Cairo University - Faculty of Engineering

Computer Engineering Department

Advanced Database Systems

Project Phase Two

Mohamed Shawky Zaky
SEC:2, BN:15

Remonda Talaat Eskarous
SEC:1, BN:19

Mohamed Ahmed Mohamed Ahmed
SEC:2, BN:10

Mohamed Ramzy Helmy
SEC:2, BN:13

Contents

List of Figures

1 Query Statistics

1.1 Query 1

1.1.1 Execution Plan Before Optimization

1.1.2 Execution Plan After Optimization

1.1.3 Parallel Query Processing

1.2 Query 2

1.2.1 Execution Plan Before Optimization

1.2.2 Execution Plan After Optimization

1.2.3 Parallel Query Processing

1.3 Query 3

1.3.1 Execution Plan Before Optimization

1.3.2 Execution Plan After Optimization

1.3.3 Parallel Query Processing

1.4 Query 4

1.4.1 Execution Plan Before Optimization

1.4.2 Execution Plan After Optimization

1.4.3 Parallel Query Processing

1.5 Query 5

1.5.1 Execution Plan Before Optimization

1.5.2 Execution Plan After Optimization

1.5.3 Parallel Query Processing

2 Optimization Details

In this section, we show the optimization details done through this work. We discuss the statistics of the new database and the schema changes. Moreover, other optimization techniques related to query, indexes and memory are discussed, as well.

2.1 New Database Statistics

In this subsection, we show the new database statistics after optimization. The record count is extracted from the database filled with 100000 records per table. Other filling sizes are considered through the analysis like 10000 and 1000000.

Table Name	Row Count	Main Key	Indexes	FK
Disaster	100000	YES	4	2
Causes	100000	YES	1	1
Precautions	100000	YES	1	1
Incident	100000	YES	4	3
Descriptions	100000	YES	1	1
Casualty	25000	YES	1	0
Government_Representative	25000	YES	1	0
Govn_Rep_Credentials	25000	YES	1	1
Citizen	25000	YES	1	0
Citizen_Credentials	25000	YES	1	1
Criminal	25000	YES	1	0
Report	100000	YES	5	4
Report_Content	100000	YES	1	1
Casualty_Incident	100000	YES *(Composite)*	3	2

Table Name	Identity Column	Max Row Size (Bytes)
Disaster	YES	52
Causes	YES	65538
Precautions	YES	65538
Incident	YES	120
Descriptions	YES	65538
Casualty	YES	105
Government_Representative	YES	116
Govn_Rep_Credentials	YES	103
Citizen	YES	116
Citizen_Credentials	YES	103
Criminal	YES	106
Report	YES	23
Report_Content	YES	65538
Casualty_Incident	NO	6

2.2 Schema Optimization

The following database schema shows the optimizations over the old schema. The schema optimizations can be summarized as follows :

1. **Denormalization** of *Person* table with all persons types. This is mainly because no duplicates between persons types. For example, no person can be a casualty and a criminal at the same time. So, in order to avoid redundant joins, the *Person* relation is merged into the four child relations.

2. **Normalization** *(Vertical Partitioning)* of variable-size data. The access of fixed-size records is faster than that of variable-size data, as the *DBMS* don't require to pre-calculate the record size. That's why, the variable-size data like *text*, *usernames* and *passwords* are separated into separate relations. One more reason for doing so is that the data in these fields aren't accessed frequently, so they better be separate from other frequently-accessed data.

3. **Minimization** of the data types based on semantic and statistical *heuristics*. The data types of different fields are reduced to the minimum possible size, in order to ease their read and write to the disk. For example, all *primary keys* are reduced to `MEDIUMINT`, because the table size is at most 1000000. Moreover, Some fields that are just limited to range from 1 to 10 are reduced to 4 bits instead of `INT`.

4. **Conversion** of variable-size data to fixed-size data. As the fixed-size record are faster to access and transfer, so each *VarChar* is converted into *Char*. This, however, can increase the storage space, as *Char* allocates the target bytes whatever they are all used or not.

5. **Usage** of `NOT NULL`, whenever possible. If the field isn't marked as `NOT NULL`, it allocates extra bits to check whether the field is *NULL* or not.

