



Departamento de Informática
Universidad Técnica Federico Santa María



Unidad VII

Procesamiento de Consultas y Transacciones

INF-239, ILI-239 Bases de Datos

Profesora Cecilia Reyes Covarrubias – Casa Central

Diapositivas realizadas con la colaboración Prof. J.Luis Martí – Campus San Joaquín



TEMARIO UNIDAD VII

7.1 Procesamiento de Consultas

7.2 Procesamiento de Transacciones



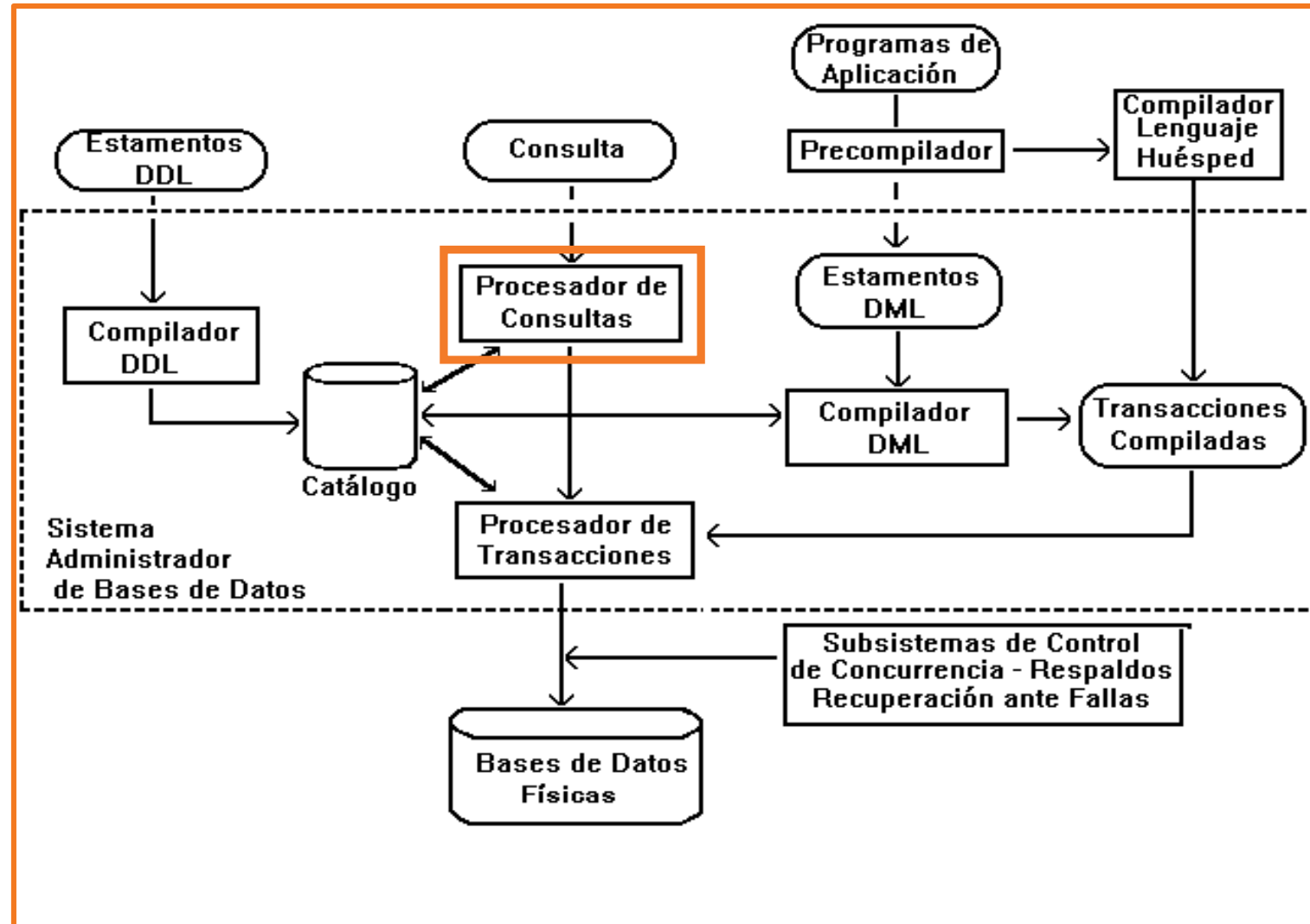
7.1 PROCESAMIENTO DE CONSULTAS



Departamento de Informática
Universidad Técnica Federico Santa María



COMPONENTES DE UN DBMS



PROCESAMIENTO DE CONSULTA

- El procesamiento de consulta tiene por objetivo identificar una forma conveniente (rápida, barata) de ejecutar una consulta sobre la BD, por lo general ingresada como un comando SQL.
- El procesador de consulta debe convertir dicho comando SQL en una secuencia de operaciones de álgebra relacional, las que se organizan entre sí mediante un esquema jerárquico (llamado plan de acceso/árbol de acceso/plan de ejecución).
- Por ejemplo, considerar el siguiente comando SQL:

```
SELECT factura.#factura, cliente.razon_social  
FROM cliente, factura, detalle, producto  
WHERE producto.tipo = 'ABC'  
        and producto.#producto = detalle.#producto  
        and detalle.#factura = factura.#factura  
        and factura.fecha >= '01/07/2012'  
        and factura.RUT_cliente = cliente.RUT;
```

PROCESAMIENTO DE CONSULTA

Comando SQL

```
SELECT factura.#factura, cliente.razon_social
FROM cliente, factura, detalle, producto
WHERE producto.tipo = 'ABC'
      and producto.#producto = detalle.#producto
      and detalle.#factura = factura.#factura
      and factura.fecha >= '01/07/2012'
      and factura.RUT_cliente = cliente.RUT;
```

Equivalencias Algebra Relacional

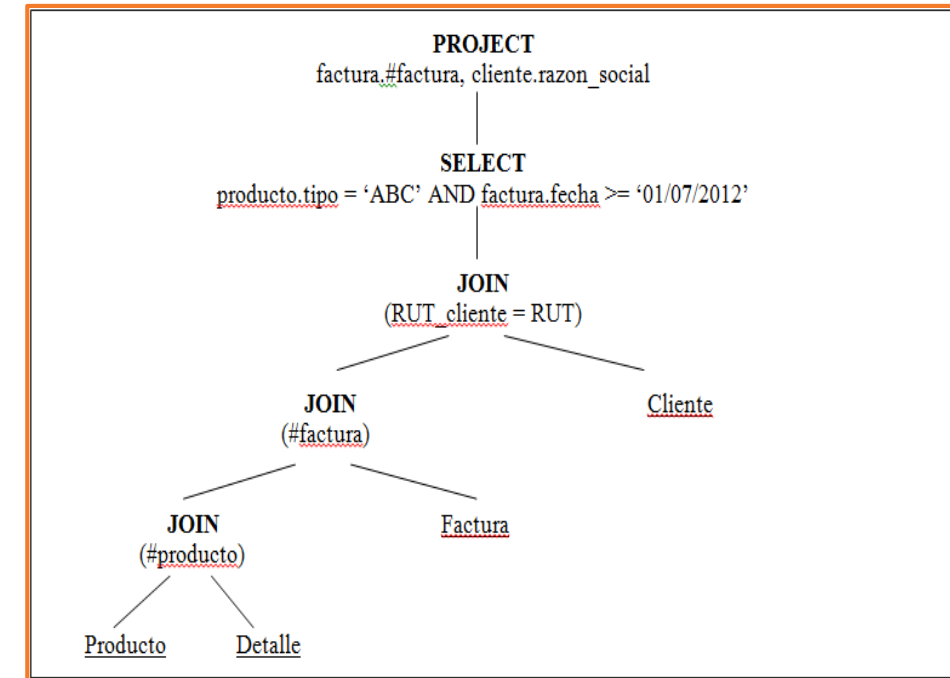
Selecciones:

producto.tipo = 'ABC'
factura.fecha >= '01/07/2012'

Joins:

producto.#producto = detalle.#producto
detalle.#factura = factura.#factura
factura.RUT_cliente = cliente.RUT;

