

Sélectionne 'N' lignes (SQL) pour DB2

par Ernaelsten Gérard ([MaitrePylos](#))

Date de publication : 17/04/2008

Cet article a pour but d'expliquer comment on fait pour sortir un nombre défini de lignes lors d'une requête SQL

- 1 - Contexte
 - 1.1 - Environnement de travail
- 2 - Fetch First n Rows Only
 - 2.1 - Requête de base
 - 2.2 - Sélectionner 10 lignes
- 3 - Sélectionner une plage de lignes
 - 3.1 - ROW_NUMBER OVER()
- 4 - Allons plus loin.....
 - 4.1 - La totale
 - 4.2 - Soyons fous
- 5 - Performances
 - 5.1 - Cause
 - 5.2 - Autres pistes
- Conclusion
- 6 - Remerciements
- Annexe
 - Table 'EMPLOYEE'

1 - Contexte

Dans certains SGBDR tel que Mysql, pour pouvoir sélectionner un nombre limité de 'records', on peut utiliser une fonction interne 'LIMIT()', or cette fonction est une sorte de Trigger interne aux SGBDR.

Ce Trigger ajoute une colonne à chaque table que vous créez, cette colonne va être numérotée afin de pouvoir faire un certain tri.

Cette solution est un choix que très peu de SGBDR mettent en place.

1.1 - Environnement de travail

Pour les exemples de cet article, nous utiliserons la table 'EMPLOYEE', fournie avec la base SAMPLE de DB2. [Voir code](#)

2 - Fetch First n Rows Only

2.1 - Requête de base

Typiquement en DB2, pour faire un 'SELECT', nous ferions comme ceci

```
SELECT * FROM GG.EMPLOYEE
```

Ou encore

```
SELECT EMPL.FIRSTNME, EMPL.LASTNAME  
FROM GG.EMPLOYEE AS EMPL
```

Cela retournera l'ensemble de la table.

2.2 - Sélectionner 10 lignes

Pour pouvoir récupérer les 10 premières lignes de votre requête, il faut utiliser la commande SQL **Fetch First n Rows Only**

```
SELECT EMPL.EMPNO, EMPL.FIRSTNME, EMPL.LASTNAME  
FROM GG.EMPLOYEE AS EMPL  
ORDER BY EMPL.EMPNO  
FETCH FIRST 10 ROWS ONLY
```

EMPNO	FIRSTNME	LASTNAME
000010	CHRISTINE	HAAS
000020	MICHAEL	THOMPSON
000030	SALLY	KWAN
000050	JOHN	GEYER
000060	IRVING	STERN
000070	EVA	PULASKI
000090	EILEEN	HENDERSON
000100	THEODORE	SPENSER
000110	VINCENZO	LUCCHESSI
000120	SEAN	O'CONNELL

10 record(s) selected [Fetch MetaData: 0/ms] [Fetch Data: 1/ms]

Pour pouvoir récupérer les 10 dernières lignes, il faut rajouter un **ORDER BY DESC**

```
SELECT EMPL.EMPNO, EMPL.FIRSTNME, EMPL.LASTNAME  
FROM GG.EMPLOYEE AS EMPL
```

```
ORDER BY EMPL.EMPNO DESC
FETCH FIRST 10 ROWS ONLY
```

EMPNO	FIRSTNME	LASTNAME
000340	JASON	GOUNOT
000330	WING	LEE
000320	RAMLAL	MEHTA
000310	MAUDE	SETRIGHT
000300	PHILIP	SMITH
000290	JOHN	PARKER
000280	ETHEL	SCHNEIDER
000270	MARIA	PEREZ
000260	SYBIL	JOHNSON
000250	DANIEL	SMITH

10 record(s) selected [Fetch MetaData: 0/ms] [Fetch Data: 1/ms]

3 - Sélectionner une plage de lignes

Ici, pour pouvoir sélectionner une plage de lignes, par exemple entre les 12 et 18 premières lignes, il faut faire appel à une commande SQL, qui va rajouter une colonne à votre requête.

