Query Processing and Optimization
Quiz Solution

Prof. Dr. Volker Markl

Alexander Alexandrov, Stephan Ewen,
Kostas Tzoumas, Fabian Hueske, Max Heimel

Fachgebiet Datenbanksysteme und Informationsmanagement
Technische Universität Berlin

http://www.dima.tu-berlin.de/

Questions 5.1 – 5.5

- Given are three relations R(<u>r1</u>, r2, r3), S(<u>r1</u>, <u>t1</u>), T(<u>t1</u>, t2, t3), where S.r1 is a foreign key on R.r1 and S.t1 a FK on T.t1.
- Following statistics are available:
  |R| = 1000
  |R.r2| = 50, min(R.r2) = 0, max(R.r2) = 50000
  |R.r3| = 500, min(R.r3) = 101, max(R.r3) = 600
  |S| = 20000
  |S.r1| = 1000
  |S.t1| = 200
  |T| = 200
  |T.t2| = 20, min(T.t2) = 0, max(T.t2) = 42
  |T.t3| = 8, min(T.t3) = 1, max(T.t3) = 256

- Given the RA expression E:

$$\pi_{r2,r3,t2}\left(\left(\left(\sigma_{(r2=75)\vee(r3>300\ \wedge r3\leqslant 400)} R\right) \bowtie_{r1} S\right) \bowtie_{t1} (\sigma_{t3=0}\, T)\right)$$

compute the cardinality estimates of the intermediate results for the following execution tree of E, as described in the lecture.

Assume independence and uniform distribution.

Questions 5.1 – 5.5

|IR1| = 216:

Selectivity (r2 = 75): S_1 = **(|R|/|R.r2|)/|R|** = (1 / 50) = 0.02

Selectivity (r3 > 300 AND r3 <= 400): S_2 = **(400-300)/(max(R.r3)-min(R.r3)+1)** = 100/500 = 0.2

Disjunctive predicate's selectivity S_3 = **1 - (1 - S_1)*(1 - S_2)** = 1 - (1 - 0.02)*(1 - 0.2) = 0.216

Cardinality: (|R| * S_3) = 1000 * 0.216 = 216

|IR2| = 4320:

S.r1 is FK of R.r1

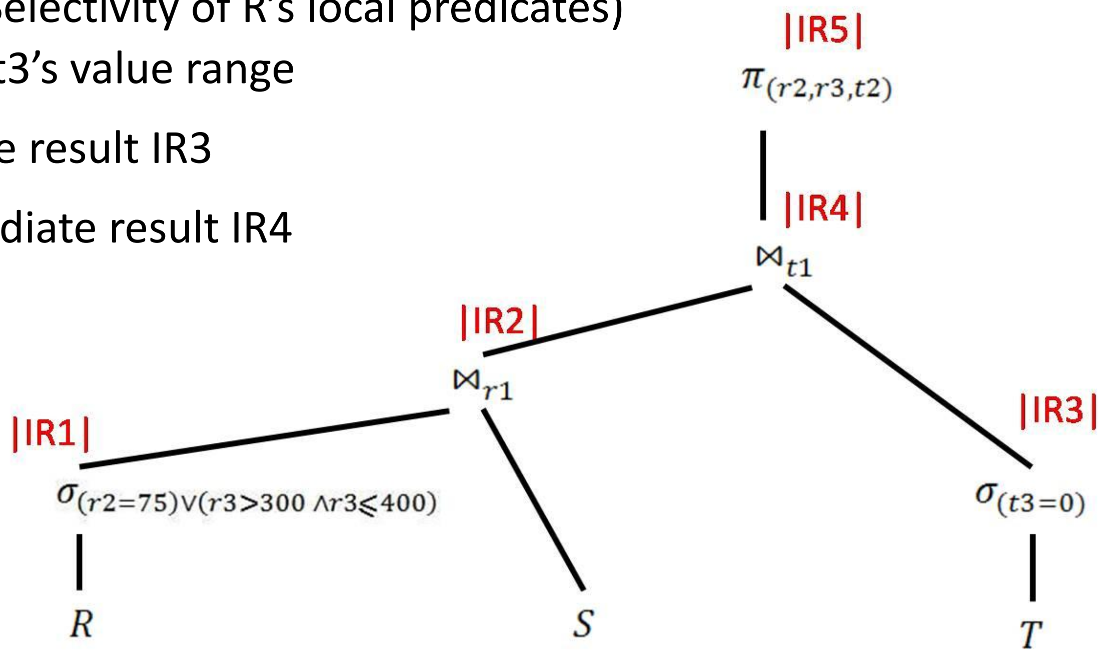
Assume uniform distribution of S.r1 and independence of R's attributes.

