

GUJARAT TECHNOLOGICAL UNIVERSITY (GTU)

Competency-focused Outcome-based Green Curriculum-2021 (COGC-2021) Semester-III

Course Title: Database Management
(Course Code: 4331603)

| Diploma programme in which this course is offered | Semester in which offered |
|---|---------------------------|
| Information Technology | Third |

1. RATIONALE

Database management course introduces students to database design using various models, SQL commands, techniques and operation. This helps students to design ER-models to represent simple databases and convert them into relational tables, populate relational databases and formulate SQL queries on data. Students will improve database design through normalization. Students will understand how database systems must provide for the safety of the stored information, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results due to multiple users concurrently accessing the same data.

2. COMPETENCY

The purpose of this course is to help the student to attain the following industry identified competency through various teaching learning experiences:

- a. Design database for a given application.
- b. Perform various database operations.

3. COURSE OUTCOMES (COs)

The practical exercises, the underpinning knowledge and the relevant soft skills associated with this competency are to be developed in the student to display the following COs:

- a) Explain the basic concepts of databases.
- b) Design database using Entity relationship approach.
- c) Implement Relational algebra in a database.
- d) Apply SQL Commands for creating, manipulating and controlling databases.
- e) Apply concepts of normalization to design an optimal database.
- f) Explain transaction management concepts for concurrent use of database.

4. TEACHING AND EXAMINATION SCHEME

| Teaching Scheme (In Hours) | | | Total Credits (L+T/2+P/2) | Examination Scheme | | | | Total Marks |
|-------------------------------|---|---|------------------------------|--------------------|-----|-----------------|-----|----------------|
| | | | | Theory Marks | | Practical Marks | | |
| L | T | P | C | CA | ESE | CA | ESE | |
| 3 | 0 | 4 | 5 | 30* | 70 | 25 | 25 | 150 |

(*): Out of 30 marks under the theory CA, 10 marks are for assessment of the micro-project to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be

taken during the semester for the assessing the attainment of the cognitive domain UOs required for the attainment of the COs.

Legends: **L**-Lecture; **T** – Tutorial/Teacher Guided Theory Practice; **P** -Practical; **C** – Credit, **CA** - Continuous Assessment; **ESE** -End Semester Examination.

5. SUGGESTED PRACTICAL EXERCISES

The following practical outcomes (PrOs) are the sub-components of the COs. These PrOs need to be attained to achieve the COs.

