



JPS Ćw 1

Praca z obiektową bazą danych ODRA i językiem SBQL

INSTALACJA

Obiektowa baza danych Odra jest dostarczona w postaci archiwum ze skompilowanymi bibliotekami Java (Jar) oraz skryptami pozwalającymi na jej uruchomienie. Do poprawnego działania potrzebna jest Java (JRE) w wersji 1.5 lub nowszej.

Uruchomienie

W głównym katalogu umieszczone zostały skrypty (zarówno dla Windows jak i Unix/MacOS) pozwalające na uruchomienie bazy danych. Dwa podstawowe polecenia to:

- server.bat – uruchamia serwer bazy danych. Dwa podstawowe parametry przekazywane do komendy:
 - -- start_expandable nazwa_bazy
 - --create_expandable nazwa_bazy 5000000 5000000 5000000

Pierwsza uruchamia serwer przy pomocy plików systemowych zapisanych pod daną nazwą. Druga, istotna przy pierwszym użyciu bazy, tworzy niezbędne pliki systemowe do których potem serwer zapisuje swoje dane.

- cli.bat – uruchamia konsolowego klienta bazy danych łączącego się z serwerem.

Dla wygody użytkownika powstał skrypt easystart.bat który jednocześnie tworzy nowe pliki bazy danych, uruchamia serwer przy ich pomocy i uruchamia konsolowego klienta.



PRZYKŁADOWA BAZA DANYCH

Do wczytania bazy danych posłużymy się komendą batch oraz przykładowym skryptem dostępnym w katalogu „./res/sampledata/batch/m0/” M0data.cli

```
batch \res\sampledata\batch\m0\M0data.cli
```

Skrypt tworzy struktury do przechowywania danych:

```
Address {
    city:string;
    street:string;
    zip:string[0..1];
}

Person {
    fName:string;
    lName:string;
    age:integer;
    married:boolean;
    address:Address;
    worksIn:Firm[0..1];
}

Car {
    model:string;
    color:string;
    power:integer;
    owner:Person[0..*];
    year:date;
}

Firm {
    name:string;
    address:Address;
}
```

Skrypt tworzy przykładowe dane oraz kompiluje moduł co pozwala od razu wysłać do niego zapytania.

Aby zobaczyć zawartość modułu wykonaj polecenie ls



OPERATORY

Kategoria	Operator
Na liczbach	+ - * / %
Logiczne	and or not
Konkatenacja	+
	Przykładowo: Person.fName + " ma kota"
Porównanie	= <> > < >= <=
Operatory nazwowe	as groupas
	Przykładowo: (Car as x).(x.model);
Operatory dat	now dateprec
Agregujące	sum count min max avg
	Przykładowo: avg (Person.age)
Na zbiorach	bag union struct , (przecinek) subtract in contains intersect unique distinct exists
	Przykładowo: Person.(fName, lName); unique (Person.address.city);
Nie-algebraiczne na zbiorach	.(kropka) join where forall forsome orderby closeby
	Przykładowo: Person.fName; Car where power>300; forall (Person.worksIn.address) (city="Warszawa");
Referencje	ref deref
Warunkowe	if ... then ... else ...
	Przykładowo: (if (forall(Person)(married)) then("tak") else("nie"));



ZAPYTANIA

Dla przykładowej bazy danych wylicz:

1. Policz $1 + 2 * 3 + 4$
2. Policz ile wynosi reszta z dzielenia 435 przez 3
3. Sprawdź czy 6549 dzieli się przez 4.
4. Wypisz nazwiska wszystkich pracowników. Posortuj tą listę.
5. Wypisz wszystkie samochody czerwone o mocy większej od 200
6. Wypisz na ekran przywitanie wszystkich pracowników po imieniu np. „Hello John!”;
7. Wypisz dla każdego pracownika nazwisko i ulicę na której pracuje.
8. Sprawdź, czy wszyscy pracownicy są zamężni/żonaci (użyj forall)
9. Policz średnią wieku mieszkających w Katowicach
10. Wypisz wszystkich pracujących w Warszawie (potrzeba użyć forsome)
11. Policz ile jest w bazie danych firm (użyj count)
12. Dla każdej firmy wypisz jej imię i liczbę osób w niej pracujących (użyj „Firm as f”)
13. Wypisz wszystkie miasta w których mieszkają pracownicy (użyj unique)
14. Dla każdego miasta wypisz średnią wieku jej mieszkańców.
15. Wypisz pracowników na zasadzie „wolny/wolna”Kowalski „zamężna/żonaty”Nowak.