LAB 4-1 SOLUTIONS

Insert, update, delete and commit data.

Use the EMP and DEPT tables to complete the following exercises:

1 Insert yourself as a new employee into the EMP table.

```
.   SQL> INSERT INTO emp
      2  VALUES (1111, 'BUSH', 'ANALYST', 7839, SYSDATE, 5000, NULL, 10);

    1 record created.
```

2 Commit the insert.

```
    SQL> COMMIT;

    Commit complete.
```

3 Insert another row into the EMP table prompting for the employee number, department number, hire date, and employee name.

```
    SQL> INSERT INTO emp (EMPNO, DEPTNO, HIREDATE, ENAME)
      2  VALUES (&EMPNO, &DEPTNO, '&HIREDATE', '&ENAME');
    Enter value for EMPNO:  7777
    Enter value for DEPTNO:  10
    Enter value for HIREDATE:  17-DEC-90
    Enter value for ENAME:  COOMBS

    1 record created.
```

4 Create a savepoint named SP1.

```
    SQL> SAVEPOINT SP1;

    Savepoint created.
```

Lab 4-1 Solutions—cont'd

5   Add a new department into the DEPT table with a department number of 99, location of *Maui*, and a department name of *Education*.

```
SQL> DESC dept
```

```
Name                        NULL?       Type
-----------------------      ---------   ---------
DEPTNO                      NOT NULL    NUMBER(2)
DNAME                                   CHAR(14)
LOC                                     CHAR(13)
```

```
SQL> INSERT INTO dept
  2  VALUES(99, 'EDUCATION', 'MAUI');

1 record created.
```

6   Rollback to savepoint SP1. Write a query to display all data from the DEPT table to verify the status of the your newly inserted data.

```
SQL> ROLLBACK TO SAVEPOINT sp1;

Rollback complete.
```

```
SQL> SELECT *
  2  FROM dept;
```

```
DEPTNO    DNAME         LOC
-------   -----------   -------
10        ACCOUNTING    NEW YORK
20        RESEARCH      DALLAS
30        SALES         CHICAGO
40        OPERATIONS    BOSTON
```

Lab 4-1 Solutions—cont'd

7 Commit the changes.

```
SQL> COMMIT;

Commit complete.
```

8 Update your own employee data by giving yourself a $1000 per month raise.

```
SQL> UPDATE emp
  2   SET sal = sal + 1000
  3   WHERE empno = 1111;

1 record updated.
```

