

# LOGIN

software

# Alen Prodan

## Korištenje histograma u optimizaciji SQL naredbi

# Agenda

- ▶ Uloga histograma
- ▶ Frequency histogram
- ▶ Height Balanced histogram
- ▶ Popularne, nepopularne vrijednosti i New Density vrijednost
- ▶ Specifične primjene histograma
- ▶ Generiranje vlastitog Frequency histograma

# Definicija histograma

## Što je histogram ?

- ▶ U statistici, histogram je grafički prikaz kojim se vizualizira distribucija podataka (<http://en.wikipedia.org/wiki/Histogram>)



# Histogrami

## Uloga histograma u Oracle RDBMS

- ▶ Oracle koristi dvije vrste histograma: Frequency i Height Based
- ▶ Histogramima se preciznije opisuje raspodijela vrijednosti unutar kolona tablice
- ▶ U odsutnosti histograma optimizier podrazumijeva ravnomjernu raspodijelu, pa se selektivnost “where kolona = konstanta” izračunava kao  $1/\text{num\_distinct}$
- ▶ Precizna procjena selektivnost bitna je za:
  - ▶ Odabir access pathova (table/index)
  - ▶ Određivanje join mehanizma
  - ▶ Određivanje redoslijeda povezivanja (join order)

# Frequency histogrami

## Osnovne značajke Frequency histograma

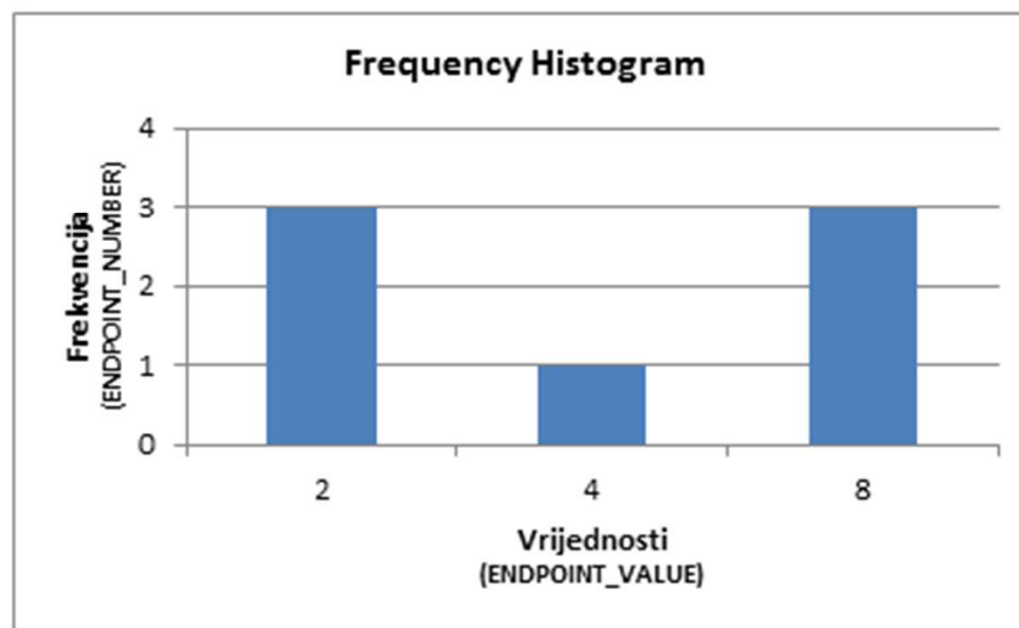
- ▶ U stupcu tablice mora biti  $\leq 254$  distinct vrijednosti
- ▶ Generira se pomoću `DBMS_STATS.gather_table_stats(method_opt)`, `DBMS_STATS.gather_schema_stats(method_opt)` poziva
- ▶ FH egzaktno bilježi frekvenciju pojave svake distinct vrijednosti
- ▶ Visoka preciznost kojom se opisuje raspodijela vrijednosti
- ▶ FH zapis se ne pohranjuje u obliku [vrijednost, frekvencija], već u obliku running totala

