Computer and Information Science
www.ccsenet.org/journal.html
Vol. 2, No. 3
August 2009

Formulas of Retrieval Rate in Database Based on Normalization Design

Bin Yan
Library, Nanjing University of Posts and Telecommunications, Nanjing, 210003, China
E-mail: yanb@njupt.edu.cn

Abstract

A normalization database is considered as mathematics logical design to avoid redundancies and inconsistencies in the stored data which brought about null data and many relationship tables in the physical database. A retrieval rate is affected directly in running a practical system of the normalization design. We argue in the paper that rate formulas given out retrieval speed via testing the vast amount of data with decomposing fields and data in practical or ideal network environment. The rate formulas show that relationship tables increased must decrease retrieval speed and a minimum table must increase retrieval speed on the contrary, which compared with normal form according to the database theory and practical requirement. A speed is predictable to calculate the rate formulas. The retrieval speed is mainly concerned with server performance base on normal form too.

Keywords: Database design, Normal form, The vast amount of data, Server group, Retrieval speed, Rate formula of dynamics

1. Introduction

Traditionally, database design activities are partitioned into distinct phases in which a logical design phase precedes physical database design. The objective of the logical design step is to eliminate redundancies and updating anomalies using the notion of data dependencies, while leaving the physical design step to consider how the database schema may be restructured to provide more efficient access. Over the last decade, management system is more and more requisition for database design following functional dependencies of all sort of normal forms. The design process keeps a series of good properties which ensure their correct of functioning and limited redundancy (P. Cordero, A. Mora, I. P. de Guzman, M. Enciso, 2008, pp. 911-923). The theory of normalization has a special prominence as it proposes some properties that database must meet in order to avoid redundancies and inconsistencies in the stored data. Three new normal forms-the relaxed 3NF, replicated 3NF and relax-replicated 3NF, induced by the strong and weak functional dependencies, provide a theoretical framework for designing database schemas which are more efficient and practical, while not compromising the integrity of the underlying database. Relations in these new normal forms will not suffer from undesirable updating anomalies (Tok Wang Ling, Cheng Hian Goh, Mong Li Lee, 1996, pp.601-608).

Attribute value can be now accomplished in general database management system as non-decomposable minimum datum unit according to 1NF. A relation schema R had a set of attributes U and a set of functional dependencies F as R (U, F) of 2NF or 3NF could present more than two concrete relationship, then have many relationship tables and increase foreign keys set associate. A divided table to minimum block according to 5NF defined can limit redundancy and must observe other normal forms. Negation normal form has been emerging recently as a new direction of research for dealing with the query operation of general propositional reasoning (Tong Chun-Ling, Zuo Yi, Li Yu-chen, 2006, pp. 228-231), which decreased connecting operation, numbers of index and foreign keys. The basic notion of a relation was augmented by the concepts of functional dependencies and normal forms in an attempt to provide integrity by reducing undesirable updating anomalies (Codd, E. F, 1997, pp.33-64). A particularly undesirable form of redundancy is the presence in a relation of an attribute whose value can always be derived from other attributes and whose value is not needed to derive other attributes' values. The relation schema may attain to 3NF but not to BCNF (Li Yuchen, Zhang Linjian, Shi Bing, 1998, pp. 907-910) if it is decomposable both integrity and function dependence. Normalization and Ab-normalization are only a logic concept, which practical thing is not always to comply with. Some perfect mathematic relation schema and database behavior are decided physical system by hardware, database size and physical design, data storage and access method, optimization level and access amount of DBMS. In current time a main frequency of CPU has achieved GHz degree and transmission ability does Gbps degree but the retrieval ability does not attain it. When retrieval function is used into an information management system high frequently, design DBMS will be main effect under software and hardware. A software interfusing hardware show, that is, directly to reflect on retrieval results and not theory analysis. Many database designers had not obeyed all normal forms and only obeyed 2NF, little obeyed 3NF because of frequency query data need (Ye Zhong-jie, 2002, pp. 34-38). Sometimes purposedly reserving partial redundancy in design may be convenient data query (Xi Yan, Pan Yu, Xu Guanghui, 2003, pp. 107-108). Sometimes software engineer lacked business knowledge to bring about redundancy in design system. We know that each time of data relation is calculated a time by a computer and the more relation tables are, the more a computer consumes resource.

