Sr.	Practical
Lab-1	**Introduction to Database, Database tools/editors, SQL and Components of SQL** 1. Why Database? Advantages of Database. 2. Different types of Databases & tools/editors available for it. 3. What is SQL? Components of SQL (DDL, DML, DCL, DQL, TCL).
Lab-2	**Introduction to SQL Server Management Studio, Database, Table, Fields, Records, Data types in SQL Server** 1. Introduction to Editor (SQL Server Management Studio). 2. Introduction to Database, Table, Field, Row, Record. 3. Introduction to various data types in SQL: INT, CHAR, VARCHAR, DATETIME, BIT, DECIMAL.

Lab-3

Implementation of the CREATE and INSERT commands in SQL

Create a database with the name: **YourName_EnrolmentNo**

Create the following tables under your database and insert the data using Query Mode.

DEPOSIT

Column_Name	DataType
ACTNO	INT
CNAME	VARCHAR(50)
BNAME	VARCHAR(50)
AMOUNT	DECIMAL(8,2)
ADATE	DATETIME

BRANCH

Column_Name	DataType
BNAME	VARCHAR(50)
CITY	VARCHAR(50)

CUSTOMERS

Column_Name	DataType
CNAME	VARCHAR(50)
CITY	VARCHAR(50)

BORROW

Column_Name	DataType
LOANNO	INT
CNAME	VARCHAR(50)
BNAME	VARCHAR(50)
AMOUNT	DECIMAL(8,2)

DEPOSIT

ACTNO	CNAME	BNAME	AMOUNT	ADATE
101	ANIL	VRCE	1000.00	01-MAR-95
102	SUNIL	AJNI	5000.00	04-JAN-96
103	MEHUL	KAROLBAGH	3500.00	17-NOV-95
104	MADHURI	CHANDI	1200.00	17-DEC-95
105	PRMOD	M.G. ROAD	3000.00	27-MAR-96
106	SANDIP	ANDHERI	2000.00	31-MAR-96
107	SHIVANI	VIRAR	1000.00	05-SEP-95
108	KRANTI	NEHRU PLACE	5000.00	02-JUL-95
109	MINU	POWAI	7000.00	10-AUG-95
110	KARAN	BANDRA	4500.00	23-FEB-94

BRANCH

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

CNAME	CITY
ANIL	CALCUTTA
SUNIL	DELHI
MEHUL	BARODA
MANDAR	PATNA
MADHURI	NAGPUR
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
KRANTI	BOMBAY
NAREN	BOMBAY

BORROW

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	PRMOD	VIRAR	8000.00
481	KRANTI	NEHRU PLACE	3000.00

Lab-4

Implementation of the SELECT command in SQL

From the tables given in Lab-3, perform the following queries using the SELECT command:

Part – A

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 table.
5. Display Loan No, Amount from BORROW table.
6. Display loan details of all customers who belongs to ‘ANDHERI’ branch.
7. Give account no and amount of depositor, whose account no is equals to 106.
8. Give name of borrowers having amount greater than 5000.
9. Give name of customers who opened account after date '01-DEC-96'.
10. Display name of customers whose account no is less than 105.
11. Display name of customer who belongs to either ‘NAGPUR’ Or ‘DELHI’. (**OR & IN**)

