

```
source create_university.sql;  
source largeRelationsInsertFile.sql;
```

1)

2)

```
MariaDB [university]> show table status like 'student' \G;  
***** 1. row *****  
      Name: student  
      Engine: InnoDB  
      Version: 10  
      Row_format: Dynamic  
      Rows: 2000  
      Avg_row_length: 73  
      Data_length: 147456  
      Max_data_length: 0  
      Index_length: 81920  
      Data_free: 0  
      Auto_increment: NULL  
      Create_time: 2019-03-05 14:17:10  
      Update_time: 2019-03-05 14:17:44  
      Check_time: NULL  
      Collation: utf8mb4_general_ci  
      Checksum: NULL  
      Create_options:  
      Comment:  
      Max_index_length: 0  
      Temporary: N  
1 row in set (0.001 sec)  
  
ERROR: No query specified
```

3)

a)

```
MariaDB [university]> set profiling = 1;  
Query OK, 0 rows affected (0.000 sec)
```

b)

```
MariaDB [university]> select * from student where name = 'wood';
```

ID	name	dept_name	tot_cred
33791	Wood	Civil Eng.	92
39876	Wood	Accounting	14
62054	Wood	Mech. Eng.	13
96085	Wood	Accounting	70

```
4 rows in set (0.002 sec)
```

```
MariaDB [university]> show profiles;
```

Query_ID	Duration	Query
1	0.00007916	select * from student where name = 'wood'
2	0.03582835	select * from takes
3	0.00008154	select * from student where name = 'wood'
4	0.00006585	reset query cache
5	0.00208418	select * from student where name = 'wood'

```
5 rows in set (0.000 sec)
```

```

MariaDB [university]> show profile;
+-----+-----+
| Status                               | Duration |
+-----+-----+
| Starting                             | 0.000026 |
| Waiting for query cache lock         | 0.000004 |
| Init                                 | 0.000004 |
| Checking query cache for query       | 0.000040 |
| Checking permissions                 | 0.000007 |
| Opening tables                       | 0.000022 |
| After opening tables                 | 0.000006 |
| System lock                          | 0.000005 |
| Table lock                           | 0.000006 |
| Waiting for query cache lock         | 0.000017 |
| Init                                 | 0.000031 |
| Optimizing                           | 0.000014 |
| Statistics                           | 0.000018 |
| Preparing                            | 0.000019 |
| Executing                            | 0.000004 |
| Sending data                         | 0.001782 |
| End of update loop                  | 0.000008 |
| Query end                            | 0.000004 |
| Commit                               | 0.000006 |
| Closing tables                       | 0.000004 |
| Unlocking tables                     | 0.000003 |
| Closing tables                       | 0.000009 |
| Starting cleanup                     | 0.000003 |
| Freeing items                        | 0.000006 |
| Updating status                      | 0.000005 |
| Waiting for query cache lock         | 0.000003 |
| Updating status                      | 0.000014 |
| Waiting for query cache lock         | 0.000003 |
| Updating status                      | 0.000003 |
| Storing result in query cache        | 0.000005 |
| Reset for next command               | 0.000004 |
+-----+-----+
31 rows in set (0.001 sec)

```

c) Bottleneck is Sending data. As the contents of 'student' table are huge and therefore is time consuming to fetch everything into RAM.

d)

```
MariaDB [university]> show processlist;
```

Id	User	Host	db	Command	Time	State	Info	Progress
3	system user		NULL	Daemon	NULL	InnoDB purge worker	NULL	0.000
1	system user		NULL	Daemon	NULL	InnoDB purge coordinator	NULL	0.000
4	system user		NULL	Daemon	NULL	InnoDB purge worker	NULL	0.000
2	system user		NULL	Daemon	NULL	InnoDB purge worker	NULL	0.000
5	system user		NULL	Daemon	NULL	InnoDB shutdown handler	NULL	0.000
41	root	localhost	university_midsem	Sleep	656		NULL	0.000
42	root	localhost	university	Query	0	Init	show processlist	0.000

7 rows in set (0.014 sec)