| Sr. No. | Practical Outcomes (PrOs) | Unit No. | Approx. Hrs. required | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------|---|-----------|-----------------------|-------------|---------|-------------|------|-------------|---------|-------|-----|------|--------------|---------|-------|-----|------|-------------|-----------|-------|----|------|-----------|-----------|--------|----|------|----------|----------|--------|-----|---|---|
| 1 | Prepare a report on current database trends, architecture and tools. | 1 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Draw ER-Diagram for Banking Management System and convert it relational schema | 2 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | <p>Aim: Design the below given schemas using SQL Command - "Create". Decide the appropriate data type for each column.</p> <p>(a) Create a table ACCOUNT with column account number, name, city, balance, loan taken.</p> <p>(b) Create a LOAN table with column loan number, account number, loan amount, interest rate, loan date, remaining loan.</p> <p>(c) Create table INSTALLMENT with column loan number, installment number, installment date and amount.</p> <p>(d) Create table TRANSACTION with column account number, transaction date, amount, type of transaction, mode of payment.</p> <p>(e) Show the structure of above tables using "Describe" command.</p> | 3 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | <p>Aim: Insert data in above tables using SQL Command "Insert".</p> <p>(a) Account</p> <table border="1" data-bbox="331 1435 1106 1890"> <thead> <tr> <th>acc_no</th> <th>Name</th> <th>City</th> <th>Balance</th> <th>Loan _taken</th> </tr> </thead> <tbody> <tr> <td>A001</td> <td>Patel Jigar</td> <td>Mehsana</td> <td>50000</td> <td>YES</td> </tr> <tr> <td>A002</td> <td>Patel Ramesh</td> <td>Mehsana</td> <td>50000</td> <td>YES</td> </tr> <tr> <td>A003</td> <td>Dave Hardik</td> <td>Ahmedabad</td> <td>75000</td> <td>NO</td> </tr> <tr> <td>A004</td> <td>SoniHetal</td> <td>Ahmedabad</td> <td>100000</td> <td>NO</td> </tr> <tr> <td>A005</td> <td>SoniAtul</td> <td>Vadodara</td> <td>100000</td> <td>YES</td> </tr> </tbody> </table> | 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 | SoniHetal | Ahmedabad | 100000 | NO | A005 | SoniAtul | Vadodara | 100000 | YES | 3 | 2 |
| 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 | SoniHetal | Ahmedabad | 100000 | NO | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A005 | SoniAtul | Vadodara | 100000 | YES | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Sr. No. | Practical Outcomes (PrOs) | Unit No. | Approx. Hrs. required | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------|--|------------|-----------------------|-------------|----------------|-------------|------|----------|-------|---|------|------|-----------|------|---|--------|------|-----------|-------|---|--------|------|-----------|-------|---|--------|------|-----------|-------|---|------|---------|--------|----------|---------------|-----------|----------------|------|------|--------|---|----------|-------|------|------|--------|---|-----------|--------|------|------|--------|----|------------|--------|---------|---------|------|--------|------|------|----------|-------|------|------|------------|-------|------|------|------------|-------|--|--|
| | <p>(b) Transaction</p> <table border="1" data-bbox="325 353 1115 790"> <thead> <tr> <th>Acc_no</th> <th>Tr_date</th> <th>Amt</th> <th>Type_of_tr</th> <th>Mode_of_pay</th> </tr> </thead> <tbody> <tr> <td>A001</td> <td>1-may-20</td> <td>10000</td> <td>D</td> <td>Cash</td> </tr> <tr> <td>A002</td> <td>3-july-20</td> <td>5000</td> <td>W</td> <td>Cheque</td> </tr> <tr> <td>A003</td> <td>12-Aug-20</td> <td>25000</td> <td>D</td> <td>Cheque</td> </tr> <tr> <td>A004</td> <td>15-may-20</td> <td>30000</td> <td>D</td> <td>Cheque</td> </tr> <tr> <td>A005</td> <td>22-oct-20</td> <td>15000</td> <td>W</td> <td>Cash</td> </tr> </tbody> </table> <p>(c) Loan</p> <table border="1" data-bbox="288 835 1153 1133"> <thead> <tr> <th>loan_no</th> <th>acc_no</th> <th>loan_amt</th> <th>Interest_rate</th> <th>loan_date</th> <th>remaining_loan</th> </tr> </thead> <tbody> <tr> <td>L001</td> <td>A001</td> <td>100000</td> <td>7</td> <td>1-jan-20</td> <td>75000</td> </tr> <tr> <td>L002</td> <td>A002</td> <td>300000</td> <td>9</td> <td>18-may-20</td> <td>150000</td> </tr> <tr> <td>L003</td> <td>A005</td> <td>500000</td> <td>11</td> <td>15-june-20</td> <td>300000</td> </tr> </tbody> </table> <p>(d) Installment</p> <table border="1" data-bbox="429 1178 1013 1469"> <thead> <tr> <th>Loan_no</th> <th>Inst_no</th> <th>Date</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>L001</td> <td>I001</td> <td>2-Feb-04</td> <td>15000</td> </tr> <tr> <td>L002</td> <td>I002</td> <td>18-June-04</td> <td>20000</td> </tr> <tr> <td>L003</td> <td>I003</td> <td>15-July-04</td> <td>20000</td> </tr> </tbody> </table> | Acc_no | Tr_date | Amt | Type_of_tr | Mode_of_pay | A001 | 1-may-20 | 10000 | D | Cash | A002 | 3-july-20 | 5000 | W | Cheque | A003 | 12-Aug-20 | 25000 | D | Cheque | A004 | 15-may-20 | 30000 | D | Cheque | A005 | 22-oct-20 | 15000 | W | Cash | loan_no | acc_no | loan_amt | Interest_rate | loan_date | remaining_loan | L001 | A001 | 100000 | 7 | 1-jan-20 | 75000 | L002 | A002 | 300000 | 9 | 18-may-20 | 150000 | L003 | A005 | 500000 | 11 | 15-june-20 | 300000 | Loan_no | Inst_no | Date | Amount | L001 | I001 | 2-Feb-04 | 15000 | L002 | I002 | 18-June-04 | 20000 | L003 | I003 | 15-July-04 | 20000 | | |
| Acc_no | Tr_date | Amt | Type_of_tr | Mode_of_pay | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A001 | 1-may-20 | 10000 | D | Cash | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A002 | 3-july-20 | 5000 | W | Cheque | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A003 | 12-Aug-20 | 25000 | D | Cheque | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A004 | 15-may-20 | 30000 | D | Cheque | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A005 | 22-oct-20 | 15000 | W | Cash | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| loan_no | acc_no | loan_amt | Interest_rate | loan_date | remaining_loan | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L001 | A001 | 100000 | 7 | 1-jan-20 | 75000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L002 | A002 | 300000 | 9 | 18-may-20 | 150000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L003 | A005 | 500000 | 11 | 15-june-20 | 300000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Loan_no | Inst_no | Date | Amount | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L001 | I001 | 2-Feb-04 | 15000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L002 | I002 | 18-June-04 | 20000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L003 | I003 | 15-July-04 | 20000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | <p>Aim: Retrieve data from tables in Practical 2 using Data SQL command- "Select"</p> <p>(a) Display all rows and all columns of table Transaction.</p> <p>(b) Display all rows and selected columns of table Installment.</p> <p>(c) Display selected rows and selected columns of table Account.</p> <p>(d) Display selected rows and all columns of table loan.</p> <p>(e) Display the branch wise balance from account table.</p> <p>(f) Display list of those branches that have balance greater than 1 Lakh rupees.</p> <p>(g) Display the list of customers in descending order of their name from account table.</p> <p>(h) Display those records where mode of payment is "cheque"</p> | 3 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | <p>Aim: Write SQL queries to use Update, alter, rename, delete, truncate and distinct.</p> | 3 | 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Sr. No. | Practical Outcomes (PrOs) | Unit No. | Approx. Hrs. required |
|---------|--|----------|-----------------------|
| | <p>Table: ACCOUNT.</p> <p>(a)Change the name 'pateljigar' to 'patelhiren'.</p> <p>(b)Change the name and city where account number is A005. (new name = 'kotharinehal'and new city = 'patan').</p> <p>(c)Display only those records where loan taken status is 'YES'.</p> <p>(d)Add the new column (address varchar2 (20)) into table ACCOUNT.</p> <p>(e)Create another table ACCOUNT_TEMP (acc_no, name, balance) from table ACCOUNT.</p> <p>(f)Rename the table ACCOUNT to ACCOUNT_MASTER.</p> <p>(g)Update the column balance for all the account holders. (Multiply the balance by 2 foreach account holders)</p> <p>(h)Delete the records whose account no is A004</p> <p>Table: LOAN.</p> <p>(a)For each loan holders Add 100000 Rs. Amount into the column loan_amt.</p> <p>(b)For each loan holders Increase the interest rate 2%.</p> <p>(c)Display only those records where loan holder taken a loan in month of January.</p> <p>(d)Modify the structure of table LOAN by adding one column credit_no varchar2 (4).</p> <p>(e)Display the Loan amount*2 of table LOAN.</p> <p>(f)Display the records of table LOAN by date wise in ascending order.</p> <p>(g)Display the records of table LOAN by account number wise in descending Order.</p> <p>(h)Increase the size 5 to 7 of column acc_no.</p> <p>Table: INSTALLMENT.</p> <p>(a)Change the Inst_Date '2-Feb-04' to '3-Mar-04'.</p> <p>(b)Reduce 5000 amount from all Installment holders.</p> <p>(c)Add the amount 5000 where loan no is 'L003' and 'L002'.</p> <p>(d)Change the column size of 5 to 7 where column name is Loan_no.</p> <p>(e)Delete row where inst_no is '1001'.</p> <p>(f)Only create a structure of table installment1 from table installment.</p> <p>Table: TRANSACTION.</p> <p>(a)Insert any duplicate value and display all the records without any duplicate rows.</p> <p>(b)Select all the records in descending order(account number wise).</p> <p>(c)Display amt, date, and type of transaction by date wise.</p> | | |

