Q1a) Page-oriented Nested Loops Join. Consider A as the outer relation. (1 mark)

Cost = NPages(A) + NTuples(A)*Npages(B)
= 10000/100 + (10000 * 200000/100)
= 100 + 20000000
= 20000100 I/O

Q1b) Block-oriented Nested Loops Join. Consider A as the outer relation. (1 mark)

NBlocks(A) = NPages(A) / (Blocksize – 2)
= (10000/100) / (52 – 2)
= 2 I/O
Cost = NPages(A) + NBlocks(A)*NPages(B)
= 10000/100 + 2*(200000/100)
= 100 + 4000
= 4100 I/O

Q1c) Sort-Merge Join. (1 mark)

Sort(A) = 2*NumPasses*NPages(A)
= 2*2*(10000/100)
= 400 I/O
Sort(B) = 2*NumPasses*NPages(B)
= 2*2*(200000/100)
= 8000 I/O
Cost = Sort(A) + Sort(B) + NPages(A) + NPages(B)
= 400 + 8000 + (10000/100) + (200000/100)
= 10500 I/O

Q1d) Hash-Join. (1 mark)

Cost = 2*NPages(A) + 2*NPages(B) + NPages(A) + NPages(B)
= 2*(10000/100) + 2*(200000/100) + (10000/100) + (200000/100)
= 6300 I/O

Q1e) What would be the lowest possible I/O cost for joining A and B using any join algorithm and how much buffer space would be needed to achieve this cost? Explain briefly. (1 mark)

The lowest possible I/O cost for joining A and B using any algorithm will be 4100 I/O with Block-oriented NLJ. The buffer pool would need to be:

Npages(A) + Input + Output = 100 + 2 = 102 Buffer Pages

Q2a) Compute the estimated result size and the reduction factors (selectivity) of this query. (1 mark)

RF(title) = 1/5
RF(level) = 1/10
Estimated Result Size = NTuples(Managers)*∏RF
= 500000*(1/5)*(1/10)
= 10000 tuples

Q2b) Compute the estimated cost of the best access path assuming that a clustered B+ tree index on (title, level) is (the only index) available. Suppose there are 200 index pages. Discuss and calculate alternative access paths. (1 mark)

Cost = (NPages(I) + NPages(Managers))*∏RF
= (200 + 5000)*(1/50)
= 104 I/O

Alternatives Access Paths:

Heap Scan = 5000 I/O

Q2c) Compute the estimated cost of the best access path assuming that an unclustered B+ tree index on (level) is (the only index) available. Suppose there are 200 index pages. Discuss and calculate alternative access paths. (1 mark)

Cost = (NPages(I) + NTuples(Managers))*RF(level)
= (200 + 500000)*(1/10)
= 50020 I/O where title = 'Architect' is checked on the fly

Alternative Access Paths:

Heap Scan = 5000 I/O

Q2d) Compute the estimated cost of the best access path assuming that an unclustered Hash index on (title) is (the only index) available. Discuss and calculate alternative access paths. (1 mark)

Cost = NTuples(Managers)*RF(title)*2.2
= 500000*(1/5)*2.2
= 220000 I/O where level > 18 is checked on the fly

Q2e) Compute the estimated cost of the best access path assuming that an unclustered Hash index on (level) is (the only index) available. Discuss and calculate alternative access paths. (1 mark)

Hash index over a range (level > 18) is not possible.
Therefore, an alternative access path could be a Heap Scan at a cost of 5000 I/O

Q3a) Compute the estimated result size and the reduction factors (selectivity) of this query (2 marks)

RF(Salary) = (100000 – 90000)/(100000 – 50000)
= 1/5
RF(Hobby) = 1/200
RF(F.id = D.id) = RF(D.id = F.id) = 1/5000

Result Size(Query) = $\prod$NPages(Tables)*$\prod$RF
= 10 Tuples

Q3b) Compute the cost of the plans shown below. Assume that sorting of any relation (if required) can be done in 2 passes. NLJ is a Page-oriented Nested Loops Join. Assume that did is the candidate key, and that 100 tuples of a resulting join between Emp and Dept fit in a page. Similarly, 100 tuples of a resulting join between Finance and Dept fit in a page. (8 marks, 2 marks per plan)

Plan 1)

Cost(Dept⋈Finance) = NPages(Dept) + NPages(Dept)*NPages(Finance)
= (5000/100) + (5000/100)*(5000/100)
= 2550 I/O
Result Size(Dept⋈Finance) = 5000*5000*(1/5000) - PIPELINE
= 50 pages
Cost(⋈Emp) = 50*(10000/100)
= 5000 I/O
Total Cost = 7550 I/O

Plan 2)

Cost(Dept⋈Finance) = 3*NPages(Dept) + 3*NPages(Finance)
= 6*(5000/100)
= 300 I/O
Result Size(Dept⋈Finance) = 5000*5000*(1/5000)
= 50 pages
Cost(⋈Emp) = Sort(Dept⋈Finance) + NPages(Emp) – PIPELINE + READ
= (2*2*50) + 50 + 100 + 50
= 350 I/O
Total Cost = 650 I/O

Plan 3)

Cost(Emp⋈Dept) = Sort(Dept) + NPages(Emp) + NPages(Dept) + READ
= (2*2*50) + 100 + 50 + 50
= 400 I/O
Result Size(Emp⋈Dept) = 10000*5000*(1/5000)
= 100 pages
Cost(⋈Finance) = 2*100 + 3*50 – PIPELINE
= 350 I/O
Total Cost = 750 I/O

Plan 4)

Cost(Emp⋈Dept)	=	3*NPages(Emp) + 3*NPages(Dept)
	=	3*100 + 3*50
	=	450 I/O
Result Size(Emp⋈Dept)	=	10000*5000*(1/5000)
	=	100 pages
Cost(⋈Finance)	=	100*50 - PIPELINE
	=	5000 I/O
Total Cost	=	5450 I/O