

Strojenie poleceń SQL – zadania

Wstęp

1. Utwórz potrzebne do ćwiczeń tabele i wypełnij je danymi. Użyj do tego celu skryptu `opt.sql`.

Utwórz relację `PLAN_TABLE`, wykonując skrypt `utlxplan.sql`

2. Wyjaśnij plan poniższego zapytania, używając polecenia `EXPLAIN_PLAN`, nadaj wygenerowanemu planowi identyfikator „p1”:

```
select nazwa, count(*) from zesp natural join prac
group by nazwa
order by count(*) desc;
```

Następnie odczytaj plan zapytania, używając:

- zapytania:

```
select operation,object_name,id,cost,parent_id
from plan_table where statement_id='p1' order by id;
```
- skryptu `utlxpls.sql`

3. Włącz dyrektywę `SET AUTOTRACE ON EXPLAIN` i pomiar czasu dyrektywą `SET TIMING ON`. Sprawdź ponownie plan wykonania zapytania z punktu 2., wykorzystując działanie zastosowanych dyrektyw.

Metody dostępu

4. Wyjaśnij plan poniższego zapytania. Jaką metodą dostępu wybrał optymalizator do wykonania tego zapytania?

```
select rowid, nazwisko, plec, placa
from prac
where id_prac = 900;
```

5. Zdefiniuj zapytanie, które odczyta te same dane, co zapytanie z zadania 4., posłuż się w tym celu adresem rekordu (`rowid`), odczytanym w zadaniu 4. Następnie wyjaśnij plan tego zapytania. Jaka metoda dostępu do danych została użyta?
6. Wyłącz dyrektywę `AUTOTRACE`. Sprawdź w słowniku bazy danych, jakie indeksy założono na relacji `PRAC`.
7. Utwórz indeks typu B-drzewo o nazwie `PRAC_IDX` na atrybucie `ID_PRAC` relacji `PRAC`. Następnie ponownie wyjaśnij plan zapytania z punktu 4. Czym różni się otrzymany plan?
8. Usuń indeks `PRAC_IDX`. Następnie na atrybucie `ID_PRAC` relacji `PRAC` zdefiniuj klucz podstawowy o nazwie `PRAC_PK`.

9. Ponownie sprawdź w słowniku bazy danych indeksy dla relacji PRAC. Wyświetl dodatkowo nazwy poindeksowanych atrybutów. Co zauważyłeś/aś?
10. Ponownie wyjaśnij plan zapytania z punktu 4. Czym różni się otrzymany plan?
11. Utwórz indeks typu B-drzewo o nazwie PRAC_NAZWISKO_IDX na atrybucie NAZWISKO relacji PRAC. Następnie wykonaj poniższe zapytania, dla każdego wyjaśniając jego plan.

```
select * from prac where nazwisko = 'Prac155';
select * from prac where nazwisko like 'Prac155%';
select * from prac where nazwisko like '%Prac155%';
select * from prac where nazwisko like 'Prac155'
        or nazwisko like 'Prac255%';
```

Dlaczego w przedostatnim zapytaniu optymalizator nie użył indeksu?

12. Usuń indeks PRAC_NAZWISKO_IDX i na jego miejsce utwórz skonkatenowany indeks typu B-drzewo o nazwie PRAC_NAZW_PLACA_IDX na atrybutach NAZWISKO i PLACA relacji PRAC. Następnie wykonaj poniższe zapytania, dla każdego wyjaśniając jego plan.

```
select count(*) from prac where nazwisko like 'Prac1%';
select count(*) from prac where nazwisko like 'Prac1%' and placa > 100;
select count(*) from prac where placa > 100;
```

Dlaczego w ostatnim zapytaniu optymalizator nie użył indeksu?