3.1 - ROW_NUMBER OVER()

La fonction **ROW_NUMBER OVER()**, va nous permettre de faire ce travail.

```
SELECT ROW_NUMBER() OVER(ORDER BY EMPL.EMPNO) AS NUM, EMPL.EMPNO, EMPL.FIRSTNME, EMPL.LASTNAME
FROM GG.EMPLOYEE AS EMPL
```

Concrètement ici, nous avons ajouté une colonne **NUM**, dans laquelle se trouve un 'integer', démarrant de 1 jusqu'à N.

NUM	EMPNO	FIRSTNME	LASTNAME
1	000010	CHRISTINE	HAAS
2	000020	MICHAEL	THOMPSON
3	000030	SALLY	KWAN
4	000050	JOHN	GEYER
5	000060	IRVING	STERN
6	000070	EVA	PULASKI
7	000090	EILEEN	HENDERSON
8	000100	THEODORE	SPENSER

Pour pouvoir utiliser cette colonne, nous devons encapsuler cette requête, afin de pouvoir en faire le traitement.

```
WITH TABLETMP (NUM, EMPNO, PRENOM, NOM) AS
(
SELECT ROW_NUMBER() OVER(ORDER BY EMPL.EMPNO) AS NUM, EMPL.EMPNO, EMPL.FIRSTNME, EMPL.LASTNAME
FROM GG.EMPLOYEE AS EMPL
)
SELECT * FROM TABLETMP
WHERE NUM BETWEEN 12 AND 18
```

Nous venons de retourner 7 lignes allant de la ligne 12 à la ligne 18.

La clause **ORDER BY** dans **OVER()**, nous garantit l'ordre des informations.

NUM	EMPNO	PRENOM	NOM
12	000140	HEATHER	NICHOLLS
13	000150	BRUCE	ADAMSON
14	000160	ELIZABETH	PIANKA
15	000170	MASATOSHI	YOSHIMURA
16	000180	MARILYN	SCOUTTEN
17	000190	JAMES	WALKER
18	000200	DAVID	BROWN

7 record(s) selected [Fetch MetaData: 0/ms] [Fetch Data: 1/ms]

[Executed: 6/03/08 12:56:00 CET] [Execution: 10/ms]

Pour avoir les lignes en partant de la fin, on lance un **ORDER BY.....DESC**.

```
WITH TABLETMP (NUM,EMPNO,PRENOM,NOM) AS
(
SELECT ROW_NUMBER() OVER(ORDER BY EMPL.EMPNO DESC) AS NUM,EMPL.EMPNO,EMPL.FIRSTNME,EMPL.LASTNAME
FROM GG.EMPLOYEE AS EMPL
)
SELECT * FROM TABLETMP
WHERE NUM BETWEEN 12 AND 18
```

NUM	EMPNO	PRENOM	NOM
12	000230	JAMES	JEFFERSON
13	000220	JENNIFER	LUTZ
14	000210	WILLIAM	JONES
15	000200	DAVID	BROWN
16	000190	JAMES	WALKER
17	000180	MARILYN	SCOUTTEN
18	000170	MASATOSHI	YOSHIMURA

7 record(s) selected [Fetch MetaData: 0/ms] [Fetch Data: 1/ms]

4 - Allons plus loin.....

Nous pourrions avoir besoin de répéter ce genre de select dans un projet, on pourrait alors en faire une procédure stockée, en lieu et place de votre langage de programmation.

```
CREATE PROCEDURE "GG"."PROCEDURE1"  
  (IN VMIN INT, IN VMAX INTEGER)  
  DYNAMIC RESULT SETS 1  
  LANGUAGE SQL  
BEGIN  
  
  DECLARE rs CURSOR WITH RETURN FOR  
    WITH TABLETMP AS  
    (  
      SELECT ROW_NUMBER() OVER(ORDER BY EMPL.EMPNO) AS NUM, EMPL.FIRSTNME, EMPL.LASTNAME  
      FROM GG.EMPLOYEE AS EMPL  
  
    )  
    SELECT * FROM TABLETMP  
    WHERE NUM BETWEEN VMIN AND VMAX  
  
  OPEN rs;  
END @
```

Ensuite nous appelons la procédure en lui passant deux paramètres, le départ de la plage et la fin.

```
CALL GG.PROCEDURE1(10,12)
```

