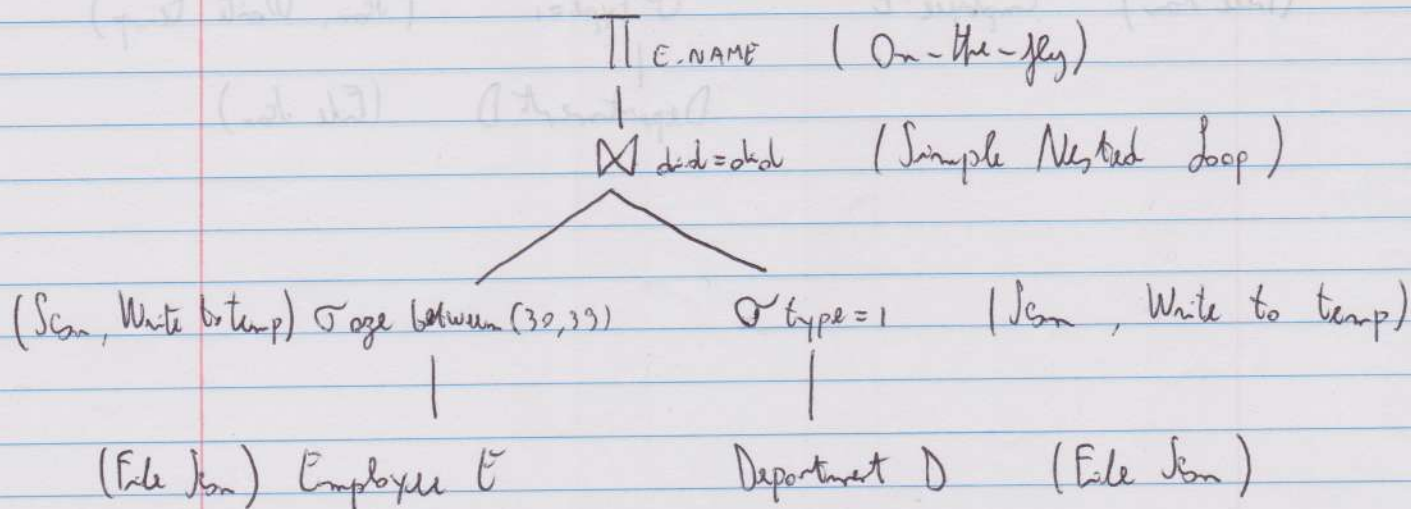
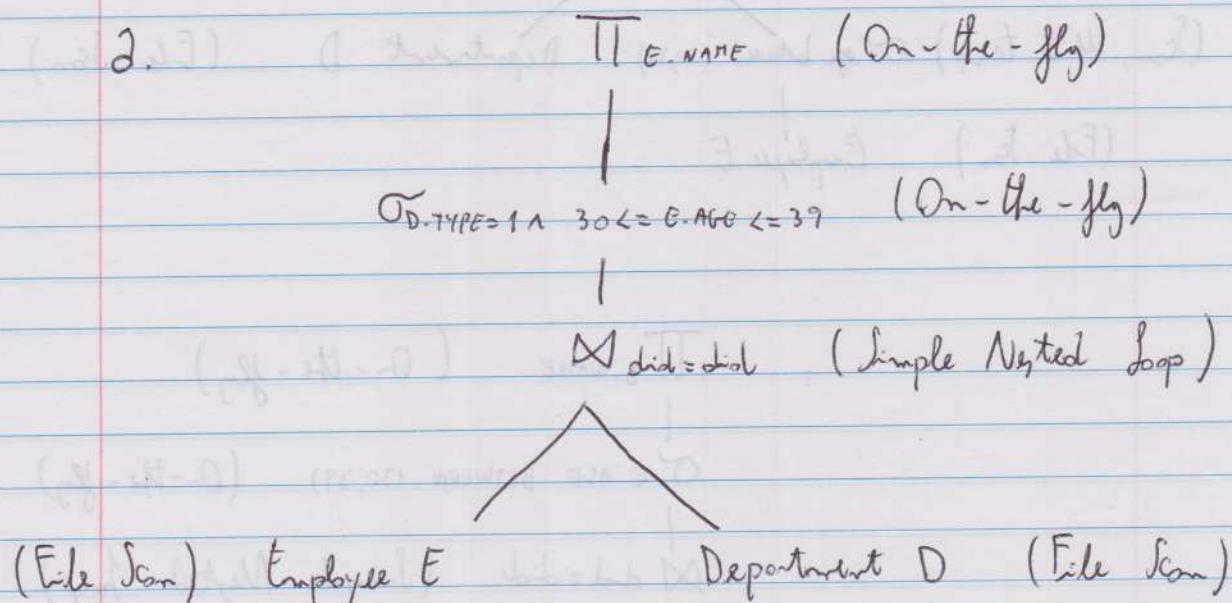


