

DBMS Oracle

úvod do systému



Pavel Kříž

Katedra informatiky a kvantitativních metod
Fakulta informatiky a managementu
Univerzita Hradec Králové
Pavel.Kriz@uhk.cz



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Cíl přednášky

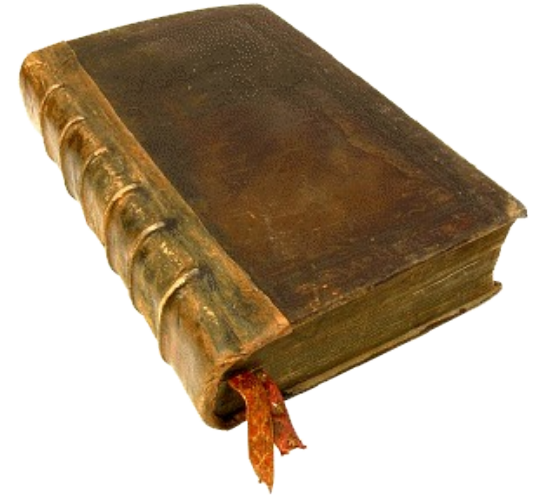
- Seznámení s
 - vybrané implementačními detaily databází Oracle
 - nejčastějšími problémy, na které v začátcích s Oracle narazíme
- Předpokládá
 - znalosti principů relačních databází
 - znalosti SQL
 - znalosti některého z procedurálních rozšíření (alespoň principiálně)

Software

- Oracle server
 - Express edition (XE) – mírně omezená, je volně ke stažení i ke komerčnímu použití
 - Standard Edition (SE) – placená základní edice
 - Enterprise Edition (EE) – pokročilá edice
- Oracle SQL Developer
 - grafický nástroj pro práci s databází (v Javě)
- SQL*plus
 - řádkový klient, vhodný pro volání z dávkových úloh nebo pro účely údržby (např. lokálně na unixovém serveru)

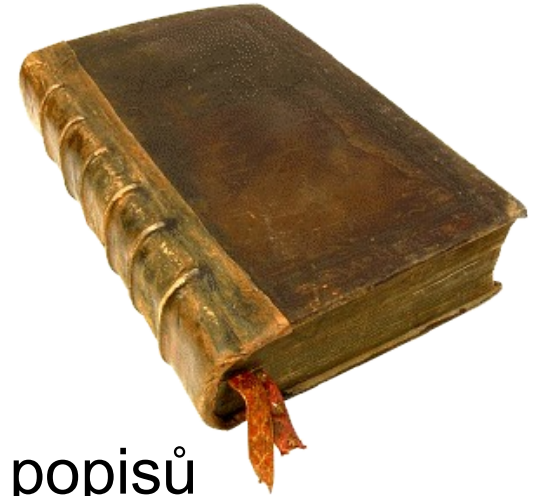
Datový slovník (Data Dictionary) 1/4

- obsahuje informace o
 - uživatelích a jejich právech
 - tabulkách, jejich sloupcích, jejich datových typech, integritních omezních, indexech
 - statistikách nad tabulkami (používá optimalizátor dotazů)
 - definice všech dalších objektů v databázi (pohledy, indexy, procedury, funkce, balíky, trigger, ...)
 - údaje o úložišti (datafiles, objekty alokovaný prostor...)
 - ...



Datový slovník (Data Dictionary) 2/4

- „metadata“
- `select * from DICT[IONARY];`
 - výpis tabulek datového slovníku, včetně popisů
- `select * from TAB;`
 - výpis tabulek, které daný (přihlášený) uživatel vlastní
- `select * from COL;`
 - výpis sloupců tabulek, které daný (přihlášený) uživatel vlastní
- v SQL*plus: `DESC[RIBE] EMPLOYEES;`



