Raport lista 1

Mateusz Tereba 280556

Tworzenie bazy w Oracle

Tabela funkcje

```
CREATE TABLE Funkcje (
    funkcja VARCHAR2(10) PRIMARY KEY,
    min_myszy NUMBER(3) CHECK (min_myszy > 5),
    max_myszy NUMBER(3),
    CONSTRAINT chk_myszy CHECK (max_myszy < 200 AND max_myszy >= min_myszy)
);
```

Tabela wrogowie

```
CREATE TABLE Wrogowie (
    imie_wroga VARCHAR2(15) PRIMARY KEY,
    stopien_wrogosci NUMBER(2) CHECK (1 <= stopien_wrogosci AND
stopien_wrogosci <= 10),
    gatunek VARCHAR2(15),
    lapowka VARCHAR2(20)
);
```

Tabela bandy klucz obcy dodany pod koniec za pomocą alter

```
CREATE TABLE Bandy (
    nr_bandy NUMBER(2) PRIMARY KEY,
    nazwa VARCHAR2(20) not null,
    teren VARCHAR2(15) UNIQUE,
    szef_bandy VARCHAR2(15)
);
```

Tabela kocury DEFFERABLE INITIALLY DEFERRED pozwala odroczyć sprawdzanie ograniczeń aż do wywołania COMMIT

```
CREATE TABLE Kocury (
    imie VARCHAR2(15) not null,
    plec VARCHAR2(1) CHECK (plec = 'M' OR plec = 'D'),
    pseudo VARCHAR2(15) PRIMARY KEY,
    funkcja VARCHAR2(10), -- foregin key
```

```
    szef VARCHAR2(15), -- foregin key
    w_stadku_od DATE default SYSDATE, -- curernt date
    przydzial_myszy NUMBER(3),
    myszy_extra NUMBER(3),
    nr_bandy NUMBER(2), -- foregin key
    FOREIGN KEY(funkcja) REFERENCES Funkcje(funkcja),
    FOREIGN KEY(szef) REFERENCES Kocury(pseudo) DEFERRABLE INITIALLY
DEFERRED,
    FOREIGN KEY(nr_bandy) REFERENCES Bandy(nr_bandy) DEFERRABLE INITIALLY
DEFERRED
);
```

Tabela wrogowie

```
CREATE TABLE Wrogowie_kocurow (
    pseudo VARCHAR2(15),
    imie_wroga VARCHAR2(15),
    data_incydentu DATE not NULL,
    opis_incydentu VARCHAR2(50),
    PRIMARY KEY(pseudo, imie_wroga),
    FOREIGN KEY(pseudo) REFERENCES Kocury(pseudo),
    FOREIGN KEY(imie_wroga) REFERENCES Wrogowie(imie_wroga)
);
```

Dodanie klucza obcego do tabeli band za pomoca ALTER

```
ALTER TABLE Bandy ADD FOREIGN KEY(szef_bandy) REFERENCES Kocury(pseudo)
DEFERRABLE INITIALLY DEFERRED;
```

Tworzenie bazy w SQL Server

Tabela funkcji

```
CREATE TABLE Funkcje (
    funkcja VARCHAR(10) PRIMARY KEY,
    min_myszy DECIMAL(3) CHECK (min_myszy > 5),
    max_myszy DECIMAL(3),
    CONSTRAINT chk_myszy CHECK (max_myszy < 200 AND max_myszy >= min_myszy)
);
```

Tabela wrogów

```
CREATE TABLE Wrogowie (
    imie_wroga VARCHAR(15) PRIMARY KEY,
    stopien_wrogosci DECIMAL(2) CHECK (1 <= stopien_wrogosci AND
```

```
stopien_wrogosci <= 10),
    gatunek VARCHAR(15),
    lapowka VARCHAR(20)
);
```

Tabela band

Klucz obcy dodany na koniec za pomocą ALTER

```
CREATE TABLE Bandy (
    nr_bandy DECIMAL(2) PRIMARY KEY,
    nazwa VARCHAR(20) not null,
    teren VARCHAR(15) UNIQUE,
    szef_bandy VARCHAR(15)
);
```

