NAME : Himanshu Dixit
ENROLL NO. : 21103262
BATCH : B10

Software Development Lab – II [15B17CI271]
Assignment Sheet
Week 11-12

Q1. *Consider Employee table*

EMPNO	EMP_NAME	DEPT	SALARY	DOJ	BRANCH
E101	Amit	Production	45000	12-MAR-00	Bangalore
E102	Amit	HR	70000	03-JUL-02	Bangalore
E103	sunita	Management	120000	11-JAN-01	mysore
E105	sunita	IT	67000	01-AUG-01	mysore
E106	mahesh	Civil	145000	20-SEP-03	Mumbai

Perform the following

1. Display all the fields of employee table
2. Retrieve employee number and their salary
3. Retrieve average salary of all employee
4. Retrieve number of employee
5. Retrieve distinct number of employee
6. Retrieve total salary of employee group by employee name and count similar names
7. Retrieve total salary of employee which is greater than >120000
8. Display name of employee in descending order
9. Display details of employee whose name is AMIT and salary greater than 50000;

Solution :

*1. select * from employee;*
2. select EMPNO, SALARY from employee;
3. select avg(SALARY) from employee;
4. select count(*) as NO_EMP from employee;
5. select count(distinct EMP_NAME) from employee;
6. select EMP_NAME,sum(SALARY),count(*) from employee group by EMP_NAME;

7. select EMP_NAME,sum(SALARY) from employee group by EMP_NAME having sum(SALARY)>120000;
8. select EMP_NAME from employee order by EMP_NAME desc;
*9. select * from employee where EMP_NAME = "Amit" AND SALARY > 50000;*

Q2. *Create a STUDENTS table with Roll No. as primary key. Name and Roll No. cannot be NULL. Assume appropriate attributes for the table.*
Solution :

```
create table student(
Roll_No int primary key,
Stu_Name char(20) Not null
);
```

Q3. *Create a "Customer" table with attributes as ID, City, LastName, FirstName, Address, TotalOrders using other tables named "User" and "Orders". The User table has following attributes UserId, City, LastName, and FirstName. Attributes of "Orders" table are UserId, LastOrderNo., TotalOrders, and Address.*
Solution :

```
create table customer as (select users.ID, users.CITY, users.LASTNAME,
users.FIRSTNAME, orders.LASTORDERNO, orders.TOTALORDER, orders.ADDRESS
from users , orders where users.ID = orders.ID );
```

Q. *Assume we have a table called employees with the following data:*

employee_number	last_name	first_name	salary	dept_id
1001	Smith	John	62000	500
1002	Anderson	Jane	57500	500
1003	Everest	Brad	71000	501
1004	Horvath	Jack	42000	501

Q4. *Write command to insert an employee record whose employee_number is 1005, employee_name is Sally Johnson, salary is $58,000, and dept_id is 500.*
Solution :

```
insert into emp2 values (1005, 'Johnson', 'Sally', 58000, 500);
```

Q5. *Write command to insert the employee information with employee_number greater than 1002 into the customers table (customer_id, last_name, first_name).*

Solution :

```
INSERT INTO customers
(customer_id, last_name, first_name)
SELECT employee_number AS customer_id, last_name, first_name
FROM employee
WHERE employee_number > 1002;
```

Q. *Table for the further questions*

EmpID	EmpName	EmpEmail	PhoneNumber	Salary	City
1	Nidhi	nidhi@sample.com	9955669999	50000	Mumbai
2	Anay	anay@sample.com	9875679861	55000	Pune
3	Rahul	rahul@sample.com	9876543212	35000	Delhi
4	Sonia	sonia@sample.com	9876543234	35000	Delhi
5	Akash	akash@sample.com	9866865686	25000	Mumbai

Q6. *Write a query to retrieve the number of employees in each city.*

Solution :

```
select count(*) , city from emp group by city;
```

Q7. *Write a query to retrieve the number of employees having different salaries in each city.*

Solution :

```
select count(distinct salary) , city from emp group by city;
```

Q8. *Write a query to retrieve the number of employees in each city, sorted in descending order.*

Solution :

```
select count(*) , city from emp group by city order by count(*) desc;
```

Q9. *Delete all records in the employees table (employee_id, last_name, first_name) where there is a record in the contacts (contact_id, last_name, first_name) table whose contact_id is less than 100, and the contact_id matches the employee_id.*

Solution :

Delete from employees where contact.contact_id<100 and employee.employee_id=contact.conact_id;

Q10. *Print all rows from the employees table where the employee_id is between 25 and 100.*

Solution :

*Select * from employees where employee_id>25 AND employee_id<100;*

---

1) *Create a map named Students where the keys will be integers (RollNo), and the values will be strings (Name). Insert values into the map Students. A key of 200 and a value of Alice will be inserted into the map. Insert values into the map Students. A key of 201 and a value of John will be inserted into the map. Use the size() function to get the size of the map named Students. Use a for loop to create an iterator named it to iterate over the elements of the map named Students. Print the values of the map Students on the console.*

Solution :

```
#include <iostream>
#include <map>
using namespace std;
int main()
{
    map<int, string> m;
    m.insert(make_pair(200,"Alice"));
    m.insert(make_pair(201,"John"));
    cout<<"Size of map is : "<<m.size()<<endl;
    map<int,string>::iterator itr;
    for(itr = m.begin(); itr!=m.end(); itr++)
```

```
    cout<<(*itr).first<<"     "<<(*itr).second<<endl;
    return 0;
}
```

2) *Create a map named m where the keys will be integers, and the values will be integers. Three entries have been made into the map. Insert a new entry into the map m. A key of 5 and a value of 6 will be inserted into the map. You can enter items into std::map using the insert() function. Remember that the std::map keys must be unique. So, it first checks whether each key is present in the map. If it's present, the entry will not be inserted, but it returns the iterator for the existing entry. If it's not present, the entry is inserted. Use the insert_or_assign() function to insert or modify an existing entry. Use a for loop to create an iterator named itr to iterate over the elements of the map named m. Print the values of the map m on the console.*

Solution :

```
#include <iostream>
#include <map>
using namespace std;
int main()
{
    map<int, int> m;
    m.insert(make_pair(5, 6));
    int a, b;
    while(m.size() != 3)
    {
        cout << "Enter key and Value : ";
        cin >> a >> b;
        map<int, int>::iterator itr;
        itr = m.find(a);
        if (itr != m.end())
            itr->second = b;
        else
            m.insert(make_pair(a, b));
    }
    map<int, int>::iterator itrr = m.begin();
    for(itrr = m.begin(); itrr!=m.end(); itrr++)
    //for(auto p : m)
    cout<<itrr->first<<"     "<<itrr->second<<endl;
    return 0;
}
```

3) *Create a map named Students whose keys will be integers and values strings. Insert values into the map Students. A key of 200 and a value of Alice will be inserted into the map. Insert values into the map Students. A key of 201 and a value of John will be inserted into the map. You can use the find() function to search for elements in a map by their keys. If the key isn't found, the function returns std::map::end. Otherwise, an iterator of the searched element will be returned. Look for the value associated with a key of 201. Use an if statement to check whether the value for the key is found. Print the value of the key alongside some text on the console.*

Solution :

```
#include <iostream>
#include <map>
using namespace std;
int main()
{
    map<int, string> m;
    m.insert(make_pair(200,"Alice"));
    m.insert(make_pair(201,"John"));
    cout<<"Search of 201  \n";
    map<int,string>::iterator itr;
    itr = m.find(201);
    if(itr != m.end())
    cout<<"Found!! ----->  "<<(*itr).first<<"     "<<(*itr).second<<endl;
    else
    cout<<"Not found!!";
    return 0;
}
```

4) *Create a map named my_map whose keys will be strings and values integers. Insert values into the map my_map. A key of Cow and a value of 1 will be inserted into the map. Insert values into the map my_map. A key of Cat and a value of 2 will be inserted into the map. Add a value 3 into the map my_map with a key of a lion. Create an iterator to iterate over the map my_map looking for the key cat. You can use the erase() function to delete a value from a map. You simply create an iterator that points to the element to be deleted. The iterator is then passed to the erase() function. Delete the element pointed to by the iterator. Use an iterator*

to iterate over the elements of the map my_map from the start to the end. Print out the contents of the map my_map on the console.