Plan de Acceso Canónico

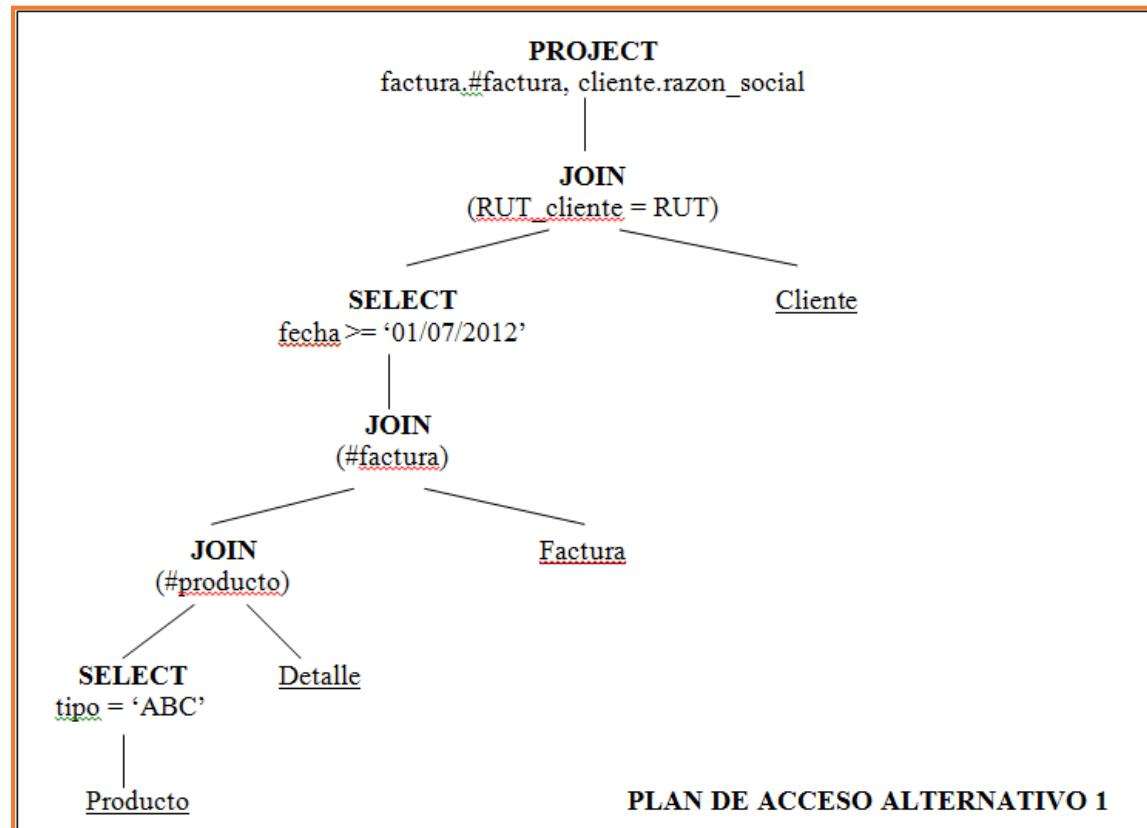


PROCESAMIENTO DE CONSULTA

- A continuación, se van probando diversas modificaciones en dicha representación, como cambios de ubicación de las operaciones y formas diferentes de ejecutar la misma operación, buscando una manera más adecuada de hallar el mismo resultado (etapa que se denomina **optimización de consultas**).
- Tras haber probado muchas posibilidades de ejecución, idealmente el total de ellas, se escoge la que se ejecuta en el menor tiempo o bien la que consume menos recursos, y es la que se lleva a cabo para responder la consulta del usuario.

