



# Hurtownie danych - przegląd technologii

**Robert Wrembel**  
**Politechnika Poznańska**  
**Instytut Informatyki**

[Robert.Wrembel@cs.put.poznan.pl](mailto:Robert.Wrembel@cs.put.poznan.pl)  
[www.cs.put.poznan.pl/rwrembel](http://www.cs.put.poznan.pl/rwrembel)



## MOLAP

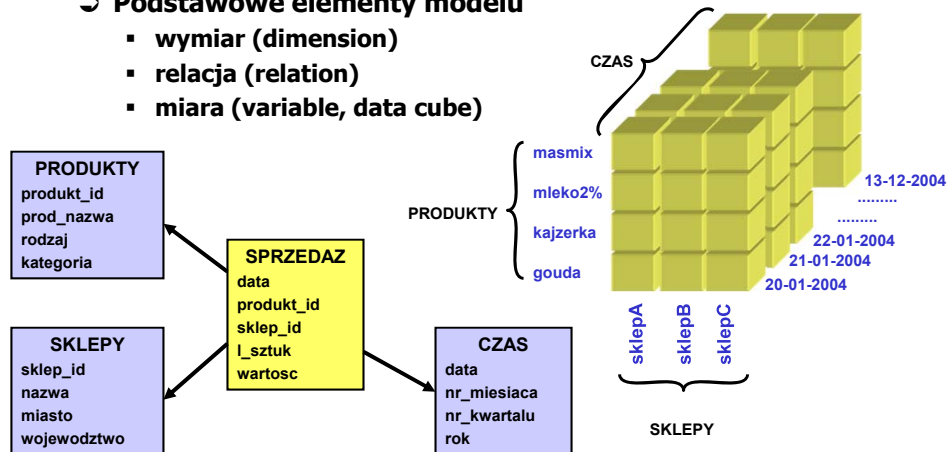
- ⇒ Implementacja
- ⇒ Porównanie MOLAP - ROLAP
- ⇒ MOLAP w implementacji Oracle
- ⇒ Wprowadzenie do MDX





# MOLAP

- Dane przechowywane w wielowymiarowych tablicach
- Podstawowe elementy modelu
  - wymiar (dimension)
  - relacja (relation)
  - miara (variable, data cube)



Robert Wrembel - Politechnika Poznańska

3



## Implementacja MOLAP

- Wielowymiarowe tablice
- Tablice haszowe (SQL Server)
- R-drzewa
- BLOB
- Składowanie na dysku
- Problem pustych przestrzeni

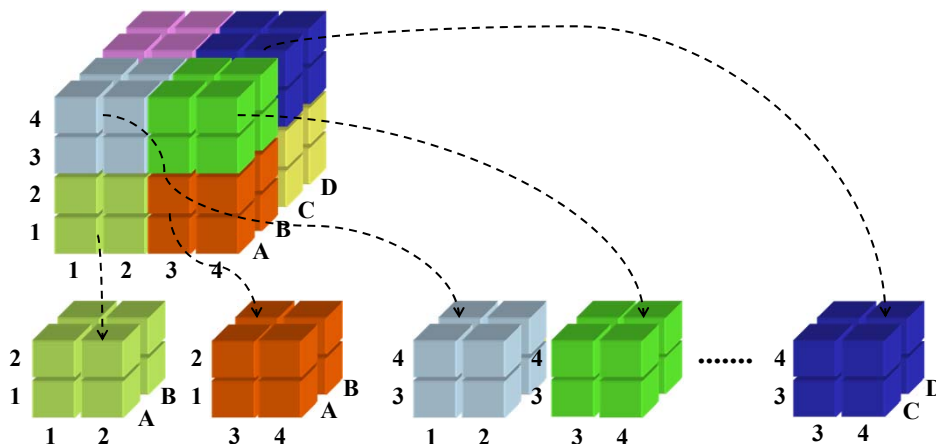
Robert Wrembel - Politechnika Poznańska

4



## Składowanie na dysku

### ➤ Podział "kostki" na części (ang. chunking)



Robert Wrembel - Politechnika Poznańska

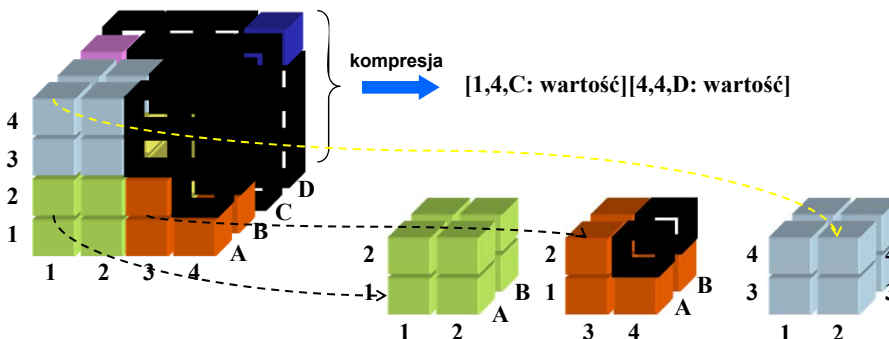
5



## Puste przestrzenie

### ➤ Kompresja - przechowywane są tylko komórki wypełnione

- jedno z podejść: kompresja tylko gdy obszar wypełniony w mniej niż 40%



Robert Wrembel - Politechnika Poznańska

6



## MOLAP ⇔ ROLAP

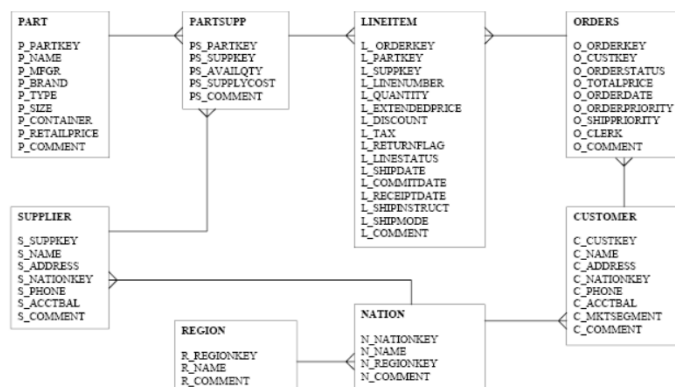
- |   |   |
|---|---|
| <ol style="list-style-type: none"><li>1. Agregaty na wszystkich poziomach hierarchii</li><li>2. Efektywne dla kilku wymiarów</li><li>3. Max 50GB danych</li><li>4. Znacznie dłuższe czasy zasilania</li><li>5. Brak odświeżania przyrostowego</li><li>6. Język analizy danych - MDX, OLAP DML</li><li>7. Zastosowanie: data marts, dane zagregowane</li></ol> | <ol style="list-style-type: none"><li>1. Konieczność definiowania perspektyw zmaterializowanych</li><li>2. Efektywność zależy od dodatkowych struktur danych (persp. zmat, indeksy)</li><li>3. Brak ograniczenia</li><li>4. Zasilanie ścieżką bezpośrednią</li><li>5. Odświeżanie przyrostowe</li><li>6. Język analizy danych - SQL</li><li>7. Zastosowanie: centralna HD, dane zagregowane i elementarne</li></ol> |
|---|---|



## Eksperyment (1)

### ⇒ Standardowy test TPC-H

- schemat
- rozmiary danych
- zbiór zapytań





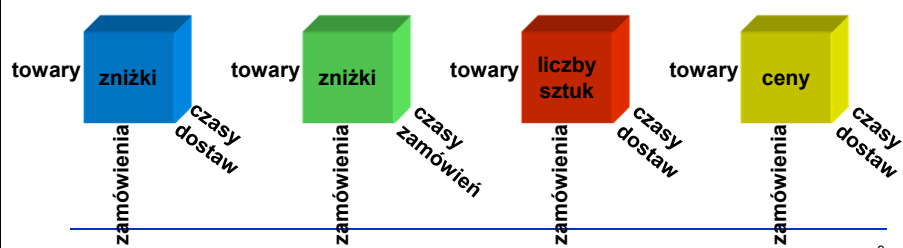
## Schemat testowy

### ➤ ROLAP

- zbiór tabel zgodnych z testem TPC-H
- zbiór indeksów typu B-drzewo na kluczach podstawowych i obcych tabel

