

Databázové systémy

5. spájanie tabuliek, referenčná integrita



KKUI
Katedra kybernetiky
a umelej inteligencie

Sekvence

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

```
SELECT seq_zamestnanec.CURRVAL FROM dual;
```

```
INSERT INTO zamestnanec VALUES(  
    seq_zamestnanec.NEXTVAL, 'Janko', 'Mrkvička',  
    1234568888, 'Košice');
```

```
ALTER SEQUENCE seq_zamestanec INCREMENT BY -2 MINVALUE 0;  
DROP SEQUENCE seq_zamestanec;
```

Spájanie tabuliek

- prečo je potrebné?

Meno	Priezvisko	Adresa	Telefon
Katarína	Biela	Hemerikova 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	Hemerikova 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

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

```
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	Contracting
206	William	Gietz	IT Helpdesk
206	William	Gietz	Sales
206	William	Gietz	Finance

2889 rows selected.

INNER JOIN –

Vnútorné spojenie

- zoznam zamestnancov a oddelení, v ktorých pracujú

```
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
105	David	Austin	IT
106	Valli	Pataballa	IT
107	Diana	Lorentz	IT
108	Nancy	Greenberg	Finance
109	Daniel	Faviet	Finance
...			
202	Pat	Fay	Marketing
203	Susan	Mavris	Human Resources
204	Hermann	Baer	Public Relations
205	Shelley	Higgins	Accounting
206	William	Gietz	Accounting

NATURAL JOIN –

Prirodzené spojenie

- realizované na základe všetkých stĺpcov s rovnakým menom v dvoch tabuľkách
- vráti riadky, ktoré majú rovnaké hodnoty v súvisiacich stĺpcoch
- nepoužívaný, náchylný na chyby
- syntax:

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

NATURAL JOIN –

Prirodzené spojenie

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME

101	Neena	Kochhar	Executive
102	Lex	De Haan	Executive
104	Bruce	Ernst	IT
105	David	Austin	IT
106	Valli	Pataballa	IT
...			
155	Oliver	Tuvault	Sales
184	Nandita	Sarchand	Shipping
185	Alexis	Bull	Shipping
186	Julia	Dellinger	Shipping
187	Anthony	Cabrio	Shipping
202	Pat	Fay	Marketing
206	William	Gietz	Accounting

32 rows selected.

aký zoznam sme dostali týmto selektom? Spojenie bolo vykonané na základe spoločných stĺpcov: department_id, manager_id

INNER JOIN USING

- alternatíva pre prirodzené spájanie
- spája záznamy tabuliek na základe rovnakých hodnôt explicitne vymenovaných stĺpcov s rovnakým pomenovaním
- syntax:

```
SELECT * FROM <tabuľka1> INNER JOIN  
<tabuľka1> USING(<meno stĺpca>);
```

Príklad:

```
SELECT * FROM employees e INNER JOIN  
    departments d USING(department_id);  
SELECT * FROM employees e INNER JOIN  
    departments d ON  
    e.department_id=d.department_id;
```

LEFT OUTER JOIN – ľavé vonkajšie spájanie

- vonkajšie spájanie sa používa k výpisu aj tých záznamov, ktoré nespĺňajú spojovacie kritérium
- vo výsledku sa zobrazí celá ľavá tabuľka: záznamy, ktoré sú v prieniku s pravou tabuľkou (spĺňajú spájaciu podmienku) a aj záznamy, ktoré nie sú, chýbajúce záznamy z pravej tabuľky sú nahradené hodnotou NULL

- syntax:

```
SELECT * FROM <tabuľka1> LEFT OUTER JOIN  
<tabuľka1> ON <podmienka>;
```

ekvivalent:

```
SELECT * FROM <tabuľka1> LEFT JOIN <tabuľka1>  
ON <podmienka>;
```

