Program	Bachelor of Technology (BTech)	**Semester - 3**
Type of Course	-	
Prerequisite		
Course Objective	This course introduces the basic concepts of the database system, its design and implementation, database modelling, relational, and hierarchical and network models. The student will also learn data manipulation language, update and manage a database, database security, integrity, concurrency, storage strategies and hands-on practice using SQL as well as NoSQL concepts.	

Teaching Scheme (Contact Hours)				Examination Scheme				
				Theory Marks		Practical Marks		
Lecture	**Tutorial**	**Practical**	**Credit**	**SEE (T)**	**CIA (T)**	**SEE (P)**	**CIA (P)**	**Total Marks**
4	0	4	6	40	30	20	10	100

***SEE** - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)*

Course Content

T - Teaching Hours | **W** - Weightage

Sr.	Topics	T	W
1	**Advanced SQL concepts** Group by, Joins, Subquery, Keys, System Functions, User Defined Functions (UDF), Stored Procedures, Parameters in SP, Procedures v/s Functions, Cursor, Triggers, Exception Handling, TCL and DCL Commands	15	22
2	**Entity-Relationship Model and Relational Database Design** **E-R Model:** Entities, attribute and its types, relationship, Mapping cardinality, E-R notations, E-R diagram, extended E-R features: specialization, generalization, reduction to relational schemas **Relational Database Design:** Overview of Functional Dependency (FD), Decomposition and its type, Anomalies, Normalization- 1NF, 2NF, 3NF Decomposition using dependency preservation, BCNF, Multivalued dependency, 4NF, Join Dependency and 5NF	10	18
3	**Query Processing & Query Optimization and Transaction Management** Query Processing, Query Optimization, Transaction Concepts, ACID Property, Transaction state diagram, schedules, serializability of transactions, Two-phase commit/locking protocol, Log-based recovery, Deadlock	10	18
4	**Introduction to NoSQL & Basic MongoDB Operations** Introduction to NoSQL, Why NoSQL?, Advantages & Disadvantages of NoSQL, Types of NoSQL databases, When should NoSQL be used?, RDBMS v/s MongoDB, Basic Database Commands, Operations & Methods	13	22
5	**Advanced MongoDB Concepts** Aggregation Commands, Indexes, Regex, Schema Validation, Embedded Documents, Create user and add role, Cursor, Backup and Restore	12	20
	Total	60	100

Suggested Distribution Of Theory Marks Using Bloom's Taxonomy

Level	Remembrance	Understanding	Application	Analyze	Evaluate	Create
Weightage	30	40	30	0	0	0

NOTE : This specification table shall be treated as a general guideline for the students and the teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcomes

At the end of this course, students will be able to:	
CO1	**implement** the advance concepts of SQL.
CO2	**apply** the basic concepts of data models, relational database design and normalization techniques.
CO3	**discuss** query processing, query optimization and concept of transactions and deadlock.
CO4	**demonstrate** the concepts of NoSQL databases.
CO5	**execute** advanced-level queries with MongoDB as a document database.

Reference Books

1.	**Database System Concepts** By Abraham Silberschatz, Henry F. Korth, S. Sudarshan | McGraw-Hill
2.	**An introduction to Database Systems** By C J Date | Pearson
3.	**Database Management Systems** By Ramakrishnan, Gehrke | Tata McGraw-Hill
4.	**Database Systems** By Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom | Pearson | 2nd
5.	**SQL Queries for Mere Mortals** By John L. Viescas, Michael J. Hernandez | 4th
6.	**Fundamentals of Database Systems** By Navathe | Pearson Education
7.	**SQL- PL/SQL** By Ivan bayross
8.	**MongoDB Basics** By Peter Membrey, David Hows, Eelco Plugge | Apress | 1st

List of Practical

1. Perform SQL queries on Create, Insert, Select and Update command

Part A:

Create following table using query according to the definition.

DEPOSIT(ACTNO INT, CNAME VARCHAR(50), BNAME VARCHAR(50), AMOUNT DECIMAL(8,2), ADATE DATETIME)
BRANCH(BNAME VARCHAR(50), CITY VARCHAR(50))
CUSTOMERS(CNAME VARCHAR(50), CITY VARCHAR(50))
BORROW(LOANNO INT,CNAME VARCHAR(50),BNAME VARCHAR(50),AMOUNT DECIMAL(8,2))

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. Insert a record (550,'JAY','AJNI',NULL)in the BORROW table.
5. Display Account No, Customer Name & Amount from DEPOSIT.
6. Display Loan No, Amount from BORROW.
7. Display loan details of all customers who belongs to 'ANDHERI' branch.
8. Give account no and amount of depositor, whose account no is equals to 106.
9. Give name of borrowers having amount greater than 5000.
10. Give name of customers who opened account after date '1-12-96'.

