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. |
|  |
|  |
| 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*). |
|  |
|  |
| 1. Write a query to create a view ***employee\_commission*** for all employees who are getting commission in their current job. |
|  |
|  |
| 1. Write a query to create a view ***employee\_address*** with columns (*employeeid, firstname, lastname, gender, line1, line2, city,* and *pin*) for all employees. |
|  |
|  |
| 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. |
|  |
|  |
| 1. Create temporary table ***temp\_employee*** alike n2employee relation. |
|  |
|  |

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. |
|  |
|  |
| 1. Display the *firstName*, lastName of all *employees*. |
|  |
|  |
| 1. Get the employee details whose employee *firstName* is ‘Fred’. |
|  |
|  |
| 1. Get all female employees. |
|  |
|  |
| 1. Get all employees whose *hobby1* is ‘Reading’. |
|  |
|  |
| 1. List all employees whose *currentjob* is ‘Software Developer’. |
|  |
|  |
| 1. List all employees whose *PREVIOUSJOB3* is ‘Trainee’. |
|  |
|  |
| 1. List all employees whose *salary* is more than 4000. |
|  |
|  |
| 1. List all employees whose *deptname* is ‘sales’. |
|  |
|  |
| 1. List all employees whose *salary* is less than 3000. |
|  |
|  |
| 1. Get all employees whose *hobby2* is ‘Ice skating’. |
|  |
|  |
| 1. Get all male employees. |
|  |
|  |
| 1. Get all employees whose *lastName* is ‘Clark’. |
|  |
|  |
| 1. List the employee whose *mobileno1* is 7032300039. |
|  |
|  |
| 1. List the employee whose *id* is 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. |
|  |
|  |
| 1. List *firstname*, *lastname* of all employees. |
|  |
|  |
| 1. List *firstname*, *lastname*, *hiredate*, and *salary* all employees. |
|  |
|  |
| 1. Display employee information of the *employee* *ID* is 15. |
|  |
|  |
| 1. List *firstname*, *lastname*, and *currentjob* of employees whose *currentjob* is ‘Sr.Assistant’. |
|  |
|  |
| 1. List all employee having *salary* greater than equal to 3000. |
|  |
|  |
| 1. List *firstname*, *lastname*, *gender*, *mobileno1*, *hobby1*, and *hobby2* whose gender is ‘M’. |
|  |
|  |
| 1. Display the employee *id*, *firstname*, *lastname*, and *currentjob* & Annual Salary for all Employees belonging to Department Name is ‘*SALES’*. |
|  |
|  |
| 1. List all employees who don’t have 2nd mobile. |
|  |
|  |
| 1. List all employees who have *salary* more than 3000Rs. |
|  |
|  |
| 1. List all employees (*firstname*, *lastname*, *gender*, *mobileno1*, *salary*, *and* *deptname*) who are in either ‘ACCOUNTING’ or ‘RESEARCH’ or ‘SALES’ department. |
|  |
|  |
| 1. List all employees whose *currentjob* is either ‘Head Clerk’ or ‘Sr. Analyst’. |
|  |
|  |
| 1. List all employees whose *gender* is ‘F’. |
|  |
|  |
| 1. List all employees whose *hobby1* is ‘Running’. |
|  |
|  |
| 1. Display the *firstname*, *lastname*, *gender*, *mobileno1*, AND *mobileno2* change the column heading of *mobile1* to ‘HOME MOBILE’, and *mobile2* to ‘OFFICE MOBILE’. |
|  |
|  |
| 1. List all employees whose *salary* is in the range of 2000 to 3000. |
|  |
|  |
| 1. Computer Total Salary by adding *salary* and *comm*. (hint: if *comm* is NULL replace *comm* with 0) |
|  |
|  |
| 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*) |
|  |
|  |
| 1. List all employees who had joined the organization on 2018-05-12. (use *hiredate*) |
|  |
|  |
| 1. List all employees of ‘OPERATIONS’ department. |
|  |
|  |
| 1. List all employees who are not receiving the *comm.* |
|  |
|  |
| 1. Display (*currentjob, previousjob1, previousjob2,* and *previousjob3*) whose employee *id* is 10. |
|  |
|  |
| 1. Display all employee who were hired on 1983. |
|  |
|  |
| 1. Display all employee whose salary is more than 4000. |
|  |
|  |
| 1. Display all employee whose salary is more than 4000 and less than 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. |
|  |
|  |
| 1. List *firstname*, *lastname* of all employees in ascending order of *firstname*. |
|  |
|  |
| 1. List *firstname*, *lastname*, *hiredate*, and *salary* for the first 5 employees. |
|  |
|  |
| 1. Display employee information of the employee *ID* is either 1, 2, 5 or 7. |
|  |
|  |
| 1. List *firstname*, *lastname*, and *currentjob* of employees whose *currentjob* is not ‘Sr.Assistant’, display first 7 rows. |
|  |
|  |
| 1. List all employee having *salary* is between to 3000 and 4000 in descending order of *salary*. |
|  |
|  |
| 1. List all employees whose gender is ‘F’, display rows between 5 and 10. |
|  |
|  |
| 1. Display the *id*, *firstname*, *lastname*, and calculate *Annual Salary* for all employees in ascending order of fourth column. |
|  |
|  |
| 1. List all employees who don’t have 2nd mobile. |
|  |
|  |
| 1. List all employees who have hired on '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*. |
|  |
|  |
| 1. List all employees whose *currentjob* is either ‘Head Clerk’ or ‘Sr. Analyst’ and firstname must be either ‘Peter’ or ‘Rosaleen’. |
|  |
|  |
| 1. List all employees whose *gender* is 'F' and *hobby1* is 'Writing'. |
|  |
|  |
| 1. List first 1 rows of all employees whose *hobby1* is 'Running'. |
|  |
|  |
| 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’. |
|  |
|  |
| 1. Display all employees whose *mobileno1* starts with '7'. Display first 5 rows only. |
|  |
|  |

