Integritní omezení (IO)

IO jsou tvrzení vymezující korektnost DB, stupeň souladu datového obrazu s předlohou (jaká data v databází mohou být a jaká již ne).

definují se na konceptuální i databázové úrovni

- KINO(NÁZEV_K, ADRESA),
- FILM(JMÉNO_F, HEREC, ROK)
- PROGRAM(NÁZEV_K, JMÉNO_F, DATUM).
- IO1: Primární klíče. Např.: Jeden film nemohou v jednom kině dávat vícekrát
- IO2: V kinech se nehraje více něž dvakrát týdně.
- IO3: Jeden film se nedává více než ve třech kinech.
- IO4 - IOn: Identifikátory (klíče – primární a unikátní), vazby (cizí klíče)
- ...

Implementace IO

- deklarativní
- procedurální na straně serveru
- procedurální na straně klienta

Deklarativní IO v RDBMS

- diskuse bude používat ukázky z Oracle
- úvod:
- - systémový katalog
- - základy DDL SQL

Systémový katalog

- Prostor v databázi, který obsahuje metainformace, tedy seznam tabulek, pohledů, jejich sloupců, integritních omezení, atd. V relační databázi je organizován jako množina tabulek, nad nimiž je pro běžné uživatele vytvořena soustava pohledů.

Systémový katalog

Příklad v databázi ORACLE

▪Prefix v názvu	▯ rozsah metainformací
▪USERS_	▪informace pouze o objektech v uživatelově schématu (co uživatel vlastní), např. USER_TABLES
▪ALL_	▪jako předešlé, navíc informace o objektech uživateli zpřístupní, např. ALL_TABLES
▪DBA_	▯ informace o všech objektech v databázi, např. ALL_TABLES

Systémový katalog

```
SELECT TABLE_NAME
FROM USER_TABLES
```

```
TABLE_NAME
------------------------------
COUNTRIES
CTENAR
CUSTOMER
DEPT
EMP
...
EXEMPLAR
60 rows selected
```

Systémový katalog

```
SELECT COLUMN_NAME,
  CASE WHEN DATA_TYPE='NUMBER' THEN
        DATA_TYPE||'('||DATA_PRECISION||','||DATA_SCALE||')'
       WHEN DATA_TYPE LIKE '%CHAR%' THEN
        DATA_TYPE||'('||DATA_LENGTH||')'
       ELSE DATA_TYPE
  END AS TYPE, NULLABLE
FROM USER_TAB_COLUMNS WHERE TABLE_NAME='EMP';
```

```
COLUMN_NAME  TYPE                 NULLABLE
-----------  -------------------- --------
EMPNO        NUMBER(4,0)          N
ENAME        VARCHAR2(10)         Y
JOB          VARCHAR2(9)          Y
MGR          NUMBER(4,0)          Y
HIREDATE     DATE                 Y
SAL          NUMBER(7,2)          Y
COMM         NUMBER(7,2)          Y
DEPTNO       NUMBER(2,0)          N
```

DDL SQL - základy

- Data Definition Languge
- umožňuje vytvářet, měnit a rušit objekty v DB
- příkazy
    - CREATE
    - DROP
    - ALTER
    - TRUNCATE
    - RENAME
    - COMENT

Objekty databáze

Objekt	Popis
Tabulka	**Základní jednotka pro ukládání dat;** **skládá se z řádků a sloupců**
Pohled	**Logicky reprezentuje podmnožiny dat** **z jedné nebo více tabulek**
Sekvence	**Generuje hodnoty primárních klíčů**
Index	**Zvyšuje výkon některých dotazů**
Synonymum	**Dává objektům alternativní názvy**

Příkaz CREATE TABLE

- Musíte mít:
  - právo CREATE TABLE
  - oblast pro uložení

```
CREATE TABLE [schéma.]tabulka
            (sloupec datový_typ [DEFAULT výraz];
```

