Section 1

Assignment

Feb19/ DBT/001

Database Technologies

Diploma in Advance Computing

February 2019

1. Create ***N1Employee*** Relation with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ID | int primary key |
| FIRSTNAME | varchar(20) |
| LASTNAME | varchar(20) |
| GENDER | char (1) |
| MOBILENO1 | varchar(12) |
| MOBILENO2 | varchar(12) |
| HOBBY1 | varchar(25) |
| HOBBY2 | varchar(25) |
| CURRENTJOB | varchar(50) |
| PREVIOUSJOB1 | varchar(50) |
| PREVIOUSJOB2 | varchar(50) |
| PREVIOUSJOB3 | varchar(50) |
| ADDRESS | varchar(100) |
| HIREDATE | date |
| SALARY | int |
| COMM | int |
| DEPTNAME | varchar(50) |
| QUALIFICATION | varchar(100) |

1. Create ***ACTOR*** Relation with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ACTORID | int |
| NAME | varchar(20) |
| GENDER | char (1) |
| RATING | int |

1. Create ***MOVIE*** Relation with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| MOVIEID | int primary key |
| NAME | varchar(20) |
| RELEASE\_DATE | date |

1. Create ***ACTOR\_MOVIE*** Relation with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ACTORID | int |
| MOVIEID | int primary key |

Assignment

Feb19/ DBT/005

Database Technologies

Diploma in Advance Computing

February 2019

1. Create N2Employee table with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ID | int primary key |
| FIRSTNAME | varchar(12) |
| LASTNAME | varchar(12) |
| GENDER | char (1) |
| HIREDATE | date |

1. Create N2Department table with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ID | int primary key |
| NAME | varchar(20) |
| LOCATION | varchar(20) |

1. Create N2Employee\_Department table with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ID | int primary key |
| DEPARTMENTID | Int [foreign key(DEPARTMENTID) references N2Department(ID)] |
| EMPLOYEEID | Int [foreign key(EMPLOYEEID) references N2Employee(ID)] |
| FROMDATE | Date |
| TODATE | Date |

1. Create N2SALARY table with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ID | int primary key |
| EMPLOYEEID | int [foreign key(EMPLOYEEID) references N2Employee(ID)] |
| FROMDATE | date |
| TODATE | date |
| SALARY | int |

1. Create N2COMMISSION table with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ID | int primary key |
| EMPLOYEEID | Int [foreign key(EMPLOYEEID) references N2Employee(ID)] |
| DATE | date |
| COMMISSION | float |

1. Create N2CONTACT table with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ID | int primary key |
| EMPLOYEEID | int [foreign key(EMPLOYEEID) references N2Employee(ID)] |
| PHONENUMBER | BIGINT(12) |
| EMAILID | varchar(40) |

1. Create N2ADDRESS table with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ID | int primary key |
| EMPLOYEEID | int [foreign key(EMPLOYEEID) references N2Employee(ID)] |
| LINE1 | varchar(30) |
| LINE2 | varchar(30) |
| CITY | varchar(20) |
| PIN | Int |

1. Create N2QUALIFICATION table with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ID | int primary key |
| EMPLOYEEID | int [foreign key(EMPLOYEEID) references N2Employee(ID)] |
| NAME | varchar(10) |
| STREAM | varchar(18) |
| ADDMISSIONYEAR | date |
| INSTITUTE | varchar(15) |
| UNIVERSITY | varchar(25) |
| YEAROFPASSING | int |
| PERCENTAGE | float(7, 2) |
| GRADE | char(6) |

1. Create N2HOBBIES table with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ID | int primary key |
| EMPLOYEEID | int [foreign key(EMPLOYEEID) references N2Employee(ID)] |
| NAME | varchar(25) |

1. Create N2ORDER table with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ID | int primary key |
| EMPLOYEEID | Int [foreign key(EMPLOYEEID) references N2Employee(ID)] |
| ORDER\_DATE | date |
| AMOUNT | float |

1. Create N2JOBHISTORY table with following columns using Workbench.

|  |  |
| --- | --- |
| Field Name | Datatype (size) |
| ID | int primary key |
| EMPLOYEEID | int [foreign key(EMPLOYEEID) references N2Employee(ID)] |
| EMPLOYER | varchar(20) |
| FROMDATE | date |
| TODATE | date |
| NATURE | varchar(50) |

Assignment

Feb19/ DBT/ 012

Database Technologies

Diploma in Advance Computing

February 2019

**Temporary tables and VIEWS**

|  |
| --- |
| 1. Write a query to create a view employee for all employees with columns id, firstName, lastName, gender, and hiredate. |
| create view e as select \* from n2employee; |
|  |
| 1. Write a query to create a view ***employee\_qualification*** for all employees along with their qualification details (ie. *firstname, lastname, gender, name, stream, institute,* and *university*). |
| create view employee\_qualification as select firstname, lastname, gender, name, stream, institute, university from n2employee e, n2qualification q where e.id = q.employeeid; |
|  |
| 1. Write a query to create a view ***employee\_commission*** for all employees who are getting commission in their current job. |
| create view employee\_commission as select e.id, firstname, lastname, gender, commission from n2employee e, n2commission c where e.id=c.employeeid and (employeeid, todate) in (select employeeid, max(todate) from n2commission group by employeeid) and commission is not null order by employeeid; |
|  |
| 1. Write a query to create a view ***employee\_address*** with columns (*employeeid, firstname, lastname, gender, line1, line2, city,* and *pin*) for all employees. |
| create view employee\_address as select e.id, firstname, lastname, gender, line1, line2, city, pin from n2employee e, n2address a where e.id = a.employeeid; |
|  |
| 1. Write a query to create a view ***employee\_address\_yearwise*** with columns (*firstname, lastname, hiredate, line1, line2, city,* and *pin*) for all employees who have joined the company in the year 1962. |
| create view employee\_address\_yearwise as select firstname, lastname, hiredate, line1,line2, city, pin from n2employee e, n2address a where e.id = a.employeeid and date\_format(hiredate, '%Y') =1962; |
|  |
| 1. Create temporary table ***temp\_employee*** alike n2employee relation. |
| create temporary table temp\_employee as select \* from n2employee; |
|  |

Section 2

Assignment

Feb19/ DBT/ 002

Database Technologies

Diploma in Advance Computing

February 2019

**DML commands: Insert data and filter data using Workbench.**

***Use “1NF DATA.xlsx” 1NF Data sheet to INSERT records in N1Employee Table.***

USE ***N1Employee*** relation to solve the following queries.

