Subject Code: 01CE0302

Subject Name: Database Management System

B.Tech. Year - II

Objective:

A major purpose of a database system is to provide users with an abstract view of the data. The lowest level of abstraction, the physical level, describes how the data are actually stored. The next-higher level of abstraction, the logical level, describes what data are stored, and what relationships exist among those data. The highest level of abstraction, the view level, describes parts of the database that are relevant to each user; application programs used to access a database form part of the view level. The SQL will help the user to retrieve the data from the database as per the requirement.

Credits Earned: 5 Credits

Course Outcomes: After learning the course, the students should be able:
- Use Relational Database and different models of Database. (Apply)
- Design ER Model for an Application. (Create)
- Apply Concepts of normalization with functional dependency to construct Data dictionary. (Apply)
- Implement Structured Query Language (SQL) and evaluate query expression. (Evaluate)
- Differentiate and Execute transactional Concepts and locking mechanism (Analyze)
- Use concepts of Database Security on Database. (Apply)

Pre-requisite of course: NA.

Teaching and Examination Scheme

Teaching Scheme (Hours)			Credits	Theory Marks			Tutorial/ Practical Marks		Total Marks
Theory	Tutorial	Practical		ESE (E)	Mid Sem (M)	Internal (I)	Viva (V)	Term work (TW)	
4	0	2	5	50	30	20	25	25	150

Contents:

Unit	Topics	Contact Hours
1	**Introductory concepts of DBMS :** • Introduction and applications of DBMS • Purpose of data base • Data Independence • Database System architecture- levels Mappings • Database, users and DBA	3
2	**Relational Model :** • Structure of relational databases • Domains • Relations • Relational algebra – fundamental operators and syntax, relational algebra queries, tuple relational calculus	4
3	**Entity-Relationship model :** • Basic concepts • Design process • Constraints, Keys, Design issues • E-R diagrams - weak entity sets, extended E-R features – generalization, specialization, aggregation, reduction to E-R database schema	5
4	**Relational Database design :** • Functional Dependency – definition, trivial and non-trivial FD • Closure of FD set • Closure of attributes • Irreducible set of FD • Normalization – 1Nf, 2NF, 3NF, Decomposition using FD-dependency preservation, BCNF, 4NF • Multi- valued dependency • Join dependency and 5NF	5
5	**Query Processing & Query Optimization :** • Introduction • Measures of query cost • Selection operation • Sorting and Join • Evaluation of expressions, transformation of relational expressions, estimating statistics of expression results, • Evaluation plans and materialized views	5
6	**Transaction Management :** • Transaction concepts • Properties of transactions • Serializability of transaction and testing for serializability,	10

	• System recovery - Two- Phase Commit protocol • Recovery and Atomicity - Log-based recovery • Concurrent executions of transactions • Concepts of dead lock using locking mechanism • Locking mechanism - two-phase locking protocol, • Isolation and Intent locking	
7	**Security:** • Overview • Discretionary access control • Mandatory Access Control • Data Encryption	2
8	**SQL Concepts** • **Working with DDL, Creating and Managing Constraints** --- NOT NULL, UNIQUE constraints, PRIMARY KEY, FOREIGN KEY, and CHECK constraints, Managing constraints, • **Constructing DML Statements, SELECT Statements and Relational Database Technology, Using the WHERE Clause, Restricting Rows ---** Logical comparisons and precedence rules, Sorting rows ,Introduction to functions – single row functions, • **Using Character, Number, and Date Functions, Conversion functions, NULL functions, Conditional expressions, Executing Database Joins---** Cross joins and natural joins, Join clauses, Inner versus outer joins, Self joins and hierarchical queries, • **Working with Group Functions, Using Complex SQL with Aggregated Data, Creating Subqueries, Creating and Managing Views--** Creating views, DML operations and views, Managing views, • **Working with Sequences---**Working with sequences, Indexes and synonyms, • **Fundamentals of Database Security---** Controlling user access, Creating and revoking object privileges, Regular expressions, • **Oracle Proprietary Join Syntax--**Cartesian product and the JOIN operations, NONEQUIJOINS, OUTER joins	13
9	**PL/SQL Concepts :** • Cursors • Stored Procedures and Stored Function, • Database Triggers	3
	Total Hours	**50**

