Computer Science:PaperIII:Basic 'C' Programming and Database Handling practicals

Sr.No.	Assignment Name	Date	Remark
Term I			
1	To demonstrate the use of data types, simple operators and expressions.		
2	To demonstrate use of decision making statements such as if and if-else.		
3	To demonstrate decision making statements (switch case)		
4	To demonstrate use of simple loops		
5	To demonstrate use of nested loops		
6	To demonstrate menu driven programs and use of standard library functions.		
7	To demonstrate writing C Programs in modular way(use of user defined functions).		
8	To demonstrate Recursion.		
9	To demonstrate use of 1-D arrays and functions.		
10	To demonstrate use of 2-D arrays and functions.		
Term II			
11	To create simple tables , with only the primary key constraint		
12	To create more than one table, with referential integrity constraint, PK constraint.		
13	To create more than one table, with referential integrity constraint, PK constraint.		
14	To Drop a table from database,to alter the table.		
15	To insert/update/delete statements.		
16	To query the tables using simple form of select statement		
17	To query table,using set operations(union,intersect,except)		
18	To query tables using nested queries.		
19	To query tables,using nested queries.		
20	To small case studies.		

Anekant Education Society's

Tuljaram Chaturchand College,

Baramati – 413102 (Pune)

DEPARTMENT OF COMPUTER SCIENCE

Computer Practical Journal

- CERTIFICATE -

University Seat No. Date:-

This is to certify that Mr./ Miss. / Smt. _____

_____ has satisfactorily completed the course in

Practical as prescribed by the University of Pune for the F.Y./ S.Y./

T.Y. B.Sc. (Computer Science-Lab Course I / II / III) in the Year 20 – 20

In-Charge Internal Examiner External Examiner Head

(Practical) Dept of Computer Science

Assignment:1

To demonstrate the use of data types, simple operators and expressions

1. Data type Table

Data	Data Format	C Data Type	C Variable declaration	Input Statement	Output Statement

quantity
month
credit-card number
	Numeric	int
		Short int
		long int
			int quantity;
			short month;
			long ccno;
				scanf("%d",&quantity);
				scanf("%d",&month);
				scanf("%ld", &ccno);
					printf("The quantity is %d", quantity);
					printf("The credit card number is %ld, ccno);

price	real	float double	float price; const double pi=3.141593;	scanf("%f",&price);	printf("The price is %5.2f", price);
grade	character		char grade;	scanf("%c",&grade)	printf("The grade is %c",grade);

2. Expression Examples

Expression	C expression
Increment by a 3	a = a + 3
Decrement b by 1	b = b-1 or b--
$2a^2 + 5b/2$	2*a*a + 5*b/2
7/13(x-5)	(float)7/13*(x-5)
5% of 56	(float)5/100*56
n is between 12 to 70	n>=12 && n<=70
$\pi r^2 h$	Pi*r*r*h
n is not divisible by 7	n % 7 != 0
n is even	n%2==0
ch is an alphabet	ch>='A' && ch<='Z' || ch>='a' && ch<='z'

Note: The operators in the above expressions will be executed according to precedence and associativity rules of operators.

1. Type the sample program given above

Follow the following guidelines

a. Type the sample program save it pnr.c

Compile the program using gcc compiler available in

Linux(red hat) gcc pnr.c

A executable file a.out is created by the compiler in current directory. The program can be executed by typing name of the file as follows giving the path.

./a.out

Write Programs to slove the following problems

1. **Write a C program to display message "Hello ! Welcome To C" using printf function.**
2. **Write a C program to demonstrate declaration of variables and functions.**
3. **Write a C program for addition , substraction, multiplication and division.**
4. **Write a C program to accept principal sum, rate of interest and number of years . Compute simple interest.**
5. **Write a C program to accept three dimensions length(l),breadth(b)and height(h) of a cuboid and print surface area and volume (Hint:Surface area=2(lb+lh+bh),volume=lbh)**
6. **Write a C program to demonstrate declaration and use of constants.**
7. **Write a C program to calculate area of circle , circumference of circle (2*PI*r) ,area of rectgangle.**
8. **Write a C program to accept temperature in celcius , convert it into farenheite (f=9*c/5+32).**
9. **Write a C program to accept marks of five subjects and display percentage.**
10. **Write a C program accept dimensions of a cylinder and print the surface area and volume.**
    (Hint: surface area=2*pi*r*r+2*pi*r*h, volume=pi*r*r*h)
11. **Write a C program to calculate squareroot , power , sine , cosine of an angle (hint : rad = PI *angle /180)**
12. **Write a C program to calculate distance between two coordinates**
    $D=\sqrt{(y2-y1)^2 + (x2-x1)^2}$
13. **Write a C program to accept two numbers and interchange them.**
14. **Write a C program to accept a character from user and display its ASCII value.**
15. **Write a C program to accept a character and display its previous , next characters ASCII code.**
16. **Write a C program to accept empid , basic salary calculate total salary including tax , dearness allowance.**
17. **Write a C program to a cashier has currency notes of denomination 1,5,10.accept the amount to be withdrawn from the user and print the total number of currency notes of each denomination the cashier will have to give.**

Signature of the Instructor Remark Date

Assignment:2

To demonstrate use of decision making statements such as if and if-else.

During problem solving, we come across situations when we have to choose one of the alternative paths depending upon the result of some condition. Condition is an expression evaluating to true or false. This is known as the Branching or decision-making statement. Several forms of If and else constructs are used in C to support decision-making.

1) if statements

```
if (condition)
{
Statement;
}
```

2) if… else

```
if(condition)
        {
                statement;
        }else
        {
         Statement;

        }
```

Write Programs to slove the following problems

1. **Write a C program to check whether number is positive and negative.**
2. **Write a C program to check whether number is even or odd.**
3. **Write a C program to check whether number is divisible by 5 and 7.**
4. **Write a C program to check whether number is within range Take limit=40 , ulimit=90.**
5. **Write a C program to Check whether character is a digit or a character or other symbol Also check character is in lowercase or uppercase (ASCII value of digit is bet 48 to 58, a__z is 97 to 123, A__Z 65 to 96)**
6. **Write a C program to Check year is leap year or not .**
7. **Write a C program to Accept marks of three subjects and find the total marks, average. Display class obtained**
8. **Write a C program to accept 3 sides of triangle as input and print whether the triangle is valid or not. (hint: the triangle is valid if sum of each of the two sides is greater than the third side)**
9. **Write a C program to Calculate roots of quadratic equation .**

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:3

To demonstrate decision making statements (switch case)

The control statement that allows us to make a decision from the number of choices is called a switch-case statement. It is a multi-way decision making statement.

