

**ORACLE®**

# ORACLE SQL Tuning

Valentin ROBU - [valentin.robu@oracle.com](mailto:valentin.robu@oracle.com)  
CSS Global Delivery, DB

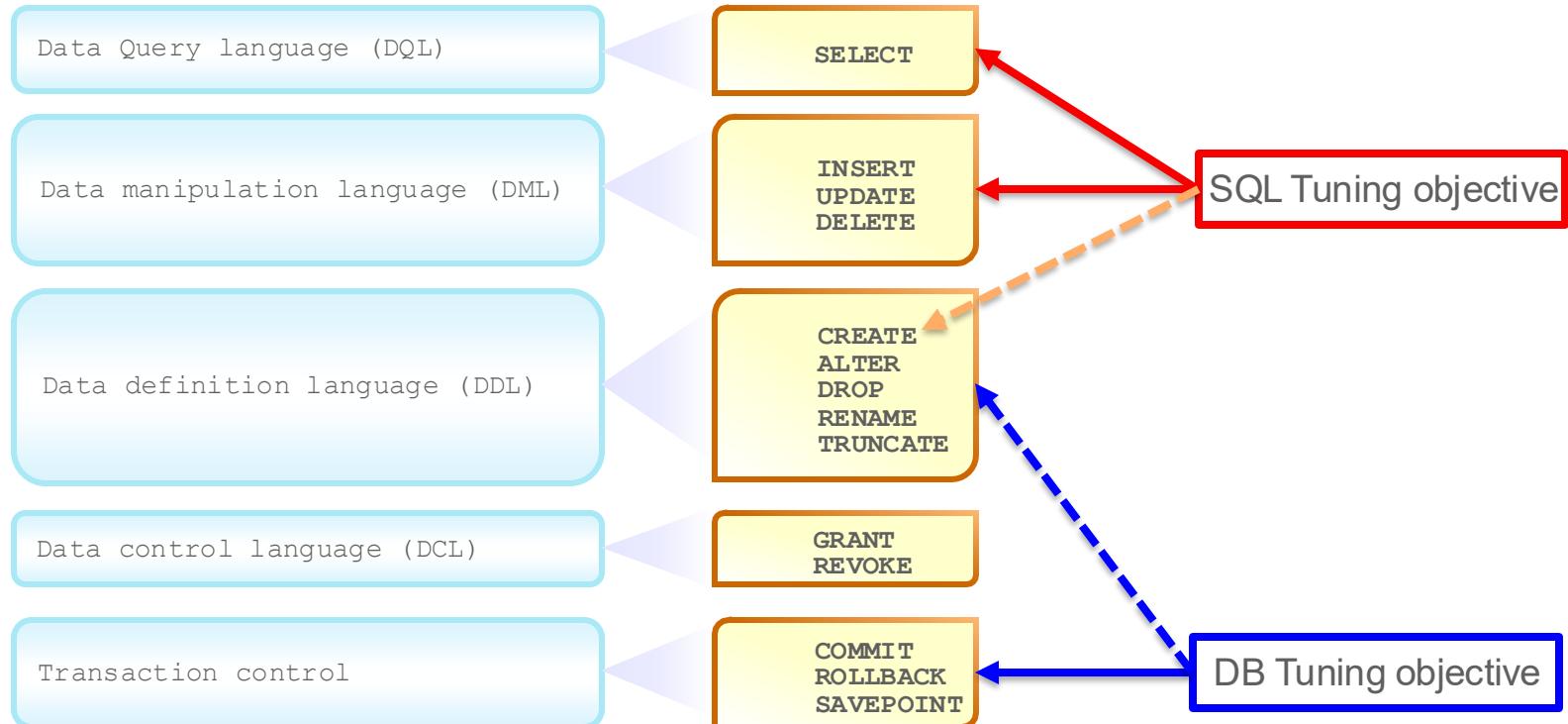
Oracle Academy - Octoberber, 2025



A woman with long brown hair and glasses, wearing a brown leather jacket over a blue patterned top, is sitting at a desk in what appears to be a library or study room. She is holding a black telephone receiver to her ear with her left hand and looking down at some papers on the desk with a focused expression. In the background, there are other people seated at desks, and large windows are visible.