# Frequency histogrami

## Primjer: Frequency histogram sa 3 distinct vrijednosti

- ▶ Frequency histogram nad stupcem sa 3 distinct vrijednosti

| VRIJEDNOST | ENDPOINT_VALUE | ENDPOINT_NUMBER | RAZLIKA     |
|------------|----------------|-----------------|-------------|
| 2          | 2              | 3               | $(3-0) = 3$ |
| 4          | 4              | 4               | $(4-3) = 1$ |
| 8          | 8              | 7               | $(7-4) = 3$ |



# Height Balanced histogrami

## Osnovne značajke Height Balanced histograma

- ▶ Broj različitih distinct vrijednosti unutar pojedine kolone je neograničen
- ▶ Mora postojati ograničenje na maksimalni broj točaka koje opisuju histogram zbog brzine parsanja i optimizacije SQL, te zbog pohrane podataka o histogramima u sistemskom tablespaceu
- ▶ Navedeni kompromis pretpostavlja smanjenje preciznosti histograma
- ▶ Generira se pomoću `DBMS_STATS.gather_table_stats(method_opt)`, `DBMS_STATS.gather_schema_stats(method_opt)` poziva
- ▶ FH egzaktno bilježi frekvenciju, HB se generira uzorkovanjem
- ▶ Uzorkovanje se provodi na način da se vrijednosti unutar kolone prvo sortiraju, a zatim podijele u jednake intervale (buckets)
- ▶ Broj intervala (buckets) određen je veličinom "SIZE N" `method_opt` parametra procedura `DBMS_STATS` paketa



# Height Balanced histogrami

## Primjer: HB histogram bez popularnih vrijednosti

- ▶ HB histogram, nad kolonom tablice sa 9 redaka i 9 distinct vrijednosti
- ▶ Histogram generiran sa  $N = 3$  intervala (bucketa)

|                 |             |   |             |   |             |   |   |   |   |
|-----------------|-------------|---|-------------|---|-------------|---|---|---|---|
| ENDPOINT_VALUE  | 1           | 3 | 6           | 9 |             |   |   |   |   |
| VRIJEDNOST      | 1           | 2 | 3           | 4 | 5           | 6 | 7 | 8 | 9 |
| ENDPOINT_NUMBER | 0           | 1 | 2           | 3 |             |   |   |   |   |
| RAZLIKA         | $(1-0) = 1$ |   | $(2-1) = 1$ |   | $(3-2) = 1$ |   |   |   |   |

# Height Balanced histogrami

## Primjer: HB histogram sa popularnom vrijednosti

- ▶ HB histogram sa popularnom vrijednosti unutar 2 uzastopna intervala
- ▶ Redak RAZLIKA označava da li se neka vrijednosti pojavljuje uzastopno i time predstavlja **popularnu vrijednost**

|                 |             |   |   |             |   |   |   |   |
|-----------------|-------------|---|---|-------------|---|---|---|---|
| ENDPOINT_VALUE  | 1           | 3 |   | 9           | 9 |   | 9 |   |
| VRIJEDNOST      | 1           | 2 | 3 | 4           | 9 | 9 | 9 | 9 |
| ENDPOINT_NUMBER | 0           | 1 |   |             | 2 |   | 3 |   |
| RAZLIKA         | $(1-0) = 1$ |   |   | $(3-1) = 2$ |   |   |   |   |

# Height Balanced histogrami

## Popularnost i uloga New Density formule

- ▶ Kod HB histograma popularne su one vrijednosti za koje vrijedi da je (tekući `ENDPOINT_NUMBER` – prethodni `ENDPOINT_NUMBER`) > 1
- ▶ Popularne vrijednosti imaju značajne statističke razlike u odnosu na nepopularne vrijednosti
- ▶ Za popularne vrijednosti frekvencija pojavnosti se određuje na temelju broja intervala u kojima je vrijednost uzorkovana
- ▶ Za nepopularne vrijednosti to nije moguće