Tabela kocurów

```
CREATE TABLE Kocury (
    imie VARCHAR(15) not null,
    plec VARCHAR(1) CHECK (plec = 'M' OR plec = 'D'),
    pseudo VARCHAR(15) PRIMARY KEY,
    funkcja VARCHAR(10), -- foregin key
    szef VARCHAR(15), -- foregin key
    w_stadku_od DATE default SYSDATETIME(), -- curernt date
    przydzial_myszy DECIMAL(3),
    myszy_extra DECIMAL(3),
    nr_bandy DECIMAL(2), -- foregin key
    CONSTRAINT fk_kocury_funkcje FOREIGN KEY(funkcja) REFERENCES
Funkcje(funkcja),
    CONSTRAINT fk_kocury_kocury FOREIGN KEY(szef) REFERENCES Kocury(pseudo),
    CONSTRAINT fk_kocury_bandy FOREIGN KEY(nr_bandy) REFERENCES
Bandy(nr_bandy)
);
```

Tabela wrogów kocurów

```
CREATE TABLE Wrogowie_kocurow (
    pseudo VARCHAR(15),
    imie_wroga VARCHAR(15),
    data_incydentu DATE not NULL,
    opis_incydentu VARCHAR(50),
    PRIMARY KEY(pseudo, imie_wroga),
    CONSTRAINT fk_wrogowie_kocurow_kocury FOREIGN KEY(pseudo) REFERENCES
Kocury(pseudo),
    CONSTRAINT fk_wrogowie_kocurow_wrogowie FOREIGN KEY(imie_wroga)
```

```
REFERENCES Wrogowie(imie_wroga)
);
```

Dodanie klucza obcego do tabeli band

```
ALTER TABLE Bandy ADD CONSTRAINT fk_bandy_kocury FOREIGN KEY(szef_bandy)
REFERENCES Kocury(pseudo);
```

Ładowanie danych w Oracle

Dane funkcji

```
INSERT INTO Funkcje(funkcja,min_myszy,max_myszy) VALUES
('SZEFUNIO',90,110),
('BANDZIOR',70,90),
('LOWCZY',60,70),
('LAPACZ',50,60),
('KOT',40,50),
('MILUSIA',20,30),
('DZIELCZY',45,55),
('HONOROWA',6,25)
;
```

Dane wrogów

```
INSERT INTO Wrogowie(imie_wroga,stopien_wrogosci,gatunek,lapowka) VALUES
('KAZIO',10,'CZLOWIEK','FLASZKA'),
('GLUPIA ZOSKA',1,'CZLOWIEK','KORALIK'),
('SWAWOLNY DYZIO',7,'CZLOWIEK','GUMA DO ZUCIA'),
('BUREK',4,'PIES','KOSC'),
('DZIKI BILL',10,'PIES',NULL),
('REKSIO',2,'PIES','KOSC'),
('BETHOVEN',1,'PIES','PEDIGRIPALL'),
('CHYTRUSEK',5,'LIS','KURCZAK'),
('SMUKLA',1,'SOSNA',NULL),
('BAZYLI',3,'KOGUT','KURA DO STADA')
;
```

