

Προηγμένα Θέματα Βάσεων Δεδομένων

Εξαμηνιαία Εργασία

Αριθμός Ομάδας: 8

Ονοματεπώνυμο – Αριθμός Μητρώου:

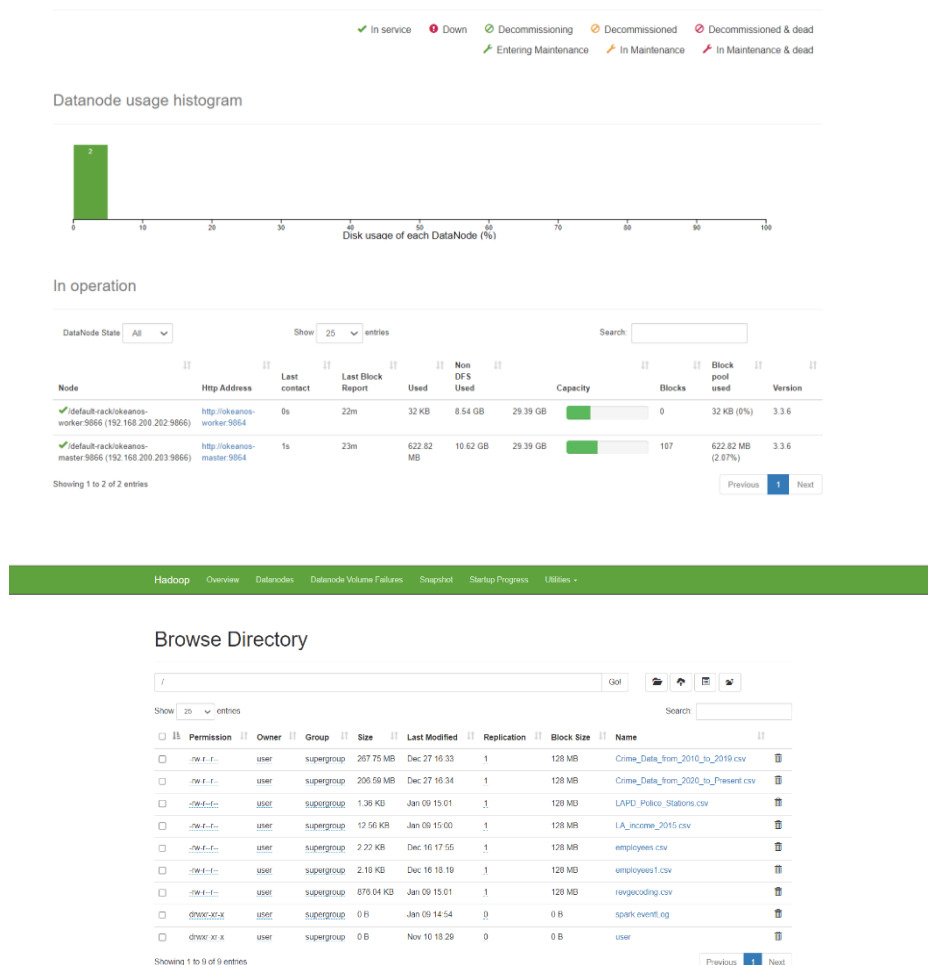
- Αριστείδης Τζιαπούρας – 03119703
- Βίκτωρας Γιαννάκης – 03119707

Link Github: <https://github.com/ntua-el19707/advancedDatabasedSystem/tree/main>


Ζητούμενο 1

Έχοντας λάβει τους απαραίτητους πόρους από τον Ωκεανό και ακολουθώντας τα βήματα για την εγκατάσταση και μορφοποίηση του περιβάλλοντος εργασίας, οι web εφαρμογές των HDFS , YARN , Spark History Server είναι πλέον προσβάσιμες και διαθέσιμες. Στο περιβάλλον εργασίας υπάρχουν 2 κόμβοι, ο Master και ο Worker

HDFS



YARN



Cluster

About Nodes Node Labels Applications

NEW NEW SAVING SUBMITTED ACCEPTED RUNNING FINISHED FAILED KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	0	Apps Pending	0	Apps Running	1	Apps Completed	0	Containers Running		Used Resources		Total Resources			
												<memory:0 B, vCores:0>		<memory:12 GB, vCores:16>	

Cluster Nodes Metrics

Active Nodes	0	Decommissioning Nodes	0	Decommissioned Nodes	0	Lost Nodes	0	Unhealthy Nodes	0
--------------	---	-----------------------	---	----------------------	---	------------	---	-----------------	---

Scheduler Metrics


Scheduler Type	Capacity Scheduler	Scheduling Resource Type	[memory-mb (unit-M), vcores]	Minimum Allocation	<memory:128, vCores:1>	Maximum Allocation	<memory:6144, vCores:4>	0
----------------	--------------------	--------------------------	------------------------------	--------------------	------------------------	--------------------	-------------------------	---

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU Vcores	Allocated Memory MB
application_1704803688650_0001	user	DataFrame Execution	SPARK		default	0	Tue Jan 9 14:53:01 +0200 2024	Tue Jan 9 14:53:04 +0200 2024	Tue Jan 9 14:54:18 +0200 2024	FINISHED	SUCCEEDED	N/A	N/A	N/A