LEFT OUTER JOIN – ľavé vonkajšie spájanie

- zoznam VŠETKÝCH zamestnancov a oddelení, v ktorých pracujú (ak zamestnanec nepatrí pod žiadne oddelenie, chýbajúce stĺpce majú hodnotu NULL)

```
SQL> SELECT e.employee_id, e.first_name, e.last_name,  
d.department_name FROM employees e LEFT OUTER JOIN  
departments d on e.department_id=d.department_id ORDER BY 1;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
-----	-----	-----	-----
206	William	Gietz	Accounting
205	Shelley	Higgins	Accounting
200	Jennifer	Whalen	Administration
102	Lex	De Haan	Executive
100	Steven	King	Executive
101	Neena	Kochhar	Executive
109	Daniel	Faviet	Finance
...			
121	Adam	Fripp	Shipping
120	Matthew	Weiss	Shipping
199	Douglas	Grant	Shipping
198	Donald	OConnell	Shipping
178	Kimberely	Grant	

107 rows selected.

RIGHT OUTER JOIN – pravé vonkajšie spájanie

- vo výsledku sa zobrazí celá pravá tabuľka: záznamy, ktoré sú v prieniku s ľavou tabuľkou (spĺňajú spájaciu podmienku) a aj záznamy, ktoré nie sú, chýbajúce záznamy z ľavej tabuľky sú nahradené hodnotou NULL
- v praxi málo používaný, môže byť nahradený LEFT OUTER JOIN
- syntax:
SELECT * FROM <tabuľka1> RIGHT OUTER JOIN <tabuľka2> ON <podmienka>;

ekvivalent:

SELECT * FROM <tabuľka1> RIGHT JOIN <tabuľka2> ON <podmienka>;

RIGHT OUTER JOIN – pravé vonkajšie spájanie

- zoznam zamestnancov a oddelení, v ktorých pracujú a aj tých oddelení, ktoré nemajú zamestnanca

```
SQL> SELECT e.employee_id, e.first_name, e.last_name,
           d.department_name FROM employees e RIGHT OUTER JOIN
           departments d on e.department_id=d.department_id ORDER BY 4;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME

205	Shelley	Higgins	Accounting
206	William	Gietz	Accounting
200	Jennifer	Whalen	Administration
			Benefits
			Construction
			Contracting
			Control And Credit
			Corporate Tax
102	Lex	De Haan	Executive
100	Steven	King	Executive
101	Neena	Kochhar	Executive
...			
196	Alana	Walsh	Shipping
197	Kevin	Feeney	Shipping
198	Donald	OConnell	Shipping
			Treasury

122 rows selected.

FULL OUTER JOIN –

Úplné vonkajšie spojenie

- zjednotenie výsledkov ľavého aj pravého vonkajšieho spojenia
- zobrazí tie záznamy, ktoré spĺňajú spájaciu podmienku a tie záznamy z oboch podmienok, ktoré ju nespĺnili
- chýbajúce atribúty su doplnené hodnotou NULL
- syntax:

```
SELECT * FROM <tabuľka1> FULL OUTER JOIN  
<tabuľka1> ON <podmienka>;
```

ekvivalent:

```
SELECT * FROM <tabuľka1> FULL JOIN <tabuľka1>  
ON <podmienka>;
```

FULL OUTER JOIN – Úplné vonkajšie spojenie

- zoznam všetkých zamestnancov a oddelení

```
SQL> SELECT e.employee_id, e.first_name, e.last_name,
           d.department_name FROM employees e FULL OUTER 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
105	David	Austin	IT
106	Valli	Pataballa	IT
			Retail Sales
			Shareholder Services
			Benefits
			IT Helpdesk
			Government Sales
			Manufacturing
			NOC
			IT Support

123 rows selected.
2022/2023

Explicitná a implicitná notácia spojenia

- explicitná notácia pomocou JOIN
 - podporuje väčšina databázových systémov
 - ANSI-92 definovaný štandard
 - sprehl'adňuje SELECT
 - odporučaný aj kvôli výkonnosti
- implicitná notácia : spojenie pomocou WHERE klauzuly(uvedená podmienka spájania sa nachádza v tejto klauzule), spájané tabuľky oddelené čiarkou
 - nezdokumentovaný štandard

Explicitná a implicitná notácia spojenia

TYP SPOJENIA	EXPLICITNÁ NOTÁCIA	IMPLICITNÁ NOTÁCIA
CROSS JOIN	SELECT * FROM <tabuľka1> CROSS JOIN <tabuľka2>;	SELECT * FROM <tabuľka1>, <tabuľka2>;
INNER JOIN	SELECT * FROM <tabuľka1> INNER JOIN <tabuľka2> ON podmienka;	SELECT * FROM <tabuľka1> , <tabuľka2> WHERE <tabuľka1>.<kľúčový atribút1>=<tabuľka2>.<kľú čový atribút2>;
LEFT OUTER JOIN	SELECT * FROM <tabuľka1> LEFT OUTER JOIN <tabuľka2> ON podmienka;	SELECT * FROM <tabuľka1> , <tabuľka2> WHERE <tabuľka1>.<kľúčový atribút1>=<tabuľka2>.<kľú čový atribút2>(+) ;
RIGHT OUTER JOIN	SELECT * FROM <tabuľka1> RIGHT OUTER JOIN <tabuľka2> ON podmienka;	SELECT * FROM <tabuľka1> , <tabuľka2> WHERE <tabuľka1>.<kľúčový atribút1>(+)=<tabuľka2>.< kľúčový atribút2>;
FULL OUTER JOIN	SELECT * FROM <tabuľka1> FULL OUTER JOIN <tabuľka2> ON podmienka;	neimplementovany, nahradza sa použitim UNION

WHERE vs ON podmienka

- filtrovať riadky vieme aj v klauzule ON pri spájaní tabuliek
- pozor, nie je to to isté ako filtrovanie riadkou v klauzule WHERE
- klauzula ON
 - definuje podmienku spájania
 - vyhodnocuje sa pred samotným spojením

Príklad:

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name  
FROM employees e RIGHT OUTER JOIN departments d ON  
    e.department_id=d.department_id and e.last_name IS NULL  
ORDER BY 1;
```

