

Laboratorio 11: Procesamiento de Consultas Distribuidas

Prof. Heider Sanchez

ACLs: Ana María Accilio, Sebastián Loza

P1. Descomposición 1

Elimine redundancias de la siguiente consulta:

```
SELECT CARGO
FROM EMP
WHERE EMP.NOMBRE = "Jorge"
      OR (NOT(EMP.CARGO = "Programador")
      AND (EMP.CARGO = "Programador" OR EMP.SUELDO < 1500)
      AND NOT(EMP.SUELDO < 1500))
```

P2. Descomposición 2

Expresé la siguiente consulta SQL en árbol de consulta algebraico optimizado:

```
SELECT NOMBRE
FROM EST, ACT, PROY
WHERE EST.COD = ACT.COD
      AND ACT.COD = PROY.COD
      AND EST.NOMBRE != "Jorge"
      AND EST.CARRERA = "CS"
      AND (ACT.HORAS = 4 OR ACT.HORAS = 6)
```

P3. Localización 1

Dada la siguiente consulta:

```
SELECT * FROM R WHERE E <= 18 AND A LIKE '%xyz%';
```

Fragmentación horizontal:

- [R1: $E < 10$]
- [R2: $10 \leq E < 20$]
- [R3: $20 \leq E < 35$]
- [R4: $E \geq 35$]

¿Cómo quedaría expresado el árbol de consulta? Reinscriba la consulta para cada servidor.

P4. Localización 2

Consulta:

```
SELECT EST.NOMBRE FROM EST, ACT
WHERE EST.COD = ACT.COD AND ACT.HORAS > 6;
```

Fragmentación:

- [EST1: COD < "201800000"]
- [EST2: COD ≥ "201800000"]
- [ACT1: ACT ⋈ EST1]
- [ACT2: ACT ⋈ EST2]

¿Cómo quedaría expresada la localización en árbol de consulta considerando los fragmentos?

P5. Localización 3

Consulta:

```
SELECT NOMBRE
FROM EST, ACT, PROY
WHERE EST.COD = ACT.COD
      AND ACT.COD = PROY.COD
      AND EST.NOMBRE != "Jorge"
      AND EST.CARGO = "CS"
      AND (ACT.HORAS = 4 OR ACT.HORAS = 6)
```

¿Cómo quedaría expresada la localización considerando la fragmentación en EST y ACT de la pregunta anterior?

P6. Optimización 1

Consulta:

```
SELECT * FROM EST ORDER BY COD;
```

Fragmentación:

- [EST1: hash(COD) % 3 == 0]
- [EST2: hash(COD) % 3 == 1]
- [EST3: hash(COD) % 3 == 2]

¿Cómo aplicaría la consulta de ordenación distribuida? Indique los pasos del algoritmo óptimo.

P7. Optimización 2

Consulta:

```
SELECT * FROM EST ORDER BY MENSUALIDAD WHERE NOM <= 'O' ;
```

Fragmentación:

- [EST1: NOM < "I"]
- [EST2: "I" ≤ NOM < "Q"]
- [EST3: COD ≥ "Q"]

¿Cómo aplicaría la consulta de ordenación distribuida? Grafique los pasos del algoritmo óptimo.

(Opcional) Proponga un algoritmo en Python para obtener una fragmentación intermedia óptima en Mensualidad, aplique el algoritmo con la siguiente información:

Fragmento	MIN mensualidad	MAX mensualidad	# tuplas
EST1	2k	7k	120
EST2	6k	18k	240
EST3	4k	9k	140

```
fragmentos = [  
    (2, 7, 120),  
    (6, 18, 240),  
    (4, 9, 140)  
]  
  
ks = encontrar_ks(fragmentos, 3) # 3 fragmentos -> 2 valores de k  
print(f"Puntos de corte: {ks}") # [6, 10]
```

P8. Optimización 3

Consulta:

```
SELECT CARRERA, COUNT(*), AVG(MENSUALIDAD) FROM EST GROUP BY CARRERA;
```

Fragmentación:

- [EST1: COD < "201800000"]
- [EST2: "201800000" ≤ COD < "202300000"]
- [EST3: COD ≥ "202300000"]

Indique los pasos para un procesamiento distribuido (incluyendo mejoras) de dicha consulta. Incluir también el SQL equivalente.

Entregable: en formato PDF.