	12. Display name of customers with branch whose amount is greater than 4000 and account no is less than 105. 13. Find all borrowers whose amount is greater than equals to 3000 & less than equals to 8000. (**AND & BETWEEN**) 14. Find all depositors who do not belongs to ‘ANDHERI’ branch. 15. Display the name of borrowers whose amount is *NULL*. **Part – B** 16. 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. 17. Display all the details of first five customers. 18. Display all the details of first three depositors whose amount is greater than 1000. 19. Display Loan No, Customer Name of first five borrowers whose branch name does not belongs to ‘ANDHERI’. 20. Retrieve all unique cities using DISTINCT. **(**Use **Customers Table)** **Part – C** 21. Retrieve all unique branches using DISTINCT. **(**Use **Branch Table)** 22. Retrieve all the records of customer table as per their city name in ascending order. 23. Retrieve all the records of deposit table as per their amount column in descending order. 24. Retrieve all the details of customers in descending order of their city name. 25. Show all unique borrowers& label the column Unique_Borrowers. (Display only Names)
Lab-5	**Implementation of the UPDATE command in SQL** **From the tables given in Lab-3, perform the following queries using the UPDATE command:** **Part – A** 1. Update deposit amount of all customers from 3000 to 5000. 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. 4. Give 10% Increment in Loan Amount. 5. Update deposit amount of all depositors to 5000 whose account no between 103 & 107. **Part – B** 6. Update amount of loan no 321 to *NULL.* 7. Change Loan number from 201 to 401 & also change branch name from VRCE to AJNI. 8. Modify customer name ANIL to ANIL JAIN. 9. Give an increment of ₹1000 in the Loan Amount. 10. Update the customer’s name from MINU to SINU and change the amount to ₹10,000. **Part – C** 11. Change Name to Ramesh, Branch Name to VRCE & Amount to 5500, Whose Account Number is 102. 12. Make Branch Name & Amount NULL, Whose Loan Number is 481 & Name is KRANTI. 13. Give 5% Increment in Loan Amount whose loan no less then 321. 14. Update the customer city from BOMBAY to MUMBAI. 15. Update the branch name to VRCE where the Loan number is 375.
Lab-6	**Implementation of DELETE, TRUNCATE and DROP commands in SQL** **From the tables given in Lab-3, perform the following queries using the DELETE, TRUNCATE and DROP commands:** **Part – A** 1. Delete records of Customer table that belongs to BOMBAY City. 2. Delete all account numbers whose amount less than equals to 1000. 3. Delete borrowers whose branch name is ‘AJNI’.

4. Delete all the borrowers whose loan number between 201 to 210.
5. Delete customers who opened account after date '01-DEC-96'. (Use **DEPOSIT** table)

Part – B

6. Delete all the records of Customers table. (Use **TRUNCATE**)
7. Remove all customers whose name is ANIL & Account Number is 101.
8. Delete all the depositors who do not belongs to ‘ANDHERI’ branch.
9. Delete loan details of customers whose amount is less than 3000.
10. Delete customers who opened account before date '01-JAN-96'. (Use **DEPOSIT** table)

Part – C

11. Delete all the borrowers whose loan amount is less than 2000 and does not belong to VRCE branch.
12. Delete all the records of Deposit table. (Use **TRUNCATE**)
13. Delete all the records of Branch table. (Use **TRUNCATE**)
14. Remove Deposit table. (Use **DROP**)
15. Remove Branch table. (Use **DROP**)

Lab-7

Implementation of the LIKE Operator in SQL

Create the following table using Design Mode and insert the data.

Students

Column_Name	DataType
StuID	INT
FirstName	VARCHAR(25)
LastName	VARCHAR(25)
Website	VARCHAR(50)
City	VARCHAR(25)
Division	VARCHAR(20)

STUDENTS

StuID	FirstName	LastName	Website	City	Division
1011	Keyur	Patel	techonthenet.com	Rajkot	II-BCX
1022	Hardik	Shah	digminecraft.com	Ahmedabad	I-BCY
1033	Kajal	Trivedi	bigactivities.com	Baroda	IV-DCX
1044	Bhoomi	Gajera	checkyourmath.com	Ahmedabad	III-DCW
1055	Harmit	Mitel	*NULL*	Rajkot	II-BCY
1066	Ashok	Jani	*NULL*	Baroda	II-BCZ

Perform the following queries on the above-given table:

Part – A

1. Display the name of students whose first name starts with ‘k’.
2. Display the name of students whose first 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 consists of ‘jer’.
9. Display all the students whose city name starts with either ‘r’ or ‘b’.
10. Display all the students name having websites.

Part – B

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 student's name whose city name consist of 'rod'.
14. Retrieve the First & Last Name of students whose website name starts with 'bi'.
15. Display student's city whose last name consists of six characters.

Part – C

16. Display all the students whose city name consist of five characters & not start with 'ba'.
17. Show all the student's whose division starts with 'II'.
18. Find out student's first name whose division contains 'bc' anywhere in division name.
19. Show student id and city name in which division consist of six characters and having website name.
20. Display all the students whose name's third character is consonant.

Lab-8

Implementation of the ALTER and RENAME commands in SQL

Create the following table using Query Mode and perform the following queries.

Student	
Column_Name	**DataType**
Enrollment_No	VARCHAR(20)
Name	VARCHAR(25)
CPI	DECIMAL(5,2)
Birthdate	DATETIME

Part – A

1. Add two more columns City VARCHAR (20) NULL and Backlog INT NOT 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.