Switch syntax:

```
switch(expression)
{
  Case value 1:statement 1;
     break;
  Case value 2:statement 2;
     break;
  Case value n:statement n;
     break;
 default : default statement;
      break;
}
```

Write Programs to slove the following problems

1. **Write a C program to Display selected number or option**
2. **Write a c program Accept two integers and perform arithmetic operations**
3. **Accept single digit and display it in words.**
4. **Write a c program having a menu with the following options and corresponding actions 1. Area of circle 2.Area of Rectangle 3. Area of triangle .**

----------	----------	----------
Signature of the Instructor	Remark	Date

Assignment:4

To demonstrate use of simple loops.

We need to perform certain actions repeatedly for a fixed number of times or till some condition holds true. These repetitive operations are done using loop control statements. The types of loop structures supported in C are

1)while statement	**2)**do-while statement	**3)for statement**
while(condition)	do{	for(expr1;expr2;expr3)
{statement 1;	statement 1;	{
Statement 2;	Statement 2;	Statement1;
}	}while(condition);	Statement2;
		}

Write Programs to slove the following problems

1. **Write a C program to Accept a number and reverse it . (if input 978 o/p 879)**
2. **Write a C program to Accept a number & find sum of digits**
3. **Write a C program to Accept a number & check whether it is a Armstrong**
4. **number (sum of cube of its digit = number itself)**
5. **Write a C program to accepts a number and checks if the number is a palindrome.**
6. **Write a c program to accept characters from the keyboard till the use**
7. **enter EOF(ctrl Z)and count the number of vowels ,spaces and line in the text.**
8. **Write a C program to accept a character and perform various operations on character using standard library function. (Hint : use isalpha, isdigit, islower, etc.)**
9. **Write a C program to accept a number and display its digits in Words. (Input : 572 output: FIVE SEVEN TWO)**
10. **Write a C program to find GCD and LCM of numbers.**
11. **Write a C program to accept a number and calculate factorial of given number.**
12. **Write a C program to accept a number and check whether number is Perfect or not.**
13. **(Hint : sum of factors = number itself)**
14. **Write a C program to find Fibonacci series i.e. 1 1 2 3 5 8…. (Hint : sum of previous two numbers is the next term)**
15. **Write a C program to accept an integer and check if it is prime or not.**
16. **Find out even & odd number from 1 to 100 number.**

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:5

To demonstrate use of nested loops

Nested loop means a loop that is contained within another loop. Nesting can be done upto any levels. However the inner loop has to be completely enclosed in the outer loop. No overlapping of loops is allowed.
let to nest any loop within another. For example, we can have a for loop inside a while or do while or a while loop inside a for.

Write C programs for the following problems.

1. **The Sample program 1 displays n lines of the following triangle. Type the program and execute it for different values of n.**

```
1
1   2
1   2   3
1   2   3   4
```

2. **Modify the sample program 1 to display n lines of the Floyd's triangle as follows (here n=4).**

```
1
2   3
4   5   6
7   8   9   10
```

3. **Write a C program to calculate sum of first n terms of series**
   1+2+3+…….

4. **Write a C program to calculate sum of first n terms of series**
   1+3+5+…….

5. **Write a C program to calculate sum of first n terms of series**
   X + 3x + 5x +7x + ……

6. **Write a C to accept a number x and integer n and calculate the sum of first n terms of series.**
   $1/x + 2/x^2 + 3/x^3$ +….

7. **Write a C program to Display following output :**

```
    *

   * *

  * * *
```

8. **Write a C program to Display following triangle :**

```
1
1  2
1  2  3
```

9. **Write a C program to display all Armstrong numbers between 1 to 500.**
10. **Write a C program to Display following output :**

```
A  B  C  D
E  F  G
H  I
J
```

11. **Write a C program to display multiplication table from 2 to 9.**

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:6

To demonstrate menu driven programs and use of standard library functions

A function is a named sub-module of a program, which performs a specific, well-defined task. It can accept information from the calling function in the form of arguments and return only 1 value. C provides a rich library of standard functions. We will explore some such function libraries and use these functions in our programs.

ctype.h : contains function prototypes for performing various operations on characters. Some commonly used functions are listed below.

Function Name	Purpose	Example
Isalpha	Check whether a character is a alphabet	if (isalpha(ch))
Isalnum	Check whether a character is alphanumeric	if (isalnum(ch))
Isdigit	Check whether a character is a digit	if (isdigit(ch))
Isspace	Check whether a character is a space	if (isspace(ch))
Ispunct	Check whether a character is a punctuation Symbol	if (ispunct(ch))
Isupper	Check whether a character is uppercase alphabet	if (isupper(ch))
Islower	Check whether a character is lowercase alphabet	if (isupper(ch))
Toupper	Converts a character to uppercase	ch = toupper(ch)
Tolower	Converts a character to lowercase	ch = tolower(ch)

math.h : This contains function prototypes for performing various mathematical operations on numeric data. Some commonly used functions are listed below.

Function Name	Purpose	Example
Cos	Cosine	a*a+b*b – 2*a*b*cos(abangle)
exp(double x)	exponential function, computes e^x	exp(x)
Log	natural logarithm	c= log(x)
log10	base-10 logarithm	y=log10(x)
pow(x,y)	compute a value taken to an exponent, x^y	y = 3*pow(x , 10)
Sin	Sine	z= sin(x) / x
Sqrt	square root	delta=sqrt(b*b – 4*a*c)

Write C programs for the following problems.

1. **Write a menu driven program to perform the following operations on a character type variable.**
   a. **Check if it is an alphabet**
   b. **Check if it is a digit.**
   c. **Check if it is lowercase.**
   d. **Check if it is uppercase.**
   e. **Convert it to uppercase.**

f. **Convert it to lowercase.**

Refer to the sample code given above and use standard functions from ctype.h

2. **Write a program, which accepts a character from the user and checks if it is an alphabet, digit or punctuation symbol. If it is an alphabet, check if it is uppercase or lowercase and then change the case.**
3. **Write a menu driven program to perform the following operations till the user selects Exit. Accept appropriate data for each option. Use standard library functions from math.h**

   i. Sine ii. Cosine iii. log iv. e^x

   v. Square Root vi. Exit
4. **Accept two complex numbers from the user (real part, imaginary part).**
5. **Write a menu driven program to perform the following operations till the user selects Exit.**

   i.Add ii.Substract iii.Multiply iv.Exit

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:7

To demonstrate writing C programs in modular way (use of user defined functions

You have already used standard library functions. C allows to write and use user defined functions. Every program has a function named main. In main you can call some functions which in turn can call other functions.

The following table gives the syntax required to write and use functions

Sr. No	Actions involving functions	Syntax	Example
1.	Function declaration	returntype function(type arg1, type arg2 …);	void display(); int sum(int x, int y);

2.	Function definition	returntype function(type arg1,
		type arg2 …)
		{
		/* statements*/
		}
			float calcarea (float r)
			{
			float area = Pi *r*r ;
			return area;
			}

3.	Function call	function(arguments); variable = function(arguments);	display(); ans = calcarea(radius);

Write C programs for the following problems

1. **Write a C program to accept two numbers and Find maximum number between two numbers using function.**
2. **Write a C program for swapping of two numbers using function.**
3. **Write a C program to Calculate factorial using function.**
4. **Write a C program to accept two nos.+,/,*,%using function.**

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:8

To demonstrate Recursion

Recursion is a process by which a function calls itself either directly or indirectly. The points to be remembered while writing recursive functions

i. Each time the function is called recursively it must be closer to the solution.
ii. There must be some terminating condition, which will stop recursion.
iii. Usually the function contains an if –else branching statement where one branch makes recursive call while other branch has non-recursive terminating condition

Write C programs for the following problems

1. Write a recursive C function to calculate the sum of digits of a number. Use this function in main to accept a number and print sum of its digits.
2. Write a recursive C function to calculate the GCD of two numbers. Use this function in main. The GCD is calculated as :

   gcd(a,b) = a if b = 0

   a. gcd (b, a mod b) otherwise
3. Write a recursive C function to calculate x^y. (Do not use standard library function)
4. Write a recursive function to calculate the nth Fibonacci number. Use this function in main to display the first n Fibonacci numbers. The recursive definition of nth Fibonacci number is as follows:

   fib(n) = 1 if n = 1 or 2

   a. = fib(n-2) + fib(n-1) if n>2
5. Write a recursive function to calculate the sum of digits of a number till you get a single digit number. Example: 961 -> 16 -> 5. (Note: Do not use a loop)

| --- | --- | --- |
| Signature of the Instructor | Remark | Date |

Assignment:9

To demonstrate use of 1-D arrays and functions.

An array is a collection of data items of the same data type referred to by a common name. Each element of the array is accessed by an index or subscript. Hence, it is also called a subscripted variable.

1. How to declare one-dimensional array
   Syntax:-Data-type arrary_name[size];
   Example:int mat1[10];

Write programs to solve the following problems

1. **Write a C program to Accept elements for an array & display elements of an Array.**
2. **Write a C program to Display even numbers from an array**
3. **Write a C program to Find smallest & largest number from an Array**
4. **Write a program to accept n numbers in the range of 1 to 25 and count the frequency of occurrence of each number.**
5. **Write a function ,which accepts an interger array and an interger as parameters and counts the occurrences of the number in the array.**

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:10

To demonstrate use of 2-D arrays and functions.

How to declare two-dimensional array

Syntax:-Data-type arrary_name[size][size];

Example:int mat1[10][10];

Accessing elements

Array_name [index1][index2] where index1is the row & index 2 is the column location of an elements in the array.

Write C programs for the following problems.

1. **Write a C program to Accept elements for an array and sort elements of an array**
2. **Write a C program to Accept elements for an array and Merge two sorted arrays into a third array.**
3. **Write a C program for Addition of two matrices.**
4. **Write a c program for multiplication of two matrices.**
5. **Write a C program to Transpose of a matrix .**
6. **Write a program to accept an mXn matrix and display an m+1 X n+1 matrix such that the m+1th row contains the sum of all elements of corresponding row and the n+1th column contains the sum of elements of the corresponding column.**

Example:

A			B			
1	**2**	**3**	**1**	**2**	**3**	**6**
4	**5**	**6**	**4**	**5**	**6**	**15**
7	**8**	**9**	**7**	**8**	**9**	**24**
			12	**15**	**18**	**45**

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:11

Assignment to create simple tables , with only the primary key constraint (as a table level constraint & as a field level constraint) (include all data types)

A table is a database object that holds data. A table must have unique name, via which it can be referred. A table is made up of columns. Each column in the table must be given a unique name within that table. Each column will also have size a data-type and an optional constraint.

The data types permitted are

Data type	Syntax	Description	Example
Character data types	Char(n)	It is fixed length char.string of default size n.it values 1 byte if n is not specified.	Account_type char(6)
	Varchar(n)	It is var,length charater string with max.size n.	Stud_name varchar(10)
	Text(n)	It is used to store large text data ,no need to define a max.	Emp_name text
Numeric data types	Integer , int , serial	Serial is same as int ,only that values are incremented automatically.	Eno int Eno serial
	Numeric	A real no.with P digits,S of them after decimal number	Sal numeric(5,2) Sal numeric(n)
	Float	Real number	Weight float
Date and time type	Date	Stores date information	Birthdate date
	time	Stores time information	Birth time

Syntax for table creation :

Create tablename (attribute list);

Attribute list : ([attribute name data type optional constraint] , …….. .)

