View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Periodicals of Engineering and Natural Sciences (PEN - International University of...

Periodicals of Engineering and Natural Sciences ISSN 2303-4521
Vol.6, No.2, December 2018, pp. 302~315
Available online at: http://pen.ius.edu.ba

A Novel Approach for Iceberg Query Evaluation on Multiple Attributes Using Set Representation

V. Chandra Shekhar Rao [1], Dr. P. Sammulal [2],
[1]Research Scalar, Department of CSE, JNTU, Hyderabad, vcsrao.kitswgl@gmail.com
[2]Associate Professor, Department of CSE, JNTU, Hyderabad, sammulalporika@g

Article Info

Article history:

Received Jun 12th, 2018
Revised Aug 20th, 2018
Accepted Aug 26th, 2018

Keyword:

set operations
iceberg query
Database
threshold

ABSTRACT

Iceberg query (IBQ) can be an really identifying kind of aggregation question that calculate aggregations up-on user given threshold (T). In data mining field, effective investigation of compounding queries was because of by the majority of investigators because the tremendous generation of information outside of industrial and businesses industries. Conclusion assist database and discovery of the majority of information connected systems largely calculate the worthiness of most fascinating features having an critical level of information from data foundations that may be tremendous. By means of the paper, we propose that an initial Manner of calculating IBQ, which builds a choice for every attribute nicely value, but additionally includes a One of a Kind events Inside the attribute column also plays specify operations for creating closing Outcomes. We formulated highly effective GUI software for just 2 characteristics, numerous traits employing egotistical prepare and several features utilizing lively plan. If data collection comprises two traits, then it truly is substantially more advanced than apply just two traits. In the event of information collection comprises multiple traits, predicated up on anyone choice suitable module could potentially be decided on. If characteristic uniqueness changes from characteristic in to the following characteristic, then vibrant variety approach is very powerful. This strategy somewhat reduces performance memory and time space contrast with additional processes. A experiment using artificial Statistics collection and actual info demonstrates our strategy will be considerably more effective compared to present apps for Nearly Every threshold.

Corresponding Author:

Dr. P. Sammulal,
Associate Professor, Department of CSE, JNTU,
Hyderabad, sammulalporika@gmail.com

1. Introduction

Industry awareness and Recognizing Detection [1] from Dealing databases/warehouses always hardy fire-arms, So that you may acquire competitive wages in the current industry community. An Iceberg Query(IBQ) could be an excellent Form of a aggregation question that divides values to a Individual Outlined threshold (T) meanings. It truly is of these differentiating comprehension of most those end users in bringing advanced level degree worth which frequently take more considerable in manufacturing companies. The Syntax with the Iceberg query as to a romantic relationship REL (C1, C2. . .Cn) is revealed beneath:

SELECT Ci, Cj, …, Cm, AGG(*) FROM R GROUP BY Ti,Tj…, Tm HAVING AGG (*) > = T

This aggregation performs,"at which by Ci, Cj,...,Tm" suggests a sub set pair of capacities in well-known mix faculties. Aggregation functions just like COUNT(),COUNT(*),MIN,MAX,SUM and AVG. The more

expensive in comparison with equal to >= could possibly be considered a index used like a contrast predicate.

1.1 Market Basket Analysis Marketplace analysts apply market place basket issues into substantial data warehouses [2] that save purchaser earnings transactions. These inquiries explain person paying for patterns, so thus by delivering thing monies (and triples) that may be attracted collectively by numerous clients. Assessing those questions will be powered with huge numbers collections. We utilize exactly the Business basket question locate normally happening term monies.

1.2 Set Operations Advantages Set up operations greatly help speed the Iceberg Queries in diminished execution period in contrast with remaining IBQ techniques like tuple scan-based prepare and lively pruning. It lowers the amount of iterations amongst put pairs of 2 exceptional faculties by occurring difference so correcting the sets that doesn't fit the ceremony worth. Moreover, it helps in repairing the locations thus decreasing the iterations Following the threshold ailment does not match.

When applications can be hard to get the job done with, it compels users to-do glitches, or no matter whether it frees users attempts to achieve their particular objectives, subsequently can hate that, irrespective of computational potential it displays or even the functionality it includes; since it ultimately ends up making a customer's perception of these applications, or so the interface must be proper. The minute we appear programs, we have to contemplate the future customers, such as profiles with the age, instruction, gender, physical talents and cultural or cultural heritage, enthusiasm, goals and disposition. For the explanation, one port design may possibly perhaps not be perfect for a number of computer system users although it might possibly be just precious to specified users.

