

What's new in MySQL Query Optimizer

Sergey.Petrunya@sun.com

MySQL University session
Dec 2008

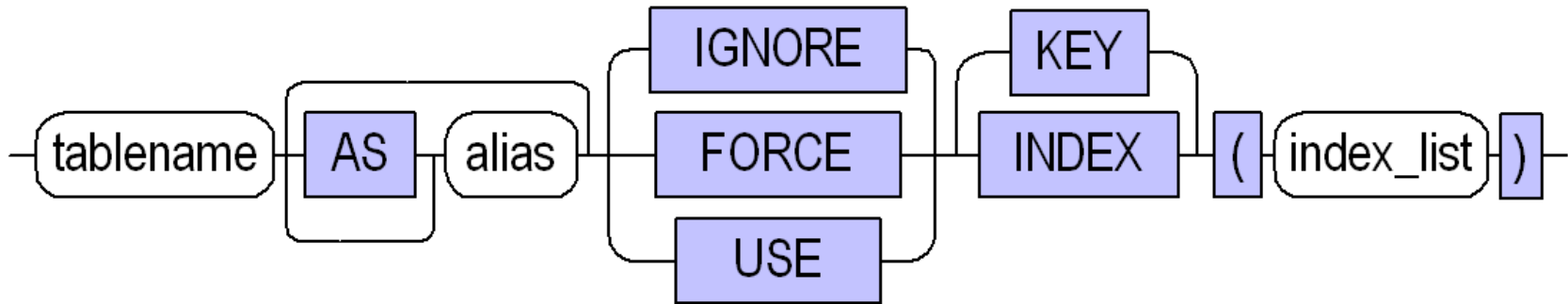
What's new in MySQL Optimizer

- MySQL 5.1 GA has been released
 - Differences between 5.0 and 5.1 optimizers
 - Overview of known problems
- MySQL 6.0 optimizer state
 - State of the new features
 - Currently unresolved issues
- Optimizer development beyond 6.0
 - Committed and uncommitted plans

Optimizer changes in MySQL 5.1

- Extended SQL hints syntax
- Join optimizer improvements
- Better ORDER BY ... LIMIT optimization
- INFORMATION_SCHEMA access optimization
- More information in EXPLAIN output

Index hint syntax and semantics in MySQL 5.0



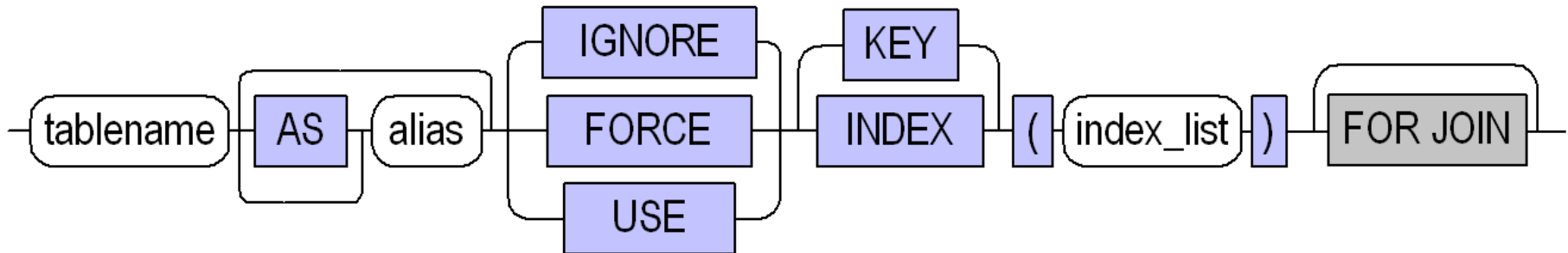
Tell the optimizer whether index(es) can be used for joining/ordering/grouping.

