

**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 1**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | 1. Introduction to SQL, features of SQL, Rules for SQL, SQL Components (DDL, DML, DCL, TCL, DQL) 2. Viewing data in the tables: (Select Statement and where condition)    * Selecting all tables    * Describe the table    * All rows and all columns (set the Environment variable line size to custom 200)    * Clear Screen (Shift + Del) 3. Filtering Data from the table    * Selected Columns    * Selected Rows (like predicate for character data type, between predicate, ‘=’, ‘>’, ‘<’, ‘<>’, NOT and NOT IN Operator)    * Select Unique values from a column: (Distinct keyword)    * Selected Column and Selected row 4. Sorting Data in the Table: (Order By, Desc clause) |
|  | **Essential Assignment** |

|  |  |
| --- | --- |
| **Exercise** | 1. Display the structure of an EMP table 2. Display the structure of DEPT table 3. Display all the records of EMP table 4. Display all the records of DEPT table 5. Display only Name of all employees 6. Display Employee’s Name and salary 7. Display only unique departments (deptno) from EMP table 8. Display employees whose name starts with ‘J’ 9. Display all the employees’ Date of Joining (HireDate) and Salary (Sal) 10. Display all clerks (job), from EMP table 11. Display all employees’ name and salary whose salary is more than 2000 12. Display all employees who are not in department number (deptno) 30 13. Display employees with their empno, ename and mgr (i.e manager’s no) 14. Display manager’s number, job profile, department number and salary of employees Allen, Adams, Jones and Blake. 15. Display only unique Salary (sal) from EMP table 16. Display the location of department number 30 17. Display the details of the department located in ‘New York’ city 18. Display all the employees according to their names in sorted order 19. Display all the employees who are not salesman nor the manager 20. Display the details of a clerk who is getting salary more than 1000 |
|  | **Desirable Assignment** |
| **Exercise** | 1. Display all the employees whose name starts with any alphabet between B and K 2. Display all employees whose name does not contain the character ‘S’ 3. Display all the employees whose salary is between 1000 and 2000 but not exact 1500 4. Select all the employees who are hired in the year 1981 (year 81) 5. Display all the employees who are hired in February (FEB) month 6. Display all the employees according to their seniority of joining 7. Display the employees who are either ‘Analyst’, ‘President’ or ‘Manager’ 8. Display all the Salesman joined in month of September and getting salary more than or equal to 1500 9. Display the job type of an employee of department number 30 and getting salary more than 1500 10. Display the unique job titles of employees in department number 20 11. Display all the employees in a sorted order to their salary highest to lowest, hiredate latest to oldest. 12. Display all the employees in a sorted order of their job title and salary. 13. Display employees in a sorted order of their salary who are clerk. 14. Display all the managers according to their salary highest to lowest 15. Display all the employees who have joined either on 3rd or 23rd date of any month in any year. |



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 2**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | 1. Oracle Data Types: Char, Varchar, Date, Number, Long, Raw/ Long Raw 2. Data Constraints: Table Level and Row Level Constraint:    1. I/O Constraints: Unique, Primary Key, Reference Key    2. Business Rule Constraints: NOT NULL, Check, default 3. Creating Table 4. Inserting Data into Table 5. Viewing Data in the Table. |
|  | **Essential Assignment** |

**Exercise**

Create below given table called “Client\_Master” with the constrains and insert minimum 5 records in it.

|  |  |  |
| --- | --- | --- |
| **Column** | **Datatype and Size** | **Constraints** |
| C\_No | Varchar (4) | Primary Key |
| Client\_Name | Varchar (25) | Not Null |
| Pincode | Number (6) | Not Null |
| Birth\_Date | Date | Not Null |
| Occupation | Varchar (15) | Not Null |

Add the following constraints with the table fields:

1. Client\_No must start with the capital letter ‘C’ only.
2. Pincode must be of exactly 6-digit number.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C\_No** | **Client\_Name** | **Pincode** | **Birth\_Date** | **Occupation** |
| C001 | Kajal Oza | 380036 | 14-july-1987 | Manager |
| C002 | Vraj Shah | 380058 | 20-nov-1986 | CEO |
| C003 | Dhyan Dave | 387903 | 17-June-1988 | Doctor |
| C004 | Mira Vaid | 398006 | 5-jan-1976 | Professor |
| C005 | Preeti Patel | 390040 | 28-feb-1971 | Clerk |

Create following tables with constraints and records as shown in the example below. Table: Product\_Master

