PostgreSQL

Lesson 3: PostgreSQL - Retrieving data

Lesson Objectives

In this lesson, you will learn about:

- SELECT statement
- Operators in PostgreSQL
- Using WHERE clause
- LIMIT key word
- ORDER BY clause
- GROUP BY clause
- HAVING clause
- DISTINCT key word

SELECT statement

▪PostgreSQL SELECT statement is used to fetch the data from a database table which returns data in the form of result set

```
SELECT * FROM table_name;
```

▪Example:

```
SELECT * FROM employee;
```

▪We can use arithmetic operators in SELECT statement:

```
SELECT name, salary*12 FORM employee;
```

▪Column name can be given alias name:

```
SELECT name, salary*12 annual_sal FORM employee;
```

▪Here “annual_sal” is an alias name

Operators in PostgreSQL

- Operators are used to perform arithmetic operations and comparison operations
- Operators are used to specify conditions and serve as conjunction for multiple conditions in a statement
- Types of operators:
  - Arithmetic

+ - * / % ^(exponentiation) |/ (square root) ||/ (cube root) !/(factorial)

  - Comparison

= != <> > < >= <=

AND OR NOT

Operators in PostgreSQL

- Using numeric expressions:

```
select 24*4 as product;
```

- Date expression gives current date and time:

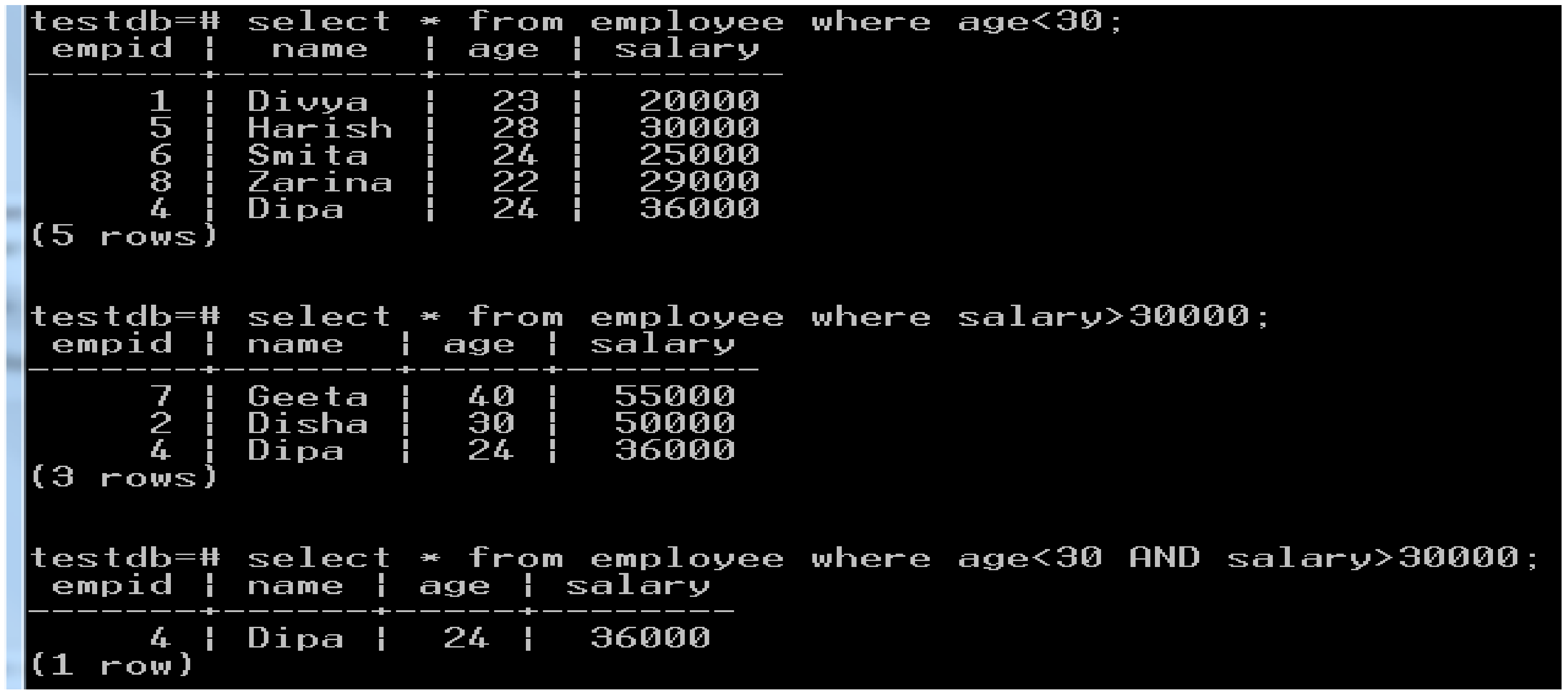
```
select CURRENT_TIMESTAMP;
```

Using WHERE clause

- In PostgreSQL WHERE clause is used to specify a condition while fetching data from one table or joining it with multiple tables
- It returns specific values from the table if the given condition is satisfied
- WHERE clause can be used in SELECT, UPDATE and DELETE statement
- Using comparison operators in SELECT statement:

Using WHERE clause

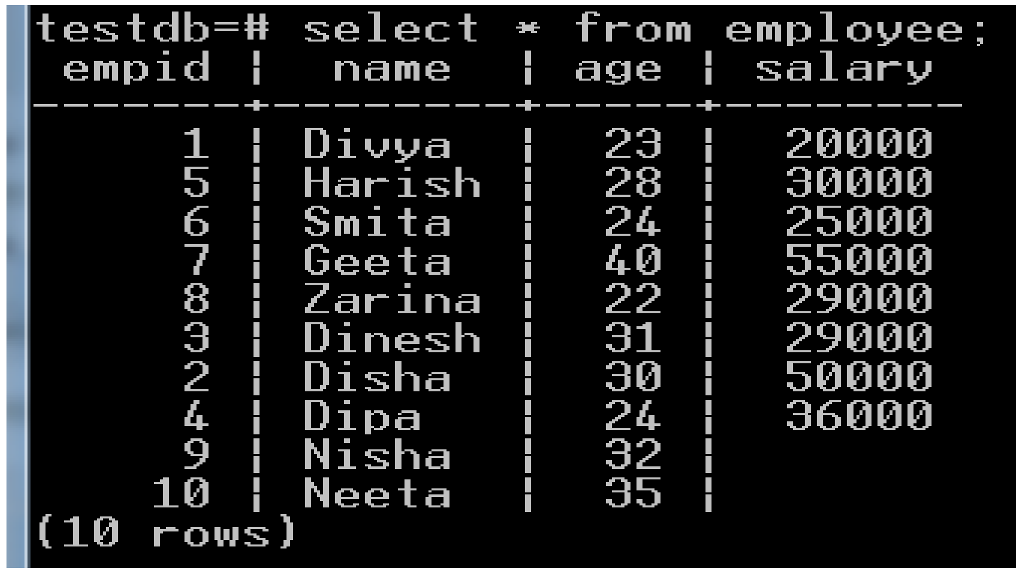
- WHERE clause can be used in SELECT, UPDATE and DELETE statement
- Using comparison operators in SELECT statement:



```
testdb=# select * from employee where age<30;
 empid |  name   | age | salary
-------+---------+-----+--------
     1 | Divya   |  23 |  20000
     5 | Harish  |  28 |  30000
     6 | Smita   |  24 |  25000
     8 | Zarina  |  22 |  29000
     4 | Dipa    |  24 |  36000
(5 rows)


testdb=# select * from employee where salary>30000;
 empid | name   | age | salary
-------+--------+-----+--------
     7 | Geeta  |  40 |  55000
     2 | Disha  |  30 |  50000
     4 | Dipa   |  24 |  36000
(3 rows)


testdb=# select * from employee where age<30 AND salary>30000;
 empid | name | age | salary
-------+------+-----+--------
     4 | Dipa |  24 |  36000
(1 row)
```



```
testdb=# select * from employee;
 empid |  name   | age | salary
-------+---------+-----+--------
     1 | Divya   |  23 |  20000
     5 | Harish  |  28 |  30000
     6 | Smita   |  24 |  25000
     7 | Geeta   |  40 |  55000
     8 | Zarina  |  22 |  29000
     3 | Dinesh  |  31 |  29000
     2 | Disha   |  30 |  50000
     4 | Dipa    |  24 |  36000
     9 | Nisha   |  32 |
    10 | Neeta   |  35 |
(10 rows)
```

