Databázové systémy

4. dátový slovník, sekvencie, vnútorné spájanie tabuliek





Opakovanie

- CREATE TABLE vytvorenie tabuľky
- ALTER TABLE
- DROP TABLE
- INSERT
- UPDATE
- DELETE
- SELECT / WHERE / ORDER BY



Úvod do pohľadov

- sú to virtuálne objekty, podmnožina údajov z jednej alebo viacerých tabuliek
- v podstate len príkaz select uložený v DBS
- nezaberá miesto
- štandardne ho nie je možné modifikovať
 CREATE VIEW <názov pohľadu> AS
 SELECT <stĺpce> FROM <názov tabuľky2> [WHERE <podmienky>];



Dátový slovník

- systémový katalóg (system catalog) pre databázu Oracle sa nazýva Dátový slovník = Data Dictionary
 - centralizované úložisko obsahujúce informácie o štruktúre a objektoch databázy
 - údaje uložené v systémovom katalógu sú meta údaje (metadata), teda údaje o údajoch
 - zoznamy tabuliek, stĺpcov, užívateľov, privilégií, štatistické údaje, údaje o výkone DB



Pohľady dátového slovníka

- dynamické v\$
 - údaje súvisiace s výkonom inštancie
- statické
 - USER_ zobrazuje informácie o privátnom prostredí daného užívateľa, teda o jeho objektoch, privilégiách, ktoré pridelil atď.
 - ALL_ zobrazuje informácie o objektoch pripojeného užívateľa, o objektoch, na ktoré mu boli pridelené práva a o objektoch, ktoré sú prístupné verejne (cez užívateľa PUBLIC)
 - DBA_ zobrazuje informácie o všetkých objektoch databázy
 - DBA_pohľady sú sprístupnené iba užívateľovi SYS(vlastník) alebo tým užívateľom, ktorý majú pridelenú rolu SELECT_CATALOG_ROLE alebo privilégium SELECT ANY DICTIONARY



Pohľady dátového slovníka

- user_objects, all_objects, dba_objects
 - user_objects zobrazí iba objekty, ktoré vytvoril užívateľ, pod ktorým ste prihlasení
 - all_objects zobrazí objekty, ktoré vlastníte, ale aj tie, na ktoré máte práva napr. HR.EMPLOYEES
 - dba_objects obsahuje zoznam všetkých objektov databázy
- user_tables, all_tables, dba_tables
- user_views, all_views, dba_views
- dba_users

SELECT * FROM dictionary WHERE table_name LIKE '%ALL%COLUMNS%';

- pomôcka, ak neviete aký presne pohľad hľadáte
- zobrazí názvy všetkých tabuliek a pohľadov z dátového slovníka, ktoré majú v názve slová ALL a COLUMNS (v takomto poradí)
- Poznámka: údaje v dátovom slovníku sú uložené veľkými písmenami



Sekvencie

- objekt používaný na generovanie postupnosti čísel
- často používaný na vytvorenie unikátnych hodnôt primárneho kľúča
- postupnosť je generovaná nezávisle od tabuľky, môže byť využívaná pri vkladaní do viacerých tabuliek
- niektoré hodnoty postupnosti môžu byť "preskočené" ->vygenerované a použité v transakcii, ktorá nebola potvrdená (ukončená príkazom rollback)
- hodnota vygenerovaná sekvenciou je iba číslo (už nie je závislá od sekvencie a naopak)



Sekvencie

```
CREATE SEQUENCE <názov sekvencie>
        [MINVALUE <hodnota>]
        [MAXVALUE <hodnota>]
        [START WITH <hodnota>]
        [INCREMENT BY <hodnota>]
        [CYCLE/NOCYCLE]
        [CACHCHE <hodnota>/NOCACHE];
Príklad:
CREATE SEQUENCE seq zamestanec
        START WITH 1000
        INCREMENT BY 1
        NOCYCLE
        CACHE 10;
```

 Vytvorili sme neohraničenú postupnosť (nezadali sme nimimálnu ani maximálnu hodnotu), ktorá začína číslom 1000, bude sa zväčšovať o 1, nie je cyklická (teda hodnoty sa nebudú opakovať) a vždy sa vygeneruje 10 hodnôt tejto postupnosti (postupne sa budú používať a keď sa minú vygenerujú sa ďalšie)



Práca so sekvenciami

- <meno sekvencie>.CURRVAL
 - vráti aktuálnu (poslednú vygenerovanú) hodnotu sekvencie v danej session (v aktuálnom prihlasení)
 - SELECT seq_zamestnanec.CURRVAL FROM dual;
- <meno sekvencie>.NEXTVAL
 - vygeneruje a zobrazí ďalšiu hodnotu
 - INSERT INTO zamestnanec VALUES(seq_zamestnanec.NEXTVAL, 'Janko', 'Mrkvička', 1234568888, 'Košice');