### ➤ MOLAP

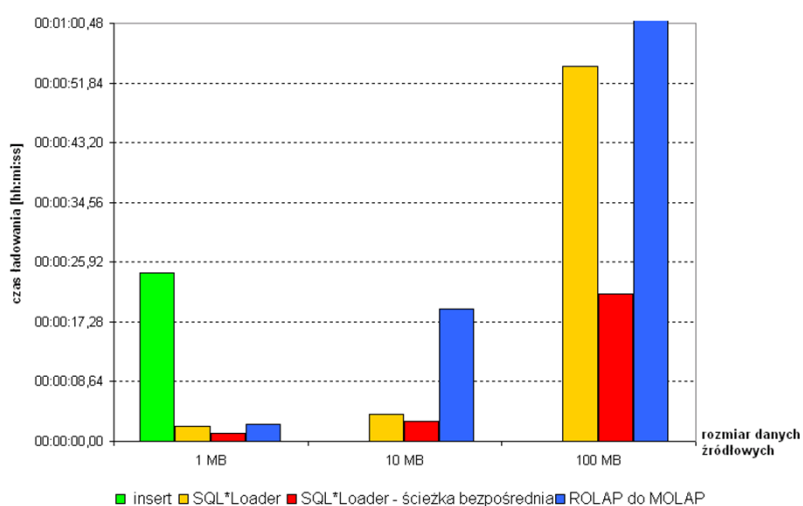
- cztery kostki
- wymiarowane skompresowanym kompozytem wymiarów



9



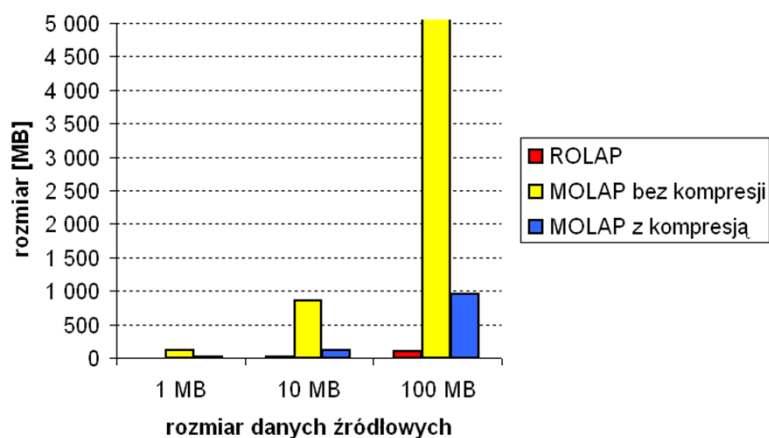
## Eksperyment - czasy ładowania



10



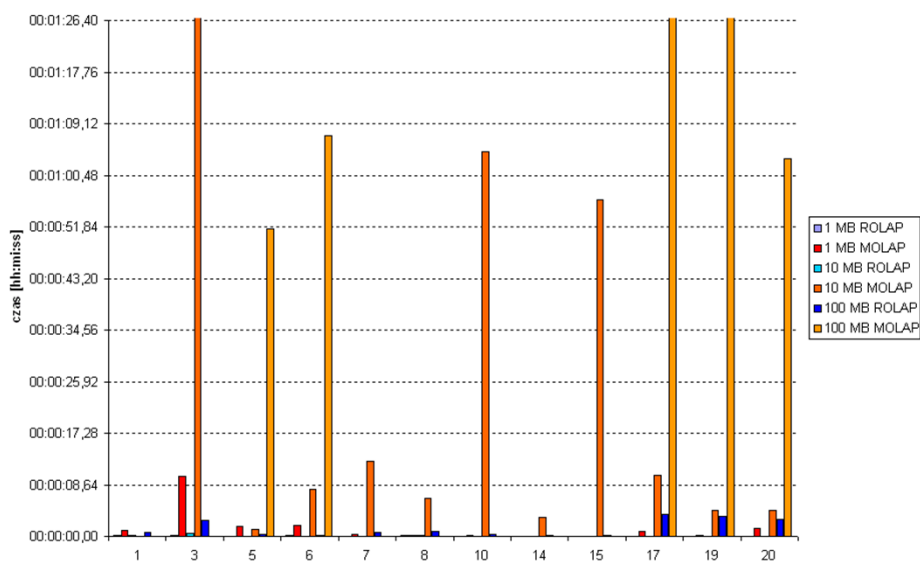
## Eksperyment - rozmiary HD



11



## Eksperyment - zapytania analityczne





## Podsumowanie (1)

---

- ⇒ Implementacja MOLAP
  - nie zawsze efektywna
- ⇒ Przewaga ROLAP nad MOLAP może wynikać
  - z istnienia dodatkowych struktur danych
    - indeksy (B-drzewa, bitmapowe, połączeniowe)
    - perspektyw zmaterializowanych i przepisywania zapytań
- ⇒ **Wyników nie należy uogólniać**
- ⇒ W innych systemach i innych wersjach Oracle efektywność może być inna
  - implementacja
  - techniki optymalizacji zapytań

---

13



## Podsumowanie (2)

---

- ⇒ Ostrożność przy wyborze technologii implementacyjnej
- ⇒ Przetestować w środowisku docelowym

---

14



## Systemy komercyjne MOLAP

---

- ⇒ **Oracle**
  - Express Server
  - DBMS Oracle9i, 10g, 11g
  - Hyperion Essbase
- ⇒ **SAS**
  - BI Server
- ⇒ **Microsoft**
  - SQL Server 2000, 2005
  - Analysis Services

---

15



## MOLAP w implementacji Oracle

---

- ⇒ **Przestrzeń analityczna**
- ⇒ **Definiowanie wymiarów**
- ⇒ **Definiowanie miar (kostek)**
- ⇒ **Wczytywanie danych**
- ⇒ **Analiza danych**
- ⇒ **Mapy agregacji**
- ⇒ **Alokacje**
- ⇒ **Przewidywanie trendów**

---

16





## Komponenty systemu MOLAP (1)

- ➔ **Calculation Engine** - wspiera operacje analizy danych; zapyty w jądrze bazy danych
- ➔ **Analytic Workspace** - przechowuje dane wielowymiarowe oraz procedury napisane w OLAP DML
- ➔ **SQL Table Functions** - predefiniowane funkcje tablicowe; interface pomiędzy środowiskiem MOLAP, a ROLAP
- ➔ **OLAP API** - interfejs programistyczny dla Javy; dostęp za pomocą JDBC
- ➔ **OLAP Catalog Metadata** - metadane opisujące dane przechowywane w tabelach lub w przestrzeni analitycznej



## Komponenty systemu MOLAP (2)

- ➔ **OLAP DML** - język zarządzania i przetwarzania danych wielowymiarowych
  - dostępne funkcje
    - Selekcja
    - Konwersje typów
    - Operacje tekstowe
    - Daty i czasu
    - Numeryczne
    - Finansowe
    - Statystyczne
    - Szeregi czasowe
    - Agregujące
    - Przewidywania i regresje
    - Agregacja
    - Alokacja
    - Odczyt / zapis plików



## Wybrane funkcje numeryczne

- **ABS** - wartość bezwzględna
- **ARCCOS, ARCSIN, ARCTAN** - oblicza wartości kątów (w radianach)
- **BITAND** - bitowe iloczyn dla dwóch wartości integer
- **CEIL** - najmniejsza całkowita wartość większa/ równa podanej liczbie
- **COS, COS, SIN, SIN, TAN, TAN** - funkcje trygonometryczne
- **DECODE**
- **EXP** - oblicza  $e^{\text{potęga}}$  ( $e = 2.71828183\dots$ )
- **FLOOR** - największa całkowita wartość mniejsza/równa podanej liczbie
- **GREATEST** - największa wartość z listy wyrażeń, wyrażenia są konwertowane do typu wyrażenia pierwszego na liście
- **LEAST** - najmniejsza wartość z listy wyrażeń, wyrażenia są konwertowane do typu wyrażenia pierwszego na liście
- **INSTRB** - wartość całkowita z ułamka
- **LOG**
- **LOG10**
- **MAX**
- **MIN**
- **NULLIF** - porównanie dwóch wyrażeń (zwraca NA jeśli są równe w przeciwnym wypadku pierwsze wyrażenie)
- **REM** - reszta z dzielenia
- **ROUND**
- **SIGN**
- **SQRT**
- **TRUNC**