2. Related work

A number of this plan that may be helpful for smaller sized database would be: Sorting REL on disc afterward proceeds aggregating and selecting the formerly recorded threshold values. These procedures don't scale into large information collections. Thus, other procedures are indispensable. A couple Them have been:

2.1 Sampling

This System samples a few of Documents by Your Bond, aggregates and extracts Which the Documents "Candidates in to the Prior Remedy" Which (the sample size) Go the threshold.

2.2 Bucket counting

As opposed to committing a counter to every single various selling price, give a counter tops for a pair of special worth, acquiring a hash functionality to split the worth into classes. These cubes create bogus favorable, values which can be thought candidates to this prior remedy but usually do not transcend this threshold.

2.3 Tuple scan based approach

More of the query advertising techniques for calculating compilation queries could be categorized whilst the tuple scanning base application, which requires the minimum minimal of 1 definite dining table scanning to navigate advice out-of disc drive. They pay give attention to lessening the reach of motions that the instant the data dimensions is not important. None has effectively mastered the territory of compounding concerns to Come Across powerful communication. This kind of tuple-scan-based strategy frequently wants a very long time to respond unanswered queries, particularly in the event your dining table remains nonetheless quite important. Besides those tuple-scan-based strategies, manufactured a two-level hardly any map index that could potentially be leveraged processing compilation queries.

You'll come across several exceptional information structures employed in analytics base to receive paid indexes used to rapidly evaluate queries. Truly one of these basic kinds of signs is termed as a bit map

indicator. Little map indices are shown to triumph [6], notably due to analysis mainly or append just data, and in addition are normally utilized at the information warehousing software and pillar stores applying little map indices, we want must attain little map indices in their aggregate faculties. Second, little map indices increased exposure of pieces rather than real tuple values.

These are strive to look for consistency, so thus empower ordinary users to use shortcuts, and gives insightful feedback, design return closing and supply simple malfunction management, and enable easy adjustment of activities, boost internal locus of direction. The Fantastic Way to introduce info to customers which are new into some subject, Ergonomics, the Way in Which stance affects efficacy and Designing a point game for your handicapped, Input and Output apparatus, New input and input apparatus produce our private lives simpler, Design methods: Success Quantities of Merchandise Using language recognition and also a whirlpool design Together with Powerful Utilization of colour within interface layout, imitating the Various Different system Growth procedures.

3. Proposed system

3.1 GUI

Later catalog places of 1 of 2 2 channels Am and B M are recovered. At case the threshold has been passed, then then your intersection performance is achieved between your pairs satisfied A B and also exceed group. If the impact of intersection established gets the urge compared to this threshold, then confirm this vector spot currently being fully a outcome and put in them into rotational effect assortment. Afterward the Am and B M Establish regions are upgraded by running the gap performance to get longterm mention. The upgraded piece map vectors of both Am and B M to evaluate little index places together side the verge, even also should they are inside of the threshold that your aforementioned process is going to be lasted. Carry-on Just the Exact Same procedure until Each Of of the vector pairs Have Been finished.

Table 1 index of bitmap

X	X1	X2	X1	X2	X1	X2	X1	X2	X1
Y	Y2	Y3	Y2	Y2	Y2	Y1	Y1	Y2	Y3

Step 1: Aggregated Attributes Bit map indices

X_1: 1 0 1 0 1 0 1 0 1
X_2: 0 1 0 1 0 1 0 1 0
Y_1: 0 0 0 0 0 1 1 0 0
Y_2: 1 0 1 1 1 0 0 1 0
Y_3: 0 1 0 0 0 0 0 0 1

Step 2: Sets Extraction: Grab an indicator position of 1 piece from every bitmap. Establish X1: {0,2,4,6,8}, X1.count=5, Establish X2: {1,3,5,7}, X2.count=4, Place Y1: {5,6}, Y1.count=2, Establish Y2: {0,2,3,4,7}, Y2.count=5,Establish Y3: {1, 8}, Y3.count=2.

