Eric THIERRY M2CCI

**TP2 Base de données**

**Agence de voyage**

*14/12/16*

**Requête 9 : Donner la liste des numéros des circuits qui passent dans toutes les villes d’un pays donné.**

Schéma et spécification :

R9 (numC)<x> appartient à R9 ⬄ Le circuit de numéro x passe par toutes les villes du pays donné.

Requête SQL :

ACCEPT nomPays PROMPT 'Donner un nom de pays (ex : France) : ';

SELECT numC

FROM (SELECT numC, COUNT(ville) AS villeVisitee

FROM (SELECT nomV AS ville

FROM AGENCE.LesVilles

WHERE pays =&nomPays)A

/\*A(ville)<x> appartient à A ⬄ La ville x est une ville du pays donné\*/

NATURAL JOIN (SELECT numC, vDep AS ville FROM AGENCE.LesCircuits

union

SELECT numC, vArr

FROM AGENCE.LesCircuits

union

SELECT numC, vEtape

FROM AGENCE.LesEtapes)B

/\*B(numC, ville)<x,y> appartient à B ⬄ La ville y fait partie du circuit x\*/

GROUP BY numC)C

/\*C(numC, villeVisitee)<x,y> appartient à C ⬄ Le circuit de numéro x visite y ville du pays donné\*/

JOIN (SELECT COUNT(nomV) AS nbville

FROM AGENCE.LesVilles

WHERE pays = &nomPays)D

/\*D(nbVille)<x> appartient à D ⬄ x est le nombre de ville du pays donné\*/

on (nbville = villeVisitee);

Résultats requête SQL :

***Résultat de la requête R9 avec Norvège comme pays donné :***

Donner un nom de pays (ex : France) : Norvege

NUMC

----------

16

21

***Résultat de la sous-requête A avec Norvège comme pays donné :***

VILLE

-------------

Cap Nord

Hammerfest

Inari

Saarijarvi

Salla

Tornio

Ylivieska

***Résultat de la sous-requête B:***

NUMC VILLE

---------- -------------

1 Londres

1 Paris

2 Londres

2 Paris

3 Amsterdam

3 Paris

4 Amsterdam

4 Paris

5 Lyon

5 Venise

6 Paris

6 Venise

7 Hoedic

7 Houat

7 Port Cotton

7 Port Maria

7 Quiberon

8 Besse

8 Clermont Ferrand

8 Laschamps

8 Orcival

9 Bath

9 Exeter

9 Londres

9 Paris

9 Salisbury

9 Sissinghurst

9 St Ives

10 Bantry

10 Cork

10 Dingle

10 Glengariff

10 Kenmare

10 Killarney

10 Shannon

11 Cleggan

11 Clifden

11 Ennis

11 Galway

11 Leenan

11 Louisburgh

11 Shannon

11 Westport

12 Ardara

12 Carrick

12 Donegal

12 Dublin

12 Dungloe

12 Killibegs

12 Letterkenny

13 Christianshab

13 Copenhague

13 Disko

13 Egedesmine

13 Godhavn

13 Holsteinborg

13 Jakobshavn

13 Paris

13 Sondre

14 Akureyri

14 Asbyrgi

14 Geysir

14 Gullfoss

14 Hveravellir

14 Myvatn

14 Reykjavik

15 Godafoss

15 Husavik

15 Paris

15 Reykholt

15 Reykjavik

16 Cap Nord

16 Hammerfest

16 Helsinky

16 Inari

16 Paris

16 Saarijarvi

16 Salla

16 Tornio

16 Ylivieska

17 Castelo de Vide

17 Evora

17 Faro

17 Fatima

17 Leiria

17 Lisbonne

17 Porto

17 Urgeirica

17 Vila Real

18 Florence

18 Paris

18 Ravenne

18 Rome

18 Venise

18 Verone

19 Bari

19 Benevento

19 Brindisi

19 Capri

19 Caserte

19 Catanzaro

19 Cosenza

19 Lecce

19 Metaponto

19 Naples

19 Pompei

19 Rome

19 Salerne

20 Briancon

21 Cap Nord

21 Florence

21 Hammerfest

21 Helsinky

21 Inari

21 Paris

21 Ravenne

21 Rome

21 Saarijarvi

21 Salla

21 Tornio

21 Venise

21 Verone

21 Ylivieska

***Résultat de la sous-requête C avec Norvège comme pays donné :***

NUMC VILLEVISITEE

---------- ------------

16 7

21 7

***Résultat de la sous-requête D avec Norvège comme pays donné :***

NBVILLE

----------

7

Tests de la requête :

**Requête 11 : Donner le numéro, le prix de base (sans tenir compte du prix des monuments visités), la date de départ et le nombre de places disponibles et dont le nombre de jours est inférieur ou égale à un entier donné.**

Schéma et spécification de la requête :

R11(numC, prix, dateDep, Dispo)<w,x,y,z> appartient à R11 ⬄ le circuit de numéro *w* a pour date de départ *y*. Son prix est *x* et il reste *z* places disponibles.