Part – B

7. Remove Column Backlog from the table.
8. Change Constraint of Name Column from NULL to NOT NULL.

Part – C

9. Rename Column Birthdate to BDate.
10. Change the datatype of ENO Column from VARCHAR (20) to VARCHAR (12).

Lab-9

Query Test – 1

Lab-10

Implementation of Math Functions in SQL

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.

Part – B

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.

Part – C

11. Find out exponential value of 2 and 3.

12. Find out logarithm having base b having value 10 of 5 and 100.
13. Find sine, cosine and tangent of 3.1415.
14. Find sign of -25, 0 and 25.
15. Generate random number using function.

Lab-11 **Implementation of String Functions in SQL**

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.

Part – B

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()).
12. Write a query to convert a float 10.58 to integer (UseCONVERT()).
13. Put 10 space before your name using function.
14. Combine two strings (Your name & Surname) using + sign as well as CONCAT ().

Part – C

15. Find reverse of “Darshan”.
16. Repeat your name 3 times.
17. Write a query to display first 3 & Last 3 characters of ‘your name’.
18. From Data, returns the first non-null value in a list. (Use COALESCE())
19. Tests whether the expression is numeric. (Use ISNUMERIC())
20. Search for "t" in string "Customer", and return its position. (Use CHARINDEX())

Lab-12 **Implementation of DATE Functions in SQL**

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.

Part – B

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.

Part – C

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. Write a query to display date & time after 30 days from today.
15. Calculate your age in years and months.

Lab-13

Implementation of Integrity Constraints in SQL

Part – A

Create the following table using Query Mode and apply the specified Integrity Constraints.

Student_Info

Column_Name	DataType	Remarks
StudentID	INT	Primary Key, Auto Increment
Enrollment_No	VARCHAR(20)	Unique Key, Not Null
Name	VARCHAR(25)	Not Null
CPI	DECIMAL(5,2)	Do not allow SPI more than 10, Null
JoiningDate	DATETIME	Set Default value getdate(), Not Null
Bklog	INT	Do not allow Bklog less than 0, Not Null
StateName	VARCHAR(50)	Default value as ‘Gujarat’ in StateName to all new records If no other value is Specified

Part – B

Create the following table using Query Mode and apply the specified Integrity Constraints.

State

Column_Name	DataType	Remarks
StateID	INT	Primary Key, Auto Increment
StateName	VARCHAR(20)	Unique Key, Not Null

City

Column_Name	DataType	Remarks
CityID	INT	Primary Key, Auto Increment, Not Null
CityName	VARCHAR(20)	Unique Key, Not Null
PinCode	VARCHAR(10)	Null
StateID	INT	Foreign key

Part – C

Create the tables given in Part-A and Part-B using Design Mode and apply the specified constraints.

Notes: *After creating all the above tables, insert the data and test the applied constraints.*

Lab-14

Practice to implement Integrity Constraints using both Query Mode and Design Mode.

Create the following table and apply the specified Integrity Constraints.

Part – A

ProductInventory

Column_Name	DataType	Remarks
ProductID	INT	Primary Key, Auto Increment
ProductName	VARCHAR(100)	Not Null
CatalogNumber	VARCHAR(20)	Unique Key
UnitPrice	DECIMAL(10,2)	Not Null, UnitPrice >= 0.50 AND UnitPrice <= 5000.00
QuantityInStock	INT	Not Null, QuantityInStock >= 0
IsActive	BIT	Not Null, Sets a default value to 1 if not specified

Part – B

DepartmentData

Column_Name	DataType	Remarks

DeptID	INT	Primary Key
DeptName	VARCHAR(50)	Not Null, Unique Key

EmployeeData

Column_Name	DataType	Remarks
EmpID	INT	Primary Key
EmpName	VARCHAR(50)	Not Null
Salary	DECIMAL(10, 2)	Not Null, Salary > 30000
DeptID	INT	Foreign Key

Part – C

Members

Column_Name	DataType	Remarks
MemberID	INT	Primary Key, Auto Increment
MemberName	VARCHAR(50)	Not Null
Email	VARCHAR(100)	Not Null, Unique Key
Phone	VARCHAR(15)	Unique Key
JoinDate	DATETIME	Not Null, Set Default value getdate()
City	VARCHAR(50)	Not Null, Sets a default value to 'Rajkot'