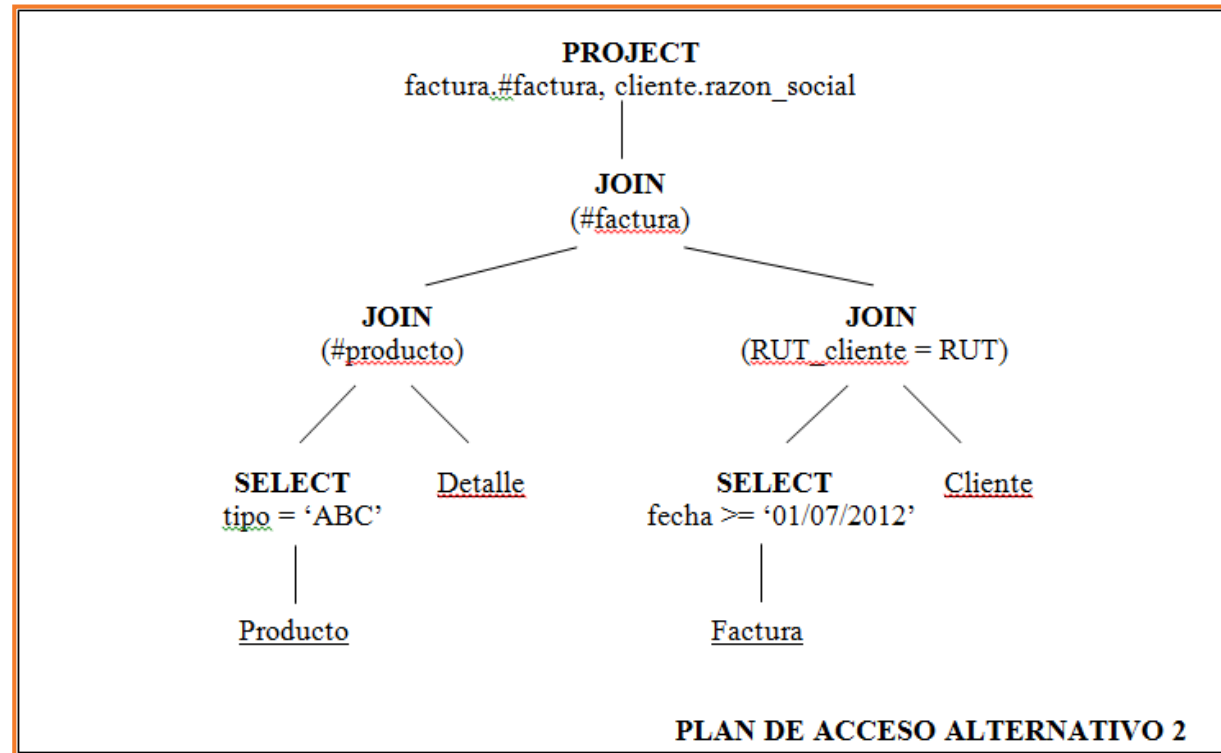
PROCESAMIENTO DE CONSULTA

EJEMPLO PLAN DE ACCESO ALTERNATIVO 1



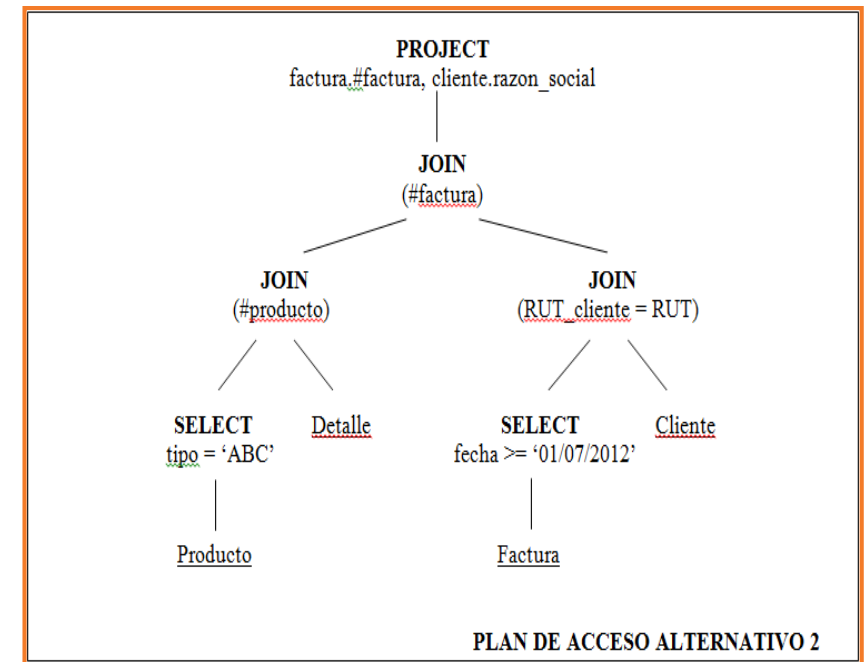