| Sr. No. | Practical Outcomes (PrOs) | Unit No. | Approx. Hrs. required |
|---------|---|----------|-----------------------|
| | (d)Delete a table TRANSACTION_TEMP. (e)Display account number where type of transaction is 'D'. | | |
| 7 | Aim: Write SQL queries to use various date functions. (a)Add 3 months in current date. (b)Display months between date: '17-02-04' and '17-02-05'. (c)Display last date of month '2-feb-06'. (d)If the date is 31-jul-13 then find out the date of friendship day. | 3 | 2 |
| 8 | Aim: Write SQL queries to use various numeric functions (a)Display integer value of 125.25. (b)Display absolute value of(-15) (c)Display ceil value of 55.65 (d)Display floor value of 100.2 (e)Display the square root of 16. (f)Display square root of 20. (g)Display result of 12 raised to 6. (h)Display result of 24 mod 2. (i)Show value of e3. (j)2 is angle in radius find out cos, sin, tan value of 2. (k)Show output of sign(-25), sign(25), sign(0). | 3 | 2 |
| 9 | Aim: Write SQL queries to use various character functions. (a)Find out length of string "hello world". (b)Change the case of 'HELLO WORLD' string to lower case. (c)Change the case of 'hello world' to upper case. (d)Display each word initial letter as capital letter : government polytechnic for girls. (e)Find 'put' from string 'computer'. (f)Add 10 star on left side of India string. (g)Add 10 star on right side of India string. (h)Trim 'ion' from 'information'. (i)Trim 'info' from 'information'. | 3 | 2 |
| 10 | Aim: Write SQL queries to use various conversion functions (a)Convert 100000 in format 99,99,99 (b)Convert today's date to MM-DD-YY format | 3 | 2 |
| 11 | Aim: Write SQL queries to use various group function and operators using tables created in Practical 1. (a)Retrieve specified information for the account holder who are not in 'Ahmedabad'. (b)Retrieve specified information for the account holder who arenot in 'Ahmedabad'or 'Vadodara'. (c)Retrieve those records of Account holder whose balance between is 50000 and 100000. (d)Retrieve those records of Account holder whose balance not between is 50000 and 100000. (e)Display only those records whose amount is 5000, 25000, | 3 | 4 |