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **P\_No** | **Description** | **Profit\_Per** | **Unit** | **Qty\_Hand** | **ReOrder** | **Sell\_P** | **Cost\_P** |
| P001 | T-Shirt | 5 | Piece | 200 | 50 | 350 | 250 |
| P002 | Jeans | 6 | Piece | 150 | 40 | 500 | 350 |
| P003 | Skirt | 6 | Piece | 100 | 50 | 350 | 200 |
| P004 | Saree | 3 | Piece | 100 | 20 | 800 | 600 |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Create Table and Insert minimum 10 records in Salesman\_Master Table as shown below. (Sample data are shown for your reference): | | | | | | | | | | |
|  | S\_No | Name | Add1 | Add2 | City | Pin  Code | State | Salary | Target | Sales |
| S001 | Aman | 3,Dev  Appt | Vagh  odia | Baroda |  | Gujarat | 5000 | 100 | 50 |
| S002 | Omkar |  |  | Bhopal |  | MP | 4500 | 200 | 150 |
| S003 | Raj | B-104 | Verli | Mumbai | 40000  2 | Mahara  shtra | 5500 | 200 | 200 |
| S004 | Ashish | 1,  Smrut i | Ghod dol | Surat |  | Gujarat | 4500 | 150 | 100 |
| Create Sales\_Order and Sales\_Order\_Details table as shown below. Insert minimum 10 records in it.  Sales\_Order  **Fields Description**  Order\_No Primary Key, Varchar (4), Starts with ‘O’ Client\_No References Client\_Master  Order\_Date Date  Salesman\_No Refers Salesman\_Master Delivery\_Type Free or Paid. Use Char(1) i.e ‘F’ or ‘P’ Bill\_Paid Yes (‘Y’) or No (‘No’)  Delivery\_Date Date  Order\_Status ‘In Process’, ‘Pending’, ‘Fulfilled’  Sales\_Order\_Details: Order\_No and Product\_No are the composite PK  **Fields Description**  Order\_No Refers Sales\_Order Table  Product\_No Refers Product\_Master  Qty\_Ordered Positive integer  Qty\_Dispatched Positive integer  Product\_Rate Product Selling Price \* Qantity Delivered | | | | | | | | | | |
|  | **Desirable Assignment** | | | | | | | | | | |
| **Exercise** | 1. Create table “Student\_Master” with all constraints and insert minimum 5 records.  Enrollment Number must be a Primary Key and is of length, exactly 11 characters. | | | | | | | | | | |

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

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Enroll\_No | Name | DOB | City | Pincode |
| 195170683150 | Ohm Zha | 12-31-1999 | Ahmedabad | 380056 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

.

* 1. Create table “Faculty Master” with constraints as shown below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Faculty\_ID | First\_ Name | Last\_ Name | DOJ | Address | Mobile | Department |

* + - Faculty\_ID must start with a first character as ‘F’ and total length of 4 characters only.
    - First Name and Last Name must be always in upper case, with length maximum 25 characters
    - DOJ – Date of Joining
    - Mobile number must be of exactly 10 digits
    - Department must be either: ‘MCA’, M. Sc.IT’ or ‘Integrated MCA’
  1. Create following tables with constraints given below. Also enter minimum 5 records in Entrance\_Test table and 10 records in each of other two tables:

APPLICANT (AID, A\_Name, Address, B\_Date) ENTRANCE\_TEST (ET\_ID, ET\_Name, Max\_Score) ETEST\_DETAILS (AID, ETID, ETest\_Date, Score)

* + - AID: Applicant ID must start with a letter ‘A’ and should be of length 4, It will be the foreign key for ETEST\_DETAILS Table
    - Max Score of any exam must not be greater than 100
    - ET\_ID is the entrance test ID, starts with ‘T’ and is length of 4, referred as Foreign Key for ETEST\_DETAILS Table
  1. Create following tables with constraints given below. Also enter minimum 5 records in Distributor table and Item table. Enter 10 records in Dist\_Item table.

Distributor (Dno, DName, City, Phone) Item (Item\_No, Item\_Name, Price, Weight) Dist\_Item (Dno, Item\_No, Qty, Date)

* + - Dno is the PK and referred as FK for 3rd table
    - Item\_No starts with ‘I’ and maximum length of 5
    - Price, weight, Qty (quantity) must be greater than 0 and not null.



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 3**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | 1. Creating a Table from a Table 2. Inserting Data into a Table from Another Table 3. Deleting records from the table: Using Delete and Truncate operations 4. Destroying Table |
|  | **Essential Assignment** |
| **Exercise** | 1. Create a table called “Clients” from the “Client\_Master” table. 2. Insert all records of “Client\_Master” into “Clients” 3. Create table “Products” from “Product\_master” 4. Copy only selected Product’s information into “Products” table. 5. Create table “Salesman” from Salesman\_Master” where “Salesman” table contains fields: Salesman number, Name, Address1, City, Target\_Get, Achieved\_Target 6. Insert all records for the fields in new table from “Salesman\_Master” table 7. Create an “Order\_Master” table form “Sales\_Order” Table 8. Insert appropriate records in “Order\_Master” from Sales\_Order” table 9. Create “Order\_Detail” table from “Sales\_Order\_Detail” table. 10. Insert appropriate records in “Order\_Detail” from Sales\_Order\_ Detail” table. 11. Destroy tables: Client\_Master, Product\_Master and Salesman\_Master table using Truncate 12. Destroy tables: “Sales\_Order” and Sales\_Order\_Details using drop operation. |
|  | **Desirable Assignment** |

|  |  |
| --- | --- |
| **Exercise** | 1. Create all the deleted tables: “Client\_Master”, “Product\_master”, “Salesman\_Master”, “Sales\_Order” and “Sales\_Order\_Details” once again with constraint 2. Insert 10 records in each recreated table. |

Prepared By: Dr. Urja Mankad RDBMS MCA-Sem-1



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 4**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | 1. Updating the contents of a Table 2. Remove all the rows and selected rows from the Table. 3. Modifying the Structure of a Table: Add a Column, Drop a Column, Modify existing Column 4. Renaming a Table |
|  | **Essential Assignment** |
| **Exercise** | 1. Remove the product details column from Product\_Master Table 2. Remove all Sales\_Orders whose QtyOrdered = 1 in “Sales\_Order\_Detail” table 3. Modify the OrderStatus to “Fulfilled” where the OrderStatus is “In Process” in Sales\_Order Table. 4. Add column City and State in Client\_Master table 5. Add a Client record for Client Name Ashwini Joshi, city = Banglore 6. Change the city of the client “Ashwini Joshi” from “Bangalore” to “Chennai” 7. Change the Cost Price (Cost\_P) of a Skirt 8. Increase the sell price of saree by Rs. 20 9. Delete all Salesman from Salesman\_Master table whose salary is more than 3500 10. Delete all Products from Product\_master where Quantity on Hand (Qty\_Hand) is less than 50 11. Add a column Mobile\_No of type Number, in the Client\_Master Table 12. Change the name of Salesman\_Master table to “SMan\_Mst” |
|  | **Desirable Assignment** |
| **Exercise** | 1. Add a column of Experience in the Faculty\_Master Table 2. Modify the column Address to Area in Faculty\_Master Table (Hint: Remove existing Column and Add new Column) |