Showing 1 to 1 of 1 entries

Spark History Server



History Server

Event log directory: hdfs://okeanos-master:54310/spark/eventLog

Last updated: 2024-01-09 14:57:09

Client local time zone: Asia/Nicosia

Show 20 entries

Search:

Version	App ID	App Name	Started	Completed	Duration	Spark User	Last Updated	Event Log
3.5.0	application_1704803688650_0001	DataFrame Execution	2024-01-09 14:52:22	2024-01-09 14:54:17	1.9 min	user	2024-01-09 14:54:18	Download
3.5.0	application_1704287765830_0037	RDD Execution	2024-01-03 17:14:55	2024-01-03 17:16:12	1.3 min	user	2024-01-03 17:16:13	Download
3.5.0	application_1704287765830_0036	DataFrame Execution	2024-01-03 17:10:50	2024-01-03 17:11:54	1.1 min	user	2024-01-03 17:11:54	Download
3.5.0	application_1704287765830_0035	DataFrame Execution	2024-01-03 17:05:11	2024-01-03 17:07:20	2.1 min	user	2024-01-03 17:07:23	Download
3.5.0	application_1704287765830_0034	DataFrame Execution	2024-01-03 17:01:20	2024-01-03 17:01:55	34 s	user	2024-01-03 17:01:55	Download
3.5.0	application_1704287765830_0033	DataFrame Execution	2024-01-03 16:56:24	2024-01-03 16:57:38	1.2 min	user	2024-01-03 16:57:38	Download
3.5.0	application_1704287765830_0032	DataFrame Execution	2024-01-03 16:48:39	2024-01-03 16:49:32	53 s	user	2024-01-03 16:49:32	Download

Ζητούμενο 2

Το βασικό σύνολο δεδομένων αποτελείται από 2 μικρότερες βάσεις οι οποίες παρουσιάζουν τα εγκλήματα που έγιναν από το 2010 – 2019 και 2020 –Σήμερα αντίστοιχα τα οποία βρίσκονται εδώ:

- <https://catalog.data.gov/dataset/crime-data-from-2010-to-2019>
- <https://catalog.data.gov/dataset/crime-data-from-2020-to-present>

Στο νέο Dataframe έχει γίνει συνένωση των 2 συνόλων και έχει τα εξής χαρακτηριστικά:

Αριθμός Γραμμών: 2925275

Τύπος Δεδομένων:

```
Number of Rows: 2925275
root
|-- DR_NO: string (nullable = true)
|-- Date Rptd: timestamp (nullable = true)
|-- Date OCC: timestamp (nullable = true)
|-- TIME OCC: string (nullable = true)
|-- AREA : string (nullable = true)
|-- AREA NAME: string (nullable = true)
|-- Rpt Dist No: string (nullable = true)
|-- Part 1-2: string (nullable = true)
|-- Crm Cd: string (nullable = true)
|-- Crm Cd Desc: string (nullable = true)
|-- Mocodes: string (nullable = true)
|-- Vict Age: integer (nullable = true)
|-- Vict Sex: string (nullable = true)
|-- Vict Descent: string (nullable = true)
|-- Premis Cd: string (nullable = true)
|-- Premis Desc: string (nullable = true)
|-- Weapon Used Cd: string (nullable = true)
|-- Weapon Desc: string (nullable = true)
|-- Status: string (nullable = true)
|-- Status Desc: string (nullable = true)
|-- Crm Cd 1: string (nullable = true)
|-- Crm Cd 2: string (nullable = true)
|-- Crm Cd 3: string (nullable = true)
|-- Crm Cd 4: string (nullable = true)
|-- LOCATION: string (nullable = true)
|-- Cross Street: string (nullable = true)
|-- LAT: double (nullable = true)
|-- LON: double (nullable = true)
```

Ζητούμενο 3

Στο Query 1 ζητείται η εύρεση των 3 μηνών με τον πιο ψηλό αριθμό καταγεγραμμένων εγκλημάτων κάθε χρόνο και η θέση του κάθε μήνα στην κατάταξη εκείνου του έτους.

Η υλοποίηση του Query 1, έγινε με 2 διαφορετικούς τρόπους, ο πρώτος με DataFrame και ο δεύτερος με SQL APIs. Και στις 2 περιπτώσεις χρησιμοποιήθηκαν 4 Spark Executors και παρακάτω παρουσιάζονται τα αποτελέσματα που προκύπτουν από την εκτέλεση αλλά και ο χρόνος εκτέλεσης των 2 υλοποιήσεων.

Μέτρηση	DataFrame 4 Executors	DataFrame 4 Executors	SQL 4 Executors	SQL 4 Executors
1	39.67sec	1.3min	36.01sec	1.2min
2	36.50sec	1.2min	38.29sec	1.4min
3	34.93sec	1.1min	39.02sec	1.2min
4	34.65sec	1.1min	38.32sec	1.2min
5	35.25sec	1.2min	37.87sec	1.3min
6	34.79sec	1.1min	39.65sec	1.3min
7	35.02sec	1.1min	39.12sec	1.3min
8	34.65sec	1.2min	38.12sec	1.2min
Average Time	35.68sec	1.1625min =69.75sec	38.30sec	1.26min = 75.75sec

Παρατηρούμε ότι ο μέσος χρόνος εκτέλεσης του SQL API είναι παρόμοιος συγκριτικά με τον χρόνο εκτέλεσης του DataFrame.