References:

1. An introduction to Database Systems, C J Date, Addition-Wesley.
2. Database System Concepts, Abraham Silberschatz, Henry F. Korth & S. Sudarshan, McGraw Hill.
3. Understanding SQL by Martin Gruber, BPB
4. SQL- PL/SQL by Ivan bayross
5. Oracle – The complete reference – TMH /oracle press

Suggested Theory distribution:

The suggested theory distribution as per Bloom's taxonomy is as per follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery and evaluation					
Remember	Understand	Apply	Analyze	Evaluate	Create
10%	30%	40%	10%	5%	5%

Suggested List of Experiments:

Practical -1

Create a table ACCOUNT

Column name	Data Type	Size
acc_no	varchar2	5
Name	varchar2	30
City	varchar2	20
Balance	Number	10,2
loan_taken	varchar2	5

Insert the following records.

acc_no	Name	City	Balance	loan_taken
A001	Patel Jigar	Mehsana	50000	YES
A002	Patel Ramesh	Mehsana	50000	YES
A003	Dave Hardik	Ahmedabad	75000	NO
A004	Soni Hetal	Ahmedabad	100000	NO
A005	Sony Atul	Vadodara	100000	YES

Create a Table LOAN

Column Name	Data Type	Size
loan_no	varchar2	5
acc_no	varchar2	5
loan_amt	number	10,2
interest_rate	number	5,2
loan_date	date	
remaining_loan	number	10,2

Insert the following Records.

Loan_no	Acc_no	Loan_amt	Interest_rate	Loan_date	Remaining_loan
L001	A001	100000	7	1-jan-04	75000
L002	A002	300000	9	18-may-04	150000
L003	A005	500000	11	15-june-04	300000

Create a table INSTALLMENT

Column Name	Data Type	Size
loan_no	varchar2	5
inst_no	varchar2	5
inst_Date	Date	
Amount	Number	10,2

Insert following Records

Loan_no	Inst_no	Date	Amount
L001	I001	2-Feb-04	15000
L002	I002	18-June-04	20000
L003	I003	15-July-04	20000

Create a Table TRANSACTION

Column Name	Data Type	Size
acc_no	Varchar2	5
tr_Date	Date	
Amt	Number	10,2
type_of_tr	Char	1
mode_of_pay	Varchar2	10

Insert a Following Records

Acc_no	Date	Amt	Type_of_tr	Mode_of_pay

A001	3-may-04	10000	D	Cash
A002	5-july-04	5000	W	Cheque
A003	12-Aug-04	25000	D	Cheque
A004	15-may-04	30000	D	Cheque
A005	22-oct-04	15000	W	Cash

List of queries

1. Display all rows and all columns of table Transaction.
2. Display all rows and selected columns of table Installment.
3. Display selected rows and selected columns of table Account.
4. Display selected rows and all columns of table loan.
5. Show the structure of the table loan, account and transaction.

PRACTICAL-2

Table: **ACCOUNT**.

1. Insert the following records if you have not inserted in PRACTIAL - 1

Acc_no	Name	City	Balance	Loan_taken
A001	Patel Jigar	Mehsana	50000	YES
A002	Patel Ramesh	Mehsana	50000	Yes
A003	Dave Hardik	Ahmedabad	75000	NO
A004	Soni Hetal	Ahmedabad	100000	NO
A005	Soni Atul	Vadodara	100000	YES

2. Change the name 'patel jigar' to 'patel hiren'.

3. Change the name and city where account number is A005. (new name = 'kothari nehal' and new city = 'patan').

4. Display only those records where loan taken status is 'YES'.

5. Add the new column (address varchar2 (20)) into table ACCOUNT.

6. Create another table ACCOUNT_TEMP (acc_no, name, balance) from table ACCOUNT.

7. Rename the table ACCOUNT to ACCOUNT_MASTER.

8. Update the column balance for all the account holders. (Multiply the balance by 2 for each account holders)

9. Describe the structure of table ACCOUNT.

10. Delete the records whose account no is A004.

Table: **LOAN**.

