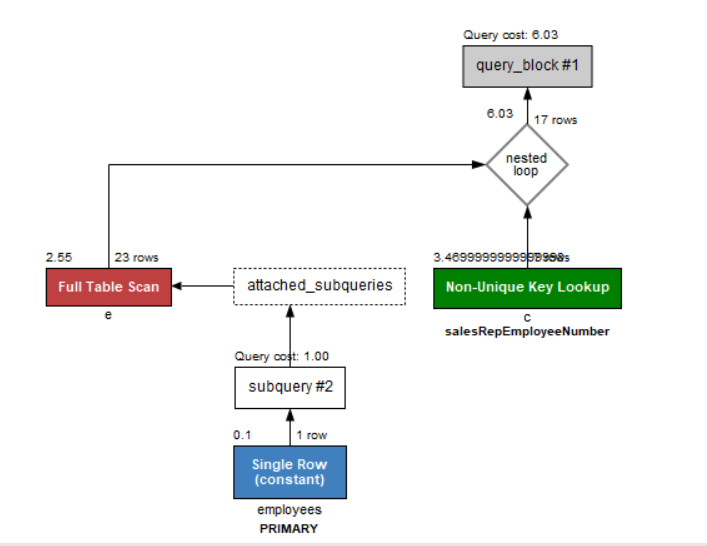
1. Write a different statement to display the same result of Question 1. [Hint: Use Nested Query/Join/Set Operation] and sort the result by the customer name in descending order.

Graphical user interface, text, application, email

Description automatically generated

1. Write a statement or click on the results tab to display the query cost and query stats of the statement in Question 3. Capture two screens and paste the information below.

- Query Cost





- Query Stats

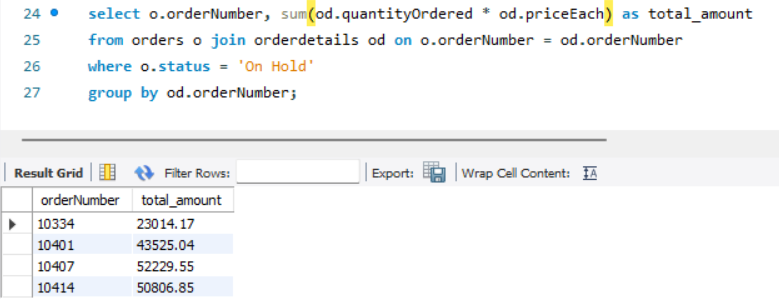
Graphical user interface, text, application

Description automatically generated

1. Compare the results in terms of cost and statistics between Question 2 and Question 4 and describe the different results based on your understanding.

Cost in Q2(6.03) is not equal to Q4(7.02), Maybe because subquery is more complicate

1. Write a statement to display order number and total amount of the orders that the status is “On Hold”.



1. Write a statement or click on the results tab to display the query cost and query stats of the statement in Question 6. Capture two screens and paste the information below.

- Query Cost

Diagram

Description automatically generated

- Query Stats

Graphical user interface, text, application

Description automatically generated

8.Please improve performance (Time/Cost) of the query in Question 6 and show your

method and result. Capture the screens and paste the information below.

**Part 2: Understanding Heuristic/Rule-based Optimization**

Rule-based Optimization

* A query can be represented as a tree data structure. Operations are at the interior nodes and data items (tables, columns) are at the leaves.
* The query is evaluated in a depth-first pattern.

