Cleverlance

Vytváření a úpravy tabulek

Cíl lekce

- Po skončení této lekce byste měli umět:
- Popsat hlavní databázové objekty
- Vytvářet tabulky
- Popsat datové typy, které mohou být použity při specifikování sloupců
- Měnit definici tabulek
- Rušit, přejmenovávat a vyprazdňovat tabulky

Databázové objekty

Objekt	Popis
Tabulka	Základní ukládací jednotka; skládá se z řádků a sloupců
Pohled	Logicky uspořádaný výběr dat z jedné nebo více tabulek
Sekvence	Numerický generátor hodnot
Index	Zvyšuje výkon při vytváření dotazů
Synonymum	Poskytuje alternativní jména objektům

Pravidla pro názvy

- Názvy tabulek a sloupců musí
  - Začínat písmenem
  - Smějí být dlouhé 1 – 30 znaků
  - Smějí obsahovat pouze A-Z, a-z, 0-9, _, $ a #
  - Nesmějí opakovat názvy jiných objektů vlastněných týmž uživatelem
  - Nesmějí to být vyhrazená slova serveru

Příkaz CREATE TABLE

- Musíte mít
  - Privilegium CREATE TABLE
  - Prostor pro uložení

```
CREATE TABLE [schema.]tabulka
        (sloupec dat_typ [DEFAULT výraz][, ...]);
```

- Specifikujete:
  - Název tabulky
  - Název sloupce, jeho datový typ a velikost

Odkazy na tabulky jiného uživatele

- Tabulky vlastněné jiným uživatelem nejsou součástí uživatelova schématu
- Můžete použít uživatelovo jméno jako prefix k názvu tabulky

Volba DEFAULT

- Specifikujte defaultní hodnotu pro sloupec během vkládání

```
... hire_date DATE DEFAULT SYSDATE, ...
```

- Platnými hodnotami jsou literál, výraz nebo SQL funkce
- Název jiného sloupce nebo pseudosloupce jsou nepřípustné
- Datový typ defaultní hodnoty musí souhlasit s datovým typem sloupce

Vytváření tabulky

- Vytvořte tabulku

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

```
DESCRIBE dept
```

Name	Null?	Type
DEPTNO		NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

Tabulky v databázi

- Uživatelské tabulky:
  - Je to kolekce tabulek vytvořených a vlastněných uživatelem
  - Obsahují uživatelovy informace
- Datový slovník:
  - Je to kolekce tabulek vytvořených a vlastněných databázovým serverem
  - Obsahují data o struktuře a stavu databáze

Dotazy do datového slovníku

Vyber názvy tabulek uživatelem vlastněných

```
SELECT table_name
FROM     user_tables ;
```

Vyber různé typy objektů uživatelem vlastněných

```
SELECT DISTINCT object_type
FROM     user_objects ;
```

Vyber tabulky, pohledy, synonyma a sekvence vlastněné uživatelem

```
SELECT *
FROM     user_catalog ;
```

Datové typy

Datový typ	Popis
VARCHAR2(*size*)	Znaková data proměnné délky
CHAR(*size*)	Znaková data pevné délky
NUMBER(*p,s*)	Numerická data proměnné délky
DATE	Hodnoty datumu a času
LONG	Znaková data proměnné délky až do 2 gigabyte
CLOB	Znaková data až do 4 gigabyte
RAW and LONG RAW	Nezpracovaná binární data
BLOB	Binární data až do 4 gigabyte
BFILE	Binární data uložená v externím souboru; až do 4 gigabyte
ROWID	Systémové číslo představující unikátní adresu řádku v tabulce.

Datové typy pro datum a čas

- Oracle 9i přinesl nové datové typy pro čas a datum
- Jsou k dispozici nové způsoby ukládání dat
- Byla rozšířena vazba na časové zóny a jejich lokalizaci

Datový typ	Popis
`TIMESTAMP`	Datum se zlomkem sekund
`INTERVAL YEAR TO MONTH`	Uložen jako interval roků a měsíců
`INTERVAL DAY TO SECOND`	Uložen jako interval dnů a hodin, minut a sekund

Datové typy pro datum a čas

- Základním typem je typ DATE
- Datový typ TIMESTAMP je rozšířením datového typu DATE
- Ukládá roky, měsíce a dny obdobně jako typ DATE, plus hodiny, minuty a sekundy a stejně zlomky sekund
- Datový typ TIMESTAMP je definován:

```
TIMESTAMP[(přesnost_vteřinového_zlomku)]
```

Vytváření tabulky s užitím poddotazu

- Vytvořte tabulku a vložte do ní řádky kombinací příkazu CREATE TABLE a volby AS *poddotaz*

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

- Počet specifikovaných sloupců musí souhlasit s počtem sloupců v poddotazu
- Definujte sloupce s názvy sloupců a defaultními hodnotami

