Προχωρημένα Θέματα Τεχνολογίας και Εφαρμογών Βάσεων Δεδομένων

Επεξεργασία Ερωτήσεων - Άσκηση

Πάνος Βασιλειάδης

pvassil@cs.uoi.gr

Μάιος 2020

www.cs.uoi.gr/~pvassil/courses/db_III/

Άσκηση

Θεωρήστε το σχήμα:

CLIENT (C_ID, C_NAME, C_ADDRESS, C_AGE, C_SPENDING-CLASS)

RESERVES (RS_C_ID, RS_R_ID, RS_START_DATE, RS_END_DATE)

ROOM (R_ID, R_FLOOR, R_VIEW, R_PRICE)

Θεωρήστε

- integer τιμές σε όλα τα αριθμητικά πεδία,
- 200 tuples per page για τον πίνακα CLIENT & 400 tuples per page για τους πίνακες ROOM & RESERVES
- 20 rooms, 1000 clients, 30000 reservations
- 3 floors, 3 types of views, 10 years of start/end dates.
- Age in 0 .. 99, 5 types of spending class.
- Price in 100 up to 250 Euros

1ο ζητούμενο: SQL, Rel.Algebra

CLIENT (C_ID, C_NAME, C_ADDRESS, C_AGE, C_SPENDING-CLASS)

RESERVES (RS_C_ID, RS_R_ID, RS_START_DATE, RS_END_DATE)

ROOM (R_ID, R_FLOOR, R_VIEW, R_PRICE)

Αν σήμερα είναι 2020.05.05, γράψτε σε SQL και σχεσιακή άλγεβρα το ερώτημα: «for each client of SpendingClass = Moderate, find Name, Age, Room View, Duration, Duration * Price, for the reservation that started at most 1 year back (i.e., RS_START >= 2019.05.05)»

1ο ζητούμενο: SQL, Rel.Algebra

CLIENT (C_ID, C_NAME, C_ADDRESS, C_AGE, C_SPENDING-CLASS)

RESERVES (RS_C_ID, RS_R_ID, RS_START_DATE, RS_END_DATE)

ROOM (R_ID, R_FLOOR, R_VIEW, R_PRICE)

Αν σήμερα είναι 2020.05.05, γράψτε σε SQL και σχεσιακή άλγεβρα το ερώτημα: «for each client of SpendingClass = Moderate, find Name, Age, Room View, Duration, Duration * Price, for the reservation that started at most 1 year back (i.e., RS_START >= 2019.05.05)»

SELECT C_NAME, C_AGE, R_VIEW, (RS_END_DATE - RS_START_DATE) AS DURATION, DURATION * PRICE AS PAYMENT_SUM

FROM CLIENTS, RESERVES, ROOM

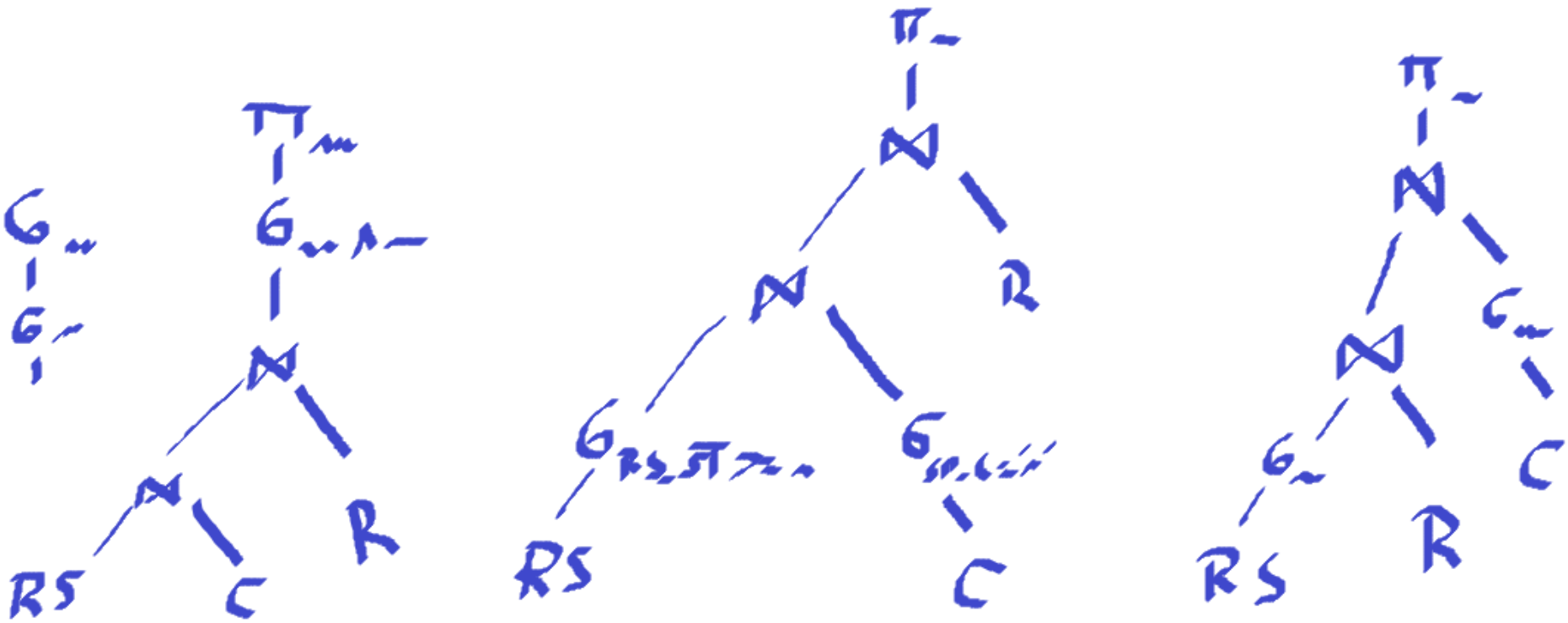
WHERE RS_C_ID = C_ID AND RS_R_ID = R_ID AND C_SPENDING_CLASS = 'MOD' AND RS_STATR_DATE >= 2019.05.05