11. Display name of customers whose account no is less than 105.
12. Display name of customer who belongs to either 'NAGPUR' Or 'DELHI'. (**OR & IN**)
13. Display name of customers with branch whose amount is greater than 4000 and account no is less than 105.
14. Find all borrowers whose amount is greater than equals to 3000 & less than equals to 8000.(**AND & BETWEEN**)
15. Find all depositors who do not belongs to 'ANDHERI' branch.
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)**
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. Update deposit amount of all customers from 3000 to 5000.
25. Change branch name of ANIL from VRCE to C.G. ROAD. (Use **Borrow Table**)
26. Update Account No of SANDIP to 111 & Amount to 5000.
27. Give 10% Increment in Loan Amount.
28. Update deposit amount of all depositors to 5000 whose account no between 103 & 107.
29. Update amount of loan no 321 to *NULL*.
30. Display the name of borrowers whose amount is NULL.

Part B:

Create following table using query according to the definition.

STUDENT(ROLLNO INT, NAME VARCHAR(50), BIRTHDATE DATETIME, SPI DECIMAL(8,2), CITY VARCHAR(50), BACKLOG INT, BRANCH VARCHAR(50))

From the above given tables perform the following queries:

1. Give RollNo and Name of students, whose RollNo is greater than 103 and backlog is greater than 0 and branch is either CE or IT.
2. Give name of students whose SPI is between 8 and 9 and branch is either CE or IT. (**OR & IN**)
3. Find all students who do not belongs to 'CE' branch.
4. Display RollNo and Name of first three students.
5. Display all the details of first three students whose SPI is greater than 8.5.
6. Retrieve all unique cities using DISTINCT.
7. Retrieve all unique branches using DISTINCT.
8. Retrieve all the records of student table as per their Backlog in descending order and then SPI in ascending order.
9. Update the branch and city of Jay to MCA and Jamangar respectively.
10. Update the backlog of Keyur and Bhoomi to *NULL*.
11. Display the name of students whose backlog is NULL and backlog is greater than 1 and branch is either CE or IT.

2. Perform SQL queries on Delete, Drop and Truncate command

Part-A:

Create following table using query according to the definition.

Employee(EmpNo INT,EmpName VARCHAR(25),JoiningDate DATETIME,Salary DECIMAL (8,2),City VARCHAR(20))

From the above given tables perform the following queries (DELETE Operation):

1. Delete all the records of Employee 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 all the records of Employee table. (Use Truncate)
5. Remove Employee table. (Use Drop)

Part-B:

	Create following table using query according to the definition. DEPOSIT(ACTNO INT,CNAME VARCHAR(50),BNAME VARCHAR(50),AMOUNT DECIMAL(8,2),ADATE DATETIME) **From the above given tables perform the following queries (DELETE Operation):** 1. Delete all the records of Account table having amount greater than and equals to 4000. 2. Delete all the accounts Bname is CHANDI. 3. Delete all the accounts having adate after 1-10-1995. 4. Delete all the records of Account table. (Use Truncate) 5. Remove Account table. (Use Drop) **Part-C:** Create following table using query according to the definition. DEPOSIT(ACTNO INT,CNAME VARCHAR(50),BNAME VARCHAR(50),AMOUNT DECIMAL(8,2),ADATE DATETIME) **From the above given tables perform the following queries (DELETE Operation):** 1. Update the amount of Anil to 5000. 2. Update amount of actno 109 to NULL. 3. Retrieve all the records of account table as per their bname in descending order. 4. Retrieve all the records of account table whose amount is between 3000 and 5000. 5. Remove Account table. (Use Drop)
3.	**Perform SQL queries on Like operator** **Part-A:** Create following table using query according to the definition. Student(Enrollment_No VARCHAR(20),Name VARCHAR(25),CPI DECIMAL(5,2),Birthdate DATETIME) **From the above given tables perform the following queries (Alter and Rename Operation):** 1. Add two more columns City VARCHAR (20) and Backlog INT. 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. 7. Remove the table STUDENT_MASTER **Part-B:** Create following table using query according to the definition. DEPOSIT(ACTNO INT,CNAME VARCHAR(50),BNAME VARCHAR(50),AMOUNT DECIMAL(8,2),ADATE DATETIME) **From the above given tables perform the following queries (Alter and Rename 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 **Part-C:** **From the above given tables perform the following queries (Alter and Rename Operation):** 1. Delete Column ADATE from the DEPOSIT table. 2. Rename Column CNAME to CustomerName. 3. Change name of table DEPOSIT to DEPOSIT_DETAIL. 4. Remove the table DEPOSIT_DETAIL.
4.	**Perform SQL queries on Group By command** **Part-A:** Create following table using query according to the definition. Student(StuID INT,FirstName VARCHAR(25),LastName VARCHAR(25),Website VARCHAR(50),City VARCHAR(25)) **From the above given tables perform the following queries (LIKE Operator):** 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'. **Part-B:** **From the above Student tables perform the following queries (LIKE Operator):** 1. Display the name of students having no website and name consists of five characters. 2. Display all the students whose last name consist of 'jer'. 3. Display all the students whose city name starts with either 'r' or 'b'. 4. Display all the name students having websites. 5. Display all the students whose name starts from alphabet A to H. 6. Display all the students whose name's second character is vowel. **Part-C:** **From the above Student table perform the following queries (LIKE Operator):** 1. Display the name of students having no website and name consists of minimum five characters. 2. Display all the students whose last name starts with 'Pat'. 3. Display all the students whose city name does not start starts with 'b'. 4. Display all the students whose name starts from alphabet A or H. 5. Display all the students whose name's second character is vowel and of and start with H.
5.	**Perform SQL queries on Group By command** **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 following table using query according to the definition. Employee(EmpNo INT,EmpName VARCHAR(25),JoiningDate DATETIME,Salary DECIMAL (8,2),City VARCHAR(20)) **From the above tables perform the following queries:** 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.
6.	**Perform SQL queries on Joins**

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 6.

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 6.

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.

7. Perform SQL queries on Joins with Group By command

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 following table using query according to the definition.

Employee(EmpNo INT,EmpName VARCHAR(25),JoiningDate DATETIME,Salary DECIMAL (8,2),City VARCHAR(20))

From the above given tables perform the following queries:

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.

8. Perform SQL queries on Joins with Group By command

Part-A:

Create following table using query according to the definition.
Computer(RollNo int, Name varchar(50))
Electrical(RollNo int, Name varchar(50))

From the above given tables perform the following queries:

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 following table using query according to the definition.
Employee(EID int, Name varchar(50))
Customer(CID int, Name varchar(50))

From the above given tables perform the following queries:

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:

Perform all the queries of Part-B but display ID and Name columns instead of Name only.

9. Perform SQL queries for data validation

Part-A:

Create following table using query according to the definition.
Cricket(Name varchar(50), City varchar(50), Age int)

From the above given tables perform the following queries:

1. Create table Worldcup 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.

Part-B:

Create following table using query according to the definition.

	Employee(Name varchar(50), City varchar(50), Age int) **From the above given tables perform the following queries:** 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 4. Insert the Data into Employee_info from Employee Whose Second Character Should Be 'A' And String 5. Should Have At least 7 Characters in Name Field. 6. Insert the Data into Employee_info from Employee Whose age is more than 32. **Part-C:** **Perform following queries on Employee table.** 1. Delete all the Rows from Employee_info. 2. Delete the Detail of employee from Employee Whose City is Rajkot. 3. Rename the Table Employee to New_Employee. 4. Delete all the records from New_Employee table. 5. Destroy table New_Employee with all the Data.
10.	**Perform SQL queries on Sub Query** **Part-A:** Create following table using query according to the definition. Employee(EID int, EName varchar(50), Department varchar(50),Salary decimal (10,2), JoiningDate datetime, City varchar(50)) **From the above given tables perform the following queries:** 1. Display the Highest, Lowest, Total, and Average salary of all employees. Label the columns Maximum, 2. Minimum, Total_Sal and Average_Sal, respectively. 3. Find total number of employees of EMPLOYEE table. 4. Give maximum salary from IT department. 5. Count total number of cities of employee without duplication. 6. Display city with the total number of employees belonging to each city. 7. Display city having more than one employee. 8. Give total salary of each department of EMPLOYEE table. 9. Give average salary of each department of EMPLOYEE table without displaying the respective department 10. name. 11. Give minimum salary of employee who belongs to Ahmedabad. 12. List the departments having total salaries more than 50000 and located in city Rajkot. **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. 4. Display total salary of each department with total salary exceeding 35000 and sort the list by total salary. 5. List out department names in which more than two employees. **Part-C:** 1. Count the number of employees living in Rajkot or Baroda. 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 in IT department. 4. Display total salary of each department with total salary exceeding 35000 and sort the list by total salary in descending order.
11.	**Perform SQL queries on Joins, Group By and Sub query** **Part-A:** Create following table using query according to the definition. Student(Rno int, Name varchar(50), Branch varchar(50)) Result(Rno int, SPI decimal(10,2)) Employee(EmployeeNo int, Name varchar(50), ManagerNo int) **From the above given tables perform the following queries:** 1. Combine information from student and result table using cross join or Cartesian product.

