



Wydział Elektroniki
i Technik Informatycznych

POLITECHNIKA WARSZAWSKA

Bazy Danych 1

edycja 21L

Laboratorium 11

Wprowadzenie i przebieg laboratorium



Wydajność w systemach bazodanowych

Co można stroić?

- Serwer bazy danych
- Schemat bazy danych
- Aplikację
 - strukturę aplikacji (gruby klient, cienki klient)
 - sposób współpracy z bazą danych
 - zapytania SQL



[Oracle database performance - docs](#)

Strojenie serwera bazy danych

- Strojenie SGA (System Global Area)
- Strojenie zasobów dyskowych
- Klastrowanie bazy danych

 [Oracle database architecture - docs](#)

Strojenie schematu bazy danych

- Normalizacja schematów relacyjnych
- Dobór typów danych
- Indeksowanie danych
- Klastrowanie tabel *(do samodzielnego opracowania)*
- Partycjonowanie - *operacja dzielenia dużej tabeli na mniejsze części. Zapytania SQL i złączenia są szybsze, ze względu na zmniejszoną liczbę operacji we/wy. Partycjonowanie może być poziome (na podstawie zakresu wartości) lub pionowe (oddzielenie kolumn). Dostęp do partycjonowanej tabeli przy użyciu zapytań SQL i instrukcji DML nie wymaga modyfikacji.*

Normalizacja schematów relacyjnych

Poziomy normalizacji (N^{th} Normal Form):

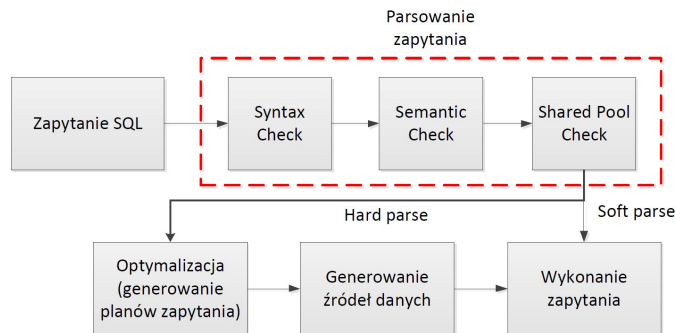
- Schematy nieznormalizowane
- Pierwsza postać normalna (1^{st} NF)
 - Normalizacja 1^{st} NF polega na podziale nieatomowych kolumn na atomowe
- Druga postać normalna (2^{nd} NF)
- Trzecia postać normalna (3^{rd} NF; Boyce-Codd NF)
- Czwarta postać normalna (4^{th} NF)
- Piąta postać normalna (5^{th} NF)
 - Normalizacja 2^{nd} NF ÷ 5^{th} NF zawsze polega na podziale tabel na mniejsze

Dobór typów danych

- Strony kodowe
 - ASCII – 1 bajt na znak
 - Unicode – 1 ÷ 4 bajtów na znak
- Typy znakowe
 - CHAR, VARCHAR (VARCHAR2), BLOB, CLOB
 - NCHAR, NVARCHAR2, NCLOB – Unicode
- Typy numeryczne
 - NUMBER – typ Oracle (do 21 bajtów)
 - BINARY_FLOAT (4 bajty), BINARY_DOUBLE (8 bajtów)

**Typy danych należy dobierać tak,
aby minimalizować przestrzeń
zajmowaną przez dane.**

Przetwarzanie zapytań SQL



- Zapytania po sprawdzeniu składni są interpretowane
- Zapytanie można wykonać na wiele sposobów opisanych za pomocą planów wykonania
- Optymalizator kosztowy wybiera plan wykonania o minimalnym koszcie (procesor, pamięć operacyjna, operacje dyskowe)

Do wersji 10 Oracle był używany również optymalizator regułowy

Optymalizator zapytań

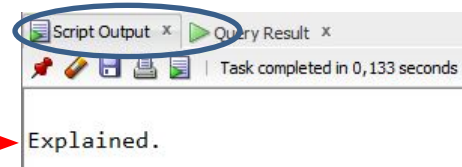
Tryby pracy określają rozmiar wynikowej puli wierszy zapytania, do której koszt dostępu należy zminimalizować:

- ALL_ROWS – (**domyślny**) wszystkie
- FIRST_ROWS – kilka pierwszych
- FIRST_ROWS_N – pierwszych N
 $N \in \{1, 10, 100, 1000\}$

```
alter session set optimizer_mode = first_rows;  
alter system set optimizer_mode = first_rows_10;
```

Plan zapytania (explain plan)

explain plan for
select * from dual;



SELECT PLAN_TABLE_OUTPUT FROM TABLE(DBMS_XPLAN.DISPLAY());

Script Output x Query Result x
All Rows Fetched: 8 in 0,054 seconds

PLAN_TABLE_OUTPUT

1 Plan hash value: 272002086

2

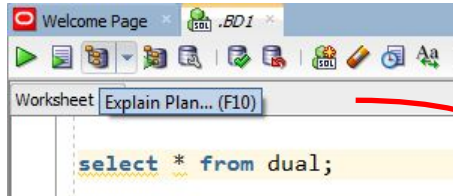
3

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	2	2 (0)	00:00:01
1	TABLE ACCESS FULL	DUAL	1	2	2 (0)	00:00:01

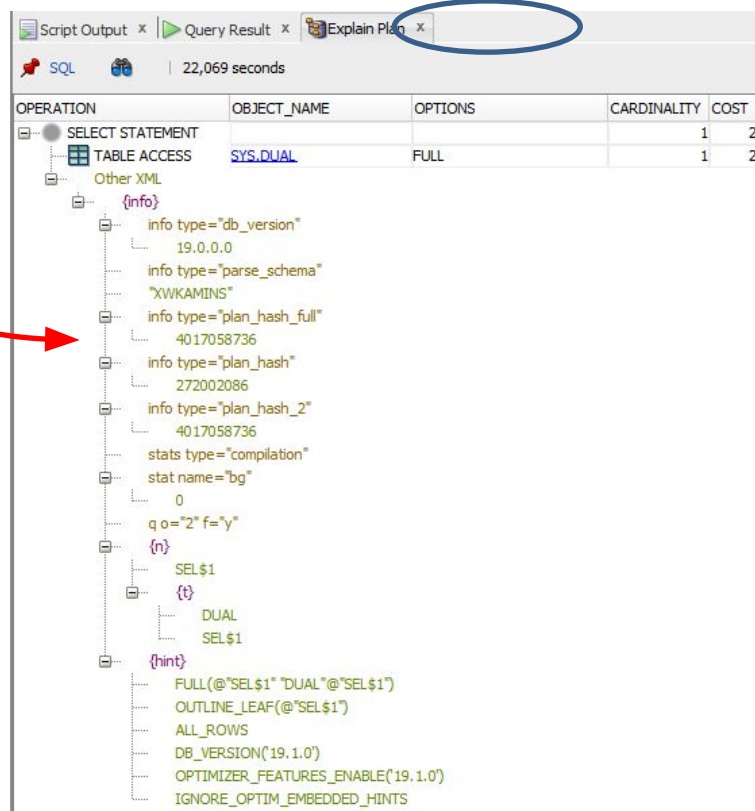
8

Plan zapytania (explain plan)

SQLDeveloper



```
select * from dual;
```



The 'Explain Plan' window displays the execution plan for the query 'select * from dual;'. The plan shows a 'SELECT STATEMENT' operation with a 'TABLE ACCESS' on 'SYS_DUAL'. The 'OPTIONS' column shows 'FULL'. The 'CARDINALITY' is 1 and the 'COST' is 2. Below the table, there is a detailed XML representation of the plan, including information about the database version, schema, and the specific table accessed.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	2
TABLE ACCESS	SYS_DUAL	FULL	1	2

