



**pane,
vy máte dneska divný pohled !!!**

Co dělat se SELECT

```
SELECT a, b, c, d, 3*(a+b), ...  
FROM TA JOIN TB USING ...  
      JOIN TC USING ...  
WHERE xyx= (SELECT ... FROM ... WHERE )  
      AND EXIST ( SELECT .... );
```

- to je ale ošklivý dotaz !!!
- ten bych ladil možná 3 hodiny
- fuj



- a co když takový dotaz bude potřeba za měsíc ?



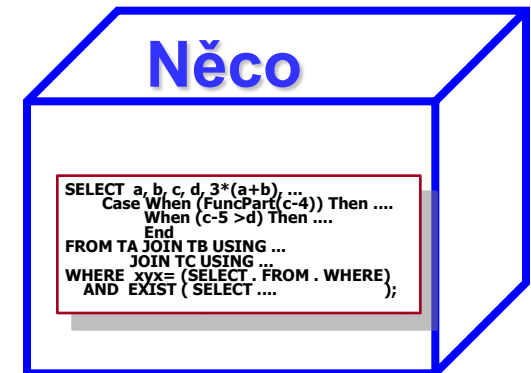
- mám je v zápisníku. Však si je pak najdu !!!



Co dělat se SELECT

```
SELECT a, b, c, d, 3*(a+b), ...  
      Case When (FuncPart(c-4)) Then ....  
            When (c-5 >d) Then ....  
      End  
FROM TA JOIN TB USING ...  
      JOIN TC USING ...  
WHERE xyx= (SELECT ... FROM ... WHERE )  
      AND EXIST ( SELECT .... );
```

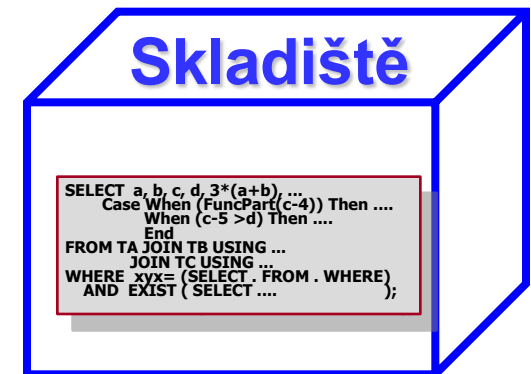
- některý dotaz SELECT je komplikovaný
 - zdlouhavě se ladí
- bylo by vhodné - nějak dotaz uskladnit
- a mít tam dotazy k dispozici



Skladiště [odladěných] dotazů

```
SELECT a, b, c, d, 3*(a+b), ...  
      Case When (FuncPart(c-4)) Then ....  
            When (c-5 >d) Then ....  
      End  
FROM TA JOIN TB USING ...  
      JOIN TC USING ...  
WHERE xyx= (SELECT ... FROM ... WHERE )  
      AND EXIST ( SELECT .... );
```

- některý dotaz SELECT je komplikovaný
 - zdlouhavě se ladí
- bylo by vhodné - nějak dotaz uskladnit
- a mít tam dotazy k dispozici



Co dělat se SELECT

```
SELECT a, b, c, d, 3*(a+b), ...  
      Case When (FuncPart(c-4)) Then ....  
      When (c-5 >d) Then ....  
      End  
FROM TA JOIN TB USING ...  
      JOIN TC USING ...  
WHERE xyx= (SELECT ... FROM ... WHERE )  
      AND EXIST ( SELECT .... );
```

■ co je výsledkem dotazu ?

➤ tabulka



x	x	x	x	x
y	y	y	y	y

■ a co by bylo výsledkem

➤ skladovaného - uloženého dotazu



➤ tabulka

x	x	x	x	x
y	y	y	y	y

Co dělat se SELECT

```
SELECT a, b, c, d, 3*(a+b), ...  
FROM TA JOIN TB USING ...  
      JOIN TC USING ...  
WHERE xyx= (SELECT ... FROM ... WHERE )  
      AND EXIST ( SELECT .... );
```

- na tabulce lze zadat dotaz [Select]
- na tabulce vzniklé dotazem
- lze také zadat další dotaz

```
Select ... From (Select ...From ...)
```

Tab					
a	a	a	a	a	a
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3



x	x	x	x	x
y	y	y	y	y



u	u	u
4	4	4



Dotaz z dotazu

```
Select 1 Jedna From Dual
```

```
Union
```

```
Select 2 Dve From Dual
```

```
Union
```

```
Select 3 Tri From Dual;
```