1. Insert the following Records if you have not inserted in PRACTICAL-1

Loan_no	Acc_no	Loan_amt	Interest_rate	Loan_date	Remaining_loan
L001	A001	100000	7	1-jan-04	75000
L002	A002	300000	9	18-may-04	150000
L003	A005	500000	11	15-june-04	300000

2. for each loan holders Add 100000 Rs. Amount into the column loan_amt.

3. for each loan holders Increase the interest rate 2%.

4. Create another table LOAN_TEMP (loan_no, Acc_no, loan_amt, loan_date) from The table LOAN.

5. Display only those records where loan holder taken a loan in month of   January.

6. Modify the structure of table LOAN by adding one column credit_no varchar2 (4).

7. Display the Loan amount*2 of table LOAN.

8. Display the records of table LOAN by date wise in ascending order.

9. Display the records of table LOAN by account number wise in descending Order.

10. Increase the size 5 to 7 of column acc_no.

Table: **INSTALLMENT**.

    1.   Insert following Records if you have not inserted in PRACTICAL-1.

Loan_no	Inst_no	Inst_Date	Amount
L001	I001	2-Feb-04	15000
L002	I002	18-June-04	20000
L003	I003	15-July-04	20000

    2.   Change the Inst_Date '2-Feb-04' to '3-Mar-04'.

    3.   Reduce 5000 amount from all Installment holders.

    4.   Add the amount 5000 where loan no is 'L003' and 'L002'.

    5.   Change the column size of 5 to 7 where column name is Loan_no.

    6.   Decrease the column size 5 to 4 where column name Inst_no.

    7. Show the structure of the Table.

    8. Change the amount 15000 to 5000 where loan number is L001

    9. Perform delete operation. (Delete only particular one record)

   10. Only create a structure of table installment1 from table installment.

Table: **TRANSACTION**.

    1.   Insert a Following Records if you have not inserted in PRACTICAL-1.

Acc_no	Trans_Date	Amt	Type_of_tr	Mode_of_pay
A001	3-may-04	10000	D	Cash
A002	5-july-04	5000	W	Check
A003	12-Aug-04	25000	D	Check
A004	15-may-04	30000	D	Check
A005	22-oct-04	15000	W	Cash

    2.   Insert any duplicate value and display all the records without any duplicate rows.

    3.   Select all the records in descending order(account number wise).

    4.   Display amt, date, and type of transaction by date wise.

    5.   Create another table TRANSACTION_TEMP from this table.

    6.   Create a another table TRANS_TEMP by change the column name acc_no to account_no.

    7.   Delete a table TRANSACTION_TEMP.

    8.   Rename the table TRANSACTION to TRANS.

9. Only create a structure of table transaction1 from table transaction.
10. Display account number where type of transaction is 'D'.

PRACTICAL-3

Note: Bold and Underline column name indicates a primary key

Create a table **ACCOUNT**.

Column name	Data Type	Size	Attributes
Acc_no	Varchar2	5	Primary key/first letter must start with 'A'
Name	Varchar2	30	NOT NULL
City	Varchar2	20	NOT NULL
Balance	Number	10,2	Balance >=500
Loan_taken	Varchar2	3	Values('NO','YES')

1. Insert the records using Practical list-1.

Create a Table **LOAN**.

Column Name	Data Type	Size	Attributes
Loan_no	Varchar2	5	Primary Key / first letter must start with 'L'
Acc_no	Varchar2	5	Foreign key References Acc_no of account table
Loan_amt	Number	10,2	NOT NULL
Interest_rate	Number	5,2	NOT NULL
Loan_date	Date		
Remaining_loan	Number	10,2	Remaining loan<loan amount

1. Insert the records using Practical list-1.

Create a table **INSTALLMENT**.

Column Name	Data Type	Size	Attributes
Loan_no	Varchar2	5	Foreign key References Loan_no of Loan table
Inst_no	Varchar2	5	first letter must start with 'I'
IDate	Date		NOT NULL
Amount	Number	10,2	NOT NULL

1. Insert the records using Practical list-1.

Create a Table **TRANSACTION**.

