Alibaba Cloud | Worldwide Cloud Services Partner

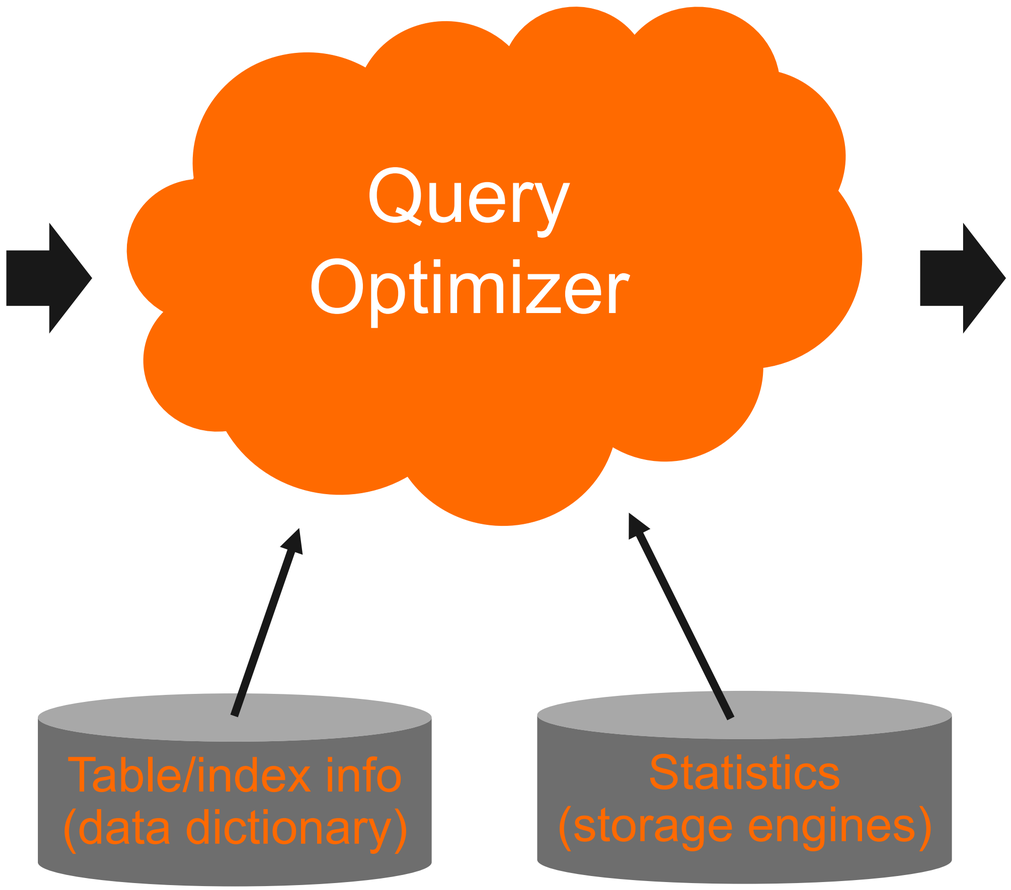
The MySQL Query Optimizer

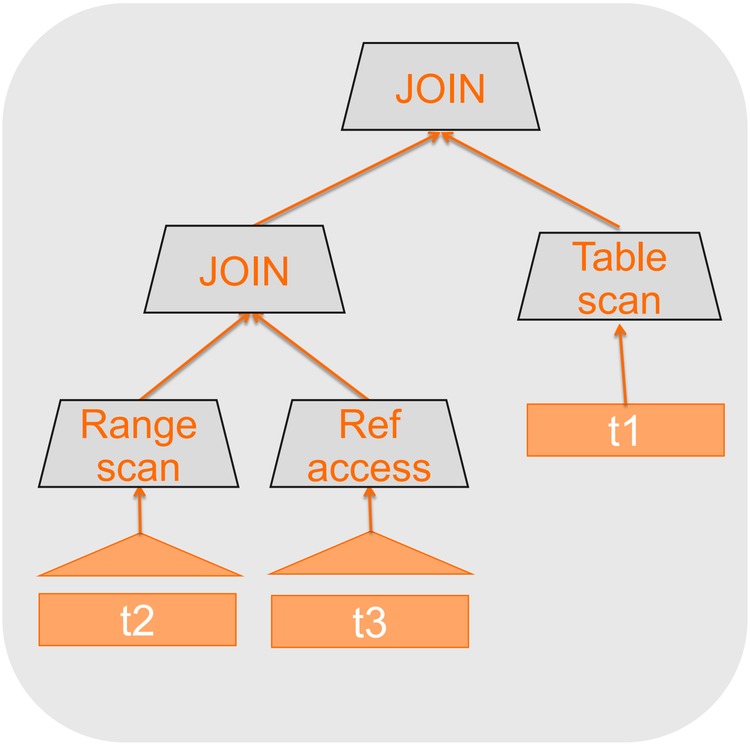
Explained Through Optimizer Trace

Øystein Grøvlen
Senior Staff Engineer
Alibaba Cloud

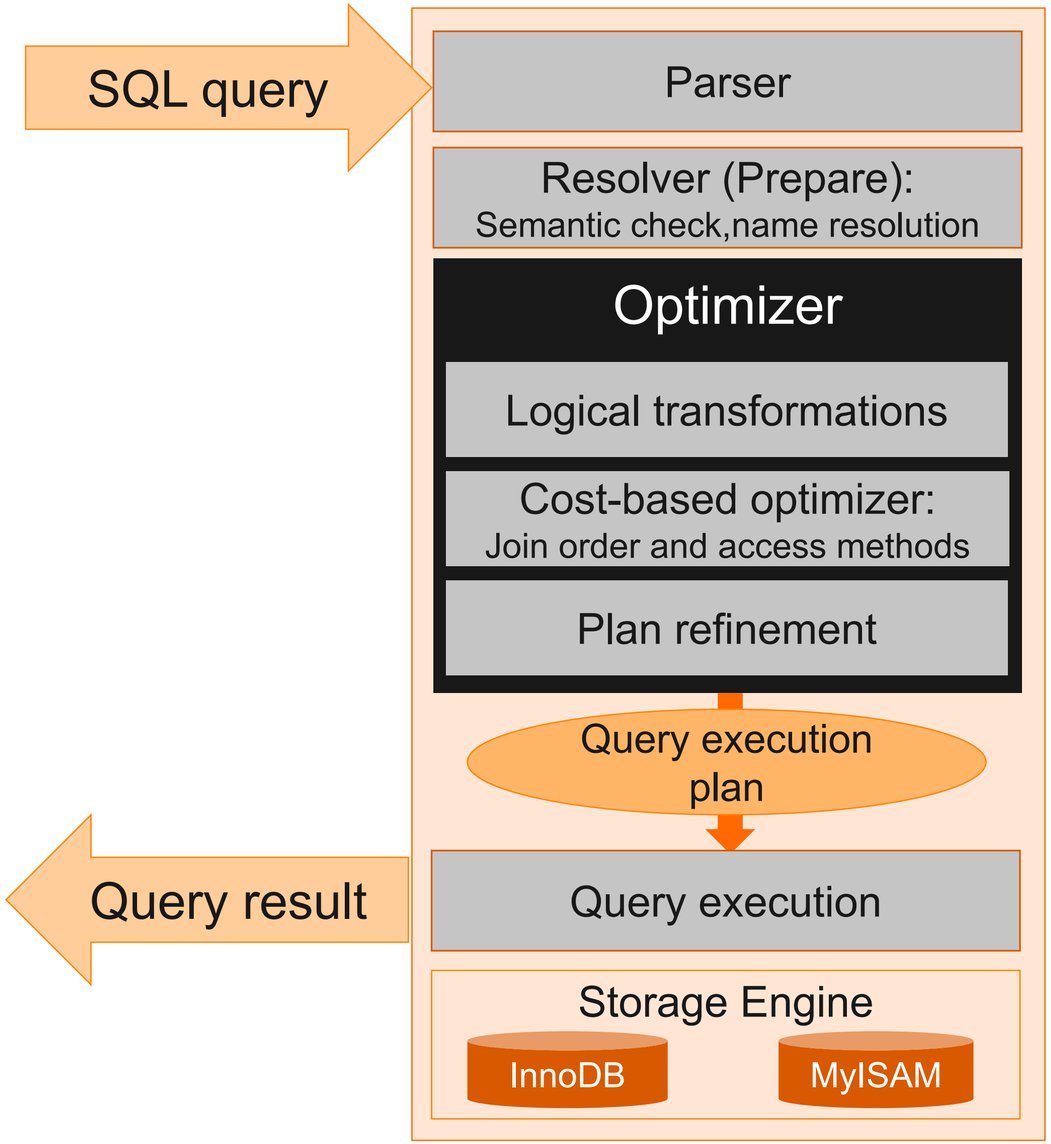
MySQL Query Optimizer


```
SELECT a, b
FROM t1, t2, t3
WHERE t1.a = t2.b
    AND t2.b = t3.c
    AND t2.d > 20
    AND t2.d < 30;
```





MySQL Architecture



Optimizer Trace

Alibaba Cloud | Worldwide Cloud Services Partner

How to generate it

- EXPLAIN shows the selected plan
- Optimizer trace shows WHY the plan was selected

```
SET optimizer_trace= "enabled=on";
SELECT * FROM t1, t2 WHERE f1=1 AND f1=f2 AND f2>0;
SELECT trace FROM information_schema.optimizer_trace
INTO OUTFILE filename LINES TERMINATED BY ”;
SET optimizer_trace="enabled=off";
```

QUERY	SELECT * FROM t1, t2 WHERE f1=1 AND f1=f2 AND f2>0;
TRACE	{ "steps": [{ "join_preparation": { "select#": 1,... } ... } ...] }
MISSING_BYTES_BEYOND_MAX_MEM_SIZE	0
INSUFFICIENT_PRIVILEGES	0

