|  |  |
| --- | --- |
| **Lab 1** | **Perform basic SQL commands like Create, Insert** |
|  | **Database Name: Branch\_DIV\_Rollno (Example: CSE\_3A\_101)**  **Note: Create all the tables under above database.**  **Create following tables and insert the data into tables using Query as shown below.**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **DEPOSIT** | | | **Column\_Name** | **DataType** | | ACTNO | INT | | CNAME | VARCHAR(50) | | BNAME | VARCHAR(50) | | AMOUNT | DECIMAL(8,2) | | ADATE | DATETIME | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ACTNO** | **CNAME** | **BNAME** | **AMOUNT** | **ADATE** | | 101 | ANIL | VRCE | 1000.00 | 1-3-95 | | 102 | SUNIL | AJNI | 5000.00 | 4-1-96 | | 103 | MEHUL | KAROLBAGH | 3500.00 | 17-11-95 | | 104 | MADHURI | CHANDI | 1200.00 | 17-12-95 | | 105 | PRMOD | M.G. ROAD | 3000.00 | 27-3-96 | | 106 | SANDIP | ANDHERI | 2000.00 | 31-3-96 | | 107 | SHIVANI | VIRAR | 1000.00 | 5-9-95 | | 108 | KRANTI | NEHRU PLACE | 5000.00 | 2-7-95 | | 109 | MINU | POWAI | 7000.00 | 10-8-95 | | |  |  | | |  |  | | --- | --- | | **BRANCH** | | | **Column\_Name** | **DataType** | | BNAME | VARCHAR(50) | | CITY | VARCHAR(50) | | |  |  | | --- | --- | | **BNAME** | **CITY** | | VRCE | NAGPUR | | AJNI | NAGPUR | | KAROLBAGH | DELHI | | CHANDI | DELHI | | DHARAMPETH | NAGPUR | | M.G. ROAD | BANGLORE | | ANDHERI | BOMBAY | | VIRAR | BOMBAY | | NEHRU PLACE | DELHI | | POWAI | BOMBAY | | |  |  | | |  |  | | --- | --- | | **CUSTOMERS** | | | **Column\_Name** | **DataType** | | CNAME | VARCHAR(50) | | CITY | VARCHAR(50) | | |  |  | | --- | --- | | CNAME | CITY | | ANIL | CALCUTTA | | SUNIL | DELHI | | MEHUL | BARODA | | MANDAR | PATNA | | MADHURI | NAGPUR | | PRAMOD | NAGPUR | | SANDIP | SURAT | | SHIVANI | BOMBAY | | KRANTI | BOMBAY | | NAREN | BOMBAY | | |  |  | | |  |  | | --- | --- | | **BORROW** | | | **Column\_Name** | **DataType** | | LOANNO | INT | | CNAME | VARCHAR(50) | | BNAME | VARCHAR(50) | | AMOUNT | DECIMAL(8,2) | | |  |  |  |  | | --- | --- | --- | --- | | LOANNO | CNAME | BNAME | AMOUNT | | 201 | ANIL | VRCE | 1000.00 | | 206 | MEHUL | AJNI | 5000.00 | | 311 | SUNIL | DHARAMPETH | 3000.00 | | 321 | MADHURI | ANDHERI | 2000.00 | | 375 | PRAMOD | VIRAR | 8000.00 | | 481 | KRANTI | NEHRU PLACE | 3000.00 | | |  |  | |
| **Lab 2** | **Perform SQL queries for Select with operators** |
|  | **SELECT Operation**  **From the above given tables perform the following queries:**   1. Retrieve all data from table DEPOSIT. 2. Retrieve all data from table BORROW. 3. Retrieve all data from table CUSTOMERS. 4. Display Account No, Customer Name & Amount from DEPOSIT. 5. Display Loan No, Amount from BORROW. 6. Display loan details of all customers who belongs to ‘ANDHERI’ branch from borrow table. 7. Give account no and amount of depositor, whose account no is equals to 106 from deposit table. 8. Give name of borrowers having amount greater than 5000 from borrow table. 9. Give name of customers who opened account after date '1-12-96' from deposit table. 10. Display name of customers whose account no is less than 105 from deposit table. 11. Display name of customer who belongs to either ‘NAGPUR’ or ‘DELHI’ from customer table. (**OR & IN**) 12. Display name of customers with branch whose amount is greater than 4000 and account no is less than 105 from deposit table. 13. Find all borrowers whose amount is greater than equals to 3000 & less than equals to 8000 from borrow table. (**AND & BETWEEN**) 14. Find all depositors who do not belongs to ‘ANDHERI’ branch from deposit table. 15. Display Account No, Customer Name & Amount of such customers who belongs to ‘AJNI’, ‘KAROLBAGH’ Or ‘M.G.ROAD’ and Account No is less than 104 from deposit table.   **Part B:**   1. Display all the details of first five customers from deposit table. 2. Display all the details of first three depositors whose amount is greater than 1000. 3. Display Loan No, Customer Name of first five borrowers whose branch name does not belongs to ‘ANDHERI’ from borrow table. 4. Retrieve all unique cities using DISTINCT. **(**Use **Customers Table)** 5. Retrieve all unique branches using DISTINCT. **(**Use **Branch Table)**   **Part C:**   1. Retrieve top 50% record from table BORROW. 2. Display top 10% amount from table DEPOSIT. 3. Display top 25% customer who deposited more than 5000. 4. Retrieve first 10% Loan Amounts in which their descending order. 5. Retrieve all unique customer names with city. 6. Retrieve all Loan records with one more column in Loan Amount as 10% extra amount. 7. Retrieve all odd/even records from Borrow table. |