|  |  |
| --- | --- |
|  | 1. Modify the Applicant Table name to Applicant\_Master 2. Add a column “Hobby” in Applicant\_Master table. 3. Add a column “Location” in the “ETest\_Detail” table. Keep it varchar(10) 4. Modify the column “Location” and increase the size upto 25 characters. 5. Change the field “Address” to “City” in Applicant\_Master Table 6. Increase the price of all products by 5% in “Item” Table 7. Change the name of the Item table to “Item\_Master” 8. Remove the Weight column from Item\_Master Table. |



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 5**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | Logical operators like: AND, OR, Combining the AND and OR operator, Not Operator  Aggregate, Math and String functions: MIN, COUNT, MAX, AVG, SUM, ABS, POWER, ROUND, SQRT, EXP, MOD, TRUNC, FLOOR, CEIL, LOWER, INITCAP, UPPER, SUBSTR, ASCII, COMPOSE, TRANSLATE, LENGTH, LTRIM, RTRIM, TRIM, LPAD, RPAD, NOW (), CURDATE (), DATE (), EXTRACT (), DATE FORMAT functions TO\_DATE (), TO\_CHAR () |
|  | **Essential Assignment** |
| **Exercise** | 1. Display all the clients whose age is more than 25 years but less than 50 yesrs 2. Display all the Salesman from Salesman\_Master table whose “Sale” is more than his “Target” and “Target” is more than 100 3. Find the Salesman who are neither from Baroda nor from Surat. 4. Display total number of clients using Client\_Master Table. 5. Display the highest salary s salesman is getting. 6. Display all item names in upper case letters only 7. Display current date and time 8. Display average target given to the salesman. 9. Display the Birth Date (DOB in Student\_Master Table) in a new format. (Eg. February 12, 1998) 10. Display Date of Joining (DOJ, Faculty\_Master) of all faculties in DD/MM/YY format 11. Display Only Birth Date and Month of all the students from Student\_Master Table |

|  |  |
| --- | --- |
|  | Employe Name Pincode DOB Address Salary Occupati e\_No on  E100 Kapil 380036 14-july- Bombay 10000 Manager  1987  E101 Manish 385670 20-nov- Pune 5000 CEO 1986  E102 Ramesh 387903 17-June- Goa 25000 Engineer  1988  E103 Rohan 390009 15-jan- Delhi 35000 Clerk 1976  E104 Raj 398006 28-feb- Abad 30000 Doctor 2000  Use the above table and write appropriate DML statements to solve following queries.   1. List the names of all employee having ‘a’ as the second letter in their names. 2. List the employee whose occupation fist letter is ‘M’. 3. List the employees who have the second character as a or o. 4. List the employee details of the named Rohan, Ramesh and Raj. 5. List all employee who stay in ‘Banglore’ or ‘Pune’. 6. List the details of employee E102 and E104. 7. List the details of employee whose salary is greater than 5000 and less than 20000. 8. List the name of employees who are not in the state ‘Bombay’. 9. List employee whose salary is less than 1000 and calculate a new incremented salary by 15% as Salary \* .15 10. Count the total number of employees. 11. Calculate the average salary of all the employees. 12. Determine the maximum and minimum salary. Rename the output as max\_salary and min\_salary respectively. 13. Count the number of employees having salary less than or equal to 15000. 14. List the details of Employee month wise in DD/MM/YY format. 15. List the DOB in the format ‘DD-Month-YY’ e.g 12-February-91 |
|  | **Desirable Assignment** |
| **Exercise** | 1. Display total orders that are fulfilled. (Hint: use the status field of Sales\_Order table) 2. Display only the Surname of all the Students. (Hint: Use Name field of Student\_Master table, which contains full name) 3. Display all Orders received in the month of “April” (Hint: Order\_Date of Sales\_Order table) 4. Display all Clients whose age is less than 30 years (Hint: Extract only year from Birth\_Date of Client\_Master Table)   Student\_ First Last DOB Address CPI Departm No name name ent  E100 Kapil Sharma 14-july- Ahmeda 96.4 MCA  1987 bad |

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

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| E101 | Rekha | Goyal | 20-nov- 1986 | Baroda | 56.8 | BBA |
| E102 | Ramesh | Joshi | 17-June- 1988 | Surat | 78.9 | BCA |
| E103 | Aliyah | Shekh | 15-jan- 1976 | Mehsan a | 65.8 | MSCIT |
| E104 | Rajesh | Gupta | 28-feb- 2000 | Himatna gar | 94.6 | PGDCA |

Use the above table and solve following queries:

1. List the names of all student having ‘e’ as the third letter in their names.
2. List the student whose name start with ‘Ra’.
3. List all student who stay in ‘Mehsana and ‘Baroda.
4. List the details of student E102 or E104.
5. List the details of student whose CPI is greater than 80.
6. List the name of students who are not in the city ‘Baroda’.
7. Count the total number of students.
8. Calculate the average CPI of all the students.
9. Determine the maximum and minimum CPI. Rename the output as max\_CPI and min\_CPI respectively.