Αποτελέσματα για Query 1

Year	Month	Total Crimes	#
2010	1	19515	1
2010	3	18131	2
2010	7	17856	3
2011	1	18133	1
2011	7	17283	2
2011	10	17034	3
2012	1	17943	1
2012	8	17661	2
2012	5	17502	3
2013	8	17440	1
2013	1	16820	2
2013	7	16644	3
2014	7	13584	1
2014	10	13433	2
2014	8	13356	3
2015	10	19218	1
2015	8	19011	2
2015	7	18709	3
2016	10	19659	1
2016	8	19490	2
2016	7	19448	3
2017	10	20431	1
2017	7	20192	2
2017	1	19833	3
2018	5	19970	1

2017	1	19833	3
2018	5	19970	1
2018	7	19874	2
2018	8	19761	3
2019	7	19121	1
2019	8	18979	2
2019	3	18854	3
2020	1	18495	1
2020	2	17255	2
2020	5	17204	3
2021	10	19303	1
2021	7	18659	2
2021	8	18374	3
2022	5	20416	1
2022	10	20269	2
2022	6	20198	3
2023	8	19712	1
2023	7	19673	2
2023	1	19627	3

Ζητούμενο 4

Στο Query 2 ζητείται να ταξινομηθούν τα 4 διαφορετικά τμήματα της ημέρας(Πρωί, Απόγευμα,Βράδυ,Νύχτα) ανάλογα με τον αριθμό των καταγεγραμμένων εγκλημάτων που συνέβησαν στον δρόμο(STREET).

Η υλοποίηση του Query 2 έγινε με 3 διαφορετικούς τρόπους, ο πρώτος με DataFrame, ο δεύτερος με SQL API και ο τρίτος με RDD API. Και στις 3 περιπτώσεις χρησιμοποιήθηκαν 4 Spark Executors και παρακάτω παρουσιάζονται τα αποτελέσματα που προκύπτουν από την εκτέλεση αλλά και ο χρόνος εκτέλεσης των 3 υλοποιήσεων.

Μέτρηση	DataFrame 4 Executors	DataFrame 4 Executors	SQL 4 Executors	SQL 4 Executors	RDD 4 Executors	RDD 4 Executors
1	37.29sec	1.7min	38.86sec	1.8min	82.70sec	2.4min
2	35.64sec	1.6min	37.09sec	1.7min	86.25sec	2.5min
3	40.32sec	1.5min	28.49sec	1.4min	78.99sec	2.1min
4	41.49sec	1.8min	30.54sec	1.9min	93.37sec	2.4min
5	37.21sec	1.8min	41.57sec	1.9min	83.90sec	2.4min
6	37.12sec	1.7min	32.82sec	1.7min	75.64sec	2.3min
7	36.06sec	1.8min	26.06sec	1.1min	89.18sec	2.5min
8	41.75sec	1.8min	36.51sec	1.7min	85.25sec	2.4min
Average Time	38.36sec	1.7125min =102.5sec	33.99sec	1.65min =99sec	84.41sec	2.375min =142.5sec

Αυτό που παρατηρούμε από τις μετρήσεις είναι πως και πάλι η υλοποίηση με DataFrame και SQL δίνουν παρόμοια αποτελέσματα ως προς τον χρόνο εκτέλεσης ενώ αντίθετα, η υλοποίηση με RDD είναι πολύ πιο αργή, καθώς χρειάζεται διπλάσιο χρόνο συγκριτικά με τα άλλα.

Αποτελέσματα για Query 2

```
+-----+
|TIME_OF_THE_DAY| count |
+-----+
|NIXTA| 232216 |
|VRADY| 182771 |
|APOGEVMA| 144329 |
|PRWI| 120649 |
+-----+
```

Ζητούμενο 5

Στο Query 3 ζητείται να βρεθεί σε φθίνουσα σειρά η καταγωγή των καταγεγραμμένων θυμάτων στο Los Angeles το 2015 σε 6 περιοχές, τις 3 με το ψηλότερο και στις 3 με το χαμηλότερο εισόδημα ανά νοικοκυριό

Για αυτό το Query , εκτός από το βασικό Data-Set: Los Angeles Crime Data, χρειάζονται επίσης τα δευτερεύοντα Dataset για το Median Household Income by ZIP Code για το 2015 και το revgecoding.csv για αντιστοίχιση ενός ζεύγους συντεταγμένων σε μια διεύθυνση, τα οποία υπάρχουν εδώ:

- <http://www.dblab.ece.ntua.gr/files/classes/data.tar.gz>

Η υλοποίηση του Query 3 έγινε με SQL API. Η εκτέλεση έγινε 3 φορές με διαφορετικό αριθμό από Spark Executors, με 2, με 3 και με 4 αντίστοιχα. Παρακάτω παρουσιάζονται τα αποτελέσματα που προκύπτουν από την εκτέλεση αλλά και ο μέσος χρόνος εκτέλεσης των 3 μετρήσεων.