- klauzula WHERE
 - definuje podmienku na filtrovanie
 - vyhodnocuje sa po spojení

Príklad:

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name  
FROM employees e RIGHT OUTER JOIN departments d ON  
    e.department_id=d.department_id  
WHERE e.last_name IS NULL ORDER BY 1;
```

Referenčná integrita

- systém pravidiel, ktoré sa využívajú k zabezpečeniu platnosti relácií medzi záznamami viazaných tabuliek
- slúži k zamedzeniu nežiaduceho odstránenia alebo zmeny súvisiacich dát

```
CREATE TABLE objednavka (  
    zakaznicke_cislo_obj NUMBER(15) UNIQUE,  
    objednavka_datum DATE DEFAULT SYSDATE,  
    dodavka_datum DATE NOT NULL,  
    interne_cislo_obj NUMBER(12) PRIMARY KEY  
        CHECK (LENGTH(interne_cislo_obj)=12  
            AND SUBSTR(interne_cislo_obj, 1,4)='OBJ-'),  
    zak_cislo NUMBER(10) CONSTRAINT fk_zak_cislo REFERENCES zakaznik,  
    vybavuje NUMBER(5)  
CONSTRAINT datумы_por  
    CHECK(dodavka_datum > objednavka_datum)  
CONSTRAINT fk_zam_cislo  
    FOREIGN KEY (vypavuje) REFERENCES zamestnanec(ev_cislo));
```

Referenčná integrita

kontroluje vzťah medzi tabuľkami nasledujúcim spôsobom:

- podriadená tabuľka
 - pridávanie záznamov
 - v prepojavacej položke je prípustná iba tá hodnota, ktorá sa vyskytuje v prepojavacej položke nadriadenej tabuľky. Pokus o pridanie záznamu, ktorý nezodpovedá žiadnemu záznamu v nadriadenej tabuľke, vyvolá chybu.
 - zmena záznamov
 - hodnotu prepojavacej položky je možné zmeniť iba na hodnotu, ktorá sa vyskytuje v prepojavacej položke nadriadenej tabuľky. Pokus o zápis hodnoty, ktorá neodpovedá nadriadenej tabuľke, vyvolá chybu.
 - vymazávanie záznamov
 - bez obmedzenia

Referenčná integrita

- nadriadená tabuľka
 - pridávanie záznamov
 - bez obmedzenia.
 - zmena záznamov
 - hodnotu prepojovacej položky je možné zmeniť iba pri tých záznamoch, ktorým nezodpovedá žiaden záznam v podriadenej tabuľke. Hodnotu prepojovacej položky v záznamoch, ktorým zodpovedá aspoň jeden záznam v podriadenej tabuľke, nie je možné meniť.
 - vymazanie záznamov:
 - kaskádny prístup: CASCADE
 - ak je zmazaný riadok v odkazujúcej tabuľke, sú zmazané aj riadky v príslušných odkazovaných tabuľkách
 - reštrikčný prístup: RESTRICT (NO ACTION)
 - záznam nemôže byť zmazaný, ak naň existuje odkaz pomocou cudzieho kľúča
 - default
 - nulitný prístup: SET NULL
 - ak je príslušný záznam v odkazovanej tabuľke odstránený alebo aktualizovaný, cudzí kľúč sa nastaví na NULL

Referenčná integrita

Operácia	podriadená tabuľka	nadriadená tabuľka
INSERT	prípustná iba tá hodnota, ktorá sa vyskytuje v prepojovacej položke nadriadenej tabuľky, inak ORA	bez obmedzenia
UPDATE	iba na hodnotu, ktorá sa vyskytuje v prepojovacej položke nadriadenej tabuľky, inak ORA	hodnotu prepojovacej položky je možné zmeniť iba pri tých záznamoch, ktorým nezodpovedá žiaden záznam v podriadenej tabuľke, inak ORA
DELETE	bez obmedzenia	záznam, ktorému nezodpovedá žiaden záznam v podriadenej tabuľke inak podľa definície cudzieho kľúča