Other XML

```
{info}
  info type="db_version"
    19.0.0.0
  info type="parse_schema"
    "XWKAMINS"
  info type="plan_hash_full"
    4017058736
  info type="plan_hash"
    272002086
  info type="plan_hash_2"
    4017058736
  stats type="compilation"
  stat name="bg"
    0
  q o="2" f="y"
  {n}
    SEL$1
    {t}
      DUAL
      SEL$1
  {hint}
    FULL(@SEL$1 "DUAL"@SEL$1)
    OUTLINE_LEAF(@SEL$1)
    ALL_ROWS
    DB_VERSION('19.1.0')
    OPTIMIZER_FEATURES_ENABLE('19.1.0')
    IGNORE_OPTIM_EMBEDDED_HINTS
```

Explain plan - ćwiczenia

Zapoznaj się z planami dla następujących zapytań:

1. `select * from employees;`
2. `select sysdate from dual;`
3. `select * from employees where employee_id > 130;`
4. `select * from employees where surname = 'Himuro';`
5. `select * from employees where salary between 1000 and 5000;`
6. `select department_id, count(*) from employees group by department_id;`
7. `select department_id, count(*) from employees where manager_id is not null group by department_id;`
8. `select department_id, count(*) from employees where manager_id is not null group by department_id having count (*) > 2;`

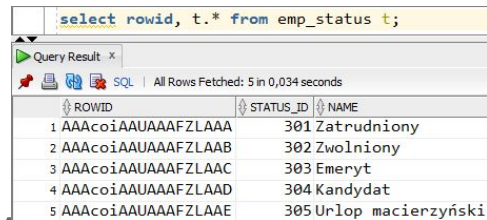
Plan zapytania: sposoby dostępu do danych

Ścieżka dostępu do danych	Tabele Heap-organized	Indeksy B-Tree	Indeksy bitmapowe	Klastry tabel
Full Table Scans	✓			
Table Access by Rowid	✓			
Sample Table Scans	✓			
Index Unique Scans		✓		
Index Range Scans		✓		
Index Full Scans		✓		
Index Fast Full Scans		✓		
Index Skip Scans		✓		
Index Join Scans		✓		
Bitmap Index Single Value			✓	
Bitmap Index Range Scans			✓	
Bitmap Merge			✓	
Cluster Scans				✓
Hash Scans				✓

Plan zapytania: sposoby dostępu do danych

Pokazują sposób dostępu do danych podczas realizacji zapytania.

- **Table Access Full** – przegląd danych **całej tabeli**.
- **Table Access by ROWID** – dostęp do wiersza tabeli za pomocą wskaźnika ROWID (zazwyczaj pobieranego z indeksu).
 - ROWID – wewnętrzna reprezentacja lokalizacji wiersza w bazie danych (plik danych + blok danych + pozycja w bloku)
 - **ROWID może się zmieniać!**
- **Sample Table Scan** – pobranie losowej próbki danych
`select * from employees sample (20);`



```
select rowid, t.* from emp_status t;
```

ROWID	STATUS_ID	NAME
1 AAACoiAAUAAAFZLAAA	301	Zatrudniony
2 AAACoiAAUAAAFZLAAB	302	Zwolniony
3 AAACoiAAUAAAFZLAAC	303	Emeryt
4 AAACoiAAUAAAFZLAAD	304	Kandydat
5 AAACoiAAUAAAFZLA AE	305	Urlop macierzyński