| **Lab 3** | **Perform SQL queries for Select into and Update** |
|  | **Select into Operation**  **Part-A:**  **Create table as per following.**   |  |  |  | | --- | --- | --- | | **Cricket** | | | | Name | City | Age | | Sachin Tendulkar | Mumbai | 30 | | Rahul Dravid | Bombay | 35 | | M. S. Dhoni | Jharkhand | 31 | | Suresh Raina | Gujarat | 30 |  1. Create table Worldcup from cricket with all the columns and data. 2. Create table T20 from cricket with first two columns with no data. 3. Create table IPL From Cricket with No Data   **Part-B:**  **Create table as per following.**   |  |  |  | | --- | --- | --- | | **Employee** | | | | Name | City | Age | | Jay Patel | Rajkot | 30 | | Rahul Dave | Baroda | 35 | | Jeet Patel | Surat | 31 | | Vijay Raval | Rajkot | 30 |  1. Create table Employee\_detail from Employee with all the columns and data. 2. Create table Employee\_data from Employee with first two columns with no data. 3. Create table Employee\_info from Employee with no Data   **Part-C:**  **Perform following queries on Employee table.**   1. Insert the Data into Employee\_info from Employee whose CITY is Rajkot 2. Insert the Data into Employee\_info from Employee whose age is more than 32.   **Update Operation**  **Part-A:**  **From the above given tables perform the following queries (UPDATE Operation):**   1. Update deposit amount of all customers from 3000 to 5000. (Use **Deposit** **Table**) 2. Change branch name of ANIL from VRCE to C.G. ROAD. (Use **Borrow Table**) 3. Update Account No of SANDIP to 111 & Amount to 5000. (Use **Deposit** **Table**) 4. Update amount of KRANTI to 7000. (Use **Deposit** **Table**) 5. Update branch name from ANDHERI to ANDHERI WEST. (Use **Branch Table**) 6. Update branch name of MEHUL to NEHRU PALACE. (Use **Deposit** **Table**) 7. Update deposit amount of all depositors to 5000 whose account no between 103 & 107. (Use **Deposit** **Table**) 8. Update ADATE of ANIL to 1-4-95. (Use **Deposit** **Table**) 9. Update the amount of MINU to 10000. (Use **Deposit** **Table**) 10. Update deposit amount of PRAMOD to 5000 and ADATE to 1-4-96 (Use **Deposit** **Table**)   **Part-B:**   1. Give 10% Increment in Loan Amount. 2. Customer deposits additional 20% amount to their account, update the same.   **Part-C:**   1. Update amount of loan no 321 to *NULL.* 2. Update branch name of KRANTI to NULL (Use **Borrow Table**) 3. Display the name of borrowers whose amount is *NULL*. 4. Display the Borrowers whose having branch. 5. Update the Loan Amount to 5000, Branch to VRCE & Customer Name to Darshan whose loan no is 481. 6. Update the Deposit table and set the date to 01-01-2021 for all the borrowers which amount is less than 2000. 7. Update the date to NULL & Branch name to ‘ANDHERI whose Account No is 110. |
| **Lab 4** | **Perform SQL queries for Alter, Delete, Truncate, and Drop** |
|  | **Alter Operation**  **Part A:**  **Use Deposit table of lab-1.**   |  |  | | --- | --- | | **DEPOSIT** | | | **Column\_Name** | **DataType** | | ACTNO | INT | | CNAME | VARCHAR(50) | | BNAME | VARCHAR(50) | | AMOUNT | DECIMAL(8,2) | | ADATE | DATETIME |   **From the above given tables perform the following queries (ALTER Operation):**   1. Add two more columns City VARCHAR (20) and Pincode INT. 2. Change the size of CNAME column from VARCHAR (50) to VARCHAR (35). 3. Change the data type DECIMAL to INT in amount Column. 4. Rename Column ActNo to ANO. 5. Delete Column City from the DEPOSIT table. 6. Change name of table DEPOSIT to DEPOSIT\_DETAIL.   **Part B:**   1. Rename Column ADATE to AOPENDATE OF DEPOSIT\_DETAIL table. 2. Delete Column AOPENDATE from the DEPOSIT\_DETAIL table. 3. Rename Column CNAME to CustomerName.   **Part C:**  **Create following table using query according to the definition.**   |  |  | | --- | --- | | **Student\_Detail** | | | **Column\_Name** | **DataType** | | Enrollment\_No | VARCHAR(20) | | Name | VARCHAR(25) | | CPI | DECIMAL(5,2) | | Birthdate | DATETIME |   **From the above given tables perform the following queries (ALTER Operation):**   1. Add two more columns City VARCHAR (20) (Not null) and Backlog INT (Null). 2. Change the size of NAME column of student from VARCHAR (25) to VARCHAR (35). 3. Change the data type DECIMAL to INT in CPI Column. 4. Rename Column Enrollment\_No to ENO. 5. Delete Column City from the STUDENT table. 6. Change name of table STUDENT to STUDENT\_MASTER.   **DELETE, Truncate, Drop Operation**  **Part A:**  **Use Deposit\_Detail table (Altered table of DEPOSIT)**   |  |  | | --- | --- | | **DEPOSIT\_DETAIL** | | | **Column\_Name** | **DataType** | | ANO | INT | | CustomerName | VARCHAR(35) | | BNAME | VARCHAR(50) | | AMOUNT | INT | | PINCODE | INT |  1. Delete all the records of DEPOSIT\_DETAIL table having amount greater than and equals to 4000. 2. Delete all the accounts CHANDI BRANCH. 3. Delete all the accounts having account number (ANO) is greater than 105. 4. Delete all the records of Deposit\_Detail table. (Use **Truncate**) 5. Remove Deposit\_Detail table. (Use **Drop**)   **Part B:**  **Create following table using query according to the definition.**   |  |  | | --- | --- | | **Employee\_Master** | | | **Column\_Name** | **DataType** | | EmpNo | INT | | EmpName | VARCHAR(25) | | JoiningDate | DATETIME | | Salary | DECIMAL (8,2) | | City | VARCHAR(20) |   **Insert the following records in the Employee\_Master table.**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **EmpNo** | **EmpName** | **JoiningDate** | **Salary** | **City** | | 101 | Keyur | 5-1-02 | 12000.00 | Rajkot | | 102 | Hardik | 15-2-04 | 14000.00 | Ahmedabad | | 103 | Kajal | 14-3-06 | 15000.00 | Baroda | | 104 | Bhoomi | 23-6-05 | 12500.00 | Ahmedabad | | 102 | Harmit | 15-2-04 | 14000.00 | Rajkot |   **From the above given tables perform the following queries (DELETE Operation):**   1. Delete all the records of Employee\_MASTER table having salary greater than and equals to 14000. 2. Delete all the Employees who belongs to ‘RAJKOT’ city. 3. Delete all the Employees who joined after 1-1-2007. 4. Delete the records of Employees whose joining date is null and Name is not null. 5. Delete the records of Employees whose salary is 50% of 20000. 6. Delete the records of Employees whose City Name is not empty. 7. Delete all the records of Employee\_MASTER table. (Use **Truncate**) 8. Remove Employee\_MASTER table. (Use **Drop**)   **Part C:**   1. Summarize Delete, Truncate and Drop |
| **Lab 5** | **Perform SQL queries for Like operator** |
|  | **Part-A:**  **Create following table using query according to the definition.**   |  |  | | --- | --- | | **Student** | | | **Column\_Name** | **DataType** | | StuID | INT | | FirstName | VARCHAR(25) | | LastName | VARCHAR(25) | | Website | VARCHAR(50) | | City | VARCHAR(25) |   **Insert the following records in the Student table.**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **StuID** | **FirstName** | **LastName** | **Website** | **City** | **Address** | | 1011 | Keyur | Patel | techonthenet.com | Rajkot | A-303 ‘Vasant Kunj’, Rajkot | | 1022 | Hardik | Shah | digminecraft.com | Ahmedabad | “Ram Krupa”, Raiya Road | | 1033 | Kajal | Trivedi | bigactivities.com | Baroda | Raj bhavan plot, near garden | | 1044 | Bhoomi | Gajera | checkyourmath.com | Ahmedabad | “Jig’s Home”, Narol | | 1055 | Harmit | Mitel | @me.darshan.com | Rajkot | B-55, Raj Residency | | 1066 | Ashok | Jani | NULL | Baroda | A502, Club House Building |   **From the above given tables perform the following queries (LIKE Operation):**   1. Display the name of students whose name starts with ‘k’. 2. Display the name of students whose name consists of five characters. 3. Retrieve the first name & last name of students whose city name ends with a & contains six characters. 4. Display all the students whose last name ends with ‘tel’. 5. Display all the students whose first name starts with ‘ha’ & ends with‘t’. 6. Display all the students whose first name starts with ‘k’ and third character is ‘y’. 7. Display the name of students having no website and name consists of five characters. 8. Display all the students whose last name consist of ‘jer’. 9. Display all the students whose city name starts with either ‘r’ or ‘b’. 10. Display all the name students having websites. 11. Display all the students whose name starts from alphabet A to H. 12. Display all the students whose name’s second character is vowel. 13. Display the name of students having no website and name consists of minimum five characters. 14. Display all the students whose last name starts with ‘Pat’. 15. Display all the students whose city name does not starts with ‘b’.   **Part-B:**   1. Display all the students whose name starts from alphabet A or H. 2. Display all the students whose name’s second character is vowel and of and start with H. 3. Display all the students whose last name does not ends with ‘a’. 4. Display all the students whose first name starts with consonant. 5. Display all the students whose website contains .net   **Part-C:**   1. Display all the students whose address consist of -. 2. Display all the students whose address contains single quote or double quote. 3. Display all the students whose website contains @ name starts with j. 4. Display all the names whose are either four or five characters. |