Step 3: The aligned collections returned from vector orientation algorithm out of resolution Estimates are SA1 and SB2. The iceberg effect is calculated as: ST= SX1={0,2,4,6,8}, SX1= SX1- SY2={0,2,4,6,8}-{0,2,3,4,7}={6,8}. SY2= SY2- ST={0,2,3,4,7}-{0,2,4,6,8}={3,7}, Csize of ST-size of both SX1=5-2=3. Since the value of C will be higher than the X1,Y2 using count 3 has been inserted into effect R and SX1 and SY2 are pushed right back into priority queues because there fresh dimensions are over two (threshold)

3.2 Discussion on implementation

This section describes the different modules Which Were Suggested in the previous section and Also the details are as Follows:

1. Building database with just two features: We shall start the creation of database using two features, database with several features having uniform database and uniqueness with multiple database using random uniqueness by randomly integrating the rows in to your database. Synthetic data collection generated with zipfian distribution.

2. Bitmap indices Creation : Generate the bit maps of 1's and 0's. By utilizing this piece maps only, we move to additional approaches. For I=1 to Dining Table Size, if value of attr1 at row is an subsequently, attr2[I]=1 attr2[I]=0.

3. Set_Generation: This module scans entire database also prepares sets for several distinct attribute values by keeping its positions in various columns.

4. It uses First_Position module to get first 1 bit element of a collection. This module guarantees of creating non empty intersection effect

5. Evaluation of IBQ utilizing set operations on two features: Iceberg queries are Conducted places using the Group Operations such as set intersection and set difference.

6. IBQ Assessment using set operations on various features using egocentric approach: Iceberg queries are Ran on Places with the Category Operations for Example set intersection and set difference.

7. Evaluation of IBQ using set operations on various attributes using dynamic programming principle procedure: This module calculates each attribute uniqueness. Inside this module attribute uniqueness is calculated and stored at variety list. Characteristic uniqueness can help in minimizing the operations in computing iceberg query result. Which also reduces the space and time, the distance optimization would be the Issue of finding the elimination with minimum conclusion predicated on characteristic uniqueness.

The below algorithm shows the functionality of IBQ using set Operations.

1. Algorithm to IBQ evaluation for two attributes[11].

2. Algorithm to IBQ evaluation for multiple attributes using greedy approach[12].

3. Algorithm to IBQ evaluation for multiple attributes using dynamic programming principle approach[13].

ALGORITHM-1 IBQ(attributes X, attribute Y)

1. For every set X_1 of attribute X, Store the position in set as set element in Sorted Set.
2. Place the vectors of attribute X into Priority Queue depending on its first 1st bit position if their size is greater than given threshold
3. if X_1.size is greater than or equals to T then
   a. SX.push(X_1)
4. for every set Y_1 of attribute Y, store its position in set as set element in sorted set
5. Insert vectors of attribute Y into Priority Queue based on its first 1st bit position if their size is greater than given threshold
6. if Y_1. size is greater than or equals to T then
   SY.push(Y1)
7. Suppose iceberg result R = null
8. Repeat following steps while both priority queues are not empty

9. Retrieve aligned sets S1 and S2 from queues SX, SY
10. Say S3=S1
11. Calculate S1=S1-S2
12. Calculate S2=S2-S3
13. Calculate c=size of S3-size of S1
14. If c is greater than T add vectors with count c into result R.
15. Insert sets into corresponding Priority Queues if their set size is greater than given threshold.
16. Return result R

Algorithm-2 [ ] computes IBQ result for multiple attributes using greedy method.

Algorithm-3 [ ] computes IBQ result for multiple attributes using dynamic programming approach method

4. Implementation

GUI Implementation : The experiments have been conducted on Pentium heart i-5 chip of 3.6GHz, 8GB main memory and also 7200rpm IDE drive; and most of algorithms are used in Java, backend is MySQL.

IBQ Implementation : This website comprises level of items out of the database. The services and products transactions are stored indoors database whenever customer perform shopping. This entire items list stays in database and kept by admin, and applying this specific database may implement aggregation on items list and display output as which category of items meets threshold selling price. This item list reflects selection of items provided in supermarket shops. Consumer will buy items determined by availability of things like specified threshold.

Implementation of Things Aggregation: This really is actually the final resulting page. Within these pages, it's display output signal as the couple of things fulfilling the threshold price. When size of the database is elevated afterward threshold appreciate for example 100,200,300 etc.. Database comprises so many features; from that feature list each single time required two features for acting aggregation. Set of things screens as output that couple of things attained service value.