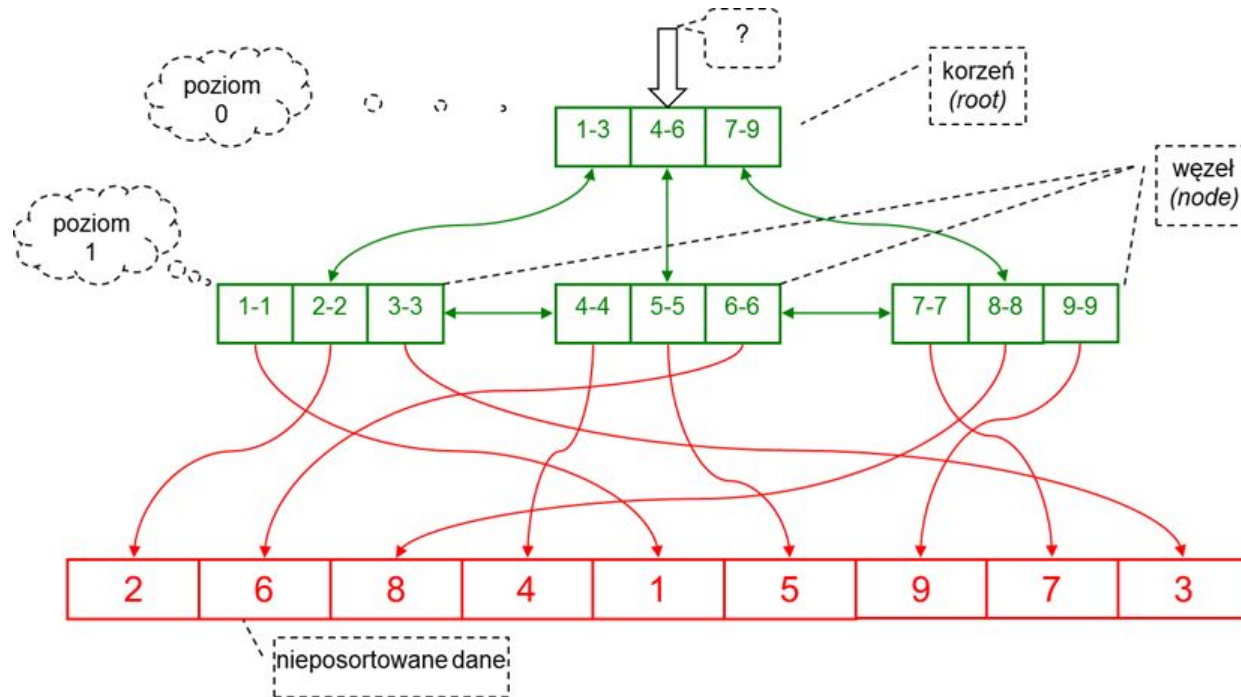
Indeksowanie

- Korzyści:
 - przyspieszenie dostępu do danych
 - pojedyncze rekordy
 - grupy rekordów
 - ułatwienie pewnych operacji
 - sortowanie
- Wady:
 - spowolnienie operacji manipulowania danymi (wstawienie, modyfikacja, usuwanie)
 - zużycie dodatkowych zasobów (pamięć dyskowa, pamięć operacyjna, procesor)
 - degradacja przy modyfikacji danych

