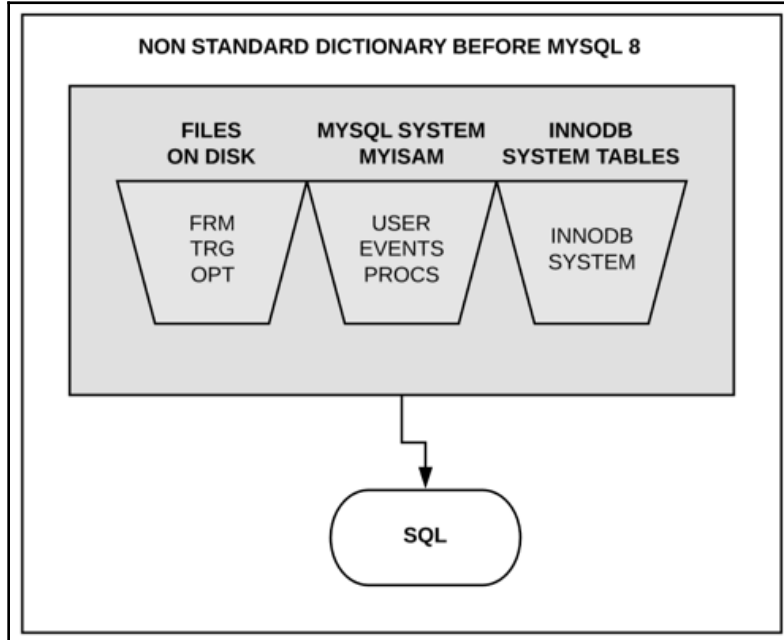
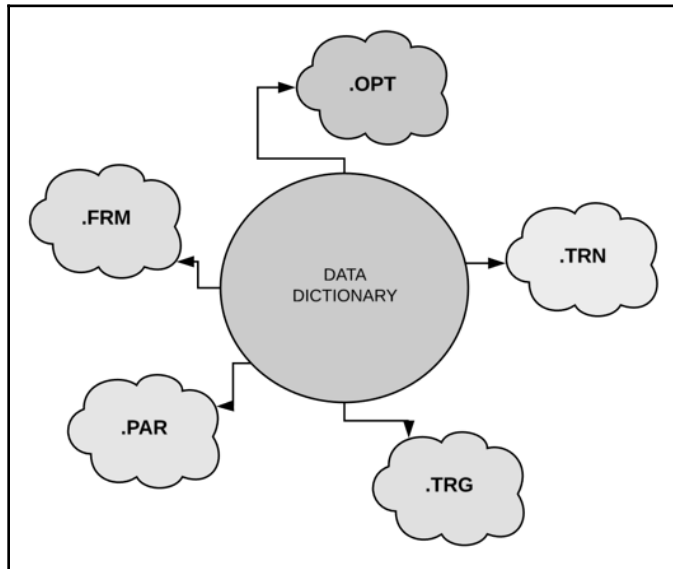


Chapter 02: MySQL 8's New Features



```

+-----+
| Grants for userdev1@localhost |
+-----+
| GRANT USAGE ON *.* TO `userdev1`@`localhost` |
| GRANT `app_developer`@`%` TO `userdev1`@`localhost` |
+-----+

```

```

+-----+
| Grants for userdev1@localhost |
+-----+
| GRANT USAGE ON *.* TO `userdev1`@`localhost` |
| GRANT ALL PRIVILEGES ON `app_db`.* TO `userdev1`@`localhost` |
| GRANT `app_developer`@`%` TO `userdev1`@`localhost` |
+-----+

```

```

***** 1. row *****
RESOURCE_GROUP_NAME: USR_default
RESOURCE_GROUP_TYPE:
USER RESOURCE_GROUP_ENABLED: 1
VCPUS_IDS: 0-8
THREAD_PRIORITY: 0
***** 2. row *****
RESOURCE_GROUP_NAME: SYS_default
RESOURCE_GROUP_TYPE: SYSTEM
RESOURCE_GROUP_ENABLED: 1
VCPUS_IDS: 0-8
THREAD_PRIORITY: 0

```

```

***** 1. row *****
RESOURCE_GROUP_NAME: Batch_job
RESOURCE_GROUP_TYPE: USER
RESOURCE_GROUP_ENABLED: 1
VCPUS_IDS: 6-8
THREAD_PRIORITY: 8

```

```
***** 1. row *****
RESOURCE_GROUP_NAME: ABCReporting
RESOURCE_GROUP_TYPE: USER
RESOURCE_GROUP_ENABLED: 1
          VCPU_IDS: 4-5
          THREAD_PRIORITY: 10
```

```
mysql> CREATE TABLE t1 (doc JSON);
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO t1 VALUES ('[1, 2, 3, 4, 5]');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT doc->"$[1 to 3]" FROM t1;
+-----+
| doc->"$[1 to 3]" |
+-----+
| [2, 3, 4]       |
+-----+
1 row in set (0.00 sec)

mysql> SELECT doc->"$[last-2]" FROM t1;
+-----+
| doc->"$[last-2]" |
+-----+
| 3                |
+-----+
1 row in set (0.00 sec)
```

```

mysql> CREATE TABLE jsonemp (
  -> c JSON,
  -> g INT GENERATED ALWAYS AS (c->"$.id"),
  -> INDEX i (g)
  -> );
Query OK, 0 rows affected (0.23 sec)

mysql> INSERT INTO jsonemp (c) VALUES
  > ('{"id": "1", "name": "Don"}'), ('{"id": "2", "name": "Lola"}'),
  > ('{"id": "3", "name": "Bob"}'), ('{"id": "4", "name": "Michele"}');
Query OK, 4 rows affected (0.05 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> SELECT c->"$.name" AS name
  > FROM jsonemp WHERE g > 2;

+-----+
| name  |
+-----+
| Don   |
| Lola  |
+-----+
2 rows in set (0.00 sec)

mysql> EXPLAIN SELECT c->"$.name" AS name
  > FROM jsonemp WHERE g > 2\G
***** 1. row *****
      id: 1
select_type: SIMPLE
      table: jsonemp
partitions: NULL
      type: range
possible_keys: i
      key: i
      key_len: 5
      ref: NULL
      rows: 2
  filtered: 100.00
  Extra: Using where

```

```
mysql> SELECT * FROM sys.schema_unused_indexes;
+-----+-----+-----+
| object_schema | object_name | index_name |
+-----+-----+-----+
| Student1      | Classroom   | s          |
| Student2      | Classroom   | s_c        |
+-----+-----+-----+
2 rows in set (0.01 sec)
```

```
ALTER TABLE classroom ALTER INDEX c INVISIBLE;
```

```
ALTER TABLE Classroom ADD INDEX c (student1) INVISIBLE;
# after some time
ALTER TABLE Classroom ALTER INDEX c VISIBLE;
```

```
SHOW INDEXES FROM my_table;
```

```
|-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name          | Seq_in_index | Column_name | Collation
| x_table | 1          | just_to_be_safe_idx | 1            | a           | A
|-----+-----+-----+-----+-----+
```

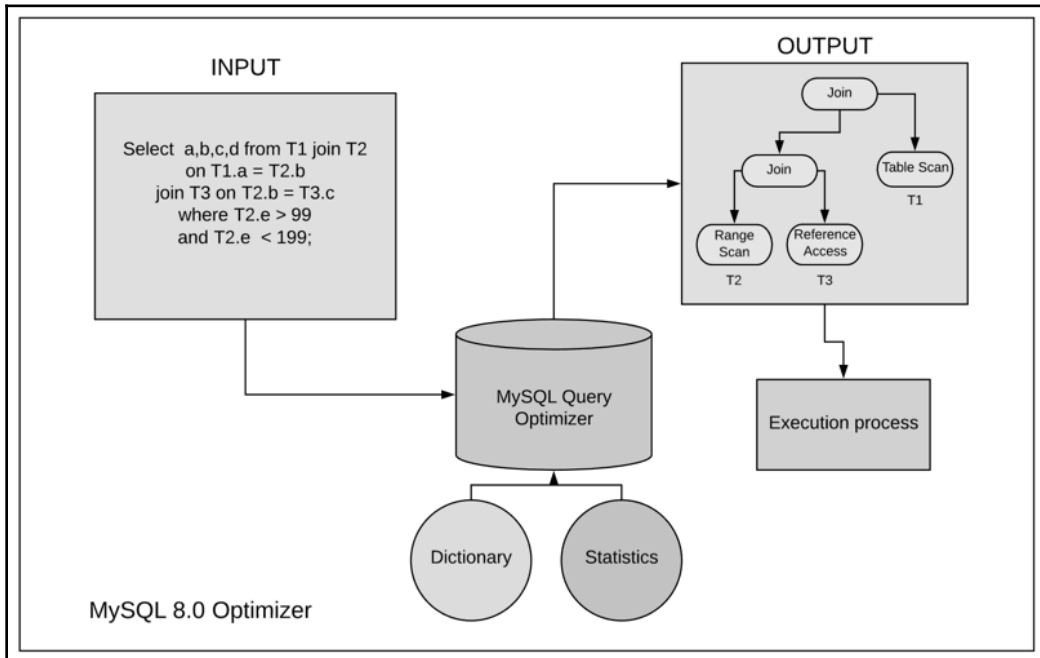
```
SELECT * FROM information_schema.statistics WHERE is_visible='NO';
```

```
***** 1. row *****
```

```
TABLE_CATALOG: def  
TABLE_SCHEMA: student1  
TABLE_NAME: Classroom  
NON_UNIQUE: 1  
INDEX_SCHEMA: student1  
INDEX_NAME: c  
SEQ_IN_INDEX: 1  
COLUMN_NAME: Classroom  
COLLATION: A  
CARDINALITY: 7  
SUB_PART: NULL  
PACKED: NULL  
NULLABLE:  
INDEX_TYPE: BTREE  
COMMENT: disabled  
INDEX_COMMENT:  
IS_VISIBLE: NO
```

Old Name	New Name
INNODB_SYS_COLUMNS	INNODB_COLUMNS
INNODB_SYS_DATAFILES	INNODB_DATAFILES
INNODB_SYS_FIELDS	INNODB_FIELDS
INNODB_SYS_FOREIGN	INNODB_FOREIGN
INNODB_SYS_FOREIGN_COLS	INNODB_FOREIGN_COLS
INNODB_SYS_INDEXES	INNODB_INDEXES
INNODB_SYS_TABLES	INNODB_TABLES
INNODB_SYS_TABLESPACES	INNODB_TABLESPACES
INNODB_SYS_TABLESTATS	INNODB_TABLESTATS
INNODB_SYS_VIRTUAL	INNODB_VIRTUAL

Chapter 03: Indexing Your Data for High Performance



Cost	Operation
40	disk_temptable_create_cost
1	disk_temptable_row_cost
2	memory_temptable_create_cost
0.2	memory_temptable_row_cost
0.1	key_compare_cost
0.2	row_evaluate_cost
1	io_block_read_cost
1	memory_block_read_cost

Type	Storage (Bytes)	Minimum Value (Signed/Unsigned)	Maximum Value (Signed/Unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

```

CREATE TABLE music_album (
  music_album_id int(10) unsigned NOT NULL,
  music_artist_id int(10) unsigned NOT NULL,
  music_album_type_id int(10) unsigned NOT NULL,
  name varchar(255) NOT NULL,
  first_released year(4) NOT NULL,
  music_country_id smallint(5) unsigned DEFAULT NULL,
  PRIMARY KEY (music_album_id),
  KEY artist_id (music_artist_id),
  KEY country_id (music_country_id),
  KEY music_album_type_id (music_album_type_id),
  KEY idx1 (music_country_id,music_album_type_id),
  KEY idx2 (music_album_type_id, music_country_id)
) ENGINE=InnoDB DEFAULT CHARSET=latin1

```

```
DROP TABLE IF EXISTS test1;
CREATE TABLE test1 LIKE music_album;
INSERT INTO test1 SELECT * FROM music_album;
DROP TABLE test1;

CREATE TABLE test1 LIKE music_album;
ALTER TABLE test1 DROP INDEX first_released, DROP
INDEX music_album_type_id,
DROP INDEX name, DROP INDEX music_country_id,
DROP INDEX idx1, DROP INDEX idx2;
INSERT INTO test1 SELECT * FROM music_album;
DROP TABLE test1;
```

```
# Insert with indexes
Query OK, 563881 rows affected (23.34 sec)

# Insert without indexes
Query OK, 563881 rows affected (7.54 sec)
```

```
CREATE TABLE music_album (
...
PRIMARY KEY (music_album_id),
KEY artist_id (music_artist_id),
KEY country_id (music_country_id),
KEY music_album_type_id (music_album_type_id),
KEY idx1 (music_country_id,music_album_type_id),
KEY idx2 (music_album_type_id, music_country_id)
...)
```

```
mysql> show create table yourexample_table\G
*****1 row*****
      Table: yourexample_table
Create Table: CREATE TABLE `yourexample_table` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) DEFAULT NULL,
  `address` varchar(140) DEFAULT NULL,
  `username` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `username` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```

