Operators
&
Functions

M. Anbarasi, AP, SCSE, VIT University

Operators

- Arithmetic operator
- Logical operator
- Comparison operator

Type	Symbol / Keyword	Where to use
Arithmetic	+ , - , * , /	To manipulate numerical column values, WHERE clause
Comparison	=, !=, <, <=, >, >=, between, not between, in, not in, like, not like	WHERE clause
Logical	and, or, not	WHERE clause, Combining two queries

Operators

Arithmetic operators +, -, *, /

Operator precedence *, /, +, -

Q. Display a salary increase of $300 for all employees and displays a new salary+300 column in the output.

A. Select last_name, salary, salary+300 from employees;

Concatenation Operator

Concatenates columns or character strings to other columns

Is represented by vertical bars(||)

Select last_name || job_id as “employees” from employees;

Comparison conditions

Comparison conditions are used in conditions that compare one expression to another value or expression.

=, >, <,>=, <=, <> ,!= and ^=

Q. Display the last name and salary from the employees table, where the employee salary is less than or equal to $3000.

A. Select last_name, salary from employees where salary <=3000;

Other Comparison operator

Between	Between two values
….and….	
IN(set)	Match any of a list of values.
LIKE	Match a character pattern
Is null	Is a null value

Using the Between condition

Q. Display all the employees from the employees table whose salary is between $2,500 and $3500.

A. Select * from employees where salary between 2500 and 3500;

IN condition

Use the IN condition to test for values in a list

Q. Displays employee no, last name, salaries and manager's employee numbers for all the employees whose manager's employee number is 100,101 or 201

A. E.g select emp_id, last_name, salary, mgr_id from employees where mgr_id in(100,101,201);

(OR)

Select emp_id, last_name, salary, mgr_id from employees where mgr_id=100 or mgr_id=101 or mgr_id =201;

LIKE Condition

Use the LIKE condition to perform wild card searches of valid search string values.

Search conditions can contain either characters or numbers

% - denotes zero or many characters.

_ - denotes one character.

Q. Display the last names and hire dates of all employees whose first name begins with an S.

A. Select first_name,hiredate from employees where first_name like ‘S%’;

NULL Conditions

A NULL value means the value is unavailable, unassigned, unknown, or not applicable.

NULL conditions include the

IS NULL condition and the IS NOT NULL condition.

Q. Display last name, mgr id and commission for all employees who are not entitled to get a commission

A. Select last_name, mgr_id, commission from employees where commission is null;

Logical Conditions

And	Returns True if both component conditions are true
OR	Returns True if either component condition is true
NOT	Returns True if the following condition is false

Q. Display the employees who have a job title that contains the string MAN and earn more than 10,000

A. Select emp_id, last_name, job_id, salary from employees where salary>=10000 and job_id like '%MAN%';

Q. Display all employees who has a job ID and salary containing MAN or earns more than $10,000

A. Select emp_id, job_id, salary from employees where salary>=10000 or job_id like '%MAN%';

Q. Display the last name and job id of all employees whose job id is not it or cse

A. Select last_name, job_id from employees where job_id not in('it' , 'cse');

Note

The NOT operator can also be used with other SQL operators, such as BETWEEN, LIKE and NULL.

.....where job_id NOT IN (‘ACCOUNTS’ , ‘SALES’);

..... Where salary NOT BETWEEN 1000 AND 5000;

..... Where last_name NOT LIKE ‘%A%’;

……. Where commission IS NOT NULL;

Sorting

ORDER BY Clause

Sort rows with the ORDER BY clause

- ASC : ascending order (the default order)
- DESC : descending order
- The ORDER BY clause comes last in the select statement
- Syntax

```
Select   expr
from     table
[where condition(s)]
[order by  {column, expr} [ASC | DESC]];
```

Q. Display the last name and salaries of all employees. Sorts the result by the most recently hired employee.

A. Select last_name, salary

from employees

order by hire_date desc;

FUNCTIONS

Single Row Functions

Group Functions

Single Row Functions

Returns only one value for every row.