NUM	FIRSTNME	LASTNAME
10	SEAN	O'CONNELL
11	DOLORES	QUINTANA
12	HEATHER	NICHOLLS

3 record(s) selected [Fetch MetaData: 0/ms] [Fetch Data: 1/ms]

[Executed: 6/03/08 12:56:38 CET] [Execution: 4/ms]

4.1 - La totale

Et voici une procédure stockée, qui prend en plus comme paramètre, la table de recherche et le champ sur lequel faire le ORDER BY.

```
CREATE PROCEDURE "GG"."PROCEDURE"  
  ( IN VTABLE VARCHAR(255),  
    IN VORDERBY VARCHAR(255),  
    IN VMIN INT,  
    IN VMAX INTEGER  
  )  
  DYNAMIC RESULT SETS 1
```

```

LANGUAGE SQL
BEGIN

DECLARE STMT_STRING VARCHAR(500);
DECLARE RC CURSOR WITH RETURN FOR STMT_NAME;

SET STMT_STRING = '
    WITH TABLETMP AS
    (
        SELECT '||VTABLE||'.*,ROW_NUMBER() OVER(ORDER BY '||VORDERBY||') AS NUM
        FROM '||VTABLE||'
    )
    SELECT * FROM TABLETMP
    WHERE NUM BETWEEN ' || CHAR(VMIN) || ' AND ' || CHAR(VMAX) || ' ';

PREPARE STMT_NAME FROM STMT_STRING;
OPEN RC;

END @

```

Dans cette procédure, on passe 4 arguments dans le CALL.

- String : Le nom de la table
- String : Le nom du champ
- Integer: Le départ de la recherche
- Integer : La fin de la recherche

```
CALL GG.PROCEDURE('GG.EMPLOYEE','EMPNO',15,24)
```

NAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE
IMURA	D11	2890	15/09/1978	DESIGNER	16	M	5/01/195
TTEN	D11	1682	7/07/1973	DESIGNER	17	F	21/02/19
ER	D11	2986	26/07/1974	DESIGNER	16	M	25/06/19
IN	D11	4501	3/03/1966	DESIGNER	16	M	29/05/19
S	D11	0942	11/04/1979	DESIGNER	17	M	23/02/19
	D11	0672	29/08/1968	DESIGNER	18	F	19/03/19
ERSON	D21	2094	21/11/1966	CLERK	14	M	30/05/19
NO	D21	3780	5/12/1979	CLERK	17	M	31/03/19
H	D21	0961	30/10/1969	CLERK	15	M	12/11/19
SON	D21	8953	11/09/1975	CLERK	16	F	5/10/193

0/ms] [Fetch Data: 3/ms]

ution: 5/ms]

4.2 - Soyons fous

On pourrait encore approfondir cette procédure, et ajouter un tableau de 'CHAMP'; à sélectionner, mais je pense qu'on aurait autant de facilité dans ce cas, à écrire la requête en SQL.

5 - Performances

Dans ce type de requête, le problème se pose au niveau des performances, car faire un 'Limit', est relativement très lourd en termes de ressources.

évidement, si nous avons 150 records, cela ira très vite, si nous en avons 1.500.000, c'est autre chose.

5.1 - Cause

En cherchant à améliorer la rapidité et suite à quelques Explain, une des causes est le Order By

```
ROW_NUMBER() OVER(ORDER BY EMPL.EMPNO DESC)
```

On pourrait ne pas mettre de clause Order By, mais alors dans ce cas, on ne peut garantir l'ordre des données entre deux requêtes.

Une des solutions que je préconise, et que lors de l'analyse des besoins de la db, on identifie les tables dont nous aurons besoin de faire un 'Limit', et de lui adjoindre un champ numérique auto incrémenté.

En effet le order by sera beaucoup plus rapide sur un champ numérique que sur un champ contenant des caractères par exemple.

5.2 - Autres pistes

Une autre solution est de pouvoir jouir d'un système de persistance, comme en java par exemple, et dès lors utiliser des Cursor Scrollable, vous trouverez [ici](#) un pdf vous expliquant très bien le processus

Conclusion

L'objectif de cet article était de répondre à une question qui revient régulièrement sur mon lieu de travail, c'est le fruit d'une partie des recherches que j'ai effectuées sur le Net.

