

Software Development Lab – II [15B17CI271]

Week 12

COURSE OUTCOMES		COGNITIVE LEVELS
C173.1	Write programs in C++ to implement OOPs concepts related to objects, classes, constructor, destructor, and friend function.	Apply Level (Level 3)
C173.2	Write programs in C++ using OOPs concept like encapsulation, inheritance, polymorphism and abstraction.	Apply Level (Level 3)
C173.3	Write programs in C++ using Standard Template Library.	Apply Level (Level 3)
C173.4	Perform exception handling in C++ programs.	Apply Level (Level 3)
C173.5	Write MySQL queries to perform operations like ADD, DELETE, UPDATE, SELECT on relational databases.	Apply Level (Level 3)

Note: Students are advised to submit their solutions to respective lab faculty. The solution file must be named as “rollno_first name_w12.doc” (here w12 represents week 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;

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

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.

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.

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

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.

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

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

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*.

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