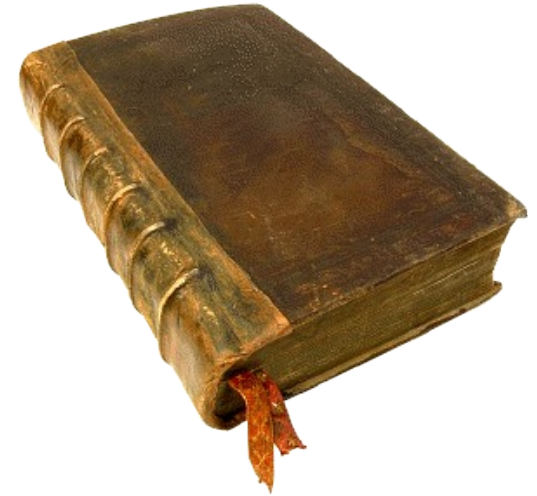
Datový slovník (Data Dictionary) 3/4

- kompletní datový slovník (pohledy):
 - **USER_***
 - informace o objektech, které daný (přihlášený) uživatel vlastní
 - **ALL_***
 - informace o všech objektech, které má daný (přihlášený) uživatel přístupné (nagantované + vlastní)
 - **DBA_***
 - informace o všech objektech v databázi, tyto pohledy má přístupné pouze administrátor



Datový slovník (Data Dictionary) 4/4

- ALL_TABLES – tabulky
- ALL_TAB_COLUMNS – sloupce tabulek
- ALL_CONSTRAINTS – integritní omezení
- ALL_VIEWS – pohledy
- ALL_CATALOG – tabulky, pohledy
- ALL_INDEXES – indexy
- ALL_TRIGGERS – triggery
- ALL_USERS – uživatelé
- ALL_OBJECT – všechny objekty (tabulky, balíky, funkce,...)
- DBA_DATA_FILES – informace o data-files (admin)



Základní datové typy

<http://www.ss64.com/orasyntax/datatypes.html>

- **char(*n*)**

- pevný počet znaků *n* (doplněné mezerami na délku *n*), max. *n* = 2000

Pozor, dat. typy mají trochu jiná omezení v SQL a PL/SQL

- **varchar2(*n*)**

- proměnný počet znaků s max. délkou *n*, max. *n* = 4000; **varchar** je synonymum pro **varchar2**

- **number(*o*, *d*)**

- číselný typ (obecně); *o* = počet platných míst, *d* = počet desetinných míst; mezní hodnoty: *o* = 1 až 38, *d* = -84 až +127
- odvozené typy: **int[eger]**, **dec[imal]**, **smallint**, **real** (jen integritní omezení, v úložišti je pořád typ **number**)

- **date** – datum a čas, **clob/blob** – znakový/bin. „large-object“⁸

Výchozí uživatelé v Oracle

- SYS

- práva pro správu objektů a struktur
- práva pro startup, shutdown, create, backup, recover database (má SYSDBA právo)
- slouží pouze pro administraci



