Alex Rogoziński 280522

Programowanie Baz Danych

Laboratorium

Lista 2 - Raport

Spis treści

Zadania rozgrzewkowe A

Zadanie 1

Oracle

```
-------------------------------------------------------------------------------
-- TASK 1
-------------------------------------------------------------------------------
SELECT
    k1.imie "W stadzie przed szefem lub bez incydentu"
FROM Kocury k1
    LEFT JOIN Wrogowie_kocurow wk ON k1.pseudo = wk.pseudo
    INNER JOIN Kocury k2 ON k1.szef = k2.pseudo
WHERE
    k1.w_stadku_od < k2.w_stadku_od
    OR
    wk.pseudo IS NULL
ORDER BY k1.imie;
```

```
W stadzie przed
--------------
LUCEK
MICKA
PUCEK
ZUZIA
```

Liczba krotek: 4

Microsoft

```
-------------------------------------------------------------------------------
-- TASK 1
-------------------------------------------------------------------------------
SELECT
    k1.imie [W stadzie przed szefem lub bez incydentu]
FROM Kocury k1
    LEFT JOIN Wrogowie_kocurow wk ON k1.pseudo = wk.pseudo
    INNER JOIN Kocury k2 ON k1.szef = k2.pseudo
WHERE
    k1.w_stadku_od < k2.w_stadku_od
    OR
    wk.data_incydentu IS NULL
ORDER BY k1.imie;
```

```
W stadzie przed szefem lub bez incydentu
----------------------------------------
LUCEK
MICKA
PUCEK
ZUZIA
((4 rows affected))
```

Zadanie 2

Oracle

```
-------------------------------------------------------------------------------
-- TASK 2
-------------------------------------------------------------------------------
SELECT
    k.pseudo          "Kotka",
    wk.imie_wroga     "jej wrog",
    wk.opis_incydentu "Przewina wroga"
FROM Kocury k
    INNER JOIN Wrogowie_kocurow wk ON k.pseudo = wk.pseudo
WHERE
    k.plec = 'D'
ORDER BY
    k.pseudo;
```

```
Kotka           jej wrog        Przewina wroga
--------------- --------------- --------------------------------------------------
DAMA            KAZIO           CHCIAL OBEDRZEC ZE SKORY
KURKA           BUREK           POGONIL
LASKA           KAZIO           ZLAPAL ZA OGON I ZROBIL WIATRAK
LASKA           DZIKI BILL      POGRYZL ZE LEDWO SIE WYLIZALA
MALA            CHYTRUSEK       ZALECAL SIE
PUSZYSTA        SMUKLA          OBRZUCILA SZYSZKAMI
SZYBKA          GLUPIA ZOSKA    UZYLA KOTA JAKO SCIERKI
UCHO            SWAWOLNY DYZIO  OBRZUCIL KAMIENIAMI
```

Liczba krotek: 8

Microsoft

```
-------------------------------------------------------------------------------
-- TASK 2
-------------------------------------------------------------------------------
SELECT
    k.pseudo          [Kotka],
    wk.imie_wroga     [jej wrog],
    wk.opis_incydentu [Przewina wroga]
FROM Kocury K
    INNER JOIN Wrogowie_kocurow wk ON k.pseudo = wk.pseudo
WHERE
    k.plec = 'D'
ORDER BY
    k.pseudo;
```

```
Kotka       jej wrog        Przewina wroga
----------  --------------  ------------------------------
DAMA        KAZIO           CHCIAL OBEDRZEC ZE SKORY
KURKA       BUREK           POGONIL
LASKA       DZIKI BILL      POGRYZL ZE LEDWO SIE WYLIZALA
LASKA       KAZIO           ZLAPAL ZA OGON I ZROBIL WIATRAK
MALA        CHYTRUSEK       ZALECAL SIE
PUSZYSTA    SMUKLA          OBRZUCILA SZYSZKAMI
SZYBKA      GLUPIA ZOSKA    UZYLA KOTA JAKO SCIERKI
UCHO        SWAWOLNY DYZIO  OBRZUCIL KAMIENIAMI
((8 rows affected))
```

Zadanie 3

Oracle

```
-------------------------------------------------------------------------------------------
-- TASK 3
-------------------------------------------------------------------------------------------
SELECT
    k.pseudo   "Szpieg",
    k.nr_bandy "Banda"
FROM Kocury k
    INNER JOIN Bandy b ON k.pseudo = b.szef_bandy
WHERE
    k.szef = 'TYGRYS'
    AND
    b.szef_bandy != 'TYGRYS';
```

```
Szpieg                Banda
--------------- ----------
ZOMBI                  3
LYSY                   2
RAFA                   4
```

Liczba krotek: 3

Microsoft

```
-------------------------------------------------------------------------------------------
-- TASK 3
-------------------------------------------------------------------------------------------
SELECT
    k.pseudo   [Szpieg],
    k.nr_bandy [Banda]
FROM Kocury k
    INNER JOIN Bandy b ON k.pseudo = b.szef_bandy
WHERE
    k.szef = 'TYGRYS'
    AND
    b.szef_bandy != 'TYGRYS';
```

```
Szpieg      Banda
----------  ----------
LYSY        2
RAFA        4
ZOMBI       3
((3 rows affected))
```

Zadanie 4

Oracle

```
-------------------------------------------------------------------------------------------
-- TASK 4
-------------------------------------------------------------------------------------------
SELECT
    NVL(k1.pseudo, 'Brak przelozonego') "Przelozony",
    NVL(k2.pseudo, 'Brak podwladnego')  "Podwladny"
FROM Kocury k1
    FULL JOIN Kocury k2 ON k1.pseudo = k2.szef
WHERE
    NVL(k1.plec, 'M') = 'M'
    AND
    NVL(k2.plec, 'M') = 'M'
ORDER BY
    "Przelozony";
```

```
Przelozony        Podwladny
----------------- ----------------
BOLEK             Brak podwladnego
Brak przelozonego TYGRYS
LYSY              PLACEK
LYSY              RURA
MALY              Brak podwladnego
MAN               Brak podwladnego
PLACEK            Brak podwladnego
RAFA              MAN
RAFA              MALY
RURA              Brak podwladnego
TYGRYS            LYSY

Przelozony        Podwladny
----------------- ----------------
TYGRYS            RAFA
TYGRYS            BOLEK
TYGRYS            ZOMBI
ZERO              Brak podwladnego
```

Liczba krotek: 15

Microsoft

```
-------------------------------------------------------------------------------------------
-- TASK 4
-------------------------------------------------------------------------------------------
SELECT
    ISNULL(k1.pseudo, 'Brak przelozonego') [Przelozony],
    ISNULL(k2.pseudo, 'Brak podwladnego')  [Podwladny]
FROM Kocury k1
    FULL JOIN Kocury k2 ON k1.pseudo = k2.szef
WHERE
    ISNULL(k1.plec, 'M') = 'M'
    AND
    ISNULL(k2.plec, 'M') = 'M'
ORDER BY
    'Przelozony';
```

Przelozony	Podwladny
BOLEK	Brak podwladneg
Brak przelozone	TYGRYS
LYSY	PLACEK
LYSY	RURA
MALY	Brak podwladneg
MAN	Brak podwladneg
PLACEK	Brak podwladneg
RAFA	MALY
RAFA	MAN
RURA	Brak podwladneg
TYGRYS	BOLEK
TYGRYS	LYSY
TYGRYS	RAFA
TYGRYS	ZOMBI
ZERO	Brak podwladneg

((15 rows affected))

Zadanie 5

Oracle

```
-------------------------------------------------------------------------------
-- TASK 5
-------------------------------------------------------------------------------
SELECT DISTINCT
    k1.pseudo                                                "PSEUDO",
    k1.przydzial_myszy                                       "PRZYDZIAL_MYSZY",
    k1.SUM_W_BANDZIE                                         "SUM_W_BANDZIE",
    ROUND(k1.przydzial_myszy * 100 / k1.SUM_W_BANDZIE, 0) "PROC_W_BANDZIE"
FROM (
    SELECT
        k.pseudo,
        k.przydzial_myszy,
        SUM(k.przydzial_myszy) OVER (PARTITION BY k.nr_bandy) "SUM_W_BANDZIE"
    FROM Kocury k
        INNER JOIN Bandy b ON k.nr_bandy = b.nr_bandy
    WHERE
        b.teren IN ('POLE', 'CALOSC')
) k1
    INNER JOIN Wrogowie_kocurow wk ON k1.pseudo = wk.pseudo
    INNER JOIN Wrogowie w ON wk.imie_wroga = w.imie_wroga
WHERE
    w.stopien_wrogosci > 5
ORDER BY
    k1.SUM_W_BANDZIE;
```

PSEUDO	PRZYDZIAL_MYSZY	SUM_W_BANDZIE	PROC_W_BANDZIE
BOLEK	50	200	25
TYGRYS	103	200	52
LASKA	24	284	8
RURA	56	284	20

Liczba krotek: 4

Microsoft

```
-- TASK 5
SELECT DISTINCT
    k1.pseudo                                   [PSEUDO],
    k1.przydzial_myszy                          [PRZYDZIAL_MYSZY],
    k1.SUM_W_BANDZIE                            [SUM_W_BANDZIE],
    k1.przydzial_myszy * 100 / k1.SUM_W_BANDZIE [PROC_W_BANDZIE]
FROM (
    SELECT
        k.pseudo,
        k.przydzial_myszy,
        SUM(k.przydzial_myszy) OVER (PARTITION BY k.nr_bandy) [SUM_W_BANDZIE]
    FROM Kocury k
        INNER JOIN Bandy b ON k.nr_bandy = b.nr_bandy
    WHERE
        b.teren IN ('POLE', 'CALOSC')
) k1
    INNER JOIN Wrogowie_kocurow wk ON wk.pseudo = k1.pseudo
    INNER JOIN Wrogowie w ON w.imie_wroga = wk.imie_wroga
WHERE
    w.stopien_wrogosci > 5
ORDER BY
    k1.SUM_W_BANDZIE;
```

PSEUDO	PRZYDZIAL_MYSZY	SUM_W_BANDZIE	PROC_W_BANDZIE
BOLEK	50	200	25
TYGRYS	103	200	51
LASKA	24	284	8
RURA	56	284	19

Ogólne spostrzeżenia po wykonaniu części *Zadania rozgrzewkowe A*

Zadania z tej części listy opierały się głównie o zastosowanie któregoś rodzaju operacji `JOIN` między tabelami. Wszystkie kwerendy były dość krótkie, z wyjątkiem ostatniej, której wykonanie bez wykorzystania podzapytania nie było możliwe. Gdy nie korzystano z podzapytania, problemem było ograniczenie zbioru kocurów, których przydziały myszy należało sumować, jedynie do tych, które spełniały warunek nadrzędny zadania. Dopiero zastosowanie podzapytania pomogło rozwiązać problem, z uwagi na dostęp do pełnej tabeli przez podzapytanie, którego wynik był obliczany niezależnie od reszty kwerendy nadrzędnej.

