# VAISHNO COLLEGE OF ENGINEERING

**Affliated to HPTU, Hamirpur and approved by AICTE**

# DBMS

# Lab Manual

# CSPC-415P (CPCS Syllabus)

**Department of Computer Science Engineering**

**VillThapkour, PO Bhardoya, Tehsil Indora,Distt. Kangra (HP)-176403**

**Contact: 094183-18394, Web: www.vaishno.edu.in**

## Vision of Institute

To emerge as an institute of eminence in the fields of engineering, technology and management in serving the industry and the nation by empowering students with a high degree of technical managerial and practical competence.

## Mission of Institute

M1 To strengthen the theoretical, practical and ethical dimensions of the learning process by fostering a cultural of research and innovation among faculty members and students.

M2 To encourage long term interaction between academia and industry through the involvement of industry for hands on implementation of the curriculum.

M3 To strengthen and molding students in professional ethical, social and environmental dimensions by encouraging participation in co-curricular extracurricular and CSR activities.

## Vision of Institute

To emerge as an institute of eminence in the fields of engineering, technology and management in serving the industry and the nation by empowering students with a high degree of technical managerial and practical competence.

## Mission of Institute

M1 To strengthen the theoretical, practical and ethical dimensions of the learning process by fostering a cultural of research and innovation among faculty members and students.

M2 To encourage long term interaction between academia and industry through the involvement of industry for hands on implementation of the curriculum.

M3 To strengthen and molding students in professional ethical, social and environmental dimensions by encouraging participation in co-curricular extracurricular and CSR activities.

## Vision of the Department

To emerge as a department of eminence in computer science and engineering in serving the industry and the nation by empowering students with high degree of technical and practical competence.

## Mission of the department

M1 To strengthen the theoretical and practical aspects of learning process by strongly encouraging a computer cultural of research, innovation and hands on learning in computer science and engineering

M2 To encourage long term interaction between the department and IT industry, through the involvement of IT industry for hands on implementation of course curriculum.

M3 To widen the awareness of students in professional, ethical, social and environmental dimensions by encouraging their participation in co-curricular extracurricular and CSR activities.

## Program Educational Objectives (PEOs) of the department

**PEO 1:** Engage in successful careers in industry, academia, and public service, by applying the acquired knowledge of Science, Mathematics and Engineering, providing technical leadership for their business, profession and community

**PEO 2:** Establish themselves as entrepreneur, work in research and development organization and pursue higher education

**PEO 3:** Exhibit commitment and engage in lifelong learning for enhancing their professional and personal capabilities.

## PROGRAM OUTCOMES

**PO1: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2: Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4: Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO6: The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7: Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO9: Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10: Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able tocomprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11: Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO 12: Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**Program Specific Outcome (PSOs)**

**PSO1:** Apply knowledge of mathematics, engineering sciences and multidisciplinary knowledge to the solution of computer science engineering problems.

**PSO2:** Apply research-based knowledge, appropriate techniques, IT tools to complex computer science engineering problems including design, analysis, interpretation of data, and synthesis of the information to provide valid conclusions.

**PSO3:** Apply ethical principles engineering profession and recognize the need of independent and lifelong learning for professional development and personnel growth.

# Lab Syllabus & List of Experiments

| CSPC-415PDBMS Lab | | | | | | | |
|---|---|---|---|---|---|---|---|
| Teaching Scheme | | | Credit | MarksDistribution | | | Durationof EndSemester Examination |
| L | T | p | C | Internal Assessment | End Semester Examination | Total | |
| 0 | 0 | 2 | 1 | MaximumMarks:30 | MaximumMarks:20 | 50 | 2Hours |
| | | | | MinimumMarks:12 | MinimumMarks:08 | 20 | |

## Course Objectives:

- To present an introduction to database management systems using programming.
- To provide skills for writing programs.
- Familiar with basic database storage structures and access techniques.

## Course Outcomes:

- Describe the fundamental elements of relational database management systems.
- Design ER-models to represent simple database application scenarios.
- Improve the database design by normalization.

