

DJILLALI LIABES UNIVERSITY OF SIDI BEL ABBES  
FACULTY OF EXACT SCIENCES  
DEPARTMENT OF COMPUTER SCIENCES



*Module : Bases de données Avancés II*  
1ST YEAR OF MASTER'S DEGREE IN  
NETWORKS, INFORMATION SYSTEMS & SECURITY (RSSI)  
2021/2022

---

**Bases de données Avancés II**  
**TP-01**

---

*Author:*  
HADJAZI M.Hisham  
AMUER Wassim Malik  
*Group: 01 / RSSI*

*Supervisor:*  
Dr. SLAMA Zohra

*A paper submitted in fulfillment of the requirements for the*  
TP-01

November 7, 2021

# Contents

<b>1</b>	<b>Solutions of Fiche TP-01</b>	<b>1</b>
1.1	Partie 1	2
1.1.1	Exprimez, lorsque cela est possible, les requêtes suivantes en algèbre relationnelle, en calcul relationnel de tuples et en SQL.)	2
	1. Donnez les numéros, noms et prénoms des employés inscrits dans les sports gérés par le local de nom «Victoire».	7
	2. Donnez les numéros des employés qui ne sont inscrits à aucun sport du Local n° 1.	8
	3. Numéros des sports assurés dans les salles 1 ou 2.	9
	4. Numéros des sports aux quels sont inscrits tous les employés.	10
	5. Numéros des employés inscrits à tout sport du local n° 1.	11
1.2	Partie 2	12
1.2.1	1. A partir des énoncés ci-dessus, créez les tables base de données et procédez à leurs alimentations par les données suivantes :	12
	Employe (numEmploye, nom, prenom, age, adresse)	12
	Sport (numSport, nomSport, numLocal, salle)	13
	Inscription (numSport, numEmploye, date, nbreHeures)	15
	Local-de-Sport (numLocal, nomLocal)	18
1.2.2	2. Une fois les tables créées et alimentées, implémentez les requêtes SQL exprimées dans la partie 1.	19
	1. Donnez les numéros, noms et prénoms des employés inscrits dans les sports gérés par le local de nom «Victoire».	19
	2. Donnez les numéros des employés qui ne sont inscrits à aucun sport du Local n° 1.	20
	3. Numéros des sports assurés dans les salles 1 ou 2.	21
	4. Numéros des sports aux quels sont inscrits tous les employés.	22
	5. Numéros des employés inscrits à tout sport du local n° 1.	23
1.2.3	3. Utilisez la commande explain plan pour calculer les coûts d'exécutions des requêtes. Comparez et commentez les résultats.	24

## Chapter 1

# Solutions of Fiche TP-01

### Notes regarding this solution :

This solution and the executions of the code in it was done in the following machine :

- *Machine*: Lenovo Ideapad S210
- *CPU*: Intel Celeron 1037U 1800 MHz
- *RAM*: 8GB DDR3l
- *OS* : Linux Mint 20.2 Cinnamon Kernel v.5.4.0-88
- *IDE* : SQL PLUS
- *Oracle version*: Express Edition 11g R2

## 1.1 Partie 1

Soit le schéma de la base de données relationnelle suivant :

**Employe** (numEmploye, nom, prenom, age, adresse)  
**Sport** (numSport, nomSport, numLocal, salle)  
**Inscription** (numSport, numEmploye, date, nbreHeures)  
**Local-de-Sport** (numLocal, nomLocal)

Inscription modélise l'inscription d'un employé identifiée par num Employe à un sport identifié par numSport. Un sport est géré dans un Local-Sport identifié par numLocal.

### 1.1.1 Exprimez, lorsque cela est possible, les requêtes suivantes en algèbre relationnelle, en calcul relationnel de tuples et en SQL.)

```
CREATE TABLE Employe(  
    numEmploye int not null PRIMARY KEY,  
    nom char(30), prenom char(30),  
    age int, adresse varchar(100));  
  
CREATE TABLE Locale_de_Sport(  
    numLocal int not null PRIMARY KEY,  
    nomLocal varchar(200));  
  
CREATE TABLE Sport(  
    numSport int not null PRIMARY KEY,  
    nomSport char(30),  
    numLocal int,  
    salle varchar(100),  
    FOREIGN KEY(numLocal) REFERENCES Locale_de_Sport(numLocal));  
  
CREATE TABLE Inscription (  
    numSport int,  
    numEmploye int,  
    Inscription_date date,  
    nbreHeures int,  
    FOREIGN KEY(numSport) REFERENCES Sport(numSport),  
    FOREIGN KEY(numEmploye) REFERENCES Employe(numEmploye));
```

We then Inserted to the tables random data.

```
INSERT INTO Employe VALUES(1, 'Hadjazi', 'Hisham', 20, 'SBA');  
INSERT INTO Employe VALUES(2, 'Obaba', 'Maroko', 20, 'Tindouf');  
INSERT INTO Employe VALUES(3, 'Sahraoui', 'Amine', 31, 'SBA');  
INSERT INTO Employe VALUES(4, 'Merioua', 'Adel', 31, 'Oran');
```

```
INSERT INTO Employe VALUES(5, 'Rais', 'Akram', 35, 'Biskra');
INSERT INTO Employe VALUES(6, 'Chanbi', 'Maroua', 25, 'Oran');
INSERT INTO Employe VALUES(7, 'Azzi', 'Sara', 20, 'Saida');
INSERT INTO Employe VALUES(8, 'Mlata', 'Moncif', 50, 'Mascara');
```

```
INSERT INTO Locale_de_Sport VALUES(1, 'MiraSport');
INSERT INTO Locale_de_Sport VALUES(2, 'Victoire');
INSERT INTO Locale_de_Sport VALUES(3, 'SBAGYM');
```

```
INSERT INTO Sport VALUES(1, 'Swim', 1, 'bobasalle');
INSERT INTO Sport VALUES(2, 'Football', 1, 'Field');
INSERT INTO Sport VALUES(3, 'Running', 1, 'no1');
INSERT INTO Sport VALUES(4, 'Swim', 3, 'Ki30');
INSERT INTO Sport VALUES(5, 'Football', 2, 'SYM30');
INSERT INTO Sport VALUES(6, 'VollyBall', 2, '1');
INSERT INTO Sport VALUES(7, 'Soccer', 1, '1');
INSERT INTO Sport VALUES(8, 'Tennis', 3, '2');
INSERT INTO Sport VALUES(9, 'Rugby', 2, '2');
```

```
INSERT INTO Inscription VALUES(1, 1, '13-SEP-07', 20);
INSERT INTO Inscription VALUES(1, 2, '13-SEP-10', 50);
INSERT INTO Inscription VALUES(2, 6, '20-DEC-21', 10);
INSERT INTO Inscription VALUES(2, 7, '10-DEC-19', 14);
INSERT INTO Inscription VALUES(3, 8, '15-DEC-20', 18);
INSERT INTO Inscription VALUES(3, 9, '15-JAN-21', 18);
INSERT INTO Inscription VALUES(5, 1, '15-JAN-21', 18);
INSERT INTO Inscription VALUES(5, 9, '15-JAN-21', 18);
INSERT INTO Inscription VALUES(4, 9, '15-JAN-21', 06);
INSERT INTO Inscription VALUES(4, 5, '15-JAN-21', 05);
INSERT INTO Inscription VALUES(4, 3, '15-JAN-21', 14);
INSERT INTO Inscription VALUES(1, 3, '15-JAN-21', 14);
INSERT INTO Inscription VALUES(1, 4, '15-JAN-21', 14);
INSERT INTO Inscription VALUES(1, 5, '15-JAN-21', 14);
INSERT INTO Inscription VALUES(1, 6, '15-JAN-21', 14);
INSERT INTO Inscription VALUES(1, 7, '15-JAN-21', 14);
INSERT INTO Inscription VALUES(1, 8, '15-JAN-21', 14);
```

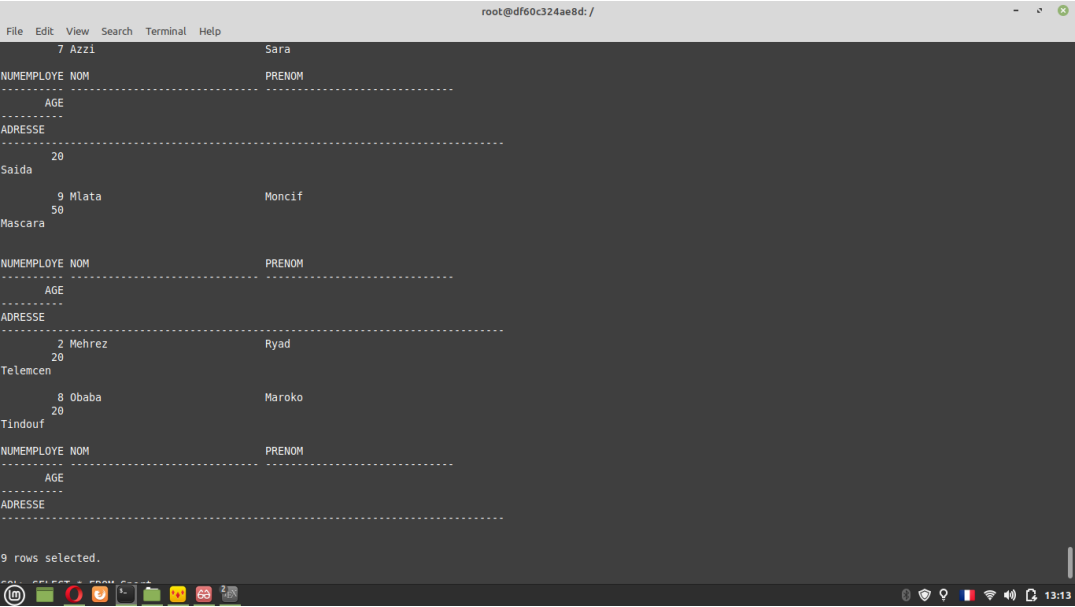
```
SQL> clear
SQL> SELECT * FROM Employee;
```