Constraints can be defined as either of the following :

Name	Description	Example
Column level	When data constraint is defined only with respect to one column & hence defined after the column definition, it	Create tablename (attribute1 datatype primary key , attribute2 datatype constraint constraint-name
	is called as column level constraint	,……)
Table Level	When data constraint spans across multiple columns & hence defined after defining all the table columns when creating or altering a table structure, it is called as table level constraint	Create tablename (attribute1 datatype , attribute2 datatype2 ,……, constraint pkey primary key(attribute1,atrribute2))

1.Strating Mysql from system prompt:mysql
Mysql>
2.issue the following commands:
to create a new database:
mysql >create database test;
here database test is created.
3.To get the list of previously created databases:
Mysql>show databases;
4. To connect to the particular database:
Mysql>use<database_name>;

1.Create a table with details as given below

Table Name	PLAYER		
Columns	Column Name	Column Data Type	Constraints
1	player_id	Integer	Primary key
2	Name	varchar (50)	
3	Birth_date	Date	
4	Birth_place	varchar(100)	
Table level constraint			

2.

Create a table with details as given below

Table Name	Student		
Columns	Column Name	Column Data Type	Constraints
1	Roll_no	Integer	
2	Class	varchar (20)	
3	Weight	numeric (6,2)	
4	Height	numeric (6,2)	
Table level constraint	Roll_no and class as primary key		

3.

Create a table with details as given below

Table Name	Project		
Columns	Column Name	Column Data Type	Constraints
1	project_id	Integer	Primary key
2	Project_name	varchar (20)	
3	Project_ description	Text	
4	Status	Boolean	
Table level constraint			

4.

Create a table with details as given below

Table Name	Donor		
Columns	Column Name	Column Data Type	Constraints
1	Donor_no	Integer	Primary key
2	Donor_name	varchar (50)	
3	Blood_group	Char(6)	
4	Last_date	Date	
Table level constraint			

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:12

To create more than one table, with referential integrity constraint, PK constraint.

The following is the list of constraints that can be defined for enforcing the referential integrity constraint.

Constraint	Use	Syntax and Example
Primary key	Designates a column or combination of columns as primary key	PRIMARY KEY (columnname[,columnname])
Foreign key	designates a column or grouping of columns as a foreign key with a table constraint	FOREIGN KEY (columnname[,columnname])
References	Identifies the primary key that is referenced by a foreign key. If only parent table name is specified it automatically references primary key of parent table	columnname datatype REFERENCES tablename[(columnname[,columnname])]
On delete Cascade	The referential integrity is maintained by automatically removing dependent foreign key values when primary key is deleted	columnname datatype REFERENCES tablename[(columnname)][ON DELETE CASCADE]
On update set null	If set, makes the foreign key column NULL, whenever there is a change in the primary key value of the referred Table	Constraint name foreign key column-name references referred-table(column-name) on update set null

Create tables for the information given below by giving appropriate integrity constraints as specified.

 1. Create the following tables :

Table Name	Property		
Columns	Column Name	Column Data Type	Constraints
1	Pnumber	Integer	Primary key
2	description	varchar (50)	Not null
3	Area	Char(10)	

Table Name	Owner		
Columns	Column Name	Column Data Type	Constraints
1	Owner-name	Varchar(50)	Primary key
2	Address	varchar (50)	
3	Phoneno	Integer	

Relationship A one-many relationship between owner & property. Define reference keys accordingly .

 2. Create the following tables :

Table Name	Hospital		
Columns	Column Name	Column Data Type	Constraints
1	Hno	Integer	Primary key
2	Name	varchar (50)	Not null
3	City	Char(10)	

Table Name	Doctor		
Columns	Column Name	Column Data Type	Constraints
1	Dno	Integer	Primary key
2	Dname	varchar (50)	
3	City	Char(10)	

Relationship A many-many relationship between hospital & doctor.

 3. Create the following tables :

Table Name	Patient		
Columns	Column Name	Column Data Type	Constraints
1	pno	Integer	Primary key
2	Name	varchar (50)	Not null
3	Address	Varchar(50)	

Table Name	Bed		
Columns	Column Name	Column Data Type	Constraints
1	Bedno	Integer	Primary key
2	Roomno	Integer	Primary key
3	description	Varchar(50)	

Relationship a one–to-one relationship between patient & bed.

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:13

To create one or more tables with Check constraint , Unique constraint, Not null constraint , in addition to the first two constraints (PK & FK)

The following is the list of additional integrity constraints.

Constraint	Use	Syntax and Example
NULL	Specifies that the column can contain null values	CREATE TABLE client_master (client_no text NOT NULL , name text NOT NULL, addr text NULL , bal_due integer));
NOT NULL	Specifies that the column can not contain null values	CREATE TABLE client_master (client_no text NOT NULL , name text NOT NULL, addr text NOT NULL , bal_due integer));
UNIQUE	Forces the column to have unique values	CREATE TABLE client_master (client_no text UNIQUE, name text ,addr text, bal_due integer));
CHECK	Specifies a condition that each row in the table must satisfy. Condition is specified as a logical expression that evaluates either TRUE or FALSE.	CREATE TABLE client_master (client_no text CHECK(client_no like 'C%'), name text CHECK(name=upper(name)) , addr text));

1.

Create a table with details as given below

Table Name	Machine		
Columns	Column Name	Column Data Type	Constraints
1	Machine_id	Integer	Primary key
2	Machine_name	varchar (50)	NOT NULL, uppercase
3	Machine_type	varchar(10)	Type in (‘drilling’, ‘milling’, ‘lathe’, ‘turning’, ‘grinding’)
4	Machine_price	Float	Greater than zero
5	Machine_cost	float	
Table level constraint	Machine_cost less than Machine_price		

2.

Create a table with details as given below

Table Name	Employee		
Columns	Column Name	Column Data Type	Constraints
1	Emp_id	integer	Primary key
2	Emp_name	varchar (50)	NOT NULL, uppercase
3	Emp_desg	varchar(10)	Designation in (‘Manager’, ‘staff’, ‘worker’)
4	Emp_sal	float	Greater than zero
5	Emp_uid	text	Unique
Table level constraint	Emp_uid not equal to Empl_id		

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:14

To drop a table from the database, to alter the schema of a table in the Database.

The Drop & Alter DDL statements :

Drop:-Deletes an object(table/view/constraint)schema from the database.