Zadania rozgrzewkowe B

Zadanie 6

Oracle

```
-------------------------------------------------------------------------------------------
-- TASK 6
-------------------------------------------------------------------------------------------
WITH Kocury_typy AS (
    SELECT
        pseudo,
        przydzial_myszy,
        nr_bandy,
        CASE
            WHEN przydzial_myszy > AVG(przydzial_myszy) OVER ()
                THEN 'Prominent'
            WHEN przydzial_myszy = MIN(przydzial_myszy) OVER (PARTITION BY nr_bandy)
                THEN 'Szarak'
        END "Typ"
    FROM Kocury
)
SELECT
    pseudo          "Pseudonim",
    przydzial_myszy "Zjada",
    nr_bandy        "Banda",
    "Typ"
FROM
    Kocury_typy
WHERE
    "Typ" IS NOT NULL
ORDER BY
    "Typ", nr_bandy;
```

```
Pseudonim            Zjada      Banda Typ
--------------- ---------- ---------- ---------
TYGRYS                  103          1 Prominent
LYSY                     72          2 Prominent
SZYBKA                   65          2 Prominent
RURA                     56          2 Prominent
PLACEK                   67          2 Prominent
KURKA                    61          3 Prominent
ZOMBI                    75          3 Prominent
RAFA                     65          4 Prominent
MALA                     22          1 Szarak
LASKA                    24          2 Szarak
PUSZYSTA                 20          3 Szarak

Pseudonim            Zjada      Banda Typ
--------------- ---------- ---------- ---------
UCHO                     40          4 Szarak
MALY                     40          4 Szarak
```

Liczba krotek: 13

Microsoft

```
-------------------------------------------------------------------------------------------
-- TASK 6
-------------------------------------------------------------------------------------------
WITH Kocury_typy AS (
    SELECT
        pseudo,
        przydzial_myszy,
        nr_bandy,
        CASE
            WHEN przydzial_myszy > AVG(przydzial_myszy) OVER ()
                THEN 'Prominent'
            WHEN przydzial_myszy = MIN(przydzial_myszy) OVER (PARTITION BY nr_bandy)
                THEN 'Szarak'
        END [Typ]
    FROM Kocury
)
SELECT
    pseudo          [Pseudonim],
    przydzial_myszy [Zjada],
    nr_bandy        [Banda],
    Typ
FROM Kocury_typy
WHERE
    Typ IS NOT NULL
ORDER BY
    przydzial_myszy DESC,
    Typ;
```

```
Pseudonim   Zjada       Banda       Typ
----------  ----------  ----------  ----------
TYGRYS      103         1           Prominent
ZOMBI       75          3           Prominent
LYSY        72          2           Prominent
PLACEK      67          2           Prominent
SZYBKA      65          2           Prominent
RAFA        65          4           Prominent
KURKA       61          3           Prominent
RURA        56          2           Prominent
MALY        40          4           Szarak
UCHO        40          4           Szarak
LASKA       24          2           Szarak
MALA        22          1           Szarak
PUSZYSTA    20          3           Szarak
((13 rows affected))
```

Zadanie 7

Oracle

```
-------------------------------------------------------------------------------
-- TASK 7
-------------------------------------------------------------------------------
WITH Kocury_sr_przydz AS (
    SELECT
        pseudo,
        AVG(przydzial_myszy) OVER (PARTITION BY nr_bandy) "Srednio w bandzie",
        plec
    FROM Kocury
)
SELECT
    pseudo "Kot",
    "Srednio w bandzie"
FROM Kocury_sr_przydz
WHERE
    plec = 'M';
```

Kot	Srednio w bandzie
TYGRYS	50
BOLEK	50
PLACEK	56,8
RURA	56,8
LYSY	56,8
ZERO	49,75
ZOMBI	49,75
MALY	49,4
MAN	49,4
RAFA	49,4

Liczba krotek: 10

Microsoft

```
-------------------------------------------------------------------------------
-- TASK 7
-------------------------------------------------------------------------------
WITH Kocury_sr_przydz AS (
    SELECT
        pseudo,
        AVG(przydzial_myszy) OVER (PARTITION BY nr_bandy) [Srednio w bandzie],
        plec
    FROM Kocury
)
SELECT
    pseudo [Kot],
    [Srednio w bandzie]
FROM Kocury_sr_przydz
WHERE
    plec = 'M';
```

```
Kot         Srednio w bandzie
----------  -----------------
BOLEK       50
TYGRYS      50
PLACEK      56
RURA        56
LYSY        56
ZERO        49
ZOMBI       49
RAFA        49
MALY        49
MAN         49
((10 rows affected))
```

Zadanie 8

Oracle

```
-------------------------------------------------------------------------------------------
-- TASK 8
-------------------------------------------------------------------------------------------
-- Wersja z wyswietlaniem przedzialu
SELECT
    b.nr_bandy             "Lepsze bandy",
    AVG(k.przydzial_myszy) "Sredni przydzial w bandzie",
    sr_wszys."Sredni przydzial"
FROM Bandy b
    INNER JOIN Kocury k ON b.nr_bandy = k.nr_bandy
    CROSS JOIN (
        SELECT AVG(przydzial_myszy) "Sredni przydzial"
        FROM Kocury
    ) sr_wszys
GROUP BY
    b.nr_bandy,
    sr_wszys."Sredni przydzial"
HAVING "Sredni przydzial w bandzie" > sr_wszys."Sredni przydzial";
```

```
Lepsze bandy Sredni przydzial w bandzie Sredni przydzial
------------ ------------------------- ----------------
           2                      56,8       51,6666667
```

Liczba krotek: 1

```
-- Wersja bez wyswietlania przedzialu
SELECT
    b.nr_bandy             "Lepsze bandy",
    AVG(k.przydzial_myszy) "Sredni przydzial w bandzie"
FROM Bandy b
    INNER JOIN Kocury k ON b.nr_bandy = k.nr_bandy
GROUP BY b.nr_bandy
HAVING "Sredni przydzial w bandzie" > (SELECT AVG(przydzial_myszy) FROM Kocury);
```

```
Lepsze bandy Sredni przydzial w bandzie
------------ --------------------------
           2                       56,8
```

Liczba krotek: 1

```sql
-------------------------------------------------------------------------------------------
-- TASK 8
-------------------------------------------------------------------------------------------
-- Wersja z wyswietlaniem przedzialu
SELECT
    b.nr_bandy              [Lepsze bandy],
    AVG(k.przydzial_myszy) [Srednio w bandzie],
    sr_wszys.[Sredni przydzial]
FROM Bandy b
    INNER JOIN Kocury k ON b.nr_bandy = k.nr_bandy
    CROSS JOIN (
        SELECT AVG(przydzial_myszy) [Sredni przydzial]
        FROM Kocury
    ) sr_wszys
GROUP BY
    b.nr_bandy,
    sr_wszys.[Sredni przydzial]
HAVING AVG(k.przydzial_myszy) > sr_wszys.[Sredni przydzial];
```

```
Lepsze bandy  Srednio w bandzie  Sredni przydzial
------------  -----------------  ---------------
2             56                 51
((1 row affected))
```

```sql
-- Wersja bez wyswietlania przedzialu
SELECT
    b.nr_bandy              [Lepsze bandy],
    AVG(k.przydzial_myszy) [Sredni przydzial w bandzie]
FROM Bandy b
    INNER JOIN Kocury k ON b.nr_bandy = k.nr_bandy
GROUP BY b.nr_bandy
HAVING AVG(k.przydzial_myszy) > (SELECT AVG(przydzial_myszy) FROM Kocury);
```

```
Lepsze bandy  Sredni przydzial w bandzie
------------  --------------------------
2             56
((1 row affected))
```

Zadanie 9

Oracle

```
-------------------------------------------------------------------------------------------
-- TASK 9
-------------------------------------------------------------------------------------------
SELECT
    UPPER(TO_CHAR(w_stadku_od, 'Month')) "Miesiac",
    COUNT(*)                             "Liczba rekrutow"
FROM Kocury
GROUP BY
    EXTRACT(MONTH FROM w_stadku_od),
    TO_CHAR(w_stadku_od, 'Month')
ORDER BY
    EXTRACT(MONTH FROM w_stadku_od);
```

```
Miesiac                            Liczba rekrutow
---------------------------------- ---------------
JANUARY                                          3
FEBRUARY                                         1
MARCH                                            2
MAY                                              2
JULY                                             2
AUGUST                                           1
SEPTEMBER                                        2
OCTOBER                                          2
NOVEMBER                                         2
DECEMBER                                         1
```

Liczba krotek: 10

Microsoft

```
-------------------------------------------------------------------------------------------
-- TASK 9
-------------------------------------------------------------------------------------------
SELECT
    UPPER(DATENAME(month, w_stadku_od)) [Miesiac],
    COUNT(*)                            [Liczba rekrutow]
FROM Kocury
GROUP BY
    MONTH(w_stadku_od),
    DATENAME(month, w_stadku_od)
ORDER BY
    MONTH(w_stadku_od);
```

```
Miesiac     Liczba rekrutow
----------  ---------------
JANUARY     3
FEBRUARY    1
MARCH       2
MAY         2
JULY        2
AUGUST      1
SEPTEMBER   2
OCTOBER     2
NOVEMBER    2
DECEMBER    1
((10 rows affected))
```

Zadanie 10

Oracle

```
-------------------------------------------------------------------------------------------
-- TASK 10
-------------------------------------------------------------------------------------------
SELECT
    f.funkcja "FUNKCJA",
    SUM (
        CASE
            WHEN b.nazwa = 'CZARNI RYCERZE'
                THEN k.przydzial_myszy + NVL(k.myszy_extra, 0)
        END
    ) "Banda CZARNI RYCERZE",
    SUM(
        CASE

            WHEN b.nazwa = 'BIALI LOWCY'
                THEN k.przydzial_myszy + NVL(k.myszy_extra, 0)
        END
    ) "Banda BIALI LOWCY"
FROM Funkcje f
    INNER JOIN Kocury k ON f.funkcja = k.funkcja
    LEFT JOIN Bandy b ON k.nr_bandy = b.nr_bandy
WHERE
    f.funkcja != 'SZEFUNIO'
GROUP BY f.funkcja
ORDER BY f.funkcja;
```

FUNKCJA	Banda CZARNI RYCERZE	Banda BIALI LOWCY
BANDZIOR	93	88
DZIELCZY		
KOT		43
LAPACZ	56	
LOWCZY	132	61
MILUSIA	52	55

Liczba krotek: 6

Microsoft

```
-------------------------------------------------------------------------------------------
-- TASK 10
-------------------------------------------------------------------------------------------
SELECT
    f.funkcja [FUNKCJA],
    SUM(
        CASE
            WHEN b.nazwa = 'CZARNI RYCERZE'
                THEN k.przydzial_myszy + ISNULL(k.myszy_extra, 0)
        END
    ) [Banda CZARNI RYCERZE],
    SUM(
        CASE
            WHEN b.nazwa = 'BIALI LOWCY'
                THEN k.przydzial_myszy + ISNULL(k.myszy_extra, 0)
        END
    ) [Banda BIALI LOWCY]
FROM Funkcje f
    INNER JOIN Kocury k ON f.funkcja = k.funkcja
    LEFT JOIN Bandy b ON k.nr_bandy = b.nr_bandy
WHERE
    f.funkcja != 'SZEFUNIO'
GROUP BY f.funkcja
ORDER BY f.funkcja;
```

```
FUNKCJA     Banda CZARNI RYCERZE  Banda BIALI LOWCY
----------  --------------------  -----------------
BANDZIOR    93                    88
DZIELCZY    NULL                  NULL
KOT         NULL                  43
LAPACZ      56                    NULL
LOWCZY      132                   61
MILUSIA     52                    55
((6 rows affected))
```

