# C. U. SHAH UNIVERSITY
## Wadhwan City

**FACULTY OF**:- Computer Science
**DEPARTMENT OF**: - Master of Computer Applications
**SEMESTER**: -I
**CODE**: - 5CS03MPF1
**NAME** – PROGRAMMING TECHNIQUE-VI (DBMS)

### Teaching and Evaluation Scheme

| Subject Code | Name of the Subject | Teaching Scheme (Hours) | | | | Credits | Evaluation Scheme | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | Theory | | | | Practical (Marks) | | | Total |
| | | Th | Tu | Pr | Total | | Sessional Exam | | University Exam | | Internal | | University | |
| | | | | | | | Marks | Hrs | Marks | Hrs | Pr/Viva | TW | Pr | |
| 5CS01MPF1 | PROGRAMMING TECHNIQUE-VI (DBMS) | - | - | 4 | 2 | - | - | - | - | - | 20 | - | 80 | **100** |

**PRACTICAL LIST:**

1 ➢ Create the following tables:

1. Create LOCATION Table with columns Location_Id, Regional_Group.

   Constraints on LOCATION table: Location_Id Primary Key.

2. Insert the following records into the table LOCATION:

| LOCATION_ID | REGIONAL_GROUP |
|---|---|
| 122 | NEW YORK |
| 123 | DALLAS |
| 124 | CHICAGO |
| 167 | BOSTON |

3. Create DEPARTMENT Table with columns Department_Id, Name, Location_ID.

   Constraints on DEPARTMENT table: Department_Id Primary Key, Location_Id references LOCATION table.

4. Insert the following records into DEPARTMENT table:

| DEPRATMEMT_ID | NAME | LOCATION_ID |
|---|---|---|
| 10 | ACCOUNTING | 122 |
| 20 | RESEARCH | 124 |

| | | |
|---|---|---|
| 30 | SALES | 123 |
| 40 | OPERATIONS | 167 |

5. Create JOB Table with columns Job_Id, Funcation.
   Constraints on JOB table: Job_ID Primary Key.

6. Insert the following records into JOB table:

| JOB_ID | FUNCTION |
|---|---|
| 667 | CLERK |
| 668 | STAFF |
| 669 | ANALYST |
| 670 | SALESPERSON |
| 671 | MANAGER |
| 672 | PRESIDENT |

7. Create EMPLOYEE Table with columns Employee_Id, Last_Name, First_Name, Middle_Name, Job_Id, Manager_Id, Hire_Date, Salary, Comm, Department_ID.
   Constraints on EMPLOYEE table: Employee_Id Primary Key, Last_Name NotNull, Department_Id references DEPARTMENT table.

8. Insert the following records into EMPLOYEE table:

| EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE | SALARY | COMM | DEPARTMENT_ID |
|---|---|---|---|---|---|---|---|---|---|
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7369 | SMITH | JOHN | Q | 667 | 7902 | 17-DEC-84 | 800 | NULL | 20 |
| 7499 | ALLEN | KEVIN | J | 670 | 7698 | 20-FEB-85 | 1600 | 300 | 30 |
| 7505 | DOYLE | JEAN | K | 671 | 7839 | 04-APR-85 | 2850 | NULL | 30 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 7506 | DENNIS | LYNN | S | 671 | 7839 | 15-MAY-85 | 2750 | NULL | 30 |
| 7507 | BAKER | LESLIE | D | 671 | 7839 | 10-JUN-85 | 2200 | NULL | 40 |
| 7521 | WARK | CYNTHIA | D | 670 | 7698 | 22-FEB-85 | 1250 | 500 | 30 |

2

➢ Perform the following queries on the tables given in Set no. 1:

1. List all job details.
2. List all the locations.
3. List out first name,last_name,salary, commission for all employees.
4. List out employee_id,last_name,department_id for all employees and rename employee_id as "ID of the employee", last_name as "Name of the employee", department_id as "department ID".
5. List out the employee's annual salary with their names only.
6. List out the employees who are working in department 20.
7. List out the employees who are earning salary between 3000 and 4500.
8. List out the employees who are working in department 10 or 20.
9. List out the employees whose name starts with "S".
10. List out the employees whose name length is 4 and start with "S"