NUMEMPLOYE	NOM	PRENOM	AGE	ADRESSE
1	Hadjazi	Hisham	20	SBA
3	Sahraoui	Amine	31	SBA
4	Merloa	Adel	31	Oran
5	Rais	Akram	35	Oran
6	Chanbi	Maroua	25	Bliskra
7	Azzi	Sara	20	Oran
9	Mlata	Moncif	50	Saida
2	Mehrez	Ryad	20	Mascara
8	Obaba	Maroko	20	Telemcen

FIGURE 1.1: Employee Table

```
ADRESSE
-----
Bliskra
-----
6 Chanbi Maroua
25
Oran
-----
7 Azzi Sara
NUMEMPLOYE NOM PRENOM
-----
AGE
-----
ADRESSE
-----
20
Saida
-----
9 Mlata Moncif
50
Mascara
-----
NUMEMPLOYE NOM PRENOM
-----
AGE
-----
ADRESSE
-----
20
Telemcen
-----
8 Obaba Maroko
20
Tindouf
-----
NUMEMPLOYE NOM PRENOM
-----
```

FIGURE 1.2: Employee Table

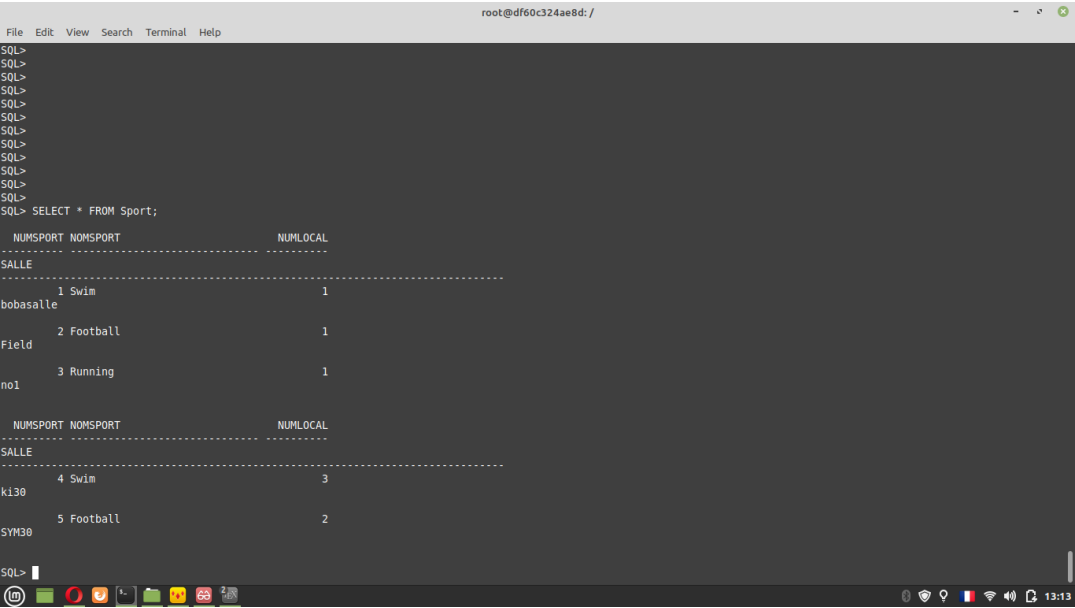


```
File Edit View Search Terminal Help
root@df60c324ae8d: /

7 Azzi Sara
NUMEMPLOIE NOM PRENOM
-----
AGE
-----
ADRESSE
-----
20
Saïda
9 Mlata Moncif
50
Mascara
NUMEMPLOIE NOM PRENOM
-----
AGE
-----
ADRESSE
-----
2 Mehrez Ryad
20
Telmen
8 Dbaba Maroko
20
Tindouf
NUMEMPLOIE NOM PRENOM
-----
AGE
-----
ADRESSE
-----

9 rows selected.
```

FIGURE 1.3: Employee Table



```
File Edit View Search Terminal Help
root@df60c324ae8d: /

SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> SELECT * FROM Sport;

NUMSPORT NOMSPORT NUMLOCAL
-----
SALLE
-----
1 Swim 1
bobasalle
2 Football 1
Field
3 Running 1
no1
NUMSPORT NOMSPORT NUMLOCAL
-----
SALLE
-----
4 Swim 3
kl30
5 Football 2
SYM30
SQL>
```

FIGURE 1.4: Sport Table

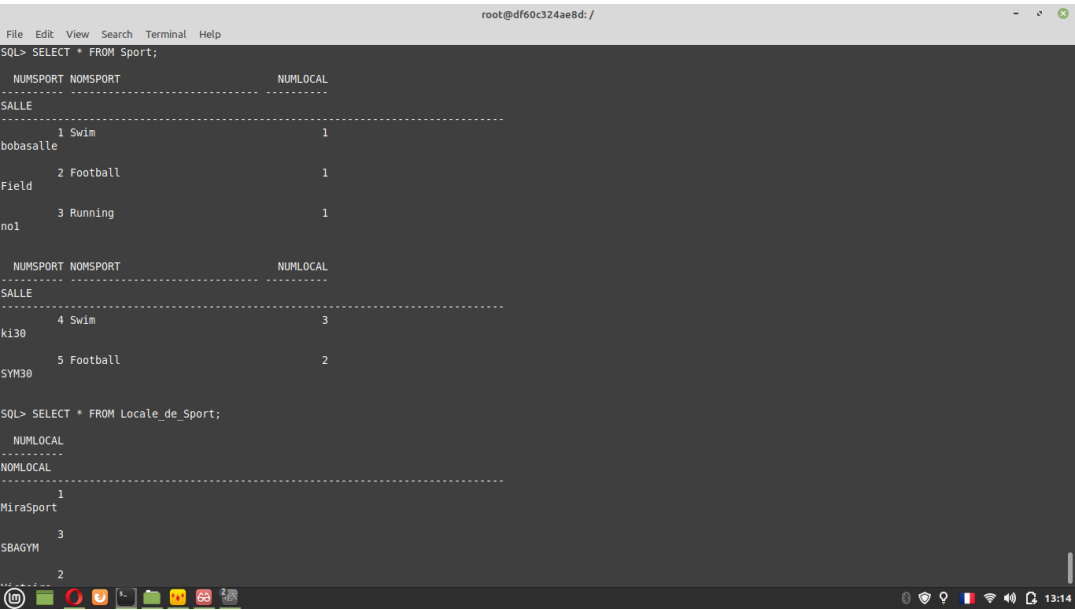
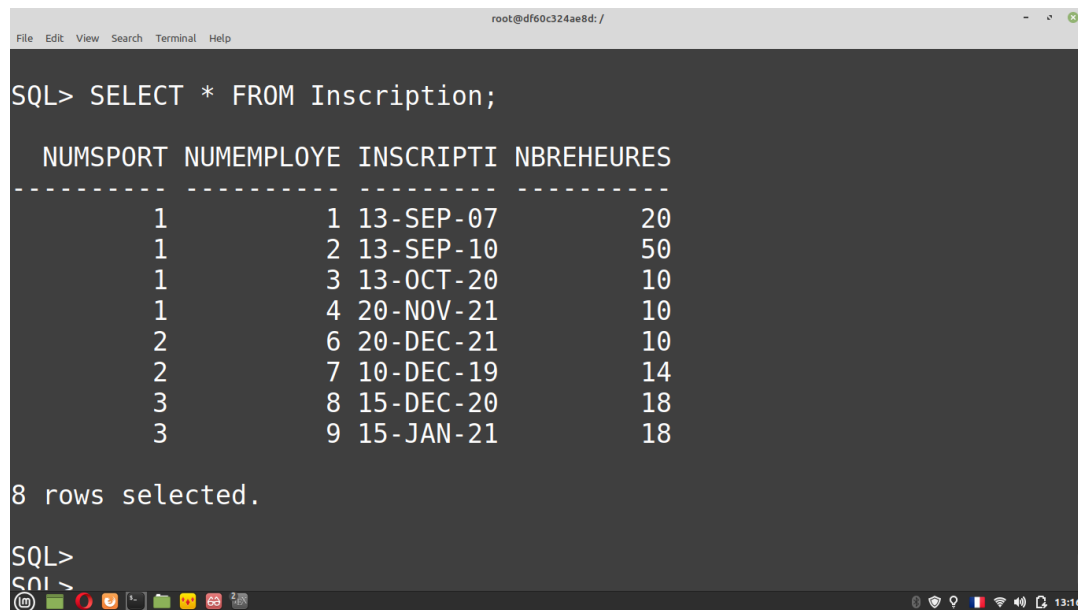