2. Display Rno, Name, Branch and SPI of all students.
3. Display Rno, Name, Branch and SPI of CE branch's student only.
4. Display Rno, Name, Branch and SPI of other than EC branch's student only.
5. Display average result of each branch.
6. Display average result of each branch and sort them in ascending order by SPI.
7. Display average result of CE and ME branch.
8. Perform the left outer join on Student and Result tables.
9. Perform the right outer join on Student and Result tables.
10. Perform the full outer join on Student and Result tables.
11. Retrieve the names of employee along with their manager name from the Employee table.

Part-B:

Create following table using query according to the definition.
Person(PersonID int, PersonName varchar(50), DepartmentID int, Salary decimal(10,2), JoiningDate datetime, City varchar(50))
Department(DepartmentID int, DepartmentName varchar(50), DepartmentCode varchar(50), Location varchar(50))

From the above given tables perform the following queries:

1. Find all persons with their department name & code.
2. Give department wise maximum & minimum salary with department name.
3. Find all departments whose total salary is exceeding 100000.
4. Retrieve person name, salary & department name who belongs to Jamnagar city.
5. Find all persons who does not belongs to any department.
6. Find department wise person counts.
7. Find average salary of person who belongs to Ahmedabad city.
8. Produce Output Like: earns from department monthly. (In Single Column)
9. List all departments who have no persons.
10. Find city & department wise total, average & maximum salaries.

Part-C:

1. Display Unique city names.
2. List out department names in which more than two persons.
3. Combine person name's first three characters with city name's last three characters in single column.
4. Give 10% increment in Computer department employee's salary.
5. Display all the person name's who's joining dates difference with current date is more than 365 days.

12. Implement Stored Procedures

Part-A:

Create following table using query according to the definition.
City(CityID int (Primary Key), Name varchar(50) (Unique Key), Pincode varchar(50) (Not Null), Remakrs varchar(50))
Village(VID int (Primary Key), Name varchar(50) (Not Null), CityID int (Foreign Key))

From the above given tables perform the following queries:

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 villag.
7. Create below student table with following constraint.

- Do not allow SPI more than 10
- Do not allow Bklog less than 0.
- Enter the default value as 'General' in branch to all new records IF no other value is specified.
- Student(Rno int (PK), Name varchar(50), Branch varchar(50), SPI decimal(10,2), Bklog int)
    - Try to update SPI of Raju from 8.80 to 12.
    - 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.

	• 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.** • 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.
13.	**Implement Triggers** **Part-A:** Create following table using query according to the definition. Student(Rno int, Name Varchar (50),City Varchar (50), DID int) Academic(Rno int, SPI decimal(10,2), Bklog int) Department(DID int, DName Varchar (50)) **From the above given tables perform the following queries:** 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. **Part-B:** 1. Display name of student who is having maximum SPI. 2. Display details of students having more than 1 backlog. 3. Display name of student who is having second highest SPI. **Part-C:** 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. 3. Display name of students whose SPI is more than 9 and who is from electrical department.
14.	**Implement Cursors** **Part-A:** Create following table using query according to the definition. Student(StuID Int, Name Varchar (100), EnrollmentNo Varchar (12), Division Varchar (50), Sem Int, BirthDate Datetime, Email Varchar (100), ContactNo Varchar (50)) **From the above given tables perform the following queries:** 1. Display Name of Student who belongs to either Semester 3 or 5. (Use IN & OR) 2. Find Student Name & Enrollment No in which Student Id between 103 to 105. 3. Find Student Name & Enrollment No with their Email Who belongs to 5th Semester. 4. Display All the Details of first three students. 5. Display Name & Enrollment no of first 30% Students who's contact number ends with 7. 6. Display Unique Semesters. 7. Retrieve All the Students who have no Enrollment. 8. Find All Students whose Name does not start with 'V'. 9. Find All Students in which Email Contains '3@G' & Only Six Characters. 10. Find Out All the Students whose First Name Starts with F And third character must be R. **Part-B:** 1. Find All the Student Details whose Contact No contains 877. 2. Display Student Name in Which Student belongs to Semester 3 & Contact Number Does Not Contains 8 & 9. 3. Find Students who born after date 01-01-1990.

