|  |  |  |  |
| --- | --- | --- | --- |
| **Practical-IV** | | **BCAP 185**  **DBMS Lab** | **48 Hrs** |
| **Practical/Week: 4 Hrs**  **Credits: 2** | | **Exercises on DBMS Problems** | **I.A.: 20**  **Exam: 80** |
| **NOTE:**   * Display all the records and describe the structure for every table in each exercise. * For any query, the result must contain at least one record. | | | |
| **PART A** | | | |
| 1 | Create a table *EMPLOYEE* using SQL command to store details of employees such as *EMPNO, NAME, DESIGNATION, DEPARTMENT, GENDER* and *SALARY*.  Specify Primary Key and NOT NULL constraints on the table Allow only ‘*M*’ or ‘*F*’ for the column *GENDER*. DEPARTMENT can be SALES, ACCOUNTS, IT  Choose DESIGNATION as CLERK, ANALYST, MANAGER, ACCOUNTANT and  SUPERVISOR that depends on department.  **Write the following SQL queries:**   1. Display EMPNO, NAME and DESIGNATION of all employees whose name ends with RAJ. 2. Display the details of all female employees who is earning salary within the range 20000 to 40000 in SALES or IT departments 3. List the different DEPARTMENTs with the DESIGNATIONs in that department 4. Display the department name, total, average, maximum, minimum salary of the DEPARTMENT only if the total salary given in that department is more than 30000. 5. List the departments which have more than two employees.   **Marks distribution:**  Creating the table with constraints: 4, Inserting records: 2, a)3 b)3 c)2 d)2 e)2 | | |
| 2 | Create a table *CLIENT* to store CLIENT\_NO, NAME, ADDRESS, STATE, BAL\_DUE. Client no must start with ‘C’. Apply the suitable structure for the columns. Specify Primary Key and NOT NULL constraints on the table  Insert 10 records.  **Write the following SQL queries:**   1. From the table CLIENT, create a new table CLIENT1 that contains only CLIENT\_NO and NAME, BAL\_DUE from specified STATE. Accept the state during run time. 2. Create a new table CLIENT2 that has the same structure as CLIENT but with no records. Display the structure and records. 3. Add a new column by name PENALTY number (10, 2) to table CLIENT. 4. Assign Penalty as 10% of BAL\_DUE for the clients C1002, C1005, C1009 and for others 8%. Display records. 5. Change the name of CLIENT1 as NEW\_CLIENT. 6. Delete the table CLIENT2.   **Marks distribution:**  Creating the table with constraints: 4 Inserting records: 2 a)2 b)3 c)2 d)3 e)1 f) 1 | | |

|  |  |
| --- | --- |
| 3 | Create a table BOOK using SQL command to store Accession No, TITLE, AUTHOR, PUBLISHER, YEAR, PRICE. Apply the suitable structure for the columns. Specify Primary Key and NOT NULL constraints on the table. Insert 10 records.  **Write the following SQL queries:**   1. List the details of publishers having ‘a’ as the second character in their names. 2. Display Accession No., TITLE, PUBLISHER and YEAR of the books published by the specified author before 2010 in the descending order of YEAR. Accept author during run time. 3. Modify the size of TITLE to increase the size by 5 characters. 4. Display the details of all books other than Microsoft press publishers. 5. Remove the records of the books published before 1990.   **Marks distribution:**  Creating the table with constraints: 4 Inserting records: 3 a)2 b)3 c)2 d)2 e)2 |
| 4 | Create a table SALES with columns SNO, SNAME, MANAGER\_NAME, JOIN\_DATE, DATE\_BIRTH, SALARY, SALES\_AMOUNT and COMMISSION. Minimum Age for joining the company must be 18 Yrs. Default value for Commission should be 0. Apply the suitable structure for the columns. Specify Primary Key and NOT NULL constraints on the table. Insert 10 records with data except commission. Manager of Manager can be Null.  **Write the following SQL queries:**   1. Display the details of Sales Persons whose salary is more than Average salary in the company. 2. Update commission as 20% of Sales Amount. 3. Display SNO, SNAME, MANAGER\_NAME, SALARY, COMMISSION, MANAGER\_SALARY of the sales persons getting sum of salary and commission more than salary of manager .(Self join) 4. Display the records of employees who finished the service of 10 years.   **Marks distribution:**  Creating the table with constraints: 5 Inserting records: 2 a)3 b)2 c)3 d)3 |
| **PART B** | |
| 1 | Create the following tables by identifying primary and foreign keys. Specify the not null property for mandatory keys.  *SUPPLIERS (SUPPLIER\_NO, SNAME, SADDRESS, SCITY*) *COMPUTER\_ITEMS (ITEM\_NO, SUPPLIER\_NO, ITEM\_NAME, IQUANTITY)*  *Consider three suppliers. A supplier can supply more than one type of items.*  **Write the SQL queries for the following:**   1. List *ITEM* and *SUPPLIER* details in alphabetical order of city name and in each city decreasing order of *IQUANTITY.* 2. List the name and address, city of the suppliers who are supplying keyboard. 3. List the supplier name, items supplied by the suppliers ‘Cats’ and ‘Electrotech’. 4. Find the items having quantity less than five and insert the details of supplier and items of these into another table NEWORDER.   **Marks distribution:**  Creating the tables with constraints: 5 Inserting records: 5 a)3 b)3 c)3 d)3 |
| 2 | Create the following tables by identifying primary and foreign keys. Specify the not null property for mandatory keys.  *EMPLOYEE\_MASTER (EMP\_ID, EMP\_NAME, EMP\_ADDRS, PHONE)*  A*TTENDANCE (EMP\_ID, MONTH, WOM, MHRS, THRS, WHRS, TRHRS, FHRS, SHRS,*  *SUHRS).* (Valid values for WOM<=5, MONTH can be 1-12). Apply appropriate constraints. Consider 3 employees. And attendance records for at least two months. |