Column Name	Data Type	Size	Attributes
Acc_no	Varchar2	5	Foreign key References Acc_no of account table
Trans_Date	Date		NOT NULL
Amt	Number	10,2	NOT NULL
Type_of_tr	Char	1	Values in ('D','W')

Mode_of_pay	Varchar2	10	Values in ('cash','check')

1. Insert the records using Practical list-1.

Using Operator: NOT,BETWEEN,NOT BETWEEN,IN,NOT IN

1. Retrieve specified information for the account holder who are not in 'Ahmedabad'.
2. Retrieve specified information for the account holder who are not in 'Ahmedabad' or 'Vadodara'.
3. Retrieve those records of Account holder whose balance between is 50000 and 100000.
4. Retrieve those records of Account holder whose balance not between is 50000 and 100000.
5. Display only those records whose amount is 5000, 25000, 30000.
6. Display only those records whose amount not in 5000, 25000, 30000.
7. Display System date.
8. Find the date,15 days after today's date.
9. Perform following operation using DUAL table.
 5*5,34+34,1000/300,length of 'uvpce',display only month of systemdate
10. Find the date, 20 days before today's date.

Function Based Queries.

1. Find the total transaction amount of account holder from transaction table.
2. Find minimum amount of transaction.
3. Find maximum amount of transaction.
4. Count the total account holders.
5. Count only those records whose made of payment is 'cash'.
6. Count only those records whose transaction made in the month of 'MAY'.
7. Find the average value of transaction.
8. Display the result of 4 rest to 4.
9. Find the square root of 25.
10. Write the query for the following Function.
LOWER,INITCAP,UPPER,SUBSTR,LENGTH,LTRIM,RTRIM,LPAD,RPAD

CONSTRAINTS Based queries.

Create a table:**STUDENT**

Name of column	Type and Size
Rollno	Varchar2(6)
Name	Varchar2(20)
Branch	Varchar2(6)
Address	Varchar2(20)

1. Add PRIMARY KEY (roll no) and provide constraint name PRIM_rollno.

2. Add NOT NULL constraint to name,branch for student table.
3. Add check constraint and check name is in capital letter.
4. Drop the primary key.
5. Drop the constraint.

Create a Table **REGISTER.**

Name of column	Type and Size
Rollno	Varchar2(6)
Name	Varchar2(20)

1. Provide foreign key references rollno of student table.
2. Add check constraint to check name's first letter is always capital.
3. Add NOT NULL constraint to name of register table.
4. Drop foreign key of REGISTER table.
5. Drop NOT NULL constraint.

PRACTICAL-4

NOTE: for following queries use TABLES of PRACTICAL-1

1. Display the sum of balance of account holders who's live in same city 'Mehsana' using group by clause.

2. Display the information about account where balance is less than total balance of all account holders.

3. Displays the information of account holders whose loan amount and balance both are same.

4. Display the name of city, remaining loan amount, account, date of loan and loan number of account holders.

5. Display name of account holder, installment number and installment amount Whose loan number is 'L001'.

6. Display name of account holder, city, loan amount and installment amount.

7. Display the balance of account holders whose balance and remaining loan both are same.

8. List of all account holders' information whose balance is same as loan amount.

9. Display the amount of transaction, name of account holders, account number and mode of payment whose mode of payment is 'CHEQUE'.

10. Display account no, loan amount, amount of transaction.

11. List of installment information whose amount is less than average amount of transaction.

12. Display the sum of installment amount and transaction amount.

13. Display the balance and amount of transaction group by amount and balance.

14. List of installment number and account number of account holders.

15. Display loan amount, transaction amount and mode of payment where transaction date and loan taken date both are done in month of 'MAY'.

16. Display all the information of installment and transaction where installment date and transaction date both are done in month of 'JULY'.

