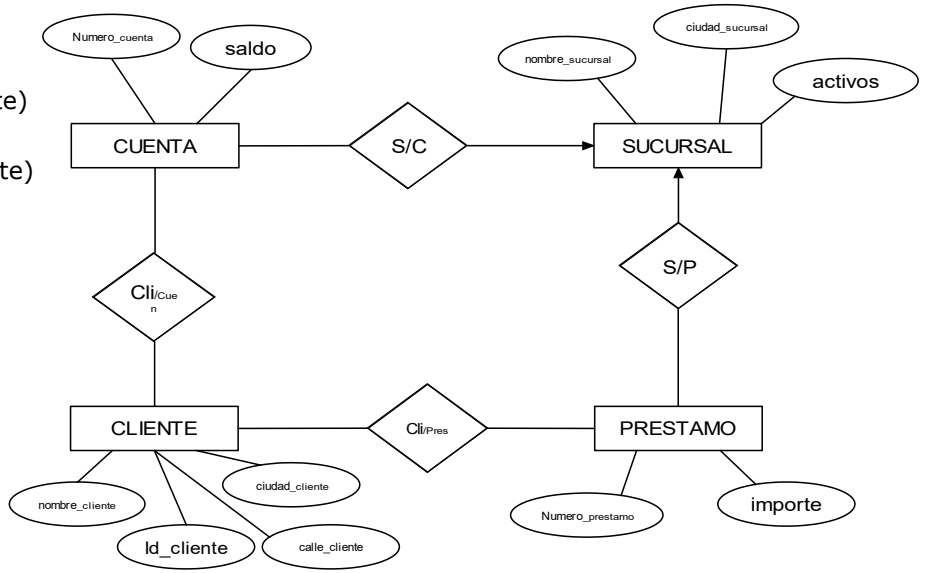
Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería

Cátedra: **Bases de datos I**

Optimización de Consultas

Esquema_sucursal=(nombre_sucursal, activo, ciudad_sucursal)
Esquema_cliente=(Id_cliente, nombre_cliente, calle, ciudad_cliente)
Esquema_cuenta=(nombre_sucursal, número_cuenta, saldo)
Esquema_préstamo=(nombre_sucursal,número_préstamo, importe)
Esquema_cli/cuen=(nombre_cliente, número_cuenta)
Esquema_Cli/Pres= (nombre_cliente, numero_prestamo)