Optimizer Trace

Alibaba Cloud | Worldwide Cloud Services Partner

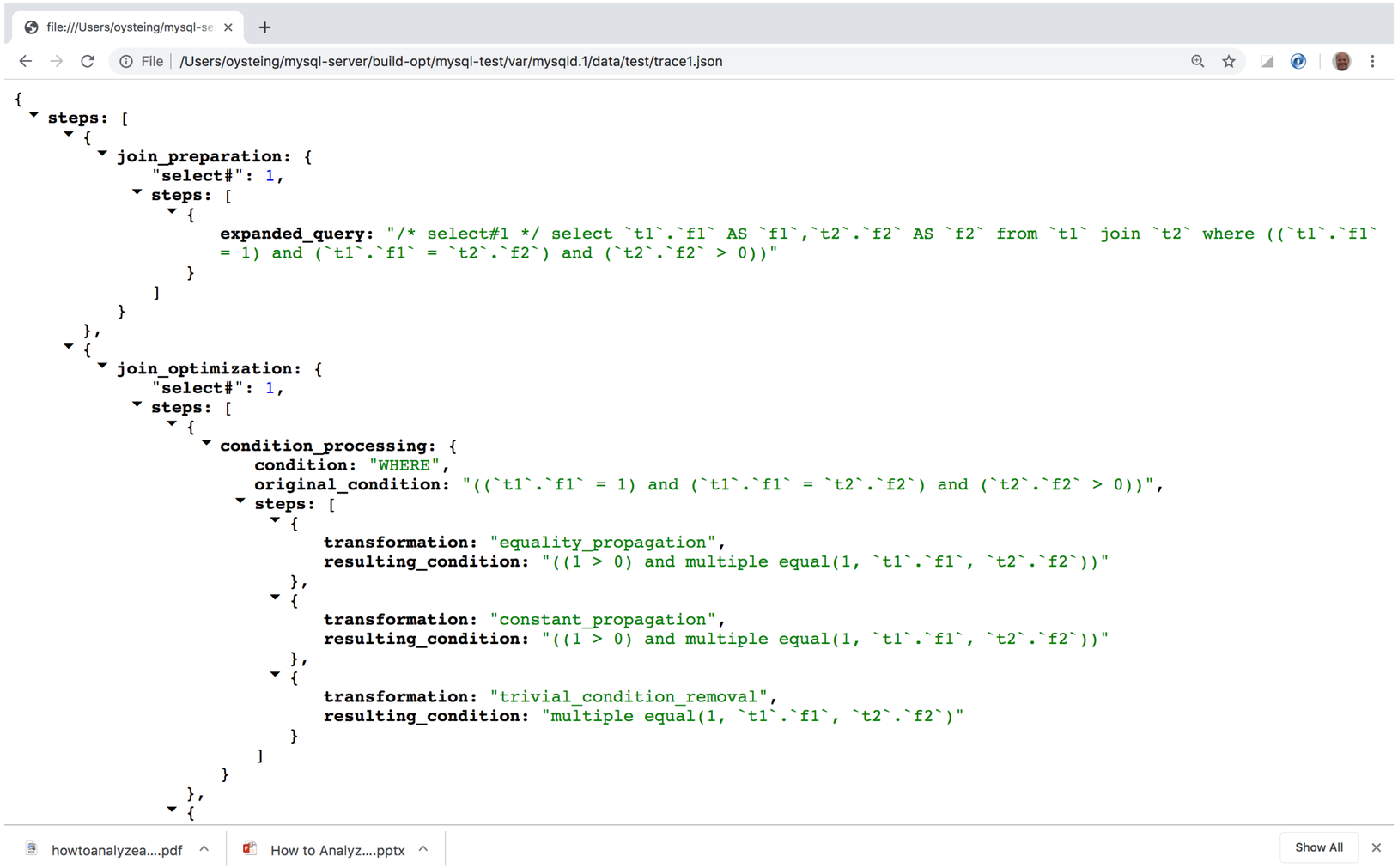
Example


```
{
  "steps": [
    {
      "join_preparation": {
        "select#": 1,
        "steps": [
          {
            "expanded_query": "/* select#1 */ select `t1`.`f1` AS `f1`,`t2`.`f2` AS `f2` from `t1` join `t2` where ((`t1`.`f1` = 1) and (`t1`.`f1` = `t2`.`f2`) and (`t2`.`f2` > 0))"
          }
        ]
      }
    },
    {
      "join_optimization": {
        "select#": 1,
        "steps": [
          {
            "condition_processing": {
              "condition": "WHERE",
              "original_condition": "((`t1`.`f1` = 1) and (`t1`.`f1` = `t2`.`f2`) and (`t2`.`f2` > 0))",
              "steps": [
                {
...
```

Optimizer Trace

Alibaba Cloud | Worldwide Cloud Services Partner

JSON browser plugin



Browser Plugin

Alibaba Cloud | Worldwide Cloud Services Partner

Collapse to see main trace objects/phases


```
{
  ▼ steps: [
      ▼ {
          ▶ join_preparation: {…}
        },
      ▼ {
          ▶ join_optimization: {…}
        },
      ▼ {
          ▶ join_execution: {…}
        }
    ]
}
```

Browser Plugin

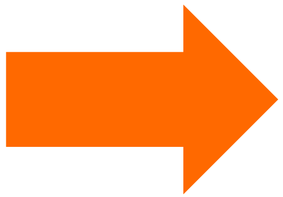
Alibaba Cloud | Worldwide Cloud Services Partner

Expand JSON objects


```
{
  steps: [
    {
      join_preparation: {
          "select#": 1,
        steps: [
          {
              expanded_query: "/* select#1 */ select `t1`.`f1` AS
              `f1`,`t2`.`f2` AS `f2` from `t1` join `t2` where ((`t1`.`f1` =
              1) and (`t1`.`f1` = `t2`.`f2`) and (`t2`.`f2` > 0))"
          }
        ]
      }
    },
    {
      join_optimization: {…}
    },
    {
      join_execution: {…}
    }
  ]
}
```

Prepare Phase

- Name resolving
  - Map names to database objects (tables, columns, …)
- Semantic checks
- Permanent transformations:
  - Conversion of outer join to inner join
  - Merging of views and derived tables
  - Subquery transformations
    - IN to EXISTS
    - IN to Semijoin (5.6)
    - EXISTS to IN (8.0.16)
    - Etc.



Simpler query to optimize and execute

+

Prepare for later optimizations

Conversion of outer join to inner join

Alibaba Cloud | Worldwide Cloud Services Partner

SELECT o_orderkey FROM orders LEFT JOIN lineitem ON o_orderkey = l_orderkey WHERE l_discount > 0.10;

```
"join_preparation": {
  "select#": 1,
  "steps": [
    {
      "expanded_query": "/* select#1 */ select `orders`.`o_orderkey` AS `o_orderkey` from (`orders` left join `lineitem` on ((`orders`.`o_orderkey` =  `lineitem`.`l_orderkey`))) where (`lineitem`.`l_discount` > 0.10)"
    },
    {
      "transformations_to_nested_joins": {
        "transformations": [
          "outer_join_to_inner_join",
          "JOIN_condition_to_WHERE",
          "parenthesis_removal"
        ],
        "expanded_query": "/* select#1 */ select `orders`.`o_orderkey` AS `o_orderkey` from `orders` join `lineitem` where ((`lineitem`.`l_discount` > 0.10) and(`orders`.`o_orderkey` = `lineitem`.`l_orderkey`))"
      } ...
```

Conversion from EXISTS-subquery to IN-subquery

New in 8.0.16

```
SELECT o_orderpriority, COUNT(*) AS order_count FROM orders
WHERE EXISTS
  (SELECT * FROM lineitem WHERE l_orderkey = o_orderkey AND l_commitdate < l_receiptdate)
GROUP BY o_orderpriority ORDER BY o_orderpriority;
```

```
SELECT o_orderpriority, COUNT(*) AS order_count FROM orders
WHERE  o_orderkey IN
   (SELECT l_orderkey FROM lineitem WHERE l_commitdate < l_receiptdate
GROUP BY o_orderpriority ORDER BY o_orderpriority;
```

```
SELECT o_orderpriority, COUNT(*) AS order_count
FROM orders SEMIJOIN lineitem ON l_orderkey = o_orderkey
WHERE l_commitdate < l_receiptdate
GROUP BY o_orderpriority ORDER BY o_orderpriority;
```

Conversion from EXISTS-subquery to IN-subquery

Alibaba Cloud | Worldwide Cloud Services Partner

Optimizer trace

```
"join_preparation": {
      "select#": 1,
      "steps": [
        {
          "join_preparation": {
            "select#": 2,
            "steps": [
              {
                "expanded_query": "/* select#2 */ select 1 from `lineitem` where ((`lineitem`.`l_orderkey` = `orders`.`o_orderkey`)
and (`lineitem`.`l_commitDATE` < `lineitem`.`l_receiptDATE`))"
              },
              {
                "transformation": {
                  "select#": 2,
                  "from": "EXISTS (SELECT)",
                  "to": "semijoin",
                  "chosen": true
                }
              }
            ]
          ...
```