| **Lab 6** | **Implement SQL In-built functions (Math, String, and Date Functions)** |
|  | **Math functions**  **Part-A:**   1. Display the result of 5 multiply by 30. 2. Find out the absolute value of -25, 25, -50 and 50. 3. Find smallest integer value that is greater than or equal to 25.2, 25.7 and -25.2. 4. Find largest integer value that is smaller than or equal to 25.2, 25.7 and -25.2. 5. Find out remainder of 5 divided 2 and 5 divided by 3. 6. Find out value of 3 raised to 2nd power and 4 raised 3rd power. 7. Find out the square root of 25, 30 and 50. 8. Find out the square of 5, 15, and 25. 9. Find out the value of PI. 10. Find out round value of 157.732 for 2, 0 and -2 decimal points. 11. Find out exponential value of 2 and 3. 12. Find out logarithm having base e of 10 and 2. 13. Find out logarithm having base b having value 10 of 5 and 100. 14. Find sine, cosine and tangent of 3.1415. 15. Find sign of -25, 0 and 25. 16. Generate random number using function.   **Part-B:**  **Create and Insert the following records in the Emp\_Master table.**   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **EmpNo** | **EmpName** | **JoiningDate** | **Salary** | **Commission** | **City** | **Dept Code** | | 101 | Keyur | 5-1-02 | 12000.00 | 4500 | Rajkot | 3@g | | 102 | Hardik | 15-2-04 | 14000.00 | 2500 | Ahmedabad | 3@ | | 103 | Kajal | 14-3-06 | 15000.00 | 3000 | Baroda | 3-GD | | 104 | Bhoomi | 23-6-05 | 12500.00 | 1000 | Ahmedabad | 1A3D | | 102 | Harmit | 15-2-04 | 14000.00 | 2000 | Rajkot | 312A |  1. Display the result of Salary plus Commission. 2. Find smallest integer value that is greater than or equal to 55.2, 35.7 and -55.2. 3. Find largest integer value that is smaller than or equal to 55.2, 35.7 and -55.2. 4. Find out remainder of 55 divided 2 and 55 divided by 3. 5. Find out value of 23 raised to 2nd power and 14 raised 3rd power.   **Part-C:**   1. Find out the square root of 36, 49 and 81. 2. Find out the square of 3, 9, and 12. 3. Find out round value of 280.8952 for 2, 0 and -2 decimal points. 4. Find sine, cosine and tangent of 4.2014. 5. Find sign of -55, 0 and 95.   **String functions**  **Part-A:**   1. Find the length of following. (I) NULL (II) ‘ hello ’ (III) Blank 2. Display your name in lower & upper case. 3. Display first three characters of your name. 4. Display 3rd to 10th character of your name. 5. Write a query to convert ‘abc123efg’ to ‘abcXYZefg’ & ‘abcabcabc’ to ‘ab5ab5ab5’ using REPLACE. 6. Write a query to display ASCII code for ‘a’,’A’,’z’,’Z’, 0, 9. 7. Write a query to display character based on number 97, 65,122,90,48,57. 8. Write a query to remove spaces from left of a given string ‘ hello world ‘. 9. Write a query to remove spaces from right of a given string ‘ hello world ‘. 10. Write a query to display first 4 & Last 5 characters of ‘SQL Server’. 11. Write a query to convert a string ‘1234.56’ to number (Use cast and convert function). 12. Write a query to convert a float 10.58 to integer (Use cast and convert function). 13. Put 10 space before your name using function. 14. Combine two strings using + sign as well as CONCAT (). 15. Find reverse of “Darshan”. 16. Repeat your name 3 times.   **Part-B: Perform following queries on Student table of practical no 5.**   1. Find the length of FirstName and LastName columns. 2. Display FirstName and LastName columns in lower & upper case. 3. Display first three characters of FirstName column. 4. Display 3rd to 10th character of Website column. 5. Write a query to display first 4 & Last 5 characters of Website column.   **Part-C: Perform following queries on Student table of practical no 5.**   1. Put 10 space before FirstName using function. 2. Combine FirstName and LastName columns using + sign as well as CONCAT (). 3. Combine all columns using + sign as well as CONCAT (). 4. Find reverse of FirstName column. 5. Repeat FirstName column 3 times 6. Give the Names which contains 5 characters. 7. Combile the result as <EmpName> Lives in <City>. 8. Combine the result as <EmpName> receives <commission> per month. 9. Separate once column of Dept Code after value 3.   **Date Functions**  **Part-A:**   1. Write a query to display the current date & time. Label the column Today\_Date. 2. Write a query to find new date after 365 day with reference to today. 3. Display the current date in a format that appears as may 5 1994 12:00AM. 4. Display the current date in a format that appears as 03 Jan 1995. 5. Display the current date in a format that appears as Jan 04, 96. 6. Write a query to find out total number of months between 31-Dec-08 and 31-Mar-09. 7. Write a query to find out total number of years between 25-Jan-12 and 14-Sep-10. 8. Write a query to find out total number of hours between 25-Jan-12 7:00 and 26-Jan-12 10:30. 9. Write a query to extract Day, Month, Year from given date 12-May-16. 10. Write a query that adds 5 years to current date. 11. Write a query to subtract 2 months from current date. 12. Extract month from current date using datename () and datepart () function. 13. Write a query to find out last date of current month. 14. Calculate your age in years and months.   **Part-B:**  **Create a table Emp\_detail and insert the following records in the table.**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **EmpNo** | **EmpName** | **JoiningDate** | **Salary** | **City** | | 101 | Keyur | 15-1-02 | 12000.00 | Rajkot | | 102 | Hardik | 15-2-04 | 14000.00 | Ahmedabad | | 103 | Kajal | 14-3-06 | 15000.00 | Baroda | | 104 | Bhoomi | 23-6-05 | 12500.00 | Ahmedabad | | 102 | Harmit | 15-2-04 | 14000.00 | Rajkot | | 105 | Jay | 12-3-07 | 12000.00 | Surat |  1. Write a query to find new date after 365 day with reference to JoiningDate. 2. Display the JoiningDate in a format that appears as may 5 1994 12:00AM. 3. Display the JoiningDate in a format that appears as 03 Jan 1995. 4. Display the JoiningDate in a format that appears as Jan 04, 96. 5. Write a query to find out total number of months between JoiningDate and 31-Mar-09. 6. Write a query to find out total number of years between JoiningDate and 14-Sep-10.   **Part-C:**   1. Write a query to extract Day, Month, Year from JoiningDate. 2. Write a query that adds 5 years to JoiningDate. 3. Write a query to subtract 2 months from JoiningDate. 4. Extract month from JoiningDate using datename () and datepart () function. 5. Calculate your age in years and months |
| **Lab 7** | **Perform SQL queries for Aggerate function and group by (without having)** |
|  | **Part-A:**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Emp** | | | | | | | EID | EName | Department | Salary | JoiningDate | City | | 101 | Rahul | Admin | 56000 | 1-Jan-90 | Rajkot | | 102 | Hardik | IT | 18000 | 25-Sep-90 | Ahmedabad | | 103 | Bhavin | HR | 25000 | 14-May-91 | Baroda | | 104 | Bhoomi | Admin | 39000 | 8-Feb-91 | Rajkot | | 105 | Rohit | IT | 17000 | 23-Jul-90 | Jamnagar | | 106 | Priya | IT | 9000 | 18-Oct-90 | Ahmedabad | | 107 | Bhoomi | HR | 34000 | 25-Dec-91 | Rajkot |   **Create table and inset records as per below.**   1. Display the Highest, Lowest, Label the columns Maximum, Minimum respectively. 2. Display Total, and Average salary of all employees. Label the columns Total\_Sal and Average\_Sal, respectively. 3. Find total number of employees of EMPLOYEE table. 4. Find highest salary from Rajkot city. 5. Give maximum salary from IT department. 6. Count employee whose joining date is after 8-feb-91. 7. Display average salary of Admin department. 8. Display total salary of HR department. 9. Count total number of cities of employee without duplication. 10. Count unique departments. 11. Give minimum salary of employee who belongs to Ahmedabad. 12. Find city wise highest salary. 13. Find department wise lowest salary. 14. Display city with the total number of employees belonging to each city. 15. Give total salary of each department of EMP table. 16. Give average salary of each department of EMP table without displaying the respective department name.   **Part-B:**   1. Count the number of employees living in Rajkot. 2. Display the difference between the highest and lowest salaries. Label the column DIFFERENCE. 3. Display the total number of employees hired before 1st January, 1991.   **Part-C:**   1. Count the number of employees living in Rajkot or Baroda. 2. Display the total number of employees hired before 1st January, 1991 in IT department. 3. Find the Joining Date wise Total Salaries. 4. Find the Maximum salary department & city wise in which city name starts with ‘R’. 5. Find out city wise duplicate names. 6. Find out department wise highest paid employee name. |
