

Optimalizacja zapytań - ćwiczenia¹

0. Utwórz tabelę PLAN_TABLE używając załączonego skryptu *utlxplan.sql*. Samodzielnie przestudiuj dokumentację i „odkryj”, po co jest ta tabela potrzebna. Skrypt ten jest zlokalizowany w katalogu \$ORACLE_HOME/rdbms/admin.

1. Utwórz potrzebne do ćwiczeń tabele i wypełnij je danymi:

```
CREATE TABLE zesp ( id_zesp number, nazwa varchar2(20));

INSERT INTO zesp VALUES (1,'Bazy danych');
INSERT INTO zesp VALUES (2,'Badania operacyjne');
INSERT INTO zesp VALUES (3,'Algorytmy');
INSERT INTO zesp VALUES (4,'Inżynieria opr. ');
INSERT INTO zesp VALUES (5,'Bezpieczeństwo');

CREATE TABLE prac
( id_prac NUMBER, id_zesp NUMBER, nazwisko VARCHAR2(50), plec CHAR(1),
czy_etat CHAR(1), placza NUMBER );

DECLARE
  v_i NUMBER := 0;
  v_plec CHAR(1);
  v_etat CHAR(1);
BEGIN
  WHILE v_i < 10000 LOOP
    IF(MOD(v_i,10) < 2)
      THEN v_plec := 'K';
    ELSE v_plec := 'M';
    END IF;
    IF(MOD(v_i,10) < 1)
      THEN v_etat := 'N';
    ELSE v_etat := 'T';
    END IF;
    INSERT INTO prac
    VALUES (v_i, MOD(v_i,5)+1, 'Prac'||v_i, v_plec, v_etat, MOD(v_i,1000) );
    IF (MOD(v_i,1000)=0)
      THEN COMMIT;
    END IF;
    v_i := v_i + 1;
  END LOOP;
END;
/
```

2. Wyjaśnij plan wykonania poniższego zapytania, korzystając z polecenia EXPLAIN PLAN:

```
EXPLAIN PLAN FOR
SELECT * FROM prac WHERE id_prac=900;
```

3. Zaobserwuj wyniki przechowywane w tabeli PLAN_TABLE.

a) samodzielnie formułując zapytanie wyświetlające OPERATION, OPTIONS, OBJECT_NAME, ID, COST, PARENT_ID

b) korzystając z poniższego zapytania:

¹ Ćwiczenie opracowano na podstawie materiałów ze strony Bartosza Bębela:
<http://www.cs.put.poznan.pl/bbebel/>

```
SELECT plan_table_output FROM
TABLE(dbms_xplan.display('plan_table',null,'serial'));
```

4. Włącz dyrektywę SET AUTOTRACE ON EXPLAIN.

5. Włącz dyrektywę SET TIMING ON.

6. Włącz optymalizator regułowy (już w wersji 8 był uważany za przestarzały, w wersji 9i był jeszcze dostępny a w wersji 10g zniknął zupełnie - AG)

7. Sprawdź plan wykonania zapytania:

```
SELECT * FROM prac WHERE id_prac=900;
```

ID_PRAC	ID_ZESP	NAZWISKO	P	C	PLACA
900	1	Prac900	K	N	900

1 wiersz został wybrany.

Całkowity: 00:00:00.00

Plan wykonywania

```
-----
0      SELECT STATEMENT Optimizer=CHOOSE
1      0      TABLE ACCESS (FULL) OF 'PRAC'
```

8. Utwórz indeks PRAC_PK_IDX na atrybucie ID_PRAC relacji PRAC.

9. Sprawdź plan wykonania zapytania:

```
SELECT * FROM prac WHERE id_prac=900;
```

10. Usuń i odtwórz indeks unikalny na atrybucie ID_PRAC relacji PRAC.

11. Utwórz indeks PRAC_PLEC_IDX na atrybucie PLEC relacji PRAC.

12. Sprawdź plan wykonania zapytania:

```
SELECT * FROM prac WHERE plec='M' AND id_prac BETWEEN 100 AND 110;
```

13. Zbierz statystyki dla relacji PRAC i powtórz ćwiczenie 12.

```
ANALYZE TABLE Prac COMPUTE STATISTICS FOR TABLE;
```

```
ANALYZE TABLE Prac COMPUTE STATISTICS FOR ALL INDEXES;
```

```
ANALYZE TABLE Prac COMPUTE STATISTICS FOR ALL INDEXED COLUMNS;
```