# Heigh Balanced histogrami

## Popularnost i uloga New Density formule

- ▶ Prikazano je 6 mogućih scenarija pojave nepopularne vrijednosti za tablicu sa 9 redaka i  $N = 3$  intervala
- ▶ Frekvencija pojavnosti nepopularnih vrijednosti se može kretati u intervalu od 1 do  $2 * N - 1$ , a bez da se takva vrijednost detektira kao popularna u HB histogramu

|          | Intervali (Buckets) za $N = 3$ |  |   |  |  |   |  |  |   |
|----------|--------------------------------|--|---|--|--|---|--|--|---|
| Scenario |                                |  |   |  |  |   |  |  |   |
| 1        |                                |  |   |  |  |   |  |  |   |
| 2        |                                |  |   |  |  |   |  |  |   |
| 3        |                                |  |   |  |  |   |  |  |   |
| 4        |                                |  |   |  |  |   |  |  |   |
| 5        |                                |  |   |  |  |   |  |  |   |
| 6        |                                |  |   |  |  |   |  |  |   |
|          | 0                              |  | 1 |  |  | 2 |  |  | 3 |

# Height Balanced histogramami

## Popularnost i uloga New Density formule

- ▶ Zbog prethodno opisanog problema, za izračun frekvencije pojavnosti nepopularnih vrijednosti koristi se druga statistička veličina: `density`
- ▶ Od verzije 11g, koristi se nova density vrijednost: `NewDensity`
- ▶ Korištenje `NewDensity` vrijednosti regulirano je skrivenim inicijalizacijskim parametrom `_optimizer_enable_density_improvements`
- ▶ `NewDensity` se ne pohranjuje u data dictionary, već je vidljiva u 10053 trace datoteci
- ▶ `NewDensity` se koristi za izračun selektivnosti SQL naredbi koje referenciraju nepopularne vrijednosti

## Height Balanced histogrami

### Što predstavlja NewDensity formula ?

- ▶ Kada CBO ne raspolaže informacijom o raspodijeli vrijednosti, pretpostavlja ravnomjernu distribuciju
- ▶ Kod HB histograma, konceptualno je moguće podijeliti tablicu na dva podskupa: popularni i nepopularni
- ▶ Za upite nad vrijednostima iz popularnog skupa koristi se histogram
- ▶ Za nepopularne vrijednosti distribucija je nepoznata, pa je NewDensity aproksimacija prosječne frekvencije pojavnosti svake pojedine vrijednosti koja nije prikazan u histogramu kao popularna vrijednost

# Height Balanced histogrami

## Primjer: Izračun i primjena NewDensity vrijednosti

```
select n1, count(*)
from t3
group by n1
order by n1;
```

| N1 | COUNT(*) |
|----|----------|
| 1  | 4        |
| 5  | 6        |
| 6  | 1        |
| 7  | 1        |
| 8  | 11       |
| 9  | 1        |
| 10 | 1        |

Nad stupcem N1 generiran je HB histogram sa 5 intervala (bucketa).

| ENDPOINT_VALUE | ENDPOINT_NUMBER | EP <sub>n</sub> - EP <sub>n-1</sub> |
|----------------|-----------------|-------------------------------------|
| 1              | 0               | 0                                   |
| 5              | 2               | 2                                   |
| 8              | 4               | 2                                   |
| 10             | 5               | 1                                   |

Distribuciju vrijednosti, alternativno možemo prikazati i na sljedeći način:

|    |   |   |   |   |   |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |    |
|----|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|
| EV | 1 | 5 | 5 | 8 | 8 | 10 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |    |
| V  | 1 | 1 | 1 | 1 | 5 | 5  | 5 | 5 | 5 | 6 | 7 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 9 | 10 |
| EN | 0 | 1 | 2 | 3 | 4 | 5  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |    |

EV = ENDPOINT\_VALUE  
V = VRIJEDNOST  
EN = ENDPOINT\_NUMBER  
R = RAZLIKA

# Height Balanced histogramami

## Primjer: Izračun i primjena NewDensity vrijednosti

```
select n1
from t3
where n1 = 1;
```

SINGLE TABLE ACCESS PATH

Single Table Cardinality Estimation for T3[T3]