| **Lab 8** | **Perform SQL queries for Group by with having and Order by** |
|  | |  |  |  |  | | --- | --- | --- | --- | | Region | Product | Sales\_Amount | Year | | North America | Watch | 1500 | 2023 | | Europe | Mobile | 1200 | 2023 | | Asia | Watch | 1800 | 2023 | | North America | TV | 900 | 2024 | | Europe | Watch | 2000 | 2024 | | Asia | Mobile | 1000 | 2024 | | North America | Mobile | 1600 | 2023 | | Europe | TV | 1500 | 2023 | | Asia | TV | 1100 | 2024 | | North America | Watch | 1700 | 2024 |   **Table: Sales\_Data**    **Part-A:**   1. Display Total Sales Amount by Region. 2. Display Average Sales Amount by Product 3. Display Maximum Sales Amount by Year 4. Display Minimum Sales Amount by Region and Year 5. Count of Products Sold by Region 6. Display Sales Amount by Year and Product 7. Display Regions with Total Sales Greater Than 5000 8. Display Products with Average Sales Less Than 10000 9. Display Years with Maximum Sales Exceeding 500 10. Display Regions with at Least 3 Distinct Products Sold. 11. Display Years with Minimum Sales Less Than 1000 12. Display Total Sales Amount by Region for Year 2023, Sorted by Total Amount   **Part-B:**   1. Display Count of Orders by Year and Region, Sorted by Year and Region 2. Display Regions with Maximum Sales Amount Exceeding 1000 in Any Year, Sorted by Region 3. Display Years with Total Sales Amount Less Than 1000, Sorted by Year Descending 4. Display Top 3 Regions by Total Sales Amount in Year 2024   **Part-C:**   1. Display Products with Average Sales Amount Between 1000 and 2000, Ordered by Product Name 2. Display Years with More Than 5 Orders from Each Region 3. Display Regions with Average Sales Amount Above 1500 in Year 2023 sort by amount in descending. |
| **Lab 9** | **Perform SQL queries for Set operator and, Subqueries** |
|  | **Sub Queries**   |  |  | | --- | --- | | **Department** | | | DID | DName | | 10 | Computer | | 20 | Electrical | | 30 | Mechanical | | 40 | Civil |  |  |  |  | | --- | --- | --- | | **Academic** | | | | Rno | SPI | Bklog | | 101 | 8.8 | 0 | | 102 | 9.2 | 2 | | 103 | 7.6 | 1 | | 104 | 8.2 | 4 | | 105 | 7.0 | 2 | | 106 | 8.9 | 3 |  |  |  |  |  | | --- | --- | --- | --- | | **Stu\_Detail** | | | | | Rno | Name | City | DID | | 101 | Raju | Rajkot | 10 | | 102 | Amit | Ahmedabad | 20 | | 103 | Sanjay | Baroda | 40 | | 104 | Neha | Rajkot | 20 | | 105 | Meera | Ahmedabad | 30 | | 106 | Mahesh | Baroda | 10 |   **Part-A:**   1. Display details of students who are from computer department. 2. Displays name of students whose SPI is more than 8. 3. Display details of students of computer department who belongs to Rajkot city. 4. Find total number of students of electrical department. 5. Display name of student who is having maximum SPI. 6. Display details of students having more than 1 backlog.   **Part-B:**   1. Display name of students who are either from computer department or from mechanical department. 2. Display name of students who are in same department as 102 studying in.   **Part-C:**   1. Display name of students whose SPI is more than 9 and who is from electrical department. 2. Display name of student who is having second highest SPI. 3. Display city names whose students branch wise SPI is 9.2.   **SET Operators**  **Part-A:**  **Create below two tables as per following data.**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Computer** | |  | **Electrical** | | | RollNo | Name |  | RollNo | Name | | 101 | Ajay |  | 105 | Ajay | | 109 | Haresh |  | 107 | Mahesh | | 115 | Manish |  | 115 | Manish |  1. Display name of students who is either in Computer or in Electrical. 2. Display name of students who is either in Computer or in Electrical including duplicate data. 3. Display name of students who is in both Computer and Electrical. 4. Display name of students who are in Computer but not in Electrical. 5. Display name of students who are in Electrical but not in Computer. 6. Display all the details of students who are either in Computer or in Electrical. 7. Display all the details of students who are in both Computer and Electrical.   **Part-B:**  **Create below two tables as per following data.**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Emp\_DATA** | |  | **Customer** | | | EID | Name |  | CID | Name | | 1 | Ajay |  | 5 | Ajay | | 9 | Haresh |  | 7 | Mahesh | | 5 | Manish |  | 5 | Manish |  1. Display name of persons who is either Employee or Customer. 2. Display name of persons who is either Employee or Customer including duplicate data. 3. Display name of persons who is both Employee as well as Customer. 4. Display name of persons who are Employee but not Customer. 5. Display name of persons who are Customer but not Employee.   **Part-C:**   1. Perform all the queries of Part-B but display ID and Name columns instead of Name only. |