# SQL Tuning – general considerations

# SQL Statement types

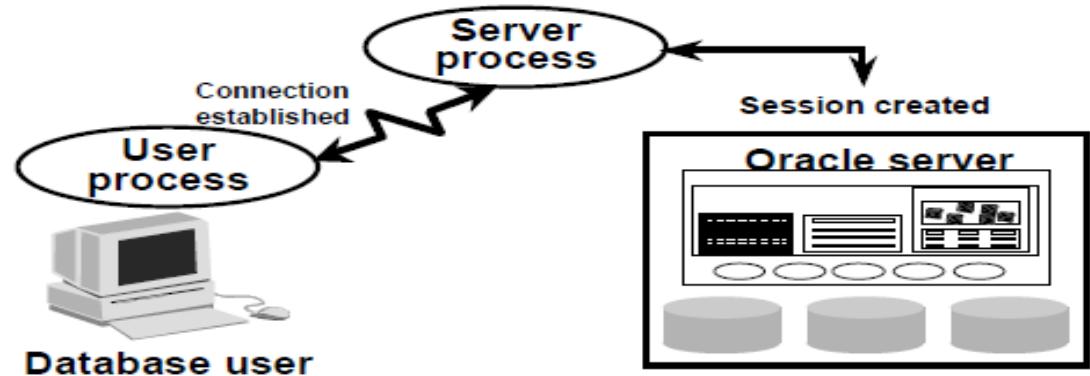


# SQL Tuning goal

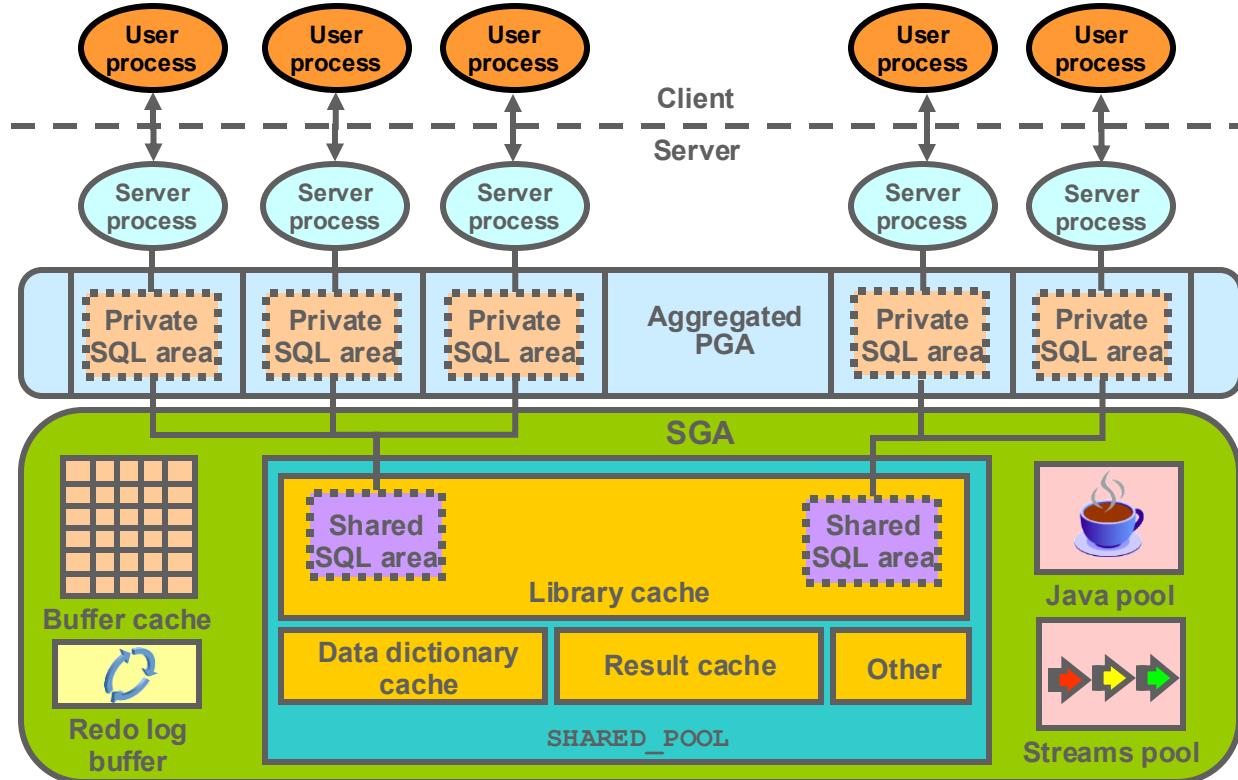
- Reduce overall resource consumption
- A **resource** could be:
  - Memory (consumed)
  - CPU (used)
  - IO operations (IO = Input Output)
  - Time:
    - Elapsed time
    - Parse time
    - Fetch time
    - Total response time
    - Etc.
  - Other

# Oracle Process types

- A **server process** is a program that directly interacts with the Oracle server.
  - It fulfills calls generated and returns results.
- A **client process** is a program running locally on client permanently connected with server process.