Books

Column_Name	DataType	Remarks
BookID	INT	Primary Key, Auto Increment (ID starts from 100)
Title	VARCHAR(100)	Not Null
Author	VARCHAR(50)	Not Null
Category	VARCHAR(30)	Not Null
Price	DECIMAL(7,2)	Not Null, Price > 0
Stock	INT	Not Null, Stock >= 0

Issue_Return

Column_Name	DataType	Remarks
IssueID	INT	Primary Key, Auto Increment (ID starts from 500)
MemberID	INT	Not Null, Foreign Key
BookID	INT	Not Null, Foreign Key
IssueDate	DATE	Not Null, Set Default value getdate()
ReturnDate	DATE	Null

Lab-15

Implementation of Aggregate Functions in SQL (Part – 1)
Create STUDENT_MARKS table with SID INT, SName VARCHAR(25) and Marks INT.
Insert the records into the Student_Marks table as given below.

Student_Marks

SID	SName	Marks
1	John	90
2	Martin	80
3	Carol	89
4	Jack	99
5	Rose	88
6	Mary	90

Part – A

1. Find the total number of students.
2. Find the total marks scored by all students.
3. Find the average marks of all students.
4. Find the minimum marks scored by any student.
5. Find the maximum marks scored by any student.

Part – B

Aggregate Functions with GROUP BY and GROUP BY Having Clause

Employee

EID	EName	Department	Salary	JoiningDate	City
101	Rahul	Admin	56000	01-Jan-90	Rajkot
102	Hardik	IT	18000	25-Sep-90	Ahmedabad
103	Bhavin	HR	25000	14-May-91	Baroda
104	Bhoomi	Admin	39000	08-Feb-91	Rajkot
105	Rohit	IT	17000	23-Jul-90	Jamnagar
106	Priya	IT	9000	18-Oct-90	Ahmedabad
107	Neha	HR	34000	25-Dec-91	Rajkot

Perform the following queries on the above-given table:

1. Display the Highest, Lowest, Total, and Average salary of all employees & label the columns Maximum, Minimum, Total_Sal and Average_Sal, respectively.
2. Find total number of employees of EMPLOYEE table.
3. Retrieve maximum salary from IT department.
4. Count total number of cities of employee without duplication.
5. Display city with the total number of employees belonging to each city.
6. Display city having more than one employee.
7. Give total salary of each department of EMPLOYEE table.
8. Give average salary of each department of EMPLOYEE table without displaying the respective department name.
9. Give minimum salary of employee who belongs to Ahmedabad.
10. List the departments having total salaries more than 50000 and located in city Rajkot.

Lab-16

Implementation of Aggregate Functions in SQL (Part – 2)

Create the following table using Design Mode and insert the given data.

Student_Data

SID	SName	Department	SPI	City
101	Jay	MCA	8.8	Rajkot
102	Deep	BCA	5.6	Ahmedabad
103	Bhavin	BBA	8.3	Baroda
104	Dharmik	BCA	9.9	Rajkot
105	Jeet	MCA	10.0	Jamnagar
106	Priya	MBA	5.5	Ahmedabad
107	Neha	BSCIT	6.5	Rajkot

Part – A

1. Display the Highest, Lowest SPI of all students & label the columns Maximum, Minimum respectively.
2. Find total number of students of student table.

3. Retrieve maximum SPI from BCA department.
4. Count total number of cities of students without duplication.
5. Display city with the total number of students belonging to each city.

Part – B

6. Display city having more than one student.
7. Give total SPI of each department of student table.
8. Give average SPI of each department of student table without displaying the respective department name.
9. Give minimum SPI of student who belongs to Ahmedabad.
10. List the departments having total SPI more than 15.

Part – C

11. Display department having more than one student.
12. Find total number of students from MCA department.
13. Retrieve maximum SPI from BBA department.
14. Count total number of department of students table without duplication.
15. List the departments having total number of students more than one.

Lab-17

Practice the basic SQL commands, various built-in functions (Math, String, Date), Aggregate functions and Group by Having clause.
Create the following table using Design Mode and insert the given data.

Employees

Column_Name	DataType
EID	INT
EName	VARCHAR(30)
Gender	VARCHAR(20)
JoiningDate	DATETIME
Salary	DECIMAL(10,2)
City	VARCHAR(20)

EMPLOYEE

EID	EName	Gender	JoiningDate	Salary	City
1	Nick	Male	01-JAN-13	4000	London
2	Julian	Female	01-OCT-14	3000	NewYork
3	Roy	Male	01-JUN-16	3500	London
4	Tom	Male	NULL	4500	London
5	Jerry	Male	01-FEB-13	2800	Sydney
6	Philip	Male	01-JAN-15	7000	NewYork
7	Sara	Female	01-AUG-17	4800	Sydney
8	Emily	Female	01-JAN-15	5500	NewYork
9	Michael	Male	NULL	6500	London
10	John	Male	01-JAN-15	8800	London

Perform the following queries on the above-given table:

1. Display all the employees whose name starts with “m” and 4th character is “h”.
2. Find the value of 3 raised to 5. Label the column as output.
3. Write a query to subtract 20 days from the current date.
4. Write a query to display name of employees whose name starts with “j” and contains “n” in their name.
5. Display 2nd to 9th character of the given string “SQL Programming”.
6. Display name of the employees whose city name ends with “ney” &contains six characters.

7. Write a query to convert value 15 to string.
8. Add department column with varchar(20) to Employees table.
9. Set the value of department to Marketing who belongs to London city.
10. Display all the employees whose name ends with either “n” or “y”.
11. Find smallest integer value that is greater than or equal to 63.1, 63.8 and -63.2.
12. Display all employees whose joining date is not available.
13. Display name of the employees in capital letters and city in small letters.
14. Change the data type of Ename from varchar(30) to char(30).
15. Display city wise maximum salary.
16. Produce output like <Ename> works at <city> and earns <salary>.
17. Find total number of employees whose salary is more than 5000.
18. Write a query to display first 4 & last 3 characters of all the employees.
19. List the city having total salaries more than 15000 and employees joined after 1^{st} January, 2014.
20. Write a query to replace “u” with “oo” in Ename.
21. Display city with average salaries and total number of employees belongs to each city.
22. Display total salaries paid to female employees.
23. Display name of the employees and their experience in years.
24. Remove column department from employees table.
25. Update the value of city from sydney to null.
26. Retrieve all the Employee’s Name, Salary & Joining date.
27. Find out combine length of Ename & Gender.
28. Find the difference between highest & lowest salary.
29. Rename a column from Ename to FirstName.
30. Rename a table from Employees to EmpMaster.

Lab-18

Implementation of SET Operators in SQL
Create the following tables and perform the following queries:

Computer	
RollNo	**Name**
101	Ajay
109	Haresh
115	Manish

Electrical	
RollNo	**Name**
105	Ajay
107	Mahesh
115	Manish

Part – A

1. Display the name of students who is either in Computer or in Electrical.
2. Display the 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.

Create the following tables and perform the following queries:

Employee_A		
EmpID	**EName**	**Department**
101	Amar	Sales
102	Boby	IT
103	Charlie	HR
104	Denish	Sales

Employee_B		
EmpID	**EName**	**Department**
101	Amar	Sales
103	Charlie	HR
105	Isha	Finance
106	Fatima	IT

Part – B

1. Find all unique employees present in either Employees_A OR Employees_B.
2. Find ALL employees from both tables, including duplicates.
3. Find employees who are present in BOTH Employees_A AND Employees_B.
4. Find employees present in Employees_A BUT NOT in Employees_B.
5. Find employees present in Employees_B BUT NOT in Employees_A.

Part – C

6. Find unique employees belonging to the 'IT' department in either table.
7. Find employees who are in 'Sales' AND appear in both tables.
8. List all names from both tables, retaining all duplicates.
9. Find employees in A with EmployeeID > 102, who are NOT in B.

Lab-19

Implementation of Select * Into command in SQL

Create the following table and perform the following queries:

Cricket

Name	City	Age
Sachin Tendulkar	Mumbai	30
Rahul Dravid	Bombay	35
M. S. Dhoni	Jharkhand	31
Suresh Raina	Gujarat	30

Part – A

1. Create table World cup from cricket with all the columns.
2. Create table T20 from cricket with first two columns with no data.
3. Create table IPL From Cricket with No Data.
4. Insert the Data into IPL from Cricket Whose Second Character Should Be ‘A’ And String Should Have At least 7 Characters in Cricket Name Field.
5. Delete All the Rows From IPL.
6. Delete the Detail of Cricketer Whose City is Jharkhand.
7. Rename the Table IPL to IPL2018.
8. Destroy table T20 with All the Data.