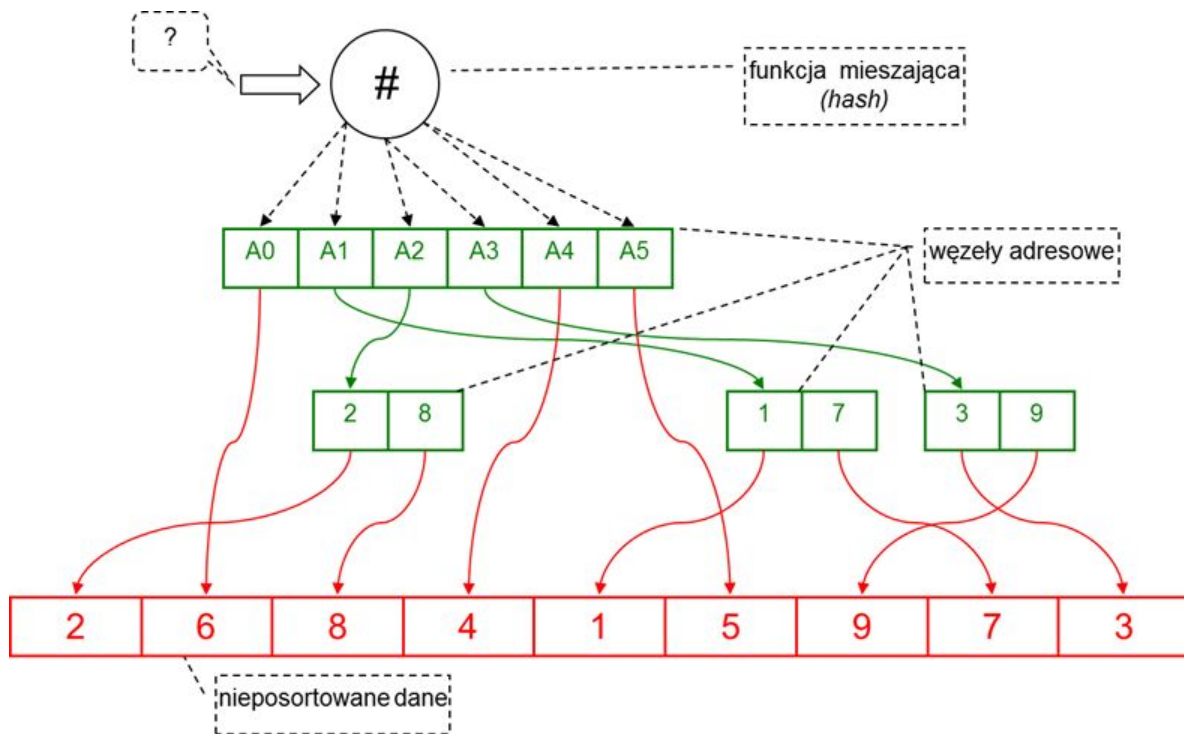
Indeksowanie

- Rodzaje indeksów:
 - sortujące – B-drzewa (*B-tree*)
 - (+) możliwość strojenia
 - (+) możliwość sortowania danych
 - (–) duże zużycie pamięci dyskowej
 - niesortujące – indeksy z funkcją mieszającą (*hash*), indeksy bitmapowe
 - (+) bardzo szybki dostęp do danych
 - (+) stosunkowo małe zużycie pamięci dyskowej
 - (–) brak możliwości sortowania

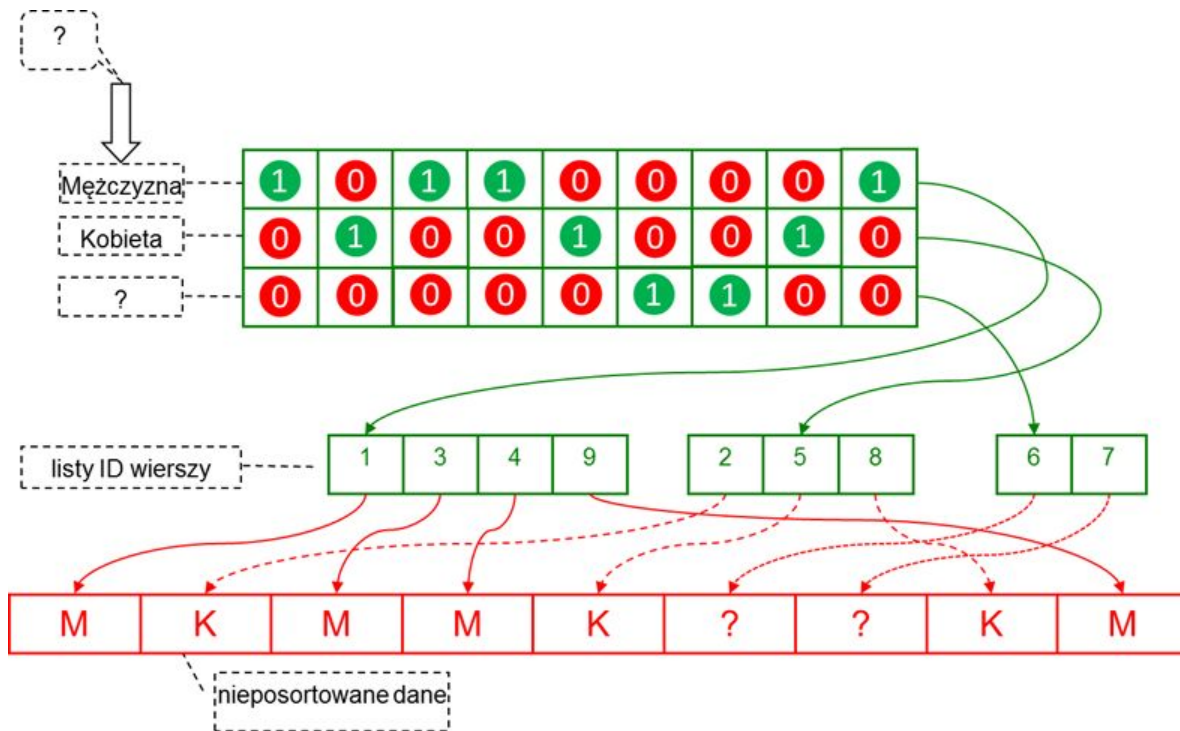
Idea indeksu B-drzewo



Idea indeksu haszującego



Idea indeksu bitmapowego



Indeksy w Oracle

- Rodzaje:
 - B-tree
 - Bitmap and bitmap join indexes
 - Function-based indexes
 - Application domain indexes