SELECT pnumber, dnum, lname

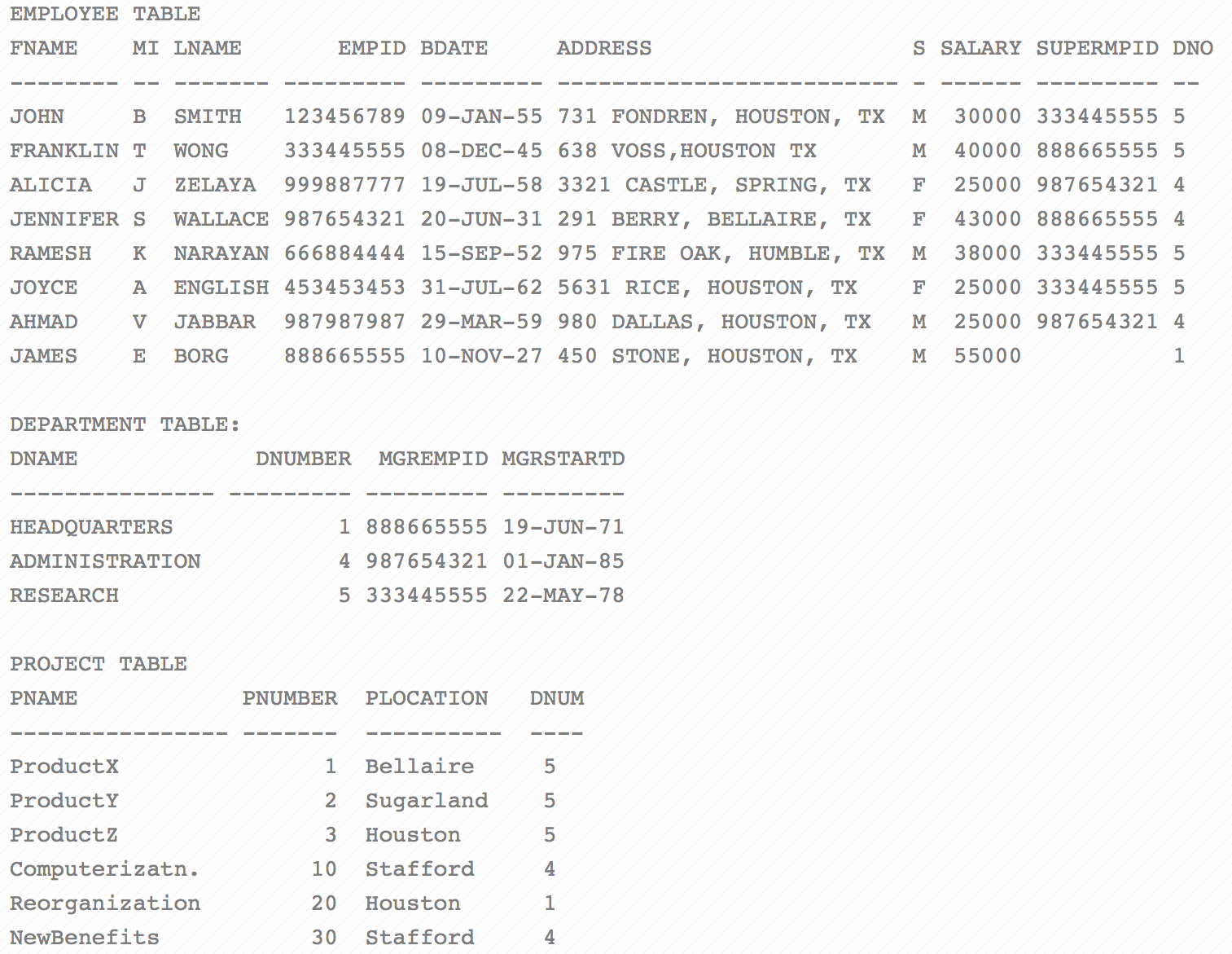
FROM project, department, employee

WHERE dnum=dnumber AND mgrempid=empid AND plocation = 'Stafford';

Or, in relational algebra:

image4.gif

on the following schema:



**Question:** Which of the following query trees is more efficient?

|  |  |  |
| --- | --- | --- |
| A | image9.gif | image3.gif |
| B | image2.gif | image10.gif |

