

Akademia Górniczo-Hutnicza w Krakowie
Wydział Informatyki, Elektroniki i Telekomunikacji



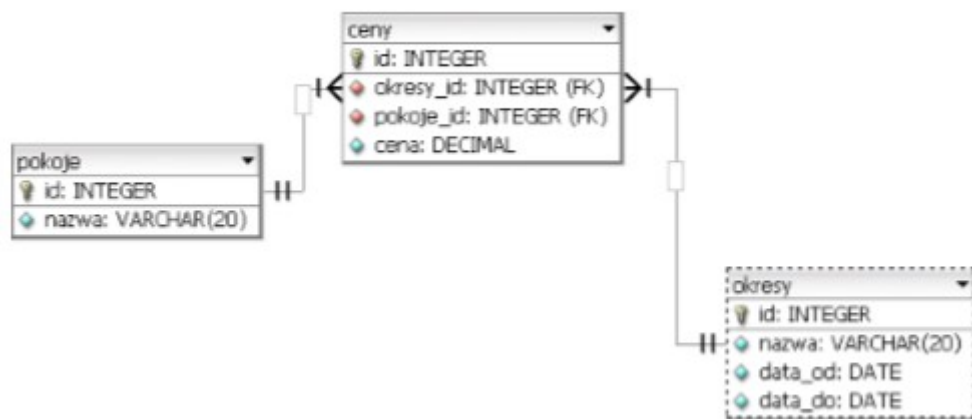
Podstawy baz danych - projekt i implementacja systemu bazodanowego

Katedra Informatyki
Studia inżynierskie
Drugi rok

Robert Bielas
Dominik Hendzel

Zadanie 2

Na podstawie danej struktury danych:



nalegało napisać procedurę dokonującą wyceny pobytu w hotelu.

1. Utworzenie tabel:

```
CREATE TABLE pokoje (  
    id INTEGER NOT NULL ,  
    nazwa VARCHAR(20) NULL,  
    PRIMARY KEY(id) );
```

```
CREATE TABLE okresy (  
    id INTEGER NOT NULL ,  
    nazwa VARCHAR(20) NULL,  
    data_od DATE NULL,  
    data_do DATE NULL,  
    PRIMARY KEY(id) );
```

```
CREATE TABLE ceny (  
    id INTEGER NOT NULL ,  
    okresy_id INTEGER NOT NULL,  
    pokoje_id INTEGER NOT NULL,  
    cena DECIMAL NULL,  
    PRIMARY KEY(id),  
    INDEX ceny_FKIndex1(pokoje_id),  
    INDEX ceny_FKIndex2(okresy_id),  
    FOREIGN KEY(pokoje_id) REFERENCES pokoje(id),  
    FOREIGN KEY(okresy_id) REFERENCES okresy(id) );
```

2. Wypełnienie tabel danymi:

Przy generowaniu przykładowych danych posłużyliśmy się językiem Python.

Plik main.py, „korzeń” generatora:

```
from Cleaning import Storage
from pokoje import wypelnijPokoje
from okresy import wypelnijOkresy

store = Storage()
w = store.w
w.write("use bielas_a\n"
        "delete from ceny\n"
        "delete from pokoje\n"
        "delete from okresy\n\n")
wypelnijPokoje(store)
wypelnijOkresy(store)
```

plik Cleaning.py, który zawiera pomocniczą strukturę danych - klasę Storage:

```
from datetime import date
class Storage:
    def __init__(self):
        self.w = open("Workfile.sql", "w")
        self.room_names =
["dwuosobowy", "trzyosobowy", "czterooosobowy", "rodzinny", "dla
zakochanych", "apartament", "jednoosobowy", "z lazienka", "z kuchnia", "z kuchnia
i lazienka"]
        self.period_names = {"lato": [date(2000, 6, 1), date(2000, 8, 31)],
                              "jesien": [date(2000, 9, 1), date(2000, 11, 30)],
                              "zima": [date(2000, 12, 1), date(2001, 2, 28)],
                              "wiosna": [date(2000, 3, 1), date(2000, 5, 31)]}
        self.adjectives_fem = ["cicha ", "urokliwa ", " tania ", "eko ",
                                "obfita ", "lagodna ",
                                "wyrozumiala ", "wymarzona "]
        self.adjectives_mas = ["ciche ", "urokliwe ", "tanie ", "eko "]
        self.years_amount = 5
        self.room_id = 0
        self.period_id = 0
        self.price_id = 0
```

