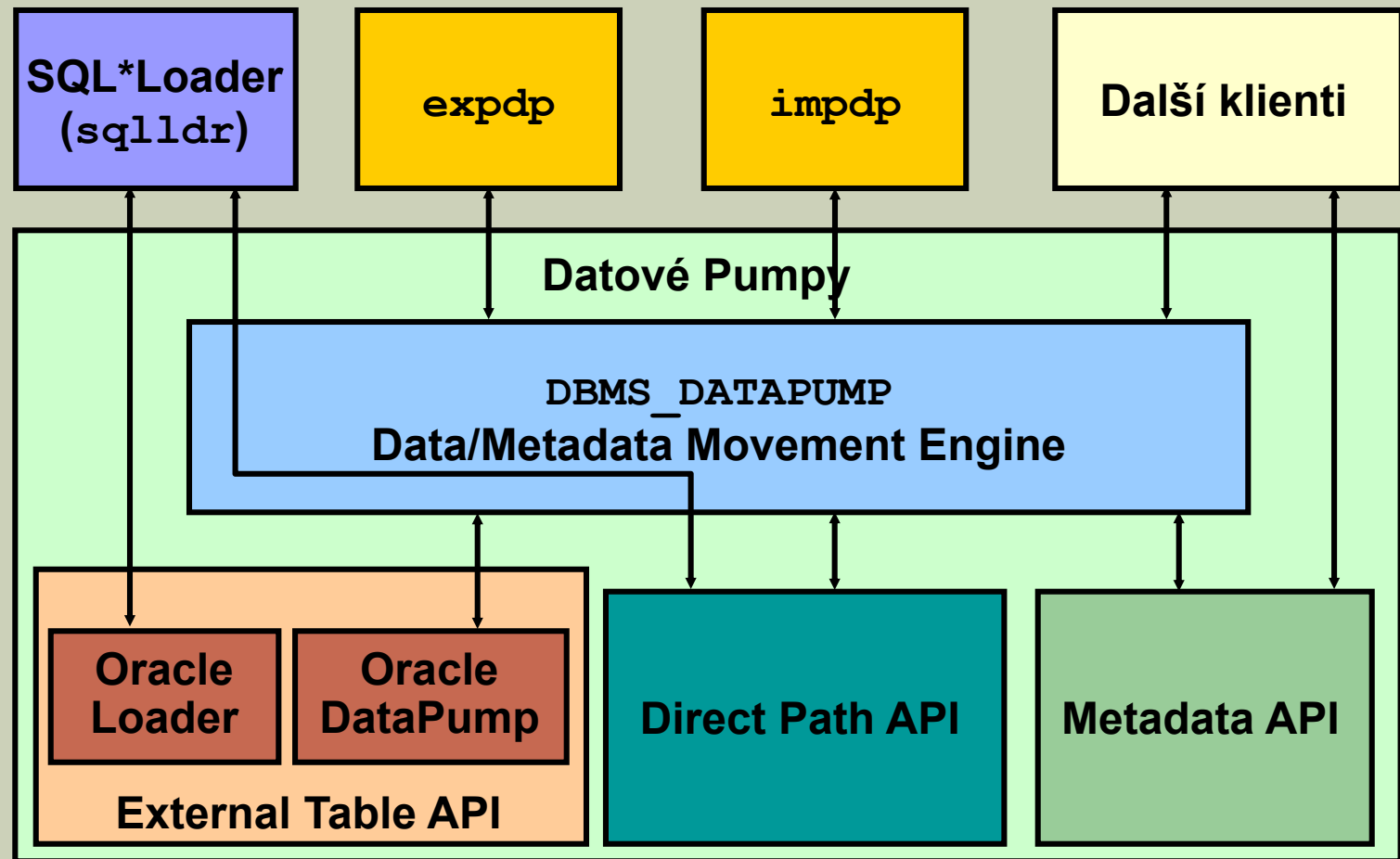


# PŘESUNY DAT

# PŘESUNY DAT: ARCHITEKTURA



# DATOVÉ PUMY: PŘEHLED

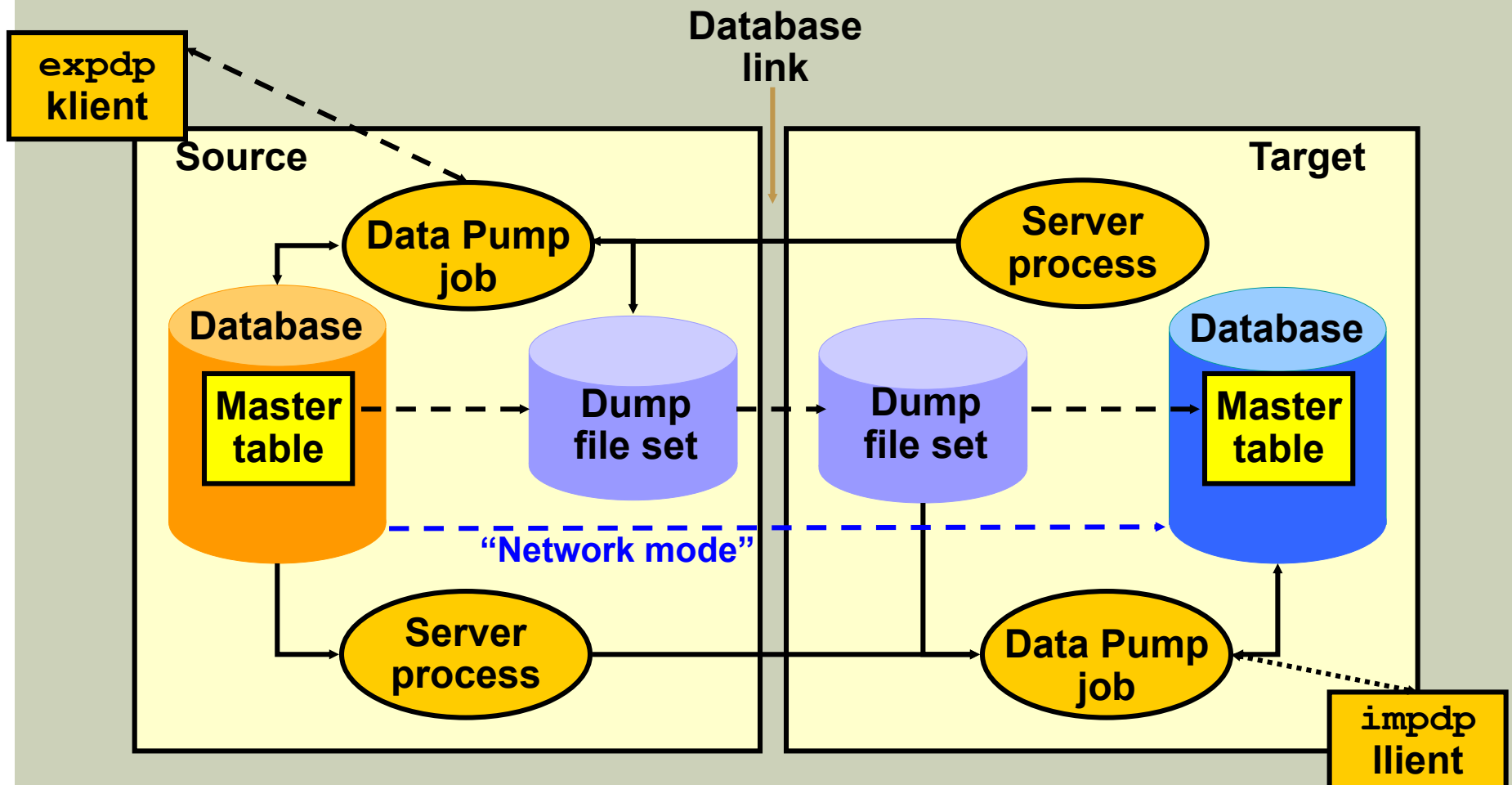
- Datové pumpy jsou instalovány na straně serveru a umožňují velmi rychlé přesuny dat, můžeme k nim přistupovat pomocí:
  - Balíku `DBMS_DATAPUMP`
  - Následujících utilit:
    - `expdp`
    - `impdp`
  - Webového rozhraní
  - Umožňuje následující čtveřici operací s daty:
    - Přímé kopírování souborů
    - Tzv. Direct path
    - Externí tabulky tables
    - Podpora síťového sdílení
  - Umožňuje úlohu spustit a odpojit se, poté je možné se kdykoliv k probíhající úloze připojit



