INFO20003 Database System Assignment 3

Juntong Angel Tan student ID: 987979

Question 1

1.(a) Page-oriented Nested Loops Join.

NPages(Parts) = NTuples(Parts) / NTuplesPerPage(Parts)
= 60,000 / 50 = 1200 pages

NPages(Supply) = NTuples(Supply) / NTuplesPerPage(Supply)
= 150,000 / 50 = 3000 pages

Cost = NPages(Parts) + NPages(Parts) x NPages(Supply)
= 1200 + 1200 x 3000 = **3601200** (I/O)

1.(b) Block-oriented Nested Loops Join.

NBlock(Parts) = NPages(Parts) / (B – 2)
= 1200 / (202 – 2) = 6

Cost = NPages(Parts) + NBlocks(Parts) x NPages(Supply)
= 1200 + 6 x 3000 = **19200** (I/0)

1.(c) Sort-Merge Join.

Cost = Sort(Parts) + Sort(Supply) + Merge
= 2 x NumPasses x NPages(Parts) + 2 x NumPasses x NPages(Supply)
+ NPages(Parts) + NPages(Supply)
= 2 x 2 x 1200 + 2 x 2 x 3000 + 1200 + 3000
= 5 x (1200 + 3000) = **21000** (I/O)

1.(d) Hash Join.

Cost = 3 x NPages(Parts) + 3 x NPages(Supply)
= 3 x 1200 + 3 x 3000 = **12600** (I/0)

1.(e)

Block Nested Loops Join with B chosen so that the smaller table fits into memory as a single block.
Due to NPages(Parts) / (B – 2) = 1 → B = 1202 pages
Cost = 1200 + 3000 = 4200 (I/O)
Therefore, the minimum buffer size is **1202** pages, the cost is **4200** (I/O).

Question 2

2.(a)

$RF_{Salary > 300,000}$ = [High(Salary) – Value] / [High(Salary) – Low(Salary)]
= (500,000 – 300,000) / (500,000 – 100,000) = **0.5**

$RF_{department = \text{"Market"}}$ = 1 / NKeys(department)
= **1 / 6**

Result size = NTuples(Employee) x Π RF
= NPages(Employee) x NTuplesPerPage(Employee) x $RF_{Salary > 300,000}$ x $RF_{department = \text{"Market"}}$
= 1200 x 120 x 0.5 x (1/6) = **12000** tuples

2.(b)

Plan 1:
Using clustered B+ tree on (department, salary):
Cost = $RF_{Salary > 300,000}$ x $RF_{department = \text{"Market"}}$ x [NPagges(I) + NPages(Employee)]
= (1/6) x 0.5 x (300 + 1200) = **125** (I/O)

Plan 2:
Using Heap Scan:
Cost = NPages(Employee) = **1200** (I/O)

125 < 1200
Therefore, the best plan is the **clustered B+ tree** on (department, salary) with a cost of **125** (I/O).

2.(c)

Plan 1:
Using unclustered B+ tree on (salary):
Cost = $RF_{Salary > 300,000}$ x [NPages(I) + NTuples(Employee)]
= 0.5 x (200 + 1200 x 120) = **72100** (I/O)

Plan 2:
Using Heap Scan:
Cost = NPages(Employee) = **1200** (I/O)

1200 < 72100
Therefore, the best plan is the **Heap Scan** with a cost of **1200** (I/O).

2.(d)

Plan 1:
Using unclustered Hash index on (department):
Cost = $RF_{department = \text{"Market"}}$ x 2.2 x NTuples(Employee)
= (1/6) x 2.2 x 1200 x 120 **= 52800** (I/O)

Plan 2:
Using Heap Scan:
Cost = NPages(Employee) = **1200** (I/O)

1200 < 52800
Therefore, the best plan is the **Heap Scan** with a cost of **1200** (I/O).

2.(e)

Hash index **cannot be** used for range queries.
The only available plan is the **Heap Scan** with a cost of **1200** (I/O).