13. Utwórz indeks typu B-drzewo o nazwie PRAC_PLEC_IDX na atrybucie PLEC relacji PRAC. Następnie wykonaj poniższe zapytanie.

```
select count(*) from prac where plec='M'
and id_prac between 100 and 110;
```

14. Przełącz się na optymalizator kosztowy i ponownie wyjaśnij plan zapytania z zadania 13. Czy zaobserwowałeś/aś jakąś różnicę? Jaki typ optymalizatora został użyty do wykonania zapytania?

15. Sprawdź, czy dla relacji PRAC zebrano statystyki. Skorzystaj z poniższych zapytań.

```
select table_name, last_analyzed, num_rows from user_tables
where table_name='PRAC';
select column_name, num_distinct, low_value, high_value
       num_buckets from user_tab_columns where table_name = 'PRAC';
select index_name, last_analyzed, num_rows from user_indexes where
table_name='PRAC';
select * from user_tab_histograms where table_name='PRAC';
```

16. Zbierz statystyki dla relacji PRAC (użyj polecenia ANALYZE TABLE). Następnie ponownie wykonaj zapytania z zadania 15.

17. Zbierz statystyki dla indeksów relacji PRAC. Następnie ponownie wykonaj zapytania z zadania 15.

18. Zbuduj histogramy dla wszystkich poindeksowanych kolumn relacji PRAC. Następnie ponownie wykonaj zapytania z zadania 15.

19. Powtórz zapytanie z punktu 13. Jakie zauważyłeś/aś różnice w stosunku do wyniku zadania 13, wykonywanego bez statystyk?

20. Wykonaj poniższe zapytanie. Co zauważyłeś/aś? Czy przy odpowiedzi na poniższe zapytanie optymalizator korzysta z relacji PRAC?

```
select count(*) from prac;
```

21. Utwórz indeks typu B-drzewo o nazwie PRAC_CZY_ETAT_IDX na atrybucie CZY_ETAT relacji PRAC. Zbierz statystyki dla tego indeksu. Następnie wykonaj poniższe zapytanie i zanalizuj jego plan.

```
select count(*) from prac where czy_etat='T' and plec='K';
```

22. Usuń indeksy na atrybutach PLEC i CZY_ETAT, wykonaj ponownie zapytanie z zadania 21. Czy widzisz różnice w obu planach?

23. Utwórz indeksy bitmapowe na relacji PRAC, na atrybutach PLEC i CZY_ETAT, o nazwach odpowiednio PRAC_PLEC_BMP_IDX i PRAC_CZY_ETAT_BMP_IDX. Zbierz statystyki dla obu indeksów.

24. Włącz optymalizator regułowy, następnie wykonaj poniższe zapytanie:

```
select count(*) from prac where czy_etat='T' and plec='K';
```

25. Włącz optymalizator kosztowy, następnie wykonaj ponownie zapytanie z zadania 24. Jakie różnice zaobserwowałeś/aś?

26. Utwórz indeks typu B-drzewo o nazwie PRAC_PLACA_IDX na atrybucie PLACA relacji PRAC. Zbierz statystyki dla tego indeksu.

27. Sprawdź plany następujących zapytań:

```
select nazwisko from prac where placa < 2;  
select nazwisko from prac where ROUND(placa) < 2;
```

Dlaczego drugie zapytanie nie korzysta z indeksu na płacy?

28. Ustaw:

```
alter session set query_rewrite_enabled=true;  
alter session set query_rewrite_integrity=trusted;
```

29. Utwórz indeks funkcyjny o nazwie PRAC_PLACA_FUN_IDX na relacji PRAC, który będzie używany przy zapytaniu o zaokrągloną wartość płacy pracownika. Wykonaj ponownie zadanie 27. Jakie różnice zaobserwowałeś/aś?