Μέτρηση	SQL 2 Executors	SQL 2 Executors	SQL 3 Executors	SQL 3 Executors	SQL 4 Executors	SQL 4 Executors
1	37.22sec	1.3min	34.21sec	1.2min	32.55sec	1.2min
2	37.19sec	1.1min	33.53sec	1.2min	33.10sec	1.2min
3	36.75sec	1.2min	31.73sec	1.1min	33.24sec	1.2min
4	38.58sec	1.3min	46.06sec	1.4min	34.21sec	1.3min
5	34.97sec	1.2min	35.75sec	1.3min	33.54sec	1.1min
6	35.38sec	1.3min	34.67sec	1.2min	33.59sec	1.1min
7	35.28sec	1.2min	35.25sec	1.2min	34.30sec	1.3min
8	37.43sec	1.2min	35.26sec	1.3min	34.94sec	1.3mn
Average Time	36.6sec	1.2375min= 74.25sec	35.81sec	1.2375min= 74.25sec	33.68sec	1.2125min= 72.75sec

Παρατηρούμε ότι όσο αυξάνεται ο αριθμός των executors, ο χρόνος εκτέλεσης του Query μειώνεται ελάχιστα αλλά σε καμία περίπτωση δεν είναι ανάλογος του αριθμού των Executors. Δηλαδή με διπλάσιους Executors, από 2 σε 4 Spark Executors, ο χρόνος εκτέλεσης μειώθηκε 3 Seconds

Αποτέλεσμα Εκτέλεσης QUERY 3

Zip Code	#	Vict Descent	victim_category
90013	8457	B	Black
90013	6728	H	Hispanic/Latin/Me...
90013	5259	W	White
90013	2348	O	Other
90013	897	A	Other Asian
90013	768	X	Unknown
90013	96	K	Korean
90013	89	C	Chinese
90013	67	F	Filipino
90013	52	J	Japanese
90013	18	I	American Indian/A...
90013	13	V	Vietnamese
90013	3	Z	Asian Indian
90013	2	S	Samoan
90013	2	U	Hawaiian
90013	2	P	Pacific Islander
90013	1	L	Laotian
90013	1	G	Guamanian
90021	7263	H	Hispanic/Latin/Me...
90021	3536	W	White
90021	3249	B	Black
90021	1716	O	Other
90021	753	X	Unknown
90021	452	A	Other Asian
90021	93	K	Korean
90021	52	C	Chinese
90021	22	F	Filipino
90021	16	I	American Indian/A...
90021	10	J	Japanese
90021	10	V	Vietnamese
90021	4	P	Pacific Islander
90021	4	Z	Asian Indian
90021	1	G	Guamanian
90021	1	U	Hawaiian
90058	4665	H	Hispanic/Latin/Me...
90058	1322	B	Black
90058	279	W	White

90058	1322	B	Black
90058	279	W	White
90058	197	O	Other
90058	106	X	Unknown
90058	42	A	Other Asian
90058	6	K	Korean
90058	3	D	Cambodian
90058	2	F	Filipino
90058	2	C	Chinese
90058	2	I	American Indian/A...
90077	785	W	White
90077	247	O	Other
90077	100	H	Hispanic/Latin/Me...
90077	54	B	Black
90077	45	X	Unknown
90077	43	A	Other Asian
90077	2	C	Chinese
90077	1	K	Korean
90272	1758	W	White
90272	445	O	Other
90272	224	H	Hispanic/Latin/Me...
90272	211	X	Unknown
90272	84	B	Black
90272	65	A	Other Asian
90272	6	C	Chinese
90272	3	F	Filipino
90272	2	K	Korean
90272	1	I	American Indian/A...
91436	3580	W	White
91436	1551	O	Other
91436	641	H	Hispanic/Latin/Me...
91436	295	B	Black
91436	139	A	Other Asian
91436	44	X	Unknown
91436	5	F	Filipino
91436	4	K	Korean
91436	4	C	Chinese
91436	2	U	Hawaiian
91436	2	P	Pacific Islander
91436	2	I	American Indian/A...
91436	1	J	Japanese

Ζητούμενο 6

Στο Query 4 ζητείται να βρεθεί:

- 1) Α) Ο αριθμός των καταγεγραμμένων εγκλημάτων που έγιναν με πυροβόλο όπλο (1xx) ανά χρονιά, μαζί με την μέση απόσταση του αστυνομικού τμήματος που ανέλαβε την έρευνα.
Β) Ο αριθμός τέτοιων εγκλημάτων που ανέλαβε ο κάθε αστυνομικός σταθμός ταξινομημένο σε φθίνουσα σειρά, μαζί με την μέση απόσταση των εγκλημάτων
- 2) Α) Ο αριθμός των καταγεγραμμένων εγκλημάτων που έγιναν με πυροβόλο όπλο (1xx) ανά χρονιά, μαζί με την μέση απόσταση του πλησιέστερου αστυνομικού τμήματος
Β) Ο αριθμός τέτοιων εγκλημάτων που θα αναλάμβανε ο κάθε πλησιέστερος αστυνομικός σταθμός ταξινομημένο σε φθίνουσα σειρά, μαζί με την μέση απόσταση των εγκλημάτων

Για αυτό το Query , εκτός από το βασικό Data-Set: Los Angeles Crime Data, χρειάζονται επίσης το δευτερεύον Dataset LA Police Stations όπου βρίσκονται οι συντεταγμένες του κάθε αστυνομικού σταθμού

- <https://geohub.lacity.org/datasets/lahub::lapd-police-stations/explore>

Η υλοποίηση του Query 4 έγινε με SQL API. Η εκτέλεση έγινε 3 φορές με διαφορετικό αριθμό από Spark Executors, με 2, με 3 και με 4 αντίστοιχα. Παρακάτω παρουσιάζονται τα αποτελέσματα που προκύπτουν από την εκτέλεση αλλά και ο χρόνος εκτέλεσης των 3 μετρήσεων.

