# ROBUST QUERY PROCESSING: *Mission Possible!*

Jayant Haritsa
Database Systems Lab
Indian Institute of Science, Bangalore

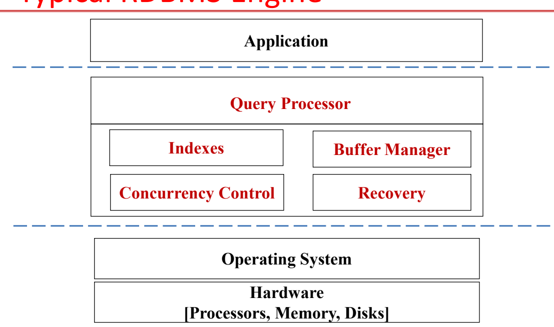


## Relational DBMS

- Workhorse of today's Information Industry
  - Commercial
    - IBM DB2, MS SQL Server, Oracle Exadata, HP SQL/MX
  - Public-domain
    - PostgreSQL, MySQL, Berkeley DB
- Extensively researched for over four decades
  - Journals
    - ACM TODS, IEEE TKDE, VLDBJ, …
  - Conferences
    - ACM SIGMOD, IEEE ICDE, VLDB, EDBT, CIKM, …



## Typical RDBMS Engine

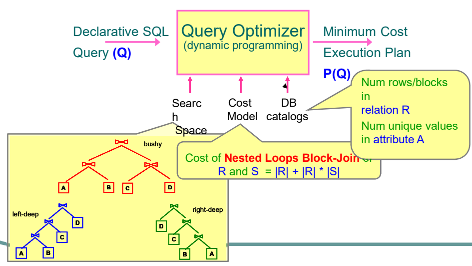





## Design of RDBMS Engines

- Transaction Processing (ACID)
  - WAL/ARIES for Atomicity/Recovery
  - 2PL for Concurrency Control
- Data Access Methods
  - B-trees/Hashing for Large Ordered Domains
  - Bitmaps for Small Categorical Domains
  - R-trees for Geometric Domains
- Memory Management
  - LRU-k (k=2 balances history and responsiveness)
- Query Processing (SQL)
  - "Black Art"

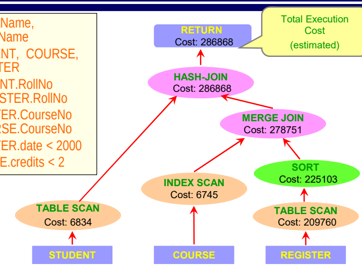
## Cost-based Query Optimization






## Plan Example

```
select  StudentName, CourseName
from    STUDENT, COURSE, REGISTER
where   STUDENT.RollNo = REGISTER.RollNo
and     REGISTER.CourseNo = COURSE.CourseNo
and     REGISTER.date < 2000
and     COURSE.credits < 2
```






## Relational Selectivities

- Cost-based Query Optimizer's choice of

  execution plan = f (query, database, system, …)

- For a given database and system setup,

  execution plan chosen for a query = f (selectivities of query's base relations)

  - selectivity is the estimated percentage of rows of a relation used in producing the query result



## Query Template QT7 [Q7 of TPC-H]

*Determines the values of goods shipped between nations in a time period*

```
select
   supp_nation, cust_nation, l_year, sum(volume) as revenue
from
(select n1.n_name as supp_nation,  n2.n_name as cust_nation,
        extract(year from l_shipdate) as l_year,
        l_extendedprice * (1 - l_discount) as volume
 from   supplier, lineitem, orders, customer, nation n1, nation n2
 where  ...okey and o_orderkey = ...
        ...key and s_nationkey = ...
        ... n2.n_nationkey and
        ((n1.n_name = 'FRANCE' and n2.n_name = 'GERMANY') or
        (n1.n_name = 'GERMANY' and n2.n_name = 'FRANCE')) and
        l_shipdate between date '1995-01-01' and date '1996-12-31'
        and  o_totalprice ≤ C1  and  c_acctbal ≤ C2  ) as shipping
group by supp_nation, cust_nation, l_year
order by supp_nation, cust_nation, l_year
```

Value determines selectivity of ORDERS relation

Value determines selectivity of CUSTOMER relation

# Query Execution Plans

- SQL is a declarative language
  - Specifies ends, not means

```
select  STUDENT.Name, COURSE.Title
from    STUDENT, COURSE, REGISTER
where   STUDENT.RollNo = REGISTER.RollNo  and
          REGISTER.CourseNo = COURSE.CourseNo
```

Unspecified: **join order** **[((S ⋈ R) ⋈ C) or ((R ⋈ C) ⋈ S) ?]**
**join technique** **[Nested-Loops / Sort-Merge / Hash?]**

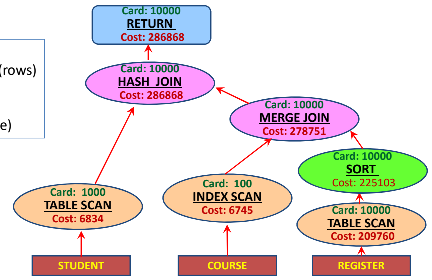
- DBMS query optimizer identifies the optimal evaluation strategy: "query execution plan"



