SQL

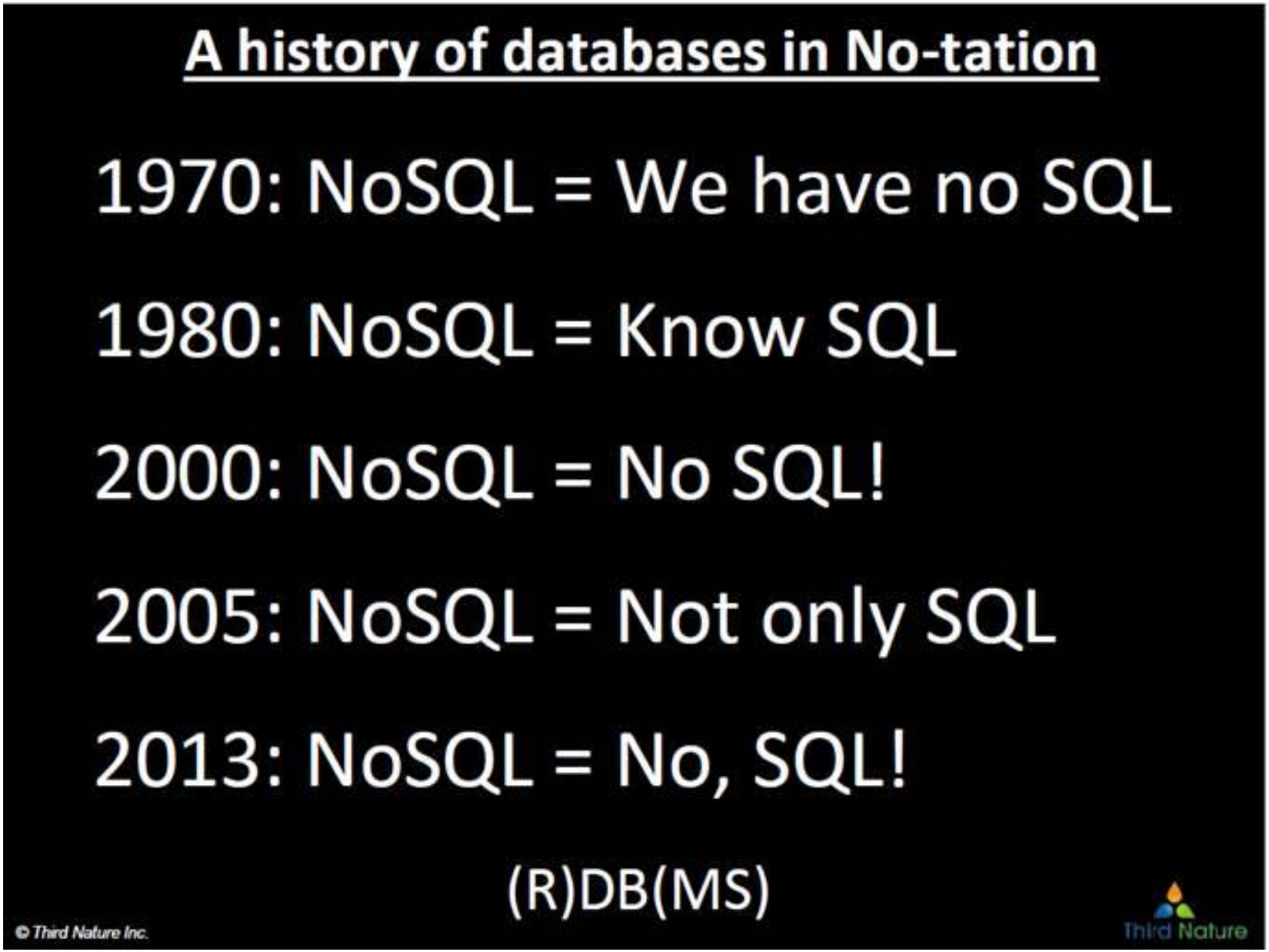
RNDr. Ondřej Zýka

Ing. Tomáš Vichta

Obsah

- Záludnosti SQL
  - Datové typy, práce s datem
  - Příkaz Select
  - Identity
  - Null
  - Join
  - Rekurzivní with, procedurální aspekty with
  - Analytické funkce

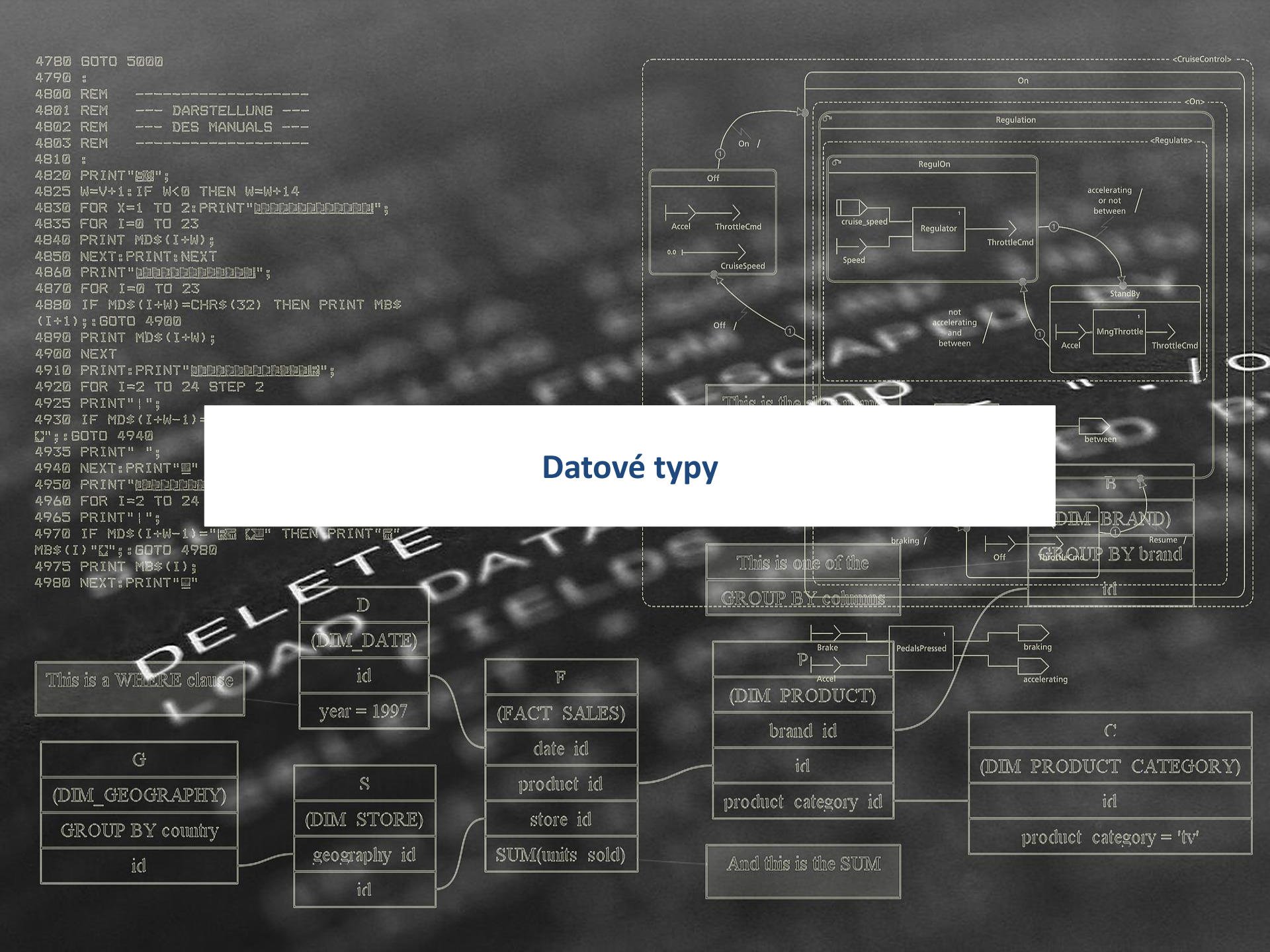
- Problematika migračních projektů


A history of databases in No-tation
1970: NoSQL = We have no SQL
1980: NoSQL = Know SQL
2000: NoSQL = No SQL!
2005: NoSQL = Not only SQL
2013: NoSQL = No, SQL!
(R)DB(MS)
© Third Nature Inc.
Third Nature

Not only SQL

```
// SQL
SELECT id, item, status FROM inventory WHERE status = 'A'

// MongoDB
db.inventory.find( { status: "A" }, { item: 1, status: 1 } )

// Amazon DynamoDB
aws dynamodb query \
    --table-name Invenotory \
    --key-condition-expression "Status = :status" \
    --expression-attribute-values  '{":status":"A"}'

// .NET - classic
db.Inventory.Where(i => i.Status = "A")
            .Select(i => new { i.Id, i.Item, i.Status })

// .NET - LINQ
from i in db.Inventory
where i.Status = "A"
select new { i.Id, i.Item, i.Status }
```