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*. |
|  |
|  |
| 1. Get employee details whose *firstname* is having at least 4 characters. |
|  |
|  |
| 1. Get the ASCII value of the 3rd character of *firstname* column. |
|  |
|  |
| 1. Get *firstname* and *lastname* in lowercase. |
|  |
|  |
| 1. Get *(hobby name)* all 7 letter hobbies. |
|  |
|  |
| 1. Get *(firstname, lastname and first 3 letters of firstname)* for all employees. |
|  |
|  |
| 1. Get *(firstname, lastname and last 3 letters of firstname)* for all employees. |
|  |
|  |
| 1. Get all *(phonenumber)* whose *phonenumber* starts with 99. |
|  |
|  |
| 1. Get employee details of first 5 employees. |
|  |
|  |
| 1. Get employee details of last 5 employees. |
|  |
|  |
| 1. Get employee details in ascending order of *firstname*. |
|  |
|  |
| 1. Get employee details in descending order of *lastname*. |
|  |
|  |
| 1. Get *(employee id, firstname, lastname, gender, phonenumber, and emailid)* for all employees whose length of email id is more than 20 characters. |
|  |
|  |
| 1. Combine to display employee *firstname* and *lastname*. |
|  |
|  |
| 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)** |
|  |
|  |
| 1. Get employee *firstname* and *lastname* in upper case. |
|  |
|  |
| 1. Get employee *firstname* and *lastname* in lower case. |
|  |
|  |
| 1. Get employee *firstname* and *lastname* in reverse order. |
|  |
|  |
| 1. Get first 4 letters of employee *firstname*. |
|  |
|  |
| 1. Get second letter of employee *firstname* to second last letter of employee *firstname*. |
|  |
|  |
| 1. Get ASCII character of employee *firstname*. |
|  |
|  |
| 1. Get 5 letter of the employee *firstname*. |
|  |
|  |
| 1. Print *salary* of all employees in the given format 3000\*\*\*\*\* for the current job. |
|  |
|  |
| 1. Get all employee who were hired in the month of ‘October’. |
|  |
|  |
| 1. Get all employee who were hired in the month of ‘December’ and gender is ‘M’. |
|  |
|  |
| 1. Get all employees who were hired on ‘Sunday’ |
|  |
|  |
| 1. Print current date and time. |
|  |
|  |
| 1. Extract month from the current date. |
|  |
|  |
| 1. Extract year from the current date. |
|  |
|  |
| 1. Get all employees who were hired in the year 1964 in ascending order of *employee id*. |
|  |
|  |
| 1. Get all employees who were hired in the 4 quarter of a year. |
|  |
|  |
| 1. Get all employees who were hired in the 43rd week of a year. |
|  |
|  |
| 1. Get all employees who were hired between 10 and 19 day. |
|  |
|  |
| 1. Count how many employees where hired in the year 1964. |
|  |
|  |
| 1. Generate the random number between 1 to 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. |
|  |
|  |
| 1. List *FIRSTNAME*, *LASTNAME* of all employees. |
|  |
|  |
| 1. Display employee information of the employee *ID* is 10. |
|  |
|  |
| 1. List of various department available from the *n2department* relation. |
|  |
|  |
| 1. List all employees having ‘A’ as second letter in their *FIRSTNAME*. |
|  |
|  |
| 1. List *ID,* *FIRSTNAME*, *LASTNAME*, and *GENDER* whose *GENDER* is ‘M’. |
|  |
|  |
| 1. Display the details of the employees who have joined on '1964-10-25'. |
|  |
|  |
| 1. List all employees having ‘R’ as first letter in their *FIRSTNAME*. |
|  |
|  |
| 1. Display the *FIRSTNAME*, *LASTNAME* from n2employee relation with Customized column headings. |