|  |  |
| --- | --- |
|  | **Write the SQL queries for the following:**   1. Display EMP\_ID, *EMP\_NAME* and *EMAIL\_ID* of all employees who are working on every Sunday of 2nd or 4th week in a month. 2. Display total hours worked by each employee in each month with EMP\_ID, 3. Display the names of the employees who never attended the duty so far (attendances not given so far). 4. Display the employee name, month, week, total hours worked for employees who have total number of hours more than 20 hours a week.   **Marks distribution:**  Creating the tables with constraints: 5 Inserting records: 5 a)3 b)3 c)3 d)3 |
| 3 | Create the following tables by identifying primary and foreign keys, specify the not null property for mandatory keys.  **Write the SQL queries for the following:**   1. Display total amount spent by C2. 2. Display the names of product for which either QtyAvailable is less than 30 or total QtySold is less than 5 (Use UNION). 3. Display the name of products and quantity purchased by C4. 4. How much Profit does the shopkeeper get on C1’s purchase? 5. How many ‘Pen Drives’ have been sold?   **Marks distribution:**  Creating the tables with constraints: 4 Inserting records: 4 a)3 b)3 c)3 d)3 e) 2 |
| 4 | Create table STUDENT\_PROFILE that includes Rollno, name, class, ECCC (Extra/Co- curricular he belongs to such as SPORTS, NSS, etc.) and another table MARKS\_REPORT that includes Rollno, Internal\_Test, Marks1, Marks2, Marks3 and ECCC\_marks.  **Constraints**   * Internal\_Test can be either 1 or 2. * Each mark can be 0 – 100. Absence in the test can be entered as -1. * Consider at least 3 classes.   Apply suitable datatype and constraints to each column. Insert 5 students marks report in the both the tests.  **Write the SQL queries for the following:**  a) Find number of students failed class-wise. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PRODUCT\_DETAIL** | | | | |
| **P\_No** | **P\_Name** | **QtyAvailable** | **Price** | **Profit (%)** |
| P0001 | Monitor | 10 | 3000 | 20 |
| P0002 | Pen Drives | 50 | 650 | 5 |
| P0003 | CD Drive | 100 | 10 | 3 |
| P0004 | Keyboard | 25 | 600 | 10 |

|  |  |  |
| --- | --- | --- |
| **PURCHASED\_DETAIL** | | |
| **CustNo** | **P\_No** | **QtySold** |
| C1 | P0003 | 2 |
| C2 | P0002 | 4 |
| C3 | P0002 | 10 |
| C4 | P0001 | 3 |
| C1 | P0004 | 2 |
| C2 | P0003 | 2 |
| C4 | P0004 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
|  | 1. Display the complete details of the students secured distinction (Percentage>=70) in I BCA 2. Display class and highest total marks in second internals in each class. 3. Display the student name with rollno and class of those who passed in I internals and failed in II internals (use SET operator).   **Marks distribution:**  Creating the tables with constraints: 5 Inserting records: 5 a)3 b)3 c)3 d)3 |
| **PART C** | |
| 1 | Write a PL/SQL program to compute the selling price of books depending on the book code and category. Use Open, Fetch and Close. The Book\_detail table contains columns Book Code, Author, Title, Category and Price. Insert 10 records. The selling price = Price – Discount.  The discount is calculated as follows:  Print the result in a tabular form with proper alignment.  Book\_code Category Title Author Price Discount % Disc.Amt Selling\_Price  ===== == ==== ===== ==== ======== ======= =========  **Marks distribution:**  Creating the tables with constraints and inserting records: 5  PL/SQL code: 10 Compiling and debugging: 5 OUTPUT: 5 |