Il existe peut-être d'autres façons de faire, notamment pour améliorer la vitesse d'exécution(via les index par exemple). Cela demande approfondissement puisqu'il y a une demande, dans mon entourage en tous-cas . :)

6 - Remerciements

Merci à **Jab**, pour ses conseils et ses pistes :).

Grand merci également à Emmanuelle, et elle seul sait pourquoi :)

Annexe

Table 'EMPLOYEE'

```
CREATE TABLE GG.EMPLOYEE (  
EMPNO CHARACTER(6) NOT NULL,  
FIRSTNME VARCHAR(12) NOT NULL,  
MIDINIT CHARACTER(1) NOT NULL,  
LASTNAME VARCHAR(15) NOT NULL,  
WORKDEPT CHARACTER(3),  
PHONENO CHARACTER(4),  
HIREDATE DATE,  
JOB CHARACTER(8),  
EDLEVEL SMALLINT NOT NULL,  
SEX CHARACTER(1),  
BIRTHDATE DATE,  
SALARY DECIMAL(9,2),  
BONUS DECIMAL(9,2),  
COMM DECIMAL(9,2)  
)  
/  
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,  
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)  
VALUES('000010', 'CHRISTINE', 'I', 'HAAS', 'A00', '3978', '1965-01-01', 'PRES ', 18, 'F',  
'1933-08-24', 52750.00, 1000.00, 4220.00)  
/  
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,  
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)  
VALUES('000020', 'MICHAEL', 'L', 'THOMPSON', 'B01', '3476', '1973-10-10', 'MANAGER ', 18, 'M',  
'1948-02-02', 41250.00, 800.00, 3300.00)  
/  
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,  
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)  
VALUES('000030', 'SALLY', 'A', 'KWAN', 'C01', '4738', '1975-04-05', 'MANAGER ', 20, 'F',  
'1941-05-11', 38250.00, 800.00, 3060.00)  
/  
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,  
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)  
VALUES('000050', 'JOHN', 'B', 'GEYER', 'E01', '6789', '1949-08-17', 'MANAGER ', 16, 'M',  
'1925-09-15', 40175.00, 800.00, 3214.00)  
/  
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,  
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)  
VALUES('000060', 'IRVING', 'F', 'STERN', 'D11', '6423', '1973-09-14', 'MANAGER ', 16, 'M',  
'1945-07-07', 32250.00, 500.00, 2580.00)  
/  
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,  
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)  
VALUES('000070', 'EVA', 'D', 'PULASKI', 'D21', '7831', '1980-09-30', 'MANAGER ', 16, 'F',  
'1953-05-26', 36170.00, 700.00, 2893.00)  
/  
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,  
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)  
VALUES('000090', 'EILEEN', 'W', 'HENDERSON', 'E11', '5498', '1970-08-15', 'MANAGER ', 16, 'F',  
'1941-05-15', 29750.00, 600.00, 2380.00)  
/  
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,  
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)  
VALUES('000100', 'THEODORE', 'Q', 'SPENSER', 'E21', '0972', '1980-06-19', 'MANAGER ', 14, 'M',  
'1956-12-18', 26150.00, 500.00, 2092.00)  
/  
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,  
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
```

```
VALUES('000110', 'VINCENZO', 'G', 'LUCCHESSI', 'A00', '3490', '1958-05-16', 'SALESREP', 19, 'M',
'1929-11-05', 46500.00, 900.00, 3720.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000120', 'SEAN', ' ', 'O'CONNELL', 'A00', '2167', '1963-12-05', 'CLERK', 14, 'M',
'1942-10-18', 29250.00, 600.00, 2340.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000130', 'DOLORES', 'M', 'QUINTANA', 'C01', '4578', '1971-07-28', 'ANALYST', 16, 'F',
'1925-09-15', 23800.00, 500.00, 1904.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000140', 'HEATHER', 'A', 'NICHOLLS', 'C01', '1793', '1976-12-15', 'ANALYST', 18, 'F',