```
mysql> explain yourexample_table;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(10)       | NO   | PRI | NULL    | auto_increment |
| name       | varchar(50)   | YES  |     | NULL    |                |
| address    | varchar(140)  | YES  |     | NULL    |                |
| username   | varchar(20)   | YES  | MUL | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc yourexample_table;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(10)       | NO   | PRI | NULL    | auto_increment |
| name       | varchar(50)   | YES  |     | NULL    |                |
| address    | varchar(140)  | YES  |     | NULL    |                |
| username   | varchar(20)   | YES  | MUL | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```

mysql> SHOW CREATE TABLE complex_example_table\G
***** 1. row *****
      Table: complex_example_table
Create Table: CREATE TABLE `complex_example_table` (
  `ID1` int(10) NOT NULL auto_increment,
  `ID2` int(10) NOT NULL default '0',
  `STORYB` varchar(200) NOT NULL default '',
  `STORYC` varchar(180) default NULL,
  PRIMARY KEY (`ID1`,`ID2`,`STORYB`),
  KEY `STORYB` (`STORYB`,`STORYC`),
  KEY `STORYC` (`STORYC`),
  CONSTRAINT `IDX_ID1_ID2` FOREIGN KEY (`ID1`,`ID2`)
REFERENCES `complex_example_table` (`ID1`,`ID2`)
ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `IDX_` FOREIGN KEY (`STORYB`,`STORYC`)
REFERENCES `second_example_table` (`STORYB`,`STORYC`)
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=INNODB CHARSET=utf8

```

1. REFERENCES `complex_example_table` (`ID1`, `ID2`) ON DELETE CASCADE ON UPDATE CASCADE
2. REFERENCES `second_example_table` (`STORYB`, `STORYC`) ON DELETE CASCADE ON UPDATE CASCADE

- CONSTRAINT `IDX_` FOREIGN KEY (`STORYB`, `STORYC`) REFERENCES `second_example_table` (`STORYB`, `STORYC`)

```
CREATE TABLE parent (  
  id INT NOT NULL AUTO_INCREMENT,  
  bogus_column char(32),  
  PRIMARY KEY (id)  
) ENGINE=InnoDB;  
  
CREATE TABLE child (  
  id INT NOT NULL AUTO_INCREMENT,  
  parent_id INT NOT NULL,  
  bogus_column char(32),  
  PRIMARY KEY (id),  
  KEY (parent_id),  
  CONSTRAINT child_ibfk_1 FOREIGN KEY (parent_id) REFERENCES parent (id)  
) ENGINE=InnoDB;  
  
INSERT INTO parent (bogus_column)  
VALUES ('aaa'), ('bbb'), ('ccc'), ('ddd'), ('eee');  
  
INSERT INTO child (parent_id,bogus_column) VALUES  
(1, 'aaa'), (2, 'bbb'), (3, 'ccc'),  
(4, 'ddd'), (5, 'eee');
```

```
START TRANSACTION; # session1  
START TRANSACTION; # session2  
  
# session1  
UPDATE child SET parent_id = 5 WHERE parent_id = 4;  
  
#session2  
UPDATE parent SET bogus_column = 'new!' WHERE id = 4;
```

```

mysql> SELECT * FROM information_schema.innodb_locks\G
***** 1. row *****
      lock_id: 87035:1300:3:6
lock_trx_id: 87035
      lock_mode: X
      lock_type: RECORD
lock_table: `test`.`parent`
lock_index: `PRIMARY`
lock_space: 1300
      lock_page: 3
      lock_rec: 6
      lock_data: 5
***** 2. row *****
      lock_id: 87034:1300:3:6
lock_trx_id: 87034
      lock_mode: S
      lock_type: RECORD
lock_table: `test`.`parent`
lock_index: `PRIMARY`
lock_space: 1300
      lock_page: 3
      lock_rec: 6
      lock_data: 5
2 rows in set (0.00 sec)

```

```

mysql> explain select * from tbl_report\G
***** 1. row *****
      id: 1
select_type: SIMPLE
      table: tbl_report
partitions: NULL
      type: ALL
possible_keys: NULL
      key: NULL
      key_len: NULL
      ref: NULL
      rows: 1
filtered: 100.00
      Extra: NULL
1 row in set, 1 warning (0.00 sec)

```

Table	Op	Msg_type	Msg_text
app.clients	analyze	status	OK

Column	JSON Name	Explanation
id	select_id	The SELECT identifier (DML)
select_type	None	The SELECT type (See below for details)
table	table_name	The table name for the output row
partitions	partitions	The matching partition names
type	access_type	The join type (Inner_join...)
possible_keys	possible_keys	The possible indexes that MySQL can use
key	key	The potential index that it can be chosen
key_len	key_length	The length of the potential key
ref	ref	The columns compared to the index chosen
rows	rows	Estimate of rows that will be taken in consideration
filtered	filtered	% of rows filtered by table condition
Extra	None	Additional helpful information

select_type Value	JSON Name	Explanation
SIMPLE	None	Simple SELECT (doesn't include UNION or subqueries)
PRIMARY	None	First most important SELECT
UNION	None	Second SELECT in a UNION
DEPENDENT UNION	dependent (true)	Second SELECT in a UNION, dependent on the outer query
UNION RESULT	union_result	Result of the UNION
SUBQUERY	None	First SELECT in a subquery
DEPENDENT SUBQUERY	dependent (true)	First SELECT in a subquery, dependent on the outer query
DERIVED	None	Derived table style SELECT (subquery in FROM clause)
MATERIALIZED	materialized_from_subquery	Materialized subquery (New option)
UNCACHEABLE SUBQUERY	cacheable (false)	A subquery where the result cannot be cached and have to be re-evaluated for each row from the outer query
UNCACHEABLE UNION	cacheable (false)	The second select in a UNION that is part of an uncacheable subquery

```
CREATE TABLE employees (
  ID INT,
  employee_name VARCHAR(50),
  employee_salary decimal(10,2),
  hired_date (date)
)
```



```
mysql> show create table people \G
***** 1. row *****
      Table: people
Create Table: CREATE TABLE `people` (
  `firstname` varchar(50) NOT NULL DEFAULT '',
  `lastname` varchar(50) NOT NULL DEFAULT '',
  `mobile` varchar(20) NOT NULL DEFAULT '',
  `birthday` date NOT NULL,
  `home_id` smallint(6) DEFAULT NULL,
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`ID`),
  KEY `birthday` (`birthday`,`lastname`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```

```
mysql> show create table people\G
***** 1. row *****
```

```
Table: people
Create Table: CREATE TABLE `people` (
  `firstname` varchar(50) NOT NULL DEFAULT '',
  lastname varchar(50) NOT NULL DEFAULT '',
  `mobile` varchar(20) NOT NULL DEFAULT '',
  `birthday` date NOT NULL,
  `home_id` smallint(6) DEFAULT NULL,
  ID int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```

```
mysql> ALTER TABLE people
ADD INDEX (birthday),
ADD INDEX (lastname);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> ALTER TABLE people DROP index birthday, drop index lastname;
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE people ADD INDEX (birthday);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE people ADD INDEX (lastname);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> explain select p.id, p.firstname, p.lastname from people p where p.birthday = '1996-05-27' and lastname = 'Thompson';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | p | NULL | ref | birthday | birthday | 55 | const,const | 1 | 100.00 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)
```

```
mysql> ALTER TABLE people DROP index birthday, drop index lastname;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> show create table people \G
***** 1. row *****
      Table: people
Create Table: CREATE TABLE `people` (
  `firstname` varchar(50) NOT NULL DEFAULT '',
  `lastname` varchar(50) NOT NULL DEFAULT '',
  `mobile` varchar(20) NOT NULL DEFAULT '',
  `birthday` date NOT NULL,
  `home_id` smallint(6) DEFAULT NULL,
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`ID`),
  KEY `birthday` (`birthday`,`lastname`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```

```
mysql> ALTER TABLE people add index (birthday, lastname);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> show create table people \G
***** 1. row *****
      Table: people
Create Table: CREATE TABLE `people` (
  `firstname` varchar(50) NOT NULL DEFAULT '',
  `lastname` varchar(50) NOT NULL DEFAULT '',
  `mobile` varchar(20) NOT NULL DEFAULT '',
  `birthday` date NOT NULL,
  `home_id` smallint(6) DEFAULT NULL,
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`ID`),
  KEY `birthday` (`birthday`,`lastname`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```

```
mysql> explain select p.id, p.firstname, p.lastname from people p where p.birthday = '1996-05-27' and lastname = 'Thompson';
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | p | NULL | ref | birthday | birthday | 55 | const,const | 1 | 100.00 | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)
```

```
mysql> explain select p.id, p.firstname, p.lastname from people p where lastname = 'Thompson' and p.birthday = '1996-05-27';
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | p | NULL | ALL | NULL | NULL | NULL | NULL | 8 | 12.50 | Using where |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)
```

```
mysql> show create table people \G
***** 1. row *****
      Table: people
Create Table: CREATE TABLE `people` (
  `firstname` varchar(50) NOT NULL DEFAULT '',
  `lastname` varchar(50) NOT NULL DEFAULT '',
  `mobile` varchar(20) NOT NULL DEFAULT '',
  `birthday` date NOT NULL,
  `home_id` smallint(6) DEFAULT NULL,
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`ID`),
  KEY `birthday` (`birthday`,`lastname`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```

```
mysql> explain select p.id, p.firstname, p.lastname from people p where p.birthday = '1996-05-27' and lastname = 'Thompson';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | p | NULL | ref | birthday | birthday | 55 | const,const | 1 | 100.00 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)
```

MySQL 8.x	MySQL 5.7	MySQL 5.6
<pre>{EXPLAIN DESCRIBE DESC} tbl_name [col_name wild]</pre>	<pre>{EXPLAIN DESCRIBE DESC} tbl_name [col_name wild]</pre>	<pre>{EXPLAIN DESCRIBE DESC} tbl_name [col_name wild]</pre>
<pre>{EXPLAIN DESCRIBE DESC} [explain_type] {explainable_stmt FOR CONNECTION connection_id}</pre>	<pre>{EXPLAIN DESCRIBE DESC} [explain_type] {explainable_stmt FOR CONNECTION connection_id}</pre>	<pre>{EXPLAIN DESCRIBE DESC} [explain_type] explainable_stmt</pre>
<pre>explain_type: { FORMAT = format_name }</pre>	<pre>explain_type: { EXTENDED PARTITIONS FORMAT = format_name }</pre>	<pre>explain_type: { EXTENDED PARTITIONS FORMAT = format_name }</pre>
<pre>format_name: { TRADITIONAL JSON }</pre>	<pre>format_name: { TRADITIONAL JSON }</pre>	<pre>format_name: { TRADITIONAL JSON }</pre>
<pre>explainable_stmt: { SELECT statement DELETE statement INSERT statement REPLACE statement UPDATE statement }</pre>	<pre>explainable_stmt: { SELECT statement DELETE statement INSERT statement REPLACE statement UPDATE statement }</pre>	<pre>explainable_stmt: { SELECT statement DELETE statement INSERT statement REPLACE statement UPDATE statement }</pre>

```


mysql> explain FORMAT=JSON select p.id, p.firstname, p.lastname from people p where lastname = 'Thompson' and p.birthday =
'1996-05-27'\G
***** 1. row *****
EXPLAIN: {
  "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "2.60" ← 1
    },
  },
  "table": {
    "table_name": "p",
    "access_type": "ALL",
    "rows_examined_per_scan": 8, ← 2
    "rows_produced_per_join": 1, ← 3
    "filtered": "12.50",
    "cost_info": {
      "read_cost": "2.40", ← 4
      "eval_cost": "0.20",
      "prefix_cost": "2.60",
      "data_read_per_join": "136" ← 5
    },
    "used_columns": [
      "firstname",
      "lastname",
      "birthday",
      "ID"
    ],
    "attached_condition": "((`addresses`.`p`.`birthday` = '1996-05-27') and (`addresses`.`p`.`lastname` = 'Thompson'))"
  }
}

```


```
mysql> explain FORMAT=JSON select p.id,
p.firstname, p.lastname from people p where
lastname = 'Thompson' and p.birthday =
'1996-05-27'\G

EXPLAIN: {
  "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "2.60"
    },
    "table": {
      "table_name": "p",
      "access_type": "ALL",
      "rows_examined_per_scan": 8,
      "rows_produced_per_join": 1,
      "filtered": "12.50",
      "cost_info": {
        "read_cost": "2.40",
        "eval_cost": "0.20",
        "prefix_cost": "2.60",
        "data_read_per_join": "136"
      },
      "used_columns": [
        "firstname",
        "lastname",
        "birthday",
        "ID"
      ],
      "attached_condition":
      "((`addresses`.`p`.`birthday` =
      '1996-05-27') and
      (`addresses`.`p`.`lastname` = 'Thompson'))"
    }
  }
}
```

```
show create table people \G
Table: people
Create Table: CREATE TABLE `people` (
  `firstname` varchar(50) NOT NULL DEFAULT '',
  `lastname` varchar(50) NOT NULL DEFAULT '',
  `mobile` varchar(20) NOT NULL DEFAULT '',
  `birthday` date NOT NULL,
  `home_id` smallint(6) DEFAULT NULL,
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
-----
mysql> explain people;
-----
| Field      | Type          | Null | Key | Default | Extra
-----
| firstname  | varchar(50)  | NO   |     |         |
| lastname   | varchar(50)  | NO   |     |         |
| mobile     | varchar(20)  | NO   |     |         |
| birthday   | date         | NO   |     | NULL    |
| home_id    | smallint(6)  | YES  |     | NULL    |
| ID         | int(11)      | NO   | PRI | NULL    | auto_increment
-----
6 rows in set (0.02 sec)
```