Zadanie 11

Oracle

```
-------------------------------------------------------------------------------------------
-- TASK 11
-------------------------------------------------------------------------------------------
SELECT
    f.funkcja "FUNKCJA",
    k.plec    "P",
    SUM (
        CASE
            WHEN b.nazwa = 'CZARNI RYCERZE'
                THEN k.przydzial_myszy + NVL(k.myszy_extra, 0)
        END
    ) "Banda CZARNI RYCERZE",
    SUM(
        CASE

            WHEN b.nazwa = 'BIALI LOWCY'
                THEN k.przydzial_myszy + NVL(k.myszy_extra, 0)
        END
    ) "Banda BIALI LOWCY",
    COUNT(*) "Liczba kotow"
FROM Funkcje f
    INNER JOIN Kocury k ON f.funkcja = k.funkcja
    LEFT JOIN Bandy b ON k.nr_bandy = b.nr_bandy
WHERE
    f.funkcja != 'SZEFUNIO'
GROUP BY
    f.funkcja,
    k.plec
ORDER BY
    f.funkcja;
```

FUNKCJA	P	Banda CZARNI RYCERZE	Banda BIALI LOWCY	Liczba kotow
BANDZIOR	M	93	88	2
DZIELCZY	M			1
KOT	D			1
KOT	M		43	2
LAPACZ	D			1
LAPACZ	M	56		2
LOWCZY	D	65	61	2
LOWCZY	M	67		2
MILUSIA	D	52	55	4

Liczba krotek: 9

Microsoft

```
-- TASK 11
SELECT
    f.funkcja [FUNKCJA],
    k.plec [P],
    SUM(
        CASE
            WHEN b.nazwa = 'CZARNI RYCERZE'
                THEN k.przydzial_myszy + ISNULL(k.myszy_extra, 0)
        END
    ) [Banda CZARNI RYCERZE],
    SUM(
        CASE
            WHEN b.nazwa = 'BIALI LOWCY'
                THEN k.przydzial_myszy + ISNULL(k.myszy_extra, 0)
        END
    ) [Banda BIALI LOWCY],
    COUNT(*) [Liczba kotow]
FROM Funkcje f
    INNER JOIN Kocury k ON f.funkcja = k.funkcja
    LEFT JOIN Bandy b ON k.nr_bandy = b.nr_bandy
WHERE
    f.funkcja != 'SZEFUNIO'
GROUP BY
    f.funkcja,
    k.plec
ORDER BY f.funkcja;
```

FUNKCJA	P	Banda CZARNI RYCERZE	Banda BIALI LOWCY	Liczba kotow
BANDZIOR	M	93	88	2
DZIELCZY	M	NULL	NULL	1
KOT	D	NULL	NULL	1
KOT	M	NULL	43	2
LAPACZ	D	NULL	NULL	1
LAPACZ	M	56	NULL	2
LOWCZY	D	65	61	2
LOWCZY	M	67	NULL	2
MILUSIA	D	52	55	4

((9 rows affected))

Ogólne spostrzeżenia po wykonaniu części *Zadania rozgrzewkowe B*

Zadania z tej części były już zdecydowanie bardziej rozbudowane. Niektóre z nich łatwiej było mi wykonać z wykorzystaniem CTE, a niektóre z wykorzystaniem klasycznych podzapytań. W tej części bardziej zauważalne były różnice między dialektami Oracle SQL i Transaction-SQL, jak np. możliwość wykorzystania nazwy kolumny nadanej w kwerendzie (widoczne w zadaniu 8.), czy bardziej złożone jak np. różne metody ekstrahowania miesiąca z daty i wypisywania go jako wartości tekstowej, a nie jedynie liczbowej (kolejno: `TO_CHAR(<data>, 'month')` i `EXTRACT(MONTH FROM <data>)` oraz `DATENAME(month, <data>)` i `MONTH(<data>)`).

Część Oracle

Zadanie 12

```
-- TASK 12
SELECT
    k.pseudo          "POLUJE W POLU",
    k.przydzial_myszy "PRZYDZIAL MYSZY",
    b.nazwa           "BANDA"
FROM Kocury k
    INNER JOIN Bandy b ON k.nr_bandy = b.nr_bandy
WHERE
    b.teren IN ('POLE', 'CALOSC')
    AND
    k.przydzial_myszy > 50
ORDER BY
    k.przydzial_myszy DESC;
```

POLUJE W POLU	PRZYDZIAL MYSZY	BANDA
TYGRYS	103	SZEFOSTWO
LYSY	72	CZARNI RYCERZE
PLACEK	67	CZARNI RYCERZE
SZYBKA	65	CZARNI RYCERZE
RURA	56	CZARNI RYCERZE

Liczba krotek: 5

Zadanie 13

```
-- TASK 13
SELECT
    k1.imie                                "IMIE",
    TO_CHAR(k1.w_stadku_od, 'YYYY-MM-DD') "POLUJE OD"
FROM Kocury k1
    INNER JOIN Kocury k2 ON k2.imie = 'JACEK'
WHERE
    k1.w_stadku_od < k2.w_stadku_od
ORDER BY
    k1.w_stadku_od DESC;
```

IMIE	POLUJE OD
MELA	2008-11-01
KSAWERY	2008-07-12
BELA	2008-02-01
PUNIA	2008-01-01
PUCEK	2006-10-15
RUDA	2006-09-17
BOLEK	2006-08-15
ZUZIA	2006-07-21
KOREK	2004-03-16
CHYTRY	2002-05-05
MRUCZEK	2002-01-01

Liczba krotek: 11

Zadanie 14

a)

```
-------------------------------------------------------------------------------
-- TASK 14
-------------------------------------------------------------------------------
-- a)
SELECT
    k1.imie           "Imie",
    k1.funkcja        "Funkcja",
    NVL(k2.imie, ' ') "Szef 1",
    NVL(k3.imie, ' ') "Szef 2",
    NVL(k4.imie, ' ') "Szef 3"
FROM Kocury k1
    LEFT JOIN Kocury k2 ON k1.szef = k2.pseudo
    LEFT JOIN Kocury k3 ON k2.szef = k3.pseudo
    LEFT JOIN Kocury k4 ON k3.szef = k4.pseudo
WHERE
    k1.funkcja IN ('KOT', 'MILUSIA');
```

Imie	Funkcja	Szef 1	Szef 2	Szef 3
LUCEK	KOT	PUNIA	KOREK	MRUCZEK
SONIA	MILUSIA	KOREK	MRUCZEK	
BELA	MILUSIA	BOLEK	MRUCZEK	
LATKA	KOT	PUCEK	MRUCZEK	
DUDEK	KOT	PUCEK	MRUCZEK	
MICKA	MILUSIA	MRUCZEK		
RUDA	MILUSIA	MRUCZEK		

Liczba krotek: 7

b)

```sql
-- b)
WITH Drzewo AS (
    SELECT
        imie    "Imie",
        funkcja "Funkcja",
        CONNECT_BY_ROOT imie AS szef_podst,
        LEVEL - 1 AS lvl
    FROM Kocury
    WHERE
        funkcja IN ('KOT', 'MILUSIA')
    CONNECT BY PRIOR pseudo = szef
)
SELECT *
FROM Drzewo
PIVOT (
    MAX(szef_podst)
    FOR lvl IN (1 AS "Szef 1", 2 AS "Szef 2", 3 AS "Szef 3")
);
```

Imie	Funkcja	Szef 1	Szef 2	Szef 3
LUCEK	KOT	PUNIA	KOREK	MRUCZEK
BELA	MILUSIA	BOLEK	MRUCZEK	
LATKA	KOT	PUCEK	MRUCZEK	
DUDEK	KOT	PUCEK	MRUCZEK	
MICKA	MILUSIA	MRUCZEK		
RUDA	MILUSIA	MRUCZEK		
SONIA	MILUSIA	KOREK	MRUCZEK	

Liczba krotek: 7

c)

```sql
-- c)
SELECT
    imie    "Imie",
    funkcja "Funkcja",
    LTRIM(
        REVERSE(
            RTRIM(
                SYS_CONNECT_BY_PATH(REVERSE(RPAD(imie, 12, ' ')), ' | '),
                imie
            )
        ),
        '| '
    )     "Imiona kolejnych szefow"
FROM Kocury
WHERE
    funkcja IN ('KOT', 'MILUSIA')
START WITH szef IS NULL
CONNECT BY PRIOR pseudo = szef;
```

Imie	Funkcja	Imiona kolejnych szefow		
MICKA	MILUSIA	MRUCZEK		
BELA	MILUSIA	BOLEK	MRUCZEK	
RUDA	MILUSIA	MRUCZEK		
DUDEK	KOT	PUCEK	MRUCZEK	
LATKA	KOT	PUCEK	MRUCZEK	
LUCEK	KOT	PUNIA	KOREK	MRUCZEK
SONIA	MILUSIA	KOREK	MRUCZEK	

Liczba krotek: 7

Zadanie 15

```
-- TASK 15
SELECT
    k.imie                                  "Imie kotki",
    b.nazwa                                 "Nazwa bandy",
    wk.imie_wroga                           "Imie wroga",
    w.stopien_wrogosci                      "Poziom wroga",
    TO_CHAR(wk.data_incydentu, 'YYYY-MM-DD') "Data inc."
FROM Kocury k
    INNER JOIN Bandy b ON k.nr_bandy = b.nr_bandy
    INNER JOIN Wrogowie_kocurow wk ON k.pseudo = wk.PSEUDO
    INNER JOIN Wrogowie w ON wk.imie_wroga = w.imie_wroga
WHERE
    k.plec = 'D'
    AND
    wk.data_incydentu > '2007-01-01'
ORDER BY
    k.imie;
```

Imie kotki	Nazwa bandy	Imie wroga	Poziom wroga	Data inc.
BELA	CZARNI RYCERZE	KAZIO	10	2009-01-07
BELA	CZARNI RYCERZE	DZIKI BILL	10	2008-12-12
LATKA	LACIACI MYSLIWI	SWAWOLNY DYZIO	7	2011-07-14
MELA	LACIACI MYSLIWI	KAZIO	10	2009-02-07
PUNIA	BIALI LOWCY	BUREK	4	2010-12-14
RUDA	SZEFOSTWO	CHYTRUSEK	5	2007-03-07
SONIA	BIALI LOWCY	SMUKLA	1	2010-11-19

Liczba krotek: 7

Zadanie 16

```
-- TASK 16
SELECT
   b.nazwa                   "Nazwa bandy",
   COUNT(DISTINCT k.pseudo) "Koty z wrogami"
FROM Bandy b
    INNER JOIN Kocury k ON k.nr_bandy = b.nr_bandy
    INNER JOIN Wrogowie_kocurow wk ON wk.pseudo = k.pseudo
GROUP BY
    b.nazwa
ORDER BY
    b.nazwa;
```

Nazwa bandy	Koty z wrogami
BIALI LOWCY	3
CZARNI RYCERZE	5
LACIACI MYSLIWI	4
SZEFOSTWO	3

Liczba krotek: 4

Zadanie 17

```
-- TASK 17
SELECT
    k.funkcja "Funkcja",
    k.pseudo  "Pseudonim kota",
    COUNT(*)  "Liczba wrogow"
FROM Kocury k
    INNER JOIN Wrogowie_kocurow wk ON wk.pseudo = k.pseudo
GROUP BY
    k.funkcja, k.pseudo
HAVING
    COUNT(*) > 1;
```