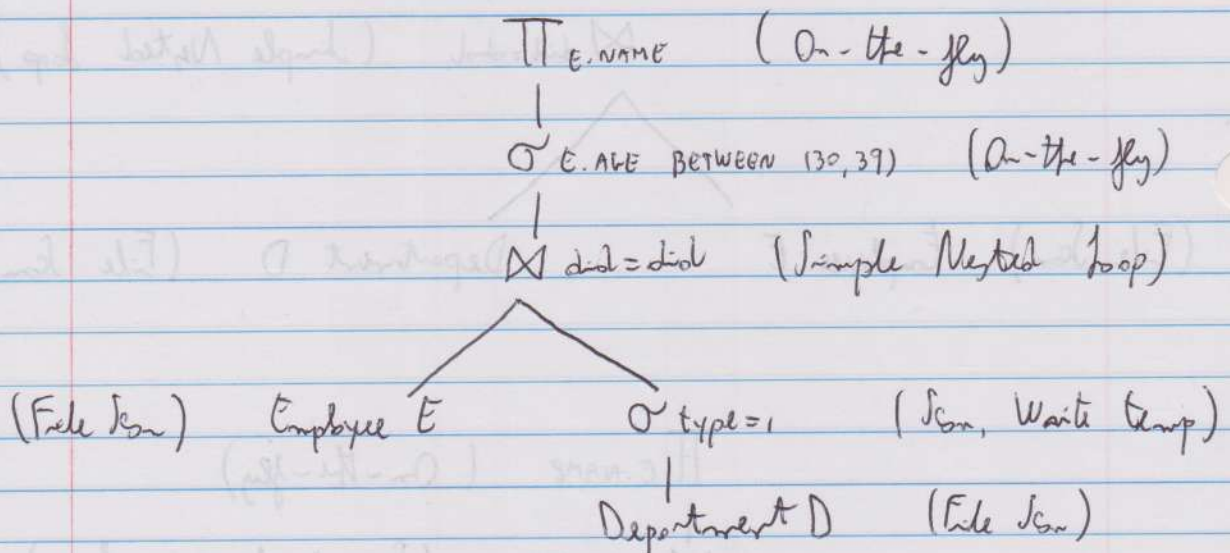
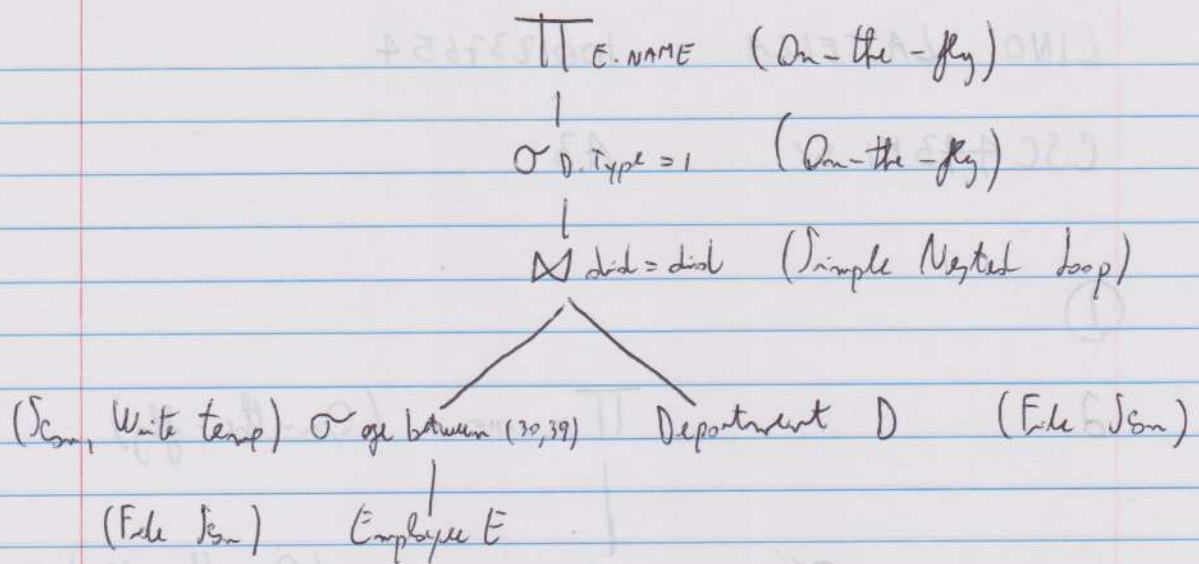
LINO LASTELLA 1001237654