Vytváření tabulky s užitím poddotazu

```
CREATE TABLE  oddel80
  AS
    SELECT   zamestnanec_id, prijmeni,
             plat*12 ROK_PLAT,
             den_nastupu
    FROM     zamestnanci
    WHERE    oddeleni_id = 80;
Table created.
```

```
 DESCRIBE oddel80
```

Název	Nezadáno?	Typ
ZAMESTNANEC_ID	NOT NULL	NUMBER(6)
PRIJMENI	NOT NULL	VARCHAR2(25)
ROK_PLAT		NUMBER
DEN_NASTUPU	NOT NULL	DATE

Příkaz ALTER TABLE

- Příkaz ALTER TABLE umožňuje
  - Přidat nový sloupec
  - Modifikovat stávající sloupec
  - Definovat defaultní hodnotu pro sloupec
  - Zrušit sloupec

Příkaz ALTER TABLE

- Užijte příkaz ALTER TABLE pro přidání, změnu a zrušení 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]...);
```

```
ALTER TABLE tabulka
DROP          (sloupec);
```

Přidání sloupce

ODDEL80

ZAMESTNANEC_ID	PRIJMENI	ROK_PLAT	DEN_NAST
134	Raus	274800	24.04.97
139	Škodová	142800	01.07.00
111	Paroubek	142800	01.05.04
125	Tihoun	144000	02.01.04

PRACE_ID

Přidej nový sloupec do tabulky ODDEL80

ODDEL80

ZAMESTNANEC_ID	PRIJMENI	ROK_PLAT	DEN_NAST	PRACE_ID
134	Raus	274800	24.04.97	
139	Škodová	142800	01.07.00	
111	Paroubek	142800	01.05.04	
125	Tihoun	144000	02.01.04	

Přidání sloupce

- Přidejte nový sloupec PRÁCE_ID

```
  ALTER TABLE oddel80
  ADD         (prace_id VARCHAR2(9));
Table altered.
```

- Nový sloupec se objeví jako poslední

ZAMESTNANEC_ID	PRIJMENI	ROK_PLAT	DEN_NAST	PRACE_ID
134	Raus	274800	24.04.97	
139	Škodová	142800	01.07.00	
111	Paroubek	142800	01.05.04	
125	Tihoun	144000	02.01.04	

Změna sloupce

- U sloupce můžete změnit datový typ, velikost a defaultní hodnotu

```
ALTER TABLE   oddel80
MODIFY     (prijmeni VARCHAR2(30));
Table altered.
```

- Změna defaultní hodnoty má vliv pouze při pozdějším vkládání dat

Zrušení sloupce

- Použijte klauzuli DROP COLUMN pro odstranění nepotřebného sloupce

```
ALTER TABLE  oddel80
DROP COLUMN  prace_id;
Table altered.
```

Odstranění tabulky

- Všechna data a struktura tabulky jsou zrušena
- Všechny probíhající transakce jsou ukončeny s potvrzením
- Všechny indexy jsou zrušeny
- Tento příkaz *nemůže* být zrušen (vrácen)

```
DROP TABLE dept80;
Table dropped.
```

Změna názvu objektu

- Ke změně názvu tabulky, pohledu, sekvence nebo synonyma použijte příkaz RENAME

```
RENAME oddeleni TO cast_oddeleni;
Table renamed.
```

- Musíte být vlastníkem objektu

Vyprázdnění tabulky

- Příkaz TRUNCATE TABLE
  - Odstraní všechny řádky z tabulky
  - Uvolní skladový prostor obsazený tabulkou

```
  TRUNCATE TABLE cast_oddeleni;
Table truncated.
```

- Jestliže použijete TRUNCATE, nelze obsah tabulky vrátit
- Alternativně můžete vyprázdnit tabulku příkazem DELETE

Co je omezení?

- Omezení (constraints) vynucují pravidla na úrovni tabulek
- Omezení zabraňují vymazání dat, na které jsou vazby z jiných tabulek
- Následující typy omezení jsou platná:
  - `NOT NULL`
  - `UNIQUE`
  - `PRIMARY KEY`
  - `FOREIGN KEY`
  - `CHECK`

Pravidla pro omezení

- Constraint může být pojmenován nebo je jeho jméno vytvářeno serverem
- Vytvořit omezení lze:
  - Při vytváření tabulky
  - Dodatečně, po vytvoření tabulky
- Omezení lze definovat na úrovni sloupce nebo tabulky
- Omezení lze prohlížet v datovém slovníku

Vytváření omezení

```
CREATE TABLE [schema.]tabulka
       (sloupec datatype [DEFAULT výraz]
       [omezení_sloupce],
       ...
       [omezení_tabulky][,...]);
```