4)

a)

```
MariaDB [university]> show storage engines;
```

Engine	Support	Comment	Transactions	XA	Savepoints
MRG_MyISAM	YES	Collection of identical MyISAM tables	NO	NO	NO
CSV	YES	Stores tables as CSV files	NO	NO	NO
MEMORY	YES	Hash based, stored in memory, useful for temporary tables	NO	NO	NO
MyISAM	YES	Non-transactional engine with good performance and small data footprint	NO	NO	NO
Aria	YES	Crash-safe tables with MyISAM heritage	NO	NO	NO
InnoDB	DEFAULT	Supports transactions, row-level locking, foreign keys and encryption for tables	YES	YES	YES
PERFORMANCE_SCHEMA	YES	Performance Schema	NO	NO	NO
SEQUENCE	YES	Generated tables filled with sequential values	YES	NO	YES

8 rows in set (0.020 sec)

Default one is clearly InnoDB.

```

MariaDB [university]> show storage engines\G;
***** 1. row *****
      Engine: MRG_MyISAM
      Support: YES
      Comment: Collection of identical MyISAM tables
Transactions: NO
          XA: NO
      Savepoints: NO
***** 2. row *****
      Engine: CSV
      Support: YES
      Comment: Stores tables as CSV files
Transactions: NO
          XA: NO
      Savepoints: NO
***** 3. row *****
      Engine: MEMORY
      Support: YES
      Comment: Hash based, stored in memory, useful for temporary tables
Transactions: NO
          XA: NO
      Savepoints: NO
***** 4. row *****
      Engine: MyISAM
      Support: YES
      Comment: Non-transactional engine with good performance and small data footprint
Transactions: NO
          XA: NO
      Savepoints: NO
***** 5. row *****
      Engine: Aria
      Support: YES
      Comment: Crash-safe tables with MyISAM heritage
Transactions: NO
          XA: NO
      Savepoints: NO
***** 6. row *****
      Engine: InnoDB
      Support: DEFAULT
      Comment: Supports transactions, row-level locking, foreign keys and encryption for tables
Transactions: YES
          XA: YES
      Savepoints: YES
***** 7. row *****
      Engine: PERFORMANCE_SCHEMA
      Support: YES
      Comment: Performance Schema
Transactions: NO
          XA: NO
      Savepoints: NO
***** 8. row *****
      Engine: SEQUENCE
      Support: YES
      Comment: Generated tables filled with sequential values
Transactions: YES
          XA: NO
      Savepoints: YES
8 rows in set (0.000 sec)

ERROR: No query specified

```

Thus hash index is supported by “MEMORY” engine.

b)

```

1
MariaDB [university]> create table takes_hash ENGINE MEMORY as (select * from takes);
Query OK, 30000 rows affected (0.111 sec)
Records: 30000 Duplicates: 0 Warnings: 0

```

```

MariaDB [university]> create index take_hash_gr on takes_hash(grade) USING HASH;
Query OK, 30000 rows affected (0.071 sec)
Records: 30000 Duplicates: 0 Warnings: 0

```

c)

```

MariaDB [university]> show index from takes_hash;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| takes_hash | 1 | take_hash_gr | 1 | grade | NULL | 9 | NULL | NULL | YES | HASH | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

```

```

MariaDB [university]> create index take_gr on takes(grade) USING HASH;
Query OK, 0 rows affected (0.451 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

```

MariaDB [university]> show index from takes;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| takes | 0 | PRIMARY | 1 | ID | A | 3814 | NULL | NULL | | BTREE | | |
| takes | 0 | PRIMARY | 2 | course_id | A | 30512 | NULL | NULL | | BTREE | | |
| takes | 0 | PRIMARY | 3 | sec_id | A | 30512 | NULL | NULL | | BTREE | | |
| takes | 0 | PRIMARY | 4 | semester | A | 30512 | NULL | NULL | | BTREE | | |
| takes | 0 | PRIMARY | 5 | year | A | 30512 | NULL | NULL | | BTREE | | |
| takes | 1 | course_id | 1 | course_id | A | 170 | NULL | NULL | | BTREE | | |
| takes | 1 | course_id | 2 | sec_id | A | 200 | NULL | NULL | | BTREE | | |
| takes | 1 | course_id | 3 | semester | A | 200 | NULL | NULL | | BTREE | | |
| takes | 1 | course_id | 4 | year | A | 200 | NULL | NULL | | BTREE | | |
| takes | 1 | take_gr | 1 | grade | A | 18 | NULL | NULL | YES | BTREE | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.001 sec)

```

Here I gave the type of 'take_gr' as 'HASH' but it got assigned as 'BTREE'.

Also all indices are of type 'BTREE'.

d)