Drop object-type
object-name;

e.g. Drop table employee;

Alter:- ALTER TABLE command of SQL is used to modify the structure of the table

It can be used for following purposes

a)adding new column b) modifying existing columns

c)add an integrity constraint d)to refine a column

Alter table tablename Add (new columnname datatype(size), new columnname datatype(size)…);

Alter table emp(add primary key(eno));

alter table student(add constraint city_ chk check city in ('pune', 'mumbai'));

Solve the following

1. Remove employee table from your database. Create table employee(eno, ename, sal). Add designation column in the employee table with values restricted to a set of values.
2. Remove student table from your database. Create table student(student_no, sname, date_of_birth). Add new column into student relation named address as a text data type with NOT NULL integrity constraint and a column phone of data type integer.
3. Consider the project relation created in the assignment 12. Add a constraint that the project name should always start with the letter 'R'
4. Consider the relation hospital created in assignment 12. Add a column hbudget of type int , with the constraint that budget of any hospital should always > 50000.

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:15

Assignment to insert / update / delete records using tables created in previous Assignments. (use simple forms of insert / update / delete statements)

The insert / update / delete statements

syntax

Insert into tablename values(list of column values);

e.g.Insert into emp values(1,'joshi',4000,'pune);

syntax

UPDATE tablename SET columnname = value where condition;

e.g. Update emp set sal = sal+1000 where eno =1;

syntax

delete from table name where condition;

e.g. Delete from emp ;

Delete from emp where eno = 1;

Consider the tables created in assignments 11,12,13,14. type and execute the below statements for these tables.

Write the output of each statement & justify your answer

1. INSERT INTO sales_order(s_order_no,s_order_date,client_no) VALUES ('A2100', now() ,'C40014');
2. INSERT INTO client_master values('A2100','NAVEEN','Natraj apt','pune_nagar road','pune',40014**);**
3. insert into client_master values ('A2100','NAVEEN',NULL,'pune_nagar road','pune',40014**);**
4. UPDATE emp SET netsal= net_sal_basic_sal*0.15;
5. UPDATE student
   SET name='SONALI',address='Jayraj apartment'
   WHERE stud_id=104 ;
6. DELETE from emp;
7. DELETE from emp
   WHERE net_sal <1000;

Solve the following

1. Create the following tables (primary keys are underlined.).

Property(pno,description, area) Owner(<u>oname,</u>address,phone)

An owner can have one or more properties, but a property belongs to exactly one owner . Create the relations accordingly ,so that the relationship is handled properly and the relations are in normalized form (3NF).

a. Insert two records into owner table.
b. insert 2 property records for each owner .
c. Update phone no of "Mr. Nene" to 9890278008
d. Delete all properties from "pune" owned by " Mr. Joshi"

2 . Create the following tables (primary keys are underlined.).

Emp(eno,ename,designation,sal)

Dept(dno,dname,dloc)

There exists a one-to-many relationship between emp & dept.

2.Create the Relations accordingly, so that the relationship is handled properly and the relations are in normalized form (3NF).

a. Insert 5 records into department table
b. Insert 2 employee records for each department.
c. increase salary of "managers" by 15%;
d. delete all employees of deparment 30;
e. delete all employees who are working as a "clerk"
f. change location of department 20 to 'KOLKATA'

 3 . Create the following tables (primary keys are underlined.).

Sales_order(s_order_no,s_order_date)

Client(client_no, name, address)

A clinet can give one or more sales_orders , but a sales_order belongs to exactly one client. Create the Relations accordingly, so that the relationship is handled properly and the relations are in normalized form (3NF).

a. insert 2 client records into client table
b. insert 3 sales records for each client
c. change order date of client_no 'C004' to 12/4/08
d. delete all sale records having order date before 10th feb. 08
e. delete the record of client "joshi"

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:16

Assignment to query the tables using simple form of Select statement

Syntax

select <attribute- list> from <relation- list> [where<condition>[group by <attribute list> [having <condition>] order by<attribute list>]];

e.g.Select * from emp;

Select eno name from emp where sal > 4000 order by eno;

Select sum(sal) from emp group by dno having sum(sal)> 100000;

The aggregate functions supported by as following

Name	Description	Usage	Example
Sum()	Gets the sum or total of the values of the specified attribute.	Sum(attribute-name)	Select sum(sal) from emp; Select dno, sum(sal) from emp group by dno;
Count()	Gives the count of members in the Group	Count(attribute); Count(*)	Select count(*) from emp; Select count(*) from emp where sal > 5000;
Max()	Gives the maximum value for an attribute, from a group of Members	Max(attribute)	Select max(sal) from emp; Select dno, max(sal) from emp group by dno having count(*) >10;
Min()	Gives the minimum value for an attribute, from a group of Members	Min(attribute)	Select min(sal) from emp; Select dno, min(sal) from emp group by dno having min(sal) >100;
Avg()	Gives the average value for an attribute, from a group of Members	Avg(attribute)	Select avg(sal) from emp; Select dno, avg(sal) from emp group by dno having count(*) >10;

As part of the self activity in exercise you have created a table employee with attributes empno, name, address, salary and deptno. You have also inserted atleast 10 records into the same.
To execute each query

type each query into the database prompt or

type queries in a file and cut and copy each query at the database prompt or

type queries in a file and type \i filename at SQL prompt. (all queries in the file will get executed one by one).

Execute following select queries & write the business task performed by each query.

1. Select * from emp;
2. Select empno, name from emp;
3. Select distinct deptno from emp;
4. Select * from emp where deptno = ___;
5. Select * from emp where address = 'pune' and sal > _____;
6. Select * from emp where address = 'pune and salary between _____ and _____;
7. Select * from emp where name like '---%'
8. Select * from emp where name like '%and%'
9. Select * from emp where salary is null;
10. Select * from emp order by eno;
11. Select * from emp order by deptno, eno desc;
12. Select deptno as department, sum(salary) as total from emp group by deptno order by deptno;
13. Select deptno as department , count(eno) as total_emp from emp group by deptno having count(eno) > _____ order by deptno;
14. select avg(salary) from emp;
15. select max(salary),deptno from emp group by deptno having max(sal) > _____;
16. select deptno, min(salary) from emp order by deptno;
17. update emp set salary = salary + 0.5*salary where deptno = (select deptno from department where dname = 'finance');
18. 18. update emp set deptno = (select deptno from department where dname = 'finance') Where deptno = (select deptno from department where dname = 'inventory');
19. insert into emp_backup(eno,ename) values(select eno,ename from emp);
20. delete from emp where deptno = (select deptno from department where dname='production');

Create the following relations person & area

Consider the relations Person (pnumber, pname, birthdate, income), Area(aname,area_type). An area can have one or more person living in it , but a person belongs to exactly one area. The attribute 'area_type' can have values as either urban or rural.
Create the Relations accordingly, so that the relationship is handled properly and the relations

are in normalized form (3NF).