- SYSTEM

- práva pro správu objektů a struktur
- slouží pouze pro administraci

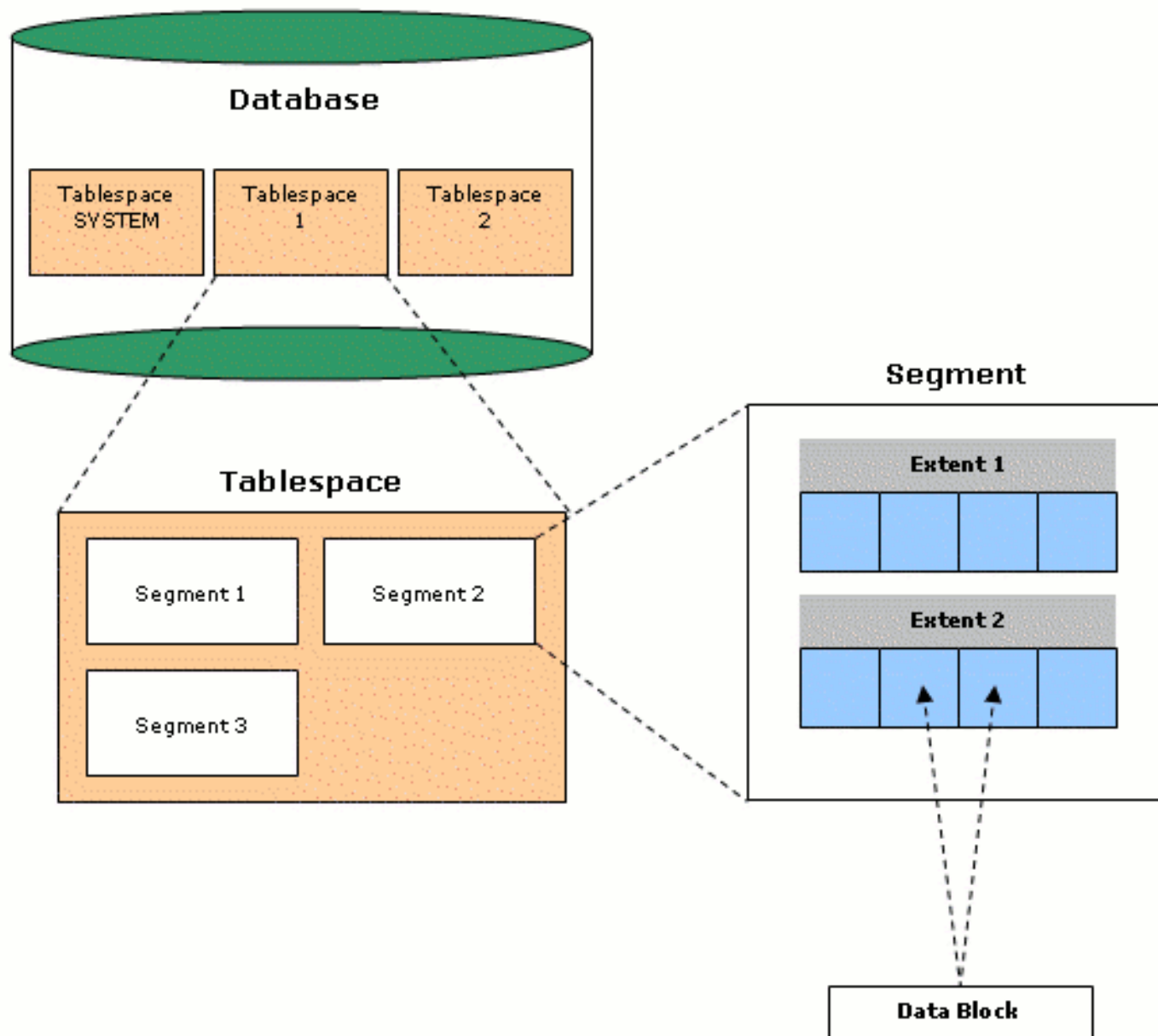
- PUBLIC

- „virtuální“ uživatel, jemuž lze udělit práva na různé objekty a tato práva přecházejí automaticky na každého uživatele systému

Logická struktura databáze

- úložiště
 - logická + fyzická struktura
- databázové schéma
 - tabulky, pohledy, indexy, sekvence,...