|  |
|  |
| 1. List all employees whose *GENDER* is ‘F’. |
|  |
|  |
| 1. List the employee *ID* from *n2hobbies* relation whose hobby is ‘Swimming’ |
|  |
|  |
| 1. Get all salary details with previous salary for employee *ID* 1 and 10. |
|  |
|  |
| 1. Get *ID*, *EMPLOYEEID*, *NUMBER*, and *EMAILID* from *n2contact* whosemobile *number* starts with ‘99’. |
|  |
|  |
| 1. List all employees who had joined the organization on '1964-10-25'. |
|  |
|  |
| 1. List all employees who had joined the organization before '1964-10-25'. |
|  |
|  |
| 1. List all employees who had joined the organization before '1964-10-25' and whose *GENDER* is ‘F’. |
|  |
|  |
| 1. Display *ID, FIRSTNAME, LASTNAME,* and *HIREDATE* of employee whose *LASTNAME* is ‘Ross’. |
|  |
|  |
| 1. Display employee details whose *FIRSTNAME* is 'Alexander'. |
|  |
|  |
| 1. Display employee details having employee *ID* is 1, 8, and 9. |
|  |
|  |
| 1. Display employee details whose *FIRSTNAME* starting with letter 'D'. |
|  |
|  |
| 1. Display employee details whose *FIRSTNAME* ending with letter 'N'. |
|  |
|  |
| 1. Display employee details whose *FIRSTNAME* starting with letter 'D' and ending with letter 'D'. |
|  |
|  |
| 1. Display employee details whose *FIRSTNAME* ‘S second letter is 'A'. |
|  |
|  |
| 1. Display the qualification details from *n2qualification* relation whose employee *ID* is 10, 12 and 14. |
|  |
|  |
| 1. Display *EMPLOYEEID, NAME, ADDMISSIONYEAR, INSTITUTE, UNIVERSITY, YEAROFPASSING, PERCENTAGE,* and *GRADE* whose employee is between 10 to 15. |
|  |
|  |
| 1. Display *EMPLOYEEID, NAME, ADDMISSIONYEAR, INSTITUTE, UNIVERSITY, YEAROFPASSING, PERCENTAGE,* and *GRADE* who have studied in “Stanford University” university. |
|  |
|  |
| 1. Display *EMPLOYEEID, NAME, ADDMISSIONYEAR, INSTITUTE, UNIVERSITY, YEAROFPASSING, PERCENTAGE,* and *GRADE* who has done “BE” from “Yale University”. |
|  |
|  |
| 1. Display *EMPLOYEEID, NAME, ADDMISSIONYEAR, INSTITUTE, UNIVERSITY, YEAROFPASSING, PERCENTAGE,* and *GRADE* whose *PERCENTAGE* is more than 60 and done ‘BE’. |
|  |
|  |
| 1. Display the hobby details from *n2hobbies* whose employee *ID* is 5 and 10. |
|  |
|  |
| 1. Display employee *ID* whose hobby *NAME* is “Running” |
|  |
|  |
| 1. Display *grade* from n2qualification table in ascending order of grade. (i.e. A++, A, B++, B, C, D, F). |
|  |
|  |
| 1. Display *phone number* and *email*-*id* of the employeeid 10. |
|  |
|  |
| 1. Display all employeeid, phone number, and email-id of all employees whose phone number starts with 9. |
|  |
|  |
| 1. Display all address (*from n2address relation*) details, who are living in ‘Las Vega’ city. |
|  |
|  |
| 1. Display all qualification (*from* *n2qualification relation*) details who have done ‘BE’ and his/her year of passing is 1964. |
|  |
|  |
| 1. Display all education (*from* *n2qualification relation*) detail who have studied in 'Yale University' |
|  |
|  |
| 1. Display all education (*from* *n2qualification relation*) detail who have studied in 'University of Chicago' and has done ‘BE’. |
|  |
|  |
| 1. Display all hobby name (*from* *n2hobbies relation*) for the employee 21. |
|  |
|  |
| 1. Display employeeid (*from* *n2hobbies relation*), whose hobby is playing ‘Football’. |
|  |
|  |
| 1. Display firstName, lastName, gender, and hiredate (*from n2employee relation*) whose firstname starts with the letter ‘S’ and gender is ‘F’. |