- Určujete:
  - název tabulky
  - název sloupce, jeho datový typ a šířku

Odkazy na tabulky jiných uživatelů

- **schéma -** logický prostor, do kterého se umísťují DB objekty (v Oracle je identický s uživatelským jménem).
- Jako prefix tabulky je třeba použít jméno vlastníka (schématu).
- SELECT * FROM **USER1.EMP**
- implicitně se předpokládá umístění objektu v schématu přihlášeného uživatele

Vytváření tabulek

– Vytvoření tabulky

```
SQL> CREATE TABLE dept
  2         (deptno  NUMBER(2),
  3          dname   VARCHAR2(14),
  4          loc     VARCHAR2(13));
Table created.
```

- Kontrola vytvoření tabulky

```
SQL> DESCRIBE dept
```

```
Name                              Null?    Type
--------------------------------- -------- ---------
DEPTNO                            NOT NULL NUMBER(2)
DNAME                                      VARCHAR2(14)
LOC                                        VARCHAR2(13)
```

Vytvoření tabulky s použitím poddotazu

- Vytvoření tabulky a vložení řádků kombinací příkazu CREATE TABLE a volby AS *poddotaz*

```
CREATE TABLE tabulka
             [sloupec(, sloupec...)]
AS poddotaz;
```

- Počet určených sloupců musí odpovídat počtu sloupců poddotazu.
- Definujte sloupce pomocí názvů a implicitních hodnot.
- Nepřebírá integritní omezení, pouze datové typy

Vytvoření tabulky s použitím poddotazu

```
SQL> CREATE TABLE   dept30
  2         AS
  3          SELECT     empno, ename, sal*12 ANNSAL, hiredate
  4          FROM       emp
  5          WHERE      deptno = 30;
Table created.
```

```
SQL> DESCRIBE dept30
```

```
Name                              Null?    Type
--------------------------------- -------- -----
EMPNO                             NOT NULL NUMBER(4)
ENAME                                      VARCHAR2(10)
ANNSAL                                     NUMBER
HIREDATE                                   DATE
```

Dotazy v datovém slovníku

- Výpis tabulek, které uživatel vlastní

```
SQL> SELECT *
  2     FROM    user_tables;
```

- - Zobrazení různých typů objektů, které uživatel vlastní

```
SQL> SELECT   DISTINCT object_type
  2      FROM   user_objects;
```

- - Zobrazení tabulek, pohledů, synonym a sekvencí, které uživatel vlastní

```
SQL> SELECT   *
  2      FROM    user_catalog;
```

Datové typy

Datový typ	Popis
VARCHAR2(*délka*)	Znaková data proměnné délky
CHAR(*délka*)	Znaková data pevné délky
NUMBER(*p*,*s*)	Číselná data proměnné délky
DATE	Hodnoty data a času
LONG	Znaková data proměnné délky do 2 gigabytů
CLOB	Jednobytová znaková data do 4 gigabytů
RAW a LONG RAW	Nezpracovaná binární data
BLOB	Binární data do 4 gigabytů
BFILE	Binární data uložená v externím souboru; až 4 gigabyty

Klauzule DEFAULT

- Slouží pro určení implicitní hodnoty sloupce při vkládání.

```
… hiredate DATE DEFAULT SYSDATE, …
```

- **Jako hodnotu lze použít literál, výraz nebo funkci SQL.**
- **Nelze použít název jiného sloupce ani pseudosloupec.**
- **Datový typ implicitní hodnoty musí odpovídat datovému typu sloupce.**

Příkaz ALTER TABLE

- Příkaz ALTER TABLE slouží pro:
  - Přidání nového sloupce
  - Změnu existujícího sloupce
  - Definici implicitní hodnoty nového sloupce

```
ALTER TABLE tabulka
ADD             (sloupec datový_typ [DEFAULT výraz]
                [, sloupec datový_typ]...);
```

```
ALTER TABLE tabulka
MODIFY       (sloupec datový_typ [DEFAULT výraz]
              [, sloupec datový_typ]...);;
```