Solution :

```
#include <iostream>
#include <map>
using namespace std;
int main()
{
    map<string, int> my_map;
    my_map.insert(make_pair("cow",1));
    my_map.insert(make_pair("cat",2));
    my_map.insert(make_pair("lion",3));
    map<string,int>::iterator itr;
    itr = my_map.find("cat");
    my_map.erase(itr);
    map<string,int>::iterator itrr;
    for(itrr = my_map.begin(); itrr != my_map.end(); itrr++)
    cout<<itrr->first<<"     "<<itr->second<<endl;
    return 0;
}
```

5) *WAP in C++ to store the information entered by the user into the file.*

Solution :

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    ofstream fout;
    fout.open("file.txt");
    string s;
    cout << "Enter what u want to store in a file : ";
    cin >> s;
    fout << s;
    fout.close();
    return 0;
}
```

6) *WAP in C++ to Retrieve information from the file which is entered in Q5.*

Solution :

```
#include <iostream>
```

```cpp
#include <fstream>
using namespace std;
int main()
{
    ifstream fin;
    fin.open("file.txt");
    string s;
    cout << "the data store in a file is : ";
    fin >> s;
    cout << s;
    fin.close();
    return 0;
}
```

7) *WAP in C++ to Copy Content of One File to Another. You have to ask from user to enter the name of source file (with extension) and target file (with extension).*

Solution :

```cpp
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    string file1, file2;
    cout << "Enter the file name from which u wants to copy the data : ";
    cin >> file1;
    cout << "Enter the file name to where u wants to paste the data : ";
    cin >> file2;
    ifstream fin;
    ofstream fout;
    fout.open(file2);
    fin.open(file1);
    char ch = fin.get();
    while (!fin.eof())
    {
        cout<<ch;
        fout.put(ch);
        ch = fin.get();
    }
    cout<<"\nData copied successfully";
    fout.close();
    fin.close();
    return 0;
}
```

8) *Write a program in C++ that merges the content of two files into the third file. To merge two files in C++ programming, you have to ask from user to enter the name of all the three files with extension*

Solution :

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    string file1, file2, file3;
    cout << "Enter the file names u have to merge : ";
    cin >> file1 >> file2;
    cout << "Enter the file name to where u wants to store the merge data : ";
    cin >> file3;
    ifstream fin1, fin2;
    ofstream fout;
    fin1.open(file1);
    fin2.open(file2);
    fout.open(file3);
    char ch = fin1.get();
    while (!fin1.eof())
    {
        fout.put(ch);
        ch = fin1.get();
    }
    ch = fin2.get();
    while (!fin2.eof())
    {
        fout.put(ch);
        ch = fin2.get();
    }
    fin1.close();
    fin2.close();
    fout.close();
    return 0;
}
```

9) *write a program in C++ that deletes a file from the current directory. To delete any file from the current directory, you have to ask from user to enter the name of file first and then perform the operation of deleting it from the directory.*

Solution :

```
#include <iostream>
#include <stdio.h>
using namespace std;
int main()
{
    int status;
    char fileName[20];
    cout << "Enter the Name of File: ";
    cin >> fileName;
    status = remove(fileName);
    if (status == 0)
        cout << "\nFile Deleted Successfully!";
    else
        cout << "\nError Occurred!";
    cout << endl;
    return 0;
}
```

10) *WAP in C++ which opens a file in reading and writing mode. After writing information inputted by the user to a file named a file.dat, the program reads information from the file and outputs it onto the screen.*

Solution :

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    fstream file;
    file.open("FILE.dat", ios::out);
    string s;
    cout<<"Enter : ";
    cin>>s;
    file.write((char *)&s , sizeof(s));
    file.close();
    string a;
    file.open("FILE.dat", ios::in);
    file.read((char *)&a , sizeof(a));
    cout<<a;
    file.close();
    return 0;
}
```