14. Sprawdź plan wykonania zapytania:

```
SELECT nazwisko
FROM prac
WHERE plec = 'K' AND id_prac = 10 OR id_prac = 20;
```

15. Następnie 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';

```

16. Sprawdź plan wykonania zapytania:

```

SELECT nazwisko
FROM prac
WHERE plec = 'K' AND (id_prac = 10 OR id_prac = 20);

```

17. Sprawdź plan wykonania zapytania:

```

SELECT * FROM prac WHERE 1=0;

```

18. Sprawdź plan wykonania zapytania:

```

SELECT COUNT(*) FROM prac WHERE plec='K';

```

19. Utwórz indeks na atrybucie CZY_ETAT relacji PRAC i sprawdź plan wykonania zapytania:

```

SELECT COUNT(*) FROM prac WHERE czy_etat='T' AND plec='K';

```

20. Usuń indeksy na atrybutach PLEC i ETAT relacji PRAC i utwórz na ich miejsce indeksy bitmapowe, po czym sprawdź plan wykonania zapytania:

```

SELECT COUNT(*) FROM prac WHERE czy_etat='T' AND plec='K';

```

21. Usuń indeksy bitmapowe i sprawdź plan wykonania dla zapytania:

```

SELECT nazwisko FROM prac
WHERE nazwisko = 'Prac3000' OR id_prac=2000;

```

22. Utwórz zwykły indeks bitmapowy na płci i sprawdź plan wykonania zapytania:

```

SELECT MAX(placa) FROM prac WHERE plec='K';

```

23. Porównaj z planem:

```

SELECT /*+ INDEX(prac prac_plec_idx) */ MAX(placa)
FROM prac WHERE plec='K';

```

24. Utwórz indeks na atrybucie PLACA relacji PRAC i sprawdź zapytania:

```

SELECT nazwisko FROM prac WHERE placa < 2;
SELECT nazwisko FROM prac WHERE ROUND(placa) < 2;

```

25. Utwórz indeks 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? Jak wygląda postać planu dla ostatniego zapytania?

26. Utwórz indeks *funkcyjny* na wyrażeniu ROUND(PLACA) relacji PRAC i sprawdź jeszcze raz plan wykonania zapytania:

```
SELECT nazwisko FROM prac WHERE ROUND(placa) < 2;
```

27. Usuń indeksy na płacy. Utwórz indeks *skonkatenowany* na atrybutach NAZWISKO i PLACA relacji PRAC i sprawdź jego wykorzystanie w następujących zapytaniach:

```
SELECT * FROM prac WHERE nazwisko LIKE 'Prac90%' AND placa < 20;
SELECT * FROM prac WHERE placa < 20 AND nazwisko LIKE 'Prac90%';
SELECT * FROM prac WHERE nazwisko LIKE 'Prac9000%';
SELECT * FROM prac WHERE placa < 1;
SELECT COUNT(*) FROM prac WHERE placa < 1;
```

28. Sprawdź plan wykonania następujących zapytań z połączeniem naturalnym:

```
SELECT count(*) FROM zesp NATURAL JOIN prac;
SELECT count(*) FROM prac NATURAL JOIN zesp;
```

29. Utwórz indeks o nazwie ZESP_ID_ZESP_IDX na atrybucie ID_ZESP relacji ZESP. Wykonaj ponownie oba zapytania.

30. Utwórz indeks o nazwie PRAC_ID_ZESP_IDX na atrybucie ID_ZESP relacji PRAC. Wykonaj ponownie oba zapytania.