- Procesamiento De Consultas

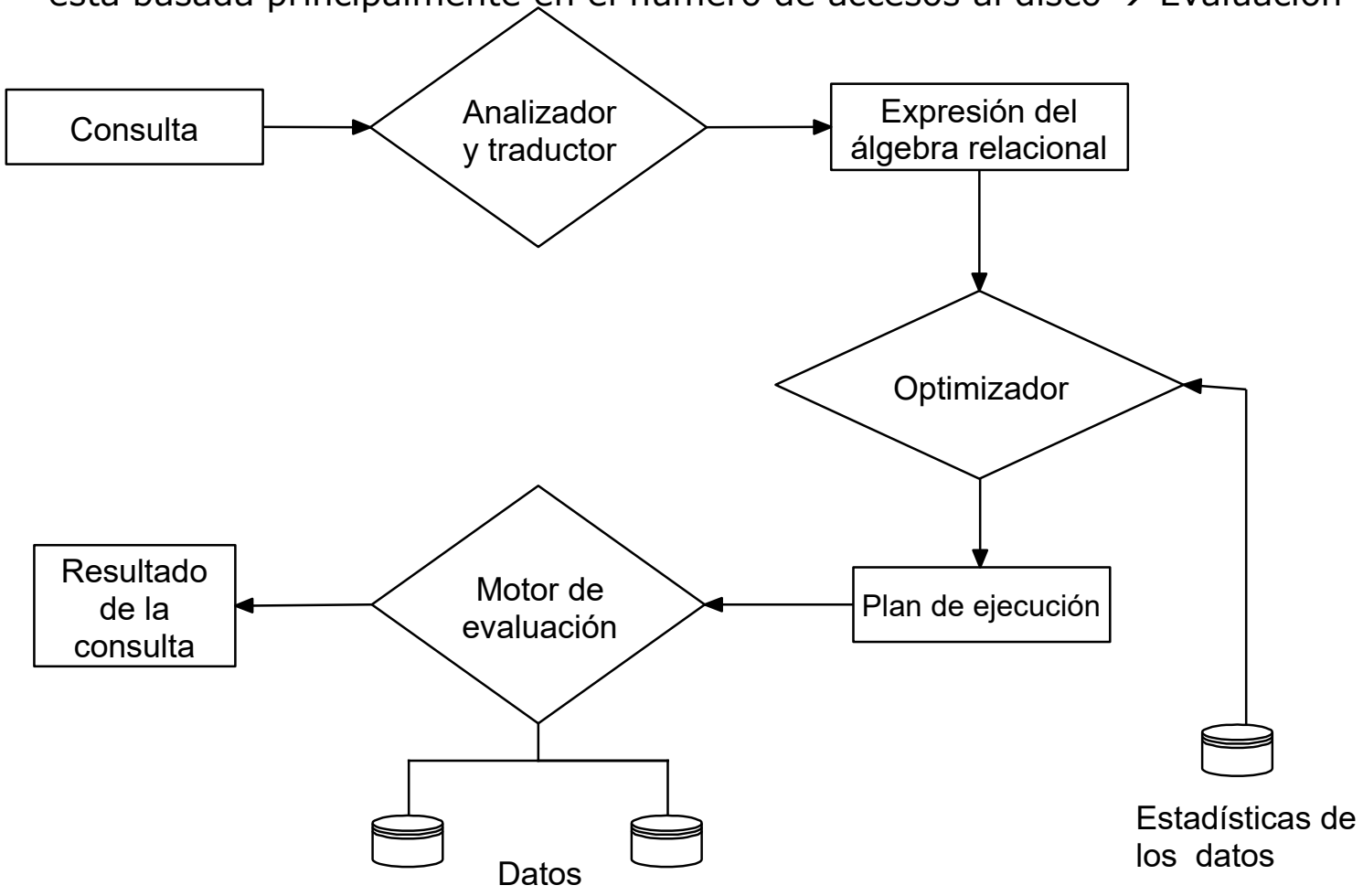
  - Cuando se presenta una consulta al sistema, es necesario hallar el mejor método para encontrar la respuesta utilizando la estructura existente de la BD. El costo de procesar una consulta, normalmente está dominado por el acceso al disco, comparativamente más lento que el acceso a memoria.

- Interpretación de consultas

  - Normalmente hay muchas estrategias posibles para procesar una consulta dada, especialmente si la consulta es compleja. Una consulta se puede expresar de varias formas y cada forma "sugiere" una estrategia para encontrar la respuesta. La diferencia entre una buena estrategia y otra mala, en términos del número de accesos a disco que necesitan, es a menudo importante. Por esta razón, vale la pena que el sistema gaste tiempo en seleccionar una buena estrategia para el procesamiento de la consulta, aunque la misma se ejecute una única vez.
  - El sistema debe encargarse de transformar una consulta como la introdujo el usuario en una consulta equivalente que pueda calcularse de manera más eficiente. Esta búsqueda se llama **optimización de consultas**.
  - En el modelo red y jerárquico la optimización la realiza el programador de aplicaciones, esto es porque el lenguaje de manipulación de datos de estos modelos tiene instrucciones incorporadas en el lenguaje de aplicación y no es fácil transformar una consulta a una equivalente si no se conoce el programa completo.
  - Al utilizar **SQL**, una consulta debe ser traducida a una forma en que se pueda trabajar → emplearemos la **forma relacional**.