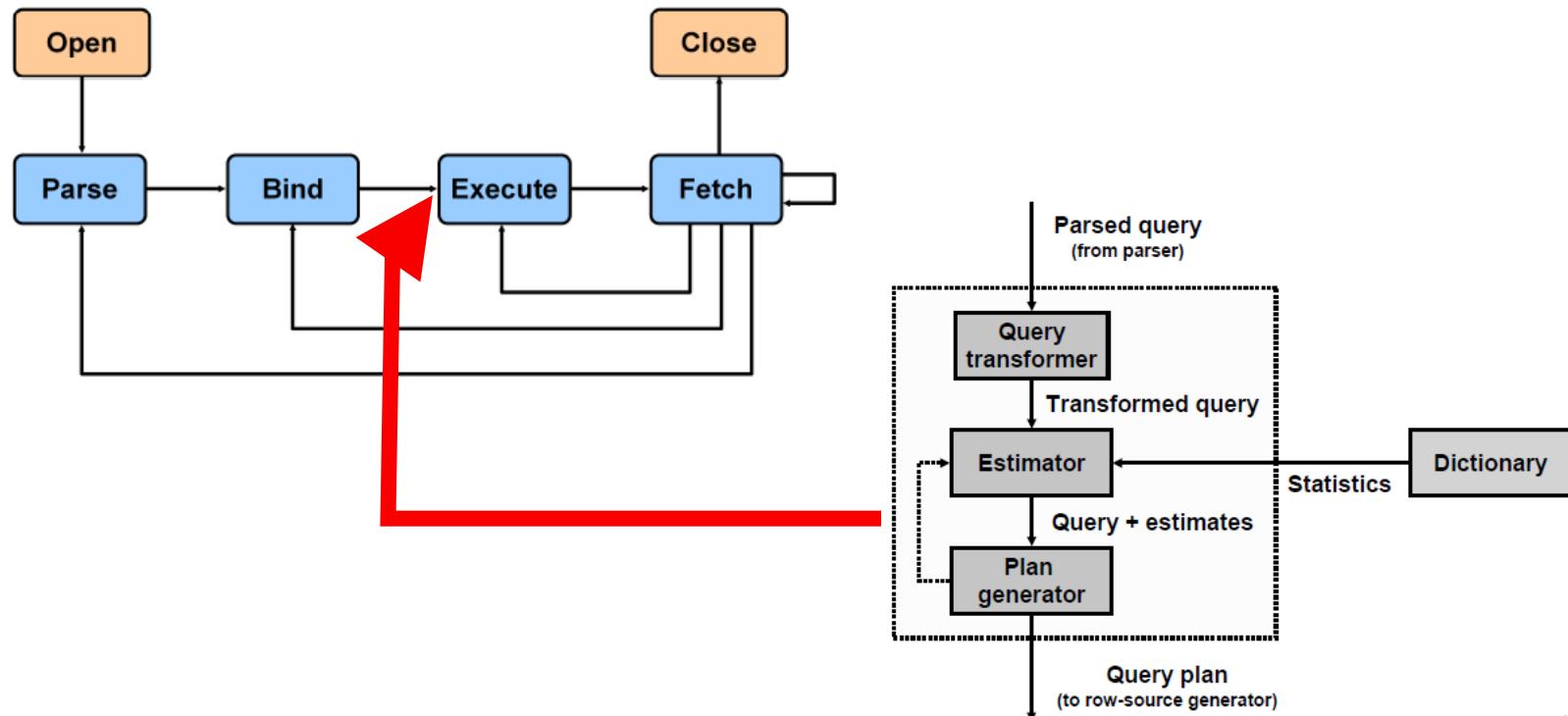


# SQL Statement Representation

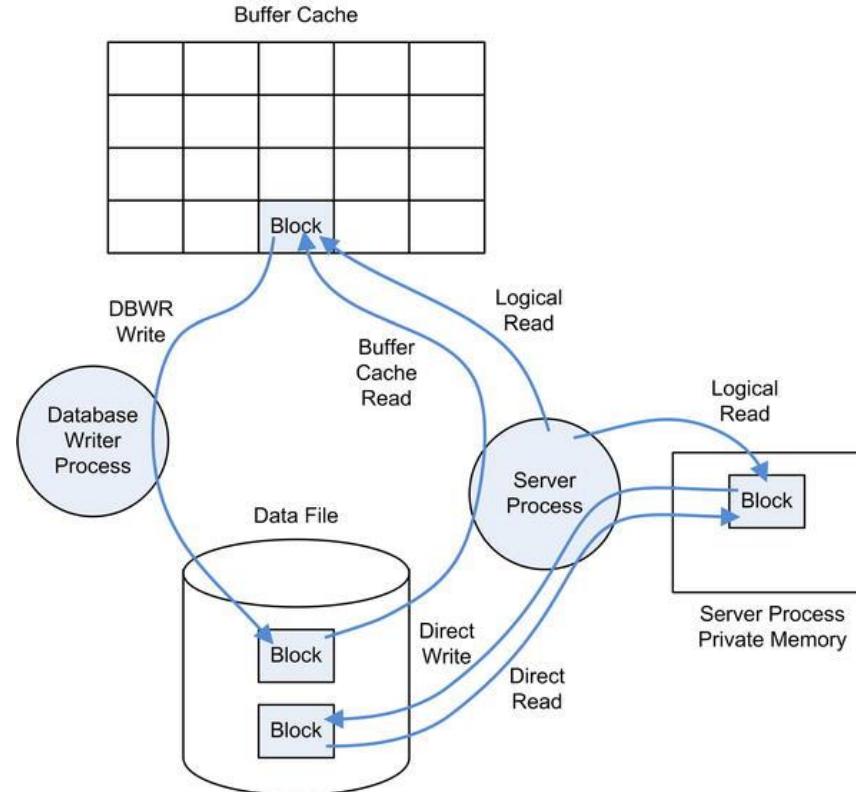


# SQL statement

# SQL Statement processing phases

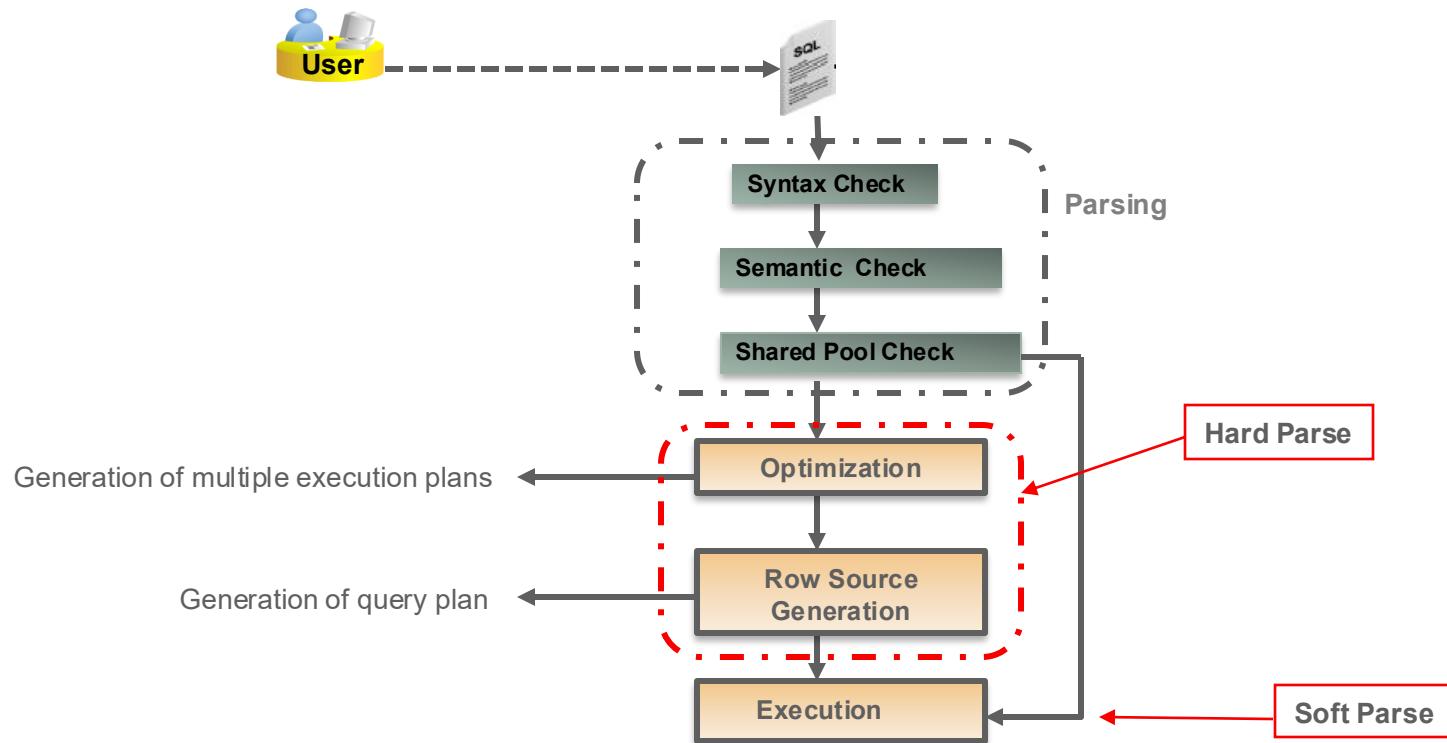


# Typical SQL Statement flow



# SQL phases details

# SQL Statement Processing phases



# Oracle Optimizer

- Determines the most efficient execution plan for a statement
- Optimizer:
  - Evaluates expressions and conditions
  - Uses objects and system's statistics
  - Decides how to access data
  - Decides how to join tables
  - Decides which access path is most efficient

# Query Estimator: COST

- Cost is the optimizer's best estimate of the number of standardized IOs it takes to execute a particular statement.
- Cost unit is a standardized single-block random read:
  - 1 cost unit = 1 SRd (SRd = Single Read)
- Example:
  - If a plan costs 1,000, the optimizer computes that it should take as long as 1,000 single-block reads.
  - Remember that it is an estimation.

# Query Estimator: Selectivity & Cardinality

- Selectivity is the estimated proportion of a row set retrieved by a particular predicate or combination of predicates.

$$\text{Selectivity} = \frac{\text{Number of rows satisfying a condition}}{\text{Total number of rows}}$$

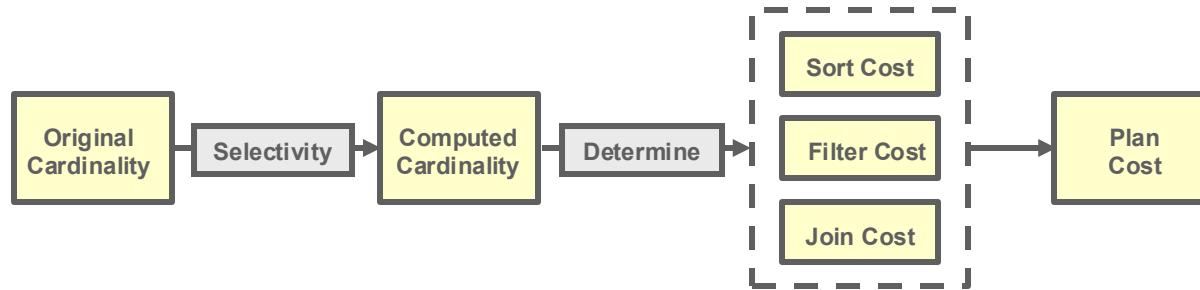
- Expected number of rows retrieved by a particular operation in the execution plan:

$$\text{Cardinality} = \text{Total number of rows} * \text{Selectivity}$$

- Selectivity is expressed as a value between 0.0 and 1.0 (or between 0 and 100%):
  - High selectivity: Small proportion of rows
  - Low selectivity: Big proportion of rows
- Selectivity computation:
  - If no statistics: Use dynamic statistics.
  - If no histograms: Assume even distribution of rows.

