INFO20003 Semester 1, 2019

Assignment 3 – Query Processing and Query Optimisation
Due: 6:00pm Friday 24 May
Submission: Via LMS https://lms.unimelb.edu.au
Weighting: 10% of your total assessment. The assignment will be graded out of 20 marks.

Question 1 (5 marks)

Consider two relations called Item and OrderItem. Imagine that relation Item has 160,000 tuples and OrderItem has 200,000 tuples. Both relations store 100 tuples per a page. Consider the following SQL statement:

```
SELECT *
FROM Item INNER JOIN OrderItem
  ON Item.ItemID = OrderItem.ItemID;
```

We wish to evaluate an equijoin between OrderItem and Item, with an equality condition Item.ItemID = OrderItem.ItemID. There are 802 buffer pages available in memory for this operation. Both relations are stored as (unsorted) heap files. Neither relation has any indexes built on it.

Consider the alternative join strategies described below and calculate the cost of each alternative. Evaluate the algorithms using the number of disk I/O's (i.e. pages) as the cost. For each strategy, provide the formulae you use to calculate your cost estimates.

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

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

c) Sort-Merge Join. Assume that Sort-Merge Join can be done in 2 passes. (1 mark)

d) Hash Join. (1 mark)

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. (1 mark)

Question 2 (5 marks)

Consider a relation with the following schema:

Student (studentid, firstname, lastname, faculty, level, wam)

The Student relation has 800 pages and each page stores 80 tuples. The *faculty* attribute can take one of ten values (“Arts”, “Architecture”, “Business”, “Education”, “Fine Arts and Music”, “Law”, “Medicine”, “Science”, “Agricultural Science”, “Engineering”) and *wam* (Weighted Average Mark) can have values between 0 and 100 ([0,100]).

Suppose that the following SQL query is executed frequently using the given relation:

```
SELECT *
FROM Student
WHERE wam > 75 AND faculty = 'Arts';
```

Your job is to analyse the query plans and estimate the cost of the *best plan* utilizing the information given about different indexes in each part.

a) Compute the estimated result size for the query, and the reduction factor of each filter. (1 mark)

b) Compute the estimated cost of the *best plan* assuming that a *clustered B+ tree* index on *(faculty, wam)* is the only index available. Suppose there are 120 index pages. Discuss and calculate alternative plans. (1 mark)

c) Compute the estimated cost of the *best plan* assuming that an un*clustered B+ tree* index on *(wam)* is the only index available. Suppose there are 60 index pages. Discuss and calculate alternative plans. (1 mark)

d) Compute the estimated cost of the *best plan* assuming that an *unclustered Hash* index on *(faculty)* is the only index available. Discuss and calculate alternative plans. (1 mark)

e) Compute the estimated cost of the *best plan* assuming that an *unclustered Hash* index on *(wam)* is the only index available. Discuss and calculate alternative plans. (1 mark)

Question 3 (10 marks)

Consider the following relational schema and SQL query. The schema captures information about customers, their orders, and orderitems.

Customer (cusid: integer, postcode: char(4), name: char(20))
Order (orderid: integer, orderdate: date, cusid: integer, departmentid: integer)
OrderItem (orderitemid: integer, handling: char(20), orderid: integer,
itemname: char(15), quantity: integer)

Consider the following query:

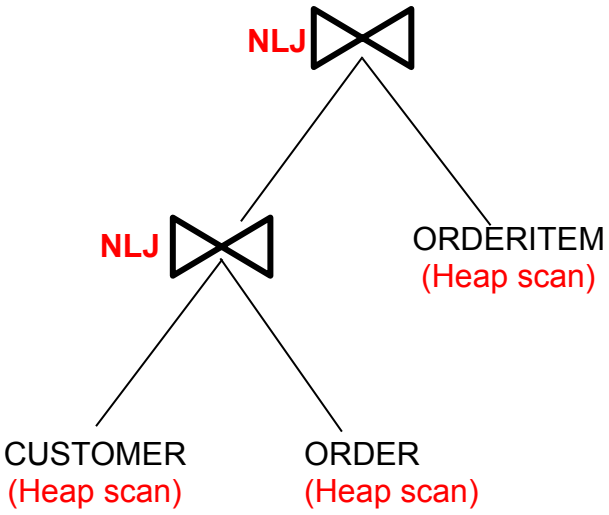
```
SELECT c.postcode, o.departmentid
FROM Customer c, Order o, OrderItem oi
WHERE c.cusid = o.cusid AND o.orderid = oi.orderid
  AND oi.quantity < 10 AND oi.itemname = 'Rotring 600';
```

The system's statistics indicate that there are 400 different *itemname* values, and *quantity* of order items range from 0 to 50 ([0,50]). There are a total of 80,000 Orderitems and 8,000 Orders and 2,000 Customers in the database. Each relation fits 100 tuples in a page. Suppose there exists a *clustered B+ tree* index on *(Order.orderid)* of size 20 pages, and suppose there is a *clustered B+ tree* index on *(Orderitem.quantity)* of size 200 pages.