Plik pokoje.py (nazwy tego pliku i następnych sugerują, jakie tabelki zostały przez nie wypełnione):

```
def wypelnijPokoje(store):
    w = store.w
    for name in store.room_names:
        w.write("insert into pokoje values(" +
                str(store.room_id) + ", " +
                "'" + name + "'" +
                ")\n")
        store.room_id += 1
```

Plik ceny.py

```
from random import randint
def getRoomId(marked, store):
    N = store.room_id - 1
    res = randint(0, N)
    while res in marked:
        res = randint(0, N)
    marked.append(res)
    return res, marked

def wypelnijCeny(store):
    w = store.w
    marked = []
    for i in range(store.room_id):
        price = randint(50, 80)
        room_id, marked = getRoomId(marked, store)
        w.write("insert into ceny values(" +
            str(store.price_id) + "," +
            str(store.period_id) + "," +
            str(room_id) + "," +
            str(price) +
            ")\n")
        store.price_id += 1
```

Plik okresy.py:

```
from datetime import timedelta

from random import randint
from ceny import wypelnijCeny
def getAdj(adjList):
    N = len(adjList) - 1
    return adjList[randint(0, N)]
def wypelnijOkresy(store):
    w = store.w
    for i in range(store.years_amount):
        for name in store.period_names.keys():
            if name == "lato":
                adjList = store.adjectives_mas
            else:
                adjList = store.adjectives_fem
            adj = getAdj(adjList)
            w.write("insert into okresy values(" +
                str(store.period_id) + "," +
                "'" + adj + name + "'," +
                "'" + str(store.period_names[name][0]) + timedelta(days = 365
* i)) + "'," +
                "'" + str(store.period_names[name][1]) + timedelta(days = 365
* i)) + "'" +
                ")\n")
            wypelnijCeny(store)
            store.period_id += 1
```

Kod procedury:

```
create procedure fill_Result2 @roomId int, @start date, @end date
as
    declare @name varchar(20), @daysAmount int, @price real, @value real, @stake_price
        real
    declare @res table([Id pokoju] int, [Nazwa pokoju] varchar(20), [Pobyt od] date,
        [Pobyt do] date, [liczba dni] int, cena real, wartosc real)
    declare @starting_date date, @ending_date date, @from_date date, @to_date date
    set @name = (select nazwa from pokoje where id = @roomId)

    while @start <= @end
    begin
        set @starting_date = @start
        set @from_date =
            (select top 1 data_od
             from okresy
             where data_od <= @starting_date
             order by 1 desc )
        set @to_date =
            (select data_do
             from okresy
             where data_od = @from_date)
        set @ending_date = dbo.minimum_date(@end, @to_date)
        set @start =
            (select top 1 data_od
             from okresy
             where data_od > @ending_date
             order by 1 asc)
        set @stake_price =
            (select cena
             from ceny as c join pokoje as p
             on p.id = c.pokoje_id
             join okresy as o
             on c.okresy_id = o.id
             where data_od = @from_date and p.id = @roomId)
        set @daysAmount = datediff(d, @starting_date, @ending_date) + 1
        set @value = @daysAmount * @stake_price
        insert into @res values(@roomId, @name, @starting_date,
            @ending_date, @daysAmount, @stake_price, @value)
    end
    select * from @res
go
```

3. Przeprowadzone testy:

```
execute fill_Result2 0, '2002-09-04', '2002-09-21'
```

	Id pokoju	Nazwa pokoju	Pobyt od	Pobyt do	liczba dni	cena	wartosc
1	0	dwuosobowy	2002-09-04	2002-09-21	18	55	990

```
execute fill_Result2 0, '2004-09-04', '2002-09-21'
```

	Id pokoju	Nazwa pokoju	Pobyt od	Pobyt do	liczba dni	cena	wartosc
--	-----------	--------------	----------	----------	------------	------	---------

```
execute fill_Result2 0, '2002-09-04', '2003-08-21'
```

	Id pokoju	Nazwa pokoju	Pobyt od	Pobyt do	liczba dni	cena	wartosc
1	0	dwuosobowy	2002-09-04	2002-11-30	88	55	4840
2	0	dwuosobowy	2002-12-01	2003-02-28	90	70	6300
3	0	dwuosobowy	2003-03-01	2003-05-31	92	65	5980
4	0	dwuosobowy	2003-06-01	2003-08-21	82	80	6560

Pierwszy test pokazuje sytuację, kiedy wprowadzono poprawne dane (data_od < data_do), kiedy obowiązywał jeden próg cenowy (pobyt „zmieścił się” w jednym okresie/sezonie).