2. Test methods and data collection

There is not previous paper to discuss on testing computer by the vast amount of data. Concurrent testing and computer simulation (J. Dennis Balenovich, David M. Gillott, John W. Motto Jr, 1973, pp. 227-235) was a testing method for circuit and was to verify capability of thyristor high-frequency. The moment equation (dm_k/dt) that is a state vector in the phase space and the evolution of time-dependent is nonlinearity in dynamical system (K. Sobczyk, 2001, pp. 475-498). The test purpose in our paper is to determine bottleneck caused retrieval speed down in a server with the vast amount of data which stored were derived from the library management system attained to 3NF. We can not test the vast amount of data for hardware and is only to see running hardware phenomena. Server used by us is Fujitsu GP7000F Model 400A (SPARC64-GP300MHz/8MB double CPU, 2GB), 18.2GB and 36.2GB 10000RPM disks. There is 15% CPU load and 30% physical memory holding in normal condition, but it took more than 30 seconds to retrieve one time in thousands or million of daily record data.

2.1 Testing data from database system

We chose one data table from book current system in library to migrate to another server in order to analyze the structure of data table and connection between tables. The BIBLI table we used was data of the bibliography information. There were 39 fields (columns) in the table included rec_ctrl_id, title, addition_title, volume,stock_num, publisher,publish_date, physical_descrip, unit_num, ISBN_1, ISBN_2, first_author, second_author, hird_author, other_respon_1, other_respon_2, subject_term_1, subject_term_2, subject_term_3, subject_term_4, subject_term_5, series_title, ISSN, abbre_spell_title, abbre_spell_author_1, abbre_spell_author_2, abbre_spell_author_3, abbre_other_respon_1, abbre_other_respon_2, book_search_no, call_no_1, call_no_2, item_type, orig_edit_mark, language_code, libary_code, cater_id, ISBN_X, check_control. The most of fields in the table had 164,906 rows of records.

The other test method was to analyze processing ability of a server. There were 13 fields in Check_in_out_log table included department_id, cur_date,book_barcode, reader_barcode, readertype_id, operate_code,user_id, call_no, language_code, workplace_id, rec_ctrl_id, binding_no, note and 2,969,373 rows of records transferred out from current system. The records were divided into 10% (325,575), 20% (600,900), 30% (904,380), 40% (1,201,500), 50% (1,507,894), 60% (1,801,500), 70% (2,101,100), 80% (2,401,445), 90% (2,726,965), 100% (2,969,373) and 110% (3,294,948) that 100% added 10% and 120% (3,570,273) that 100% added 20%. These parts were copied to different servers. Thus it is a more simple method to analyze ability of each server.

2.2 Equipment and its parameters

(1) Stopwatch: JD-3A model digital stopwatch, 1ms precision.

(2) UPS: USA ACP SmartCellXR 3KV.

(3) Computers: COMPAQ PROLIANT 800 (Pentium Pro, 200MHz, 256K Cathe) /64MB (two sets), SCSI 2G, 4G and 9G disks, 10/100 adapter; LONGSHAO 6060 (Pentium 3,930MHz) /128 MB, IDE 40G disk, 10/100 adapter; Fujistu Primergy MS 610 (double CPU PIII Xeon 700) /2G, IDE 18.2G, 10/100, adds ESCORT DA 3500 array disks (SCSI 73G×8, 128K Cathe, RIAD 0+1).

(4) Network: Cisco 1548, IBM 8271 and 999 9S3024B 10/100 Fast Ethernet switch.

(5) Software: WINDOWS NT 4.0, WINDOWS 2000, PowerBuilder 6.5 and MS SQL Server 7.0.

2.3 Testing method and environment

(1) Choice of Time Segment. We selected three time segments according to rush hour and low hour in network, such as AM 8:15~9:30, noon 12:00~1:15 and PM 2:15~3:30. The test kept two weeks that the first week was on the idea network condition and the second week on the practice one.

(2) Choice of Method. There were six items to test circularly and each item tested three times sequentially. The methods of different enter were used by different reader hobby to find differential connection involving refresh and reenter.

(3) Choice of Network. The network of library is sub-network in campus network and connects 105 sets of computers. The practice network can connect to Internet and the idea one can not do in order to judge out-network affection. The idea network is composed of one computer and two servers connected with hub.