|  |
| --- |
| 1. Display all records. |
| select \* from n1employee; |
|  |
| 1. Display the *firstName*, lastName of all *employees*. |
| select firstname, lastname from n1employee; |
|  |
| 1. Get the employee details whose employee *firstName* is ‘Fred’. |
| select \* from n1employee where firstname ='Fred'; |
|  |
| 1. Get all female employees. |
| select \* from n1employee where gender = 'F'; |
|  |
| 1. Get all employees whose *hobby1* is ‘Reading’. |
| select \* from n1employee where hobby1 = 'Reading'; |
|  |
| 1. List all employees whose *currentjob* is ‘Software Developer’. |
| select \* from n1employee where currentjob = 'Software Developer'; |
|  |
| 1. List all employees whose *PREVIOUSJOB3* is ‘Trainee’. |
| select \* from n1employee where previousjob3 = 'Trainee'; |
|  |
| 1. List all employees whose *salary* is more than 4000. |
| select \* from n1employee where salary > 4000; |
|  |
| 1. List all employees whose *deptname* is ‘sales’. |
| select \* from n1employee where deptname = 'sales'; |
|  |
| 1. List all employees whose *salary* is less than 3000. |
| select \* from n1employee where salary < 3000; |
|  |
| 1. Get all employees whose *hobby2* is ‘Ice skating’. |
| select \* from n1employee where hobby2 = 'Ice skating'; |
|  |
| 1. Get all male employees. |
| select \* from n1employee where gender = 'm'; |
|  |
| 1. Get all employees whose *lastName* is ‘Clark’. |
| select \* from n1employee where lastname = 'Clark'; |
|  |
| 1. List the employee whose *mobileno1* is 7032300039. |
| select \* from n1employee where mobileno1 = 7032300039; |
|  |
| 1. List the employee whose *id* is 7. |
| select \* from n1employee where id = 7; |
|  |

Assignment

Feb19/ DBT/ 003

Database Technologies

Diploma in Advance Computing

February 2019

**DML commands: Select data with WHERE clause.**

USE ***N1Employee*** relation to solve the following queries.

|  |
| --- |
| 1. List all employees. |
| select \* from n1employee; |
|  |
| 1. List *firstname*, *lastname* of all employees. |
| select firstname, lastname from n1employee; |
|  |
| 1. List *firstname*, *lastname*, *hiredate*, and *salary* all employees. |
| select firstname, lastname, hiredate, salary from n1employee; |
|  |
| 1. Display employee information of the *employee* *ID* is 15. |
| select \* from n1employee where id=5; |
|  |
| 1. List *firstname*, *lastname*, and *currentjob* of employees whose *currentjob* is ‘Sr.Assistant’. |
| select firstname, lastname, currentjob from n1employee where currentjob = 'Sr.Assistant'; |
|  |
| 1. List all employee having *salary* greater than equal to 3000. |
| select \* from n1employee where salary >=3000; |
|  |
| 1. List *firstname*, *lastname*, *gender*, *mobileno1*, *hobby1*, and *hobby2* whose gender is ‘M’. |
| select firstname, lastname, gender, hobby1, hobby2 from n1employee where gender = 'M'; |
|  |
| 1. Display the employee *id*, *firstname*, *lastname*, and *currentjob* & Annual Salary for all Employees belonging to Department Name is ‘*SALES’*. |
| select id, firstname, lastname, currentjob, salary\*12 as "annual salary" from n1employee where deptname = 'sales'; |
|  |
| 1. List all employees who don’t have 2nd mobile. |
| select \* from n1employee where mobileno2 is null; |
|  |
| 1. List all employees who have *salary* more than 3000Rs. |
| select \* from n1employee where salary > 3000; |
|  |
| 1. List all employees (*firstname*, *lastname*, *gender*, *mobileno1*, *salary*, *and* *deptname*) who are in either ‘ACCOUNTING’ or ‘RESEARCH’ or ‘SALES’ department. |
| select firstname, lastname gender, mobileno1, salary, deptname from n1employee where deptname in ('accounting' ,'research', 'sales'); |
|  |
| 1. List all employees whose *currentjob* is either ‘Head Clerk’ or ‘Sr. Analyst’. |
| select \* from n1employee where currentjob = 'Head Clerk' or currentjob='Sr. Analyst'; |
|  |
| 1. List all employees whose *gender* is ‘F’. |
| select \* from n1employee where gender = 'F'; |
|  |
| 1. List all employees whose *hobby1* is ‘Running’. |
| select \* from n1employee where hobby1 = 'Running'; |
|  |
| 1. Display the *firstname*, *lastname*, *gender*, *mobileno1*, AND *mobileno2* change the column heading of *mobile1* to ‘HOME MOBILE’, and *mobile2* to ‘OFFICE MOBILE’. |
| select firstname, lastname, gender, mobileno1 as "HOME MOBILE", mobileno2 as "OFFICE MOBILE" from n1employee; |
|  |
| 1. List all employees whose *salary* is in the range of 2000 to 3000. |
| select \* from n1employee where salary>=2000 and salary <= 3000; |
|  |
| 1. Computer Total Salary by adding *salary* and *comm*. (hint: if *comm* is NULL replace *comm* with 0) |
| select salary + ifNull(comm,0) as "Total Salary" from n1employee; |
|  |
| 1. Display, what will be the New Commission, after increasing the *comm* by .25 %. If commission is null give 1000 as New Commission (hint: display *comm* and *New Commission*) |
| select comm, ifNull(comm + comm\*.25, 1000) from n1employee; |
|  |
| 1. List all employees who had joined the organization on 2018-05-12. (use *hiredate*) |
| select \* from n1employee where hiredate = '2018-05-12'; |
|  |
| 1. List all employees of ‘OPERATIONS’ department. |
| select \* from n1employee where deptname = 'OPERATIONS'; |
|  |
| 1. List all employees who are not receiving the *comm.* |
| select \* from n1employee where comm is null; |
|  |
| 1. Display (*currentjob, previousjob1, previousjob2,* and *previousjob3*) whose employee *id* is 10. |
| select currentjob, previousjob1, previousjob2, previousjob3 from n1employee where id=10; |
|  |
| 1. Display all employee who were hired on 1983. |
| select \* from n1employee where date\_format(hiredate, '%Y') = 1983; |
|  |
| 1. Display all employee whose salary is more than 4000. |
| select \* from n1employee where salary > 4000; |
|  |
| 1. Display all employee whose salary is more than 4000 and less than 5000. |
| select \* from n1employee where salary > 4000 and salary < 5000; |
|  |

Assignment

Feb19/ DBT/ 004

Database Technologies

Diploma in Advance Computing

February 2019

**DML commands: Select data with WHERE, LIMIT, and ORDER BY clause.**

USE ***N1Employee*** relation to solve the following queries.

