**Database Management System**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Practical No : 3**    **Aim :** 1. Functions   * Aggregate, Numeric, Date, String and Conversion  1. **Aggregate Functions**   AVG: returns average value  *Select Avg(<ColumnName>) from <table\_name>*  MIN: returns minimum value  *Select min(<ColumnName>) frsom <table\_name>*  COUNT: returns no of rows where expression is not NULL  *Select count(<ColumnName>) from <table\_name>*  COUNT(\*): returns no of rows in the table including duplicates and those with NULL  *Select count(\*) from <table\_name>*  MAX: returns maximum value  *Select max(<ColumnName>) from <table\_name>*  SUM: returns sum of the values  *Select sum(<ColumnName>) from <table\_name>*   1. **Numeric Functions**   ABS: returns the absolute value of n  *Select abs(n); e.g. select abs(-24.5); =>24.5*  POWER: returns m raised to nth power  *Select power(m,n); e.g. Select power(2,3); =>8*  ROUND: returns n rounded to m places to the right of the decimal point  *Select round(n,m); e.g. Select round(5.2333, 2);*  SQRT: returns square root of n  *Select sqrt(n);*  EXP: returns e raised to nth power  *Select exp(n);*  FLOOR: returns the largest integer value that is equal to or less than a number  *Select floor(n);*  CEIL: returns the smallest integer value that is equal to or greater than a number  *Select ceil (n);*  RAND: returns a random number or a random number within a range.  ***Select Rand();***  *Rand()\*(upper value-lower value) e.g. select rand()\*(10-1)*   1. **String Functions:**   LEN: returns the length of the specified string.  *Select LENGTH( ‘string’ ) from dual;*  CONCAT: allows you to concatenate strings together.  *Select CONCAT( ‘string1’, ‘string2’, ... ‘string\_n’) from dual;*  LOWER: converts all letters in s specified string to lowercase (same for Upper).  *Select LOWER( ‘string’ ) from dual, or select lower(ename) from employees;*  UPPER: converts all letters in the specified string to lowercase (same for Upper).  *Select UPPER( ‘string’ ) from dual, or select upper(ename) from employees;*  INITCAP(‘string’) proper case  *E.g. select Initcap(‘ALPHA’) from dual; =>Alpha*  SUBSTR(argument, starting point, length) extract substring from string.  *E.g. select Substr(‘Alpha’,2,3) from dual =>pha*  INSTR(String,char) Returns Position of char  *Select INSTR(String,char) from dual;*  REPLACE: Replace all occurrences of a substring by another substring in a string.  *Select (stringfrom, string to find, string to replace)*  *Select REPLACE(‘JACK AND JOND’,’J’,’BL’);=>  ‘BLACK AND BLOND’*   1. **Date Functions:**   TO\_DATE(‘char\_string’, ‘Oracle format’): Converts a character string to an Oracle date format  *SELECT TO\_DATE( '5 Jan 2017', 'DD MON YYYY' ) FROM dual.*  TO\_CHAR(Input, format)  SELECT CHAR( sysdate 'DD MON YYYY' ) FROM dual;  ADD\_MONTHS(date, number): Adds number of months to the date yielding another date  *SELECT ADD\_MONTHS( DATE '2016-02-29', 1 ) FROM dual;*  MONTH\_BETWEEN(date1.date2): Yields the number of months between date 1 & date 2  *SELECT MONTHS\_BETWEEN( DATE '2017-07-01', DATE '2017-01-01' ) FROM DUAL;*  LAST\_DAY(date1): Returns the last day of the month  *SELECT LAST\_DAY(SYSDATE) FROM dual;*  NEXT\_DAY(date,no. of day in week) :Returns Date Of Next comming Day  e.g.(Jn\_date,1) Returns Date of Next Monday of Jn\_date  Date + value(n) => ﻿select (sysdate+5) from dual;  Returns Date after n days of Date  **se**  CURRENT\_TIMESTAMP: returns the current date and time (GETDATE function can also be used).  Select CURRENT\_TIMESTAMP from dual;  EXTRACT:    **Arithmetic operations**  Select <ColumnName1> <ColumnName2><ColumnName3><arithmetic expression>  **Renaming column name while using arithmetic operations**  ***Select <ColumnName1> <ColumnName2>arithmetic expression> “<ColumnName>”***  **Use of logical operators in where condition**  Select <ColumnName1> <ColumnName2> where <cond1> and <cond2>  Select <ColumnName1> <ColumnName2> where <cond1> or <cond2>  Select <ColumnName1> <ColumnName2> where not<cond1>  **Range searching**  Select <ColumnName1> <ColumnName2> where < ColumnName> between <val1> and <val2>  **Pattern Matching**  Select <ColumnName1> <ColumnName2> where < ColumnName> like <pattern>   |  |  | | --- | --- | | WHERE EName LIKE 'a%' | Finds any values that start with "a" | | WHERE EName LIKE '%a' | Finds any values that end with "a" | | WHERE EName LIKE '%or%' | Finds any values that have "or" in any position | | WHERE EName LIKE '\_r%' | Finds any values that have "r" in the second position | | WHERE EName LIKE 'a\_\_%' | Finds any values that start with "a" and are at least 3 characters in length | | WHERE EName LIKE 'a%o' | Finds any values that start with "a" and ends with "o" | | '[a-c]%' | starting with "a", "b", or "c": | | '[bsp]%' | starting with "b", "s", or "p" | | '[!bsp]%' or **not like '[bsp]%'** | NOT starting with "b", "s", or "p" |   Example: **WHERE SALARY LIKE '200%' =>** Finds any values that start with 200.  **In and not in predicates**  Select <ColumnName1> <ColumnName2> where < ColumnName> in (list)  **Execute prac\_3.sql script**  **Steps :**   1. Download prac\_3.sql from course website, copy to d:\ 2. Run this script on SQLPLUSW using following command   SQL>@d:\prac\_3.sql  **Do as directed:**   1. Print the incremented salary of all employee with 10% increment. 2. Find the (salary + Commission) as Total salary of all employees. 3. Find the gross salary of employee by following formula:   Gross\_Salary= Basic Salary+DA+HRA+TA+MA  DA=110% of Basic Salary  HRA=30% of Basic Salary  TA= 1500  MA= 1000   1. Display the first name in lower case and last name in upper case, for all employees whose employee number is in the range between 80 and 150. 2. Generating new email address   a) For each employee, display the first name, last name, and email address. The email address will be composed from the first letter of first name, concatenated with the three first letters of last name, concatenated with *@oracle.com*.  b) For each employee, display the first name, last name, and email address. The email address will be composed from the first letter of first name, concatenated with the three last letters of last name, concatenated with *@oracle.com*.   1. For each employee, display the first name concatenated with the last name, concatenated with hire date. 2. Display the last name for all employees where last name’s length is greater than 8 characters. 3. For each employee, display the first name, last name, phone number and a new phone number. In the new phone number, replace all occurrences of *515* with *815.* 4. For each employee, display first name, salary, salary after a raise of 12%, salary after a raise of 12%, expressed as a whole number, salary after a raise of 12%, round down to the nearest whole number. 5. For each employee, display the first name, hire date, hire date minus 10 days, hire date plus one month, and the day difference between current date and hire date. 6. For each employee, display the first name, last name, hire date, number of months he works in the company, and number of years he works in the company. 7. For each employee, display the first name, hire date, and hire date plus one year. 8. For each employee, display the first name, hire date, hire date rounded up to the nearest year, and hire date rounded up to the nearest month. 9. For each employee, display the first name, the day of his hire date, and the year of his hire date. 10. Display the last name in upper case, the salary in format model : ‘9,999.999’, and hire date in format model: ‘DD/MM/YYYY’, for all employees whose last name begins with the letter *D* or *K* (without using like). 11. For each employee, display the first name, last name, salary and commission percentage. If an employee doesn’t earn a commission, display 0 instead of NULL. 12. For each employee, display the first name, last name, salary and commission percentage. If an employee doesn’t earn a commission, display “No Commission” instead of NULL. 13. ***For each employee, display the first name, last name, salary, and a salary grade based on these conditions :***   ***a) if the salary is between 0 and 5000 – salary grade level is A***  ***b) if the salary is between 5001 and 15000 – salary grade level is B***  ***c) if the salary is between 15001 and 20000 – salary grade level is C***  ***d) for any other range – salary grade level is D***   1. Display the first name, salary and round the salary to thousands. 2. Display all the employees who joined in the month of May. 3. ***Display the first word in job title.*** 4. Display the length of first name for employees whose last name contains character ‘b’ after the third position. (Without using like). 5. Display first name of the employees whose experience in more than 5 years. |
|  |
|  |