3

➢ Perform the following queries on the tables given in Set no. 1:

1. List out the employee id, last name in ascending order based on the employee id.
2. List out the employee id, name in descending order based on salary column.
3. List out the employee details according to their last_name in ascending order and salaries in descending order.
4. List out the employee details according to their last_name in ascending order and then on department_id in descending order.
5. How many employees who are working in different departments wise in the organization
6. List out the department wise maximum salary, minimum salary, average salary of the employees
7. List out the no. of employees for each month and year, in the ascending order based on the year, month.
8. List out the department id having at least four employees.
9. How many employees in January month.

| | |
|---|---|
| | 10. Which is the department id, having greater than or equal to 3 employees joined in April 1985. |
| 4 | ➢ Perform the following queries on the tables given in Set no. 1:<br>1. Display the employee who got the maximum salary.<br>2. Display the employees who are working in Sales department.<br>3. Display the employees who are working as "Clerk".<br>4. Find out no. of employees working in "Sales" department.<br>5. List our employees with their department names.<br>6. Display employees with their designations (jobs).<br>7. How many employees who are working in different departments and display with department name.<br>8. How many jobs in the organization with designations.<br>9. Display employee details with all departments.<br>10. List out the common jobs in Research and Accounting Departments in ascending order. |
| 5 | ➢ Create the following tables:<br>1. Create STUDENT Table with fields rollno, name, class, birthdate<br>Constraints on STUDENT table: rollno primary key and rollno must start with latter 'R'.<br><br>2. Insert the following records into Student Table:<br><br>ROLLNO NAME CLASS BIRTHDATE<br>---------- ---------- -------- ----------<br>R1 Pritesh Patel A 23-FEB-89<br>R2 Sugeet Patel A 05-SEP-85<br>R3 Dipesh Patel B 24-MAR-76<br>R4 Chandresh patel B 17-APR-87<br>R5 Bhavin Jilvaani A 25-DEC-75<br><br>3. Create COURSE Table with fields courseno, coursename, max_marks, pass_marks<br><br>Constraints on COURSE table: courseno primary key, check for max_mark>0, also check for pass_mark>0 and pass_marks<max_marks.<br>4. Insert the following records into Course Table: |

# C. U. SHAH UNIVERSITY
## Wadhwan City

| COURSENO | COURSENAME | MAX_MARKS | PASS_MARKS |
|---|---|---|---|
| 610001 | FOP | 90 | 40 |
| 610002 | FOP Prac | 90 | 40 |
| 610003 | MATHS | 90 | 40 |
| 610004 | COMP ORG | 90 | 40 |
| 610005 | DBMSI | 90 | 40 |
| 610006 | SQL & PL/SQL | 90 | 40 |
| 610007 | ERFM | 90 | 40 |

1. Create SC Table with fields rollno, courseno, marks:

   Constraints on Sc table: marks must be greater than 0, rollno, courseno primary key, rollno references students and couseno references course.

2. Insert the following records into SC Table:

| ROLLNO | COURSENO | MARKS |
|---|---|---|
| R3 | 610005 | 70 |
| R3 | 610001 | 70 |
| R3 | 610002 | 68 |
| R3 | 610003 | 58 |
| R3 | 610004 | 74 |
| R3 | 610006 | 59 |
| R3 | 610007 | 55 |
| R1 | 610001 | 80 |
| R1 | 610002 | 89 |
| R1 | 610003 | 78 |
| R1 | 610004 | 88 |
| R1 | 610005 | 76 |
| R1 | 610006 | 85 |
| R1 | 610007 | 90 |
| R2 | 610001 | 90 |