17. Display the last three row of account table.

18. Display the balance, mode of payment, loan taken status whose mode of payment is 'CHEQUE' and loan taken is 'YES'.

19. Retrieve only rows 2 to 5 from account table.

PRACTICAL-5

TABLE: **SALESMEN**

Column Name	Data Type	Size	Attributes
SNUM	Varchar2	6	Primary key/first letter must start with 'S'
SNAME	Varchar2	20	Not null
CITY	Varchar2	15	
COMM	Number	5,2	

Insert the following records:

SNUM	SNAME	CITY	COMM
S1001	Piyush	London	0.12
S1002	Niraj	San jose	0.13
S1003	Miti	London	0.11
S1004	Rajesh	Barcelona	0.15
S1005	Haresh	New york	0.10

| S1006 | Ram | Bombay | 0.10 |
| S1007 | Nehal | Delhi | 0.09 |

TABLE: **CUSTOMER**

Column Name	Data Type	Size	Attributes
CNUM	Varchar2	6	Primary key/first letter must start with 'C'
CNAME	Varchar2	20	Not null
CITY	Varchar2	15	
RATING	Number	5	
SNUM	Varchar2	6	

Insert the following records

CNUM	CNAME	CITY	RATING	SNUM
C2001	Hardik	London	100	S1001
C2002	Geeta	Rome	200	S1003
C2003	Kavish	San jose	200	S1002
C2004	Dhruv	Berlin	300	S1002
C2005	Pratham	London	100	S1001
C2006	Vyomesh	San jose	300	S1007
C2007	Kirit	Rome	100	S1004

TABLE: **ORDER**

Column Name	Data Type	Size	Attributes
ONUM	Varchar2	6	Primary key/first letter must start with 'O'
AMT	Number	10,2	Not null
ODATE	Date		
CNUM	Varchar2	6	
SNUM	Varchar2	6	

Insert the following records

ONUM	AMT	ODATE	CNUM	SNUM
O3001	18.69	10-Mar-90	C2008	S1007

O3003	767.19	10-Mar-90	C2001	S1001
O3002	1900.10	03-Oct-90	C2007	S1004
O3005	5160.45	04-Oct-90	C2003	S1002
O3006	1098.16	10-Mar_90	C2008	S1007
O3009	1713.23	10-April-90	C2002	S1003
O3007	75.75	10-April-90	C2004	S1002
O3008	4723.00	10-May-90	C2006	S1001
O3010	1309.95	10-May-90	C2004	S1002
O3011	9891.88	10-June-90	C2006	S1001

Perform following queries.

SELECT

1. Display all the information of salesmen.
2. Display snum,sname,city from salesmen table.
3. Display odate,snum,onum and amt from orders.
4. Display the information of orders without duplication.
5. List of sname, city from salesmen where city is 'LONDON'.
6. List all records of customers where rating is equal to 100.
7. Write a select command that produces the order number,amount and date for all rows in the order table.
8. Produces all rows from the customer table for which the salesperson's number is S1001.
9. Display the salesperson table with the column in the following order: city,sname,snum,comm.
10. Write a select command that produces the rating followed by the name of each customer in SAN JOSE.
11. Display SNUM values of all salesmen without any repeat.

OPERATORS

12. List all customers with a rating above 200.
13. List all customers in SAN JOSE who have a rating above 200.
14. List all customers who were either located in SAN JOSE or had a rating above 200.
15. List of all customers who were either located in SAN JOSE or not rating above 200.
16. List of all customers who were not located in SAN JOSE or rating is not above 200.
17. Write a query that will give you all orders for more than $1000.

18. Write a query that will give you the names and cities of all salesmen in LONDON with a commission above 0.10.

19. Write a query on the customers table whose output will exclude all customers with a rating <= 100 and they are located in ROME.

SPECIAL OPERATORS

20. Display all salesmen that were located in either BARCELONA or LONDON(use IN keyword).

21. Find all customers matched with salesmen S1001,S1007 and S1004.

22. Display all salesmen with commission between 0.10 and 0.12.

23. Select all customers whose names fall in a 'A' and 'G' alphabetical.

LIKE OPERATORS.

24. List all the customers whose names begin with 'G'.

25. List all salesmen whose sname start with letter 'P' and end letter is 'H'.

NULL OPERATORS.