# DATOVÉ PUMPY: VÝHODY

- Základní vlastnosti datových pump:
  - Možnost selektivního výběru konkrétních dat
  - Explicitní specifikaci verze databáze
  - Paralelní spouštění
  - Odhad potřebného místa
  - Možnost síťových přesunů
  - Možnost přemapování (tablesace, uživatel ...)
  - Vzorkování dat a komprese metadat
  - Komprese dat během exportu
  - Šifrování
  - Možnost exportovat XMLType do CLOBu
  - Legacy mód pro kompatibilitu se s utilitami export a import

# DATOVÉ PUMPY: PŘEHLED



# DATOVÉ PUMPY: OVLÁDÁNÍ

- Export pomocí datových pump můžeme provádět pomocí:
  - Příkazové řádky
  - Souboru parametrů
  - Interaktivně
  - Grid Control Enterprise Manageru
- Módy pro export a import pomocí datových pump:
  - Full
  - Schema
  - Table
  - Tablespace
  - Transportable tablespace

# IMPDP

- Import pomocí datových pump můžeme vyvolat z příkazové řádky včetně všech parametrů:

```
$ impdp hr DIRECTORY=DATA_PUMP_DIR \  
DUMPFILE=HR_SCHEMA.DMP \  
PARALLEL=1 \  
CONTENT=ALL \  
TABLES="EMPLOYEES" \  
LOGFILE=DATA_PUMP_DIR:import_hr_employees.log \  
JOB_NAME=importHR
```

# PŘEMAPOVÁNÍ

## ■ Co můžeme přemapovat:

- **Datové soubory** REMAP\_DATAFILE
- **Tablespace** REMAP\_TABLESPACE
- **Schéma** REMAP\_SCHEMA
- **Tabulky** REMAP\_TABLE
- **Data** REMAP\_DATA

```
REMAP_TABLE = 'EMPLOYEES' : 'EMP'
```



# UMÍSTĚNÍ SOUBORŮ

- Originální `exp` a `imp` utility: Úplné názvy souborů
- Datové pumpy – jeden adresář pro všechny soubory
  - Default: `DATA_PUMP_DIR` adresář
  - Uživatel, který provádí export nebo import pomocí datových pump musí mít do `: DATA_PUMP_DIR` právo zápisu/čtení
  - Založíme pomocí `CREATE DIRECTORY` a práva přidáme pomocí příkazu `GRANT`

```
SQL> create or replace directory DATA_PUMP_DIR as '/tmp';
```

Adresář vytvořen.

```
SQL>
```

# UMÍSTĚNÍ SOUBORŮ

```
SQL> select owner,directory_name,directory_path
from all_directories where
directory_name='DATA_PUMP_DIR';
```

```
OWNER
```

```
-----
```

```
DIRECTORY_NAME
```

```
-----
```

```
DIRECTORY_PATH
```

```
-----
```