No indexes



Same output as the command DESC

Only 1 column	2 columns	3 columns et plus
We include these instructions: <ol style="list-style-type: none"> 1. Operators: <>= 2. LIKE & NOT LIKE 3. IN() & NOT IN() 	We include these instructions: <ol style="list-style-type: none"> 1. BETWEEN & NOT 2. AND & OR 3. Operators: <>= 4. NOT 5. LIKE & NOT LIKE 	We include these instructions: <ol style="list-style-type: none"> 1. All instructions from column #1 and #2 2. Parenthesis We exclude all sub-queries.
Index strategies	Index strategies	Index strategies
<ul style="list-style-type: none"> ● Primary key ● Unique index ● Partial index 	<ul style="list-style-type: none"> ● Partial index ● Compound 	<ul style="list-style-type: none"> ● Compound

```
mysql> show create table people \G
***** 1. row *****
      Table: people
Create Table: CREATE TABLE `people` (
  `firstname` varchar(50) NOT NULL DEFAULT '',
  `lastname` varchar(50) NOT NULL DEFAULT '',
  `mobile` varchar(20) NOT NULL DEFAULT '',
  `birthday` date NOT NULL,
  `home_id` smallint(6) DEFAULT NULL,
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```

```
explain FORMAT=JSON SELECT firstname, lastname, COUNT(*) as number
FROM people
WHERE birthday BETWEEN '1964-01-01' AND '2018-09-01'
GROUP BY lastname
ORDER BY birthday;
```



```

EXPLAIN: {
  "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "2.60"
    },
    "ordering_operation": {
      "using_filesort": true,
      "grouping_operation": {
        "using_temporary_table": true,
        "using_filesort": false,
        "table": {
          "table_name": "people",
          "access_type": "ALL",
          "rows_examined_per_scan": 8,
          "rows_produced_per_join": 1,
          "filtered": "12.50",
          "cost_info": {
            "read_cost": "2.40",
            "eval_cost": "0.20",
            "prefix_cost": "2.60",
            "data_read_per_join": "136"
          },
          "used_columns": [
            "firstname",
            "lastname",
            "birthday",
            "ID"
          ],
          "attached_condition": "('addresses`.`people`.`birthday` between '1964-01-01' and '2018-09-01')"
        }
      }
    }
  }
}

```

```

"used_columns": [
  "firstname",
  "lastname",
  "birthday",
  "ID"
]

```

```

mysql> ALTER TABLE people ADD INDEX (firstname,lastname,birthday,ID);
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

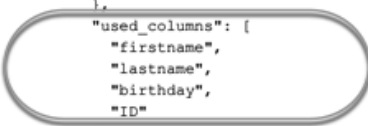
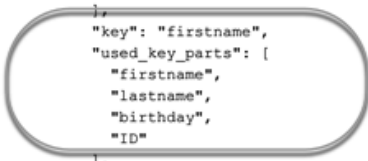
```

mysql> show create table people \G
***** 1. row *****
      Table: people
Create Table: CREATE TABLE `people` (
  `firstname` varchar(50) NOT NULL DEFAULT '',
  `lastname` varchar(50) NOT NULL DEFAULT '',
  `mobile` varchar(20) NOT NULL DEFAULT '',
  `birthday` date NOT NULL,
  `home_id` smallint(6) DEFAULT NULL,
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`ID`),
  KEY `firstname` (`firstname`,`lastname`,`birthday`,`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
1 row in set (0.00 sec)

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	people	NULL	index	firstname	firstname	111	NULL	8	12.50	Using where; Using index; Using temporary; Using filesort

```
EXPLAIN: {
  "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "2.60"
    },
    "ordering_operation": {
      "using_filesort": true,
      "grouping_operation": {
        "using_temporary_table": true,
        "using_filesort": false,
        "table": {
          "table_name": "people",
          "access_type": "index",
          "possible_keys": [
            "firstname"
          ],
          "key": "firstname",
          "used_key_parts": [
            "firstname",
            "lastname",
            "birthday",
            "ID"
          ],
          "key_length": "111",
          "rows_examined_per_scan": 8,
          "rows_produced_per_join": 1,
          "filtered": "12.50",
          "using_index": true,
          "cost_info": {
            "read_cost": "2.40",
            "eval_cost": "0.20",
            "prefix_cost": "2.60",
            "data_read_per_join": "136"
          },
          "used_columns": [
            "firstname",
            "lastname",
            "birthday",
            "ID"
          ],
          "attached_condition": "('addresses`.`people`.`birthday` between '1964-01-01' and '2018-09-01')"
```



Chapter 04: Advanced Data Techniques for Large Queries

```
***** 303. row *****
Variable_name: Select_full_join
      Value: 0
***** 304. row *****
Variable_name: Select_full_range_join
      Value: 0
***** 305. row *****
Variable_name: Select_range
      Value: 1
***** 306. row *****
Variable_name: Select_range_check
      Value: 0
***** 307. row *****
Variable_name: Select_scan
      Value: 56
```

```
PARTITION BY RANGE (Column ID) (
  PARTITION p0 VALUES LESS THAN (Num1),
  PARTITION p1 VALUES LESS THAN (Num2),
  PARTITION p2 VALUES LESS THAN (Num3),
  PARTITION p3 VALUES LESS THAN (Num4)
);
```

```
CREATE TABLE employees (  
    id INT NOT NULL,  
    firstname VARCHAR(30),  
    lastname VARCHAR(30),  
    datehired DATE NOT NULL DEFAULT '1999-01-01',  
    datefinished DATE NOT NULL DEFAULT '9999-12-31',  
    job_id INT NOT NULL,  
    resto_id INT NOT NULL  
)  
PARTITION BY RANGE (resto_id) (  
    PARTITION p0 VALUES LESS THAN (6),  
    PARTITION p1 VALUES LESS THAN (11),  
    PARTITION p2 VALUES LESS THAN (16),  
    PARTITION p3 VALUES LESS THAN (21),  
    PARTITION p4 VALUES LESS THAN (31));
```

```
CREATE TABLE employees (  
    id INT NOT NULL,  
    firstname VARCHAR(30),  
    lastname VARCHAR(30),  
    datehired DATE NOT NULL DEFAULT '1999-01-01',  
    datefinished DATE NOT NULL DEFAULT '9999-12-31',  
    job_id INT NOT NULL,  
    resto_id INT NOT NULL  
)  
PARTITION BY RANGE (resto_id) (  
    PARTITION p0 VALUES LESS THAN (6),  
    PARTITION p1 VALUES LESS THAN (11),  
    PARTITION p2 VALUES LESS THAN (16),  
    PARTITION p3 VALUES LESS THAN (21),  
    PARTITION p4 VALUES LESS THAN (31),  
    PARTITION p5 VALUES LESS THAN MAXVALUE);
```

```
CREATE TABLE inventory_monthly_report (  
    inv_report_id INT NOT NULL,  
    inv_report_status VARCHAR(20) NOT NULL,  
    inv_report_status_updated TIMESTAMP NOT NULL DEFAULT  
    CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP  
)  
PARTITION BY RANGE ( UNIX_TIMESTAMP(inv_report_status_updated))(  
    PARTITION p0 VALUES LESS THAN ( UNIX_TIMESTAMP('2017-01-01  
00:00:00') ),  
    PARTITION p1 VALUES LESS THAN ( UNIX_TIMESTAMP('2017-05-01  
00:00:00') ),  
    PARTITION p2 VALUES LESS THAN ( UNIX_TIMESTAMP('2017-08-01  
00:00:00') ),  
    PARTITION p3 VALUES LESS THAN ( UNIX_TIMESTAMP('2017-10-01  
00:00:00') ),  
    PARTITION p4 VALUES LESS THAN ( UNIX_TIMESTAMP('2018-01-01  
00:00:00') ),  
    PARTITION p5 VALUES LESS THAN ( UNIX_TIMESTAMP('2018-02-01  
00:00:00') ),  
    PARTITION p9 VALUES LESS THAN (MAXVALUE)  
);
```

```
PARTITION BY LIST(resto_id) (  
    PARTITION p1 VALUES IN (v1,v2,v3..),  
    PARTITION p2 VALUES IN (v1,v2,v3..),  
    PARTITION p3 VALUES IN (v1,v2,v3..),  
    PARTITION p4 VALUES IN (v1,v2,v3..)
```

```
CREATE TABLE employees (  
    id INT NOT NULL,  
    firstname VARCHAR(30),  
    lastname VARCHAR(30),  
    datehired DATE NOT NULL DEFAULT '1999-01-01',  
    datefinished DATE NOT NULL DEFAULT '9999-12-31',  
    job_id INT NOT NULL,  
    resto_id INT NOT NULL  
)
```

Region	Resto ID
R1	1, 2, 3, 5
R2	4, 7, 8, 9
R3	10, 22, 23, 13
R4	14, 12, 16, 17

```

CREATE TABLE employees (
  id INT NOT NULL,
  firstname VARCHAR(30),
  lastname VARCHAR(30),
  datehired DATE NOT NULL DEFAULT '1999-01-01',
  datefinished DATE NOT NULL DEFAULT '9999-12-31',
  job_id INT NOT NULL,
  resto_id INT NOT NULL
)
PARTITION BY LIST(resto_id) (
  PARTITION pR1 VALUES IN (1,2,3,5),
  PARTITION pR2 VALUES IN (4,7,8,9),
  PARTITION pR3 VALUES IN (10,22,23,13),
  PARTITION pR4 VALUES IN (14,12,16,17)
);

```

```

CREATE TABLE employees (
  id INT NOT NULL,
  firstname VARCHAR(30),
  lastname VARCHAR(30),
  datehired DATE NOT NULL DEFAULT '1999-01-01',
  datefinished DATE NOT NULL DEFAULT '9999-12-31',
  job_id INT NOT NULL,
  resto_id INT NOT NULL
)
PARTITION BY HASH(resto_id)
PARTITIONS 4;

```

```
CREATE TABLE employees (  
    id INT NOT NULL,  
    firstname VARCHAR(30),  
    lastname VARCHAR(30),  
    datehired DATE NOT NULL DEFAULT '1999-01-01',  
    datefinished DATE NOT NULL DEFAULT '9999-12-31',  
    job_id INT NOT NULL,  
    resto_id INT NOT NULL  
)  
PARTITION BY HASH(resto_id)  
PARTITIONS 2;
```

```
CREATE TABLE employees (  
    id INT NOT NULL,  
    firstname VARCHAR(30),  
    lastname VARCHAR(30),  
    datehired DATE NOT NULL DEFAULT '1999-01-01',  
    datefinished DATE NOT NULL DEFAULT '9999-12-31',  
    job_id INT NOT NULL,  
    resto_id INT NOT NULL  
)  
PARTITION BY RANGE (resto_id) (  
    PARTITION p0 VALUES LESS THAN (6),  
    PARTITION p1 VALUES LESS THAN (11),  
    PARTITION p2 VALUES LESS THAN (16),  
    PARTITION p3 VALUES LESS THAN (21),  
    PARTITION p4 VALUES LESS THAN (31),  
    PARTITION p5 VALUES LESS THAN MAXVALUE);
```



```
mysql> select * from sys.schema_unused_indexes;
+-----+-----+-----+
| object_schema | object_name      | index_name      |
+-----+-----+-----+
| employees     | People           | LCode           |
| employees     | Language         | LCode           |
+-----+-----+-----+
2 rows in set (0.01 sec)
```

```
update performance_schema.setup_consumers set enabled = 'yes' où name
= 'events_waits_current';
```

```
update performance_schema.setup_instruments set activé = 'yes' où name
= 'wait / io / table / sql / handler';
```

```
mysql> select * from schema_redundant_indexes\G
***** 1. row *****
      table_schema: employees
      table_name: Language
      redundant_index_name: LCode
      redundant_index_columns: LCode
      redundant_index_non_unique: 1
      dominant_index_name: PRIMARY
      dominant_index_columns: LCode,Language
      dominant_index_non_unique: 0
      subpart_exists: 0
      sql_drop_index: ALTER TABLE `employees`.`language` DROP
INDEX `LCode`
1 row in set (0.00 sec)
```

```

mysql> select * from employees.language where lang = 'E';
56b204785fe7a5befa67b135c58gt655

746 rows in set (0.00 sec)