Datové typy


4780 GOTO 5000
4790 :
4800 REM ----------
4801 REM --- DARSTELLUNG ---
4802 REM --- DES MANUALS ---
4803 REM ----------
4810 :
4820 PRINT"
4825 W=V+1:IF W<0 THEN W=W+14
4830 FOR X=1 TO 2:PRINT"
4835 FOR I=0 TO 23
4840 PRINT MD$(I+W);
4850 NEXT:PRINT:NEXT
4860 PRINT"
4870 FOR I=0 TO 23
4880 IF MD$(I+W)=CHR$(32) THEN PRINT MB$
(I+1);:GOTO 4900
4890 PRINT MD$(I+W);
4900 NEXT
4910 PRINT:PRINT"
4920 FOR I=2 TO 24 STEP 2
4925 PRINT"|";
4930 IF MD$(I+W-1)=
";:GOTO 4940
4935 PRINT" ";
4940 NEXT:PRINT"
4950 PRINT"
4960 FOR I=2 TO 24
4965 PRINT"|";
4970 IF MD$(I+W-1)="
MB$(I)"";:GOTO 4980
4975 PRINT MB$(I);
4980 NEXT:PRINT"
<CruiseControl>
On
<On>
Regulation
<Regulate>
RegulOn
On /
Off
Accel
ThrottleCmd
0.0
CruiseSpeed
cruise_speed
Speed
Regulator
ThrottleCmd
accelerating or not between
StandBy
not accelerating and between
Off /
Accel
MngThrottle
ThrottleCmd
between
braking /
Off
ThrottleCmd
Resume /
This is one of the
GROUP BY columns
GROUP BY brand
id
Brake
Accel
P
PedalsPressed
braking
accelerating
(DIM PRODUCT)
brand id
id
product category id
C
(DIM PRODUCT CATEGORY)
id
product category = 'tv'
This is a WHERE clause
D
(DIM_DATE)
id
year = 1997
F
(FACT SALES)
date id
product id
store id
SUM(units sold)
And this is the SUM
G
(DIM_GEOGRAPHY)
GROUP BY country
id
S
(DIM STORE)
geography id
id
DELETE
LOAD DATA
LOAD FIELDS

Datové typy a jejich problematické rysy

- Číselné
  - Rozsah, přesnost, interní implementace, výkonnost, velikost na disku
- Řetězce
  - Národní znaky, skutečná velikost, počáteční a koncové mezery, prázdný řetězec
- Datum a čas
  - Rozsah, aritmetika s datumy, převod na a z řetězce, časové zóny, přesnost, porovnání
- Text
  - Využití v SQL, způsob uložení, operace nad textem
- Boolean
  - Efektivita, indexy
- XML, JSON, Geometrické
  - Množství funkcionality, efektivita použití
- Binární typy
  - Binární operace, převod na a z binární hodnoty

Microsoft datetime – datové typy

Data type	Format	Range	Accuracy	Storage size (bytes)
time	hh:mm:ss.nnnnnnn	00:00:00.0000000 - 23:59:59.9999999	100 ns	3 to 5
date	YYYY-MM-DD	0001-01-01 - 9999-12-31	1 day	3
smalldatetime	YYYY-MM-DD hh:mm:ss	1900-01-01 - 2079-06-06	1 minute	4
datetime	YYYY-MM-DD hh:mm:ss.nnn	1753-01-01 - 9999-12-31	0.00333 s	8
datetime2	YYYY-MM-DD hh:mm:ss.nnnnnnn	0001-01-01 00:00:00.0000000 - 9999-12-31 23:59:59.9999999	100 ns	6 to 8
datetimeoffset	YYYY-MM-DD hh:mm:ss.nnnnnnn [+-]hh:mm	0001-01-01 00:00:00.0000000 - 9999-12-31 23:59:59.9999999 (in UTC)	100 ns	8 to 10

Microsoft datetime - funkce

Syntax	Return data type
SYSDATETIME ()	**datetime2(7)**
SYSDATETIMEOFFSET ( )	**datetimeoffset(7)**
SYSUTCDATETIME ( )	**datetime2(7)**
CURRENT_TIMESTAMP	**datetime**
GETDATE ( )	**datetime**
GETUTCDATE ( )	**datetime**

Microsoft datetime - funkce

Syntax	Return value	Return data type
DATENAME (*datepart* , *date*)	Returns a character string that represents the specified *datepart* of the specified date.	**nvarchar**
DATEPART (*datepart* , *date*)	Returns an integer that represents the specified *datepart* of the specified *date*.	**int**
DAY (*date*)	Returns an integer that represents the day day part of the specified *date*.	**int**
MONTH (*date*)	Returns an integer that represents the month part of a specified *date*.	**int**
YEAR (*date*)	Returns an integer that represents the year part of a specified *date*.	**int**
DATEDIFF (*datepart* , *startdate* , *enddate*)	Returns the number of date or time *datepart* boundaries that are crossed between two specified dates.	**int**
DATEADD (*datepart* , *number* , *date*)	Returns a new **datetime** value by adding an interval to the specified *datepart* of the specified *date*.	The data type of the *date* argument

Microsoft datetime - convert

```
CONVERT ( data_type [ ( length ) ] , expression [ , style ] )
```

Without century (yy) (1)	With century (yyyy)	Standard	Input/Output (3)
-	**0** or **100** (1, 2)	Default	mon dd yyyy hh:miAM (or PM)
1	**101**	U.S.	mm/dd/yyyy
2	**102**	ANSI	yy.mm.dd
3	**103**	British/French	dd/mm/yyyy
4	**104**	German	dd.mm.yy
5	**105**	Italian	dd-mm-yy
6	**106** (1)	-	dd mon yy
7	**107** (1)	-	Mon dd, yy
8	**108**	-	hh:mi:ss
-	**9** or **109** (1, 2)	Default + milliseconds	mon dd yyyy hh:mi:ss:mmmAM (or PM)

Oracle datetime – datové typy

Date type	Timezone	Fractional seconds
January 1, 4712 BC, to December 31, 9999 AD		
DATE	No	No
TIMESTAMP	No	Yes
TIMESTAMP WITH TIME ZONE	Explicit	Yes
TIMESTAMP WITH LOCAL TIME ZONE	Relative	Yes

Oracle datetime - aritmetika

```
with datum as (
  select
    TO_DATE('18.10.2018:23:30:14','DD.MM.YYYY:HH24:MI:SS') value2,
    TO_DATE('03.12.2004:10:34:24','DD.MM.YYYY:HH24:MI:SS') value1
  from dual
)
SELECT
TO_CHAR(value2,'DD.MM.YYYY:HH24:MI:SS') "Value2",
TO_CHAR(value1,'DD.MM.YYYY:HH24:MI:SS') "Value1",
trunc(86400*(value2-value1))-60*(trunc((86400*(value2-
value1))/60)) "Sec",
trunc((86400*(value2-value1))/60)-60*(trunc(((
  86400*(value2-value1))/60)/60)) "Min",
trunc(((86400*(value2-value1))/60)/60)
  -24*(trunc((((86400*(value2-value1))/60)/60)/24)) "Hrs",
trunc((((86400*(value2-value1))/60)/60)/24) "Days"
FROM datum;
```

Oracle TIMESTAMP - aritmetika

```
with datum as (
  select
    TO_TIMESTAMP('18.10.2018:23:30:14','DD.MM.YYYY:HH24:MI:SS'
    TO_TIMESTAMP('03.12.2004:10:34:24','DD.MM.YYYY:HH24:MI:SS') value1
  from dual
)
SELECT
  TO_CHAR(value2,'DD.MM.YYYY:HH24:MI:SS') "Value2",
  TO_CHAR(value1,'DD.MM.YYYY:HH24:MI:SS') "Value1",
  (value2-value1),
  extract (second from (value2-value1) ) "Sec",
  extract (minute from (value2-value1) ) "Min",
  extract (hour from (value2-value1) ) "Hrs",
  extract (day from (value2-value1) ) "Days"
FROM datum;
```

Oracle datetime - funkce

- **ADD_MONTHS**
  - `SELECT ADD_MONTHS(TO_DATE('28-01-2007'), 1) FROM dual;`
  - `SELECT ADD_MONTHS(TO_DATE('30-01-2007'), 1) FROM dual;`
- **CURRENT_DATE**
  - `SELECT SESSIONTIMEZONE, CURRENT_DATE FROM dual;`
- **INTERVAL '<integer>' <unit>**
  - `SELECT TO_CHAR(SYSDATE + INTERVAL '10' MINUTE, 'HH:MI:SS') FROM dual;`
- **MONTHS_BETWEEN**
  - `SELECT MONTHS_BETWEEN(SYSDATE+365, SYSDATE-365) FROM dual;`
- **TO_DATE**(<string1>, [format_mask], [nls_language])

Select


4780 GOTO 5000
4790 :
4800 REM ----------
4801 REM --- DARSTELLUNG ---
4802 REM --- DES MANUALS ---
4803 REM ----------
4810 :
4820 PRINT"
4825 W=V+1:IF W<0 THEN W=W+14
4830 FOR X=1 TO 2:PRINT"
4835 FOR I=0 TO 23
4840 PRINT MD$(I+W);
4850 NEXT:PRINT:NEXT
4860 PRINT"
4870 FOR I=0 TO 23
4880 IF MD$(I+W)=CHR$(32) THEN PRINT MB$
(I+1);:GOTO 4900
4890 PRINT MD$(I+W);
4900 NEXT
4910 PRINT:PRINT"
4920 FOR I=2 TO 24 STEP 2
4925 PRINT"|";
4930 IF MD$(I+W-1)=
";:GOTO 4940
4935 PRINT" ";
4940 NEXT:PRINT"
4950 PRINT"
4960 FOR I=2 TO 24
4965 PRINT"|";
4970 IF MD$(I+W-1)="
MB$(I)"";:GOTO 4980
4975 PRINT MB$(I);
4980 NEXT:PRINT"
<CruiseControl>
On
<On>
Regulation
<Regulate>
RegulOn
On /
Off
Accel
ThrottleCmd
0.0
CruiseSpeed
cruise_speed
Speed
Regulator
ThrottleCmd
accelerating or not between
StandBy
Accel
MngThrottle
ThrottleCmd
not accelerating and between
Off /
between
braking /
Off
Resume /
This is one of the
GROUP BY columns
DIM BRAND)
GROUP BY brand
id
Brake
Accel
P
PedalsPressed
braking
accelerating
(DIM PRODUCT)
brand id
id
product category id
C
(DIM PRODUCT CATEGORY)
id
product category = 'tv'
D
(DIM_DATE)
id
year = 1997
This is a WHERE clause
F
(FACT SALES)
date id
product id
store id
SUM(units sold)
And this is the SUM
G
(DIM_GEOGRAPHY)
GROUP BY country
id
S
(DIM STORE)
geography id
id
DELETE
LOAD DATA
LOAD FIELDS

Příkaz select – několik příkladů

```
select * from author
```

- Základní příkaz
- Mnoho modifikací a rozšíření
- Neexistují vhodná a ustálená pravidla pro formátování
- Jeden příkaz x procedurální definice
- Složitost x efektivita x čitelnost

Příklad

```
SELECT DISTINCT DECODE(town_cd, 1, town) town, -- bud jednoznacny Town nebo null
                DECODE(district_cd, 1, district) district, -- bud jednoznacny District nebo null
                DECODE(region_cd, 1, region) region, -- bud jednoznacny Region nebo null
                zip3
  INTO l_zip.town, l_zip.district, l_zip.region, l_zip.zip3
  FROM (SELECT RTRIM(town) town,
               district,
               region,
               MIN(zip3) OVER() AS zip3, -- nejmensi ZIP pro danou kombinaci
               COUNT(DISTINCT town) OVER() AS town_cd, -- kardinalita Town
               COUNT(DISTINCT district) OVER() AS district_cd, -- kardinalita District
               COUNT(DISTINCT region) OVER() AS region_cd -- kardinalita Region
          FROM c_zip
         WHERE (l_s_profile.district IS NULL OR district = l_s_profile.district) -- district nevyplnen, nebo = c_zip
           AND (l_s_profile.town IS NULL OR town = l_s_profile.town) -- town nevyplnen, nebo = c_zip
           AND (l_s_profile.region IS NULL OR region = l_s_profile.region)); -- region nevyplnen, nebo = c_zip
```

```
select souhrnnyli0_.id as id9_62_, souhrnnyli0_.datum_pocatku_platnosti as datum2_9_62_, souhrnnyli0_.datum_storna as datum3_9_62_, souhrnnyli0_.id_limitu as id4_9_62_, souhrnnyli0_.nazev as nazev9_62_,
souhrnnyli0_.smlouva as smlouva9_62_, nazevlimit1_.klic as klic23_0_, nazevlimit1_.hodnota as hodnota23_0_, pojisteni2_.souhrnny_limit as souhrnny8_64_, pojisteni2_.id as id64_, pojisteni2_.id as id4_1_,
pojisteni2_.cislo_dodatku_storna as cislo2_4_1_, pojisteni2_.datum_storna as datum3_4_1_, pojisteni2_.platnost_od as platnost4_4_1_, pojisteni2_.id_pojisteni as id5_4_1_, pojisteni2_.max_pojistne_plneni as max6_4_1_,
pojisteni2_.nazev_id as nazev14_4_1_, pojisteni2_.predmet_id as predmet11_4_1_, pojisteni2_.sazebnik_id as sazebnik13_4_1_, pojisteni2_.sleva as sleva4_1_, pojisteni2_.smlouva_id as smlouva10_4_1_,
pojisteni2_.souhrnny_limit as souhrnny8_4_1_, pojisteni2_.spoluucast_id as spoluucast12_4_1_, pojisteni2_.typ as typ4_1_, nazevpojis3_.klic as klic25_2_, nazevpojis3_.hodnota as hodnota25_2_, policka4_.pojisteni_id as
pojisteni1_65_, policko5_.id as policko2_65_, policko5_.id as id6_3_, policko5_.ciselna_hodnota as ciselna2_6_3_, policko5_.policko as policko6_3_, policko5_.textova_hodnota as textova3_6_3_, policko5_.vyctova_hodnota as
vyctova4_6_3_, cpolicko6_.id as id48_4_, cpolicko6_.typ as typ48_4_, hodnoty7_.policko as policko66_, hodnoty7_.klic as klic66_, hodnoty7_.klic as klic43_5_, hodnoty7_.hodnota as hodnota43_5_, hodnoty7_.policko as
policko43_5_, chodnotapo8_.klic as klic43_6_, chodnotapo8_.hodnota as hodnota43_6_, chodnotapo8_.policko as policko43_6_, predmet9_.id as id7_7_, predmet9_.cislo_dodatku_storna as cislo2_7_7_, predmet9_.datum_storna
as datum3_7_7_, predmet9_.platnost_od as platnost4_7_7_, predmet9_.predmet_id as predmet5_7_7_, predmet9_.misto_id as misto9_7_7_, predmet9_.nazev_predmetu as nazev11_7_7_, predmet9_.sazebnik as sazebnik7_7_,
predmet9_.smlouva_id as smlouva7_7_7_, predmet9_.specifikace_predmetu as specifik6_7_7_, predmet9_.typ_pojistne_hodnoty as typ10_7_7_, predmet9_.typ_predmetu as typ8_7_7_, predmet9_.vlastnictvi_predmetu as
vlastni12_7_7_, mistopojis10_.id as id1_8_, mistopojis10_.cislo_dodatku_storna as cislo2_1_8_, mistopojis10_.datum_storna as datum3_1_8_, mistopojis10_.platnost_od as platnost4_1_8_, mistopojis10_.misto_id as misto5_1_8_,
mistopojis10_.cinnost_id as cinnost9_1_8_, mistopojis10_.popis as popis1_8_, mistopojis10_.zona_id as zona7_1_8_, mistopojis10_.smlouva_id as smlouva8_1_8_, adresy11_.misto_pojisteni as misto6_67_, adresy11_.id as id67_,
adresy11_.id as id0_9_, adresy11_.psc as psc0_9_, adresy11_.ulice as ulice0_9_, adresy11_.adresa_id as adresa5_0_9_, adresy11_.misto_pojisteni as misto6_0_9_, podnikatel12_.klic as klic21_10_, podnikatel12_.nazev_cinnosti as
nazev2_21_10_, podnikatel12_.odvetvi as odvetvi21_10_, podnikatel13_.klic as klic22_11_, podnikatel13_.nazev_odvetvi as nazev2_22_11_, rizikovapo14_.klic as klic26_12_, rizikovapo14_.hodnota as hodnota26_12_,
rizikovapo14_.platnost_do as platnost3_26_12_, rizikovapo14_.platnost_od as platnost4_26_12_, pojistnasm15_.id as id5_13_, pojistnasm15_.cetnost_placeni as cetnost14_5_13_, pojistnasm15_.cislo_dodatku as cislo2_5_13_,
pojistnasm15_.cislo_navrhu as cislo3_5_13_, pojistnasm15_.cislo_ps as cislo4_5_13_, pojistnasm15_.konec_platnosti as konec5_5_13_, pojistnasm15_.pocatek_platnosti as pocatek6_5_13_, pojistnasm15_.datum_uzavreni as
datum7_5_13_, pojistnasm15_.korespondence as korespo18_5_13_, pojistnasm15_.pobocka_produkce as pobocka16_5_13_, pojistnasm15_.popis as popis5_13_, pojistnasm15_.sleva as sleva5_13_, pojistnasm15_.stav as
stav5_13_, pojistnasm15_.typ_pojistneho as typ17_5_13_, pojistnasm15_.cislo_uctu_pojistnika as cislo10_5_13_, pojistnasm15_.kod_banky as kod15_5_13_, pojistnasm15_.predcisli_uctu_pojistnika as predcisli11_5_13_,
pojistnasm15_.specificky_symbol_uctu_pojistnika as specificky12_5_13_, pojistnasm15_.vysledna_pml as vysledna13_5_13_, cetnostpla16_.klic as klic15_14_, cetnostpla16_.hodnota as hodnota15_14_, koresponde17_.klic as
klic20_15_, koresponde17_.hodnota as hodnota20_15_, osoby18_.smlouva_id as smlouva11_68_, osoby18_.id as id68_, osoby18_.id as id3_16_, osoby18_.adresa_id as adresa10_3_16_, osoby18_.koresp_adresa_id as
koresp14_3_16_, osoby18_.cislo_pasu as cislo2_3_16_, osoby18_.evidencni_vypis as evidencni3_3_16_, osoby18_.funkce as funkce3_16_, osoby18_.ico as ico3_16_, osoby18_.jmeno as jmeno3_16_, osoby18_.nazev_firmy as
nazev7_3_16_, osoby18_.prijmeni as prijmeni3_16_, osoby18_.rodne_cislo as rodne9_3_16_, osoby18_.role as role3_16_, osoby18_.smlouva_id as smlouva11_3_16_, osoby18_.statni_prislusnost as statni15_3_16_,
osoby18_.titul_id as titul13_3_16_, osoby18_.typ as typ3_16_, adresa19_.id as id0_17_, adresa19_.psc as psc0_17_, adresa19_.ulice as ulice0_17_, adresa19_.adresa_id as adresa5_0_17_, adresa19_.misto_pojisteni as
misto6_0_17_, adresa19_.typ as typ0_17_, adresa20_.id as id0_18_, adresa20_.psc as psc0_18_, adresa20_.ulice as ulice0_18_, adresa20_.adresa_id as adresa5_0_18_, adresa20_.misto_pojisteni as misto6_0_18_, adresa20_.typ as
typ0_18_, roleosoby21_.klic as klic27_19_, roleosoby21_.hodnota as hodnota27_19_, statnipris22_.klic as klic29_20_, statnipris22_.hodnota as hodnota29_20_, titul23_.klic as klic31_21_, titul23_.hodnota as hodnota31_21_,
typosoby24_.klic as klic33_22_, typosoby24_.hodnota as hodnota33_22_, cislopoboc25_.klic as klic16_23_, cislopoboc25_.hodnota as hodnota16_23_, pojisteni26_.smlouva_id as smlouva10_69_, pojisteni26_.id as id69_,
pojisteni26_.id as id4_24_, pojisteni26_.cislo_dodatku_storna as cislo2_4_24_, pojisteni26_.datum_storna as datum3_4_24_, pojisteni26_.platnost_od as platnost4_4_24_, pojisteni26_.id_pojisteni as id5_4_24_,
pojisteni26_.max_pojistne_plneni as max6_4_24_, pojisteni26_.nazev_id as nazev14_4_24_, pojisteni26_.predmet_id as predmet11_4_24_, pojisteni26_.sazebnik_id as sazebnik13_4_24_, pojisteni26_.sleva as sleva4_24_,
pojisteni26_.smlouva_id as smlouva10_4_24_, pojisteni26_.souhrnny_limit as souhrnny8_4_24_, pojisteni26_.spoluucast_id as spoluucast12_4_24_, pojisteni26_.typ as typ4_24_, csazebnik27_.id as id50_25_, csazebnik27_.nazev as
nazev50_25_, csazebnik27_.platnost_do as platnost3_50_25_, csazebnik27_.platnost_od as platnost4_50_25_, predmetyap28_.sazebnik as sazebnik70_, predmetyap28_.platnost_do as platnost2_70_, predmetyap28_.platnost_od
as platnost3_70_, predmetyap28_.pojisteni as pojisteni70_, predmetyap28_.predmet as predmet70_, cpojisteni29_.id as id47_26_, cpojisteni29_.nazev as nazev47_26_, konverze30_.pojisteni_id as pojisteni5_71_, konverze30_.id as
id71_, konverze30_.id as id44_27_, konverze30_.id_oj as id2_44_27_, konverze30_.platnost_do as platnost3_44_27_, konverze30_.platnost_od as platnost4_44_27_, konverze30_.pojisteni_id as pojisteni5_44_27_,
okamzikyaa31_.pojisteni_id as pojisteni1_72_, okamzikyaa31_.akce_id as akce2_72_, okamzikyaa31_.okamzik_id as okamzik3_72_, okamzikyaa31_.poradi as poradi72_, okamzikyaa31_.platnost_do as platnost5_72_,
okamzikyaa31_.platnost_od as platnost6_72_, cakce32_.id as id41_28_, cakce32_.java_program as java2_41_28_, cokamzik33_.id as id45_29_, cokamzik33_.nazev as nazev45_29_, policka34_.pojisteni_id as pojisteni1_73_,
policka34_.platnost_do as platnost2_73_, policka34_.platnost_od as platnost3_73_, policka34_.policko_id as policko4_73_, cpolicko35_.id as id48_30_, cpolicko35_.typ as typ48_30_, souhrnneli36_.pojisteni_id as pojisteni1_74_,
csouhrnnyl37_.id as souhrnne2_74_, csouhrnnyl37_.id as id51_31_, csouhrnnyl37_.nazev as nazev51_31_, cpredmet38_.id as id49_32_, cpredmet38_.misto_poj_povinne as misto2_49_32_, cpredmet38_.nazev as nazev49_32_,
cpredmet38_.typ as typ49_32_, typpredmet39_.klic as klic37_33_, typpredmet39_.hodnota as hodnota37_33_, spoluucast40_.klic as klic28_34_, spoluucast40_.hodnota as hodnota28_34_, ctyppojist41_.klic as klic34_35_,
ctyppojist41_.hodnota as hodnota34_35_, otazky42_.pojisteni_id as pojisteni1_75_, otazky42_.otazka_id as otazka2_75_, otazky42_.platnost_do as platnost3_75_, otazky42_.platnost_od as platnost4_75_, cotazka43_.id as
id46_36_, cotazka43_.poradi as poradi46_36_, cotazka43_.typ as typ46_36_, zavisina44_.pojisteni_id as pojisteni1_76_, zavisina44_.master_id as master2_76_, zavisina44_.pozadovany_vysledek as pozadovany3_76_, cotazka45_.id
as id46_37_, cotazka45_.poradi as poradi46_37_, cotazka45_.typ as typ46_37_, predmety46_.smlouva_id as smlouva7_77_, predmety46_.id as id77_, predmety46_.id as id7_38_, predmety46_.cislo_dodatku_storna as
cislo2_7_38_, predmety46_.datum_storna as datum3_7_38_, predmety46_.platnost_od as platnost4_7_38_, predmety46_.predmet_id as predmet5_7_38_, predmety46_.misto_id as misto9_7_38_, predmety46_.nazev_predmetu
as nazev11_7_38_, predmety46_.sazebnik as sazebnik7_38_, predmety46_.smlouva_id as smlouva7_7_38_, predmety46_.specifikace_predmetu as specifik6_7_38_, predmety46_.typ_pojistne_hodnoty as typ10_7_38_,
predmety46_.typ_predmetu as typ8_7_38_, predmety46_.vlastnictvi_predmetu as vlastni12_7_38_, nazevpredm47_.klic as klic24_39_, nazevpredm47_.hodnota as hodnota24_39_, nemovitost48_.predmet as predmet78_,
nemovitost48_.id as id78_, nemovitost48_.id as id2_40_, nemovitost48_.cena as cena2_40_, nemovitost48_.index as index2_40_, nemovitost48_.predmet as predmet2_40_, nemovitost48_.psc as psc2_40_, nemovitost48_.ulice as
ulice2_40_, nemovitost48_.specifikace as specifik7_2_40_, policka49_.predmet_id as predmet1_79_, policko50_.id as policko2_79_, policko50_.id as id6_41_, policko50_.ciselna_hodnota as ciselna2_6_41_, policko50_.policko as
policko6_41_, policko50_.textova_hodnota as textova3_6_41_, policko50_.vyctova_hodnota as vyctova4_6_41_, csazebnik51_.id as id50_42_, csazebnik51_.nazev as nazev50_42_, csazebnik51_.platnost_do as platnost3_50_42_,
csazebnik51_.platnost_od as platnost4_50_42_, typpojistn52_.klic as klic36_43_, typpojistn52_.hodnota as hodnota36_43_, typpredmet53_.klic as klic37_44_, typpredmet53_.hodnota as hodnota37_44_, vlastnictv54_.klic as
klic39_45_, vlastnictv54_.hodnota as hodnota39_45_, vozidla55_.predmet as predmet80_, vozidla55_.vozidlo_id as vozidlo1_80_, vozidla55_.vozidlo_id as vozidlo1_12_46_, vozidla55_.cena as cena12_46_, vozidla55_.index as
index12_46_, vozidla55_.predmet as predmet12_46_, vozidla55_.cena_nova as cena4_12_46_, vozidla55_.cislo_karoserie as cislo5_12_46_, vozidla55_.druh as druh12_46_, vozidla55_.registracni_znacka as registra6_12_46_,
vozidla55_.rok_vyroby as rok7_12_46_, vozidla55_.typ_provedeni as typ9_12_46_, vozidla55_.znacka as znacka12_46_, druhvozidl56_.klic as klic18_47_, druhvozidl56_.hodnota as hodnota18_47_, typprovede57_.klic as klic38_48_,
typprovede57_.hodnota as hodnota38_48_, znackavozi58_.klic as klic40_49_, znackavozi58_.hodnota as hodnota40_49_, zarizeni59_.predmet as predmet81_, zarizeni59_.zarizeni_id as zarizeni1_81_, zarizeni59_.zarizeni_id as
zarizeni1_13_50_, zarizeni59_.cena as cena13_50_, zarizeni59_.index as index13_50_, zarizeni59_.predmet as predmet13_50_, zarizeni59_.rok_vyroby as rok4_13_50_, zarizeni59_.specifikace as specifik5_13_50_, zarizeni59_.typ as
typ13_50_, zarizeni59_.vyrobni_cislo as vyrobni7_13_50_, prilohy60_.smlouva_id as smlouva4_82_, prilohy60_.id as id82_, prilohy60_.id as id8_51_, prilohy60_.druh as druh8_51_, prilohy60_.id_prilohy as id2_8_51_,
prilohy60_.pocet_stran as pocet3_8_51_, prilohy60_.smlouva_id as smlouva4_8_51_, druhpriloh61_.klic as klic17_52_, druhpriloh61_.hodnota as hodnota17_52_, souhrnneli62_.smlouva as smlouva83_, souhrnneli62_.id as id83_,
souhrnneli62_.id as id9_53_, souhrnneli62_.datum_pocatku_platnosti as datum2_9_53_, souhrnneli62_.datum_storna as datum3_9_53_, souhrnneli62_.id_limitu as id4_9_53_, souhrnneli62_.nazev as nazev9_53_,
souhrnneli62_.smlouva as smlouva9_53_, specialniu63_.smlouva_id as smlouva3_84_, specialniu63_.id as id84_, specialniu63_.id as id10_54_, specialniu63_.smlouva_id as smlouva3_10_54_, specialniu63_.text as text10_54_,
spravci64_.smlouva_id as smlouva1_85_, spravce65_.id as spravce2_85_, spravce65_.id as id11_55_, spravce65_.cislo_pobocky as cislo8_11_55_, spravce65_.cislo_spravce as cislo2_11_55_, spravce65_.email as email11_55_,
spravce65_.jmeno as jmeno11_55_, spravce65_.osobni_cislo as osobni5_11_55_, spravce65_.prijmeni as prijmeni11_55_, spravce65_.telefon as telefon11_55_, cislopoboc66_.klic as klic16_56_, cislopoboc66_.hodnota as
hodnota16_56_, stavsmlouv67_.klic as klic30_57_, stavsmlouv67_.hodnota as hodnota30_57_, typpojistn68_.klic as klic35_58_, t
```