Can be used in SELECT command and included in WHERE clause

Types

- Character functions
- Numeric functions
- Date functions

Character Functions

Syntax	Description
initcap (char)	Changes first letter to capital
lower (char)	Changes to lower case
upper (char)	Changes to upper case
ltrim (char, set)	Removes the set from left of char
rtrim (char, set)	Removes the set from right of char
translate(char, from, to)	Translate ‘from’ anywhere in char to ‘to’
replace(char, search string, replace string)	Replaces the search string to new

Character Function con……

Syntax	Description
substring(char, m , n)	Returns chars from m to n length
lpad(char, length, special char)	Pads special char to left of char to Max of length
rpad(char, length, special char)	Pads special char to right of char to Max of length
chr(number)	Returns char equivalent
length(char)	Length of string

Examples for Character Functions

Function	Result
INITCAP (‘oracle software’)	Oracle Software
LOWER (‘ORACLE Software’)	oracle software
UPPER (‘Oracle Software’)	ORACLE SOFTWARE
LTRIM (‘ Oracle’,’ ‘)	Oracle
RTRIM (‘Oracle’,’e’)	Oracl
TRANSLATE('223abcd', '0123abc', '9999XXX')	999XXXd
REPLACE('JACK and JUE','J','BL')	BLACK and BLUE

Examples for Character Functions

Function	Result
SUBSTR('ABCDEFG',3,3)	CDE
LPAD('Page 1',15,'*')	*********Page 1
RPAD('Page 1',15,'*')	Page 1*********
chr(65)	A
length(‘oracle’)	6

Numeric Functions

Syntax	Description
abs ( )	Returns the absolute value
ceil ( )	Rounds the argument
cos ( )	Cosine value of argument
exp ( )	Exponent value
floor( )	Truncated value
power (m,n)	N raised to m
mod (m,n)	Remainder of m / n
round (m,n)	Rounds m’s decimal places to n
trunc (m,n)	Truncates m’s decimal places to n
sqrt (m)	Square root value

Numeric Functions

Function	Result
abs (-15)	15
ceil (11.7)	12
cos (0*3.14/180)	1
exp(4)	54.59815
floor(11.7)	11
power (2,3)	8
mod (1000,300)	100
round(15.136,2)	15.14
trunc (15.136,2)	15.13
sqrt (9)	3

Date Functions

Syntax	Description
add_months(date,no. of months)	Return the date after adding the number of months
last_day(date)	Returns the last date corresponding to the last day of the month
months_between(date1,date2)	Returns the numeric value of the difference between the months.
round(date, [format])	Format – ‘day’, ‘month’ , ‘year’ rounded to the nearest format specified
next_day(date, day)	Returns the next date of the day
trunc(date, [format])	Format – ‘day’, ‘month’ , ‘year’ Day – previous nearest Sunday Month – start date of the month Year – start date of the year

SYSDATE Function

SYSDATE is a date function that returns the current database server date and time.

Select sysdate from dual;

DATE Functions

Sysdate = 08-Aug-09

Function	**Result**
Add_months('11-jan-09',6)	11-JUL-09
Last_day(sysdate)	31-AUG-09
Months_between(sysdate,'11-jan-2009')	6.92621378
Next_day('08-aug-09','wednesday')	15-AUG-09

DATE Functions

Sysdate = 08-Aug-09

Function	Result
Round(sysdate,'month')	01-AUG-09
Round(sysdate,'year')	01-JAN-10
Trunc(sysdate,'month')	01-AUG-09
Trunc(sysdate,'month')	01-JAN-09

Group Functions:

Result based on group of rows.

Group functions operate on set of rows to give one result per group
Employees

Dept_id	Salary
90	5000
90	10000
90	10000
60	5000
60	5000

The maximum salary in the employees table

Max(salary)

10000

Types of Group Functions

Syntax	Description
count (*), count (column name), count (distinct column name)	Returns number of rows
min (column name)	Min value in the column
max (column name)	Max value in the column
avg (column name)	Avg value in the column
sum (column name)	Sum of column values

Group Functions Syntax

```
Select      [column,]  group_function(column),..
  From        table
  [where      condition]
  [GROUP BY column];
```

Q. Display the average,highest, lowest and sum of salaries for all the sales representatives.

A. Select avg(salary), max(salary), min(salary), sum(salary)
From employees where job_id like ‘%rep%’;

Groups of Data

Divide rows in a table in to smaller groups by using the group by clause

Employees

Dept_id	Salary
10	4000
10	5000
10	6000
50	5000
50	3000

The average salary in employees table for each department

D_id	Avg(Salary)
10	5000
50	4000

Select dept_id, avg(salary) from employees group by dept_id;