## Wybrane funkcje statystyczne

- **CATEGORIZE** - grupuje wartości wyrażenia liczbowego
- **CORRELATION**
- **NORMAL** - losowa wartość z rozkładu normalnego dla zadanej średniej i odchylenia standardowego
- **RANDOM** - losowa wartość z zadanego przedziału
- **STDDEV** - odchylenie standardowe dla zadanych wartości



## Wybrane funkcje tekstowe (1)

- **ASCII** - reprezentacja dziesiętna pierwszego znaku danego wyrażenia
- **BLANKSTRIP** - usuwa początkowe i końcowe znaki spacji
- **CHANGECHARS** - zamienia wystąpienie podanego tekstu w innym tekście
- **EXTCHARS** - ekstrakcja części tekstu
- **FINDCHARS** - podaje pozycję znaku w tekście
- **GREATEST**
- **INITCAP**
- **INSCARS** - wkłada znak(i) do wyrażenia znakowego
- **INSTR** - wyszukuje pozycję tekstu w innym tekście
- **JOINCHARS** - łączy dwa lub więcej wyrażenia tekstowe
- **LEAST**
- **LIKECASE** - ustawia 'case sensitive' dla operatora LIKE
- **LOWCASE**
- **LTRIM/ RTRIM** - usuwanie znaków z lewej/prawej strony tekstu
- **MAXCHARS** - liczba znaków w najdłuższej linii wieloliniowego wyrażenia tekstowego
- **NULLIF**



## Wybrane funkcje tekstowe (2)

- **NUMCHARS** - liczba znaków w wyrażeniu
- **OBSCURE** - szyfruje wyrażenie tekstowe (dostępne dwie metody)
- **REMCHARS** - usuwa znak(i) z tekstu
- **REPLCHARS** - zamienia znak(i) w wyrażeniu
- **SUBSTR**
- **UPCASE**



## Wybrane funkcje szeregów czasowych

- **CUMSUM** - suma kumulacyjna dla wymiaru
- **LAG** - wartość zmiennej lub wyrażenia dla określonego okna wymiaru poprzedzającego aktualną wartość tego wymiaru
- **LAGABSPCT** - procentowa różnica między bieżącą wartością zmiennej lub wyrażenia dla określonego okna wymiaru a wartością poprzedzającą
- **LAGDIF** - różnica między bieżącą wartością zmiennej lub wyrażenia dla określonego okna wymiaru a wartością poprzedzającą
- **LEAD** - wartość zmiennej lub wyrażenia dla określonego okna wymiaru następującego po aktualnej wartości tego wymiaru
- **MOVINGAVERAGE** - średnia wartości zmiennej lub wyrażenia dla zadanego okna czasowego
- **MOVINGMAX** - maksymalna wartość zmiennej lub wyrażenia dla zadanego okna czasowego
- **MOVINGMIN** - minimalna wartość zmiennej lub wyrażenia dla zadanego okna czasowego
- **MOVINGTOTAL** - suma wartości zmiennej lub wyrażenia dla zadanego okna czasowego



## Wybrane funkcje finansowe

- **Spadek wartości (deprecjacja) kosztów dla zbioru wkładów kapitałowych (aktywów)**
  - **DEPRDECL**
  - **DEPRDECLSW**
  - **DEPRSL**
  - **DEPRSOYD**
- **Stopa procentowa, wielkość spłat, odsetki dla określonej pożyczki (stałe raty)**
  - **FINTSCHED**
  - **FPMTSCHED**
- **Stopa zysku dla określonych danych związanych z przepływem gotówki**
  - **IRR**
- **Stopa procentowa, wielkość spłat, odsetki dla określonej pożyczki (zmienne raty)**
  - **VINTSCHED**
  - **VPMTSCHED**
  - **temp wzrostu**
  - **GROWRATE:  $\{(ostatnia/pierwsza)^{(1/(n-1))}\} - 1$**



## Wybrane funkcje konwersji typów

- **TO\_CHAR**
  - `nls_date_language='italian'`
  - `show to_char(sysdate, 'day, month, yyyy: hh24:mi:ss')`
- **TO\_DATE**
  - `show to_date('January 15, 2002, 11:00 A.M.', 'Month dd, YYYY, HH:MI A.M.')`
- **TO\_NUMBER**
  - `show to_number('zł800 300,91', 'L999G999D99')`



## Narzędzia programistyczne

- ⇒ Enterprise Manager
- ⇒ Warehouse Builder
- ⇒ Analytic Workspace Manager
- ⇒ OLAP Worksheet (część AWM)
- ⇒ SQL



## Przestrzeń analityczna

### ➔ Utworzenie

```
DBMS_AW.EXECUTE('AW CREATE test_ws;')  
DBMS_AW.EXECUTE('UPDATE; COMMIT;')
```

```
aw create test_ws  
update  
commit
```

### ➔ Dołączenie w trybie RW i RO

```
DBMS_AW.EXECUTE('AW ATTACH test_ws RW;')
```

```
aw attach test_ws rw
```

```
DBMS_AW.EXECUTE('AW ATTACH test_ws RO;')
```

```
aw attach test_ws RW after analysis
```

```
aw attach test_ws RW before sales
```

### ➔ Odłączenie

```
DBMS_AW.EXECUTE('AW DETACH test_ws;')
```

```
aw detach test_ws
```

### ➔ Usunięcie

```
DBMS_AW.EXECUTE('AW DELETE test_ws;')
```

```
aw delete test_ws
```



## Program AUTOGO

### ➔ Wywoływany w czasie dołączania przestrzeni analitycznej

```
define autogo program  
program  
  aw attach expense RW after analysis  
  aw attach revenue RW after analysis  
end
```

← program zdefiniowany  
w przestrzeni ANALYSIS

```
aw attach analysis autogo
```

← spowoduje uruchomienie  
programu AUTOGO



## Prawa dostępu

- ➔ Właściciel obszaru roboczego może nadać innym użytkownikom uprawnienia SELECT, UPDATE, INSERT, DELETE, ALTER, ALL do swojego obszaru
- ➔ Nadawanie uprawnień - polecenie GRANT
- ➔ Obszar jest implementowany w tabeli o nazwie AW\$nazwa\_obszaru
- ➔ Uprawnienia nadaje się do tabeli AW\$nazwa\_obszaru
- ➔ Dostęp do cudzego obszaru - należy jego nazwę poprzedzić nazwą właściciela
- ➔ Odbieranie uprawnień - polecenie REVOKE

```
SQL> grant select on aw$test_ws  
2> to user_star;
```

```
aw attach user_molap1.test_ws rw
```

```
aw detach user_molap1.test_ws
```

```
SQL> revoke all on aw$test_ws  
2> from user_star;
```



## Definiowanie zmiennej prostej

```
DEFINE zmienna VARIABLE typ [TEMP | SESSION]
```

### typ

- INTEGER ( $-2^{31} \rightarrow 2^{31}-1$ )
- LONGINT ( $-2^{63} \rightarrow 2^{63}-1$ )
- SHORTINT ( $-2^{15} \rightarrow 2^{15}-1$ )
- NUMBER(m, n)
- DECIMAL (liczby do 15 cyfr znaczących)
- SHORTDECIMAL (liczby do 7 cyfr znaczących)
- TEXT (n) (maksymalnie 4000B)
- ID (maksymalnie znaków - 8B)
- BOOLEAN (YES/NO, TRUE/FALSE, ON/OFF)
- DATE (dd-mm-yyyy)
- DATETIME (dd-mm-yyyy:hh24:mi:ss)

```
DEFINE x VARIABLE DECIMAL  
x=500  
show x  
DEFINE z VARIABLE DECIMAL  
z=x*10  
show z  
delete x  
delete z
```

- TEMP → zmienna jako obiekt jest trwała, natomiast jej wartość jest przechowywana tylko w ramach bieżącej sesji
- SESSION → zmienna jako obiekt jest automatycznie usuwana po zakończeniu bieżącej sesji



## Definiowanie prostego wymiaru

```
DEFINE wymiar DIMENSION typ [SESSION]
```

typ

- TEXT WIDTH(n)
- ID
- INTEGER
- NUMBER(m, n)
- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

```
SQL> exec DBMS_AW.EXECUTE('DEFINE lokalizacja DIMENSION TEXT W 15; -  
                           DEFINE produkty DIMENSION TEXT W 20; -  
                           DEFINE czas DIMENSION MONTH;')
```