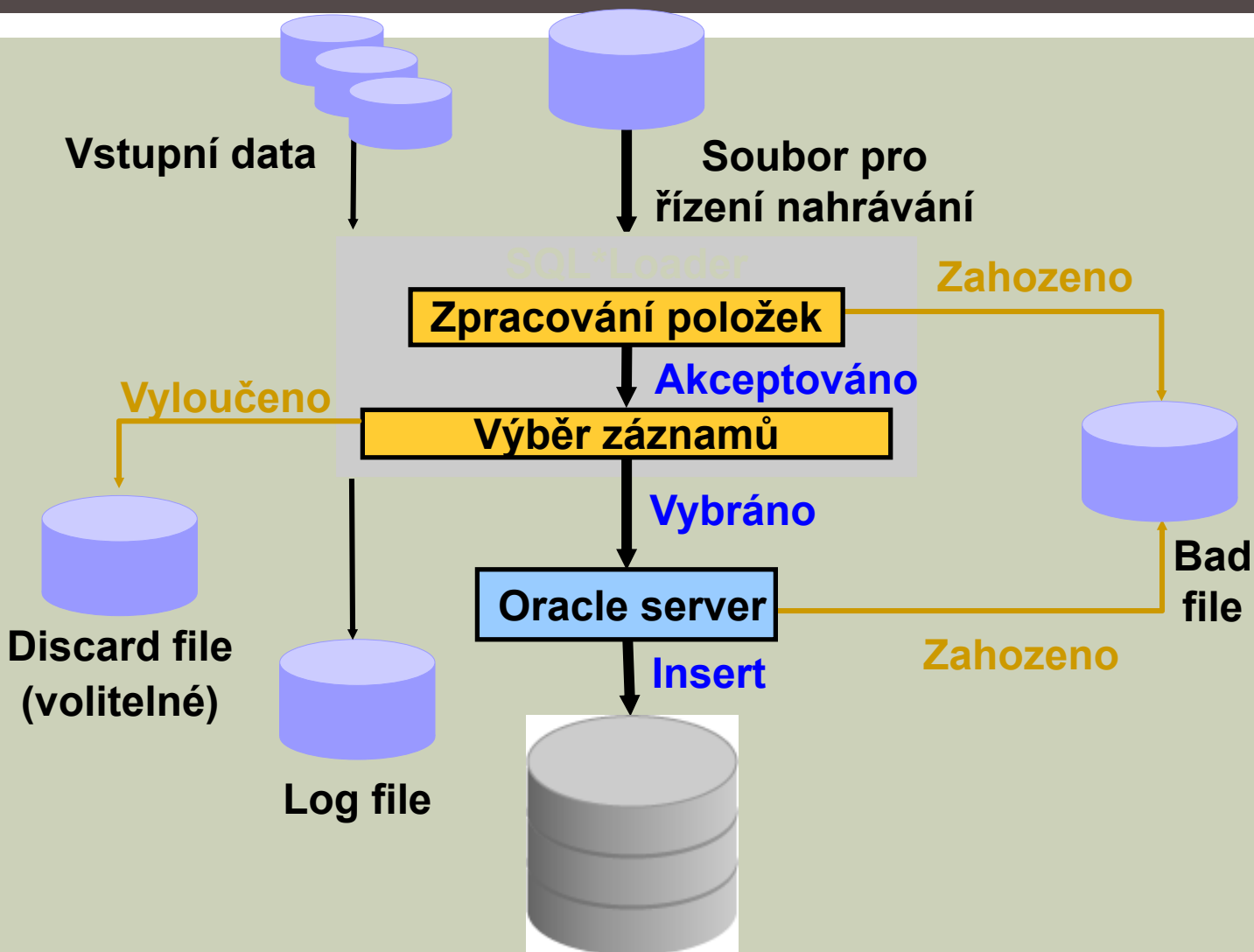
```
SYS
```

```
DATA_PUMP_DIR
```

```
/home/oracle/admin/orcl/dpdump/
```