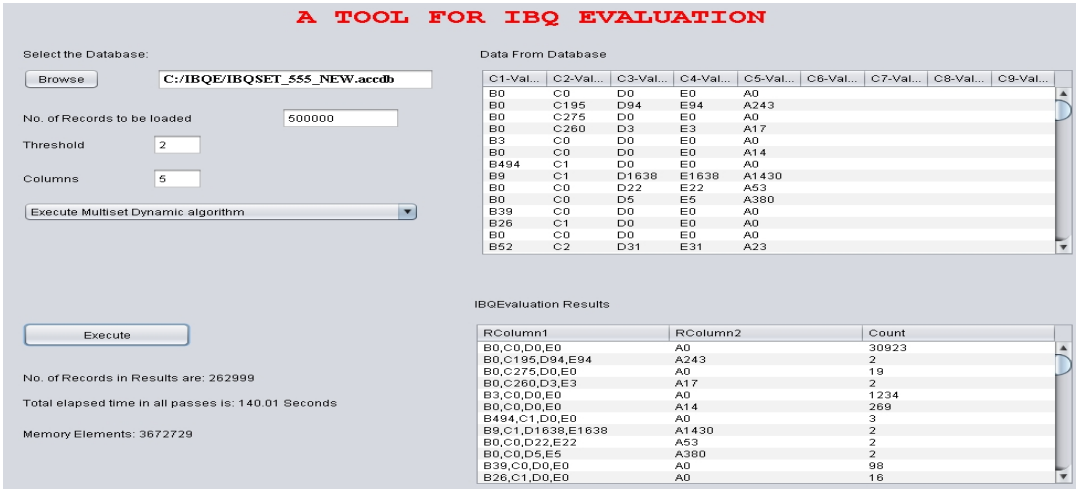

Figure 1: Sample output screen

The above figure 1 shows a sample output screen, which allows database name, threshold, database size and in turn it gives IBQ result for given threshold in output window, the input selected as database is 5 lakhs, threshold is 2, number of attributes is 5.

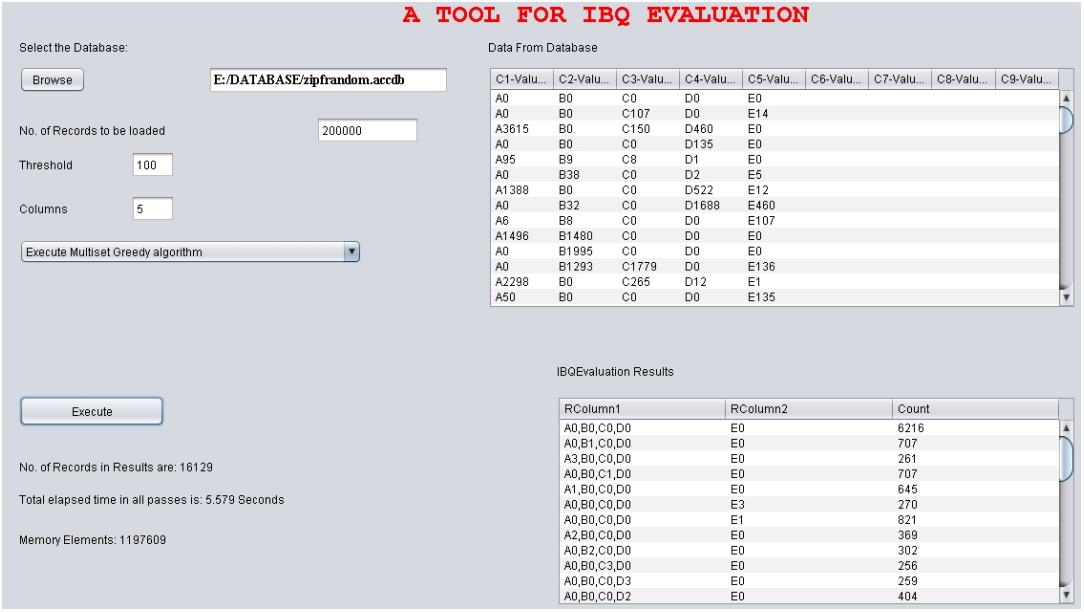

Figure 2: Sample output screen

The above figure 2 is a sample output screen, which allows database name, threshold, database size and in turn it gives IBQ result for given threshold in output window, the input selected as database is 2 lakhs, threshold is 100, number of attributes is 5.

5. Results

In this section, the results obtained by experimentation in the previous section are recorded and analyzed in the following tables with various thresholds, various database sizes among existing and proposed approaches.