| | |
|---|---|
| | R2 610002 85<br>R2 610003 78<br>R2 610004 75<br>R2 610005 68<br>R2 610006 59<br>R2 610007 74<br>R4 610001 75<br>R4 610002 45<br>R4 610003 58<br>R4 610004 68<br>R4 610005 78<br>R4 610006 62<br>R4 610007 63<br>R5 610001 70<br>R5 610002 78<br>R5 610003 52<br>R5 610004 79<br>R5 610005 85<br>R5 610006 76<br>R5 610007 80 |
| 6 | ➢ Perform the following queries on the tables given in Set no. 5:<br>1. Add constraint that marks entered are between 0 to 100 only.<br>2. While creating COURSE table, primary key constraint was forgotten. Add the primary key now.<br>3. Display details of student where course is 'Data Base Management System'<br>4. Select student names who have scored more than 70% in Computer Networks and have not failed in any subject.<br>5. Select names and class of students whose names begin with 'A' or 'B'.<br>6. Display average marks obtained by each student.<br>7. Select all courses where passing marks are more than 30% of average maximum marks.<br>8. Select the course where Second and third characters are 'AT'<br>9. Display details of students born in 1975 or 1976. |

| | |
|---|---|
| | 10. Find out the cousewise average marks from SC table. |
| 7 | ➢ Perform the following miscellaneous queries:<br>1. Add 15 days to current date.<br>2. Add and subtract 5 months from current month.<br>3. Calculate months between current months and '3-7-2008'<br>4. Find last day of current month.<br>5. How many days left in a current month?<br>6. Find ASCII value of letter 'R'.<br>7. Find name of all constraint based on particular table.<br>8. Find difference between current date and specified date.<br>9. Find username and userid from current login.<br>10. Find the occurrence of 'or' in the string. |
| 8 | ➢ Create the following tables:<br>1. Create the table SCREEN with the fields (screen_id, location, seating_cap)<br><br>Constraints on SCREEN table: screen_id primary key, location not null, seating_cap not null, Screen_Id must start with S, location values must be either FF,SF or TF, seating_cap must be greater then 0.<br><br>2. Insert the following records into SC Table:<br><br>SCREEN_ID LOCATION SEATING_CAP<br>---------- ---------- ----------<br>S1 SF 400<br>S2 TF 350<br>S3 TF 250<br>S4 SF 300<br>S5 TF 170<br><br>3. Create the table MOVIE with the fields (movie_id, movie_name, date_of_release)<br>Constraints on MOVIE table: movie_id primary key, movie_name unique, date_of_release not null. |

4. Insert the following records into the table MOVIE:

| MOVIE_ID | MOVIE_NAME | DATE_OF_RELEASE |
|---|---|---|
| M01 | Star Wars III | 11-SEP-09 |
| M02 | Oceans 13 | 10-JUL-09 |
| M03 | Armageddon | 18-FEB-05 |
| M04 | Step up | 27-SEP-02 |
| M05 | Terminator-3 | 25-OCT-05 |

5. Create the table CURRENT1 with the fields (screen_id, movie_id, date_of_arrival, date_of_closure)

Constraints on CURRENT1 table: screen_id references SCREEN table, movie_id references MOVIE, date_of_arrival not null, date_of_closure not null, check for date_of_arrival<date_of_closure.

6. Insert the following records into the table CURRENT1:

| SCREEN_ID | MOVIE_ID | DATE_OF_ARRIVAL | DATE_OF_CLOSURE |
|---|---|---|---|
| S1 | M01 | 13-JUL-09 | 26-AUG-09 |
| S2 | M03 | 25-APR-04 | 03-MAY-04 |
| S3 | M02 | 05-JAN-09 | 25-FEB-09 |
| S4 | M04 | 16-MAR-09 | 20-APR-09 |
| S5 | M05 | 03-MAY-05 | 09-JUL-05 |

9

- Perform the following queries on the tables given in Set no. 8:
  1. Get the name of movie which has run the longest in the multiplex so far.
  2. Get the average duration of a movie on screen number 'S4'.
  3. Get the details of movie that closed on date 24-november-2004.
  4. Movie 'star wars III' was released in the 7th week of 2005. Find out the date of its release considering that a movie releases only on Friday.
  5. Get the full outer join of the relations screen and current.

1

- Create the following tables:

# C. U. SHAH UNIVERSITY
## Wadhwan City

0

1. Create the table DISTRIBUTOR with the fields (DNO, DNAME, DADDRESS, DPHONE)

   Constraints on table DISTRIBUTOR: dno primary key, dname not null.