– Acciones a tomar en una consulta:

1. Traducirla a su forma relacional.
2. Se intenta encontrar una expresión equivalente, pero que pueda ejecutarse de manera eficiente → Optimización
3. Seleccionar una estrategia detallada para procesar la consulta. La elección final está basada principalmente en el número de accesos al disco → Evaluación



- Antes de empezar el procesamiento de una consulta, el sistema debe traducirla a una forma interna. Un lenguaje como SQL es adecuado para el uso humano, pero es poco apropiado para la representación interna, por eso se traduce a una representación basada en el álgebra relacional. Es un proceso similar al que realiza el analizador de un compilador.
- Durante la generación del formato interno de una consulta, el analizador comprueba la sintaxis, verifica que los nombres de las relaciones que aparecen en la consulta sean nombres de relaciones de la BD, etc. Luego se construye un árbol para el análisis de la consulta, que se transformará en una expresión del álgebra relacional.
- Cada consulta en SQL se puede traducir en una expresión del álgebra relacional de varias formas. Por ejemplo, consideremos la siguiente consulta:

```
Select saldo
From cuenta
Where saldo>500
```

- Esta consulta se puede traducir de las siguientes formas:

$$\pi_{saldo}\,(\sigma_{saldo>"500"}\,(cuenta))$$

$$\sigma_{saldo>"500"}\,(\pi_{saldo}\,(cuenta))$$

- Estas expresiones son equivalentes. Una secuencia de operaciones que se pueden utilizar para evaluar una consulta establecen un plan de evaluación o de ejecución de la consulta. Los diferentes planes de evaluación para una consulta pueden tener costes distintos. No se espera que el usuario elija la consulta con menor coste, esto es responsabilidad del sistema.

- La optimización de consultas es el **proceso de selección del plan de evaluación de la consulta más eficiente**.
- Para decidirse entre distintos planes de evaluación, el optimizador debe estimar el costo de cada plan de evaluación. Normalmente no es posible evaluar el costo real sin ejecutar el plan, en lugar de esto, los optimizadores hacen uso de información estadística acerca de las relaciones, como los tamaños de las relaciones, los índices si existen, para hacer una buena optimización.

• Equivalencia de expresiones

- El álgebra relacional es un lenguaje procedural o procedimental, es decir que representa una secuencia determinada de operaciones. Consideremos la siguiente expresión de AR para la consulta: Encontrar todos los clientes que tengan una cuenta en cualquier sucursal de Madryn

$$\pi_{Id\_cliente}(\sigma_{ciudad\_sucursal='Madryn'}(sucursal \bowtie (cuenta \bowtie cli/cuen)))$$

- Esta expresión construye una relación intermedia muy grande, sin embargo estamos interesados en una pocas tuplas de esta relación (aquellas que pertenecen a sucursales de Madryn) y sólo en uno de los seis atributos, id_cliente.
- El resultado intermedio probablemente será demasiado grande para conservarlo en la memoria principal, por lo que debe almacenarse en disco. Esto significa que además de acceder al disco para leer las relaciones cli/cuen, cuenta y sucursal, el sistema necesita acceder al disco para leer y escribir resultados intermedios. ¿Hay una forma de reducir el tamaño del resultado intermedio?

- Dada una expresión del AR, es trabajo del optimizador de la consulta **alcanzar un plan de evaluación** que calcule el mismo resultado que la expresión dada, pero de la forma menos costosa.

- La generación de planes de evaluación de consultas implica **generar expresiones que son lógicamente equivalentes a la expresión dada** → El optimizador de consultas genera las distintas expresiones mediante las reglas de equivalencia, que especifican cómo transformar una expresión en otra lógicamente equivalente.

- Además, se debe elegir un plan para evaluar las expresiones y elegir las menos costosas, esto se llama **optimización basada en el coste**. Puesto que el coste es una estimación, puede ser que el plan seleccionado no sea el menos costoso, sin embargo, mientras que la estimación sea buena, probablemente el plan sea el menos costoso o uno no mucho más costoso que éste.

- Reglas de equivalencia (Heurísticas)

  – Una regla de equivalencia dice que las expresiones de dos formas son equivalentes, es decir se puede transformar una en otra mientras se preserva la equivalencia. Por preservar la equivalencia se quiere decir que las relaciones generadas por las dos expresiones tienen el mismo conjunto de atributos y el mismo conjunto de tuplas, aunque sus atributos pueden estar ordenados de manera distinta.