## List of Experiments

1. Design a Database and create required tables. For e.g .Bank, College Database
2. Apply the constraints like Primary Key ,Foreign key, NOT NULL to the tables.
3. Write a SQL statement for implementing ALTER, UPDATE and DELETE
4. Write the queries to implement the joins
5. Write the query for implementing the following functions: MAX(),MIN(), AVG(), COUNT().
6. Write the query to implement the concept of lntegrity constraints.
7. Write the query to create the views.
8. Perform the queries for triggers.
9. Perform the following operation for demonstrating the insertion, updation and deletion using the referential integrity constraints.
10. Write the query for creating the users and their role.

## Evaluation Scheme

**Internal Assessment: 30 marks (pass marks:12)**

Distribution of marks for internal assessment:

- Written/presentation/Demonstration: 05
- Viva-voice: 05
- Teacher assessment: Lab Work performance/Report/File Work:15
- Attendance: 05

**External Assessment: 20 marks (pass marks: 08)**

**Total marks 30+20=50, Pass marks = 20**

**Note: Student has to pass internal & external assessment separately.**

## GENERAL GUIDELINES AND SAFETY INSTRUCTIONS

1. You may use the computers in the lab only when a teacher is present.
2. Please place your bags at the front of the lab.
3. Do not eat or drink in the lab.
4. Keep the lab clean and neat at all times.
5. Use only the computer you are assigned to.
6. Report any hardware fault immediately to your teacher. Never attempt to dismantle the different parts of the computer.
7. Each student must log in to his/her account. No sharing of accounts is permitted.
8. The computers are for your academic use. Playing computer games for entertainment is strictly not allowed.
9. Shut down the computer properly after use.
10. Do not charge your personal mobile devices in the lab.

### Cleanliness

- Keep your workspace clean and free of clutter
- Don't eat or drink in the lab
- Don't litter
- Don't remove cables or items from the lab

### Fire safety

- Have a fire extinguisher and first-aid kit available
- Follow fire safety guidelines
- Be aware of the possibility of an accidental fire
- Know how to react to a fire
- Have a planned fire escape route

### Eye and body safety

- Avoid eye fatigue by blinking often or closing your eyes for a few minutes
- Sit straight and in a comfortable posture
- Spread your fingers apart or rotate your wrists at regular intervals
- Wear proper lab attire
- Practice good hygiene

### **Other safety guidelines**

- Don't spill liquids on the computer

- Don't touch hot or high voltage areas of printers
- Don't open a power supply or CRT monitor
- Don't tamper with wires or network cables
- Don't use illegal software
- Don't attempt to compromise network security

## Experiment No: 1

**Aim:** Design a Database and create required tables. For e.g .Bank, College Database
**SOFTWARE/APPARATUSREQUIRED:-** MySQL**,** Personal Computer

**Program :**

```
CREATE TABLE User (
    UserID INT PRIMARY KEY,
    Username VARCHAR(255),
    Email VARCHAR(255),
    Password VARCHAR(255),
    UserType VARCHAR(50)
);

CREATE TABLE Course (
    CourseID INT PRIMARY KEY,
    CourseName VARCHAR(255),
    Description TEXT,
    Price DECIMAL(10, 2)
);

CREATE TABLE CourseContent (
    ContentID INT PRIMARY KEY,
    CourseID INT,
    ContentType VARCHAR(50),
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)
);

CREATE TABLE Enrollment (
    EnrollmentID INT PRIMARY KEY,
    UserID INT,
    CourseID INT,
    EnrollmentDate DATE,
    CompletionStatus VARCHAR(50),
    FOREIGN KEY (UserID) REFERENCES User(UserID),
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)
);

CREATE TABLE Payment (
    PaymentID INT PRIMARY KEY,
    UserID INT,
    PaymentDate DATE,
    Amount DECIMAL(10, 2),
    PaymentMethod VARCHAR(50),
    FOREIGN KEY (UserID) REFERENCES User(UserID)
);

CREATE TABLE Result (
    ResultID INT PRIMARY KEY,
    UserID INT,
    CourseID INT,
    QuizID INT,
    Score INT,
    FOREIGN KEY (UserID) REFERENCES User(UserID),
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID),
    FOREIGN KEY (QuizID) REFERENCES Quiz(QuizID)
);
```

```
CREATE TABLE Quiz (
    QuizID INT PRIMARY KEY,
    CourseID INT,
    QuizName VARCHAR(255),
    Description TEXT,
    TotalMarks INT,
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)
```