| Sr. No. | Practical Outcomes (PrOs) | Unit No. | Approx. Hrs. required |
|---------|--|----------|-----------------------|
| | 30000. (f) Display only those records whose amount not in 5000, 25000, 30000. (g) Find the total transaction amount of account holder from transaction table. (h) Find minimum amount of transaction. (i) Find maximum amount of transaction. (j) Count the total account holders. (k) Count only that record that's made of payment is 'cash'. (l) Count only those records whose transaction made in the month of 'MAY' (m) Find the average value of transaction. (n) Display total balance for each branch from account table. (o) Display total balance for account in Ahmadabad city. (p) Find total amount of mode of pay 'cash' from transaction table. | | |
| 12 | Aim: Write SQL query for set operators and join operations. (Use tables of Practical 1) (a) Display all customer account number who have account or taken loan from bank. (b) Display all customer account number who have account and also taken loan from bank. (c) Find the list of all account number of customer who have no loan in bank. (d) Display name of all customer whose remaining loan amount is greater than 50000. (e) Display account number of customer who have given installment of their loan on date 18-June-04. (f) Display name of all customer who have selected mode of payment as cash. | 3 | 4 |
| 13 | Aim: Apply the concept of integrity/data constraints while creating/altering a table: (a) Create table SalesPeople where Snum varchar2(4) P.K (first letter should start with S) , Sname varchar2(20) NOT NULL , City Varchar2(15), Mobile_No Number(10). (b) Create table Customer where cnum varchar2(4) P.K (first letter should start with C), Cname varchar2(20) NOT NULL , city varchar2(20), Rating number(3) DEFAULT 10, Snum number(4) F.K (where snum refers salespeople table) (c) Create table Order where Order_No number(4) P.K , Amount number(5), odate varchar2(10), cnum varchar2(4) F.K , (where cnum refers customer table), snum varchar2(4) F.K (where snum refers Salespeople table) | 3 | 2 |
| 14 | Aim: Write SQL queries for CREATE USER, GRANT, REVOKE AND DROP USER command. | 3 | 2 |