Příkaz select

```
select
   city,
   count(*)
from author
where city like '%o%'
group by city
having count(*) > 1
order by city
```

- Základní části příkazu
- Funkce
- Třídění
- Jména tabulek
- Case sensitive/insensitive

Příkaz select

```
select
   l_name,
   NumOfAuthors
from author,
   (select
      city,
      count(*) as NumOfAuthors
    from author
    group by city) CityCount
where
   author.city = CityCount.city
   and CityCount.NumOfAuthors > 1
order by author.f_name
```

- Select *
- Odvozené tabulky (Derived tables)
- „Klasický“ join

Příkaz select

```
with CityCount as
 (select
   city,
   count(*) as NumOfAuthors
  from author
  group by city)
select
  l_name,
  NumOfAuthors
from author join CityCount on
   (author.city = CityCount.city)
where CityCount.NumOfAuthors > 1
order by author.f_name desc
```

- With
- Kvalifikovaná jména (dbo)
- Přepsání jména sloupců
- Unikátnost jmen sloupců
- Třídění
- Ansi join

With a Rekurze

WITH – strukturovaný dotaz

```
WITH cte_sales AS(
    SELECT EmployeeID,
           COUNT(OrderID) as OrdersCount,
           ShipperID
    FROM Orders
    GROUP BY EmployeeID, ShipperID
),

cte_shipper AS(
    SELECT EmployeeID,
           OrdersCount,
           ShipperID
    FROM cte_sales
    WHERE ShipperID IN (2, 3)
),

SELECT ShipperID,
       AVG(OrdersCount) average_order_per_employee
FROM cte_shipper
GROUP BY ShipperID;
```

Rekurze

Vytvořte tabulku s hodnotami
1 až 100

```
ID
-----
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
...
```

Rekurze

- Oracle
- „Klasická“ konstrukce pomocí connect
- Ještě před ANSI definicí

```
SELECT ROWNUM AS ID
FROM dual
CONNECT BY LEVEL <= 10
```

- Rekurzivní with
- ANSI norma, Microsoft, Oracle

```
with x (id) as
    (select 1 as id
     union all
     select id+1 from x
     where id<10)
select id from x
```

Příklad rekurze: Vypište obsah kategorie eshopu

```
;WITH categoryTree(CategoryId, CategoryName, ParentCategoryId, Level) as
(
  SELECT CategoryId, Name, ParentCategoryId, Level = 0
    FROM Nop_Category main
    WHERE main.Name='Drogerie'
  UNION ALL
  SELECT subCat.CategoryId, subCat.Name, subCat.ParentCategoryId, Level =
[Level] + 1
    FROM Nop_Category subCat
    JOIN categoryTree on categoryTree.CategoryId = subCat.ParentCategoryID
)
SELECT * FROM categoryTree
```

CategoryId	CategoryName	ParentCategoryId	Level
5	Drogerie	NULL	0
11	Prací prostředky	5	1
101	Aviváže	11	2
105	Prací gely	11	2
22	Úklidové potřeby	5	1
108	Mopy	22	2
109	Kbelíky	22	2

Diskuze

› Nevýhody rekurze?

Identity - sekvence

Identity/Sekvence

Generování posloupnosti

- Umělé klíče (Id)
- Bez vlivu transakcí
- Požadavek vysokého výkonu
- Alokace bloků, výpadky při pádu serveru nebo v clusteru. (Hi-Lo key generation)
- Identity (MS SQL) – speciální typ sloupce, složitější administrace
- Sequence – další objekt v databázi, složitější administrace
- Nelze zaručit spojitost

Oracle - Sequence

```
CREATE SEQUENCE SQ_TEST
   INCREMENT BY 1
   START WITH 1;

CREATE TABLE T_TEST (ID number, tx varchar (100));

select SQ_TEST.nextval from dual;
select SQ_TEST.currval from dual;

insert into T_TEST values (SQ_TEST.nextval, 'text');
```

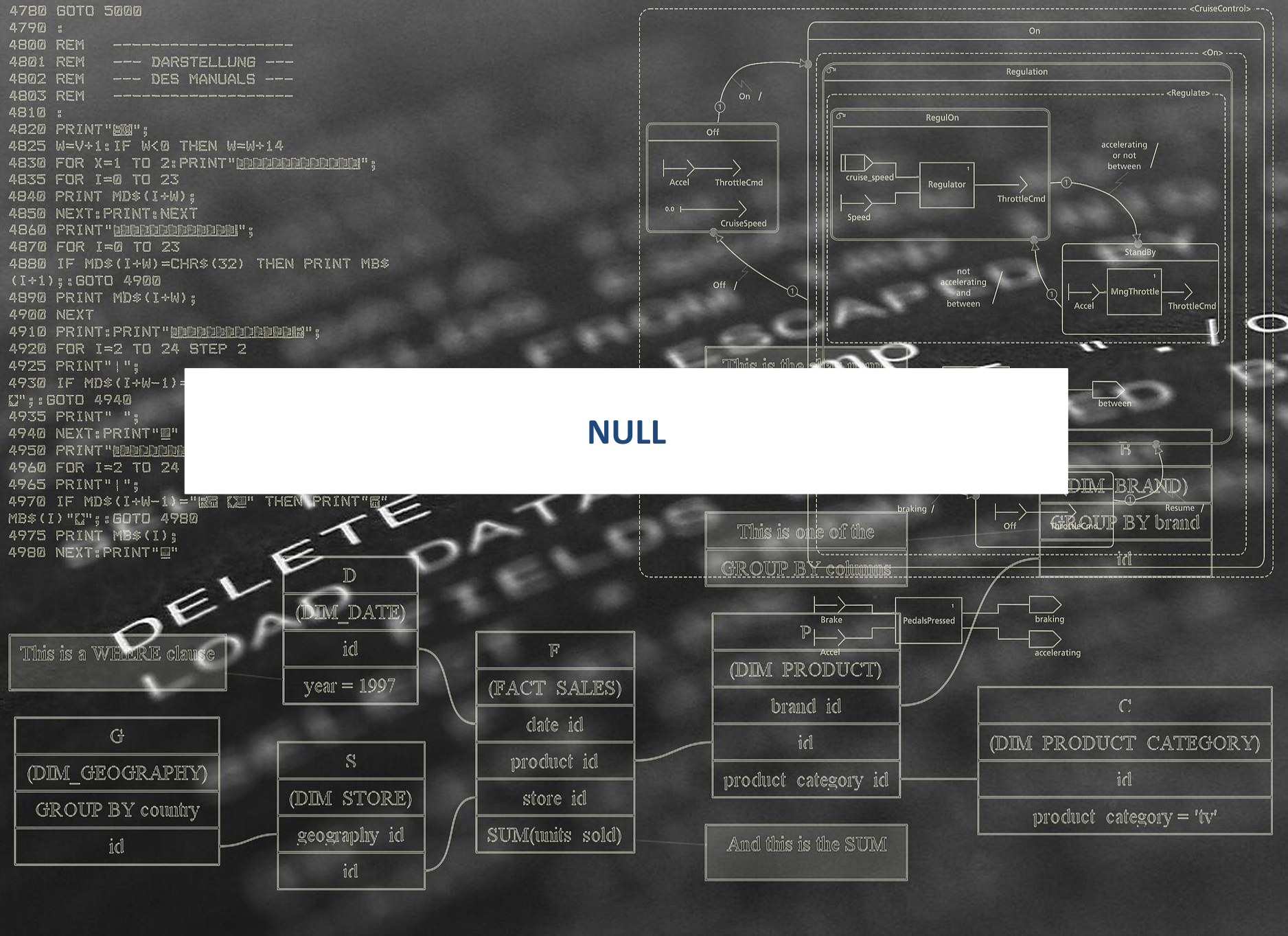
Microsoft - Identity

```
create table T_TEST
(id numeric(10,0) identity,
 tx varchar(100)
)

insert T_TEST values ('text')
```

Otázka a diskuze

- Generovat umělý klíč (ID) v databázi nebo v aplikaci?
- Datový typ klíče?

