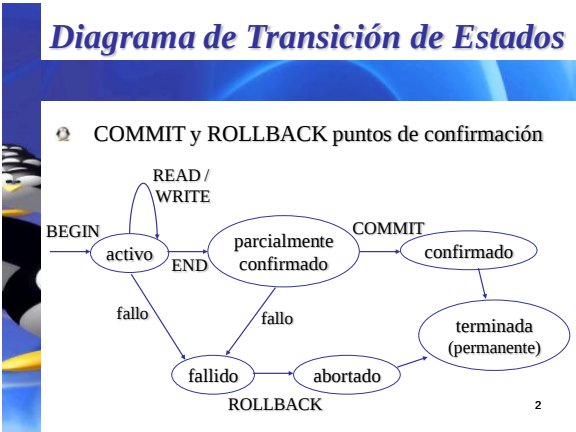






Ejemplo de Commit

```
USE NorthWind
DECLARE @Error int
BEGIN TRAN
UPDATE Products SET UnitPrice=20 WHERE ProductName ='Pan'
SET @Error=@@ERROR
IF (@Error<>0) GOTO TratarError
UPDATE Products SET UnitPrice=40 WHERE ProductName ='Huevo'
SET @Error=@@ERROR
IF (@Error<>0) GOTO TratarError
COMMIT TRAN
TratarError:
If @@Error<>0 THEN
BEGIN
PRINT 'Ha ocurrido un error. Abortamos la transacción'
ROLLBACK TRAN
END
```

Ejemplo de Commit

```
CREATE PROCEDURE spDiscounts(@TYPE VARCHAR(40), @STORE
CHAR(4), @LOW SMALLINT, @HIGH SMALLINT, @DISCOUNT
NUMERIC(9,2))
AS BEGIN
IF @DISCOUNT > 7
BEGIN
RAISERROR ('You entered %d, the discount can not be greater than
7', 10, 1, @DISCOUNT)
END
ELSE BEGIN
BEGIN TRANSACTION INSERT INTO
DISCOUNTS(DISCOUNTTYPE, STOR_ID, LOWQTY, HIGHQTY,
DISCOUNT)
VALUES(@TYPE, @STORE, @LOW, @HIGH, @DISCOUNT)
IF @@ERROR <> 0
ROLLBACK TRANSACTION
ELSE COMMIT TRANSACTION
END
```

4

Ejemplo de Commit

```
START TRANSACTION;
SELECT @A:=SUM(salary) FROM
table1 WHERE type=1;
UPDATE table2 SET summary=@A
WHERE type=1;
IF @@ERROR <> 0
ROLLBACK TRANSACTION
ELSE COMMIT TRANSACTION;
```

5

Transacciones en SQL

- ⌚ Cada una de las transacciones es tratada por los sistemas de BD como una transacción, emitiendo un COMMIT automático si se realizó bien la transacción o en su defecto un ROLLBACK.
- ⌚ Sin embargo algunos sistemas permiten inhibir estos COMMIT y ROLLBACK
- ⌚ **Cuidado con bloqueos a la BD**

6

Recuperación del Sistema

- ¿Qué pasa cuando ocurre una falla del sistema y se pierde la información en la memoria RAM?
- Parte de la información de la bitácora que estaba en la RAM se pierde al reiniciar el sistema.

7

Punto de Verificación

- A determinados intervalos especificados el sistema en forma automática realiza un punto de verificación. Estos intervalos usualmente están especificados por cantidades de transacciones.

8

Recuperación del Sistema



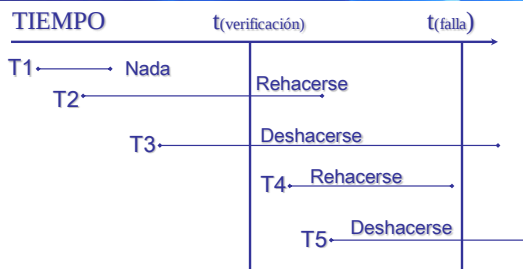
"Somebody broke into your computer, but it looks like the work of an inexperienced hacker."