```

MariaDB [university]> show profiles;
+-----+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+-----+
| 11 | 0.00009279 | create table takes_hash as (select * from takes) STORAGE ENGINE MEMORY |
| 12 | 0.00009178 | create table takes_hash as (select * from takes) STORAGE ENGINE = MEMORY |
| 13 | 0.00009599 | create table takes_hash STORAGE ENGINE = MEMORY as (select * from takes) |
| 14 | 0.00007464 | create table takes_hash STORAGE ENGINE MEMORY as (select * from takes) |
| 15 | 0.00115000 | describe takes |
| 16 | 0.00009743 | create table takes_hash STORAGE ENGINE=MEMORY as (select * from takes) |
| 17 | 0.11060519 | create table takes_hash ENGINE MEMORY as (select * from takes) |
| 18 | 0.07144564 | create index take_hash_gr on takes_hash(grade) USING HASH |
| 19 | 0.00116125 | describe takes_hash |
| 20 | 0.00040430 | show index from takes_hash |
| 21 | 0.45129783 | create index take_gr on takes(grade) USING HASH |
| 22 | 0.00038809 | show index from takes |
| 23 | 0.00700222 | reset query cache |
| 24 | 0.15349564 | select * from takes where grade like '%C%' |
| 25 | 0.01111117 | select * from takes_hash where grade like '%C%' |
+-----+-----+-----+-----+
15 rows in set (0.000 sec)

```

Here takes_hash had an hash index on grade whereas takes had an BTREE index on grade.

Now HASH index for this case turned out to be faster than BTREE as BTREE will have $O(\log N)$ time complexity whereas HASH for suitable bucket size has $O(1)$ complexity for fetching data.

e)

```

MariaDB [university]> create unique index take_grU on takes(grade);
ERROR 1062 (23000): Duplicate entry 'B+' for key 'take_grU'

```


Here the column is not unique (like 'B+' has duplicate entries) thus failing to create index.
f & g)

```
MariaDB [university]> create index id_courseID on takes(ID, course_id);
Query OK, 0 rows affected (0.407 sec)
Records: 0 Duplicates: 0 Warnings: 0

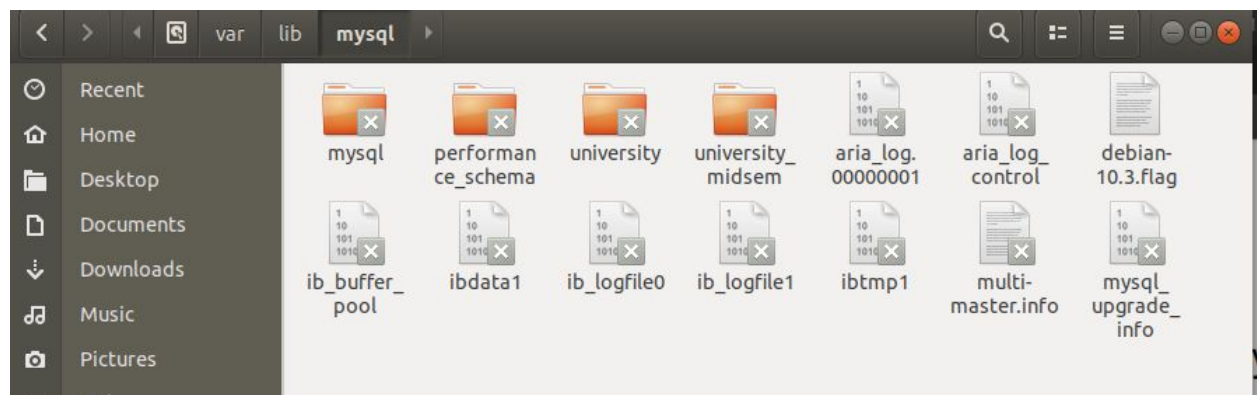
MariaDB [university]> show index from takes;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
takes	0	PRIMARY	1	ID	A	3814	NULL	NULL		BTREE		
takes	0	PRIMARY	2	course_id	A	30512	NULL	NULL		BTREE		
takes	0	PRIMARY	3	sec_id	A	30512	NULL	NULL		BTREE		
takes	0	PRIMARY	4	semester	A	30512	NULL	NULL		BTREE		
takes	0	PRIMARY	5	year	A	30512	NULL	NULL		BTREE		
takes	1	course_id	1	course_id	A	170	NULL	NULL		BTREE		
takes	1	course_id	2	sec_id	A	200	NULL	NULL		BTREE		
takes	1	course_id	3	semester	A	200	NULL	NULL		BTREE		
takes	1	course_id	4	year	A	200	NULL	NULL		BTREE		
takes	1	take_gr	1	grade	A	18	NULL	NULL	YES	BTREE		
takes	1	id_courseID	1	ID	A	4358	NULL	NULL		BTREE		
takes	1	id_courseID	2	course_id	A	30512	NULL	NULL		BTREE		

```
12 rows in set (0.001 sec)
```