4320 = 20000 * 0.216 (Cardinality S * Selectivity of R's local predicates)

|IR3| = 0, predicate value (0) is out of t3's value range

|IR4| = 0, join with empty intermediate result IR3

|IR5| = 0, projection of empty intermediate result IR4

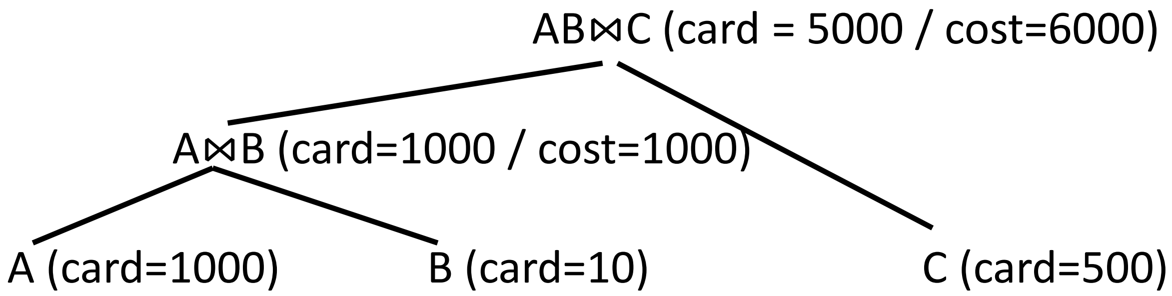


Questions 5.6 – 5.9

- Given are:
  - four relations R, S, T, U
    with cardinalities |R| = 20, |S| = 100000, |T| = 1000, |U| = 500
  - Local predicates on R (selectivity = 0.5) and U (selectivity = 0.2)
  - The following join-selectivities (based on size of the cross product of the inputs)

2-Table Joins	Selectivity	3-Table Joins	Selectivity	4-Table Joins	Selectivity
R ⋈ S	0,05	RS ⋈ T	0,001	RST ⋈ U	0,002
R ⋈ T	1,0	RS ⋈ U	1,0	RSU ⋈ T	0,000002
R ⋈ U	1,0	RT ⋈ S	0,00005	RTU ⋈ S	0,00005
S ⋈ T	0,001	RT ⋈ U	0,002	STU ⋈ R	0,05
S ⋈ U	1,0	RU ⋈ S	0,05	RS ⋈ TU	0,001
T ⋈ U	0,002	RU ⋈ T	0,002	RT ⋈ SU	0,0000001
		ST ⋈ R	0,05	RU ⋈ ST	0,0001
		ST ⋈ U	0,002		
		SU ⋈ R	0,05		
		SU ⋈ T	0,000002		
		TU ⋈ R	1,0		
		TU ⋈ S	0,001		

- Compute the costs for join order using a Dynamic Programming approach
- Costs are modeled as follows:
  - Table access is free (cost = 0)
  - The cost of a join is the sum of the resulting cardinalities of all previous joins (including the cardinality of the own result)
  - Example:



Questions 5.12 – 5.15

- Sheet and paper solution in two steps
  - Step 1 (Preprocessing): Compute IR cardinalities
    - Removes duplicate computations in the next step
    - Independent on the join order, e.g.
      $|(R \bowtie S) \bowtie T| = |R \bowtie (S \bowtie T)| = |(R \bowtie T) \bowtie S|$
    - IR cardinality decomposes into base relation cardinality and local and join predicate selectivity factors

  - Step 2 (Optimization): Salinger style enumeration of plans
    - Bottom-up enumeration

      for *k* in *[1 to N]* tables:

      generate all plans of *k* tables
    - Dynamic Programming: save and reuse results up to *k* for *k+1* tables
    - Pruning: keep only the best plan joining the same subset of k tables

Questions 5.12 – 5.15 (IR Cardinalities)