## Definiowanie hierarchii wymiaru

### ➤ Wymiar skonkatenowany (concat dimension)

- umożliwia zgrupowanie zbioru prostych wymiarów w jeden
- wykorzystywany do definiowania hierarchii wymiarów

```
exec DBMS_AW.EXECUTE('DEFINE d_prod_nazwa DIMENSION TEXT W 17; -  
                     DEFINE d_prod_rodzaj DIMENSION TEXT W 16; -  
                     DEFINE d_prod_kategoria DIMENSION TEXT W 10;')  
  
exec DBMS_AW.EXECUTE('DEFINE d_produkty DIMENSION CONCAT (d_prod_kategoria -  
                                                           d_prod_rodzaj -  
                                                           d_prod_nazwa);')  
  
exec DBMS_AW.EXECUTE('DEFINE rel_d_produkty RELATION d_produkty <d_produkty>')
```





## Definiowanie kostki

```
exec DBMS_AW.EXECUTE -  
  ('DEFINE l_sztuk -  
    VARIABLE DECIMAL <lokalizacja produkty czas>;')
```

```
DEFINE wartosc VARIABLE DECIMAL <lokalizacja czas>;')
```



## Definiowanie kompozytu

- **Kostka (zmienna) rzadka w wymiarach <dimA, dimB, dimC>**
  - zalecane utworzenie **kompozytu** (composite) ⇒ bardziej efektywne przechowywanie danych
- **Kompozyt nienazwany** (unnamed composite)
  - utworzony przez dodanie SPARSE
  - automatycznie jest tworzona struktura do przechowywania zmiennej x

```
DEFINE x DATATYPE decimal SPARSE <dimA, dimB, dimC>
```

- **Kompozyt nazwany** (named composite)
  - utworzony jawnie - reprezentuje wymiary

```
DEFINE market.product COMPOSITE <market product>  
DEFINE expenses DECIMAL <month market.product <market product>>
```



## Dodawanie opisu obiektów

- Polecenie **LD**
- Dodaje opis do bieżącego obiektu
  - obiektem bieżącym jest:
    - przed chwilą utworzony
    - wskazany poleceniem **CONSIDER**

```
CONSIDER sklepy
LD wymiar sklepy
DESCRIBE sklepy
```

```
DEFINE SKLEPY DIMENSION TEXT WIDTH 15
LD wymiar sklepy
```

- Polecenie **DESCRIBE** obiekt
  - wyświetla definicję obiektu



## Zawartość AW

- Program **LISTNAMES**
  - wyświetla nazwy wszystkich obiektów w bieżącej przestrzeni analitycznej

parametr

- DIMENSION
- VARIABLE
- PROGRAM
- FORMULA
- RELATION
- COMPOSITE
- VALUE
- AGGMAP

```
LISTNAMES AW [nazwa_przestrzeni | '*']
[TYP_OBIEKTU | ALL]
[LIKE '%wzo_rzec']
```

- domyślnie LISTNAMES adresuje bieżącą przestrzeń
- '\*' oznacza wszystkie dołączone przestrzenie

```
LISTNAMES dimension variable relation
```

- Program **AWDESCRIBE**
  - raport zawartości przestrzeni

```
AWDESCRIBE
```



## Polecenie AW

- **AW CREATE** przestrzeń
- **AW ATTACH** przestrzeń
- **AW DETACH** przestrzeń
- **AW DELETE** przestrzeń
- **AW LIST** ⇒ wyświetla nazwy dołączonych przestrzeni

```
aw list
TEST_WS    R/O UNCHANGED USER_MOLAP1.TEST_WS
(1 other user writing)
EXPRESS    R/O UNCHANGED SYS.EXPRESS
(1 other user reading)
```



## Funkcja AW

- **Podaje informacje nt. dołączonej przestrzeni**

parametr

- **ATTACHED** - zwraca Yes jeśli przestrzeń o podanej nazwie jest dołączona
- **DATE** - data ostatniej modyfikacji przestrzeni
- **LIST** - lista dołączonych przestrzeni
- **LISTNAMES** - lista wszystkich obiektów w przestrzeni
- **NAME** - nazwa przestrzeni
- **RO** - zwraca Yes jeśli przestrzeń dołączono w trybie RO
- **RW** - zwraca Yes jeśli przestrzeń dołączono w trybie RW
- lista obiektów wskazanego typu:
  - COMPOSITE
  - DIMENSION
  - FORMULA
  - PROGRAM
  - RELATION
  - VARIABLE

**REPORT AW (parametr)**

**report aw(RW)**

```
AW (RW)
-----
no
```

**report aw(attached 'test\_ws')**

```
AW (ATTACHED 'test_ws')
-----
yes
```



## Usuwanie obiektów

- ⇒ Polecenie **DELETE** usuwa dowolny obiekt z bieżącej przestrzeni analitycznej

```
DELETE nazwa_obiektu
```



## Dodawanie danych do wymiarów i kostek

- ⇒ Wypełnianie wymiarów
- poleceniem **MAINTAIN**
  - programem wczytującym dane z pliku lub tabeli
- ⇒ Wypełnianie kostek
- poleceniem **LIMIT** i operatorem przypisania
  - Qualified Data Reference (QDR) i operatorem przypisania
  - programem wczytującym dane z pliku lub tabeli
- ⇒ ... \molap\demo\demo\_wypelnianie.sql
- ⇒ ... \molap\demo\demo\_wypeln\_ztabeli.sql



## Wyświetlanie danych

### ➤ Polecenie **REPORT**

- wyświetla wartości wymiarów, zmiennych prostych i kostek

### ➤ Polecenie **SHOW**

- wyświetla pojedynczą wartość, np. zmiennej typu prostego, komórki kostki, wymiaru

REPORT sprzedaz1	cała kostka	
SHOW sprzedaz1 ←		pierwsza komórka
SHOW VALUES (czas)	cały wymiar	
SHOW czas ←		pierwsza komórka
SHOW sysdate		
show arcsin(0.5)		



## Polecenie **REPORT**

```
report sprzedaz1
```

-----SPRZEDAZ1-----			
-----CZAS-----			
LOKALIZACJA	JAN04	FEB04	MAR04
Poznań	NA	NA	NA
Gniezno	NA	NA	NA
Warszawa	NA	NA	NA

```
report w 15 down czas w 20 across lokalizacja: sprzedaz1
```

-----SPRZEDAZ1-----			
-----LOKALIZACJA-----			
CZAS	Poznań	Gniezno	Warszawa
JAN04	NA	NA	NA
FEB04	NA	NA	NA
MAR04	NA	NA	NA

```
REPORT W 0 DOWN czas ACROSS lokalizacja:sprzedaz1
```

↑ kolumna nie jest wyświetlana



## Formatowanie kolumn

### ➤ Domyślna szerokość kolumny: parametr **COLWIDTH**

- domyślnie 14 znaków
- max 4000 znaków

```
COLWIDTH=20  
exec DBMS_AW.EXECUTE('COLWIDTH=20')
```

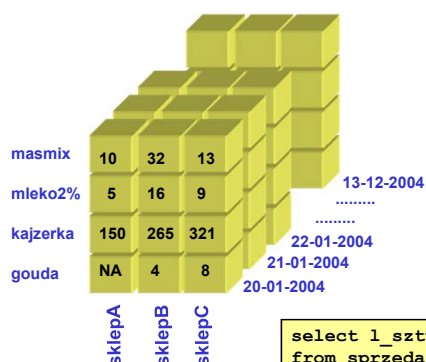
### ➤ Domyślna szerokość nagłówka: parametr **LCOLWIDTH**

- domyślnie 10 znaków
- max 4000 znaków

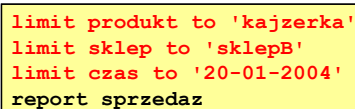
```
LCOLWIDTH=20  
exec DBMS_AW.EXECUTE('LCOLWIDTH=20')
```



## Polecenie LIMIT (1)



```
select l_sztuk  
from sprzedaz sp, produkty pr, sklepy sk, czas cz  
where sp.produkt_id=pr.produkt_id  
and sp.sklep_id=sk.sklep_id  
and sp.data=cz.data  
and pr.prod_nazwa='kajzerka'  
and sk.nazwa='sklepB'  
and cz.data='20-01-2004'
```