Drugi test pokazuje, że gdy wprowadzimy niepoprawne dane (tzn. data_od > data_do) procedura nie zakończy się rzuceniem wyjątku, tylko zwróci pustą tabelkę wyniku.

Trzeci test pokazuje najciekawszą sytuację, gdy np. klient zdecyduje się wynająć pokój na przełomie sezonów i w okresie jego pobytu ceny za dany pokój zmieniają się. Wtedy procedura zwraca tabelkę wypełnioną więcej niż jednym wierszem, a każdy wiersz opisuje sytuację/koszt za pokój w każdym z przedziałów czasowych, kiedy obowiązywały różne stawki za pokój.

Pomocnicze polecenia sql:

- Polecenie zwracające informacje o pokoju i posortowane okresy, w których obowiązują dane ceny:

```
select p.id, p.nazwa, o.id,o.nazwa, cena,data_od,data_do
from pokoje as p join ceny as c
on c.pokoje_id = p.id
join okresy as o
on o.id = c.okresy_id
order by 1,6
```

- Polecenie zwracające obowiązującą cenę za pokój, gdy wynajęto go na określoną datę, przykładowo dla 2000-03-15 i pokoju o id 0

```
select cena
from ceny as c join pokoje as p
on p.id = c.pokoje_id
join okresy as o
on c.okresy_id = o.id
where data_od =
    (select top 1 data_od
     from okresy
     where data_od <= '2000-03-15'
     order by 1 desc ) and p.id = 0
```

- Polecenie zwraca początek następnego sezonu:

```
select top 1 data_od
from okresy
where data_od > '2000-05-31'
order by 1 asc
```

- Dla danej daty rozpoczęcia sezonu, polecenie zwraca jego koniec

```
select data_do
from okresy
where data_od = '2001-06-01'
```

Pomocnicza procedura zwracająca minimum z dwóch podanych na wejście dat:

```
create function minimum_date(@val1 date, @val2 date)
returns date
as
begin
    if @val1 < @val2
        return @val1
    return isnull(@val2,@val1)
end
go
```