Table 2: Performance on various thresholds with two attributes

Threshold	IBQ_ MAIN	IBQ_ SET	IBQ_ MAIN	IBQ_ SET	IBQ_ MAIN	IBQ_ SET	IBQ_ MAIN	IBQ_ SET
	Database: 1 Lakh		Database: 2 Lakhs		Database: 3 Lakhs		Database: 4 Lakhs	
2	1.204	0.766	3.089	1.605	5.107	2.889	7.173	3.56
3	0.81	0.632	2.382	1.411	4.462	2.611	6.387	3.50
4	0.686	0.508	1.908	1.246	3.734	2.324	5.944	3.42
5	0.541	0.481	1.529	1.158	3.167	2.25	5.357	3.261
6	0.528	0.443	1.299	1.016	2.675	1.943	4.489	3.166
7	0.498	0.434	1.155	0.954	2.268	1.744	3.961	2.866
8	0.472	0.417	0.994	0.884	2.034	1.508	3.546	2.512
9	0.452	0.377	0.968	0.85	1.758	1.422	3.054	2.362
10	0.418	0.309	0.921	0.806	1.687	1.388	2.784	2.117

The above result demonstrates IBQ evaluation on two attributes, the table 2 consists of 9 columns and 10 rows. The columns defines the database size by ranging from 1 lakh to 4 lakhs, Execution times are shown

for various thresholds and different database size on two attributes.. The final result demonstrates the fall towards execution time for proposed approach(IBQ_SET) compared to exiting approach(IBQ_MAIN) on two attributes.

Table 3 : Performance on various thresholds with five attributes having different uniqueness

Threshold	IBQ_GM	IBQ_DM	IBQ_GM	IBQ_DM	IBQ_GM	IBQ_DM	IBQ_GM	IBQ_DM
	Database: 1 Lakh		Database: 2 Lakhs		Database: 3 Lakhs		Database: 5 Lakhs	
2	13.756	7.138	22.945	21.193	48.867	43.638	157.049	138.576
3	7.978	5.538	16.968	14.164	37.464	33.45	117.702	106.638
4	7.756	4.111	15.288	13.404	33.588	30.484	108.742	93.232
5	6.826	3.789	13.691	10.736	32.009	27.466	105.688	84.741
6	6.385	3.373	13.342	10.194	30.079	26.556	103.742	81.136
7	6.561	3.279	12.424	9.225	27.484	25.113	93.113	78.670
8	6.210	2.811	12.211	9.230	26.044	24.512	90.399	74.240
9	5.986	2.617	12.083	8.257	25.327	24.201	88.727	70.019
10	5.817	2.582	10.482	7.657	23.871	18.811	86.839	64.887
20	5.091	2.018	8.368	6.349	18.044	14.923	69.067	60.248
30	4.826	1.587	8.170	4.781	14.397	12.091	57.628	53.630
40	4.649	1.453	6.838	4.138	11.125	10.623	47.064	48.159
50	4.373	1.402	6.462	4.002	10.866	10.087	44.983	42.001
100	4.161	1.222	5.579	3.017	6.587	5.855	30.334	28.388
200	4.054	0.874	5.080	1.977	4.874	4.192	24.302	19.572
300	4.019	0.797	4.702	1.961	4.098	3.284	18.975	15.143
400	4.001	0.780	4.643	1.641	2.975	2.791	15.705	12.878
500	3.716	0.724	4.428	1.285	2.848	2.001	13.198	11.686
1000	3.381	0.674	3.918	1.061	2.254	1.648	9.334	7.951

(Attribute a=5000,b=4000,c=3000,d=2000,3=1000)

The above result demonstrates IBQ evaluation on multiple attributes having different uniqueness for attributes, the table 3 columns describes the database size by ranging from 1 lakh to 5 lakhs, Execution times are shown for various thresholds and different database size on multiple(five) attributes. The result specifies the fall in the execution time for proposed approach (IBQ_DM) compared to exiting approach (IBQ_GM) on multiple attributes wherein database generated with different attribute uniqueness. Attribute a=5000, b=4000, c=3000,d=2000, e=1000.

Table 4 : Performance on various thresholds with five attributes having equal uniqueness (IBQ_GM) on multiple attributes wherein database generated with equal attribute uniqueness.

