****

**Format No. QSP/7.1/01.F01 (B) Issue No.05 Rev. No 5 Dated: Jan 1, 2017**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**UNIVERSITY OF PETROLEUM & ENERGY STUDIES**

**College of Engineering Studies**

**Dehradun**

**COURSE PLAN**

Programme : B. Tech (CS-CCVT)

Course : Advanced Database Management Systems Lab

Course Code : CSEG 236

No. of credits : 1

Semester : III

Session : 2018-19

Batch : 2017-21

Prepared by : Dr. Jagdish Chandra Patni

Email : [jcpatni@ddn.upes.ac.in](mailto:jcpatni@ddn.upes.ac.in)

Co-Faculties : Dr. Hitesh Kumar Sharma, Mr. Vishwas Rathi, Mr. Pardeep Singh, Mr. Sudeep varshney, Ms. Richa Chaudhary, Ms. Nitika Goenka, Ms. Ankita Shukla, Ms. Gaytri Bakshi

**Approved By**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Subject Faculty HOD

UPES Campus Tel : +91-135-2770137

“Energy Acres” Fax : +91 135- 27760904

P.O. Bidholi, Via Prem Nagar, Dehradun Website : [www.upes.ac.in](http://www.upes.ac.in)

**COURSE PLAN**

1. **PREREQUISITE:**
   1. Elementary knowledge of Database Management System
   2. Basic knowledge of SQL.
2. **PROGRAM OUTCOMES (POs) and PROGRAM SPECIFIC OUTCOMES (PSOs) for CCVT:**
3. **B1. PROGRAM OUTCOMES (POs)**

Engineering Graduates will be able to:

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, review 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 to comprehend 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.

PO12. 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.

1. **B2. Program Specific Outcomes (PSOs)**

Computer Science Engineering with specialization in Cloud Computing and Virtualization Engineering Graduates will be able to:

PSO1. Perform system and application programming using computer system concepts, concepts of Data Structures, algorithm development, problem solving and optimizing techniques,

PSO2. Apply software development and project management methodologies using concepts of front-end and back-end development and emerging technologies and platforms.

PSO3: Ability to understand and apply Cloud Computing architecture for scalable, secure and dynamically provisioned business oriented environment with optimized performance tuning and data reliability.

1. **COURSE OUTCOMES FOR AUTOMOTIVE TRANSMISSION SYSTEMS: At the end of this course student should be able to**

CO1. Able to work on Oracle Databases.

CO2: Understand the implementation of searching methods of the Database Management Systems.

CO3: Learn PL/SQL Programming

CO4: Learn the implementation & working of Cursors & Triggers.

**Table: Correlation of POs and PSOs v/s COs**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PO/CO | PO  1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO  9 | PO  10 | PO  11 | PO  12 | PSO  1 | PSO  2 | PSO  3 |
| CO1 |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |
| CO2 | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| CO3 |  |  |  |  |  |  |  |  |  |  |  |  | 3 |  |  |
| CO4 |  |  |  |  |  |  |  |  |  |  |  |  | 3 |  |  |

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

1. **PEDAGOGY**

* **Power Point Presentation,**
* **Experimental learning with continuous evaluation,**

1. **COURSE COMPLETION PLAN**

|  |  |
| --- | --- |
| **Total Lab sessions** | 12 |
| **Total Quizzes/ Viva** | 03 |
| **Total Test** | 01 |

One Session = 120 minutes

1. **EVALUATION & GRADING**

Students will be evaluated based on the following 3 stages.

1. Continuous Evaluation - 40%
2. Attendance & General Discipline - 20%
3. Test, Viva & Quiz - 40%

**F1. CONTINUOUS EVALUATION**

|  |  |  |
| --- | --- | --- |
| S.No | Description | % of Weightage out of 40 |
| 1 | Performance & Records | 80% |
| 2 | Lab File | 20% |

Lab performance and record evaluation shall be a continuous process throughout the semester. Minimum three Viva voice/ Quiz based on practical sessions shall be conducted during the semester.  A student should attend at least two Viva voice/ Quiz otherwise he/she will be failed.

**F2. Performance Records:**

|  |  |
| --- | --- |
| Example ( 10 Marks for each Experiment ) | |
| Algorithm Design | 3 Marks |
| Coding Syntax | 2 Marks |
| Execution /Bug Finding | 2 Marks |
| Records | 3 Marks |

**F3. Viva voce / Quiz**

External lab examination shall be of 2 hours duration. The lab examination will be based on the lab classes and practical oriented problems.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Viva of 10-15 students in one lab session So one complete viva is in 3 Lab Sessions. | | | | | | | |
| 3 Viva Means 9-10 session as Continuous evaluations in a semester. | | | | | |  |  |
| Or take 1 Quiz & 2 Viva |  |  |  |  |  |  |  |

**F4. GENERAL DISCIPLINE:** Based on student’s regularity, punctuality, sincerity and participation in the interactions.

*The marks obtained by the students will be displayed on Blackboard at the end of semester.*

**F6. GRADING:**

The overall marks obtained at the end of the semester comprising all the above three mentioned shall be converted to a grade.