2. Insert the following records into the table DISTRIBUTOR

| DNO | DNAME | DADDR | DPHONE |
|---|---|---|---|
| D01 | Hardik Ode | 9315462 | |
| D02 | Dhaval | Anand | 9325135 |
| D03 | AAAAOH | Baroda | 9563154 |
| D04 | Mr. Talkative | Vasad | 9321354 |
| D05 | Dipen | Thasara | 9345432 |

3. Create the table ITEM1 with the fields (ITEMNO, ITEMNAME, COLOR, WEIGHT)

   Constraints on table ITEM1: itemno primary key, itemname not null, check for weight>0

4. Insert the following records into the table ITEM1:

| ITEMNO | ITEMNAME | COLOUR | WEIGHT |
|---|---|---|---|
| I01 | Screw | Black | 20 |
| I02 | Bolt | white | 100 |
| I03 | Nut | red | 50 |
| I04 | Hammer | green | 75 |
| I05 | Washer | red | 110 |
| I06 | Wire | Gray | 37 |
| I07 | Nail | Green | 46 |

5. Create the table DIST_ITEM with the fields (DNO, ITEMNO, QTY):
   Constraints of table DIST_ITEM: dno references DISTRIBUTOR table, itemno references ITEM table

6. Insert the records into the table DIST_ITEM:

# C. U. SHAH UNIVERSITY
## Wadhwan City

| | |
|---|---|
| | DNO ITEMNO QTY<br>D01 I02 130<br>D02 I01 500<br>D03 I05 420<br>D04 I03 320<br>D05 I06 160<br>D02 I04 190<br>D01 I07 462<br>D05 I01 256<br>D03 I04 315 |
| 11 | ➢ Perform the following queries on the tables given in Set no. 10:<br>1. Add column CONTACT_PERSON to the distributor table with the not null constraint.<br>2. Create a view LONDON_DIST on DIST_ITEM which contains only those records where distributors are from London. Make sure that this condition is checked for every DML against this view.<br>3. Display detail of all those item that have never been supplied. Select * from item1 where itemno not in(select itemno from dist_item) no rows selected.<br>4. Delete all those items that have been sulpplied only once.<br>5. List the names of distributors who have an 'A' and also a 'B' somewhere in their names. |
| 12 | ➢ Perform the following queries on the tables given in Set no. 10:<br>1. Count the number of items having the same color but not having weight between 20 and 100<br>2. Display all those distributors who have supplied more than 1000 parts of the same type.<br>3. Display the average weight of items of same colour provided at least one items have that colour.<br>4. Display the position where a distributor name has an 'OH' in its spelling somewhere after the forth character.<br>5. Count the number of distributors who have a phone connection and are supplying item number 'I100'. |
| 13 | ➢ Perform the following queries on the tables given in Set no. 10:<br>1. Create a view on the table in such a way that the view contains the distributor name, item name and the quantity supplied.<br>2. List the name, address and phone number of distributors who have the same three digits in their number as 'Mr. Talkative'.<br>3. List all distributor names who supply either item I01 or I07 and the quantity supplied is more than 100.<br>4. Display the data of the top three heaviest ITEMS. |

| | |
|---|---|
| | 5. Count the total quantity group by itemno. |
| 1 | ➢ Create the following tables: |
| 4 | 1. Create the table WORKER with the fields (worker_id, name, wage_per_hour, specialized_in, manager_id)<br>Constraints on table WORKER: worker_id primary key, name not null, manager_id primary key, check for wage_per_hour>=0. |

2. Insert the following records into the table WORKER:

| WOR | NAME | WAGE_PER_HOUR | SPECIALISED_IN | MAN |
|---|---|---|---|---|
| W01 | Mr.Cacophonix | 50 | Polishing | M01 |
| W02 | Dhaval | 40 | Polishing | M02 |
| W03 | Dipen | 35 | Fitting | M03 |
| W04 | Hardik | 30 | Marketing | M04 |
| W05 | Jigar | 55 | Fitting | M05 |

3. Create the table JOB with the fields (job_id, type_of_job, status):

4. Insert the following records into the table JOB:

| JOB | TYPE_OF_JOB | S |
|---|---|---|
| J01 | Packing | A |
| J02 | Editing | A |
| J03 | Moulding | B |
| J04 | Accounting | B |
| J05 | Printing | B |

5. Create the table JOB_ASSIGNED with the fields (worker_id, job_id, starting_date, number_of_days)

Constraints on table JOB_ASSIGNED: worker_id references WORKER table, job_id references JOB table.