Threshold	IBQ_GM	IBQ_DM	IBQ_GM	IBQ_DM	IBQ_GM	IBQ_DM
	Database: 1 Lakh		Database: 3 Lakhs		Database: 5 Lakhs	
2	9.627	6.777	47.848	34.044	146.604	124.919
3	8.02	5.419	32.642	26.137	96.845	91.537
4	7.063	4.048	27.628	22.319	92.038	86.28
5	6.713	3.56	25.822	21.511	89.291	78.676
6	6.37	3.076	24.945	20.451	80.978	76.427
7	6.077	3.068	23.912	19.818	77.37	75.879
8	6.044	3.028	23.112	19.435	74.346	71.632
9	5.982	2.531	22.989	18.307	74.049	70.758
10	5.917	2.283	22.664	17.337	72.12	67.686
20	5.385	1.814	16.369	12.687	62.502	58.499
30	4.974	1.743	12.852	10.513	51.106	48.37
40	4.518	1.328	10.018	9.108	43.968	40.916
50	4.377	1.198	9.906	8.556	38.789	36.381
100	4.233	0.894	6.238	5.921	27.202	25.474
200	3.929	0.681	4.29	3.899	19.589	17.937
300	3.921	0.637	3.955	3.702	15.585	13.987
400	3.919	0.534	2.85	2.770	14.438	11.498
500	3.910	0.528	2.706	2.70	11.48	8.006
1000	3.784	0.412	2.004	1.944	9.355	6.488

The preceding result shows IBQ test on multiple features using uniform feature uniqueness, the dining table consists of the aforementioned result dining table is made up of 1 9 columns and 9 rows. The columns is referred to that database size by which it ranges from 1 lakh to five lakhs. Execution times are displayed for a variety of thresholds and separate database size to multiple(five) features. . The outcomes reveal that the drop at the implementation time for suggested strategy (IBQ_DM) when compared with leaving approach

The link between preceding algorithms are compared with all original , existing approach "Successful Iceberg Query Assessment utilizing Compressed Bitmap Index[1]" using "IBQ test utilizing place rendering [1-1]" on 2 features, next one multiple characteristics of egotistical approach[12] with dynamic programming approach with arbitrary uniqueness, third among multiple features of egocentric strategy with multiple dynamic programming process using uniform characteristic uniqueness. We discovered that the suggested procedures showing better performance compared to existing systems.

We provide the research on implementation time detected by each suggested algorithm by calculating advantage between proposed and existing calculations we tabulated the implementation times on several thresholds for just two features along with numerous features. Dependent on these values calculated profit percent.

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\, n\Sigma x^2 - (\Sigma x)^2]\,[\, n\Sigma y^2 - (\Sigma y)^2]}}$$ Equation ------1

r denotes correlation, and considered as Eq.(1) and reproduced as here under

r= n Σ (xy) – (Σ x)(Σ y) / SQRT (n Σ x^2 Σ y^2) here Σ x and Σ y are sum of thresholds and gain in execution time for existing and proposed approaches to find the correlation r between them. The correlation r value indicates the percentage of faster than existing method. The following 3 tables demonstrate the gain percentages for various thresholds and correlation among existing and proposed approaches.

Table 5: Execution time comparison on two attributes with one lakh database.

Database is 1 lakh / Threshold	Existing Method Execution time	Proposed Method Execution time	Diff of Execution time = Existing - proposed	Gain %
2	1.204	0.766	0.438	36.37874
3	0.81	0.632	0.178	21.97531
4	0.686	0.508	0.178	25.94752
5	0.541	0.481	0.06	11.09057
6	0.528	0.443	0.085	16.09848
7	0.498	0.434	0.064	12.85141
8	0.472	0.417	0.055	11.65254
9	0.452	0.377	0.075	16.59292
10	0.418	0.309	0.109	26.07656
11	0.385	0.275	0.11	28.57143

Table 6 Gain percentage and correlation calculations

Threshold(x)	**Gain % (y)**	**X = x-x^1 x^1 =6.5**	**Y = y-y^1 y^1 =20.73**	**X * y**
2	36.37874	-4.5	15.64	72.75748
3	21.97531	-3.5	1.24	65.92593
4	25.94752	-2.5	5.21	103.7901

Threshold(x)	**Gain % (y)**	**X = x-x^1 x^1 =6.5**	**Y = y-y^1 y^1 =20.73**	**X * y**
5	11.09057	-1.5	-9.64	55.45287
6	16.09848	-0.5	-4.64	96.59091
7	12.85141	0.5	-7.88	89.95984
8	11.65254	1.5	-9.08	93.22034
9	16.59292	2.5	-4.14	149.3363
10	26.07656	3.5	5.34	260.7656
11	28.57143	4.5	7.84	314.2857
Σ x=65 x^1=6.5	Σ y = 207.385 y^1 = 20.73	0.0	0.0	Σ xy = 1302.085

r = (1,302.085) / (65 x 207.385), r= 0.0966 means 9.66 % faster than existing method.