Using WHERE clause

- Using OR and NOT:

```
select * from employee where age<30 OR salary>30000
select * from employee where salary IS NOT NULL;
```

- Using special operators like IN, between and :

```
select * from employee where salary between 25000 and 30000;
select * from employee where age IN (28,30);
```

- **Using LIKE operator to match wild card characters _ and %:**

```
      select * from employee where name LIKE 'Di%';
select * from employee where name LIKE '___ta';
```

- _ is for single character and % is for 1 or more characters

LIMIT clause

- LIMIT clause limits the data returned by the SELECT statement
- Example : if we need first four rows in employee table then

```
select * from employee limit 4;
```

- To display data starting from row 4, show next 2 rows we should use:

```
select * from employee limit 2 offset 3;
```

- Limit 2 is number of rows and
- offset 3 means start from row 4

ORDER BY clause

- PostgreSQL sorts the data in ascending or descending order, based on 1 or more columns
- Example : Sort employee data according to age:

```
select * from employee order by age;
```

- Sort employee data according to deptno and then by name:

```
select * from employee order by deptno, name;
```

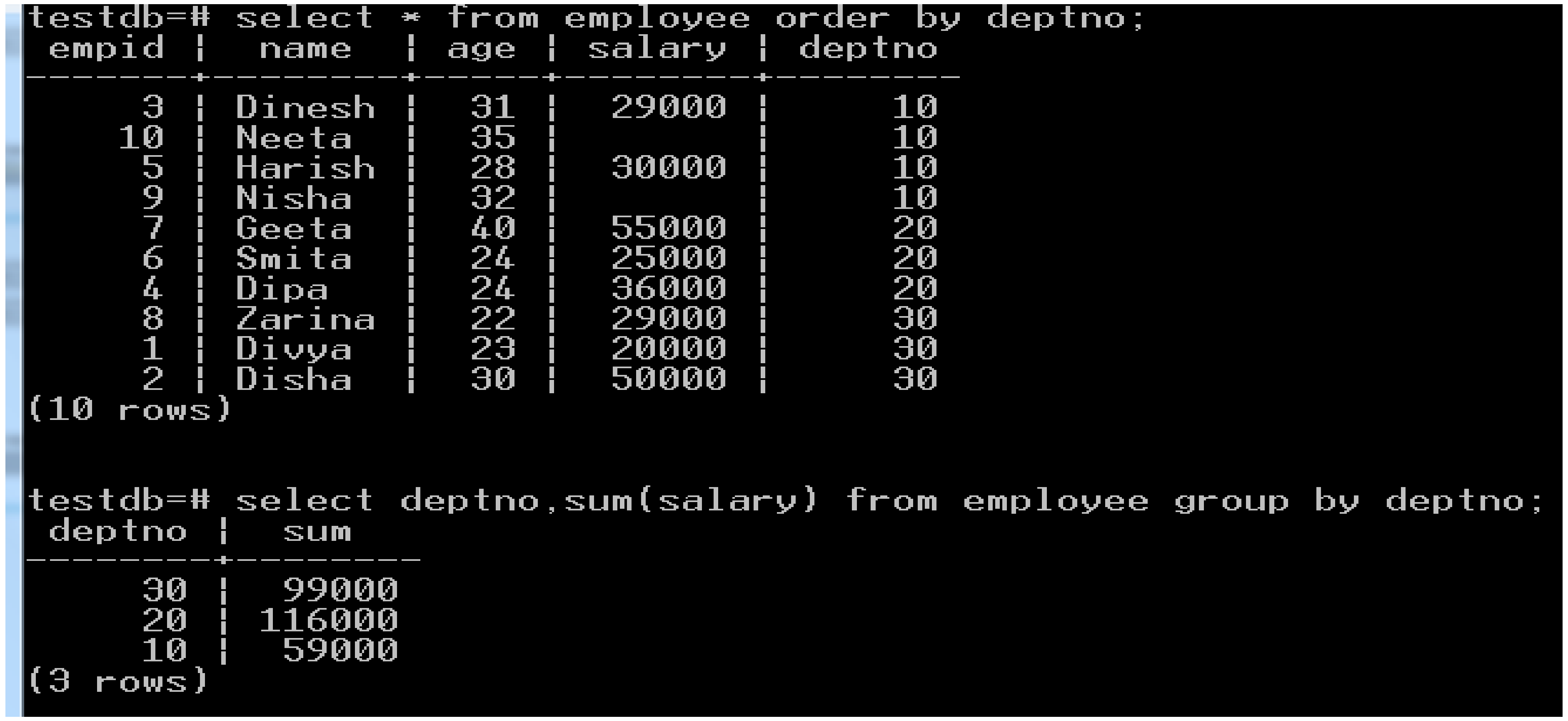
- Sort employee data according to salary in descending order:

```
select * from employee order by salary desc;
```