Conversion from EXISTS-subquery to IN-subquery

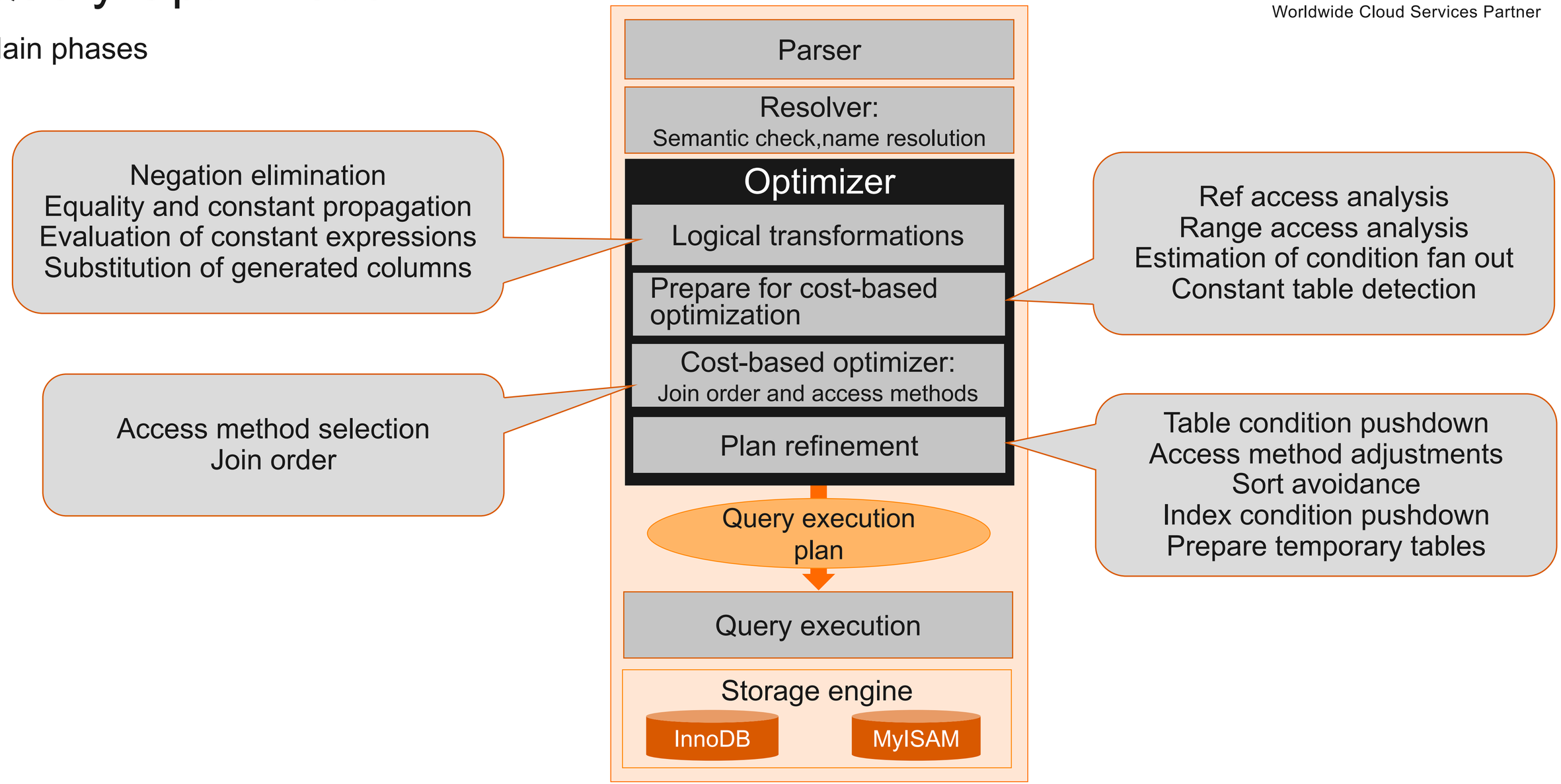
Optimizer trace, cont.


```
join_preparation: {
  "select#": 1,
  steps: [
    {
      join_preparation: {…}
    },
    {
      expanded_query: "/* select#1 */ select `orders`.`o_orderpriority` AS `o_orderpriority`,count(0) AS
      `order_count` from `orders` where exists(/* select#2 */ select 1 from `lineitem` where
      ((`lineitem`.`l_orderkey` = `orders`.`o_orderkey`) and (`lineitem`.`l_commitDATE` <
      `lineitem`.`l_receiptDATE`))) group by `orders`.`o_orderpriority` order by `orders`.`o_orderpriority`"
    },
    {
      transformation: {
        "select#": 2,
        from: "IN (SELECT)",
        to: "semijoin",
        chosen: true,
        evaluating_constant_semijoin_conditions: [ ]
      }
    },
    {
      transformations_to_nested_joins: {
        transformations: [
          "semijoin"
        ],
        expanded_query: "/* select#1 */ select `orders`.`o_orderpriority` AS `o_orderpriority`,count(0) AS
        `order_count` from `orders` semi join (`lineitem`) where ((`lineitem`.`l_commitDATE` <
        `lineitem`.`l_receiptDATE`) and (`orders`.`o_orderkey` = `lineitem`.`l_orderkey`)) group by
        `orders`.`o_orderpriority` order by `orders`.`o_orderpriority`"
      }
```

Query Optimization

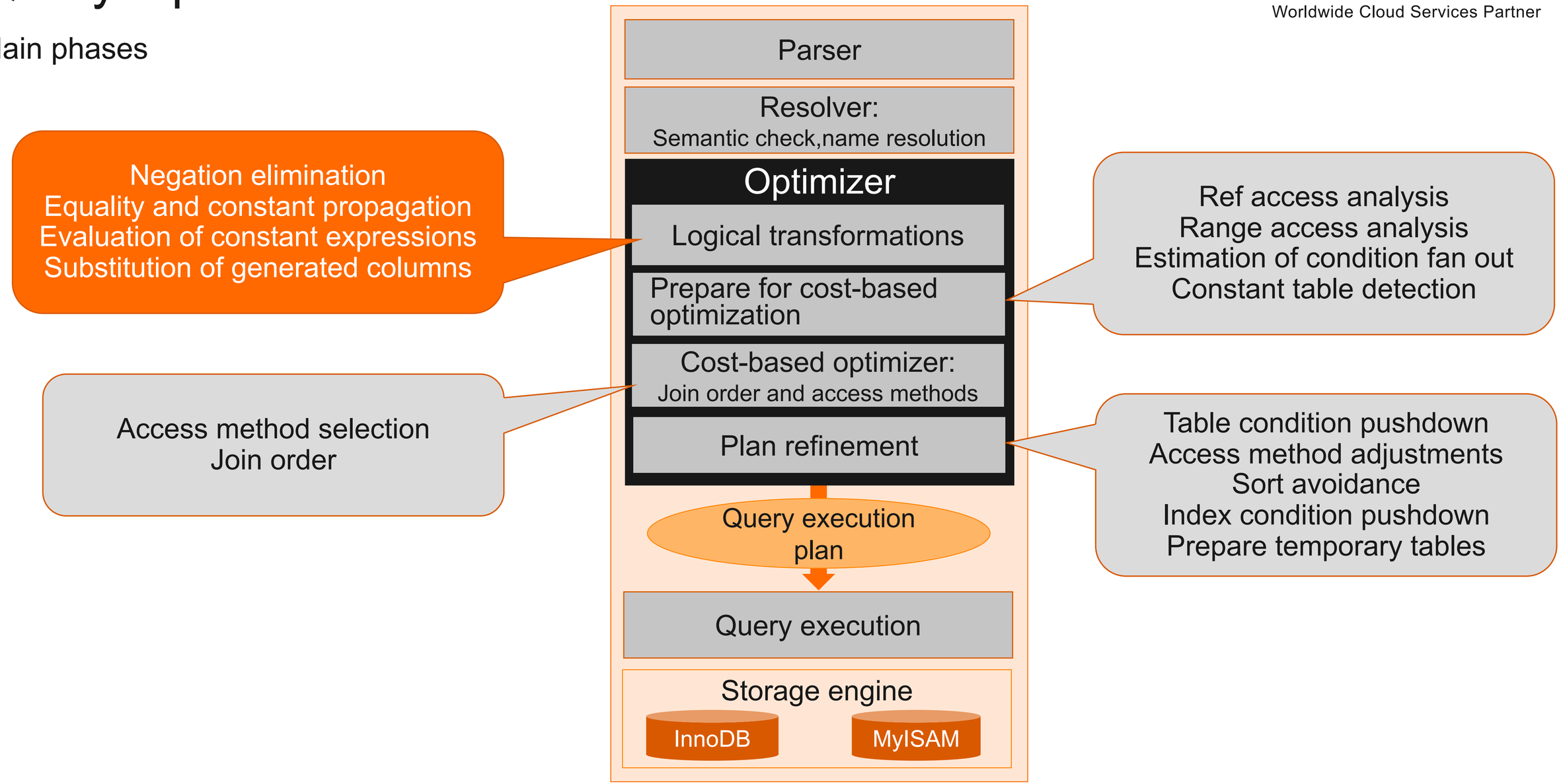
Alibaba Cloud | Worldwide Cloud Services Partner

Main phases



Query Optimization

Main phases



Logical Transformations

Alibaba Cloud | Worldwide Cloud Services Partner

Condition processing

SELECT * FROM t1, t2 WHERE
t1.a = t2.a AND t2.a = 9 AND (NOT (t1.a > 10 OR t2.b > 3) OR (t1.b = t2.b + 7 AND t2.b = 5));

Negation elimination

t1.a = t2.a AND t2.a = 9 AND (t1.a <= 10 AND t2.b <= 3 OR (t1.b = t2.b + 7 AND t2.b = 5));

Equality/const propagation

t1.a = 9 AND t2.a = 9 AND (9 <= 10 AND t2.b <= 3 OR (t1.b = 5 + 7 AND t2.b = 5));

Evaluate const expressions

t1.a = 9 AND t2.a = 9 AND (9 <= 10 AND t2.b <= 3 OR (t1.b = 12 AND t2.b = 5));

=TRUE

Trivial condition removal

t1.a = 9 AND t2.a = 9 AND (t2.b <= 3 OR (t1.b = 12 AND t2.b = 5));

Logical Transformations

Alibaba Cloud | Worldwide Cloud Services Partner

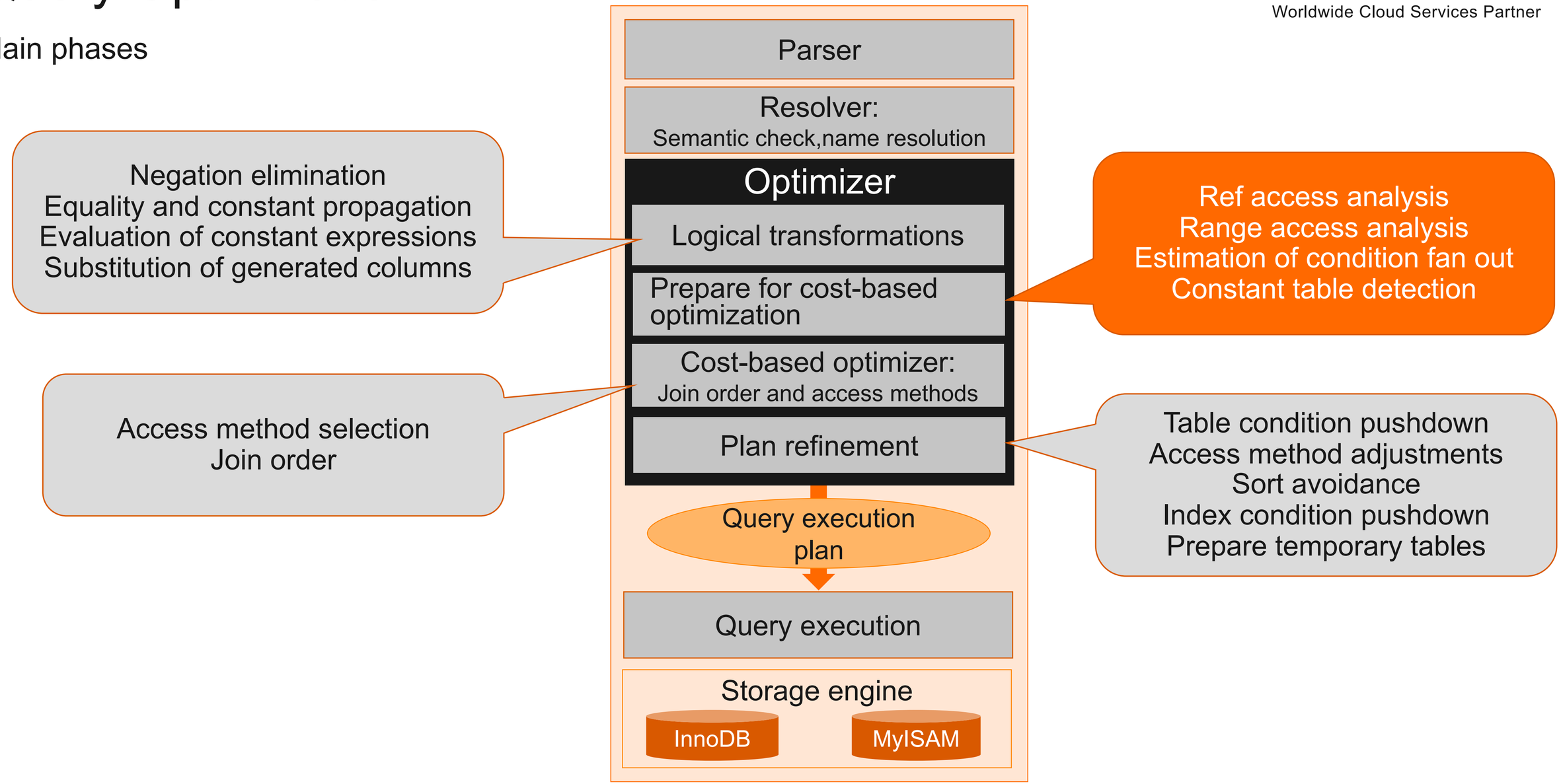
Optimizer Trace


```
"join_optimization": {
      "select#": 1,
      "steps": [
       {
         "condition_processing": {
          "condition": "WHERE",
          "original_condition": "((`t1`.`a` = `t2`.`a`) and (`t2`.`a` = 9) and (((`t1`.`a` <= 10) and (`t2`.`b` <= 3)) or ((`t1`.`b` =(`t2`.`b` + 7)) and
(`t2`.`b` = 5))))",
          "steps": [
           {
            "transformation": "equality_propagation",
            "resulting_condition": "((((9 <= 10) and (`t2`.`b` <= 3)) or ((`t1`.`b` = (5 + 7)) and multiple equal(5, `t2`.`b`))) and multiple equal(9,
`t1`.`a`, `t2`.`a`))"
           },
           {
            "transformation": "constant_propagation",
            "resulting_condition": "((((9 <= 10) and (`t2`.`b` <= 3)) or ((`t1`.`b` = 12) and multiple equal(5, `t2`.`b`))) and multiple equal(9, `t1`.`a`
, `t2`.`a`))"
           },
           {
            "transformation": "trivial_condition_removal",
            "resulting_condition": "(((`t2`.`b` <= 3) or ((`t1`.`b` = 12) and multiple equal(5, `t2`.`b`))) and multiple equal(9, `t1`.`a`, `t2`.`a`))"
           }
```

