SYSDATE:

SYSDATE zwraca obecną datę i czas z systemu operacyjnego na którym silnik bazodanowy działa
Przykład:

```
SELECT TO_CHAR(SYSDATE, 'DD-MM-YYYY HH24:MI:SS') "TERAZ"
     FROM DUAL;
```

Wynik:

```
TERAZ
-------------------
17-04-2018 12:45:51
```

MONTHS_BETWEEN:

MONTHS_BETWEEN(date1,date2) – zwraca liczbę miesięcy pomiędzy date1 oraz date2. Jeżeli date1 następuje po date2 wynik jest dodatni. Jeżeli date1 występuje przed date2 wynik jest ujemny. Gdy dni miesiąca zawarte w datach są takie same lub jeśli są to ostatnie dni miesiąca wynik jest liczą całkowitą. W przeciwnym wypadku jest obliczany ułamek bazując na 31 dniowym miesiącu.

Przykład:

```
SELECT MONTHS_BETWEEN
   (TO_DATE('02-02-1995','MM-DD-YYYY'),
    TO_DATE('01-01-1995','MM-DD-YYYY') ) "MIESIACE"
    FROM DUAL;
```

Wynik:

```
    MIESIACE
----------
1.03225806
```

DISTINCT:

Klauzula DISTINCT jest używana w celu usunięcia duplikujących się rekordów z wyniku.

Składnia: `SELECT DISTINCT` *`wyrażenia`* `FROM` *`tabele`* `[WHERE` *`warunki`*`];`

HAVING:

Klauzula HAVING użyta wraz z klauzulą GROUP BY organicza grupy zwracanych wierszy do takich, które spełniają warunek.

Składnia:

```
SELECT wyrażenie1, wyrażenie _n,
       Funkcja_agregująca(wyrażenie agregowane)
FROM tabele
[WHERE warunki]
GROUP BY wyrażenie1, wyrażenie_n
HAVING warunek_HAVING;
```

Przykład:

```
SELECT department, SUM(sales) AS "Sprzedaz calkowita"

FROM order_details

GROUP BY department

HAVING SUM(sales) > 25000;
```

NEXT_DAY:

NEXT_DAY(date,day) zwraca datę pierwszego dnia o nazwie (day), następującego po date.

Przykład:

```
SELECT NEXT_DAY('02-FEB-2001','TUESDAY') "NEXT DAY"
     FROM DUAL;

NEXT DAY
-----------
06-FEB-2001
```

Pierwszy wtorek po 2 lutego 2001.

CASE:

Wyrażenie `CASE` wybiera spośród postawionych warunków spełniony i wykonuje następujący po nim ciąg czynności.

```
CASE { WHEN wyrażenie logiczne THEN {wyrażenie;} ... }...

[ ELSE {wyrażenie;}... ]

END CASE [ nazwa_etykiety ];
```

Przykład:

```
SELECT table_name,
CASE
  WHEN owner='SYS' THEN 'The owner is SYS'
  WHEN owner='SYSTEM' THEN 'The owner is SYSTEM'
  ELSE 'The owner is another value'
END
FROM all_tables;
```

ZADANIA DO WYKONANIA:

1. Oblicz na ile dni upłynęło od wpisania ostatniej oceny (użyj funkcji sysdate). Wyświetl kto ją wpisał i z jakiego przedmiotu.
2. Wyświetl dane wykładowców (imię, nazwisko, tytuł naukowy), którzy w ciągu ostatnich 12 miesięcy wpisywali oceny. Kolumnie nadaj nazwę DANE.
3. Wyświetl minimalna, średnią oraz maksymalną ocenę. Nazwy kolumn wynikowych mają być odpowiednio: MINIMALNA OCENA, ŚREDNIA Z OCEN, MAKSYMALNA OCENA.
4. Wyświetl nazwy przedmiotów oraz wszystkie dane prowadzącego, który wpisał ocenę mającą najstarszą datę.
5. Wyświetl studenta /ów (imię, nazwisko, nr albumu), który ma najwięcej ocen.
6. Wyświetl liczbę studentów w każdej grupie studenckiej na kierunku informatyka.
7. Wyświetl daty trzech najbliższych poniedziałków, przypadających po ostatniej ocenie wpisanej do bazy.
8. Wyświetl nazwiska i imiona (pierwsze dwie kolumny) wykładowców oraz informację czy prowadzą oni przedmiot. Jeżeli dany wykładowca ma w nazwisku ‘sk’, informacja (w trzeciej kolumnie) brzmi: „BRAK PRZEDMIOTU”. W przeciwnym razie „PROWADZI PRZEDMIOT”.
9. Wyświetl dane na temat budynku, w którym odbywa się najwięcej zajęć.
10. Wyświetl nazwę przedmiotu (ów), które mają największą liczbę pkt. ECTS.
11. Wyświetl dane o studentach, którzy studiują na kierunku, na którym jest najmniejsza liczba studentów.
12. Wyświetl nazwę grupy, która ma zajęcia w budynkach „Wydział Inżynierii Elektrycznej i Komputerowej” i „Wydział Inżynierii i Technologii Chemicznej”.
13. Wyświetl salę oraz charakter przedmiotu, który jest wykładany przez tego wykładowcę, który ma zajęcia we wtorek.
14. Wyświetl numer grupy, z której pochodzą studenci, którzy mają zajęcia prowadzone przez prof. dr. hab. inż.