

## 2-1 Columns Characters and Rows

- Concatenation (pridruživanje) stanje povezanosti kao kod lanca; unija u povezanim serijama

### DESCRIBE

- Korišćenje DESCRIBE (DESC) komande za prikaz strukture tabele:  
`DESCRIBE <table_name>;`
- DESC vraća ime tabele, tip podataka, osnovni i strani ključ, kolone sa null vrednostima..

```
DESC departments;
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPARTMENTS	DEPARTMENT_ID	NUMBER	-	4	0	1	-	-	-
	DEPARTMENT_NAME	VARCHAR2	30	-	-	-	-	-	-
	MANAGER_ID	NUMBER	-	6	0	-	-	-	-
	LOCATION_ID	NUMBER	-	4	0	-	-	-	-

- Uopšteno, SQL službene reči se ne mogu pisati skraćeno; DESCRIBE je Oracle Proprietary komanda, pa zato se može skraćeno napisati kao DESC
- Važno je znati kada se ubacuje novi red u tabelu kakvi su tipovi podataka po kolonama i da li se kolona može ostaviti praznom

### Concatenation Operator

- Pridruživanje znači povezivanje ili linkovanje u serijama
- Simbol za pridruživanje su dve verikalne crte (pipes)
- Vrednosti sa obe strane || operatora su kombinovane da bi napravile jednu izlaznu kolonu
- Sintaksa je: `string1 || string2 || string_n`
- Kada su vrednosti pridružene, rezultujuća vrednost je karakter string
- U SQL operator pridruživanja može linkovati kolone sa drugim kolonama, aritmetičkim izrazima ili konstantnim vrednostima za kreiranje karakter izraza
- Operator pridruživanja se koristi za kreiranje čitljivog tekstualnog izlaza
- U primeru, `department_id` je pridruženo sa `department_name`

```
SELECT department_id || department_name
FROM departments;
```

DEPARTMENT_ID    DEPARTMENT_NAME
10Administration
20Marketing
50Shipping
60IT
...

- U ovoj varijaciji prošlog primera, `|| ' ' ||` se koristi za pravljenje praznog mesta između `department_id` i `department_name`
- Karakter 'space' između apostrofa kreira prazno mesto između vrednosti kolona

```
SELECT department_id || ' ' || department_name
FROM departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
50	Shipping
60	IT
...	

### Concatenation and Column Aliases

- Alijasi kolona su korisni pri korišćenju operatora pridruživanja tako da difolt SELECT linija se ne pojavljuje kao naslov kolone the column heading.

```
SELECT department_id || ' ' ||
       department_name AS "Department Info "
FROM departments;
```

Department Info
10 Administration
20 Marketing
50 Shipping
60 IT
...

```
SELECT first_name || ' ' ||
       last_name AS "Employee Name"
FROM employees;
```

Employee Name
Ellen Abel
Curtis Davies
Lex De Haan

### Concatenation and Literal Values

- Literal vrednost je fiksna vrednost podatka kao što su karakteri, brojevi ili datumi ('dollars', 1000, 'January 1, 2009'); korišćenjem pridruživanja i literala može se kreirati izlaz koji izgleda kao iskaz ili rečenica
- Literalne vrednosti se mogu uključiti u SELECT listu sa operatorima pridruživanja
- Karakter i datumi moraju biti unutar seta apostrofa
- Svaki red koji se vrati iz upita sa literalnim vrednostima će imati isti karakter string u njemu
- U sledećem primeru, King zarađuje 24000\$ mesečno
- Stringovi 'has a monthly salary of' i 'dollars' su primeri literala
- SQL iskaz koji proizvodi izlaz u ovom formatu izgleda:

```
SELECT last_name || ' has a monthly
       salary of ' || salary || '
       dollars.' AS Pay
FROM employees;
```

PAY
King has a monthly salary of 24000 dollars.
Kochhar has a monthly salary of 17000 dollars.
De Haan has a monthly salary of 17000 dollars.
Whalen has a monthly salary of 4400 dollars.
Higgins has a monthly salary of 12000 dollars.
Gietz has a monthly salary of 8300 dollars.
-

- Videti da postoji karakter praznog mesta koji se nalazi odmah posle ' na početku literala i pre ' na kraju literala
- U literal se mogu uključiti brojevi kao literalne vrednosti
- U sledećem primeru, broj 1 je pridružen stringu 'has a' i 'year salary of'

```
SELECT last_name || ' has a ' || 1 || ' year salary of ' || salary*12 ||
       ' dollars.' AS Pay
FROM employees;
```

PAY
King has a 1 year salary of 288000 dollars.
Kochhar has a 1 year salary of 204000 dollars.
De Haan has a 1 year salary of 204000 dollars.
Whalen has a 1 year salary of 52800 dollars.
Higgins has a 1 year salary of 144000 dollars.
...