Column (#1):

NewDensity:0.040000, OldDensity:0.100000 BktCnt:5, PopBktCnt:4, PopValCnt:2, NDV:7

gdje je:

|            |   |
|------------|---|
| OldDensity | stara density vrijednost pohranjena u data dictionary         |
| BktCnt     | broj intervala (bucketa)                                      |
| PopBktCnt  | broj popularnih intervala                                     |
| PopValCnt  | broj popularnih vrijednosti                                   |
| NDV        | broj različitih vrijednosti (NDV = number of distinct values) |

NewDensity se izračunava koristeći sljedeću formulu

$$\text{NewDensity} = ((\text{BktCnt} - \text{PopBktCnt}) / \text{BktCnt}) / (\text{NDV} - \text{PopValCnt})$$

odnosno, ako uvrstimo gornje vrijednosti dobivamo:

$$\text{NewDensity} = ((5 - 4) / 5) / (7 - 2) = 0.2 / 5 = 0.04$$



# Height Balanced histogrami

## Stabilnost Height Balanced histograma

- ▶ Ponekad čak i male promjene kao dodavanje, brisanje ili ažuriranje retka, mogu uzrokovati nestajanje nepopularnih vrijednosti iz histograma
- ▶ Popularne vrijednosti su otpornije na taj fenomen, no češći je slučaj da postanu nepopularne

Ažuriramo li samo jedan redak koji ima vrijednosti 9 u vrijednosti 10:

|                 |           |   |   |   |           |   |   |   |   |
|-----------------|-----------|---|---|---|-----------|---|---|---|---|
| ENDPOINT_VALUE  | 1         | 3 |   | 9 | 9         |   |   |   | 9 |
| VRIJEDNOST      | 1         | 2 | 3 | 4 | 9         | 9 | 9 | 9 | 9 |
| ENDPOINT_NUMBER | 0         | 1 |   |   | 2         |   |   |   | 3 |
| RAZLIKA         | (1-0) = 1 |   |   |   | (3-1) = 2 |   |   |   |   |

vrijednost 9 postaje nepopularna vrijednosti:

|                 |           |   |   |           |           |   |   |   |    |
|-----------------|-----------|---|---|-----------|-----------|---|---|---|----|
| ENDPOINT_VALUE  | 1         | 3 |   | 9         | 10        |   |   |   |    |
| VRIJEDNOST      | 1         | 2 | 3 | 4         | 9         | 9 | 9 | 9 | 10 |
| ENDPOINT_NUMBER | 0         | 1 |   | 2         |           |   |   |   | 3  |
| RAZLIKA         | (1-0) = 1 |   |   | (2-1) = 1 | (3-2) = 1 |   |   |   |    |

# Specifične primjene histograma

## Primjer 1: Korištenje neprimjerenih tipova podataka

- ▶ Primjena histograma nije nužno ograničena isključivo na slučajeve sa neravnomjernom raspodjelom vrijednosti
- ▶ Histogrami pomažu u optimizaciji SQL naredbi za slučajeve gdje se koristi number tip podataka za pohranu datumskih vrijednosti
- ▶ Pretpostavimo tablicu sa 32 retka, za interval od 01.12.10 do 01.01.11
- ▶ Tablica sadrži 2 stupca, u oba su pohranjene iste datumske vrijednosti, ali koristeći dva različita tipa podatka: D – date tip, N – number tip

```

create table t
as
select to_date('01.12.2010', 'dd.mm.yyyy')+rownum-1 d,
       to_number(to_char(to_date('01.12.2010', 'dd.mm.yyyy')+rownum-1, 'yyyymmdd')) n
from dual
connect by level <= 32;

exec dbms_stats.gather_table_stats(user, 'T', method_opt=>'FOR ALL COLUMNS SIZE 1' );