The above table demonstrates first column is threshold denoted by (x), the second column represents gain in execution time, third column indicates X and Y, computes statistical calculations required for r.

Table 7: Execution time comparison on five attributes(multiple) with one lakh database having uniform uniqueness

Database = 1 lakh Threshold	Existing Method Execution time	Proposed Method Execution time	Diff of Execution time = Existing - proposed	Gain %
2	13.756	7.138	6.618	48.10992
3	7.978	5.538	2.44	30.58411
4	7.756	4.111	3.645	46.99587
5	6.826	3.789	3.037	44.49165
6	6.385	3.373	3.012	47.17306
7	6.561	3.279	3.282	50.02286
8	6.21	2.811	3.399	54.7343
9	5.986	2.617	3.369	56.28132
10	5.817	2.582	3.235	55.61286
11	5.291	2.218	3.073	58.07976

Table 8: Gain percentage correlation calculations.

Threshold(x)	Gain % (y)	$X = x - x^1$ $x^1 = 6.5$	$Y = y - y^1$ $y^1 = 49.20$	X * y
2	48.10992	-4.5	-1.10	96.21983
3	30.58411	-3.5	-18.62	91.75232
4	46.99587	-2.5	-2.21	187.9835
5	44.49165	-1.5	-4.71	222.4582
6	47.17306	-0.5	-2.03	283.0384
7	50.02286	0.5	0.82	350.16
8	54.7343	1.5	5.53	437.8744
9	56.28132	2.5	7.08	506.5319
10	55.61286	3.5	6.41	556.1286
11	58.07976	4.5	8.87	638.8773
Σ x=65 x^1=6.5	Σ y = 492.0857 Y^1 = 49.20	0.0	0.0	Σ xy = **3371.025**

r = (3371.025) / (65 x 492.0857.0), r= 0.1053 means 10.53 % faster than existing method.

The above table demonstrates first column is threshold denoted by (x), the second column represents gain in execution time, third column indicates X and Y, computes statistical calculations required for r.

Table 9: Execution time comparison on five attributes with one lakh database having random uniqueness

Database = 1 lakh Threshold	Execution time for existing method	Execution time for proposed method	Diff of Execution time = Existing - proposed	Gain %
2	9.627	6.777	2.85	29.60424
3	8.02	5.419	2.601	32.43142
4	7.063	4.048	3.015	42.68724
5	6.713	3.56	3.153	46.96857
6	6.37	3.076	3.294	51.71115
7	6.077	3.068	3.009	49.51456
8	6.044	3.028	3.016	49.90073
9	5.982	2.531	3.451	57.68974
10	5.917	2.283	3.634	61.41626
11	5.385	2.014	3.371	62.59981

Table 10: Gain percentage and correlation calculations.

Threshold(x)	Gain % (y)	$X = x-x^1$ $x^1 = 6.5$	$Y = y-y^1$ $y^1 = 48.45$	X * y
2	29.60424	-4.5	-18.85	59.20848
3	32.43142	-3.5	-16.02	97.29426
4	42.68724	-2.5	-5.77	170.749
5	46.96857	-1.5	-1.49	234.8428
6	51.71115	-0.5	3.26	310.2669
7	49.51456	0.5	1.06	346.6019
8	49.90073	1.5	1.45	399.2058
9	57.68974	2.5	9.23	519.2076
10	61.41626	3.5	12.96	614.1626
11	62.59981	4.5	14.14	688.598
$\Sigma x = 65$ $x^1 = 6.5$	$\Sigma y = 484.523$ $y^1 = 48.45$	0.0	0.0	$\Sigma xy = 3440.137$

r = (3440.137) / (65 x 484.523), r= 0.1092 means 10.92 % faster than existing method.

The above table demonstrates first column is threshold denoted by (x), the second column represents gain in execution time, third column indicates X and Y, computes statistical calculations required for r.

5.1 Space Optimization:

The following tables 11 and 12 shows the number of locations saved for various thresholds on different databases. This shows the memory space reduced between existing approach and proposed approach.