2-Table Joins	Cardinality	3-Table Joins	Cardinality	4-Table Joins	Cardinality
R ⋈ S	50000	R ⋈ S ⋈ T	50000	R ⋈ S ⋈ T ⋈ U	10000
R ⋈ T	10000	R ⋈ S ⋈ U	5000000		
R ⋈ U	1000	R ⋈ T ⋈ U	2000		
S ⋈ T	100000	S ⋈ T ⋈ U	20000		
S ⋈ U	10000000				
T ⋈ U	200				

Questions 5.12 – 5.15 (Step 2)

2-Tables	Card.	Cost	3-Tables	Card.	Cost	4-Tables	Card.	Cost
R $\bowtie$ S	$5*10^4$	$5*10^4$	~~RS $\bowtie$ T~~	~~$5*10^4$~~	~~$1*10^5$~~	~~RST $\bowtie$ U~~	~~$1*10^4$~~	~~$7*10^4$~~
R $\bowtie$ T	$1*10^4$	$1*10^4$	~~RS $\bowtie$ U~~	~~$5*10^6$~~	~~$505*10^4$~~	~~RSU $\bowtie$ T~~	~~$1*10^4$~~	~~$5011*10^3$~~
R $\bowtie$ U	$1*10^3$	$1*10^3$	RT $\bowtie$ S	$5*10^4$	$6*10^4$	RTU $\bowtie$ S	$1*10^4$	$122*10^2$
S $\bowtie$ T	$1*10^5$	$1*10^5$	~~RT $\bowtie$ U~~	~~$2*10^3$~~	~~$12*10^3$~~	~~STU $\bowtie$ R~~	~~$1*10^4$~~	~~$302*10^2$~~
S $\bowtie$ U	$1*10^7$	$1*10^7$	RU $\bowtie$ S	$5*10^6$	$5001*10^3$	~~RS $\bowtie$ TU~~	~~$1*10^4$~~	~~$602*10^2$~~
T $\bowtie$ U	$2*10^2$	$2*10^2$	~~RU $\bowtie$ T~~	~~$2*10^3$~~	~~$3*10^3$~~	~~RT $\bowtie$ SU~~	~~$1*10^4$~~	~~$1002*10^4$~~
			~~ST $\bowtie$ R~~	~~$5*10^4$~~	~~$15*10^4$~~	~~RU $\bowtie$ ST~~	~~$1*10^4$~~	~~$111*10^3$~~
			~~ST $\bowtie$ U~~	~~$2*10^4$~~	~~$12*10^4$~~			
			~~SU $\bowtie$ R~~	~~$5*10^6$~~	~~$15*10^6$~~			
			~~SU $\bowtie$ T~~	~~$2*10^4$~~	~~$102*10^4$~~			
			TU $\bowtie$ R	$2*10^3$	$22*10^2$			
			TU $\bowtie$ S	$2*10^4$	$202*10^2$			

Questions 5.6 – 5.9

- Q 5.6 Best left-deep plan:
  $(((T \bowtie U) \bowtie R) \bowtie S)$ -> 200 + 2000 + 10000 = 12200
- Q 5.7 Worst left-deep plan (pruned in the table on the prev. slide):
  $(((S \bowtie U) \bowtie R) \bowtie T)$ -> 10000000 + 5000000 + 10000
  = 15010000
- Q 5.8 Best bushy plan:
  $(T \bowtie U) \bowtie (R \bowtie S)$ -> 200 + 50000 + 10000 = 60200
- Q 5.9 Worst bushy plan:
  $(S \bowtie U) \bowtie (T \bowtie R)$ -> 10000000 + 10000 + 10000 = 10020000

Question 5.10

- Given is a relation R with tuple size of 256 Byte.
  Block size of the disc is 4KB and 128MB of memory are available.
  Assume for simplicity that blocks have no header, 1MB=1024KB, 1KB=1024Byte

- What is the max cardinality of R to sort it with a single pass method?

Answer: *524288*

Memory size / Tuple size:
128 MB / 256 B = 524288 = 2^19

- Given is a relation R with tuple size of 256 Byte.
  Block size of the disc is 4KB and 128MB of memory are available.
  Assume for simplicity that blocks have no header, 1MB=1024KB, 1KB=1024Byte
  (same as in 5.1)

- What is the max. cardinality of R to sort it with a TPMMS?