```

# Specifične primjene histograma

## Primjer 1: Korištenje neprimjerenih tipova podataka

- ▶ SQL upitom selektirati ćemo retke iz tablice za razdoblje 31.12.2010 do 01.01.2011, prvo za stupac D – nativni date tip
- ▶ Očekivano, upit vraća dva retka, a procjena selektivnosti je vrlo točna

```
select d, n
from t_dtype
where d between to_date('31.12.2010', 'dd.mm.yyyy') and to_date('01.01.2011', 'dd.mm.yyyy')
```

```
D                N
-----
31.12.10    20101231
01.01.11    20110101
```

### Execution Plan

Plan hash value: 696206505

| Id  | Operation         | Name    | Rows | Bytes | Cost (%CPU) | Time     |
|-----|-------------------|---------|------|-------|-------------|----------|
| 0   | SELECT STATEMENT  |         | 2    | 28    | 2 (0)       | 00:00:01 |
| * 1 | TABLE ACCESS FULL | T_DTYPE | 2    | 28    | 2 (0)       | 00:00:01 |

# Specifične primjene histograma

## Primjer 1: Korištenje neprimjerenih tipova podataka

- ▶ Ponovimo li isti upit sa filterom nad N stupcem, upit će vratiti ista dva retka, ali uz krivu procjenu selektivnosti

```
select d, n
from t_dtype
where n between 20101231 and 20110101
```

```
D                N
-----
31.12.10    20101231
01.01.11    20110101
```

Execution Plan

Plan hash value: 696206505

| Id  | Operation         | Name    | Rows | Bytes | Cost (%CPU) | Time     |
|-----|-------------------|---------|------|-------|-------------|----------|
| 0   | SELECT STATEMENT  |         | 32   | 448   | 2 (0)       | 00:00:01 |
| * 1 | TABLE ACCESS FULL | T_DTYPE | 32   | 448   | 2 (0)       | 00:00:01 |

# Specifične primjene histograma

## Primjer 1: Korištenje neprimjerenih tipova podataka

- ▶ Zašto dolazi do pogrešne selektivnosti ?
- ▶ Zbog pogrešnog korištenja number tipa za pohranu datumskih vrijednosti, Oracle ne razumije da su 20121231 i 20110101 u kalendarskom smislu dvije uzastopne vrijednosti
- ▶ Traženi rang vrijednosti iznosi: 20110101-20101231 = 8870
- ▶ Ukupni rang vrijednosti iznosi: 20110101- 20101201 = 8890
- ▶ Parcijalnim izračunom vidimo da je selektivnost gotovo 100%:  
 $8870 / 8890 = 99.6\%$ , a to u konkretnom slučaju iznosi 32 retka

# Specifične primjene histograma

## Primjer 1: Korištenje neprimjerenih tipova podataka

- ▶ Kako ublažiti posljedice ?
- ▶ Generiranjem HB histograma sa 20 intervala nad N stupcem
- ▶ Ponovimo li isti SQL, selektivnost je ponovno precizna
- ▶ Uz pomoć histograma optimizer zna da svaki interval sadrži  $32 / 20 = 1.6$  redaka

```

D                N
-----
31.12.10      20101231
01.01.11      20110101
    
```

Execution Plan

Plan hash value: 696206505

| Id  | Operation         | Name    | Rows | Bytes | Cost (%CPU) | Time     |
|-----|-------------------|---------|------|-------|-------------|----------|
| 0   | SELECT STATEMENT  |         | 2    | 28    | 2 (0)       | 00:00:01 |
| * 1 | TABLE ACCESS FULL | T_DTYPE | 2    | 28    | 2 (0)       | 00:00:01 |

# Specifične primjene histograma

## Primjer 2: Nizovi sa ekstremnim vrijednostima

- ▶ Pretpostavimo tablicu sa 365 redaka, 1 redak za svaki dan u 2010.god.
- ▶ Pretpostavimo da obrađene retke aplikacija ažurira na 01.01.1970 god.
- ▶ Kreiramo tablicu i generiramo statistike, bez histograma

```

create table t_eks
as
select to_date('01.01.2010', 'dd.mm.yyyy')+rownum-1 d
from dual
connect by level <=365;

exec dbms_stats.gather_table_stats(user, 'T_EKS', method_opt=>'FOR ALL COLUMNS SIZE 1' );

