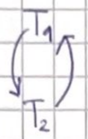
② a)

T_1	X(A) R(A) W(A)		X(C) R(C)					! X(B)		
T_2		X(B) R(B)			W(B)		!X(A)			! X(A)
T_3				!X(C)		!X(C)			! X(B)	

b)



c)

T_1	TS(T_1) X(A) R(A) W(A)		X(C) R(C)		
T_2		TS(T_2) X(B) R(B)			W(B)
T_3				"IF" ! X(C) ↦ TS(T_1) < TS(T_3) ... ABORT	

↪ T_3 | "IF" ! X(C) ↦ TS(T_1) < TS(T3) ... ABORT

d)

T_1	R(A) W(A) R(C)		R(B) W(B)			C
T_2				R(B) W(B) R(A) W(A)		C
T_3		R(C)			R(B) W(B)	C

① Part: Evaluation & Optimisation

$\pi_\{$Skiis.model, Skiis.lenght $(\delta_($Skiis.brand = 'Elan' AND Purchase.date > '31.12.2023'$)$
$($Skiis $\bowtie_{}$Skiis.sid = Purchase.sid Purchase$))\}$

* Buffer can store 2500 pages, 1 page on the disk is 8K

|Data Entry| = |index + pointer| = 8B + 8B = 16B

# data entries per page = 8000B/16 = 500

size of level of the leaves = 200000/500 = 400

size of B+ tree = 1,5 · (400+1) = 601,5

I will assume that 50% of purchases were made after '31.12.2023'

0,5 · 601,5 = 300,75 pages will be stored in T_1

Next assume 50% of records corresponding to the Skiis brand

50% · 250 · 30 = 3750 pages stored into T_2

Sort - Merge - Join

$2|T_1|(1+\log(B-1)(|T_1|/B)) + 2|T_2|(1+\log(B-1)(|T_2|/B)) + T_1 + T_2 =$

$2 \cdot 301 \cdot \left(1+ \underset{0}{\cancel{\frac{\log_{10}\frac{301}{2499}}{\log_{10}2499}}}\right) + 2 \cdot 3750 \cdot \left(1+ \underset{\overset{2}{0}}{\cancel{\frac{\log_{10}\frac{3750}{2499}}{\log_{10}2499}}}\right) + 301 + 3750 =$

602 + 7500 + 301 + 3750 = 12 153

Total: 301 + 7500 + 3750 + 12153 = 23 704

Index - Nested - Loops - Join

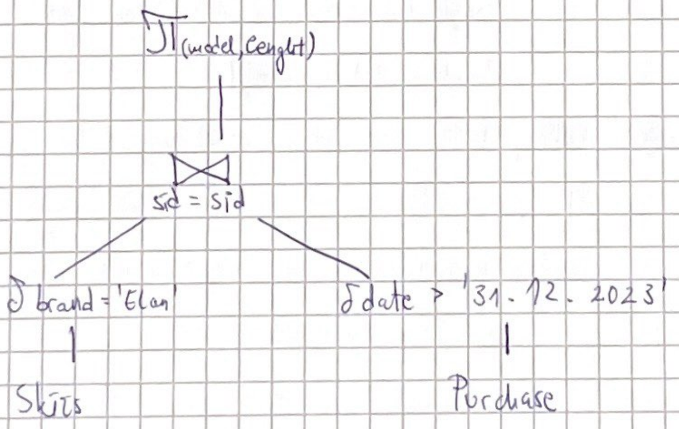
$N + M \cdot p_{rn} \cdot 1{,}2 = |T_2| + |T_1| \cdot$ rec. per pages $T_2 \cdot 1{,}2 = 3750 + 301 \cdot 30 \cdot 1{,}2 = 14586$

Total = 301 + 3750 + 250 + 14586 = 18 887 pages

Hash - Join cannot be done

So Index - Nested - Loops - Join is the most optimal solution out of all of them.

* I forgot to draw this on the previous page:



89224055
Anastasija
Tomova

③ Part: Normalization

a) The R schema is in 1NF (First Normal Form) because every attribute that is in the schema is atomic.

b) To normalize the schema, I will devide it into two parts.

From 1NF to 2NF: (removing transitive dependences)

$R_1 = ABD$, $F_{R_1} = \{B \to D\}$

$R_2 = AC$, $F_{R_2} = \{A \to C, C \to A\}$

From 2NF to 3NF:

Because R_1 and R_2 are already in 3NF no decomposition is needed so

$R_1 = ABD$, Keys $= \{A, B\}$, $F_{R_1} = \{B \to D\}$

$R_2 = AC$, Keys $= \{A, C\}$, $F_{R_2} = \{A \to C, C \to A\}$