|  |
| --- |
| 1. List all employees. |
| select \* from n1employee; |
|  |
| 1. List *firstname*, *lastname* of all employees in ascending order of *firstname*. |
| select firstname, lastname from n1employee order by firstname; |
|  |
| 1. List *firstname*, *lastname*, *hiredate*, and *salary* for the first 5 employees. |
| select firstname, lastname, salary from n1employee limit 5; |
|  |
| 1. Display employee information of the employee *ID* is either 1, 2, 5 or 7. |
| select \* from n1employee where id in (1, 2, 5, 7); |
|  |
| 1. List *firstname*, *lastname*, and *currentjob* of employees whose *currentjob* is not ‘Sr.Assistant’, display first 7 rows. |
| select firstname, lastname, currentjob from n1employee where currentjob <> 'Sr.Assistant' limit 7; |
|  |
| 1. List all employee having *salary* is between to 3000 and 4000 in descending order of *salary*. |
| select \* from n1employee where salary between 3000 and 4000 order by salary desc; |
|  |
| 1. List all employees whose gender is ‘F’, display rows between 5 and 10. |
| select \* from n1employee where gender = 'F' limit 5, 10; |
|  |
| 1. Display the *id*, *firstname*, *lastname*, and calculate *Annual Salary* for all employees in ascending order of fourth column. |
| select id, firstname, lastname, salary\*12 as "Annual Salary" from n1employee order by 4; |
|  |
| 1. List all employees who don’t have 2nd mobile. |
| select \* from n1employee where mobileno2 is null; |
|  |
| 1. List all employees who have hired on '1980-05-02'. |
| select \* from n1employee where hiredate = '1980-05-02'; |
|  |
| 1. List all employees (id, *firstname*, *lastname*, *gender*, *mobileno1*, *salary*, *and* *deptname*) who are in either ‘ACCOUNTING’ or ‘RESEARCH’ or ‘SALES’ department order by *deptname*. |
| select id, firstname, lastname, gender, mobileno1,salary, deptname from n1employee where deptname in ('accounting', 'research', 'sales') order by deptname; |
|  |
| 1. List all employees whose *currentjob* is either ‘Head Clerk’ or ‘Sr. Analyst’ and firstname must be either ‘Peter’ or ‘Rosaleen’. |
| select \* from n1employee where currentjob in ('Head Clerk', 'Sr. Analyst') and firstname in ('peter', 'Rosaleen'); |
|  |
| 1. List all employees whose *gender* is 'F' and *hobby1* is 'Writing'. |
| select \* from n1employee where gender = 'F' and hobby1 = 'writing'; |
|  |
| 1. List first 1 rows of all employees whose *hobby1* is 'Running'. |
| select \* from n1employee where hobby1 = 'running' limit 1; |
|  |
| 1. Display the id, *firstname*, *lastname*, *gender*, *mobileno1*, and *mobileno2* change the column heading of *mobile1* to ‘HOME MOBILE’, and *mobile2* to ‘OFFICE MOBILE’ in ascending order of ‘HOME MOBILE’. |
| select id, firstname, lastname, gender, mobileno1 as "HOME MOBILE", mobileno2 as "OFFICE MOBILE" from n1employee order by "HOME MOBILE"; |
|  |
| 1. Display all employees whose *mobileno1* starts with '7'. Display first 5 rows only. |
| select \* from n1employee where mobileno1 like '7%' limit 5; |
|  |

Assignment

Feb19/ DBT/ 006

Database Technologies

Diploma in Advance Computing

February 2019

**String, Date, Math functions, and Date formats.**

USE *n2employee, n2department, n2employee\_department, n2salary, n2commission, n2contact, n2address, n2qualification, n2hobbies, n2order, and n2jobhistory*relation to solve the following queries.

|  |
| --- |
| 1. Get employee *firstname* with how many characters are there in their *firstname*. |
| select firstname, length(firstname) from n2employee; |
|  |
| 1. Get employee details whose *firstname* is having at least 4 characters. |
| select \* from n2employee where length(firstname) = 4; |
|  |
| 1. Get the ASCII value of the 3rd character of *firstname* column. |
| select firstname, ascii(substr(firstname,3,1)) from n2employee; |
|  |
| 1. Get *firstname* and *lastname* in lowercase. |
| select lower(firstname) , lower(lastname) from n2employee; |
|  |
| 1. Get *(hobby name)* all 7 letter hobbies. |
| select name from n2hobbies where length(name) = 7; |
|  |
| 1. Get *(firstname, lastname and first 3 letters of firstname)* for all employees. |
| select firstname, lastname, left(firstname, 3) from n2employee; |
|  |
| 1. Get *(firstname, lastname and last 3 letters of firstname)* for all employees. |
| select firstname, lastname, right(firstname, 3) from n2employee; |
|  |
| 1. Get all *(phonenumber)* whose *phonenumber* starts with 99. |
| select phonenumber from n2contact where phonenumber like '99%'; |
|  |
| 1. Get employee details of first 5 employees. |
| select e.\* from n2employee e limit 5; |
|  |
| 1. Get employee details of last 5 employees. |
| select @cnt:=@cnt+1 R1, emp.\* from emp, (select @cnt:=0) x order by R1 desc limit 5; |
|  |
| 1. Get employee details in ascending order of *firstname*. |
| select \* from n2employee order by firstname; |
|  |
| 1. Get employee details in descending order of *lastname*. |
| select \* from n2employee order by lastname desc; |
|  |
| 1. Get *(employee id, firstname, lastname, gender, phonenumber, and emailid)* for all employees whose length of email id is more than 20 characters. |
| select id, firstname, lastname, gender, phonenumber, and emailid from n2employee e, n2contact c where e.id = c.employeeid and length(emailid) > 20; |
|  |
| 1. Combine to display employee *firstname* and *lastname*. |
| select concat(firstname, ' ', lastname) from n2employee; |
|  |
| 1. Write a query to display the following output for all employees. If (*firstname*, *lastname or hiredate)* is null then replace it with a blank space.   **(Bhoopali Nanadikar and hired on 1962-04-10)** |
| select concat(ifnull(firstname, ' '), ' ', ifnull(lastname, ' '), "and hired on ", ifnull(hiredate, ' ' )) as R1 from n2employee; |
|  |
| 1. Get employee *firstname* and *lastname* in upper case. |
| select ucase (firstname), ucase(lastname) from n2employee; |
|  |
| 1. Get employee *firstname* and *lastname* in lower case. |
| select lcase (firstname), lcase(lastname) from n2employee; |
|  |
| 1. Get employee *firstname* and *lastname* in reverse order. |
| select reverse(firstname), reverse(lastname) from n2employee; |
|  |
| 1. Get first 4 letters of employee *firstname*. |
| select firstname, substr(firstname, 1, 4) as R1, left(firstname,4) as R2 from n2employee; |
|  |
| 1. Get second letter of employee *firstname* to second last letter of employee *firstname*. |
| select firstname, substr(firstname, 2, length(firstname) -2 ) as R1 from n2employee; |
|  |
| 1. Get ASCII character of employee *firstname*. |
| select firstname, ascii(firstname)from n2employee; |
|  |
| 1. Get 5 letter of the employee *firstname*. |
| select firstname, substr(firstname, 1, 5), left(firstname, 5) from n2employee; |
|  |
| 1. Print *salary* of all employees in the given format 3000\*\*\*\*\* for the current job. |
| select rpad(salary, 9, "\*") from n2salary where (employeeid, todate) in (select employeeid, max(todate) from n2salary group by employeeid); |
|  |
| 1. Get all employee who were hired in the month of ‘October’. |
| select \* from n2employee where date\_format(hiredate, '%M') = 'October'; |
|  |
| 1. Get all employee who were hired in the month of ‘December’ and gender is ‘M’. |
| select \* from n2employee where date\_format(hiredate,'%M') = 'December' and gender = 'm'; |
|  |
| 1. Get all employees who were hired on ‘Sunday’ |
| select \* from n2employee where date\_format(hiredate, '%W') = 'Sunday' |
|  |
| 1. Print current date and time. |
| select now() ; |
|  |
| 1. Extract month from the current date. |
| select now(), extract(month from now()); |
|  |
| 1. Extract year from the current date. |
| select now(), extract(year from now()); |
|  |
| 1. Get all employees who were hired in the year 1964 in ascending order of *employee id*. |
| select \* from n2employee where extract(year from hiredate) = 1964 order by id; |
|  |
| 1. Get all employees who were hired in the 4 quarter of a year. |
| select \* from n2employee where extract(quarter from hiredate) = 4; |
|  |
| 1. Get all employees who were hired in the 43rd week of a year. |
| select \* from n2employee where extract(week from hiredate) = 43; |
|  |
| 1. Get all employees who were hired between 10 and 19 day. |
| select \* from n2employee where extract(day from hiredate) between 10 and 19; |
|  |
| 1. Count how many employees where hired in the year 1964. |
| select count(\*) from n2employee where extract(year from hiredate) = 1964; |
|  |
| 1. Generate the random number between 1 to 100 |
| select round(rand() \* 100); |
|  |