Answer: 17179344896

1. Data is sorted in memory (see 5.1) and spilled to disk
2. Sorted chunks are read from disk (one block per chunk) and merged

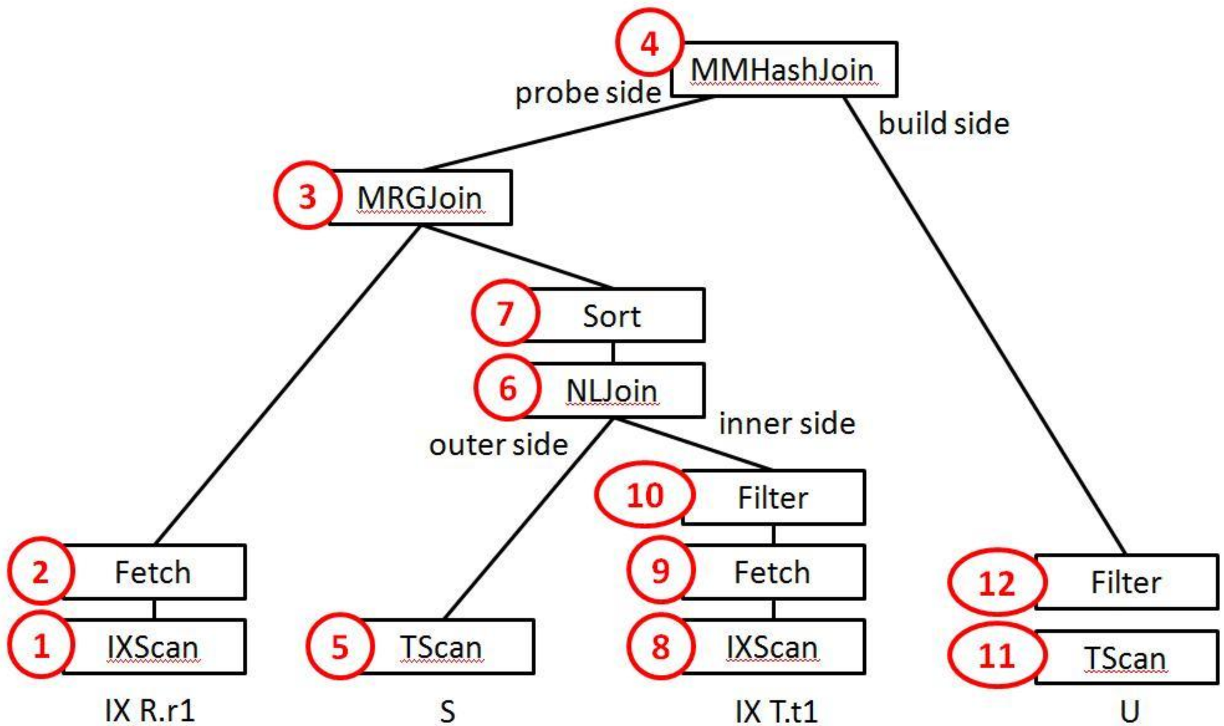
#Tuples sortable in memory (see 5.1) * Number of blocks in memory

*2^19 * ((128 MB / 4 KB) – 1) = 17179344896 ≈ 2^39*

1 block reserved for the output

Question 5.12

- Given the following query execution plan:

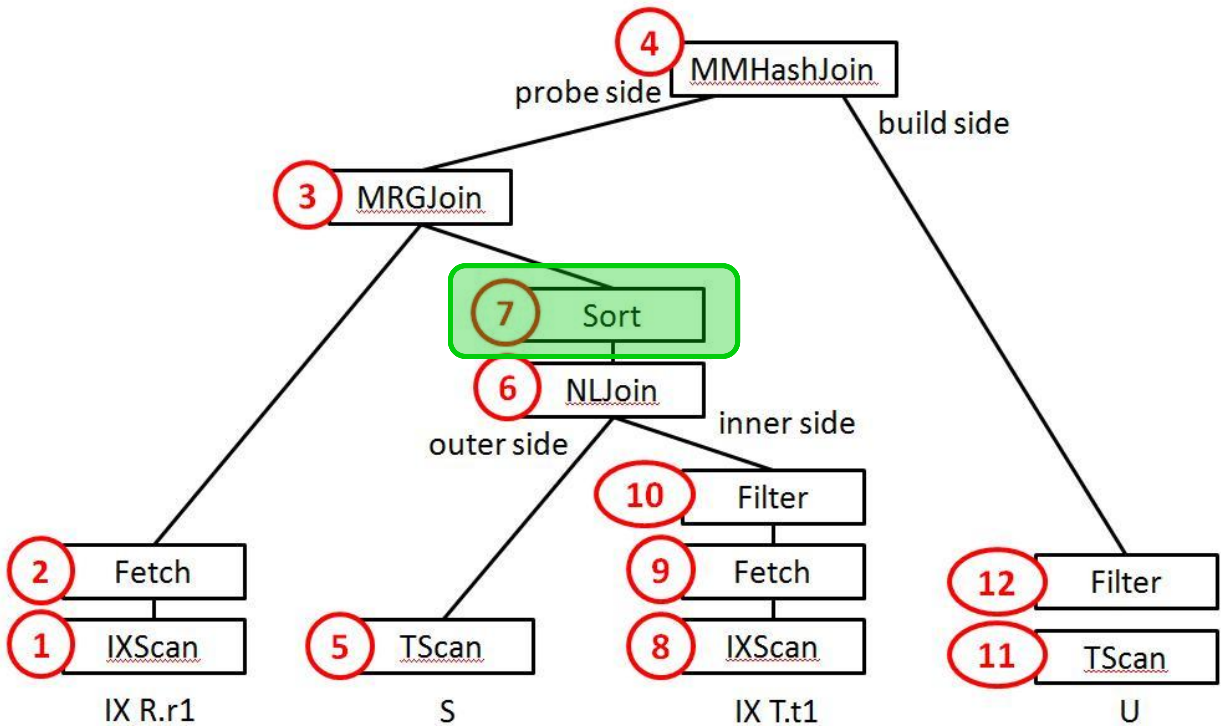


- Decide which operator acts as the first dam for the tuples of R. Operator IDs are given in red, if none operator is daming answer 0.