Dane kocurów

```
INSERT INTO
Kocury(imie,plec,pseudo,funkcja,szef,w_stadku_od,przydzial_myszy,myszy_extra
,nr_bandy) VALUES
('JACEK','M','PLACEK','LOWCZY','LYSY',TO_DATE('2008-12-01', 'yyyy-mm-
dd'),67,NULL,2),
```

```
('BARI','M','RURA','LAPACZ','LYSY',TO_DATE('2009-09-01', 'yyyy-mm-
dd'),56,NULL,2),
('MICKA','D','LOLA','MILUSIA','TYGRYS',TO_DATE('2009-10-14', 'yyyy-mm-
dd'),25,47,1),
('LUCEK','M','ZERO','KOT','KURKA',TO_DATE('2010-03-01', 'yyyy-mm-
dd'),43,NULL,3),
('SONIA','D','PUSZYSTA','MILUSIA','ZOMBI',TO_DATE('2010-11-18', 'yyyy-mm-
dd'),20,35,3),
('LATKA','D','UCHO','KOT','RAFA',TO_DATE('2011-01-01', 'yyyy-mm-
dd'),40,NULL,4),
('DUDEK','M','MALY','KOT','RAFA',TO_DATE('2011-05-15', 'yyyy-mm-
dd'),40,NULL,4),
('MRUCZEK','M','TYGRYS','SZEFUNIO',NULL,TO_DATE('2002-01-01', 'yyyy-mm-
dd'),103,33,1),
('CHYTRY','M','BOLEK','DZIELCZY','TYGRYS',TO_DATE('2002-05-05', 'yyyy-mm-
dd'),50,NULL,1),
('KOREK','M','ZOMBI','BANDZIOR','TYGRYS',TO_DATE('2004-03-16', 'yyyy-mm-
dd'),75,13,3),
('BOLEK','M','LYSY','BANDZIOR','TYGRYS',TO_DATE('2006-08-15', 'yyyy-mm-
dd'),72,21,2),
('ZUZIA','D','SZYBKA','LOWCZY','LYSY',TO_DATE('2006-07-21', 'yyyy-mm-
dd'),65,NULL,2),
('RUDA','D','MALA','MILUSIA','TYGRYS',TO_DATE('2006-09-17', 'yyyy-mm-
dd'),22,42,1),
('PUCEK','M','RAFA','LOWCZY','TYGRYS',TO_DATE('2006-10-15', 'yyyy-mm-
dd'),65,NULL,4),
('PUNIA','D','KURKA','LOWCZY','ZOMBI',TO_DATE('2008-01-01', 'yyyy-mm-
dd'),61,NULL,3),
('BELA','D','LASKA','MILUSIA','LYSY',TO_DATE('2008-02-01', 'yyyy-mm-
dd'),24,28,2),
('KSAWERY','M','MAN','LAPACZ','RAFA',TO_DATE('2008-07-12', 'yyyy-mm-
dd'),51,NULL,4),
('MELA','D','DAMA','LAPACZ','RAFA',TO_DATE('2008-11-01', 'yyyy-mm-
dd'),51,NULL,4)
;
```

Dane band

```
INSERT INTO Bandy(nr_bandy,nazwa,teren,szef_bandy) VALUES
(1,'SZEFOSTWO','CALOSC','TYGRYS'),
(2,'CZARNI RYCERZE','POLE','LYSY'),
(3,'BIALI LOWCY','SAD','ZOMBI'),
(4,'LACIACI MYSLIWI','GORKA','RAFA'),
(5,'ROCKERSI','ZAGRODA',NULL)
;
```

Dane wrogów kocurów

```
INSERT INTO
Wrogowie_kocurow(pseudo,imie_wroga,data_incydentu,opis_incydentu) VALUES
('TYGRYS','KAZIO',TO_DATE('2004-10-13', 'yyyy-mm-dd'),'USILOWAL NABIC NA
WIDLY'),
('ZOMBI','SWAWOLNY DYZIO',TO_DATE('2005-03-07', 'yyyy-mm-dd'),'WYBIL OKO Z
PROCY'),
('BOLEK','KAZIO',TO_DATE('2005-03-29', 'yyyy-mm-dd'),'POSZCZUL BURKIEM'),
('SZYBKA','GLUPIA ZOSKA',TO_DATE('2006-09-12', 'yyyy-mm-dd'),'UZYLA KOTA
JAKO SCIERKI'),
('MALA','CHYTRUSEK',TO_DATE('2007-03-07', 'yyyy-mm-dd'),'ZALECAL SIE'),
('TYGRYS','DZIKI BILL',TO_DATE('2007-06-12', 'yyyy-mm-dd'),'USILOWAL
POZBAWIC ZYCIA'),
('BOLEK','DZIKI BILL',TO_DATE('2007-11-10', 'yyyy-mm-dd'),'ODGRYZL UCHO'),
('LASKA','DZIKI BILL',TO_DATE('2008-12-12', 'yyyy-mm-dd'),'POGRYZL ZE LEDWO
SIE WYLIZALA'),
('LASKA','KAZIO',TO_DATE('2009-01-07', 'yyyy-mm-dd'),'ZLAPAL ZA OGON I
ZROBIL WIATRAK'),
('DAMA','KAZIO',TO_DATE('2009-02-07', 'yyyy-mm-dd'),'CHCIAL OBEDRZEC ZE
SKORY'),
('MAN','REKSIO',TO_DATE('2009-04-14', 'yyyy-mm-dd'),'WYJATKOWO NIEGRZECZNIE
OBSZCZEKAL'),
('LYSY','BETHOVEN',TO_DATE('2009-05-11', 'yyyy-mm-dd'),'NIE PODZIELIL SIE
SWOJA KASZA'),
('RURA','DZIKI BILL',TO_DATE('2009-09-03', 'yyyy-mm-dd'),'ODGRYZL OGON'),
('PLACEK','BAZYLI',TO_DATE('2010-07-12', 'yyyy-mm-dd'),'DZIOBIAC
UNIEMOZLIWIL PODEBRANIEKURCZAKA'),
('PUSZYSTA','SMUKLA',TO_DATE('2010-11-19', 'yyyy-mm-dd'),'OBRZUCILA
SZYSZKAMI'),
('KURKA','BUREK',TO_DATE('2010-12-14', 'yyyy-mm-dd'),'POGONIL'),
('MALY','CHYTRUSEK',TO_DATE('2011-07-13', 'yyyy-mm-dd'),'PODEBRAL PODEBRANE
JAJKA'),
('UCHO','SWAWOLNY DYZIO',TO_DATE('2011-07-14', 'yyyy-mm-dd'),'OBRZUCIL
KAMIENIAMI')
;
```