Skrypt z przykładowymi danymi wygenerowanymi przez pythonowski generator, na których przeprowadziliśmy powyższe trzy testy:

```
delete from ceny
delete from pokoje
delete from okresy

insert into pokoje values(0, 'dwuosobowy')
insert into pokoje values(1, 'trzyosobowy')
insert into pokoje values(2, 'czterooosobowy')
insert into pokoje values(3, 'rodzinny')
insert into pokoje values(4, 'dla zakochanych')
insert into pokoje values(5, 'apartament')
insert into pokoje values(6, 'jednoosobowy')
insert into pokoje values(7, 'z lazienka')
insert into pokoje values(8, 'z kuchnia')
insert into pokoje values(9, 'z kuchnia i lazienka')
insert into okresy values(0, 'wyrozumiala jesien', '2000-09-01', '2000-11-30')
insert into ceny values(0,0,0,68)
insert into ceny values(1,0,9,73)
insert into ceny values(2,0,5,64)
insert into ceny values(3,0,6,55)
insert into ceny values(4,0,2,62)
insert into ceny values(5,0,7,60)
insert into ceny values(6,0,4,60)
insert into ceny values(7,0,3,57)
insert into ceny values(8,0,8,79)
insert into ceny values(9,0,1,51)
insert into okresy values(1, 'tanie lato', '2000-06-01', '2000-08-31')
insert into ceny values(10,1,2,71)
insert into ceny values(11,1,4,70)
insert into ceny values(12,1,0,75)
insert into ceny values(13,1,5,65)
insert into ceny values(14,1,8,74)
insert into ceny values(15,1,3,63)
insert into ceny values(16,1,7,65)
insert into ceny values(17,1,1,65)
insert into ceny values(18,1,9,74)
insert into ceny values(19,1,6,57)
insert into okresy values(2, 'tania zima', '2000-12-01', '2001-02-28')
insert into ceny values(20,2,7,63)
insert into ceny values(21,2,0,63)
insert into ceny values(22,2,5,70)
insert into ceny values(23,2,3,62)
insert into ceny values(24,2,8,50)
insert into ceny values(25,2,1,56)
insert into ceny values(26,2,2,61)
insert into ceny values(27,2,9,68)
insert into ceny values(28,2,4,65)
insert into ceny values(29,2,6,57)
insert into okresy values(3, 'urokliwa wiosna', '2000-03-01', '2000-05-31')
insert into ceny values(30,3,7,69)
insert into ceny values(31,3,2,63)
insert into ceny values(32,3,1,60)
insert into ceny values(33,3,0,65)
insert into ceny values(34,3,6,78)
insert into ceny values(35,3,4,75)
insert into ceny values(36,3,3,73)
insert into ceny values(37,3,5,52)
insert into ceny values(38,3,8,75)
insert into ceny values(39,3,9,54)
insert into okresy values(4, 'cicha jesien', '2001-09-01', '2001-11-30')
insert into ceny values(40,4,1,52)
insert into ceny values(41,4,0,62)
```

```

insert into ceny values(42,4,5,70)
insert into ceny values(43,4,8,61)
insert into ceny values(44,4,7,57)
insert into ceny values(45,4,4,52)
insert into ceny values(46,4,2,66)
insert into ceny values(47,4,9,57)
insert into ceny values(48,4,3,52)
insert into ceny values(49,4,6,59)
insert into okresy values(5,'ciche lato','2001-06-01','2001-08-31')
insert into ceny values(50,5,6,50)
insert into ceny values(51,5,5,59)
insert into ceny values(52,5,9,68)
insert into ceny values(53,5,0,57)
insert into ceny values(54,5,4,60)
insert into ceny values(55,5,2,63)
insert into ceny values(56,5,3,73)
insert into ceny values(57,5,1,68)
insert into ceny values(58,5,7,69)
insert into ceny values(59,5,8,68)
insert into okresy values(6,'eko zima','2001-12-01','2002-02-28')
insert into ceny values(60,6,8,56)
insert into ceny values(61,6,6,72)
insert into ceny values(62,6,2,56)
insert into ceny values(63,6,7,71)
insert into ceny values(64,6,4,59)
insert into ceny values(65,6,0,50)
insert into ceny values(66,6,1,53)
insert into ceny values(67,6,3,72)
insert into ceny values(68,6,5,65)
insert into ceny values(69,6,9,76)
insert into okresy values(7,'lagodna wiosna','2001-03-01','2001-05-31')
insert into ceny values(70,7,9,65)
insert into ceny values(71,7,4,68)
insert into ceny values(72,7,6,55)
insert into ceny values(73,7,3,59)
insert into ceny values(74,7,1,70)
insert into ceny values(75,7,5,75)
insert into ceny values(76,7,0,51)
insert into ceny values(77,7,2,58)
insert into ceny values(78,7,7,63)
insert into ceny values(79,7,8,78)
insert into okresy values(8,'cicha jesien','2002-09-01','2002-11-30')
insert into ceny values(80,8,4,55)
insert into ceny values(81,8,5,77)
insert into ceny values(82,8,9,72)
insert into ceny values(83,8,7,58)
insert into ceny values(84,8,2,66)
insert into ceny values(85,8,0,55)
insert into ceny values(86,8,3,67)
insert into ceny values(87,8,6,71)
insert into ceny values(88,8,1,73)
insert into ceny values(89,8,8,80)
insert into okresy values(9,'urokliwe lato','2002-06-01','2002-08-31')
insert into ceny values(90,9,2,80)
insert into ceny values(91,9,0,69)
insert into ceny values(92,9,3,61)
insert into ceny values(93,9,1,79)
insert into ceny values(94,9,9,56)
insert into ceny values(95,9,7,54)
insert into ceny values(96,9,8,50)
insert into ceny values(97,9,5,68)
insert into ceny values(98,9,4,54)
insert into ceny values(99,9,6,77)
insert into okresy values(10,'wrozumiala zima','2002-12-01','2003-02-28')
insert into ceny values(100,10,3,72)