Query Optimization

Alibaba Cloud | Worldwide Cloud Services Partner

Main phases

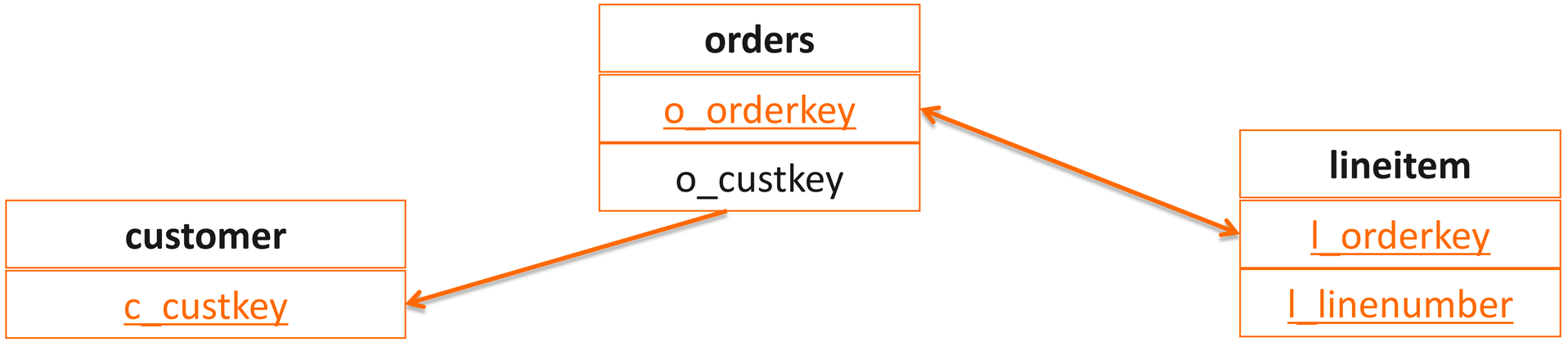


Ref Access Analysis

Alibaba Cloud | Worldwide Cloud Services Partner

Determine which indexes that can be used for index lookup in a join

```
SELECT l_orderkey, sum(l_extendedprice * (1 - l_discount)) AS revenue,
       o_orderdate, o_shippriority
FROM customer JOIN orders ON c_custkey = o_custkey
              JOIN lineitem ON l_orderkey = o_orderkey
WHERE c_mktsegment = 'FURNITURE'
  AND o_orderdate < '1997-04-15' AND l_shipdate > '1997-04-15'
GROUP by l_orderkey, o_orderdate, o_shippriority
ORDER by revenue desc, o_orderdate
LIMIT 10;
```



Ref Access Analysis

Alibaba Cloud | Worldwide Cloud Services Partner

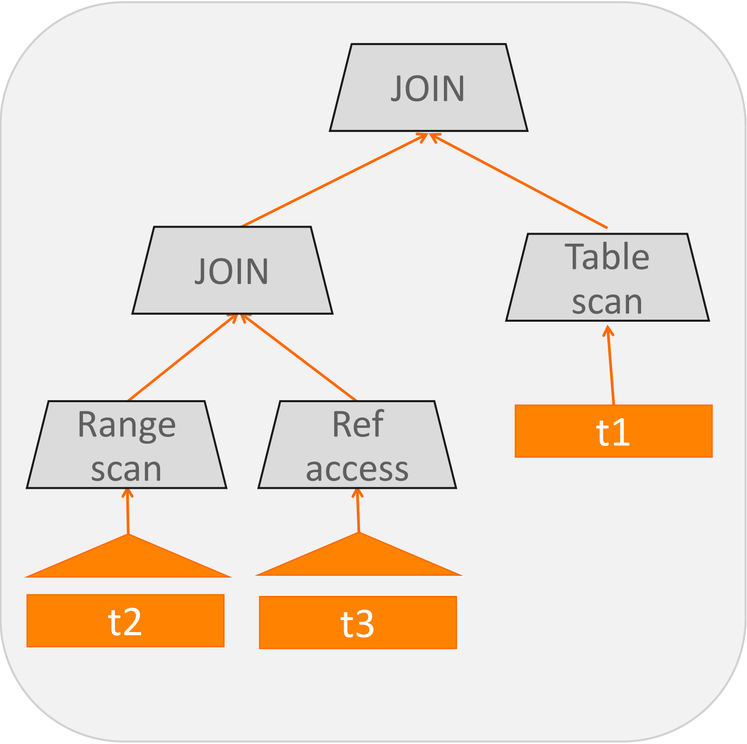
Optimizer trace


```
{
        "ref_optimizer_key_uses": [
         {
           "table": "`customer`",
           "field": "c_custkey",
           "equals": "`orders`.`o_custkey`",
           "null_rejecting": true
         },
         {
           "table": "`orders`",
           "field": "o_orderkey",
           "equals": "`lineitem`.`l_orderkey`",
           "null_rejecting": false
         },
         {
           "table": "`lineitem`",
           "field": "l_orderkey",
           "equals": "`orders`.`o_orderkey`",
           "null_rejecting": false
         },
         {
           "table": "`lineitem`",
           "field": "l_orderkey",
           "equals": "`orders`.`o_orderkey`",
           "null_rejecting": false
         }
       ]
},
```

Cost-based Query Optimization

General idea:

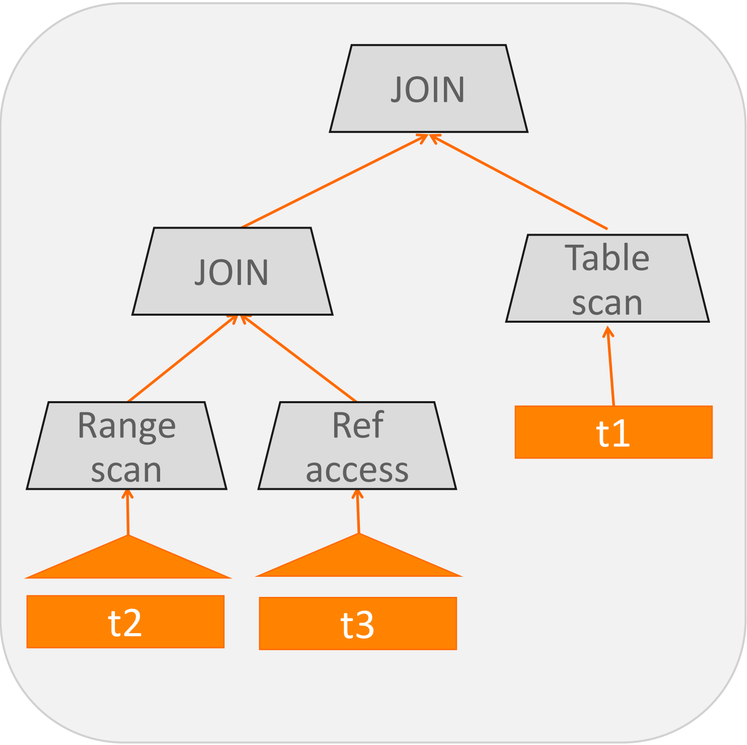
- Assign cost to operations
- Assign cost to partial or alternative plans
- Search for plan with lowest cost