Table 11: Memory locations saved for one lakh database

Threshold/One Lakh database	Existing IBQ_GM (occupied memory locations)	Proposed Algorithm IBQ_DM (occupied memory locations)	Number of memory locations saved
50	613821	547524	66297
150	592722	489489	103233
250	584006	463922	120084
350	578486	446594	131892
450	573936	433812	140124

Table 12: Memory locations saved for three lakhs database

Threshold/Three Lakhs database	Existing: IBQ_GM (Number of memory locations)	Proposed: Algorithm IBQ_DM (Number of memory locations)	Number of memory locations saved
100	1865030	1706369	158661
200	1823935	1594144	229791
300	1802420	1531885	270535
400	1784783	1485707	299076
500	1773071	1451063	322008

The above mentioned results includes 5 row and 4 columns, Row indicates different thresholds and pillar indicates memory positions required for place operations for existing process utilizing egocentric procedure (IBQ_GM)[12] and suggested system (IBQ_DM)[13] with dynamic programming principle strategy. The final column means quantity of memory locations stored for a variety of thresholds and differing database sizes. With this monitoring it's additionally shown that it's time-efficient additionally distance efficient. We also detected the more effective distance optimization is for whenever features with distinct uniqueness among features

6. Conclusion & future scope

This paper provides a fresh IBQ test for processing of numerous features using place representation procedure. The collections are utilized for processing of IBQ by running established intersection operation between adapting places just. In this procedure consistently, a set of columns chosen using low amount of sets having more set dimensions. Employing this strategy, it owns less space compared to the raw information. Within this paper, we exploited the land of bitmap index and indicator places are reflected in sets. This instrument assesses iceberg outcomes for 2 characteristics, multiple characteristics using greedy strategy and numerous characteristics with dynamic programming principle strategy it's also noted that this suggested system time efficient and space efficient. It's a space effective whenever feature uniqueness is arbitrary.

The experimental results will be shown and found that IBQ test time for 2 characteristics, multiple characteristics using greedy strategy and numerous characteristics with dynamic programming principle strategy better than present methods. The future study management of this work could be centered on memory resident region will be decreased using large data strategy using map decrease frame with HIVE infrastructure instrument, Which Might further maximizes the implementation time to Assess iceberg queries for big database.

References

[1] Bin He, Hu i-I Hsiao, Ziyang Liu, Yu Huang and Yi Chen, “Efficient Iceberg Query Evaluation Using Compressed Bit map Index”, IEEE Transactions On Knowledge and Data Engineering, vol 24, issue 9, sept 2011, pp.1570-1589.

[2] R. Agrawal, T. Imielinski, and A.N. Swami, “Mining Association Rules between Sets of Items in Large Databases,” Proc. A CM SIGM OD Int’l Conf. Management of Data, pp. 207-216, 1993.

[3] Shneiderman. B and Plaisant. C, “The Eight Golden Rules of Interface Design”, Designing the User Interface: Strategies for Effective Human Computer Interaction: Fifth Edit ion, Addison- Wesley Publ. Co., MA, Ch.2,Sec 2.3.4, 2010

[4] http://wiki.up.ac.za/ index.php/HCI_research_topics

[5] Gossen, Tatiana, Marcus Nitsche, and Andreas Nurnberger. "Knowledge journey: A web search interface for young users." Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval. A CM, 2012.

[6] Dayal, Nab ilKamel, Gunter Schlageter,andKyu-Young Whang, editors,VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt, pages 329–338. Morgan Kaufmann, 2000.

[7] www.w3schools.com/php/default.asp

[8] Scott W. Ambler "User Interface Design Tips, Techniques, and Principles."

[9] R.C.C. Guntupalli, "User Interface Design-Methods and Qualities of Good User Interface Design", M.Thesis, University West, Sweden, Jun. 2008.

[10] Alan Dix, Janet Finalay, Gregory D. Abowd, Russell Beale "The Hu man, The Co mputer, The Interaction", Human Computer Interaction, 3rd edition, England,Pearson Publishers, Ch.1-3, pp.11-123, 2004

[11] Rao V.C.S, Sammulal P, "Efficient Iceberg Query Evaluation Using Set Representation", IEEE Explorer, INDICON, Pune, INDIA, ISBN 978-1-4799-5362-2, December 2014, pp.1-5

[12] Rao, P. Sammulal. "Efficient Iceberg Query Evaluation on Multiple attributes Using Set Representation", International Journal of Control Theory and Appliations, ISSN 0674-5572, Volume 10, Number 6, 2017.

[13] V. Chandra shekhar Rao, P. Sammulal "Efficient Iceberg Query Evaluation on Multiple Attributes using Dynamic Programming Approach:, Journal of Advanced Research in Dynamical and Control Systems, ISSN 1943-023X, pages 1703-1715, Issue-10,2018