CS 587 Database Implementation Database Benchmarking Project (Part 1)

(Punam Rani Pal, Weiwei Chen)

Github Link: https://github.com/waiwaixiaochen/CS587-Database-Benchmarking

Description:

In this project, we have followed the Wisconsin benchmark project that have been applied on the different relational database management systems to measure the performance on various sets of queries and predict its efficiency. This project is a good practice for us to increase our understanding of Database Performance. We can get some experience with a relational-style data management system of our choice as well as programming experience.

we accomplished two tasks for the first part of the project:

- **Data generation:** For the data generation, we used Python script. The data model and the schema is based on the Wisconsin Benchmark. We generated the data directly into a system.
- **System selection:** We selected MySQL for database implementation and running queries.

System used:

For this project MySQL has been chosen because we got an opportunity to use postgresql in the database introductory class (CS586) and now we planned to experience a different system since MySQL is a stable and reliable open-source database, it also features a distinct storage-engine framework that facilitates system management to configure the MySQL database server for a flawless performance.

Following are the prominent benefits:

- User friendly
- On demand high scalability
- Data security
- High performance
- Supports transactional feature
- Ease of connection with different systems

Furthermore, it also ensures high speed and unique memory caches for enhanced performance on many cloud databases.

Data sample loaded:

We followed the relational schema specified on the Wisconsin paper. There are three tables in the database: OENKTUP of size 1000, TENKTUP1 and TENKTUP2 of size 10000 respectively. All the tables share the same schema given below:

```
CREATE TABLE TENKTUP1
( unique1
                  integer NOT NULL,
   unique2
                  integer NOT NULL PRIMARY KEY,
   two
                  integer NOT NULL,
   four
                  integer NOT NULL,
                  integer NOT NULL,
   ten
                  integer NOT NULL,
   twenty
   hundred
                  integer NOT NULL,
                  integer NOT NULL,
   thousand
   twothous
                  integer NOT NULL,
   fivethous
                  integer NOT NULL,
   tenthous
                  integer NOT NULL,
   odd100
                  integer NOT NULL,
   even100
                  integer NOT NULL,
   stringu1
                  char (52) NOT NULL,
   stringu2
                  char (52) NOT NULL,
   string4
                  char (52) NOT NULL
)
```

Following are the constraints of each attribute:

Attribute Name	Range of Values Ord	der Comme	ent
unique1	0-(MAXTUPLES-1)	random	unique, random order
unique2	0-(MAXTUPLES-1)	sequential	unique, sequential
two	0-1	random	(unique1 mod 2)
four	0-3	random	(unique1 mod 4)
ten	0-9	random	(unique1 mod 10)
twenty	0-19	random	(unique1 mod 20
onePercent	0-99	random	(unique1 mod 100)
tenPercent	0-9	random	(unique1 mod 10)
twentyPercent	0 - 4	random	(unique1 mod 5)
fiftyPercent	0-1	random	(unique1 mod 2)
unique3	0-(MAXTUPLES-1)	random	unique1
evenOnePercent	0,2,4,,198	random	(onePercent * 2)
oddOnePercent	1,3,5,,199	random	(onePercent * 2)+1
stringu1	_	random	candidate key
stringu2	=	random	candidate key
string4	<u> </u>	cyclic	

Method for data generation:

- Established connection between MySQL server and the local machine Python editor.
- Populated records in each table by Python script.
- Exported the generated tables from MySQL Workbench as CSV files.

Examples of the generated tables:

Table ONEKTUP:

Α	В	C	D	E	F	G	Н	1	J	K	L	М	N	0	P	Q	R	
unique1	unique2	two	four	ten	twenty	onePercer	tenPercen	t twentyPero	fiftyPercen	unique3	evenOnePeo	ddOnePe	stringu1	stringu2	stringu4			
436	5	0	1	2	8	14 95	5	0	1	601	L 86	63	UQAAAA	AAAAAA	Ax AAAAxxxx	xxxxxxxxx	xxxxxxxx	XXXX
161		1	1	1	6	11 76		3	0	340	34	125	FGAAAAA	х ВААААА	Ax HHHHxxxx	xxxxxxxxxx	xxxxxxxx	cxxxx
300)	2	0	3	2	7 18		4	1	388	3 158	1	OLAAAAA	X CAAAAAA	Ax OOOOxxx	xxxxxxxxx	xxxxxxxx	XXXXX
754	l.	3	0	2	6	9 78	7	7 3	1	221	1 70	115	ADBAAAA	x DAAAAA	Ax VVVVxxxx	xxxxxxxxx	xxxxxxxx	xxxx
126	5	4	0	1	3	8 14	. 2	2 1	1	52	2 0	195	WEAAAA	A EAAAAAA	x AAAAxxxx	xxxxxxxx	xxxxxxxx	XXXX
37	,	5	0	2	4	13 91		0	1	35	76	7	LBAAAAA	x: FAAAAAA	x: HHHHxxxx	xxxxxxxxx	xxxxxxxx	(XXXX
245	5	6	1	2	8	10 39	(5 4	1	323	64	119	LJAAAAA	x GAAAAA	Ax OOOOxxx	xxxxxxxxx	xxxxxxxx	XXXX
428	3	7	0	1	2	16 47	7	7 0	0	332	2 34	79	MQAAAA	A HAAAAA	Ax VVVVxxxx	xxxxxxxx	xxxxxxxx	XXXX
903	3	8	0	1	1	0 13	(0	1	528	3 150	43	TIBAAAA	X IAAAAAA	xx AAAAxxxx	xxxxxxxxx	xxxxxxxx	xxxxx
452	2	9	1	0	8	0 72	5	0	0	491	184	67	KRAAAAA	x JAAAAAA	хэ ННННхххэ	xxxxxxxxxx	xxxxxxxx	CXXXX
758	3 1	10	0	1	1	19 66	7	7 0	0	299	180	165	EDBAAAA	x KAAAAAA	x OOOOxxx	xxxxxxxxx	xxxxxxxx	XXXXX
990) 1	11	0	0	6	5 76		3	0	194	38	1	CMBAAA	A LAAAAAA	x: VVVVxxxx	xxxxxxxxx	xxxxxxxx	XXXXX
47	1	12	0	2	9	11 71	3	3 1	0	492	2 50	55	VBAAAAA	x MAAAAA	A: AAAAxxxx	xxxxxxxxx	xxxxxxxx	XXXX
462	1	13	1	3	9	16 96	1	. 2	1	967	7 114	103	URAAAAA	X NAAAAA	Ax HHHHxxxx	xxxxxxxxxx	xxxxxxxx	(XXXX
			-	-	-													

Table TENKTUP1:

А	В	С	D	E	F	G	Н	l i	J		K	L	М	N	0	Р	Q	R	
unique1	unique2	two	four	ten	twenty	onePercen	tenPercen	nt twentyPe	rcfiftyPer	cen uni	que3	evenOnePe	oddOnePe	erstringu1	stringu2	stringu4			
436		0	1	2 8	3 14	95	5	5	0	1	601	86	63	UQAAAAA	AAAAAAA	AAAAxxxx	xxxxxxxxx	xxxxxxxx	XXXXX
161		1	1	1 6	5 11	. 76		4	3	0	340	34	125	FGAAAAA	BAAAAAA	HHHHXXXX	XXXXXXXXXX	XXXXXXXX	(XXXXX
300		2	0	3 2	2 7	18	4	4	4	1	388	158		OLAAAAA	CAAAAAA	0000xxx	xxxxxxxxx	xxxxxxxx	XXXXX
754		3	0	2 6	5 9	78	7	7	3	1	221	70	115	ADBAAAA:	DAAAAAA	VVVVxxxx	xxxxxxxxx	xxxxxxxx	xxxxx
126		4	0	1 3	3 8	14	2	2	1	1	52	0	195	WEAAAAA	EAAAAAA	AAAAxxxx	xxxxxxxxx	xxxxxxxx	xxxxx
37		5	0	2	1 13	91	9	9	0	1	35	76		7 LBAAAAA	FAAAAAA	HHHHXXXX	xxxxxxxxxx	XXXXXXXX	(XXXXX
245		6	1	2 8	3 10	39	6	5	4	1	323	64	119	LJAAAAAx	GAAAAAA	0000xxx	xxxxxxxxx	xxxxxxxx	xxxxx
428		7	0	1 2	2 16	47	7	7	0	0	332	34	79	MQAAAAA	HAAAAAA	VVVVxxxx	xxxxxxxxx	xxxxxxxx	xxxxx
903		8	0	1 1	L C	13	6	5	0	1	528	150	43	TIBAAAAx	(IAAAAAAx	AAAAxxxx	xxxxxxxxx	xxxxxxxx	xxxxx
452		9	1	0 8	3 0	72		5	0	0	491	184	67	KRAAAAA	JAAAAAA	HHHHXXXX	xxxxxxxxx	xxxxxxxx	(XXXXX)
758	1	.0	0	1 1	19	66	7	7	0	0	299	180	165	EDBAAAA	KAAAAAA	0000xxx	xxxxxxxxx	xxxxxxxx	xxxxx
990	1	.1	0	0 6	5 5	76		5	3	0	194	38		1 CMBAAAA	LAAAAAA	VVVVxxxx	xxxxxxxxx	xxxxxxxx	xxxxx
47	1	.2	0	2 9	11	71	3	3	1	0	492	50	55	VBAAAAA	MAAAAAA	AAAAxxxx	xxxxxxxxx	xxxxxxxx	xxxxx
462	1	.3	1	3 9	16	96	1	1	2	1	967	114	103	B URAAAAA	NAAAAAA	HHHHxxxx	xxxxxxxxxx	xxxxxxxx	(XXXX)

