Question 1 – SQL Server Databases

Present SQL Server T-SQL commands for accomplishing the following tasks:

(a) Create a database named ***NutrientsDB***, containing **one log file** and **three different data files**, in **three distinct filegroups** (i.e., one data file in each filegroup). The log file should have an initial size of 25MB and a maximum size of 250MB. All data files should have an unlimited maximum size, except the one in the primary filegroup, which should have a maximum size of 1GB). The first data file on the first secondary filegroup should have an initial size of 100MB, and the remaining files should have an initial size of 50MB. All files should grow at a rate of 50%, except for the data file in the primary filegroup, which should grow by 5MB, every time this is required.

(b) Create a table named ***Cheese*** in the ***NutrientsDB*** database. The table should have a numeric attribute named ***cheeseID***, that identifies the individual records, an alphanumeric attribute named ***Type***, and four other numeric attributes named ***Calories, Proteins, Carbohidrates,*** and ***Fat.*** The table should be partitioned so that all tuples where *cheeseID* is less or equal than 50 are physically stored in the primary filegroup, all tuples where the *cheeseID* is greater than 50, but less or equal than 100, are physically stored in the first secondary filegroup, and the remaining tuples are physically stored in the second secondary filegroup.

(c) In the table named Cheese**,** the amount of calories is stored in an attribute named ***Calories,*** in Kcals per 100 grams. Create an index over the table with a search key corresponding to the calories in cals per 100 grams, including the amount of protein and fat as additional attributes that are not part of the search key. The index should be physically stored in the primary filegroup. Indicate also if the index is clustered or non-clustered, justifying.

Question 2 – B+Tree Index Structures

Consider the problem of inserting the following keys, in the given order, into an empty B+-tree where nodes can hold up to 3 values:

Parmesão, Ilha, Camembert, Fresco, Requeijão, Azeitão, Alverca, Serra, Alcobaça, Roquefort, Flamengo, Emmental, Évora, Creme, Serpa, Quark

(a) Draw the tree after each insertion.

(b) Delete the following keys from the B+tree data structure from the previous exercise: *Ilha; Flamengo; Emmental; Serpa*. Draw the tree after each deletion.

3 – Extendable Hashing Index Schemes

Consider the extendable hashing indexing mechanism introduced in the theoretical classes. Show how data records with the following keys can be stored in buckets which individually can hold 2 records, using extendable hashing and considering the **least significant bits first** in the directory of the resulting data structure.

Azeitão, Camembert, Emmental, Serra , Camembert, Camembert, Creme, Alverca curado

Consider that binary representations for keys were determined by a simple hash function, resulting in:

Azeitão	11100011
Camembert	01101000
Emmental	01001111
Serra	11001101
Creme	11001110
Alverca	00110110

4 – Estimating the Cost of Relational Algebra Operations

Consider the following relational schema:

CheeseProvenance(cheese-name, region-name)
Location(region-name, climate-type)

The relation CheeseProvenance stores information about the region where each cheese type is produced, and the relation Location stores information about the the regions that produce cheese.

All tuples have fixed size. The relation CheeseProvenance takes 1000 blocks and the relation Location has 2300 blocks. Each page of CheeseProvenance contains 120 tuples and each page of Location contains 100 tuples.

Compute the number of I/Os performed by each of the following algorithms:

(a) Selection on the Location relation where the filtering condition is *climate-type = 'Dry'*, assuming there is an index on the table over the attribute climate-type.
(b) Block Nested Loop Join, with CheeseProvenance as the outer relation and the join condition is on region-name. Present the costs of the worst and best cases.
(c) Sort-Merge Join, assuming that only the relation Location is ordered on region-name, the relation CheeseProvenance is ordered on cheese-name and that you can have 3 pages in memory when sorting the relations.

5 – Query Optimization and Estimation of Join Sizes

Consider the two relations of Question 4. Consider also the following information, regarding the two relations:

V(climate, Location) = 20

Estimate the number of tuples that results from the expression:

$$CheeseProvenance \bowtie (\sigma_{climate='dry'} Location)$$

6 - External talk: *Casos Reais na Administração de Bases de Dados*

Answer the following questions, on the subject of the invited talk given by engineer Wilson Lucas, that you had the opportunity to attend on the 24th of May 2015.

(a) One of the tasks of a DBA is to assure the high availability of the Databases being managed. This can be done in several ways. However, independently of the technique used, there are always trade-offs that one must take into account when choosing how to implement it (or even if it is worth implementing). Name and explain one such trade-off.

(b) In theory, once our database is fully optimized, it should not be necessary to change it any further. In practice, on a database that is being used in a functioning organization, this is not the case. Explain why.