Assignment

Feb19/ DBT/ 007

Database Technologies

Diploma in Advance Computing

February 2019

**DML commands: Select data with WHERE, GROUP BY, HAVING, ORDER BY and LIMIT clause.**

USE *n2employee, n2department, n2employee\_department, n2salary, n2commission, n2contact, n2address, n2qualification, n2hobbies, n2order,* and *n2jobhistory*relation to solve the following queries.

|  |
| --- |
| 1. List all employees. |
| select \* from n2employee; |
|  |
| 1. List *FIRSTNAME*, *LASTNAME* of all employees. |
| select firstname, lastname from n2employee; |
|  |
| 1. Display employee information of the employee *ID* is 10. |
| select \* from n2employee where id = 10; |
|  |
| 1. List of various department available from the *n2department* relation. |
| select \* from n2department; |
|  |
| 1. List all employees having ‘A’ as second letter in their *FIRSTNAME*. |
| select \* from n2employee where firstname like 'A%'; |
|  |
| 1. List *ID,* *FIRSTNAME*, *LASTNAME*, and *GENDER* whose *GENDER* is ‘M’. |
| select firstname, lastname, gender from n2employee where gender = 'M'; |
|  |
| 1. Display the details of the employees who have joined on '1964-10-25'. |
| select \* from n2employee where hiredate = '1964-10-25'; |
|  |
| 1. List all employees having ‘R’ as first letter in their *FIRSTNAME*. |
| select \* from n2employee where firstname like 'R%'; |
|  |
| 1. Display the *FIRSTNAME*, *LASTNAME* from n2employee relation with Customized column headings. |
| select firstname as "First Name", lastname as "Last Name" from n2employee; |
|  |
| 1. List all employees whose *GENDER* is ‘F’. |
| select \* from n2employee where gender = 'F'; |
|  |
| 1. List the employee *ID* from *n2hobbies* relation whose hobby is ‘Swimming’ |
| select employeeid from n2hobbies where name='swimming'; |
|  |
| 1. Get all salary details with previous salary for employee *ID* 1 and 10. |
| select \* from n2salary where employeeid = 1 or employeeid = 10; |
|  |
| 1. Get *ID*, *EMPLOYEEID*, *NUMBER*, and *EMAILID* from *n2contact* whosemobile *number* starts with ‘99’. |
| select id, employeeid, phonenumber, emailid from n2contact where phonenumber like '99%'; |
|  |
| 1. List all employees who had joined the organization on '1964-10-25'. |
| select \* from n2employee where hiredate = '1964-10-25'; |
|  |
| 1. List all employees who had joined the organization before '1964-10-25'. |
| select \* from n2employee where hiredate < '1964-10-25'; |
|  |
| 1. List all employees who had joined the organization before '1964-10-25' and whose *GENDER* is ‘F’. |
| select \* from n2employee where hiredate < '1964-10-25' and gender = 'F'; |
|  |
| 1. Display *ID, FIRSTNAME, LASTNAME,* and *HIREDATE* of employee whose *LASTNAME* is ‘Ross’. |
| select id, firstname, lastname, hiredate from n2employee where lastname = 'ross'; |
|  |
| 1. Display employee details whose *FIRSTNAME* is 'Alexander'. |
| select \* from n2employee where firstname = 'Alexander'; |
|  |
| 1. Display employee details having employee *ID* is 1, 8, and 9. |
| select \* from n2employee where id = 1 or id=8 or id=9; |
|  |
| 1. Display employee details whose *FIRSTNAME* starting with letter 'D'. |
| select \* from n2employee where firstname like 'd%'; |
|  |
| 1. Display employee details whose *FIRSTNAME* ending with letter 'N'. |
| select \* from n2employee where firstname like '%n'; |
|  |
| 1. Display employee details whose *FIRSTNAME* starting with letter 'D' and ending with letter 'D'. |
| select \* from n2employee where firstname like 'D%D'; |
|  |
| 1. Display employee details whose *FIRSTNAME* ‘S second letter is 'A'. |
| select \* from n2employee where firstname like '\_A%'; |
|  |
| 1. Display the qualification details from *n2qualification* relation whose employee *ID* is 10, 12 and 14. |
| select \* from n2qualification where employeeid = 10 or employeeid = 12 or employeeid = 14; |
|  |
| 1. Display *EMPLOYEEID, NAME, ADDMISSIONYEAR, INSTITUTE, UNIVERSITY, YEAROFPASSING, PERCENTAGE,* and *GRADE* whose employee is between 10 to 15. |
| select employeeid, name, addmissionyear, institute, university, yearofpassing, percentage, grade from n2qualification limit 9, 10 |
|  |
| 1. Display *EMPLOYEEID, NAME, ADDMISSIONYEAR, INSTITUTE, UNIVERSITY, YEAROFPASSING, PERCENTAGE,* and *GRADE* who have studied in “Stanford University” university. |
| select employeeid, name, addmissionyear, institute, university, yearofpassing, percentage, grade from n2qualification where university = 'Stanford university'; |
|  |
| 1. Display *EMPLOYEEID, NAME, ADDMISSIONYEAR, INSTITUTE, UNIVERSITY, YEAROFPASSING, PERCENTAGE,* and *GRADE* who has done “BE” from “Yale University”. |
| select employeeid, name, addmissionyear, institute, university,  yearofpassing, percentage, grade from n2qualification where name ='be' and university = 'yale university'; |
|  |
| 1. Display *EMPLOYEEID, NAME, ADDMISSIONYEAR, INSTITUTE, UNIVERSITY, YEAROFPASSING, PERCENTAGE,* and *GRADE* whose *PERCENTAGE* is more than 60 and done ‘BE’. |
| select employeeid, name, addmissionyear, institute, university,  yearofpassing, percentage, grade from n2qualification where name ='be' and percentage >60; |
|  |
| 1. Display the hobby details from *n2hobbies* whose employee *ID* is 5 and 10. |
| select \* from n2hobbies where employeeid = 5 or employeeid = 10; |
|  |
| 1. Display employee *ID* whose hobby *NAME* is “Running” |
| select employeeid from n2hobbies where name = 'running'; |
|  |
| 1. Display *grade* from n2qualification table in ascending order of grade. (i.e. A++, A, B++, B, C, D, F). |
| select grade from n2qualification order by if(grade='A+',1, if(grade="A", 2, if(grade="B+", 3, if(grade="B", 4, if(grade="C", 5, if(grade="D" ,6, 7)))))); |
|  |
| 1. Display *phone number* and *email*-*id* of the employeeid 10. |
| select phonenumber, emailid from n2contact where employeeid = 10; |
|  |
| 1. Display all employeeid, phone number, and email-id of all employees whose phone number starts with 9. |
| select employeeid, phonenumber, emailid from n2contact where phonenumber like '9%'; |
|  |
| 1. Display all address (*from n2address relation*) details, who are living in ‘Las Vega’ city. |
| select \* from n2address where city='Las Vega'; |
|  |
| 1. Display all qualification (*from* *n2qualification relation*) details who have done ‘BE’ and his/her year of passing is 1964. |
| select \* from n2qualification where name = 'BE' and yearofpassing = 1964; |
|  |
| 1. Display all education (*from* *n2qualification relation*) detail who have studied in 'Yale University' |
| select \* from n2qualification where university = 'Yale University'; |
|  |
| 1. Display all education (*from* *n2qualification relation*) detail who have studied in 'University of Chicago' and has done ‘BE’. |
| select \* from n2qualification where university = ' University of Chicago' and name='BE'; |
|  |
| 1. Display all hobby name (*from* *n2hobbies relation*) for the employee 21. |
| select \* name n2hobbies where employeeid=21; |
|  |
| 1. Display employeeid (*from* *n2hobbies relation*), whose hobby is playing ‘Football’. |
| select employeeid from n2hobbies where name = 'Football'; |
|  |
| 1. Display firstName, lastName, gender, and hiredate (*from n2employee relation*) whose firstname starts with the letter ‘S’ and gender is ‘F’. |
| select firstName, lastName, gender, hiredate from n2employee where firstname like 'S%' and gender='F'; |
|  |
| 1. Get employee details in descending order of gender and firstname. |
| select \* from n2employee order by gender, firstname; |
|  |