```

# Specifične primjene histograma

## Primjer 2: Nizovi sa ekstremnim vrijednostima

- ▶ Sljedećim SQL upitom selektiramo podatke za 12/2010
- ▶ Upit donosi 31 redak, procjena selektivnosti vrlo precizna

```
select count(*)
from t_eks
where d between
to_date('01.12.2010' , 'dd.mm.yyyy') and to_date('31.12.2010' , 'dd.mm.yyyy')
```

```

COUNT(*)
-----
          31
```

### Execution Plan

Plan hash value: 2718855673

| Id  | Operation         | Name  | Rows | Bytes | Cost (%CPU) | Time     |
|-----|-------------------|-------|------|-------|-------------|----------|
| 0   | SELECT STATEMENT  |       | 1    | 8     | 2 (0)       | 00:00:01 |
| 1   | SORT AGGREGATE    |       | 1    | 8     |             |          |
| * 2 | TABLE ACCESS FULL | T_EKS | 31   | 248   | 2 (0)       | 00:00:01 |



# Specifične primjene histograma

## Primjer 2: Nizovi sa ekstremnim vrijednostima

- ▶ Jedan redak u tablici će biti ažuriran na 01.01.1970, nakon čega će se prikupiti statistike, ponovno bez histograma
- ▶ Ponovnim izvođenje testnog SQL upita, dobivamo točan rezultat, ali uz pogrešnu procjenu selektivnosti

```
update t_eks set d = to_date('01.01.1970', 'dd.mm.yyyy') where rownum = 1;
commit;
exec dbms_stats.gather_table_stats(user, 'T_EKS', method_opt=>'FOR COLUMNS D SIZE 1');
```

```

COUNT(*)
-----
          31

```

### Execution Plan

Plan hash value: 2718855673

| Id  | Operation         | Name  | Rows | Bytes | Cost (%CPU) | Time     |
|-----|-------------------|-------|------|-------|-------------|----------|
| 0   | SELECT STATEMENT  |       | 1    | 8     | 2 (0)       | 00:00:01 |
| 1   | SORT AGGREGATE    |       | 1    | 8     |             |          |
| * 2 | TABLE ACCESS FULL | T_EKS | 2    | 16    | 2 (0)       | 00:00:01 |

# Specifične primjene histograma

## Primjer 2: Nizovi sa ekstremnim vrijednostima

- ▶ Što je uzrok problema ?
- ▶ Oracle izračunava selektivnost “between” ranga vrijednosti kao:  
 $\text{željeni\_rang} / \text{ukupni\_rang} + 2 / \text{num\_distinct}$
- ▶ U prvom slučaju to iznosi:
- ▶  $(31.12.2010 - 01.12.2010) / (31.12.2010 - 01.01.2010) + 2 / 365$   
 $= 30 / 364 + 2/365 = 0.0879 * 365 = 32$
- ▶ U drugom slučaju zbog promjene MIN vrijednosti, mijenja se ukupni raspon vrijednosti:
- ▶  $(31.12.2010 - 01.12.2010) / (31.12.2010 - 01.01.1970) + 2 / 365$   
 $= 30 / 14974 + 2/365 = 0.00748 * 365 = 2$

# Specifične primjene histograma

## Primjer 2: Nizovi sa ekstremnim vrijednostima

- ▶ Rješenje ?
- ▶ Kreiranje HB histograma, sa barem 2 intervala. Zbog histograma, interval se više ne aplicira na cijeli rang vrijednosti u tablici, već na MIN/MAX vrijednosti intervala koji pokriva traženi rang vrijednosti

| ENDPOINT | ENDPOINT_NUMBER | WIDTH | HEIGHT |
|----------|-----------------|-------|--------|
| 01.01.70 | 0               |       |        |
| 02.07.10 | 1               | 14792 | ,0022  |
| 31.12.10 | 2               | 182   | ,1758  |