	4. Update Division to BCX-5 & Semester 5 whose Student Id Is 102. 5. Change the Student's Name to Firoz Sherasiya in which Email is Firoz.Me@Gmail.Com & Contact No is 888599992. **Part-C:** 1. Add one more Column City Varchar (50) in Student Table. 2. Remove All BCX-3 Division Students. 3. Change Column Name Email to EmailAddress. 4. Remove All the Data from Student Table Using Truncate. 5. Write an SQL query to clone a new table Student_New from Student table with all data.
15.	**Implementation of Functions** **Part-A:** Create following table using query according to the definition. Employee(EID Int, EName Varchar (100), Gender Varchar (10), JoiningDate Datetime, Salary Decimal (8,2), City Varchar (100)) **From the above given tables perform the following queries:** 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". **Part-B:** 1. Display name of the employees whose city name ends with "ney" &contains six characters. 2. Write a query to convert value 15 to string. 3. Add department column with varchar (20) to Employee table. 4. Set the value of department to Marketing who belongs to London city. 5. Display all the employees whose name ends with either "n" or "y". **Part-C:** 1. Find smallest integer value that is greater than or equal to 63.1, 63.8 and -63.2. 2. Display all employees whose joining date is not available. 3. Display name of the employees in capital letters and city in small letters. 4. Change the data type of Ename from varchar (30) to char (30). 5. Display city wise maximum salary.
18.	**SQL Like operator and group by** **Consider the same Employee table of practical no 17 and perform the following queries** **Part-A:** 1. Produce output like works at and earns . 2. Find total number of employees whose salary is more than 5000. 3. Write a query to display first 4 & last 3 characters of all the employees. 4. List the city having total salaries more than 15000 and employees joined after 1st January, 2014. 5. Write a query to replace "u" with "oo" in Ename. **Part-B:** 1. Display city with average salaries and total number of employees belongs to each city. 2. Display total salaries paid to female employees. 3. Display name of the employees and their experience in years. 4. Remove column department from employee table. 5. Update the value of city from syndey to null. **Part-C:** 1. Retrieve all Employee name, Salary & Joining date. 2. Find out combine length of Ename & Gender. 3. Find the difference between highest & lowest salary. 4. Rename a column from Ename to FirstName. 5. Rename a table from Employee to EmpMaster.
19.	**SQL constraints and group by**

Create following table using query according to the definition.
Person(PersonID Int Primary Key, PersonName Varchar (100) Not Null, DepartmentID Int Foreign Key and Null, Salary Decimal (8,2) Not Null, JoiningDate Datetime Not Null, City Varchar (100) Not Null)
Department(DepartmentID Int Primary Key, DepartmentName Varchar (100) Not Null and Unique, DepartmentCode Varchar (50) Not Null and Unique, Location Varchar (50) Not Null)

Part-A:

1. Find all persons with their department name & code.
2. Give department wise maximum & minimum salary with department name.
3. Find all departments whose total salary is exceeding 100000.
4. Retrieve person name, salary & department name who belongs to Jamnagar city.
5. Find all persons who does not belongs to any department.

Part-B:

1. Find department wise person counts.
2. Find average salary of person who belongs to Ahmedabad city.
3. Produce Output Like: earns from department monthly. (In Single Column)
4. List all departments who have no persons.
5. Find city & department wise total, average & maximum salaries.

Part-C:

1. Display Unique city names.
2. List out department names in which more than two persons.
3. Combine person name's first three characters with city name's last three characters in single column.
4. Give 10% increment in Computer department employee's salary.
5. Display all the person name's who's joining dates difference with current date is more than 365 days.

20. SQL View

Create following table using query according to the definition.
Student(RNo int Primary Key, Name varchar(50), Branch varchar(50), SPI varchar(50), Bklog varchar(50))

Part-A:

1. Create a view Personal with all columns.
2. Create a view Student_Details having columns Name, Branch & SPI.
3. Create a view Academic having columns RNo, Name, Branch.
4. Create a view Student_Data 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 Academic 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 Academic view.

