**QUESTION 1**

**Not examinable anymore**

**QUESTION 2**

**ai)**

| trades\_on | = 25000

Key = (cname, xname) => | key | = 20 + 20 = 40B  
Value = ptr to tuple => | value | = 4B (32 bit addressable)

=> | tuple | = 40 + 4 = 44B

Num\_tuples\_per\_page = (pg\_size - extra\_ptr\_size) / tuple\_size = (2048 - 4) / 44 = 46.45...

Pages are unspanned so 46 tuples per page

Depth = floor( log\_46 (25000) ) = 2

Number of pages:

Depth 2 (leaves): ceil( 25000 / 46) = 544

Depth 1: ceil (544 / 46) = 12

Depth 0 (root): ceil ( 12 / 46 ) = 1

**aii)** (unsure about this)

RowSize(company) = 2 + 20 + 6 = 28B

RowSize(trades\_on) = 20 + 20 + 6 = 46B

Total pages for company = ceil ( (28 \* 20000) / 2048 ) = 274

Total pages for trades\_on = ceil ( (46 \* 25000) / 2048 ) = 562

Scanning cost = 274 + 562 = 836 faults

**aiii A)**

s = 0.15, n = 46, we have p(s, n) = 1 - (1 – s)^n = 0.9943...

Ceil ( p(s, n) \* (25000 / 46) ) = 544

... ?

**aiii B)** (also unsure about this)

Depth = 2 => 3 page faults for each tuple

| country | = 20000

=> 20000 \* 3 = 60000 faults

**bi)**

Push projection inside join

Only cname and sector from S1 to S2

(20 + 6) \* | country | = 520000B

**bii)**

Send cnames of companies with assets >= 10000 from S2 to S1

20 \* 18000 \* 0.2 = 72000

Send cname and sector for (18000 \* 0.2) companies from S1 to S2

18000 \* 0.2 \* (20 + 6) = 93600

Total = 72000 + 93600 = 165600B

**biii)**

Equality when

(20 \* 18000 \* f) + (26 \* 18000 \* f) = 520000

=> f = 520000 / (18000 \* 46 ) = 62.8%

**QUESTION 3**

**ai)**

company\_and\_country =

JOIN company BY hq,

country BY iso\_code;

min\_company\_and\_country =

FOREACH company\_and\_country

GENERATE country::iso\_code AS code,

company::sector AS sector;

country\_companies =

GROUP min\_company\_and\_country

BY code;

country\_data =

FOREACH country\_companies {

tech = FILTER min\_company\_and\_country

BY sector == ‘Tech’;

oil = FILTER min\_company\_and\_country

BY sector == ‘Oil’;

GENERATE group AS code,

COUNT(tech) AS no\_tech,

COUNT(oil) AS no\_oil;

}

// is there a way to get short\_name without the extra join

country\_data\_w\_detail =

JOIN country\_data BY code,

country BY iso\_code;

country\_data\_w\_name =

FOREACH country\_data\_w\_detail

GENERATE country::short\_name,

country\_data::no\_tech,

country\_data::no\_oil;

**aii)**

Join REPLICATED (copy the country table to each map node since it fits)

**bi)**

Possible to split any table with cname as key

office\_i = office ⋉ company\_i

public\_company\_i = public\_company ⋉ company\_i

trades\_on\_i = trades\_on ⋉ company\_i

**bii)**

LTM.S1: same query without where clause and without grouping

LTM.S2: same query without grouping

LTM.S3: -

GTM: union of S1 and S2 and then Grouped by hq

**QUESTION 4**

**ai)**

r\_1[Cus] = r1us because im lazy

r1us, r2fr, r2us, r2gb, r2ie, **w1us, w2us,** r1gb, w1gb, c2, c1

**aii)**

All of H1 without the commit, all of H3, c1

**aiii)**

Reads from 2, reads from 3, write from 2, write from 3, c2, c3

**bi)**

All of the products whose stock has always been at least 100

HRO6

**bii)**

Products which have been in stock at the same time as HR03, since HR03 came into stock

DR10, DR11, HR03, HR06

**biii)**

Π\_(max(stock\_level)) Γ(pno) [◆ product]