|  |
|  |
| 1. Get employee details in descending order of gender and 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. |
|  |
|  |
| 1. Display *(firstname, lastname, gender, name, and institute)* name from *n2employee* and *n2qualification* relations. |
|  |
|  |
| 1. Display *(firstname, lastname, institute, and university)* who have studied in 'Yale University'. *(Use n2employee, and n2qualification relation)* |
|  |
|  |
| 1. Display all employee's with their hobby detail whose hobby *NAME* is 'Swimming'. *(Use n2employee, and n2hobbies relation)* |
|  |
|  |
| 1. Display *(employeeid, name, stream institute, university, and grade)* whose stream is ‘Commerce’. *(Use n2employee, and n2qualification relation)* |
|  |
|  |
| 1. Display the hobby detail of all employees who have hobby same as ‘SMITH’ (lastName). |
|  |
|  |
| 1. Display *(firstname, lastname, gender, line1, line2, and city)* of all employees who are staying in “Las Vega” *city*. |
|  |
|  |
| 1. Display *(department name)* who are currently working in departments 10 or 20. |
|  |
|  |
| 1. Display *(employee id, firstname, lastname, department name, location, fromdate, and todate)* of employeeid 10. |
|  |
|  |
| 1. Display *(firstname, lastname, phonenumber, and emailid)* of employeeid 14. |
|  |
|  |
| 1. Display *(firstname and count the total number of phone an employee is having)* for all employees. |
|  |
|  |
| 1. Get employee’s *(firstname, lastname,* *gender, phonenumber, and emailid)* whose employeeid is 14. |
|  |
|  |
| 1. Get *(firstname, lastname, gender, and all department details)* of the employee 21. |
|  |
|  |
| 1. Get *(employee id, firstname, lastname, gender, and all hobby name)* for all employees. |
|  |
|  |
| 1. Get highest salary of the current employee. |
|  |
|  |
| 1. Display employee details and his job history details for all employees. |
|  |
|  |
| 1. Display *(employee id, firstname, lastname, gender, his previous employeer, fromdate, and todate)* for the employee whose id is 20. |
|  |
|  |
| 1. Display *(employee id, firstname, lastname, gender, his/her employeer, fromdate, and todate)* who had previously worked under ‘leena’ |
|  |
|  |
| 1. Get the first name, last name, department number and department name, for all employees for current department ID is 10 or 70. |
|  |
|  |
| 1. Get all department details where no employees are working. |
|  |
|  |
| 1. Get employee firstname and phone no employee id is 7. |
|  |
|  |
| 1. Get employee details with hobbies. |
|  |
|  |
| 1. Get the list of employees having hobby is ‘Running’ |
|  |
|  |
| 1. Display all employee and address details who are staying in ‘New York’ city. |
|  |
|  |
| 1. Display *employeeid, phonenumber, emailid, line1, line2*, and *city* (*from n2contact and n2address relation*). |
|  |
|  |
| 1. Display *firstname, lastname, gender, line1, line2, city,* and *pin* whose *pin* starts with 3. |
|  |
|  |
| 1. Get all employee details whose qualification is ‘M.Com.’. |
|  |
|  |
| 1. Display all employee, qualification name, and grade whose grade is ‘A’ and has done ‘BE’. |
|  |
|  |
| 1. Display all employee details and his qualification name, and stream who have done ‘arts’. |
|  |
|  |
| 1. Display all employee details and his qualification name, and stream who have done ‘arts’ is 12th. |
|  |
|  |