Part-B:

Create following table using query according to the definition.
Student(Rno Int Primary Key, Name Varchar (50) Not Null, Branch Varchar (50) Not Null, SPI Decimal (4,2) Not Null, Bklog Int Not Nul)

1. Create a view Personal with all columns.
2. Create a view Student_Details having columns Name, Branch & SPI.
3. Create a view Academic having columns RNo, Name, Branch.
4. Create a view Student_Data having all columns but students whose bklogs are 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 Academic view. (107, Meet, ME). Remaining all columns must be null.
7. Update the branch of Amit from CE to ME in Student_Details view.

Part-C:

1. Delete a student whose roll number is 104 from Academic view.
2. Create a view that displays information of all students whose spi is above 8.5.
3. Create a view that displays 0 backlog students.
4. Create a view Computer that displays CE branch data only.
5. Create a view Result_EC that displays the name and SPI of students with SPI less than 5 of branch EC.
6. Update the result of student Sanjay to 4.90 in Result_EC view.

	7. Create a view Stu_Bklog with RNo, Name and Bklog columns in which name starts with 'M' and having bklogs more than 5. 8. Drop Computer view form the database.
21.	**SQL Stored procedure** Create following table using query according to the definition. Person(WorkerID Int Primary Key and Auto Increment,FirstName Varchar (100) Not Null, LastName Varchar (100) Not Null, Salary Decimal (8,2) Not Null, JoiningDate Datetime Not Null, DepartmentID Int Foreign Key and Null, DesignationID Int Foreign Key and Null) Department(DepartmentID Int Primary Key, DepartmentName Varchar (100) Not Null and Unique) Designation(DesignationID Int Primary Key, DesignationName Varchar (100) Not Null and Unique) **From the above given tables create Stored Procedures:** **Part-A:** 1. Department, Designation & Person Table's INSERT, UPDATE & DELETE Procedures. 2. Department, Designation & Person Table's SELECTBYPRIMARYKEY 3. Department, Designation & Person Table's (If foreign key is available then do write join and take columns on select list) 4. Create a Procedure that shows details of the first 3 persons. **Part-B:** 1. Create a Procedure that takes the department name as input and returns a table with all workers working in that department. 2. Create Procedure that takes department name & designation name as input and returns a table with worker's first name, salary, joining date & department name. 3. Create a Procedure that takes the first name as an input parameter and display all the details of the worker with their department & designation name. 4. Create Procedure which displays department wise maximum, minimum & total salaries. 5. Create Procedure which displays designation wise average & total salaries. **Part-C:** 1. Create Procedure that Accepts Department Name and Returns Person Count. 2. Create Procedure that accepts Department Name & Designation as a parameter with given test cases and returns a table with FirstName, LastName, Salary, JoiningDate, DepartmentName & Designation. 3. Create Procedure that returns DepartmentID, DepartmentName & Count of all person belongs to that department. i.e. 1 | Admin | 2
22.	**SQL Trigger** Create following table using query according to the definition. Person(PersonID Int Primary Key,Person Varchar (100) Not Null, Salary Decimal (8,2) Not Null, JoiningDate Datetime Not Null, City Varchar (100) Not Null, Age Int Null, BirthDate Datetime Not Null) PersonLog(PLogID Int Primary Key and Auto increment, PersonID Int Not Null, PersonName Varchar (250) Not Null, Operation Varchar (50) Not Null, UpdateDate Datetime Not Null) **From the above given tables create Stored Procedures:** **Part-A:** 1. Create a trigger that fires on INSERT, UPDATE and DELETE operation on the Person table to display a message "Record is Affected." 2. Create a trigger that fires on INSERT, UPDATE and DELETE operation on the Person table. For that, log all operations performed on the person table into PersonLog. **Part-B:** 1. Create an INSTEAD OF trigger that fires on INSERT, UPDATE and DELETE operation on the Person table. For that, log all operations performed on the person table into PersonLog. 2. Create a trigger that fires on INSERT operation on the Person table to convert person name into uppercase whenever the record is inserted. **Part-C:** 1. Create a trigger that fires on INSERT operation on person table, which calculates the age and update that age in Person table. 2. Create DELETE trigger on PersonLog table, when we delete any record of PersonLog table it prints 'Record deleted successfully from PersonLog'.