FIGURE 1.5: Local-de-Sport Table



FIGURE 1.6: Local-de-Sport Table





The screenshot shows a terminal window with a dark background. At the top, there's a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. Below the menu bar, the prompt 'SQL>' is followed by the command 'SELECT \* FROM Inscription;'. The output is a table with four columns: 'NUMSPORT', 'NUMEMPLOYE', 'INSCRIPTI', and 'NBREHEURES'. The table contains 8 rows of data. Below the table, it says '8 rows selected.' and then 'SQL>' is shown again. The terminal window has a title bar that says 'root@df60c324ae8d: /'.

```
SQL> SELECT * FROM Inscription;
```

NUMSPORT	NUMEMPLOYE	INSCRIPTI	NBREHEURES
1	1	13-SEP-07	20
1	2	13-SEP-10	50
1	3	13-OCT-20	10
1	4	20-NOV-21	10
2	6	20-DEC-21	10
2	7	10-DEC-19	14
3	8	15-DEC-20	18
3	9	15-JAN-21	18

```
8 rows selected.
```

```
SQL>
```

FIGURE 1.7: Local-de-Sport Table

1. Donnez les numéros, noms et prénoms des employés inscrits dans les sports gérés par le local de nom «Victoire».

```
SELECT e.numEmploye, e.nom, e.prenom
FROM Employe e
INNER JOIN Inscription i
    ON e.numEmploye = i.numEmploye
INNER JOIN Sport s
    ON i.numSport = s.numSport
INNER JOIN Locale_de_Sport l
    ON s.numLocal = l.numLocal
WHERE l.nomLocal = 'Victoire';
```

```

root@df60c324ae8d: /
File Edit View Search Terminal Help
  3      8 15-DEC-20      18
  3      9 15-JAN-21      18

8 rows selected.

SQL> INSERT INTO Inscription VALUES(5, 9, '15-JAN-21', 18);
1 row created.

SQL> INSERT INTO Inscription VALUES(5, 1, '15-JAN-21', 18);
1 row created.

SQL> SELECT e.numEmploye, e.nom, e.prenom
FROM Employe e
INNER JOIN Inscription i
ON e.numEmploye = i.numEmploye
INNER JOIN Sport s
ON i.numSport = s.numSport
INNER JOIN Locale_de_Sport l
ON s.numLocal = l.numLocal
WHERE l.nomLocal = 'Victoire';
  2      3      4      5      6      7      8      9

NUMEMPLOYE  NOM                                PRENOM
-----
          1  Hadjazi                                Hisham
          9  Mlata                                  Moncif

SQL>

```

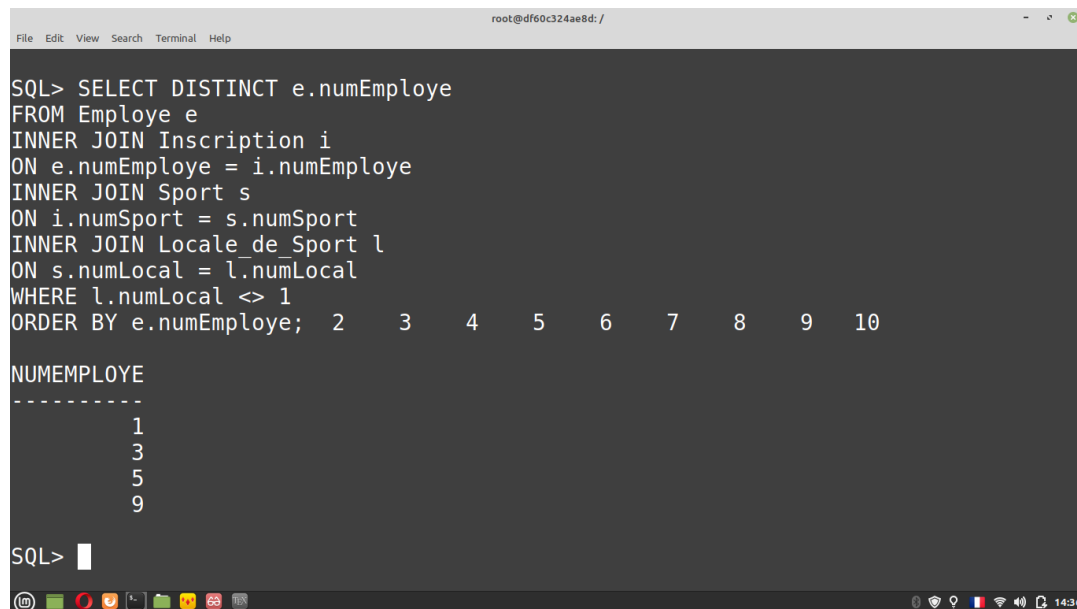
FIGURE 1.8: Query 1

2. Donnez les numéros des employés qui ne sont inscrits à aucun sport du Local n° 1.

```

SELECT DISTINCT e.numEmploye
  FROM Employe e
 INNER JOIN Inscription i
    ON e.numEmploye = i.numEmploye
 INNER JOIN Sport s
    ON i.numSport = s.numSport
 INNER JOIN Locale_de_Sport l
    ON s.numLocal = l.numLocal
 WHERE l.numLocal <> 1
    ORDER BY e.numEmploye;

```



```
root@df60c324ae8d: /
File Edit View Search Terminal Help

SQL> SELECT DISTINCT e.numEmploye
FROM Employee e
INNER JOIN Inscription i
ON e.numEmploye = i.numEmploye
INNER JOIN Sport s
ON i.numSport = s.numSport
INNER JOIN Locale_de_Sport l
ON s.numLocal = l.numLocal
WHERE l.numLocal <> 1
ORDER BY e.numEmploye; 2    3    4    5    6    7    8    9    10

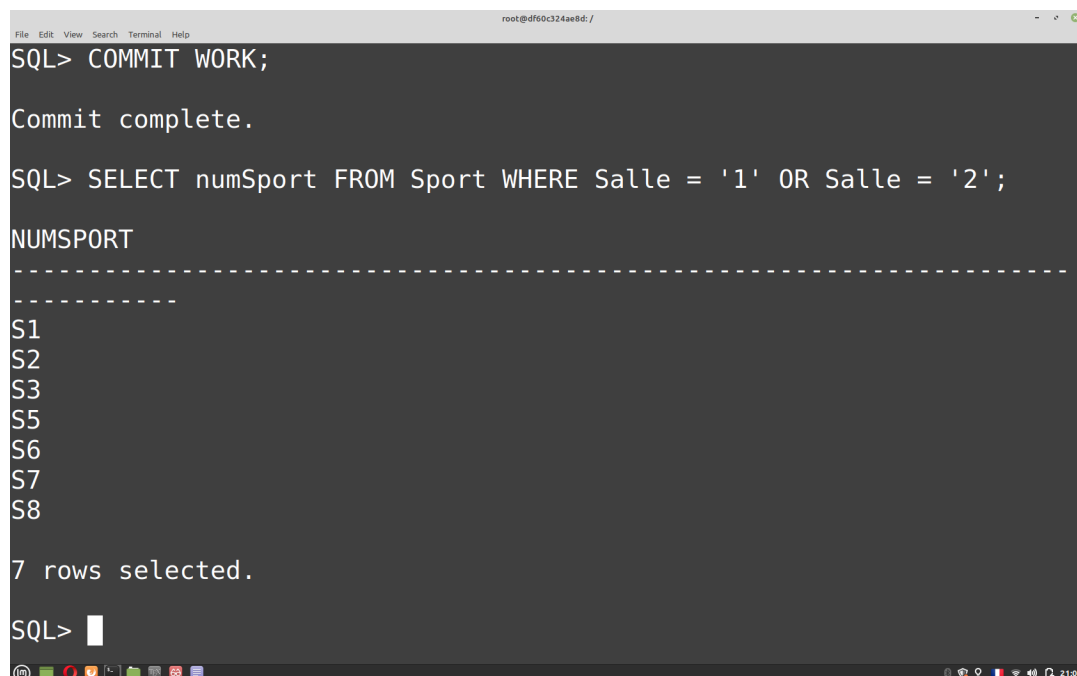
NUMEMPLOYE
-----
          1
          3
          5
          9

SQL> 
```