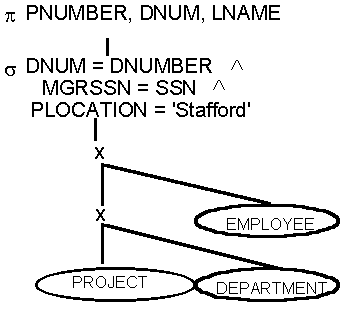
* Note the two cross product operations. These require lots of space and time (nested loops) to build.
* After the two cross products, we have a temporary table with 144 records (6 projects \* 3 departments \* 8 employees).
* **An overall rule for heuristic query optimization is to perform as many select and project operations as possible before doing any joins.**

These transformations can be used in various combinations to optimize queries. Some general steps follow:

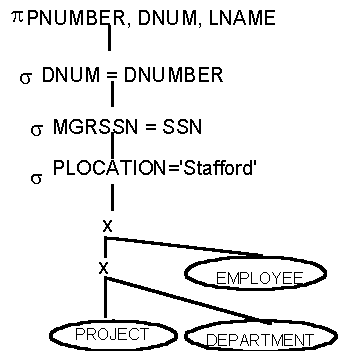
1. Using the cascading selections rule, break up conjunctive selection conditions and chain them together.
2. Using the commutativity rules, move the selection operations as far down the tree as possible.
3. Using the associativity rules, rearrange the leaf nodes so that the most restrictive selection conditions are executed first. For example, an equality condition is likely more restrictive than an inequality condition (range query).
4. Combine cartesian product operations with associated selection conditions to form a single Join operation.
5. Using the commutativity of projection rules, move the projection operations down the tree to reduce the sizes of intermediate result sets.
6. Finally, identify subtrees that can be executed using a single efficient access method.

**Example of** Rule-based Optimization

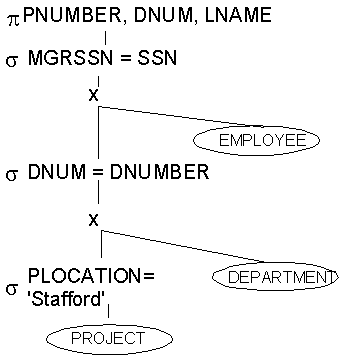
1. Original Query Tree



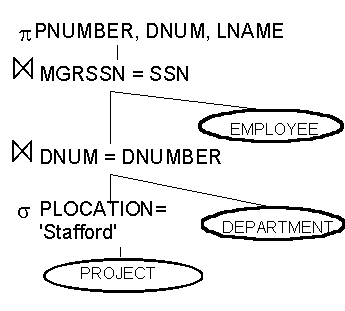
1. Use the cascading selections rule to break up cascading selections



1. Commute selection with cross product



1. Combine cross product and selection to form join



**Practice:**

1**.** Evaluate the following SQL statement:

SELECT e.lname

FROM employee e, works\_on w, project p

WHERE p.name = ‘Aquarius’ AND p.no = w.no

AND w.ssn = e.ssn AND e.birthdate > ’31-DEC-1957’ ;

Transform the SQL statement to Relational Algebra Tree and use Heuristic Rules to convert the Relational Algebra Tree (RAT) into an equivalent form that is more efficient.

Write a Relational Algebra Tree step by step.

2. Evaluate the following SQL statement:

SELECT  S.FNAME, P.STREET

FROM BRANCH B ,  STAFF S ,  PROPERTYFORRENT P

WHERE B.BRANCHNO = S.BRANCHNO

AND   S.STAFFNO = P.STAFFNO

AND   ( (S.POSITION='Assistant' AND P.CITY='London')

OR

  ( S.POSITION='Assistant' AND P.ROOMS>5  ) ) ;

Transform the SQL statement to Relational Algebra Tree and use Heuristic Rules to convert the Relational Algebra Tree (RAT) into an equivalent form that is more efficient.

Write a Relational Algebra Tree step by step.

(#Hint: use the rules of Boolean algebra to simplify the Boolean expressions.)