| **Lab 10** | **Implement SQL View** |
|  | **Part-A:**  **Views (First create a view then display all views)**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Student\_INFO** | | | | | | RNo | Name | Branch | SPI | Bklog | | 101 | Raju | CE | 8.80 | 0 | | 102 | Amit | CE | 2.20 | 3 | | 103 | Sanjay | ME | 1.50 | 6 | | 104 | Neha | EC | 7.65 | 1 | | 105 | Meera | EE | 5.52 | 2 | | 106 | Mahesh | EC | 4.50 | 3 |  1. Create a view Personal with all columns. 2. Create a view Student\_Details having columns Name, Branch & SPI. 3. Create a view AcademicData having columns RNo, Name, Branch. 4. Create a view Student\_ bklog having all columns but students whose bklog more than 2. 5. Create a view Student\_Pattern having RNo, Name & Branch columns in which Name consists of four letters. 6. Insert a new record to AcademicData view. (107, Meet, ME) 7. Update the branch of Amit from CE to ME in Student\_Details view. 8. Delete a student whose roll number is 104 from AcademicData view.   **Part-B:**   1. Create a view that displays information of all students whose spi is above 8.5. 2. Create a view that displays 0 backlog students. 3. Create a view Computerview that displays CE branch data only.   **Part – C**   1. Create a view Result\_EC that displays the name and SPI of students with SPI less than 5 of branch EC. 2. Update the result of student MAHESH to 4.90 in Result\_EC view. 3. Create a view Stu\_Bklog with RNo, Name and Bklog columns in which name starts with ‘M’ and having bklogs more than 5. 4. Drop Computerview form the database. |
| **Lab 11** | **Implement SQL Joins** |
|  | **JOINS (Create below tables as per following data)**   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Stu\_Info | | |  | Result | |  | Employee\_Data | | | | Rno(PK) | Name | Branch | Rno(FK) | SPI |  | EmployeeNo | Name | ManagerNo | | 101 | Raju | CE | 101 | 8.8 |  | E01 | Tarun | NULL | | 102 | Amit | CE | 102 | 9.2 |  | E02 | Rohan | E02 | | 103 | Sanjay | ME | 103 | 7.6 |  | E03 | Priya | E01 | | 104 | Neha | EC | 104 | 8.2 |  | E04 | Milan | E03 | | 105 | Meera | EE | 105 | 7.0 |  | E05 | Jay | E01 | | 106 | Mahesh | ME |  | 107 | 8.9 |  | E06 | Anjana | E04 |   **Part-A:**   1. Combine information from student and result table using cross join or Cartesian product. 2. Perform inner join on Student and Result tables. 3. Perform the left outer join on Student and Result tables. 4. Perform the right outer join on Student and Result tables. 5. Perform the full outer join on Student and Result tables. 6. Display Rno, Name, Branch and SPI of all students. 7. Display Rno, Name, Branch and SPI of CE branch’s student only. 8. Display Rno, Name, Branch and SPI of other than EC branch’s student only. 9. Display average result of each branch. 10. Display average result of CE and ME branch.   **Part-B:**   1. Display average result of each branch and sort them in ascending order by SPI. 2. Display highest SPI from each branch and sort them in descending order.   **Part-C:**   1. Retrieve the names of employee along with their manager’s name from the Employee table. |
| **Lab 12** | **Implement Complex Joins** |
|  | Create Database with Name: Person\_Info  Create following table under Person\_Info database. (Using Design Mode)   |  |  |  | | --- | --- | --- | | Person | | | | Column\_Name | DataType | Constraints | | PersonID | Int | Primary Key | | PersonName | Varchar (100) | Not Null | | DepartmentID | Int | Foreign Key, Null | | Salary | Decimal (8,2) | Not Null | | JoiningDate | Datetime | Not Null | | City | Varchar (100) | Not Null |  |  |  |  | | --- | --- | --- | | Dept | | | | Column\_Name | DataType | Constraints | | DepartmentID | Int | Primary Key | | DepartmentName | Varchar (100) | Not Null, Unique | | DepartmentCode | Varchar (50) | Not Null, Unique | | Location | Varchar (50) | Not Null |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | PersonID | PersonName | DepartmentID | Salary | JoiningDate | City | | 101 | Rahul Tripathi | 2 | 56000 | 01-01-2000 | Rajkot | | 102 | Hardik Pandya | 3 | 18000 | 25-09-2001 | Ahmedabad | | 103 | Bhavin Kanani | 4 | 25000 | 14-05-2000 | Baroda | | 104 | Bhoomi Vaishnav | 1 | 39000 | 08-02-2005 | Rajkot | | 105 | Rohit Topiya | 2 | 17000 | 23-07-2001 | Jamnagar | | 106 | Priya Menpara | NULL | 9000 | 18-10-2000 | Ahmedabad | | 107 | Neha Sharma | 2 | 34000 | 25-12-2002 | Rajkot | | 108 | Nayan Goswami | 3 | 25000 | 01-07-2001 | Rajkot | | 109 | Mehul Bhundiya | 4 | 13500 | 09-01-2005 | Baroda | | 110 | Mohit Maru | 5 | 14000 | 25-05-2000 | Jamnagar |  |  |  |  |  | | --- | --- | --- | --- | | DepartmentID | DepartmentName | DepartmentCode | Location | | 1 | Admin | Adm | A-Block | | 2 | Computer | CE | C-Block | | 3 | Civil | CI | G-Block | | 4 | Electrical | EE | E-Block | | 5 | Mechanical | ME | B-Block |   **From the above given table perform the following queries:**  **Part – A**  1. Find all persons with their department name & code.  2. Find the person's name whose department is in C-Block.  3. Retrieve person name, salary & department name who belongs to Jamnagar city.  4. Retrieve person name, salary & department name who does not belong to Rajkot city.  5. Retrieve person’s name of the person who joined the Civil department after 1-Aug-2001.   1. Find details of all persons who belong to the computer department. 2. Display all the person's name with the department whose joining date difference with the current date is more than 365 days. 3. Find department wise person counts. 4. Give department wise maximum & minimum salary with department name. 5. Find city wise total, average, maximum and minimum salary. 6. Find the average salary of a person who belongs to Ahmedabad city. 7. Produce Output Like: <PersonName> lives in <City> and works in <DepartmentName> Department. (In single column)   **Part – B**   1. Produce Output Like: <PersonName> earns <Salary> from <DepartmentName> department monthly. (In single column) 2. Find city & department wise total, average & maximum salaries. 3. Find all persons who do not belong to any department. 4. Find all departments whose total salary is exceeding 100000.   **Part – C**   1. List all departments who have no person. 2. List out department names in which more than two persons are working. 3. Give a 10% increment in the computer department employee’s salary. (Use Update) |
| **Lab 13** | **Implement Advanced level Joins** |
|  | |  |  |  | | --- | --- | --- | | **Village** | | | | VID  **(Primary Key)** | Name  **(Not Null)** | CityID  **(Foreign Key)** | | 101 | Raiya | 1 | | 102 | Madhapar | 1 | | 103 | Dodka | 3 | | 104 | Falla | 4 | | 105 | Bhesan | 5 | | 106 | Dhoraji | 5 |   **Part-A: Create table as per following data.**   |  |  |  |  | | --- | --- | --- | --- | | **City** | | | | | CityID  **(Primary Key)** | Name  **(Unique Key)** | Pincode **(Not Null)** | Remakrs | | 1 | Rajkot | 360005 | Good | | 2 | Surat | 335009 | Very Good | | 3 | Baroda | 390001 | Awesome | | 4 | Jamnagar | 361003 | Smart | | 5 | Junagadh | 362229 | Historic | | 6 | Morvi | 363641 | Ceramic |  1. Display all the villages of Rajkot city. 2. Display city along with their villages & pin code. 3. Display the city having more than one village. 4. Display the city having no village. 5. Count the total number of villages in each city. 6. Count the number of cities having more than one village.   **Create below table with following constraints**   1. Do not allow SPI more than 10 2. Do not allow Bklog less than 0. 3. Enter the default value as ‘General’ in branch to all new records IF no other value is specified.  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Stu\_Master** | | | | | | Rno(PK) | Name | Branch | SPI | Bklog | | 101 | Raju | CE | 8.80 | 0 | | 102 | Amit | CE | 2.20 | 3 | | 103 | Sanjay | ME | 1.50 | 6 | | 104 | Neha | EC | 7.65 | 0 | | 105 | Meera | EE | 5.52 | 2 | | 106 | Mahesh |  | 4.50 | 3 |  1. Try to update SPI of Raju from 8.80 to 12. 2. Try to update Bklog of Neha from 0 to -1.   **Part-B: Create table as per following schema with proper validation and try to insert data which violate your validation.**   1. Emp(Eid, Ename, Did, Cid, Salary, Experience)   Dept(Did, Dname)  City(Cid, Cname)  **Part-C: Create table as per following schema with proper validation and try to insert data which violate your validation.**   1. Emp(Eid, Ename, Did, Cid, Salary, Experience)   Dept(Did, Dname)  City(Cid, Cname, Did)  District(Did, Dname, Sid)  State(Sid, Sname, Cid)  Country(Cid, Cname)   1. Insert 5 records in each table. 2. Display employeename, departmentname, Salary, Experience, City, District, State and country of all employees. |