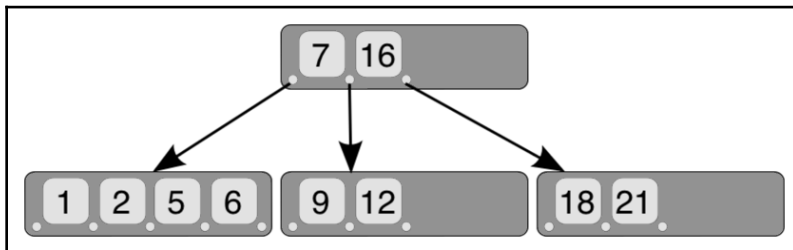
mysql> select * from statements_with_full_table_scans\G
***** 1. row *****
      query: select * from employees.language where lang = ?
         db: employees
         exec_count: 1
         total_latency: 533.34 us
         no_index_used_count: 1
no_good_index_used_count: 0
         no_index_used_pct: 100
           rows_sent: 1743
           rows_examined: 946
           rows_sent_avg: 642
           rows_examined_avg: 946
         first_seen: 2018-02-05 20:44:33
         last_seen: 2018-02-05 20:44:33
         digest: ab627af0817666c891221bar39f24231

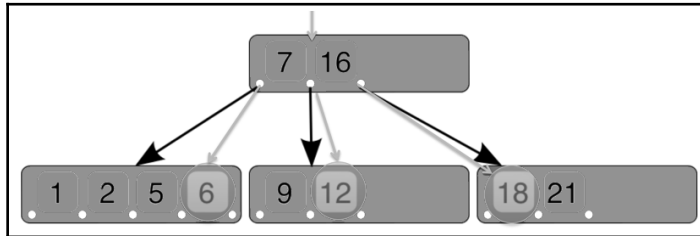
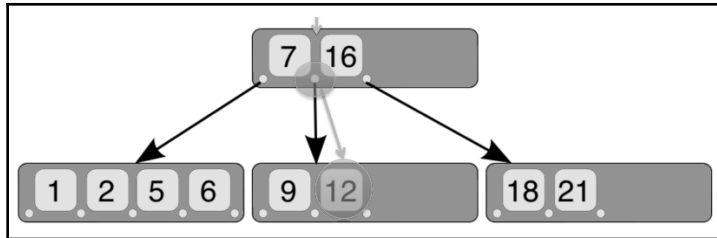
```

```

mysql> explain select * from employees.language where lang = 'E'\G
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: language
      type: ALL
possible_keys: NULL
      key: NULL
      key_len: NULL
      ref: NULL
      rows: 946
      Extra: Using where

```





```

SHOW CREATE TABLE employees (
  id INT NOT NULL,
  firstname VARCHAR(30),
  lastname VARCHAR(30),
  datehired DATE NOT NULL DEFAULT '1999-01-01',
  datefinished DATE NOT NULL DEFAULT '9999-12-31',
  job_id INT NOT NULL,
  resto_id INT NOT NULL
  Primary key (id),
  Key job_id (job_id) )as emp;

```

```
mysql> explain select * from employees where ID = 1;
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| table | type | possible_keys | key      | key_len | ref      | rows | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+
| emp   | const | PRIMARY       | PRIMARY | 4        | const   | 1    |      |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

```
mysql> explain select * from employees where job_id = 33;
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| table | type | possible_keys | key      | key_len | ref      | rows | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+
| emp   | ref  | job_id       | job_id  | 3        | const   | 563 | Using where |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

```
SHOW CREATE TABLE employees (  
  id INT NOT NULL,  
  firstname VARCHAR(30),  
  lastname VARCHAR(30),  
  datehired DATE NOT NULL DEFAULT '1999-01-01',  
  datefinished DATE NOT NULL DEFAULT '9999-12-31',  
  job_id char(3) NOT NULL,  
  resto_id INT NOT NULL  
  Primary key (id),  
  Key job_id (job_id) )as emp;
```

```
mysql> explain select * from employees where job_id = 33 \G  
***** 1. row *****  
  
table: employees  
type: ref  
possible_keys: combine  
key: combine  
key_len: 3 <--- Use the first column  
ref: const  
rows: 563  
*****
```

```
explain select * from employees where resto_id = 33 and lastname like
'%Tremb%'\G

***** 1. row *****

table: employees

type: ALL

possible_keys: NULL

key: NULL

key_len: NULL

ref: NULL

rows: 1023

*****
```

```
SHOW CREATE TABLE employees (
  id INT NOT NULL,
  firstname VARCHAR(30),
  lastname VARCHAR(30),
  datehired DATE NOT NULL DEFAULT '1999-01-01',
  datefinished DATE NOT NULL DEFAULT '9999-12-31',
  job_id char(3) NOT NULL,
  resto_id INT NOT NULL
  Primary key (id));

mysql> Select max(id), job_id, resto_id from employees where job_id =
7 and resto_id > 10 group by datehired;

...
      type: ALL
possible_keys: NULL
      key: NULL
      key_len: NULL
      ref: NULL
      rows: 5354
      Extra: Using where; Using temporary; Using filesort
```

```
mysql> Select max(id), job_id, resto_id from employees where job_id =
7 and resto_id > 10 group by datehired,resto_id\G
...
      type: range
possible_keys: combine
      key: combine
      key_len: 2
         ref: NULL
        rows: 2443
  Extra: Using where; Using index; Using temporary; Using
filesort
```

```
SHOW CREATE TABLE employees (
  id INT NOT NULL,
  firstname VARCHAR(30),
  lastname VARCHAR(30),
  datehired DATE NOT NULL DEFAULT '1999-01-01',
  datefinished DATE NOT NULL DEFAULT '9999-12-31',
  job_id char(3) NOT NULL,
  resto_id INT NOT NULL
  Primary key (id));

mysql> explain select * from employees where job_id = 100 order by
datehired\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: employees
         type: ALL
possible_keys: NULL
         key: NULL
      key_len: NULL
         ref: NULL
        rows: 345
  Extra: Using where; Using filesort
```

```
mysql> explain select * from employees where job_id = 100 order by
datehired\G
***** 1. row *****
id: 1
table: employees
type: ref
possible_keys: combine
key: combine
key_len: 3
ref: const
rows: 563
Extra: Using where;
```

- 2 types of temporary tables
 - MEMORY
 - On-disk

```
mysql> show global variables like 'tmp_table_size';
+-----+-----+
| Variable_name | Value      |
+-----+-----+
| tmp_table_size | 468435456 |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> show global status like 'created_tmp_disk_tables';
```

```
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| Created_tmp_disk_tables | 248551 |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> show global status like 'created_tmp_tables';
```

```
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| Created_tmp_tables     | 1618533 |
+-----+-----+
1 row in set (0.00 sec)
```

La formule suivante vous aidera à déterminer le pourcentage:

```
Tmp_disk_tables=((created_tmp_disk_tables*100/(created_tmp_tables+crea
ted_tmp_disk_tables))
= ((248551*100/(1618533 + 248551))
= 13.31%
```

```
SELECT * FROM employees WHERE ((employees_id=59235 AND employees_flag1 = 1) OR
(employees_division_employees_id=87459234 AND employees_flag2 = 1)) AND
(employees_id=59235 OR other_employees_id=59235) AND (employees_id <> 38465 AND
other_employees_id <> 38465)AND (employees_prop1 IS NOT NULL) AND
(employees_rangedate >= '2018-01-01') OR (employees_rangedate_min >
'2018-03-04' AND employees_rangedate_max < '2018-04-05') AND (NOT EXISTS (
SELECT 1 FROM employees_party WHERE fk_employeess_id = employeess.id AND
Important_id BETWEEN 1 and 4 AND diff_value BETWEEN 1 and 3));
```



```

SELECT * FROM employees
WHERE
    ((employees_id=59235 AND employees_flag1 = 1)
OR
    (employees_division_employees_id=87459234 AND employees_flag2 = 1))
AND
    (employees_id=59235 OR other_employees_id=59235)
AND
    (employees_id <> 38465 AND other_employees_id <> 38465)
AND
    (employees_prop1 IS NOT NULL)
AND
    (employees_rangedate >= '2018-01-01')
OR
    (employees_rangedate_min > '2018-03-04'
AND
    employees_rangedate_max < '2018-04-05')
AND
    (NOT EXISTS ( SELECT 1 FROM employees_party WHERE fk_employees_id =
employees.id AND
    Important_id BETWEEN 1 and 4 AND diff_value BETWEEN 1 and 3));

```

```

***** 1. row *****
id: 1
select_type: PRIMARY
table: employees
type: ref
possible_keys: idx_employees_id,idx_division_employees_id,idx_prop1
key: idx_employees_id
key_len: 4
rows: 917943
Extra: Using where
***** 2. row *****
id: 2
select_type: DEPENDENT SUBQUERY
table: employees_party
type: subquery
possible_keys: PRIMARY,fk_employees_id
key: PRIMARY
key_len: 4
rows: 1
Extra: Using where
2 rows in set (0.01 sec)

```

```

***** 1. row *****
id: 1
select_type: SIMPLE
table: ep
type: ref
possible_keys: idx_important_id_diff_value (see note 1)
key: idx_important_id_diff_value
key_len: 4
ref: const
rows: 43844
Extra: Using where
***** 2. row *****
id: 1
select_type: SIMPLE
table: t
type: ref
possible_keys: PRIMARY
key: PRIMARY
key_len: 4
ref: ts.fk_employees_id
rows: 5
Extra: Using where
2 rows in set (0.01 sec)

```

```

mysql> explain select empl.c from empl, empl e where empl.k=5 and e.empl.c=e.k order by e.k limit 5;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys | key | key_len | ref | rows | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | empl | ref | PRIMARY,k | k | 4 | const | 1 | Using temporary; Using filesort |
| 1 | SIMPLE | e | ref | k | k | 4 | empl.empl.c | 1 | Using where; Using index |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

```

mysql> explain select empl.c from empl, empl e where empl.k=5 and e.empl.c=e.k order by e.k limit 5;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys | key | key_len | ref | rows | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | empl | const | PRIMARY,k | k | 4 | const | 1 | |
| 1 | SIMPLE | e | ref | k | k | 4 | const | 1 | Using index |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

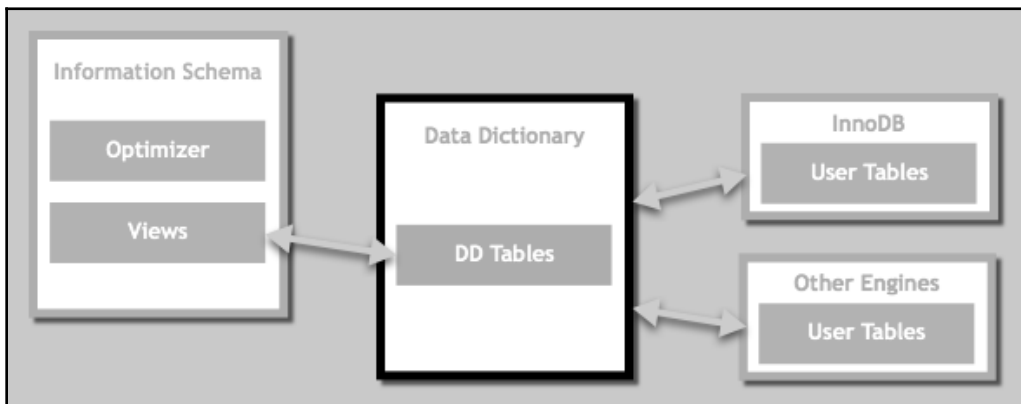
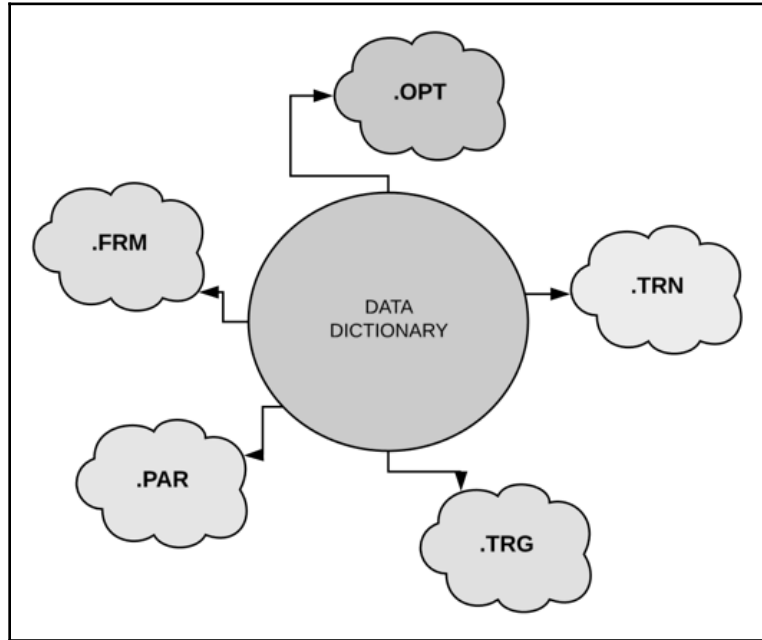
```