### Using DISTINCT to eliminate Duplicate Rows

- Puno puta, želite znati koliko jedinstvenih instanci nečega postoji
- Npr, šta ako želite listu svih sektora od kojih postoje zaposleni
- Može se napisati upit za selektovanje department\_ids iz employees tabele:

```
SELECT department_id
FROM employees;
```

- Pogledati sve duplicirane redove
- Kako se može modifikovati iskaz za eliminaciju dupliciranih redova?
- Sem ako nije drugačije traženo, izlaz SQL upita će prikazati rezultate bez eliminisanja dupliciranih redova
- U SQL, DISTINCT službena reč se koristi za eleimisanje dupliciranih redova

```
SELECT DISTINCT department_id
FROM employees;
```

- DISTINCT utiče na sve navedene kolone i vraća svaku distinct kombinaciju kolona u SELECT rečenici
- Službena reč DISTINCT se mora pojaviti odmah posle SELECT ključne reči

DEPARTMENT_ID
90
90
90
10
110
110
80
80
80
...

DEPARTMENT_ID
-
90
20
110
80
50
10
60

### EXECUTE, SAVE i EDIT u APEX

- Sada treba i sačuvati iskaze za kasnije da bi se ponovo mogli startovati ili da bi se modifikovali

### 2-2 Limit Rows Selected

- Da bi se mogla ograničiti količina informacija za procesuiranje u jednom trenutku, koristi se komanda WHERE
- Sa WHERE se bira informacija koja se traži iz tabele

### SELECT iskaz

- Sa SELECT se vraća informacija iz db; SELECT iskaz mora uključiti najmanje SELECT i FROM komande, dok je WHERE komanda opciona

```
SELECT*|[DISTINCT] column | expression alias]..}
FROM table
[WHERE condition(s)];
```

- [] znači opciono; | znači ili

## WHERE komanda

- Pri dobijanju podataka iz db, možda vam zatreba limitiranje redova podataka koji su prikazani
- To se može izvesti korišćenjem WHERE naredbe
- WHERE naredba sadrži uslov koji se mora ispuniti a on direktno prati iz FROM rečenice u SQL iskazu
- Sintaksa za WHERE naredbu:

```
WHERE column_name comparison_condition comparison_value
```

- Alijasi se ne smeju koristiti u WHERE rečenici
- Primer iz Employees db:

```
SELECT employee_id, first_name, last_name
FROM employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
100	Steven	King
101	Neena	Kochhar
102	Lex	De Haan

- Dodavanjem WHERE dela, redovi su limitirani na one redove gde vrednost od employee\_id je 101:

```
SELECT employee_id, first_name, last_name
FROM employees
WHERE employee_id = 101;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
101	Neena	Kochhar

## Operatori upoređivanja unutar WHERE komande

- Znak = se može koristiti unutar WHERE rečenice (=, <, >, >=, <=, <> ili != ili ^=)
- U sledećem primeru, department\_id kolona se koristi u WHERE rečenici, sa operatorom upoređivanja =
- Svi zaposleni sa department\_id od 90 se vraćaju

```
SELECT employee_id, last_name, department_id
FROM employees
WHERE department_id = 90;
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90

- Karakter stringovi i datumi u WHERE rečenici moraju biti zatvoreni u apostrofima; ali brojevi ne trebaju biti zatvoreni u apostrofima
- Primer: WHERE rečenica sadrži string i zatvorena je u apostrofima

```
SELECT first_name, last_name
FROM employees
WHERE last_name = 'Taylor';
```

- Šta bi se desilo da je WHERE napisana kao: `WHERE last_name = 'jones';`
- Sve pretrage po karakterima su osetljivi na veličinu slova
- Pošto su sva prezimena sa velikim prvim slovom u tabeli, prikazan primer neće vratiti nikakvu vrednost
- Operatori upoređivanja moraju biti korišćeni na sledeći način:

```
WHERE hire_date < '01-Jan-2000'
```

```
WHERE salary >= 6000
```

```
WHERE job_id = 'IT_PROG'
```

- Poput karakter stringova, Date vrednosti moraju takođe biti zatvorene pod apostrofima
- Dozvoljeni formati za Datume su: '20-MAR-1999', '20/MAR/1999', '20/mar/1999', '20/Mar/1999'
- Koji redovi će biti izabrani sa:

```
SELECT last_name, salary
FROM employees
WHERE salary <= 3000;
```