- ▶ Uz prisustvo HB histograma sa 2 intervala, selektivnost se izračunava na sljedeći način:

$$\begin{aligned}
 & (31.12.2010 - 01.12.2010) / (31.12.2010 - 02.07.2010) + \\
 & 2 * \text{density} \\
 & = 30/182 + 2*0.002739726 = 0.164835*182 = 31
 \end{aligned}$$

# Specifične primjene histograma

## Primjer 2: Nizovi sa ekstremnim vrijednostima

- ▶ Ponovimo li testni SQL upit, možemo vidjeti da je uz pomoć histograma ponovno točno procjenjena selektivnost

```
select count(*)
from t_eks
where d between
to_date('01.12.2010' , 'dd.mm.yyyy') and to_date('31.12.2010' , 'dd.mm.yyyy')
```

```

COUNT(*)
-----
          31
```

Execution Plan

-----  
Plan hash value: 2718855673

| Id  | Operation         | Name  | Rows | Bytes | Cost (%CPU) | Time     |
|-----|-------------------|-------|------|-------|-------------|----------|
| 0   | SELECT STATEMENT  |       | 1    | 8     | 2 (0)       | 00:00:01 |
| 1   | SORT AGGREGATE    |       | 1    | 8     |             |          |
| * 2 | TABLE ACCESS FULL | T_EKS | 30   | 240   | 2 (0)       | 00:00:01 |

## Izrada vlastitog FH histograma

### Zašto koristiti vlastiti FH histogram ?

- ▶ Veće tablice nerijetko sadržavaju više od 254 distinct vrijednosti u stupcima, pa za njih nije moguće generirati precizan FH histogram
- ▶ Preostaje generiranje HB histograma, ali uz gubitak preciznosti kojom se opisuje raspodijela vrijednosti u tablici
- ▶ Mogući su slučajevi u kojima retke sa visokom frekvencijom pojavnosti Oracle ne detektira kao popularne
- ▶ Ukoliko su baš te vrijednosti važne za korisnike sustava, potrebno je osigurati dobru procjenu selektivnosti, kao temelj dobrih planova izvršavanja

# Izrada vlastitog FH histograma

## Primjer izrade vlastitog FH histograma

- Generiramo tablicu sa 100.000 redaka i 100.000 distinct vrijednosti

```
create table t4
as
select
rownum n1
from dual
connect by level <= 100000;
```

Ciljano uvodimo 5 popularnih vrijednosti sa visokom frekvencijom ponavljanja (100,200,300,400 i 500):

```
insert into t4 select 100 from dual connect by level <= 100; -- ponavljanje 100 puta
insert into t4 select 200 from dual connect by level <= 200; -- ponavljanje 200 puta
insert into t4 select 300 from dual connect by level <= 300; -- ponavljanje 300 puta
insert into t4 select 400 from dual connect by level <= 400; -- ponavljanje 400 puta
insert into t4 select 500 from dual connect by level <= 500; -- ponavljanje 500 puta
```

Na kraju, insertirati ćemo još 100 redaka sa vrijednostima od 1 do 100 kako bi dodatno otežali izradu Height balanced histograma:

```
insert into t4 select rownum from dual connect by level <= 100;
```

# Izrada vlastitog FH histograma

## Primjer izrade vlastitog FH histograma

- Prikupljamo statistike i generiramo HB histogram sa 254 intervala, međutim niti jedna od popularnih vrijednosti nije detektirana u HB histogramu

```
begin
  dbms_stats.gather_table_stats(
    ownname=>user,
    tabname=>'T4',
    method_opt=>'FOR COLUMNS N1 SIZE 254'
  );
end;
```

| ENDPOINT_VALUE | ENDPOINT_NUMBER |
|----------------|-----------------|
| 2              | 0               |
| 100            | 1               |
| 300            | 2               |
| 384            | 3               |
| 486            | 4               |
| 500            | 5               |
| 935            | 6               |
| 1284           | 7               |