Use below given tables to solve queries based on it:

Distributor (Dno, DName, City, Phone) Item (Item\_No, Item\_Name, Price, Weight) Dist\_Item (Dno, Item\_No, Qty, Date)

1. Count total distributors
2. Find the total price of all items
3. Find the item wise total quantity of each item
4. Display all items whose name starts with “S”
5. Display Distribution details of all items with month (from the date) only.
6. Display the distributors whose city name starts with “B”
7. Find the highest weight of an item
8. Display the distribution details like Item\_No, Qty and Year (from the date)
9. Display the year wise average quantity of items that are distributed



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 6**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | Select data using: Order By, Group by and Having clause |
|  | **Essential Assignment** |
| **Exercise** | 1. Use following tables and solve given queries below it.   APPLICANT (AID, A\_Name, City, B\_Date) ENTRANCE\_TEST (ET\_ID, ET\_Name, Max\_Score) ETEST\_DETAILS (AID, ETID, ETest\_Date, Score)   * 1. How many applicants have appeared for each test   2. Display highest score for each test   3. Display applicant’s ID who appeared for more than 3 tests   4. Calculate applicant’s average score across all test they have appeared in   5. Display number of applicants by city   6. Display ETID and Average score where average score is more than 50.   7. Count date wise total entrance test to be held.  1. Use the following tables and solve below given queries.   Distributor (Dno, DName, City, Phone) Item (Item\_No, Item\_Name, Price, Weight) Dist\_Item (Dno, Item\_No, Qty, Date)   * 1. Display city wise total number of distributors   2. List distributors’ no by who distributed more than 50 items in month of July   3. List Item\_No with more than 5000 Qty delivered |

|  |  |
| --- | --- |
|  | 1. List Dno who delivered more than 50 items for each month 2. Display item details in descending order of price and ascending order of weight 3. Show all distributors in alphabetical order of City and DName 4. Calculate average quantity of items distributed on each day. 5. Find the weight wise average price of items |
|  | **Desirable Assignment** |
| **Exercise** | 1. Use the following tables and solve below given queries.   Worker (Id, Name, Wages\_Per\_Hr) Job (Job\_Id, Job\_Type)  Assigned (W\_Id, J\_Id, Start\_Date, Status, Total\_Hrs)   * 1. Display all workers in descending order of their Wages\_Per\_Hr   2. Count workers in each J\_Id   3. List all the W\_Id who has worked for more than 100 hrs   4. Count total jobs which are pending till date   5. Display all Job\_Id according to alphabetical order of their type  1. Create below given relations with appropriate key constraints. Also add a constraint specifying that the marks entered should not be negative. Solve the below given queries    1. Display all the students whose first name starts with the character ‘C’ and last name starts with ‘P’    2. Insert a new column address in student table where datatype is varchar and size is 30.    3. Display those students whose DOB is before 1st march 2000.    4. Display total number of students in each course.    5. Display the names of students who have last name like ‘Patel’, ‘Shah’ or ‘Desai’    6. Display the names of students who have not failed in any subject.    7. Display the age of all the students.    8. Student name initial letter must be capital. Add a constraint using Alter command.    9. Print the details of students who got second highest marks.    10. Fond out the details of students who have not enrolled in any course.    11. Print the details of students who born in current year.    12. Find out in details of students who have enrolled for maximum courses. |



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 7**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | Subqueries   * Using sub-query in the FROM clause * Using co-related sub-queries * Using multi-column sub-queries * Using sub-query in an Order-By clause * Using Exits and Not Exists operator |
|  | **Essential Assignment** |
| **Exercise** | Use Following Tables and solve given queries:  (I)  STUDENT (rollno, fname, lname, dob)  COURSE (rollno,courseno,fees, coursename, max\_marks, pass\_marks)   1. Find all students whose marks is higher than the average marks of the students in their course.(using corelate sub query) 2. List the student detail whose marks is same as ‘Kapil’. 3. Display the name of students who have enrolled for ‘html’ course. 4. Display details of students who have enrolled in courses. 5. Update max\_marks =80 where courseno is 105. 6. Delete those rows where pass\_marks is less than 40. 7. Give the name of the course in which maximum number of students are enrolled. 8. Display the first name and last name of student who have enrolled for ‘oracle’ course.   (II)  Employee (emp\_no, fname, lname, dob,address) Company(comp\_id,emp\_no,company\_name,salary,department,designation) Customer(cust\_no,fname,lname,address)   1. Find all students whose salary is higher than the average salary of the employees in their |

|  |  |
| --- | --- |
|  | departments.   1. List the employee detail whose salary is same as “Jay”. 2. Display the name of employees who work in ‘mrkt’ department. 3. Display details of employees not belong to any department. 4. Find out all the customers having same name as the employees. (Using multi column subquery). 5. Delete those rows where department=’sales’ 6. Update salary=salary\*0.10 where emp\_no=105.   (III)  PRODUCTS (Prod\_ID, Prod\_Name, Supplier\_ID, Cat\_ID, Unit, Price) ORDER\_DETAILS (OrderDetail\_ID, Order\_ID, Prod\_Id, Quantity)   1. Lists the ProductName if ANY records in the Order\_Details table has Quantity equal to 10 2. lists the ProductName if ALL the records in the OrderDetails table has Quantity more than 15 3. Display Products which are not ordered 4. List the products which is ordered for more than 200 Quantity 5. List the products whose Unit is Kg or Lt (Use IN operator) |
|  | **Desirable Assignment** |
| **Exercise** | Use Following Tables to solve given queries Table: Sales\_Order/ Order\_Master  **Fields Description**  Order\_No Primary Key, Varchar (4), Starts with ‘O’ Client\_No References Client\_Master  Order\_Date Date  Salesman\_No Refers Salesman\_Master Delivery\_Type Free or Paid. Use Char(1) i.e ‘F’ or ‘P’ Bill\_Paid Yes (‘Y’) or No (‘No’)  Delivery\_Date Date  Order\_Status ‘In Process’, ‘Pending’, ‘Fulfilled’  Table: Sales\_Order\_Details: Order\_No and Product\_No are the composite PK  **Fields Description**  Order\_No Refers Sales\_Order Table  Product\_No Refers Product\_Master  Qty\_Ordered Positive integer  Qty\_Dispatched Positive integer  Product\_Rate Product Selling Price \* Qantity Delivered   1. Display all the sales order details for the sales order\_no ‘O101’ 2. Find the Salesman\_No whose order contains less than 5 products |