# Experiment No: 2

**AIM:** Apply the constraints like Primary Key, Foreign key, NOT NULL to the tables.

**SOFTWARE/APPARATUSREQUIRED:-** MySQL**,** Personal Computer

**Program:**

```
CREATE TABLE Student
(
ID int(6) NOT NULL UNIQUE,
NAME varchar(10),
ADDRESS varchar(20),
PRIMARY KEY(ID)
);

CREATE TABLE Student
(
ID int(6) NOT NULL UNIQUE,
NAME varchar(10),
ADDRESS varchar(20)
);



CREATE TABLE Orders

(

O_ID int NOT NULL,

ORDER_NO int NOT NULL,

C_ID int,

PRIMARY KEY (O_ID),

FOREIGN KEY (C_ID) REFERENCES Customers(C_ID)

)
```

## Experiment No: 3

**AIM**: Write a SQL statement for implementing ALTER, UPDATE and DELETE.

**SOFTWARE/APPARATUSREQUIRED:-** MySQL**,** Personal Computer

**Program:**

```
CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
ALTER TABLE Customers
ADD Email varchar(255);

UPDATE Customers
SET ContactName='Juan'
WHERE Country='Mexico';

DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
```

## Experiment No: 4

**AIM**: Write the queries to implement the joins

**SOFTWARE/APPARATUSREQUIRED:-** MySQL**,** Personal Computer

**Program**

```
join the Customers and Orders tables
-- based on the common values of their customer_id columns

SELECT Customers.customer_id, Customers.first_name, Orders.item
FROM Customers
JOIN Orders
ON Customers.customer_id = Orders.customer_id;
SELECT columns_from_both_tables
FROM table1
JOIN table2
ON table1.column1 = table2.column2
-- join Customers and Orders tables based on
-- customer_id of Customers and customer column of Orders

SELECT Customers.customer_id, Customers.first_name, Orders.amount
FROM Customers
JOIN Orders
ON Customers.customer_id = Orders.customer;
```

# Experiment No: 5

**AIM**: Write the query for implementing the following functions: MAX(),MIN(), AVG(), COUNT().

**SOFTWARE/APPARATUSREQUIRED:-** MySQL**,** Personal Computer

**Program**

The SQL aggregate functions — AVG, COUNT, DISTINCT, MAX, MIN, SUM — all return a value computed or derived from one column's values, after discarding any NULL values. The syntax of all these functions is:

```
SELECT AGGREGATE_FUNCTION( 'column_name' )

FROM TABLE_NAME
```

The aggregate functions are:

| **function** | **returns** |
|---|---|
| AVG() | the mean average of the elements in the column |
| COUNT() | the total number of elements in the column |
| DISTINCT() | the number of distinct values across the column |
| MAX() | the largest-value element in the column |
| MIN() | the smallest-value element in the column |
| SUM() | the arithmetic total of all values in the column |

We'll examine the workings of the SQL aggregate functions by interacting with some menu items and prices from the former Brasserie Les Halles, perhaps best-known for it's one-time executive chef, Anthony Bourdain. (The SQL needed to create and populate the tables are given at the bottom of this blog entry.)

### AVG() — The Mean Average

The AVG() aggregate function returns the mean average of the numeric values in the specified column. For example,

```
SELECT AVG(price) FROM les_halles_menu ;
```

| AVG(price) |
|---|
| 17.4706 |

To constrain the returned result to two decimal places, as is typically expected for currency, use the FORMAT() function — FORMAT(AVG(price), '2') returns 17.47. To have the value formatted properly for a given locale use something like FORMAT(AVG(price), '2', 'fr_FR'), which returns 17,47 (as is expected in France) (The fr_FR is composed from ISO-3166 Country Codes and ISO-639 Language Codes.) Check the syntax for your chosen database implementation as many variations may be available to you.