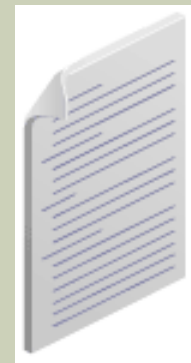
```
SQL>
```

# SQL\*LOADER: PŘEHLED



# SQL\*LOADER ŘÍDÍCÍ SOUBOR

- SQL\*Loader řídicí soubor dává instrukce SQL\*Loaderu o:
  - Umístění souboru s nahrávanými daty
  - Formátu dat
  - Detailech konfigurace:
    - Správa paměti
    - Odmítnutí záznamů
    - Co dělat v případě přerušení nahrávání
  - Detailech manipulace s daty



# SQL\*LOADER ŘÍDÍCÍ SOUBOR

```
load data
infile 'example.dat'
into table example
fields terminated by ','
optionally enclosed by '"'
(col1 char(5),
 col2 char(7))
```

```
example.dat:
hello,cd - OK
010world,im - KO
012my,name is, - KO
```

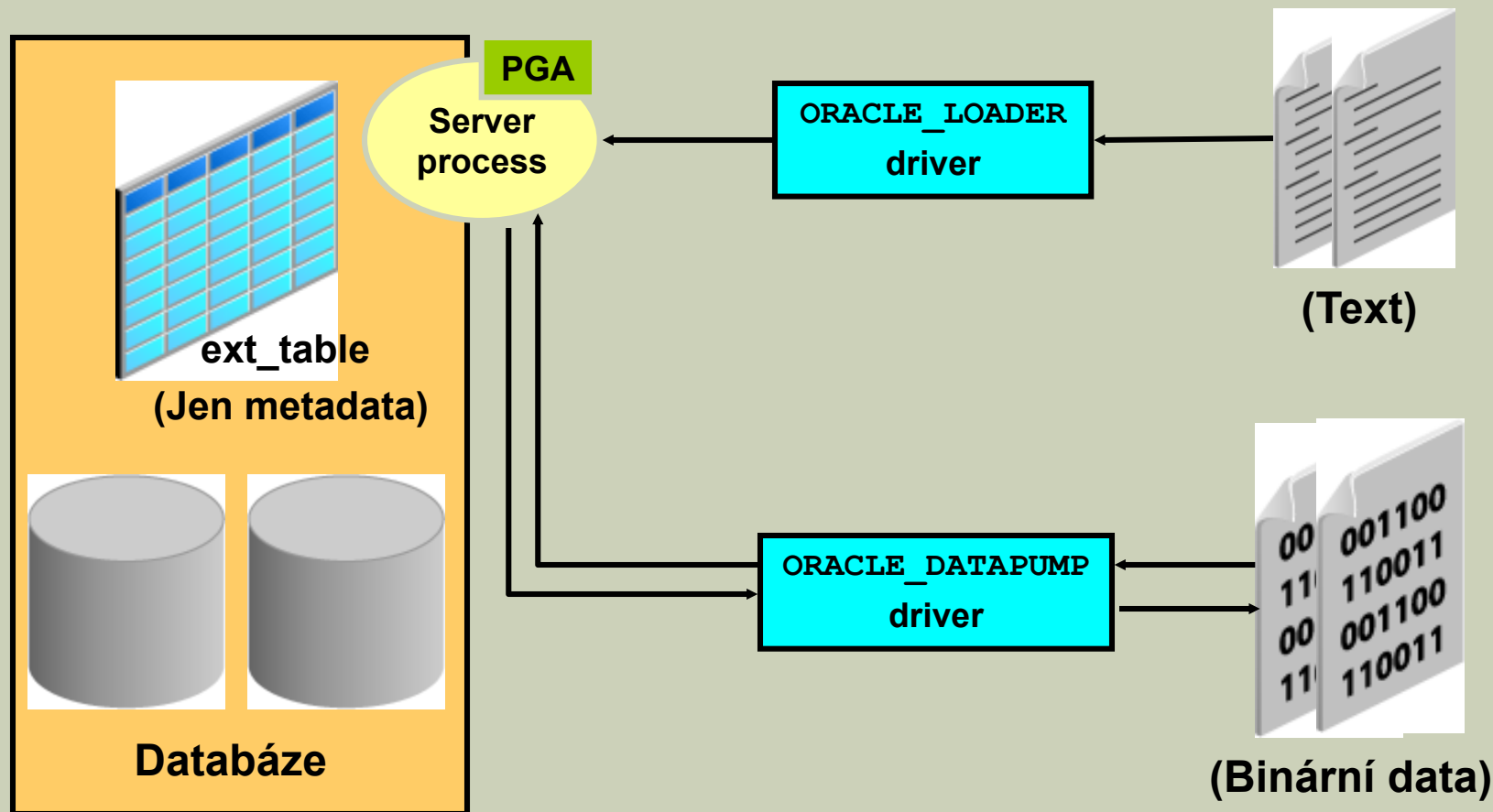


# METODY NAHRÁVÁNÍ DAT

Standardní nahrávání	Direct Path
Používá <code>COMMIT</code>	Ukládá data přímo do souboru (rychlejší)
Generuje transakční žurnál	Negeneruje transakční žurnál (většinou)
Hlídá všechny constrianty	Hlídá jen <code>PRIMARY KEY</code> , <code>UNIQUE</code> , a <code>NOT NULL</code>
Spouští <code>INSERT</code> trigger	Nespustí <code>INSERT</code> trigger
Zvládá i klastrované tabulky	Nezvládá klastrované tabulky
Uživatelé mohou modifikovat data v tabulkách, do kterých nahráváme	Neumožňuje jiným uživatelům změny v tabulkách, do kterých nahráváme data
Po každém insertu aktualizuje indexy	Index aktualizuje až na konci nahrávání