Jedna
1
2
3

Dotaz z dotazu

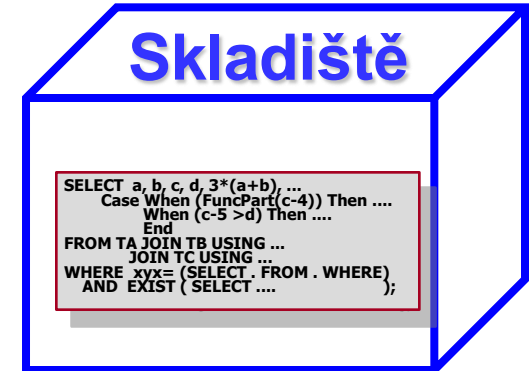
```
Select * From (  
  Select * From (  
    Select * From (  
      Select 1 Jedna From Dual  
    Union  
      Select 2 Dve From Dual  
    Union  
      Select 3 Tri From Dual  
    )  
  )  
);
```



Jedna
1
2
3

jak vyvolat uložený dotaz

- a jak z toho skladiště
- ten dotaz zase dostaneme ?
- jak jej použijeme ?



- na dotaz lze dát dotaz
- zkusíme dát dotaz i na uložený dotaz !
- dotazy budeme ukládat do DB - katalogu
- uloženému dotazu budeme říkat pohled - VIEW
- pohled budeme vyvolávat dotazem



■ **uskladnit dotaz !!!**

■ **vytvořit pohled !!!**

Pohled - VIEW

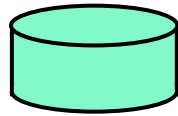
- uskladněný dotaz
- uskladněný [a **pojmenovaný**] povel **Select**
- DB objekt [vytváříme **Create**]
- neobsahující žádná vlastní data
- **virtuální tabulka** bez dat
- má sloupce, nemá řádky
- (**kukátko**) - na skutečná data
- pokud se přistoupí do pohledu
 - pohled se obnoví
 - data se obnoví
 - provede se uskladněný dotaz **Select**



JmenoPohledu				
a	b	c	d	e



Tabulka



- trvalá tabulka
- data uskladněna trvale v databázi (na disku)

Pohled



- data jsou převzata vždy znovu a znovu z jiných tabulek
- provedení katalogizovaného dotazu [Select]
- data - dočasně uložena v paměti počítače

Pohled - VIEW

- pohledy - okna (kukátka)
- pomocí nichž si prohlížíme
- reálná data z reálných tabulek
- virtuální tabulka
 - jejíž obsah je definován dotazem
- pro uživatele DB pohled vypadá
 - jako skutečná tabulka
 - pojmenované sloupce
 - řádky
 - jméno tabulky (pohledu)
- SQL vytváří **iluzi** tabulky

Vytvoření pohledu

Osoba			
Jmeno	Mesto	Adresa	Stat
Alois Švehla	Benešov	Uliční 1	CZ
Jan Novák	Praha	Krásná 4	CZ
Alois Švehla	Ostrava	Ošklivá 7	CZ
Lucie Černá	Bratislava	Ušmouněná 555	SK

```
CREATE VIEW WOsobaS AS
  SELECT Jmeno, Stat
  FROM Osoba
```

- toť vše
- co se do katalogu uložilo

WOsobaS	
Jmeno	Stat
Alois Švehla	CZ
Jan Novák	CZ
Alois Švehla	CZ
Lucie Černá	SK

```
SELECT * FROM WOsobaS
```

Alois Švehla	CZ
Jan Novák	CZ
Alois Švehla	CZ
Lucie Černá	SK

Vytvoření pohledu

Auto				
cAuto	cModel	cBarva	Cena	Pocet
10	1	3	1 000 000	6
11	3	4	500 000	5
12	1	2	2 000 000	1
13	4	1	100 000	20
14	2	3	700 000	10

Model	
cModel	Model
1	BMV
2	Audi
3	Skoda
4	Trabi

Barva	
cBarva	Barva
1	red
2	blue
3	black
4	white

```
CREATE VIEW WAuto AS  
  
SELECT cAuto,Model,Barva,Cena,Pocet  
  
FROM Auta  
JOIN Barva Using (cBarva)  
JOIN Model Using (cModel)
```

```
Select * FROM WAuto
```

WAuto				
cAuto	Model	Barva	Cena	Pocet

10	BMV	black	1 000 000	6
11	Skoda	white	500 000	5
12	BMV	blue	2 000 000	1
13	Trabi	red	100 000	20
14	Audi	black	700 000	10