1. Las operaciones de selección conjuntas se pueden desglosar en una secuencia de selecciones individuales. Entonces se sugiere efectuar las **selecciones tan pronto como sea posible** $\sigma_{\theta_1 \cap \theta_2}(E) = \sigma_{\theta_1}(\sigma_{\theta_2}(E))$
2. Las operaciones de **selección son conmutativas** $\sigma_{\theta_1}(\sigma_{\theta_2}(E)) = \sigma_{\theta_2}(\sigma_{\theta_1}(E))$
3. La **proyección reduce el tamaño de las relaciones**. Así, siempre que necesitemos generar una relación temporal, resulta provechoso aplicar inmediatamente todas las proyecciones posibles, para poder eliminar varios atributos del esquema, los únicos que debemos conservar son aquellos que aparecen en el resultado de la consulta o se necesitan para procesar operaciones posteriores. Al eliminar atributos que no se necesitan reducimos el número de columnas del resultado intermedio.
4. Elegir un **ordenamiento óptimo de operaciones de producto**. El producto natural es asociativo. Así, para todas las relaciones r1, r2 y r3:

   $(r1 \bowtie r2) \bowtie r3 = r1 \bowtie (r2 \bowtie r3)$

   Esta regla es importante porque puede ser que las relaciones r1 y r2 no tengan atributos en común, entonces el producto natural se transforma en un producto cartesiano, que tiene como resultado un número muy grande de tuplas.

5. El **producto natural es conmutativo**.
   $(r1 \bowtie r2) = (r2 \bowtie r1)$

6. Los productos naturales, proyecciones y selecciones se presentan con frecuencia en la práctica. Los productos cartesianos y naturales son unas de las operaciones más costosas en el procesamiento de consultas. Sin embargo, hacemos notar que se cumplen equivalencias similares a las ya mencionadas en el caso de operaciones de unión y diferencia de conjuntos. **Equivalencias**:

   $\sigma_P (r1 \cup r2) = \sigma_P (r1) \cup \sigma_P (r2)$

   $\sigma_P (r1 - r2) = \sigma_P (r1) - \sigma_P (r2)$

   $(r1 \cup r2) \cup r3 = r1 \cup (r2 \cup r3)$

   $r1 \cup r2 = r2 \cup r1$

↳ La elección final de la estrategia se hace sólo después de analizar detalladamente cada una y de estimar su costo de procesamiento.

- Estimación del costo de Procesamiento de Consultas

  - La estrategia que elijamos para una consulta dependerá del **tamaño de cada relación y de la distribución de los valores dentro de las columnas**.
  - En el ejemplo utilizado, la fracción de sucursales situadas en Madryn tiene un impacto importante en la utilidad de las técnicas.
  - Para poder elegir una estrategia basada en información fiable, los SGBD pueden almacenar estadísticas para cada relación r. Estas estadísticas incluyen:

    - ✓ n_r número de tuplas en la relación r.
    - ✓ s_r tamaño de una tupla de la relación r en bytes
    - ✓ V(A, r) número de valores distintos que aparecen en r para el atributo A.

  - Las dos primeras estadísticas nos permiten estimar con exactitud el tamaño de un producto cartesiano:
    - El producto cartesiano **r x s** contiene $\mathbf{n_r}$ y $\mathbf{n_s}$ tuplas.
    - Cada tupla **r x s** ocupa $\mathbf{s_r + s_s}$ bytes.

- Estimación del costo de Procesamiento de Consultas (Cont.)

  - ✓ n_r número de tuplas en la relación r.
  - ✓ s_r tamaño de una tupla de la relación r en bytes
  - ✓ V(A, r) número de valores distintos que aparecen en r para el atributo A.

