Question 1

a) Page-oriented Nested Loops Join. Consider A as the outer relation.
A = 80000 tuples = 800 pages
B = 100000 tuples = 1000 pages
Cost (PNJL) = NPages(Outer) + NPages(Outer) * NPages(Inner)
= 800 + 800 * 1000
= 800800 I/O

b) Block-oriented Nested Loops Join. Consider A as the outer relation.
102 buffer pages available
NBlocks(Outer) = $NPages(Outer)/(B-2)$ = 800/(102-2) = 8 blocks
Cost (BNJL) = NPages(Outer) + NBlocks(Outer) * NPages(Inner)
= 800 + 8 * 1000
= 8800 I/O

c) Sort-Merge Join. Assume that Sort-Merge Join can be done in 2 passes.
NumPasses = 2
Sort(R) = External Sort Cost = 2*NumPasses*NPages(R)
Cost (SMJ) = Sort(Outer) + Sort(Inner) + NPages(Outer) + NPages(Inner)
= 5 * 800 + 5 * 1000
= 9000 I/O

d) Hash Join
Cost (HJ) = 2 * NPages(Outer) + 2* NPages(Inner) + NPages(Outer) + NPages(Inner)
= 3 * 800 + 3 * 1000
= 5400 I/O

e) What would be the lowest possible cost to perform this query, assuming that no indexes are built on any of the two relations, and assuming that sufficient buffer space is available? What would be the minimum buffer size required to achieve this cost? Explain briefly.
Out of the join strategies calculated, a Hash Join initially seems to have the lowest cost. However, given a sufficient number of buffer pages the Block-oriented Nested Loops Join becomes optimal by solving for the number of blocks required to match the Hash Join:
800 + 800/(x – 2) * 1000 = 5400 I/O
800/(x – 2) = 4.6
x = 800/4.6 + 2 ≈ 175.913 = 176 buffer pages (need whole number of pages)

Question 2

a) Compute the reduction factors and the estimated result size in number of tuples.

8 cities

Salary ranges from 60000 to 100000

NTuples (JobSeekers) = 10000 pages * 100 tuples/page = 1000000 tuples

Col = value: RF = 1/NKeys(Col)

RF (city) = 1 / 8

= 0.125

Col > value RF = (High(Col) – value) / (High(Col) – Low(Col))

RF (soughtsalary) = (160000 – 80000) / (160000 – 60000)

= 80000 / 100000

= 0.8

ResultSize = $\boldsymbol{NTuples(R)}$ $\Pi_{i=1..n} RF_i$

= 1000000 * 0.125 * 0.8

= 100,000 tuples

b) Compute the estimated cost in number of disk I/O's of the best plan if a clustered B+ tree index on (city, soughtsalary) is the only index available. Suppose there are 2,000 index pages. Discuss and calculate alternative plans.

Cost (B+Tree) = (NPages(I) + NPages(R)) * $\Pi_{i=1..n} RF_i$

= (2000 + 10000) * 0.125 * 0.8 = 1200 I/O

Cost (Heap) = NPages(R)

= 10000 I/O

The cost of using the clustered B+ tree is far less than the heap scan, so using the index is the optimal plan in this case.

c) Compute the estimated cost in number of disk I/O's of the best plan if an unclustered B+ tree index on (soughtsalary) is the only index available. Suppose there are 2,000 index pages. Discuss and calculate alternative plans.

Cost (B+Tree) = (NPages(I) + NTuples(R)) * $\Pi_{i=1..n} RF_i$

= (2000 + 1000000) * 0.8

= 801600 I/O

Cost (Heap) = NPages(R)

= 10000 I/O

With the provided unclustered index, the resulting cost is far greater than if a simple Heap scan was used, making a Heap scan the optimal plan in this case.

d) Compute the estimated cost in number of disk I/O's of the best plan if an unclustered Hash index on (city) is the only index available. Discuss and calculate alternative plans.

Cost (HashIndex) = NTuples(R) * $\Pi_{i=1..n}$ RF_i * 2.2
= 1000000 * 0.125 * 2.2
= 275000 I/O

Cost (Heap) = NPages(R)
= 10000 I/O

The cost of a Heap scan is again much less than the provided unclustered Hash index, making a Heap scan the optimal plan in this case.

e) Compute the estimated cost in number of disk I/O's of the best plan if an unclustered Hash index on (soughtsalary) is the only index available. Discuss and calculate alternative plans.

Cost (HashIndex) = NTuples(R) * $\Pi_{i=1..n}$ RF_i * 2.2
= 1000000 * 0.8 * 2.2
= 1760000 I/O

Cost (Heap) = NPages(R)
= 10000 I/O

The cost of a Heap scan is again far less than the provided unclustered Hash index, making a Heap scan the optimal plan in this case.

Question 3

a) Compute the reduction factors and the estimated result size in number of tuples.

Sal ranges from 50000 to 150000

50 different hobbies

Col > value: RF = (High(Col) – value) / (High(Col) – Low(Col))

RF (sal) = (150000 – 100000) / (150000 – 50000)

= 50000 / 100000

= 0.5

Col = value: RF = 1/NKeys(Col)

RF (hobby) = 2 / 50 (‘diving’ and ‘soccer’)

= 0.04

Col_A = Col_B (for joins): RF = 1/ (Max (NKeys(Col_A), NKeys(Col_B)))

RF (did Emp & Dept) = 1 / (Max (25000, 1200)

= 1 / 25000

Col_A = Col_B (for joins): RF = 1/ (Max (NKeys(Col_A), NKeys(Col_B)))

RF (did Dept & Finance) = 1 / (Max (1200, 1200))

= 1/ 1200

ResultSize = $\Pi_{j=1..k}$ *NTuples*(R_j) $\Pi_{i=1..n}$ RF_i

= (25000 * 1200 * 1200) * 0.5 * 0.04 * (1 / 1200) * (1 / 25000)

= 24 tuples

b) Compute the cost in number of disk I/O’s 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 50 tuples of a resulting join between Emp and Dept fit in a page. Similarly, 50 tuples of a resulting join between Finance and Dept fit in a page. Any selections/projections not indicated on the plan are performed “on the fly” after all joins have been completed.