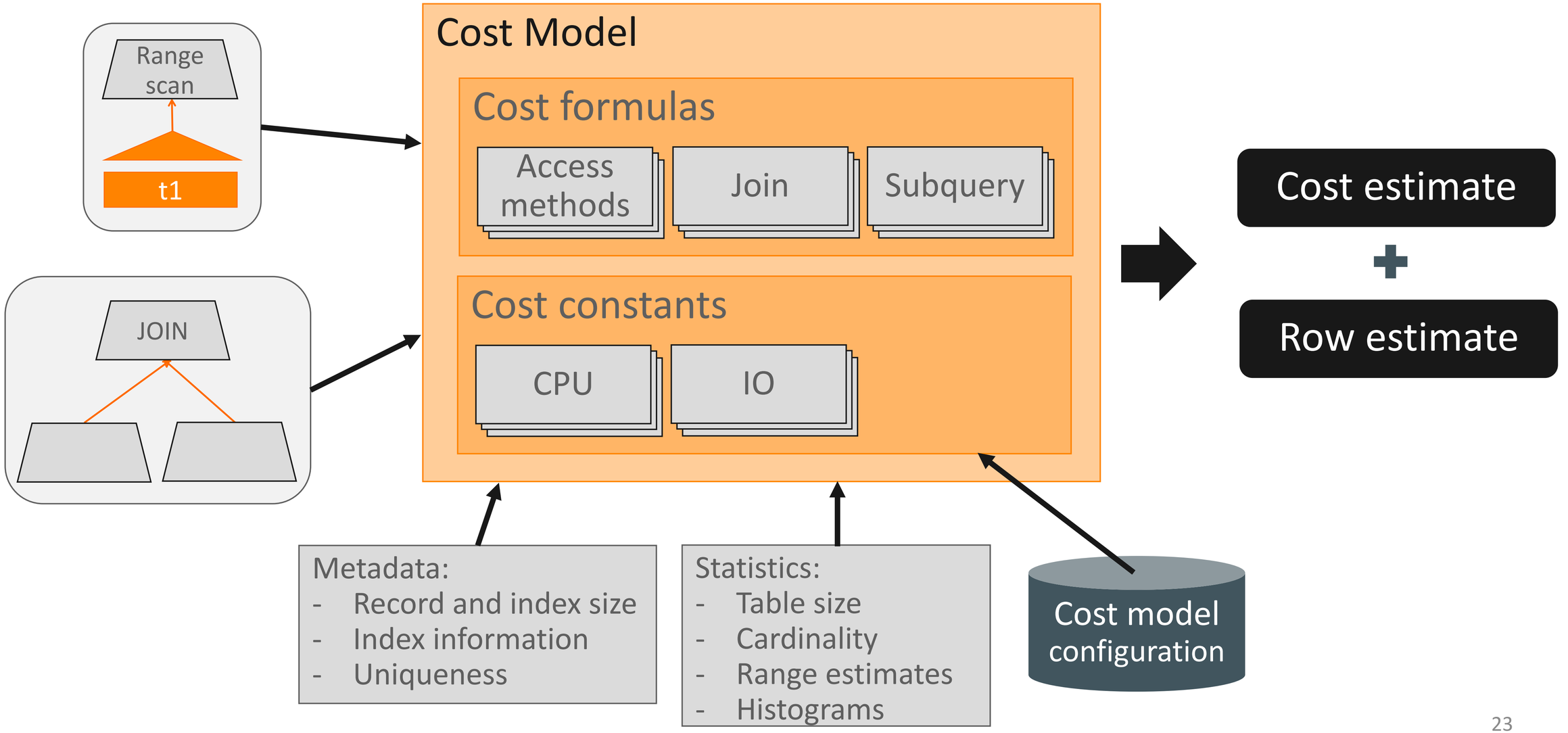
Cost-based Query Optimizations

The main cost-based optimizations:

- Index and access method:
  - Table scan
  - Index scan
  - Range scan
  - Index lookup (ref access)
- Join order
- Join buffering strategy
- Subquery strategy



Optimizer Cost Model



Cost Estimates

- The **cost** for **executing** a query
- **Cost unit:**
  - “read a random data page from disk”
- Main cost factors:
  - IO cost:
    - #pages read from table
    - #pages read from index
  - CPU cost:
    - Evaluating query conditions
    - Comparing keys/records
    - Sorting keys

Main cost constants

Cost	MySQL 5.7	MySQL 8.0
Read a random disk page	1.0	1.0
Read a data page from memory buffer	1.0	0.25
Evaluate query condition	0.2	0.1
Compare keys/records	0.1	0.05

Range analysis

Alibaba Cloud | Worldwide Cloud Services Partner

Compare cost of table scan to index scan

- Cost of table scan
  - Based on number of pages to read
- Cost of index scan alternatives
  - Index scan (covering)
  - Index range scan
  - Index merge
  - Index for grouping
  - Skip scan (new in 8.0)
  - Loose-index scan

Range Analysis Example

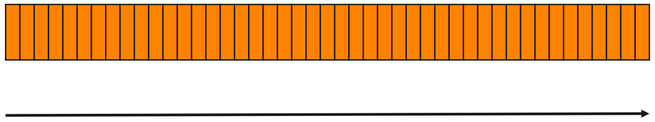
Alibaba Cloud | Worldwide Cloud Services Partner

Table scan vs Index range scan

SELECT SUM(o_totalprice) FROM orders
WHERE o_orderdate BETWEEN '1994-01-01' AND '1994-12-31';

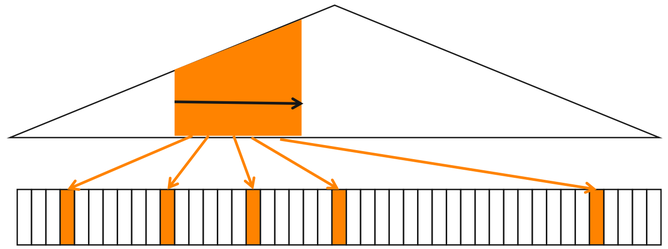
Table scan:

- IO-cost: #pages in table * IO_BLOCK_READ_COST
- CPU cost: #rows * ROW_EVALUATE_COST



Range scan (on secondary index):

- IO-cost: #rows_in_range * IO_BLOCK_READ_COST
- CPU cost: #rows_in_range * ROW_EVALUATE_COST



Range Analysis Example

Alibaba Cloud | Worldwide Cloud Services Partner

Table scan vs index range scan cnt.

EXPLAIN SELECT SUM(o_totalprice) FROM orders
WHERE o_orderdate BETWEEN '1994-01-01' AND '1994-12-31';

id	select type	table	type	possible keys	key	key len	rows	filtered	Extra
1	SIMPLE	orders	ALL	i_o_orderdate	NULL	NULL	1480845	31.13	Using where

EXPLAIN SELECT SUM(o_totalprice) FROM orders
WHERE o_orderdate BETWEEN '1994-01-01' AND '1994-06-30';

Id	select type	table	type	possible keys	key	key len	rows	filtered	Extra
1	SIMPLE	orders	range	i_o_orderdate	i_o_orderdate	4	222102	100.00	Using index condition

Range Analysis Example

Alibaba Cloud | Worldwide Cloud Services Partner

Optimizer trace

```
"rows_estimation": [
 {
  "table": "`orders`",
  "range_analysis": {
   "table_scan": {
    "rows": 1480845,
    "cost": 151138
   },
   "potential_range_indexes": [
    {
     "index": "PRIMARY",
     "usable": false,
     "cause": "not_applicable"
    },
    {
     "index": "i_o_orderdate",
     "usable": true,
     "key_parts": [
      "o_orderDATE",
      "o_orderkey"
     ]      }     ],
```

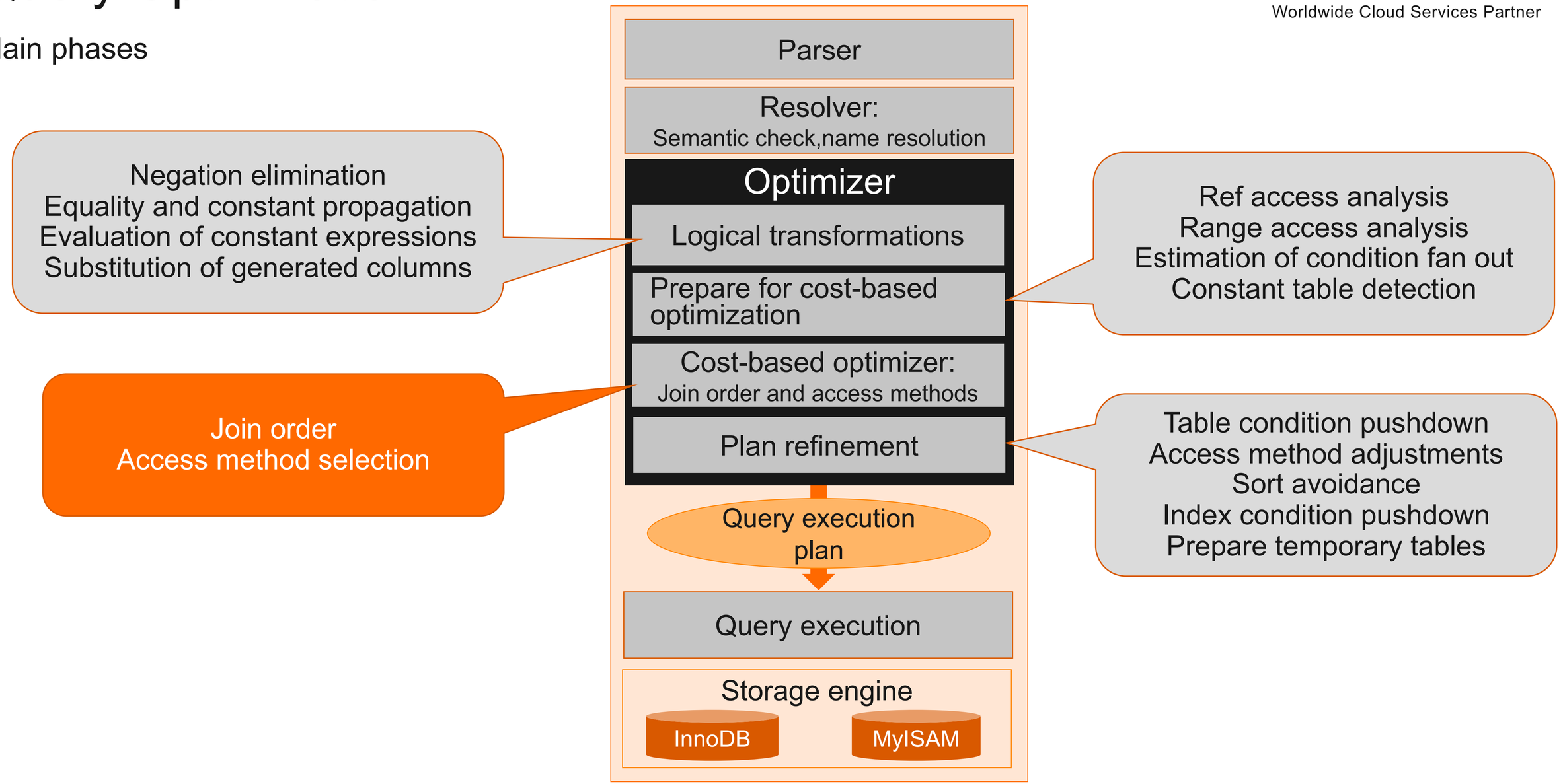
```
"setup_range_conditions": [
],
"group_index_range": {
 "chosen": false,
 "cause": "not_group_by_or_distinct"
},
"skip_scan_range": {
 "potential_skip_scan_indexes": [
  {
   "index": "i_o_orderdate",
   "usable": false,
   "cause": "query_references_nonkey_column"
  }
 ]
},
```

```
"analyzing_range_alternatives": {
    "range_scan_alternatives": [
     {
      "index": "i_o_orderdate",
      "ranges": [
       "0x21940f <= o_orderDATE <= 0x9f950f"
      ],
      "index_dives_for_eq_ranges": true,
      "rowid_ordered": false,
      "using_mrr": false,
      "index_only": false,
      "rows": 460938,
      "cost": 161329,
      "chosen": false,
      "cause": "cost"
     }
    ],
    "analyzing_roworder_intersect": {
     "usable": false,
     "cause": "too_few_roworder_scans"
    }    }   } }]
```