FIGURE 1.9: Query 2

### 3. Numéros des sports assurés dans les salles 1 ou 2.

```
SELECT numSport FROM Sport WHERE Salle = '1' OR Salle = '2';
```



```
root@df60c324ae8d: /
File Edit View Search Terminal Help

SQL> COMMIT WORK;

Commit complete.

SQL> SELECT numSport FROM Sport WHERE Salle = '1' OR Salle = '2';

NUMSPORT
-----
S1
S2
S3
S5
S6
S7
S8

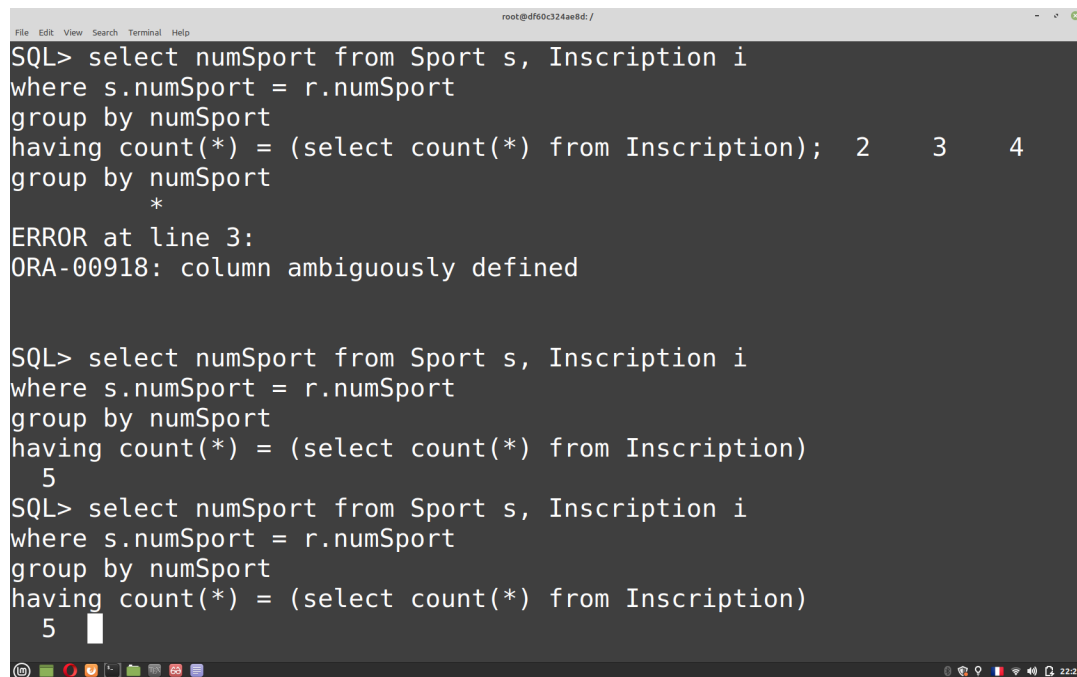
7 rows selected.

SQL> 
```

FIGURE 1.10: Query 3

**4. Numéros des sports aux quels sont inscrits tous les employés.**

```
select numSport from Sport s, Inscription i
where s.numSport = r.numSport
group by numSport
having count(*) = (select count(*) from Inscription)
```



The screenshot shows a terminal window with the following content:

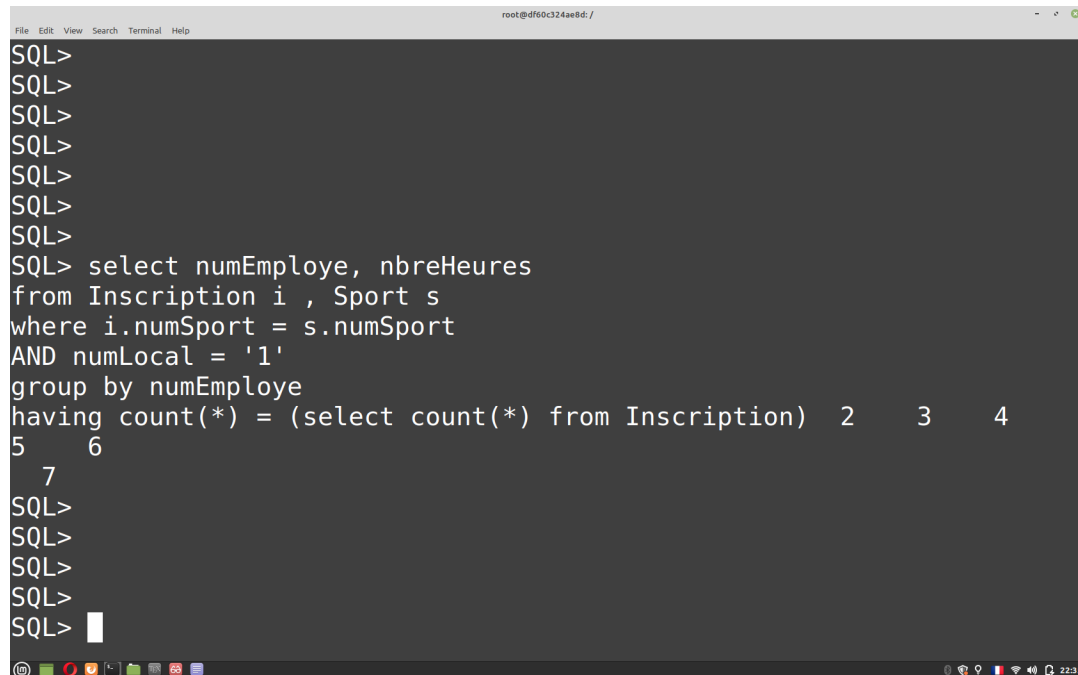
```
SQL> select numSport from Sport s, Inscription i
where s.numSport = r.numSport
group by numSport
having count(*) = (select count(*) from Inscription); 2    3    4
group by numSport
              *
ERROR at line 3:
ORA-00918: column ambiguously defined

SQL> select numSport from Sport s, Inscription i
where s.numSport = r.numSport
group by numSport
having count(*) = (select count(*) from Inscription)
5
SQL> select numSport from Sport s, Inscription i
where s.numSport = r.numSport
group by numSport
having count(*) = (select count(*) from Inscription)
5
```

FIGURE 1.11: Query 3

**5. Numéros des employés inscrits à tout sport du local n° 1.**

```
select numEmploye, nbreHeures
from Inscription i , Sport s
where i.numSport = s.numSport
AND numLocal = '1'
group by numEmploye
having count(*) = (select count(*) from Inscription)
```



```
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> select numEmploye, nbreHeures
from Inscription i , Sport s
where i.numSport = s.numSport
AND numLocal = '1'
group by numEmploye
having count(*) = (select count(*) from Inscription) 2    3    4
5    6
7
SQL>
SQL>
SQL>
SQL>
SQL>
```

FIGURE 1.12: Query 3

## 1.2 Partie 2

### 1.2.1 1. A partir des énoncés ci-dessus, créez les tables base de données et procédez à leurs alimentations par les données suivantes :

Employe (numEmploye, nom, prenom, age, adresse)

```
(100, 'ABID', 'Mohamed', '35', 'SBA');
(101, 'BEY', 'Amina', '41', 'SBA');
(102, 'MIR', 'Ilyas', '29', 'Oran');
(103, 'DIAFI', 'Adel', '25', 'Oran');
(104, 'AMAR', 'Ali', '38', 'Oran');
(105, 'BENSAID', 'Salima', '39', 'Alger');
(106, 'NOUR', 'Khadija', '49', 'Alger');
(107, 'NACER', 'Fatima', '29', 'Oran');
```

```
INSERT INTO Employe VALUES(100, 'ABID', 'Mohamed', 35, 'SBA');
INSERT INTO Employe VALUES(101, 'BEY', 'Amina', 41, 'SBA');
INSERT INTO Employe VALUES(102, 'MIR', 'Ilyas', 29, 'Oran');
INSERT INTO Employe VALUES(103, 'DIAFI', 'Adel', 25, 'Oran');
INSERT INTO Employe VALUES(104, 'AMAR', 'Ali', 38, 'Oran');
INSERT INTO Employe VALUES(105, 'BENSAID', 'Salima', 39, 'Alger');
INSERT INTO Employe VALUES(106, 'NOUR', 'Khadija', 49, 'Alger');
INSERT INTO Employe VALUES(107, 'NACER', 'Fatima', 29, 'Oran');
```