Robert Wrembel - Politechnika Poznańska

45



## Robert Wrembel - Politechnika Poznańska

46



## Operatory

- NOT
- AND
- OR
- EQ (equal)
- NE (not equal)
- GT (greater than)
- LT (less than)
- GE (greater than or equal to)
- LE (less than or equal to)
- IN
- LIKE
- = (przypisania)

### operacje na kostkach

```
kostka2 = kostka1  
kostka3 = kostka2*2  
clear from kostka3  
kostka3 = NA
```

➔ dalsze przykłady ... \molap\demo\demo\_analazy.sql



## Przykłady wykorzystania funkcji (1)

REPORT HEADING 'łącznie' W 10 TOTAL(sprzedaz1, czas)

CZAS	łącznie
JAN04	14,00
FEB04	24,00
MAR04	39,00
APR04	60,00
MAY04	80,00

szerokość okna: rekord bieżący i dwa przed

REPORT HEADING '3 months total' MOVINGTOTAL(sprzedaz1, -2, 0, 1, czas)

krok przesuwu okna

LOKALIZACJA	JAN04	FEB04	MAR04	APR04	MAY04
p1	2,00	4,00	6,00	8,00	10,00
p2	2,00	2,00	6,00	8,00	11,00

LOKALIZACJA	JAN04	FEB04	MAR04	APR04	MAY04
p1	2,00	6,00	12,00	18,00	24,00
p2	2,00	4,00	10,00	16,00	25,00





## Przykłady wykorzystania funkcji (2)

### ➤ MIN/MAX (wyrażenie1, wyrażenie2)

- wybiera mniejszą/większą wartość wyrażenia

```
nowa_cena = MAX(cena_2003, cena_2004)*1.2
```

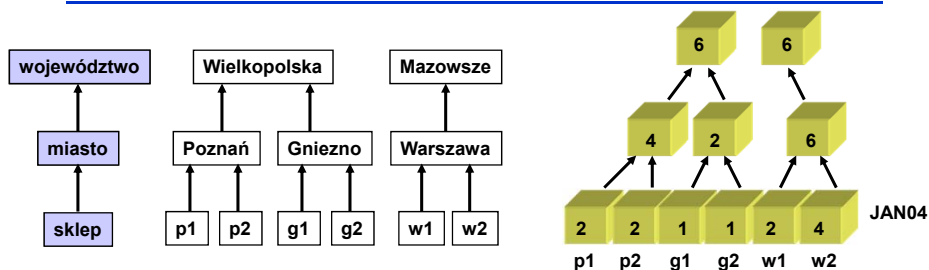
### ➤ LAG/LEAD (zmienna, przesunięcie, wymiar)

```
REPORT DOWN czas -  
HEADING 'prev' LAG(sprzedaz1, 1, czas) -  
HEADING 'now' sprzedaz1 -  
HEADING 'next' LEAD(sprzedaz1, 1, czas)
```

CZAS	-----p1-----			-----p2-----		
	prev	now	next	prev	now	next
JAN04	NA	2,00	4,00	NA	2,00	2,00
FEB04	2,00	4,00	6,00	2,00	2,00	6,00
MAR04	4,00	6,00	8,00	2,00	6,00	8,00
APR04	6,00	8,00	10,00	6,00	8,00	11,00



## Agregacja (1)



➤ demo: ..\molap\demo\demo\_aggreg.sql

```
define sprzedaz_agg variable decimal <czas lokalizacja>  
sprzedaz_agg=sprzedaz1  
  
define full.agg aggmap  
aggmap  
RELATION lokalizacja.r OPERATOR SUM  
RELATION czas.r OPERATOR SUM  
end  
  
aggregate sprzedaz_agg using full.agg
```



## Agregacja (2)

### ⇒ Operatory agregacji m.in.:

- SUM (domyślnie)
- WSUM - suma ważona
- AVERAGE
- WAVERAGE - średnia ważona
- MAX
- MIN
- FIRST - pierwsza wartość != NA
- LAST - ostatnia wartość != NA

### ⇒ PRECOMPUTE

- ogranicza agregację do wskazanych wartości poziomu

```
define full.agg aggmap
aggmap
  RELATION lokalizacja.r PRECOMPUTE ('Poznań' 'Warszawa')
  RELATION czas.r PRECOMPUTE ('Q1.04')
end
```



## Agregacja (3)

### ⇒ COUNTVAR

- zlicza wartości w poziomie podrzędnym, różne od NA, które posłużyły do wyliczenia agregatu na poziomie nadrzędnym

```
define sprzedaz_aggavg variable decimal <czas lokalizacja>
define v_count variable integer <czas lokalizacja>
sprzedaz_aggavg=NA

define avg.agg aggmap
consider avg.agg
aggmap JOINLINE (-
'RELATION lokalizacja.r OPERATOR AVERAGE' -
'RELATION czas.r OPERATOR SUM' -
'END')

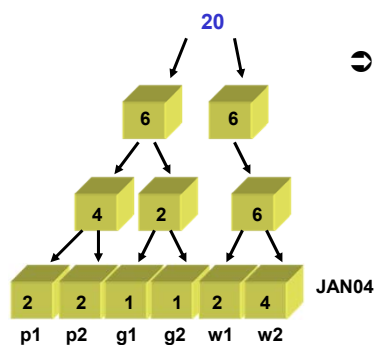
aggr sprzedaz_aggavg using avg.agg FROM sprzedaz1 COUNTVAR v_count
```

⇒ demo: ..\molap\demo\demo\_aggreg.sql



## Alokacja (1)

- ⇒ Dystrybuuje wartości w dół hierarchii wymiaru
  - podział budżetu firmy pomiędzy oddziały



### ⇒ Rodzaje

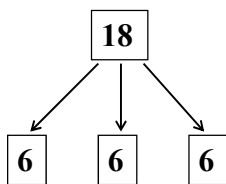
- równomierna
- proporcjonalna

⇒ demo: `..\molap\demo\demo_aggreg.sql`

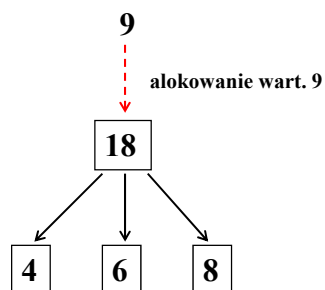
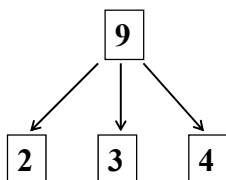


## Alokacja (2)

### ⇒ Równomierna



### ⇒ Proporcjonalna





## Alokacja (3)

### ➤ Alokacja proporcjonalna

LOKALIZACJA	JAN04
p1	2,00
p2	2,00
g1	1,00
g2	1,00
w1	2,00
w2	4,00
l1	1,00
l2	1,00
Poznań	4,00
Gniezno	2,00
Warszawa	6,00
Łódź	2,00
Wielkopolska	6,00
Mazowsze	8,00
Polska	14,00

LOKALIZACJA	JAN04
p1	2,86
p2	2,86
g1	1,43
g2	1,43
w1	2,86
w2	5,71
l1	1,43
l2	1,43
Poznań	5,71
Gniezno	2,86
Warszawa	8,57
Łódź	2,86
Wielkopolska	8,57
Mazowsze	11,43
Polska	20,00

Polska<sup>old</sup> = 14

Wielkopolska<sup>old</sup> = 6

Mazowsze<sup>old</sup> = 8

Wielkopolska<sup>new</sup> = 20 \* (Wielkopolska<sup>old</sup> / Polska<sup>old</sup>)

Mazowsze<sup>new</sup> = 20 \* (Mazowsze<sup>old</sup> / Polska<sup>old</sup>)



## Alokacja (4)

### ➤ Alokacja proporcjonalna

```
DEFINE ALOKPROP.AGG AGGMAP  
ALLOCMAP  
RELATION lokalizacja.r OPERATOR PROPORTIONAL ARGS MIN 3  
END
```

➤ **MIN *minval*** - jeżeli zaalokowana wartość jest mniejsza niż *minval*, wówczas do komórki jest przypisywana wartość *minval*

➤ **MAX *maxval*** - jeżeli zaalokowana wartość jest większa niż *maxval*, wówczas do komórki jest przypisywana wartość *maxval*