Answer: 0, R is completely pipelined

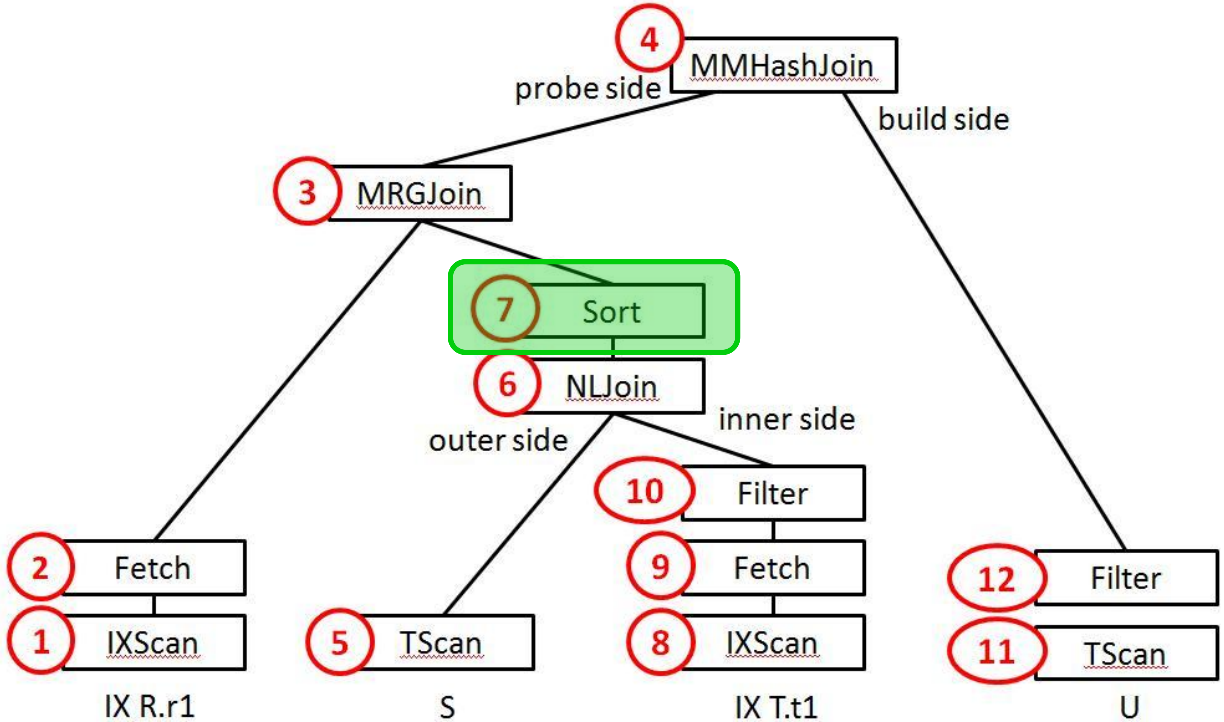
Question 5.13

- Given the following query execution plan:



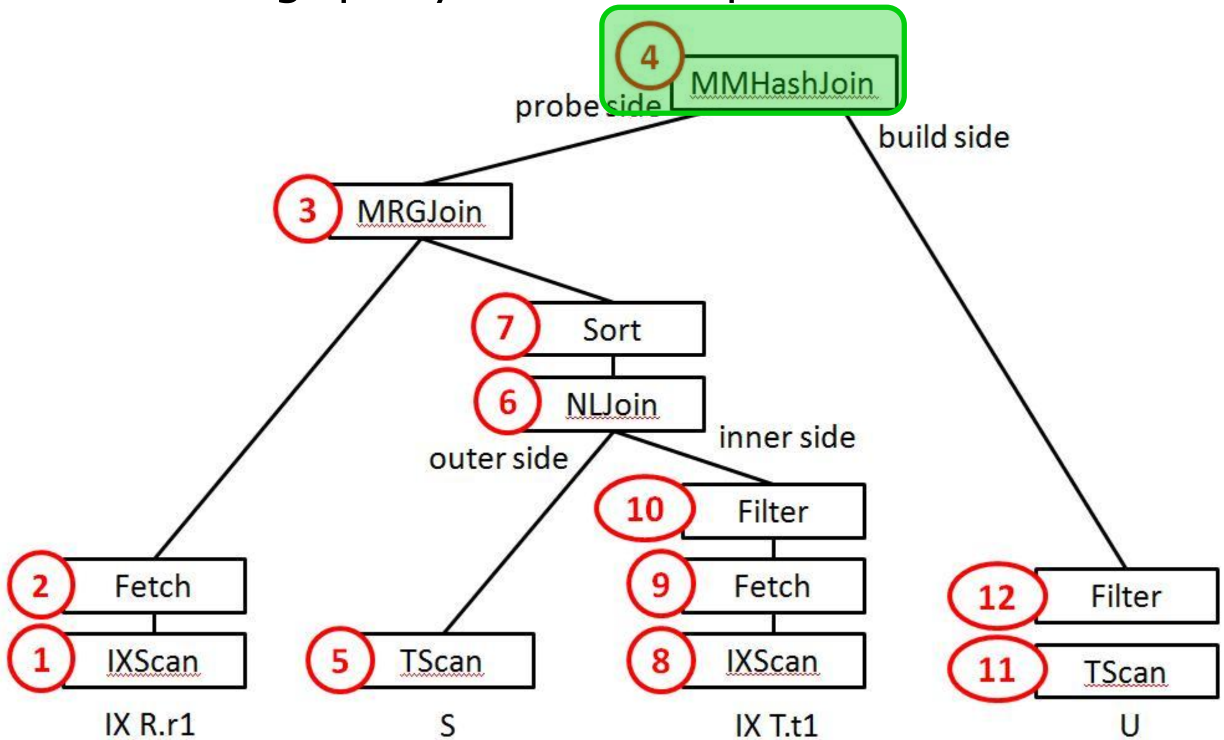
- Decide which operator acts as the first dam for the tuples of S. Operator IDs are given in red, if none operator is daming answer 0.

- Given the following query execution plan:



- Decide which operator acts as the first dam for the tuples of T. Operator IDs are given in red, if none operator is daming answer 0.

- Given the following query execution plan:



- Decide which operator acts as the first dam for the tuples of U. Operator IDs are given in red, if none operator is damming answer 0.