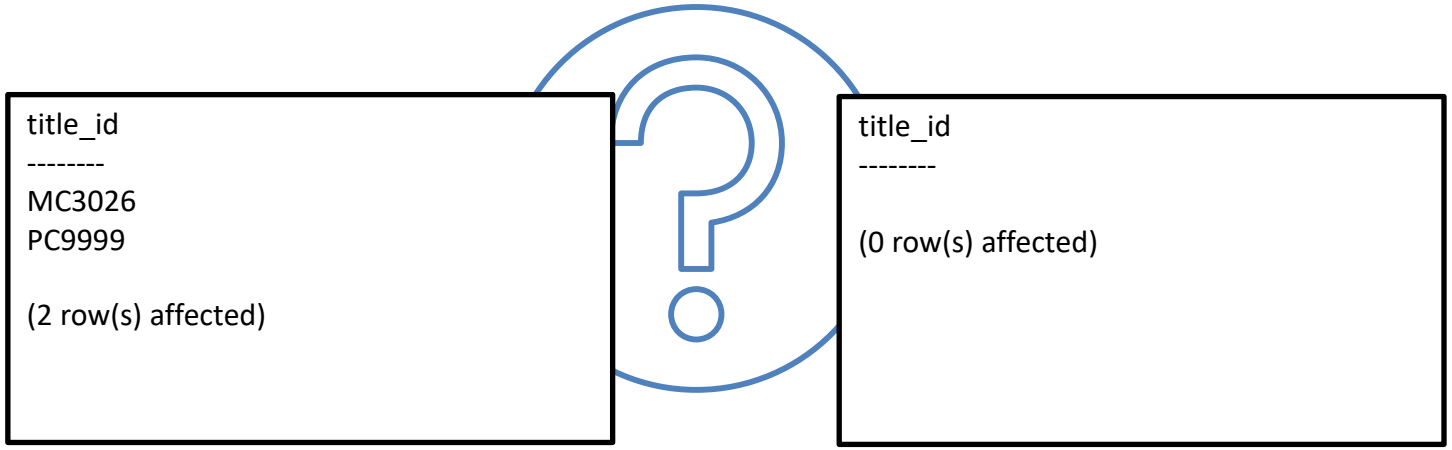
NULL


4780 GOTO 5000
4790 :
4800 REM ---
4801 REM --- DARSTELLUNG ---
4802 REM --- DES MANUALS ---
4803 REM ---
4810 :
4820 PRINT"
4825 W=V+1:IF W<0 THEN W=W+14
4830 FOR X=1 TO 2:PRINT"
4835 FOR I=0 TO 23
4840 PRINT MD$(I+W);
4850 NEXT:PRINT:NEXT
4860 PRINT"
4870 FOR I=0 TO 23
4880 IF MD$(I+W)=CHR$(32) THEN PRINT MB$
(I+1);:GOTO 4900
4890 PRINT MD$(I+W);
4900 NEXT
4910 PRINT:PRINT"
4920 FOR I=2 TO 24 STEP 2
4925 PRINT"|";
4930 IF MD$(I+W-1)=
";:GOTO 4940
4935 PRINT" ";
4940 NEXT:PRINT"
4950 PRINT"
4960 FOR I=2 TO 24
4965 PRINT"|";
4970 IF MD$(I+W-1)="
MB$(I)"";:GOTO 4980
4975 PRINT MB$(I);
4980 NEXT:PRINT"
<CruiseControl>
On
<On>
Regulation
<Regulate>
RegulOn
On /
Off
Accel
ThrottleCmd
0.0
CruiseSpeed
cruise_speed
Speed
Regulator
ThrottleCmd
accelerating or not between
StandBy
Accel
MngThrottle
ThrottleCmd
not accelerating and between
Off /
between
braking /
Off
Resume /
This is one of the
GROUP BY columns
DIM BRAND)
GROUP BY brand
id
D
(DIM_DATE)
id
year = 1997
This is a WHERE clause
F
(FACT SALES)
date id
product id
store id
SUM(units sold)
P
Brake
Accel
PedalsPressed
braking
accelerating
(DIM PRODUCT)
brand id
id
product category id
C
(DIM PRODUCT CATEGORY)
id
product category = 'tv'
G
(DIM_GEOGRAPHY)
GROUP BY country
id
S
(DIM STORE)
geography id
id
And this is the SUM

Null

```
select title_id from title where price = null
```

title_id	title	type	pub_id	price	advance	total_sales
MC3026	The Psychology of Computer Cooking	psychology	0877	(null)	(null)	2606
PC9999	Net Etiquette	popular_comp	1389	(null)	(null)	3528



Null

```
set ANSI_NULLS on
set ANSI_NULLS off
```

- Použití null s = je programátorská chyba
- Záleží na kontextu
- Chování může být mimo vliv programátora

```
select title_id from title where price is null
select title_id from title where price is not null
```

ANSI NULL

- Rozdílné chování než v programovacích jazycích
  - = vs. IS, IS NOT
  - Agregačních funkce nezapočítají NULL
    - `COUNT(city) vs. COUNT(*)`
  - `WHERE CITY IS NULL`
  - `COALESCE, NVL, NVL2, NULLIF`
  - NULL interpretujte jako UNKNOWN – lépe se to chápe
  - V Oracle neexistuje prázdný string!
    - `UPDATE employee set degree='' --nastaví NULL`
    - `SELECT employee WHERE degree=''  --nenajde nikdy nic`

X	Y	X OR Y	X AND Y	NOT X	=
TRUE	**UNKNOWN**	TRUE	UNKNOWN	FALSE	UNKNOWN
FALSE	**UNKNOWN**	UNKNOWN	FALSE	TRUE	UNKNOWN
UNKNOWN	**UNKNOWN**	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN

Příklad

```
CREATE TABLE colors (
    color   VARCHAR(10) NOT NULL PRIMARY KEY
);

INSERT INTO colors VALUES ( 'Blue' );
INSERT INTO colors VALUES ( 'Red' );
INSERT INTO colors VALUES ( 'Green' );
INSERT INTO colors VALUES ( 'Black' );


CREATE TABLE products (
    product_id             NUMBER NOT NULL PRIMARY KEY,
    product_description    VARCHAR(30) NOT NULL,
    color                  VARCHAR(10)
);

INSERT INTO products VALUES(1, 'Ball', 'Red');
INSERT INTO products VALUES(2, 'Bike', 'Blue');
INSERT INTO products VALUES(3, 'Tent', NULL);
```

Najdi nepoužité barvy

Colors

color
Blue
Red
Green
Black

Products

product_id	product_description	color
1	Ball	Red
2	Bike	Blue
3	Tent	NULL

```
SELECT c.color
FROM colors c
WHERE c.color NOT IN (
      SELECT p.color
      FROM products p
);
```

Řešení

```
/* EXISTS */


SELECT c.color

FROM colors c

WHERE NOT EXISTS (SELECT * FROM products p WHERE c.color =
p.color);



/* IS NOT NULL in the subquery */

SELECT c.color

FROM colors c

WHERE c.color NOT IN (SELECT p.color FROM products p

                      WHERE p.color IS NOT NULL

                     );
```

Řešení

```sql
/* MINUS */
SELECT color FROM colors
MINUS
SELECT color FROM products;
```

```sql
/* LEFT OUTER JOIN */
SELECT c.color
FROM colors c
    LEFT OUTER JOIN products p ON c.color = p.color
WHERE
    p.color IS NULL;
```

Dotaz na null hodnotu v kódu

```
CREATE OR REPLACE FUNCTION get_color_count_1 (
    p_color VARCHAR
) RETURN NUMBER AS
    l_num   NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO l_num
    FROM products
    WHERE color = p_color;
    RETURN l_num;
END;
/

SELECT get_color_count_1('Red') FROM dual;

1

SELECT get_color_count_1(null) FROM dual;

0
```

Řešení

```
CREATE OR REPLACE FUNCTION get_color_count_2 (
    p_color VARCHAR
) RETURN NUMBER AS
    l_num   NUMBER;
BEGIN
    IF p_color IS NULL THEN
        SELECT COUNT(*) INTO l_num FROM products WHERE
            color IS NULL;
    ELSE
        SELECT COUNT(*) INTO l_num FROM products WHERE
            color = p_color;
    END IF;
    RETURN l_num;
END;
/
SELECT get_color_count_2(NULL) FROM dual;

SELECT get_color_count_2('Red') FROM dual;
```

Jiné řešení

```
CREATE OR REPLACE FUNCTION get_color_count_3 (
    p_color VARCHAR
) RETURN NUMBER AS
    l_num   NUMBER;
BEGIN
    SELECT COUNT(*) INTO l_num FROM products
        WHERE NVL(color,'N/A') = NVL(p_color,'N/A');
    RETURN l_num;
END;
/



SELECT get_color_count_3(NULL) FROM dual;

SELECT get_color_count_3('Red') FROM dual;
```

Porovnání řádků

```
create or replace FUNCTION compare_products_1(
    p_product_id_1 NUMBER,
    p_product_id_2 NUMBER
) RETURN NUMBER AS
    l_row1 products%ROWTYPE;
    l_row2 products%ROWTYPE;
BEGIN
    select * into l_row1 from products
        where product_id = p_product_id_1;
    select * into l_row2 from products
        where product_id = p_product_id_2;
    if l_row1.product_description = l_row2.product_description
        and l_row1.color = l_row2.color
    then
        return 0;
    else
        return 1;
    end if;
END;
/
```

Porovnávání NULL

- NULL **není** jedna speciální hodnota (jako například v java nebo C)
- Nutno rozdělit případy kdy se může vyskytnout při porovnání null hodnota a kdy ne
- Konstrukce `NVL(color,'N/A') = NVL(p_color,'N/A');`
- Je nutné zvolit vhodnou nevýznamovou hodnotu (`'N/A'`).
  - Jde pro dlouhé řetězce
  - Jde pro datum
  - Nejde pro krátké řetězce (všechny hodnoty mohou být významové)
  - Nejde pro čísla
- Vždy jde použít

```
( (color = p_color) or (color is null and p_color is null) )
```

Join


4780 GOTO 5000
4790 :
4800 REM ----------
4801 REM --- DARSTELLUNG ---
4802 REM --- DES MANUALS ---
4803 REM ----------
4810 :
4820 PRINT"
4825 W=V+1:IF W<0 THEN W=W+14
4830 FOR X=1 TO 2:PRINT"
4835 FOR I=0 TO 23
4840 PRINT MD$(I+W);
4850 NEXT:PRINT:NEXT
4860 PRINT"
4870 FOR I=0 TO 23
4880 IF MD$(I+W)=CHR$(32) THEN PRINT MB$
(I+1);:GOTO 4900
4890 PRINT MD$(I+W);
4900 NEXT
4910 PRINT:PRINT"
4920 FOR I=2 TO 24 STEP 2
4925 PRINT"|";
4930 IF MD$(I+W-1)=
";:GOTO 4940
4935 PRINT" ";
4940 NEXT:PRINT"
4950 PRINT"
4960 FOR I=2 TO 24
4965 PRINT"|";
4970 IF MD$(I+W-1)="
MB$(I)"";:GOTO 4980
4975 PRINT MB$(I);
4980 NEXT:PRINT"
DELETE
LOAD DATA
LOAD FIELDS
<CruiseControl>
On
<On>
Regulation
<Regulate>
RegulOn
On /
Off
Accel
ThrottleCmd
0.0
CruiseSpeed
cruise_speed
Speed
Regulator
ThrottleCmd
accelerating or not between
StandBy
not accelerating and between
Off /
Accel
MngThrottle
ThrottleCmd
between
braking /
Off
ThrottleCmd
Resume /
This is one of the
GROUP BY columns
DIM BRAND)
GROUP BY brand
id
Brake
Accel
P
PedalsPressed
braking
accelerating
(DIM PRODUCT)
brand id
id
product category id
C
(DIM PRODUCT CATEGORY)
id
product category = 'tv'
This is a WHERE clause
D
(DIM_DATE)
id
year = 1997
F
(FACT SALES)
date id
product id
store id
SUM(units sold)
And this is the SUM
G
(DIM_GEOGRAPHY)
GROUP BY country
id
S
(DIM STORE)
geography id
id

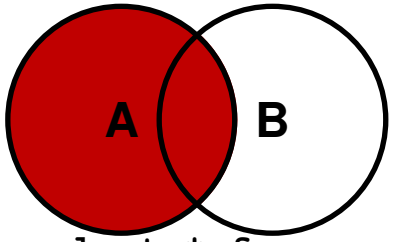
Join NULL hodnot

```
select a.title_id, b.title_id
from title a, title b
where a.price = b.price
```