Funkcja	Pseudonim kota	Liczba wrogow
SZEFUNIO	TYGRYS	2
DZIELCZY	BOLEK	2
MILUSIA	LASKA	2

Liczba krotek: 3

Zadanie 18

```
-------------------------------------------------------------------------------
-- TASK 18
-------------------------------------------------------------------------------
SELECT
    imie "IMIE",
    12 * (przydzial_myszy + myszy_extra) "DAWKA ROCZNA",
    'powyzej 864' "DAWKA"
FROM Kocury
WHERE
    myszy_extra IS NOT NULL
    AND
    12 * (przydzial_myszy + myszy_extra) > 864

UNION

SELECT
    imie                                "IMIE",
    12 * (przydzial_myszy + myszy_extra) "DAWKA ROCZNA",
    LPAD('864', 11, ' ')                "DAWKA"
FROM Kocury
WHERE
    myszy_extra IS NOT NULL
    AND
    12 * (przydzial_myszy + myszy_extra) = 864

UNION

SELECT
    imie                                "IMIE",
    12 * (przydzial_myszy + myszy_extra) "DAWKA ROCZNA",
    'ponizej 864'                       "DAWKA"
FROM Kocury
WHERE
    myszy_extra IS NOT NULL
    AND
    12 * (przydzial_myszy + myszy_extra) < 864

ORDER BY
    "DAWKA ROCZNA" DESC;
```

IMIE	DAWKA ROCZNA	DAWKA
MRUCZEK	1632	powyzej 864
BOLEK	1116	powyzej 864
KOREK	1056	powyzej 864
MICKA	864	864
RUDA	768	ponizej 864
SONIA	660	ponizej 864
BELA	624	ponizej 864

Liczba krotek: 7

Zadanie 19

a)

```
-------------------------------------------------------------------------------
-- TASK 19
-------------------------------------------------------------------------------
-- a)
SELECT
    b.nr_bandy "NR BANDY",
    b.nazwa    "NAZWA",
    b.teren    "TEREN"
FROM Bandy b
    LEFT JOIN Kocury k ON k.nr_bandy = b.nr_bandy
WHERE k.nr_bandy IS NULL;
```

NR BANDY	NAZWA	TEREN
5	ROCKERSI	ZAGRODA

Liczba krotek: 1

b)

```
-- b)
SELECT
    nr_bandy "NR BANDY",
    nazwa    "NAZWA",
    teren    "TEREN"
FROM Bandy

MINUS

SELECT
    b.nr_bandy "NR BANDY",
    b.nazwa    "NAZWA",
    b.teren    "TEREN"
FROM Bandy b
    INNER JOIN Kocury k ON k.nr_bandy = b.nr_bandy;
```

NR BANDY	NAZWA	TEREN
5	ROCKERSI	ZAGRODA

Liczba krotek: 1

Zadanie 20

```
-------------------------------------------------------------------------------
-- TASK 20
-------------------------------------------------------------------------------
WITH Max_przydzial AS (
    SELECT *
    FROM (
        SELECT
            k.przydzial_myszy AS przydzial
        FROM Kocury k
            INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
        WHERE
            k.funkcja = 'MILUSIA'
            /*
               Aby kwerenda zwracała wynik identyczny jak w przykładzie,
               nalezy pominąc nastepny warunek.
            */
            AND
            b.teren = 'SAD'
        ORDER BY
            przydzial DESC
    )
    WHERE
        ROWNUM = 1
)
SELECT
    k.imie            "IMIE",
    k.funkcja         "FUNKCJA",
    k.przydzial_myszy "PRZYDZIAL MYSZY"
FROM Kocury k
    CROSS JOIN Max_przydzial mp
WHERE
    k.przydzial_myszy >= 3 * mp.przydzial
ORDER BY
    k.przydzial_myszy;
```

IMIE	FUNKCJA	PRZYDZIAL MYSZY
PUNIA	LOWCZY	61
PUCEK	LOWCZY	65
ZUZIA	LOWCZY	65
JACEK	LOWCZY	67
BOLEK	BANDZIOR	72
KOREK	BANDZIOR	75
MRUCZEK	SZEFUNIO	103

Liczba krotek: 7

Wynik dla pominiętego drugiego warunku:

IMIE	FUNKCJA	PRZYDZIAL MYSZY
KOREK	BANDZIOR	75
MRUCZEK	SZEFUNIO	103

Liczba krotek: 2

Zadanie 21

```
--------------------------------------------------------------------------------
-- TASK 21
--------------------------------------------------------------------------------
WITH Avg_przydzialy AS (
    SELECT
        funkcja,
        AVG(przydzial_myszy + NVL(myszy_extra, 0))                                  AS przydzial,
        DENSE_RANK() OVER (ORDER BY AVG(przydzial_myszy + NVL(myszy_extra, 0)))     AS min_rank,
        DENSE_RANK() OVER (ORDER BY AVG(przydzial_myszy + NVL(myszy_extra, 0)) DESC) AS max_rank
    FROM Kocury
    WHERE
        funkcja != 'SZEFUNIO'
    GROUP BY
        funkcja
)
SELECT
    funkcja             "Funkcja",
    ROUND(przydzial, 0) "Srednio najw. i najm. myszy"
FROM Avg_przydzialy
WHERE
    min_rank = 1
    OR
    max_rank = 1;
```

```
Funkcja    Srednio najw. i najm. myszy
---------- ---------------------------
BANDZIOR                            91
KOT                                 41
```

Liczba krotek: 2

Zadanie 22

Poniższe wyniki są dla podanego $n = 6$.

a)

```
-------------------------------------------------------------------------------
-- TASK 22
-------------------------------------------------------------------------------
-- a)
SELECT
    k1.pseudo                                   "PSEUDO",
    k1.przydzial_myszy + NVL(k1.myszy_extra, 0) "ZJADA"
FROM Kocury k1
WHERE &n >= (
    SELECT
        COUNT(*) AS cnt
    FROM Kocury k2
    WHERE
        k1.przydzial_myszy + NVL(k1.myszy_extra, 0) < k2.przydzial_myszy + NVL(k2.myszy_extra,
0)
)
ORDER BY
    "ZJADA" DESC;
```

PSEUDO	ZJADA
TYGRYS	136
LYSY	93
ZOMBI	88
LOLA	72
PLACEK	67
RAFA	65
SZYBKA	65

Liczba krotek: 7

b)

```sql
-- b)
WITH Przydzialy AS (
    SELECT DISTINCT
        przydzial_myszy + NVL(myszy_extra, 0) AS przydz
        FROM Kocury
        ORDER BY
            przydz DESC
)
SELECT
    pseudo                              "PSEUDO",
    przydzial_myszy + NVL(myszy_extra, 0) "ZJADA"
FROM Kocury
WHERE
    przydzial_myszy + NVL(myszy_extra, 0) IN (
        SELECT *
        FROM Przydzialy
        WHERE ROWNUM <= &n
    );
```

PSEUDO	ZJADA
TYGRYS	136
LYSY	93
ZOMBI	88
LOLA	72
PLACEK	67
SZYBKA	65
RAFA	65

Liczba krotek: 7

c)

```sql
-- c)
SELECT
    k1.pseudo                                       "PSEUDO",
    AVG(k1.przydzial_myszy + NVL(k1.myszy_extra, 0)) "ZJADA"
FROM Kocury k1
    LEFT JOIN Kocury k2 ON k1.przydzial_myszy + NVL(k1.myszy_extra, 0) < k2.przydzial_myszy +
NVL(k2.myszy_extra, 0)
GROUP BY
    k1.pseudo
HAVING
    COUNT(*) <= &n
ORDER BY
    "ZJADA" DESC;
```

PSEUDO	ZJADA
TYGRYS	136
LYSY	93
ZOMBI	88
LOLA	72
PLACEK	67
RAFA	65
SZYBKA	65

Liczba krotek: 7

d)

```sql
-- d)
SELECT
    pseudo   "PSEUDO",
    zjada    "ZJADA"
FROM (
    SELECT
        pseudo,
        przydzial_myszy + NVL(myszy_extra, 0)                                  AS zjada,
        DENSE_RANK() OVER (ORDER BY przydzial_myszy + NVL(myszy_extra, 0) DESC) AS rnk
    FROM Kocury
    ORDER BY
        zjada DESC
)
WHERE rnk <= &n;
```

PSEUDO	ZJADA
TYGRYS	136
LYSY	93
ZOMBI	88
LOLA	72
PLACEK	67
SZYBKA	65
RAFA	65

Liczba krotek: 7

Zadanie 23

```
-------------------------------------------------------------------------------
-- TASK 23
-------------------------------------------------------------------------------
WITH Lata AS (
    SELECT
        EXTRACT(YEAR FROM w_stadku_od) AS rok,
        COUNT(*)                       AS liczba
    FROM Kocury
    GROUP BY EXTRACT(YEAR FROM w_stadku_od)
),
Srednia AS (
    SELECT
        ROUND(AVG(liczba), 7) AS sr
    FROM Lata
),
-- Najblizsze mniejsze od sredniej
Dol AS (
    SELECT *
    FROM Lata, Srednia
    WHERE
        liczba <= sr
    ORDER BY
        (sr - liczba)
    FETCH FIRST 1 ROWS WITH TIES
),
-- Najbliższe wieksze od sredniej
Gora AS (
    SELECT *
    FROM Lata, Srednia
    WHERE
        liczba > sr
    ORDER BY
        (liczba - sr)
    FETCH FIRST 1 ROWS WITH TIES
)
-- Łączenie wynikowych tabel
SELECT
    TO_CHAR(rok),
    liczba
FROM Dol

UNION ALL

SELECT
    'Srednia',
    sr
FROM Srednia

UNION ALL

SELECT
    TO_CHAR(rok),
    liczba
FROM Gora
```

```
ORDER BY
    liczba;
```

TO_CHAR(ROK)	LICZBA
2009	2
2010	2
2011	2
2002	2
Srednia	2,5714286
2006	4

Liczba krotek: 6

Zadanie 24

a)

```
--------------------------------------------------------------------------------
-- TASK 24
--------------------------------------------------------------------------------
-- a)
SELECT
    k1.imie                                                 "IMIE",
    k1.przydzial_myszy + NVL(k1.myszy_extra, 0)             "ZJADA",
    k1.nr_bandy                                             "NR BANDY",
    ROUND(AVG(k2.przydzial_myszy + NVL(k2.myszy_extra, 0)), 2) "SREDNIA BANDY"
FROM Kocury k1
    INNER JOIN Kocury k2 ON k2.nr_bandy = k1.nr_bandy
WHERE
    k1.plec = 'M'
GROUP BY
    k1.imie, "ZJADA", k1.nr_bandy
HAVING
    "ZJADA" < "SREDNIA BANDY"
ORDER BY
    "SREDNIA BANDY";
```