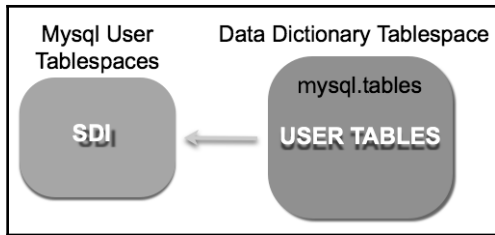
```
CREATE TABLE `metrics` (  
  `metric_timestamp` datetime NOT NULL,  
  `station_name` varchar(255) DEFAULT NULL,  
  `gas1_mtsperhour` int(11) NOT NULL,  
  `gas2_mtsperhour` int(11) NOT NULL,  
  `gas_qty` int(3) NOT NULL,  
  `rain_mm` decimal(5,2),  
  `temperature` int(5),  
  `humidity` int(5),  
  `barometric_pressure` decimal(10,2) NOT NULL,  
  `barometric_temperature` decimal(10,0) NOT NULL,  
  `lux` decimal(7,2),  
  `is_plugged` tinyint(1),  
  `battery_level` int(3),  
  KEY `metric_timestamp` (`metric_timestamp`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `partitioned_metrics` (  
  `metric_timestamp` datetime NOT NULL,  
  `station_name` varchar(255) DEFAULT NULL,  
  `gas1_mtsperhour` int(11) NOT NULL,  
  `gas2_mtsperhour` int(11) NOT NULL,  
  `gas_qty` int(3) NOT NULL,  
  `rain_mm` decimal(5,2),  
  `temperature` int(5),  
  `humidity` int(5),  
  `barometric_pressure` decimal(10,2) NOT NULL,  
  `barometric_temperature` decimal(10,0) NOT NULL,  
  `lux` decimal(7,2),  
  `is_plugged` tinyint(1),  
  `battery_level` int(3),  
  KEY `metric_timestamp` (`metric_timestamp`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `test`.`partitioned_metrics`  
  PARTITION BY RANGE (YEAR(metric_timestamp)) (  
    PARTITION to_metric_logs VALUES LESS THAN (2017),  
    PARTITION to_metric_prev_year_logs VALUES LESS THAN (2018),  
    PARTITION to_metric_current_logs VALUES LESS THAN (MAXVALUE)  
  );
```

Chapter 05: MySQL Data Dictionary in MySQL 8.0





```
mysql> select count(*), engine from information_schema.tables where table_schema = 'mysql' group by engine;
+-----+-----+
| count(*) | ENGINE|
+-----+-----+
|      2  | CSV  |
|     30  | InnoDB|
+-----+-----+
2 rows in set (0.00 sec)
```

Chapter 06: MySQL Server Settings

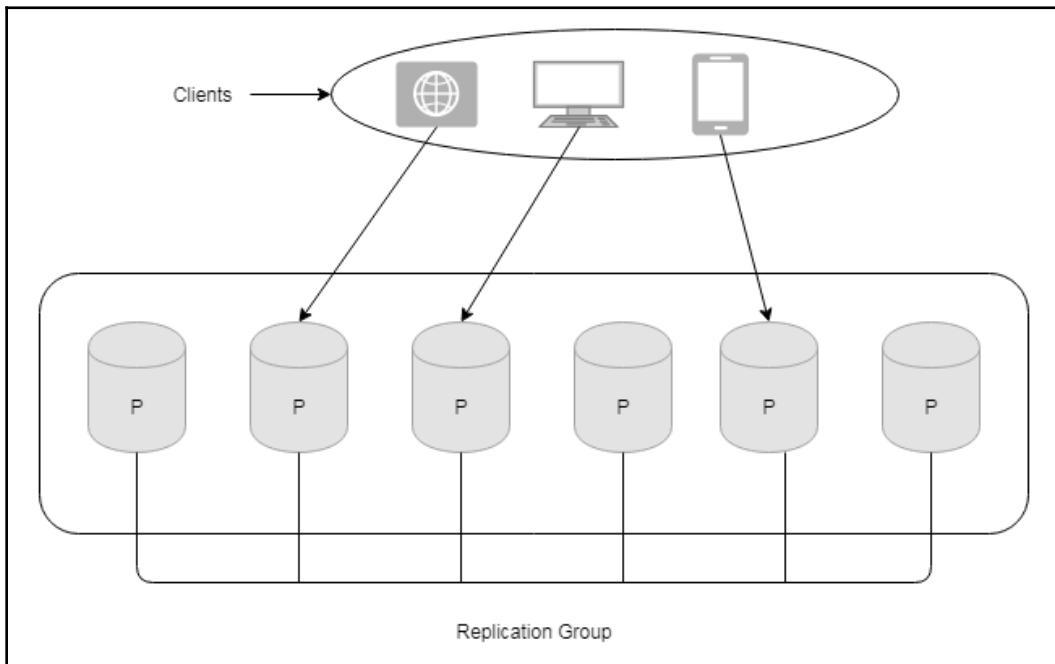
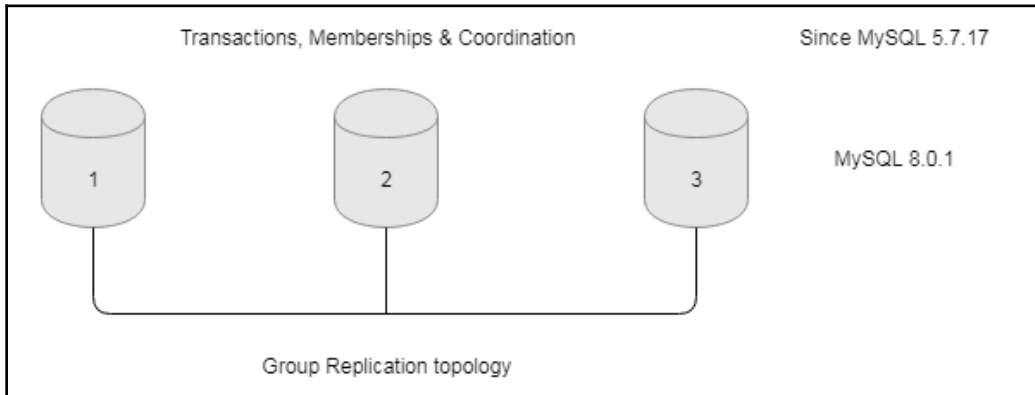
```
top - 8:45:35 up 53 days, 12:8, 1 user, load average: 25.35, 17.43, 26.03
```

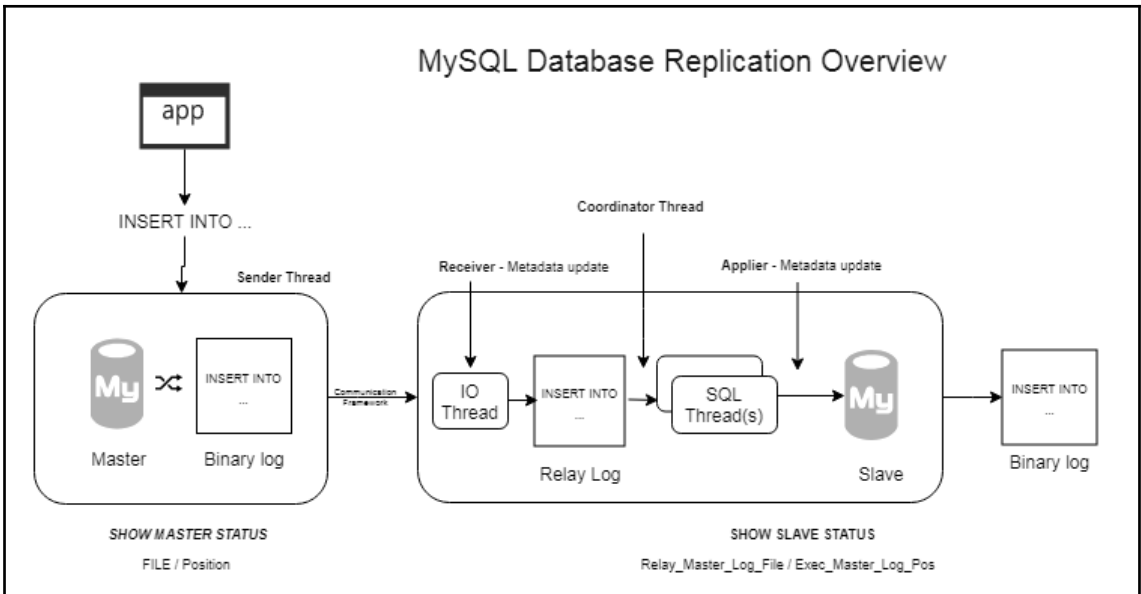
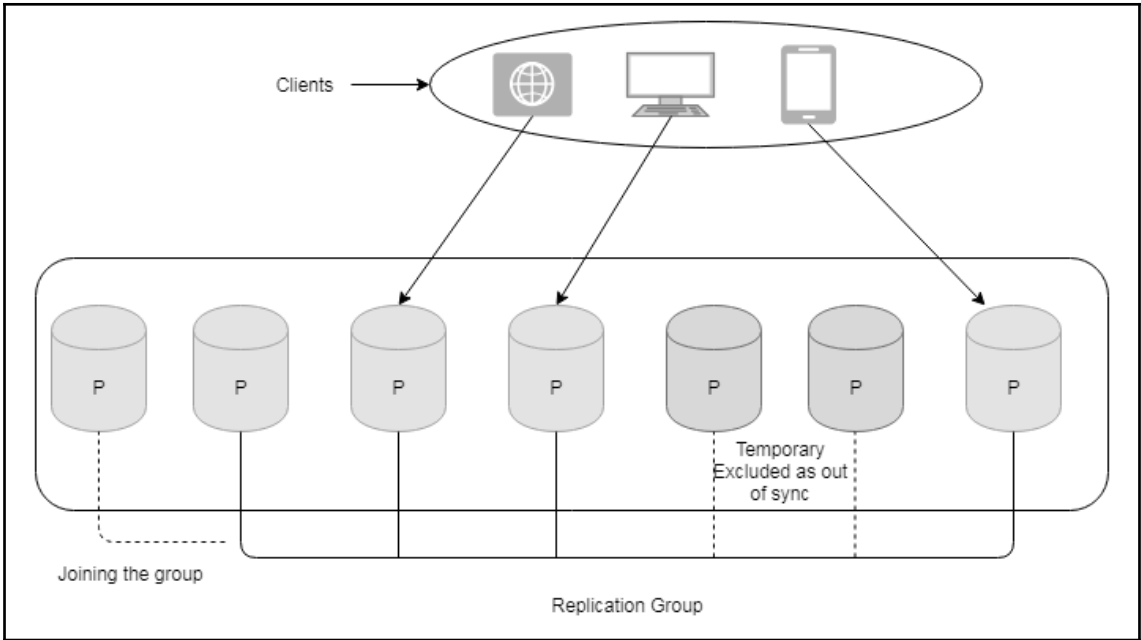
```
Tasks: 68 total, 6 running, 62 sleeping, 0 stopped, 0 zombie
```

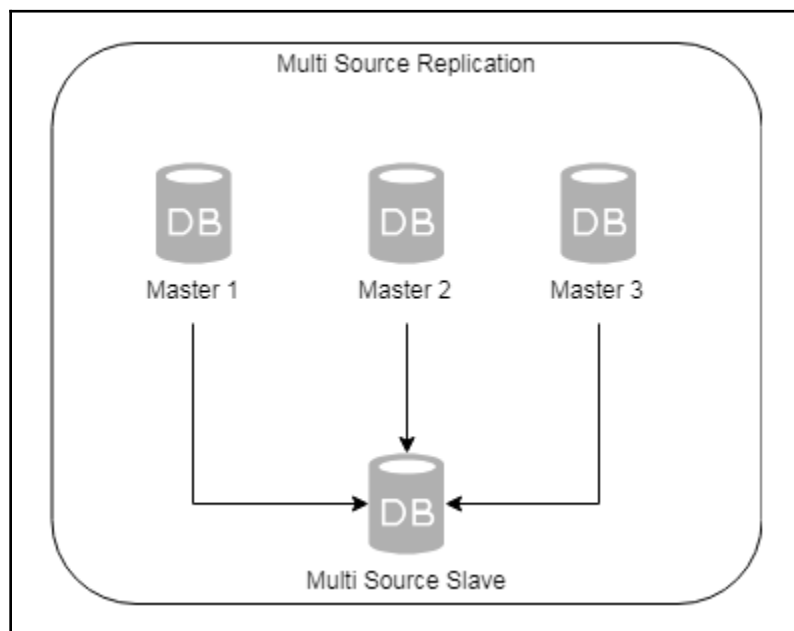
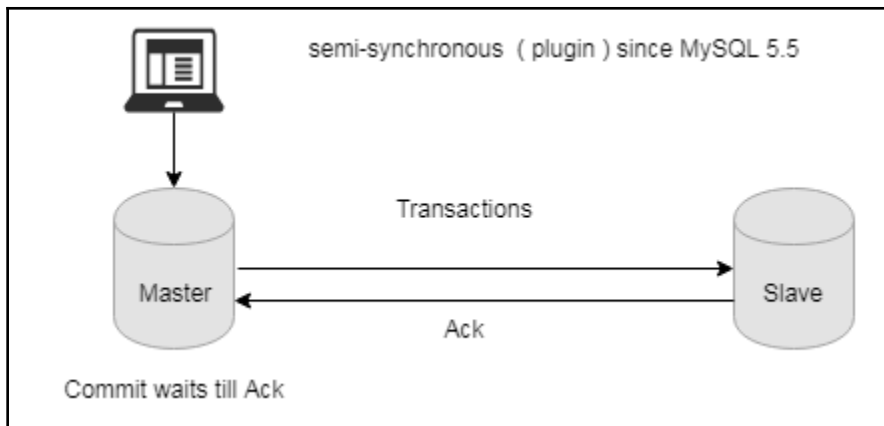
```
Cpu(s): 89.8%us, 0.4%sy, 0.0%ni, 12.5%id, 0.0%wa, 0.2%hi, 0.5%si, 0.0%st
```