'1946-01-19', 28420.00, 600.00, 2274.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000150', 'BRUCE', ' ', 'ADAMSON', 'D11', '4510', '1972-02-12', 'DESIGNER', 16, 'M',
'1947-05-17', 25280.00, 500.00, 2022.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000160', 'ELIZABETH', 'R', 'PIANKA', 'D11', '3782', '1977-10-11', 'DESIGNER', 17, 'F',
'1955-04-12', 22250.00, 400.00, 1780.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000170', 'MASATOSHI', 'J', 'YOSHIMURA', 'D11', '2890', '1978-09-15', 'DESIGNER', 16, 'M',
'1951-01-05', 24680.00, 500.00, 1974.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000180', 'MARILYN', 'S', 'SCOUTTEN', 'D11', '1682', '1973-07-07', 'DESIGNER', 17, 'F',
'1949-02-21', 21340.00, 500.00, 1707.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000190', 'JAMES', 'H', 'WALKER', 'D11', '2986', '1974-07-26', 'DESIGNER', 16, 'M',
'1952-06-25', 20450.00, 400.00, 1636.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000200', 'DAVID', ' ', 'BROWN', 'D11', '4501', '1966-03-03', 'DESIGNER', 16, 'M',
'1941-05-29', 27740.00, 600.00, 2217.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000210', 'WILLIAM', 'T', 'JONES', 'D11', '0942', '1979-04-11', 'DESIGNER', 17, 'M',
'1953-02-23', 18270.00, 400.00, 1462.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000220', 'JENNIFER', 'K', 'LUTZ', 'D11', '0672', '1968-08-29', 'DESIGNER', 18, 'F',
'1948-03-19', 29840.00, 600.00, 2387.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000230', 'JAMES', 'J', 'JEFFERSON', 'D21', '2094', '1966-11-21', 'CLERK', 14, 'M',
'1935-05-30', 22180.00, 400.00, 1774.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000240', 'SALVATORE', 'M', 'MARINO', 'D21', '3780', '1979-12-05', 'CLERK', 17, 'M',
'1954-03-31', 28760.00, 600.00, 2301.00)
/
```

```
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000250', 'DANIEL', 'S', 'SMITH', 'D21', '0961', '1969-10-30', 'CLERK', 15, 'M',
'1939-11-12', 19180.00, 400.00, 1534.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000260', 'SYBIL', 'P', 'JOHNSON', 'D21', '8953', '1975-09-11', 'CLERK', 16, 'F',
'1936-10-05', 17250.00, 300.00, 1380.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000270', 'MARIA', 'L', 'PEREZ', 'D21', '9001', '1980-09-30', 'CLERK', 15, 'F',
'1953-05-26', 27380.00, 500.00, 2190.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000280', 'ETHEL', 'R', 'SCHNEIDER', 'E11', '8997', '1967-03-24', 'OPERATOR', 17, 'F',
'1936-03-28', 26250.00, 500.00, 2100.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000290', 'JOHN', 'R', 'PARKER', 'E11', '4502', '1980-05-30', 'OPERATOR', 12, 'M',
'1946-07-09', 15340.00, 300.00, 1227.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000300', 'PHILIP', 'X', 'SMITH', 'E11', '2095', '1972-06-19', 'OPERATOR', 14, 'M',
'1936-10-27', 17750.00, 400.00, 1420.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000310', 'MAUDE', 'F', 'SETRIGHT', 'E11', '3332', '1964-09-12', 'OPERATOR', 12, 'F',
'1931-04-21', 15900.00, 300.00, 1272.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000320', 'RAMLAL', 'V', 'MEHTA', 'E21', '9990', '1965-07-07', 'FIELDREP', 16, 'M',
'1932-08-11', 19950.00, 400.00, 1596.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000330', 'WING', ' ', 'LEE', 'E21', '2103', '1976-02-23', 'FIELDREP', 14, 'M',
'1941-07-18', 25370.00, 500.00, 2030.00)
/
INSERT INTO GG.EMPLOYEE(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB,
EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM)
VALUES('000340', 'JASON', 'R', 'GOUNOT', 'E21', '5698', '1947-05-05', 'FIELDREP', 16, 'M',
'1926-05-17', 23840.00, 500.00, 1907.00)
/
```