1. **COURSE DELIVERY PLAN**
2. **EXPERIMENT-1**

**Title: Working with Oracle, Working with Views, Read only and updatable views**

* 1. Create the Tables STUDENT (sid, firstname, lastname, address, contact, courseid) and COURSE(courseid, cname, cduration, instructor).
     1. Design the tables with Referential and Domain constraints as applicable (like primary key, not null , foreign key). Input some sample records into both the tables.
     2. Delete Record 2 from COURSE table
     3. Delete Record 2 from STUDENT table
     4. Drop the COURSE Table
     5. Drop the STUDENT table
  2. Create table of table name: EMPLOYEES

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Width | Attributes |
| Employee\_id | Character | 10 | PK |
| First\_Name | Character | 30 | NN |
| Last\_Name | Character | 30 | NN |
| DOB | Date |  |  |
| salary | Number | 25 | NN |
| Department\_id | character | 10 |  |

1. Create View of name emp\_view and the column would be Employee\_id, Last\_Name, salary and department\_id only.
2. Insert values into view(remove the NOT NULL constraint and then insert values)
3. Modify, delete and drop operations are performed on view.
4. creates a view  named salary\_view. The view shows the employees in department 20 and their annual salary.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1. **EXPERIMENT-2**

**Title: Working with various Indexes**

* 1. Create an index of name employee\_idx on EMPLOYEES with column Last\_Name, Department\_id .
  2. Find the ROWID for the above table and create a unique index on employee\_id column of the EMPLOYEES.
  3. Create a reverse index on employee\_id column of the EMPLOYEES.
  4. Create a unique and composite index on employee\_id and check whether there is duplicity of tuples or not.
  5. Create Function-based indexes defined on the SQL functions UPPER(column\_name) or LOWER(column\_name) to facilitate case-insensitive searches(on column Last\_Name).
  6. Drop the function based index on column Last\_Name.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**3. EXPERIMENT-3**

**Title: Working with Sequences**

1. Create a sequence by name EMPID\_SEQ starting with value 100 with an interval of 1.
2. Write a SQL for finding the current and the next status of EMPID\_SEQ.
3. Change the Cache value of the sequence EMPID\_SEQ to 20 and maxvalue to 1000.
4. Insert values in employees table using sequences for employee\_id column.
5. Drop sequence EMPID\_SEQ. Create a sequence called REVERSE to generate numbers in the descending order from 10000 to 1000 with a decrement of 5.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**EXPERIMENT-4**

**Title: PL/SQL Programming - 1**

1. Write a PLSQL code block to swap two numbers without taking third variable.
2. Write a PL/SQL code to accept the value of A, B & C display which is greater.
3. Using PL/SQL Statements create a simple loop that display message “Welcome to PL/SQL Programming” 20 times.
4. Write a PLSQL code block to reverse the string.
5. Write a PL/SQL code block to check whether the given number is Armstrong number or not.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**EXPERIMENT-5**

**Title: PL/SQL Programming - 2**

1. Write a PL/SQL code block to find total and average of 6 subjects and display the grade.
2. Write a PLSQL code block to accept table name and display number of rows in a table.
3. Write a PLSQL code block generate table of a given number and store result in two column table.
4. Write a PLSQL code block to generate simple interest and store Principal, Interest rate, time and simple Interest in a table.
5. Write a PL/SQL Code block program to accept a deptno and display who are working in that dept?
6. Consider the following table to write PL/SQL code as specified under
   * Teacher (t\_no, f\_name, l\_name, salary, supervisor, joiningdate, birthdate, title)
   * Class (class\_no, t\_no, room\_no)
   * Payscale (Min\_limit, Max\_limit, grade)
7. Accept a range of salary and print the details of teachers from teacher table.
8. Calculate the bonus amount to be given to a teacher depending on the following conditions:
   1. if salary > 10000 then bonus is 10% of the salary.
   2. if salary is between 10000 and 20000 then bonus is 20% of the salary.
   3. if salary is between 20000 and 25000 then bonus is 25% of the salary.
   4. if salary exceeds 25000 then bonus is 30% of the salary.
9. Using a simple LOOP structure, list the first 10 records of the ‘teachers’ table.
10. Accept the room number and display the teacher details like t\_no, f\_name, l\_name,
    1. birthdate, title from table Teacher.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**EXPERIMENT- 6**

**Title: Working with Functions/Procedures**

1. Write a PL/SQL function that accepts 2 numbers and returns the addition of passed values.
2. Write a PL/SQL function that accepts number and returns the reverse of that number.
3. Write a PL/SQL procedure to calculate the sum of first N number where N is passed as parameter.
4. Write a PL/SQL function that accepts department number and returns the total salary of the department. Use table EMPLOYEES(as given in Experiment 1- Q.6)
5. Write a PL/SQL procedure to display the average salary of each department. Use table EMPLOYEES(as given in Experiment 1- Q.6)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**EXPERIMENT- 7**

**Title: Working with IMPLICIT CURSORS**

1. Using implicit cursor update the salary by an increase of 10% for all the records in EMPLOYEES table, and finally display how many records have been updated. If no records exist display the message None of the salaries were updated.
2. WAP to accept the employee\_id and display all the details of employees.If employee doesnot exist display the appreciate message