- IGNORE – don't use the listed indexes
- USE – use only listed indexes
- FORCE – use only listed indexes and also assume full table scan to be very expensive

Special cases

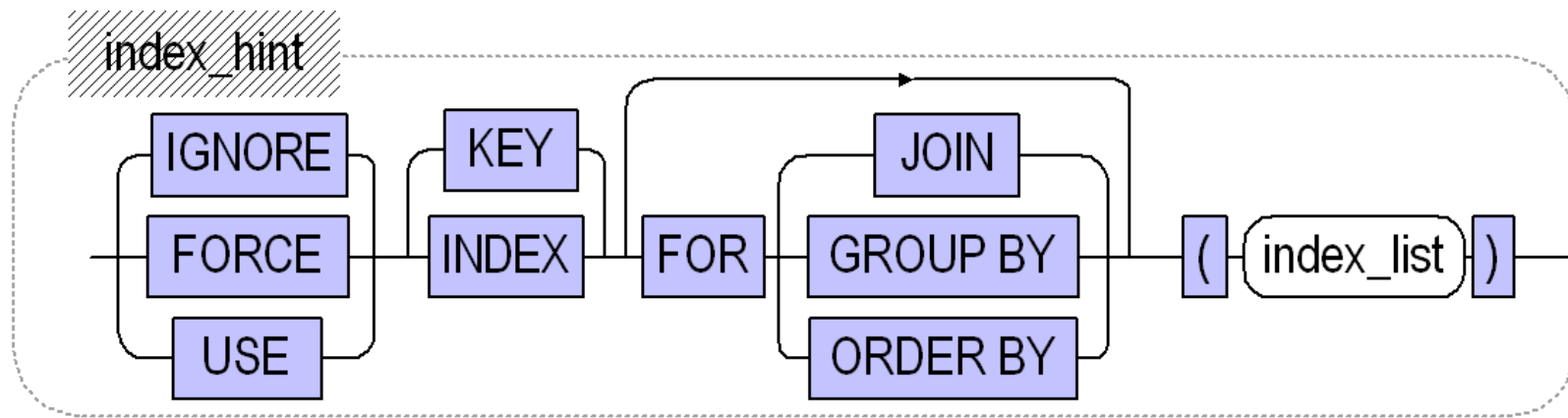
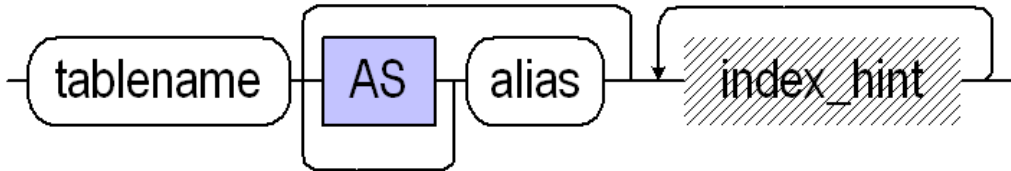
- Fulltext index hints are ignored (will be honored for BOOLEAN MODE searches after the next 5.0.x release, see BUG#38842)

Index hints in MySQL 5.0: late additions



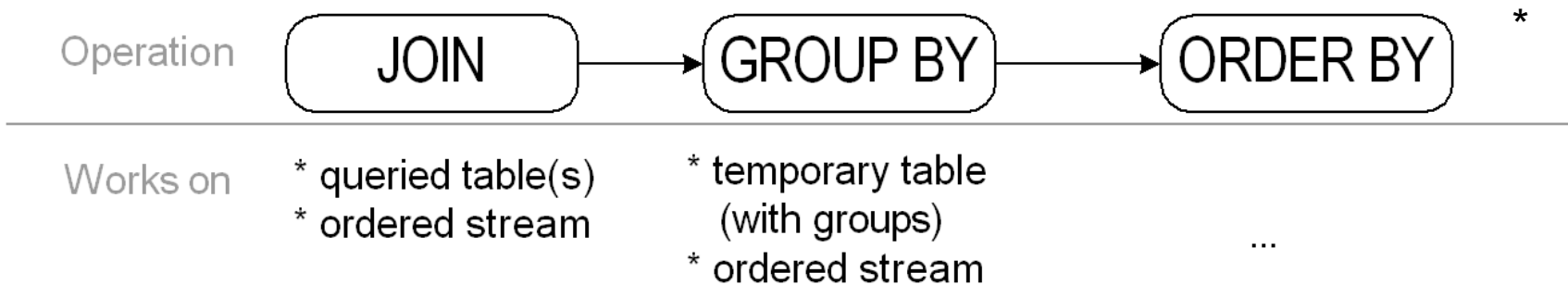
- Can now specify empty `tablename USE INDEX()` which means “Use no indexes”
- “FOR JOIN” clause added for compatibility with 5.1
 - it doesn't affect the meaning

New in 5.1: Extended index hint syntax



- Hints are used to specify three sets of indexes
 - to use for range/index_merge/joins
 - to use for resolving ORDER BY
 - to use for resolving GROUP BY
 - and DISTINCT when it can be reduced to GROUP BY

New in 5.1: Extended index hint syntax (2)



Meaningful combinations match this pattern

```
table USE      INDEX FOR JOIN(idx)
      IGNORE  INDEX FOR ORDER BY(idx)
```

where you can substitute ORDER for GROUP and/or change USE/FORCE to IGNORE or vice versa.