Assume appropriate data types for all the attributes. Add any new attributes as required, depending on the queries. Insert sufficient number of records in the relations / tables with appropriate values as suggested by some of the queries**.**

Write select queries for following business tasks and execute them.

1. List the names of all people living in '_____ 'area.
2. List details of all people whose names start with the alphabet '_'
3. List the names of all people whose birthday falls in the month of _____.
4. Give the count of people who are born on '_____'
5. Give the count of people whose income is below _____
6. List names of all people whose income is between _____ and _____;
7. List the names of people with average income
8. List the sum of incomes of people living in '____'
9. List the names of the areas having people with maximum income
   (duplicate areas must be omitted in the result)
10. Give the count of people in each area
11. List the details of people living in '____' and having income greater than _____;
12. List the details pf people, sorted by person number
13. List the details of people, sorted by area, person name
14. List the minimum income of people.
15. Transfer all people living in 'pune' to 'mumbai'.
16. delete information of all people staying in 'urban' area

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment:17

Assignment to query table,using set operation (union ,intersect)

SQL Set operations :

Name	description	Syntax	Example
Union	Returns the union of two sets of values, eliminating duplicates.	<select query> Union <select query>	Select cname from depositor Union Select cname from borrower;
Union all	Returns the union of two sets of values, retaining all duplicates.	<select query> Union all <select query>	Select cname from depositor Union all Select cname from borrower;
Intersect	Returns the intersection of two sets of values, eliminating duplicates	<select query> intersect <select query>	Select cname from depositor intersect Select cname from borrower;
Intersect all	Returns the intersection of two sets of values, retaining duplicates	<select query> Intersect all <select query>	Select cname from depositor Intersect all Select cname from borrower;
except	Returns the difference between two set of values, I.e returns all values from set1 , not contained in set2 .eliminates duplicates	<select query> except <select query>	Select cname from depositor except Select cname from borrower;
Except all	Returns the difference between two set of values, i.e. returns all values from set1 , not contained in set2 .Retains all duplicates	<select query> Except all <select query>	Select cname from depositor Except all Select cname from borrower;

The relations participating in the SQL operations union, intersect & except must be compatible i.e. the following two conditions must hold :

a)The relation r and s must be of the same arity. That is , they must have the same number of attributes.

b) The domains of the i^{th} attribute of r and the i^{th} attribute of s must be the same , for all i.

Consider the following relations, non-teaching, teaching, department.
One deparment can have one or more teaching & non-teaching staff, but a teaching or non-teaching staff belongs to exactly one department. Hence dno is a foreign key in the both the relations. Create these relations in your database .

Non-teaching (empno int primary key, name varchar(20), address varchar(20), salary int,dno references department)
Teaching(empno int primary key, name varchar(20), address varchar(20), salary int,dno references department)
Department(dno int primary key,dname)

- insert at least 10 records into both the relations.
- type the following select queries & write the output and the business task performed by each query

1. Select empno from non-teaching union select empno from teaching;
2. Select empno from non-teaching union all select empno from teaching;
3. Select name from non-teaching intersect select name from teaching;
4. Select name from non-teaching intersect all select name from teaching;
5. Select name from non-teaching except select name from teaching;
6. Select name from non-teaching except all select name from teaching;

Create the following relations, for an investment firm

emp(emp-id ,emp-name, address, bdate)
Investor(inv-name , inv-no, inv-date, inv-amt)

An employee may invest in one or more investments, hence he can be an investor.
But an investor need not be an employee of the firm.
Create the Relations accordingly, so that the relationship is handled properly and the relations are in normalized form (3NF).
Assume appropriate data types for the attributes. Add any new attributes , as required by the queries. Insert sufficient number of records in the relations / tables with appropriate values as suggested by some of the queries**.**

Write the following queries & execute them.

1. List the distinct names of customers who are either employees, or investors or both.
2. List the names of customers who are either employees , or investors or both.
3. List the names of emloyees who are also investors.
5. List the names of employees who are not investors.

---------- ---------- ----------
Signature of the Instructor Remark Date

Assignment:18

Assignment to query tables using nested queries

A subquery is a select-from-where expression that is nested within another query.

Set membership	the 'in' & 'not in' connectivity tests for set membership & absence of set membership respectively.
Set comparison	the < some, > some, <= some, >= some, = some, <> some are the constructs allowed for comparison. = some is same as the 'in' connectivity. <> some is not the same as the 'not n'i connectivity. Similarly sql also provides < all, >all, <=all, >= all, <> all comparisons. <>all is same as the 'not in' construc.
Set cardinality	The 'exists' construct returns thevalue true if the argument subquery is nonempty. We can test for the non-existence of tuples in a subquery by using the 'not exists' construct.The 'not exists ' construct can also be used to simulate the set containment operation (the super set). We can write "relation A contains relation B" as "not exists (B except A)".

The complete Syntax of select statement containing connectivity or Comparison operators is as follows

```
select <attribute-list> from <relation-list>
        where <connectivity / comparison > { sub-query };
```

Create the following relation in your database(primary keys underlined)

Employee(ename, street, city)
Works(ename, company-name, salary)
Company(company-name, city)
Manages(ename, manager-name)

An employee can work in one or more companies, a company can have one or more employees working in it. Hence the relation 'works' with keyattributes as ename, company-name.

An employee manages one or more employees, but an employee is managed by exactly one employee (a recursive relationship), hence the relation 'manages' with key ename.

