Exercise Sheet 1

1 SQL99/SQL2003 Queries

Pompier

id_pompier	id_caserne	nom	nom_ville
1	2	Valjean	Papou
1	3	Montant	Shadok
2	3	Montant	Papou
3	3	Savairien	Papou
2	2	Conchon	Le Luc
4	3	Jumper	Le Luc

Primary Key: id_pompier, id_caserne
Foreign Key: Pompier.nom_ville→ Ville.nom_ville

Ville

nom_ville	nb_hab	cp
Draguignan	20000	83240
Brignoles	1650	83620
Le Luc	4580	83620
Lajoie	25000	83450
Shadok	14630	83666
Papou	3655	83220

Primary Key: nom_ville

Figure 1: A synthetic database

Exercise 1 (SQL Queries)

1. Give the result of each query below:[1]

(a)
```
SELECT id_caserne, id_pompier,
CASE WHEN GROUPING(id_pompier) = 1
  THEN 'nb_p:'|| COUNT(*)
  ELSE 'no:'|| RANK() OVER (PARTITION BY id_caserne ORDER BY nom)
END AS nbp
FROM Pompier
GROUP BY id_caserne, ROLLUP((id_pompier,nom))
ORDER BY id_caserne, nbp
```

(b)
```
SELECT REGEXP_REPLACE(nom, '^(.)(.*)(.)$','\1***\3') n
FROM Pompier
ORDER BY n
```

2. For each of the queries below, indicat if it is correct (i.e., can be evaluated)? If not, point out all the mistakes that prevent its proper execution. If the query is correct but poorly written (i.e., can be simplified), simplify it (the alternative query must of course return the same result).

(a)
```
SELECT nom_ville,
SUM(SUM(nb_hab)) OVER (PARTITION BY nom_ville) pop
FROM ville
GROUP BY nom_ville
ORDER BY nom_ville
```

(b)
```
SELECT SUM(SUM(nb_hab)) OVER () pop
FROM ville
```

Exercise 2 (Regular Expressions)

1. Give a regular expression that validates (french) phone numbers: 5 groups of 2 digits separated by a "." or " "(the same separator must be used between all groups). We will also restrict the first digit to be a 0 and the second to be some digit between 1 and 9 (we do not validate special numbers).

2. Give a `sed` instruction to reorder columns of a csv file having 3 columns separated with commas[2]. We expect the file to contain the columns in reverse order after the instruction is executed (column3, then 2, then 1 on every line). sed -i.b -E 's/([^ ,]),([^ ,]),(*)/ \3,\3,\1 /g' file.txt

3. Give an SQL instruction for Oracle that replaces any sequence of 2 or more spaces with a single space in column `nom_ville` of table `Pompier`.

[1] It may help to recall that `||` is the SQL syntax for string concatenation. We will assume that NULLs are ranked last in increasing order (this is Oracle's default, with NULLS FIRST in decreasing order).

[2] Remark (fyi): a previous version of this exercise sheet discussed the same issue with `.tsv` files, i.e., files separated with tabs, sometimes abusively referred to as csv files. But depending on the system, sed may not recognize \t as a tab (Linux does, but on FreeBSD Mac: type a real tab instead with `Ctrl+v then Ctrl+i`)

UPDATE Pompier
SET nom_ville REGEXP_REPLACE(nom_ville,'{2,}',' ')