(*) there was a whole session about ORDER BY and GROUP BY handling. will provide a link at the end of this session

Index hints: what's missing and future

What's missing

- Most common complaint: no way to control `index_merge`
 - You need to do it, for both good (correlated data) and bad (BUG#32254) reasons
 - But it's not possible to force or disable `index_merge`
 - The only thing you can do is to disallow `index_merge` by disabling all indexes except one
 - Why we don't have a hint already?
 - There are many ways to perform index merge
 - It's not easy to extend our SQL grammar

The future

- Want hints for `index_merge` and more
- The major roadblock: no hint syntax support system
- There will be `@@optimizer_switch` for every optimization

Optimizer changes in MySQL 5.1

- Extended SQL hints syntax
- **Join optimizer improvements**
- Better ORDER BY ... LIMIT optimization
- INFORMATION_SCHEMA access optimization
- More information in EXPLAIN output

New in 5.1: join optimizer improvements: 'filtered'

“filtered” column in EXPLAIN EXTENDED and its meaning

```
EXPLAIN EXTENDED
```

```
SELECT * FROM teachers T, courses C
```

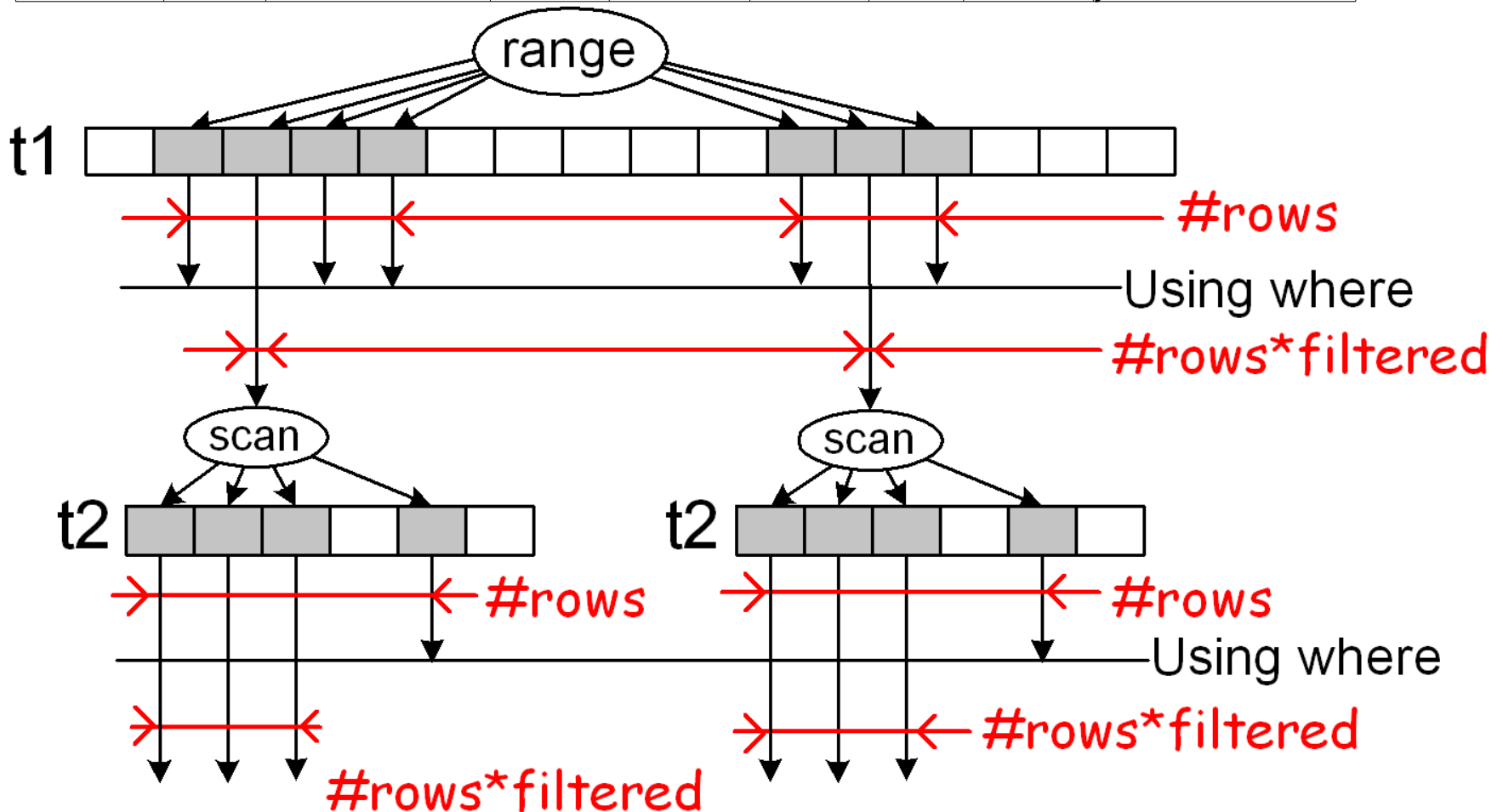
```
WHERE T.cert_level > 5 AND C.title LIKE 'foo%';
```