1)

Calculating Cost:

Dept x Finance

1200 Dept tuples / 100 tuples/page = 12 pages

1200 Finance tuples / 100 tuples/page = 12 pages

25000 Emp tuples / 100 tuples/page = 250 pages

Cost (PNJL) = NPages(Outer) + NPages(Outer) * NPages(Inner)

Cost (Dept x Finance) = 12 + 12 * 12

= 156

(Dept x Finance) x Emp

ResultSize (Dept x Finance) = $\Pi_{j=1..k}$ *NTuples*(R_j) $\Pi_{i=1..n}$ RF_i

= 1200 * 1200 * 1 / 1200 = 1200 tuples / 50 tuples/page = 24 pages

Cost (PNJL) = NPages(Outer) + NPages(Outer) * NPages(Inner)

Cost (x Emp) = 24 + 24 * 250

= 6000

Total Cost = 12 + 12 * 12 + 24 * 250

= 6156 I/O

2)
Calculating Cost:
Dept x Finance
1200 Dept tuples / 100 tuples/page = 12 pages
1200 Finance tuples / 100 tuples/page = 12 pages
25000 Emp tuples / 100 tuples/page = 250 pages
2 sorting passes
Cost (HJ) = 2 * NPages(Outer) + 2* NPages(Inner) + NPages(Outer) + NPages(Inner)
= 3 * 12 + 3 * 12
= 72
(Dept x Finance) x Emp
ResultSize (Dept x Finance) = $\Pi_{j=1..k}$ *NTuples*(R_j) $\Pi_{i=1..n}$ RF_i
= 1200 * 1200 * 1 / 1200 = 1200 tuples / 50 tuples/page = 24 pages
Cost (SMJ) = Sort(Outer) + Sort(Inner) + NPages(Outer) + NPages(Inner) Sort(R) = External
Sort Cost = 2*NumPasses*NPages(R)
Cost (x Emp) = 2 * 2 * 24 – 24 + 2 * 2 * 250 + 24 + 250 – 250
= 72 + 1000 + 24
= 1096
Total Cost = 3 * 12 + 3 * 12 + 2 * 2 * 24 – 24 + 2 * 2 * 250 + 24 + 250 – 250
= 1168 I/O

3)
Calculating Cost:
Emp x Dept
1200 Dept tuples / 100 tuples/page = 12 pages
1200 Finance tuples / 100 tuples/page = 12 pages
25000 Emp tuples / 100 tuples/page = 250 pages
2 sorting passes
Cost (SMJ) = Sort(Outer) + Sort(Inner) + NPages(Outer) + NPages(Inner) Sort(R) = External
Sort Cost = 2*NumPasses*NPages(R)
Cost (Emp x Dept) = 5 * 250 + 5 * 12
= 1310
(Emp x Dept) x Finance
ResultSize (Emp x Dept) = $\Pi_{j=1..k}$ *NTuples*(R_j) $\Pi_{i=1..n}$ RF_i
= 25000 * 1200 * 1 / 25000 = 1200 tuples / 50 tuples/page = 24 pages
Cost (HJ) = 2 * NPages(Outer) + 2* NPages(Inner) + NPages(Outer) + NPages(Inner)
Cost (x Finance) = 3 2 * 24 + 3 * 12
= 84
Total Cost = 5 * 250 + 5 * 12 + 2 * 24 + 3 * 12 = 84
= 1394 I/O

4)

Calculating Cost:

Emp

1200 Dept tuples / 100 tuples/page = 12 pages

1200 Finance tuples / 100 tuples/page = 12 pages

25000 Emp tuples / 100 tuples/page = 250 pages

Cost (B+Tree) = (NPages(I) + NPages(R))* $\Pi_{i=1..n}$ RF_i

Cost ($\boldsymbol{\sigma}_{Emp.sal > 100000}$) = (50 + 250) * 0.5

= 150

$\boldsymbol{\sigma}_{Emp.sal > 100000}$ x Dept

ResultSize ($\boldsymbol{\sigma}_{Emp.sal > 100000}$) = *NTuples*($R_j$) $\Pi_{i=1..n}$ RF_i

= 25000 * 0.5

= 12500 tuples / 100 tuples/page = 125 pages

Cost (HJ) = 2 * NPages(Outer) + 2* NPages(Inner) + NPages(Outer) + NPages(Inner)

Cost ($\boldsymbol{\sigma}_{Emp.sal > 100000}$ x Dept) = 3 2 * 125 + 3 * 12

= 286

($\boldsymbol{\sigma}_{Emp.sal > 100000}$ x Dept) x Finance

ResultSize ($\boldsymbol{\sigma}_{Emp.sal > 100000}$ x Dept) = $\Pi_{j=1..k}$ *NTuples*(R_j) $\Pi_{i=1..n}$ RF_i

= 12500 * 1200 * 1 / 25000

= 600 tuples / 50 tuples/page = 12 pages

Cost (PNJL) = NPages(Outer) + NPages(Outer) * NPages(Inner)

Cost (x Finance) = 12 + 12 * 12

= 144

Total Cost = (50 + 250) * 0.5 + 2 * 125 + 3 * 12 + 12 * 12

= 580 I/O