```



```

insert into ceny values(101,10,5,76)
insert into ceny values(102,10,9,67)
insert into ceny values(103,10,1,52)
insert into ceny values(104,10,4,62)
insert into ceny values(105,10,2,68)
insert into ceny values(106,10,6,65)
insert into ceny values(107,10,7,77)
insert into ceny values(108,10,8,66)
insert into ceny values(109,10,0,70)
insert into okresy values(11,'cicha wiosna','2002-03-01','2002-05-31')
insert into ceny values(110,11,7,52)
insert into ceny values(111,11,8,77)
insert into ceny values(112,11,6,56)
insert into ceny values(113,11,3,52)
insert into ceny values(114,11,9,76)
insert into ceny values(115,11,0,58)
insert into ceny values(116,11,1,62)
insert into ceny values(117,11,5,64)
insert into ceny values(118,11,4,59)
insert into ceny values(119,11,2,72)
insert into okresy values(12,'tania jesien','2003-09-01','2003-11-30')
insert into ceny values(120,12,8,58)
insert into ceny values(121,12,7,60)
insert into ceny values(122,12,9,66)
insert into ceny values(123,12,4,51)
insert into ceny values(124,12,0,62)
insert into ceny values(125,12,3,73)
insert into ceny values(126,12,2,60)
insert into ceny values(127,12,5,58)
insert into ceny values(128,12,6,68)
insert into ceny values(129,12,1,66)
insert into okresy values(13,'tanie lato','2003-06-01','2003-08-31')
insert into ceny values(130,13,3,63)
insert into ceny values(131,13,2,58)
insert into ceny values(132,13,0,80)
insert into ceny values(133,13,5,56)
insert into ceny values(134,13,4,55)
insert into ceny values(135,13,8,59)
insert into ceny values(136,13,7,53)
insert into ceny values(137,13,6,57)
insert into ceny values(138,13,1,51)
insert into ceny values(139,13,9,67)
insert into okresy values(14,'wrozumiala zima','2003-12-01','2004-02-28')
insert into ceny values(140,14,9,50)
insert into ceny values(141,14,7,59)
insert into ceny values(142,14,8,62)
insert into ceny values(143,14,0,51)
insert into ceny values(144,14,6,53)
insert into ceny values(145,14,5,57)
insert into ceny values(146,14,2,67)
insert into ceny values(147,14,4,52)
insert into ceny values(148,14,1,70)
insert into ceny values(149,14,3,66)
insert into okresy values(15,'cicha wiosna','2003-03-01','2003-05-31')
insert into ceny values(150,15,7,67)
insert into ceny values(151,15,6,62)
insert into ceny values(152,15,2,76)
insert into ceny values(153,15,1,69)
insert into ceny values(154,15,0,65)
insert into ceny values(155,15,4,52)
insert into ceny values(156,15,8,77)
insert into ceny values(157,15,5,54)
insert into ceny values(158,15,3,74)
insert into ceny values(159,15,9,79)
insert into okresy values(16,'obfita jesien','2004-08-31','2004-11-29')

```

```
insert into ceny values(160,16,9,57)
insert into ceny values(161,16,6,74)
insert into ceny values(162,16,5,62)
insert into ceny values(163,16,8,59)
insert into ceny values(164,16,0,60)
insert into ceny values(165,16,2,63)
insert into ceny values(166,16,3,58)
insert into ceny values(167,16,1,65)
insert into ceny values(168,16,7,50)
insert into ceny values(169,16,4,57)
insert into okresy values(17, 'tanie lato', '2004-05-31', '2004-08-30')
insert into ceny values(170,17,4,71)
insert into ceny values(171,17,0,75)
insert into ceny values(172,17,5,69)
insert into ceny values(173,17,1,61)
insert into ceny values(174,17,6,76)
insert into ceny values(175,17,2,59)
insert into ceny values(176,17,9,56)
insert into ceny values(177,17,7,71)
insert into ceny values(178,17,3,70)
insert into ceny values(179,17,8,53)
insert into okresy values(18, 'obfita zima', '2004-11-30', '2005-02-27')
insert into ceny values(180,18,2,70)
insert into ceny values(181,18,0,61)
insert into ceny values(182,18,9,76)
insert into ceny values(183,18,3,74)
insert into ceny values(184,18,1,69)
insert into ceny values(185,18,4,50)
insert into ceny values(186,18,7,63)
insert into ceny values(187,18,5,53)
insert into ceny values(188,18,6,79)
insert into ceny values(189,18,8,75)
insert into okresy values(19, 'wyrozumiala wiosna', '2004-02-29', '2004-05-30')
insert into ceny values(190,19,2,68)
insert into ceny values(191,19,7,54)
insert into ceny values(192,19,0,55)
insert into ceny values(193,19,1,55)
insert into ceny values(194,19,6,67)
insert into ceny values(195,19,9,63)
insert into ceny values(196,19,3,51)
insert into ceny values(197,19,4,52)
insert into ceny values(198,19,8,62)
insert into ceny values(199,19,5,78)
```