Requête SQL :

ACCEPT choixJour PROMPT 'Choisissez le nombre de jours pour votre voyage :';

SELECT numC, dateDep, prix, Dispo

FROM (SELECT numC, dateDep, prix, (nbplaces - nvl(nbReserve, 0)) AS Dispo

FROM AGENCE.LesProgrammations NATURAL LEFT OUTER JOIN

(SELECT numC, datedep, SUM(nbRes) AS nbReserve FROM AGENCE.LesReservations

GROUP BY numC, datedep)A

/\*A(numC, dateDep, nbReserve)<x,y,z> appartient à A ⬄ le circuit de numéro *x* a pour date de départ *y* et possède *z* places de reservées\*/

NATURAL JOIN AGENCE.LesCircuits

WHERE (nbplaces - nvl(nbReserve, 0)) > 0)B

/\*B(numC, dateDep, prix, Dispo)<w,x,y,z> appartient à B ⬄ le circuit de numéro *w* a pour date de départ *x*. Son prix est *y* et il reste *z* places disponibles. \*/

NATURAL JOIN (SELECT numC, SUM(nbjours) AS totaljour FROM AGENCE.LesEtapes

GROUP BY numC

HAVING SUM(nbjours) <= &choixJour)C

/\*C(numC, totaljour)<y,z> appartient à C ⬄ Le circuit de numéro *y* dure *z* jours \*/

ORDER BY numC, dateDep;

Résultats de la requête SQL :