Logická struktura úložiště

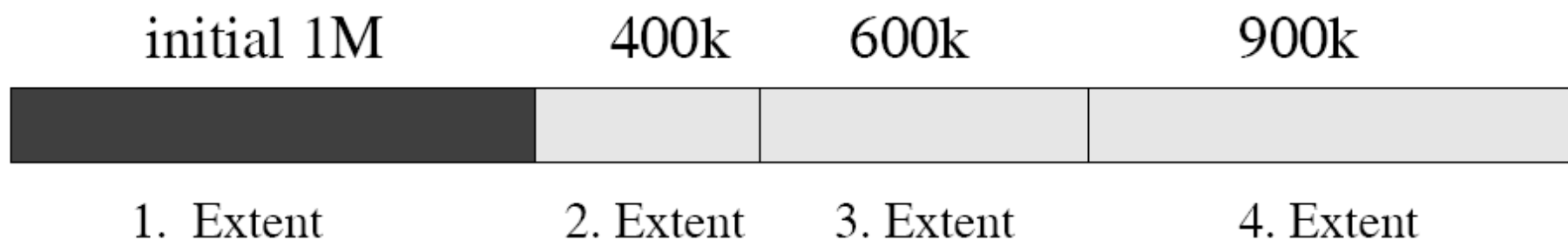


Log. struktura úložiště – Tablespace

- Tablespace
 - každý datový objekt (tabulka,index,...) v databázi je uložen v nějakém tablespace
 - uživatel má nastaven výchozí tablespace pro své objekty
 - tablespace explicitně lze zvolit při vytváření tabulky (CREATE TABLE) v rámci „storage parametrů“
 - skládá se z data-files (fyziké úložiště)
- k čemu jsou Tablespaces dobré
 - řízení využití diskové kapacity (DB nezaplní disk)
 - nastavení kvót pro uživatele (uživatel nezaplní DB)
 - distribuce fyzických úložišť mezi více fyzických disků

Ladění storage parametrů

```
create table STOCKS
  (ITEM varchar2(30) ,
   QUANTITY number(4))
storage (initial 1M next 400k
        minextents 1 maxextents 20
        pctincrease 50);
```

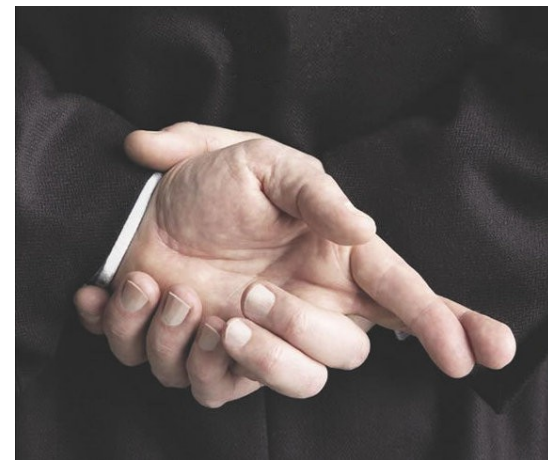


Modifikovatelné pohledy (view)

- data v pohledu lze obecně modifikovat (UPDATE, INSERT nad pohledem)
- nelze modifikovat pohled, který obsahuje alespoň jeden z násl. konstruktů:
 - spojení (join)
 - agregační funkce (sum, min, max, ...)
 - poddotaz s množinovým operátor (in, any, all, exists)
 - group by nebo distinct clause

Typické potíže s Oracle v začátcích

- prázdný řetězec je totéž co NULL
 - v dokumentaci se píše, že v budoucích verzích Oracle se toto chování může změnit, ale nevěřte jim
 - nepříjemné důsledky
 - výraz **x=' '** (prázdný řetězec [apostrofy]) není nikdy pravdivý (nehledě na hodnotu X), neboť se v něm vyskytuje NULL (' ' je s NULL ekvivalentní), tudíž také výraz **' '= ' '** není pravdivý
 - když uložíte někam prázdný řetězec a přečtete jej zpět, výsledek bude NULL
 - zda je řetězec prázdný musíme testovat pomocí operátoru IS NULL



Typické potíže s Oracle v začátcích

- Oracle neumí omezit výsledek dotazu pomocí LIMIT



- musíme použít pseudo-sloupec **rownum**

- pokud neřadíme výsledek, můžeme jednoduše takto:

```
select * from emp where rownum <= 20;
```

- pokud řadíme, musíme SELECT vnořit a omezit nadřazený dotaz (vysvětlení: rownum se chová podobně jako skutečný sloupec a selekce dle podmínky ve WHERE klauzili proběhne vždy dříve, než řazení):

```
select * from (select * from emp order by salary) where rownum <= 20;
```