# EXTERNÍ TABULKY

- Externí tabulky jsou read-only tabulky uložené ve formě souborů mimo databázi Oracle



# VÝHODY EXTERNÍCH TABULEK

- Data mohou být použita přímo ze souboru
- Můžeme se na ně dotazovat i spojovat je s daty ze standardních tabulek
- Můžeme kombinovat data z různých externích tabulek



# EXTERNÍ TABULKY ORACLE\_LOADER

```
CREATE TABLE extab_employees
    (employee_id      NUMBER(4) ,
     first_name        VARCHAR2(20) ,
     last_name         VARCHAR2(25) ,
     hire_date         DATE)

ORGANIZATION EXTERNAL
( TYPE ORACLE_LOADER DEFAULT DIRECTORY extab_dat_dir
  ACCESS PARAMETERS
    ( records delimited by newline
      badfile extab_bad_dir:'empxt%a_%p.bad'
      logfile extab_log_dir:'empxt%a_%p.log'
      fields terminated by ',' missing field values are null
      (employee_id, first_name, last_name,
       hire_date char date_format date mask "dd-mon-yyyy")
    )
    LOCATION ('empxt1.csv', 'empxt2.csv')
)
PARALLEL REJECT LIMIT UNLIMITED;
```

# EXTERNÍ TABULKY ORACLE\_LOADER – JSON

```
CREATE TABLE json_file_contents (json_document BLOB)
  ORGANIZATION EXTERNAL (TYPE ORACLE_LOADER DEFAULT
  DIRECTORY order_entry_dir
  ACCESS PARAMETERS
    (RECORDS DELIMITED BY 0x'0A'
    DISABLE_DIRECTORY_LINK_CHECK
    FIELDS (json_document CHAR(5000)))
  LOCATION (order_entry_dir:'PurchaseOrders.json'))
  PARALLEL
  REJECT LIMIT UNLIMITED;
```