➤ Stosowane wyłącznie z operatorem PROPORTIONAL

➤ **demo: ..\molap\demo\demo\_aggreg.sql**



## Prosta regresja (1)

### ➔ Polecenie **REGRESS**

```
REGRESS [NOINTERCEPT] zm_zal zm_niezal1 ...  
zm_niezalN [WEIGHTBY zm_wag]
```

- **NOINTERCEPT** - nie wylicza współczynnika b w równaniu prostej
- **zm\_zal** - zmienna zależna
- **zm\_niezal1** - pierwsza zmienna niezależna
- **zm\_niezalN** - N-ta zmienna niezależna
- **WEIGHTBY** - dla regresji ważonej, np. dane z określonych sklepów mogą być mniej ważne
- **zm\_wag** - zmienna wymiarowana, zawierająca wagi, np. dla sklepów

```
REGRESS sprzedaz1 czas lokalizacja
```



## Prosta regresja (2)

### ➔ Wynik regresji: polecenie

```
REGRESS.REPORT
```

### ➔ Informacje o obliczeniach: funkcja

parametr

```
INFO (REGRESS parametr)
```

- **DEPENDENT** - nazwa zmiennej zależnej
- **WEIGHTED** - TRUE jeśli regresja ważona
- **STDERROR** - wartość odchylenia standardowego
- **NUMOBS** - liczba wykorzystanych komórek
- **FORMULA** - równanie wykorzystanej prostej
- ...

```
REPORT INFO (REGRESS FORMULA)
```

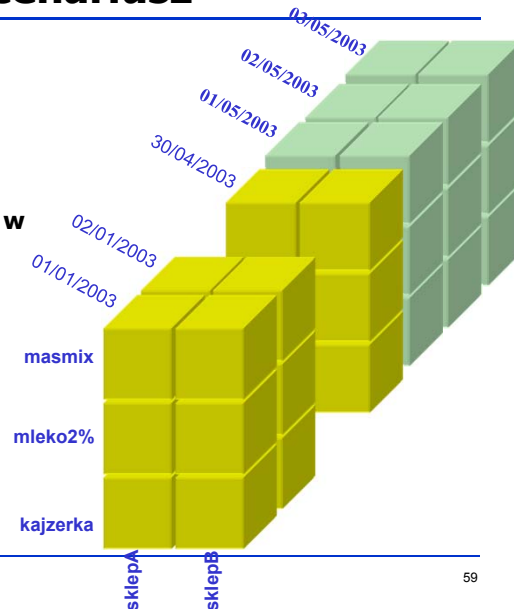
```
REPORT INFO (REGRESS NUMOBS)
```

➔ demo: **..\molap\demo\demo\_regress.sql**



## Zaawansowana predykcja - scenariusz

- Dane historyczne nt. liczby sprzedanych towarów w ostatnich n miesiącach
- Oszacować sprzedaż w kolejnych dniach maja 2003
- Oszacować sumaryczną sprzedaż produktu kajzerka w sklepie A w maju 2003



## Zaawansowana predykcja - implementacja (1)

- Przygotowanie środowiska
  1. Dodać do wymiaru czasu okres predykcji
  2. Zdefiniować zmienną wynikową przechowującą wynik predykcji
  3. Przygotować parametry predykcji
  4. Uruchomić program predykcji
  5. Wyświetlić wyniki



## Narzędzia predykcji

### ⇒ FCOPEN

- funkcja tworząca i otwierająca kontekst predykcji

### ⇒ FCSET

- polecenie ustalające parametry/charakterystykę predykcji

### ⇒ FCEXEC

- polecenie uruchamiające proces predykcji

### ⇒ FCQUERY

- funkcja udostępniająca wykorzystane parametry/charakterystykę predykcji

### ⇒ FCCLOSE

- polecenie zamykające kontekst predykcji



## Funkcja FCOPEN

### ⇒ Tworzy i otwiera kontekst predykcji

### ⇒ Zwraca uchwyt do kontekstu (typu INTEGER)

```
FCOPEN ('nazwa_kontekstu' [uchwyt_do_innego_kontekstu])
```

```
VARIABLE cont_handle INTEGER  
cont_handle = FCOPEN ('sprzedazMAJ04')
```

- uchwyt\_do\_innego\_kontekstu umożliwia wskazanie wcześniej zdefiniowanego kontekstu w celu wykorzystania jego parametrów/charakterystyki predykcji



## Polecenie FCSET

### ➤ Dobiera parametry wskazanego kontekstu predykcji (cont\_handle)

```
FCSET cont_handle parametr1 'wartość1' ... parametrN 'wartośćN'
```

### ➤ Parametry

- **METHOD** – wybór metody przewidywania
  - parametr jest aktywny gdy **APPROACH** ma wartość **'APPMANUAL'**
- **HISTORYPERIODS** – określa liczbę rekordów historycznych wykorzystywanych do predykcji
  - zakres: 1 - 5000
- **TRIALS** – liczba prób predykcji w celu dobrania najlepszej metody
  - zakres: 1 - 3
- **MAXFACTOR** – górna granica wartości przewidywanych danych
  - granicę tę otrzymuje się mnożąc wartość tego parametru przez największą wartość historyczną
  - domyślnie 100
- **MINFACTOR** – dolna granica wartości przewidywanych danych
  - granicę tę otrzymuje się mnożąc wartość tego parametru przez najmniejszą wartość historyczną
  - zakres: 0.0 – 1.0
  - wartość domyślna: 0.0

```
FCSET cont_handle METHOD 'LINREG' HISTPERIODS 30  
APPROACH 'APPMANUAL'
```

Robe



## Polecenie FCEXEC

### ➤ Uruchamia predykcję dla wskazanego kontekstu (cont\_handle)

```
FCEXEC cont_handle -  
TIME nazwa_wymiaru -  
[INTO kostka_wyj]] -  
kostka_wej
```

- **cont\_handle** - uchwyt do kontekstu utworzonego funkcją **FCOPEN**
- **TIME nazwa\_wymiaru** – wskazuje wymiar reprezentujący czas
- **INTO kostka\_wyj** – wskazuje zmienną wyjściową, która będzie przechowywała wyniki predykcji
  - jeśli parametr zostanie pominięty, wówczas dane będą zapisywane w zmiennej wejściowej (dane wejściowe mogą zostać nadpisane)
- **kostka\_wej** - zmienna wejściowa z wymiarem czasu, zawiera dane historyczne wykorzystywane w predykcji

```
FCEXEC cont_handle TIME czas INTO wynik sprzedaz
```





## Polecenie FCCLOSE

⇒ Zamyka wskazany kontekst (cont\_handle)

```
FCCLOSE cont_handle
```

⇒ demo: ..\molap\demo\_forecast



## Metody predykcji

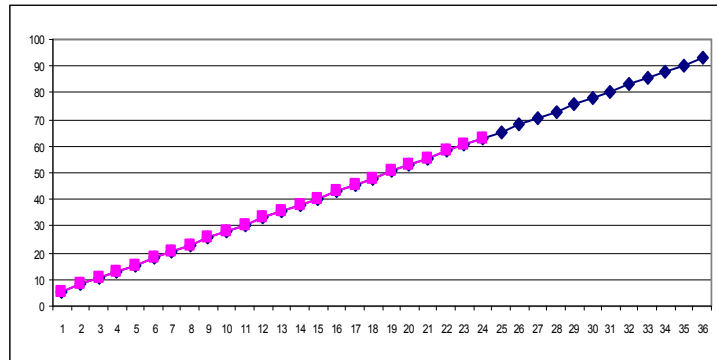
⇒ W Oracle10g są dostępne następujące metody predykcji:

- regresja liniowa (METHOD='LINREG')
- regresje nieliniowe (METHOD='NLREGi')
  - NLREG1:  $y = a * x + b$
  - NLREG2:  $y = c * x^a$
  - NLREG3:  $a * \log(x) + b$
  - NLREG4:  $x / (a + b * x)$
  - NLREG5:  $(c * e^{a * x}) / (1 + c * e^{a * x})$
- single exponential smoothing (METHOD='SESMOOTH')
- double exponential smoothing (METHOD='DESMOOTH')
- metoda Holt/Winters (METHOD='HOLT/WINTERS')
- automatyczna (METHOD='AUTOMATIC') - Oracle dobierze najlepszą metodę