Assignment

Feb19/ DBT/ 008

Database Technologies

Diploma in Advance Computing

February 2019

**Joins**

USE *n2employee, n2department, n2employee\_department, n2salary, n2commission, n2contact, n2address, n2qualification, n2hobbies, n2order, and n2jobhistory*relation to solve the following queries.

|  |
| --- |
| 1. Display *(firstname, lastname, gender, line1, line2, city,* and *pin)* from *n2employee* and *n2address* relation. |
| select firstname, lastname, gender, line1, line2, city, pin from n2employee e, n2address a where e.id = a.employeeid; |
|  |
| 1. Display *(firstname, lastname, gender, name, and institute)* name from *n2employee* and *n2qualification* relations. |
| select firstname, lastname, gender, name, institute from n2employee e, n2qualification q where e.id = q.employeeid; |
|  |
| 1. Display *(firstname, lastname, institute, and university)* who have studied in 'Yale University'. *(Use n2employee, and n2qualification relation)* |
| select firstname, lastname, institute, university from n2employee e, n2qualification q where e.id = q.employeeid and university='Yale University'; |
|  |
| 1. Display all employee's with their hobby detail whose hobby *NAME* is 'Swimming'. *(Use n2employee, and n2hobbies relation)* |
| select \* from n2employee e, n2hobbies h where e.id = h.employeeid; |
|  |
| 1. Display *(employeeid, name, stream institute, university, and grade)* whose stream is ‘Commerce’. *(Use n2employee, and n2qualification relation)* |
| select employeeid, name, stream institute, university, grade from n2employee e, n2qualification q where e.id = q.employeeid and stream ='commerce'; |
|  |
| 1. Display the hobby detail of all employees who have hobby same as ‘SMITH’ (lastName). |
| select \* from n2hobbies h2 where exists (select name from n2hobbies h, n2employee e where e.id=h.employeeid and lastname='smith' and h2.name=h.name); |
|  |
| 1. Display *(firstname, lastname, gender, line1, line2, and city)* of all employees who are staying in “Las Vega” *city*. |
| select firstname, lastname, gender,line1, line2,city from n2employee e, n2address a where e.id = a.employeeid and city = "Las Vega"; |
|  |
| 1. Display *(department name)* who are currently working in departments 10 or 20. |
| select employeeid, departmentid, todate, d.name from n2employee\_department e1, n2department d where e1.departmentid = d.id and todate in (select max(todate) from n2employee\_department group by employeeid) and departmentid in(10, 20); |
|  |
| 1. Display *(employee id, firstname, lastname, department name, location, fromdate, and todate)* of employeeid 10. |
| select e.id, e.firstname, e.lastname, d.name, d.location, e1.fromdate, e1.todate from n2employee e, n2department d, n2employee\_department e1 where e.id=e1.employeeid and e1.departmentid = d.id and e.id=10; |
|  |
| 1. Display *(firstname, lastname, phonenumber, and emailid)* of employeeid 14. |
| select firstname, lastname, phonenumber, emailid from n2employee e, n2contact c where e.id = c.employeeid and e.id=14; |
|  |
| 1. Display *(firstname and count the total number of phone an employee is having)* for all employees. |
| select firstname, count(\*) "R1" from n2employee e, n2contact c where e.id = c.employeeid group by firstname, c.employeeid; |
|  |
| 1. Get employee’s *(firstname, lastname,* *gender, phonenumber, and emailid)* whose employeeid is 14. |
| select firstname, lastname, gender, phonenumber, emailid from n2employee e, n2contact c where e.id=c.employeeid and e.id=14; |
|  |
| 1. Get *(firstname, lastname, gender, and all department details)* of the employee 21. |
| select firstname, lastname, gender, d.name, d.location, e1.fromdate, e1.todate from n2employee e, n2department d, n2employee\_department e1 where e.id = e1.employeeid and e1.departmentid = d.id and e1.employeeid=21; |
|  |
| 1. Get *(employee id, firstname, lastname, gender, and all hobby name)* for all employees. |
| select e.id, firstname, lastname, gender, h.name from n2employee e, n2hobbies h where e.id = h.employeeid; |
|  |
| 1. Get highest salary of the current employee. |
| select max(salary) from n2salary where (employeeid, todate) in (select employeeid, max(todate) from n2salary group by employeeid); |
|  |
| 1. Display employee details and his job history details for all employees. |
| select \* from n2employee e, n2jobhistory j where e.id = j.employeeid; |
|  |
| 1. Display *(employee id, firstname, lastname, gender, his previous employeer, fromdate, and todate)* for the employee whose id is 20. |
| select id, firstname, lastname, gender, employeer, fromdate, todate from n2employee e, n2jobhistory j where e.id = j.employeeid and e.id=20; |
|  |
| 1. Display *(employee id, firstname, lastname, gender, his/her employeer, fromdate, and todate)* who had previously worked under ‘leena’ |
| select e.id, e.firstname, e.lastname, e.gender, j.employeer, j.fromdate, j.todate from n2employee e, n2jobhistory j where e.id = j.employeeid and employeer = 'leena'; |
|  |
| 1. Get the first name, last name, department number and department name, for all employees for current department ID is 10 or 70. |
| select \* from n2employee e, n2department d where (e.id, d.id) in (select employeeid, departmentid from n2employee\_department where (employeeid, todate) in (select employeeid, max(todate) from n2employee\_department group by employeeid) and departmentid in (10, 70)); |
|  |
| 1. Get all department details where no employees are working. |
| select \* from n2department d where d.id not in (select departmentid from n2employee\_department where (employeeid, todate) in (select employeeid, max(todate) from n2employee\_department group by employeeid)); |
|  |
| 1. Get employee firstname and phone no employee id is 7. |
| select firstname, phonenumber from n2employee e, n2contact c where e.id = c.employeeid and e.id=7; |
|  |
| 1. Get employee details with hobbies. |
| select \* from n2employee e, n2hobbies h where e.id = h.employeeid; |
|  |
| 1. Get the list of employees having hobby is ‘Running’ |
| select \* from n2employee e, n2hobbies h where e.id = h.employeeid and name = 'Running'; |
|  |
| 1. Display all employee and address details who are staying in ‘New York’ city. |
| select \* from n2employee e, n2address a where e.id = a.employeeid and a.city='New York'; |
|  |
| 1. Display *employeeid, phonenumber, emailid, line1, line2*, and *city* (*from n2contact and n2address relation*). |
| select c.employeeid, phonenumber, emailid,line1, line2,city from n2contact c, n2address d where c.employeeid = d.employeeid order by c.employeeid; |
|  |
| 1. Display *firstname, lastname, gender, line1, line2, city,* and *pin* whose *pin* starts with 3. |
| select firstname, lastname, gender, line1, line2, city, pin from n2employee e, n2address a where e.id = a.employeeid and pin like '3%'; |
|  |
| 1. Get all employee details whose qualification is ‘M.Com.’. |
| select e.\* from n2employee e, n2qualification q where e.id = q.employeeid and name='M.Com.'; |
|  |
| 1. Display all employee, qualification name, and grade whose grade is ‘A’ and has done ‘BE’. |
| select e.\*,name, grade from n2employee e, n2qualification q where e.id = q.employeeid and grade = 'A' and name='BE'; |
|  |
| 1. Display all employee details and his qualification name, and stream who have done ‘arts’. |
| select e.\*,name, grade, stream from n2employee e, n2qualification q where e.id = q.employeeid and stream ='arts'; |
|  |
| 1. Display all employee details and his qualification name, and stream who have done ‘arts’ is 12th. |
| select e.\*,name, grade, stream from n2employee e, n2qualification q where e.id = q.employeeid and stream ='arts' and name=12; |
|  |