IMIE	ZJADA	NR BANDY	SREDNIA BANDY
DUDEK	40	4	49,4
LUCEK	43	3	61,75
BARI	56	2	66,6
CHYTRY	50	1	80,5

Liczba krotek: 4

b)

```sql
-- b)
SELECT
    k1.imie                                     "IMIE",
    k1.przydzial_myszy + NVL(k1.myszy_extra, 0) "ZJADA",
    k1.nr_bandy                                 "NR BANDY",
    ROUND(k2.sred, 2)                           "SREDNIA BANDY"
FROM Kocury k1
    INNER JOIN (
        SELECT
            nr_bandy,
            AVG(przydzial_myszy + NVL(myszy_extra, 0)) AS sred
        FROM Kocury
        GROUP BY
            nr_bandy
    ) k2 ON k2.nr_bandy = k1.nr_bandy AND k1.przydzial_myszy + NVL(k1.myszy_extra, 0) < k2.sred
WHERE
    k1.plec = 'M'
ORDER BY
    "SREDNIA BANDY";
```

IMIE	ZJADA	NR BANDY	SREDNIA BANDY
DUDEK	40	4	49,4
LUCEK	43	3	61,75
BARI	56	2	66,6
CHYTRY	50	1	80,5

Liczba krotek: 4

c)

```sql
-- c)
SELECT
    k1.imie                                     "IMIE",
    k1.przydzial_myszy + NVL(k1.myszy_extra, 0) "ZJADA",
    k1.nr_bandy                                 "NR BANDY",
    ROUND((
        SELECT
            AVG(przydzial_myszy + NVL(myszy_extra, 0))
        FROM Kocury
        WHERE
            nr_bandy = k1.nr_bandy
    ), 2) "SREDNIA BANDY"
FROM Kocury k1
WHERE
    k1.plec = 'M'
    AND
    k1.przydzial_myszy + NVL(k1.myszy_extra, 0) < (
        SELECT
            AVG(przydzial_myszy + NVL(myszy_extra, 0))
        FROM Kocury
        WHERE
            nr_bandy = k1.nr_bandy
    )
ORDER BY
    "SREDNIA BANDY";
```

IMIE	ZJADA	NR BANDY	SREDNIA BANDY
DUDEK	40	4	49,4
LUCEK	43	3	61,75
BARI	56	2	66,6
CHYTRY	50	1	80,5

Liczba krotek: 4

Zadanie 25

```
-------------------------------------------------------------------------------------------
-- TASK 25
-------------------------------------------------------------------------------------------
WITH Daty AS (
    SELECT
    imie,
    TO_CHAR(data_wst, 'YYYY-MM-DD') AS data_wst_format,
    nazwa_bandy,
    staz_max,
    staz_min
    FROM (
        SELECT
            k.imie AS imie,
            k.w_stadku_od AS data_wst,
            b.nazwa AS nazwa_bandy,
            MIN(k.w_stadku_od) OVER (PARTITION BY b.nr_bandy) staz_max,
            MAX(k.w_stadku_od) OVER (PARTITION BY b.nr_bandy) staz_min
        FROM Kocury k
            INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
    )
)
-- Kocury ze stazami posrednimi
SELECT
    d.imie            "IMIE",
    d.data_wst_format "WSTAPIL DO STADKA"
FROM Daty d
WHERE
    d.data_wst_format NOT IN (d.staz_max, d.staz_min)

UNION
-- Kocury z najmniejszymi stazami
SELECT
    d.imie "IMIE",
    d.data_wst_format || ' <--- NAJMLODSZY STAZEM W BANDZIE ' || d.nazwa_bandy
FROM Daty d
WHERE
    d.data_wst_format = staz_min

UNION
-- Kocury z najwiekszymi stazami
SELECT
    d.imie "IMIE",
    d.data_wst_format || ' <--- NAJSTARSZY STAZEM W BANDZIE ' || d.nazwa_bandy
FROM Daty d
WHERE
    d.data_wst_format = staz_max

ORDER BY
    "IMIE";
```

```
IMIE            WSTAPIL DO STADKA
--------------- ------------------------------------------------------------
BARI            2009-09-01 <--- NAJMLODSZY STAZEM W BANDZIE CZARNI RYCERZE
BELA            2008-02-01
BOLEK           2006-08-15
CHYTRY          2002-05-05
DUDEK           2011-05-15 <--- NAJMLODSZY STAZEM W BANDZIE LACIACI MYSLIWI
JACEK           2008-12-01
KOREK           2004-03-16 <--- NAJSTARSZY STAZEM W BANDZIE BIALI LOWCY
KSAWERY         2008-07-12
LATKA           2011-01-01
LUCEK           2010-03-01
MELA            2008-11-01

IMIE            WSTAPIL DO STADKA
--------------- ------------------------------------------------------------
MICKA           2009-10-14 <--- NAJMLODSZY STAZEM W BANDZIE SZEFOSTWO
MRUCZEK         2002-01-01 <--- NAJSTARSZY STAZEM W BANDZIE SZEFOSTWO
PUCEK           2006-10-15 <--- NAJSTARSZY STAZEM W BANDZIE LACIACI MYSLIWI
PUNIA           2008-01-01
RUDA            2006-09-17
SONIA           2010-11-18 <--- NAJMLODSZY STAZEM W BANDZIE BIALI LOWCY
ZUZIA           2006-07-21 <--- NAJSTARSZY STAZEM W BANDZIE CZARNI RYCERZE
```

Liczba krotek: 18

Ogólne spostrzeżenia po wykonaniu zadań z części *Część Oracle*

Zadania z tej części były zdecydowanie bardziej wymagające i złożone. Rozwiązania dla wielu z nich zapewne można wykonać na wiele sposobów, a tu zaprezentowane są tymi, które były pierwszymi działającymi, które przyszły mi do głowy. W niektórych przypadkach natomiast zmuszony byłem zmienić koncepcję, co wynikało z ograniczeń pierwotnie stosowanych rozwiązań, jak np. kolejność wykonywania operacji czy to jakie operacje są dozwolone przy korzystaniu z CTE - z tego powodu w niektórych przypadkach prościej było wykorzystać klasyczne podzapytania.

Dość dużą trudność sprawiły mi podpunkty b) i c) zadania 14., gdzie narzucone do wykorzystania operacje wymagały opracowania przeze mnie często wielu pomysłów i łączenia ich ze sobą - tu szczególnie podpunkt b) i skorzystanie z `LEVEL` do tworzenia kolumn za pomocą `PIVOT` oraz, w podpunkcie c), formatowanie wynikowych wierszy z wielokrotnym złożeniem operacji `REVERSE()` czy `LTRIM()` i `RTRIM()`.

Co istotne, w jednym zadaniu wyniki przykładowe nie zgadzały się w zupełności z wynikami otrzymanymi przeze mnie, pomimo logicznej poprawności kwerendy, a ręczne sprawdzenie potwierdza poprawność prezentowanego rozwiązania. Dotyczy to zadania 20., w którym, aby otrzymać wynik zgodny z przykładowym, należy pominąć zapis „[...] WSZYSTKICH MILUŚ **OPERUJĄCYCH W SADZIE**", czyli zastosować kryterium `k.funkcja = 'MILUSIA'` i porzucić kryterium `b.teren = 'SAD'`.

Część SQL Server

Zadanie 26

```
-------------------------------------------------------------------------------
-- TASK 26
-------------------------------------------------------------------------------
WITH Kotki AS (
    SELECT
        pseudo
    FROM Kocury
    WHERE
        plec = 'D'
),
Incydenty AS (
    SELECT DISTINCT
        k.pseudo AS psd
    FROM Kotki k
        INNER JOIN Wrogowie_kocurow wk ON wk.pseudo = k.pseudo
        INNER JOIN Wrogowie w ON w.imie_wroga = wk.imie_wroga AND w.stopien_wrogosci > 5
)
SELECT
    psd [Zadziorne kotki]
FROM Incydenty
ORDER BY
    psd DESC;
```

```
Zadziorne kotki
---------------
UCHO
LASKA
DAMA
((3 rows affected))
```

Zadanie 27

```
-------------------------------------------------------------------------------------------
-- TASK 27
-------------------------------------------------------------------------------------------
WITH Hierarchia AS (
    -- Baza
    SELECT
        1 AS poziom,
        pseudo,
        funkcja,
        nr_bandy,
        plec
    FROM Kocury
    WHERE
        plec = 'M'
        AND
        funkcja = 'BANDZIOR'

    UNION ALL
    -- Rekurencyjne wywolanie
    SELECT
        poziom + 1,
        k.pseudo,
        k.funkcja,
        k.nr_bandy,
        k.plec
    FROM Kocury k
        INNER JOIN Hierarchia h ON h.pseudo = k.szef
)
SELECT
    poziom   [Poziom],
    pseudo   [Pseudonim],
    funkcja  [Funkcja],
    nr_bandy [Nr bandy]
FROM Hierarchia
WHERE
    plec = 'M'
ORDER BY
    nr_bandy;
```

```
Poziom      Pseudonim   Funkcja     Nr bandy
----------  ----------  ----------  ----------
1           LYSY        BANDZIOR    2
2           PLACEK      LOWCZY      2
2           RURA        LAPACZ      2
1           ZOMBI       BANDZIOR    3
3           ZERO        KOT         3
((5 rows affected))
```

Zadanie 28

```
-------------------------------------------------------------------------------
-- TASK 28
-------------------------------------------------------------------------------
WITH Hierarchia AS (
    -- Baza
    SELECT
        0 AS poziom,
        szef,
        funkcja,
        pseudo,
        imie
    FROM Kocury
    WHERE
        szef IS NULL
        AND
        myszy_extra IS NOT NULL

    -- Rekurencyjne wywolanie
    UNION ALL

    SELECT
        h.poziom + 1,
        k.szef,
        k.funkcja,
        k.pseudo,
        k.imie
    FROM Kocury k INNER JOIN Hierarchia h ON h.pseudo = k.szef AND k.myszy_extra IS NOT NULL
)
SELECT
    CONCAT(
        REPLICATE('===>', poziom),
        poziom,
        REPLICATE(' ', 15 - 4 * poziom),
        imie
    )                               [Hierarchia],
    ISNULL(szef, 'Sam sobie panem') [Pseudo szefa],
    funkcja                         [Funkcja]
FROM Hierarchia;
```

```
Hierarchia              Pseudo szefa     Funkcja
----------------------  ---------------  ----------
0              MRUCZEK  Sam sobie panem  SZEFUNIO
===>1          MICKA    TYGRYS           MILUSIA
===>1          BOLEK    TYGRYS           BANDZIOR
===>1          RUDA     TYGRYS           MILUSIA
===>1          KOREK    TYGRYS           BANDZIOR
===>===>2      SONIA    ZOMBI            MILUSIA
===>===>2      BELA     LYSY             MILUSIA
((7 rows affected))
```

Zadanie 29

```
-------------------------------------------------------------------------------
-- TASK 29
-------------------------------------------------------------------------------
SELECT
    k1.pseudo [Do przeczolgania],
    b.nazwa  [Nazwa bandy]
FROM Kocury k1
    INNER JOIN Bandy b ON b.nr_bandy = k1.nr_bandy
    INNER JOIN Funkcje f ON f.funkcja = k1.funkcja
WHERE
    NOT EXISTS (
        SELECT *
        FROM Kocury k2
        WHERE
            k2.szef = k1.pseudo
    )
   AND
    EXISTS (
        SELECT *
        FROM Wrogowie_kocurow wk
        WHERE wk.pseudo = k1.pseudo
    )
    AND
    k1.przydzial_myszy > f.min_myszy + (f.max_myszy - f.min_myszy) / 3
ORDER BY
    [Nazwa bandy], [Do przeczolgania];
```

```
Do przeczolgania  Nazwa bandy
----------------  --------------
LASKA             CZARNI RYCERZE
PLACEK            CZARNI RYCERZE
RURA              CZARNI RYCERZE
SZYBKA            CZARNI RYCERZE
BOLEK             SZEFOSTWO
((5 rows affected))
```

Ogólne spostrzeżenia po wykonaniu zadań z części *Część SQL Server*

Dwa spośród zadań z tej części znajdowały się na liście pierwszej, jednak należało wykonać je jedynie w dialekcie Oracle. Teraz natomiast, gdy należało wykonać je w dialekcie Transact-SQL, bardzo widoczne stały się różnice między tymi dialektami. Bardzo brakuje tu mechanizmu kwerend hierarchicznych z wykorzystaniem `CONNECT BY`, ponieważ znacząco ułatwiało to i przyspieszało pracę. Tu należało posłużyć się sposobem nieco bardziej klasycznym, czyli rekurencyjnymi CTE. Najbardziej uciążliwe w tym sposobie jest obliczanie poziomu drzewa, `LEVEL` jest znacznym ułatwieniem.