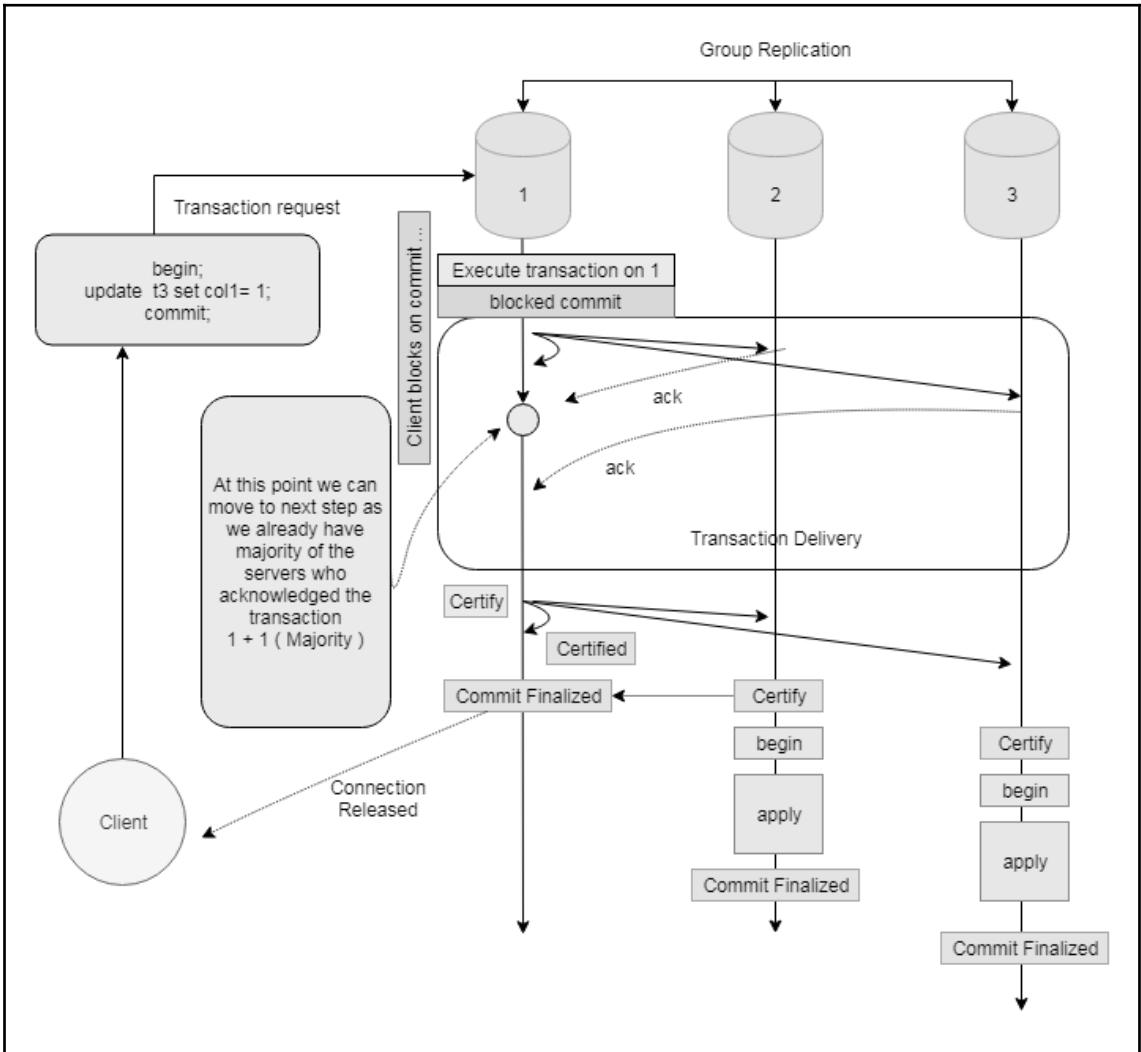
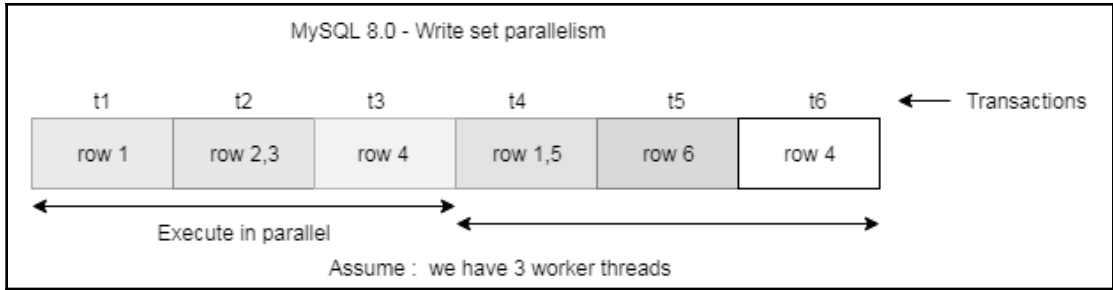
```
Mem: 2975930k total, 1118314k used, 1857616k free, 267407k buffers
```

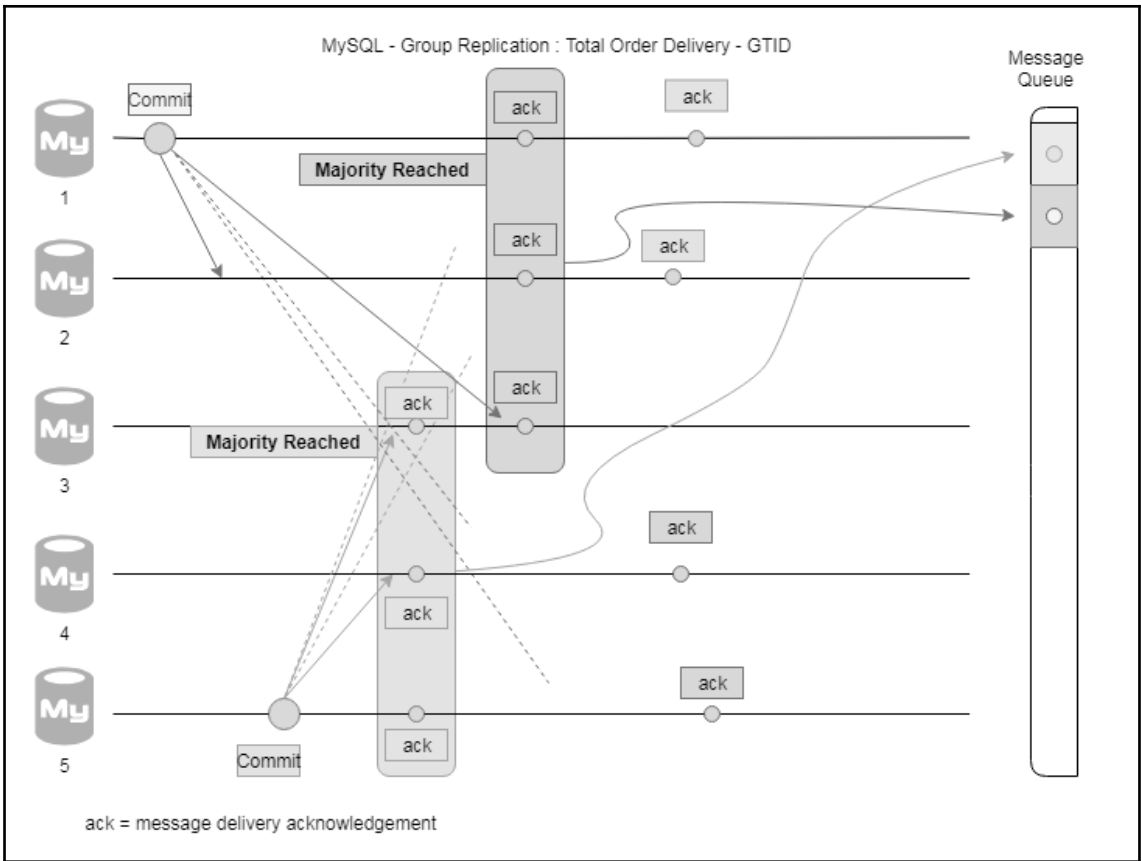
Chapter 07: Group Replication in MySQL 8.0

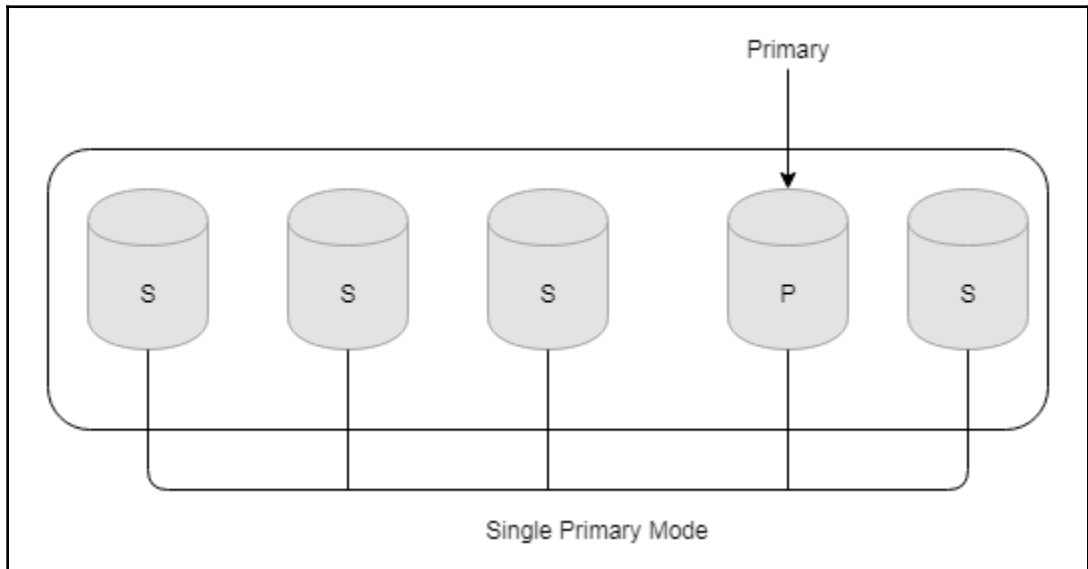
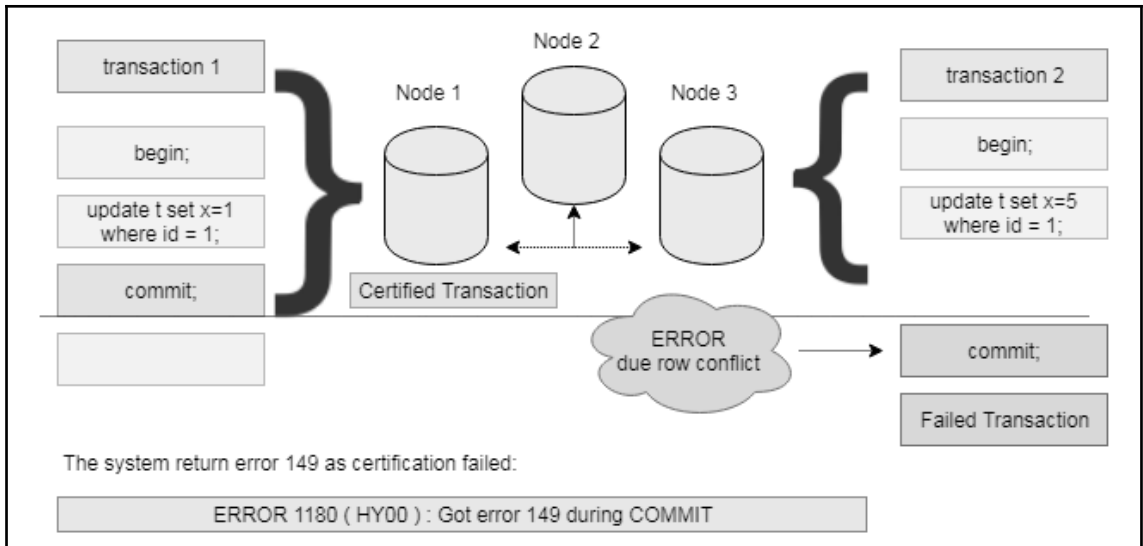