Typické potíže s Oracle v začátcích

- historicky Oracle nepodporoval ANSI/ISO standardizovaný zápis spojení (join)
 - neznal CROSS JOIN, NATURAL JOIN, JOIN...USING..., JOIN...ON..., ...OUTER JOIN...
 - nejvíc zřejmě překvapí zápis OUTER JOIN:

ISO/ANSI syntaxe:

```
SELECT e.last_name,  
       d.department_name  
FROM   employees e  
       LEFT OUTER JOIN departments d  
       ON (e.department_id =  
           d.department_id);
```

Oracle původní propriet. syntaxe:

```
SELECT e.last_name,  
       d.department_name  
FROM   employees e,  
       departments d  
WHERE  e.department_id =  
       d.department_id(+);
```

Úvod do PL/SQL

- procedurální rozšíření
 - uložené procedury a funkce (zapouzdření logiky, úkony související se správou databáze,...)
 - triggery – reagují na DML operace (obsluha událostí, auditing,...)
 - syntaxe vychází z jazyka Ada („příbuzný“ jazyka Pascal → velmi podobná syntaxe s Pascalem)
- PL/SQL blok
 - anonymní
 - pojmenovaný - procedury,..

PL/SQL blok

- jednoduchý anonymní PL/SQL blok:

DECLARE

-- volitelná část deklarací

BEGIN

-- vlastní tělo (program)

EXCEPTION

-- volitelná obsluha vyjímek

END

/* víceřádkový komentář */

PL/SQL - proměnné

- deklarace proměnné:

`<variable name> [constant] <data type> [not null] [:= <expression>];`

- data type

- datové typy z SQL (pozor, že mohou mít jiná omezení než v SQL)
- strukturované datové typy
- speciální dat. typy pro PL/SQL
- místo typu lze „odkázat“ na dat. typ sloupce z tabulky:

`EMPLOYEE.EMPLOYEE_ID%TYPE`

případně na definici celého záznamu (prom. typu záznam):

`EMPLOYEE%ROWTYPE`

PL/SQL blok 2

- anonymní PL/SQL blok s programem:

```
DECLARE
    number1 NUMBER(2) := 17;
    greeting VARCHAR2(255) := 'Hello';
    date1 DATE := SYSDATE;
    fname VARCHAR2(255);
BEGIN
    SELECT first_name INTO fname
    FROM address
    WHERE last_name = 'Smith';
    DBMS_OUTPUT.PUT_LINE(greeting||' '||fname||' today is '||
                          to_char(date1));
END;
```

PL/SQL systémové vyjímky

- Zachytávání vyjímek (v tomto případě „systémových“)

```
DECLARE
```

```
    emp_sal emp.sal%TYPE;
```

```
    emp_no emp.emp_no%TYPE;
```

```
BEGIN
```

```
    SELECT empno, sal INTO emp_no, emp_sal
```

```
        FROM emp WHERE ename = 'KING';
```

```
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
```

```
        DBMS_OUTPUT.PUT_LINE('No King found');
```

```
END;
```

- typické sys. vyjímky: **NO_DATA_FOUND**, **TOO_MANY_ROWS**

Procedure, funkce, balíky

- Procedure – nemají návratovou hodnotu
`CREATE OR REPLACE PROCEDURE ...`
- Funkce – mají návratovou hodnotu, lze je volat i z prostředí SQL (v SELECTech)

`CREATE OR REPLACE FUNCTION ...`

- Procedure a funkce lze organizovat do balíků (package), jméno balíku je předponou jména funkce (např. main_api.my_procedure)

`CREATE OR REPLACE PACKAGE ...`

`CREATE OR REPLACE PACKAGE BODY ...`

- Balík má část deklarací (hlavičky procedur a fcí) a tělo (body - implementace)