Query 4.1

Αποτέλεσμα Εκτέλεσης QUERY 3

year	average_distance	count
2010	16.656527733269883	5304
2011	17.130008356255697	4617
2012	17.82669250762691	4121
2013	18.17947340222288	3787
2014	11.367572520663321	2457
2015	18.399574332889284	4407
2016	17.99578778201601	5002
2017	18.098020611682575	5044
2018	17.71577819582119	4632
2019	17.712188784071895	4574
2020	18.394293287173173	5496
2021	18.12688112902872	5951
2022	18.00779823624805	6058
2023	18.13320450561431	5309

division	average_distance	count
RAMPART	9.385629843529065	16522
TOPANGA	43.892796636535586	11804
HOLLYWOOD	11.942783822408574	9594
FOOTHILL	5.347552660899223	4147
DEVONSHIRE	30.696743873501077	3963
OLYMPIC	8.525161831416845	3839
VAN NUYS	9.603534433688283	3662
NORTH HOLLYWOOD	2.7214117194048204	3335
WILSHIRE	22.992561218999324	2778
NORTHEAST	21.880865669863763	2639
WEST VALLEY	6.781817019999322	2378
MISSION	14.854213071301885	2098

Μέτρηση	SQL 2 Executors	SQL 2 Executors	SQL 3 Executors	SQL 3 Executors	SQL 4 Executors	SQL 4 Executors
1	54.81sec	1.8min	56.50sec	1.9min	58.46sec	2.1min
2	58.39sec	1.8min	54.82sec	1.8min	53.91sec	2.0min
3	59.76sec	2.1min	52.99sec	1.7min	61.18sec	1.9min
4	56.55sec	1.8min	57.19sec	1.9min	57.28sec	2.1min
5	58.30sec	1.9min	64.05sec	2.1min	56.04sec	1.9min
Average Time	57.56sec	1.88min =112.8sec	57.11sec	1.88min =112.8sec	57.37sec	2.00min =120 sec

Σε αυτή την περίπτωση δεν βλέπουμε οποιαδήποτε βελτίωση του χρόνου εκτέλεσης ανάλογα με τον αριθμό των Executors

Query 4.2

year	average_distance	count
2010	2.434235131073048	8212
2011	2.46100507843099	7232
2012	2.5055255743371387	6532
2013	2.4555437568989595	5838
2014	2.3915472779761844	4586
2015	2.3872613200248236	6763
2016	2.4281950357376467	8100
2017	2.391618932774625	7786
2018	2.4082079737438598	7413
2019	2.4294088109777405	7129
2020	2.383615837920021	8487
2021	2.352716378850509	9745
2022	2.3120803748064067	10025
2023	2.270586785715524	8583

division	average_distance	count
77TH STREET	1.720516809530909	13295
SOUTHWEST	2.279758145034117	11183
SOUTHEAST	2.213211553395432	10859
NEWTON	1.5691528311018892	7142
WILSHIRE	2.445609338096738	6232
HOLLENBECK	2.638149080357802	6150
HOLLYWOOD	2.003464451832188	5317
HARBOR	3.8995576321257572	5299
OLYMPIC	1.663148125434012	5080
RAMPART	1.3967648289758168	4679
VAN NUYS	2.9533148232235624	4585
FOOTHILL	3.6007589003843794	4367
CENTRAL	1.0197110483325327	3562
NORTH HOLLYWOOD	2.7302047146999158	3304
NORTHEAST	3.754395584281293	3088
WEST VALLEY	2.781373619433168	2701
MISSION	3.801258877877068	2654
PACIFIC	3.70082473460895	2515
TOPANGA	3.0413726498688973	2186
DEVONSHIRE	2.981839678996518	1220
WEST LOS ANGELES	2.769236423329846	1013

Μέτρηση	SQL Execution Time	SQL Spark History
1	567.59sec	10min
2	550.33sec	12min
Average Time	558.96sec	11min

Ζητούμενο 7

Για τα Query 3 και 4α, χρησιμοποιούμε τα διαφορετικά είδη Join που υπάρχουν. Οι χρόνοι εκτελέσεων του κάθε τρόπου για κάθε Query εμφανίζονται πιο κάτω

Αυτά που χρησιμοποιήθηκαν είναι τα:

BROADCAST, MERGE, SHUFFLE_HASH, SHUFFLE_REPLICATE_NL

Επίσης όλες οι μετρήσεις έγιναν με 4 Spark Executors

Query 3

Μέτρηση	SQL 4 Executors	SQL 4 Executors BROADCAST	SQL 4 Executors MERGE	SQL 4 Executors SHUFFLE_HASH	SQL 4 Executors SHUFFLE_REPLICATE_NL
1	44.40sec	52.49sec	47.94sec	67.20sec	3768sec = 62.8min

Μέτρηση	SQL 4 Executors	SQL 4 Executors BROADCAST	SQL 4 Executors MERGE	SQL 4 Executors SHUFFLE_HASH	SQL 4 Executors SHUFFLE_REPLICATE_NL
1	1.7min	2min	1.8min	2.4min	1.1h

Παρατηρούμε ότι ο χρόνος εκτέλεσης με BROADCAST και MERGE JOIN είναι πανομοιότυπος. Το SHUFFLE_HASH χρειάζεται λίγο περισσότερο χρόνο και τέλος για το SHUFFLE_REPLICATE_NL ο χρόνος εκτέλεσης είναι τεράστιος.

BROADCAST

```

= Physical Plan ==\nAdaptiveSparkPlan (38)\n+- Project (37)\n  +- Sort (36)\n  +- Exchange (35)\n  +- Project (34)\n  +- BroadcastHashJoin Inner BuildLeft (33)\n    :- BroadcastExchange (12)\n      +- Union (11)\n        :- Filter (5)\n        :- Project (3)\n        :- Scan csv (1)\n        :- TakeOrderedAndProject (4)\n      +- Filter (2)\n      +- Project (8)\n      +- Scan csv (6)\n      +- HashAggregate (32)\n      +- Exchange (31)\n      +- HashAggregate (30)\n      +- Project (29)\n      +- BroadcastHashJoin Inner BuildRight (28)\n      :- HashAggregate (24)\n      +- Exchange (23)\n      +- HashAggregate (22)\n      +- Union (21)\n      :- Filter (14)\n      +- Scan csv (13)\n      +- Filter (18)\n      +- Filter (16)\n      +- Scan csv (15)\n      +- Filter (20)\n      +- Scan csv (19)\n      +- BroadcastExchange (27)\n      +- Filter (26)\n      +- Scan csv (25)\n      +- Scan csv (1)\n      +- Scan csv \nOutput [3]: [Zip Code#373, Community#374, Estimated Median Income#375]\nBatched: false\nLocation: InMemoryFileIndex [hdfs://oceanos-master:54310/LA_income_2015.csv]\nPushedFilters: [IsNotNull(Community), StringContains(Community, Los Angeles)]\nReadSchema: struct<Zip Code:string,Community:string,Estimated Median Income:string>\n(2) Filter\nInput [3]: [Zip Code#373, Community#374, Estimated Median

```

MERGE

```

[== Physical Plan ==\nAdaptiveSparkPlan (44)\n+- Project (43)\n  +- Sort (42)\n    +- Exchange (41)\n      +- Project (40)\n        +- SortMergeJoin Inner (39)\n          :- Sort (13)\n            : +- Exchange (12)\n              : +- Union (11)\n                :- Filter (5)\n                  : +- Tak
eOrderedAndProject (4)\n                  : +- Project (3)\n                    : +- Filter (2)\n                      : +- Scan csv (1)\n                        +- Filter (7)\n                          +- Filter (10)\n                            +- TakeOrderedAndProject (9)\n                              +- Project (8)\n                                +- Sort (38)\n                                  +- Exchange (37)\n                                    +- HashAggreg
ate (36)\n                                      +- Exchange (35)\n                                        +- HashAggregate (34)\n                                          +- Project (33)\n                                            +- Sort (27)\n                                              +- Exchange (26)\n                                                +- HashAggregate (23)\n                                                  +- Scan csv (16)\n                                                    +- Union (22)\n                                                      +- Filter (17)\n                                                        +- Filter (15)\n                                                          +- Scan csv (14)\n                                                            +- Filter (21)\n                                                              +- Filter (19)\n                                                                +- Scan csv (20)\n                                                                  +- Scan csv (18)\n                                                                    +- Sort (31)\n                                                                      +- Exchange (30)\n                                                                        +- Filter (29)\n                                                                          +- Scan csv (28)\n                                                                            +- Scan csv \nOutput [3]: [Zip Code#373, Community#374, Estimated Median Income#375]\nBatched: false\nLocation: InMemoryFileIndex [hdfs://okeanos-master:54310/LA_income_2015.csv]\nPushedFilters: [IsNotNull(Community), StringContains(Community, Los Angeles)]\nReadSchema: struct<Zip Code:string, Community:string, Estimated Median Income:string>\nInput [3]: [Zip Code#373, Community#374, Estimated Median Income#375]\nCondition: (isNotNull(Community#374) AND Contains(Community#

```

SHUFFLE_HASH

```
|== Physical Plan ==\nAdaptiveSparkPlan (40)\n+- Project (39)\n    +- Sort (38)\n        +- Exchange (37)\n            +- Project (36)\n                +- ShuffledHashJoin Inner BuildLeft (35)\n                    :- Exchange (12)\n                        +- Union (11)\n                            :+- Filter (5)\n                                :+- TakeOrderedAndProject (4)\n                                    :+- Project (3)\n                                        :+- Filter (2)\n                                            :+- Scan csv (1)\n                                                :+- Filter (10)\n                                                    +- Scan csv (6)\n                                                        +- Exchange (34)\n                                                            +- HashAggregate (33)\n                                                                +- Exchange (32)\n                                                                    +- HashAggrate (31)\n                                                                        +- Project (30)\n                                                                            +- ShuffledHashJoin Inner BuildRight (29)\n                                                                                :- Exchange (25)\n                                                                                    :+- HashAggregate (24)\n                                                                                        :+- Exchange (23)\n                                                                                            :+- HashAggregate (22)\n                                                                                                :+- Union (21)\n                                                                                                    :- Filter (14)\n                                                                                                        :+- Scan csv (13)\n                                                                                                            :+- Filter (18)\n                                                                                                                :+- Scan csv (17)\n                                                                                                                    :+- Scan csv (15)\n                                                                                                                        +- Filter (20)\n                                                                                                                            :+- Filter (27)\n                                                                                                                                +- Scan csv (26)\n                                                                                                                                    \\\nScan csv\nOutput [3]: [Zip Code#373, Community#374, Estimated Median Income#375]\nBatched: false\nLocation: InMemoryFileIndex[hdfs://okeanos-master-54310/LA_income_2015.csv]\nPushedFilters: [IsNotNull(Community), StringContains(Community, Los Angeles)]
```

SHUFFLE REPLICATE NL

```

== Physical Plan ==\nAdaptiveSparkPlan (36)\n+- Project (35)\n   +- Sort (34)\n      +- Exchange (33)\n         +- Project (32)\n            +- CartesianProduct Inner (31)\n               +- Union (11)\n                  :- Filter (5)\n                     :- TakeOrderedAndProject (4)\n                        +- Project (3)\n                           +- Filter (2)\n                              +- Scan csv (1)\n                                 +- TakeOrderedAndProject (9)\n                                    +- Project (8)\n                                       +- Filter (7)\n                                          +- Scan csv (6)\n                                             +- HashAggregate (30)\n                                                +- CartesianProduct Inner (26)\n                                                   +- Exchange (29)\n                                                      +- HashAggregate (28)\n                                                         +- Project (27)\n                                                            +- Exchange (22)\n                                                               +- HashAggregate (21)\n                                                                  +- Union (20)\n                                                                     +- Filter (13)\n                                                                        +- Scan csv (12)\n                                                                           +- Filter (15)\n                                                                              +- Scan csv (14)\n                                                                                 +- Filter (19)\n                                                                                    +- Scan csv (17)\n                                                                                       +- Scan csv (16)\n                                                                                          +- Filter (25)\n                                                                                             +- Scan csv (24)\n                                                                                                +- Scan csv (1)\n                                                                                                   +- Scan csv (nOutput [3]: [Zip code#373, Community#374, Estimated Median Income#375])\nBatched: false\nLocation: InMemoryFileIndex [hdfs://okeanos-master:54318/LA_income_2015.csv]\nPushedFilters: [IsNotNull(Community), StringContains

```

Query 4.1

Μέτρηση	SQL 4 Executors	SQL 4 Executors BROADCAST	SQL 4 Executors MERGE	SQL 4 Executors SHUFFLE_HASH	SQL 4 Executors SHUFFLE_REPLICATE_NL
1	60.76sec	62.20sec	54.06sec	60.26sec	64.22sec

Μέτρηση	SQL 4 Executors	SQL 4 Executors BROADCAST	SQL 4 Executors MERGE	SQL 4 Executors SHUFFLE_HASH	SQL 4 Executors SHUFFLE_REPLICATE_NL
1	2.3min	2.0min	2.7min	2.6min	2min

Παρατηρούμε ότι οι όλοι οι χρόνοι κυμαίνονται στα ίδια επίπεδα. Με SHUFFLE_REPLICATE_NL ο χρόνος εκτέλεσης είναι σημαντικά μικρότερος συγκριτικά με το Query 3

BROADCAST

```
=====+
|== Physical Plan ==\nAdaptiveSparkPlan (18)\n+- Project (17)\n  +- BroadcastHashJoin Inner BuildRight (16)\n    :- HashAggregate (12)\n      +- Exchange (11)\n        +- HashAggregate (10)\n          +- Union (9)\n            :- Filter (2)\n              +- Scan csv (1)\n                :- Filter (4)\n                  +- Scan csv (3)\n                    :- Filter (6)\n                      +- Scan csv (5)\n                        +- Filter (8)\n                          +- Scan csv (7)\n                            +- BroadcastExchange (15)\n                              +- Filter (14)\n                                +- Scan csv (13)\n                                  +- Scan csv (1)\n                                    +- Scan csv\n\nOutput [28]: [DR_NO#17, Date Rptd#18, DATE OCC#19, TIME OCC#20, AREA #21, AREA NAME#22, Rpt Dist No#23, Part 1-2#24, Crm Cd#25, Crm Cd Desc#26, Mocodes#27, Vict Age#28, Vict Sex#29, Vict Descent#30, Premis Cd#31, Premis Desc#32, Weapon Used Cd#33, Weapon Desc#34, Status#35, Status Desc#36, Crm Cd 1#37, Crm Cd 2#38, Crm Cd 3#39, Crm Cd 4#40, LOCATION#41, Cross Street#42, LAT#43, LON#44]\n\nBatched: false\nLocation: InMemoryFileIndex [hdfs://okeanos-master:54310/crimesCsv_part1.csv]\n\nPushedFilters: [IsNotNull(Weapon Used Cd),
```