Query Optimization

Alibaba Cloud | Worldwide Cloud Services Partner

Main phases



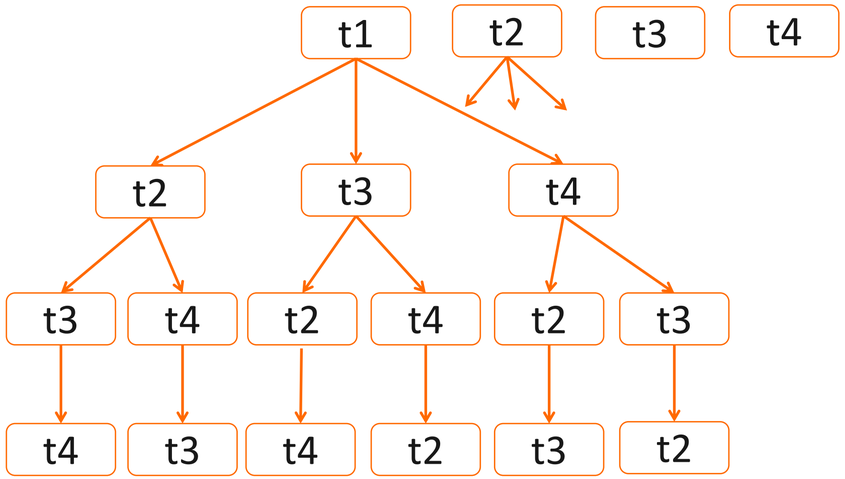
Join Optimizer

”Greedy search strategy”

- Goal: Given a JOIN of N tables, find the best JOIN ordering
- Only considers left-deap plans

N! possible plans

- Strategy:
  - Start with all 1-table plans
    - Sorted based on *size* and *key dependency*
  - Expand each plan with remaining tables
    - Depth-first
  - If “cost of partial plan” > “cost of best plan”:
    - “Prune” plan
  - Heuristic pruning:
    - Prune less promising partial plans
    - May in rare cases miss most optimal plan (turn off with **set optimizer_prune_level = 0**)

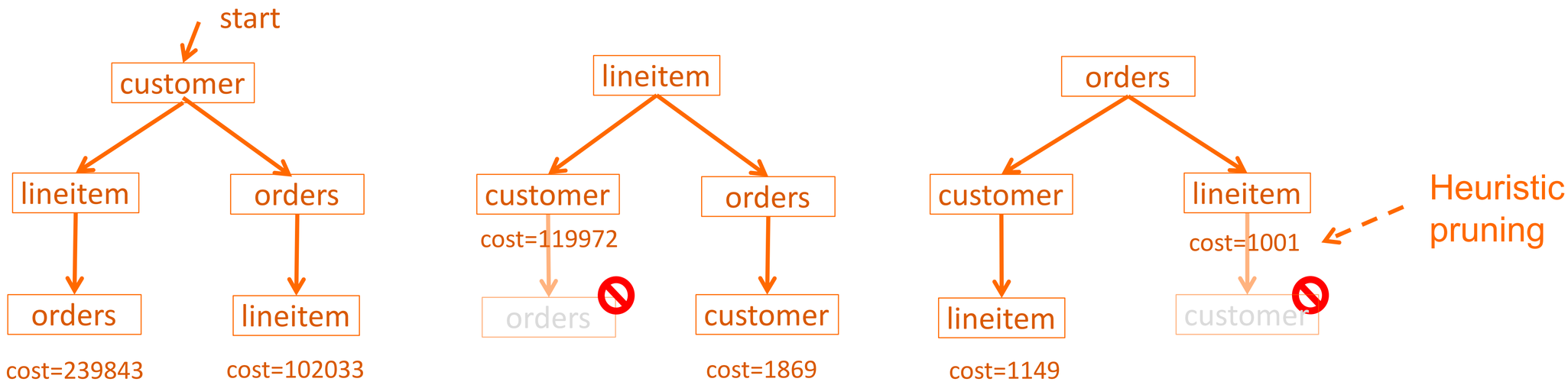


Join Optimizer Illustrated

Alibaba Cloud | Worldwide Cloud Services Partner

DBT3-Q3

```
SELECT l_orderkey, sum(l_extendedprice * (1 - l_discount)) AS revenue,
       o_orderdate, o_shippriority
FROM customer JOIN orders ON c_custkey = o_custkey
              JOIN lineitem ON l_orderkey = o_orderkey
WHERE c_mktsegment = 'FURNITURE'
  AND o_orderdate < '1997-04-15' AND l_shipdate > '1997-04-15
...
```



Join Optimizer

Alibaba Cloud | Worldwide Cloud Services Partner

Optimizer trace


```
considered_execution_plans: [
  {
    plan_prefix: [ ],
    table: "`customer`",
    best_access_path: {…},
    condition_filtering_pct: 100,
    rows_for_plan: 478.9,
    cost_for_plan: 244.2,
    rest_of_plan: [
      {
        plan_prefix: [
            "`customer`"
        ],
        table: "`lineitem`",
        best_access_path: {…},
        condition_filtering_pct: 100,
        rows_for_plan: 1.19e6,
        cost_for_plan: 119735,
        rest_of_plan: [
          {
            plan_prefix: [
                "`customer`",
                "`lineitem`"
            ],
            table: "`orders`",
            best_access_path: {…},
            condition_filtering_pct: 5,
            rows_for_plan: 59742,
            cost_for_plan: 239843,
            chosen: true
          }
        ]
      },
      {
        plan_prefix: [
            "`customer`"
        ],
        table: "`orders`",
        best_access_path: {…},
        condition_filtering_pct: 0.0471,
        rows_for_plan: 478.9,
        cost_for_plan: 101865,
        rest_of_plan: [
          {
            plan_prefix: [
                "`customer`",
                "`orders`"
            ],
            table: "`lineitem`",
            best_access_path: {…},
            condition_filtering_pct: 100,
            rows_for_plan: 486.3,
            cost_for_plan: 102033,
            chosen: true
          }
        ]
      }
    ]
  },
  {
    plan_prefix: [ ],
    table: "`lineitem`",
    best_access_path: {…},
    condition_filtering_pct: 100,
```

Choosing Access Path

Alibaba Cloud | Worldwide Cloud Services Partner

Ref access vs. table/index scan

- Ref Access (index look-up)
  - Read all records with a given key value using an index
  - Examples:

    SELECT * FROM t1 WHERE t1.key = 7;

    SELECT * FROM t1, t2 WHERE t1.key = t2.key;
  - “eq_ref”:
    - Reading from a unique index, max one record returned
  - “ref”:
    - Reading from a non-unique index or a prefix of an unique index, possibly multiple records returned
    - The record estimate is based on cardinality number from index statistics
  - Indexes available for ref access depends on the join prefix
- Table/index scan
  - The “winner” of the earlier ”range_analysis”
- Also considers:
  - Join buffering (BNL/BKA)
  - Filtering effects of conditions

Choosing the Access Path

customer


```
"plan_prefix": [
],
"table": "`customer`",
"best_access_path": {
  "considered_access_paths": [
    {
      "access_type": "ref",
      "index": "PRIMARY",
      "usable": false,
      "chosen": false
    },
    {
      "rows_to_scan": 2367,
      "filtering_effect": [
        {
          "condition": "(`customer`.`c_mktsegment` = 'FURNITURE')",
          "histogram_selectivity": 0.2023
        }
      ],
      "final_filtering_effect": 0.2023,
      "access_type": "scan",
      "resulting_rows": 478.9,
      "cost": 244.2,
      "chosen": true
    }
  ]
},
"condition_filtering_pct": 100,
"rows_for_plan": 478.9,
"cost_for_plan": 244.2,
```

Choosing the Access Path

Alibaba Cloud | Worldwide Cloud Services Partner

customer → lineitem


```
"rest_of_plan": [
 {
  "plan_prefix": [
   "`customer`"
  ],
  "table": "`lineitem`",
  "best_access_path": {
   "considered_access_paths": [
    {
     "access_type": "ref",
     "index": "PRIMARY",
     "usable": false,
     "chosen": false
    },
    {
     "access_type": "ref",
     "index": "i_l_orderkey",
     "usable": false,
     "chosen": false
    },
    {
     "rows_to_scan": 2495,
     "filtering_effect": [
     ],
     "final_filtering_effect": 1,
     "access_type": "scan",
     "using_join_cache": true,
     "buffers_needed": 1,
     "resulting_rows": 2495,
     "cost": 119491,
     "chosen": true
    }
   ]
  },
  "condition_filtering_pct": 100,
  "rows_for_plan": 1.19e6,
  "cost_for_plan": 119735,
```

Choosing the Access Path

customer → lineitem → orders


```
"rest_of_plan": [
 {
  "plan_prefix": [
   "`customer`",
   "`lineitem`"
  ],
  "table": "`orders`",
  "best_access_path": {
   "considered_access_paths": [
    {
     "access_type": "eq_ref",
     "index": "PRIMARY",
     "rows": 1,
     "cost": 120108,
     "chosen": true,
     "cause": "clustered_pk_chosen_by_heuristics"
    },
    {
     "rows_to_scan": 2502,
     "filtering_effect": [
     ],
     "final_filtering_effect": 0.8477,
     "access_type": "scan",
     "using_join_cache": true,
     "buffers_needed": 292,
     "resulting_rows": 2121,
     "cost": 2.53e8,
     "chosen": false
    }
   ]
  },
  "condition_filtering_pct": 5,
  "rows_for_plan": 59742,
  "cost_for_plan": 239843,
  "chosen": true
 }
]
```

Condensed Trace For the Join Optimizer

- With many tables, the trace for join optimization may be thousands of lines
  - The trace for DBT3 Q8 (8 tables) is 16000 lines
  - Beware: Size for optimizer trace is limited by session variable `optimizer_trace_max_mem_size`
    - Default MySQL 5.7: 16 kB
    - Default MySQL 8.0: 1 MB
  - If `information_schema.optimizer_trace.missing_bytes_beyond_max_mem_size > 0`, increase `optimizer_trace_max_mem_size`
- joinopttrace.js
  - A script to present the trace in a condensed form
  - Available at https://github.com/ogrovlen/opttrace
  - Usage