Vytvoření jiného pohledu pohledu [horizontální]

Auto				
cAuto	cModel	cBarva	Cena	Pocet
10	1	3	1 000 000	6
11	3	4	500 000	5
12	1	2	2 000 000	1
13	4	1	100 000	20
14	2	3	700 000	10

Model	
cModel	Model
1	BMV
2	Audi
3	Skoda
4	Trabi

Barva	
cBarva	Barva
1	red
2	blue
3	black
4	white

```
CREATE VIEW WAutoBMV AS  
  
SELECT cAuto,Model,Barva,Cena,Pocet  
  
FROM   Auta  
       JOIN Barva   Using (cBarva)  
       JOIN Model   Using (cModel)  
WHERE Model.Model = "BMV"
```

```
Select * FROM WAutoBMV
```

WAutoBMV				
cAuto	Model	Barva	Cena	Pocet

- negeneruje výstup
- vrací potvrzení
- byl vytvořen nový DB objekt

Vytvoření jiného pohledu pohledu [horizontální]

Auto				
cAuto	cModel	cBarva	Cena	Pocet
10	1	3	1 000 000	6
11	3	4	500 000	5
12	1	2	2 000 000	1
13	4	1	100 000	20
14	2	3	700 000	10

Model	
cModel	Model
1	BMV
2	Audi
3	Skoda
4	Trabi

Barva	
cBarva	Barva
1	red
2	blue
3	black
4	white

```
CREATE VIEW WAutoBMV AS  
  
SELECT cAuto,Model,Barva,Cena,Pocet  
  
FROM Auta  
      JOIN Barva Using (cBarva)  
      JOIN Model Using (cModel)  
WHERE Model.Model = "BMV"
```

WAutoBMV				
cAuto	Model	Barva	Cena	Pocet

10	BMV	black	1 000 000	6
12	BMV	blue	2 000 000	1

```
Select * FROM WAutoBMV
```


Vytvoření pohledu

```
CREATE VIEW WAuto AS  
  
SELECT cAuto,Model,Barva,Cena,Pocet  
  
FROM Auta  
      JOIN Barva Using (cBarva)  
      JOIN Model Using (cModel)
```

```
Select * FROM WAuto
```

```
CREATE VIEW WAutoBMV AS  
  
SELECT cAuto,Model,Barva,Cena,Pocet  
  
FROM Auta  
      JOIN Barva Using (cBarva)  
      JOIN Model Using (cModel)  
WHERE Model.Model = "BMV"
```

```
Select * FROM WAutoBMV
```

WAuto				
cAuto	Model	Barva	Cena	Pocet
10	BMV	black	1 000 000	6
11	Skoda	white	500 000	5
12	BMV	blue	2 000 000	1
13	Trabi	red	100 000	20
14	Audi	black	700 000	10

WAutoBMV				
cAuto	Model	Barva	Cena	Pocet
10	BMV	black	1 000 000	6
12	BMV	blue	2 000 000	1

Vytvoření pohledu

```
CREATE VIEW WAuto AS  
  
SELECT cAuto,Model,Barva,Cena,Pocet  
  
FROM Auta  
JOIN Barva Using (cBarva)  
JOIN Model Using (cModel)
```

```
Select * FROM WAuto
```

```
CREATE VIEW WAutoBMV AS  
  
SELECT cAuto,Model,Barva,Cena,Pocet  
  
FROM Auta  
JOIN Barva Using (cBarva)  
JOIN Model Using (cModel)  
WHERE Model.Model = "BMV"
```

```
Select * FROM WAutoBMV
```

WAuto				
cAuto	Model	Barva	Cena	Pocet
10	BMV	black	1 000 000	6
11	Skoda	white	500 000	5
12	BMV	blue	2 000 000	1
13	Trabi	red	100 000	20
14	Audi	black	700 000	10

WAutoBMV				
cAuto	Model	Barva	Cena	Pocet
10	BMV	black	1 000 000	6
12	BMV	blue	2 000 000	1

Pohledy

```
CREATE VIEW NazevPohledu [(pojmenováníSloupců)]  
AS SELECT .....
```

- pro pohled lze zadat jména sloupců
- neuvádí se datový typ
 - ten je usouzen z datových typů sloupců v SELECTu
 - počet jmen musí být = počtu sloupců v SELECT

Jména sloupců se musí zadat, když [porad'te mi !!! 