# Sample Execution Plan

**Card:**
Output Cardinality (rows)

**Cost:**
Execution Cost (time)






# Query Optimization Framework

*Declarative Query (Q)* → **Query Optimizer** (Dynamic Programming) → *Optimal Plan P(Q)* [Min Cost]

| Operator Execution Cost Estimation Model | Operator Output Cardinality Estimation Model |
|---|---|
| Function of Hardware and DB Engine | Function of Data Distributions and Correlations |



# Run-time Sub-optimality

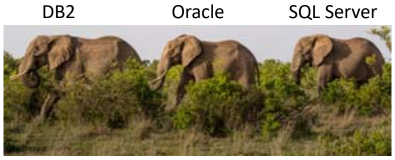
The supposedly optimal plan-choice may actually turn out to be highly sub-optimal (e.g. a 1000 times worse!) when the query is executed with this plan. This adverse effect is due to errors in:

(a) cost model
- Reasons: Simple linear models, operator-agnostic features, fixed coefficients, system dynamics …

(b) cardinality model
- Reasons: Coarse statistics, outdated statistics, attribute value independence (AVI) assumption, multiplicative error propagation, query construction, …

## What have the QP guys been doing all these years?

DB2 Oracle SQL Server



"Elephants"† are highly sensitive animals!

(† Stonebraker-speak for enterprise DBMS)



## Cardinality Sensitivity Example

EMPLOYEE

| EID | Name | Age |
|---|---|---|
| 1 | Cohen | 25 |
| 2 | Giuliani | 25 |
| 3 | Manafort | 25 |
| 4 | Melania | 25 |
| 5 | Ivanka | 25 |
| 6 | Donald | 25 |
| 7 | Jared | 25 |
| …. | …. | 25 |
| $10^9$ | Eric | 25 |

MANAGER

| MID | Name | Age |
|---|---|---|
| 1 | Trump | 50 |
| 2 | Pence | 50 |
| 3 | Mnuchin | 50 |
| 4 | Shanahan | 50 |
| 5 | Whitaker | 50 |
| 6 | Bernhardt | 50 |
| 7 | Perdue | 50 |
| …. | …. | 50 |
| $10^6$ | Ross | 50 |

**EMPLOYEE.AGE ⋈ MANAGER.AGE**

- Output cardinality of the join is **ZERO**
- One new employee aged **50** joins the company
- Output cardinality of the join jumps to a **million!**
- No summary mechanism can capture such **"nanoscopic"** changes



## Proof by Authority [Guy Lohman, IBM]

Snippet from April 2014 Sigmod blog post on

**"Is Query Optimization a "Solved" Problem?"**

The root of all evil, the Achilles Heel of query optimization, is the estimation of the size of intermediate results, known as cardinalities. The cardinality model can easily introduce errors of many orders of magnitude! With such errors, the wonder isn't "Why did the optimizer pick a bad plan?" Rather, the wonder is "Why would the optimizer ever pick a decent plan?"



## Sound-bites

- Little difference between worst-case and average-case in Query Processing
- It is far easier to win money at the Macau casinos than to get query processing right!

## Prior DB Research (lots!)

- Sophisticated estimation techniques
  - SIGMOD 1999/2010, VLDB 2001/2009/2011, …, CIDR 2019
  - e.g. wavelet histograms, self-tuning histograms, deep learning
- Selection of stable plans
  - SIGMOD 1994/2005/2010, PODS 1999/2002, VLDB 2008, …, VLDB 2017
  - e.g. Variance-aware plan selection
- Runtime re-optimization techniques
  - SIGMOD 1998/2000/2004/2005, …, arXiv 2019 [Stonebraker et al]
  - e.g. POP (progressive optimization), RIO (re-optimizer)

**Several novel ideas and formulations, but are they robust?**



## Is there any hope?

Over last decade, several promising advances that collectively promise to soon make robustness a contemporary reality – we will survey these techniques in the tutorial.



## *QP Robustness*



## ROBUSTNESS DEFINITION

- Multiple perspectives, no consensus
  - If worst-case performance is improved at the expense of average-case performance, is that acceptable?
  - Is it to be defined on a query instance basis, or "in expectation"?
  - …
- Ultimately, robustness definition is application dependent
- Graceful performance profile – no "cliffs"
- Seamless scaling with workload complexity, database size, distributional skew, join correlations
- Provable guarantees on worst-case performance (relative to an offline ideal that makes all the right decisions)

## Picasso Visualizer

Picasso is a (free) Java tool that, given an *n*-dimensional SQL query template and a choice of database engine, automatically generates plan and cost diagrams

- Operational on
  - DB2 • Oracle • SQLServer • Sybase • PostgreSQL • MySQL
- Additional Diagrams:
  - Cardinality Diagram
  - Plan-tree Diagram
  - Plan-difference diagram
  - Abstract-plan diagram
  - ….

DEMO