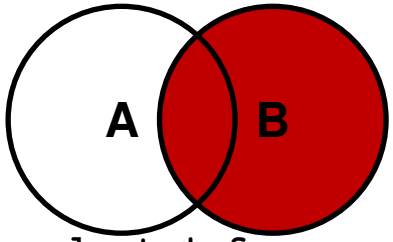
- Výsledek bude bez NULL price

```
… join on (a.price = b.price)
… join on (a.price = b. price or (a. price is null
  and b. price is null) )
… join on (nvl(a.price,'N/A') = nvl(b.price,'N/A'))
```

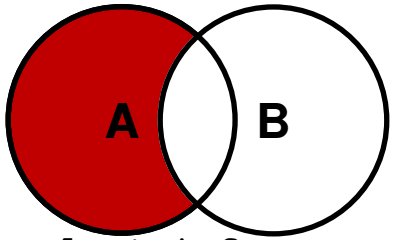
Join



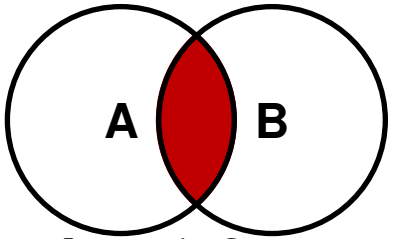

```
select * from
A left join B
on A.PK = B.PK
```



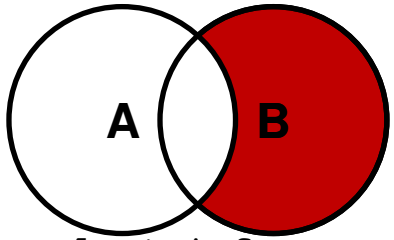

```
select * from
A right join B
on A.PK = B.PK
```



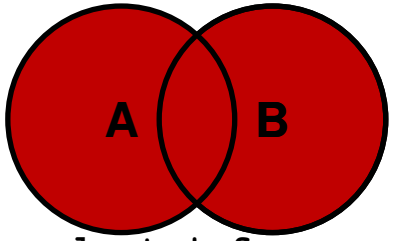

```
select * from
A left join B
on A.PK = B.PK
where B.PK is null
```



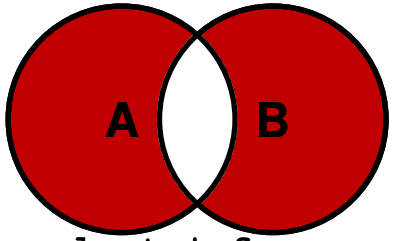

```
select * from
A inner join B
on A.PK = B.PK
```




```
select * from
A right join B
on A.PK = B.PK
where A.PK is null
```




```
select * from
A full outer join B
on A.PK = B.PK
```




```
select * from
A full outer join B
on A.PK = B.PK
where A.PK is null
   or B.PK is null
```

Join

```
select store.store_id, sale.ord_num
from store, sale
where store.store_id = sale.store_id
and sale.ord_date = '01.01.2010'
```

- Klasická notace
- Definice přes kartézský součin a podmínky ve where
- Nedoporučovaný zápis – mladí programátoři už zápisu nerozumí

Join

```
select store.store_id, sale.ord_num
from store
join sale on store.store_id = sale.store_id
where sale.ord_date = '01.01.2010'

select store.store_id, sale.ord_num
from store
join sale on (store.store_id = sale.store_id
              and sale.ord_date = '01.01.2010')
```

- Ansi notace
- Přesně definovaný výsledek operace join

Join

```
select l_name, title from
title join title_author join author
   on (title_author.au_id = author.au_id)
   on (title.title_id = title_author.title_id)

select l_name, title from
title join title_author
on (title.title_id = title_author.title_id)
   join author
on (title_author.au_id = author.au_id)
```

- Ansi notace
  - Definuje pořadí spojení
  - Definuje kdy vyhodnocovat podmínky
- Podstatné u databází, které neoptimalizují pořadí v join operaci

Outer join

```
select store.store_id, count(sale.ord_num)
from store, sale
where store.store_id *= sale.store_id
      and sale.ord_date = '2010-01-01'
group by store.store_id
```

- Microsoft, Sybase – stará (klasická) notace
- Definice přes kartézský součin

Outer join

```
select store.store_id, count(sale.ord_num)
from store, sale
where store.store_id = sale.store_id (+)
and sale.ord_date(+) = TO_DATE('01.01.2010','DD.MM.YYYY')
group by store.store_id;


select store.store_id, count(dd.ord_num)
from store,
  (select store_id, ord_date, ord_num
   from sale
   where ord_date = TO_DATE('01.01.2010','DD.MM.YYYY')) dd
where store.store_id = dd.store_id (+)
group by store.store_id;
```

Oracle používá jinou notaci a semantiku, (+) u datumu je nutné

Outer join

```
select store.store_id, count(sale.ord_num)
from store
left outer join sale on (
            store.store_id = sale.store_id
            and sale.ord_date = TO_DATE('01.01.2010','DD.MM.YYYY'))
group by store.store_id
```

ANSI notace funguje

Datum musí být v podmínce

Chybně:

```
select store.store_id, count(sale.ord_num)
from store
right outer join sale on (store.store_id = sale.store_id)
where sale.ord_date = TO_DATE('01.01.2010','DD.MM.YYYY')
group by store.store_id, sale.ord_date
```

Outer join - shrnutí

- Klasická notace
  - Liší se u jednotlivých databází
  - Lze na ní narazit
  - !!! Není jednoznačná
  - Některé nástroje ji už nepodporují

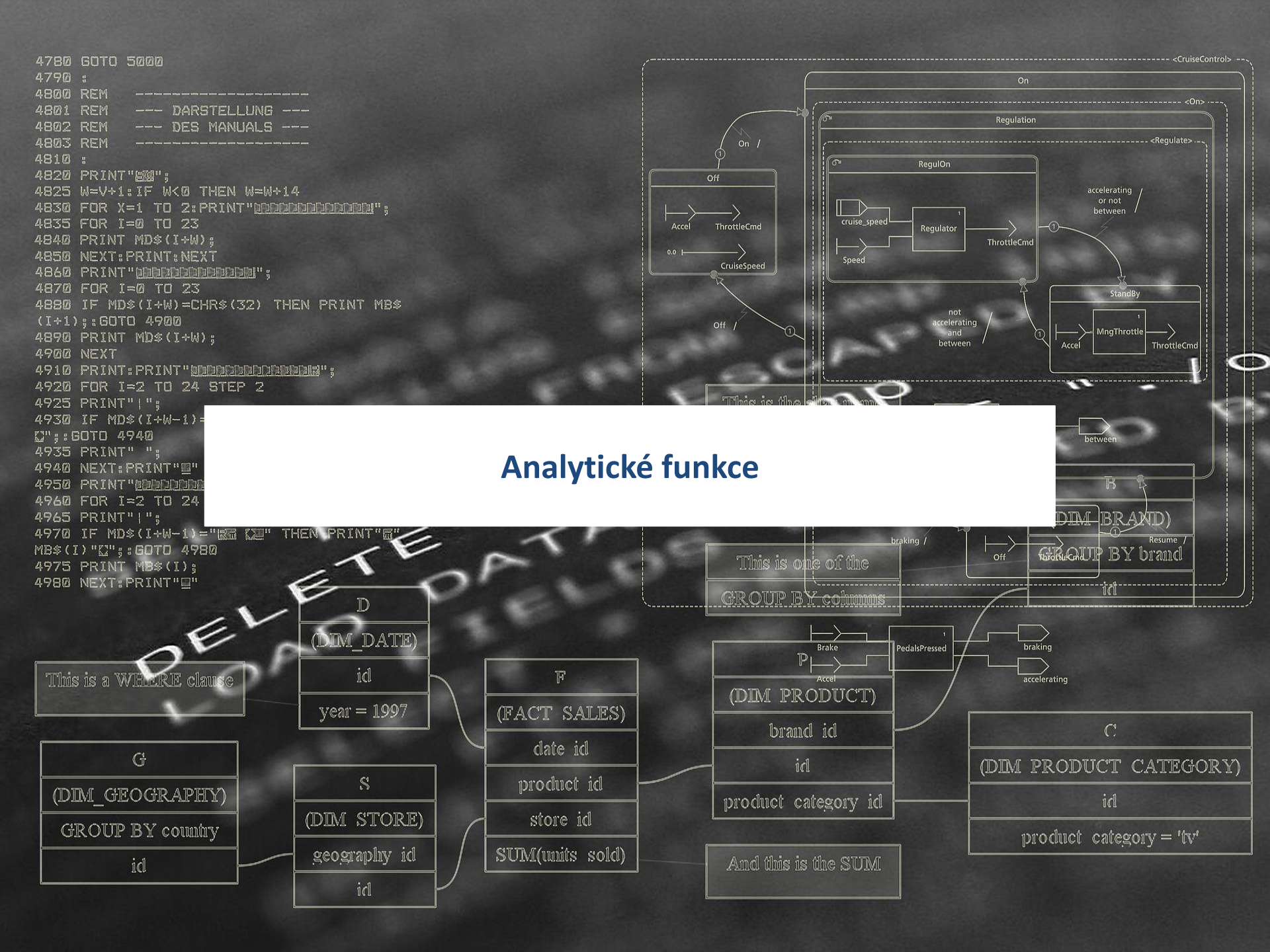
- ANSI notace
  - Jednoznačná
  - Mnoho typů joinů
    - [Inner] Join
    - [Left | right] outer join
    - Full join
    - Cross join
    - !!!! Natural Join

Join - shrnutí

- Podmínka joinu
  - =
  - <
  - Jiná
- Selfjoin
- Join velkého počtu tabulek

```
Select
   count(b.store_id) as poradi,
   a.store_id
from store a join store b on (a.store_id <= b.store_id)
group by a.store_id
order by 1
```