CSC 443 H1 W A3

①

2.





b.

Cost of first plan:

- E outer, D inner  $\rightarrow 100 + 100 \cdot 4 = 500$

- D outer, E inner  $\rightarrow 4 + 4 \cdot 100 = 404$

Cost of second plan:

Let  $T_1$  be the temporary file for E, and let  $T_2$  be the temporary file for D. Then,

- Outer  $T_1$ , inner  $T_2 \rightarrow$  Scan E (100) +  
Write  $T_1$  (IF 1250 records, uniform, 25 pages) +

Scan D (4) +  
Write  $T_2$  (IF 60 records, uniform, 1 page) + SNL

$= 130 + 25 + 25 \cdot 1 = 180$

- Outer  $T_2$ , inner  $T_1 \rightarrow 130 + 1 + 1 \cdot 25 = 156$

Cost of third plan:

Scan E (100) + WRITE T (IF 1250 records, uniform, 25 pages)

- outer  $T_1$ , inner D  $\rightarrow 125 + 25 + 25 \cdot 4 = 250$

- outer D, inner T  $\rightarrow 125 + 4 + 4 \cdot 25 = 229$



Cost of best plan:

Join D (4) + Write T (IF 60 results, uniform, 1 page)

- outer T, inner E  $\rightarrow 5 + 1 + 1 \cdot 100 = 106$
- outer E, inner T  $\rightarrow 5 + 100 + 100 \cdot 1 = 205$

The most optimised plan is plan #4 and having the temporary file as outer loop when performing the join.

②

d.

LSN	prevLSN	TID	type	PID	old	new	undo LSN	next undo LSN
1	—	T1	U	P1	A	B	—	—
2	—	T2	U	P2	C	D	—	—
3	2	T2	U	P1	B	E	—	—
4	1	T1	U	P2	D	F	—	—
5	4	T1	COM	—	—	—	—	—
6	3	T2	U	P1	E	G	—	—
7	5	T1	END	—	—	—	—	—
8	6	T2	U	P2	F	H	—	—

### TRANSACTION TABLE

TID	STATUS	LAST LSN
<del>T1</del>	<del>COMMIT</del>	<del>5</del>
T2	ACTIVE	8

### DIRTY PAGE TABLE

PID	rec LSN
P1	1
P2	2

### MEMORY :

P1 CONTAINS VALUE G, LSN IS 6  
P2 CONTAINS VALUE H, LSN IS 8

b.

### ANALYSIS

AFTER THE CRASH, EVERYTHING THAT WAS NOT FLUSHED TO DISK IS LOST.

IN OUR CASE, ONCE THE SYSTEM RESTARTS, IT FINDS ONLY LOG ENTRIES UP TO LSN 5.