  – La tercera estadística se usa para estimar cuántas tuplas satisfacen un predicado de selección de la forma:
  <nombre_atributo> = <valor>

  – Sin embargo, para realizar una estimación así, necesitamos saber cuántas veces aparece cada valor en una columna.
  – Si suponemos que la distribución de valores es uniforme (es decir, cada valor tiene la misma probabilidad de aparecer), entonces se estima que la consulta

  $\sigma_{A=a}(r)$ tiene (tuplas): $\frac{n_r}{V(A,r)}$

  – Sin embargo, puede que no sea realista la suposición de que cada valor aparezca con la misma probabilidad. El atributo nombre_sucursal en la relación cuenta es un ejemplo de este caso y existe una tupla en la relación cuenta para cada cuenta. Es razonable esperar que las sucursales grandes tengan más cuentas que las pequeñas. Por tanto, algunos valores de nombre_sucursal tienen mayor probabilidad de aparecer que otros.

- Estimación del costo de Procesamiento de Consultas (Cont.)
  - A pesar del hecho de que la suposición de distribución no siempre se cumple, en muchos casos es una buena aproximación a la realidad. Por ello, muchos procesadores de consulta hacen la suposición al elegir la estrategia. Por simplicidad, supondremos distribución uniforme.

  - La estimación del tamaño de un producto natural es algo más complicada que la estimación del tamaño de una selección o de un producto cartesiano. Sean r1(R1) y r2(R2) relaciones:

  - ✓ Si **R1 ∩ R2 =** ∅, entonces r1 ⋈ r2 es lo mismo que r1 x r2, y podemos usar la técnica de estimación para los productos cartesianos.
  - ✓ Si **R1 ∩ R2 es una clave de R1**, entonces sabemos que una tupla de r2 se va a unir con exactamente una tupla de r1. Por tanto, el número de tuplas en r1 ⋈ r2 no es mayor que el número de tuplas en r2.
  - ✓ El caso más difícil de considerar es cuando **R1 ∩ R2 no es clave ni de R1 ni de R2**. En este caso utilizamos la tercera estadística y suponemos que cada valor tiene la misma probabilidad de aparecer. Considérese una tupla t de r1, suponga que R1 ∩ R2 = {A}. Estimamos que la tupla t produce $\frac{n_{r_2}}{V(A, r_2)}$

  - tuplas en r1 ⋈ r2, ya que éste es el número de tuplas en r2 con un valor A de t[A].

- Considerando todas las tuplas en r1, estimamos que hay $\frac{n_{r_1} \cdot n_{r_2}}{V(A, r_2)}$ ¿
- tuplas en r1 ⋈ r2. Observe que si se invierten los papeles de r1 y r2 en esta estimación, obtenemos una estimación de $\frac{n_{r_1} \cdot n_{r_2}}{V(A, r_1)}$ ¿
- tuplas de r1 ⋈ r2. Las dos estimaciones difieren si V(A,r1) ≠ V(A,r2). Si se da esta situación, es probable que queden algunas tuplas colgantes que no participen en el producto. Así, es probable que la estimación más baja sea la mejor.
- Es posible que la estimación anterior del tamaño del producto sea demasiado alta si los valores de V(A,r1) para el atributo A en r1 tienen pocos valores en común con los valores de V(A,r2) para el atributo A en r2. Con todo, es poco probable que la estimación esté muy lejos del valor real en la práctica, ya que lo normal es que las tuplas colgantes sean sólo una pequeña parte de las tuplas de una relación del mundo real. Si las tuplas colgantes aparecen frecuentemente, pueden aplicarse un factor de corrección a las estimaciones.

- Si deseamos mantener estadísticas exactas, entonces cada vez que se modifique una relación debemos actualizar también las estadísticas. Esto requiere un tiempo adicional considerable. Por tanto, la mayor parte de los sistemas no actualizan las estadísticas en cada modificación. En cambio, se actualizan durante períodos de carga ligera del sistema. Como resultado, es posible que las estadísticas que se usan para elegir una estrategia de procesamiento de consultas no sean completamente exactas. Sin embargo, si el intervalo entre actualizaciones de las estadísticas no es demasiado largo, éstas serán lo suficientemente aproximadas para proporcionar una buena estimación del tamaño de los resultados de las expresiones.