GROUP BY clause

- GROUP BY clause is used in SELECT statement to group together rows in the table that have identical data
- Example :

select deptno, sum(salary) from employee GROUP BY deptno;



```
testdb=# select * from employee order by deptno;
 empid |  name  | age | salary | deptno
-------+--------+-----+--------+--------
     3 | Dinesh |  31 |  29000 |     10
    10 | Neeta  |  35 |        |     10
     5 | Harish |  28 |  30000 |     10
     9 | Nisha  |  32 |        |     10
     7 | Geeta  |  40 |  55000 |     20
     6 | Smita  |  24 |  25000 |     20
     4 | Dipa   |  24 |  36000 |     20
     8 | Zarina |  22 |  29000 |     30
     1 | Divya  |  23 |  20000 |     30
     2 | Disha  |  30 |  50000 |     30
(10 rows)

testdb=# select deptno,sum(salary) from employee group by deptno;
 deptno |  sum
--------+--------
     30 |  99000
     20 | 116000
     10 |  59000
(3 rows)
```

HAVING clause

- HAVING clause restricts groups
- WHERE clause places conditions on the selected columns, whereas the HAVING clause places conditions on groups created by the GROUP BY clause
- Example: Get data for those departments which have average salary more that 30000

```
select deptno, avg(salary) from employee group by deptno having avg(salary)>30000 ;
```

- Get all department numbers which have less then 4 employees

```
select deptno, count(*) from employee group by deptno having count(*)<4;
```

DISTINCT key word

- DISTINCT clause gets unique values after removing duplicate values from the data
- Example : if we need to get all deptno values in employee table then

```
select DISTINCT deptno from employee;
```

- To display data starting from row 4, show next 2 rows we should use:

```
select * from employee limit 2 offset 3;
```

- Limit 2 is number of rows and
- offset 3 means start from row 4

Lab

Lab 2

Summary

In this lesson, you have learn about:

- Use SELECT statement to retrieve rows form table in database
- WHERE clause will restrict data based on conditions
- Aggregate functions can be used with GROUP BY clause and HAVING clause will restrict groups
- DISTICT key word will help remove duplicate values

Review Question

Question 1: We need employees who work in department 10 and have salaries in the range 20000 and 30000. Which of the following is a correct query?

- Select * from emp where deptno=10 and salary IN (20000,30000)
- Select * from emp where deptno=10 and salary between 20000 and 30000
- Select * from emp where deptno=10 or salary between 20000 and 30000

Question 2: Which of the following is useful to get unique values from a repeated group?

- Distinct
- Primary key
- Unique

People matter, results count.

About Capgemini

A global leader in consulting, technology services and digital transformation, Capgemini is at the forefront of innovation to address the entire breadth of clients' opportunities in the evolving world of cloud, digital and platforms. Building on its strong 50-year heritage and deep industry-specific expertise, Capgemini enables organizations to realize their business ambitions through an array of services from strategy to operations. Capgemini is driven by the conviction that the business value of technology comes from and through people. It is a multicultural company of 200,000 team members in over 40 countries. The Group reported 2016 global revenues of EUR 12.5 billion.

Visit us at

www.capgemini.com