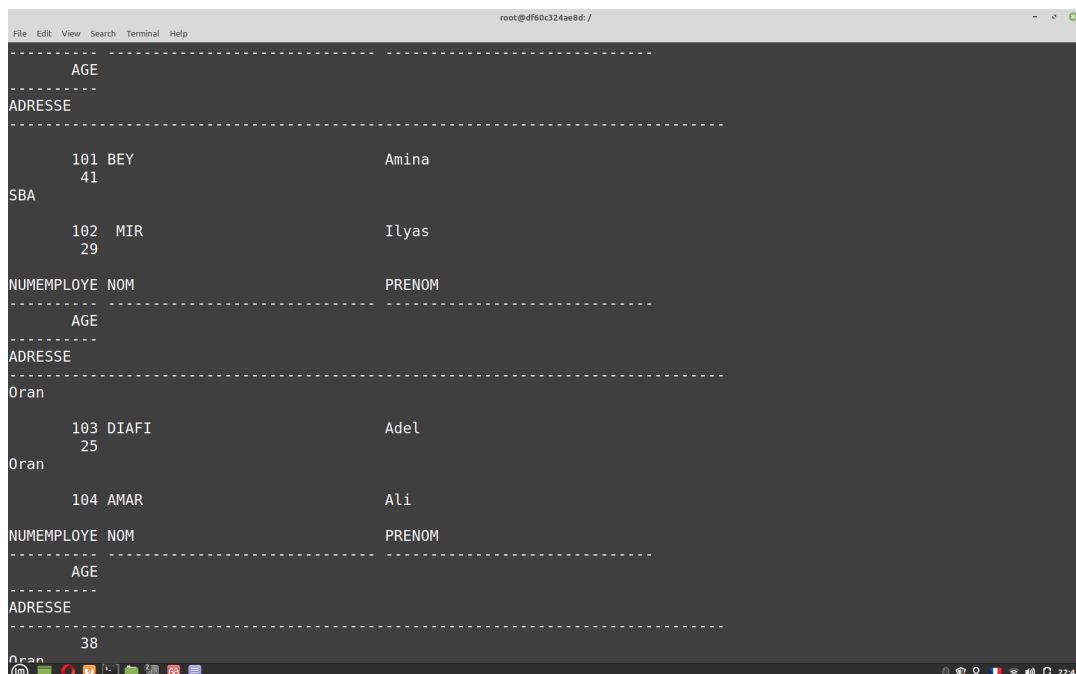


FIGURE 1.13: Query 1.1

```

root@df60c324ae8d:/
File Edit View Search Terminal Help
-----
38
Oran
105 BENSAID Salima
39
Alger
NUMEMPLOYE NOM PRENOM
-----
AGE
-----
ADRESSE
-----
106 NOUR Khadija
49
Alger
107 NACER Fatima
29
Oran
NUMEMPLOYE NOM PRENOM
-----
AGE
-----
ADRESSE
-----
16 rows selected.
SQL>

```

FIGURE 1.14: Query 1.2

**Sport (numSport, nomSport, numLocal, salle)**

```

('S1', 'Aérobic', 'L1', '3');
('S2', 'Athlétisme', 'L1', '1');
('S3', 'Basket-ball', 'L2', '1');
('S4', 'Hand-ball', 'L3', '2');

```

```

UPDATE Sport SET
    nomSport = 'Aerobic',
    numLocal = 'L1',
    salle = '3',

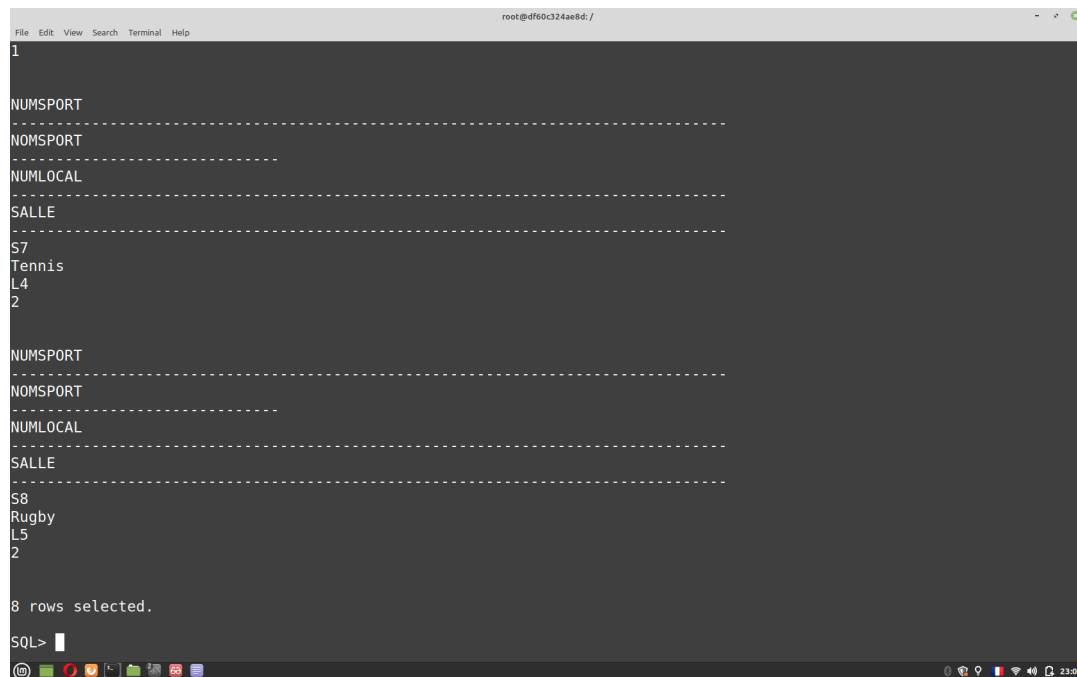
WHERE
    numSport = 'S1';

UPDATE Sport SET
    nomSport = 'Athletisme',
    numLocal = 'L1',
    salle = '1',

WHERE
    numSport = 'S2';

UPDATE Sport SET
    nomSport = 'Basket-ball',
    numLocal = 'L1',
    salle = '1',

```

**WHERE**`numSport = 'S3';`**UPDATE Sport SET**`nomSport = 'Hand-ball',``numLocal = 'L3',``salle = '2',`**WHERE**`numSport = 'S4';`

The screenshot shows a terminal window with a dark background. At the top, a menu bar contains 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal title is 'root@df60c324ae8d: /'. The output of a query is displayed as follows:

```
1
NUMSPORT
-----
NOMSPORT
-----
NUMLOCAL
-----
SALLE
-----
S7
Tennis
L4
2

NUMSPORT
-----
NOMSPORT
-----
NUMLOCAL
-----
SALLE
-----
S8
Rugby
L5
2

8 rows selected.
SQL> 
```

The terminal window has a standard Linux desktop environment at the bottom with various icons and a system tray showing the time as 23:05.

FIGURE 1.15: Query 2.1



**Inscription (numSport, numEmploye, date, nbreHeures)**

```

('S1','101','15-01-2017','3');
('S1','100','25-02-2019','6');
('S2','100','20-03-2017','3');
('S1','104','10-05-2019','4');
('S3','102','15-01-2018','4');
('S1','103','25-10-2017','6');
('S4','100','10-02-2019','6');
('S1','107','20-12-2018','8');
('S2','104','10-07-2019','4');
('S3','100','05-06-2019','6');
('S4','105','30-04-2017','6');
('S1','102','12-02-2019','6');
('S1','106','05-10-2018','6');
('S1','105','21-09-2019','4');
('S1','105','06-09-2021','4');

```

```

INSERT INTO Inscription VALUES( 'S1 ' , '101 ' , '15-01-2017 ' ,3);
INSERT INTO Inscription VALUES( 'S1 ' , '100 ' , '25-02-2019 ' ,6);
INSERT INTO Inscription VALUES( 'S2 ' , '100 ' , '20-03-2017 ' ,3);
INSERT INTO Inscription VALUES( 'S2 ' , '104 ' , '10-05-2019 ' ,4);
INSERT INTO Inscription VALUES( 'S3 ' , '102 ' , '15-01-2018 ' ,4);
INSERT INTO Inscription VALUES( 'S1 ' , '103 ' , '25-10-2017 ' ,6);
INSERT INTO Inscription VALUES( 'S4 ' , '100 ' , '10-02-2019 ' ,6);
INSERT INTO Inscription VALUES( 'S1 ' , '107 ' , '20-12-2018 ' ,8);
INSERT INTO Inscription VALUES( 'S2 ' , '104 ' , '10-07-2019 ' ,4);
INSERT INTO Inscription VALUES( 'S3 ' , '100 ' , '05-06-2019 ' ,6);
INSERT INTO Inscription VALUES( 'S4 ' , '105 ' , '30-04-2017 ' ,6);
INSERT INTO Inscription VALUES( 'S1 ' , '102 ' , '12-02-2019 ' ,6);
INSERT INTO Inscription VALUES( 'S1 ' , '106 ' , '05-10-2018 ' ,6);
INSERT INTO Inscription VALUES( 'S1 ' , '105 ' , '21-09-2019 ' ,4);
INSERT INTO Inscription VALUES( 'S1 ' , '105 ' , '06-09-2021 ' ,4);

```

```
SQL>
SQL> SELECT * FROM Inscription;

NUMSPORT
-----
NUMEMPLOYE INSCRIPT NBREHEURES
-----
S1
      1 15-01-17      20
S1
      1 15-01-19      20
S2
      2 05-01-20      20

NUMSPORT
-----
NUMEMPLOYE INSCRIPT NBREHEURES
-----
S3
      6 15-01-20      20
S4
      7 25-01-21      20
S2
      2 05-01-20      20

NUMSPORT
-----
```

FIGURE 1.16: Query 3.1

```
NUMSPORT
-----
NUMEMPLOYE INSCRIPT NBREHEURES
-----
S2
      2 05-01-20      20
S1
      1 15-01-19      20
S3
      6 15-01-20      20

NUMSPORT
-----
NUMEMPLOYE INSCRIPT NBREHEURES
-----
S3
      6 15-01-20      20
S1
      1 15-01-19      20
S4
      7 25-01-21      20

NUMSPORT
-----
NUMEMPLOYE INSCRIPT NBREHEURES
-----
S4
      7 25-01-21      20
```