|  |  |  |
| --- | --- | --- |
| **Book Code** | **Category** | **Discount Percentage** |
| A | Novels | 10% of Price |
| Technology | 12.5% of Price |
| B | Commerce | 18% of Price |
| Science | 19% of Price |
| C | Songs | 25% of Price |
| Sports | 24% of Price |
| D | Others | 28% of Price |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |
|  |  |  |
|  |  |
|  |  |  |
|  |  |
|  |  |  |

|  |  |
| --- | --- |
| 2 | Write a PL/SQL program to display employee pay bill (using Cursor For loop). Use a **procedure** to receive basic pay and to compute DA, HRA, Tax, PF, Gross Pay and Net pay (Use OUT). Base table contains the columns empnum, empname, basic pay. Insert 3 records.  Allowances are computed as follows:  Basic Pay DA HRA  <=20000 35% of Basic 8% of Basic  >20000 and <=30000 38% of Basic 9% of Basic  >30000 and <=40000 40% of Basic 10% of Basic  >40000 45% of Basic 10% of Basic  Gross = Basic + DA + HRA; PF = 12% of Gross or Rs. 2000, whichever is minimum. PT = Rs. 100 upto Gross is 25,000; else Rs. 200, Net = Gross – (PF +PT)  Print Pay slip as follows:  **Marks distribution:**  Creating the tables with constraints and inserting records: 5  PL/SQL code: 5, Procedure code: 6, Compiling and debugging: 4, OUTPUT: 5 |
| 3 | Given the following tables: ITEM\_MASTER (itemno, name, stock, unit\_price) [Apply the Primary key and check constraint for stock and price as >0) [Insert 5 records]  ITEM\_TRANS (itemno, quantity and trans\_date)  Create a **package** PCK\_ITEM includes a function CHK\_ITEM and a procedure PROC\_ITEM. **Function** CHK\_ITEM gets one argument itemno and is used to check whether the parameter itemno exists in ITEM\_MASTER and should return 1 if exist. Otherwise 0 and display proper message.  **Procedure** PROC\_ITEM gets two arguments itemno and quantity, and is used to perform the following if item exists. If required quantity is not available, give appropriate message. If available, insert a record of this transaction to ITEM\_TRANS and modify the stock in ITEM\_MASTER.  Write a **PL/SQL** program to accept ITEM\_NO and Quantity needed of required item. Use  **Package** to do the transaction process (Transaction date can be current date). OUTPUT to be shown as follows:  **Marks distribution:**  Creating the tables with constraints and inserting records: 5  PL/SQL code: 3, Package specification: 2 Procedure: 4, Function: 3  Compiling and debugging: 3, OUTPUT: 5 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| 4 | Consider the following tables:  LIBRARY (Accession no, Title, Author, Publication, Status). Status can be A for available and I for Issued. Insert 3 records with status ‘A’ for all initially.  ISSUE (Rollno, Accession no, Borrowdate, returndate). OUTDATED (Accession no, Title, Author, Publication, tdate). Write the following Trigger programs.   1. Whenever the book is to be issued, insert a new record to ISSUE without having return date. When the record is **inserted** to ISSUE table, trigger TRIG\_ISSUE to be executed to update status in LIBRARY as ‘*I’.* 2. Whenever book is returned, update return date of that record as todays date in ISSUE table. When the record is **updated** to ISSUE table, trigger TRIG\_ISSUE to be executed to update status in LIBRARY as ‘*A’.* 3. Whenever the book is **deleted** by accepting Accession no. for status ‘A’ (at SQL >), trigger TRIG\_OUTDATE has to be executed to insert a record to OUTDATED.   Write a PL/SQL program to accept Rollno, Accession no. and transaction (B for Borrow and R for Return). Check for the existence of a given Accession no. and proceed as follows.   * If does not exist , display the message ‘Given accession no. is not available’ * If exist and transaction is B, check the status as ‘A’, then insert to ISSUE, and display the message with accno, author, title, publication and roll no to whom it is issued . * If exist and transaction is R, then update return date as current system date in ISSUE by accepting Rollno and Aceession no (for the record having return date empty.)   If searched record is not available, raise the predefined exception.  **Marks distribution:**  Creating the tables with constraints and inserting records: 5 PL/SQL code: 5, Trig\_ISSUE: 4, Trig\_OUTDATE: 3  Compiling and debugging: 3, OUTPUT: 5 |

Scheme of Examination

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Details** | **Marks** |
| 1 | PART A | 18 |
| 2 | PART B | 22 |
| 3 | PART C | 25 |
| 4 | Class Records | 10 |
| 5 | Viva – Voce | 5 |
| **Total Marks** | | **80** |