Question 3

3.(a)

RF_{eid} = 1/NKeys(eid) = 1/NTuples(Employee) = **1 / 5000**

RF_{projid} = 1/NKeys(projid) = 1/NTuples(Project) = **1/60,000**

$RF_{salary<300,000}$ = [Value – Low(Salary)] / [High(Salary) – Low(Salary)]
= (300,000 – 100,000) / (500,000 – 100,000) = **0.5**

$RF_{code\ =\ \text{“alpha 340”}}$ = 1 / NKeys(code) = **1/1000**

Result size = NTuples(Employee) x NTuples(projid) x NTuples(department)
x RF_{eid} x RF_{projid} x $RF_{salary<300,000}$ x $RF_{code\ =\ \text{“alpha 340”}}$
= 5000 x 60000 x 20000 x (1/5000) x (1/60000) x 0.5 x (1/1000)
= **10** tuples

3.(b)

Plan [1]:

NPages(Employee) = 5000/100 = 50 pages
NPages(Project) = 60000/100 = 600 pages
NPages(Department) = 20000/100 = 200 pages

Cost to join Employee ⋈ Project = NPages(Employee) + NPages(Employee) x NPages(Project)
= 50 + 50 x 600 = **30050** (I/O)

Result size of Employee ⋈ Project = NTuples(Employee) x NTuples(Project) x (1/NKeys(eid))
= 5000 x 60000 x (1/5000) = 6000 tuples

NPages(Employee ⋈ Project) = 60000/100 = 600 pages

Cost to join with Department = NPages(Employee ⋈ Project) + NPages(Employee ⋈ Project)
x NPages(Department) – Npages(Employee ⋈ Project)
= 600 x 200 = **120000** (I/O) <u>#due to pipelining</u>

Overall cost = 30050 + 120000 = **150050** (I/O)

Plan [2]:

Cost to join Porject ⋈ Department = 3 x NPages(Project) + 3 x NPages(Department)
= 3 x (600 + 200) = **2400** (I/O)

Result size of Porject ⋈ Department = NTuples(Project) x NTuples(Department) x (1/NKeys(projid))
= 60000 x 20000 x (1/60000) = 20000 tuples

NPages(Porject ⋈ Department) = 20000/100 = 200 pages

Cost to join with Employee = 2 x NumPasses x NPages(Porject ⋈ Department)
+ 2 x NumPasses x NPages(Employee) + NPages(Porject ⋈ Department)
+ NPages(Employee) – Npages(Porject ⋈ Department)
= 4 x 200 + 5 x 50 = **1050** (I/O) <u>#due to pipelining</u>

Overall cost = 2400 + 1050 = **3450** (I/O)

Plan [3]:

Cost to select from Employee = $RF_{salary<300,000}$ x (NPages(Index on salary) + NPages(Employee))
= 0.5 x (10 + 50) = 30 (I/O)

Result size of $\sigma_{salary<300,000}$ (Employee) = NTuples(Employee) x $RF_{salary<300,000}$
= 5000 x 0.5 = 2500 tuples

NPages($\sigma_{salary<300,000}$ (Employee)) = 2500/100 = 25 pages

Cost to join $\sigma_{salary<300,000}$ (Employee) ⋈ Project = NPages($\sigma_{salary<300,000}$ (Employee))
+ NPages($\sigma_{salary<300,000}$ (Employee)) x NPages(Project)
– Npages($\sigma_{salary<300,000}$ (Employee))
= 25 x 600 = **15000** I/O #due to pipelining

Result size of $\sigma_{salary<300,000}$ (Employee) ⋈ Project
= NTuples($\sigma_{salary<300,000}$ (Employee)) x NTuples(Project)
x RF($\sigma_{salary<300,000}$ (Employee)))
= 2500 x 60000 x (1/2500) = 60000 tuples

NPages($\sigma_{salary<300,000}$ (Employee) ⋈ Project) = 60000 / 100 = 600 pages

Cost to join with Department = 3 x NPages($\sigma_{salary<300,000}$ (Employee) ⋈ Project)
+ 3 x NPages(Department) – Npages($\sigma_{salary<300,000}$ (Employee) ⋈ Project)
= 2 x 600 + 3 x 200 = **1800** (I/O) #due to pipelining

Overall cost = 30 + 15000 + 1800 = **16830** (I/O)

Plan [4]:

the index on Project.projid is clustered. The data pages of Project are already sorted by projid, so there is no need to sort Project.

Cost to join Porject ⋈ Department = 2 x NumPasses x NPages(Department) + NPages(Department)
+ (NPages(Index on projid) + NPages(Project))
= 5 x 200 + (200 + 600) = **1800** (I/O)

Result size of Porject ⋈ Department = 20000 tuples (from Plan [2])

NPages(Porject ⋈ Department) = 20000 / 100 = 200 pages

Cost to join with Employee = 2 x NumPasses x NPages(Porject ⋈ Department)
+ 2 x NumPasses x NPages(Employee) + NPages(Porject ⋈ Department)
+ NPages(Employee) – Npages(Porject ⋈ Department)
= 4 x 200 + 5 x 50 = **1050** (I/O)

Overall cost = 1800 + 1050 = **2850** (I/O)