FIGURE 1.17: Query 3.2

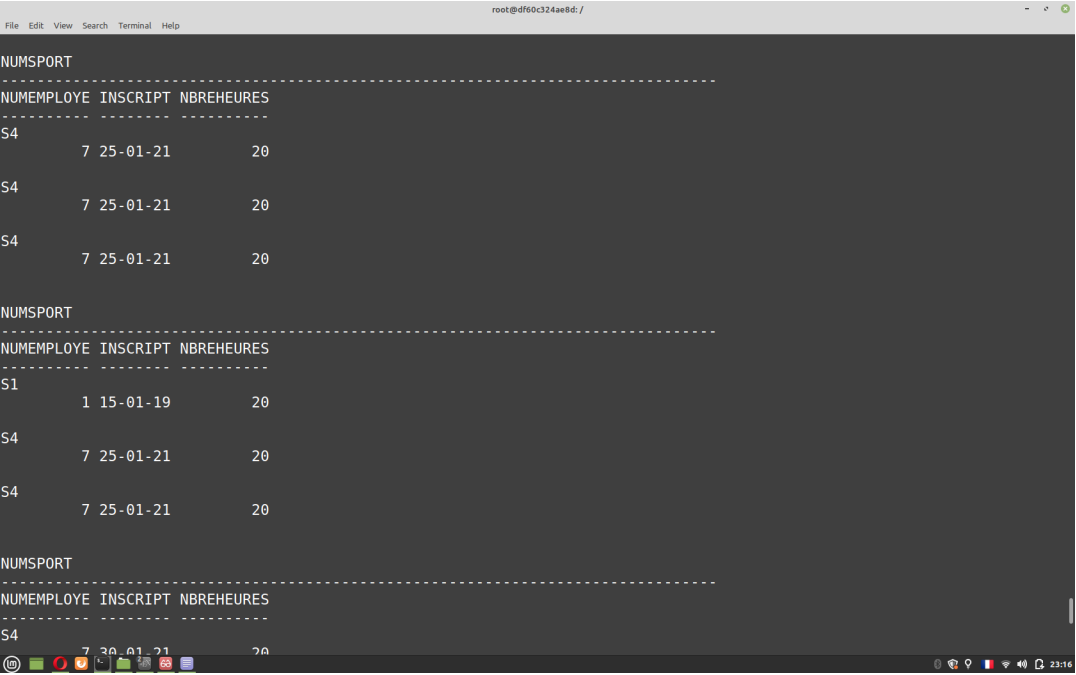


FIGURE 1.18: Query 3.3

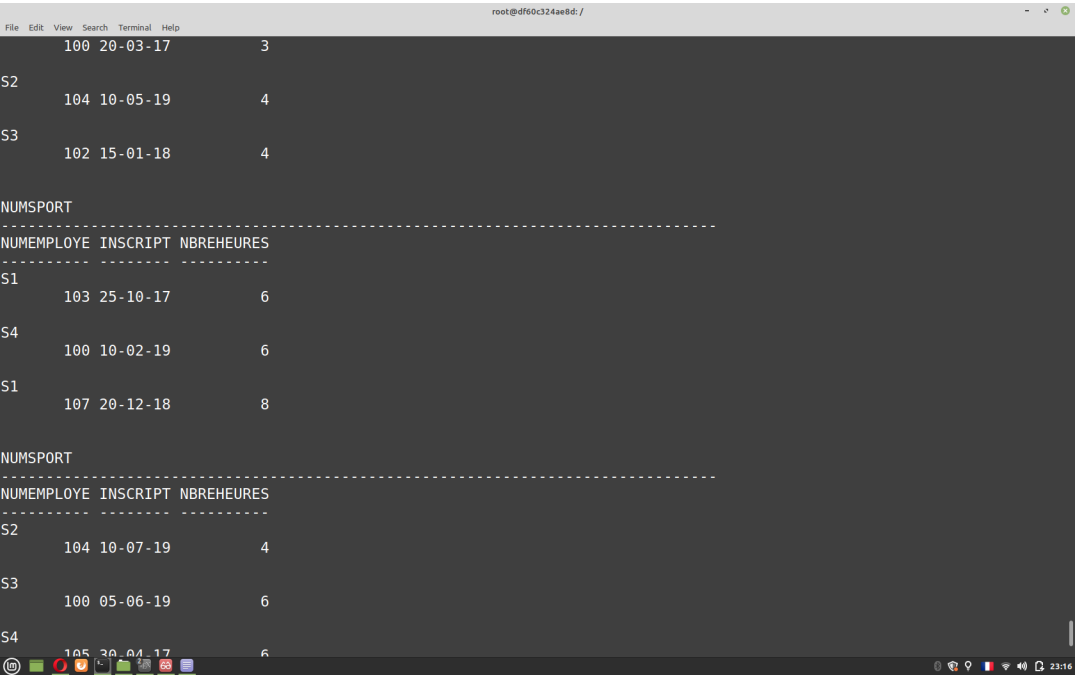
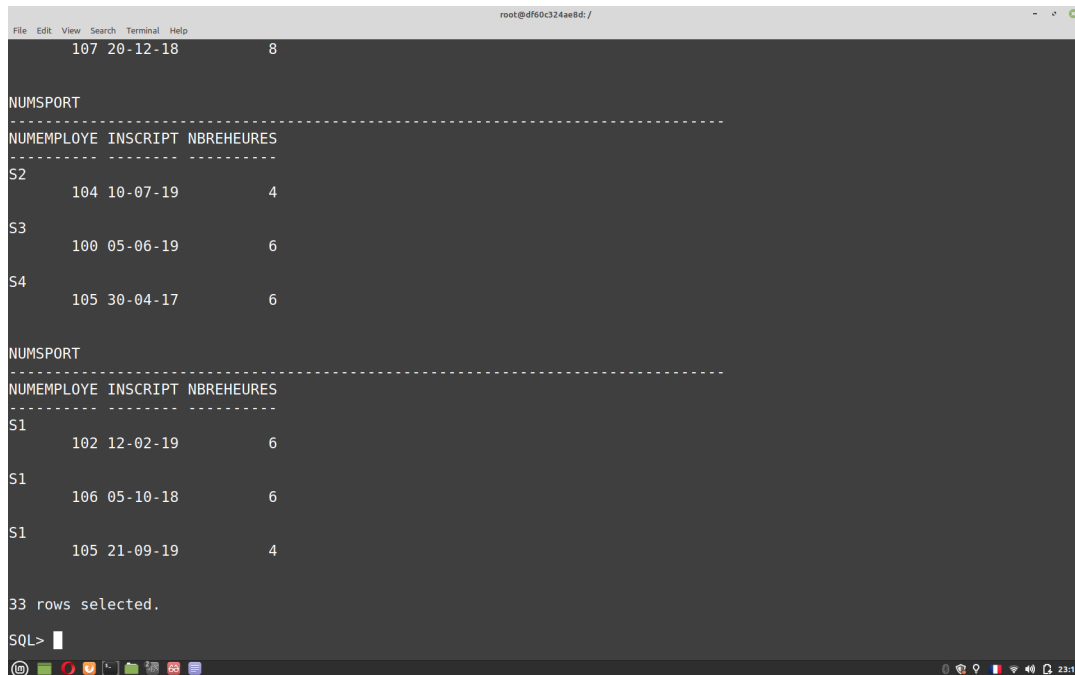


FIGURE 1.19: Query 3.4



```
107 20-12-18      8

NUMSPORT
-----
NUMEMPLOYE INSCRIPT NBREHEURES
-----
S2
      104 10-07-19      4
S3
      100 05-06-19      6
S4
      105 30-04-17      6

NUMSPORT
-----
NUMEMPLOYE INSCRIPT NBREHEURES
-----
S1
      102 12-02-19      6
S1
      106 05-10-18      6
S1
      105 21-09-19      4

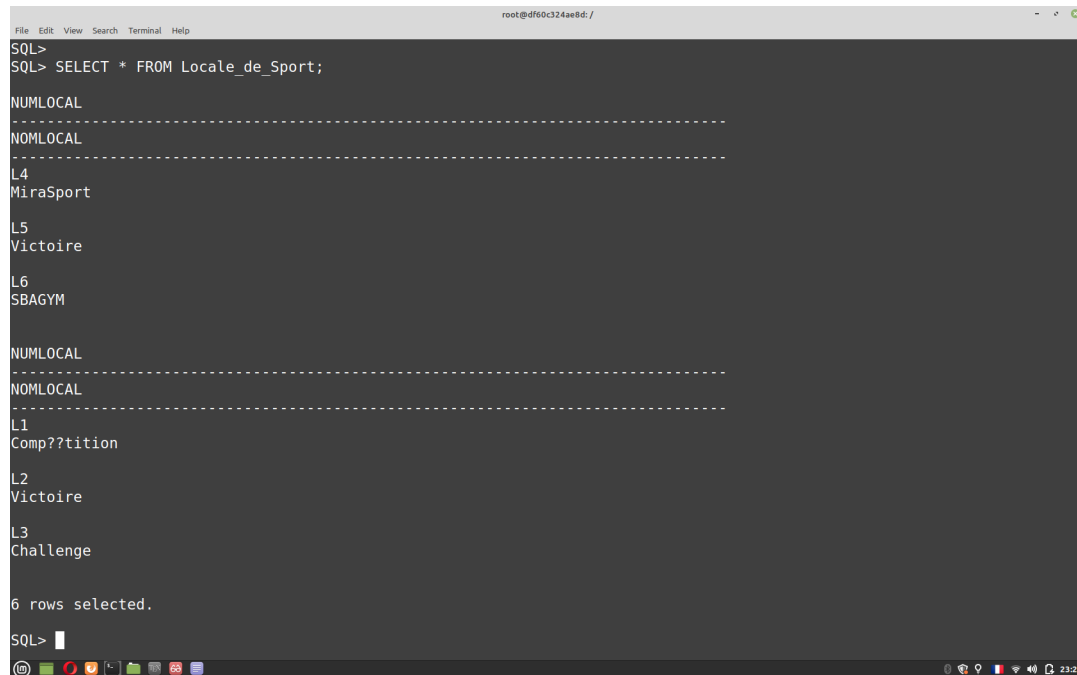
33 rows selected.

SQL>
```

