Bazy Danych
LABORATORIUM 2

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

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

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

```
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.

```
-- Wyliczenie sredniej pensji wedlug menedzera i departamentu
   , sortowane rosnaco po sredniej pensji
SELECT
  manager_id    AS "Menedzer",
  department_id AS "Departament",
  AVG(salary)   AS "Srednia pensja"
FROM
  employees
GROUP BY
  manager_id,
  department_id
ORDER BY
  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.

```
-- Wyliczenie sredniej pensji pracownikow, zarabiajacych
   ponizej 3000, mniejszej niz 5000, wedlug menedzera i
   departamentu, sortowane rosnaco po sredniej pensji
SELECT
  manager_id    AS "Menedzer",
  department_id AS "Departament",
  AVG(salary)   AS "Srednia pensja"
FROM
  employees
WHERE
  salary < 3000
GROUP BY
  manager_id,
  department_id
HAVING
  AVG(salary) < 5000
ORDER BY
  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.

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

Funkcja TO_DATE
Funkcja TO_DATE konwertuje zadany ciąg znaków o zadanym formacie na datę.

```
SELECT
  TO_DATE(
    <ciag znakow>,
    <format daty>
  )
FROM
  <tabela>;
```

Funkcja ADD_MONTHS:
Funkcja ADD_MONTHS dodaje do zadanej daty zadaną liczbę miesięcy.

```
SELECT
  ADD_MONTHS(
    <data>,
    <ilosc miesiecy>
  )
FROM
  tabela;
```

Funkcja MONTHS_BETWEEN:
Funkcja obliczająca różnicę między dwoma datami w miesiącach.

```
SELECT
  MONTHS_BETWEEN(
    <data 1>,
    <data 2>
  )
FROM
  <tabela>;
```

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

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

Funkcja TO_CHAR:
Funkcja konwertująca zadaną datę na ciąg znaków o zadanym formacie.

```
SELECT
  TO_CHAR(
    <data>,
    <format>
  )
FROM
  <tabela>;
```

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

```
SELECT
  TRUNC(
    <data>,
    <ograniczenie>
  )
FROM
  <tabela>;
```

Funkcja LAST_DAY:
Funkcja zwracająca ostatni dzień miesiąca, obecnego we wskazanej dacie.

```
SELECT
  LAST_DAY(<data>)
FROM
  <tabela>;
```

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

```
-- Zwraca date najblizszego dnia tygodnia od zadanej daty
SELECT
  NEXT_DAY(
    <data>,
    <numer dnia tygodnia>
  )
FROM
  <tabela>;
```

Arytmetyka dat:
Do daty można dodać lub odjąć stałą, aby dodać lub objąć od/do niej dni.

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

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

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

Operator IS NULL:
Operator służący do określania czy wartość w kolumnie jest null-em.

```
SELECT <kolumny> FROM <tabela> WHERE IS NULL;
```

Operator IS NOT NULL:
Operator służący do określania czy wartość w kolumnie nie jest NULL-em.

```
  SELECT <kolumny> FROM <tabela> WHERE IS NOT NULL;
```

Funkcja COALESCE:
Funkcja podmieniająca występujące w zadanych kolumnach wartości NULL na wartość pierwszej z kolumn (albo stałych) z zadanego zbioru, która nie jest NULL-em.

```
-- Gdy manager_id bedzie NULL to zastap go salary badz na
   odwrot
SELECT COALESCE(manager_id, salary) FROM employeees;
```

Funkcja NULLIF:
Funkcja zwracająca NULL, gdy dwie zadane w niej wartości są równe. W przeciwnym wypadku zwraca wartość pierwszą.

```
SELECT   <kolumny>
FROM     <tabela>
WHERE    NULLIF(<kolumna/kolumna>, <kolumna/wartosc>) IS NULL;
```

Porównania z NULL-em:
Porównania z zawierające NULL zawsze dadzą wynik NULL.

Operacje arytmetyczne z NULL-em:
Operacje arytmetyczne, w których co najmniej jeden argument jest NULL-em zwracają NULL.

Funkcje agregujące z NULL-em:
Funkcje agregujące po prostu ignorują komórki zawierające wartości NULL. Przykładowo funkcja COUNT nie zlicza takich komórek.

Funkcje logiczne z NULL-em:
Jeżeli argumentem funkcji logicznej jest NULL to zwraca ona NULL.

Funkcja NVL:
Funkcja podmienia wartość pierwszego argumentu na drugi jeżeli pierwszy jest NULL-em.

```
SELECT
  NVL(<kolumna/wartosc>, <kolumna/wartosc>)
FROM <tabela>;
```

Funkcja NVL2:
Funkcja podmienia wartość pierwszego argumentu na drugi jeżeli pierwszy nie jest NULL-em lub na trzeci gdy pierwszy jest NULL-em.

```
SELECT
  NVL2(
    <kolumna/wartosc>,
    <wartosc gdy pierwszy nie NULL>,
    <wartosc gdy pierwszy NULL>
  )
FROM <tabela>;
```