MERGE

```
+
|== Physical Plan ==\nAdaptiveSparkPlan (21)\n+- Project (20)\n  +- SortMergeJoin Inner (19)\n    :- Sort (14)\n      +- Exchange (13)\n        +- HashAggregate (12)\n          +- Exchange (11)\n            +- HashAggregate (10)\n              +- Union (9)\n                :- Filter (2)\n                  +- Scan csv (1)\n                    :- Filter (4)\n                      +- Scan csv (3)\n                        +- Filter (6)\n                          +- Scan csv (5)\n                            +- Filter (8)\n                              +- Scan csv (7)\n                                +- Sort (18)\n                                  +- Exchange (17)\n                                    +- Filter (16)\n                                      +- Scan csv (15)\n                                        +- Scan csv (1)\n                                          +- Scan csv\n\nOutput [28]: [DR_NO#17, Date Rptd#18, DATE OCC#19, TIME OCC#20, AREA #21, AREA NAME#22, Rpt Dist No#23, Part 1-2#24, Crm Cd#25, Crm Cd Desc#26, Mocodes#27, Vict Age#28, Vict Sex#29, Vict Descent#30, Premis Cd#31, Premis Desc#32, Weapon Used Cd#33, Weapon Desc#34, Status#35, Status Desc#36, Crm Cd 1#37, Crm Cd 2#38, Crm Cd 3#39, Crm Cd 4#40, LOCATION#41, Cross Street#42, LAT#43, LON#44]\n\nBatched: false\nLocation: InMemoryFileIndex [hdfs://okeanos-master:54310/crimesCsv_part1.csv]\n\nPushedFilters: [IsNotNull(Weapon Used Cd),
```

SHUFFLE HASH

```
=====+
|== Physical Plan ==\nAdaptiveSparkPlan (19)\n+- Project (18)\n  +- ShuffledHashJoin Inner BuildRight (17)\n    :- Exchange (13)\n      +- HashAggregate (12)\n        +- Exchange (11)\n          +- HashAggregate (10)\n            +- Union (9)\n              :- Filter (2)\n                +- Scan csv (1)\n                  :- Filter (4)\n                    +- Scan csv (3)\n                      +- Filter (6)\n                        +- Scan csv (5)\n                          +- Filter (8)\n                            +- Scan csv (7)\n                              +- Exchange (16)\n                                +- Filter (15)\n                                  +- Scan csv (14)\n                                    +- Scan csv (1)\n                                      +- Scan csv\n\nOutput [28]: [DR_NO#17, Date Rptd#18, DATE OCC#19, TIME OCC#20, AREA #21, AREA NAME#22, Rpt Dist No#23, Part 1-2#24, Crm Cd#25, Crm Cd Desc#26, Mocodes#27, Vict Age#28, Vict Sex#29, Vict Descent#30, Premis Cd#31, Premis Desc#32, Weapon Used Cd#33, Weapon Desc#34, Status#35, Status Desc#36, Crm Cd 1#37, Crm Cd 2#38, Crm Cd 3#39, Crm Cd 4#40, LOCATION#41, Cross Street#42, LAT#43, LON#44]\n\nBatched: false\nLocation: InMemoryFileIndex [hdfs://okeanos-master:54310/crimesCsv_part1.csv]\n\nPushedFilters: [IsNotNull(Weapon Used Cd),
```

SHUFFLE REPLICATE NL

```
=====+
|== Physical Plan ==\nAdaptiveSparkPlan (17)\n+- Project (16)\n  +- CartesianProduct Inner (15)\n    :- HashAggregate (12)\n      +- Exchange (11)\n        +- HashAggregate (10)\n          +- Union (9)\n            :- Filter (2)\n              +- Scan csv (1)\n                :- Filter (4)\n                  +- Scan csv (3)\n                    :- Filter (6)\n                      +- Scan csv (5)\n                        +- Filter (8)\n                          +- Scan csv (7)\n                            +- Filter (14)\n                              +- Scan csv (13)\n                                +- Scan csv (1)\n                                  +- Scan csv\n\nOutput [28]: [DR_NO#17, Date Rptd#18, DATE OCC#19, TIME OCC#20, AREA #21, AREA NAME#22, Rpt Dist No#23, Part 1-2#24, Crm Cd#25, Crm Cd Desc#26, Mocodes#27, Vict Age#28, Vict Sex#29, Vict Descent#30, Premis Cd#31, Premis Desc#32, Weapon Used Cd#33, Weapon Desc#34, Status#35, Status Desc#36, Crm Cd 1#37, Crm Cd 2#38, Crm Cd 3#39, Crm Cd 4#40, LOCATION#41, Cross Street#42, LAT#43, LON#44]\n\nBatched: false\nLocation: InMemoryFileIndex [hdfs://okeanos-master:54310/crimesCsv_part1.csv]\n\nPushedFilters: [IsNotNull(Weapon Used Cd),
```