***Résultat de la requête R11 avec 4 comme entier entré :***

Choisissez le nombre de jours pour votre voyage : 4

NUMC DATEDEP PRIX DISPO TOTALJOUR

---------- --------- ---------- ---------- ----------

1 04-JAN-10 1160 34 2

1 04-FEB-10 1160 8 2

1 06-FEB-10 1160 34 2

1 24-JUL-10 1160 10 2

2 06-JAN-10 1160 2 2

2 07-JAN-10 1160 9 2

2 06-FEB-10 1160 12 2

2 05-SEP-10 1160 30 2

3 24-DEC-09 1040 13 3

3 31-DEC-09 1040 45 3

3 03-JUL-10 1040 1 3

4 30-JUN-10 1270 11 4

4 06-AUG-10 1270 10 4

4 31-AUG-10 1270 1 4

4 06-NOV-10 1270 18 4

6 06-FEB-10 2520 22 4

6 06-SEP-10 2520 10 4

6 06-OCT-10 2520 12 4

6 16-NOV-10 2520 34 4

***Résultat de la sous-requête A :***

NUMC DATEDEP NBRESERVE

---------- --------- ----------

8 28-FEB-10 7

3 03-JUL-10 11

7 16-DEC-09 51

13 01-JAN-10 10

12 06-FEB-10 79

18 06-DEC-10 11

8 14-FEB-10 1

8 16-FEB-10 1

4 30-JUN-10 88

5 31-AUG-10 2

2 07-JAN-10 25

19 15-APR-10 20

5 06-NOV-10 43

21 15-JAN-10 24

13 31-DEC-09 48

14 26-JUL-10 21

20 27-JAN-10 5

1 21-JUL-10 10

2 05-FEB-10 99

1 04-FEB-10 4

10 01-JAN-10 2

9 30-OCT-10 10

10 11-FEB-10 2

***Résultat de la sous-requête B :***

NUMC DATEDEP PRIX DISPO

---------- --------- ---------- ----------

1 24-JUL-10 1160 10

1 06-FEB-10 1160 34

1 04-FEB-10 1160 8

1 04-JAN-10 1160 34

2 05-SEP-10 1160 30

2 06-FEB-10 1160 12

2 07-JAN-10 1160 9

2 06-JAN-10 1160 2

3 03-JUL-10 1040 1

3 31-DEC-09 1040 45

3 24-DEC-09 1040 13

4 06-NOV-10 1270 18

4 31-AUG-10 1270 1

4 06-AUG-10 1270 10

4 30-JUN-10 1270 11

5 06-NOV-10 2740 3

5 31-AUG-10 2740 64

6 16-NOV-10 2520 34

6 06-OCT-10 2520 12

6 06-SEP-10 2520 10

6 06-FEB-10 2520 22

7 31-OCT-10 2500 39

7 31-AUG-10 2500 56

7 26-FEB-10 2500 1

7 06-JAN-10 2500 1

7 16-DEC-09 2500 1

8 14-NOV-10 2140 3

8 31-AUG-10 2140 10

8 21-JUL-10 2140 14

8 16-MAY-10 2140 12

8 26-APR-10 2140 31

8 28-FEB-10 2140 4

8 21-FEB-10 2140 13

8 16-FEB-10 2140 11

8 31-DEC-09 2140 10

8 24-DEC-09 2140 18

9 30-OCT-10 5700 21

9 06-FEB-10 5700 3

10 28-FEB-10 6170 22

10 11-FEB-10 6170 28

10 21-JAN-10 6170 45

10 01-JAN-10 6170 78

11 30-JUN-10 6270 12

11 29-MAY-10 6270 34

11 28-FEB-10 6270 3

11 06-FEB-10 6270 13

12 06-FEB-10 6170 111

13 31-DEC-10 18590 44

13 30-JUN-10 18590 52

13 31-MAY-10 18590 3

13 06-MAY-10 18590 60

13 30-APR-10 18590 15

13 06-FEB-10 18590 99

13 01-JAN-10 18590 58

13 31-DEC-09 18590 3

14 26-JUL-10 7700 4

14 06-JUL-10 7700 12

15 31-AUG-10 8560 18

15 06-FEB-10 8560 11

16 31-AUG-10 10180 17

16 06-FEB-10 10180 12

16 06-JAN-10 10180 3

17 31-DEC-10 3990 3

17 26-FEB-10 3990 34

17 16-FEB-10 3990 12

17 06-FEB-10 3990 3

18 06-DEC-10 6690 1

18 06-OCT-10 6690 40

18 30-JUL-10 6690 90

18 30-APR-10 6690 15

19 20-DEC-10 10400 11

19 10-DEC-10 10400 28

19 16-SEP-10 10400 10

19 06-SEP-10 10400 20

19 05-AUG-10 10400 12

19 15-APR-10 10400 25

19 06-FEB-10 10400 10

21 06-APR-10 15000 99

21 26-FEB-10 15000 12

21 25-JAN-10 15000 3

21 15-JAN-10 15000 42

20 03-APR-10 450 87

20 01-APR-10 450 13

20 06-FEB-10 450 1

20 02-FEB-10 450 14

20 20-JAN-10 450 12

20 22-DEC-09 450 18

Tests de la requête :

**Requête 13 : Pour chaque programmation de circuit, retrouver les pays dans lequel passe le circuit et la date à laquelle le circuit arrive dans ce pays**

Schéma et spécification de la requête :

R13(numC, dateDep, pays, dateEntree)<w,x,y,z> appartient à R13 ⬄ la programmation de la date x du circuit w arrive dans le pays y à la date z

Requête SQL :

SELECT numC, dateDep, pays, (NVL(nbJ,0) + dateDep) AS dateEntree

FROM (SELECT A1.numC, A1.rang, A1.pays

FROM (SELECT numC, rang, pays

FROM AGENCE.LesEtapes JOIN AGENCE.LesVilLes ON (vEtape = nomV))A1

JOIN (SELECT numC, rang, pays

FROM AGENCE.LesEtapes JOIN AGENCE.LesVilLes ON (vEtape = nomV))A2 ON (A1.pays <> A2.pays AND A2.rang = (A1.rang - 1) AND A1.numC = A2.numC)

UNION

SELECT numC, rang, pays

FROM AGENCE.LesEtapes JOIN AGENCE.LesVilLes ON (vEtape = nomV)

WHERE rang = 1)B NATURAL LEFT OUTER JOIN (SELECT E1.numC, E1.rang, SUM(E2.nbJours) AS nbJ

FROM AGENCE.LesEtapes E1 JOIN AGENCE.LesEtapes E2 ON (E1.rang > E2.rang AND E1.numC = E2.numC)

GROUP BY E1.numC, E1.rang)C NATURAL JOIN AGENCE.LesProgrammations

UNION

SELECT numC, dateDep, pays, dateDep as dateEntree

FROM (SELECT numC, pays, dateDep

FROM AGENCE.LesCircuits JOIN AGENCE.LesVilLes ON (vDep = nomV) NATURAL JOIN AGENCE.LesProgrammations)D

UNION

SELECT numc, dateDep, pays, (nbJ + dateDep) AS dateEntree

FROM (SELECT numC, SUM(nbJours) as nbJ

FROM AGENCE.LesEtapes

GROUP BY numC)E NATURAL JOIN AGENCE.LesProgrammations NATURAL JOIN AGENCE.LesCircuits JOIN AGENCE.LesVilLes ON (vArr = nomV)

ORDER BY numC, dateDep, dateEntree;

Résultats de la requête SQL :

Tests de la requête SQL :