Database Programming with SQL
kurs 2017 – database design and programming with sql
students slajdovi

7-1 Oracle Equijoin and Cartesian Product

- Prethodna sekcija se bavila upitima preko više od jedne tabele u rdb korišćenjem ANSI/ISO SQL: 99 sintakse
- Starija verzija Oracle db je zahtevala udruživanje korišćenjem Oracle Proprietary join sintakse
- Sada se bavimo Oracle Proprietary join sintaksom (pre Oracle 9i) za Equijoins i Cartesian Product i njihovim ANSI/ISO SQL:99 odgovarajućim pandanima

Join Commands

- Postoje dva seta komandi ili sintaksi koje se mogu koristiti za pravljenje veza između tabela u db: Oracle proprietary joins i ANSI/ISO SQL 99 compliant standard joins

Oracle Proprietary Join	ANSI/ISO SQL: 1999 Equivalent
Cartesian Product	Cross Join
Equijoin	NATURAL JOIN JOIN USING clause JOIN ON clause (if the equality operator is used)
Non-equijoin	ON clause

Oracle Proprietary Joins

- Za upite nad više od jedne tabele korišćenjem Oracle proprietary sintakse koristiti join uslov u WHERE izrazu:

```
SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column1 = table2.column2;
```

- Problem se javlja ako postoje dva studenta u istom razredu sa istim prezimenom pa im se obraćaju sa imenom oa prezimenom; isti je razlog stavljati ime tabele ispred imena kolone što se naziva “qualifying your columns“
- Kombinacija imena tabele i imena kolone pomaže eliminisanju ambiguous imena kada dve tabele sadrže kolonu sa istim imenom kolone
- Kada se isto ime kolone pojavi u obe tabele, ime kolone mora biti posle imena tabele
- Za kvalifikovanje kolone koristi se sintaksa tablename.columnname

EQUIJOIN

- Zove se još “simple“ ili “inner“ join i predstavlja udruživanje tabela koje kombinuju redove koji imaju iste vrednosti za specificirane kolone
- Naredba equijoin je ekvivalentna sa ANSI naredbama: NATURAL JOIN, JOIN USING; JOIN ON (kada join uslov koristi =)

```
SELECT employees.last_name, employees.job_id, jobs.job_title
FROM employees, jobs
WHERE employees.job_id = jobs.job_id;
```

What? Where? How?

LAST_NAME	JOB_ID	JOB_TITLE
King	AD_PRES	President
Kochhar	AD_VP	Administration Vice President
De Haan	AD_VP	Administration Vice President
Whalen	AD_ASST	Administration Assistant
Higgins	AC_MGR	Accounting Manager
Gietz	AC_ACCOUNT	Public Accountant
Zlotkey	SA_MAN	Sales Manager
Abel	SA_REP	Sales Representative
...	...	...

- Primer:

```
SELECT employees.last_name, departments.department_name
FROM employees, departments
WHERE employees.department_id = departments.department_id;
```

LAST_NAME	DEPARTMENT_NAME
Whalen	Administration
Hartstein	Marketing
Fay	Marketing
Mourgos	Shipping
Rajs	Shipping
Davies	Shipping
Matos	Shipping
...	...

- Kolone koje se koriste za udruživanje obe tabele ne moraju da budu u SELECT listi kolona

Cartesian Product Join

- Ako dve tabele u join upitu nemaju join uslov specificiran u WHERe izrazu ili join uslov nije dobar, Oracle Server vraća Kartezijan proizvod te dve tabele; to je kombinacija svakog reda jedne tabele sa svakim redom druge tabele
- Kartezijan proizvod je ekvivalentan sa ANSI CROSS JOIN
- Da bi se izbegao Kartezijan proizvod uvek treba uključiti validan join uslov u WHERE izraz
- U sledećem upitu, join uslov nije uključen:

```
SELECT employees.last_name, departments.department_name
FROM employees, departments;
```

LAST_NAME	DEPARTMENT_NAME
Abel	Administration
Davies	Administration
De Haan	Administration
Ernst	Administration
Fay	Administration
Gietz	Administration
Grant	Administration
...	...

160 rows returned in 0.01 seconds

Restricting Rows In a Join

- Kao i sa upitima nad jednom tabelom, WHERE izraz se može koristiti za ograničenje redova uključenih u jednu ili više tabela
- Upit pokazuje korišćenje AND operatora za ograničenje vraćenih redova

```
SELECT employees.last_name, employees.job_id, jobs.job_title
FROM employees, jobs
WHERE employees.job_id = jobs.job_id
  AND employees.department_id = 80;
```

LAST_NAME	JOB_ID	JOB_TITLE
Zlotkey	SA_MAN	Sales Manager
Abel	SA_REP	Sales Representative
Taylor	SA_REP	Sales Representative

Aliases

- Skraćivanje sintakse korišćenjem alijasa; alijasi se koriste za razlikovanje kolona koji imaju ista imena ali su u drugačijim tabelama
- Alijas tabele je sličan alijasu kolone, on reimenuje objekat unutar iskaza
- Alijas se pravi unosom novog imena za tabelu odmah posle iemna tabele unutar from

Table Aliases

```
SELECT last_name, e.job_id, job_title
FROM employees e, jobs j
WHERE e.job_id = j.job_id
  AND department_id = 80;
```

LAST_NAME	JOB_ID	JOB_TITLE
Zlotkey	SA_MAN	Sales Manager
Abel	SA_REP	Sales Representative
Taylor	SA_REP	Sales Representative