$\pi_{N,A,V,(EDT-SDT),\,(EDT-SDT)*PR}\,(\sigma_{C\_SC='MOD}\,\{\sigma_{RS\_ST\_DT>=19.5.5}\,[(CLIENT\ |><|_{CID=RS\_CID}\ RESERVES)\ |><|_{RID=RS\_RID}\ ROOM\ ]\})$

2ο ζητούμενο: πλάνα

- Δώστε το τυπικό δέντρο σχεσιακής άλγεβρας που περιγράφει την ερώτηση, όπως θα προέκυπτε με «τυφλή» μετάφραση της SQL σε σχ. άλγεβρα. Δώστε δύο άλλα πλάνα που θα εξέταζε ένας βελτιστοποιητής.

$\pi_{N,A,V,(EDT-SDT),(EDT-SDT)*PR}$ ($\sigma_{C\_SC='MOD}$ {$\sigma_{RS\_ST\_DT>=19.5.5}$ [(CLIENT |><|$_{CID=RS\_CID}$ RESERVES) |><|$_{RID=RS\_RID}$ ROOM]})



3ο ζητούμενο: κόστος

- Διαλέξτε ένα από τα δύο πλάνα αυτά και προσδιορίστε για κάθε τελεστή του δέντρου, τι εκτίμηση θα κάνει ο βελτιστοποιητής για τον αριθμό των εγγραφών που θα παράγει.
- Προσδιορίστε το κόστος του πλάνου που δώσατε, θεωρώντας ότι οι συνδέσεις γίνονται με hash-join?
- Θεωρείστε ότι έχετε διαθέσιμες $\sqrt{|R|}$ buffers ($|R|$ το μέγεθος της μέγιστης σχέσης σε σελίδες, από αυτές που συμμετέχουν στην ερώτηση) σελίδες στη μνήμη. Ζωγραφίστε την ανάθεση στη μνήμη.

Size of intermediate results

C’ = σ(C) : 1/5 * 1000 = 200 tuples => **1 page**

RS’ = σ(RS): 1/10 * 30000 = 3000 tuples => upper(7,5) = **8 pages**

RSC <- RS’ |><| C’: 1/5 * 3000 * 1 = 600 tuples = **5 pages**

RSCR <- RSC |><| R : 600 * 1 = 600 tuples = **6 pages**

Result sizes

200 tuples per page για τον πίνακα CLIENT

400 tuples per page για τους πίνακες ROOM & RESERVES

=>

1/200 + 1/400 => each tuple of RSC takes 3/400 of a page

600 * 3 / 400 = upper(4.5) = **5**

Each final tuple takes1/200 + 1/400 + 1/400 = 4/400 = 1/100 of a page

600 * 1/100 = **6 pages** output

Cost of intermediate results

Με κόκκινο οι διορθώσεις σε σχέση με τη διάλεξη + (α) δείχνω τι θα έβγαινε αν υπήρχε pipelining + (β) τι γίνεται αν το output του 1ου join γίνει pipeline ως input στο 2ο join