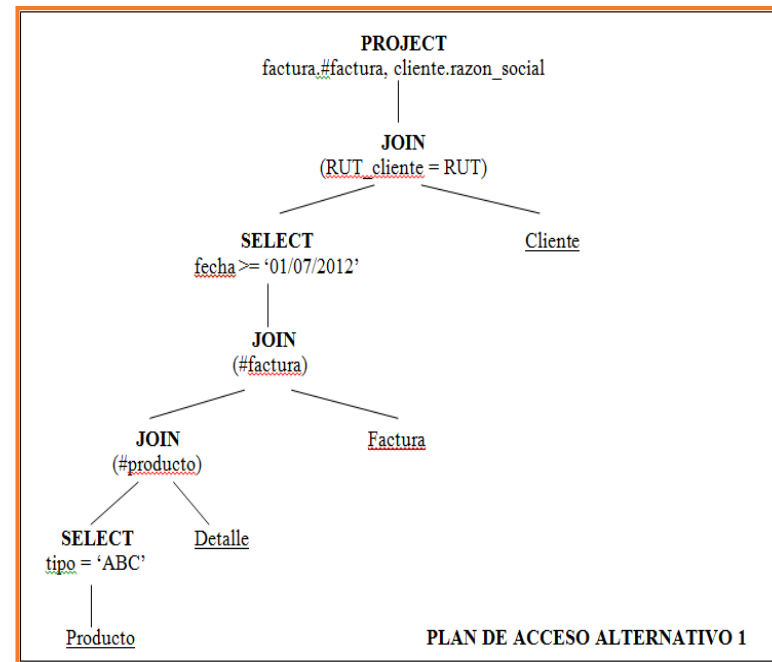
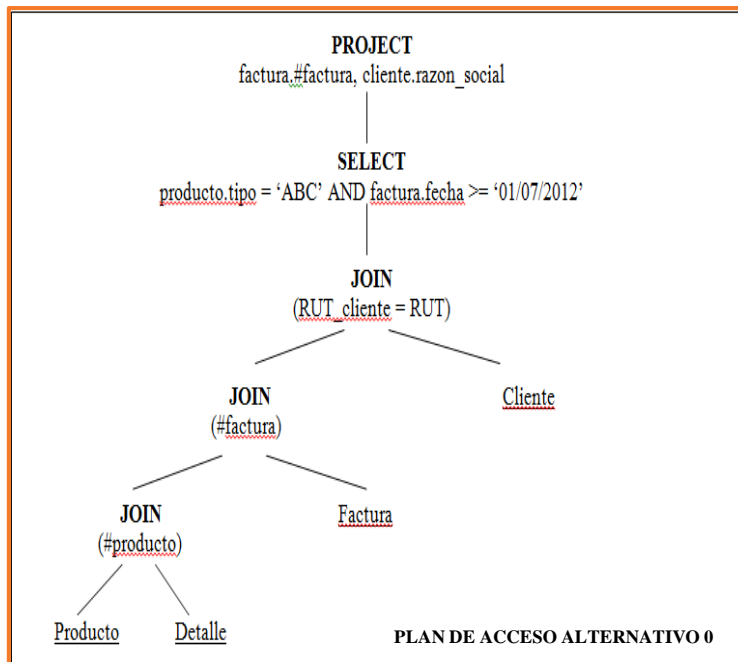
PROCESAMIENTO DE CONSULTA