- dotaz obsahuje vypočítané sloupce (výraz), konstantu, ...
- několik sloupců (z různých tabulek) se jmenuje stejně

Vytvoření pohledu

Auta				
cAuto	cModel	cBarva	Cena	Pocet
10	1	3	1 000 000	6
11	3	4	500 000	5
12	1	2	2 000 000	1
13	4	1	100 000	20
14	2	3	700 000	10

Barvy	
cBarva	Barva
1	red
2	blue
3	black
4	white

Modely	
cModel	Model
1	BMV
2	Audi
3	Skoda
4	Trabi

Create View WAutoProv
(Cislo, Model, Bar, Cena, Prov)

AS

```
SELECT cAuto,  
       Model,  
       Barva,  
       Cena,  
       Cena * 0.015
```

```
FROM Auta Join Barvy Using (cBarva)  
       Join Modely Using (cModel);
```

```
SELECT *FROM WAutoProv
```

WAutoProv				
Cislo	Model	Bar	Cena	Prov

10	BMV	black	1 000 000	15 000
11	Skoda	white	500 000	7 500
12	BMV	blue	2 000 000	30 000
13	Trabi	red	100 000	1 500
14	Audi	black	700 000	10 500

Vytvoření pohledu

Auta				
cAuto	cModel	cBarva	Cena	Pocet
10	1	3	1 000 000	6
11	3	4	500 000	5
12	1	2	2 000 000	1
13	4	1	100 000	20
14	2	3	700 000	10

Barvy	
cBarva	Barva
1	red
2	blue
3	black
4	white

Modely	
cModel	Model
1	BMV
2	Audi
3	Skoda
4	Trabi

Create View WAutoProv

AS

```
SELECT cAuto          as Cislo,  
       Model,  
       Barva          as Bar,  
       Cena,  
       Cena * 0,015   as Prov
```

```
FROM Auta Join Barvy Using (cBarva)  
       Join Modely Using (cModel);
```

```
SELECT *FROM WAutoProv
```

WAutoProv				
Cislo	Model	Bar	Cena	Prov

10	BMV	black	1 000 000	15 000
11	Skoda	white	500 000	7 500
12	BMV	blue	2 000 000	30 000
13	Trabi	red	100 000	1 500
14	Audi	black	700 000	10 500

Citlivá data

ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
4	Dana	z	1990	Brno	30 000	hluch
5	Ema	z	1970	Zlin	20 000	dobry
6	Fany	z	1980	Vary	10 000	ryma

ZamNas	
PK	<u>cZam</u>
	Jmeno Poh RokN Pob Plat ZdStav

- některá osobní data
- velmi citlivá
- přístup ke všem datům
- pouze perzonalista
- pro každou pobočku
- samostatný perzonalista
- měl by vidět pouze zaměstnance
- ze "své" pobočky

Citlivá data

ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
4	Dana	z	1990	Brno	30 000	hluch
5	Ema	z	1970	Zlin	20 000	dobry
6	Fany	z	1980	Vary	10 000	ryma

ZamNas	
PK	<u>cZam</u>
	Jmeno Poh RokN Pob Plat ZdStav

- některá osobní data
- velmi citlivá
- přístup ke všem datům
- pouze perzonalista
- pro každou pobočku
- samostatný perzonalista
- měl by vidět pouze zaměstnance
- ze "své" pobočky

Citlivá data

- nutno - 3 pohledy
- pro perzonalisty

```
Create View WZamBrno as  
Select * From ZamNas  
Where Pob = 'Brno';
```

```
Create View WZamVary as  
Select * From ZamNas  
Where Pob = 'Vary';
```

```
Create View WZamZlin as  
Select * From ZamNas  
Where Pob = 'Zlin';
```

ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
4	Dana	z	1990	Brno	30 000	hluch
5	Ema	z	1970	Zlin	20 000	dobry
6	Fany	z	1980	Vary	10 000	ryma

WZamBrno						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
4	Dana	z	1990	Brno	30 000	hluch

WZamVary						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
6	Fany	z	1980	Vary	10 000	ryma

WZamZlin						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
5	Ema	z	1970	Zlin	20 000	dobry

Horizontální pohledy

- omezení přístupu uživatele
- pouze na vybrané řádky
- každý prodejce
- vidí pouze do své pobočky