26. Find all records in customer table with NULL values in the city column.

27. Write a two queries that will produce all orders taken on October 3rd or 4th ,1990

    (use IN operator and Use BETWEEN operator)

28. Write a query that selects all of the customers matched with S1001 and S1002.

29. Write a query that will produce all of the customers whose names begin with a letter from A to H.

30. Write a query that selects all customers whose names begin with 'C'.

31. Write a query that selects all orders without ZEROS or NULLS in amt field..

FUNCTIONS

32. Display sum of amt,average of orders.

33. To count the numbers of salesmen without duplication in the orders tables.

34. Count the rating of customers (with NULL and without NULL).

35. Find the largest order taken by each salesperson.(hint: use group by)

36. Find the largest order taken by each salesperson on each date.

37. Find out which day had the higher total amount ordered.

38. Write a query that counts all orders for October 3rd.

39. Write a query that counts the number of different non-NULL city in the customer table.

40. Write a query that selects the first customer in alphabetical order whose name begin with 'G'.

41. write a query that selects each customers smallest order.

42. Write a query that selects the highest rating in each city.

43. Write a query that counts the number of salesmen registering orders for each day(if a salesperson has more than one order on a given day , he or she should be counted only once)

44. Display all the information in descending orders(use column CNUM).

45. Display all the information in descending orders(use column CNUM,AMT).

46. Display sname and comm. From salesmen in descending order(in place of column name use column number).

47. Assume each salesperson has a 0.12 commission. Write a query on the orders table that will produce the order number,the salesperson number and the amount of the salesperson's commission for that order.

48. Write a query on the customers table that will find the highest rating in each city. Put the output in this form.

    For the city (city) , the highest rating is: (rating).

49. Write a query that lists customers in descending order of rating. Output the rating field first, followed by the customer's name and number.

50. Write a query that totals the orders for each day and places the results in descending order.

    JOIN

51. Show the names of all customers matched with the salesmen serving them.

52. Write a query that lists each order number followed by the name of the customer who made the order.

53. Write a query that gives the names of both the salesperson and the customer for each order after the order number.

54. Write a query that produces all customers serviced by salesmen with a commission above 0.12. Output the customer's name, the salesperson's name and the salesperson's rate of commission.

55. Write a query that calculates the amount of the salesperson's commission on each order by a customer with a rating above 100.

    OTHERS

56. List all customer located in cities where salesperson 'PIYUSH' has customer.

57. List all salesmen who are living in same city without duplicate rows.

58. Extract all orders of 'PIYUSH'.

59. Extract all orders of LONDON'S salesmen.

60. Find all customers whose cnum is 1000 above than the snum of 'PIYUSH'.

61. Count the no. of customers with the rating above than average Rating of 'LONDON'.

62. Produce the name and rating of all customers who have above average Rating.

63. List all salesmen with customers located in their cities.

64. Select all customers whose rating doesn't match with any rating customer of 'SAN JOSE'.

65. Create a union of two queries that shows the names,cities and ratings of all customers. Those with rating of >=200 should display 'HIGH RATING' and those with <200 should display 'LOW RATING'.

66. Find all customers with orders on 3rd october 1990 using correlate sub query.

67. Find all customers having rating greater than any customer in 'ROME'.

68. Insert a row into salesmen table with the values snum is s1008,salesmen name is 'RAKESH', city is unknown and commission is 14%.

69. Create another table London_staff having same structure as salesmen table.

70. Delete all orders from customer 'PIYUSH' from the order table.

71. Set the ratings of all the customers of PIYUSH to 400.

72. Increase the rating of all the customers in ROME by 100.

73. Double the commission of all salesmen of LONDON.

74. Set ratings for all customers in LONDON to NULL.

75. Delete all salesmen who have at least one customer with a rating of 100 from salesmen table.

Instructional Method:

a. The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

b. The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.

c. Practical examination will be conducted at the end of semester for evaluation of performance of students in laboratory.

d. Students will use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory

Supplementary Resources:

1. https://apex.oracle.com/en/

2. https://ilearning.oracle.com/

3. https://lagunita.stanford.edu/courses/DB/2014/SelfPaced/about