EJEMPLO PLAN DE ACCESO ALTERNATIVO 2



PROCESAMIENTO DE CONSULTA

- De los tres planes de acceso anteriores, ¿cuál es el que, probablemente, consume menos tiempo?



PROCESAMIENTO DE CONSULTA

- Para poder apoyar la **etapa de optimización**, es necesario trabajar con **funciones de costo** sobre cada forma de ejecutar cada operación de álgebra relacional, de modo de contar con un mecanismo de comparación entre las diversas formas de ejecución, e identificar correctamente la más conveniente.

PROCESAMIENTO DE CONSULTA

- Funciones de costo para operador Selección, los parámetros a considerar son:
 - b: número de bloques del archivo.
 - r: número de registros del archivo.
 - fb: factor de bloqueo del archivo.
 - s: cardinalidad de la selección.
 - x: número de niveles de un índice.
 - b1: número de bloques del último nivel del índice (o nivel de las hojas)

PROCESAMIENTO DE CONSULTA

- Funciones de costo para evaluar un Join, a partir de los siguientes parámetros presentes en un diccionario de datos...
 - bR: número de bloques del archivo R.
 - r: número de registros del archivo R.
 - bS: número de bloques del archivo S.
 - s: número de registros del archivo S.
 - fbS: factor de bloqueo del archivo S
 - sS: número de registros del archivo S (cardinalidad de la selección del archivo S).
 - xS: número de niveles de un índice sobre el atributo de join de S.
 - sJ: selectividad del Join.