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

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

|  |  |
| --- | --- |
|  | 1. Display all the sales\_order\_date on which the Qty\_Ordered and Qty\_Dispatched are not same 2. Display all sales details like, Order\_No, Order\_Date, Order\_Status and Total Bill Amount (Hint: Total bill amount will be the sum of all product\_rate of a particular order) 3. Display the product purchased by the client\_no C101 4. Display all products (product\_no) whose delivery date is ‘12-Dec-2021’ |



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 8**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | What is Join?  Explain types of Join.  Understand Equi Join with examples |
|  | **Essential Assignment** |
| **Exercise** | (I) Use following tables and solve given queries below it.  APPLICANT (AID, A\_Name, City, B\_Date) ENTRANCE\_TEST (ET\_ID, ET\_Name, Max\_Score) ETEST\_DETAILS (AID, ETID, ETest\_Date, Score)   1. Display Entrance Test ID (ETID) Wise highest marks scored by any applicant. 2. Count ETID wise total number of applicants appeared for the test 3. Find the minimum number of applicants in the entrance test. 4. Count city wise number of applicants registered 5. Display all the entrance test details for which the applicant “Ohm Patel” appeared   (II) Use the following tables and solve below given queries.  Distributor (Dno, DName, City, Phone) Item (Item\_No, Item\_Name, Price, Weight) Dist\_Item (Dno, Item\_No, Qty, Date)   1. Display all the distributor’s name who supplies Item\_No 5 2. Display the item which is distributed maximum time 3. Display all the items that are distributed by the distributor “Dev Shah” 4. Display the Item\_Name and Quantity that are received in month of July in 2021 |

|  |  |
| --- | --- |
|  | 5. Display all the items whose price is less than 1000 and received Qty =10 |
|  | **Desirable Assignment** |
| **Exercise** | Use the following tables and solve below given queries.  Worker (Id, Name, Wages\_Per\_Hr) Job (Job\_Id, Job\_Type)  Assigned (W\_Id, J\_Id, Start\_Date, Status, Total\_Hrs)   1. Display all the workers assigned for Job “Assemble” 2. Display total hours of the worker “Deep Jha” spent on Job “Packing” 3. Display all the job types that are starting from Sept-2021. 4. What are the total hours allocated for Job “Cutting” 5. Display all the jobs that are still pending |



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 9**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | Inner Join, Outer Join, Cross Join |
|  | **Essential Assignment** |
| **Exercise** | Deposit (AccNo, Cust\_No, Bname, Amount, Adate) Branch (Bname,city)  Customers (Cust\_No, Cname, City)  Borrow (LoanNo, Cust\_No, Bname, Amount)   1. Get the details of the customers ‘Deepak’ 2. Give name of customer who are borrowers and depositors and having living city Ahmedabad 3. Give city as their city name of customers having same living branch 4. Write a query to display the Customer name with his Deposited Amount, deposited date and branch name in which he has deposited. 5. Create a unique listing of all customers who have taken loan from “Ahmedabad’ 6. Display all account details of Baroda branch 7. Display customer’s details who have deposited less than 10,000 and borrowed more than 50,000 8. Display all account details where there is any deposited after the year 2020. |

|  |  |
| --- | --- |
|  | **Desirable Assignment** |
| **Exercise** | Use the following tables and solve below given queries.  Colleges (CollegeId, CollName, City) Course (CourseID, Cname, Duration)  Student (StudentId, Sname, DOB, Pincode, City) Admission (CollegeId, StudentID, CourseId, Fees, date)   1. Get the details of the student ‘Sumit’. 2. Give name of student who are taken admission in ‘LJ’ college. 3. Give city as their city name of students having same city of college. 4. Write a query to display the student’s name, college name, and course name for all students. 5. Create a query to display the name and fees paid date of all students those have paid the fees after student ‘Jay’. |



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 10**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | Set Operation: UNION, UNION ALL, INTERSECTION, MINUS clause |
|  | **Essential Assignment** |
| **Exercise** | (I) Use the following tables and solve below given queries.  Worker (Id, Name, Wages\_Per\_Hr) Job (Job\_Id, Job\_Type)  Assigned (W\_Id, J\_Id, Start\_Date, Status, Total\_Hrs)   1. Display all the workers’ Id assigned for Job 1 and 2 both (J\_Id =1 and J\_Id=2) 2. List the workers who are working for J\_Id =1 and not for J\_Id = 2 3. Display the workers Id who are working for more than 6 hours and status is pending. And Union them all with the worker’s Id who are working for less than 6 hours and status is completed   (II) Use following tables to solve below given queries.  FYRankers (Enrol\_No, Name, SPI) SYRankers (Enrol\_No, Name, SPI)   1. Display the name of the student who is ranker in ‘FY’ or ‘SY’ 2. Display the name of the student who is ranker in ‘FY’ or ‘SY’ including duplicate data 3. Display the name of the students who is ranker in ‘FY’ or ‘SY’ and having SPI more than 7 4. Display the name of the student who is ranker in both FY and SY 5. Display the name of the student who is ranker in FY but not in SY 6. Display the name of the student who is ranker in SY but not in FY |

|  |  |
| --- | --- |
|  | **Desirable Assignment** |
| **Exercise** | Use the following tables and solve below given queries.  customer (cust\_id, name, city, country) supplier (supp\_id, name, city, country, cust\_id)   1. Display distinct city of both the customer and supplier table in sorted order. 2. Display city of both the tables with duplicate data. 3. Display the details of Ahmedabad city of both customer and supplier table. 4. Display the name of students who have enrolled for ‘html’ course. 5. Display customer details who are living in ‘ahmedabad’ and name start with ‘A’. 6. Find the details of the supplier who are not customer. |