```
CREATE TABLE zamestanci (
     zamestnanec_id  NUMBER(6),
     jmeno           VARCHAR2(20),
     ...
     funkce_id       VARCHAR2(10) NOT NULL,
  CONSTRAINT ZAM_ID_PK
     PRIMARY KEY ( ZAMESTNANEC_ID ) );
```

Vytváření omezení

- Omezení na úrovni sloupců

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

- Omezení na úrovni tabulky

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

Omezení NOT NULL

- Zajišťuje, že do sloupce nemůže být vložena hodnota NULL

ZAMESTNANEC_ID	JMENO	PRIJMENI	EMAIL	DEN_NAST	ODDELENI_ID
146	Karla	Boučková	KBOUCKOV	01.03.96	150
147	Lenka	Suková	LSUKOVA	01.10.01	150
148	Veronika	Rohanová	VROHANOV	01.03.01	150
102	Karel	Velebný	KVELEBNY	12.03.98	70
103	František	Mařena	FMARENA	03.06.91	110
107	Zdeněk	Bajaja	ZBAJAJA	02.01.92	20
116	David	Getzler	DGETZLER	01.02.02	140
151	Dan	Wiener	DWIENER	03.01.04	

NOT NULL constraint (žádný řádek nesmí být nulový)

NOT NULL constraint

Není NOT NULL constraint

Omezení NOT NULL

- Definuje se na úrovni sloupce

```
CREATE TABLE zamestnanci(
    zamestnanec_id NUMBER(6),
    prijmeni       VARCHAR2(25) NOT NULL,
    plat           NUMBER(8,2),
    premie_pct     NUMBER(2,2),
    den_nastupu    DATE
                   CONSTRAINT zam_nast_dat_nn
                   NOT NULL,
...
```

Systémem pojmenován

Uživatelem pojmenován

Omezení UNIQUE

- Zabezpečuje, že v určitém sloupci budou pouze jedinečné hodnoty (bez opakování)

ID	JMENO	PRIJMENI	EMAIL
144	František	Půlpán	FPULPAN
145	Walter	Fritz	WFRITZ
146	Karla	Boučková	KBOUCKOV
147	Lenka	Suková	LSUKOVA
148	Veronika	Rohanová	VROHANOV
149	Ludvík	Danihelka	LDANIHEL
150	Jiří	Hlávka	JHLAVKA
151	Dan	Wiener	DWIENER
152	Jan	Vančura	JVANCURA
153	Jiří	Vančura	JVANCURA

UNIQUE

LZE

NELZE

Omezení UNIQUE

- Lze jej definovat na úrovni řádku i tabulky

```
CREATE TABLE zamestnanci(
    zamestnanec_id    NUMBER(6),
    prijmeni          VARCHAR2(25)
NOT NULL,
    email             VARCHAR2(25),
    palt              NUMBER(8,2),
    premie_pct        NUMBER(2,2),
    den_nastupu       DATE NOT NULL,
...
    CONSTRAINT zam_email_uk
UNIQUE(email));
```

Omezení PRIMARY KEY

- Omezení PRIMARY KEY je sloupec, který jednojednoznačně identifikuje určitý řádek v tabulce

Omezení PRIMARY KEY

- Lze definovat jak na úrovni tabulky, tak řádku

```
CREATE TABLE   oddeleni(
    oddeleni_id           NUMBER(4),
    oddeleni_nazev        VARCHAR2(30)
      CONSTRAINT odd_nazev_nn NOT NULL,
    manager_id            NUMBER(6),
    lokalita_id           NUMBER(4),
     CONSTRAINT odd_id_pk PRIMARY KEY(oddeleni_id));
```

Omezení FOREIGN KEY

ODDELENI_ID	ODDELENI_NAZEV	MANAGER_ID	LOKALITA_ID
10	Ředitelství	100	1
20	Zásobování	113	1
30	Účtárna	108	1
40	Oblastní závod Kladno	117	2
50	Oblastní závod Brno	128	3
60	Sklad Líheznice	104	4

Cizí klíč

JMENO	PRIJMENI	ODDELENI_ID
Karla	Boučková	150
Lenka	Suková	150
Veronika	Rohanová	150

INSERT

Dan	Wiener	60
Jan	Vančura	9

Přípustné

Nelze – klíč neexistuje

FOREIGN KEY constraint

- Lze definovat na úrovni tabulky i sloupce