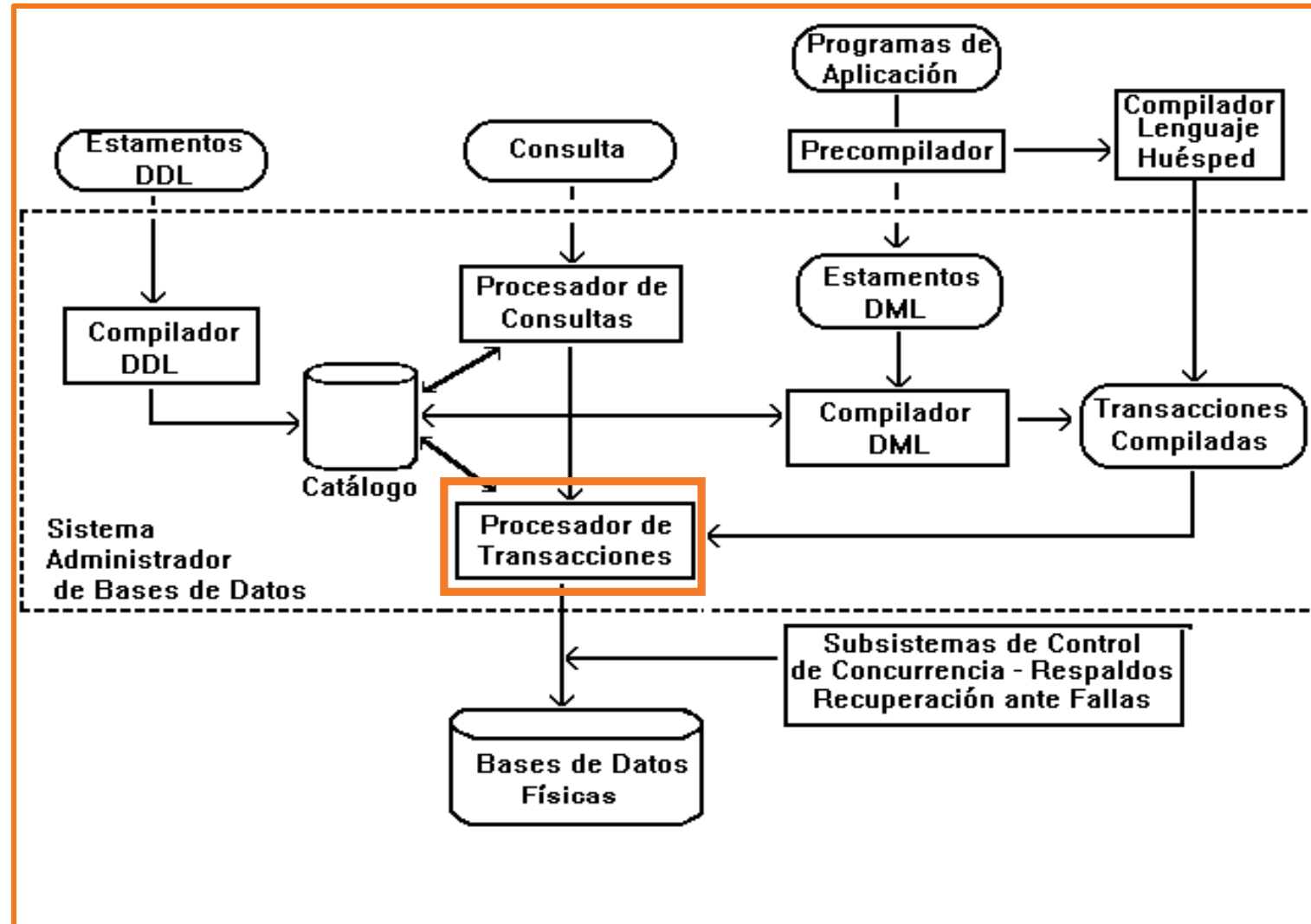
7.2 PROCESAMIENTO DE TRANSACCIONES



Departamento de Informática
Universidad Técnica Federico Santa María



COMPONENTES DE UN DBMS



PROCESAMIENTO DE TRANSACCION

RECORDAR...

- Una Transacción es un conjunto de operaciones (u órdenes) que se ejecutan en forma indivisible (atómica) sobre una BD.
- El DBMS debe mantener la integridad de los datos, haciendo que estas transacciones no puedan finalizar en un estado intermedio.
- Si por algún motivo se debe cancelar la transacción (por ejemplo: división por cero, privilegios de acceso, etc.), el DBMS empieza a deshacer (ROLLBACK) las órdenes ejecutadas hasta dejar la BD en su estado inicial (llamado punto de integridad), como si la orden de la transacción nunca se hubiese realizado.
- Compromiso (COMMIT) de una Transacción: indicación de que una transacción se ha ejecutado por completo, y que todos los cambios se deben guardar “para siempre” en la base de datos.
- Un DBMS debe proveer las funciones necesarias para que sus transacciones tengan las características ACID (Atomicity, Consistency, Isolation and Durability).

