

Bazy Danych

LABORATORIUM 2

Funkcja STDDEV:

Funkcja STDDEV służy do obliczania odchylenia standardowego wybranych komórek z danej kolumny.

```
1 SELECT STDDEV(<kolumna>) FROM <nazwa tabeli>;
```

Funkcja VARIANCE:

Funkcja VARIANCE służy do obliczania wariancji wybranych komórek z danej kolumny.

```
1 SELECT VARIANCE(<kolumna>) FROM <tabela>;
```

Klauzula GROUP BY:

Klauzula GROUP BY może zostać użyta do grupowania rekordów według wartości poszczególnych kolumny/kolumn. Pozwala to na wyliczanie wyrażeń agregujących dla określonych grup w danej tabeli.

```
1 -- Wyliczenie sredniej pensji wedlug menedzera i departamentu
   , sortowane rosnaco po sredniej pensji
2 SELECT
3     manager_id      AS "Menedzer",
4     department_id   AS "Departament",
5     AVG(salary)     AS "Srednia pensja"
6 FROM
7     employees
8 GROUP BY
9     manager_id ,
10    department_id
11 ORDER BY
12     AVG(salary);
```

Klauzula HAVING:

Klauzula HAVING jest klauzulą warunkową dla wyników funkcji agregujących. Często jest używana wraz z GROUP BY. Nie można użyć HAVING dla zwykłych kolumn, a WHERE do kolumn będących wynikami grupowania.

```
1  -- Wyliczenie sredniej pensji pracownikow, zarabiajacych
   ponizej 3000, mniejszej niz 5000, wedlug menedzera i
   departamentu, sortowane rosnaco po sredniej pensji
2  SELECT
3      manager_id      AS "Menedzer",
4      department_id   AS "Departament",
5      AVG(salary)     AS "Srednia pensja"
6  FROM
7      employees
8  WHERE
9      salary < 3000
10 GROUP BY
11     manager_id,
12     department_id
13 HAVING
14     AVG(salary) < 5000
15 ORDER BY
16     AVG(salary);
```

Funkcja SYSDATE:

Funkcja SYSDATE zwraca aktualną datę systemową. Wymagane jest jednak podanie tabeli, aby się do niej dostać. Przykładowo można użyć tabeli dummy o nazwie DUAL.

```
1  SELECT SYSDATE FROM <tabela>; -- Pobranie daty systemowej
```

Funkcja TO_DATE

Funkcja TO_DATE konwertuje zadany ciąg znaków o zadanej formie na datę.

```
1  SELECT
2      TO_DATE(
3          <ciąg znakow>,
4          <format daty>
5      )
6  FROM
7      <tabela>;
```

Funkcja ADD_MONTHS:

Funkcja ADD_MONTHS dodaje do zadanej daty zadaną liczbę miesięcy.

```
1 SELECT
2   ADD_MONTHS (
3     <data>,
4     <ilosc miesiecy>
5   )
6 FROM
7   tabela;
```

Funkcja MONTHS_BETWEEN:

Funkcja obliczająca różnicę między dwoma datami w miesiącach.

```
1 SELECT
2   MONTHS_BETWEEN (
3     <data 1>,
4     <data 2>
5   )
6 FROM
7   <tabela>;
```

Funkcja EXTRACT:

Funkcja pozwalająca na wyciągnięcie roku, dnia lub miesiąca z daty.

```
1 SELECT EXTRACT(DAY FROM <data>) FROM dual; -- wyjmij dzien
2 SELECT EXTRACT(MONTH FROM <data>) FROM dual; -- wyjmij mies.
3 SELECT EXTRACT(YEAR FROM <data>) FROM dual; -- wyjmij rok
```

Funkcja TO_CHAR:

Funkcja konwertująca zadaną datę na ciąg znaków o zadanym formacie.

```
1 SELECT
2   TO_CHAR (
3     <data>,
4     <format>
5   )
6 FROM
7   <tabela>;
```

Funkcja TRUNC:

Obcina datę do pierwszego dnia wskazanej jednostki czasu. Na przykład pierwszego dnia miesiąca, roku, dnia, tygodnia, kwartału.

```
1 SELECT
2     TRUNC (
3         <data>,
4         <ograniczenie>
5     )
6 FROM
7     <tabela>;
```

Funkcja LAST_DAY:

Funkcja zwracająca ostatni dzień miesiąca, obecnego we wskazanej dacie.

```
1 SELECT
2     LAST_DAY(<data>)
3 FROM
4     <tabela>;
```

Funkcja NEXT_DAY:

Funkcja zwracająca datę, w której wystąpi najbliższy zadany dzień tygodnia (względem zadanej daty).

```
1 -- Zwraca date najbliższego dnia tygodnia od zadanej daty
2 SELECT
3     NEXT_DAY (
4         <data>,
5         <numer dnia tygodnia>
6     )
7 FROM
8     <tabela>;
```

Arytmetyka dat:

Do daty można dodać lub odjąć stałą, aby dodać lub odjąć od/do niej dni.

```
1 -- Zmiana daty o pewna ilosc dni
2 SELECT <data>+/-<stala> FROM <tabela>;
```

Można obliczyć różnicę między datami w dniach za pomocą operacji odejmowania.

```
1 -- Zmiana daty o pewna ilosc dni
2 SELECT <data> - <data> FROM DUAL
```

Operowanie na NULL-ach: