RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

DEPARTEMENT OF INFORMATION TECHNOLOGY

|  |  |
| --- | --- |
| |  | | --- | |  |   IT333 DATABASE LAB  SEMESTER 5 |

LAB CYCLE (2017- 21 BATCH)

INSTRUCTIONS TO STUDENTS

 Students should be regular and come prepared for the lab practice.

1. In case a student misses a class, it is his/her responsibility to complete that missed experiment(s) before he or she comes for the second lab after the missed class.
2. Students should bring the lab record with cycle stuck. Prescribed textbook and class notes can be kept ready for reference if required.
3. Once the experiment(s) get executed, they should show the results to the instructors and copy the same in their observation book and get it verified before the next lab session.

PROCEDURE FOR EVALUATION

The entire lab course consists of 100 marks.

The marking scheme is as follows

1. Practical records /Outputs 60 marks (Internally by the College)
2. Regular class Viva 10 marks (Internally by the College)
3. Final written test/quiz 30 marks

|  |  |
| --- | --- |
| Continuous Evaluation (i and ii) | 70 marks |
| End Semester Lab Exam | 30 marks |
| Total | 100 marks |

Scheme for continuous evaluation

Students will be evaluated weekly. Each evaluation carries 15 marks. The scheme is as follows

|  |  |
| --- | --- |
| Program and Execution(till date) | 7 marks |
| Record | 3 marks |
| Viva-voce | 5 marks |
| Total | 15 marks  *(15 marks\*9 weeks=135 marks) Which will be converted to 70 marks* |

Scheme for end semester lab exam

End semester lab exam will be conducted after the completion of all the weekly exercises. The student will not be allowed for exam if he/she is found short of attendance (<75%) and has not completed and signed all the experiments. The marking scheme for end semester lab exam is as follows

|  |  |
| --- | --- |
| Algorithm | 5 marks |
| Program execution | 10 marks |
| Results for all inputs | 5 marks |
| Viva-voce | 10 marks |
| Total | 30 marks |

**LAB CYCLE**

Week 1

**Exercise I: DDL Commands**

1. Create a table students with the following structure

|  |  |  |
| --- | --- | --- |
| Column name | Constraint | Type of data |
| Id | Not null | Integer value |
| Name |  | Text |
| Email |  | Text |
| Phoneno |  | Numeric value |
| DOB |  | Date |
| Address |  | Text |
| Height\_in\_cm |  | Float |
| Deptno | Not null | Numeric value |

1. Create a table department with the following structure

|  |  |  |
| --- | --- | --- |
| Column name | Constraint | Type of data |
| Deptno | Not null | Integer value |
| Dname | Unique | Text |
| Loc |  | Text |

1. Modify student table to add a column called date\_of\_joining with data type date.
2. Modify student table to add a column called stipend with not null constraint.
3. Add a check constraint to the student table to verify that the stipend is always greater than 0.
4. Modify student table to change id to a primary key constraint.
5. Modify student table by adding a referential integrity constraint (foreign key) on deptno to department table.
6. Drop the column date\_of\_joining, adress from student table.
7. Modify the column width of dname of department table to 50.
8. Drop the unique constraint on dname of department table.

Week 2

**Exercise II: DML commands**

1. Insert the following data into students table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Name** | **Email** | **Phoneno** | **DOB** | **Height\_in\_cm** | **stipend** | **Deptno** |
| 1 | Smith | Smith@abc.com | 9874747474 | 17-dec-80 | 160.2 | 10000 | 10 |
| 2 | Allen | Allen@abc.com | 9874747475 | 20-feb-81 | 161.2 | 10200 | 20 |
| 3 | Ward | Ward@abc.com | 9874747476 | 22-feb-81 | 162.2 | 10400 | 30 |
| 4 | Jones | [Jones@abc.com](mailto:Jones@abc.com) | 9874747477 | 02-apr-81 | 163.2 | 10600 | 20 |
| 5 | Martin | [Martin@abc.com](mailto:Martin@abc.com) | 9874747478 | 28-sep-81 | 164.2 | 10800 | 20 |
| 6 | Blake | [Blake@abc.com](mailto:Blake@abc.com) | 9874747479 | 01-may-81 | 165.2 | 11000 | 30 |
| 7 | Clark | [Clark@abc.com](mailto:Clark@abc.com) | 9874747480 | 09-jun-81 | 166.2 | 11200 | 10 |
| 8 | Scott | [Scott@abc.com](mailto:Scott@abc.com) | 9874747481 | 09-dec-82 | 167.2 | 11400 | 20 |
| 9 | King | [King@abc.com](mailto:King@abc.com) | 9874747482 | 17-nov-81 | 168.2 | 11600 | 10 |
| 10 | Turner | [Turner@abc.com](mailto:Turner@abc.com) | 9874747483 | 08-sep-81 | 163.2 | 11800 | 30 |
| 11 | Adam | [Adam@abc.com](mailto:Adam@abc.com) | 9874747484 | 12-jan-83 | 164.2 | 12000 | 20 |
| 12 | James | [James@abc.com](mailto:James@abc.com) | 9874747485 | 03-dec-81 | 165.2 | 12200 | 30 |
| 13 | Ford | [Ford@abc.com](mailto:Ford@abc.com) | 9874747486 | 03-dec-81 | 166.2 | 12400 | 20 |
| 14 | Miller | [Miller@abc.com](mailto:Miller@abc.com) | 9874747487 | 23-jan-82 | 173.2 | 12600 | 10 |

1. Insert the following data into department table

|  |  |  |
| --- | --- | --- |
| **Deptno** | **Dname** | **Loc** |
| 10 | IT | New York |
| 20 | CS | Dallas |
| 30 | ECE | Chicago |
| 40 | ME | Boston |

1. Cancel the mail id of all IT students.
2. Modify the mail id, phone no and DOB of Jones.
3. List empno and name of all students of IT department.
4. List details of students who belong to department 10 or 30.
5. List unique heights.
6. List students whose name starts with ‘Sm’.
7. List name, stipend and updated stipend which is calculated as 10% of stipend of students. Display updated stipend as a separate column.
8. List the name, mail id and phone number of all students born between February 20, 1981 and May 1, 1981. Order the query in ascending order of start date.
9. List the maximum, average stipend and number of students in department 30.
10. Write SQL query to find the total number of CS students.
11. Write SQL query to find which department has exactly one student having height <165.
12. Write SQL query to find which department has the highest number of students? Show the deptno and count.
13. Write SQL query to find the total number of students in each department.
14. List the lowest stipend for different department and list them in descending order.
15. Which department’s average stipend is the lowest among all? Show the deptno, average stipend.
16. List the minimum, maximum and average stipend for each department.
17. Retrieve all the students who are working in deptno=10 and who earn stipend at least as much as any students working in deptno=30.
18. List the entire departments who have no students.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Week 3  **Exercise IV: Simple PL/SQL Programs**   1. Write a pl/sql program to check whether a given number is Armstrong or not. 2. Write a pl/sql program to display the reverse of a string. 3. Write a pl/sql program to check whether a given number is palindrome or not. 4. Write a pl/sql code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns: radius & area. 5. Write a pl/sql block to accept the marks for three subjects from a student and classify according to the grade and insert into tables named gradea, gradeb and gradec.   (Hint: the student table contains rollno, name, m1, m2, m3 and tot\_marks. If the total marks >75% insert into gradea table, between 60 and 75% insert into gradeb table and <60% gradec table. Grade table has the following fields : rollno, name, %marks)  Week 4  **Exercise V: Cursors**   1. Write a program in PL/SQL to create an implicit cursor with for loop to display the details of students. 2. Write a program in PL/SQL to create an explicit cursor with for loop to display the details of students. 3. Write a program in PL/SQL to create a cursor displays the name and stipend of each student in the students table whose stipend is less than that specified by a passed-in parameter value. 4. A bank has an ‘ACCMASTER’ table where it holds the current status of a client’s bank account. Another table called ‘ACCTRAN’ holds each transaction as it occurs at bank (ie, Deposits/withdrawals of clients). The ‘ACCTRAN’ table must hold a flag indicating whether the transaction was credit or debit. Write SQL procedure to update the ‘ACCMASTER’ table and sets the balance depending upon whether the account is debited or credited. The updation should be done only for those records that are not processed.  |  |  | | --- | --- | | **ACCMASTER** | **ACCTRAN** | | Accno(PK) | Accno(FK) | | Name | TransactionDate | | Balance | Deb\_cred Debit/Credit | |  | Amount | |  | Processed Yes/No flag |  1. An HRD manager has decided to raise the salary of all employees in department number by 0.005. Whenever any such rise is given to the employees, a record for the same is maintained in the Emp\_Raise table. Write a PL/SQL block to update the salary of each employee and insert a record in the Emp\_Raise table.   **Employee Table**   |  |  |  | | --- | --- | --- | | **Field** | **Data type** | **Constraints** | | Name | Varchar2(6) |  | | EmpNo | Number(5) | Primary key | | Salary | Number(5) |  | | Deptno | Number(5) |  |   **Emp\_Raise Table**   |  |  | | --- | --- | | **Field** | **Data type** | | Emp\_no | Varchar2(6) | | Raise\_date | Date | | Raise\_amt | Number |   Week 5  **Exercise VI: Exception Handling, Procedures**   1. Write SQL procedure for a banking application which accepts the account number and amount withdraw. If the balance after withdrawal is less than 1000, give an error message using user defined exception. Else update the table with new balance. 2. Write a PL/SQL procedure that takes three numbers as parameters and displays the LCM and GCD of the three numbers. 3. Create table Student(RegNo, name, marks). 4. Write a procedure to display average mark of the students from the Student table. 5. Write a procedure to display mark of a student with the given student register number.   Week 6  **Exercise VII: Functions** |
| 1. Sales of two different products in 1 week is recorded   Product (product\_id, product\_name, grade)  Sales (product\_id, sales, sales\_date, sales\_day)  Do the following:  1) Write a function that retrieves the product name and grade of a product with the given product\_id.  2) Whenever the sum of the sales of a particular product becomes greater than a target value it is given an A grade. If there is no sale for a product, an exception is to be raised and no more sales for that product should be allowed. Sales are not allowed on Sundays.  2. Consider a database that contains the following tables:  Candidate (cand-id, cand-name, party)  Voters (voter-id, voter-name)  Voted (cand-id, party, voter-id)   1. Write a program to ensure that only valid voters are permitted to vote. 2. Write a function which calculates the total number of votes of a particular candidate. 3. Write a procedure which count the votes and declare the winner.   Week 7  **Exercise VIII: Trigger**  Question 1  Create a trigger that can be used to keep track of all the transactions performed on the employee table. If any employee is deleted, a new row containing the details of this employee is stored in a table called xemployee. Similarly, if a new employee is inserted, a new row is created in another table called nemployee, and so on. If employee table is updated, then updated rows have to be inserted into another table called cemployee.  Question 2  Create a trigger that limits the DML actions to the employee table to weekdays from 8.30am to 6.30pm. If a user tries to insert/update/delete a row in the EMPLOYEE table, a warning message will be prompted.  Question 3  Create the following tables: BookRec (BookName, Author, TotalCopies, AvailableCopiesMembershipRec(MemberID, Name)BookDetails(BookID, BookName, MemberID)CirculationRec(BookID, MemberID, IssueDate, ReturnDate)  1. Write a PL/SQL program to add a new book into the library   Hint: Read Book Name, Author and total copies from the keyboard  b) Autogenerate the BookID field in the BookDetails table  c) The entry into the BookDetails table should be done using Trigger  Question 4  Create the following tables: BookRec (BookName, Author, TotalCopies, AvailableCopiesMembershipRec(MemberID, Name)BookDetails(BookID, BookName, MemberID)CirculationRec(BookID, MemberID, IssueDate, ReturnDate) Write a function to issue a book to a member. If book is available the function should return the  returnDate of the book otherwise display a message  Hint:1. Read Book name and MemberID from the keyboard  2. If sufficient copies of the book are available   1. Make necessary updations to BookRec and BookDetails table 2. Insert a new row into the CirculationRec table 3. Return date should be 30 days after the issue date   Question 5    Create the above tables with foreign key constraint and insert values.    Write a trigger to update value of  **no-of-courses** when a new row is inserted to ENROLLMENT table  Week 8  **Exercise IX: Develop an online train ticket reservation system**  PROCEDURE   1. A home page in which user can register and the details are entered to the database 2. Registered user can login to the homepage by unique username and password 3. User can book tickets, cancel tickets and enquire PNR status 4. User can view train schedule, booked ticket history and details of cancelled tickets 5. User can also track train status 6. User can update account details, add/delete favorite journey list |
|  |
|  |