Create the following table and perform the following queries:

SalesOrders

Column_Name	DataType
OrderID	INT PRIMARY KEY
ProductID	INT
ProductName	VARCHAR(50)
Qty	INT
SaleDate	DATE
Region	VARCHAR(50)

SALESRECORDS

OrderID	ProductID	ProductName	Qty	SaleDate	Region
1	10	Laptop	25	2025-10-01	North
2	20	Monitor	60	2025-10-05	South
3	30	Keyboard	150	2025-10-10	East
4	10	Laptop	10	2025-10-15	West
5	40	Mouse	45	2025-10-20	North
6	50	Webcam	55	2025-10-25	South

7	60	Speaker	20	2025-10-30	East
8	20	Monitor	70	2025-11-01	West

Part – B

1. Create a full backup of the *SalesRecords* table and name it *SalesRecords_Backup_Full* as a new table.
2. Create a table *SalesInfo* from the *SalesRecords* table with only the OrderID, ProductName, and Quantity columns.
3. Create a new empty table with the same schema as SalesRecords and name it *SalesRecords_Empty*.

Part – C

4. Create a table *Sales_North_West* from the *SalesRecords* table containing only the sales records for the North and West regions.
5. Create a table *Product_Summary* from the *SalesRecords* table that summarizes the total quantity sold for each product.
6. Create an archive table for all sales made before Oct 15, 2025.

Lab-20 **Query Test – 2**

Lab-21 **Implementation of JOIN in SQL**

Create the following tables and perform the following queries:

Student		
Rno	Name	Branch
101	Raju	CE
102	Amit	CE
103	Sanjay	ME
104	Neha	EC
105	Meera	EE
106	Mahesh	ME

Result	
Rno	SPI
101	8.8
102	9.2
103	7.6
104	8.2
105	7.0
107	8.9

Employee		
EmployeeNo	Name	ManagerNo
E01	Tarun	*NULL*
E02	Rohan	E02
E03	Priya	E01
E04	Milan	E03
E05	Jay	E01
E06	Anjana	E04

Part – A

1. Combine information from student and result table using cross join or Cartesian product.
2. Display Rno, Name, Branch and SPI of CE branch’s student only.
3. Display Rno, Name, Branch and SPI of other than EC branch’s student only.
4. Display average result of each branch.
5. Display average result of each branch and sort them in ascending order by SPI.

Part – B

6. Display average result of CE and ME branch.
7. Perform the left outer join on Student and Result tables.
8. Perform the right outer join on Student and Result tables.
9. Perform the full outer join on Student and Result tables.
10. Retrieve the names of employee along with their manager’s name from the Employee table.

Part – C

11. Display maximum result of each branch.
12. Display minimum result of each branch.
13. Display name of student whose result is not generated.
14. Display overall average result.
15. Display Rno, Name, Branch and SPI of those students whose SPI greater than.

Lab-22 **Implementation of advanced JOIN operations in SQL**

Create the following tables and perform the following queries:

City

CityID (Primary Key)	City Name (Unique Key)	Pincode	Remakrs
1	Rajkot	360005	Good
2	Surat	335009	Good
3	Baroda	390001	Awesome
4	Jamnagar	361003	Smart
5	Junagadh	362229	Historic
6	Morvi	363641	Ceramic

Village

VillageID (Primary Key)	Village Name	CityID (Foreign Key)
101	Raiya	1
102	Madhapar	1
103	Dodka	3
104	Falla	4
105	Bhesan	5
106	Dhoraji	5

Part – A

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.

Part – B

6. Display number of villages of Rajkot city.
7. Display Rajkot city along with their villages & pin code.
8. Display villages of Rajkot & Junagadh city.

Part – C

9. Display number of villages of Surat city.
10. Display city along with their villages & pin code whose Remarks is Good.
11. Display the city name of Falla village.

Lab-23 Implementation of Sub Queries in SQL

Create the following tables and perform the following queries:

Student

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

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

Department

DID	DName
10	Computer
20	Electrical
30	Mechanical
40	Civil

Part – A

1. Display details of students who are from computer department.
2. Display 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.

Part – B

6. Display details of students having more than 1 backlog.
7. Display name of student who is having second highest SPI.
8. Display name of students who are either from computer department or from mechanical department.

Part – C

9. Display name of students who are in same department as 102 studying in.
10. Display name of students whose SPI is more than 9 and who is from electrical department.