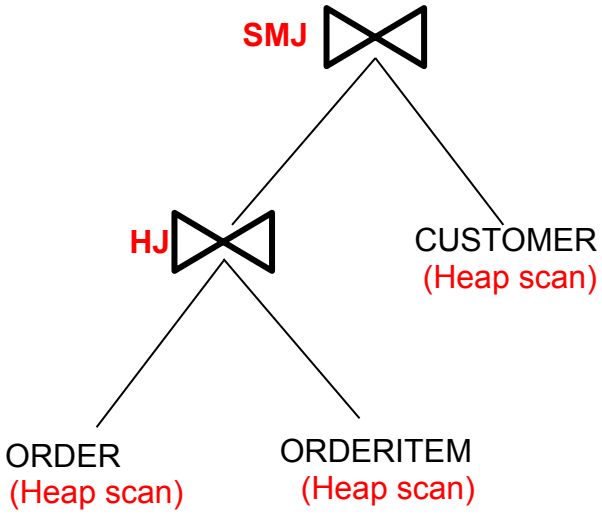
a) Compute the estimated result size and the reduction factors (selectivity) of this query. (2 marks)

b) 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 *cusid* is the candidate key of the Customer relation, and *orderid* is the candidate key of the Order relation. Assume that 100 tuples of a resulting join between Customer and Order fit in a page. Similarly, 100 tuples of a resulting join between Order and OrderItem fit in a page. If selection over filtering predicates is not marked in the plan, assume it will happen on-the-fly after all joins are performed, as the last operation in the plan. (8 marks, 2 marks per plan)

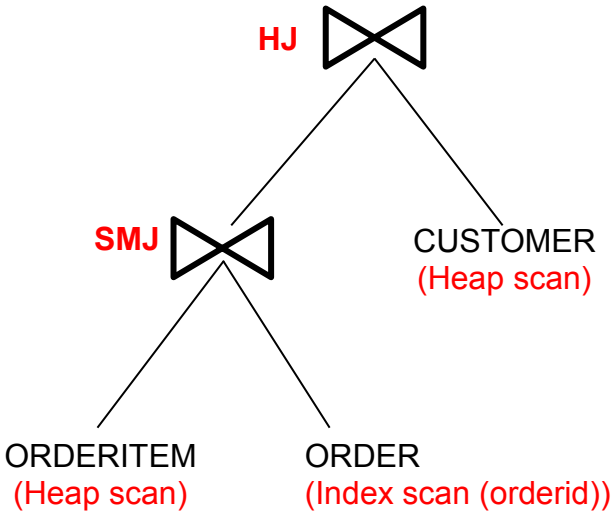
[1]


NLJ
NLJ
ORDERITEM
(Heap scan)
CUSTOMER
(Heap scan)
ORDER
(Heap scan)

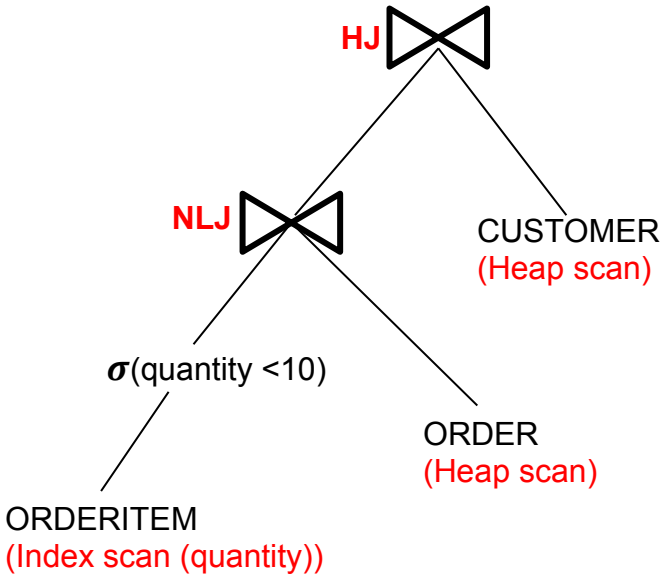
[2]


SMJ
HJ
CUSTOMER
(Heap scan)
ORDER
(Heap scan)
ORDERITEM
(Heap scan)

[3]


HJ
SMJ
CUSTOMER
(Heap scan)
ORDERITEM
(Heap scan)
ORDER
(Index scan (orderid))

[4]


HJ
NLJ
CUSTOMER
(Heap scan)
σ(quantity <10)
ORDER
(Heap scan)
ORDERITEM
(Index scan (quantity))

Formatting Requirements:

For each question, present an answer in the following format:

- Show the question number before each question. You do not need to include the text of the question itself.
- Show your answer in **blue** text (please type your answers on a computer).
- Start Question 2 and Question 3 on a new page.
- For each of the calculations, provide all the formulae you used to calculate your cost estimates, and show your working, not only the result.

Submission Process:

Submit a single PDF showing your answers to all questions to the Assessment page on LMS by 6pm on the due date of Friday 24 May. Name your file 'STUDENT_ID'.pdf, where STUDENT_ID corresponds to YOUR student id.

Requesting a Submission Deadline Extension

If you need an extension due to a valid (medical) reason, you will need to provide evidence to support your request by 9pm, Thursday 23 May. Medical certificates need to be at least two days in length.

To request an extension:

1. Email the head tutor, Alan Thomas (alan.thomas@unimelb.edu.au) from your university email address, supplying your student ID, the extension request and supporting evidence.
2. If your submission deadline extension is granted, you will receive an email reply granting the new submission date. Replies may take up to 12 hours, so please be patient.

Reminder: INFO20003 Hurdle Requirements

To pass INFO20003 you must pass two hurdles:

- Hurdle 1: Obtain at least 50% (15/30) or higher for the three assignments (each worth 10%)
- Hurdle 2: Obtain a grade of 50% (35/70) or higher for the End of Semester Exam

Therefore, it is our recommendation to students that you attempt every assignment and every question in the exam.

GOOD LUCK!