Insert sufficient number of records in the relations / tables with appropriate values as suggested by some of the queries**.**

Type the following queries , execute them and give the business task performed by each query

1. select ename from works w where salary >= all (select max(salary) from works));
2. select ename form works w where salary = (select max(salary) from works
3. w1 where w1.company-name = w.company-name));

4. select manager-name from manages where manager-name in
   (select ename from works where company-name = "_____");
5. select manager-name from manages where manager-name not in(select ename from works where company-name = "_____");
6. select ename from works w where salary > some (select salary from works where company-name not in (select company-name from company where city = "____"));
7. select ename from employee e where city = (select city from employee e1 , manages m where m.ename = e.ename and m.manager-name = e1.ename);
8. select * from employee where ename in (select manager-name from manages)
9. select city count(*) from employee group by city having count(*) >= all (select count(*) from employee group by city)
10. select ename from works w where salary <> all (select salary from works where ename <> w.ename);
11. select company-name, sum(salary) from works w group by company-name having sum(sal) >= all (select sum(sal) from works group by company-name)
12. select ename from employee e where city in('_____','_____');
13. select ename from employee e where city = (select city from company c, works w where w.ename = e.name and c.company-name = w.company-name);

Create the following relations :

Emp(eno,name,dno,salary)

Project(pno,pname,control-dno,budget)

Each employee can work on one or more projects, and a project can have many employees working in it. The number of hours worked on each project , by an employee also needs to be stored.

Create the Relations accordingly, so that the relationship is handled properly and the relations are in normalized form (3NF).

Assume appropriate data types for the attributes. Add any new attributes , new relations as required by the queries.

Insert sufficient number of records in the relations / tables with appropriate values as suggested by some of the queries**.**

Write the queries for following business tasks & execute them.

1. list the names of departments that controls projects whose budget is greater than ___.
2. list the names of projects, controlled by department No __, whose budget is greater than atleast one project controlled by department No __.
3. list the details of the projects with second maximum budget
4. list the details of the projects with third maximum budget.
5. list the names of employees, working on some projects that employee number __ is working.
6. list the names of employees who do not work on any project that employee number __ works on
7. list the names of employees who do not work on any project controlled by '_____' department
8. list the names of projects along with the controlling department name, for those projects which has atleast __ employees working on it.
9. list the names of employees who is worked for more than 10 hrs on atleast one

project controlled by '_____' dept.

10. list the names of employees , who are males , and earning the maximum salary in their department.

11. list the names of employees who work in the same department as '_____'.

12. list the names of employees who do not live in _____ or _____**.**

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment: 19

Assignment to query tables,using nested queries(use of exists,not exists)

SQL includes a feature for testing whether a sub query has any tuples in its result, using the following clauses:

Name	Description	Syntax	Example
Exists	The 'exists' construct returns the value true if the argument sub query is nonempty	select <attribute-list> from <relation-list> where <exists> { sub-query} ;	Select cname from borrower b where exists(select * from depositor where dname = b.cname);
Not exists	We can test for the non-existence of tuples in a sub query by using the 'not exists' Construct. The 'not exists ' construct can also be used to simulate the set containment operation (The super Set). We can write "relation A contains relation B" as "not exists (B except A)"	select <attribute-list> from <relation-list> where <not exists> { sub-query};	Select cname from borrower b where not exists(select * from depositor where dname = b.cname);

Consider the table you have prepared as part of self activity of exercise 18, Type the following queries, execute them and give the business task performed by each query

1. Select company-name from company c where not exists (select city from company where company-name = "_____" except (select city from company where company-name = c.company-name));
2. Select ename from employee e where exists (select manager-name from manages where manager-name = e.ename group by manager-name having count(*) >3);
3. Select company-name from company c where not exists (select city from company where company-name = c.company-name except (select city from company where company-name = "_____"));

4. Select ename from employee e where exists (select city from employee where city = e.city and ename <> e.ename group by city having count(*) > 5)

5. Select company-name from company c where not exists (select company-name from company where city = c.city and company-name <> c.company-name)

Consider the table you have prepared as part of Assessment work set A of exercise 18, Type the following queries, execute them and give the business task performed by each query

1. List the names of employees who work in all the projects that "_____" works on.
2. List the names of employees who work on only some projects that "_____" works on
3. List the names of the departments that have at least one project under them. (Write using 'exists 'clause)
4. List the names of employees who do not work on "sales" project (write using 'not exists') clause
5. List the names of employees who work only on those projects that are controlled by their department.
6. List the names of employees who do not work on any projects that are controlled by their department.

---------- ---------- ----------

Signature of the Instructor Remark Date

Assignment: 20

Assignment related to small case studies (Each case study will involve creating tables with specified constraints, inserting records to it & writing queries for extracting records from these tables)

You should read following topics before starting this exercise 1.
All the assignments from 11 to 19

Steps in solving a case study:

1. Read through the given case study carefully.
2. Create the given relations in the database. The database thus created should be in 3NF. (no data duplication, appropriate handling of the relationships)
3. Insert sufficient number of records in the relations / tables.
4. Create a new file with all the select queries in it.
5. To execute each query .

1. Consider the following case study:

A 4-wheeler rental company needs to develop a database to store the following information:
The information about the cars , like the registration number, the chassis number, the type of the vehicle (car, jeep, SUV etc). the vehicles may have one or more luxurious features like AC, Stereo, tape, DVD player etc).
The company also needs to maintain the information about its drivers like driver license no, name, address, age etc.
A car is driven by different drivers on different days , a driver may drive different cars on different days . The company also needs information regarding the different places to which the car had been driven down, the names of drivers who have driven it to these places along with the name of customers who had booked the car to that place. The information of the different destinations to which the cars from this company can be driven down, also needs to be stored.
Regarding customers, customers can book more than one car to a place. The customers are allowed to book multiple cars to different places, in a single booking transaction. The name, address, no of passengers travelling in the car, the destination, the rental cost etc needs to be stored.

The following constraints are to be defined for the vehicles, drivers, and destination places:

a) The vehicle make should be after the year 2000.
b) Only vehicles of maruti, Tata are used by the company
c) Drivers should be above 20 years of age
d) Drivers should be staying in “Pune” city
e) The destination places should be within 500km radius from Pune.