Prepared By: Dr. Urja Mankad RDBMS MCA-Sem-1



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 11**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | View   * Introduction to View and benefits of creating Views * Creating View * Selecting data from the View * Renaming column of the View * Update a View * Drop a View. |
|  | **Essential Assignment** |
| **Exercise** | (I) Use the EMP and DEPT tables to perform following queries:   1. Create a view called Emp\_View from Employee table. 2. Renaming the columns of Emp view. 3. Select Employee Name where department is ‘Marketing’ or ‘Loan’. 4. Update Name=’Vaishali’ where name is ‘Sharanam’. 5. Delete a record where name is ‘Vaishali’. 6. Remove a view Emp from database.   (II) Use following tables and solve given below queries:  Branch (bno, bname)  Address (addno, bno, type, Addr1, Addr2, City, State, Pincode)   1. Create bno in Branch as a primary key, which is the reference key in address table 2. Add constraint for type field will have value ‘H’ or ‘B’ (H-Head, B-Branch) 3. Create a view Branch\_Master from Brach and Address Table. 4. Update Pincode=400079 where bno=102. 3. 5. Delete records where bno=102; 6. Remove a view Branch\_Master from database |

Prepared By: Dr. Urja Mankad RDBMS MCA-Sem-1



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 12**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
|  | REVISION – All SQL Statements to solve mixed queries |
|  | **Essential Assignment** |
| **1** | Emloyee (**EmpNo**, Ename, Salary, Designation) Dept (**EmpNo**, DeptNo)   1. Display all rows for salary greater than 5000. 2. Display the deptno for the name ‘shyam’. 3. Add a new column DeptName in Dept table. 4. Change the designation of ename=’ram’ from ‘clerk’ to ‘senior clerk’. 5. Find the total salary of all the rows. 6. Display EmpNo, Ename, DeptNo, DeptName. 7. Drop the table Employee. |
| **2** | Student (**StuNo**,sname,marks,college) Course (**StuNo**,CourseId)   1. Display all rows for student greater than 80. 2. Display the CouseId for the name ‘shyam’. 3. Add a new column CollegeName in Course table. 4. Change the college of sname=’ram’ from ‘LJ’ to ‘new LJ’. 5. Find the total marks of all the rows. 6. Display StuNo,sname,CoursId,CourseName. |
| **3.** | BOOK\_CATALOG (book\_code, title, ISBN\_No, Publisher\_Name, yr\_of\_release, total\_copies )  MEMBER (member\_code, member\_name, mem\_ship\_dt)  ISSUE (Issue\_id, member\_code, book\_code, issue\_date, issue\_ret\_dt) |

|  |  |
| --- | --- |
|  | 1. Create the above tables with appropriate key constraints 2. Publisher name should be entered in capital letters 3. Display the book details which contain ‘Database’ somewhere in the book title 4. Display the member and book details for books issued between 1st January 2014 and 30th March 2015 5. Display book details whose all copies are issued 6. Display the book details of ‘Pearson’ publications 7. Display those books having name Database and SQL 8. Create a table LIBRARY\_USER having the same structure of MEMBER table with no records 9. Drop table Library\_USER from database |
| **4.** | STUDENT (rollno, fname, lname, dob)  COURSE (courseno, coursename, max\_marks, pass\_marks)   1. Create the above tables with appropriate key constraints. 2. Marks cannot be less than 0. 3. Display the names of students who have last name like ‘Patel’, ‘Shah’ or ‘Desai’ 4. Display the names of students who have not failed in any subject 5. Display the age of all the students |
| **5.** | CUSTOMER (cno, cust\_name)  ITEM (item\_no, item\_name, item\_price, stock)  CUST\_ITEM (cust\_no, item\_no, qty\_purchased, date\_of\_trans)   1. Create the above tables with appropriate key constraints. 2. Qty\_Purchased cannot be 0. 3. Retrieve the name of customers who have purchased the costliest item from the item list. 4. Display the total item price. 5. Alter table ITEM and add column item class, which can have values as A, B or C. 6. Display those customers who have purchased ‘chair’.   7. Display total number of items purchased by each customer. 6. Display the customer details whose name start with ‘n’. |
| **6.** | EMPLOYEE (empid, ename, gender, DOB, J\_date, designation) PROJECT (pid, pname, tot\_duration, location)  WORKS\_ON (empid, pid, type\_work, num\_days)   1. Create the above tables with appropriate key constraints. 2. Add a constraint that tot\_duration (total duration for the project given) in the table PROJECT is between 7 to 30 days. 3. Display those employees whose designation is Manager. 4. Display those employees whose joining date is between 1st march to 20th march. 5. Display those employees who has worked for 25 Days. |
| **7** | Competition (Comp\_code, Comp\_name (Dancing, Painting, GK, etc.))  Participants (Part\_no, Part\_name, DOB, Address, EmailID, Contact\_number ) Scorecard (Part\_no, Comp\_code, Judge\_no [1, 2, 3], Marks) |