Lab-24 Implementation of advanced Sub Queries in SQL

Create the following tables and perform the following queries:

Departments

DeptID	DeptName	Location
10	HR	Surat
20	Sales	Rajkot
30	IT	Baroda
40	Finance	Surat
50	Operations	Ahmedabad

Emp_Detail

EmpID	Name	DeptID	Salary	HireDate
101	Ali	10	60000.00	2022-01-15
102	Boby	20	75000.00	2021-05-20
103	Charu	10	62000.00	2023-03-10
104	Danish	30	90000.00	2020-11-01
105	Emran	20	78000.00	2022-08-25
106	Faizal	30	92000.00	2021-07-01
107	Gita	40	85000.00	2023-11-12
108	Hena	50	55000.00	2024-02-01
109	Isha	20	75000.00	2021-05-20
110	Jay	50	58000.00	2024-05-01

Part – A

1. List all employees who earn a salary greater than the average salary of all employees.
2. Find the names of employees who work in the **'Surat'** location.
3. List all employees who do not work in the **'Rajkot'** and **'Ahmedabad'** location.

Part – B

4. List the names of departments that have more than two employees.
5. List all employees who earn the same salary as Boby.
6. List employees hired after the employee named **'Ali'**.

Part – C

7. List employees whose salary is either the highest or the lowest within the 'HR' department.
8. List employees who work in a **'Surat'** location **AND** have a salary **less than the average salary** of all employees.
9. List the name and salary of all employees who earn **more than the minimum salary** paid in the 'IT' department.

Lab-25 **Query Test – 3**

Lab-26 **Practice LAB to implement SELECT, UPDATE, ALTER, RENAME, TRUNCATE, DELETE, DROP, the LIKE Operator**
Create the following table and insert the given data:

DU_Students

Column_Name	DataType
StuID	INT
Name	VARCHAR(20)
EnrollmentNo	INT
Division	VARCHAR(10)
Sem	VARCHAR(100)
BirthDate	DATETIME
Email	VARCHAR(100)
ContactNo	VARCHAR(10)

DU_Students

StuID	Name	EnrollmentNo	Division	Sem	Email	ContactNo
101	Naimish Patel	090200107051	BCX-3	3	naimishp49@gmail.com	8866205253
102	Firoz A. S.	090200107090	BCY-3	3	firoz.me@yahoo.com	8885999922
103	Krunal Vyas	090243107101	BCZ-5	5	krunal.vyas@gmail.com	9990888877
104	Vijay Patel	090200107102	BCX-5	5	vijay.patel123@gmail.com	8787878787
105	Vimal Trivedi	090200107103	BCY-3	3	vimal123@yahoo.com	8789564512

Perform the following queries on the above-given table:

1. Display all the records of DU_Students table.
2. Display students studying in BCY-3.
3. Display 5th semester students.
4. Display contact number of all students.
5. Display name and division of all students.
6. Extract last 3 digits of ContactNo.
7. Display students sorted by Name in ascending order.
8. Display Name of Student who belongs to either Semester 3 or 5. (Use IN & OR)
9. Find Student Name & Enrollment No in which Student Id between 103 to 105.
10. Find Student Name & Enrollment No with their Email Who belongs to 5th Semester.
11. Display All the Details of first three students.
12. Display length of each student name.
13. Update semester to 5 and Contact number to 98984477 whose id is 101.
14. Display Name & Enrollment no of first 30% Students.
15. Display current date with student name.
16. Display Unique Semesters.
17. Display semester 5 student’s division and Enrollment No.
18. Retrieve All the Students who have no Enrollment No.
19. Display students whose email ends with 'gmail.com'.
20. Find Students who born after date 01-01-1990.
21. Modify ContactNo data type to VARCHAR(15).
22. Display students whose name contains 'Patel'.
23. Display student name in uppercase.
24. Rename column Name to StudentName
25. Update Division BCX-3 to BCX-4.
26. Update Division to BCX-5 & Semester to 5 whose Student Id Is 102.
27. Add one more Column City Varchar (50) in DU_Students table.
28. Remove All BCX-4 Division Students.
29. Truncate the table (removes all data).
30. Remove table DU_Students from the database.

