

Ottimizzazione delle performance I

- Esercitazione in aula -

Siano date le seguenti query SQL

```
select O_ORDERSTATUS,N_NAME, count(O_ORDERKEY)
from   ORDERS,CUSTOMER,NATION,REGION
where  O_CUSTKEY=C_CUSTKEY and C_NATIONKEY = N_NATIONKEY
       and N_REGIONKEY = R_REGIONKEY and R_NAME= 'Europe'
group by O_ORDERSTATUS,N_NAME
```

```
select sum(L_QUANTITY)
from   LINEITEM,ORDERS,PART,CUSTOMER,NATION
where  L_ORDERKEY=O_ORDERKEY and O_CUSTKEY=C_CUSTKEY
       and C_NATIONKEY=N_NATIONKEY
       and L_PARTKEY=P_PARTKEY and N_NAME= 'Canada'
```

```
select DISTINCT(P_NAME)
from   PART,PARTSUPP,SUPPLIER
where  P_TYPE='PROMO PLATED STEEL' and S_ACCBAL>1000
       and P_PARTKEY=S_PARTKEY and PS_SUPPKEY=S_SUPPKEY;
```

Determinare l'albero di esecuzione ottimizzato rispetto ai criteri euristici presentati a lezione

Ottimizzazione delle performance

- Esercitazione in laboratorio -

Database TPCD_SRVORACLE

user=usersi password=usersi

1) Si disegni l'albero di esecuzione dell'ottimizzatore per le seguenti interrogazioni:

```
select s_name, s_address
from   supplier, nation, region
where  s_nationkey = n_nationkey
       and n_regionkey = r_regionkey and r_name='Europe';
```

```
select count(distinct p_brand) from part;
```

```
select * from PART ORDER BY P_NAME;
```

```
select * from PART ORDER BY P_PARTKEY;
```

2) Dopo aver disegnato l'albero di esecuzione dell'ottimizzatore per l'interrogazione:

```
select o_clerk, p_type, sum(l_quantity), avg(l_discount)
from   lineitem, orders, part, supplier, nation
where  l_partkey = p_partkey and l_orderkey = o_orderkey
       and l_suppkey = s_suppkey
       and s_nationkey = n_nationkey
       and o_clerk = 'Clerk#000000955'
       and n_name = 'KENYA'
group by o_clerk, p_type;
```

si verifichi:

- Come e perchè varia l'albero di esecuzione rimuovendo la condizione su o_clerk
- Come e perchè varia l'albero di esecuzione rilassando la condizione su o_clerk a o_clerk > 'Clerk#000000955'

3) Data la seguente interrogazione:

```
select n_name, s_name, sum(ps_supplycost)
from partsupp, part, supplier, nation
where n_nationkey = s_nationkey
      and s_suppkey = ps_suppkey
      and p_partkey = ps_partkey
      and p_name > 'saddle metallic grey smoke sky'
group by n_name, s_name;
```

- ottimizzare l'albero di esecuzione in base alle regole euristiche viste a lezione e confrontarlo con quello proposto dall'ottimizzatore
- verificare come varia il piano di esecuzione (e il relativo costo) imponendo
 - un diverso ordine di esecuzione dei join
 - diversi algoritmi di join
 - evitando l'utilizzo di indici

4) Sia data la seguente query SQL

```
select sum(L_QUANTITY)
from LINEITEM, ORDERS, PART, CUSTOMER, NATION
where L_ORDERKEY=O_ORDERKEY and O_CUSTKEY=C_CUSTKEY
      and C_NATIONKEY=N_NATIONKEY
      and N_NAME= 'BRAZIL'
```

Ottimizzare l'albero di esecuzione in base alle regole euristiche viste a lezione e confrontarlo con quello proposto dall'ottimizzatore

5) Sia data la seguente query SQL

```
select O_ORDERKEY, O_TOTALPRICE
from ORDERS
where O_ORDERPRIORITY= '0' and O_ORDERSTATUS= '5-LOW';
```

Disegnare l'albero di esecuzione, ottimizzarlo mediante l'approccio euristico ed effettuando l'effettivo calcolo dei costi:

Calcolare il miglior piano di accesso ai dati sapendo che:

$NT_{Orders}=1.500.000$
 $Len(Orders)=115$ byte
 $D = 4$ KB
 $u = 1.0$
 $NK_{O_ORDERPRIORITY} = 5$
 $NK_{O_ORDERSTATUS} = 3$
 $Len(O_ORDERPRIORITY) = 15$
 $Len(O_ORDERSTATUS) = 1$

Si hanno a disposizione due indici bitmap unclustered su `O_ORDERPRIORITY` e su `O_ORDERSTATUS` (con altezza $h = 2$)

6) Sia data la seguente query SQL

```
select  P_PARTKEY, P_NAME, sum(L_QUANTITY)
from    LINEITEM, PART
where   P_PARTKEY=L_PARTKEY and P_MFGR='Manufacturer#4'
        and L_EXTENDEDPRICE > 50.000
group by P_PARTKEY, P_NAME;
```

Disegnare l'albero di esecuzione, e confrontarlo con il piano dei costi ottimizzato mediante l'approccio euristico. come cambia il piano di esecuzione rimuovendo la clausola su P_MFGR? Perché?

Si effettui inoltre l'effettivo calcolo dei costi sapendo che:

$NT_{Part}=200.000$ $NT_{LINEITEM}= 6.012.150$
 $Len(Part)=133$ byte $Len(Lineitem)=113$ byte
 $Len(P_NAME) = 32$ $Len(P_PARTKEY) = 4$

$D = 4$ KB

$ut = 1.0$

$NK_{P_MFGR} = 5$

$NK_{L_EXTENDEDPRICE} = 934.059$

$Min(L_EXTENDEDPRICE) = 901$ - $Max(L_EXTENDEDPRICE)=104.949$

$NB = 101$ (per l'ordinamento)

Sono a disposizione gli indici sulle chiavi primarie e un indice su L_PARTKEY:

$h_{L_ORDERKEY,L_LINENUMBER}=2$ $h_{L_PARTKEY}=2$ $h_{P_PARTKEY}=2$

$len(L_PARTKEY)=Len(P_PARTKEY) =Len(Pointer) = 4$ byte

$ui(L_PARTKEY) = 0.69$