9 Delete yourself from the EMP table.

```
SQL> DELETE FROM emp
  2   WHERE empno = 1111;

1 record deleted.
```

10 Commit the changes.

```
SQL> COMMIT;

Commit complete.
```

Lab 4-1 Solutions—cont'd

11 Re-create the view named EMP_DEPTNO_TEN, use the WITH CHECK OPTION,
that includes the employee name, department number, and employee number from
the EMP table. The only employees that will be visable through this view will be
employees in department 10.

```
SQL> CREATE VIEW emp_deptno_ten(NAME, DNO, ENO)
  2  AS SELECT ename, deptno, empno
  3  FROM emp
  4  WHERE deptno= 10
  5  WITH CHECK OPTION;

View created.
```
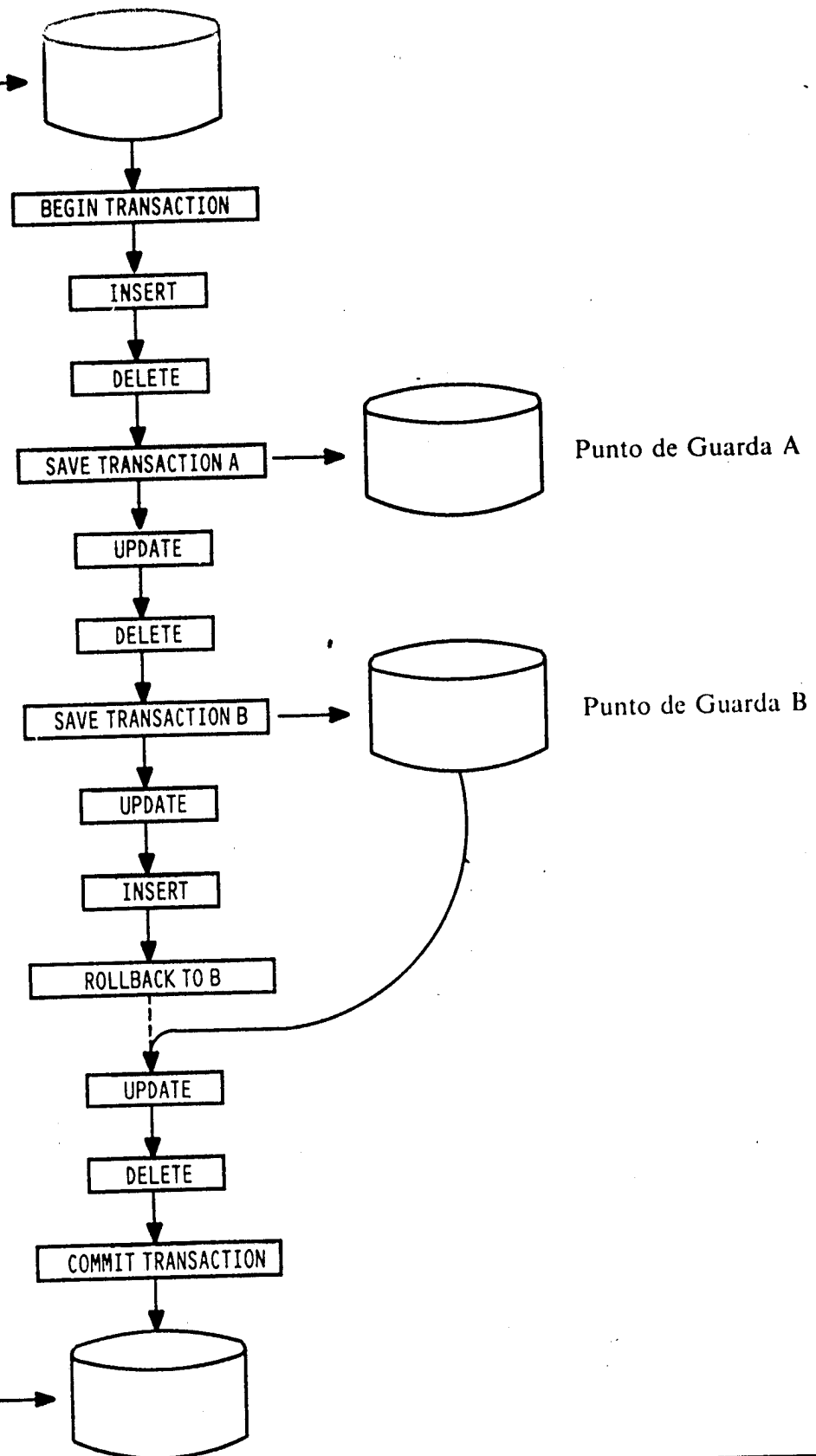