| Sr. No. | Practical Outcomes (PrOs) | Unit No. | Approx. Hrs. required |
|---------|--|----------|-----------------------|
| | (a) Create a new user User1 . (b) Grant connect, resource and dba rights to user1. (c) Grant all permissions of account table to user1. (d) Remove select permission on account table for user1. (e) Delete user1. | | |
| 15 | Write a program in PL/SQL to show the user of Cursor. | 3 | 2 |
| 16 | Write a program in PL/SQL to show the user of Stored Procedures | 3 | 2 |
| 17 | Write a program in PL/SQL to show the user of Stored Function. | 3 | 2 |
| 18 | Write a program in PL/SQL to show the user of Database Triggers | 3 | 2 |
| 19 | Aim: Normalization of database. Consider the following relational schema: a) Employee(eid, ename, salary, mngr id, mngrname, mngrage, ecity) b) Student(no, name, bdate, city, semester, subname, submarks, percentage) For above schemas, answer the following questions:- a) Determine the primary key for given relation. b) Is there any pitfall in design of relation schema? Explain it. c) Find out various dependencies among given attributes of given relation schema. d) Normalize given relation schema up to possible normal form | 4 | 4 |
| 20 | Prepare a report on transaction management concepts for concurrent access of database by multiple users. | 5 | 4 |
| | Total | | 56 |

Note

*i. More **Practical Exercises** can be designed and offered by the respective course teacher to develop the industry relevant skills/outcomes to match the COs. The above table is only a suggestive list.*

*i. The following are some **sample** 'Process' and 'Product' related skills (more may be added/deleted depending on the course) that occur in the above listed **Practical Exercises** of this course required which are embedded in the COs and ultimately the competency..*

| Sr. No. | Sample Performance Indicators for the PrOs | Weightage in % |
|---------|---|----------------|
| 1 | Explain the basic concepts of databases | 8% |
| 2 | Design ER Diagram for a given problem and map it to relational schema | 12% |
| 3 | Perform SQL queries to create, store data or retrieve data from database. | 60% |
| 4 | Normalize the given database. | 10% |
| 5 | Explain transaction management concepts. | 10% |
| | Total | 100 |

6. MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

These major equipment with broad specifications for the PrOs is a guide to procure them by the administrators to usher in uniformity of practical in all institutions across the state.

| Sr. No. | Equipment Name with Broad Specifications | PrO. No. |
|---------|---|----------|
| 1 | Desktop Computer P-IV processor or higher | All |
| 2 | Oracle latest version | 3 |

7. AFFECTIVE DOMAIN OUTCOMES

The following **sample** Affective Domain Outcomes (ADOs) are embedded in many of the above mentioned COs and PrOs. More could be added to fulfil the development of this competency.

- a) Work as a database administrator.
- b) Follow ethical practices.

The ADOs are best developed through the laboratory/field based exercises. Moreover, the level of achievement of the ADOs according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

- i. 'Valuing Level' in 1st year
- ii. 'Organization Level' in 2nd year.
- iii. 'Characterization Level' in 3rd year.