5)

a)



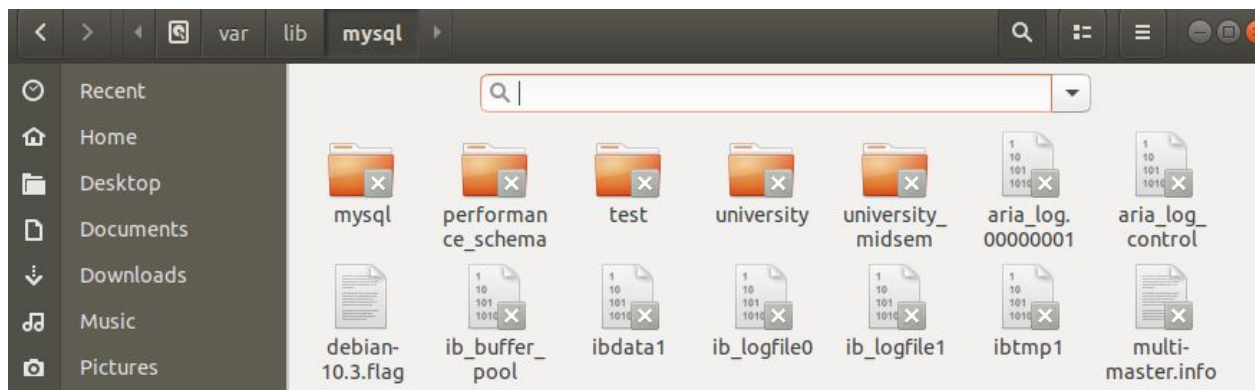
b)

```
MariaDB [university]> create database test;
Query OK, 1 row affected (0.016 sec)

MariaDB [university]> use test;
Database changed
MariaDB [test]> create table candidates(
    -> columns_id INT PRIMARY KEY,
    -> name VARCHAR(20));
Query OK, 0 rows affected (0.253 sec)

MariaDB [test]> insert into candidates values(1, 'tom');
Query OK, 1 row affected (0.068 sec)

MariaDB [test]> insert into candidates values(2, 'jerry');
Query OK, 1 row affected (0.054 sec)
```



```
root@dbms-VirtualBox:/var/lib/mysql/test# vim candidates.ibd
root@dbms-VirtualBox:/var/lib/mysql/test# vim candidates.frm
root@dbms-VirtualBox:/var/lib/mysql/test#
```

c)

```
MariaDB [test]> create table places(city varchar(20), country varchar(20), primary key (city, country));
Query OK, 0 rows affected (0.095 sec)
```

```
MariaDB [test]> insert into places values ('ludhiana', 'India');
Query OK, 1 row affected (0.016 sec)
```

```
MariaDB [test]> insert into places values ('jalandhar', 'India');
Query OK, 1 row affected (0.121 sec)
```

```
root@dbms-VirtualBox:/var/lib/mysql# cd test/
root@dbms-VirtualBox:/var/lib/mysql/test# ls
candidates.frm candidates.ibd db.opt places.frm places.ibd view_location.frm
```

d)

```
MariaDB [test]> create view view_location as (select A.columns_id, A.name, B.city, B.country from (select candidates.*, row_number() over (order by columns_id) as seqnum from candidates) as A join (select places.*, row_number() over (order by city) as seqnum from places) as B on A.seqnum = B.seqnum);
Query OK, 0 rows affected (0.014 sec)
```

```
MariaDB [test]> select * from view_location;
+-----+-----+-----+-----+
| columns_id | name | city      | country |
+-----+-----+-----+-----+
| 1          | tom  | jalandhar | India   |
| 2          | jerry | ludhiana  | India   |
| 3          | micky | Moga      | India   |
+-----+-----+-----+-----+
3 rows in set (0.011 sec)
```



```

root@dbms-VirtualBox:/var/lib/mysql/test# ls -l
total 208
-rw-rw---- 1 mysql mysql  973 Mar  5 15:28 candidates.frm
-rw-rw---- 1 mysql mysql 98304 Mar  5 15:29 candidates.ibd
-rw-rw---- 1 mysql mysql   65 Mar  5 15:28 db.opt
-rw-rw---- 1 mysql mysql  986 Mar  5 16:04 places.frm
-rw-rw---- 1 mysql mysql 98304 Mar  5 16:05 places.ibd
-rw-rw---- 1 mysql mysql   753 Mar  5 16:08 view_location.frm