Equivalencia de expresiones

Sea la consulta: Encontrar todos los clientes que tengan una cuenta en cualquier sucursal de Madryn

$$\pi_{\text{Id\_cliente}} (\sigma_{\text{ciudad\_sucursal='Madryn'}} (\text{sucursal} \bowtie (\text{cuenta} \bowtie \text{cli/cuen})))$$

Esta expresión construye una relación intermedia muy grande, sin embargo estamos interesados en una pocas tuplas de esta relación (aquellas que pertenecen a sucursales de Madryn) y sólo en uno de los seis atributos, id_cliente.

El resultado intermedio probablemente será demasiado grande ¿Hay una forma de reducir el tamaño del resultado intermedio?

$$\pi_{\text{Id\_cliente}} (\text{cli/cuen} \bowtie (\text{cuenta} \bowtie (\sigma_{\text{ciudad\_sucursal='Madryn'}} \text{sucursal})))$$

¿Hay alguna otra opción para reducir el tamaño de las relacines intermedias?

→ **Proyección**

Equivalencia de expresiones

Recordemos los esquemas con los que estamos trabajando:

sucursal=(nombre_sucursal, activo, ciudad_sucursal)

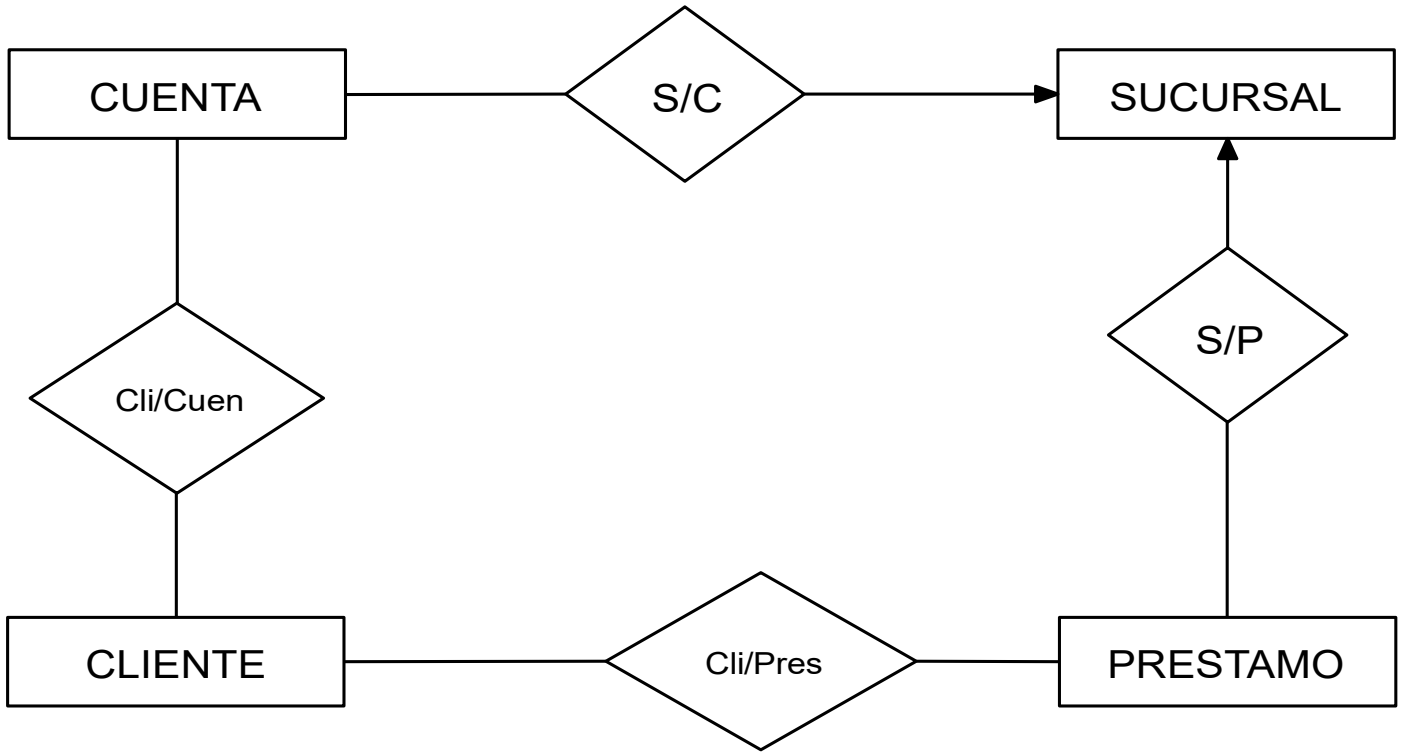
cliente=(id_cliente, nombre_cliente, calle_cliente, ciudad_cliente)

