TEMPLATE FOR ASSIGNMENT #4 REPORT

Student Name and CCID:

A. __JASPREET KAUR SOHAL__________ (Name) jsohal (CCID)
B. __RAJ SHREYAS PENUKONDA______ (Name) penukond __ (CCID)
C. ________________________________ (Name) ___________ (CCID)

By submitting this assignment the students named above confirm that they have worked on it themselves without any help by other people. If any external resources were used please state which ones and how they were used:

PART 1

Task A (no index):

Cardinality of Table Parts	**Average Processing time for index free Q1 (ms)**
100	3.997623920440674
1000	5.222156047821045
10,000	8.235352039337158
100,000	276.75642013549805
1,000,000	2860.9025597572327

Cardinality of Table Parts	**Average Processing time for index free Q2 (ms)**
100	4.017789363861084
1000	6.154077053070068
10,000	8.115711212158203
100,000	149.4702434539795
1,000,000	1896.1724781990051

Task B:

Compare, contrast and explain the trends observable in both tables above (Task A)
We notice that the average query time required for answering query 2 for each version of table Parts is a **little higher** than average query time required for answering query 1. As NP increases, the difference in time (in ms) also increases. The space taken by the databases are: 1) A4V100 - 5 KB 2) A4V1k - 28 KB 3) A4V10k - 254 KB 4) A4V100k - 2562 KB 5) A4V1M - 25764KB

Task C (using index):

Cardinality of Table Parts	**Average Processing time for indexed Q1 (ms)**
100	2.7170729637145996
1000	6.113598346710205
10,000	8.360369205474854
100,000	46.86593770980835
1,000,000	460.6753134727478

Cardinality of Table Parts	**Average Processing time for indexed Q2 (ms)**
100	2.867729663848877
1000	3.85770320892334
10,000	8.89223575592041
100,000	46.46853446960449
1,000,000	454.20751094818115

Task D:

Compare, contrast and explain the trends observable in both tables above (Task C)

After creating an index for table Parts, we notice that the average query time required for answering query 2 for each version of table Parts is a **little higher** than average query time required for answering query 1.
As NP increases, the difference in time (in ms) also increases.
The space taken by the databases are:

6) A4V100- 8 KB
7) A4V1k - 47 KB
8) A4V10k - 434KB
9) A4V100k-4351KB
10) A4V1M- 43640KB

Task E:

Compare, contrast and explain the trends observed in Task D to the trends observed in Task B. Discuss the cost-benefit of the index space cost and query performance.

Although the average processing time for the cardinality 100, 1000, 10000 were approximately equal with or without the index, there is **significant improvement** in time for the 100000 and 1,000,000 data point files.
For the file containing 1,000,000 data points, the extra space the index took was around ~18000KB which is beneficial as the time decreases significantly for 100K and 1M.

PART 2

Task F (no index):

Cardinality of Table Parts	**Average Processing time for index-free Q3 (ms)**
100	1.643843650817871
1000	2.750225067138672
10,000	3.8777995109558105
100,000	116.5021014213562
1,000,000	1508.620638847351

Task G (using index):

Cardinality of Table Parts	**Average Processing time for indexed Q3 (ms)**
100	1.548168659210205
1000	1.8506073951721191
10,000	1.8785858154296875
100,000	2.109372615814209
1,000,000	1013.3221459388733

Task H:

Compare, contrast and explain the trends observed in Task F to the trends observed in Task G. Discuss the cost-benefit of the index space cost and query performance.
We notice that the average query time required for answering the query after creating an index for MadeIn is **significantly lower** than average query time required for answering queries without an index. Hence, we notice an **improvement.** For the file containing 1,000,000 (1 Million) data points, the extra space the index took was around 14000 kb which is beneficial as the time **decreases significantly..**

PART 3

Task I (no index):

Cardinality of Table Parts	Average Processing time for no-index Q4 (ms)
100	0.33833980560302734
1000	0.7647466659545898
10,000	2.250354290008545
100,000	56.11529588699341
1,000,000	1127.1074843406677

Task J:

Define an index that you believe will optimize Q4 and explain why you think so.
An index on both “madeIn” and “partPrice” will optimize the query in question 4. This is because the DBMS iterates through all the rows to find the rows corresponding to the country, the index on “madeIn” will speed this up. Then it has to iterate through all the prices in the countries, the index on “partPrice” will speed this up.

Task K (using index):

Cardinality of Table Parts	Average Processing time for indexed Q4 (ms)
100	0.6828212738037109
1000	1.0190463066101074
10,000	2.715468406677246
100,000	25.406317710876465
1,000,000	277.35379219055176

Task L:

Compare, contrast and explain the trends observed in Task K to the trends observed in Task I. Discuss the cost-benefit of the index space cost and query performance.
Although the average processing time for the cardinality 100, 1000, 10000 were approximately equal with or without the index, there is **significant improvement** in time for the 100000 and 1,000,000 data point files. For the file containing 1,000,000 data points, the extra space the index took was around 16000kb **which is beneficial as the time decreases significantly (by over 75%).**

PART 4

Task M (no index):

Cardinality of Table Parts	**Average Processing time for index-free Q5 (ms)**
100	1.2431883811950684
1000	2.267594337463379
10,000	5.4315876960754395
100,000	168.35609197616577
1,000,000	2794.5112776756287

Task N (no index):

Cardinality of Table Parts	**Average Processing time for index-free Q6 (ms)**
100	1.2390875816345215
1000	3.558213710784912
10,000	8.630115985870361
100,000	181.27856969833374
1,000,000	3728.606264591217

Task O:

Compare, contrast and explain the trends observed in Task M to the trends observed in Task N

We notice that the average query time required for answering query 2 for each version of table Parts is **higher** than average query time required for answering query 1.
As NP increases, the difference in time (in ms) also increases.

Task P:

Define an index that you believe will optimize Q6 and explain why you think so

An index on needsParts will optimize the query in question 6. This is because for each part in the table, the DBMS iterates through all the values of needsParts. Since the part number is a primary key, we should create an index on needsParts to make this iteration faster.

Task Q (with index):

Cardinality of Table Parts	Average Processing time for indexed Q6 (ms)
100	1.4150619506835938
1000	1.749734878540039
10,000	5.369746685028076
100,000	39.49692487716675
1,000,000	2327.9979968070984

Task R:

Compare, contrast and explain the trends observed in Task N to the trends observed in Task Q. Discuss the cost-benefit of the index space cost and query performance.

Although the average processing time for the cardinality 100, 1000, 10000 were decreased by a few milliseconds, there is **significant improvement** in time for the 100000 and 1,000,000 data point files. For the file containing 1,000,000 data points, the extra space the index took was around 17800 kb which is beneficial but we get around the same time using “EXISTS”.