The diagram shows a table with 5 columns and 10 rows. The top row is highlighted in blue. The next two rows are highlighted in grey. The next three rows are enclosed in a red rectangular box. The next three rows are enclosed in a green rectangular box. The bottom row is enclosed in an orange rectangular box. This illustrates how different horizontal filters (queries) would affect the data visible to different users.

```
SELECT * FROM Pobocky WHERE IdPobocky in (22,23)
```

```
SELECT * FROM Pobocky WHERE Oblast="Praha"
```

```
SELECT * FROM ZAKAZNICI WHERE 500 000 >  
      (SELECT SUM (CASTKA) FROM OBJEDNAVKY  
       WHERE Zakaznik=CisloZakaznika)
```

Citlivá data

ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
4	Dana	z	1990	Brno	30 000	hluch
5	Ema	z	1970	Zlin	20 000	dobry
6	Fany	z	1980	Vary	10 000	ryma

- některá osobní data
- velmi citlivá
- zdravotní stav kromě perzonalistů
- nemá vidět nikdo
- plat a rokNarození - též poněkud citvivé
- data pouze pro účetní
- [jedno účetní oddělení pro celou formu]

Citlivá data

ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
4	Dana	z	1990	Brno	30 000	hluch
5	Ema	z	1970	Zlin	20 000	dobry
6	Fany	z	1980	Vary	10 000	ryma

- některá osobní data
- velmi citlivá
- zdravotní stav kromě perzonalistů
- nemá vidět nikdo
- plat a rokNarození - též poněkud citvivé
- data pouze pro účetní
- [jedno účetní oddělení pro celou formu]

Citlivá data

- nutno - 2 pohledy
- pro účetní
- standard pro ostatní

ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
4	Dana	z	1990	Brno	30 000	hluch
5	Ema	z	1970	Zlin	20 000	dobry
6	Fany	z	1980	Vary	10 000	ryma

Create View **WZam** as

Select **cZam, Jmeno, Poh, Pob**

From ZamNas;

WZam			
cZam	Jmeno	Poh	Pob
1	Adam	m	Brno
2	Bobo	m	Vary
3	Cecil	m	Vary
4	Dana	z	Brno
5	Ema	z	Zlin
6	Fany	z	Vary

Create View **WZamUcet** as

Select **cZam, Jmeno, Poh, RokN, Pob, Plat**

From ZamNas;

WZamUcet					
cZam	Jmeno	Poh	RokN	Pob	Plat
1	Adam	m	1960	Brno	20 000
2	Bobo	m	1970	Vary	30 000
3	Cecil	m	1980	Vary	10 000
4	Dana	z	1990	Brno	30 000
5	Ema	z	1970	Zlin	20 000
6	Fany	z	1980	Vary	10 000

Vertikální pohledy

- omezení přístupu uživatele
- pouze na vybrané sloupce
- osobní (jména, datnar,...)
- finanční (plat,...)
- data v tab užívají různí uživatelé

The diagram shows a table with 8 columns and 7 rows. The top row is highlighted in blue. Three vertical rectangles are drawn over the table: a red one covering the first two columns, a green one covering the third column, and an orange one covering the last three columns. This illustrates how different users can see different subsets of columns from the same table.

```
SELECT Jmeno, Prijmeni, DatNar FROM Osoba
```

```
SELECT Jmeno, Prijmeni, Plat FROM Osoba
```

Přehledy

- a pane účetní
- jak si v tom podniku stojíte
- de to ? de to ?



ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
4	Dana	z	1990	Brno	30 000	hluch
5	Ema	z	1970	Zlin	20 000	dobry
6	Fany	z	1980	Vary	10 000	ryma

- tady přehled o pobočkách

Select

Pob Pobočka,

Count(*) PocZam,

Sum(Plat) Platy,

Avg(Plat) Prumer

From ZamNas

Group By Pob;

Pobočka	PocZam	Platy	Prumer
Brno	2	50 000	25 000
Vary	3	50 000	16 666.6667
Zlin	1	20 000	20 000

- ošklivý součet



Přehledy

- a pane účetní
- jak si v tom podniku stojíte
- de to ? de to ?



ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
4	Dana	z	1990	Brno	30 000	hluch
5	Ema	z	1970	Zlin	20 000	dobry
6	Fany	z	1980	Vary	10 000	ryma

- tady přehled o pobočkách

Select

```
Pob                Pobočka,  
Count(*)           PocZam,  
Sum(Plat)          Platy,  
Round(Avg(Plat)) Prumer
```

From ZamNas

Group By Pob;

Pobočka	PocZam	Platy	Prumer
Brno	2	50 000	25 000
Vary	3	50 000	16 666
Zlin	1	20 000	20 000

- o málo lepší



Seskupené pohledy

Create View WPobSum as

```
Select
  Pob                Pobočka,
  Count(*)           PocZam,
  Sum(Plat)          Platy,
  Round(Avg(Plat), -3) Prumer
From ZamNas
Group By Pob;
```

```
Select * From WPobSum;
```

ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
4	Dana	z	1990	Brno	30 000	hluch
5	Ema	z	1970	Zlin	20 000	dobry
6	Fany	z	1980	Vary	10 000	ryma

Pobočka	PocZam	Platy	Prumer
Brno	2	50 000	25 000
Vary	3	50 000	17 000
Zlin	1	20 000	20 000

Seskupené pohledy

- SELECT obsahuje
 - agregační funkce
 - klauzuli GROUP BY

Spojené pohledy

ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
4	Dana	z	1990	Brno	30 000	hluch
5	Ema	z	1970	Zlin	20 000	dobry
6	Fany	z	1980	Vary	10 000	ryma

WZamBrno						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
4	Dana	z	1990	Brno	30 000	hluch

WZamVary						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
6	Fany	z	1980	Vary	10 000	ryma

WZamZlin						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
5	Ema	z	1970	Zlin	20 000	dobry

```
Create View Wxxxx as
Select * From ZamNas
Where Pob = 'xxx';
```

- pohledy z jedné tabulky
- pohled spojením více tabulek

Auto				
cAuto	cModel	cBarva	Cena	Pocet
10	1	3	1 000 000	6
11	3	4	500 000	5
12	1	2	2 000 000	1
13	4	1	100 000	20
14	2	3	700 000	10

Model	
cModel	Model
1	BMV
2	Audi
3	Skoda
4	Trabi

Barva	
cBarva	Barva
1	red
2	blue
3	black
4	white

```
CREATE VIEW WAuto AS
SELECT cAuto,Model,Barva,Cena,Pocet
FROM Auto
JOIN Barva Using (cBarva)
JOIN Model Using (cModel);
```

WAuto				
cAuto	Model	Barva	Cena	Pocet
10	BMV	black	1 000 000	6
11	Skoda	white	500 000	5
12	BMV	blue	2 000 000	1
13	Trabi	red	100 000	20
14	Audi	black	700 000	10

Spojené pohledy

ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
	Adam	m	1960	Brno	20 000	dobry

WZamBrno						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav

rozlišování pohledů

- pohled z jedné tabulky
- pohled z více tabulek [spojený **join**-pohled]



má **význam** pouze pro

- udělování práv [mírný význam]
- **aktualizovatelné pohledy** [zásadní význam]

JOIN Model Using (cModel);

13	Trabi	red	100 000	20
14	Audi	black	700 000	10

Horizontální pohledy

- nutno - 3 pohledy
- pro perzonalisty

```
Create View WZamBrno as  
Select * From ZamNas  
Where Pob = 'Brno';
```

```
Create View WZamVary as  
Select * From ZamNas  
Where Pob = 'Vary';
```

```
Create View WZamZlin as  
Select * From ZamNas  
Where Pob = 'Zlin';
```

ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
4	Dana	z	1990	Brno	30 000	hluch
5	Ema	z	1970	Zlin	20 000	dobry
6	Fany	z	1980	Vary	10 000	ryma

WZamBrno						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
4	Dana	z	1990	Brno	30 000	hluch

WZamVary						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
6	Fany	z	1980	Vary	10 000	ryma

WZamZlin						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
5	Ema	z	1970	Zlin	20 000	dobry

Zoufalá personalistka Brněnská

ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry

- potřebuji upravit/opravit Adama

```
Update ZamNas  
Set .....
```

Where Jmeno ='Adam';



- ale vždyť ta paní nemá na tab ZamNas žádná práva
- má jenom svůj pohled WZamBrno !!!



```
Update WZamBrno  
Set .....
```

Where Jmeno ='Adam';

WZamBrno						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
4	Dana	z	1990	Brno	30 000	hluch

- to potřebuje měnitelné/aktualizovatelné pohledy



Změnové pohledy

Create View **WZamBrno** as
Select * From ZamNas
Where Pob = '**Brno**';

Update **WZamBrno**
Set Plat = Plat+ 1 000,
Pob = 'Vary'
Where Jmeno = 'Adam';

Insert Into **WZamBrno**
Values (..., 'Vary', ...)