Zatiwierdzenie transakcji

```
COMMIT;
```

Ładowanie danych w SQL Server

Dane funkcji

```
INSERT INTO Funkcje(funkcja,min_myszy,max_myszy) VALUES
('SZEFUNIO',90,110),
('BANDZIOR',70,90),
('LOWCZY',60,70),
```

```
('LAPACZ',50,60),
('KOT',40,50),
('MILUSIA',20,30),
('DZIELCZY',45,55),
('HONOROWA',6,25)
;
```

Dane wrogów

```
INSERT INTO Wrogowie(imie_wroga,stopien_wrogosci,gatunek,lapowka) VALUES
('KAZIO',10,'CZLOWIEK','FLASZKA'),
('GLUPIA ZOSKA',1,'CZLOWIEK','KORALIK'),
('SWAWOLNY DYZIO',7,'CZLOWIEK','GUMA DO ZUCIA'),
('BUREK',4,'PIES','KOSC'),
('DZIKI BILL',10,'PIES',NULL),
('REKSIO',2,'PIES','KOSC'),
('BETHOVEN',1,'PIES','PEDIGRIPALL'),
('CHYTRUSEK',5,'LIS','KURCZAK'),
('SMUKLA',1,'SOSNA',NULL),
('BAZYLI',3,'KOGUT','KURA DO STADA')
;
```

Wyłączenie sprawdzania ograniczeń dla klucza obcego w tabeli kocury

```
ALTER TABLE Kocury NOCHECK CONSTRAINT fk_kocury_bandy;
```

Dane kocurów

```
INSERT INTO
Kocury(imie,plec,pseudo,funkcja,szef,w_stadku_od,przydzial_myszy,myszy_extra
,nr_bandy) VALUES
('JACEK','M','PLACEK','LOWCZY','LYSY','2008-12-01',67,NULL,2),
('BARI','M','RURA','LAPACZ','LYSY','2009-09-01',56,NULL,2),
('MICKA','D','LOLA','MILUSIA','TYGRYS','2009-10-14',25,47,1),
('LUCEK','M','ZERO','KOT','KURKA','2010-03-01',43,NULL,3),
('SONIA','D','PUSZYSTA','MILUSIA','ZOMBI','2010-11-18',20,35,3),
('LATKA','D','UCHO','KOT','RAFA','2011-01-01',40,NULL,4),
('DUDEK','M','MALY','KOT','RAFA','2011-05-15',40,NULL,4),
('MRUCZEK','M','TYGRYS','SZEFUNIO',NULL,'2002-01-01',103,33,1),
('CHYTRY','M','BOLEK','DZIELCZY','TYGRYS','2002-05-05',50,NULL,1),
('KOREK','M','ZOMBI','BANDZIOR','TYGRYS','2004-03-16',75,13,3),
('BOLEK','M','LYSY','BANDZIOR','TYGRYS','2006-08-15',72,21,2),
('ZUZIA','D','SZYBKA','LOWCZY','LYSY','2006-07-21',65,NULL,2),
('RUDA','D','MALA','MILUSIA','TYGRYS','2006-09-17',22,42,1),
('PUCEK','M','RAFA','LOWCZY','TYGRYS','2006-10-15',65,NULL,4),
```

```
('PUNIA','D','KURKA','LOWCZY','ZOMBI','2008-01-01',61,NULL,3),
('BELA','D','LASKA','MILUSIA','LYSY','2008-02-01',24,28,2),
('KSAWERY','M','MAN','LAPACZ','RAFA','2008-07-12',51,NULL,4),
('MELA','D','DAMA','LAPACZ','RAFA','2008-11-01',51,NULL,4)
;
```