9

Punto de Verificación

- 🕒 El punto de verificación involucra:
 - 🕒 Escritura forzada, disco físico, de la información de la BD
 - 🕒 Escribir físicamente un registro de punto de verificación especial en la bitácora física, en disco.
 - 🕒 Este punto, da una lista de todas las transacciones que estaban en progreso en la BD en el momento que se realizó.¹⁰

Ejemplo



11

Volver a un Punto de Verificación

- 🕒 Una vez que se reinicia el sistema primeramente tiene que DESHACER y luego tiene que REHACER las transacciones.
- 🕒 Una vez terminado esto el sistema de BD está listo para recibir más transacciones.

12

Ejemplo de Punto de Verificación

- 🐧 Instrucción de Punto de Verificación
- 🐧 SAVEPOINT identifier
- 🐧 Instrucción de Recuperación a un Punto de Verificación
- 🐧 ROLLBACK TO SAVEPOINT identifier

13

OQL y SQL 3



OQL y SQL3

- 🐧 OQL, {Object Query Language}, esfuerzo de los programas Orientados a Objetos de incorporar SQL
- 🐧 SQL3 es la intención de utilizar programación Orientada a Objetos dentro de SQL.
- 🐧 Son muy similares

15

Ejemplo 1

```
❏ CREATE TABLE Ventas
    (art# CHAR(5),
    cant INTEGER ARRAY[3],
    PRIMARY KEY (art#));
❏ SELECT art#
    FROM Ventas
    WHERE cant[2] > 100;
❏ INSERT INTO Ventas (art#, cant)
❏ VALUES ('casa1', ARRAY [0,0,0]);
```

16

Ejemplo 2

```
❏ CREATE TABLE Cliente
    (cliente# CHAR(3),
    Dir ROW (calle CHAR(50),
    número INTEGER,
    ciudad CHAR(20)),
    PRIMARY KEY (cliente#));
❏ SELECT cliente#
    FROM Cliente
    WHERE Dir.ciudad = 'Santiago';
❏ INSERT INTO Cliente (cliente#, Dir)
❏ VALUES ( '45', ROW ('Apoquindo',250, 'Santiago'17));
```

Herencia de Tipo

```
❏ CREATE TYPE Punto
    AS (x FLOAT, y FLOAT) FINAL;
❏ CREATE TYPE Elipse
    AS (a Longitud, b Longitud, centro Punto)
    NOT FINAL;
❏ CREATE TYPE Circulo UNDER Elipse
    AS (r Longitud)
    NOT FINAL; //Significa que puede tener subtipos
```

18

Herencia de Tipo

```
CREATE TABLE ciudades
(nombre TEXT,
población INTEGER,
altitud INTEGER);

CREATE TABLE capital
( region CHAR(2) ) INHERITS (ciudades);
```

19

Herencia de Tipo

```
CREATE TYPE PersonaTipo AS OBJECT
(Rut INTEGER,
Nombre CHAR(20),
Apellido CHAR(20),
Domicilio CHAR(30),
Ciudad CHAR(20),
FechaNac DATE,
MEMBER FUNCTION edad RETURN INTEGER)
NOT FINAL;
```

20

Herencia de Tipo

```
CREATE TYPE EstudianteTipo UNDER
PersonaTipo AS
(NumAlum INTEGER,
Especialidad CHAR(20),
AñoIngreso INTEGER,
MEMBER FUNCTION añosEstudio RETURN
INTEGER)
NOT FINAL;
```

21

Herencia de Tipo

```
CREATE TYPE ProfesorTipo UNDER PersonaTipo
AS
  (NumProf INTEGER,
   Cargo CHAR(20),
   Fecha_Ingreso DATE,
   MEMBER FUNCTION antigüedad RETURN
   INTEGER)
NOT FINAL;
```

22

FIN

Andrés Moreno S.