cuenta=(nombre_sucursal, número_cuenta, saldo)

préstamo=(nombre_sucursal, número_préstamo, importe)

cli/cuen=(id_cliente, número_cuenta)

cli/pres= (id_cliente, numero_prestamo)



- Estimación del costo de Procesamiento de Consultas

  - La estrategia que elijamos para una consulta dependerá del **tamaño de cada relación y de la distribución de los valores dentro de las columnas**.
  - Para poder elegir una estrategia basada en información fiable, los SGBD pueden almacenar estadísticas para cada relación r. Estas estadísticas incluyen:

    - ✓ n_r número de tuplas en la relación r.
    - ✓ s_r tamaño de una tupla de la relación r en bytes
    - ✓ V(A, r) número de valores distintos que aparecen en r para el atributo A.

  - Las dos primeras estadísticas nos permiten estimar con exactitud el tamaño de un producto cartesiano:
    - El producto cartesiano **r x s** contiene $\mathbf{n_r}$ y $\mathbf{n_s}$ tuplas.
    - Cada tupla **r x s** ocupa $\mathbf{s_r + s_s}$ bytes.

  - Para realizar una estimación, necesitamos saber cuántas veces aparece cada valor en una columna → suponemos que la distribución de valores es uniforme, entonces se estima que la consulta

    $\sigma_{A=a}$ (r) tiene (tuplas): $\dfrac{n_r}{V(A,r)}$

  - Sin embargo, puede que no sea realista la suposición de que cada valor aparezca con la misma probabilidad. Por simplicidad, supondremos distribución uniforme.

- Estimación del costo de Procesamiento de Consultas (Cont.)
  - La estimación del tamaño de un producto natural es algo más complicada que la estimación del tamaño de una selección o de un producto cartesiano. Sean r1(R1) y r2(R2) relaciones:

  ✓ Si **R1** $\cap$ **R2 =** $\varnothing$, entonces r1 $\bowtie$ r2 es lo mismo que r1 x r2, y podemos usar la técnica de estimación para los productos cartesianos. Sean las relaciones Préstamo y Cliente:

nombre_sucursal	nro_préstamo	importe
Centro	*P-170*	*600*
Sur	*P-230*	*800*
Norte	*P-260*	*340*

id_cliente	nombre_cliente	calle_cliente	ciudad_cliente
C-123	*López*	*San Martín*	*CRD*
C-230	*Sosa*	*Rivadavia*	*TRW*

  ✓ ¿Cómo quedaría Préstamo X Cliente?
  ✓ El producto cartesiano **r x s** contiene $\mathbf{n_r}$ y $\mathbf{n_s}$ tuplas → ¿6 tuplas?
  ✓ Cada tupla **r x s** ocupa $\mathbf{s_r + s_s}$ bytes.

- Estimación del costo de Procesamiento de Consultas (Cont.)
  - La estimación del tamaño de un producto natural es algo más complicada que la estimación del tamaño de una selección o de un producto cartesiano. Sean r1(R1) y r2(R2) relaciones:

  ✓ Si **R1 ∩ R2 es una clave de R1**, entonces sabemos que una tupla de r2 se va a unir con exactamente una tupla de r1. Por tanto, el número de tuplas en r1⋈r2 no es mayor que el número de tuplas en r2. Sean las relaciones Préstamo y Prestatario:

nombre_sucursal	nro_préstamo	importe
Centro	*P-170*	*600*
Sur	*P-230*	*800*
Norte	*P-260*	*340*

nombre_cliente	nro_préstamo
Santos	*P-170*
Gómez	*P-230*

nombre_sucursal	nro_préstamo	importe	nombre_cliente
Centro	*P-170*	*600*	*Santos*
Sur	*P-230*	*800*	*Gómez*