# Importance of Selectivity & Cardinality

- Selectivity affects the estimates of I/O cost.
- Selectivity affects the sort cost.
- **Cardinality is important to determine join, filter, and sort costs.**
- If incorrect selectivity and cardinality are used, the optimizer estimates the plan cost incorrectly.



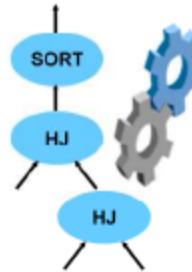
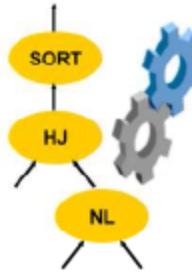
# Plan Generator

```
select e.last_name, d.department_name
from employees e, departments d
where e.department_id = d.department_id;
```

```
Join order[1]: DEPARTMENTS[D]#0 EMPLOYEES [E]#1
NL Join: Cost: 41.13 Resp: 41.13 Degree: 1
SM cost: 8.01
HA cost: 6.51
Best:: JoinMethod: Hash
Cost: 6.51 Degree: 1 Resp: 6.51 Card: 106.00
Join order[2]: EMPLOYEES [E]#1 DEPARTMENTS [D]#0
NL Join: Cost: 121.24 Resp: 121.24 Degree: 1
SM cost: 8.01
HA cost: 6.51
Join order aborted
Final cost for query block SEL$1 (#0)
All Rows Plan:
Best join order: 1
+-----+
| Id | Operation          | Name      | Rows | Bytes | Cost |
+-----+
| 0  | SELECT STATEMENT   |           |       |        |    7 |
| 1  | HASH JOIN           |           | 106  | 6042  |    7 |
| 2  | TABLE ACCESS FULL  | DEPARTMENTS| 27   | 810   |    3 |
| 3  | TABLE ACCESS FULL  | EMPLOYEES  | 107  | 2889  |    3 |
+-----+
```

# Oracle Execution Plan

An execution plan is a set of steps that optimizer performs when executes an SQL statement and performs an operation



# Reading an Execution Plan

Plan hash value: 1343509718

Id	Operation	Name	Rows	Bytes	Cost	(%CPU)
0	SELECT STATEMENT		106	2862	6	(17)
1	MERGE JOIN		106	2862	6	(17)
2	TABLE ACCESS BY INDEX ROWID	DEPARTMENTS	27	432	2	(0)
3	INDEX FULL SCAN	DEPT_ID_PK	27	1	1	(0)
* 4	SORT JOIN		107	1177	4	(25)
5	TABLE ACCESS FULL	EMPLOYEES	107	1177	3	(0)

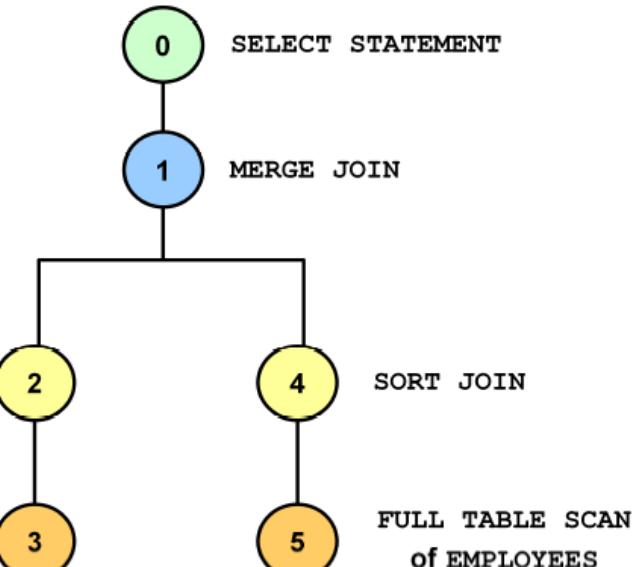
Predicate Information (identified by operation id):