Assignment

Feb19/ DBT/ 009

Database Technologies

Diploma in Advance Computing

February 2019

**Aggregate Functions.**

USE *n2employee, n2department, n2employee\_department, n2salary, n2commission, n2contact, n2address, n2qualification, n2hobbies, n2order, and n2jobhistory*relation to solve the following queries.

|  |
| --- |
| 1. Count total number of employees. |
| select count(\*) from n2employee; |
|  |
| 1. Count total number of female employees. |
| select count(\*) from n2employee where gender = 'F'; |
|  |
| 1. Count total number of female employees whose firstname starts with the letter ‘F’. |
| select count(\*) from n2employee where gender = 'F' and firstname like 'B%'; |
|  |
| 1. Count total number employee who were hired in the year 1962. |
| select count(\*) from n2employee where date\_format(hiredate, '%Y') = 1962; |
|  |
| 1. Count how many phonenumber an employeeid 3 is having. |
| select count(\*) from n2contact where employeeid=3; |
|  |
| 1. Count number of hobbies every employee is having. |
| select count(\*) from n2hobbies group by employeeid; |
|  |
| 1. Count total number of unique hobbies. |
| select count(distinct name) from n2hobbies; |
|  |
| 1. Count how many employees has done ‘BE’. |
| select count(\*) from n2qualification where name = 'BE'; |
|  |
| 1. Stream wise count of employees who have taken admission in ‘BE’. |
| select stream, count(\*) from n2qualification where name = 'BE' group by stream; |
|  |
| 1. Stream wise count of employees who have taken admission in ‘BE’ and have secured ‘A’ grade. |
| select stream, count(\*) from n2qualification where name='BE' and grade='A' group by stream; |
|  |
| 1. Count unique cities from n2address relation. |
| select count(distinctrow city) from n2address; |
|  |
| 1. Count how many employee are staying in ‘Pune’ city. |
| select count(\*) from n2address where city = 'pune'; |
|  |
| 1. Count the number of employee who have more than 60% in ‘BE’. |
| select count(\*) from n2qualification where percentage > 60 and name = 'BE'; |
|  |
| 1. Stream wise count of employee who have more than 60% in ‘BE’. |
| select stream, count(\*) from n2qualification where percentage > 60 and name = 'BE' group by stream; |
|  |
| 1. Count how many employees are from ‘PUNE’ city. (*use n2address relation*) |
| select count(\*) from n2address where city = 'PUNE'; |
|  |

Assignment

Feb19/ DBT/ 011

Database Technologies

Diploma in Advance Computing

February 2019

**Sub-queries**

USE *n2employee, n2department, n2employee\_department, n2salary, n2commission, n2contact, n2address, n2qualification, n2hobbies, n2order, and n2jobhistory*relation to solve the following queries.