(4) Choice of Retrieval Word. The retrieval word must be retrieved out 20 rows more or less in those fields in which there was not same keyword. We chose keyword “the principle of operate system” as retrieval word in title field and could retrieve out 20 rows with fuzzy retrieval method.

We chose keyword “753586” as retrieval word in book_barcode field to exist in each data part for the test of different data quantity. They were 10% (8 rows), 20% (16 rows), 30% (26 rows), 40% (36 rows), 50% (42 rows), 60% (46 rows), 70% (51 rows), 80% (55 rows), 90% (55 rows), 100% (55 rows), 110% (63 rows) and 120% (71 rows).

(5) Data Processed. We counted the average of three times for each item and data of same method every day and reserved 4-bit significance digits.

(6) Divided Data. The single table preserved primary all fields and data as a referencer. The new tables were made of the table of existed data and the table of null data in fields related with rec_ctrl_id field so called the table of divided fields. The divided data table was one divided into two from primary data with rec_ctrl_id field related. Each table was placed in different servers in order to analyze effect of server group.

(7) Test Network. The system of network was used as a client/server mode.

3. Results: analysis

3.1 Comparing to divide fields in a table

Dividing fields in a table means those fields of no data in an original table would be deleted and created new table with rec_ctrl_id field relation. There were 14 fields that had data in the original table such as rec_ctrl_id, title, publisher, publish_date, physical_descrip, first_author, second_author, series_title, book_search_no, call_no_1, item_type, orig_edit_mark, language_code, cater_id except 25 fields of no data. The test data for original table were as the standard. See Table 1:

The retrieval speed of the dividing tables was slower than the original table. But when directly retrieve for the table of data, the speed was quicker than the original table after deleting the no data fields. See Table 2:

The result was that the speed of a narrow table was quicker than a wide table. But the retrieval speed was slow if a prescribed method was to divide data into two tables related each other. The method used the narrow table and a few related tables could raise retrieval speed.

$$t_1=(1+am-bn)\cdot t_0 \quad (1)$$

In Eq. (1) it is shown retrieval time t_1 for affect factors of database can be completed successfully provided *am* and *bn,* where *a* is divided table coefficient, we set value 0.12, and *m* is number of tables related, only one data table here, *b* is deleting no data fields coefficient, we set value 0.015, *n* is number of deleting no data fields, about 25 fields deleted. The symbol t_0 is initial retrieval time, without loss no data fields and setting new tables related.

3.2 Dividing data into two tables

The data divided tables were that 164,906 rows of records in original table were divided two parts. One part had 81,693 rows of records in a table and the other part had 83,213 rows of records in other table. Two tables were related with rec_ctrl_id field. One of them was set in the same server or another server. Server group were composed of two more than servers and could use to conjugate retrieval. Test results see the Table 3.

We can see a result in Table 3 that data-divided-tables are obviously quicker than one table but data divided tables in server group are quicker than in one server. Although the processing possibility increased a time with two computers than only one computer if the computers were same, test speed was only to raise a little more than 20% under 50% data quantity. It is the best method for retrieving the vast amount of data. The retrieval time *t* should amount to various terms, which we have it practically in Eq. (2):

$$t=(t_1+t_2)/2 \quad (2)$$

$$t_2=(1-cp-dq)\cdot t_0 \quad (3)$$

In Eq. (3) it is a express formula of the retrieval time t_2 in detail, where t_2 denotes the retrieval time of dividing data in data table, therein *c* is dividing data coefficient, we obtain result 0.125 via calculating data, and *p* is number of data divided times, and *d* is server group coefficient, there is 0.029, and *q* is server number of composed of server group.

3.3 Comparing in a whole day

From test data in a whole day the table and field divided tables had a bit fluctuation but data divided tables had big fluctuation no matter one server or server group. See Table 4:

This fluctuation was not regular and was a bit improvement to steady for space after starting server.

3.4 Comparing the vast amount of data divided tables

We created new database log_ana in LONGSHAO model 6060 computer and data of log_ana were set 2,969,373 rows from Check_in_out_log data table in the current system. The first time test datum was discarded because connecting back end database caused retrieval time protraction. Test records show in Table 5 at 21°C after test data was stable with no other server connection. Counting out retrieval numbers per second sees Table 6 and average is 180,436 rows per second.