Część Oracle + SQL Server

Zadanie 30

Oracle

Perspektywa i jej zawartość:

```
-------------------------------------------------------------------------------
-- TASK 30
-------------------------------------------------------------------------------
CREATE OR REPLACE VIEW Statystyki_band (
    "NAZWA BANDY",
    "SRE_SPOZ",
    "MAX_SPOZ",
    "MIN_SPOZ",
    "KOTY",
    "KOTY_Z_DOD"
) AS
SELECT
    b.nazwa                AS nazwa,
    AVG(k.przydzial_myszy) AS sre_spoz,
    MAX(k.przydzial_myszy) AS max_spoz,
    MIN(k.przydzial_myszy) AS min_spoz,
    COUNT(k.pseudo)        AS koty,
    COUNT(k.myszy_extra)   AS koty_z_dod
FROM Bandy b
    INNER JOIN Kocury k ON k.nr_bandy = b.nr_bandy
GROUP BY
    nazwa
ORDER BY
    max_spoz DESC;

-- Zawartosc perspektywy
SELECT *
FROM Statystyki_band;
```

```
View STATYSTYKI_BAND created.


NAZWA BANDY           SRE_SPOZ   MAX_SPOZ   MIN_SPOZ       KOTY KOTY_Z_DOD
-------------------- ---------- ---------- ---------- ---------- ----------
SZEFOSTWO                    50        103         22          4          3
BIALI LOWCY                49,75        75         20          4          2
CZARNI RYCERZE              56,8        72         24          5          2
LACIACI MYSLIWI             49,4        65         40          5          0
```

Zapytanie i wynik dla wprowadzonego przez użytkownika kota 'PLACEK':

```
-- Zapytanie dla kota wskazanego przez uzytkownika
SELECT
    k.pseudo                                        "PSEUDONIM",
    k.imie                                          "IMIE",
    k.funkcja                                       "FUNKCJA",
    k.przydzial_myszy                               "ZJADA",
    'OD ' || sb.min_spoz || ' DO ' || sb.max_spoz "GRANICE SPOZYCIA",
    TO_CHAR(k.w_stadku_od, 'YYYY-MM-DD')            "LOWI OD"
FROM Kocury k
    INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
    INNER JOIN Statystyki_band sb ON sb."NAZWA BANDY" = b.nazwa
WHERE
    k.pseudo = UPPER(&pseudo_input);
```

	PSEUDONIM	IMIE	FUNKCJA	ZJADA	GRANICE SPOZYCIA	LOWI OD
1	PLACEK	JACEK	LOWCZY	67	OD 24 DO 72	2008-12-01

Microsoft

Perspektywa i jej zawartość:

```
--------------------------------------------------------------------------------
-- TASK 30
--------------------------------------------------------------------------------
CREATE OR ALTER VIEW Statystyki_band (
    [NAZWA_BANDY],
    [SRE_SPOZ],
    [MAX_SPOZ],
    [MIN_SPOZ],
    [KOTY],
    [KOTY_Z_DOD]
) AS
SELECT
    b.nazwa                AS nazwa,
    AVG(k.przydzial_myszy) AS sre_spoz,
    MAX(k.przydzial_myszy) AS max_spoz,
    MIN(k.przydzial_myszy) AS min_spoz,
    COUNT(k.pseudo)        AS koty,
    COUNT(k.myszy_extra)   AS koty_z_dod
FROM Kocury k
    INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
GROUP BY
    nazwa;

GO

-- Zawartosc perspektywy
SELECT *
FROM Statystyki_band;
```

```
NAZWA_BANDY      SRE_SPOZ    MAX_SPOZ    MIN_SPOZ    KOTY        KOTY_Z_DOD
---------------  ----------  ----------  ----------  ----------  ----------
BIALI LOWCY      49          75          20          4           2
CZARNI RYCERZE   56          72          24          5           2
LACIACI MYSLIWI  49          65          40          5           0
SZEFOSTWO        50          103         22          4           3
((4 rows affected))
```

Zapytanie i wynik dla kota ‘PLACEK’:

```sql
-- Zapytanie dla kota wskazanego przez uzytkownika
DECLARE @pseudo_input VARCHAR(15);
SET @pseudo_input = 'PLACEK'

SELECT
    k.pseudo                                          [PSEUDONIM],
    k.imie                                            [IMIE],
    k.funkcja                                         [FUNKCJA],
    k.przydzial_myszy                                 [ZJADA],
    CONCAT('OD ', sb.[MIN_SPOZ], ' DO ', sb.[MAX_SPOZ]) [GRANICE SPOZYCIA],
    k.w_stadku_od                                     [LOWI OD]
FROM Kocury k
    INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
    INNER JOIN Statystyki_band sb ON sb.[NAZWA_BANDY] = b.nazwa
WHERE
    k.pseudo = UPPER(@pseudo_input);
```

```
PSEUDONIM   IMIE        FUNKCJA     ZJADA       GRANICE SPOZYCIA  LOWI OD
----------  ----------  ----------  ----------  ----------------  ----------
PLACEK      JACEK       LOWCZY      67          OD 24 DO 72       2008-12-01
((1 row affected))
```

Zadanie 31

Oracle

```
-- ------------------------------------------------------------------------------------------
-- TASK 31
-- ------------------------------------------------------------------------------------------
CREATE OR REPLACE VIEW Najdluzsze_staze (
    pseudonim,
    plec,
    myszy,
    extra,
    num_bandy
) AS
WITH Koty_z_czarni_rycerze AS (
    SELECT
        k.pseudo
    FROM Kocury k
        INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
    WHERE
        b.nazwa = 'CZARNI RYCERZE'
    ORDER BY
        k.w_stadku_od
    FETCH FIRST 3 ROWS ONLY
),
Koty_z_laciaci_mysliwi AS (
    SELECT
        k.pseudo
    FROM Kocury k
        INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
    WHERE
        b.nazwa = 'LACIACI MYSLIWI'
    ORDER BY
        k.w_stadku_od
    FETCH FIRST 3 ROWS ONLY
)
SELECT
    k.pseudo,
    k.plec,
    k.przydzial_myszy,
    k.myszy_extra,
    k.nr_bandy
FROM Kocury k
    LEFT JOIN Koty_z_czarni_rycerze cr ON k.pseudo = cr.pseudo
    LEFT JOIN Koty_z_laciaci_mysliwi lm ON k.pseudo = lm.pseudo
WHERE
    cr.pseudo IS NOT NULL
    OR
    lm.pseudo IS NOT NULL;
```

```
-- Przed podywzkami
SELECT
    pseudonim "Pseudonim",
    plec      "Plec",
    myszy     "Myszy przed podw.",
    extra     "Extra przed podw."
```

```
FROM Najdluzsze_staze;
```

Wynik przed podwyżką:

```
View NAJDLUZSZE_STAZE created.


Pseudonim       P Myszy przed podw. Extra przed podw.
--------------- - ----------------- -----------------
RAFA            M                65
MAN             M                51
DAMA            D                51
LYSY            M                72                21
SZYBKA          D                65
LASKA           D                24                28

6 rows selected.
```

Podwyżka:

```
-- Podwyzka
UPDATE Najdluzsze_staze
SET
    myszy = myszy + DECODE(
        plec,
        'D', 0.1 * (SELECT MIN(przydzial_myszy) FROM Kocury),
        'M', 10
    ),
    extra = NVL(extra, 0) + 0.15 * (
        SELECT
            AVG(NVL(k.myszy_extra, 0) )
        FROM Kocury k
        WHERE
            k.nr_bandy = num_bandy
    );
```

```
-- Po podwyzce
SELECT
    pseudonim "Pseudonim",
    plec      "Plec",
    myszy     "Myszy po podw.",
    extra     "Extra po podw."
FROM Najdluzsze_staze;

ROLLBACK;
```

Wynik po podwyżce:

```
6 rows updated.


Pseudonim       P Myszy po podw. Extra po podw.
--------------- - --------------- --------------
RAFA            M              75              0
MAN             M              61              0
DAMA            D              53              0
LYSY            M              82             22
SZYBKA          D              67              1
LASKA           D              26             29

6 rows selected.


Rollback complete.
```

Microsoft

```
-------------------------------------------------------------------------------------------------
-- TASK 31
-------------------------------------------------------------------------------------------------
GO

CREATE OR ALTER VIEW Najdluzsze_staze (
    pseudonim,
    plec,
    myszy,
    extra,
    num_bandy
) AS
WITH Koty_z_czarni_rycerze AS (
    SELECT TOP(3)
        k.pseudo
    FROM Kocury k
        INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
    WHERE
        b.nazwa = 'CZARNI RYCERZE'
    ORDER BY
        k.w_stadku_od
),
Koty_z_laciaci_mysliwi AS (
    SELECT TOP(3)
        k.pseudo
    FROM Kocury k
        INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
    WHERE
        b.nazwa = 'LACIACI MYSLIWI'
    ORDER BY
        k.w_stadku_od
)
SELECT
    k.pseudo,
    k.plec,
    k.przydzial_myszy,
    k.myszy_extra,
    k.nr_bandy
FROM Kocury k
    LEFT JOIN Koty_z_czarni_rycerze cr ON cr.pseudo = k.pseudo
    LEFT JOIN Koty_z_laciaci_mysliwi lm ON lm.pseudo = k.pseudo
WHERE
    cr.pseudo IS NOT NULL
    OR
    lm.pseudo IS NOT NULL;

GO
```

```
-- Przed podwyzkami
SELECT
    pseudonim        [Pseudonim],
    plec             [Plec],
    myszy            [Myszy przed podw.],
    ISNULL(extra, 0) [Extra przed podw.]
```

```
FROM Najdluzsze_staze;
```

Wynik przed podwyżką:

```
Pseudonim   Plec        Myszy przed podw.  Extra przed podw.
----------  ----------  -----------------  -----------------
DAMA        D           51                 0
LASKA       D           24                 28
LYSY        M           72                 21
MAN         M           51                 0
RAFA        M           65                 0
SZYBKA      D           65                 0
((6 rows affected))
```