|  |
| --- |
| 1. Display all employee and hobby details of those employees who have more than or equal to 4 hobbies. |
| select \* from n2employee e, n2hobbies h where e.id = h.employeeid and e.id in (select employeeid from n2hobbies group by employeeid having count(\*) >=4); |
|  |
| 1. Display the employee’s detail who have least hobbies. |
| select \* from n2employee where id = (select employeeid from n2hobbies group by employeeid having count(\*) = (select min(r1) from (select count(\*) r1 from n2hobbies group by employeeid) e)); |
|  |
| 1. Display all employee’s detail who have least hobbies for *gender* ‘M’. |
| select \* from n2employee where id in (select employeeid from n2hobbies group by employeeid having count(\*) in (select min(r1) from (select employeeid, count(\*) r1 from n2employee e, n2hobbies h where e.id = h.employeeid and gender = 'M' group by employeeid) e1)) and gender='M'; |
|  |
| 1. Display *(firstname, lastname)* who are having more than 4 hobbies. |
| select firstname, lastname from n2employee e where exists (select h.employeeid from n2hobbies h where e.id = h.employeeid group by h.employeeid having count(\*)>4); |
|  |
| 1. Get all department details where no employees are working. |
| select \* from n2department d where not exists (select distinct departmentid from n2employee\_department where (employeeid, todate) in (select employeeid, max(todate) from n2employee\_department group by employeeid) and d.id = n2employee\_department.departmentid); |
|  |
| 1. Get all department details where employees are working. |
| select \* from n2department d where exists (select distinct departmentid from n2employee\_department where (employeeid, todate) in (select employeeid, max(todate) from n2employee\_department group by employeeid) and d.id = n2employee\_department.departmentid); |
|  |
| 1. Get the *(employeeid,* and *salary)* for those employees who earn less than the employee earn whose *employeeID* is 10. |
| select employeeid ,salary from n2salary where (employeeid, todate) in (select employeeid, max(todate) from n2salary group by employeeid) and salary < (select salary from n2salary where (todate, employeeid) = (select max(todate), employeeid from n2salary where employeeid=10)); |
|  |
| 1. Display all salary details who are having same *salaries* for the current jobs. |
| select \* from (select \* from n2salary where (employeeid ,todate) in (select employeeid, max(todate) from n2salary group by employeeid)) e1, (select salary from n2salary where (employeeid, todate) in (select employeeid, max(todate) from n2salary group by employeeid) group by salary having count(\*)>1) e2 where e1.salary = e2.salary order by 5; |
|  |
| 1. Display *(salary, and count of salaries)* of all employees who same salary for the current job. |
| select salary, count(\*) from n2salary where (employeeid, todate) in (select employeeid, max(todate) from n2salary group by employeeid) group by salary having count(\*) > 1; |
|  |
| 1. Get all employees whose employer is either 'sharmin' or 'saleel' |
| select \* from n2employee e, n2jobhistory j where e.id=j.employeeid and employeer in ('sharmin', 'saleel') and (employeeid, todate) in (select employeeid, max(todate) from n2jobhistory group by employeeid); |
|  |
| 1. Get salary details of the current employees, whose salary is below 2500. |
| select \* from n2salary where (employeeid, todate) in (select employeeid, max(todate) from n2salary group by employeeid) and salary < 2500 order by employeeid; |
|  |
| 1. Display all employees who are getting commission. |
| select e.\* from n2employee e where exists (select \* from n2commission c where e.id = c.employeeid); |
|  |
| 1. Display all employees who are not getting commission. |
| select e.\* from n2employee e where not exists (select \* from n2commission c where e.id = c.employeeid); |
|  |
| 1. Display all employees’ commission details for the currentjob who are not getting commission. |
| select \* from n2commission where (employeeid, todate) in (select employeeid, max(todate) from n2commission group by employeeid) and commission is null; |
|  |
| 1. Display all employees’ commission details for the currentjob who are getting commission. |
| select \* from n2commission where (employeeid, todate) in (select employeeid, max(todate) from n2commission group by employeeid) and commission is not null; |
|  |
| 1. Display all employee details who have three or more emailid. |
| select \* from n2employee where id in (select employeeid from n2contact group by employeeid having count(emailid)>=3); |
|  |
| 1. Get lowest salary of employee working in current job? |
| select min(salary) from n2salary where (employeeid, todate) in (select employeeid, max(todate) from n2salary group by employeeid); |
|  |
| 1. Get lowest salary of employee working in current job? |
| select min(salary) from n2salary where (employeeid, todate) in (select employeeid, max(todate) from n2salary group by employeeid); |
|  |
| 1. Get average salary of employee working in current job? |
| select avg(salary) from n2salary where (employeeid, todate) in (select employeeid, max(todate) from n2salary group by employeeid); |
|  |
| 1. Get sum salary of employee working in current job? |
| select sum(salary) from n2salary where (employeeid, todate) in (select employeeid, max(todate) from n2salary group by employeeid); |
|  |
| 1. Get highest *commission* of employee for the current job. |
| select max(commission) from n2commission where (employeeid, todate) in (select employeeid, max(todate) from n2commission group by employeeid); |
|  |
| 1. Get second highest *salary* of employee for the current job. |
| select max(commission) from n2commission where (employeeid, todate) in (select employeeid, max(todate) from n2commission group by employeeid) and commission < (select max(commission) from n2commission where (employeeid, todate) in (select employeeid, max(todate) from n2commission group by employeeid)); |
|  |
| 1. Print *salary* of all employees in the given format 3000\*\*\*\*\* for the current job. |
| select rpad(salary, 9, "\*") from n2salary where (employeeid, todate) in (select employeeid, max(todate) from n2salary group by employeeid); |
|  |
| 1. Get the *(department name, and count)*, than how many employees are working in which department for the current job. |
| select d.name, count(\*) from n2employee e, n2department d, n2employee\_department e1 where e.id=e1.employeeid and e1.departmentid = d.id and (e1.employeeid, todate) in (select employeeid, max(todate) from n2employee\_department group by employeeid) group by d.name; |
|  |
| 1. Display the name of stream where maximum number of employees has taken admission in ‘BE’. |
| select stream from n2qualification where name='BE' group by stream having count(\*) = (select max(R1) from (select count(\*) R1 from n2qualification where name = 'BE' group by stream) s ); |
|  |
| 1. Count how many employee are not getting commission for the current job. |
| select count(\*) from n2commission where (employeeid, todate)in (select employeeid, max(todate) from n2commission group by employeeid) and commission is null; |
|  |
| 1. Count how many employee are getting commission for the current job. |
| select count(\*) from n2commission where (employeeid, todate)in (select employeeid, max(todate) from n2commission group by employeeid) and commission is not null; |
|  |
| 1. Display all commission details of those employees who are not getting commission for the current job. |
| select \* from n2commission where todate in (select max(todate) from n2commission group by employeeid) and commission is null; |
|  |

Section 3

Assignment

Feb19/ DBT/125

Database Technologies

Diploma in Advance Computing

February 2019

**Basic Programming**

1. Write a basic PL/SQL programme to create two variables and store some default value and print them.
2. Write a simple procedure to print ‘Hello World’
3. Write a simple procedure to print a table of a given number?
4. Write a procedure to print the maximum number of 3 inputted numbers.

Assignment

Feb19/ DBT/126

Database Technologies

Diploma in Advance Computing

February 2019

**Procedure**

|  |
| --- |
| 1. Create a LOGIN table (username, password, and email). Write a procedure (named ***addUser***) to pass the username, password, and email-ID through the procedure and store the data in the LOGIN table. |
| create table login (username varchar(20), password varchar(20), email varchar(20));  drop procedure if exists addUser;  delimiter $$  create procedure addUser (userName varchar(20), password varchar(20), email varchar(20))  begin  insert into login values(userName, password, email);  Select "Record inserted..." as Message;  end$$  delimiter ; |
|  |
| 1. Create a LOG table (id (auto\_increment), curr\_date, curr\_time, and message). Write a procedure (named ***checkUser***) to pass the email-ID as an input, check whether passed email-ID is available in LOGIN table or not available. If the email-ID is available then display the username and his password. If the email-ID is not available then, insert (id, curr\_date, curr\_time, and message) in LOG table. |
| create table log (id int, curr\_date date, curr\_time time, message varchar(20));  drop procedure if exists checkUser;  delimiter $$  create procedure checkUser (\_email varchar(20))  begin  declare \_username varchar(20);  declare \_password varchar(20);  select username, password into \_username, \_password from login where email = \_email;    if \_username is not null and \_password is not null then  select \_username, \_password;  else  Insert into log (curr\_date, curr\_time, message) values ((select current\_date), (select current\_time), "Record not found...");  end if;  end$$  delimiter ; |
|  |
| 1. Write a procedure (named getQualification) that takes employeeID as a parameter. If employeeID is present in the N2EMPLOYEE table, then print his employee details along with N2QUALIFICATION details and if the employeeID is not present display message “Employee not found…” (Use: N2EMPLOYEE, and N2QUALIFICATION tables) |
| drop procedure if exists getQualification;  delimiter $$  create procedure getQualification (\_id int)  begin  declare id int;  select distinct e.id into id from n2employee e, n2qualification q where e.id = q.employeeid and e.id = \_id;    if id is not null then  select \* from n2employee e, n2qualification q where e.id = q.employeeid and e.id = \_id;  else  select "Employee not found…";  end if;  end$$  delimiter ; |
|  |
| 1. Write a procedure (named addDepartment) that inserts a new department into the N2DEPARTMENT table. (Use: N2DEPARTMENT table) |
| drop procedure if exists addDepartment;  delimiter $$  create procedure addDepartment (\_id int, \_name varchar(20), \_location varchar(20))  begin  declare id int;  insert into n2department values(\_id, \_name, \_location);  end$$  delimiter ; |
|  |
| 1. Write a procedure (named addQualification) that takes employeeID, and qualification details as a parameter. If employeeID is present in the N2EMPLOYEE table, then insert the qualification in N2QUALIFICATION table and return a message “Record inserted” or else print ‘Employee not found’. (hint: using OUT parameter) (Use: N2EMPLOYEE, and N2QUALIFICATION tables) |
| drop procedure if exists addQualification;  delimiter $$  create procedure addQualification (\_EMPLOYEEID int, \_ID int, \_NAME varchar(25), \_Stream varchar(25), \_ADDMISSIONYEAR date, \_INSTITUTE varchar(25), \_UNIVERSITY varchar(25), \_YEAROFPASSING int, \_PERCENTAGE float, \_GRADE varchar(10), out \_message varchar(100))  begin  declare x int;  select id into x from n2employee where id = \_EMPLOYEEID;  if x is not null then  insert into n2qualification values (\_ID, \_EMPLOYEEID, \_NAME, \_Stream, \_ADDMISSIONYEAR, \_INSTITUTE, \_UNIVERSITY, \_YEAROFPASSING, \_PERCENTAGE, \_GRADE);  select "Record inserted...";  else  set \_message := "Employee not found";  end if;  end$$  delimiter ; |
|  |