Analytické funkce



Agregační funkce

```
select count(*) from title
select count(price) from title
select count(distinct price) from title
```

- Počet řádků
- Počet neNULLových hodnot price
- Počet různých neNULLových hodnot price

Analytické funkce

CORR(<expression1>, <expression2>) OVER (<analytic clause>)

COVAR_POP(<expression1>, <expression2>) OVER (<analytic clause>)

COVAR_SAMP(<expression1>, <expression2>) OVER (<analytic clause>)

CUME_DIST(<value>) OVER (<partition_clause> <order by clause>)

DENSE_RANK() OVER (<query_partition_clause> <order_by_clause>)

FIRST_VALUE(<expression> [IGNORE NULLS]) OVER (<analytic clause>)

LAG(<value expression>, <offset>, <default>) OVER ([<query partition clause>] <order_by_clause>)

LAST_VALUE (<expression> IGNORE NULLS) OVER (<analytic clause>)

LEAD(<expression, offset, default>) [(<query_partition_clause>)] OVER (<order_by_clause>)

NTILE (<expression>) OVER ([query_partition_clause] <order by clause>)

PERCENT_RANK(<value>) OVER (<partition_clause> <order_by_clause>)

Analytické funkce

PERCENTILE_CONT(<value>) WITHIN GROUP (ORDER BY <expression> [ASC | DESC])
OVER (<partition_clause>)

PERCENTILE_DISC(<expression>) WITHIN GROUP (ORDER BY <order_by_clause>)

RANK(<value>) OVER (<partition_clause> ORDER BY <order_by_clause>)

RATIO_TO_REPORT(<value>) OVER (<partition_clause>)

ROW_NUMBER(<value>) OVER (<partition_clause> ORDER BY <order_by_clause>)

STDDEV([DISTINCT | ALL] <expression>) OVER (<analytic_clause>)

STDDEV_POP(<expression>) OVER (<analytic_clause>)

STDDEV_SAMP(<expression>) OVER (<analytic_clause>)

VAR_POP(<value>) OVER (<analytic_clause>)

VAR_SAMP(<value>) OVER (<analytic_clause>)

VARIANCE([DISTINCT | ALL] <value>) OVER (<analytic_clause>)

Count – analytická funkce

V kolika různých typech knih se vyskytuje kniha se stejnou cenou:

TITLE_ID	TYPE	PRICE	TYPE_COUNT
BU2075	business	2.99	2
MC3021	mod_cook	2.99	2
PS2091	psychology	10.95	1
BU1111	business	11.95	1
BU1032	business	19.99	3
BU7832	business	19.99	3
MC2222	mod_cook	19.99	3
PC1035	popular_comp	19.99	3
PC8888	popular_comp	20	1
PC9999	popular_comp		2
MC3026	psychology		2

Řešení

```
select
    title_id,
    type,
    price,
    count(type) over (partition by price) as type_count
from title
order by type;
```

```
select
    title_id,
    type,
    price,
    count(distinct type) over (partition by price) as type_count
from title
order by type;
```

Řazení – číslování řádků

Seřaďte knihy podle prodejů.

ORDER	TOTAL_SALES	TYPE	TITLE_ID
1	111	psychology	PS2106
2	375	psychology	PS1372
3	375	trad_cook	TC3218
4	2032	mod_cook	MC2222
5	2045	psychology	PS2091
6	3336	psychology	PS7777
7	3876	business	BU1111
8	4072	psychology	PS3333
9	4095	trad_cook	TC7777
10	4095	popular_comp	PC8888
11	4095	business	BU1032
12	4095	business	BU7832
13	8780	popular_comp	PC1035
14	15096	trad_cook	TC4203

Řešení

```
select
        row_number()
        over (ORDER BY TOTAL_SALES) as "ORDER",
        total_sales,
        type,
        title_id
from
        title
```

Rozhodování - CASE

Skryjte opakující se hodnoty.

ORDER	TOTAL_SALES	TYPE	TITLE_ID
1	111	psychology	PS2106
2	375	psychology	PS1372
	375	trad_cook	TC3218
3	2032	mod_cook	MC2222
4	2045	psychology	PS2091
5	3336	psychology	PS7777
6	3876	business	BU1111
7	4072	psychology	PS3333
8	4095	trad_cook	TC7777
	4095	business	BU7832
	4095	business	BU1032
	4095	popular_comp	PC8888
9	8780	popular_comp	PC1035
10	15096	trad_cook	TC4203

Řešení

```
select
    case LAG("ORDER") over (order by "ORDER")
        when "ORDER" then null
        else "ORDER" end as order,
    total_sales,
    type,
    title_id
from
(
    select dense_rank() over (order by total_sales) as "ORDER",
           total_sales,
           type,
           title_id
    from title
)
```

Příklady rozdílů SQL dialektů

Příklady rozdílů SQL dialektů

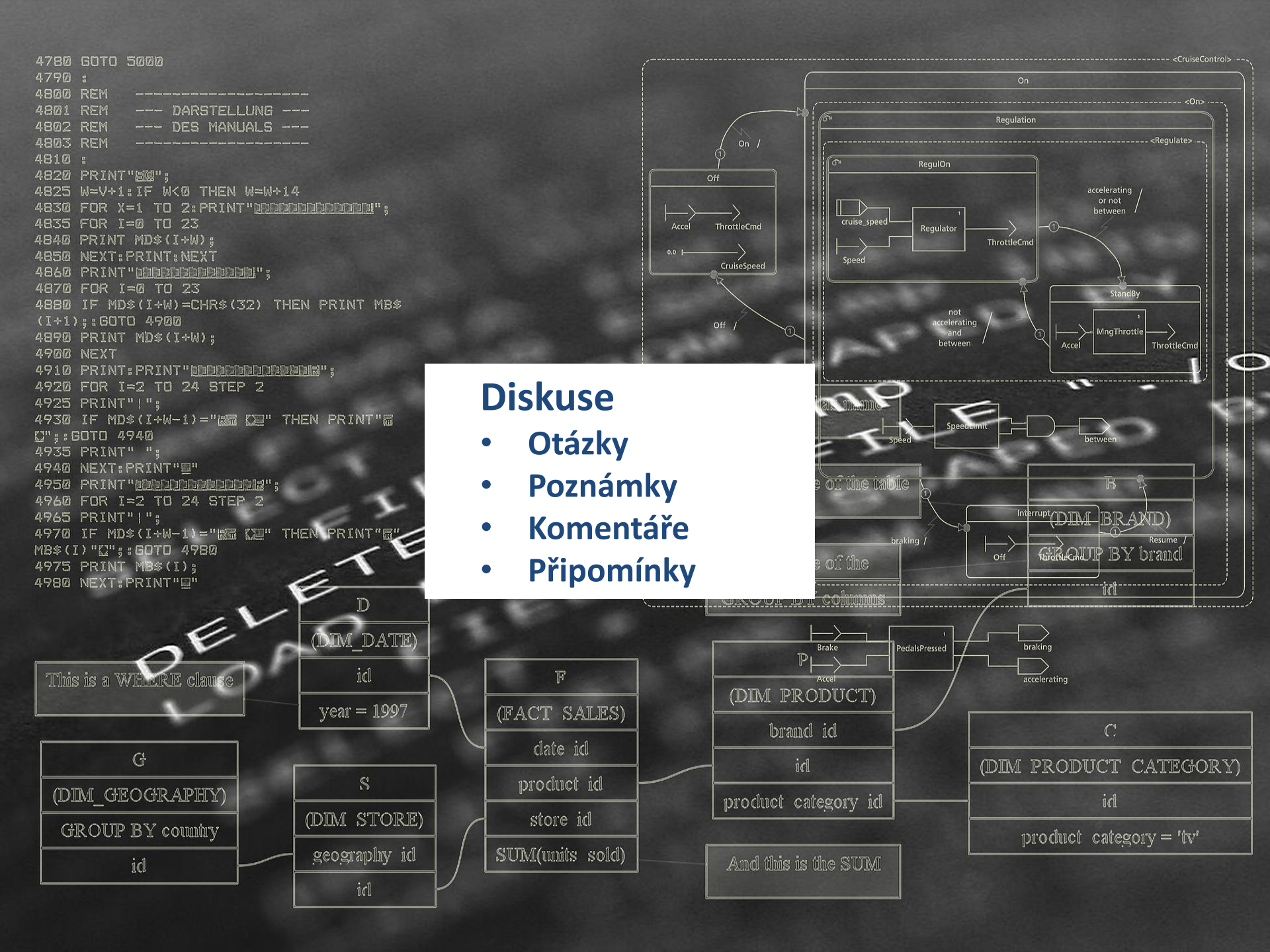
```
ORACLE: SELECT uid, user FROM dual;
MS SQL: SELECT SUSER_ID(), SUSER_SNAME()

ORACLE: ALTER SESSION
        set NLS_LANG='ENGLISH_UNITED KINGDOM.WE8ISO8859P1'
MS SQL: SET LANGUAGE 'British'

ORACLE: ALTER INDEX "IX_Customer_Name" UNUSABLE
MS SQL: ALTER INDEX [IX_Customer_Name] ON dbo.Customer DISABLE

ORACLE: SELECT TOP 100 * FROM Orders
MS SQL: SELECT * FROM Orders WHERE ROWNUM < 101
```

Závěr

- SQL jazyk je velice proprietární
- Mnoho specialit serverů
- Psaní univerzálního kódu obtížné a omezující

Co si zapamatovat

- Na co je potřeba dát pozor při práci s hodnotami typu datum
- Základní části příkazu select
- Jaký je rozdíl mezi implementací rostoucí řady pomocí identity a sekvencí
- Co jsou základní chyby při práci s null hodnotou
- Jaké typy operace join existují, na co si je potřeba dát pozor při jejich použití
- Jak a k čemu se používá klauzule with
- Co to jsou analytické funkce, k čemu slouží



Diskuse

- Otázky
- Poznámky
- Komentáře
- Připomínky