THE ANALYSIS PHASE STARTS READING FROM THE BEGINNING OF THE LOG AND TRIES TO REBUILD THE TRANSACTION TABLE AND THE DIRTY PAGE TABLE



THE RESULT IN OUR CASE WOULD BE AS FOLLOWS :

### TRANSACTION TABLE

TID	STATUS	LAST LSN
T1	COMMIT	5
T2	UNKNOWN	3

### DIRTY PAGE TABLE

PID	rec LSN
P1	1
P2	2

### REDO

REDO STARTS AT FIRST LSN WHICH IS THE SMALLEST LSN IN THE DIRTY PAGE TABLE (IN OUR CASE LSN 1).

FOR EVERY UPDATE OR CLR THE REDO PHASE REDOES THE CHANGE WHEN NECESSARY.

THE PROCEDURE IS AS FOLLOWS :

TO CHECK IF THE SYSTEM NEEDS TO MAKE CHANGES TO THE PAGE IT FIRST CHECKS IF THE PAGE IS IN THE DIRTY PAGE TABLE. IF IT IS, IT CHECKS THAT recLSN FOR THAT PAGE IS  $\leq$  THE LSN OF THE CHANGE UNDER CONSIDERATION.

IF IT IS, THE SYSTEM READS THE PAGE FROM DISK AND CHECKS IF THE PAGE LSN IS  $<$  CURRENT LSN.

IF IT IS, THEN THE SYSTEM REDOES THE CHANGE, ELSE SKIP.

IN OUR CASE :

LSN	ACTION
1	REDONE
2	REDONE
3	NO REDO SINCE PAGE LSN IS ALREADY 3
4	REDONE
5	SHIPPED

UNDO

THE SYSTEM MUST UNDO T2 SINCE IT'S THE ONLY TRANSACTION IN THE TRANSACTION TABLE. (T1 IS REMOVE RIGHT AT THE END OF THE REDO PHASE) SO START FROM LSN 3

LSN	prev LSN	TID	type	PID	old	New	undo LSN	next undo LSN
1	-	T1	U	P1	A	B	-	-
2	-	T2	U	P2	C	D	-	-
3	2	T2	U	P1	B	E	-	-
4	1	T1	U	P2	D	F	-	-
5	4	T1	COM	-	-	-	-	-
6	5	T1	END	-	-	-	-	-
7	-	T2	CLR	UNDO T2 LSN 3	E	B	3	2
8	-	T2	CLR	UNDO T2 LSN 2	D	C	2	-
9	8	T2	END	-	-	-	-	-



c.

THERE ARE MANY POSSIBLE SCENARIOS IF THE SYSTEM CRASHES ONCE AGAIN DURING RECOVERY PHASE

- SYSTEM FLUSHED ENTIRE NEW LOG TO DISK :

THEN TRANSACTION TABLE AT THE END OF ANALYSIS PHASE BECOMES EMPTY AND THERE WOULD NOT BE ANY UNDO.

- SYSTEM CRASHES AFTER CLR WITH LUN 7:

THEN THE UNDO PHASE STARTS FROM THAT CLR WHEN UNDOING T2. IT WOULD FOLLOW THE next UNDO LUN FIELD, ADD ONE MORE CLR ENTRY INTO THE LOG AND THEN END.

- SYSTEM CRASHES BEFORE ANYTHING GETS FLUSHED TO DISK:

THEN ALL THE STEPS (ANALYSIS, UNDO, REDO) WILL BE PERFORMED AGAIN JUST AS DESCRIBED IN SECTION b.

- SYSTEM CRASHES BUT NOT END ENTRY WAS NOT YET FLUSHED :

THEN AFTER ANALYSIS PHASE, ONLY T2 IS IN THE TRANSACTION TABLE, AND THE DIRTY PAGE TABLE IS UNCHANGED.

REDO IS THE SAME UP TO LUN 5. FOR LUN 6 NOTHING HAPPENS. FOR LUN 7, IF PAGE WAS FLUSHED THEN IT'S SKIPPED, OTHERWISE REDONE. LIKEWISE FOR LUN 8.

THEN UNDO PHASE WOULD UNDO T2, BUT SINCE THERE IS A CLR ENTRY WITH next UNDO LUN, IT JUST ADDS AN END ENTRY.