6. Insert the following records into the table JOB_ASSIGNED:

| WOR | JOB | STARTING_ | NUMBER_OF_DAYS |
|---|---|---|---|
| W01 | J01 | 15-SEP-09 | 35 |
| W02 | J01 | 20-SEP-08 | 34 |
| W03 | J04 | 12-OCT-09 | 39 |
| W01 | J05 | 19-OCT-09 | 10 |

| | |
|---|---|
| | W02 J04 12-SEP-08 25 |
| 15 | ➢ Perform the following queries on the tables given in Set no. 14:<br>1. Display the date on which each worker is going to end his presently assigned job.<br>2. Display how many days remain for each worker to finish his job.<br>3. Display the STARTING_DATE in the following format - 'The fifth day of month of October, 2004'.<br>4. Change the status to 'Complete' for all those jobs, which started in year 2008.<br>5. Display job details of all those jobs where at least 25 workers are working.<br>6. Display all those jobs that are already incompleted. |
| 16 | ➢ Perform the following queries on the tables given in Set no. 14:<br>1. Find all the jobs, which begin within the next two weeks.<br>2. List all workers who have their wage per hour ten times greater than the wage of their managers.<br>3. List the names of workers who have been assigned the job of Packing.<br>4. What is total number of days allocated for printing on the goods for all the workers together.<br>5. Which workers receive higher than average wage per hour. |
| 17 | ➢ Perform the following queries on the tables given in Set no. 14:<br>1. Display details of workers who are working on more than one job.<br>2. Which workers having specialization in polishing start their job in September?<br>3. Display details of workers who are specialized in the same field as that of Mr.Cacophonix or have a wage per hour more than any of the workers.<br>4. Find the names of the workers who are getting more then 50 Rs. as wages per hour.<br>5. Find the jobs which are assigned after 31-DEC-2008. |
| 18 | 1. Create the following table named table as CUSTOMER with following fields-Cust_No, First_Name, Last_Name, Address, City, State, Pin, B_Date, Status.<br>Constraints on table CUSTOMER: Cust_No Primary Key, First_Name Not Null and the values for status must be in ('V','I','A').<br>2. Insert the following records into the table CUSTOMER: |

| CUST_NO | FIRST_NAME | LAST_NAME | ADDRESS | CITY | STATE | PIN | B_DATE | STATUS |
|---|---|---|---|---|---|---|---|---|
| ------- | ---------- | ---------- | ---------- | ----- | ---------- | ---------- | ------- | ------- |
| 1003 | RAJ | BAHADUR | SHANTI | UDP | KARNATAK | 576101 | 1-AUG- | V |

# C. U. SHAH UNIVERSITY
## Wadhwan City

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | VILLA | | A | | 70 | |
| | 1004 | FELIX | SIMON | M-J-56 | PJM | GOA | 403002 | 12-FEB-71 | A |
| | 1005 | RAJAN | KUTTY | A1 TRADERS | KNR | KERALA | 670001 | 9-JUN-71 | A |
| | 1006 | SHILPA | PAI | 12/4B | MNG | KARNATAKA | 574154 | 11-DEC-70 | I |
| | 1007 | BOSCO | RAKSHIT | R.K. PLAZA | BNG | KARNATAKA | 576201 | 1-JAN-71 | A |

19

➢ Perform the following queries on the tables given in Set no. 18:

1. Display all the records from the table where state is KARNATAKA.
2. Delete the row from the table where PIN CODE is 576201.
3. Change the ADDRESS as “KAVI MUDDANNA MARG” AND PIN=576104 where CUST_NO=1003.
4. Delete the records of KARNATAKA state from the table and then retrieve all the records back.
5. Select all the records with single occurrence of state from the table.
6. Sort and display the customer data, in the alphabetic order of state.
7. Sort and display the state field in the in descending order.
8. Retrieve records of Karnataka / Kerala customers who are ACTIVE (‘A’).
9. Retrieve rows where name contains the word RAJ embedded it.
10. Display all the rows whose dates are in the range of 10-JAN-70 and 31-JUL-96.