```
CREATE TABLE zamestnanci(
    zamestnanec_id    NUMBER(6),
    prijmeni          VARCHAR2(25)
NOT NULL,
    email             VARCHAR2(25),
    plat              NUMBER(8,2),
    premie_pct        NUMBER(2,2),
    den_nastupu       DATE NOT NULL,
...
    oddeleni_id       NUMBER(4),
    CONSTRAINT zam_odd_fk FOREIGN
KEY (oddeleni_id)
      REFERENCES
oddeleni(oddeleni_id),
    CONSTRAINT zam_email_uk
UNIQUE(email));
```

FOREIGN KEY constraint

Klíčová slova

- FOREIGN KEY: identifikuje sloupec v dceřiné tabulce, který odpovídá vzoru
- REFERENCES: identifikuje otcovskou tabulku a sloupec
- ON DELETE CASCADE: při zrušení řádku v otcovské tabulce zruší závislé řádky v dceřiné tabulce
- ON DELETE SET NULL: při zrušení řádku v otcovské tabulce nastaví závislé cizí klíče na NULL

Omezení CHECK

- Definuje podmínky, které musí každý řádek splňovat
- Následující výrazy jsou nepřípustné
  - Odkazy na pseudosloupce CURRVAL, NEXTVAL, LEVEL a ROWNUM
  - Užití funkcí SYSDATE, UID, USER a USERENV
  - Dotazy, které používají hodnoty v jiných řádcích

```
..., plat NUMBER(2)
     CONSTRAINT zam_plat_min
            CHECK (plat > 0),...
```

Přidávání omezení

- Použijte příkaz ALTER TABLE
  - Přidat nebo odstranit omezení beze změny struktury
  - Povolit nebo zakázat omezení
  - Přidat NOT NULL omezení pomocí MODIFY klauzule

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

Přidání omezení

- Přidejte omezení FOREIGN KEY, který zajistí, že manažer zaměstnance již musí existovat jako platný zaměstnanec v tabulce ZAMESTNANCI

```
ALTER TABLE     zamestnanci
ADD CONSTRAINT  zam_manager_fk
  FOREIGN KEY(manager_id)
  REFERENCES zamestnanci(employee_id);
Table altered.
```

Odstranění omezení

- Odstraňte manažerské omezení z tabulky ZAMESTANCI

```
ALTER TABLE       zamestnanci
DROP CONSTRAINT   zam_manager_fk;
Table altered.
```

- Odstraňte PRIMARY KEY na tabulce ODDELENI a odstraňte související omezení FOREIGN KEY na sloupci ZAMESTNANCI.ODDELENI_ID

```
ALTER TABLE   departments
DROP PRIMARY KEY CASCADE;
Table altered.
```

Deaktivace omezení

- Použijte DISABLE klauzuli příkazu ALTER TABLE k deaktivaci integritního omezení
- Aplikujte volbu CASCADE k deaktivaci souvisejících integritních omezení

```
ALTER TABLE      zamestnanci
DISABLE CONSTRAINT  zam_zam_id_pk CASCADE;
Table altered.
```

Aktivace omezení

- Aktivujte integritní omezení v tabulce deaktivované užitím klauzule ENABLE

```
ALTER TABLE      employees
ENABLE CONSTRAINT   emp_emp_id_pk;
Table altered.
```

- UNIQUE nebo PRIMARY KEY indexy se automaticky vytvoří při aktivaci omezení UNIQUE nebo PRIMARY KEY

Prohlížení omezení

- Dotaz do tabulky USER_CONSTRAINTS ukáže všechny definice a jména omezení

```
SELECT constraint_name, constraint_type,
    search_condition
FROM   user_constraints
WHERE  table_name = 'ZAMESTNANCI';
```

CONSTRAINT_NAME	C	SEARCH_CONDITION
ZAM_ID_NN	C	"ZAMESTNANEC_ID" IS NOT NULL
ZAM_PRIJMENI_NN	C	"PRIJMENI" IS NOT NULL
ZAM_DEN_NASTUP_NN	C	"DEN_NASTUPU" IS NOT NULL
ZAM_ODDELENI_FK	R	
ZAM_PLAT_MIN	C	PLAT > 0
ZAM_ID_PK	P	
ZAM_EMAIL_UK	U	
ZAM_MANAGER_FK	R	

Prohlížení sloupců spojených s omezeními

- Pro prohlížení sloupců spojených se jménem omezení použijeme pohled USER CONS COLUMNS

```
SELECT constraint_name, column_name
FROM   user_cons_columns
WHERE  table_name = 'ZAMESTNANCI';
```

CONSTRAINT_NAME	COLUMN_NAME
ZAM_DEN_NASTUP_NN	DEN_NASTUPU
ZAM_EMAIL_UK	EMAIL
ZAM_ID_NN	ZAMESTNANEC_ID
ZAM_ID_PK	ZAMESTNANEC_ID
ZAM_MANAGER_FK	MANAGER_ID
ZAM_ODDELENI_FK	ODDELENI_ID
ZAM_PLAT_MIN	PLAT
ZAM_PRIJMENI_NN	PRIJMENI