table	type	possible_keys	key	key_len	ref	rows	filtered	Extra
t1	range	b	b	5	NULL	334	100	Using where
t2	ALL	b	NULL	NULL	NULL	8000	51.04	Using where; Using join buffer

- **rows** is how many records we expect to scan in the table
- **filtered** is shows percentage of records that will be left after attached part of the WHERE was applied.

New in 5.1: join optimizer improvements: 'filtered' (2)

table	type	possible_keys	key	key_len	ref	rows	filtered	Extra
t1	range	b	b	5	NULL	334	100	Using where
t2	ALL	b	NULL	NULL	NULL	8000	51.04	Using where; Using join buffer



New in 5.1: join optimizer improvements: 'filtered' (3)

How can one calculate the 'filtered' ratio?

```

SELECT ... FROM ... tbl ...
WHERE ...
    tbl.key1 < 123 AND           -- (1)
    tbl.key2 = 'foo' AND        -- (2)
    tbl.not_key1 = 'bar' AND    -- (3)
    tbl.key3 = othertbl.col     -- (4)

```

- (1) and (2) – can get estimates from the range optimizer
 - but not for “(1) AND (2)”
- (3) – we don't know anything about data in non-key columns
- (4) – requires knowledge of data in a non-key column in another table. perhaps not in the whole table but in the subset the query enumerates.

New in 5.1: join optimizer improvements: 'filtered' (3)

```

SELECT ... FROM ... tbl ...
WHERE ...
    tbl.key1 < 123 AND           -- (1)
    tbl.key2 = 'foo' AND        -- (2)
    tbl.not_key1 = 'bar' AND    -- (3)
    tbl.key3 = othertbl.col     -- (4)

```

Classic solution

- Assume conditions on different columns are satisfied independently
- Use histograms data
- Assume fixed selectivity for unknown predicates. eg. 50%

MySQL's solution

- Pessimistic: assume conditions on different columns are true together.
- Re-use estimates we got from range access
- Heuristic: if doing full scan, tbl.key=othertbl.col filters out 25%

Optimizer changes in MySQL 5.1

- Extended SQL hints syntax
- Join optimizer improvements
- **Better ORDER BY ... LIMIT optimization**
- INFORMATION_SCHEMA access optimization
- More information in EXPLAIN output

ORDER BY ... [LIMIT] optimization

MySQL 5.0

- Run the join optimizer and choose the best join order
- Look at the indexes of the first non-constant table
 - If there is an index which we could use to resolve ORDER BY/GROUP BY without doing filesort(), switch to using that index

MySQL 5.1

- Run the join optimizer and choose the best join order
- Look at the indexes of the first non-constant table
 - Make a cost-based choice between using index or filesort()

New optimization properties

- You can control it with USE/IGNORE INDEX FOR ORDER BY hints (well, will be after BUG#NNNN is fixed)
- High risk of error: ... WHERE x ORDER BY y LIMIT 1