Podwyżka:

```
-- Podwyzka
BEGIN TRANSACTION;

UPDATE Najdluzsze_staze
SET
    myszy = myszy +
    CASE
        WHEN plec = 'D' THEN 0.1 * (SELECT MIN(przydzial_myszy) FROM Kocury)
        WHEN plec = 'M' THEN 10
    END,
    extra = ISNULL(extra, 0) + 0.15 * (
        SELECT
            AVG(ISNULL(k.myszy_extra, 0))
        FROM Kocury k
        WHERE
            k.nr_bandy = num_bandy
    );
```

```
-- Po podwyzce
SELECT
    pseudonim        [Pseudonim],
    plec             [Plec],
    myszy            [Myszy po podw.],
    ISNULL(extra, 0) [Extra po podw.]
FROM Najdluzsze_staze;

ROLLBACK;
```

Wynik po podwyżce:

```
Pseudonim   Plec        Myszy po podw.  Extra po podw.
----------  ----------  --------------  --------------
DAMA        D           53              0
LASKA       D           26              29
LYSY        M           82              22
MAN         M           61              0
RAFA        M           75              0
SZYBKA      D           67              1
((6 rows affected))
```

Zadanie 32

Oracle

a)

```
-------------------------------------------------------------------------------
-- TASK 32
-------------------------------------------------------------------------------
--a)
-- Statystyki wg band i plci
CREATE OR REPLACE VIEW Statystyki_band_plec (
    "NAZWA BANDY",
    "PLEC",
    "ILE",
    "SZEFUNIO",
    "BANDZIOR",
    "LOWCZY",
    "LAPACZ",
    "KOT",
    "MILUSIA",
    "DZIELCZY",
    "SUMA"
) AS
SELECT
    DECODE(k.plec, 'D', b.nazwa, 'M', ' '),
    DECODE(k.plec, 'D', 'Kotka', 'M', 'Kocor'),
    TO_CHAR(COUNT(k.pseudo)),
    TO_CHAR(SUM(DECODE(
        k.funkcja,
        'SZEFUNIO', k.przydzial_myszy + NVL(k.myszy_extra, 0),
        0
    ))),
    TO_CHAR(SUM(DECODE(
        k.funkcja,
        'BANDZIOR', k.przydzial_myszy + NVL(k.myszy_extra, 0),
        0
    ))),
    TO_CHAR(SUM(DECODE(
        k.funkcja,
        'LOWCZY', k.przydzial_myszy + NVL(k.myszy_extra, 0),
        0
    ))),
    TO_CHAR(SUM(DECODE(
        k.funkcja,
        'LAPACZ', k.przydzial_myszy + NVL(k.myszy_extra, 0),
        0
    ))),
    TO_CHAR(SUM(DECODE(
        k.funkcja,
        'KOT', k.przydzial_myszy + NVL(k.myszy_extra, 0),
        0
    ))),
    TO_CHAR(SUM(DECODE(
        k.funkcja,
        'MILUSIA', k.przydzial_myszy + NVL(k.myszy_extra, 0),
        0
    ))),
```

```
    TO_CHAR(SUM(DECODE(
        k.funkcja,
        'DZIELCZY', k.przydzial_myszy + NVL(k.myszy_extra, 0),
        0
    ))),
    TO_CHAR(SUM(k.przydzial_myszy + NVL(k.myszy_extra, 0)))
FROM Kocury k
    INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
GROUP BY
    b.nazwa,
    k.plec
ORDER BY
    b.nazwa;

SELECT *
FROM Statystyki_band_plec

UNION ALL

-- Separator
SELECT 'Z----------------', '------', '----', '---------', '---------', '---------', '---------
', '---------', '---------', '---------', '-------'
FROM dual

UNION ALL

-- Podsumowanie
SELECT
    'ZJADA RAZEM',
    ' ',
    ' ',
    TO_CHAR(SUM("SZEFUNIO")),
    TO_CHAR(SUM("BANDZIOR")),
    TO_CHAR(SUM("LOWCZY")),
    TO_CHAR(SUM("LAPACZ")),
    TO_CHAR(SUM("KOT")),
    TO_CHAR(SUM("MILUSIA")),
    TO_CHAR(SUM("DZIELCZY")),
    TO_CHAR(SUM("SUMA"))
FROM Statystyki_band_plec;
```

	NAZWA BANDY	PLEC	ILE	SZEFUNIO	BANDZIOR	LOWCZY	LAPACZ	KOT	MILUSIA	DZIELCZY	SUMA
1	BIALI LOWCY	Kotka	2	0	0	61	0	0	55	0	116
2		Kocor	2	0	88	0	0	43	0	0	131
3	CZARNI RYCERZE	Kotka	2	0	0	65	0	0	52	0	117
4		Kocor	3	0	93	67	56	0	0	0	216
5	LACIACI MYSLIWI	Kotka	2	0	0	0	51	40	0	0	91
6		Kocor	3	0	0	65	51	40	0	0	156
7	SZEFOSTWO	Kotka	2	0	0	0	0	0	136	0	136
8		Kocor	2	136	0	0	0	0	0	50	186
9	Z----------	------	----	---------	---------	---------	---------	---------	---------	---------	--------
10	ZJADA RAZEM			136	181	258	158	123	243	50	1149

b)

```
--b)
-- Widok dla pivota
CREATE OR REPLACE VIEW Statystyki_band_plec_B (
    nazwa_bandy,
    plec_kota,
    funkcja_kota,
    spozycie
) AS
SELECT
    b.nazwa,
    k.plec,
    k.funkcja,
    k.przydzial_myszy + NVL(k.myszy_extra, 0)
FROM Kocury k
    INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy;

-- Widok do dołączenia - suma spozycia i liczba kotow danej plci w bandach
CREATE OR REPLACE VIEW Bandy_sumy_spozycia_per_liczba_kotow (
    nazwa,
    plec,
    liczba_kotow,
    suma_spozycia
) AS
SELECT
    nazwa,
    plec,
    COUNT(pseudo),
    SUM(przydzial_myszy + NVL(myszy_extra, 0))
FROM Kocury k1
    INNER JOIN Bandy b1 ON b1.nr_bandy = k1.nr_bandy
GROUP BY nazwa, plec;

-- Podsumowanie
SELECT *
FROM (
    SELECT
        DECODE(plec_kota, 'D', nazwa_bandy, 'M', ' ') "NAZWA BANDY",
        DECODE(plec_kota, 'D', 'Kotka', 'M', 'Kocor') "PLEC",
        TO_CHAR(liczba_kotow)                         "ILE",
        TO_CHAR(NVL(szefunio, 0))                     "SZEFUNIO",
        TO_CHAR(NVL(bandzior, 0))                     "BANDZIOR",
        TO_CHAR(NVL(lowczy, 0))                       "LOWCZY",
        TO_CHAR(NVL(lapacz, 0))                       "LAPACZ",
        TO_CHAR(NVL(kot, 0))                          "KOT",
        TO_CHAR(NVL(milusia, 0))                      "MILUSIA",
        TO_CHAR(NVL(dzielczy, 0))                     "DZIELCZY",
        TO_CHAR(suma_spozycia)                        "SUMA"
    FROM Statystyki_band_plec_B
    PIVOT (
        SUM(spozycie)
        FOR funkcja_kota IN (
            'SZEFUNIO' szefunio,
            'BANDZIOR' bandzior,
            'LOWCZY'   lowczy,
            'LAPACZ'   lapacz,
```

```
            'KOT'      kot,
            'MILUSIA'  milusia,
            'DZIELCZY' dzielczy
        )
    )
        INNER JOIN Bandy_sumy_spozycia_per_liczba_kotow
            ON nazwa = nazwa_bandy AND plec = plec_kota
    ORDER BY
        nazwa_bandy
)

UNION ALL

-- Separator
SELECT
    'Z----------------', '------', '----', '---------', '---------', '---------', '---------',
'---------', '---------', '---------', '-------'
FROM dual

UNION ALL

-- Podsumowanie
SELECT
    'ZJADA RAZEM',
    ' ',
    ' ',
    TO_CHAR(szefunio),
    TO_CHAR(bandzior),
    TO_CHAR(lowczy),
    TO_CHAR(lapacz),
    TO_CHAR(kot),
    TO_CHAR(milusia),
    TO_CHAR(dzielczy),
    TO_CHAR(suma)
FROM (
    SELECT
        k.funkcja                                AS funkcja_kota,
        k.przydzial_myszy + NVL(k.myszy_extra, 0) AS spozycie
    FROM Kocury k
        INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
)
PIVOT (
    SUM(spozycie)
    FOR funkcja_kota IN (
        'SZEFUNIO' szefunio,
        'BANDZIOR' bandzior,
        'LOWCZY'   lowczy,
        'LAPACZ'   lapacz,
        'KOT'      kot,
        'MILUSIA'  milusia,
        'DZIELCZY' dzielczy
    )
)
 CROSS JOIN (
    SELECT
        SUM(przydzial_myszy + NVL(myszy_extra, 0)) AS suma
    FROM Kocury
```

```
);
```

	NAZWA BANDY	PLEC	ILE	SZEFUNIO	BANDZIOR	LOWCZY	LAPACZ	KOT	MILUSIA	DZIELCZY	SUMA
1	BIALI LOWCY	Kotka	2	0	0	61	0	0	55	0	116
2		Kocor	2	0	88	0	0	43	0	0	131
3	CZARNI RYCERZE	Kotka	2	0	0	65	0	0	52	0	117
4		Kocor	3	0	93	67	56	0	0	0	216
5	LACIACI MYSLIWI	Kotka	2	0	0	0	51	40	0	0	91
6		Kocor	3	0	0	65	51	40	0	0	156
7	SZEFOSTWO	Kotka	2	0	0	0	0	0	136	0	136
8		Kocor	2	136	0	0	0	0	0	50	186
9	Z----------	------	----	--------	--------	--------	--------	--------	--------	--------	-------
10	ZJADA RAZEM			136	181	258	158	123	243	50	1149