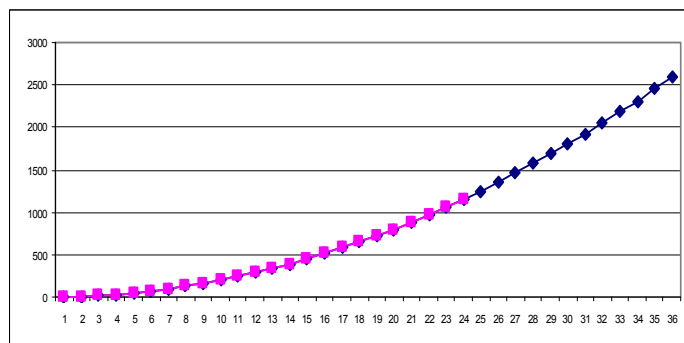
⇒ Metodę wskazuje się parametrem **METHOD** polecenia **FCSET**



## METHOD='LINREG'

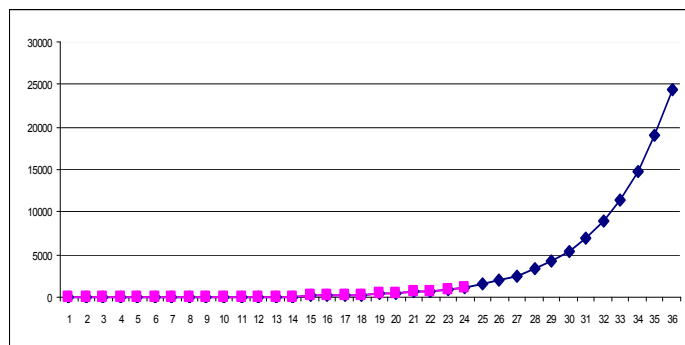


## METHOD='NLREG1'

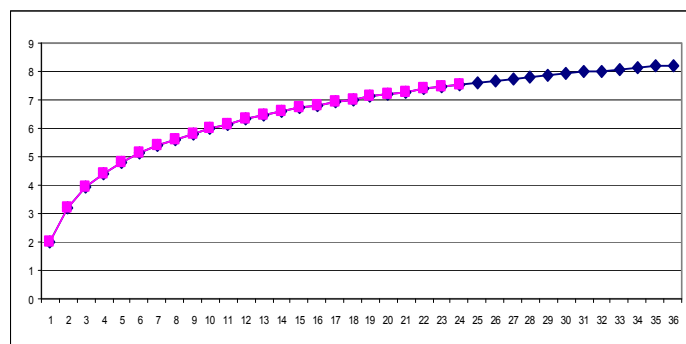




## METHOD='NLREG2'

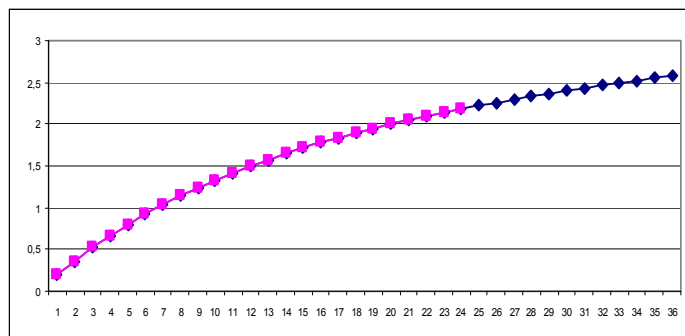


## METHOD='NLREG3'

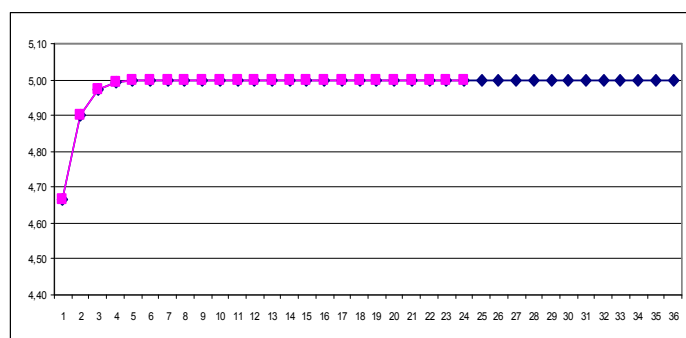




## METHOD='NLREG4'

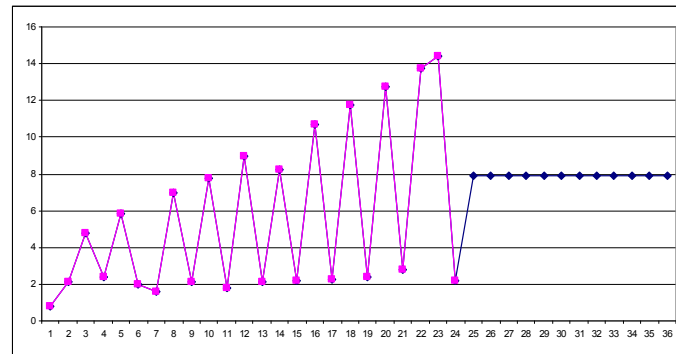


## METHOD='NLREG5'

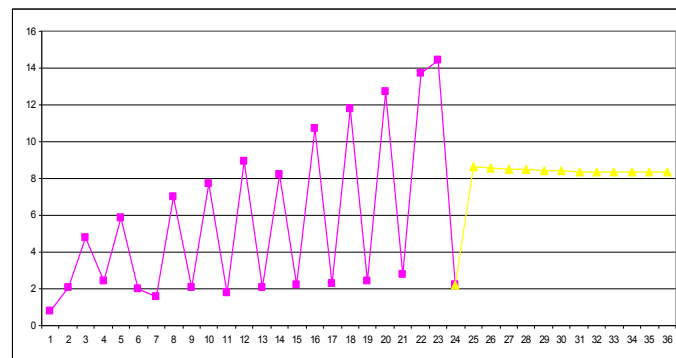




## METHOD='SESMMOOTH'

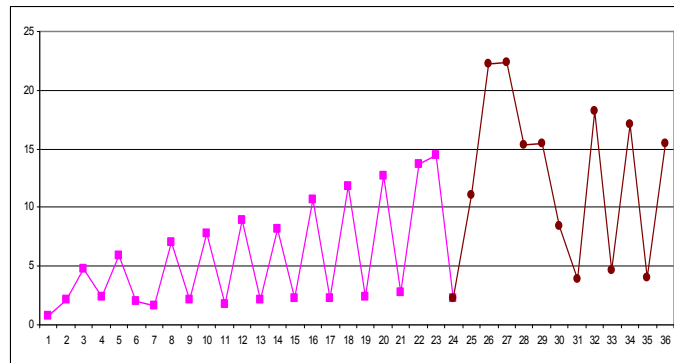


## METHOD='DESMOOTH'

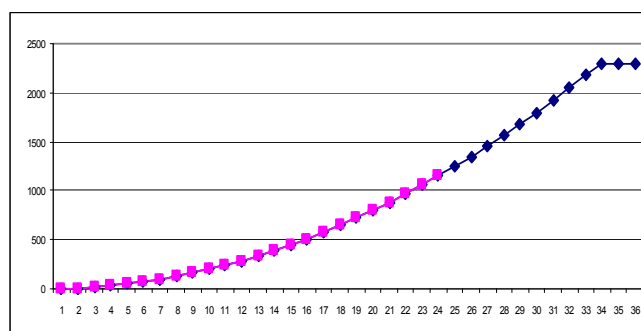




## METHOD='HOLT/WINTERS'

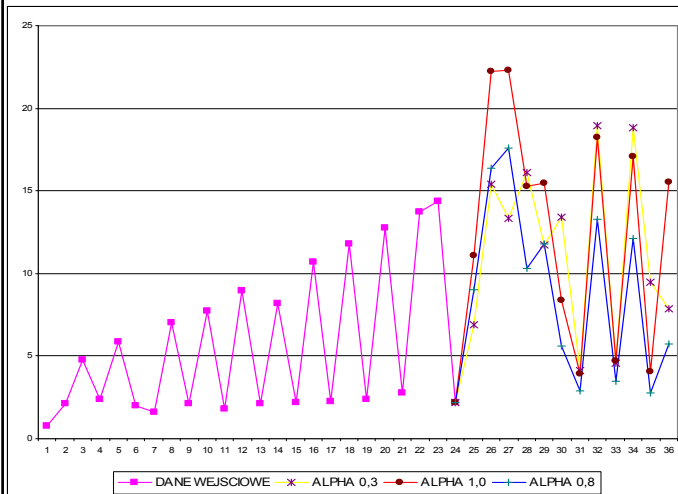


## MAXFACTOR





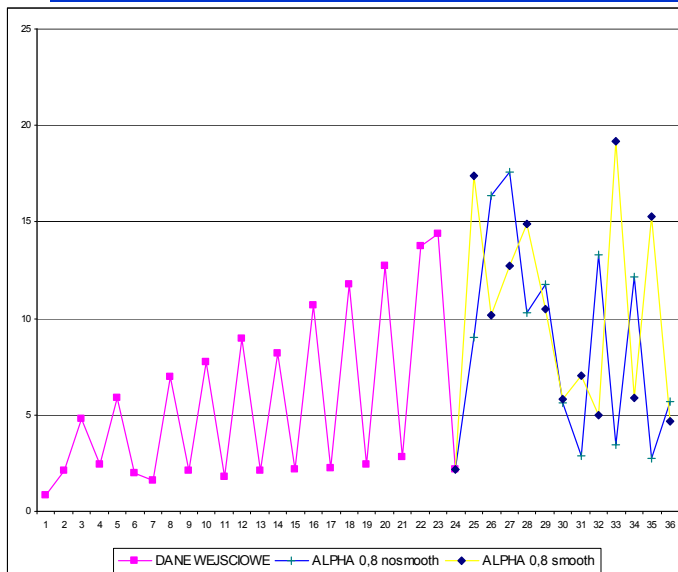
## Współczynnik ALPHA



■ wykorzystywany w metodach wygładzania i Holt/Winters



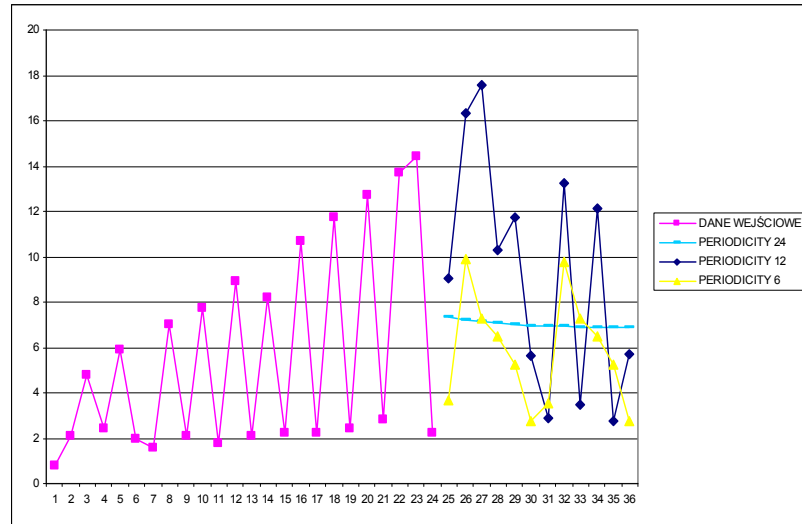
## Współczynnik SMOOTHING



■ SMOOTHING = YES/NO



## Współczynnik PERIODICITY



Robert Wrembel - Politechnika Poznańska

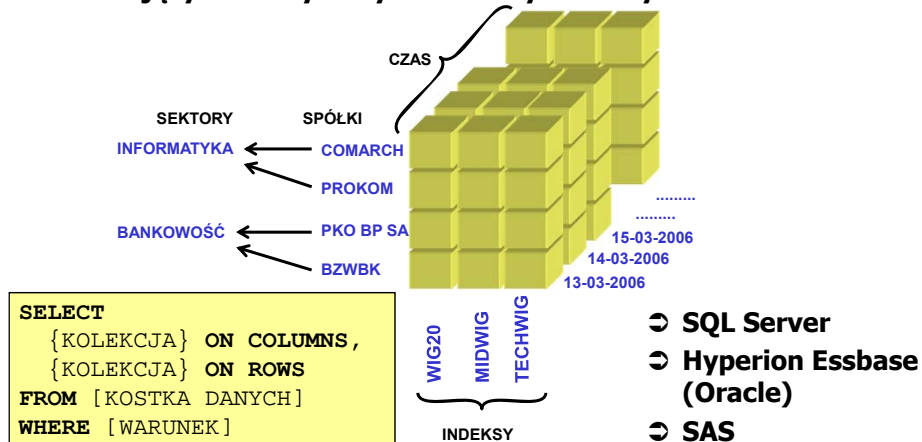
79



## MDX

### ➤ MultiDimensional eXpressions

- język analizy danych wielowymiarowych



Robert Wrembel - Politechnika Poznańska

80





## MDX

### ⇒ Obrót z dnia 09.05.2005 dla spółek sektora bankowego wchodzących w skład indeksu WIG20

```
select {[Obrót]} on rows,  
{(  
  [czas].[May 09],  
  [rok].[2005],  
  [sektor].[banki],  
  [index].[wig20]  
)} on columns  
from SPOLKI.spolkiDB;
```

Axis-1	(May 09, 2005, Banki, WIG20)
(Obrót)	317007



## MDX

### ⇒ Obrót wszystkich spółek wchodzących w skład indeksu WIG20 (dla wszystkich notowań w bazie danych)

```
select  
  {[Obrót]} on rows,  
  {[wig20]} on columns  
from  
  SPOLKI.spolkiDB;
```

Axis-1	(WIG20)
(Obrót)	1608242944.16667



## MDX

### ➤ Obróty spółki Techwig w 2005 i obroty wszystkich spółek wchodzących w skład indeksu WIG20

```
select
  {[Obrót]} on rows,
  {[techwig],[2005]}, ([wig20], [2005]) on columns