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**EXPERIMENT- 8**

**Title: Working with EXPLICIT CURSORS**

1. Using explicit cursor fetch the employee name, employee\_id and salary of all the records from EMPLOYEES table.
2. Using explicit cursor Insert the records from EMPLOYEES table for the columns employee\_id, Last\_Name and salary for those records whose salary exceeds 2500 into a new table TEMP\_EMP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**EXPERIMENT- 9**

**Title: Working with Triggers:**

**CUSTOMER Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | NAME | AGE | ADDRESS | SALARY |
| 1 | Ramesh | 32 | Ahmedabad | 2000.00 |
| 2 | Khilan | 25 | Delhi | 1500.00 |
| 3 | Kaushik | 23 | Kota | 2000.00 |
| 4 | Chaitali | 25 | Mumbai | 6500.00 |
| 5 | Hardik | 27 | Bhopal | 8500.00 |
| 6 | Komal | 22 | MP | 4500.00 |

1. The following program creates a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values:
2. First, [create a new table](http://www.mysqltutorial.org/mysql-create-table/) named employees\_audit to keep the changes of the employee table. Next, create a BEFORE UPDATE trigger that is invoked before a change is made to the employees table.
3. Scenario: We want to keep a log file containing data from rows that have been deleted from the EMPLOYEES Table.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**EXPERIMENT- 10 (Virtual Lab** [**http://vlabs.ac.in**](http://vlabs.ac.in/computer-science-and-engineering-labs.html) **)**

**Title: Working with Enhanced Entity relationship Model**

**Link to access :** <http://sage.virtual-labs.ac.in/home/pub/14/>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**EXPERIMENT- 11 (Virtual Lab** [**http://vlabs.ac.in**](http://vlabs.ac.in/computer-science-and-engineering-labs.html) **)**

**Title: Managing Records**

**Link to access :** <http://sage.virtual-labs.ac.in/home/pub/21/>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**EXPERIMENT- 12 (Virtual Lab** [**http://vlabs.ac.in**](http://vlabs.ac.in/computer-science-and-engineering-labs.html) **)**

**Title: Retrieving Data**

**Link to access :** <http://sage.virtual-labs.ac.in/home/pub/22/>

1. **SUGGESTED READINGS:**

**H1. TEXT BOOK:**

1. SQL, PL/SQL: The Programming Language Of Oracle by Ivan Bayross, BPB Publications.

**GUIDELINES**

***Cell Phones and other Electronic Communication Devices*:** Cell phones and other electronic communication devices (such as Blackberries/Laptops) are not permitted in classes during Tests or the Mid/Final Examination. Such devices MUST be turned off in the class room.

***E-Mail and online learning tool:*** Each student in the class should have an e-mail id and a pass word to access the LMS system regularly. Regularly, important information – Date of conducting class tests, guest lectures, via online learning tool. The best way to arrange meetings with us or ask specific questions is by email and prior appointment. All the assignments preferably should be uploaded on online learning tool. Various research papers/reference material will be mailed/uploaded on online learning platform time to time.

***Attendance:*** Students are required to have **minimum attendance of 75%** in each subject. Students with less than said percentage shall **NOT** be allowed to appear in the end semester examination.

**Course outcome assessment:** To assess the fulfilment of course outcomes two different approaches have been decided. Degree of fulfillment of course outcomes will be assessed in different ways through direct assessment and indirect assessment. In Direct Assessment, it is measured through quizzes, tests, assignment, Mid-term and/or End-term examinations. It is suggested that each examination is designed in such a way that it can address one or two outcomes (depending upon the course completion). Indirect assessment is done through the student survey which needs to be designed by the faculty (sample format is given below) and it shall be conducted towards the end of course completion. The evaluation of the achievement of the Course Outcomes shall be done by analyzing the inputs received through Direct and Indirect Assessments and then corrective actions suggested for further improvement. Capping

***Passing criterion:*** Student has to secure minimum 40% marks of the “highest marks in the class scored by a student in that subject (in that class/group class)” individually in both the ‘End-Semester examination’ and ‘Total Marks’ in order to pass in that paper.

* Passing Criterion for B. Tech: Minimum 40% of the highest marks in the class applicable to the students.

**Sample format for Indirect Assessment of Course outcomes**

|  |
| --- |
| NAME: |
| ENROLLMENT NO: |
| SAP ID: |
| COURSE: |
| PROGRAM: |

Please rate the following aspects of course outcomes of Automotive transmissions systems.

Use the scale 1-4\*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sl. No. |  | 1 | 2 | 3 | 4 |
| 1 | CO1. Able to work on Oracle Databases. |  |  |  |  |
| 2 | CO2. Understand the implementation of searching methods of the Database Management Systems. |  |  |  |  |
| 3 | CO3. Learn PL/SQL Programming |  |  |  |  |
| 4 | CO4. Learn the implementation & working of Cursors & Triggers. |  |  |  |  |

3

Below Average

Good

1

**\***

Very Good

Average

4

2