Table TENKTUP2:

A	В	C		D	E	F		G	Н	1	J	K		L	M	N	0	Р	Q	R
unique1	unique2	two	fc	our	ten	twenty	/	onePercen	tenPercer	t twentyPer	fiftyPercer	unique3	e	venOnePeo	ddOnePe	stringu1	stringu2	stringu4		
436		0	1		2	8	14	95	5	5 0	1	60	01	86	63	UQAAAA	AAAAAAA	x AAAAxxxx	xxxxxxxxx	xxxxxxxxxxx
161		1	1		1	6	11	76	- 2	1 3	0	34	40	34	125	FGAAAAA	x BAAAAAA	к ННННхххх	XXXXXXXXXX	(XXXXXXXXXXXXXX
300		2	0		3	2	7	18	4	1 4	1	38	88	158	1	OLAAAAA	x CAAAAAa	0000xxx	xxxxxxxxx	xxxxxxxxxxx
754		3	0		2	6	9	78		7 3	1	22	21	70	115	ADBAAAA	x DAAAAAA	x VVVV xxxx	xxxxxxxxx	xxxxxxxxxxx
126	1	4	0		1	3	8	14	- 2	2 1	1	5	52	0	195	WEAAAA	A EAAAAAA	CAAAAxxxx	xxxxxxxxx	xxxxxxxxxxx
37		5	0		2	4	13	91	9	0	1	3	35	76	7	LBAAAAA	x: FAAAAAA	HHHHXXXX	XXXXXXXXXX	(XXXXXXXXXXXXX
245		6	1		2	8	10	39	(5 4	1	32	23	64	119	LJAAAAA	x GAAAAAA	x0000xxx	xxxxxxxxx	xxxxxxxxxxx
428		7	0		1	2	16	47	- 17	7 0	0	33	32	34	79	MQAAAA	А НААААА	xVVVVxxxx	xxxxxxxxx	xxxxxxxxxxx
903		8	0		1	1	0	13	6	5 0	1	52	28	150	43	TIBAAAA	x IAAAAAA x	AAAAxxxx	xxxxxxxxx	xxxxxxxxxxx
452	1	9	1		0	8	0	72		5 0	0	49	91	184	67	KRAAAAA	x JAAAAAAx	HHHHXXXX	xxxxxxxxx	(XXXXXXXXXXXXX
758	1	0	0		1	1	19	66		7 0	0	29	99	180	165	EDBAAAA	x KAAAAAA	0000xxx	xxxxxxxxx	xxxxxxxxxxx
990	1	1	0		0	6	5	76		5 3	0	19	94	38	1	CMBAAA	A LAAAAAA	c VVVVxxxx	xxxxxxxxx	xxxxxxxxxxx
47	1	2	0		2	9	11	71	3	3 1	0	49	92	50	55	VBAAAAA	х МААААА	AAAAxxxx	xxxxxxxxx	xxxxxxxxxxx
462	1	3	1		3	9	16	96		1 2	1	96	67	114	103	URAAAAA	x NAAAAAA	HHHHXXXX	XXXXXXXXXX	(XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
478	1	4	0		0	2	1	44	(5 4	1	35	59	158	63	KSAAAAA	x: OAAAAAA	×0000xxx	xxxxxxxxx	xxxxxxxxxxx
788	1	5	1		0	4	12	90		2 2	0	4	41	154	163	IEBAAAA	x PAAAAAA	VVVVxxxx	xxxxxxxxx	xxxxxxxxxxx

Discussion:

Through this project we gained a better understanding of how systems manages data, generating data based on a specific schema and uploading data into the system using a generated script. We also got familiar with how to establish connection from an editor to any system and populate table instead of directly uploading from a CSV file which we have done in our previous projects in database. It was a great experience learning about these technical aspects and last but not the least learning from others researches.

We have encountered some issues while installing softwares for connecting MySQL and an editor (Intellij). While uploading data, we also got some problems regarding tempering the integrity of the relational schema. For example, the attribute which is the primary key was mentioned not null and it shouldn't be mentioned, because it is by default unique.