ZamNas						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
4	Dana	z	1990	Brno	30 000	hluch
5	Ema	z	1970	Zlin	20 000	dobry
6	Fany	z	1980	Vary	10 000	ryma

WZamBrno						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Brno	20 000	dobry
4	Dana	z	1990	Brno	30 000	hluch

WZamVary						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
6	Fany	z	1980	Vary	10 000	ryma

WZamBrno						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
4	Dana	z	1990	Brno	30 000	hluch

WZamVary						
cZam	Jmeno	Poh	RokN	Pob	Plat	ZdStav
1	Adam	m	1960	Vary	20 000	dobry
2	Bobo	m	1970	Vary	30 000	rak
3	Cecil	m	1980	Vary	10 000	syfilis
6	Fany	z	1980	Vary	10 000	ryma



```
CREATE VIEW NavezPohledu [(pojmenováníSloupců)]  
AS SELECT .....
```

[WITH [CASCADED | LOCAL] CHECK OPTION]

CHECK OPTION

- pro aktualizovatelné pohledy
- kontrola u **INSERT** a **UPDATE**
 - ➔ zda vložený či měněný řádek patří též do pohledu
- **Local**
- **Cascaded**

Aktualizovatelné pohledy - omezení

```
UPDATE Pohled SET Sloupec= hodnota [,...]
```

pohled musí

- jaká omezení
- pro změnový pohled
- zkuste typovat !!!
- co musí
- co nesmí

Aktualizovatelné pohledy - omezení

```
UPDATE Pohled SET Sloupec= hodnota [,...]
```

pohled musí

- FROM pouze z jedné tabulky
- jen sloupcové dotazy

pohled nesmí obsahovat

- UNION, INTERSECT, EXCEPT
- DISTINCT
- pouze sloupce, žádný výraz, žádná funkce, konstanty
- žádné agregace
- žádný GROUP BY, žádný HAVING
- žádný sloupec 2x

Aktualizovatelné pohledy - trend vývoje

- DB stroje tolerantnější
- nejsou dogmatické dle normy
- co může být bez konfliktu provedeno
- to se provede

Update

Insert

```
Create View Wxyz as
```

```
Select Jmeno,  
Plat,
```

```
Plat + 1000 PlatSOdmenou,  
'--Pridavani--' Popis
```

```
From ZamNas;
```

- dle ANSI normy
- **NotUpdatable**

```
UPDATE Wxyz  
SET Plat= Plat+1 000  
Where Jmeno='Adam';
```

ok

```
UPDATE Wxyz SET PlatSOdmenou = 0 Where Jmeno='Adam';
```

Column 'PlatSOdmenou' is not updatable

Aktualizovatelné pohledy - trend vývoje

- DB stroje tolerantnější
- nejsou dogmatické dle normy
- co může být bez konfliktu provedeno
- to se provede

Update

Insert

```
Create View Wxyz as
```

```
Select Jmeno,  
Plat,
```

```
Plat + 1000 PlatSOdmenou,  
'--Pridavani--' Popis
```

```
From ZamNas;
```

- dle ANSI normy
- **NotUpdatable**

```
UPDATE Wxyz  
SET Plat= Plat+1 000  
Where Jmeno='Adam';
```

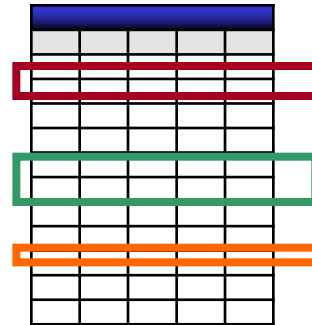
ok

```
UPDATE Wxyz SET Popis = '--Nepridam--' Where Jmeno='Adam';
```

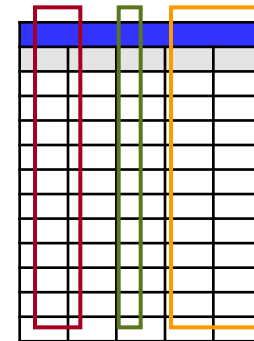
Column 'Popis' is not updatable

Typy pohledů

- horizontální



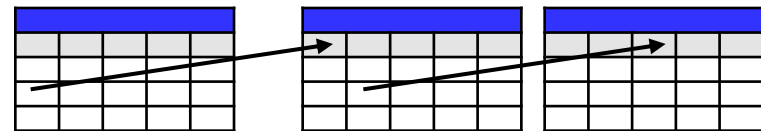
- vertikální



- seskupené

Σ sum ,agr ..