    `node joinopttrace.js` *tracefile*
  - Please:
    - Test it out
    - Suggest/submit improvements
    - Report issues

joinopttrace

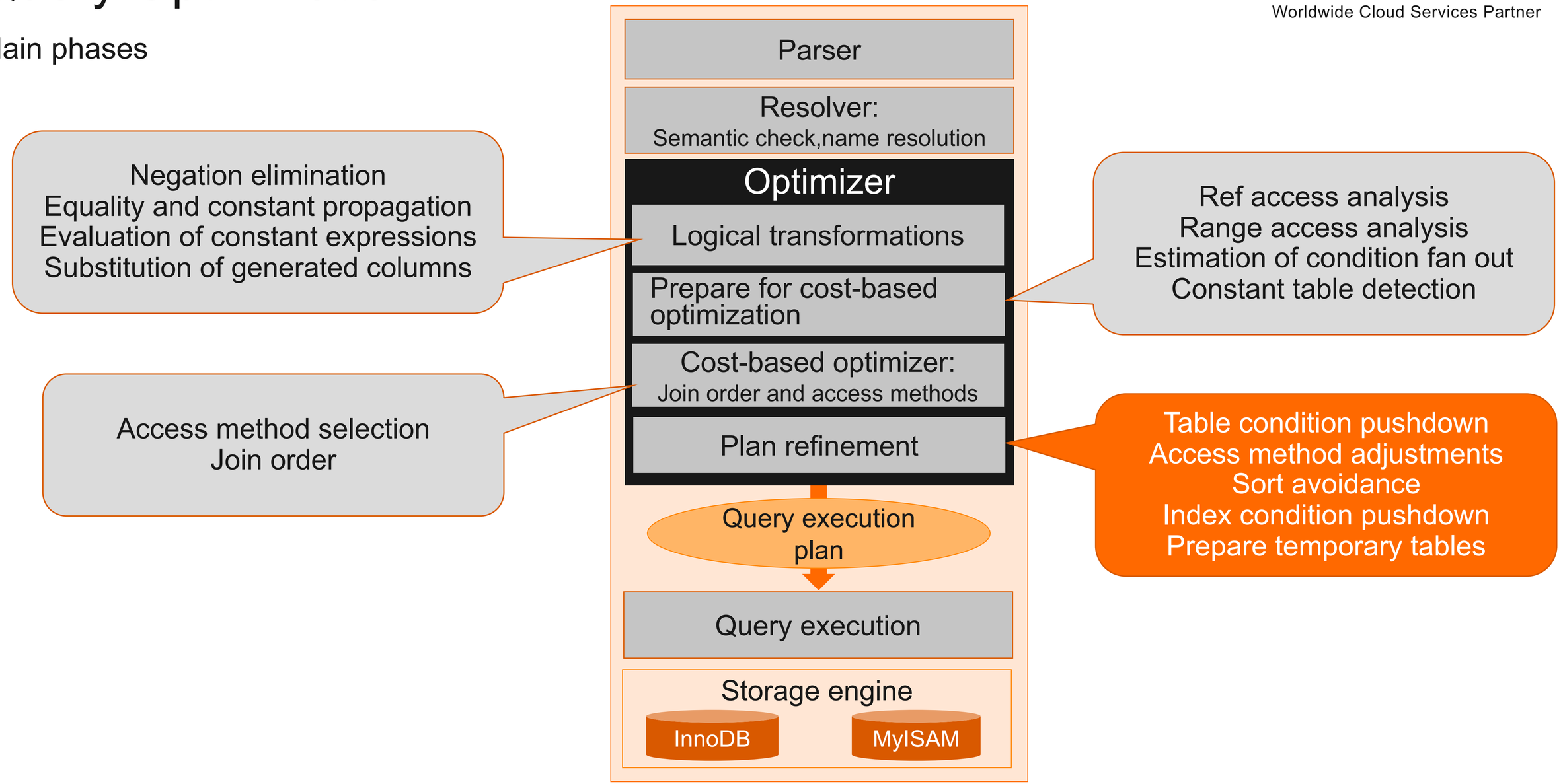
Example output (DBT3-Q3)

```
Table AccessType:IndexName Rows/Cost TotalRows/TotalCost
------------------------------------------------------
`customer` scan 2367/244.2 478.9/244.2
  `lineitem` scan 2495/119491 1190000/119735
    `orders` eq_ref:PRIMARY 1/120108 59742/239843 *** NEW BEST PLAN ***
  `orders` scan 2502/101621 478.9/101865
    `lineitem` ref:PRIMARY 1.0155/168.35 486.3/102033 *** NEW BEST PLAN ***
`lineitem` scan 2495/255.5 2495/255.5
  `customer` scan 2367/119716 1190000/119972 PRUNED(cost)
  `orders` eq_ref:PRIMARY 1/873.25 2115.1/1128.8
    `customer` eq_ref:PRIMARY 1/740.27 427.92/1869 *** NEW BEST PLAN ***
`orders` scan 2502/255.7 2121/255.7
  `customer` eq_ref:PRIMARY 1/742.35 429.12/998.05
    `lineitem` ref:PRIMARY 1.0155/150.86 435.76/1148.9 *** NEW BEST PLAN ***
  `lineitem` ref:PRIMARY 1.0155/745.63 2153.8/1001.3 PRUNED(heuristic)
```

Query Optimization

Alibaba Cloud | Worldwide Cloud Services Partner

Main phases



Assigning Query Conditions to Tables

Alibaba Cloud | Worldwide Cloud Services Partner

Evaluate conditions as early as possible in join order


```
 "attaching_conditions_to_tables": {
    "original_condition": "((`customer`.`c_custkey` = `orders`.`o_custkey`) and (`lineitem`.`l_orderkey` = `orders`.`o_orderkey`) and
 (`customer`.`c_mktsegment` = 'FURNITURE') and (`orders`.`o_orderDATE` < DATE'1997-04-15') and (`lineitem`.`l_shipDATE` > DATE'1997-
04-15'))",
    "attached_conditions_computation": [
     {
      "table": "`orders`",
      "rechecking_index_usage": {
       "recheck_reason": "low_limit",
       "limit": 10,
       "row_estimate": 2121 } } ],
    "attached_conditions_summary": [
     {
      "table": "`orders`",
      "attached": "((`orders`.`o_orderDATE` < DATE'1997-04-15') and (`orders`.`o_custkey` is not null))" },
     {
      "table": "`customer`",
      "attached": "((`customer`.`c_custkey` = `orders`.`o_custkey`) and (`customer`.`c_mktsegment` = 'FURNITURE'))" },
     {
      "table": "`lineitem`",
      "attached": "((`lineitem`.`l_orderkey` = `orders`.`o_orderkey`) and (`lineitem`.`l_shipDATE` > DATE'1997-04-15'))" } ] }
```

Assigning Query Conditions to Tables

Alibaba Cloud | Worldwide Cloud Services Partner

Remove conditions satisfied by ref access


```
"finalizing_table_conditions": [
 {
  "table": "`orders`",
  "original_table_condition": "((`orders`.`o_orderDATE` < DATE'1997-04-15') and (`orders`.`o_custkey` is not null))",
  "final_table_condition   ": "((`orders`.`o_orderDATE` < DATE'1997-04-15') and (`orders`.`o_custkey` is not null))" },
 {
  "table": "`customer`",
  "original_table_condition": "((`customer`.`c_custkey` = `orders`.`o_custkey`) and (`customer`.`c_mktsegment` = 'FURNITURE'))",
  "final_table_condition   ": "(`customer`.`c_mktsegment` = 'FURNITURE')" },
 {
  "table": "`lineitem`",
  "original_table_condition": "((`lineitem`.`l_orderkey` = `orders`.`o_orderkey`) and (`lineitem`.`l_shipDATE` > DATE'1997-04-15'))",
  "final_table_condition   ": "(`lineitem`.`l_shipDATE` > DATE'1997-04-15')" }
]
```

ORDER BY Optimization

Alibaba Cloud | Worldwide Cloud Services Partner

Avoid sorting, if possible

- Change to a different index that provides result in sorted order
- Read in descending order
- Example:

EXPLAIN SELECT * FROM orders WHERE o_totalprice > 400000 ORDER BY o_orderdate DESC LIMIT 10;

id	select type	table	type	possible keys	key	key len	rows	filtered	Extra
1	SIMPLE	orders	index	NULL	i_o_orderdate	4	10	0.57	Using where; Backward index scan

- Join queries:
  - Join order is already fixed.
  - Sorting can only be avoided if ordering is provided by an index from the first table in the join order

Assigning Query Conditions to Tables

Alibaba Cloud | Worldwide Cloud Services Partner

Evaluate conditions as early as possible in join order

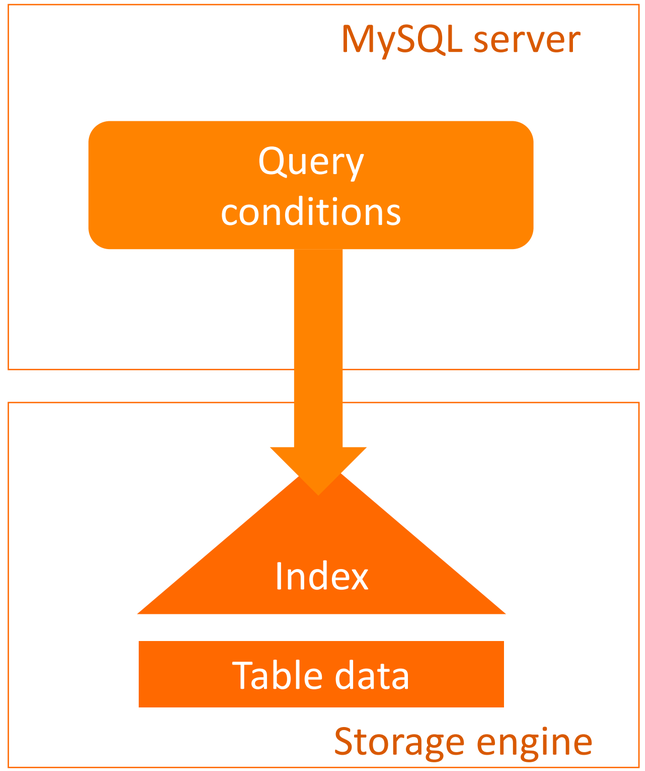

```
{
  "optimizing_distinct_group_by_order_by": {
    "simplifying_order_by": {
      "original_clause": "`orders`.`o_orderDATE` desc",
      "items": [
        {
          "item": "`orders`.`o_orderDATE`"
        }
      ],
      "resulting_clause_is_simple": true,
      "resulting_clause": "`orders`.`o_orderDATE` desc"
    }
  }
},
```

```
{
  "reconsidering_access_paths_for_index_ordering": {
    "clause": "ORDER BY",
    "steps": [
    ],
    "index_order_summary": {
      "table": "`orders`",
      "index_provides_order": true,
      "order_direction": "desc",
      "index": "i_o_orderdate",
      "plan_changed": true,
      "access_type": "index"
    }
  }
},
```


- Unfortunately, no information on cost calculations!