### COUNT — How Many Exist?

The COUNT() aggregate function returns the number of items in the column. For example, to count the number of the French-language menu items:

```
SELECT COUNT(item_fr) FROM les_halles_menu ;
```

| COUNT(item_fr) |
|---|
| 17 |

### DISTINCT — How Many Unique?

The DISTINCT() aggregate function returns the set of unique values in the column, discarding multiples. For example, to see exactly one of each of the prices appearing on the menu,

```
SELECT DISTINCT(price) FROM les_halles_menu ;
```

| price |
|---|
| 16 |
| 25 |
| 26 |

| |
|---|
| 32 |
| 30 |
| 24 |
| 9 |
| 10 |
| 11 |
| 8 |
| 15 |

**MAX & MIN — The Largest and Smallest**

The MAX() aggregate function returns the largest value from the column. For example, to find the most expensive menu item:

```
SELECT MAX(price) FROM les_halles_menu ;
```

| MAX(price) |
|---|
| 32 |

MAX() also works on text values; to find the last item in the names of the English-language menu items:

```
SELECT MAX(item_en) FROM les_halles_menu ;
```

| MAX(item_en) |
|---|

| |
|---|
| Truffle Fries |

MIN() provides the same functionality, with a focus on the smallest values.

**SUM — All Together Now**

The SUM() aggregate function returns the sum of all the numeric values from the column. For example, to calculate the total cost of a dinner consisting of one of each menu item:

```
SELECT SUM(price) FROM les_halles_menu ;
```

| **SUM(price)** |
|---|
| 297 |

## Experiment No: 6

**AIM**: Write the query to implement the concept of lntegrity constraints
**SOFTWARE/APPARATUSREQUIRED:-** MySQL**,** Personal Computer

**Program:**

```
CREATE TABLE students (
    student_id INT PRIMARY KEY,
    first_name TEXT NOT NULL,
    last_name TEXT NOT NULL,
    email TEXT UNIQUE NOT NULL,
    major TEXT DEFAULT 'Undeclared',
    enrollment_year INT,
    CONSTRAINT year_check CHECK (enrollment_year >= 1900),
    CHECK (major IN (
        'Undeclared',
        'Computer Science',
        'Mathematics',
        'Biology',
        'Physics',
        'Chemistry',
        'Biochemistry'
    ))
);
```

# Experiment No: 7

**AIM**: Write the query to create the views.
**SOFTWARE/APPARATUSREQUIRED:-** MySQL**,** Personal Computer

**Program:**

```
CREATE TABLE StudentDetails (
   S_ID INT PRIMARY KEY,
   NAME VARCHAR(255),
   ADDRESS VARCHAR(255)
);

INSERT INTO StudentDetails (S_ID, NAME, ADDRESS)
VALUES
   (1, 'Harsh', 'Kolkata'),
   (2, 'Ashish', 'Durgapur'),
   (3, 'Pratik', 'Delhi'),
   (4, 'Dhanraj', 'Bihar'),
   (5, 'Ram', 'Rajasthan');
```

## Experiment No: 8

**AIM**: perform the queries for triggers
**SOFTWARE/APPARATUSREQUIRED:-** MySQL**,** Personal Computer

**Program**:

```
CREATE TABLE employees (
    employee_id INT,
    name VARCHAR(100),
    department VARCHAR(100)
);
CREATE TABLE employees_log (
    employee_id INT,
    name VARCHAR(100),
    action VARCHAR(100)
);
INSERT INTO employees (employee_id, name, department)
VALUES (1, 'Alice', 'HR'), (2, 'Bob', 'IT'), (3, 'Charlie', 'Sales'), (4, 'David', 'IT');
```

## Experiment No: 9

**AIM**: Perform the following operation for demonstrating the insertion, updation and deletion using the referential integrity constraints.
**SOFTWARE/APPARATUSREQUIRED:-** MySQL**,** Personal Computer

**Program:**

```
-- Create StudentDetails table
CREATE TABLE departments (
  department_id INT PRIMARY KEY,
  department_name VARCHAR(255)
);

// Insert some data into the departments table
INSERT INTO departments (department_id, department_name)
VALUES (1, 'IT'), (2, 'HR'), (3, 'Sales');

// Create the employees table with a foreign key constraint
CREATE TABLE employees (
  employee_id INT PRIMARY KEY,
  employee_name VARCHAR(255),
  department_id INT,
  FOREIGN KEY (department_id) REFERENCES departments(department_id)
);

// Insert some data into the employees table
INSERT INTO employees (employee_id, employee_name, department_id)
VALUES (1, 'John', 1), (2, 'Jane', 2), (3, 'Jack', 3);

update
// Update the department_id of the HR department to 4
UPDATE departments
SET department_id = 4
WHERE department_id = 2;

// Update the department_id of employees in the HR department to 4
UPDATE employees
SET department_id = 4
WHERE department_id = 2;
```

## A list of common DBMS viva questions includes:

### Basic Concepts:

What is a DBMS and what is its utility?