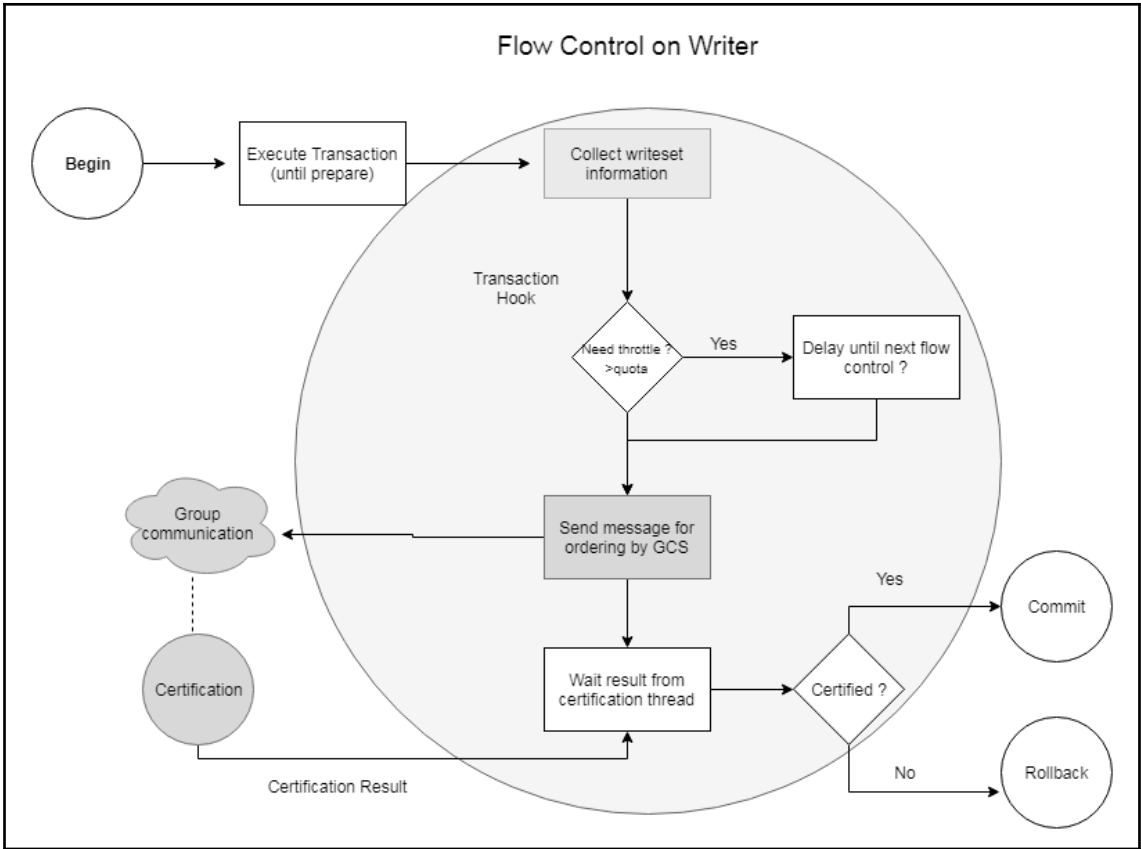




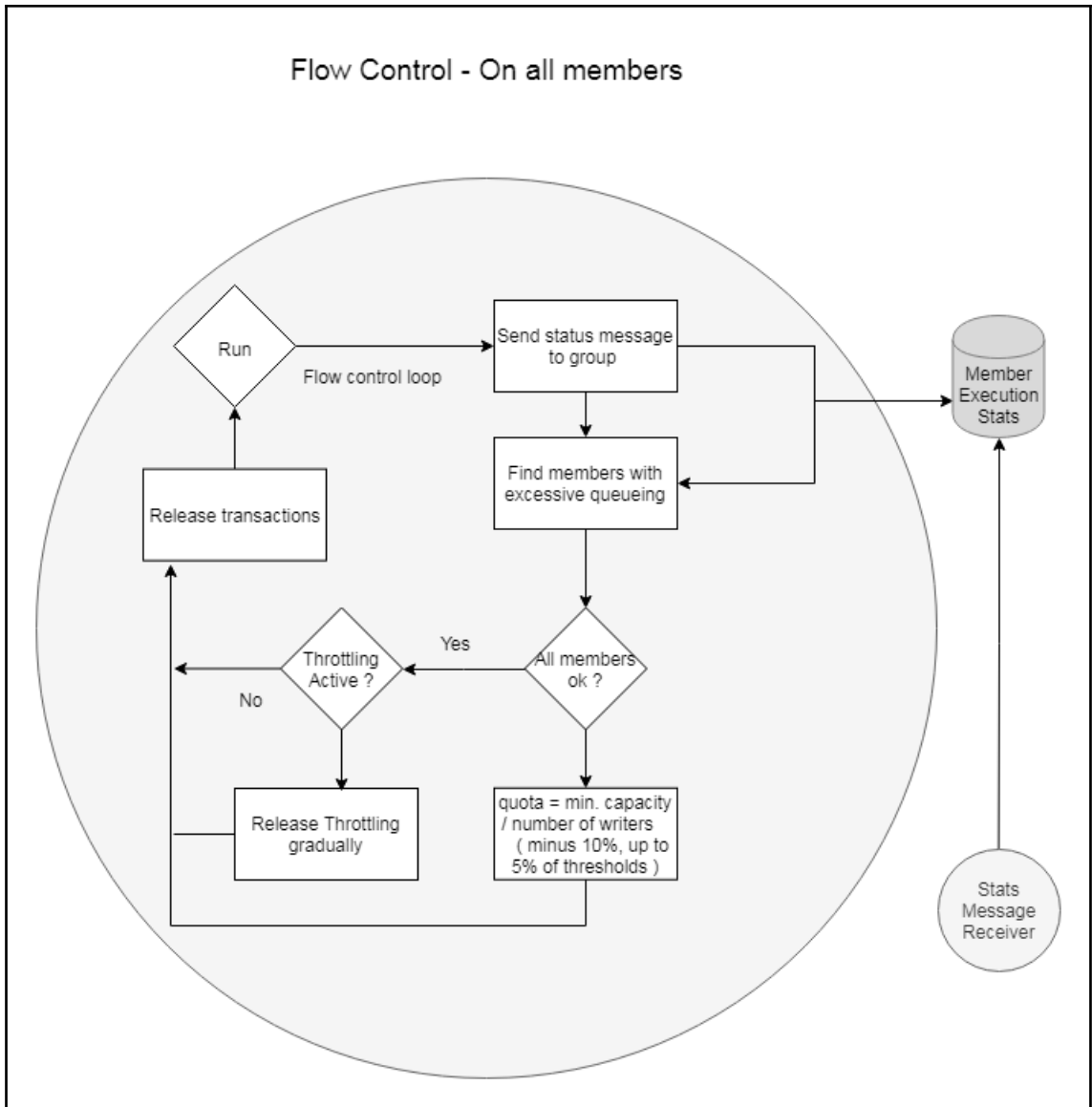




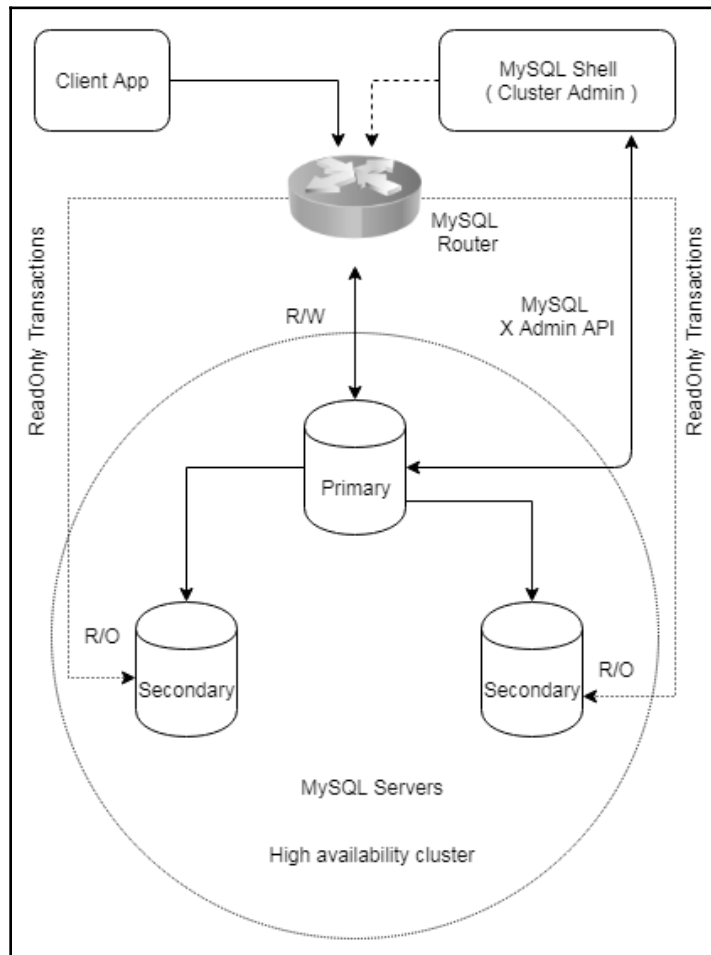




Flow Control - On all members



Chapter 08: InnoDB Cluster in MySQL 8.0



GLOBAL OBJECTS For “js” mode and “py” mode

dba	Used for InnoDB cluster administration.
mysql	Support for connecting to MySQL servers using the classic MySQL protocol.
mysqlx	Used to work with X Protocol sessions using the MySQL X DevAPI.
shell	Gives access to general purpose functions and properties.
sys	Gives access to system specific parameters.
util	Global object that groups miscellaneous tools like upgrade checker and JSON import.

Chapter 09: Monitoring Your Large Distributed Databases

The screenshot displays the 'Overview' page of a monitoring tool. At the top, there are four summary cards: '504 Total Servers', '0 Servers Down', '7570 Critical Alerts', and '4074 Warnings'. Below these is a table titled 'TOP 10 QUERIES (across all servers based on Total Time)'. The table has four columns: 'Query', 'Count', 'Total Time', and 'Average Latency'. The queries listed include 'SHOW FULL PROCESSLIST', 'SELECT * FROM (SELECT digest AS 'Digest', SCHEMA_NAME AS 'Db', 'digest_text' AS 'Query', 'count_star' AS 'Count', 'IFNULL('sum_time...', '0.000000')) AS 'sum_time...', 'SHOW GLOBAL VARIABLES', 'SHOW GLOBAL STATUS', 'SELECT * FROM (SELECT digest AS 'Digest', schema_name AS 'Db', digest_text AS 'Query', count_star AS 'Count', IFNULL(sum_timer_wait * 0.000000, '0.000000')) AS 'sum_timer_wait', 'SELECT UNIX_TIMESTAMP (DATE_SUB (NOW (), INTERVAL ? SQL_TSI_SECOND)) AS 'starttime'', 'SHOW SCHEMAS LIKE ?', 'SET NAMES ?', and 'SELECT * FROM `mysql`.`user`'. The footer contains 'Monog Ultimate 7.04 © 2017 Webyog Registered to Webyog' and links for 'Show Tooltip', 'Documentation', and 'Feedback'.

Query	Count	Total Time	Average Latency
SHOW FULL PROCESSLIST	61M	01:03:30.000	0
SHOW FULL PROCESSLIST	8M	01:00:16.000	0
SELECT * FROM (SELECT digest AS 'Digest', SCHEMA_NAME AS 'Db', 'digest_text' AS 'Query', 'count_star' AS 'Count', 'IFNULL('sum_time...', '0.000000')) AS 'sum_time...'	87K	51:36.000	00.036
SHOW GLOBAL VARIABLES	1M	10:12.000	00.001
SHOW GLOBAL STATUS	1M	07:52.000	0
SELECT * FROM (SELECT digest AS 'Digest', schema_name AS 'Db', digest_text AS 'Query', count_star AS 'Count', IFNULL(sum_timer_wait * 0.000000, '0.000000')) AS 'sum_timer_wait'	5K	05:23.000	00.064
SELECT UNIX_TIMESTAMP (DATE_SUB (NOW (), INTERVAL ? SQL_TSI_SECOND)) AS 'starttime'	1M	01:01.000	0
SHOW SCHEMAS LIKE ?	1M	53.000	0
SET NAMES ?	1M	50.000	0
SELECT * FROM `mysql`.`user`	1M	49.000	0

Monitors
Arg-Testi...
🔍

- General Info
- Security
- Excessive Privileges
- ▲ MySQL Logs
- **Connection History**
- Current Connections
- ▲ MyISAM Key Cache
- InnoDB Cache
- InnoDB Deadlocks
- ▲ InnoDB Logs
- ▲ InnoDB - Others
- Threads
- Table Cache & Locks
- Query Cache
- Index Usage

Monitors ⊕
Arg-Testing ☰

Attempts	53.00 (0.014/sec)	▮▮▮
Successful	53.00 (0.014/sec)	▮▮▮
Percentage of max allowed reached	🚩 <div style="display: flex; align-items: center; margin-left: 10px;"> <div style="width: 28.48%; height: 15px; background-color: #ccc; border: 1px solid #000;"></div> 28.48% </div>	▮▮▮
Refused	0.00	▮▮▮
Percentage of refused connections	🚩 <div style="display: flex; align-items: center; margin-left: 10px;"> <div style="width: 0.00%; height: 15px; background-color: #000; border: 1px solid #000;"></div> 0.00% </div>	▮▮▮
Terminated abruptly	1.00 (0.000/sec)	▮▮▮
Bytes received from all clients	2.18M (618.280/sec)	▮▮▮
Bytes sent to all clients	191.03M (52.941K/sec)	▮▮▮

Install your first Datadog Agent

The Datadog Agent collects metrics and events from your systems and apps. Install at least one Agent anywhere, even on your workstation.