Přidání sloupce

- Pro přidávání sloupců slouží klauzule ADD.

```
SQL> ALTER TABLE dept30
  2  ADD        (job VARCHAR2(9));
Table altered.
```

- Nový sloupec se stane posledním sloupcem.

```
    EMPNO ENAME          ANNSAL HIREDATE  JOB
--------- ---------- --------- --------- ----
     7698 BLAKE           34200 01-MAY-81
     7654 MARTIN          15000 28-SEP-81
     7499 ALLEN           19200 20-FEB-81
     7844 TURNER          18000 08-SEP-81
...
6 rows selected.
```

Úpravy sloupce

- Můžete změnit datový typ sloupce, jeho šířku a implicitní hodnotu.

```
ALTER TABLE   dept30
MODIFY        (ename VARCHAR2(15));
Table altered.
```

- Změna implicitní hodnoty ovlivní pouze následná vkládání do tabulky.

Odstranění tabulky

- Všechna data v tabulce jsou odstraněna.
- Všechny rozpracované transakce jsou potvrzeny.
- Všechny indexy jsou odstraněny.
- Příkaz *nelze* vrátit zpět.

```
SQL> DROP TABLE dept30;
Table dropped.
```

Změna názvu objektu

- Pro změnu názvu tabulky, pohledu, sekvence nebo synonyma slouží příkaz RENAME.

```
SQL> RENAME dept TO department;
Table renamed.
```

Odstranění všech řádků z tabulky

- Příkaz TRUNCATE TABLE:
  - odstraní všechny řádky z tabulky
  - uvolní místo, které tabulka zaujímala

```
SQL> TRUNCATE TABLE department;
Table truncated.
```

- Při použití příkazu TRUNCATE nelze odstranění řádků vrátit zpět
- Odstranit řádky můžete též příkazem DELETE

SQL DDL pro tabulky - shrnutí

Příkaz	Popis
CREATE TABLE	Vytvoření tabulky
ALTER TABLE	Změna struktury tabulky
DROP TABLE	Odstranění řádků a struktury tabulky
RENAME	Změna názvu tabulky, pohledu, sekvence nebo synonyma
TRUNCATE	Odstranění všech řádků z tabulky a uvolnění místa, které zaujímala
COMMENT	Přidání poznámky k tabulce nebo pohledu

Co jsou deklarativní IO?

- Omezení definované na úrovni objektu (tabulky).
- Deklarativní IO typu cizí klíč zabraňují odstranění tabulky, pokud existují závislosti.
- V systému Oracle (a ve stanadardu SQL92) jsou zavedený tyto IO:
  - NOT NULL
  - UNIQUE Key
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK

Pravidla tvorbu deklarativních IO

- Dejte omezení název, jinak Oracle Server vytvoří název ve formátu SYS_C*n*.
- Vytvářejte omezení:
  - Současně s vytvořením tabulky
  - Po vytvoření tabulky
- Definujte omezení na úrovni sloupce nebo tabulky.
- Omezení si lze zobrazit v datovém slovníku.

Zavedení IO - příklad

```
CREATE TABLE [schéma.]tabulka
            (sloupec datový_typ [DEFAULT výraz]
             [omezení_sloupce],
             …
             [omezení_tabulky]);
```

```
CREATE TABLE emp
            (empno  NUMBER(4),
             ename  VARCHAR2(10),
             …
             deptno  NUMBER(7,2) NOT NULL,
             CONSTRAINT emp_empno_pk
                          PRIMARY KEY (EMPNO));
```

Deklarativní IO

- Omezení na úrovni sloupce

```
sloupec [CONSTRAINT název_omezení] typ_omezení,
```

- Omezení na úrovni tabulky

```
sloupec,...
  [CONSTRAINT název_omezení] typ_omezení
  (sloupec, ...),
```

Omezení NOT NULL

- Zajišťuje, že ve sloupci nejsou povoleny prázdné hodnoty

EMP

EMPNO	ENAME	JOB	...	COMM	DEPTNO
7839	KING	PRESIDENT			10
7698	BLAKE	MANAGER			30
7782	CLARK	MANAGER			10
7566	JONES	MANAGER			20
...					