Dane band

```
INSERT INTO Bandy(nr_bandy,nazwa,teren,szef_bandy) VALUES
(1,'SZEFOSTWO','CALOSC','TYGRYS'),
(2,'CZARNI RYCERZE','POLE','LYSY'),
(3,'BIALI LOWCY','SAD','ZOMBI'),
(4,'LACIACI MYSLIWI','GORKA','RAFA'),
(5,'ROCKERSI','ZAGRODA',NULL)
;
```

Włączenie sprawdzania ograniczeń dla klucza obcego

```
ALTER TABLE Kocury WITH CHECK CHECK CONSTRAINT fk_kocury_bandy;
```

Dane wrogów kocurów

```
INSERT INTO
Wrogowie_kocurow(pseudo,imie_wroga,data_incydentu,opis_incydentu) VALUES
('TYGRYS','KAZIO','2004-10-13','USILOWAL NABIC NA WIDLY'),
('ZOMBI','SWAWOLNY DYZIO','2005-03-07','WYBIL OKO Z PROCY'),
('BOLEK','KAZIO','2005-03-29','POSZCZUL BURKIEM'),
('SZYBKA','GLUPIA ZOSKA','2006-09-12','UZYLA KOTA JAKO SCIERKI'),
('MALA','CHYTRUSEK','2007-03-07','ZALECAL SIE'),
('TYGRYS','DZIKI BILL','2007-06-12','USILOWAL POZBAWIC ZYCIA'),
('BOLEK','DZIKI BILL','2007-11-10','ODGRYZL UCHO'),
('LASKA','DZIKI BILL','2008-12-12','POGRYZL ZE LEDWO SIE WYLIZALA'),
('LASKA','KAZIO','2009-01-07','ZLAPAL ZA OGON I ZROBIL WIATRAK'),
('DAMA','KAZIO','2009-02-07','CHCIAL OBEDRZEC ZE SKORY'),
('MAN','REKSIO','2009-04-14','WYJATKOWO NIEGRZECZNIE OBSZCZEKAL'),
('LYSY','BETHOVEN','2009-05-11','NIE PODZIELIL SIE SWOJA KASZA'),
('RURA','DZIKI BILL','2009-09-03','ODGRYZL OGON'),
('PLACEK','BAZYLI','2010-07-12','DZIOBIAC UNIEMOZLIWIL PODEBRANIEKURCZAKA'),
('PUSZYSTA','SMUKLA','2010-11-19','OBRZUCILA SZYSZKAMI'),
('KURKA','BUREK','2010-12-14','POGONIL'),
('MALY','CHYTRUSEK','2011-07-13','PODEBRAL PODEBRANE JAJKA'),
('UCHO','SWAWOLNY DYZIO','2011-07-14','OBRZUCIL KAMIENIAMI')
;
```

Część 1 w Oracle

Zadanie 1

```
SELECT imie_wroga, DATA_INCYDENTU From Wrogowie_kocurow WHERE data_incydentu
>= '01-01-2009' AND data_incydentu <= '31-12-2009';
```

Zadanie 2

```
SELECT imie, w_stadku_od FROM KOCURY WHERE plec = 'D' AND W_STADKU_OD
BETWEEN '01-09-2005' AND '31-07-2007';
```

Zadanie 3

```
SELECT imie_wroga, gatunek, stopien_wrogosci FROM WROGOWIE WHERE LAPOWKA IS
NULL ORDER BY STOPIEN_WROGOSCI ASC;
```

Zadanie 4

```
SELECT
imie || ' zwany ' || pseudo || ' (fun. '  || funkcja || ') lowi myszki w
bandzie ' || nr_bandy ||  ' od ' || w_stadku_od
AS "Wszystko o Kocurach" From KOCURY Where PLEC = 'M';
```

Zadanie 5

REGEXP_REPLACE pozwala na zamianę podanego wystąpienia danego ciągu znaków od podanej litery , INSTR zwraca 0 jeśli nie znajduje podanej litery

```
SELECT pseudo, REGEXP_REPLACE(REGEXP_REPLACE(pseudo, 'L', '#', 1, 1), 'A',
'%', 1, 1) as Replacement
FROM KOCURY Where INSTR(pseudo, 'A') > 0 AND INSTR(pseudo, 'L') > 0;
```