-- Sprawdzenie jakie są indeksy na tabeli

```
SELECT * FROM ALL_INDEXES where owner = 'HR' and table_name = 'EMPLOYEES';  
SELECT * FROM USER_INDEXES where table_name = 'EMPLOYEES';
```

Plan zapytania: sposoby dostępu do danych (2)

- **Index Unique Scans** – dostęp do pojedynczego wiersza tabeli z wykorzystaniem unikalnego indeksu przy warunkach selekcji na równość.
- **Index Range Scans** – dostęp do pojedynczego lub wielu wierszy tabeli z wykorzystaniem nieunikalnego indeksu przy warunkach selekcji na równość lub definiujących zakres.
- **Index Full Scans** – dostęp do wierszy w porządku wyznaczonym przez indeks w celu eliminacji oddzielnej operacji sortowania.
- **Index Fast Full Scans** – odczyt danych wynikowych zapytania bezpośrednio z indeksu a nie z tabeli.

Obsługa indeksów (B-tree)

- Tworzenie indeksu

```
CREATE [UNIQUE] INDEX nazwa_indeksu  
ON nazwa_tabeli (nazwa_kolumny {, nazwa_kolumny});
```

- Modyfikacja indeksu

```
ALTER INDEX nazwa_indeksu [ VISIBLE | INVISIBLE | UNUSABLE |  
REBUILD ];
```

- Usuwanie indeksu

```
DROP INDEX nazwa_indeksu;
```

Indeksy - ćwiczenia

1. Załóż indeks na kolumnie surname w tabeli employees.
2. Przeanalizuj plan zapytania z warunkiem równościowym na tej kolumnie. Przeanalizuj zapytanie z warunkiem równościowym na kluczu głównym tabeli employees. Porównaj uzyskane plany.
3. Zaindeksuj kolumnę tabeli zawierającą NULL-e i sprawdź plan wykonania przy zapytaniu z warunkiem „kolumna IS NULL” oraz „kolumna IS NOT NULL”.
4. Utwórz indeks B-tree na dwóch kolumnach (np. w tabeli positions na min_salary i max_salary) i sprawdź plan wykonania dla wszystkich kombinacji występowania kolumn indeksujących w klauzuli WHERE. Kiedy wykorzystany jest indeks?
5. Sprawdź plany wykonania zapytań z frazą ORDER BY dotyczącą kolumn zaindeksowanych null/not null.
6. Sprawdź plany wykonania zapytań z/bez fraz(ą/y) ORDER BY dotyczącą kolumn zaindeksowanych indeksami unikalnymi i nieunikalnymi, przy czym lista selekcji ma być ograniczona wyłącznie do tych kolumn.
7. Jakie czynniki mają wpływ na to czy indeks zostanie wykorzystany?

Obsługa indeksów funkcyjnych

- Tworzenie indeksu

```
CREATE [UNIQUE] INDEX nazwa_indeksu  
ON nazwa_tabeli (nazwa_funkcji(nazwa_kolumny));
```

- Modyfikacja indeksu

```
ALTER INDEX nazwa_indeksu  
ON nazwa_tabeli (nazwa_funkcji(nazwa_kolumny));  
  
ALTER INDEX nazwa_indeksu [ ENABLE | DISABLE ];
```



- DROP INDEX nazwa_indeksu;

Usunięcie indeksu

Indeksy funkcyjne - ćwiczenia

1. Wykonaj zapytanie na tabeli employees z warunkiem
surname = 'Himuro'

oraz z warunkiem

UPPER(surname) = 'HIMURO'.