Práca so sekvenciami

```
SQL> CREATE SEQUENCE seg zamestanec START
WITH 1000 INCREMENT BY 1 NOCYCLE CACHE 10;
Sequence created.
SQL> select seq_zamestanec.currval from
dual;
select seq zamestanec.currval from dual
ERROR at line 1:
ORA-08002: sequence SEQ ZAMESTANEC.CURRVAL
is not yet defined in this session
- funkcia currval vráti poslednú hodnotu, ktorá bola
vygenerovaná v aktuálnej session (v tvojom prihlasení
) t.j. zobrazí hodnotu až po zavolaní-vygenerovaní
hodnoty funkciou nextval
SQL> select seg zamestanec.nextval from
dual;
   NEXTVAL
      1000
```

```
SQL> select seg zamestanec.currval from dual;
   CURRVAL
      1000
SQL> select seq zamestanec.nextval from dual;
   NEXTVAL
      1001
SQL> select seg zamestanec.nextval from dual;
   NEXTVAL
      1002
SQL> select seg zamestanec.currval from dual;
   CURRVAL
     1002
```



Práca so sekvenciami

zmena vlastností sekvencie

ALTER SEQUENCE seq_zamestanec INCREMENT BY -2 MINVALUE 0;

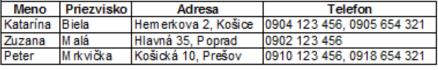
zmazanie sekvencie

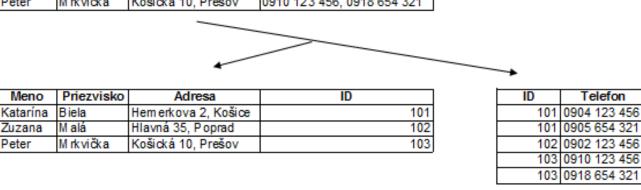
DROP SEQUENCE seq_zamestanec;



Spájanie tabuliek

• prečo je potrebné?





 - jedným z procesov, ktorým si pri vytváraní prejdeme je normalizácia. Počas tohto procesu často rozdeľujeme tabuľky na menšie, aby boli dodržané všetky vlastnosti, ktoré databáza má mať. Ale v reálnom svete tieto vlastnosti vystupujú spolu a preto aj my ich potrebujeme vybrať z databázy spolu a na to nám slúži spájanie tabuliek



Spájanie tabuliek

- dotaz, ktorý kombinuje riadky z dvoch alebo viacerých tabuliek pohľadov alebo materializovaných pohľadov
- DB vykonáva spojenie, ak v klauzule FROM nájde dve alebo viac tabuliek
- spojenie prebieha pomocou rovnakých hodnôt v stĺpci
- pri rovnakých názvoch stĺpcov sa musí použiť kvalifikácia stĺpcov <meno tabuľky>.<meno stĺpca>; môžeme definovať prefixy (platí len v danom príkaze SELECT)
- pri spájaní troch a viacerých tabuliek systém najprv spojí prvé dve a následne výsledok spojí s ďalšou tabuľkou resp. ďalšími tabuľkami v poradí
- typy spájania:
 - karteziánsky súčin (CROSS JOIN)
 - vnútorné spojenie (INNER JOIN)
 - vonkajšie spojenie (LEFT/RIGHT/FULL OUTER JOIN)
 - Prirodzené spojenie (NATURAL JOIN)



Karteziánsky súčin

 kombinuje každý riadok z 1.tabuľky s každým riadkom z 2.tabuľky

 $\{(a,b), a \in A, b \in B\}$

 výsledný počet riadkov je daný ako súčin počtu riadkov 1.tabuľky s počtom riadkov 2.tabuľky

$$|\{(a,b), a \in A, b \in B\}| = |A|.|B|$$

syntax:

```
SELECT * FROM <tabuľka1> CROSS JOIN
<tabuľka2>;
```



Karteziánsky súčin

SQL> SELECT e.employee_id, e.first_name, e.last_name,
 d.department_name FROM employees e CROSS JOIN departments
 d ORDER BY 1;

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
100 100 100	Steven Steven Steven Steven Steven	King King King King King King	Administration Marketing Purchasing Human Resources Shipping
100	Steven Steven Steven	King King King	IT Public Relations Sales
206 206 206 206	William William William William William William William	Gietz Gietz Gietz Gietz Gietz Gietz	Human Resources Government Sales Contracting IT Helpdesk Sales Finance

2889 rows selected.

INNER JOIN – vnútorné spojenie



- najčastejšie používaný typ spojenia
- spojí riadky dvoch tabuliek, ktoré majú rovnakú hodnotu kľúča – často spája cudzí a primárny kľúč
- výsledkom sú riadky, ktoré spĺňajú spájaciu podmienku
- výsledok z pohľadu množín: PRIENIK
- syntax:

```
SELECT * FROM <tabulka1> INNER JOIN
<tabulka2> ON podmienka;
```

INNER JOIN – vnútorné spojenie



- zoznam zamestnancov a oddelení, v ktorých pracujú
- zobrazí iba tie riadky z tabuľky employees, ku ktorým vieme priradiť oddelenie z tabuľky departments t.j. e splnená podmienka

SQL> SELECT e.employee_id, e.first_name, e.last_name,
 d.department_name FROM employees e INNER JOIN departments d
 on e.department id=d.department id ORDER BY 1;

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
101 102 103	Steven Neena Lex Alexander Bruce	King Kochhar De Haan Hunold Ernst	Executive Executive Executive IT IT
204 205	Pat Susan Hermann Shelley William	Fay Mavris Baer Higgins Gietz	Marketing Human Resources Public Relations Accounting Accounting

106 rows selected.