Lab-27

Practice Lab to implement Aggregate Functions using GROUP BY and GROUP BY Having clauses

Create the following table and insert the given data:

Employee_Performance	
Column_Name	**DataType**
EmployeeID	INT PRIMARY KEY
Name	VARCHAR(50)
Team	VARCHAR(50)
YearsService	INT
ProjectsCompleted	INT
AnnualBonus	DECIMAL(10, 2)

Employee_Performance					
EmployeeD	**Name**	**Team**	**YearsService**	**ProjectsCompleted**	**AnnualBonus**
101	Ashok	Marketing	5	8	4000.00

102	Bina	Sales	2	5	2500.00
103	Chirag	Marketing	8	12	6000.00
104	Deep	IT	1	3	1000.00
105	Krish	Sales	6	10	5500.00
106	Shruti	IT	4	7	3000.00
107	Bhargav	Sales	3	6	3000.00
108	Rajni	Marketing	10	15	8000.00

Perform the following queries on the above-given table:

1. Calculate the total number of projects completed by each Team.
2. Calculate the average years of service and the maximum annual bonus awarded within each Team.
3. Find the number of employees belonging to each Team.
4. Calculate the total AnnualBonus grouped by both Team and YearsService.
5. Find Teams that have more than two employees.
6. Find Teams whose employees have completed a combined total of more than 20 projects.
7. Find Teams where the average annual bonus is less than 3500.00.
8. Find Teams where the minimum number of projects completed by any single member is less than 5.
9. Find Teams that have at least one employee with more than 8 years of service.
10. Find Teams where the average annual bonus is greater than 3000.00. Order the final result by the average bonus in descending order.
11. Find Teams where the sum of all members' years of service is less than 15 years.
12. Find Teams where the maximum years of service is greater than 5 AND the minimum annual bonus awarded in that team is less than 4000.00.

Lab-28

Practice Lab to implement advanced JOIN operations in SQL
Create the following tables and insert the given data:

StudentAcademicData

Column_Name	DataType	Remarks
StuID	INT	PRIMARY KEY
Name	VARCHAR(50)	Not Null
City	VARCHAR(50)	Not Null

Course

Column_Name	DataType	Remarks
CourseID	INT	PRIMARY KEY
CourseName	VARCHAR(50)	Not Null
Credits	INT	Not Null

Enrolments

Column_Name	DataType	Remarks
Enroll_ID	INT	PRIMARY KEY
StuID	INT	FOREIGN KEY
CourseID	INT	FOREIGN KEY
Marks	INT	Not Null

StudentAcademicData

StuID	Name	City
101	Amit	Rajkot

102	Neha	Surat
103	Rohan	Baroda
104	Meera	Rajkot
105	Jatin	Ahmedabad
106	Kajal	Surat

Course		
CourseID	CourseName	Credits
1	DBMS	4
2	Operating System	3
3	Computer Networks	4
4	Data Structures	3
5	Python Programming	4

Enrolments			
Enroll_ID	StuID	CourseID	Marks
1	101	1	85
2	101	2	78
3	101	5	92
4	102	1	88
5	102	3	75
6	103	2	67
7	103	4	73
8	104	5	95
9	105	1	56
10	105	3	61
11	106	2	82
12	106	4	77

Perform the following queries on the above-given tables:

1. Display student names with the courses they enrolled in.
2. Display students and their marks.
3. Display students who enrolled in DBMS.
4. Display all students and their enrolments (include non-enrolled students).
5. Display courses that have no enrolments.
6. Display students with city and course credits.
7. Display total number of courses each student has enrolled in.
8. Display student names with marks greater than 80.
9. Display students and courses in which credits = 4.
10. Display average marks of each student.
11. Display highest marks obtained in each course.
12. Display students who scored below 60 in any course.
13. Display students and courses from Rajkot city only.
14. Display total marks gained by each student.
15. Display the list of students who have taken at least 3 courses.
16. Display students who have the highest marks in their courses.
17. Display students who scored above the average marks of that course.
18. Display each student’s highest and lowest marks with course names.
19. Display students enrolled in at least one 4-credit course.

	20. Display the total marks of each course and arrange them from highest to lowest.
Lab-29	**Design an E-R Diagram for an enterprise-level database (Part – 1)** 1. Prepare an E-R Diagram for Bank Management System. 2. Prepare an E-R Diagram for Library Management System.
Lab-30	**Design an E-R Diagram for an enterprise-level database (Part – 2)** 1. Prepare an E-R Diagram for Hospital Management System. 2. Prepare an E-R Diagram for College Management System.