FIGURE 1.20: Query 3.5

**Local-de-Sport (numLocal, nomLocal)****('L1', 'Competition');****('L2', 'Victoire');****('L3', 'Challenge');**

```
INSERT INTO Locale_de_Sport VALUES( 'L1', 'Competition' );
INSERT INTO Locale_de_Sport VALUES( 'L2', 'Victoire' );
INSERT INTO Locale_de_Sport VALUES( 'L3', 'Challenge' );
```



```
SQL>
SQL> SELECT * FROM Locale_de_Sport;

NUMLOCAL
-----
NOMLOCAL
-----
L4
MiraSport

L5
Victoire

L6
SBAGYM

NUMLOCAL
-----
NOMLOCAL
-----
L1
Comp??tition

L2
Victoire

L3
Challenge

6 rows selected.

SQL>
```

FIGURE 1.21: Query 4.1

## 1.2.2 2. Une fois les tables créées et alimentées, implémentez les requêtes SQL exprimées dans la partie 1.

1. Donnez les numéros, noms et prénoms des employés inscrits dans les sports gérés par le local de nom «Victoire».

```
SELECT DISTINCT e.numEmploye, e.nom, e.prenom
FROM Employe e
INNER JOIN Inscription i
ON e.numEmploye = i.numEmploye
INNER JOIN Sport s
ON i.numSport = s.numSport
INNER JOIN Locale_de_Sport l
ON s.numLocal = l.numLocal
WHERE l.nomLocal = 'Victoire';
```

```
root@df60c324ae8d:/
```

File Edit View Search Terminal Help

6 rows selected.

```
SQL> SELECT DISTINCT e.numEmploye, e.nom, e.prenom  
FROM Employe e  
INNER JOIN Inscription i  
ON e.numEmploye = i.numEmploye  
INNER JOIN Sport s  
ON i.numSport = s.numSport  
INNER JOIN Locale_de_Sport l  
ON s.numLocal = l.numLocal  
WHERE l.nomLocal = 'Victoire';
```

NUMEMPLOYE	NOM	PRENOM
104	AMAR	Ali
100	ABID	Mohamed
2	Amuer	Wasim

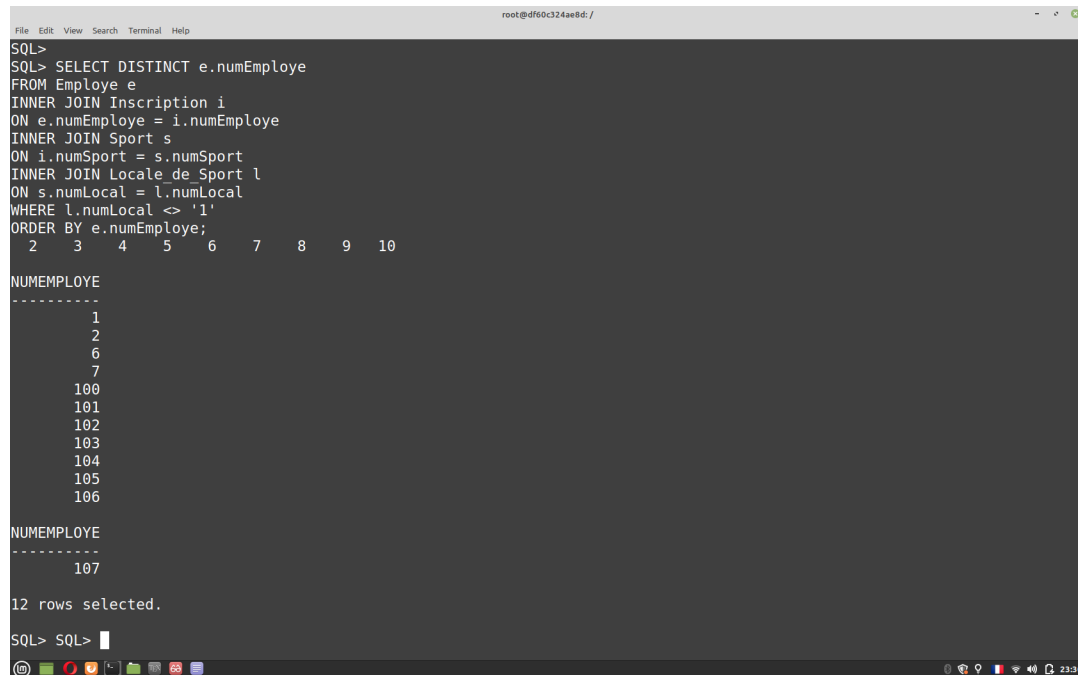
```
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>
```

Linux desktop environment with terminal window showing SQL query results.

FIGURE 1.22: Query 6.1

**2. Donnez les numéros des employés qui ne sont inscrits à aucun sport du Local n° 1.**

```
SELECT DISTINCT e.numEmploye
FROM Employe e
INNER JOIN Inscription i
ON e.numEmploye = i.numEmploye
INNER JOIN Sport s
ON i.numSport = s.numSport
INNER JOIN Locale_de_Sport l
ON s.numLocal = l.numLocal
WHERE l.numLocal <> '1'
ORDER BY e.numEmploye;
```



```

root@df60c324ae8d:/
File Edit View Search Terminal Help
SQL>
SQL> SELECT DISTINCT e.numEmployee
FROM Employe e
INNER JOIN Inscription i
ON e.numEmployee = i.numEmployee
INNER JOIN Sport s
ON i.numSport = s.numSport
INNER JOIN Locale_de_Sport l
ON s.numLocal = l.numLocal
WHERE l.numLocal <> '1'
ORDER BY e.numEmployee;
  2   3   4   5   6   7   8   9  10

NUMEMPLOYEE
-----
      1
      2
      6
      7
     100
     101
     102
     103
     104
     105
     106

NUMEMPLOYEE
-----
     107

12 rows selected.

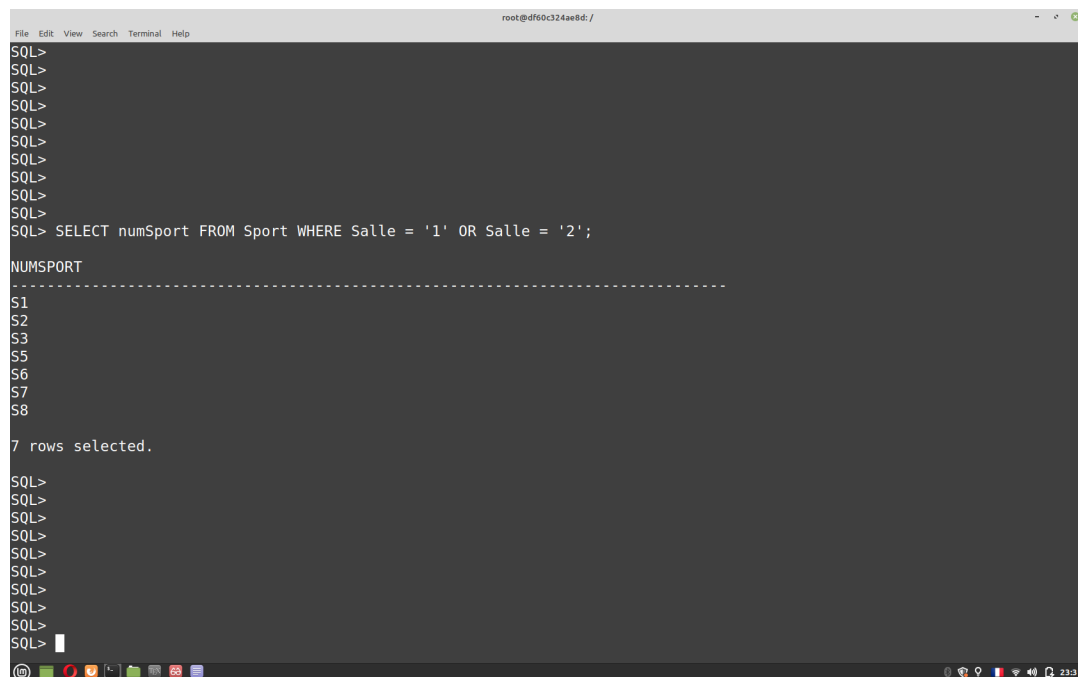
SQL> SQL>