Omezení NOT NULL (žádný řádek nesmí v tomto sloupci obsahovat prázdnou hodnotu)

Bez omezení NOT NULL (každý řádek může mít v tomto sloupci prázdnou hodnotu)

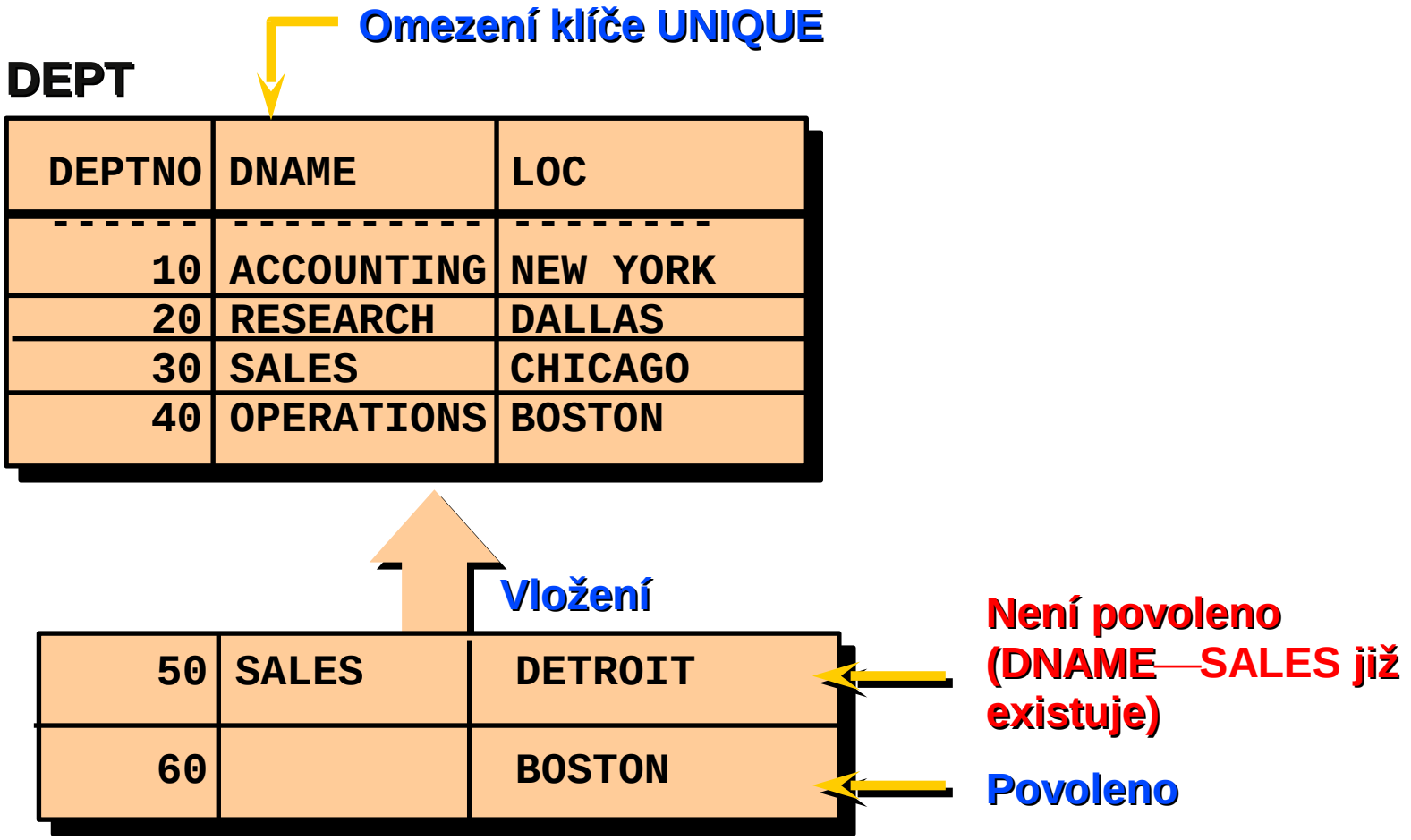
Omezení NOT NULL

NOT NULL

- Definované na úrovni sloupce

```
SQL> CREATE TABLE emp(
  2       empno     NUMBER(4),
  3       ename     VARCHAR2(10) NOT NULL,
  4       job       VARCHAR2(9),
  5       mgr       NUMBER(4),
  6       hiredate  DATE,
  7       sal       NUMBER(7,2),
  8       comm      NUMBER(7,2),
  9       deptno    NUMBER(7,2) NOT NULL);
```