Microsoft

a)

```
-------------------------------------------------------------------------------------------
-- TASK 32
-------------------------------------------------------------------------------------------
--a)
GO

-- Statystyki wg band i plci
CREATE OR ALTER VIEW Statystyki_band_plec_A (
    nazwa_bandy,
    plec,
    ile,
    szefunio,
    bandzior,
    lowczy,
    lapacz,
    kot,
    milusia,
    dzielczy,
    suma
) AS
SELECT
    CASE
        WHEN k.plec = 'D' THEN b.nazwa
        WHEN k.plec = 'M' THEN ' '
    END,
    CASE
        WHEN k.plec = 'D' THEN 'Kotka'
        WHEN k.plec = 'M' THEN 'Kocor'
    END,
    COUNT(k.pseudo),
    SUM(CASE
            WHEN k.funkcja = 'SZEFUNIO' THEN k.przydzial_myszy + ISNULL(k.myszy_extra, 0)
            ELSE 0
        END),
    SUM(CASE
            WHEN k.funkcja = 'BANDZIOR' THEN k.przydzial_myszy + ISNULL(k.myszy_extra, 0)
            ELSE 0
        END),
    SUM(CASE
            WHEN k.funkcja = 'LOWCZY' THEN k.przydzial_myszy + ISNULL(k.myszy_extra, 0)
            ELSE 0
        END),
```

```
    SUM(CASE
            WHEN k.funkcja = 'LAPACZ' THEN k.przydzial_myszy + ISNULL(k.myszy_extra, 0)
            ELSE 0
        END),
    SUM(CASE
            WHEN k.funkcja = 'KOT' THEN k.przydzial_myszy + ISNULL(k.myszy_extra, 0)
            ELSE 0
        END),
    SUM(CASE
            WHEN k.funkcja = 'MILUSIA' THEN k.przydzial_myszy + ISNULL(k.myszy_extra, 0)
            ELSE 0
        END),
    SUM(CASE
            WHEN k.funkcja = 'DZIELCZY' THEN k.przydzial_myszy + ISNULL(k.myszy_extra, 0)
            ELSE 0
        END),
    SUM(k.przydzial_myszy + ISNULL(k.myszy_extra, 0))
FROM Kocury k
    INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
GROUP BY
    b.nazwa,
    k.plec;

GO


SELECT
    nazwa_bandy               [NAZWA BANDY],
    plec                      [PLEC],
    CAST(ile AS VARCHAR)      [ILE],
    CAST(szefunio AS VARCHAR) [SZEFUNIO],
    CAST(bandzior AS VARCHAR) [BANDZIOR],
    CAST(lowczy AS VARCHAR)   [LOWCZY],
    CAST(lapacz AS VARCHAR)   [LAPACZ],
    CAST(kot AS VARCHAR)      [KOT],
    CAST(milusia AS VARCHAR)  [MILUSIA],
    CAST(dzielczy AS VARCHAR) [DZIELCZY],
    CAST(suma AS VARCHAR)     [SUMA]
FROM Statystyki_band_plec_A

UNION ALL

-- Separator
SELECT
    'Z----------------' AS Col1,
    '------'            AS Col2,
    '----'              AS Col3,
    '---------'         AS Col4,
    '---------'         AS Col5,
    '---------'         AS Col6,
    '---------'         AS Col7,
    '---------'         AS Col8,
    '---------'         AS Col9,
    '---------'         AS Col10,
    '-------'           AS Col11

UNION ALL
```

```
-- Podsumowanie
SELECT
    'ZJADA RAZEM',
    ' ',
    ' ',
    CAST((SUM([SZEFUNIO])) AS VARCHAR),
    CAST((SUM([BANDZIOR])) AS VARCHAR),
    CAST((SUM([LOWCZY])) AS VARCHAR),
    CAST((SUM([LAPACZ])) AS VARCHAR),
    CAST((SUM([KOT])) AS VARCHAR),
    CAST((SUM([MILUSIA])) AS VARCHAR),
    CAST((SUM([DZIELCZY])) AS VARCHAR),
    CAST((SUM([SUMA])) AS VARCHAR)
FROM Statystyki_band_plec_A;
```

```
NAZWA BANDY      PLEC       ILE        SZEFUNIO   BANDZIOR   LOWCZY     LAPACZ     KOT        MILUSIA    DZIELCZY   SUMA
---------------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
BIALI LOWCY      Kotka      2          0          0          61         0          0          55         0          116
CZARNI RYCERZE   Kotka      2          0          0          65         0          0          52         0          117
LACIACI MYSLIWI  Kotka      2          0          0          0          51         40         0          0          91
SZEFOSTWO        Kotka      2          0          0          0          0          0          136        0          136
                 Kocor      2          0          88         0          0          43         0          0          131
                 Kocor      3          0          93         67         56         0          0          0          216
                 Kocor      3          0          0          65         51         40         0          0          156
                 Kocor      2          136        0          0          0          0          0          50         186
Z---------------- ------     ----       ---------- ---------- ---------- ---------- ---------- ---------- ---------- --------
ZJADA RAZEM                             136        181        258        158        123        243        50         1149
((10 rows affected))
```

b)

```
--b)
-- Widok dla pivota
GO

CREATE OR ALTER VIEW Statystyki_band_plec_B (
    nazwa_bandy,
    plec_kota,
    funkcja_kota,
    spozycie
) AS
SELECT
    b.nazwa,
    k.plec,
    k.funkcja,
    k.przydzial_myszy + ISNULL(k.myszy_extra, 0)
FROM Kocury k
    INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy;

GO

-- Widok do dołączenia - suma spozycia i liczba kotow danej plci w bandach
CREATE OR ALTER VIEW Bandy_sumy_spozycia_per_liczba_kotow (
    nazwa,
    plec,
    liczba_kotow,
    suma_spozycia
) AS
SELECT
```

```
    nazwa,
    plec,
    COUNT(pseudo),
    SUM(przydzial_myszy + ISNULL(myszy_extra, 0))
FROM Kocury k1
    INNER JOIN Bandy b1 ON b1.nr_bandy = k1.nr_bandy
GROUP BY nazwa, plec;

GO

-- Statystyki
SELECT *
FROM (
    SELECT
        CASE
            WHEN plec_kota = 'D' THEN nazwa_bandy
            WHEN plec_kota = 'M' THEN ' '
        END [NAZWA BANDY],
        CASE
            WHEN plec_kota = 'D' THEN 'Kotka'
            WHEN plec_kota = 'M' THEN 'Kocor'
        END [PLEC],
        CAST(liczba_kotow AS VARCHAR)        [ILE],
        CAST(ISNULL(szefunio, 0) AS VARCHAR) [SZEFUNIO],
        CAST(ISNULL(bandzior, 0) AS VARCHAR) [BANDZIOR],
        CAST(ISNULL(lowczy, 0) AS VARCHAR)   [LOWCZY],
        CAST(ISNULL(lapacz, 0) AS VARCHAR)   [LAPACZ],
        CAST(ISNULL(kot, 0) AS VARCHAR)      [KOT],
        CAST(ISNULL(milusia, 0) AS VARCHAR)  [MILUSIA],
        CAST(ISNULL(dzielczy, 0) AS VARCHAR) [DZIELCZY],
        CAST(suma_spozycia AS VARCHAR)       [SUMA]
    FROM Statystyki_band_plec_B
    PIVOT (
        SUM(spozycie)
        FOR funkcja_kota IN (
            SZEFUNIO,
            BANDZIOR,
            LOWCZY,
            LAPACZ,
            KOT,
            MILUSIA,
            DZIELCZY
        )
    ) AS pvt
        INNER JOIN Bandy_sumy_spozycia_per_liczba_kotow
            ON nazwa = nazwa_bandy AND plec = plec_kota
) AS T

UNION ALL

-- Separator
SELECT
    'Z----------------' AS Col1,
    '------'            AS Col2,
    '----'              AS Col3,
    '---------'         AS Col4,
    '---------'         AS Col5,
```

```
    '---------'         AS Col6,
    '---------'         AS Col7,
    '---------'         AS Col8,
    '---------'         AS Col9,
    '---------'         AS Col10,
    '-------'           AS Col11

UNION ALL

-- Podsumowanie
SELECT
    'ZJADA RAZEM',
    ' ',
    ' ',
    CAST(szefunio AS VARCHAR),
    CAST(bandzior AS VARCHAR),
    CAST(lowczy AS VARCHAR),
    CAST(lapacz AS VARCHAR),
    CAST(kot AS VARCHAR),
    CAST(milusia AS VARCHAR),
    CAST(dzielczy AS VARCHAR),
    CAST(suma AS VARCHAR)
FROM (
    SELECT
        k.funkcja                                   AS funkcja_kota,
        k.przydzial_myszy + ISNULL(k.myszy_extra, 0) AS spozycie
    FROM Kocury k
        INNER JOIN Bandy b ON b.nr_bandy = k.nr_bandy
) AS dane
PIVOT (
    SUM(spozycie)
    FOR funkcja_kota IN (
        SZEFUNIO,
        BANDZIOR,
        LOWCZY,
        LAPACZ,
        KOT,
        MILUSIA,
        DZIELCZY
    )
) AS pvt
 CROSS JOIN (
    SELECT
        SUM(przydzial_myszy + ISNULL(myszy_extra, 0)) AS suma
    FROM Kocury
) AS sumy;
```

```
NAZWA BANDY        PLEC       ILE        SZEFUNIO   BANDZIOR   LOWCZY     LAPACZ     KOT        MILUSIA    DZIELCZY   SUMA
-----------------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------
BIALI LOWCY        Kotka      2          0          0          61         0          0          55         0          116
CZARNI RYCERZE     Kotka      2          0          0          65         0          0          52         0          117
LACIACI MYSLIWI    Kotka      2          0          0          0          51         40         0          0          91
SZEFOSTWO          Kotka      2          0          0          0          0          0          136        0          136
                   Kocor      2          0          88         0          0          43         0          0          131
                   Kocor      3          0          93         67         56         0          0          0          216
                   Kocor      3          0          0          65         51         40         0          0          156
                   Kocor      2          136        0          0          0          0          0          50         186
Z-----------------  ------     ----       ---------  ---------  ---------  ---------  ---------  ---------  ---------  -------
ZJADA RAZEM                   136        181        258        158        123        243        50         1149
((10 rows affected))
```

Ogólne spostrzeżenia po wykonaniu zadań z części *Część Oracle + SQL Server*

W zadaniach z tej części różnice między SZBD Oracle i SZBD SQL Server były widoczne bardzo mocno. Mówiąc o różnicach należy w pierwszej kolejności wspomnieć oczywiście o funkcjach, które mają niemal identyczne działanie i składnię, lecz różnią się przede wszystkim nazwą czy kontekstem zastosowania; mowa tu o m.in. `TO_CHAR(<expression>)` z SZBD Oracle oraz `CAST(<expression> AS <type>)`, które służyły w ostatnim zadaniu do konwersji danych na typ tekstowy, aby możliwe było zastosowanie `UNION ALL`, pomimo teoretycznie różnych typów w kolumnach. W SQL Server brakowało bardzo `DECODE()`, które można używać w Oracle, a stosowanie w jego miejscu `CASE-WHEN` było dość uciążliwe i przede wszystkim, zabierało dość dużo miejsca, jeśli chce zadbać się o czytelność kodu.

Z kolejnych różnic, wymienić należy składnie polecenia `PIVOT`. W SQL Server każdą tabelę należało wzbogacić o alias, podczas gdy Oracle tego nie wymagało.

Kolejna, może i subtelna różnica, miała miejsce w przypadku tworzenia separatora tabeli w ostatnim zadaniu. W SQL Server brakowało prostego mechanizmu podobnego do `dual` z SZBD Oracle.

Bardzo widoczną różnicą (bez której niemożliwe byłoby wykonanie tych zadań, szczególnie w formie, którą wybrałem) jest konieczność jawnego dzielenia skryptu na batch'e w SQL Oracle przy pomocy instrukcji `GO`. W przypadku Oracle dzielenie to odbywało się automatycznie, z uwagi na fakt parsowania od razu całego pliku. Oba sposoby mogą być mniej lub bardziej wygodne, zależnie od kontekstu zastosowania, osobiście forma rozwiązania tego problemu przez SQL Server podoba mi się bardziej, gdyż daje użytkownikowi większą jawną kontrolę.

Istotną różnicą było także definiowanie zmiennych i możliwość komunikacji z użytkownikiem. W przypadku SZBD Oracle definiowanie zmiennych odbywało się (na potrzeby zadań) przez `&<variable_name>`, co sprawiało, że przy każdym uruchomieniu kwerendy wykorzystującej daną zmienną, system prosił użytkownika o wprowadzenie jej wartości, co stanowi warstwę interaktywności. W przypadku SQL Server niestety niemożliwe było osiągnięcie tego stopnia interaktywności (przynajmniej korzystając z wtyczki do Visual Studio Code). Deklarowanie i definiowanie zmiennych odbywało się poprzez instrukcję `DECLARE @<variable_name> <type>` a ustalanie jej wartości poprzez instrukcję `SET <variable_name> = <value>`, bezpośrednio w kodzie.