1

a.

Npage(Parts) = NTuple(Parts)/ TupleperPage

= 60000/50

= 1200

Npage(Supply) = NTuple(Supply)/ TupleperPage

= 150000/50

= 3000

Cost = NPages(Parts) + NPages(Parts) x NPages(Supply)

= 1200 + 1200 x 3000 = **3601200 I/O**

b.

NBlocks(Parts) = ceil(NPages(Parts)/(Buffer - 2)) = ceil(1200/200) = 6

Cost = NPages(Parts) + NBlocks(Parts) x NPages(Supply)

= 1200 + 6 x 3000 = **19200 I/O**

c.

Cost = Sort(Parts) + Sort(Supply) + Merge

= 2 × NumPasses × NPages(Parts) + (2 × NumPasses × NPages(Supply))

+ NPages(Parts) + NPages(Supply)

= 5 × (1200 + 3000) = **21000 I/O**

d.

Cost = 3 × NPages(Parts) + 3 × NPages(Supply)

= 3 × (1200 + 3000) = **12600 I/O**

e.

Block Nested Loops Join with Buffer chosen so that the smaller table fits into

memory as a single block.

NPages(Parts)/(Buffer – 2) = 1 ⇒ Buffer = **1202 pages**

Cost = 1200 + 3000 = **4200 I/O**

2

a.

RF(Salary > 300000) = (High(Salary) – Value)/(High(Salary) – Low(Salary))
= (500000 – 300000)/ (500000 – 100000) = 1/2

RF(Department = 'Marketing') = 1/NKeys(Department) = 1/6

Result size = NTuples(Employee) × πRF

= NPages(Employee) × NTuplesPerPage(Employee) × RF(Salary > 300000)× RF(Department = 'Marketing')

= 1200 × 120 × 1/2 × 1/6 = **12000 tuples**

b.

Using clustered B+ tree on (Department, Salary):

Cost = RF(Department = 'Marketing') × RF(Salary > 300000)× (NPages(Index) + NPages(Employee))

= 1/6 × 1/2 × (300 + 1200) = 125 I/O

Using full table scan:

Cost = NPages(Employee) = 1200 I/O

The best plan is the clustered B+ tree on (Department, Salary) with a cost of **125 I/O**.

c.

Using unclustered B+ tree on (Salary):

Cost = RF(salary > 300000) × (NPages(Index) + NTuples(Employee))

= 1/2 × (200 + 1200 × 120) = 72100 I/O

Using full table scan: Cost = NPages(Employee) = 1200 I/O

The best plan is the full table scan with a cost of **1200 I/O.**

d.

Using unclustered hash on (Department):

Cost = RF(department = ‘Marketing’) × 2.2 × NTuples(Employee)

= 1/6 × 2.2 × 1200 × 120 = 52800 I/O

Using full table scan:

Cost = NPages(Employee) = 1200 I/O

The best plan is the full table scan with a cost of **1200 I/O**.

e.

Hash index cannot be used for range queries.

The only available plan is the full table scan with a cost of 1200 I/O.

3

a.

RFeid = 1/ NKeys(eid) = 1/ NTuples(Employee) = 1/ 5000

RFprojid =1/ NKeys(projid) = 1/ NTuples(Project) =1/ 60000

RF(salary < 300000) = (Value – Low(salary))/ (High(salary) – Low(salary)) =

(300000 – 100000) / (500000 - 100000) = 1/2

RF(code = 'alpha 340') = 1/ NKeys(code) = 1/ 1000

Result size = NTuples(Employee) × NTuples(Project) ×

NTuples(Department) × RFeid × RFprojid × RF(salary < 300000) × RF(code = 'alpha 340')

= 5000 × 60000 × 20000 × 1/5000 × 1/60000 × 1/2 × 1/1000

= **10 tuples**

b.(plan1)

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)

× NPages(Project)

= 50 + 50 × 600 = 30050 I/O

Result size of Employee ⋈ Project = NTuples(Employee) × NTuples(Project)

× 1/NKeys(eid)

= 5000 × 60000 × 1/5000 = 60000 tuples

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

Cost to join with Department = NPages(Employee ⋈ Project) +
NPages(Employee ⋈ Project) ×
NPages(Department) – NPages(Employee ⋈ Project)
= 200 × 600 = 120000 I/O
Overall cost = 30050 + 120000 = **150050 I/O**

b.(plan2)

Cost to join Project ⋈ Department = 3 × NPages(Project) + 3 ×
NPages(Department)
= 3 × (600 + 200) = 2400 I/O
Result size of Project ⋈ Department = NTuples(Project) ×
NTuples(Department) × 1/NKeys(projid)
= 60000 × 20000 × 1/60000 = 20000 tuples
NPages(Project ⋈ Department) = 20000 / 100 = 200 pages
Cost to join with Employee = 2 × NumPasses × NPages(Project ⋈
Department) + 2 × NumPasses × NPages(Employee) + NPages(Project ⋈
Department) + NPages(Employee) – NPages(Project ⋈ Department)
= 4 × 200 + 5 × 50 = 1050 I/O
Overall cost = 2400 + 1050 = **3450 I/O**

b.(plan3)

Cost to select from Employee
= RF(salary < 300000)× (NPages(Index on salary) + NPages(Employee))
= 1/2 × (10 + 50) = 30 I/O

Result size of σsalary < 300000(Employee) = NTuples(Employee) ×

RF(salary < 300000)

= 5000 × 1/2 = 2500 tuples

NPages(σsalary < 300000 (Employee)) = 2500 / 100 = 25 pages

Cost to join σsalary < 300000 (Employee) ⋈ Project = NPages(salary < 300000 (Employee)) + NPages(σsalary < 300000 (Employee)) × NPages(Project) – NPages(σsalary < 300000 (Employee))

= 25 × 600 = 15000 I/O

Result size of σsalary < 300000 (Employee) ⋈ Project

= NTuples(σsalary < 300000 (Employee)) × NTuples(Project) × 1/NKeys(projid)

= 2500 × 60000 × 1/60000 = 2500 tuples

NPages(σsalary < 300000 (Employee) ⋈ Project)

= 2500 / 100 = 25 pages

Cost to join with Department = 3 × NPages(σsalary < 300000 (Employee) ⋈ Project) + 3 × NPages(Department) – NPages(σsalary < 300000 (Employee) ⋈ Project)

= 2 × 25 + 3 × 200 = 650 I/O

Overall cost = 30 + 15000 + 650 = **15680 I/O**

b.(plan4)

Cost to join Department ⋈ Project = 2 × NumPasses × NPages(Department) + NPages(Department) + (NPages(Index on projid) + NPages(Project))

= 5 × 200 + (200 + 600) = 1800 I/O

Result size of Department ⋈ Project same as Plan 2 = 200 pages

Cost to join with Employee = 3 × NPages(Project ⋈ Department) + 3 × NPages(Employee) – NPages(Project ⋈ Department)

= 2 × 200 + 3 × 50 = 550 I/O

Overall cost = 1800 + 550 = **2350 I/O**