PROCESAMIENTO DE TRANSACCION

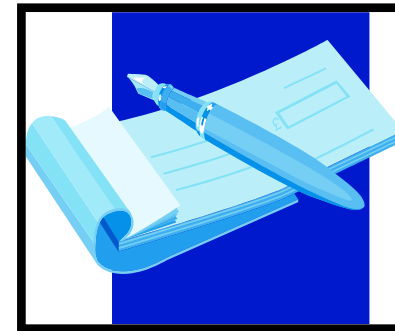
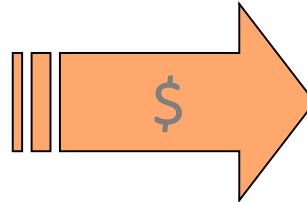
RECORDAR...

Ejemplo Típico: Transferencias Bancarias



Cuenta Origen

Operación 1: girar dinero
(disminuir SALDO Cuenta Origen)



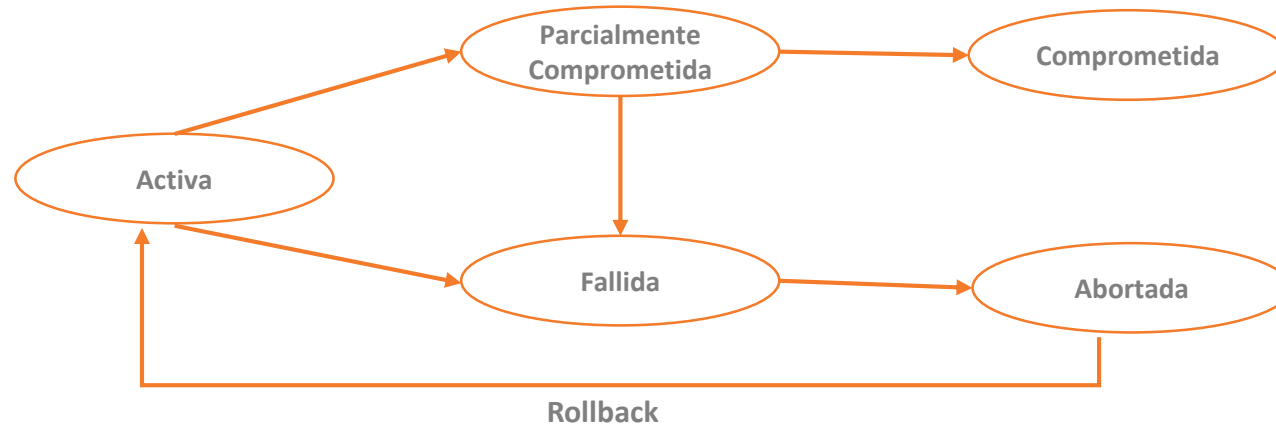
Cuenta Destino

Operación 2: depositar dinero
(aumentar SALDO Cuenta Destino)

Se debe garantizar que no se pierda el dinero, para ello las dos operaciones deben ser atómicas, es decir, el DBMS debe garantizar que, bajo cualquier circunstancia (incluso una caída del sistema), el resultado final sea: se han realizado las dos operaciones o ninguna de ellas.

PROCESAMIENTO DE TRANSACCION

- Una transacción puede tener los siguientes estados:
 - Activa (Active): estado inicial y durante su ejecución.
 - Parcialmente comprometida (Uncommitted): después de ejecutarse la última transacción.
 - Fallida (Failed): cuando detecta que no se puede seguir con la ejecución normal.
 - Abortada (Rolled Back): deshecha la transacción y restablecida la BD a sus estado anterior.
 - Comprometida (Committed): completada con éxito la transacción.



PROCESAMIENTO DE TRANSACCION EN SQL

Tipos de transacciones:

- **Unitaria o simple (implícitas):** un solo comando SQL requiere ser ejecutado. Por ejemplo:

```
UPDATE Productos SET Precio=200 WHERE NombreProducto ='Clavo'
```

- **Definida por el usuario (explícitas):** múltiples comandos SQL requieren ser ejecutados por ello se necesita delimitar su alcance a través de BEGIN TRANSACTION y END (COMMIT) TRANSACTION.

```
BEGIN TRANSACTION T1
```

```
UPDATE Productos SET Precio = Precio*2 WHERE CodProd = 11
```

```
UPDATE Productos SET Precio = Precio*1.1 WHERE CodProd = 12
```

```
....
```

