

Databázové systémy

4. dátový slovník, sekvencie,
vnútorné spájanie tabuliek



KKUI
Katedra kybernetiky
a umelej inteligencie

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 prideliť 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 prihlásení
 - 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]  
    [CACHE <hodnota>/NOCACHE];
```

Príklad:

```
CREATE SEQUENCE seq_zamestanec  
    START WITH 1000  
    INCREMENT BY 1  
    NOCYCLE  
    CACHE 10;
```

- Vytvorili sme neohraničenú postupnosť (nezadali sme minimálnu 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 prihlásení)
 - `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 seq_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 prihlásení
) t.j. zobrazí hodnotu až po zavolaní-vygenerovaní
hodnoty funkciou nextval

```
SQL> select seq_zamestanec.nextval from  
dual;
```

```
      NEXTVAL  
-----  
      1000
```

```
SQL> select seq_zamestanec.currval from dual;
```

```
      CURRVAL  
-----  
      1000
```

```
SQL> select seq_zamestanec.nextval from dual;
```

```
      NEXTVAL  
-----  
      1001
```

```
SQL> select seq_zamestanec.nextval from dual;
```

```
      NEXTVAL  
-----  
      1002
```

```
SQL> select seq_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?

Meno	Priezvisko	Adresa	Telefon
Katarína	Biela	Hemerkova 2, Košice	0904 123 456, 0905 654 321
Zuzana	Malá	Hlavná 35, Poprad	0902 123 456
Peter	Mrkvička	Košická 10, Prešov	0910 123 456, 0918 654 321



Meno	Priezvisko	Adresa	ID
Katarína	Biela	Hemerkova 2, Košice	101
Zuzana	Malá	Hlavná 35, Poprad	102
Peter	Mrkvička	Košická 10, Prešov	103

ID	Telefon
101	0904 123 456
101	0905 654 321
102	0902 123 456
103	0910 123 456
103	0918 654 321

- 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

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

- 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	Steven	King	Administration
100	Steven	King	Marketing
100	Steven	King	Purchasing
100	Steven	King	Human Resources
100	Steven	King	Shipping
100	Steven	King	IT
100	Steven	King	Public Relations
100	Steven	King	Sales
...			
206	William	Gietz	Human Resources
206	William	Gietz	Government Sales
206	William	Gietz	Contracting
206	William	Gietz	IT Helpdesk
206	William	Gietz	Sales
206	William	Gietz	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 <tabuľka1> INNER JOIN  
<tabuľka2> 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
-----	-----	-----	-----
100	Steven	King	Executive
101	Neena	Kochhar	Executive
102	Lex	De Haan	Executive
103	Alexander	Hunold	IT
104	Bruce	Ernst	IT
...			
202	Pat	Fay	Marketing
203	Susan	Mavris	Human Resources
204	Hermann	Baer	Public Relations
205	Shelley	Higgins	Accounting
206	William	Gietz	Accounting

106 rows selected.