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. |
|  |
|  |
| 1. Count total number of female employees. |
|  |
|  |
| 1. Count total number of female employees whose firstname starts with the letter ‘F’. |
|  |
|  |
| 1. Count total number employee who were hired in the year 1962. |
|  |
|  |
| 1. Count how many phonenumber an employeeid 3 is having. |
|  |
|  |
| 1. Count number of hobbies every employee is having. |
|  |
|  |
| 1. Count total number of unique hobbies. |
|  |
|  |
| 1. Count how many employees has done ‘BE’. |
|  |
|  |
| 1. Stream wise count of employees who have taken admission in ‘BE’. |
|  |
|  |
| 1. Stream wise count of employees who have taken admission in ‘BE’ and have secured ‘A’ grade. |
|  |
|  |
| 1. Count unique cities from n2address relation. |
|  |
|  |
| 1. Count how many employee are staying in ‘Pune’ city. |
|  |
|  |
| 1. Count the number of employee who have more than 60% in ‘BE’. |
|  |
|  |
| 1. Stream wise count of employee who have more than 60% in ‘BE’. |
|  |
|  |
| 1. Count how many employees are from ‘PUNE’ city. (*use n2address relation*) |
|  |
|  |

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. |
|  |
|  |
| 1. Display the employee’s detail who have least hobbies. |
|  |
|  |
| 1. Display all employee’s detail who have least hobbies for *gender* ‘M’. |
|  |
|  |
| 1. Display *(firstname, lastname)* who are having more than 4 hobbies. |
|  |
|  |
| 1. Get all department details where no employees are working. |
|  |
|  |
| 1. Get all department details where employees are working. |
|  |
|  |
| 1. Get the *(employeeid,* and *salary)* for those employees who earn less than the employee earn whose *employeeID* is 10. |
|  |
|  |
| 1. Display all salary details who are having same *salaries* for the current jobs. |
|  |
|  |
| 1. Display *(salary, and count of salaries)* of all employees who same salary for the current job. |
|  |
|  |
| 1. Get all employees whose employer is either 'sharmin' or 'saleel' |
|  |
|  |
| 1. Get salary details of the current employees, whose salary is below 2500. |
|  |
|  |
| 1. Display all employees who are getting commission. |
|  |
|  |
| 1. Display all employees who are not getting commission. |
|  |
|  |
| 1. Display all employees’ commission details for the currentjob who are not getting commission. |
|  |
|  |
| 1. Display all employees’ commission details for the currentjob who are getting commission. |
|  |
|  |
| 1. Display all employee details who have three or more emailid. |
|  |
|  |
| 1. Get lowest salary of employee working in current job? |
|  |
|  |
| 1. Get lowest salary of employee working in current job? |
|  |
|  |
| 1. Get average salary of employee working in current job? |
|  |
|  |
| 1. Get sum salary of employee working in current job? |
|  |
|  |
| 1. Get highest *commission* of employee for the current job. |
|  |
|  |
| 1. Get second highest *salary* of employee for the current job. |
|  |
|  |
| 1. Print *salary* of all employees in the given format 3000\*\*\*\*\* for the current job. |
|  |
|  |
| 1. Get the *(department name, and count)*, than how many employees are working in which department for the current job. |
|  |
|  |
| 1. Display the name of stream where maximum number of employees has taken admission in ‘BE’. |
|  |
|  |
| 1. Count how many employee are not getting commission for the current job. |
|  |
|  |
| 1. Count how many employee are getting commission for the current job. |
|  |
|  |
| 1. Display all commission details of those employees who are not getting commission for the current job. |
|  |
|  |