Test reports show in Table 7 in the same condition for COMPAQ PROLIANT 800 server with additional 9G SCSI disk as slave. System construction adopted client/server pattern that LANGCHAO model 6060 was viewed as client and COMPAQ PROLIANT 800 regarded as server. Table 8 shows rows per second counted and their average was 78,726 rows per second.

The ideal network condition for test could be composed of Fujistu Primergy MS 610 server with ESCORT DA 3500 array disks and LANGCHAO model 6060 computer through connecting Cisco 8541 exchanger. The Table 9 is the test report at 21°C and the Table 10 is data to count row number per second retrieval, averages 609,840 row/s.

This purpose was to know the ability of processing the vast amount of data by computer and found inflection point of test value in order to improve most processing ability. Those data could be moved out after data of the inflection point, called the vast amount of data split. But the phenomenon did not happen from beginning to end during the test work. The results see as follow every table.

(1) Determining the effect of the vast amount of data

The synthetically processing ability of computer shows better blend technique of hardware and software. The result of any collocation will be a reflection of output equipment, showing on the screen of a monitor. There was not obvious change at retrieval speed to test different data quantity. We counted out row number per second of retrieval and again out average row number per second. There were maximum, minimum and average values for each server and the absolute value between max or min and average and their percent of average value could be seen in the Table 11.

Max and min values were lower than 10% and a server was normally processing ability. But the values were over 10% and appeared unstable of retrieval speed. It was that a computer caused effect for data processing ability and, further speaking, the computer emerged the effect of the vast amount of data. We propose that concept of the vast amount of data is to determine retrieval ability of a computer and decrease depositing data capacity in a computer for developing most ability of effectively processing data.

(2) Data quantity direct proportion retrieval speed

It can be seen in table 5, table 7 and table 9 that retrieval time was prolonged for different servers according to increase data quantity. The average number was approximate to the same value of processing data quantity by computer if different percent data quantity divided by every average time of test item made numbers per second. The numbers per second were in swing in the average numbers, and they, processing quantity per second, were linear that data quantity increased and time was consumed long. So we assume that data quantity and retrieval time are in direct proportion (see Eq. (4))

$$r=ka\cdot(1+f) \tag{4}$$

where r denotes the dynamics rate (data amount/time), therein k is rate constant associated with computer integrated process capability, i.e. Hz, disk(r/m), a is normal processing rate, f is vibrating coefficient and its value ranges from 0.15 to 0.30 that is over 10% with appeared unstable of retrieval speed.

$$k=\frac{nt_0}{\sum_{i=1}^{n} t_i} \tag{5}$$

In Eq. (5) rate constant k is placed a status in great data amount when $\Sigma t_i \approx t_0$, $k=1$.

(3) Server performance main effect factor for retrieval

The whole performance of server determined processing ability under client/server network pattern because data were deposited in database in server. The client computer was changed and server only changed each from COMPAQ PROLIANT 800 to LANGCHAO model 6060 to Fujistu Primergy MS 610. The test results in table 5, 7 and 9 shows the performance from better to good to best and client performance was unrelated.

(4) CPU as main performance

We noted that 30% and 70% in table 7 shows lower retrieval speed than 20% and both sides. We farther found the phenomenon was CUP occupancy resource from 30% to 60% during retrieval high percent data.

4. Conclusion

Above individual event retrieval speed tested we obtained some influence coefficients. All sorts of influence factors could become much more intricacy during computer processing data and transmission in network. The final rate we discussed should be a function of data processed and transduction $r=r(x)$and x is relevant to time. We give a rate formula:

$$r(x)=k_xa(1+f) \tag{6}$$

In Eq. (6) we know computer processed data amount unit time *a* is invariable and the rate coefficient k_x affects by more factors. The problem of increasing rate becomes to resolve the problem of rate coefficient *k* in fact.

