

TWORZENIE WŁASNYCH TYPÓW INDEKSÓW – ZADANIA

Krzysztof Jankiewicz, Instytut Informatyki Politechnika Poznańska

Zadanie 1 – indeks przestrzenny CONTAINS (quad-drzewo)

Utwórz indeks, który będzie zastępował indeks typu SPATIAL_INDEX.

- Zakładamy, że w bazie danych będą istniały tabele punktów wykorzystujących typ SDO_GEOMETRY.
- Indeks ma wspierać wyszukiwanie tych wierszy, które zawierają punkty mieszczące się w zadanym prostokącie.
- Operator wykorzystujący nowostworzony indeks to MY_CONTAINS. Jego postać ma być następująca:
MY_CONTAINS(GEOM, X1, Y1, X2, Y2); gdzie GEOM to kolumna zawierająca poszukiwane punkty, X1, Y1 lewy dolny kąt prostokąta definiującego przeszukiwany obszar, X2, Y2 prawy górny kąt tego prostokąta.
- Podczas definiowania, tworzenia i wykorzystywania indeksu nie mogą być używane żadne wbudowane operatory przestrzenne.

Zadanie 2 – indeks tekstowy

Utwórz indeks, który będzie zastępował indeks typu CONTEXT.

- Zakładamy, że w bazie danych będą istniały tabele zawierające duże obiekty tekstowe (VARCHAR2(4000)).
- Indeks ma wspierać wyszukiwanie w tych wierszy, które zawierają określone słowa lub ich sekwencje.
- Operator wykorzystujący nowostworzony indeks to MY_CONTAINS. Jego postać ma być następująca:
MY_CONTAINS(TEXT, SZUKANY_CIAĞ); gdzie TEXT to kolumna zawierająca ciągi tekstu zawierająca poszukiwane słowa, SZUKANY_CIAĞ będzie definiował poszukiwane słowa lub ciąg słów. Przykładowa zawartość parametru SZUKANY_CIAĞ to 'śrubka' – poszukiwanie dokumentu zawierającego zadane słowo, 'zając & śrubka' – poszukiwanie dokumentu zawierającego zadane słowa, 'zając śrubka' – poszukiwanie dokumentu zawierającego zadaną sekwencję.

Zadanie 3 – wyszukiwanie zbiorów

Utwórz indeks, który będzie przyspieszał wyszukiwanie zbiorów.

- Zakładamy, że w bazie danych będą istniały tabele zawierające obiekty, których zawartością będą tablice o zmiennej długości liczb. Tablice te będą zawierały nieuporządkowane zbiory liczb.
- Indeks ma wspierać wyszukiwanie tych wierszy, które zawierają obiekt posiadający zadany zbiór liczb.
- Operator wykorzystujący nowostworzony indeks to MY_CONTAINS. Jego postać ma być następująca:
MY_CONTAINS(OBIEKT, ZBIÓR); gdzie OBIEKT to kolumna zawierająca obiekty posiadające w swojej zawartości atrybut będący zbiorem liczb, ZBIÓR to instancja tego samego typu przechowująca poszukiwany przez nas zbiór liczb.

Zadanie 4 – wyszukiwanie klas ekwiwalencji blokad

Metody kontroli współbieżnego dostępu do baz danych są w wielu przypadkach oparte na blokadach i macierzach kompatybilności blokad. Podczas opracowywania zestawu blokad wyznacza się tzw. klasy ekwiwalencji blokad. Blokad X i Y są ekwiwalentne (należą do tej samej klasy ekwiwalencji) jeżeli dla dowolnej blokady Z: $COMP(X, Z) = COMP(Y, Z)$ oraz $COMP(Z, X) = COMP(Z, Y)$, gdzie COMP jest funkcją kompatybilności.

Przykładowe dane wejściowe:

```
desc CC_BLOKADY
```

Nazwa	Typ
B_ID	NUMBER(4)
B_NAZWA	VARCHAR2(100)

```
desc CC_MACIERZ_KOMPATYBILNOSCI
```

Nazwa	Typ
M_B_ID_HELD	NUMBER(4)
M_B_ID_REQUESTED	NUMBER(4)
M_KOMPATYBILNE	NUMBER(1)

- Indeks ma być zakładany na tabeli z blokadami i ma umożliwiać wyszukiwanie blokad znajdujących się w tej samej klasie ekwiwalencji, co zadana blokada.
- Operator wykorzystujący nowostworzony indeks to EQCLASS. Jego postać ma być następująca: EQCLASS(OBIEKT, TYP_BLOKADY); gdzie OBIEKT to kolumna z tabeli CC_BLOKADY na której zostanie założony indeks, TYP_BLOKADY to nazwa blokady, dla której poszukujemy blokad znajdujących się w tej samej klasie ekwiwalencji.