|  |  |
| --- | --- |
|  | 1. Apply appropriate key constraints on the above tables. 2. Participants’ contact number must be of exactly 10 digits. 3. Find those participants who have registered both for ‘Dancing’ and ‘Painting’ 4. Display Names of Participants in CAPITAL latter in descending order of their DOB. 5. Display those part5iucipants with highest marks. 6. Display all the Participants whose first name starts with the character ‘C’ and last name starts with ‘P’ 7. List the names of c who have an ‘A’ and also a ‘B’ somewhere in their names. 8. List out competition wise participant name. |
| **8** | SUPPLIER (s\_id, s\_name, contact)  PARTS (part\_id, part\_name, colour, unit\_rate) CATALOG (s\_id, part\_id, qty)   1. Apply appropriate key constraints on the above tables. 2. Part colour can be black, white and grey. 3. Add the column Contact\_Person in the Supplier table with NOT NULL constraint. 4. List the names of Suppliers who have an ‘A’ and also a ‘B’ somewhere in their names.   5. Display those parts whose unit rate is more than 500rs. |



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 13**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | Introduction to PL/SQL: Advantages of PL/SQL  The Generic PL/SQL Block, PL/SQL Execution Environment, Control Structure   * Conditional Control |
|  | **Essential Assignment** |
| **Exercise** | 1. Print a static string “Hello Every One…!” using an anonymous PLSQL block and execute it. 2. Write a PLSQL block to display a greeting message like: “Hi!! Today is 3rd November 2021, Friday. 3. Declare a string variable to store student’s name, define three integer variables to store marks of 3 subjects (out of 50) of that student. Write a PLSQL code to calculate total of all three subjects and print the result in percentage. 4. Write a program to divide 2 numbers and if the denominator if 0 then handle the exception. 5. Write a user defined exception for above program 3 where if marks are less than 0 then appropriate error message must be shown as exception. 6. Write a PLSQL block to find the largest of three numbers |
|  | **Desirable Assignment** |
| **Exercise** | 1. Display a greeting message as “Good Morning”, “Good Afternoon” or “Good Evening” according to the current time. 2. Implement PL/SQl Block which exchanges value of two number variables without using third variable. |

|  |  |
| --- | --- |
|  | 3. Implement PL/SQL Block which performs Menu driven Mathematical operations as shown below:  Enter x: 2  Enter y: 1  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   1. Addition 2. Subtraction 3. Division 4. Multiplication 5. Power 6. EXIT   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 14**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | PL/SQL Control Structure (Cont…)   * Iterative Control * Sequential Control |
|  | **Essential Assignment** |
| **Exercise** | 1. Write a PLSQL block to print all the prime numbers between 1 to 50 2. Display all the integer numbers between 4 to 40 which are divisible by 3 using “Exit When” statement. 3. Implement a PL/SQL Block which takes input number of rows and displays triangle.   \*  \*\*  \*\*\*  \*\*\*\*  \*\*\*\*\*   1. Use following tables and write below given PL/SQL blocks.   PRODUCTS (Prod\_ID, Prod\_Name, Supplier\_ID, Cat\_ID, Unit, Price) ORDER\_DETAILS (OrderDetail\_ID, Order\_ID, Prod\_Id, Quantity)   * 1. Write a PLSQL block to display total number of products ordered in Order\_ID = 3   2. Write a PLSQL block to update the price (actual price + 5) of product with Id = 2   3. Write a PLSQL block to delete the products of Cat\_Id = 3   4. Write a PLSQL block to insert any product whose cat\_id = 3   5. Write a PLSQL block to display Supplier\_Id and their total number of products they supply |

|  |  |
| --- | --- |
|  | **Desirable Assignment** |
| **Exercise** | 1. Implement PL/SQL block that inputs a string and checks palindrome or not. Note: String is Palindrome if reverse of string equals to the original string. Eg. Input String = “aibohphobia”   Reverse of string = “aibohphobia” Output: “aibohphobia” is Palindrome.  Eg Input String = “level” Reverse of string = “level”  Output: “level” is Palindrome”   1. Create a table AreaofCircle (Radius, Area) and write a PLSQL block to insert multiple rows for radius 3 to 10 2. Write a PLSQL block to display all the Product’s details which are not ordered yet. 3. Update the Products table for Product\_name, if the name does not begin with a capital letter, then make it capital. |
|  |  |



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 15**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | PLSQL Character Set, Data Types and Constants  Transaction control commands – Commit, Rollback, Savepoint |
|  | **Essential Assignment** |
| **Exercise** | Essential Assignment:  PRODUCTS (Prod\_ID, Prod\_Name, Supplier\_ID, Cat\_ID, Unit, Price) ORDER\_DETAILS (OrderDetail\_ID, Order\_ID, Prod\_Id, Quantity)   1. Write a PLSQL block to display total number of products ordered in Order\_ID = 3 2. Write a PLSQL block to update the price (actual price + 5) of product with Id = 2 3. Write a PLSQL block to delete the products of Cat\_Id = 3 4. Write a PLSQL block to insert any product whose cat\_id = 3 5. Write a PLSQL block to display Supplier\_Id and their total number of products they supply |
|  | **Desirable Assignment** |
| **Exercise** | 1. Create a table AreaofCircle (Radius, Area) and write a PLSQL block to insert multiple rows for radius 3 to 10 2. Write a PLSQL block to display all the Product’s details which are not ordered yet. 3. Update the Products table for Product\_Name, if the name does not begin with a capital letter, then make it capital. |