UNIQUE

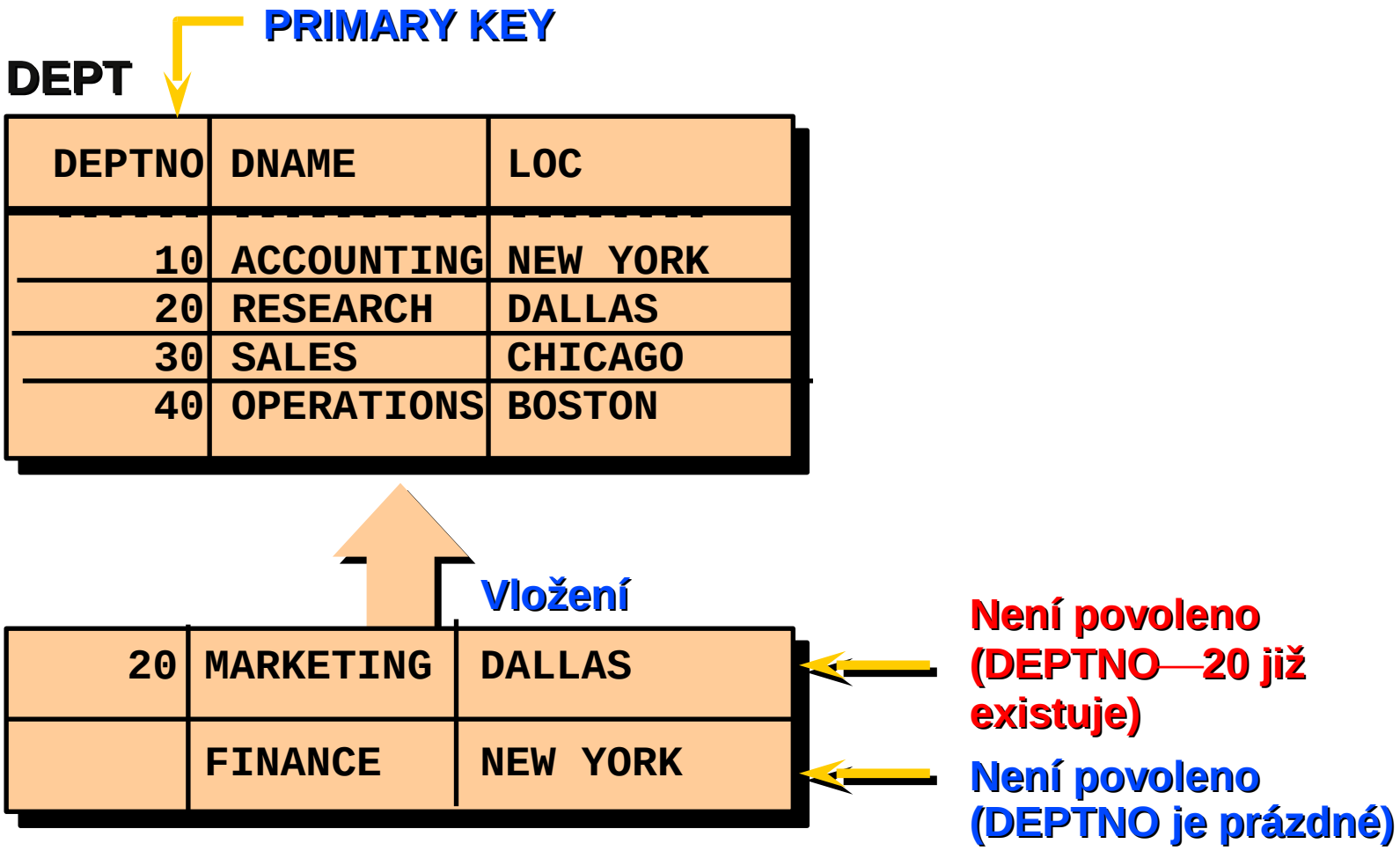

Omezení klíče UNIQUE
DEPT
DEPTNO DNAME LOC
10 ACCOUNTING NEW YORK
20 RESEARCH DALLAS
30 SALES CHICAGO
40 OPERATIONS BOSTON
Vložení
50 SALES DETROIT
60 BOSTON
Není povoleno (DNAME—SALES již existuje)
Povoleno

UNIQUE

- Definované na úrovni tabulky nebo sloupce

```
SQL> CREATE TABLE dept(
  2       deptno    NUMBER(2),
  3       dname     VARCHAR2(14),
  4       loc       VARCHAR2(13),
  5       CONSTRAINT dept_dname_uk UNIQUE(dname));
```