C' = σ(C) : 1/5 * 1000 = 200 tuples => **1 page**

RS' = σ(RS): 1/10 * 30000 = 3000 tuples => upper(7,5) = **8 pages**

RSC <- RS' |><| C': 1/5 * 3000 * 1 = 600 tuples = **5 pages**

RSCR <- RSC |><| R : 600 * 1 = 600 tuples = **6 pages**

Without pipelining: had we assumed there is no pipelining, and the output of the first join is stored to disk

A. Hash join: scan(RS) + writeHashed(RS') + scan(C) + writeHashed(C') + RS' + C' -> produces RSC

75 + 8 + 5 + 1 + 8 + 1 = 98

B. Output RSC: 5 pages (assuming there is no pipelining and the output of the 1st join is stored to disk)

C. Hash join: scan(RSC') + writeHashed(RSC') + scan(R) + writeHashed(R) + RSC' + R -> produces final RSCR

5 + 5 + 1 + 1 + 5 + 1 = 18

SUM = 98 + 5 + 18 = 121 pages (assuming no pipelining)

Pipelining: the output of the first join is pipelined into the next join for hashing

A. Hash join: scan(RS) + writeHashed(RS') + scan(C) + writeHashed(C') + RS' + C' -> produces RSC

75 + 8 + 5 + 1 + 8 + 1 = 98

B. Hash join: ~~scan(RSC')~~ + writeHashed(RSC') + scan(R) + writeHashed(R) + RSC' + R -> produces RSCR

~~5~~ + 5 + 1 + 1 + 5 + 1 = 13

SUM = 98 + 13 = 111 pages (assuming pipelining, you do not have to read the left input from disk)

Άσκηση: Compute the cost for all possible join algo's

- R: 10000 tuples, 10 tuples per page
- S: 2000 tuples, 10 tuples per page
- 52 buffers
- Εκτελώ το R $\bowtie_{A=B}$ S
- Εκτιμήστε το μέγεθος του αποτελέσματος και τον απαιτούμενο χώρο στο δίσκο για να γραφτεί το αποτέλεσμα αυτό. Προσοχή: δεν σας δίνονται άλλες πληροφορίες (π.χ., εξωτερικά κλειδιά) και άρα πρέπει να σκεφθείτε μια λογική εκτίμηση για τη χειρότερη περίπτωση ☺

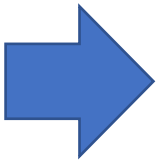
- Αιτιολογήστε το ακριβές, ελάχιστο και μέγιστο μέγεθος του αποτελέσματος της ερώτησης αν

- το B είναι πρωτεύον κλειδί και το A ξένο κλειδί στο B,
- το B είναι πρωτεύον κλειδί
- τίποτε από τα προηγούμενα δεν ισχύει

Foreign keys & join result size

R
1
1
2
1
2
3
3
3
4
1

Key-FK



S
1
2
3
4
5

Foreach r in R, **exactly one** match

R
1
17
25
18
2
3
33
33
44
1

Key @S

S
1
2
3
4
5

Foreach r in R, **at most one** match

R
10
17
25
18
2
3
33
33
44
19

S
1
2
3
4
5

Compute the cost for all possible join algo's

- R:10000 tuples, 10 tuples per page => 1000 pages
- S: 2000 tuples, 10 tuples per page => 200 pages
- S.*b:* primary key για το S.
- 52 buffers
- Υπολογίστε το κόστος του R $\rhd\!\lhd_{A=B}$ S αν ο αλγόριθμος είναι:
  - Simple NL
  - Block-based NL
  - SMJ
  - HJ
- Τι θα γινόταν αν είχα 10 buffers αντί για 52?
- Τι θα γινόταν αν είχα 250 buffers αντί για 52?
- Πιο το ελάχιστο κόστος και με πόσους buffers επιτυγχάνεται?
- Πόσες εγγραφές θα παραχθούν στο αποτέλεσμα κατά μέγιστο? Πόσες σελίδες χρειάζονται για να το αποθηκεύσουν?

Formulae

Algo	Cost formula
Simple NL *which one? Pick:*	M + M*N N + N*M
Block NL *Similarly…*	$M + \left\lceil \frac{M}{B} \right\rceil * N$
Sort-Merge Join	3*(M+N), if B > max($\sqrt{M}, \sqrt{N}$)
	2M(1+$\lceil \log_{B-1}[M/B] \rceil$) + 2N(1+$\lceil \log_{B-1}[N/B] \rceil$) + M+N, otherwise
Hash join w/o overflows	3*(M+N)