Index Condition Pushdown

- Pushes conditions that can be evaluated on the index down to storage engine
  - Works only on indexed columns
- Goal: evaluate conditions without having to access the actual record
  - Reduces number of disk/block accesses
  - Reduces CPU usage



Index Condition Pushdown

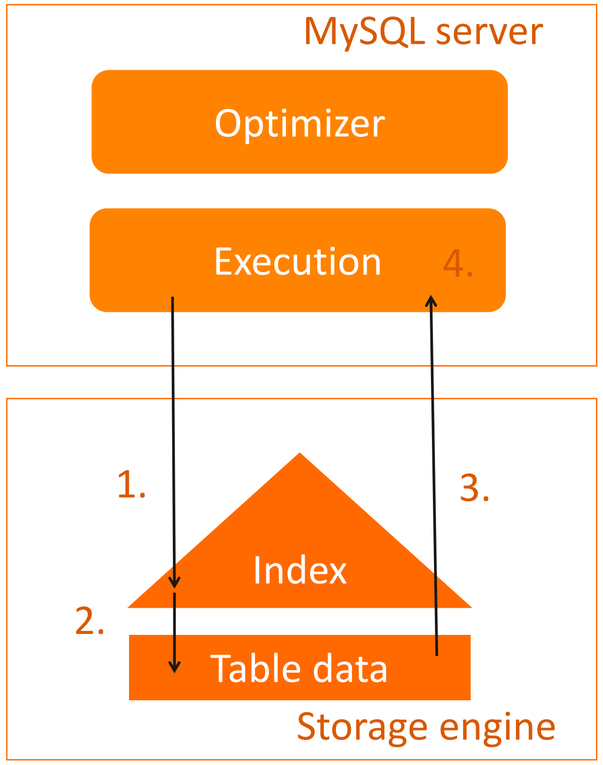
How it works

Alibaba Cloud | Worldwide Cloud Services Partner

Without ICP:

Storage Engine:
1. Reads index
2. Reads record
3. Returns record

Server:
4. Evaluates condition



With ICP:

Storage Engine:
1. Reads index and **evaluates pushed index condition**
2. Reads record
3. Returns record

Server:
4. Evaluates **rest of** condition

Index Condition Pushdown

Alibaba Cloud | Worldwide Cloud Services Partner

Optimizer trace

Index: lineitem(l_partkey,l_quantity,l_shipmode)


```
{
 "refine_plan": [
   {
     "table": "`part`"
   },
   {
     "table": "`lineitem` FORCE INDEX (`i_l_partkey_plus`)",
     "pushed_index_condition": "((`lineitem`.`l_quantity` >= 1.00) and (`lineitem`.`l_quantity` <= <cache>((1 + 10))) and
 (`lineitem`.`l_shipmode` in ('AIR','AIR REG')))",
     "table_condition_attached": "(`lineitem`.`l_shipinstruct` = 'DELIVER IN PERSON')"
   }
 ]
},
```

Prepare Temporary Tables

Alibaba Cloud | Worldwide Cloud Services Partner

DBT3-Q3

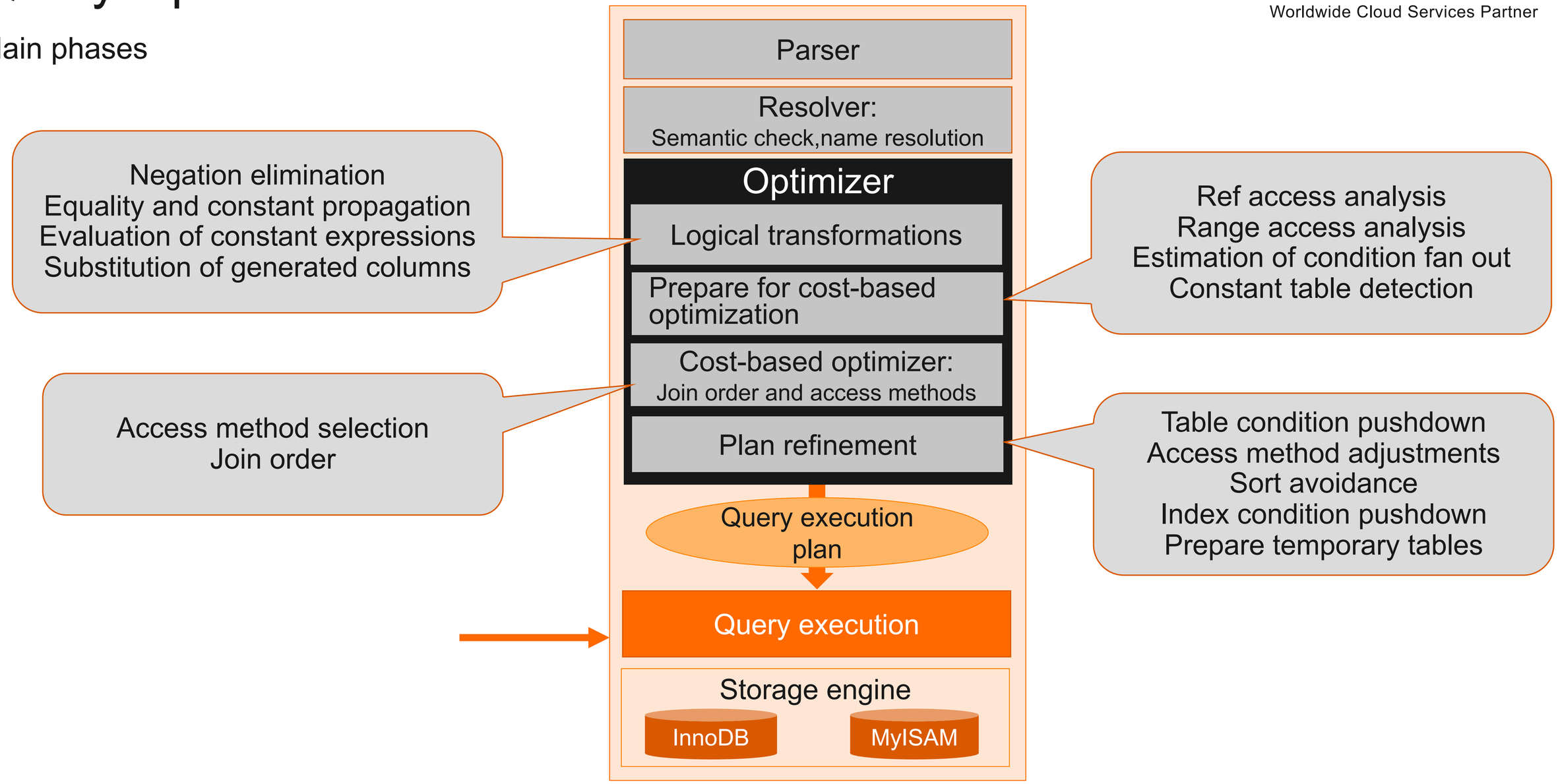
```
SELECT l_orderkey, sum(l_extendedprice * (1 - l_discount)) AS revenue,
       o_orderdate, o_shippriority
FROM customer JOIN orders ON c_custkey = o_custkey
              JOIN lineitem ON l_orderkey = o_orderkey
WHERE c_mktsegment = 'FURNITURE'
  AND o_orderdate < '1997-04-15' AND l_shipdate > '1997-04-15'
GROUP by l_orderkey, o_orderdate, o_shippriority
ORDER by revenue desc, o_orderdate
LIMIT 10;
```

```
{
  "considering_tmp_tables": [
    {
      "adding_tmp_table_in_plan_at_position": 3,
      "write_method": "continuously_update_group_row"
    },
    {
      "adding_sort_to_table_in_plan_at_position": 3
    }
  ]
}
```

Query Optimization

Alibaba Cloud | Worldwide Cloud Services Partner

Main phases



Query Execution Trace

- Information on temporary table creation
- Sort parameters and summary
- Subquery execution
- Dynamic range optimization
- May create large volumes of trace
  - Not available after EXPLAIN
  - Volume may be reduced by setting session variable:

```
SET optimizer_trace_features = “dynamic_range=off, repeated_subselect=off”;
```

Query Execution Trace

Alibaba Cloud | Worldwide Cloud Services Partner

Temporary table information


```
{
  "join_execution": {
    "select#": 1,
    "steps": [
      {
        "temp_table_aggregate": {
          "select#": 1,
          "steps": [
            {
              "creating_tmp_table": {
                "tmp_table_info": {
                  "in_plan_at_position": 3,
                  "columns": 4,
                  "row_length": 34,
                  "key_length": 13,
                  "unique_constraint": false,
                  "makes_grouped_rows": true,
                  "cannot_insert_duplicates": false,
                  "location": "TempTable"
                }            }          }        ]      }
```

Query Execution Trace

Sort information


```
{
  "sorting_table_in_plan_at_position": 3,
  "filesort_information": [
    {
      "direction": "desc",
      "field": "revenue"
    },
    {
      "direction": "asc",
      "field": "o_orderDATE"
    }
  ],
  "filesort_priority_queue_optimization": {
    "limit": 10,
    "chosen": true
  },
  "filesort_execution": [
  ],
  "filesort_summary": {
    "memory_available": 262144,
    "key_size": 33,
    "row_size": 33,
    "max_rows_per_buffer": 11,
    "num_rows_estimate": 452,
    "num_rows_found": 442,
    "num_initial_chunks_spilled_to_disk": 0,
    "peak_memory_used": 451,
    "sort_algorithm": "std::sort",
    "unpacked_addon_fields": "using_priority_queue",
    "sort_mode": "<fixed_sort_key, rowid>"
  }
}
```

A Last Tip

Alibaba Cloud | Worldwide Cloud Services Partner

Use MySQL JSON functionality to process the optimizer trace

- Example: Present the filesort summary from the last execution:

```
SELECT JSON_PRETTY(trace->'$.steps[*].join_execution.steps[*].filesort_summary')
FROM information_schema.optimizer_trace;
```

- Queries against `information_schema.optimizer_trace` will not generate a new trace, so you can execute multiple queries on the same trace.

More information

Alibaba Cloud | Worldwide Cloud Services Partner

- The optimizer trace manual: https://dev.mysql.com/doc/internals/en/optimizer-tracing.html
  - Enable/disable trace of selected features of the optimizer
  - Tune trace purging (by default an optimizer trace overwrites the previous trace)
  - Make it more human readable (but no longer valid JSON)

- My blog: https://oysteing.blogspot.com

- For information about MySQL related work at Alibaba:

  Attend the **AliSQL & POLARDB** track tomorrow

Thank You!

Please, remember to rate this session in the Percona Live APP!