PRIMARY KEY

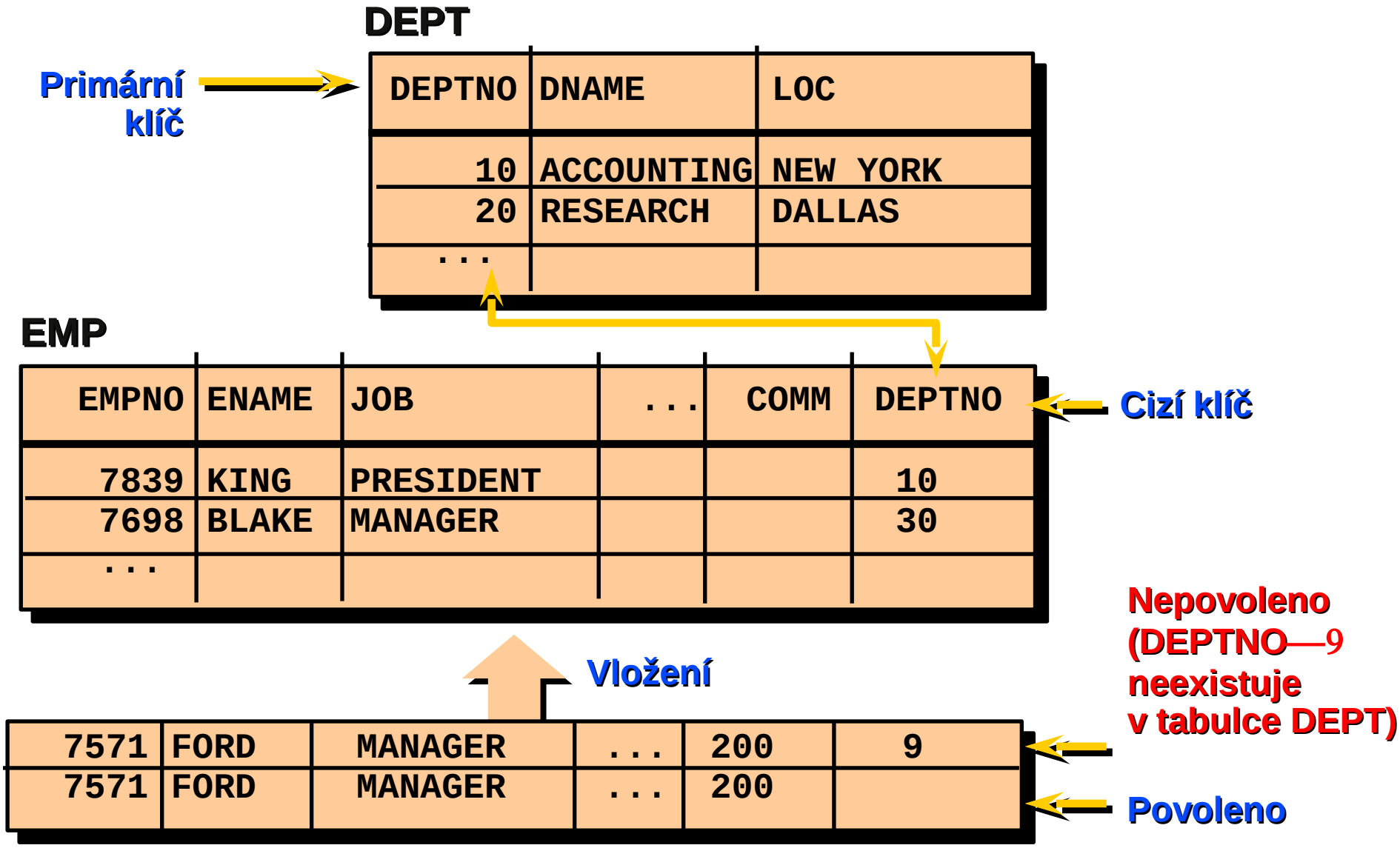

PRIMARY KEY
DEPT
DEPTNO DNAME LOC
10 ACCOUNTING NEW YORK
20 RESEARCH DALLAS
30 SALES CHICAGO
40 OPERATIONS BOSTON
Vložení
20 MARKETING DALLAS
FINANCE NEW YORK
Není povoleno (DEPTNO—20 již existuje)
Není povoleno (DEPTNO je prázdné)

PRIMARY KEY

- Definované na úrovni tabulky nebo sloupce

```
SQL> CREATE TABLE dept(
  2       deptno   NUMBER(2),
  3       dname    VARCHAR2(14),
  4       loc      VARCHAR2(13),
  5       CONSTRAINT dept_dname_uk UNIQUE(dname),
  6       CONSTRAINT dept_deptno_pk PRIMARY KEY(deptno));
```

FOREIGN KEY


DEPT
Primární klíč
DEPTNO DNAME LOC
10 ACCOUNTING NEW YORK
20 RESEARCH DALLAS
...
EMP
EMPNO ENAME JOB ... COMM DEPTNO
Cizí klíč
7839 KING PRESIDENT 10
7698 BLAKE MANAGER 30
...
Vložení
7571 FORD MANAGER ... 200 9
7571 FORD MANAGER ... 200
Nepovoleno (DEPTNO—9 neexistuje v tabulce DEPT)
Povoleno

FOREIGN KEY

- Definované na úrovni tabulky nebo sloupce

```
SQL> CREATE TABLE emp(
  2       empno    NUMBER(4),
  3       ename    VARCHAR2(10) NOT NULL,
  4       job      VARCHAR2(9),
  5       mgr      NUMBER(4),
  6       hiredate DATE,
  7       sal      NUMBER(7,2),
  8       comm     NUMBER(7,2),
  9       deptno   NUMBER(7,2) NOT NULL,
 10       CONSTRAINT emp_deptno_fk FOREIGN KEY (deptno)
 11                 REFERENCES dept (deptno));
```

Klíčová slova klauzule FOREIGN KEY

- FOREIGN KEY
  - definuje sloupec v podřízené tabulce na úrovni omezení tabulky
- REFERENCES
  - identifikuje tabulku a sloupec v rodičovské tabulce
- ON DELETE CASCADE
  - umožňuje odstraňování v rodičovské tabulce odstraněním závislých řádků v podřízené tabulce