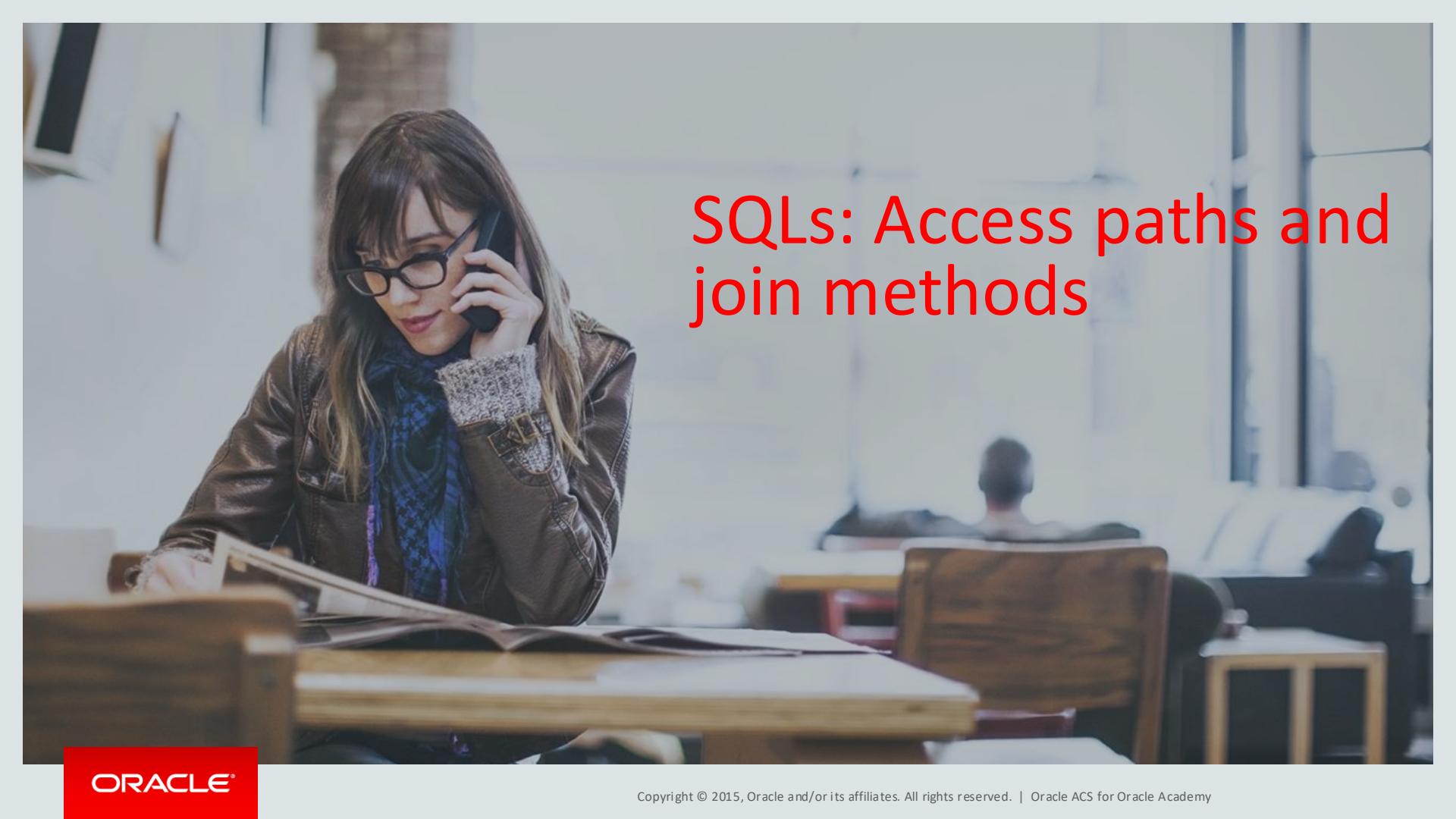
```
4 - access("E"."DEPARTMENT_ID"="D"."DEPARTMENT_ID")
      filter("E"."DEPARTMENT_ID"="D"."DEPARTMENT_ID")
```

18 rows selected.

TABLE ACCESS BY  
INDEX ROWID  
of DEPARTMENTS

INDEX FULL SCAN  
DEPT\_ID\_PK



A woman with long brown hair and glasses, wearing a brown leather jacket over a blue patterned top, is sitting at a desk in what appears to be a library or study area. She is holding a black telephone receiver to her ear with her left hand and looking down at some papers on the desk with a focused expression. In the background, there are other people seated at desks, and large windows are visible.

# SQLs: Access paths and join methods

# Main Structures and Access Paths

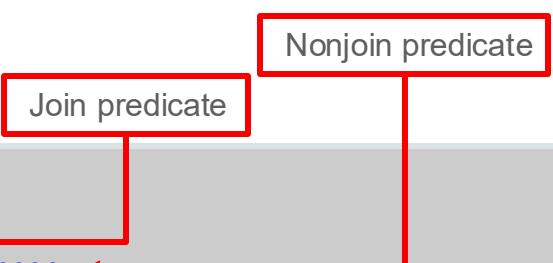
Structures	Access Paths
Tables	<ol style="list-style-type: none"><li>1. Full Table Scan</li><li>2. ROWID Scan</li><li>3. Sample Table Scan</li></ol>
Indexes	<ol style="list-style-type: none"><li>4. Index Scan (Unique)</li><li>5. Index Scan (Range)</li><li>6. Index Scan (Full)</li><li>7. Index Scan (Fast Full)</li><li>8. Index Scan (Skip)</li><li>9. Index Scan (Index Join)</li><li>10. Using Bitmap Indexes</li><li>11. Combining Bitmap Indexes</li></ol>

# Join Methods

A join:

- Defines the relationship between two row sources
- Is a method of combining data from two data sources
- Is controlled by join predicates, which define how the objects are related
- Join methods:
  - Nested loops
  - Sort-merge join
  - Hash join
  - Cartesian join

```
SELECT e.ename, d.dname
FROM dept d, emp e
WHERE e.deptno = d.deptno
AND e.job = 'ANALYST' OR e.empno = 9999;
```



A woman with long brown hair and glasses, wearing a brown leather jacket over a blue patterned top, is sitting at a desk in what appears to be a library or study room. She is holding a black telephone receiver to her ear with her left hand and looking down at some papers on the desk with a focused expression. In the background, there are other people seated at desks, and large windows are visible.