Optimizer changes in MySQL 5.1

- Extended SQL hints syntax
- Join optimizer improvements
- Better ORDER BY ... LIMIT optimization
- **INFORMATION_SCHEMA access optimization**
- More information in EXPLAIN output

New in 5.1: INFORMATION_SCHEMA optimizations

- MySQL has no centralized data dictionary
 - Databases are directories, tables have .frm files
 - Table's DDL info is in .frm file
 - Table's triggers are in .trg file(s)
 - Statistics is in the storage engine, you need to “open” the table to get it

- Now consider a query

```
SELECT ...
FROM information_schema.tables T
WHERE T.table_name='t1' AND ...
```

- Naïve execution strategy
 - Create a temporary table T according to information_schema.tables' DDL.
 - Scan through all tables in all databases and populate table T
 - Run the query against T.

INFORMATION_SCHEMA optimization in MySQL 5.0

```
populate_IS_pseudo_table($IS_table) {  
  
    part_cond= <part of WHERE that refers only to  
                $IS_table.TABLE_NAME and  
                $IS_table.TABLE_SCHEMA>  
  
    for each database D and its table T {  
        if (part_cond(T,D)) {  
            open table (T,D);  
            copy its data into temptable;  
        }  
    }  
}
```

INFORMATION_SCHEMA optimization in MySQL 5.0

```

populate_IS_pseudo_table($IS_table) {
+   $open_method= <analyze what columns of $IS_table are
+                   needed and see if we can
+                   - just open the table
+                   - or even just open the frm
+   $part_cond= <part of WHERE that refers only to
+               $IS_table.TABLE_NAME and
+               $IS_table.TABLE_SCHEMA>;

+   $dbs= <try to find if we only need one db>;
+   $table= <try to find if we only need one table>;

-   for each database D and its table T {
+   for each database D in $dbs and its table T in $tbls {
+       if ($part_cond(T,D)) {
-           open table (T,D);
+           open table with $open_method (T,D);
+           copy its data into temptable;
+       }
+   }
}

```

Optimizer changes in MySQL 5.1

- Extended SQL hints syntax
- Join optimizer improvements
- Better ORDER BY ... LIMIT optimization
- INFORMATION_SCHEMA access optimization
- **More information in EXPLAIN output**

New in 5.1: EXPLAIN changes summary

- **EXPLAIN EXTENDED** shows “filtered” column
- Introduced **EXPLAIN PARTITIONS** which shows “partitions” column with lists of used partitions
- **Extra** column may contain
 - INFORMATION_SCHEMA optimization keywords
 - `Skip_open_table`
 - `Open_frm_only`
 - `Open_full_table`
 - `Scanned (0|1|all) databases`
 - “Using join buffer”

Join buffering was there at least since 4.1, we've just made it visible in EXPLAIN. See links and references for detailed explanation of what it is.

Known MySQL 5.1 problems

Overall status

- One known bug in optimizer (index_merge over MERGE)
- But plenty of slowdowns

Top-most bugs

- BUG#32254 (index_merge used unnecessarily)
- BUG#36569 (UPDATE ... ORDER BY always does a filesort when required)
- ...
- A proposal how to deal with them:
<http://s.petrunia.net/blog/?p=48>

News in MySQL 6.0 optimizer

New optimizer features in MySQL 6.0

- MultiRangeRead and Index Condition Pushdown
 - MyISAM
 - InnoDB (probably)
 - Falcon(in BKA tree)
 - Maria (in BKA tree)
- Subquery optimizations
- Batched Key Access

Not a query optimizer features

- DTrace support (unsuitable for debugging query optimizer ATM)

Links and references

- **Index hints page in MySQL 5.1 manual**
<http://dev.mysql.com/doc/refman/5.1/en/index-hints.html>
- **How MySQL handles ORDER BY, GROUP BY, and DISTINCT**
http://forge.mysql.com/wiki/How_MySQL_Handles_ORDER_BY%2C_GROUP_BY%2C_and_DISTINCT
- **Use of join buffer is now visible in EXPLAIN**
<http://s.petrunia.net/blog/?p=18>
- **Reference Manual: What's new in MySQL 5.1**
<http://dev.mysql.com/doc/refman/5.1/en/mysql-nutshell.html>
-