23. SQL Cursor

Create following table using query according to the definition.
Products(Product_id Int Primary Key, Product_Name Varchar (250) Not Null, Price Decimal (10,2) Not Null)

From the above given tables perform the following queries:

Part-A:

1. Create a cursor Product_Cursor to fetch all the rows from a products table.
2. Create a cursor Product_Cursor_Fetch to fetch the records in form of ProductID_ProductName. (Example: 1_Smartphone)
3. Create a cursor Product_CursorDelete that deletes all the data from the Products table.

Part-B:

1. Create a cursor Product_CursorUpdate that retrieves all the data from the products table and increases the price by 10%.

Part-C:

1. Create a cursor to insert details of Products into the NewProducts table if the product is “Laptop” (Note: Create NewProducts table first with same fields as Products table)

24. SQL Scalar valued functions

Perform the following queries.

Part-A:

1. Write a function to print "hello world".
2. Write a function which returns addition of two numbers.
3. Write a function to print a cube of a given number.
4. Write a function to check whether the given number is ODD or EVEN.
5. Write a function which returns a table with details of a person whose first name starts with B.
6. Write a function which returns a table with unique first names from the person table.

Part-B:

1. Write a function to compare two integers and return the comparison result. (Using Case statement)
2. Write a function to print number from 1 to N. (Using while loop)
3. Write a function to print the sum of even numbers between 1 to 20.

Part-C:

1. Write a function to check whether a given number is prime or not.
2. Write a function which accepts two parameters start date & end date, and returns a difference in days.
3. Write a function which accepts two parameters year & month in integer and returns total days in a given month & year.
4. Write a function which accepts departmentID as a parameter & returns a detail of the persons.

25. MongoDB Create, Insert, Update and Drop methods

Perform the following queries in MongoDB.

Part-A:

1. Create Database with Name: BANK_INFO.
2. Create a collection name Deposite with filds are ACTNO, CNAME, BNAME, AMOUNT, ADATE.
3. Insert following documents in Deposite collection
    - 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
4. Retrieve/Display every document of Deposite collection.
5. Retrieve/Display every document of Deposite collection. (Use: pretty())
6. Display only one document of Deposite collection. (Use: findOne())
7. Insert following document to Deposite collection. (Use: insertOne())
    - 109, KIRTI, VIRAR, 3000.00, 3-5-97
8. Insert following documents to your collection. (Use: insertMany())
    - 110, MITALI, ANDHERI, 4500.00, 4-9-95

- 111, RAJIV, NEHRU PLACE, 7000.00, 2-10-98

9. Display documents with CNAME, BNAME and AMOUNT fields.
10. Display every document of Deposite collection on ascending order by CNAME.
11. Display every document of Deposite collection on descending order by BNAME.
12. Display every document of Deposite collection on ascending order by ACTNO and descending order by AMOUNT.
13. Display only two documents of Deposite collection.
14. Display 3rd document of Deposite collection.
15. Display 6th and 7th documents of Deposite collection.
16. Display the count of documents in Deposite collection.

Part-B:

1. Insert following document to Deposite collection. (Use: insertOne())
   - 112, MANISH, ANDHERI, 8000.00, 9-5-98
2. Display 9th document of Deposite collection.
3. Display 11th and 12th documents of Deposite collection.

Part-C:

1. Display every document of Deposite collection on ascending order by AMOUNT and descending order by BNAME.
2. Display only five documents of Deposite collection.
3. Delete the collection Deposite.
4. Drop BANK_INFO database.

26. MongoDB Create, Insert, Find, Update, Regex and Drop methods

Perform the following queries in MongoDB.

Part-A:

1. Create Database with Name: EMPLOYEE_INFO.
2. Create a collection name employee with filds are EID, ENAME, GENDER, JOININGDATE, SALARY, CITY.
3. Insert following documents in Deposite collection
   - 1, Nick, Male, 01-JAN-13, 4000, London
   - 2, Julian, Female, 01-OCT-14, 3000, New York
   - 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, New York
   - 7, Sara, Female, 01-AUG-17, 4800, Sydney
   - 8, Emily, Female, 01-JAN-15, 5500, New York
   - 9, Michael, Male, NULL, 6500, London
   - 10, John, Male, 01-JAN-15, 8800, London
4. Display employees whose gender is Male.
5. Display employees who belong to London city.
6. Display employees whose salary is greater than 3500.
7. Display employees whose joining date is before 2015-01-01.
8. Display employees whose EID is greater than or equal to 7.
9. Display employees whose city is Landon or New York (use:IN)
10. Display employees who do not belongs to Landon or New York (use: NOT IN)
11. Display the EID of those employee who lives in city London.
12. Display first 2 employee names who lives in New york.
13. Display next 2 employee after skipping first 2 whose city is London.
14. Display Male employees who lives Sydney.
15. Display EID, ENAME, CITY and SALARY of those employees who belongs to London or Sydney.
16. Display ENAME, SALARY, and CITY of those employee whose salary is more than 7000.
17. Display documents whose name start with E.
18. Display documents whose name starts with S or M in your collection.
19. Display documents where city starts with A to M in your collection.
20. Display documents where city name ends in 'ney'.
21. Display employee info whose name contains n. (Both uppercase(N) and lowercase(n))