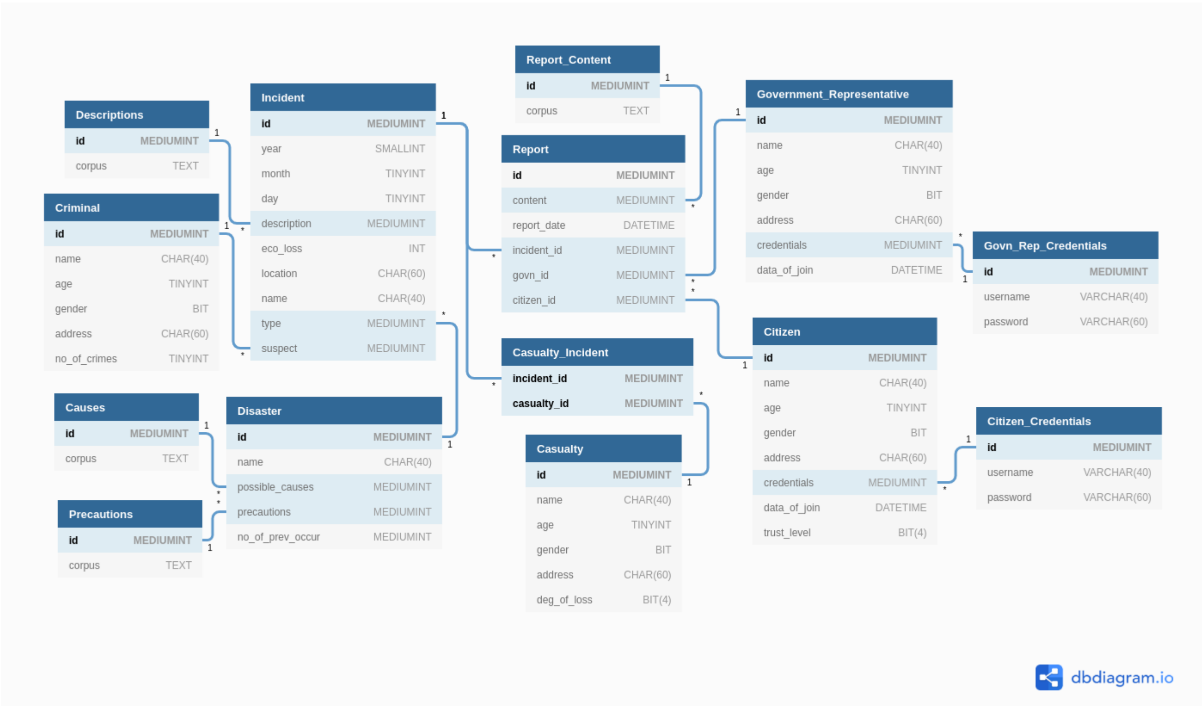

Figure 1: New Optimized Database Schema.

2.3 Memory Optimization

2.4 Index Tuning

2.5 Query Optimization

2.5.1 Query 1

2.5.2 Query 2

2.5.3 Query 3

2.5.4 Query 4

2.5.5 Query 5

3 Validation Details

3.1 Time and Space Analysis

In this subsection, we evaluate both time and space improvements of each optimization on each query. We consider both before and after *disk cache*. Moreover, the space improvement is considering the **total size** of the transferred tables between memory and disk. Execution time is measured in *seconds*.

Query 1	Before Cache			After Cache		
	Time	Time %	Space %	Time	Time %	Space %
Initial Query	16.78	-	-	1.77	-	-
Index Opt.	-	-	-	-	-	-
Query Opt.	10.87	35%	25%	0.39	78%	25%
Schema Opt.	8.8	19%	99.8%	0.33	18%	99.8%
Memory Opt.	7.8	11%	-	0.3	9%	-

Query 2	Before Cache			After Cache		
	Time	Time %	Space %	Time	Time %	Space %
Initial Query	1535	-	-	1463	-	-
Index Opt.	9.49	99%	-	0.78	99.9%	-
Query Opt.	7.75	18%	-	0.68	13%	-
Schema Opt.	6.8	12%	99.8%	0.65	4%	99.8%
Memory Opt.	5.77	15%	-	0.65	0%	-

Query 3	Before Cache			After Cache		
	Time	Time %	Space %	Time	Time %	Space %
Initial Query	0.36	-	-	0.07	-	-
Index Opt.	0.23	36%	-	0.01	85.7%	-
Query Opt.	0.19	17%	-	0.01	0%	-
Schema Opt.	0.14	26%	99.8%	0	100%	99.8%
Memory Opt.	0.12	14%	-	0	0%	-