You haven't installed any Agents yet. Let's do it now!

Mac OS X

Windows

Debian

Ubuntu

Amazon Linux

CentOS/RedHat

Fedora

SUSE

CoreOS

Docker

Kubernetes

Openshift

Chef

Puppet

Ansible

Cloud Foundry

From Source

Installing on Mac OS X

1 The Datadog Agent can be installed on OS X as easily as:

```
DD_API_KEY=278375898eee5adc594e6a49d58986d5 bash -c "$(curl -L https://raw.github
```

You can also [download the DMG package](#) and install it (you will have to add your `api_key`: `278375898eee5adc594e6a49d58986d5` in `/opt/datadog-agent/etc/datadog.yaml`).

To update an already installed agent from version 5 to 6:

```
DD_UPGRADE=true bash -c "$(curl -L https://raw.githubusercontent.com/DataDog/datadog-agent
```

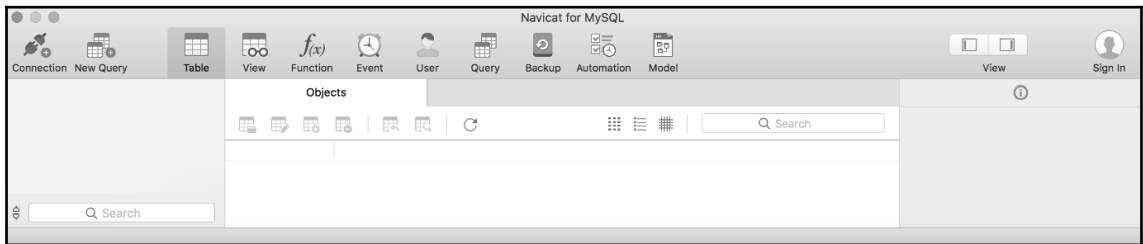
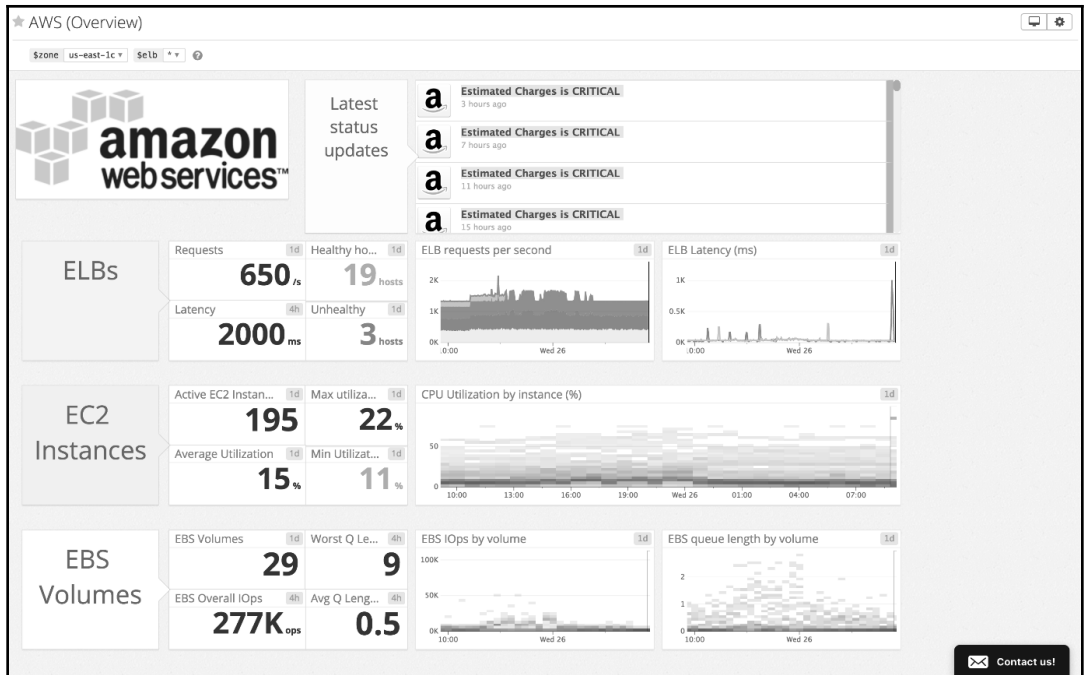
2 Manage the Agent:

Use the Datadog Agent app in the system tray, or the command line `datadog-agent` (located in `/usr/local/bin`).

Enable/disable integrations in `/opt/datadog-agent/etc/conf.d`.

3 By default, the agent will run at login (you can disable it via the system tray). If you want to run it at boot time, execute these commands:

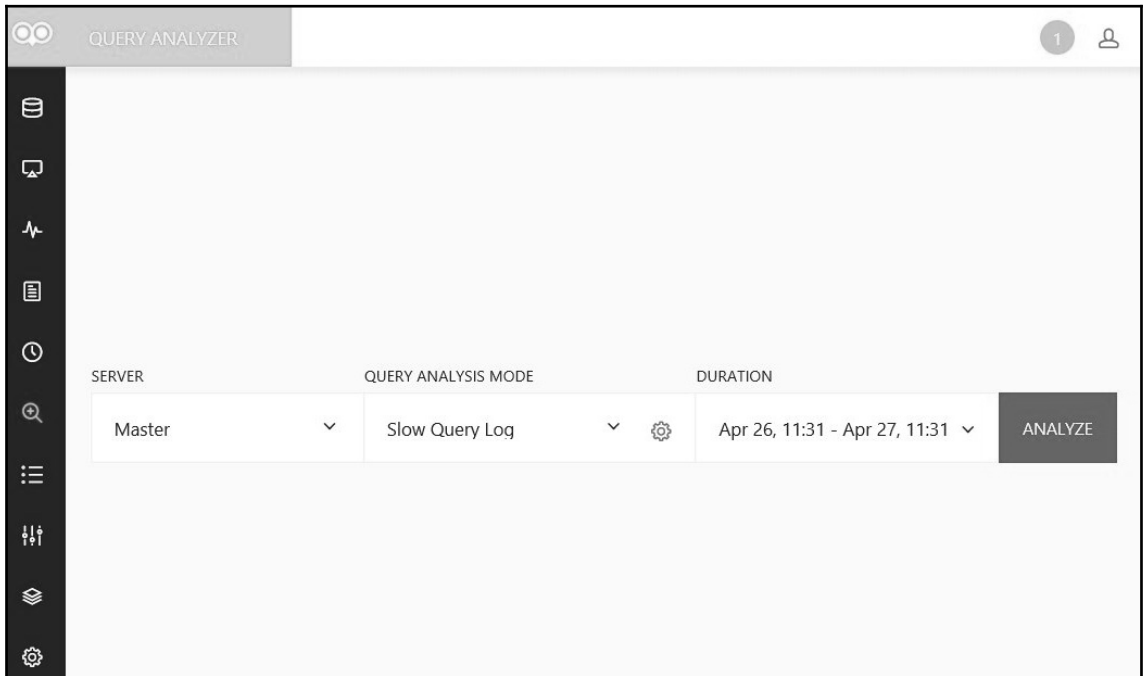
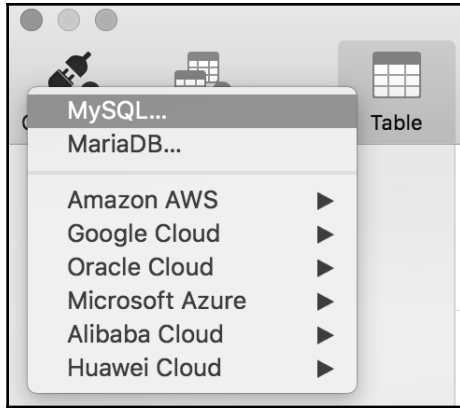
```
sudo cp '/opt/datadog-agent/etc/com.datadoghq.agent.plist.example' /Library/LaunchDaemons/
sudo launchctl load -w /Library/LaunchDaemons/com.datadoghq.agent.plist
```





```
Query Builder Query Editor
```

```
1 SELECT
2 orders.EmpNo,
3 cust.CustNo
4 FROM
5 customer AS cust
6 Inner Join orders ON orders.CustNo = cust.CustNo
7 WHERE
8 cust.CustNo > 1000
9 GROUP BY
10 cust.CustNo
11 ORDER BY
12 cust.CustNo ASC
13 LIMIT 1, 5
14
```

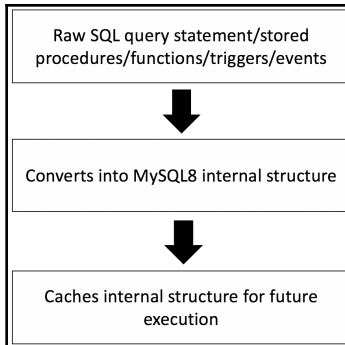


Chapter 11: Advanced MySQL Performance Tips and Techniques

```
Terminal — docker
mysql> SHOW VARIABLES LIKE 'have_query_cache';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_query_cache | NO   |
+-----+-----+
1 row in set (0.00 sec)

mysql> SHOW VARIABLES LIKE 'query_cache_size';
Empty set (0.00 sec)

mysql> _
```



```

Terminal — mysql
mysql> USE mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT Host, Db FROM db;
+-----+-----+
| Host      | Db      |
+-----+-----+
| localhost | performance_schema |
| localhost | sys     |
+-----+-----+
2 rows in set (0.00 sec)

mysql> EXPLAIN SELECT Host, Db FROM db;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | db    | NULL        | index | NULL          | User | 96      | NULL | 2 | 100.00 | Using index |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql> EXPLAIN SELECT Host, count(Db) FROM db GROUP BY Host;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | db    | NULL        | index | PRIMARY,User | PRIMARY | 468    | NULL | 2 | 100.00 | Using index |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql>

```

```

Terminal — mysql
mysql> USE mysql
Database changed
mysql> EXPLAIN SELECT Host, Db FROM db WHERE Select_priv = 'Y' LIMIT 1;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | db    | NULL        | ALL | NULL          | NULL | NULL    | NULL | 2 | 50.00 | Using where |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql> EXPLAIN SELECT Host, Db FROM db WHERE Select_priv = 'Y';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | db    | NULL        | ALL | NULL          | NULL | NULL    | NULL | 2 | 50.00 | Using where |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql>

```


userId	firstName	lastName
uid1	aaaaa	bbbb
uid2	abcd	dcab
uid3	bbbd	dabbb
uid4	efgh	ijk
.	.	.
.	.	.
uidn	zzzz	xyz

offers	
offerId<VARCHAR>	itemId<VARCHAR>
orders	
userId<VARCHAR>	itemId<MEDIUMINT>

offers	
offerId<VARCHAR>	itemId<VARCHAR> (indexed)
orders	
userId<VARCHAR>	itemId<VARCHAR>(indexed)

a11	a12	a1n
a21	a22	a2n
.
.
.
an1	an2	ann

a21	a22	a2n
-----	-----	-------	-----

count	a11	a12	a1n
	a21	a22	a2n

	an1	an2	ann

count	a11
	a21
	.
	.
	.
	an1

mobileNumber	firstName	lastName
mob1	mob1fName	mob1lName
mob2	mob2fName	mob2lName
.	.	.
.	.	.
mobn	mobnfName	mobnlName

mobileNumberOrEmail	firstName	lastName
mob1	mob1fName	mob1lName
email1	email1fName	email1lName
.	.	.
.	.	.
mobn	mobnfName	mobnlName

userId	userIdentifier	firstName	lastName
id1	mob1	mob1fName	mob1lName
id2	email1	email1fName	email1lName
id3	pager1	pager1fName	pager1lName
.	.	.	.
.	.	.	.
idn	mobn	mobnfName	mobnlName

userId	userIdentifier	firstName	lastName	gender
id1	mob1	mob1fName	mob1lName	male
id2	email1	email1fName	email1lName	female
id3	pager1	pager1fName	pager1lName	other
.
.
idn	mobn	mobnfName	mobnlName	male

Value	Index
NULL	NULL
male	0
female	1
other	2