- spojené



- aktualizovatelné

Update

Insert

- materializované (indexové)

Datové sklady

Pohledy - další klauzule

```
CREATE [SQL SECURITY { DEFINER | INVOKER }]
      VIEW NazevPohledu [(pojmenováníSloupců)]
      AS SELECT .....
```

SQL **Security** klauzule určuje

- čí práva se prověřují při použití pohledu
- **DEFINER**
 - právo musí mít definující uživatel [projektant ?]
- **INVOKER**
 - právo musí mít uživatel pohledu, ten
 - co provádí Select ... From Pohled [uživatel aplikace ?]

```
CREATE [Temporary | Temp]
      VIEW NazevPohledu [(pojmenováníSloupců)]
      AS SELECT .....
```

Pohled s klauzulí **Temporary**

- dočasný
- platí pouze po dobu sezení (spojení)
- na konci sezení bude zrušen z katalogu

- pohledy pro **hraní** [**testování**] během sezení
- není třeba jimi "zaprášit" DB

View a změny tabulek

```
CREATE VIEW WBeta AS  
SELECT * FROM Beta
```

Beta			
A	B	C	D
1	1	1	1
2	2	2	2
3	3	3	3

WBeta			
A	B	C	D
1	1	1	1
2	2	2	2
3	3	3	3

```
ALTER TABLE Beta  
ADD COLUMN Text varchar.,  
ADD COLUMN Cislo Int;
```

Beta					
A	B	C	D	Text	Cislo

WBeta			
A	B	C	D

```
SELECT * FROM WBeta;
```

```
CREATE VIEW WAifa AS  
SELECT * FROM Alfa
```

Alfa			
A	B	C	D
1	1	1	1
2	2	2	2
3	3	3	3

WAifa			
A	B	C	D
1	1	1	1
2	2	2	2
3	3	3	3

```
ALTER TABLE Alfa  
DROP COLUMN D;
```

Alfa		
A	B	C
1	1	1
2	2	2
3	3	3

WAifa			
A	B	C	D
1	1	1	1
2	2	2	2
3	3	3	3

```
SELECT * FROM WAifa;
```

View

```
CREATE VIEW WBeta AS
```

```
SE
```

Beta			
A	B	C	D

WBeta	
C	D

- pohled po definici zůstává ZAMRAŽEN
 - ve tvaru tabulek tak
 - jak jsou v okamžiku definice pohledu
- následné změny tabulek
 - se do existujících pohledů nepromítnou



```
A  
DRC
```

2	2	2
3	3	3

```
SELECT * FROM WAlfa;
```

- a proč je pohled již při definici zamražen ?
- proč se to neřeší až při použití ?



- s pohledem se pracuje [téměř] stejně
 ➤ jako s každou jinou tabulkou



- musí mít definovaný počet sloupců
- jména sloupců
- typy sloupců

Pokusy o pohledy

```
Create View W_01 as
Select *
From (Select 1 A Union
      Select 2);
```

A
1
2

View contains subquery in FROM

```
Create Table TW1 (Alfa int);
Create Table TW2 (Alfa int);
```

TW1
Alfa

TW2
Alfa

```
Create View W_02 as
Select TW1.Alf,
       TW2.Alf
From TW1, TW2;
```

W_02	
Alfa	Alfa

Duplicate column name 'Alfa'

Hodnocení pohledů

Výhody

[co myslíte ?]

- zabezpečení
- jednoduchost dotazu
- jednoduchost struktury

- izolování od změn [od jakých ?]
- integrita dat

Nevýhody

[co myslíte ?]

- výkon
- omezení aktualizací

Pohled - shrnutí

umožňují

- nastavit zobrazení databáze tak, aby
 - různí uživatelé mohli DB vidět
 - z různých pohledů
- omezit přístup k datům
 - různí uživatelé mohou vidět
 - pouze určité řádky
 - pouze určité sloupce
- zjednodušit přístup k DB [protože]
 - zobrazují strukturu uložených dat způsobem
 - pro každého uživatele nejpřirozenějším

Pohled - shrnutí

- pojmenovaný dotaz trvale uložený v DB
- jméno pohledu = jméno výsledku dotazu
 - lze použít v jiném dotazu
 - jako tabulku [FROM]
- nemá v DB uloženy hodnoty
- datový objekt, který neobsahuje žádná vlastní data
- pohled je vlastně virtuální tabulka
- virtuální tabulka vypadá jako skutečná
- dotaz [pohledu] se provede
 - kdykoliv je pohled užít jako tabulka
 - v klauzuli FROM



**pane,
vy máte dneska divný pohled !!!**



**promiňte pane,
nevěděla jsem
že to byl pohled
databázový !!!**