from SPOLKI.spolkiDB;
```

Axis-1	(TECHWIG, 2005)	(WIG20, 2005)
(Obrót)	1449650	1025137649.83333



## MDX

### ➤ Obrót spółek z notowań od stycznia do marca, z podziałem na lata

```
select {[Obrót],[2005]}, ([Obrót],[2006]) on rows,
  {[Jan]:[Mar]} on columns
from SPOLKI.spolkiDB;
```

```
select {[Obrót],[2005]}, ([Obrót],[2006]) on rows,
  {MemberRange([Jan],[Mar])} on columns
from SPOLKI.spolkiDB;
```

Axis-1	(Jan)	(Feb)	(Mar)
(Obrót, 2005)	88104579	104873583	99285346
(Obrót, 2006)	267968635	133036826	129383238



## MDX

### ⇒ Obroty w 2005 wszystkich spółek sektora Informatyka

```
select {[Obrót],[2005]} on rows,  
       {[Informatyka].children} on columns  
from SPOLKI.spolkiDB;
```

Axis-1	(PROKOM)	(COMARCH)	(COMP)
(Obrót, 2005)	4892227	1588614	393603



## MDX

### ⇒ Obroty w 2005 wszystkich spółek + agregacja wartości na poziomie sektora

```
select crossjoin  
(  
    {[Obrót]},  
    {members(sektor.levels(1)),  
      members(sektor.levels(0))}  
) on rows,  
   {[2005]} on columns  
from SPOLKI.spolkiDB;
```

Axis-1	(2005)
(Obrót, banki)	235469677
(Obrót, informatyk	6868499
(Obrót, p_metalowy	303889846
(Obrót, telekomuni	593986399
(Obrót, usługi)	18396037
(Obrót, PKOBP)	221255682
(Obrót, INGBSK)	857474
(Obrót, BZVMBK)	13356521
(Obrót, PROKOM)	4892227
(Obrót, COMARCH)	1588614
(Obrót, COMP)	387658
(Obrót, KGHM)	161102739
(Obrót, STALEXP)	142787107
(Obrót, ATM)	1056047
(Obrót, TPSA)	592930352
(Obrót, ORBIS)	13470269
(Obrót, AMREST)	4925768



## MDX

### ⇒ Obroty w 2005 wszystkich spółek spoza sektora bankowego

```
select
  {[Obrót],[2005]} on rows,
  except ([banki].children) on columns
from SPOLKI.spolkiDB
```



## MDX

### ⇒ Funkcjonalność

- eliminowanie duplikatów
- sortowanie
- operacje na zbiorach (union, intersect, minus)
- funkcje wbudowane (matematyczne, nawigowanie w hierarchii wymiaru, około 200 funkcji)
- predykcja regresją liniową, nieliniową