Zadanie 6

EXTRACT pozwala na uzyskanie miesiąca z daty

```
SELECT imie, w_stadku_od, przydzial_myszy/1.1 as Zjadal,
ADD_MONTHS(w_stadku_od, 6) AS Podwyzka, przydzial_myszy as Zjada FROM KOCURY
where
(SYSDATE - w_stadku_od) / 365 > 15 AND
EXTRACT(MONTH FROM w_stadku_od) BETWEEN 3 AND 9
;
```

Zadanie 7

```
SELECT imie, przydzial_myszy * 3 as Myszy, myszy_extra * 3 as Dodatki FROM
KOCURY WHERE PRZYDZIAL_MYSZY > (MYSZY_EXTRA * 2) AND PRZYDZIAL_MYSZY >= 55;
```

Zadanie 8

COALESCE zwraca pierwszy argument nie będący nullem, jest to funkcja dostępna w Oracle i SQL Server

```
SELECT imie, CASE
    WHEN (przydzial_myszy + COALESCE(MYSZY_EXTRA, 0)) > 55 THEN
TO_CHAR((przydzial_myszy + COALESCE(MYSZY_EXTRA, 0)) * 12)
    WHEN (przydzial_myszy + COALESCE(MYSZY_EXTRA, 0)) = 55 THEN 'LIMIT'
    WHEN (przydzial_myszy + COALESCE(MYSZY_EXTRA, 0)) < 55 THEN 'PONIZEJ
660'
END AS Zjada_rocznie
FROM KOCURY;
```

Zadanie 9a

CONCAT pozwala na łączenie ciągów znaków w Oracle i SQL Server

```
SELECT CONCAT(pseudo,
CASE
    WHEN COUNT(PSEUDO) = 1 THEN ' - Unikalny'
    ELSE ' - Nie unikalny'
END) as Unikalny FROM KOCURY GROUP BY pseudo;
```

Zadanie 9b

```
SELECT CONCAT(SZEF,
CASE
    WHEN COUNT(SZEF) = 1 THEN ' - Unikalny'
    ELSE ' - Nie unikalny'
END) as Unikalny FROM KOCURY WHERE SZEF IS NOT NULL GROUP BY SZEF;
```

Zadanie 10

COUNT zlicza wystąpienia tej samej wartości w danej kolumnie, natomiast HAVING pozwala ograniczyć zwracane wartości

```
SELECT pseudo, COUNT(pseudo) as Liczba_worgow FROM Wrogowie_kocurow GROUP BY
PSEUDO HAVING Liczba_worgow >= 2;
```

Część 1 w SQL Server

Zadanie 1

BETWEEN zastępuje dwie nierówności

```
SELECT imie_wroga, DATA_INCYDENTU From Wrogowie_kocurow WHERE data_incydentu
BETWEEN '2009-01-01' AND '2009-12-31';
```

Zadanie 2

```
SELECT imie, w_stadku_od FROM KOCURY WHERE plec = 'D' AND W_STADKU_OD
BETWEEN '01-09-2005' AND '31-07-2007';
```

Zadanie 3

```
SELECT imie_wroga, gatunek, stopien_wrogosci FROM WROGOWIE WHERE LAPOWKA IS
NULL ORDER BY STOPIEN_WROGOSCI ASC;
```

Zadanie 4

```
SELECT Concat(
imie, ' zwany ', pseudo, ' (fun. ', funkcja, ') lowi myszki w bandzie ',
nr_bandy,  ' od ', w_stadku_od)
AS "Wszystko o Kocurach" From KOCURY Where PLEC = 'M';
```

Zadanie 5

CHARINDEX tylko w sql server zwraca 0 jeśli znaku nie ma w ciągu, STUFF pozwala na zamianę znaku na wybranej pozycji innym znakiem

```
SELECT pseudo, STUFF(STUFF(pseudo, CHARINDEX('L', pseudo), 1, '#'),
CHARINDEX('A', pseudo), 1, '%') as Replacement
FROM KOCURY Where CHARINDEX(pseudo, 'A') > 0 AND CHARINDEX(pseudo, 'L') > 0;
```