Kiedy jest wykorzystany indeks?

2. Stwórz indeks funkcyjny na tabeli employees na kolumnie surname poddanej działaniu funkcji upper.
3. Wykonaj zapytanie z warunkiem upper(surname) = 'HIMURO'.
Czy jest wykorzystany indeks?

Plan wykonania: złączenia

- **Nested Loops**
 - Zbiory danych dzieli się na zewnętrzne i wewnętrzne
 - Dla każdego wiersza ze zbioru zewnętrznego wyszukuje się wiersze ze zbioru wewnętrznego (możliwe użycie indeksów w zbiorze wewnętrznym)
- **Sort Merge**
 - Odmiana Nested Loops
 - Zestawy danych są sortowane (operacje SORT JOIN)
 - Dla każdego wiersza w pierwszym zestawie danych sprawdzany jest drugi zestaw danych pod kątem zgodnych wierszy i łączy je (operacja MERGE JOIN)
 - Pozycja początkowa iteracji bazuje na dopasowaniu dokonanym w poprzedniej iteracji
- **Hash Join**
 - Dla mniejszego zbioru danych budowana jest tablica mieszająca na kluczu złączenia przechowywana w pamięci
 - Większy zbiór danych jest przeszukiwany ze względu na dopasowanie do wartości w tablicy mieszającej

Plan wykonania

złączenie (zaindeksowane tylko klucze główne)

The screenshot shows the SQL Developer interface with a query in the Worksheet and its execution plan in the Explain Plan tab.

Query:

```
select * from employees e inner join departments d on e.department_id = d.department_id;
```

Execution Plan:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			51	4
HASH JOIN			51	4
Access Predicates				
E.DEPARTMENT_ID=D.DEPARTMENT_ID				
TABLE ACCESS	DEPARTMENTS	FULL	19	2
TABLE ACCESS	EMPLOYEES	FULL	51	2
Filter Predicates				
E.DEPARTMENT_ID IS NOT NULL				
Other XML				

Wskazówki (hints)

Wskazówki:

- umożliwiają wpływanie na wybór planu wykonania przez optymalizator
- Są przekazywane za pomocą specjalnie sformatowanych komentarzy w treści zapytania

```
SELECT /*+ hint_text */ ...
```

```
/*+ NO_INDEX(employees emp_manager) */
```

```
/*+ INDEX(emp_dep_pk prac_kod_s) */
```

```
/*+ use_nl (emp dep) */ albo use_merge, use_hash
```

Złączenia - ćwiczenia

1. Jak zmieni się plan wykonania, jeśli w poprzednim przykładzie w definicji złączenia zastosujemy frazę USING?
2. Dodaj indeks B-tree na kolumnie klucza obcego z poprzedniego przykładu i sprawdź plan wykonania.
Jakiego rodzaju musi być to indeks (unikalny, nieunikalny)?
Czy wydajność złączenia uległa poprawie?
3. Za pomocą wskazówek zmień typ złączenia tabel.
4. Zmień warunek złączenia na nierównościowy, zweryfikuj jak zmienił się plan wykonania.

⚡ Następne zajęcia - sprawdzian ⚡

Zakres:

- widoki, zmaterializowane widoki, sekwencje, synonimy
- aspekty wydajności bazy danych



Praca domowa

1. Wykonaj ćwiczenia z przebiegu laboratorium
2. Co to jest klastrowanie tabel?
3. Do schematu bazy danych dodaj klaster tabel EMPLOYEES_CL oraz DEPARTMENTS_CL o strukturze odpowiadającej tabelom EMPLOYEES oraz DEPARTMENTS.
4. Przekopiuj dane z tabel EMPLOYEES oraz DEPARTMENTS do odpowiednich tabel w klastrze.
5. Sprawdź plany wykonania zapytań do pojedynczych tabel klastra z warunkami do różnych kolumn.
6. Sprawdź plan wykonania operacji złączenia sklastrowanych tabel.
7. Stwórz indeks bitmapowy na kolumnie status_id w tabeli pracownicy. Wykonaj zapytania na tej tabeli i sprawdź, czy indeks jest wykorzystywany.