What is a database?

Explain the advantages of using a DBMS compared to traditional file-based systems.

What are the different languages present in DBMS (e.g., DDL, DML, DCL)?

### Data Integrity and Consistency:

What is data integrity and how does a DBMS enforce it?

Explain different types of integrity constraints (entity integrity, referential integrity)

What is data consistency and how does a DBMS maintain it?

### Database Design:

Explain the concept of normalization and its different forms

What are the different types of keys in a database (primary key, foreign key, candidate key)?

How would you design a database for a given scenario (e.g., an online store)?

### Concurrency Control:

What is concurrency control and why is it important?

Explain different concurrency control mechanisms (locking, timestamping)

What are the ACID properties and their significance in database transactions?

### Data Access and Manipulation:

Explain the concept of SQL and its different operations (SELECT, INSERT, UPDATE, DELETE)

What are different types of joins in SQL?

How would you write a query to retrieve specific data from a database?

### Data Independence:

What is data independence and its different levels (physical, logical, external)?

How does data independence benefit database management?

### Security and Backup:

How does a DBMS implement data security measures?

What are the different types of database backups and their importance?

Explain the concept of user access control in a DBMS

**DBMS Types and Architecture:**

What are the different types of DBMS (relational, hierarchical, network, object-oriented)?

Explain the three-tier database architecture

Laboratory Experiment Evaluation Rubric

| **Category** | **Outstanding (Up to 100%)** | **Accomplished (Up to 75%)** | **Developing (Up to 50%)** | **Beginner (Up to 25%)** |
|---|---|---|---|---|
| **Written/Presentation/Demonstration** | The write-up is clear, well-organized, and follows the prescribed format. All required sections (aim, apparatus, theory, procedure, diagram, etc.) are present and well-written. Demonstration is clear and thorough. | The report follows the specified format, but some sections (like the diagram or theory) are missing or incomplete. The demonstration is understandable but lacks depth. | The report includes most sections but lacks clarity, coherence, or completeness in some parts (e.g., diagram missing, unclear theoretical explanation). The demonstration is incomplete or unclear. | The report is poorly written and organized. Many sections are missing or incorrect (e.g., no diagram, incomplete procedure). The demonstration lacks clarity or is missing. |
| **Viva-Voice** | Demonstrates a deep understanding of the experiment, underlying principles, and outcomes. Answers questions confidently and accurately. | Demonstrates a general understanding of the experiment and principles but struggles with some aspects. Provides correct answers to most questions. | Struggles with some fundamental concepts and principles. Answering questions requires additional prompts, with a few errors in understanding. | Lacks a basic understanding of the experiment. Unable to answer most questions accurately. Demonstrates significant gaps in knowledge. |

| Category | Outstanding (Up to 100%) | Accomplished (Up to 75%) | Developing (Up to 50%) | Beginner (Up to 25%) |
|---|---|---|---|---|
| **Performance/Report/File Work** | Performs the experiment accurately and efficiently. The report is thorough, with correct observations, calculations, and analysis. Data is recorded neatly and with appropriate units. All relevant calculations and interpretations are included. | Performs the experiment well with minor errors or delays. The report is complete but may contain some inaccuracies or missing components in calculations or observations. | Completes the experiment but with notable mistakes, either in the setup or the data. The report has several missing or inaccurate components, including incorrect or incomplete calculations. | Struggles to perform the experiment correctly. Significant errors in setup, data collection, and analysis. The report is poorly structured with major inaccuracies or missing sections. |
| **Attendance** | Consistently attends all lab sessions, actively participates, and engages with the experiment and group discussions. | Attends most lab sessions with occasional absences. Participation is generally good but lacks consistency or depth. | Attends some lab sessions but has frequent absences or minimal participation. | Misses several lab sessions and shows minimal to no participation in class or group activities. |