We can give the evaluated dynamic rate formula through above analysis for test data. The fastest retrieval speed is narrow data table according to 5NF after deleting no data fields. And faster speed that is data divided tables according to 3NF in server group, but unstable phenomena could occur. The longest retrieval time is the first logging database because of needing time to connect it and retrieval time is stable once connection in success. Major factors of the effected retrieval speed are not room temperature and bandwidth of network but server performance. Server performance can be changed if some hardware such as disk is renewed, otherwise some strange phenomena could happen such as retrieving out double same rows. Data quantity is main factor to affect server performance and available divided data may relieve processing limit of server. It is an effective pathway of protective equipment investment that performance of different servers used reasonably has got server group of management systems.

These formulas are different from the results of information dynamics. One of formulas was proofed a=0.125, b=0.018 within formula (1) by my student undergraduate thesis. Our test results highlight that the server group has more processing ability to compare with a server under same data quantity. The processing ability is how to change along data quantity increase in same condition of network, equipments and temperature with the different ability and different data quantity of each server. The testing result may judge how many data quantity moved into other server in order to relieve burden of the server.

Acknowledgments

I thank Ms. Yang Jinjing and Mr. Wang Haifeng and Mr. Chen Kai, my students, of Nanjing University of Posts and Telecommunications for testing work during their undergraduate theses. The project was supported by the Scientific Research Foundation of Nanjing University of Posts and Telecommunications (NY206024).

References

Codd, E. F. (1997). Further normalization of the data base relational model. In Data Base System, R. Rustin, Ed., *Courant Computer Science Symposium*, vol. 6, Prentice-Hall, Englewood Cliffs, N. J., pp. 33-64.

J. Dennis Balenovich, David M. Gillott and John W. Motto Jr. (1973). Thyristor high-frequency ratings by concurrent testing and computer simulation, *IEEE Transactions on Industry Application*, Vol. IA.9, No. 2, pp.227-235 MARCH/APRIL.

K. Sobczyk. (2001). Information Dynamics: Premises, Challenges and Results, *Mechanical Systems and Signal Processing*, 15(3), pp.475-498.

Li, Yuchen, Zhang, Linjian and Shi, Bing. (1998). A consistency algorithm for realizing the reconfiguration of partitioned database in DDBS. *Journal of Computer Research and Development*, ISSN: 1000-1239, 35(10):907-910.

P. Cordero, A. Mora, I. P. de Guzman and M. Enciso. (2008). Non-deterministic ideal operators: An adequate tool for formalization in Data Bases. *Discrete Applied Mathematics*, ISSN: 0166-218X,156 pp. 911-923.

Tok Wang Ling, Cheng Hian Goh and Mong Li Lee. (1996). Extending classical functional dependencies for physical database design. *Information and Software Technology*, ISSN: 0950-5849, 38, pp.601-608.

Tong, Chun-Ling, Zuo, Yi and Li, Yu-chen. (2006). Normalization and Abnormalization of Relation Schema Design. *Journal of Jinan University (Sci.&Tech.)*, ISSN: 1671-3559, 20(3):228-231.

Xi, Yan, Pan, Yu and Xu, Guanghui. (2003) Design Formula of Database. *Modern Electronic Technique*, ISSN: 1004-373X, (24) pp.107-108.

Ye, Zhong-jie. (2002). Database normal formula and application. *Journal of Zhejiang vocational and Technical Institute of Transportation*, Vol. 3 No. 4, Dec.pp. 34-38.

Table 1. One server and one table comparing with fields dividing tables (second)

First day average		Second day average		Third day average		Fourth day average	
table	tables	table	tables	table	tables	table	tables
28.75	32.23	28.78	32.21	28.70	32.16	28.84	32.57

Table 2. One server and one table comparing with table of all data fields (second)

First day average		Second day average		Third day average		Fourth day average	
table	data table	table	data table	table	data table	table	data table
28.75	21.48	28.78	21.47	28.70	21.45	28.84	21.62

Table 3. One table comparing with tables in one server(1) and servers(2) (second)

First day average			Second day average			Third day average			Fourth day average		
table	tables 1	tables 2	table	tables 1	tables 2	table	tables 1	tables 2	table	tables 1	tables 2
28.75	25.25	23.23	28.78	24.80	23.40	28.70	24.94	23.15	28.84	25.50	23.97

Table 4. Data divided tables in server or servers (second)