9. UNDERPINNING THEORY

Only the major Underpinning Theory is formulated as higher level UOs of *Revised Bloom's taxonomy* in order development of the COs and competency is not missed out by the students and teachers. If required, more such higher level UOs could be included by the course teacher to focus on attainment of COs and competency.

| Unit | Unit Outcomes (UOs) (4 to 6 UOs at Application and above level) | Topics and Sub-topics |
|--|--|--|
| Unit – I Introduction to Database Systems | 1a. Describe basic concepts of database and its applications 1b. Differentiate between databases and file systems | 1.1 Introduction 1.1.1 Data and Information 1.1.2 Database and Database Management System 1.1.3 Metadata 1.1.4 Data items, fields & records 1.1.5 Data Dictionary 1.2 Purpose of Database System 1.3 File oriented System versus database system 1.4 Application of DBMS 1.5 Database Administrator 1.5.1 Roles and responsibilities of DBA 1.6 Schema, Sub-Schema, Instances |

| Unit | Unit Outcomes (UOs) (4 to 6 UOs at Application and above level) | Topics and Sub-topics |
|--|--|---|
| | 1c. Explain the concepts of data abstraction and data independence 1d. Explain the different types of data models | 1.7 Data Abstraction 1.7.1 Internal Level 1.7.2 Conceptual Level 1.7.3 External Level 1.8 Data Independence 1.9 Database Architecture 1.9.1 ER Model 1.9.2 Relational Model 1.9.3 Object oriented data model 1.9.4 Network Data Model 1.9.5 Hierarchical Data Model |
| Unit – II ER Model and Relational Algebra | 2a. Design ER Diagram for an application 2b. Convert ER Diagram to database | 2.1 Basic concepts of E-R 2.1.1 Entity 2.2 Attributes 2.2.3 Relationship 2.2.3.1 Participation 2.2.3.2 Recursive relationships 2.2.3.3 Degree of relationshipset 2.2 Mapping Cardinality 2.3 Key Constraints 2.3.1 Primary 2.3.2 Foreign 2.3.3 Super 2.3.4 Candidate 2.4 ER Diagrams 2.5 Weak Entity Sets 2.6 Enhanced ER Model 2.6.1 Subclass & Super Class 2.6.2 Generalization 2.6.3 Specialization 2.6.4 Aggregation 2.7 Converting ER Diagrams to database |
| | 2c. Write relational algebra queries for an application. | 2.8 Structure of Relational Databases 2.9 Operations and Queries of Relational algebra 2.9.1 The Select Operation 2.9.2 The Project Operation 2.9.3 The Union Operation 2.9.4 The Set Difference(Minus) Operation 2.9.5 The Set Intersection Operation 2.9.6 The Rename Operation 2.9.7 The Cartesian Product Operation |

| Unit | Unit Outcomes (UOs) (4 to 6 UOs at Application and above level) | Topics and Sub-topics |
|---|---|---|
| | | 2.9.8 The Natural Join Operation 2.9.9 The Outer Join Operation 2.9.10 The Assignment Operation |
| Unit-III Structured Query Language | 3a. Write queries to create database, store and retrieve data from database | 3.1 SQL Data types 3.2 DDL Commands 3.2.1 create 3.2.2 alter 3.2.3 truncate 3.2.4 drop 3.3 DML Commands 3.3.1 insert 3.3.2 select 3.3.3 update 3.3.4 delete 3.4 Privilege command 3.4.1 grant 3.4.2 revoke 3.5 SQL views |
| | 3b. Use SQL functions for different Operations | 3.6 Single row function 3.7 Date functions 3.8 Numeric & Character function 3.9 Conversion function 3.10 Miscellaneous function 3.11 Group function |
| | 3c. Write SQL sub-queries to retrieve data | 3.13 Operators 3.14 Arithmetic 3.15 Comparison 3.16 Logical Group by 3.17 Having and order by clause 3.18 Set operators 3.18.1 Union 3.18.2 union all 3.18.3 intersect 3.18.4 minus 3.19 Joins 3.19.1 simple join 3.19.2 equi join 3.19.3 non equi join 3.19.4 self join 3.19.5 outer join |
| | 3d. SQL Constraints | 3.20 Need of Constraints 3.21 Domain Integrity constraints 3.21.1 Not null |