Sortowanie

30. Porównaj plany wykonania następujących zapytań

```

select * from prac order by id_prac;
select * from prac order by id_prac desc;
select * from prac order by nazwisko;
select distinct nazwisko from prac;
select nazwisko from prac group by nazwisko;

```

Tabele IOT

31. Zbuduj tabelę o organizacji indeksowej o nazwie PRAC_ZESP_IOT i zapełnij danymi z tabel prac i zesp, wykonując poniższe zapytanie.

```

CREATE TABLE prac_zesp_iot (
id_prac NUMBER PRIMARY KEY,
nazwisko VARCHAR2(20),
nazwa VARCHAR2(20) )
ORGANIZATION INDEX
PCTTHRESHOLD 20
OVERFLOW TABLESPACE USERS;

insert into prac_zesp_iot select id_prac, nazwisko, nazwa
from prac, zesp where prac.id_zesp = zesp.id_zesp;

```

Sprawdź plan wykonania poniższego zapytania do takiej tabeli.

```

select * from prac_zesp_iot where id_prac < 100;

```

Klastry

32. Zbuduj klastrowy indeks o nazwie PRAC_ZESP_CI do przechowywania danych pracowników i zespołów.

```

create cluster prac_zesp_ci(id_zesp number);
create index prac_zesp_ci_idx on cluster prac_zesp_ci;
create table zesp_ci
cluster prac_zesp_ci(id_zesp) as select * from zesp;
create table prac_ci
cluster prac_zesp_ci(id_zesp) as select * from prac;

```

33. Sprawdź plan wykonania zapytania do tabel klastrowanych:

```

select count(*) from zesp_ci natural join prac_ci
where nazwa = 'ALGORYTMY';

```

Tabele partycjonowane

34. Utwórz tabelę partycjonowaną PRAC_PART, w której dane pracowników są podzielone na następujące partycje zgodnie z wartością atrybutu PLACA:

- „biedni” – płaca mniejsza niż 300 zł,
- „sredni” – płaca większa lub równa 300 zł, mniejsza niż 800
- „bogaci” – płaca większa lub równa 800 zł.

Wstaw następnie do tabeli dane z tabeli PRAC. Sprawdź rozkład danych pomiędzy poszczególnymi partycjami tabeli PRAC_PART;

Połączenia

35. Włącz optymalizator regułowy.

```
alter session set optimizer_goal = rule;
```

Ustaw tryb wyjaśniania zapytań dyrektywą

```
set autotrace traceonly explain statistics
```

36. Sprawdź plan wykonania następujących zapytań z połączeniem naturalnym:

```
select count(*) from zesp natural join prac;  
select count(*) from prac natural join zesp;
```

37. Utwórz indeks typu B-drzewo o nazwie ZESP_ID_ZESP_IDX na atrybucie ID_ZESP relacji ZESP. Wykonaj ponownie oba zapytania z zadania 36.

38. Utwórz indeks typu B-drzewo o nazwie PRAC_ID_ZESP_IDX na atrybucie ID_ZESP relacji PRAC. Wykonaj ponownie oba zapytania z zadania 36.

39. Włącz optymalizator kosztowy, zbierz statystyki dla tabeli ZESP i jej indeksów oraz tabeli PRAC i jej indeksu PRAC_ID_ZESP_IDX. Wykonaj ponownie oba zapytania z zadania 36.

40. Usuń wszystkie indeksy (poza indeksem na kluczu głównym tabeli PRAC). Wykonaj poniższe zapytanie i przeanalizuj jego plan wykonania.

```
select nazwa, count(*) from prac natural join zesp group by nazwa;
```

Statystyki, histogramy

41. Usuń statystyki dla tabeli PRAC. Następnie dokonaj oszacowania statystyk dla tabeli PRAC na podstawie próbki 10%. Wyłącz tryb wyświetlania planu zapytania i sprawdź informacje o statystykach dla tabeli PRAC.

42. Poznaj rozkład wartości atrybutu PLACA_DOD w tabeli PRAC.

43. Usuń statystyki dla tabeli PRAC. Utwórz indeks typu B-drzewo o nazwie PRAC_PLACA_DOD_IDX na atrybucie PLACA_DOD tabeli PRAC. Włącz tryb wyświetlania planu zapytania.

44. Wyświetl plany poniższych zapytań

```
select * from prac where placa_dod = 100;  
select * from prac where placa_dod = 999;
```

Co zaobserwowałeś/aś?