```

Clearly size of the view is comparatively very small than table sizes.

e)

```

MariaDB [test]> delimiter //
MariaDB [test]> create trigger check_city BEFORE INSERT ON places
-> for each row begin
->     set @atemp = SUBSTRING(NEW.city, 1, 1);
->     set @atemp = UCASE(@atemp);
->     set @btemp = SUBSTRING(NEW.city, 2);
->     set @ctemp = CONCAT(@atemp, @btemp);
->     set NEW.city = @ctemp;
-> end //
Query OK, 0 rows affected (0.033 sec)

MariaDB [test]> delimiter ;
MariaDB [test]> insert into places values ('moga', 'India');
Query OK, 1 row affected (0.004 sec)

MariaDB [test]> select * from places;
+-----+-----+
| city      | country |
+-----+-----+
| jalandhar | India   |
| ludhiana  | India   |
| Moga      | India   |
+-----+-----+
3 rows in set (0.000 sec)

```

```

root@dbms-VirtualBox:/var/lib/mysql/test# ls -l
total 216
-rw-rw---- 1 mysql mysql  973 Mar  5 15:28 candidates.frm
-rw-rw---- 1 mysql mysql 98304 Mar  5 15:29 candidates.ibd
-rw-rw---- 1 mysql mysql   38 Mar  5 16:30 check_city.TRN
-rw-rw---- 1 mysql mysql   65 Mar  5 15:28 db.opt
-rw-rw---- 1 mysql mysql  986 Mar  5 16:04 places.frm
-rw-rw---- 1 mysql mysql 98304 Mar  5 16:30 places.ibd
-rw-rw---- 1 mysql mysql  479 Mar  5 16:30 places.TRG
-rw-rw---- 1 mysql mysql  753 Mar  5 16:08 view_location.frm

```

```

TYPE=TRIGGERNAME
trigger_table=places

```

```

"check_city.TRN" 2L, 38C

```

```

TYPE=TRIGGERS
triggers='CREATE DEFINER='root'@'localhost' trigger check_city BEFORE INSE
RT ON places\nfor each row begin \n    set @atemp = SUBSTRING(NEW.city, 1,
1);\n    set @atemp = UCASE(@atemp);\n    set @btemp = SUBSTRING(NEW.city
, 2);\n    set @ctemp = CONCAT(@atemp, @btemp);\n    set NEW.city = @ctemp
;\nend'
sql_modes=1411383296
definers='root@localhost'
client_cs_names='utf8'
connection_cl_names='utf8_general_ci'
db_cl_names='latin1_swedish_ci'
created=155178363431

```

This is places.TRG

f)

a , b and c:-

```
MariaDB [test]> show global variables like 'storage_engine';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| storage_engine | InnoDB |
+-----+-----+
1 row in set (0.001 sec)
```

```
MariaDB [test]> show global variables like '%buffer_size%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| aria_pagecache_buffer_size | 134217728 |
| aria_sort_buffer_size | 268434432 |
| bulk_insert_buffer_size | 16777216 |
| innodb_log_buffer_size | 8388608 |
| innodb_sort_buffer_size | 1048576 |
| join_buffer_size | 262144 |
| key_buffer_size | 134217728 |
| mrr_buffer_size | 262144 |
| myisam_sort_buffer_size | 536870912 |
| preload_buffer_size | 32768 |
| read_buffer_size | 2097152 |
| read_rnd_buffer_size | 1048576 |
| sort_buffer_size | 4194304 |
+-----+-----+
13 rows in set (0.001 sec)
```

```
MariaDB [university]> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.00013871 | reset query cache |
| 2 | 0.00030185 | set read_buffer_size = (select CAST((@read_buffer_size/16) AS INT)) |
| 3 | 0.03552815 | select * from takes |
| 4 | 0.00011123 | reset query cache |
| 5 | 0.00017499 | set read_buffer_size = (select CAST((@read_buffer_size*16) AS INT)) |
| 6 | 0.01824848 | select * from takes |
+-----+-----+-----+
6 rows in set (0.000 sec)
```

Thus multiplying by a factor 16 decreases the speed by half.