12 Update KING's department from 10 to 20 using the view EMP_DEPTNO_TEN.

```
SQL> UPDATE emp_deptno_ten
  2  SET dno = 20
  3  WHERE name = 'KING';
```
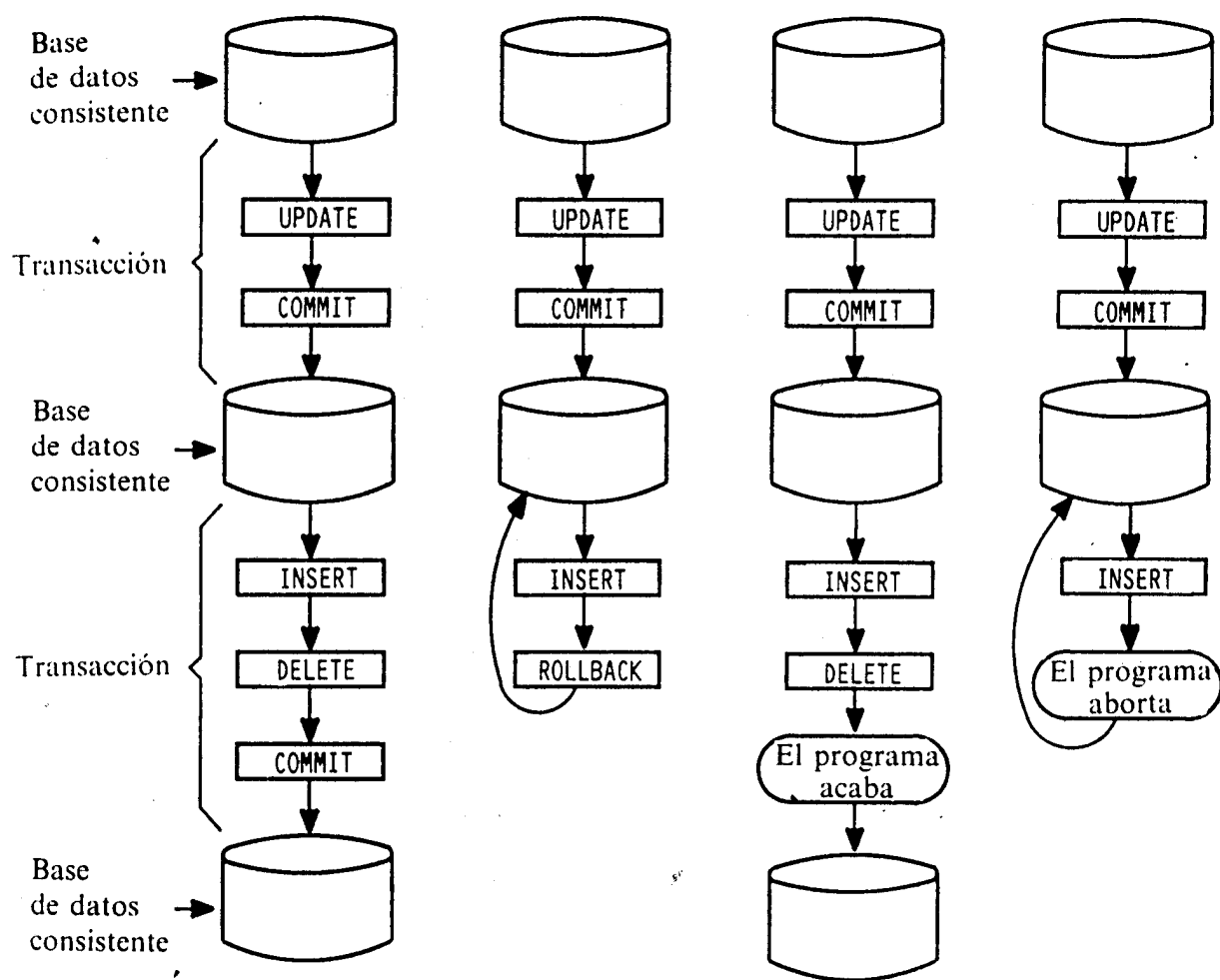
What happened?

```
ERROR at line 2:
ORA-01402: view WITH CHECK OPTION where-clause violation
```

Estado de la base
de datos antes ——→
de la transacción

BEGIN TRANSACTION

INSERT

DELETE

SAVE TRANSACTION A ——→ Punto de Guarda A

UPDATE

DELETE

SAVE TRANSACTION B ——→ Punto de Guarda B

UPDATE

INSERT

ROLLBACK TO B

UPDATE

DELETE

COMMIT TRANSACTION

Estado de la base
de datos después——→
de la transacción

Figura 12.4. *El modelo de transacción de Sybase.*

Figura 12.3. *Transacciones cumplimentadas* COMMIT *y vueltas atrás* ROLLBACK.