Design the relational database for the above company, so that the following queries can be answered:

1. List the names of drivers who have driven a car to “Mumbai”
2. List the name of customers who have booked a “SUV” to “Satara”

3. List the names of customers who have booked cars to Pune or Mumbai or Lonavla
4. List the details of cars that have never driven down to “Mumbai”
5. List the details of the place to which maximum number of customers have driven down.
6. List the details of the driver who have driven all the vehicles of the company.
7. List the names of the drivers who have driven at least two cars to “Mumbai
8. List the names of drivers who have also driven some vehicles to “Mumbai”
9. List the details of customers who have booked more than two vehicles to “Solapur”
10. List the names of customers who have booked maximum number of vehicles

2. Consider the following case study:

An insurance agent sells policies to clients. Each policy is of a particular type like vehicle insurance, life insurance, accident insurance etc, and there can be many policies of a particular type. Each policy will have many monthly premiums, and each premium is associated to only one policy. Assume appropriate attributes for agents, policy, premiums and policy-types.

The following constraints have to be defined on the relations

a. The policy types can be only accident, life and vehicle.
b. b. The agents can be only from Pune, Mumbai and Chennai.
c. c. The policy amount should be greater than 20000.
d. The policy-sale-date should be greater than the policy-intro-date.

Design the relational database for the above , so that the following queries can be answered:

1. List the names of agents living in ‘_____’
2. List the names of policy holders , who have bought policies from the agent ‘joshi’
3. List the names of policyholders, who have bought more than two policies from ‘joshi’
4. List the names of agents, who have sold policies to only customers who live in their own City.
5. List the names of agents who have sold at least two policies.
6. List the names of cities, which have the maximum number of agents.
7. List the names of customers who have bought the maximum number of policies.
8. List the details of all premiums , paid or the policy number _____
9. Update all policy amount to _____for all policies bought by customers from _____ city.
10. Delete all policies , bought from ‘joshi’
11. _____
12. _____
13. _____
14. _____
15. _____

Instructor should fill in the blanks with appropriate values.

3. Consider the following case study:

A movie studio wants to develop a database to manage their office information , related to movies, actors, directors, producers.

The following facts are relevant

a. Each actor has acted in one or more movies
b. Each director has directed many movies.
c. Each producer has produced many movies.
d. Each movie is directed by one and only one director, but can be produced by more than one producers.
e. Each movie has one or more actors acting in it, in different roles.
f. Each actor & director has several addresses. Each address is made up of a house-no, street, city, state.

The following constraints are defined on the relations.

a. Each movie can have a maximum budget of 10 lakhs
b. Each actor can charge a maximum of Rs. 10 lakhs for a movie.
c. The roles that an actor can act in a movie can be any of the following : villain, hero, heroine, support.

Design the relational database for the above, so that the following queries can be answered:

1. List the names of movies in which _____ have acted.
2. List the names of actors who have acted in at least one movie, in which shahrukh has acted.
3. List the names of actors who have acted in every movie in which ---------- has acted.
4. List the names of actors who have acted as a 'villain' in every movie, in which the actor has acted.
5. List the names of movies with the highest budget.
6. List the names of movies with the second highest budget.
7. List the names of actors who have acted in the maximum number of movies.
8. _____
9. _____
10. _____
11. Update the address of producer _____ . set the city to _____
12. Delete information of all actors who have an address in pune.
13. _____
14. _____
15 List the names of movies, produced by more than one producer.

Instructor should fill in the blanks with appropriate values.

4. Consider the following case study:

A music company wants to go in for automation of their requirements. They want to develop a database for maintaining the information of their music albums, singers, musicians, instruments. The following facts are relevant:

a. Each album is produced by many musicians, a musician can produce many albums
b. A singer can sing for many albums, but an album consists of songs of only one singer.
c. A musician can play many instruments, an instrument can be played by many musicians.

The following constraints are to be placed on the relations

a. each musician is paid a minimum of 50000 Rs. for each album
b. all singers are from either pune, Mumbai or Chennai
c. each instrument cost is maximum 10000

Design the relational database for the above, so that the following queries can be answered:

1. _____
2. List the names of musicians who have played guitar for the album _____
3. List the names of musicians who palsy at least one instrument same as the one “joshi” plays.
4. List the names of albums , in which “_____” has sung.
5._____
6._____
7. List the names of albums released in 1998
8. List the names of albums that have more than two instruments being played in it
9. Delete all information of singers who have not sung in any album
10. Delete all information of musicians , who have worked in the album “_____”

Instructor should fill in the blanks with appropriate values.

5. Consider the following case study:

A housing society needs to manage the administrative information related to the society.
The society is made up of different types of flats like 2BHK, 1BHK, 3BHK. Each type has a well defined square-feet area. The outright sale rate & the rental value of the flat depend on the type of the flat. Each flat has a single owner. Each owner can have one or more flats in his name. The name, address, phone etc of the owner need to be maintained. For each flat, its type, the floor no, any internal specifications needs to be maintained.
The society also contains a club-house, which is rented out to flat owners, at a nominal rate for conducting various functions / programmes. Society would like to print reports like number of functions held in the club-house during a month / period etc.
Every month maintenance amount is collected from the owners of the flats. Society needs to maintain this finance information, like how much amount collected for a month, whether any defaulters for a month, sending reminders to the defaulters etc.
The expenditure information includes money spent on maintenance of the society like paying the sweepers, cleaners of the common area of the society, any emergency expense, salaries of the security etc.

Every month the society would like to print a report of expenditure versus collection.

Design the relational database for the above , so that the following queries can be answered:
1. List the flats of 2bhk type.
2. List the 3bhk flats that are currently vacant.
3. List the functions held in clubhouse during the month of “_____”
4. List the names of owners, who have never conducted any functions in the clubhouse.
5. List the payment defaulters for the month of “April”
6. List the total expenditure for the month of _____

7. List the month with the least expenditure.
8. Transfer the flat in the name of _____ to _____
9. _____
10. List the names of owners , who own both a 2bhk and a _____
11. _____
12. _____

---	---	---
Signature of the Instructor	Remark	Date