CHECK

- Definuje podmínku, kterou musí splňovat každý řádek
- Výrazy, které nejsou povoleny:
  - Odkazy na pseudosloupce CURRVAL, NEXTVAL, LEVEL a ROWNUM
  - Volání funkcí SYSDATE, UID, USER a USERENV
  - Dotazy, které se odkazují na jiné hodnoty v jiných řádcích

```
..., deptno   NUMBER(2),
      CONSTRAINT emp_deptno_ck
            CHECK (DEPTNO BETWEEN 10 AND 99),...
```

Dodatečné přidání dekl. IO

```
ALTER TABLE tabulka
ADD [CONSTRAINT omezení] typ (sloupec);
```

- Omezení lze přidávat a odstraňovat, nikoli upravovat
- Aktivace a deaktivace omezení
- Přidání omezení NOT NULL pomocí klauzule MODIFY

Dodatečné přidání dekl. IO

- Přidejte do tabulky EMP omezení FOREIGN KEY, které zajistí, že manažer již musí existovat jako platný zaměstnanec v tabulce EMP.

```
SQL> ALTER TABLE    emp
  2  ADD CONSTRAINT emp_mgr_fk
  3           FOREIGN KEY(mgr) REFERENCES emp(empno);
Table altered.
```

Odstranění dekl. IO

- Odstraňte omezení pro manažera z tabulky EMP.

```
SQL> ALTER TABLE           emp
  2  DROP CONSTRAINT       emp_mgr_fk;
Table altered.
```

- Odstraňte omezení PRIMARY KEY z tabulky DEPT a odpovídající omezení FOREIGN KEY pro sloupec EMP.DEPTNO.

```
SQL> ALTER TABLE dept
  2  DROP PRIMARY KEY CASCADE;
Table altered.
```

Deaktivace dekl. IO

- Použijte klauzuli DISABLE příkazu ALTER TABLE pro deaktivaci integritního omezení.
- Použijte volbu CASCADE pro deaktivaci závislých integritních omezení.

```
SQL> ALTER TABLE            emp
  2  DISABLE CONSTRAINT     emp_empno_pk CASCADE;
Table altered.
```

Aktivace dekl. IO

- Pro deaktivaci integritního omezení v definici tabulky, které je právě deaktivované, použijte klauzuli ENABLE.

```
SQL> ALTER TABLE            emp
  2  ENABLE CONSTRAINT      emp_empno_pk;
Table altered.
```

-

- Pokud aktivujete omezení UNIQUE nebo PRIMARY KEY, je automaticky vytvořen index UNIQUE nebo PRIMARY KEY.

Zobrazení dekl. IO v dat. slov.

- Pro zobrazení všech definic a názvů omezení použijte dotaz v tabulce USER_CONSTRAINTS.

```
SQL>  SELECT  constraint_name, constraint_type,
  2           search_condition
  3   FROM    user_constraints
  4   WHERE   table_name = 'EMP';
```

```
CONSTRAINT_NAME           C SEARCH_CONDITION
------------------------- - ------------------------
SYS_C00674                C EMPNO IS NOT NULL
SYS_C00675                C DEPTNO IS NOT NULL
EMP_EMPNO_PK              P
...
```

Zobrazení sloupců sdružených s dekl. IO (složená IO)

- Pro zobrazení sloupců sdružených s názvy omezení použijte pohled USER_CONS_COLUMNS

```
SQL> SELECT   constraint_name, column_name
  2  FROM     user_cons_columns
  3  WHERE    table_name = 'EMP';
```

```
CONSTRAINT_NAME              COLUMN_NAME
---------------------------- --------------------
EMP_DEPTNO_FK                DEPTNO
EMP_EMPNO_PK                 EMPNO
EMP_MGR_FK                   MGR
SYS_C00674                   EMPNO
SYS_C00675                   DEPTNO
```

Deklarativní IO - shrnutí

- Existují následující typy omezení:
  - NOT NULL
  - UNIQUE Key
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK
- Pro zobrazení všech definic a názvů dotazů použijte dotaz v tabulce USER_CONSTRAINTS.