22. Display employee info whose name starts with E and having 5 characters.
23. Display employee whose name start with S and ends in a.
24. Display EID, ENAME, CITY and SALARY whose name starts with 'Phi'.
25. Display ENAME, JOININGDATE and CITY whose city contains 'dne' as three letters in city name.
26. Display ENAME, JOININGDATE and CITY who does not belongs to city London or Sydney.
27. Delete the documents whose city is New York.
28. Update ENAME of Nick to 'Naysa' and GENDER to 'Female'.

Part-B:

1. Create a collection name student with filds are ROLLNO, SNAME, DEPARTMENT, FEES, SEM, GENDER, CITY.
2. Insert following documents in Deposite collection
    - 101, Vina, CE, 15000, 3, Female, Rajkot
    - 102, Krisha, EC, 8000, 5, Female, Ahmedabad
    - 103, Priti, Civil, 12000, 7, Female, Baroda
    - 104, Mitul, CE, 15000, 3, Male, Rajkot
    - 105, Keshav, CE, 15000, 3, Male, Jamnagar
    - 106, Zarna, Civil, 12000, 5, Female, Ahmedabad
    - 107, Nima, EE, 9000, 5, Female, Rajkot
    - 108, Dhruv, Mechanical, 10000, 5, Male, Rajkot
    - 109, Krish, Mechanical, 10000, 7, Male, Baroda
    - 110, Zeel, EE, 9000, 3, Female, Jamnagar
3. Display Female students.
4. Display students who belong to Rajkot city.
5. Display students studying in 7th sem.
6. Display students not studying in 3rd sem.
7. Display students whose roll no is greater than 107.
8. Display students whose city is Jamnagar or Baroda (use:IN)
9. Display students whose fees is less than 9000.
10. Display the roll no of those students who belongs to Mechanical department.
11. Display first 2 students names who lives in Baroda.
12. Display Male students who studying in 3rd sem.
13. Display sname and city and fees of those students whose roll no is less than 105.
14. Display documents where sname start with K.
15. Display documents where sname starts with Z or D in your collection.
16. Display documents where city starts with A to R in your collection.
17. Display students' info whose name start with P and ends in i.
18. Display students' info whose department name starts with 'C'.
19. Display name, sem, fees, and department whose city contains 'med' as three letters somewhere in city name.
20. Display name, sem, fees, and department who does not belongs to city Rajkot or Baroda.
21. Delete the documents whose city is Jamnagar.
22. Update sname of Krish to 'fenny' and gender to 'Female'.

Part-C:

1. Display next 2 students after skipping first 2 whose city is Ahmedabad.
2. Display rollno, sname, fees, and department of those students who is from Baroda and belongs to CE department.
3. Display documents where city name ends in 'oda'.
4. Display students' info whose name contains v. (Both uppercase(V) and lowercase(v))
5. Display students' info whose name starts with V and having 4 characters.

27. **MongoDB Find and Group by methods**

Perform the following queries in MongoDB (use employee collection of Lab 26).

Part-A:

1. Display distinct city.
2. Display city wise number of persons.
3. Display sum of salary in your collection.
4. Display average of salary in your document.

5. Display maximum and minimum salary of your document.
6. Display city wise total salary in your collection.
7. Display gender wise maximum salary in your collection.
8. Display city wise maximum and minimum salary.
9. Display count of persons lives in Sydney city in your collection.
10. Display average salary of New York city.

Part-B:

1. Display distinct department.
2. Display city wise number of students.
3. Display sum of fees in your collection.
4. Display average of fees in your document.
5. Display maximum and minimum fees of your document.

Part-C:

1. Display department wise total fees in your collection.
2. Display gender wise maximum fees in your collection.
3. Display department maximum and minimum fees.
4. Display count of persons lives in Rajkot city in your collection.
5. Display department wise number of students.