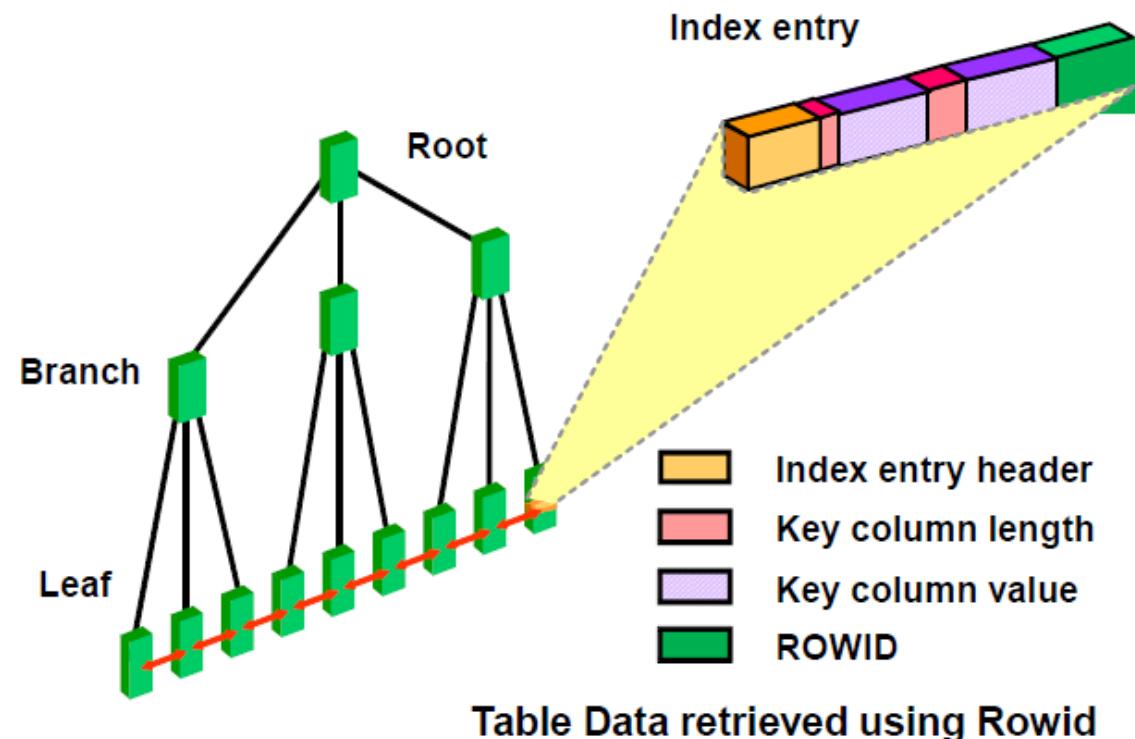
# SQL: using indexes

# Indexes types

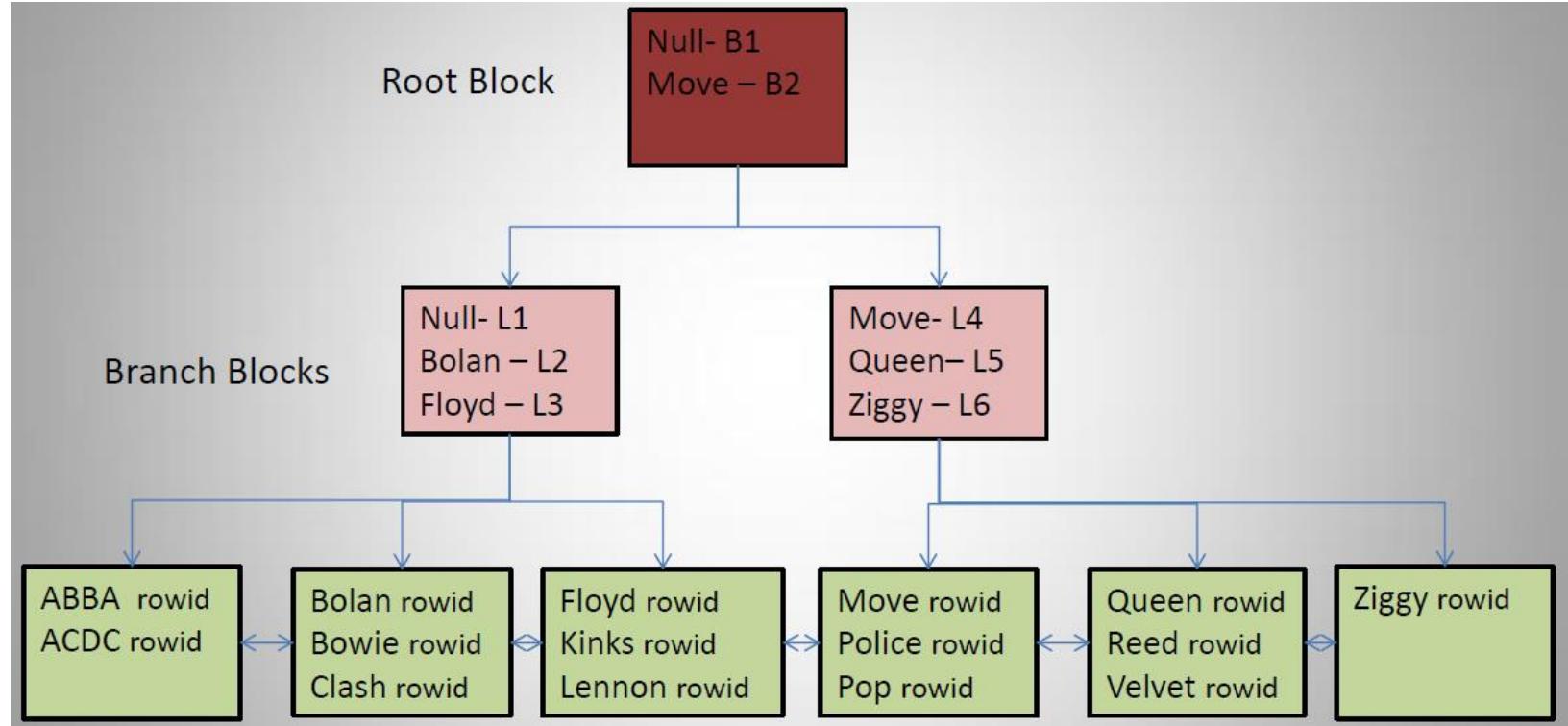
- B\*Tree Index
- Bitmap Index
- Bitmap Join Index
- Function-Based Index
- Index-Organized Table IOT
- Application Domain Index

# Oracle Index Types

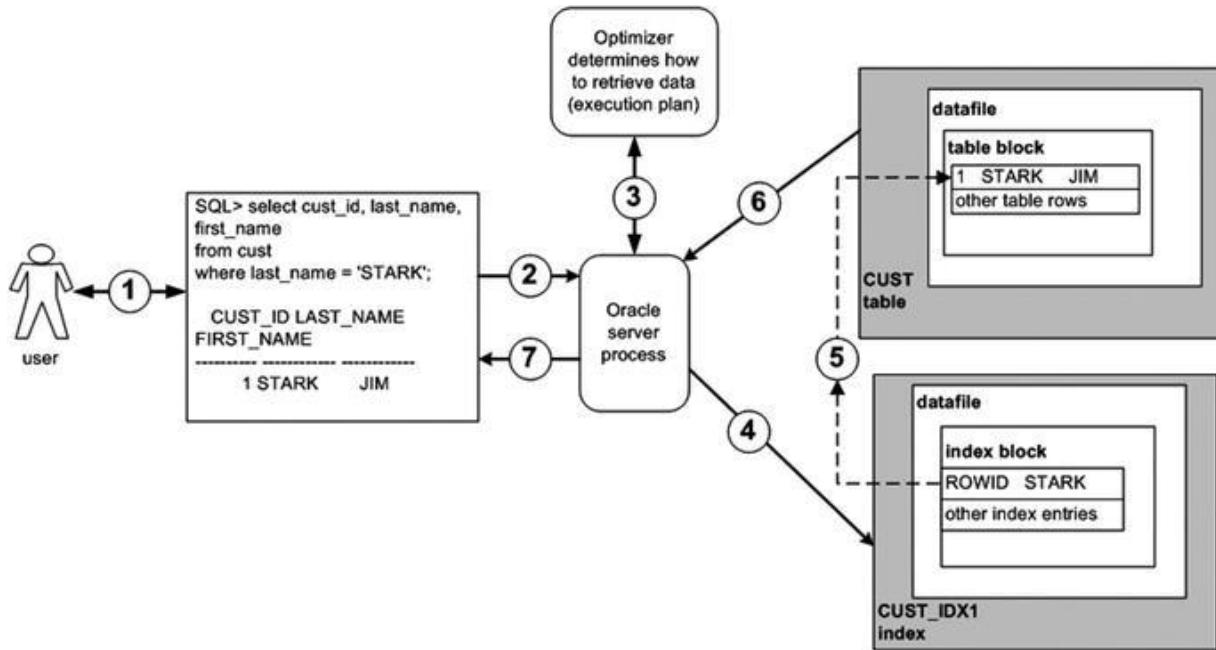
## B\*-Tree Indexes



# B\* Tree Index (B means BALANCED!)



# How SQL use an index



# How an SQL use an index (cont.)

To access a Table Row using an Index Range Scan we need to read:

- Index Root block and all branch
- Traverse the Index leaf blocks (ascending or descending).
- For each selected Index row read matching Table row by reading that Table block.



[valentin.robu@oracle.com](mailto:valentin.robu@oracle.com)