Zadanie 5 – wyszukiwanie wolnych pokoi

W hotelowej bazie danych znajdują się dane dotyczące pokoi hotelowych oraz rezerwacji. Zdefiniuj indeks, który będzie przyspieszał wyszukiwanie wolnych pokoi w zadanym interwale.

Przykładowe dane wejściowe:

```
desc H_POKOJE
```

Nazwa	Typ
P_ID	NUMBER(4)
P_NAZWA	VARCHAR2(100)

```
desc H_REZERWACJE
```

Nazwa	Typ
R_B_ID	NUMBER(4)
R_OD	DATE
R_DO	DATE

- Indeks ma być zakładany na tabeli z pokojami.
- Operator wykorzystujący nowostworzony indeks to WOLNY_W_DNIACH. Jego postać ma być następująca: WOLNY_W_DNIACH(OBIEKT, OD_DNIA, DO_DNIA); gdzie OBIEKT to kolumna z tabeli H_POKOJE, na której zostanie założony indeks, OD_DNIA i DO_DNIA to graniczne daty wyznaczające okres na jaki potrzebna jest rezerwacja.

Zadanie 6 – indeks przestrzenny NEAREST (R-drzewo)

Utwórz indeks, który będzie zastępował indeks typu SPATIAL_INDEX.

- Zakładamy, że w bazie danych będą istniały tabele prostokątów zoptymalizowanych wykorzystujących typ SDO_GEOMETRY.
- Indeks ma wspierać wyszukiwanie tych prostokątów, które znajdują się najbliżej zadanego punktu.
- Operator wykorzystujący nowostworzony indeks to MY_NEAREST. Jego postać ma być następująca:
MY_NEAREST(GEOM, X, Y, LICZBA); gdzie GEOM to kolumna zawierająca poszukiwane prostokąty, X, Y punkt, dla którego szukamy najbliższych prostokątów, LICZBA – liczba poszukiwanych prostokątów.
- Podczas definiowania, tworzenia i wykorzystywania indeksu nie mogą być używane żadne wbudowane operatory przestrzenne.

Zadanie 7 – wyszukiwanie samochodów

Utwórz indeks, który będzie przyspieszał wyszukiwanie samochodów posiadających określone cechy.

Przykładowe dane wejściowe:

```
desc S_SAMOCODY
```

Nazwa	Typ
S_ID	NUMBER(4)
S_NAZWA	VARCHAR2(100)

```
desc S_CECHY
```

Nazwa	Typ
C_ID	NUMBER(4)
C_NAZWA	VARCHAR2(100)

```
desc S_CECHY_SAMOCODÓW
```

Nazwa	Typ
CS_S_ID	NUMBER(4)
CS_C_ID	NUMBER(4)

- Indeks ma być zakładany na tabeli z samochodami.
- Operator wykorzystujący nowostworzony indeks to ZNAJDZ_SAMOCOD. Jego postać ma być następująca:
ZNAJDZ_SAMOCOD(OBIEKT, TABLICA_LICZB); gdzie OBIEKT to kolumna z tabeli S_SAMOCODY, na której zostanie założony indeks, TABLICA_LICZB to identyfikatory cech, które poszukiwany samochód powinien posiadać. TABLICA_LICZB może być tablicą o zmiennej długości lub instancją typu obiektowego z odpowiednimi atrybutami przechowującymi identyfikatory cech.

Zadanie 8 – wyszukiwanie list

Utwórz indeks, który będzie przyspieszał wyszukiwanie list.

- Zakładamy, że w bazie danych będą istniały tabele zawierające obiekty, których zawartością będą tablice o zmiennej długości liczb. Tablice te będą zawierały uporządkowane listy liczb.
- Indeks ma wspierać wyszukiwanie tych wierszy, które zawierają obiekt posiadający listę, w której będzie zawarty poszukiwany ciąg liczb.
- Operator wykorzystujący nowostworzony indeks to MY_CONTAINS. Jego postać ma być następująca:
MY_CONTAINS(OBIEKT, LISTA); gdzie OBIEKT to kolumna zawierająca obiekty posiadające w swojej zawartości atrybut będący listą liczb, LISTA to instancja tego samego typu przechowująca poszukiwaną przez nas listę liczb.

Harmonogram prac:

Tydzień 1 – zaprojektowanie struktury indeksu, oraz metody jego wykorzystania. Oprogramowanie tworzenia indeksu.

Tydzień 2 – oprogramowanie mechanizmów odpowiadających za aktualizację indeksu.

Tydzień 3 – oprogramowanie mechanizmów odpowiadających za wykorzystanie indeksu podczas wyszukiwania danych.

Tydzień 4 – oddawanie projektów.