Assignment

Feb19/ DBT/127

Database Technologies

Diploma in Advance Computing

February 2019

**Function**

|  |
| --- |
| 1. Pass employeeID to the function (named sumSalary) and calculate the sum of salary till date.(Use: N2SALARY table) |
| drop function if exists sumSalary;  delimiter $$  create function sumSalary(\_employeeID int) returns int  begin  declare x int;  select sum(salary) into x from n2salary where employeeid = \_employeeID;  return(x);  end$$  delimiter ; |
|  |
| 1. Write a function to return auto generate deptno and return the new value (Use: N2DEPARTMENT table). |
| drop function if exists autoNumber;  delimiter $$  create function autoNumber() returns int  begin  declare newNumber int;  select max(id) + 1 into newNumber from n2department;  return(newNumber);  end$$  delimiter ; |
|  |
| 1. Write a function which will accept email-ID from the user, if the email-ID is present return his username, and password or else `Return “Employee not exists”. (Use: LOGIN table) |
| drop function if exists checkUser;  delimiter $$  create function checkUser(\_email varchar(20)) returns varchar(1000)  begin  declare \_userName varchar(20);  declare \_password varchar(20);  select userName, password into \_userName, \_password from login where email = \_email;  if \_userName is not null and \_password is not null then  return(concat(\_userName, ' ', \_password));  else  return("Employee not exists");  end if;  end$$  delimiter ; |
|  |

Section 4

Assignment

Feb19/ DBT/128

Database Technologies

Diploma in Advance Computing

February 2019

**Trigger**

|  |
| --- |
| 1. Write a trigger (named insertDepartment) that saves the message "Record inserted successfully" in LOG table as soon as you insert the record in N2DEPARTMENT table. |
| drop trigger if exists insertDepartment;  delimiter $$  create trigger insertDepartment before insert on n2department for each row  begin    insert into log(curr\_date, curr\_time, message) values (current\_date, current\_time, 'Record Inserted');  end$$  delimiter ; |
|  |
| 1. Write a trigger (named insertDuplicate) on N2EMPLOYEE table, that as when we INSERT a record in N2EMPLOYEE table the same record should get duplicated (INSERTED) in EMP\_LOG table. (Create EMP\_LOG table, having the same structure as N2EMPLOYEE table). |
| drop trigger if exists insertDuplicate;  delimiter $$  create trigger insertDuplicate before insert on n2employee for each row  begin    insert into emp\_log (ID, FIRSTNAME, LASTNAME, GENDER, HIREDATE) values(new.ID, new.FIRSTNAME, new.LASTNAME, new.GENDER, new.HIREDATE);  end$$  delimiter ; |
|  |
| 1. Write a trigger on N2EMPLOYEE table, that as soon as we UPDATE any column data in N2EMPLOYEE table, the update record should get inserted in EMP\_LOG table. |
| drop trigger if exists updateemployee;  delimiter $$  create trigger updateemployee before update on n2employee for each row  begin  insert into emp\_log (id, firstname, lastname, gender, hiredate) values(old.id, old.firstname, old.lastname, old.gender, old.hiredate);  end$$  delimiter ; |
|  |
| 1. Write a trigger on N2EMPLOYEE table, that as soon as we DELETE any record from N2EMPLOYEE table, then that record should get inserted into EMP\_LOG table. |
| drop trigger if exists deleteemployee;  delimiter $$  create trigger deleteemployee before delete on n2employee for each row  begin  insert into e(id, firstname, lastname, gender, hiredate) values(old.id, old.firstname, old.lastname, old.gender, old.hiredate);  end$$  delimiter ; |
|  |
| 1. Write a trigger on N2EMPLOYEE table, that if today is Sunday then, no record should get inserted in EMP table. |
| drop trigger if exists insertValidation;  delimiter $$  create trigger insertValidation after insert on n2employee for each row  begin  if DATE\_FORMAT (now(), '%W') = 'Tuesday' then  signal sqlstate '42000' set message\_text = 'Record cannot be inserted';  end if;  end$$  delimiter ; |
|  |

Section 5

Assignment

Feb19/ DBT/ 129

Database Technologies

Diploma in Advance Computing

February 2019

**MongoDB**

USE ***EMP***collection.

|  |
| --- |
| 1. Display all databases. |
| show dbs;  show databases; |
|  |
| 1. Display the current database. |
| db  db.getName(); |
|  |
| 1. Display all collection. |
| db.getCollectionNames(); |
|  |
| 1. Display the current version of MongoDB. |
| version(); |
|  |
| 1. Display the current host details. |
| hostname(); |
|  |
| 1. Get the current IP address and the PORT number. |
| db.getMongo(); |
|  |
| 1. Display all documents from EMP collection. |
| db.emp.find();  db.getCollection('emp').find(); |
|  |
| 1. Display all documents from EMP collection in JSON format. |
| db.emp.find().pretty();  db.emp.find().forEach(printjson) |
|  |
| 1. Display first 5 documents from EMP collection. |
| db.emp.find().limit(5); |
|  |
| 1. Display employee name, and his address from EMP collection. |
| db.emp.find({}, { ename: true, address: true }) |
|  |
| 1. Display all building and coord details of all employee from EMP collection. |
| db.getCollection('emp').find({}, { \_id: false, "address.building": true, "address.coord": true }); |
|  |
| 1. Display all documents who are staying in building number “2780”. |
| db.getCollection('emp').find({"address.building":"2780"}); |
|  |
| 1. Display all female employee documents. |
| db.getCollection('emp').find({ gender: "female" }); |
|  |
| 1. Display all employee working in department number 40. |
| db.getCollection('emp').find({ deptno: 40 }); |
|  |
| 1. Enter 5 documents in EMP collection in the following format.   empid:number, ename:str, address:{ building:str,"coord" : [number,number], street:str, zipcode:number }, isActive : bool, gender:char, canVote:bool, canDrive:bool, favouriteColor[,..], favouriteFruit[,..], aadhar:str, job:str, mgr:number, hiredate:date, sal:number, comm:number, deptno:number |
| db.emp.insertMany( [ {}, {}, {}, {}, {} ] ) |
|  |
| 1. Count total documents in EMP collection. |
| db.emp.countDocuments({}) |
|  |
| 1. Display ename, sal, comm fields from the collection, who are getting some comm. |
| db.getCollection('emp').find( {comm: { $ne: null} }, {ename: true, sal: true, comm: true}) |
|  |
| 1. Count the documents of ‘Computer Programmer’ |
| db.emp.countDocuments({ job: "Computer Programmer" } ) |
|  |
| 1. Display ename, job, and salary fields from EMP collection in ascending order of ename. |
| db.emp.find({}, { ename: true, job: true, sal: true }).sort( { ename: 1 } ) |
|  |
| 1. Display all documents between 5 and 10. |
| db.emp.find().skip(5).limit(5) |
|  |