# Izrada vlastitog FH histograma

## Primjer izrade vlastitog FH histograma

- ▶ Upitom iz tablice možemo potvrditi da popularne vrijednosti nisu prepoznate, pa je samim time i selektivnost pogrešno izračunata

```
select n1 from t4 where n1 = 100;
```

| Id  | Operation         | Name | Rows | Bytes | Cost (%CPU) | Time     |
|-----|-------------------|------|------|-------|-------------|----------|
| 0   | SELECT STATEMENT  |      | 1    | 5     | 45 (3)      | 00:00:01 |
| * 1 | TABLE ACCESS FULL | T4   | 1    | 5     | 45 (3)      | 00:00:01 |



# Izrada vlastitog FH histograma

```

declare
    l_statrec DBMS_STATS.statrec;
    l_val_array DBMS_STATS.numarray;
    l_distcnt number;
    l_density number;
    l_nullcnt number;
    l_avgclen number;

begin
    dbms_stats.get_column_stats(
        ownname=>user,
        tabname=>'T4',
        colname=>'N1',
        distcnt=> l_distcnt,
        density=> l_density,
        nullcnt=> l_nullcnt,
        srec => l_statrec,
        avgclen => l_avgclen
    );

    select n1, c
    bulk collect into l_val_array, l_statrec.bkvals
    from
    (
        select n1, c
        from
        (
            select n1, count(*) c
            from t4
            group by n1
            order by c desc
        )
        where rownum <=253
    )
    union all
    select 10000000, 98246
    from dual
    order by 1;

```

```

l_statrec.epc := l_val_array.count;
dbms_output.put_line('Count: '||l_val_array.count);

dbms_stats.prepare_column_values(
    srec => l_statrec,
    numvals=>l_val_array
);

select 1/(2*count(*)) into l_density from t4;

dbms_stats.set_column_stats(
    ownname=>user,
    tabname=>'T4',
    colname=>'N1',
    distcnt=> l_distcnt,
    density => l_density,
    nullcnt => l_nullcnt,
    srec => l_statrec,
    avgclen => l_avgclen
);

END;

```

# Izrada vlastitog FH histograma

## Primjer izrade vlastitog FH histograma

- ▶ Analizirajući generirani FH histogram, vidimo da su ovog puta popularne vrijednosti zabilježene kao takve, što je vidljivo iz USER\_TAB\_HISTOGRAMS viewa

| ENDPOINT_VALUE | ENDPOINT_NUMBER |
|----------------|-----------------|
| ...            | ...             |
| 98             | 196             |
| 99             | 198             |
| 100            | 300             |
| 123            | 301             |
| 200            | 502             |
| 300            | 803             |
| 400            | 1204            |
| 500            | 1705            |
| ...            | ....            |

```
select n1 from t4 where n1 = 100
```

Execution Plan

Plan hash value: 2560505625

| Id  | Operation         | Name | Rows | Bytes | Cost (%CPU) | Time     |
|-----|-------------------|------|------|-------|-------------|----------|
| 0   | SELECT STATEMENT  |      | 104  | 520   | 45 (3)      | 00:00:01 |
| * 1 | TABLE ACCESS FULL | T4   | 104  | 520   | 45 (3)      | 00:00:01 |

## Zaključak

- ▶ Frequency histogrami su precizniji od Height balanced histograma i predstavljaju preporučenu opciju kada je  $NDV \leq 254$
- ▶ Kod Height balanced histograma tablicu je moguće promatrati kao dva podskupa: popularne i nepopularne vrijednosti
- ▶ Selektivnost popularnih vrijednosti određuje se pomoću HB histograma, dok za nepopularne koristimo NewDensity
- ▶ Primjena histograma nije ograničena isključivo na neravnomjerne raspodjele vrijednosti, već i na rješavanje problema selektivnosti u određenim specijalnim slučajevima
- ▶ U slučaju da je HB histogram nedovoljno precizan, moguće je izraditi vlastiti Frequency histogram za ciljani skup popularnih vrijednosti

Pitanja

i

Odgovori