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. |
|  |
|  |
| 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. |
|  |
|  |
| 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) |
|  |
|  |
| 1. Write a procedure (named addDepartment) that inserts a new department into the N2DEPARTMENT table. (Use: N2DEPARTMENT table) |
|  |
|  |
| 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) |
|  |
|  |

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) |
|  |
|  |
| 1. Write a function to return auto generate deptno and return the new value (Use: N2DEPARTMENT table). |
|  |
|  |
| 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) |
|  |
|  |

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. |
|  |
|  |
| 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). |
|  |
|  |
| 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. |
|  |
|  |
| 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. |
|  |
|  |
| 1. Write a trigger on N2EMPLOYEE table, that if today is Sunday then, no record should get inserted in EMP table. |
|  |
|  |

Section 5

Assignment

Feb19/ DBT/ 129

Database Technologies

Diploma in Advance Computing

February 2019

**MongoDB**

USE ***EMP***collection.

|  |
| --- |
| 1. Display all databases. |
|  |
|  |
| 1. Display the current database. |
|  |
|  |
| 1. Display all collection. |
|  |
|  |
| 1. Display the current version of MongoDB. |
|  |
|  |
| 1. Display the current host details. |
|  |
|  |
| 1. Get the current IP address and the PORT number. |
|  |
|  |
| 1. Display all documents from EMP collection. |
|  |
|  |
| 1. Display all documents from EMP collection in JSON format. |
|  |
|  |
| 1. Display first 5 documents from EMP collection. |
|  |
|  |
| 1. Display employee name, and his address from EMP collection. |
|  |
|  |
| 1. Display all building and coord details of all employee from EMP collection. |
|  |
|  |
| 1. Display all documents who are staying in building number “2780”. |
|  |
|  |
| 1. Display all female employee documents. |
|  |
|  |
| 1. Display all employee working in department number 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 |
|  |
|  |
| 1. Count total documents in EMP collection. |
|  |
|  |
| 1. Display ename, sal, comm fields from the collection, who are getting some comm. |
|  |
|  |
| 1. Count the documents of ‘Computer Programmer’ |
|  |
|  |
| 1. Display ename, job, and salary fields from EMP collection in ascending order of ename. |
|  |
|  |
| 1. Display all documents between 5 and 10. |
|  |
|  |