Zadanie 6

DATEADD dodaje wartość do daty we wskazanym interwalem, DATEDIFF oblicza rónice ze wskazanym interwałem

```
SELECT imie, w_stadku_od, przydzial_myszy/1.1 as Zjadal, DATEADD(month, 6,
w_stadku_od) AS Podwyzka, przydzial_myszy as Zjada FROM KOCURY where
    DATEDIFF(year, w_stadku_od, SYSDATETIME()) > 15
    AND MONTH(w_stadku_od) BETWEEN 3 AND 9;
;
```

Zadanie 7

```
SELECT imie, przydzial_myszy * 3 as Myszy, myszy_extra * 3 as Dodatek FROM
KOCURY WHERE PRZYDZIAL_MYSZY > (MYSZY_EXTRA * 2) AND PRZYDZIAL_MYSZY >= 55;
```

Zadanie 8

CONVERT pozwala zamieniać na wybrany typ, natomiast COALESCE zwraca pierwszy argument nie będący nullem jest w Oracle i SQL server

```
SELECT imie, CASE
    WHEN (przydzial_myszy + COALESCE(MYSZY_EXTRA, 0)) > 55 THEN
CONVERT(VARCHAR(15), przydzial_myszy + COALESCE(MYSZY_EXTRA, 0) * 12)
    WHEN (przydzial_myszy + COALESCE(MYSZY_EXTRA, 0)) = 55 THEN 'LIMIT'
    WHEN (przydzial_myszy + COALESCE(MYSZY_EXTRA, 0)) < 55 THEN 'PONIZEJ
660'
END AS Zjada_rocznie
FROM KOCURY;
```

Zadanie 9a CONCAT pozwala na łączenie ciągów znaków w Oracle i SQL Server

```
SELECT CONCAT(pseudo,
CASE
    WHEN COUNT(PSEUDO) = 1 THEN ' - Unikalny'
    ELSE ' - Nie unikalny'
END) as Unikalny FROM KOCURY GROUP BY pseudo;
```

Zadanie 9b

```
SELECT CONCAT(SZEF,
CASE
    WHEN COUNT(SZEF) = 1 THEN ' - Unikalny'
    ELSE ' - Nie unikalny'
END) as Unikalny FROM KOCURY WHERE SZEF IS NOT NULL GROUP BY SZEF;
```

Zadanie 10

COUNT zlicza wystąpienia tej samej wartości w danej kolumnie, natomiast HAVING pozwala ograniczyć zwracane wartości

```
SELECT pseudo, COUNT(pseudo) as Liczba_worgow FROM Wrogowie_kocurow GROUP BY
PSEUDO HAVING COUNT(pseudo) >= 2;
```

Część 2 w Oracle

Zadanie 11a

jeśli następna środa jest w innym miesiącu szukaj ostatniej środy w następnym miesiącu w przeciwnym razie szukaj w postępuj zgodnie z zasadami.

```
SELECT pseudo, w_stadku_od,
CASE
    WHEN EXTRACT(MONTH FROM NEXT_DAY(DATE '2024-10-29', 'WEDNESDAY')) =
EXTRACT(MONTH FROM DATE '2024-10-29')
    THEN
        CASE
            WHEN EXTRACT(DAY FROM W_STADKU_OD) <= 15 THEN
NEXT_DAY(LAST_DAY(DATE '2024-10-29') - 7, 'WEDNESDAY')
            ELSE NEXT_DAY(LAST_DAY(ADD_MONTHS(DATE '2024-10-29', 1)) - 7,
'WEDNESDAY')
        END
    ELSE NEXT_DAY(LAST_DAY(ADD_MONTHS(DATE '2024-10-29', 1)) - 7,
'WEDNESDAY')
END AS wyplata
FROM KOCURY ORDER BY W_STADKU_OD;
```

Zadanie 11b

```
SELECT pseudo, w_stadku_od,
CASE
    WHEN EXTRACT(MONTH FROM NEXT_DAY(DATE '2024-10-31', 'WEDNESDAY')) =
EXTRACT(MONTH FROM DATE '2024-10-31')
    THEN
        CASE
            WHEN EXTRACT(DAY FROM W_STADKU_OD) <= 15 THEN
NEXT_DAY(LAST_DAY(DATE '2024-10-31') - 7, 'WEDNESDAY')
            ELSE NEXT_DAY(LAST_DAY(DATE '2024-10-31') - 7, 'WEDNESDAY')
        END
    ELSE NEXT_DAY(LAST_DAY(ADD_MONTHS(DATE '2024-10-31', 1)) - 7,
'WEDNESDAY')
END AS wyplata
FROM KOCURY ORDER BY W_STADKU_OD;
```