- Kada imena kolona nisu duplicirana između dve tabele, ne treba dodati ime tabele ili alijas imenu kolone
- Ako se alijas tabele koristi iz FROM izraza onda taj alijas tabele mora biti zamenjen za ime tabele kroz ceo SELECT iskaz
- Korišćenje imena tabele u SELECT izrazu kojem je dat alijas u FROM izrazu dovešće do greške

```
SELECT last_name, employees.job_id, job_title
FROM employees e, jobs j
WHERE e.job_id = j.job_id
  AND department_id = 80;
```


ORA-00904: "EMPLOYEES"."JOB_ID": invalid identifier

JOIN Syntax Example

- Ako treba udružiti tri tabele koliko joins treba koristiti ?
- Da bi se udružile tri tabele, treba dodati još jedan join uslov u WHERE izraz korišćenjem AND operatora

- Ako treba napraviti izveštaj o zaposlenima i gradu gde su smešteni njihovi sektori, treba udružiti tri tabele: employees, departments i locations

```
SELECT last_name, city
FROM employees e, departments d, locations l
WHERE e.department_id = d.department_id
   AND d.location_id = l.location_id;
```

LAST_NAME	CITY
Hartstein	Toronto
Fay	Toronto
Zlotkey	Oxford
Abel	Oxford
...	...

7-2 Oracle Nonequijoins and Outer Joins

- Šta se dešava ako hoćemo dobiti podatak iz tabele koja nema odgovarajuću kolonu u drugoj tabeli ?
- Npr, procentualna ocena iz matematike je 92 i smeštena je u GRADES koloni u jednoj tabeli; ocena u obliku slova je smeštena u LETTER_GRADE koloni u drugoj tabeli
- Kada je podatak zapisan korišćenjem opsega, dobijanje takvog podatka je posao nonequijoin
- Do sada se nisu uzimali u obzir redovi koji nisu ispunjavali određene uslove ali je ponekad potrebno dobiti podatke iz jedne tabele čak iako nijedan podatak ne odgovara drugoj tabeli

Nonequijoin

- Primer: Želimo znati grade_level za svaku platu zaposlenog; job_grades tabela nema zajedničku (common) kolonu sa employees tabelom; korišćenje nonequijoin omogućava udruživanje dve tabele
- Pošto ne postoji tačno poklapanje između dve kolone u svakoj od tabela, operator jednakosti = se ne može koristiti
- Iako uslovi upoređivanja <= i >= se mogu koristiti, BETWEEN...AND je mnogo efikasniji način izvršavanja nonequijoin
- Nonequijoin je ekvivalentan sa ANSI JOIN ON (gde je korišćeni uslov neki drugi a ne =)

job_grades table

GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

```
SELECT last_name, salary, grade_level, lowest_sal, highest_sal
FROM employees, job_grades
WHERE (salary BETWEEN lowest_sal AND highest_sal);
```

LAST_NAME	SALARY	GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
Vargas	2500	A	1000	2999
Matos	2600	A	1000	2999
Davies	3100	B	3000	5999
Rajs	3500	B	3000	5999
Lorentz	4200	B	3000	5999
Whalen	4400	B	3000	5999
Mourgos	5800	B	3000	5999
Fay	6000	C	6000	9999
...				

Outer Join

- Outer join se koristi da bi se videli redovi koji imaju odgovarajuće vrednosti u drugoj tabeli plus oni redovi u jednoj od tabela koji nemaju odgovarajuće vrednosti u drugoj tabeli
- Za indikaciju koja tabela može imati nedostajuće vrednosti koristi se Oracle Join Syntax, doda se + posle imena kolone u tabeli unutar WHERE izraza upita
- Upit će vratiti sva prezimena zaposlenih, uključujući ona koja su dodeljena sektoru i ona koja nisu
- Isti rezultat se dobija korišćenjem ANSI LEFT OUTER JOIN

```
SELECT e.last_name, d.department_id,
 d.department_name
FROM employees e, departments d
WHERE e.department_id =
    d.department_id(+);
```

LAST_NAME	DEPT_ID	DEPT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
Vargas	50	Shipping
…		
Higgins	110	Accounting
Grant	-	-

- Outer join će vratiti sve department ID i department names, koji imaju zaposlene dodeljene njima i one koji to nemaju
- Isti rezultat se može dobiti korišćenjem ANSI RIGHT OUTER JOIN

```
SELECT e.last_name, d.department_id,
 d.department_name
FROM employees e, departments d
WHERE e.department_id(+) =
    d.department_id;
```

LAST_NAME	DEPT_ID	DEPT_NAME
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Mourgos	50	Shipping
…		
Gietz	110	Accounting
-	190	Contracting

- Nije moguće imati ekvivalent FULL OUTER JOIN dodavanjem + znaka obema kolonama u join uslovu
- Ovakav pokušaj bi doprineo grešci:

```
SELECT e.last_name, d.department_id, d.department_name
FROM employees e, departments d
WHERE e.department_id(+) =  d.department_id(+);
```


ORA-01468: a predicate may reference only one outer-joined table

- Moguće je napraviti full outer join korišćenjem Set operatora
- Varijacije u sintaksi od outer join su :

```
SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column = table2.column(+);
```

```
SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column(+) = table2.column;
```

```
SELECT table1.column, table2.column
FROM table1, table2
NEVER table1.column(+) = table2.column(+);
```

X

Outer Join and ANSI equivalents

ANSI/ISO SQL	Oracle Syntax
`LEFT OUTER JOIN departments d ON (e.department_id = d.department_id);`	`WHERE e.department_id = d.department_id(+);`
`RIGHT OUTER JOIN departments d ON (e.department_id = d.department_id);`	`WHERE e.department_id(+) = d.department_id;`
`FULL OUTER JOIN departments d ON (e.department_id = d.department_id);`	No direct equivalent.