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 16**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | Understanding Cursor, Cursor FOR loops Parameterized Cursor, Cursor within Cursor |
|  | **Essential Assignment** |
| **Exercise** | (I)  Create a table Student (R\_No, Name, Sub1, Sub2, Sub3, Total, Grade)  Insert 10 records in the above table where each subject carries maximum 100 marks. Don’t enter the total and grade manually.   1. Write a PLSQL block to calculate and update the Total for each and every student. 2. Calculate the grade of all students, based to total (>70 AA, >60 A, >50 B, >35 C, else Fail) 3. Write a Cursor to find the first 3 rankers based on the total marks.   (II)   1. Add a Salary and Bonus column in the Faculty\_Master Table and calculate the bonus of each faculty of “MCA” department which is based on the 5% of their salary. If the salary in less than 25000, then raise the exception. 2. Display name of 2 faculties getting maximum bonus.   (III)  Use following tables are create a report as shown below. supplier (sid, sname, contactnum)  parts (pid, pname, color, unitrate)  catalog (sid, pid, qty) [primary key(sid,pid)] |

Prepared By: Dr. Urja Mankad RDBMS MCA-Sem-1

|  |  |
| --- | --- |
|  | Write a PLSQl Cursor to take a supplier’s name as an input from the user and prepare a report in the following format:  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  PART ID PART NAME QUANTITY UNIT PRICE TOTAL  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Grand Total:  Raise a user defined exception when the supplier’s name not found in the database table. |
|  | **Desirable Assignment** |
| **Exercise** | (I)  Create following tables and perform operations given below. Competition (Comp\_code, Comp\_name )  Participants (Part\_no, Part\_name, DOB, Address, EmailID, Contact\_number ) Scorecard (Part\_no, Comp\_code, Judge\_no ,Marks) [F.K. Part\_No and Comp\_Code]   1. Create a Competition wise participants’ report using PL/SQL block. Competition Name Participants’ Name Marks 2. Handle the exception where no record found. (II)   Use Following Tables:  Emp\_Master (Emp\_Code, Emp\_Name, Birth\_Date) Dept\_Master (Dept\_Code, Dept\_Name, Budget) Salary (Dept\_Code, Emp\_Code, Salary)  Write a pl/sql block to call a procedure that counts total employees in each department and display them. Also handles the exception where no record found.  (III)  Use Following Tables:  Employee\_master(EmpCode , Emp\_Name , Dept\_Id, Emp\_Address , DOB , Basic\_Salary) Department\_master(Dept\_Code ,Dept\_Name)  Create a PL/ SQL block to prepare the salary slip for the employee in the following format, whose Employee Code is provided by the user.  Salary Slip for the month <current month and year>  Employee Code: <E102> Employee Name: <John Smith> Department Name: <Finance>  --- |

|  |  |
| --- | --- |
|  | Basic Salary DA HRA Medical P.F.  ---  Total Salary :  Note:  HRA is 15% of basic salary DA is 30% of basic salary Medical is 1% of basic salary  P.F. is 10% of basic salary  Raise an exception if the record not found for the entered EmpCode. |



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 17**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | Procedure: IN, OUT and INOUT parameters Function  Difference between procedure and function |
|  | **Essential Assignment** |
| **Exercise** | 1. Write a procedure which will take Faculty ID as an input and will display all the information of that faculty 2. Write a stored procedure that uses an INOUT parameter and an IN parameter. The user will supply 'M' or 'F' through IN parameter (emp\_gender) to count a number of male or female from Employee table. The INOUT parameter (mfgender) will return the result to a user. 3. Write a procedure which will take minimum limit and maximum limit of salary and the execution of the procedure will display name of the employees having salary between the range.   Use following tables and do as directed:  Movie (movie\_id, movie\_name, date\_of\_release) Screen (screen\_id, location, max\_capacity)  Current (movie\_id,screen\_id, date\_of\_arrival, date\_of\_closure)   1. Consider the above table and write a function to return the movie name which arrived today. 2. Write a function that will return max\_capacity of a screen by providing Screen\_Id as a parameter. |
|  | **Desirable Assignment** |
| **Exercise** | PRODUCTS (Prod\_ID, Prod\_Name, Supplier\_ID, Cat\_ID, Unit, Price) ORDER\_DETAILS (OrderDetail\_ID, Order\_ID, Prod\_Id, Quantity)   1. Write a PLSQL block to display total number of products ordered in a user entered |

|  |  |
| --- | --- |
|  | Order\_ID   1. Write a PLSQL block to update the price of a product with User entered Id and increment amount. (eg. ID=2 amount = 5, means the price of prod\_ID will be updated by 5) 2. Write a PLSQL block to delete the products of user mentioned Category (Cat\_Id) 3. Write a PLSQL block to insert a product in the category mentioned by the user as input (Cat\_Id) 4. Write a PLSQL block to take Supplier\_Id as input and display total number of products they supply |

Prepared By: Dr. Urja Mankad RDBMS MCA-Sem-1



**COURSE NAME: MCA, 1st Semester Course Code: 040110103**

**Subject Name: Relational Database Management System Day 18**

|  |  |
| --- | --- |
| **Detailed Schedule** | |
| **Topics** | Explain Triggers. Use of Trigger Types of Triggers |
|  | **Essential Assignment** |
| **Exercise** | 1. Create a trigger that when the new record is inserted in the Movie table 2. Create a trigger that will be fired when a Movie is deleted 3. Create a trigger when the max\_capacity of the screen is more than 100. |
|  | **Desirable Assignment** |
| **Exercise** | 1. Insert records in Student\_Master table where the Enrollment Numbers must be inserted by 1 increment of the previous one. 2. Insert the record in the Employee Master table where the employee code must be inserted automatically as <dept\_code><no> where the no will start from 1 in each department and must increment by 1 from the previous one. |

Prepared By: Dr. Urja Mankad RDBMS MCA-Sem-1