- Estimación del costo de Procesamiento de Consultas (Cont.)
  - La estimación del tamaño de un producto natural es algo más complicada que la estimación del tamaño de una selección o de un producto cartesiano. Sean r1(R1) y r2(R2) relaciones:

  ✓ El caso más difícil de considerar es cuando **R1 ∩ R2 no es clave ni de R1 ni de R2**. En este caso utilizamos la tercera estadística y suponemos que cada valor tiene la misma probabilidad de aparecer. Considérese una tupla t de r1, suponga que R1 ∩ R2 = {A}. Sean las relaciones Cliente y Empleado:

id_cliente	**nombre_cliente**	**calle**	**ciudad_sucursal**
C-123	*Lopez*	*San Martin*	*CRD*
C-230	*Sosa*	*Rivadavia*	*TRW*
C-212	*Herrera*	*Alem*	*MDY*
C-452	*Hernández*	*Sarmiento*	*CRD*

  ✓ Cliente ∩ Empleado = {calle}

leg_empleado	**calle**	**interno**
125	*San Martin*	*100*
456	*Rivadavia*	*110*
159	*Alem*	*111*

- Estimación del costo de Procesamiento de Consultas (Cont.)

(cliente $\bowtie_{\text{cliente.calle=empleado.calle}}$ empleado)

id_cliente	nombre_cliente	calle	ciudad_sucursal	leg_empleado	calle	interno
C-123	*Lopez*	*San Martin*	*CRD*	*125*	*San Martin*	*100*
C-230	*Sosa*	*Rivadavia*	*TRW*	*456*	*Rivadavia*	*110*
C-212	*Herrera*	*Alem*	*MDY*	*159*	*Alem*	*111*

- ✓ Estimamos que la tupla t produce $\frac{n_{r_2}}{V(A,r_2)}$

  tuplas en r1$\bowtie$r2, ya que éste es el número de tuplas en r2 con un valor A de t[A].
- – Considerando todas las tuplas en r1, estimamos que hay $\frac{n_{r_1} \cdot n_{r_2}}{V(A,r_2)}$ ¿

  tuplas en r1 $\bowtie$r2. Observe que si se invierten los papeles de r1 y r2 en esta estimación, obtenemos una estimación de $\frac{n_{r_1} \cdot n_{r_2}}{V(A,r_1)}$ ¿

  tuplas de r1$\bowtie$r2.
- – Las dos estimaciones difieren si V(A,r1) ≠ V(A,r2). Si se da esta situación, es probable que queden algunas tuplas colgantes que no participen en el producto. Así, es probable que la estimación más baja sea la mejor.

- Es posible que la estimación anterior del tamaño del producto sea demasiado alta si los valores de V(A,r1) para el atributo A en r1 tienen pocos valores en común con los valores de V(A,r2) para el atributo A en r2. Con todo, es poco probable que la estimación esté muy lejos del valor real en la práctica, ya que lo normal es que las tuplas colgantes sean sólo una pequeña parte de las tuplas de una relación del mundo real. Si las tuplas colgantes aparecen frecuentemente, pueden aplicarse un factor de corrección a las estimaciones.

- Si deseamos mantener estadísticas exactas, entonces cada vez que se modifique una relación debemos actualizar también las estadísticas. Esto requiere un tiempo adicional considerable. Por tanto, la mayor parte de los sistemas no actualizan las estadísticas en cada modificación. En cambio, se actualizan durante períodos de carga ligera del sistema. Como resultado, es posible que las estadísticas que se usan para elegir una estrategia de procesamiento de consultas no sean completamente exactas. Sin embargo, si el intervalo entre actualizaciones de las estadísticas no es demasiado largo, éstas serán lo suficientemente aproximadas para proporcionar una buena estimación del tamaño de los resultados de las expresiones.