```

FIGURE 1.23: Query 6.2

### 3. Numéros des sports assurés dans les salles 1 ou 2.

```
SELECT numSport FROM Sport WHERE Salle = '1' OR Salle = '2';
```



```

root@df60c324ae8d:/
File Edit View Search Terminal Help
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> SELECT numSport FROM Sport WHERE Salle = '1' OR Salle = '2';

NUMSPORT
-----
S1
S2
S3
S5
S6
S7
S8

7 rows selected.

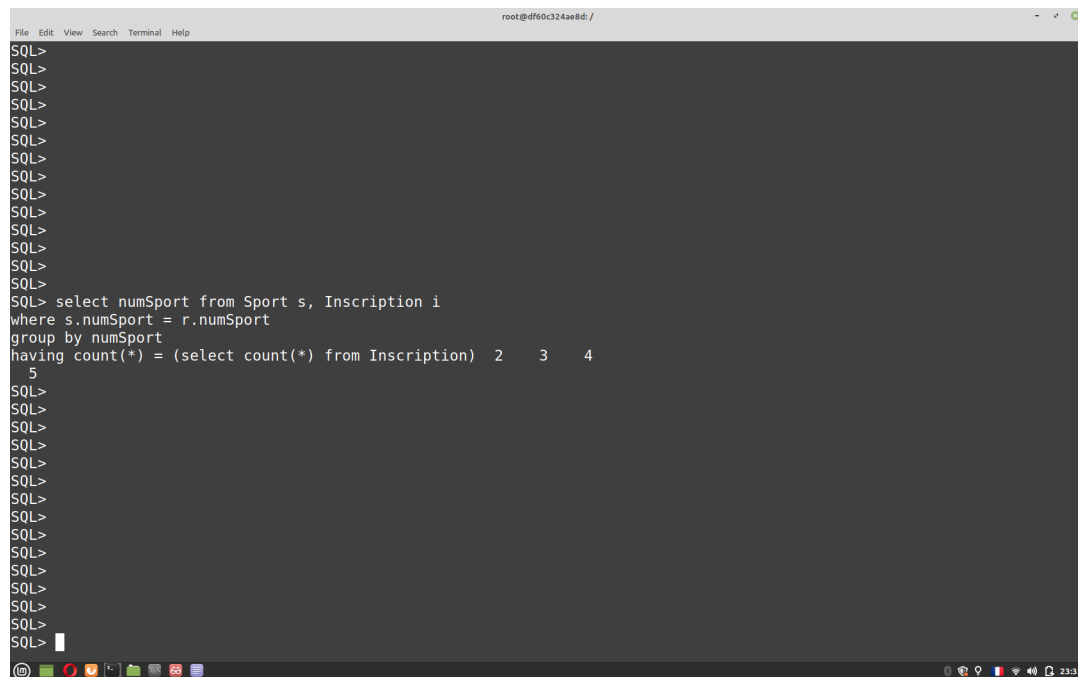
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>

```

FIGURE 1.24: Query 3

**4. Numéros des sports aux quels sont inscrits tous les employés.**

```
select numSport from Sport s, Inscription i
where s.numSport = r.numSport
group by numSport
having count(*) = (select count(*) from Inscription)
```



The screenshot shows a terminal window with a dark background. The title bar at the top reads 'root@df60c324ae8d: /'. The terminal content shows a series of 'SQL>' prompts. The first 14 prompts are empty. The 15th prompt is followed by the SQL query: `select numSport from Sport s, Inscription i where s.numSport = r.numSport group by numSport having count(*) = (select count(*) from Inscription)`. The output of this query is displayed on the next line: `2 3 4`. Following this, there are 14 more 'SQL>' prompts, each followed by a single digit: `5`, `5`, `5`, `5`, `5`, `5`, `5`, `5`, `5`, `5`, `5`, `5`, `5`, and `5`. The terminal window has a standard Linux desktop environment at the bottom with various icons and a system tray showing the time as 23:32.

FIGURE 1.25: Query 6.4





### 1.2.3 3. Utilisez la commande explain plan pour calculer les coûts d'exécutions des requêtes. Comparez et commentez les résultats.

```

root@df60c324ae8d: /
File Edit View Search Terminal Help

SET LINESIZE 130
SET PAGESIZE 0
SELECT * FROM table(DBMS_XPLAN.DISPLAY); 2 3 4 SQL> SQL> SQL>
Plan hash value: 2174506114

-----
| Id | Operation          | Name          | Rows | Bytes | Cost (%CPU)| Time |
-----
| 0  | SELECT STATEMENT   |               |      |      |              |      |
| 1  |   SORT UNIQUE      |               | 33   | 7293 | 6 (34)     | 00:00:01 |
| * 2 |    HASH JOIN       |               | 33   | 7293 | 5 (20)     | 00:00:01 |
| 3  |      NESTED LOOPS  |               | 8     | 1248 | 2 (0)      | 00:00:01 |
| * 4 |        TABLE ACCESS FULL | SPORT        | 8     | 832  | 2 (0)      | 00:00:01 |
| * 5 |          INDEX UNIQUE SCAN | SYS_C007005  | 1     | 52   | 0 (0)      | 00:00:01 |
| * 6 |            TABLE ACCESS FULL | INSCRIPTION  | 33    | 2145 | 2 (0)      | 00:00:01 |
-----

Predicate Information (identified by operation id):
-----
 2 - access("I"."NUMSPORT"="S"."NUMSPORT")
 4 - filter("S"."NUMLOCAL"<>'1')
 5 - access("S"."NUMLOCAL"="L"."NUMLOCAL")
    filter("L"."NUMLOCAL"<>'1')
 6 - filter("I"."NUMEMPLOYE" IS NOT NULL)

Note
-----
   - dynamic sampling used for this statement (level=2)

26 rows selected.

SQL>

```

FIGURE 1.27: Query 7.1

```

root@df60c324ae8d: /
File Edit View Search Terminal Help

SET LINESIZE 130
SET PAGESIZE 0
SELECT * FROM table(DBMS_XPLAN.DISPLAY); 2 3 4 5 6 7 8 9 10 11 12 SQL> SQL> SQL>
Plan hash value: 2174506114

-----
| Id | Operation          | Name          | Rows | Bytes | Cost (%CPU)| Time |
-----
| 0  | SELECT STATEMENT   |               |      |      |              |      |
| 1  |   SORT UNIQUE      |               | 33   | 7293 | 6 (34)     | 00:00:01 |
| * 2 |    HASH JOIN       |               | 33   | 7293 | 5 (20)     | 00:00:01 |
| 3  |      NESTED LOOPS  |               | 8     | 1248 | 2 (0)      | 00:00:01 |
| * 4 |        TABLE ACCESS FULL | SPORT        | 8     | 832  | 2 (0)      | 00:00:01 |
| * 5 |          INDEX UNIQUE SCAN | SYS_C007005  | 1     | 52   | 0 (0)      | 00:00:01 |
| * 6 |            TABLE ACCESS FULL | INSCRIPTION  | 33    | 2145 | 2 (0)      | 00:00:01 |
-----

Predicate Information (identified by operation id):
-----
 2 - access("I"."NUMSPORT"="S"."NUMSPORT")
 4 - filter("S"."NUMLOCAL"<>'1')
 5 - access("S"."NUMLOCAL"="L"."NUMLOCAL")
    filter("L"."NUMLOCAL"<>'1')
 6 - filter("I"."NUMEMPLOYE" IS NOT NULL)

Note
-----
   - dynamic sampling used for this statement (level=2)

26 rows selected.

SQL>

```

FIGURE 1.28: Query 7.2

```
root@df60c324ae8d: /
File Edit View Search Terminal Help

SET LINESIZE 130
SET PAGESIZE 0
SELECT * FROM table(DBMS_XPLAN.DISPLAY); 3 4 5 6 7 8 9 10 11 12 13 SQL> SQL> SQL>
Plan hash value: 2174506114

-----
| Id | Operation          | Name          | Rows | Bytes | Cost (%CPU)| Time |
-----+-----+-----+-----+-----+-----+-----+-----
|  0 | SELECT STATEMENT   |               |    33 | 7293 |       7 (43)| 00:00:01 |
|  1 |   SORT UNIQUE      |               |    33 | 7293 |       6 (34)| 00:00:01 |
| * 2 |    HASH JOIN       |               |    33 | 7293 |       5 (20)| 00:00:01 |
|  3 |      NESTED LOOPS  |               |      8 | 1248 |        2 (0)| 00:00:01 |
| * 4 |        TABLE ACCESS FULL | SPORT        |      8 | 832 |        2 (0)| 00:00:01 |
| * 5 |          INDEX UNIQUE SCAN | SYS C007005  |      1 | 52 |        0 (0)| 00:00:01 |
| * 6 |            TABLE ACCESS FULL | INSCRIPTION  |    33 | 2145 |        2 (0)| 00:00:01 |
-----

Predicate Information (identified by operation id):
-----
  2 - access("I"."NUMSPORT"="S"."NUMSPORT")
  4 - filter("S"."NUMLOCAL"<>'1')
  5 - access("S"."NUMLOCAL"="L"."NUMLOCAL")
    filter("L"."NUMLOCAL"<>'1')
  6 - filter("I"."NUMEMPLOYE" IS NOT NULL)

Note
-----
   - dynamic sampling used for this statement (level=2)

26 rows selected.

SQL>
```

FIGURE 1.29: Query 7.2