Zadanie 12

```
SELECT
'Liczba kotow=' || COUNT(funkcja) || ' lowi jako ' || funkcja || ' i zjada
max. ' || MAX(COALESCE(PRZYDZIAL_MYSZY, 0) + COALESCE(MYSZY_EXTRA, 0)) || '
myszy miesiecznie'
```

```
FROM Kocury WHERE
funkcja != 'SZEFUNCIO' AND plec != 'M' GROUP BY funkcja HAVING
AVG(COALESCE(PRZYDZIAL_MYSZY, 0) + COALESCE(MYSZY_EXTRA, 0)) > 50;
```

Zadanie 13

```
SELECT nr_bandy, plec, MIN(PRZYDZIAL_MYSZY) FROM KOCURY GROUP BY NR_BANDY,
PLEC;
```

Część 3 w Oracle

Zadanie 14

CONNECT BY PRIOR pozwala na utworzenie struktury drzewa gdzie węzły są połączone kiedy szef kota jest równy pseudonimowi kota stanowiącego węzeł. Starts with pozwala określić korzeń drzewa, jeśli wiele rekordów spełnia ten warunek utworzone jest wiele drzew.

```
SELECT level, pseudo, funkcja, nr_bandy
From KOCURY WHERE plec = 'M'
START WITH funkcja = 'BANDZIOR' CONNECT by Prior PSEUDO = szef;
```

Zadanie 15

Level pozwala na określenie poziomu drzewa na którym jest dany rekord.

```
SELECT LPAD(level - 1 || ' ' || imie, (level - 1) * 4 + Length(level - 1 ||
imie) + (level-1)/10, '===>'), SZEF, funkcja FROM KOCURY
WHERE COALESCE(MYSZY_EXTRA, 0) != 0
START WITH SZEF IS NULL
CONNECT BY PRIOR PSEUDO = SZEF;
```

Zadanie 16

Aby wyświetlić hierarchię kotów należy zacząć od kotów spełniających zadane warunki, a następnie wybrać koty których pseudonim jest równy pseudonimowi szefa danego rekordu.

```
SELECT LPAD(pseudo, 4 * (level-1) + LENGTH(pseudo))
FROM KOCURY
START WITH plec = 'M'
AND COALESCE(MYSZY_EXTRA, 0) = 0
AND DATE '2024-07-17' - W_STADKU_OD > 365 * 15
CONNECT BY PRIOR SZEF = PSEUDO
```

	LPAD(PSEUDO,4*(LEVEL-1)+LENGTH(PSEUDO))
1	BOLEK
2	TYGRYS
3	MAN
4	RAFA
5	TYGRYS
6	PLACEK
7	LYSY
8	TYGRYS
9	RAFA
10	TYGRYS

Podsumowanie

Praca z bazą danych Oracle była łatwiejsza ze względu na wygodne funkcje których nie ma w SQL Server, ładowanie danych w Oracle wydaje się być bardziej przemyślane za pomocą DEFFERABLE INNITIALLY DEFFERABLE i COMMIT, natomiast w SQL Server jedynym sensownym sposobem na wprowadzenie tych jaki znalazłem jest wyłączenie ograniczeń, wstawienie danych i włączenie ich ze sprawdzeniem danych.
Największe różnice występują przy tworzeniu tabel i ich usuwaniu, typy danych różnią się pomiędzy bazami, usuwanie tabel jest o wiele prostsze w Oracle ponieważ można usuwać powiązania pomiędzy tabelami razem z tabelami, w SQL Server należy usunąć najpierw ograniczenia a następnie usunąć tabele.
Wiele funkcji jest dostępnych w obu dialektach np. COALESCE(), CONCAT() jednak Oracle i SQL Serwer posiadają różniące się odpowiedniki dla COALESCE() są to NVL() i ISNULL(), natomiast dla CONCAT() są to operatory || i +. Co ułatwia przenoszenie zapytań pomiędzy bazami, niestety nie znalazłem uniwersalnych odpowiedników funkcji używanych do obsługi dat.