Query 4	Before Cache			After Cache		
	Time	Time %	Space %	Time	Time %	Space %
Initial Query	10.41	-	-	0.27	-	-
Index Opt.	-	-	-	-	-	-
Query Opt.	-	-	-	-	-	-
Schema Opt.	6.37	39%	99.8%	0.15	44%	99.8%
Memory Opt.	6.01	5.5%	-	0.13	13%	-

Query 5	Before Cache			After Cache		
	Time	Time %	Space %	Time	Time %	Space %
Initial Query	6.14	-	-	0.33	-	-
Index Opt.	-	-	-	-	-	-
Query Opt.	4.39	28.5%	-	0.26	21%	-
Schema Opt.	2.94	33%	99.85%	0.21	19%	99.85%
Memory Opt.	2.34	20%	-	0.2	4.7%	-

3.2 Database Size Effect

The following plots show the effect of increasing database sizes on the execution time of our 5 queries. We consider both before and after *disk cache*.

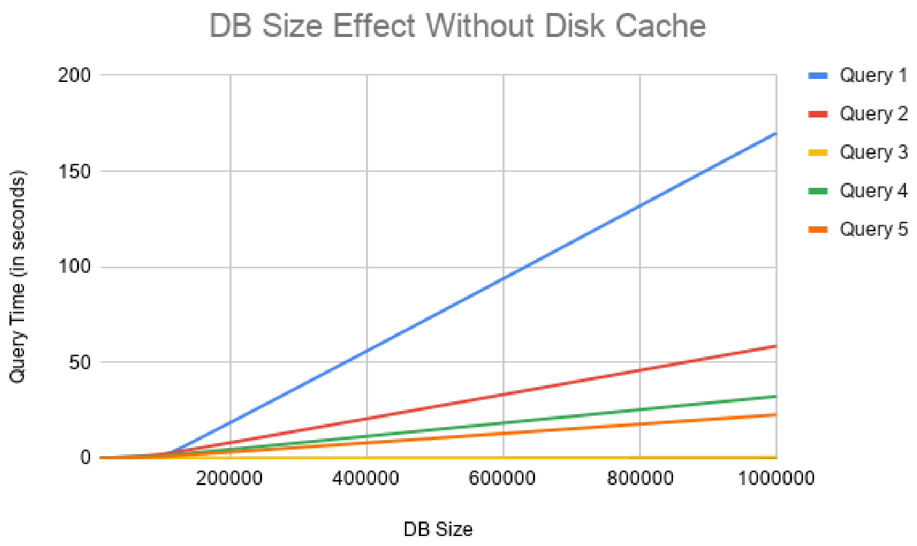

Figure 2: Database Size Effect Without OS (Disk) Cache.

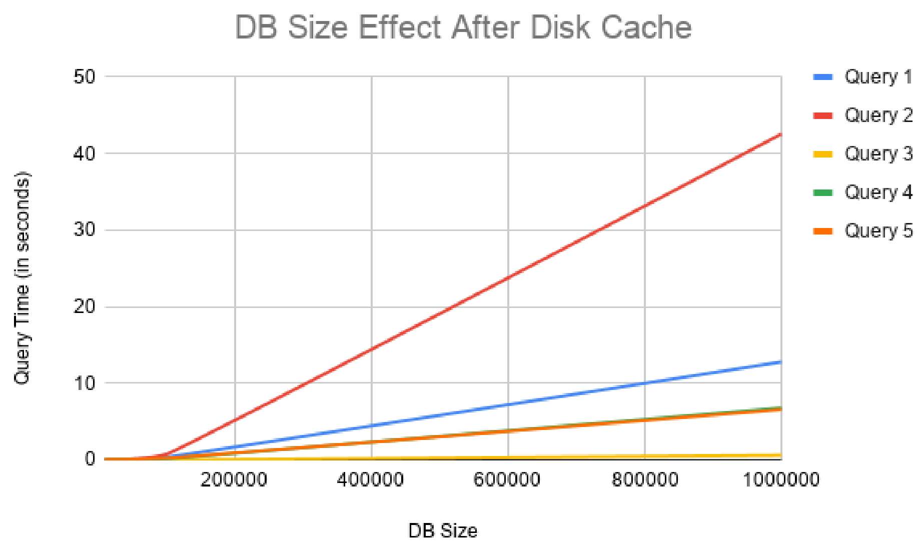

Figure 3: Database Size Effect After OS (Disk) Cache.

3.3 Optimized SQL vs. NoSQL

3.4 Hardware Effect

4 Final Remarks