# EXTERNÍ TABULKY ORACLE\_LOADER – XML

```
CREATE TABLE tab_ext (  
  doc1 VARCHAR2(4000)  
)  
ORGANIZATION EXTERNAL (  
  TYPE oracle_loader  
  DEFAULT DIRECTORY tmp_dir1  
  ACCESS PARAMETERS (  
    RECORDS  
    XMLTAG ("employee")  
    FIELDS NOTRIM  
    MISSING FIELD VALUES ARE NULL  
  )  
  LOCATION ('test.xml')  
)  
REJECT LIMIT UNLIMITED;
```

```
<employees>  
  <employee><employee_number>7369</employee_number><employee_name>SMITH</employee_name><job>CLERK</job></employee>  
  <employee><employee_number>7499</employee_number><employee_name>ALLEN</employee_name><job>SALESMAN</job></employee>  
  <employee><employee_number>7521</employee_number><employee_name>WARD</employee_name><job>SALESMAN</job></employee>  
</employees>
```

# EXTERNÍ TABULKY ORACLE\_DATAPUMP

```
CREATE TABLE ext_emp_query_results
  (first_name, last_name, department_name)
ORGANIZATION EXTERNAL
  (
    TYPE ORACLE_DATAPUMP
    DEFAULT DIRECTORY ext_dir
    LOCATION ('emp1.exp', 'emp2.exp', 'emp3.exp')
  )
PARALLEL
AS
SELECT e.first_name, e.last_name, d.department_name
FROM   employees e, departments d
WHERE  e.department_id = d.department_id AND
       d.department_name in
           ('Marketing', 'Purchasing');
```

# INLINE EXTERNAL TABLES

## ■ Od 18c

```
SELECT * FROM      EXTERNAL (
    (time_id          DATE NOT NULL,
     prod_id          INTEGER NOT NULL,
     quantity_sold    NUMBER(10,2),
     amount_sold      NUMBER(10,2))
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY data_dir1
  ACCESS PARAMETERS (
      RECORDS DELIMITED BY NEWLINE
      FIELDS TERMINATED BY '|' )
  LOCATION ('sales_9.csv') REJECT LIMIT UNLIMITED)
sales_external;
```

# POUŽITÍ EXTERNÍCH TABULEK

- Dotazy na externími tabulkami:

```
SQL> SELECT * FROM extab_employees;
```

- Dotazy a spojování externích tabulek se standardními tabulkami

```
SQL> SELECT e.employee_id, e.first_name, e.last_name,  
d.department_name FROM departments d, extab_employees e  
WHERE d.department_id = e.department_id;
```

- Přidání dat z externí tabulky do standardní

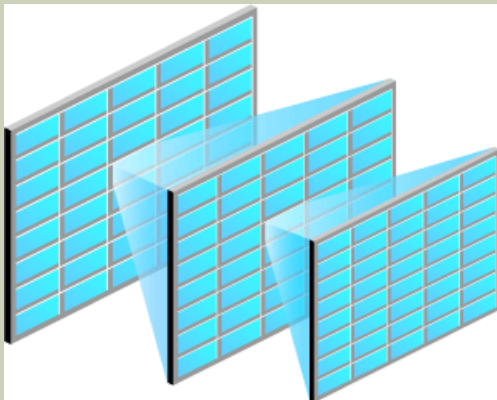
```
SQL> INSERT /*+ APPEND */ INTO hr.employees SELECT * FROM  
extab_employees;
```

- Hint APPEND používáme pouze pro externí tabulky, zamezuje použití buffer cache

# DATA DICTIONARY

## ■ Informace o externích tabulkách najdeme v:

- [DBA | ALL | USER]\_EXTERNAL\_TABLES
- [DBA | ALL | USER]\_EXTERNAL\_LOCATIONS
- [DBA | ALL | USER]\_TABLES
- [DBA | ALL | USER]\_TAB\_COLUMNS
- [DBA | ALL]\_DIRECTORIES



**DOTAZY?**