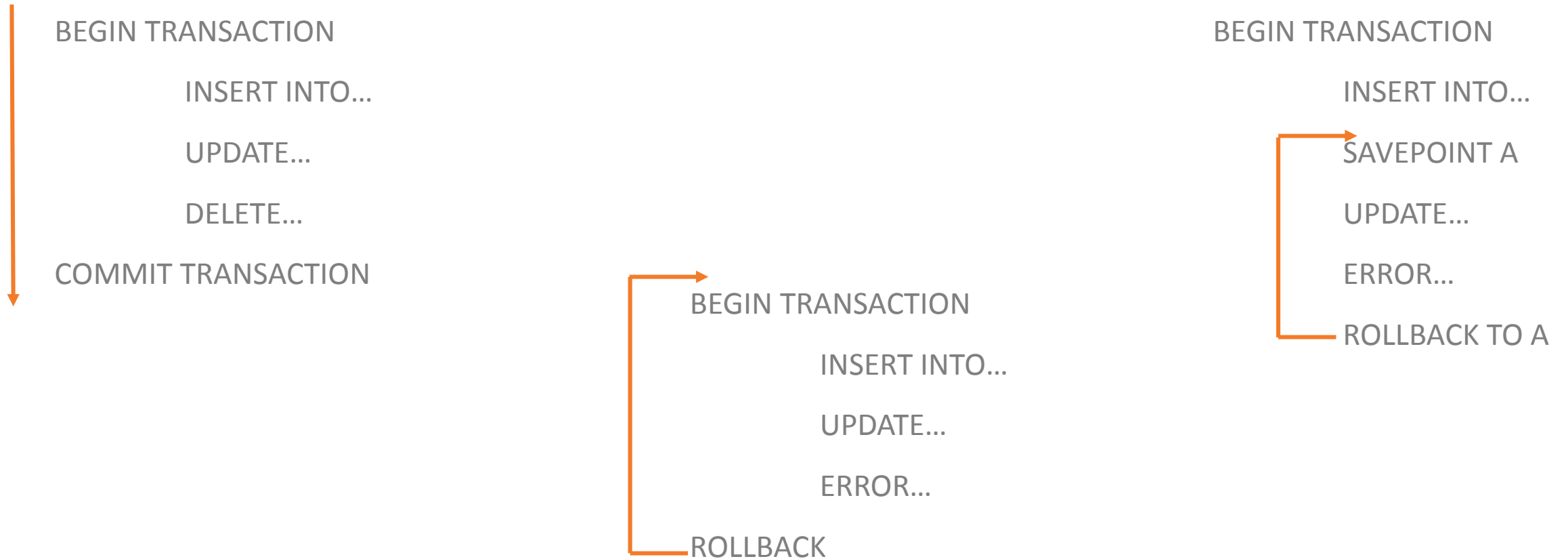
```
COMMIT TRAN T1
```

PROCESAMIENTO DE TRANSACCION EN SQL

Los siguientes **comandos SQL** son utilizados para tener control de las transacciones:

- **BEGIN TRANSACTION:** establece el punto de partida de una transacción explícita.
- **ROLLBACK TRANSACTION:** deshace la transacción y deja la BD en el estado en que estaban justo antes del inicio de la transacción.
- **COMMIT TRANSACTION:** finaliza la transacción si no se han encontrado errores.
- **SAVEPOINT:** define un punto dentro de la misma transacción al que se puede volver si parte de la transacción se cancela (solo se deshacen los cambios hasta la señal indicada).

PROCESAMIENTO DE TRANSACCION EN SQL



PROCESAMIENTO DE TRANSACCION EN SQL

Integridad de transacciones:

- **Autocommit:** en el caso de transacciones simples, algunos RDBMS automáticamente realizan un COMMIT o un ROLLBACK después de ejecutado el comando.
- **Log File:** archivo creado con el BEGIN TRANSACTION para ir registrando todos los cambios (inserciones, eliminaciones y actualizaciones) a la BD. Cuando se llega al END TRANSACTION o COMMIT se toma contenido del LogFile y se aplica a la BD, así los cambios se vuelven permanentes.
- **SET AUTOCOMMIT (ON/OFF):** comando que especifica si los cambios serán o no permanente. Las transacciones definidas por el usuario mejoran el desempeño pues los cambios se hacen cuando se ejecutan todas las operaciones exitosamente, por lo que para ellas conviene que esté OFF.