	First day average		Second day average		Third day average		Fourth day average	
No.	server	servers	server	servers	server	servers	server	servers
1	25.92	25.84	25.66	25.86	25.59	25.93	25.80	25.73
2	25.71	25.53	25.64	26.09	25.75	25.20	25.98	25.02
3	25.71	25.50	25.69	26.00	25.60	25.43	25.94	25.06
4	25.54	25.65	25.66	25.97	25.54	25.22	25.84	25.13
5	25.68	25.20	25.62	26.07	25.65	25.79	25.72	25.01
6	25.61	25.27	25.56	26.08	25.56	25.15	25.73	25.28
7	26.33	22.16	24.40	22.11	24.41	21.98	26.02	25.03
8	26.10	22.03	24.38	22.10	24.41	22.00	25.94	24.17
9	26.32	22.06	24.57	21.95	24.49	22.05	25.65	24.28
10	26.01	22.12	24.10	22.04	24.44	21.68	25.66	24.42
11	24.53	21.90	24.29	21.96	24.56	21.82	25.70	24.33
12	24.70	22.09	24.41	22.05	24.94	21.96	25.74	24.43
13	24.33	22.11	24.38	22.16	25.01	22.06	25.71	24.03
14	24.31	22.16	24.42	22.24	24.70	22.09	25.80	24.26
15	24.32	22.17	24.44	22.30	24.58	22.05	25.67	24.24
16	24.34	22.02	24.34	22.11	24.65	22.07	25.61	24.36
17	24.42	22.28	24.48	22.07	24.58	22.07	25.70	23.99
18	24.55	22.12	24.36	22.08	24.43	22.07	25.91	24.23
average	25.25	23.23	24.80	23.40	24.94	23.15	25.78	24.61

Table 5. LANGCHAO model 6060 server test report (second)

	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
first	1.89	3.36	4.87	6.71	8.10	9.83	11.56	13.37	14.74	15.63
second	2.08	3.98	4.96	6.89	8.03	9.66	11.60	13.17	14.99	15.46
third	1.93	3.37	4.87	6.74	8.16	9.51	11.80	13.23	14.83	15.43
average	1.97	3.57	4.90	6.78	8.10	9.67	11.65	13.26	14.85	15.51

Table 6. LANGCHAO model 6060 server (rows per second)

data	10%	20%	30%	40%	50%
row/s	165,266	168,319	184,567	177,212	186,160
data	60%	70%	80%	90%	100%
row/s	186,298	180,352	181,104	183,634	191,449

Table 7. COMPAQ PROLIANT 800 server(9G)test report (second)

	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	110%	120%
first	4.14	9.74	8.64	14.73	19.86	26.03	30.45	37.43	43.43	39.73	32.14	36.06
second	4.14	9.76	8.54	14.86	19.86	25.95	30.39	37.24	43.43	39.49	32.26	35.99
third	4.14	9.79	8.53	14.84	19.81	26.01	30.37	37.40	43.43	39.48	32.29	36.04
average	4.14	9.76	8.57	14.81	19.84	26.00	30.40	37.36	43.43	39.57	32.23	36.03

Table 8. COMPAQ PROLIANT 800 (9G) server (rows per second)

data	10%	20%	30%	40%	50%	60%
row/s	78,641	61,568	105,529	81,128	76,003	69,288
data	70%	80%	90%	100%	110%	120%
row/s	69,115	64,279	62,790	75,041	102,232	99,092

Table 9. Fujistu Primergy MS 610 server test report (second)

	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
first	0.58	1.08	1.60	2.04	2.46	2.85	3.28	3.78	4.12	4.54
second	0.66	1.13	1.55	2.05	2.43	2.89	3.24	3.76	4.07	4.53
third	0.53	1.10	1.59	1.96	2.49	2.89	3.35	3.74	4.07	4.51
average	0.59	1.11	1.58	2.02	2.46	2.88	3.29	3.76	4.09	4.53

Table 10. Fujistu Primergy MS 610 server (rows per second)

data	10%	20%	30%	40%	50%
row/s	551,822	541,351	572,392	594,802	612,965
data	60%	70%	80%	90%	100%
row/s	625,521	638,632	638,682	666,740	655,491

Table 11. Average row number per second to retrieve and fluctuation ratio

server	LANGCHAO	COMPAQ	Fujistu
average	180,436	78,726	609,840
max	6%	34%	9%
min	8%	22%	11%