45. Zbierz statystyki dla tabeli PRAC. Powtórz zapytania z zadania 43.

46. Utwórz histogram dla atrybutu PLACA_DOD tabeli PRAC. Powtórz zapytania z zadania 43.

Wskazówki

47. Cel optymalizacji. Wykonaj poniższe zapytanie, wyświetl plan tego zapytania.

```
select count(*) from prac where plec = 'K';
```

Następnie dodaj do zapytania taką wskazówkę, aby zapytanie to zrealizował optymalizator regułowy.

48. Usuń indeks bitmapowy PRAC_PLEC_BMP_IDX, założony na atrybucie PLEC tabeli PRAC. Następnie utwórz na tym samym atrybucie indeks B-drzewo o nazwie PRAC_PLEC_IDX.

49. Dostęp do danych. Sprawdź plan poniższego zapytania. Następnie dodaj do zapytania taką wskazówkę, aby optymalizator kosztowy użył utworzonego w zadaniu 47. indeksu.

```
select count(*) from prac where id_prac < 100 and plec = 'K';
```

50. Dostęp do danych. Wykonaj ponownie zapytanie z zadania 48, tym razem dodaj wskazówkę, która spowoduje, że optymalizator nie użyje żadnego indeksu przy dostępie do tabeli PRAC.

51. Dostęp do danych. Sprawdź plan poniższego zapytania.

```
select * from prac
where id_prac = 100 or id_prac = 200 or id_prac = 300;
```

Następnie dodaj do zapytania wskazówkę USE_CONCAT. Jak zmienił się plan zapytania?

52. Kolejność łączenia tabel. Sprawdź plan wykonania poniższego zapytania.

```
select * from prac natural join zesp;
```

Następnie dodaj do zapytania wskazówkę ORDERED. Co zaobserwowałeś/aś? Zmień kolejność relacji w klauzuli from i ponownie wykonaj zapytanie.

53. Porównaj koszty wykonania połączenia naturalnego tabel PRAC i ZESP za pomocą każdego algorytmu łączenia tabel i dla każdej kolejności łączonych tabel (6 kombinacji). Określ plan charakteryzujący się najniższym kosztem.

Ćwiczenie indywidualne

Utwórz tabele potrzebne do ćwiczeń i wypełnij je danymi.

```
drop table zamowienia;
drop table pozycje;
create table zamowienia( data_zamowienia date,lp_zam number,
id_klienta number,data_realizacji date,uwagi varchar2(200));

create table pozycje(data_zamowienia date,lp_zam number,lp_pozycji number,
id_towaru number,ilosc number,kwota number);

drop sequence id_zam;
create sequence id_zam;
drop sequence id_poz;
create sequence id_poz;
```

```
insert into zamowienia values(sysdate, -1, -1, sysdate, 'no comments');
commit;
```

```
declare
  number;
begin
  for i in 1..14 loop
    insert into zamowienia (data_zamowienia ,lp_zam ,id_klienta,
      data_realizacji ,uwagi )
    select to_date('1997','yyyy')+round(id_zam.nextval/4),
      mod(id_zam.currval,4),mod(id_zam.currval+13,1023),
      to_date('1997','yyyy')+round(id_zam.currval/4)+10,
      'no comments'
    from zamowienia;
    commit;
  end loop;
end;
/
```

```
declare
  i number;
begin
  for i in 1..12 loop
    insert into pozycje (data_zamowienia ,lp_zam ,lp_pozycji , id_towaru ,
      ilosc ,kwota )
    select data_zamowienia, lp_zam,i,mod(lp_zam+17,253),
      mod(lp_zam+13,7),mod(lp_zam+31,2047)from zamowienia;
    commit;
  end loop;
end;
/
```

Zoptymalizuj zapytanie: „Podaj liczbę różnych klientów obsłużonych w styczniu 1999 i sumaryczną wartość ich zamówień.”