| Unit | Unit Outcomes (UOs) (4 to 6 UOs at Application and above level) | Topics and Sub-topics |
|--|--|--|
| | | 3.21.2 Check 3.22 Entity Integrity constraints 3.22.1 Unique 3.22.2 Primary key 3.23 Referential integrity Constraints 3.23.1 Foreign key 3.23.2 Reference key |
| | 3e. Apply PL/SQL Concepts | 3.24 Cursors 3.25 Stored Procedures 3.26 Stored Function 3.27 Database Triggers |
| Unit– IV Refining database design through Normalization | 4a. Describe the Normalization process. 4b. Design database using different levels of normalization | 4.1 Importance of Normalization 4.2 Functional Dependencies 4.2.1 Partial Functional Dependency 4.2.2 Full Functional Dependency 4.2.3 Transitive Dependency 4.3 Normal Forms 4.3.1 First Normal Form 4.3.2 Second Normal Form 4.3.3 Third Normal Form |
| Unit– V Transaction Management | 5a. Apply transaction concepts 5b. Use Serializability 5c. Understand concurrent transaction problems and their solutions. | 5.1 Transaction concepts, properties of transactions. 5.2 Serializability of transactions 5.3 Concurrent executions of transactions and related problems. 5.4 Locking mechanism, solution to concurrency related problems, deadlock. |

Note: The UOs need to be formulated at the 'Application Level' and above of 'Revised Bloom's Taxonomy' to accelerate the attainment of the COs and the competency.

10. SUGGESTED SPECIFICATION TABLE FOR QUESTION PAPER DESIGN

| Unit No. | Unit Title | Teaching Hours | Distribution of Theory Marks | | | |
|--------------|--|----------------|------------------------------|-----------|-----------|-------------|
| | | | R Level | U Level | A Level | Total Marks |
| I | Introduction to Database Systems | 06 | 4 | 4 | 2 | 10 |
| II | ER Model and Relational Algebra | 10 | 5 | 5 | 6 | 16 |
| III | Structured Query Language | 16 | 6 | 6 | 10 | 22 |
| IV | Refining database design through Normalization | 04 | 3 | 3 | 6 | 12 |
| V | Transaction Management | 06 | 2 | 4 | 4 | 10 |
| Total | | 42 | 20 | 22 | 28 | 70 |

Legends: R=Remember, U=Understand, A=Apply and above (Revised Bloom's taxonomy)

Note: This specification table provides general guidelines to assist student for their learning and to teachers to teach and question paper designers/setters to formulate test items/questions assess the attainment of the UOs. The actual distribution of marks at different taxonomy levels (of R, U and A) in the question paper may vary slightly from above table.

11. SUGGESTED STUDENT ACTIVITIES

Other than the classroom and laboratory learning, following are the suggested student-related **co-curricular** activities which can be undertaken to accelerate the attainment of the various outcomes in this course: Students should conduct following activities in group and prepare reports of about 5 pages for each activity, also collect/record physical evidences for their (student's) portfolio which will be useful for their placement interviews:

- a) Give a presentation on database case studies.
- b) Visit a software company and discuss their practices adopted for database development.

12. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

These are sample strategies, which the teacher can use to accelerate the attainment of the various outcomes in this course:

- a) Massive open online courses (**MOOCs**) may be used to teach various topics/sub topics.
- b) Guide student(s) in undertaking micro-projects.
- c) '**L**' in **section No. 4** means different types of teaching methods that are to be employed by teachers to develop the outcomes.
- d) About **20% of the topics/sub-topics** which are relatively simpler or descriptive in nature is to be given to the students for **self-learning**, but to be assessed using different assessment methods.
- e) With respect to **section No.11**, teachers need to ensure to create opportunities and provisions for **co-curricular activities**.

13. SUGGESTED MICRO-PROJECTS

Only one micro-project is planned to be undertaken by a student that needs to be assigned to him/her in the beginning of the semester. In the first four semesters, the micro-project are group-based. However, in the fifth and sixth semesters, it should be preferably be **individually** undertaken to build up the skill and confidence in every student to become problem solver so that s/he contributes to the projects of the industry. In special situations where groups have to be formed for micro-projects, the number of students in the group should **not exceed three**.

The micro-project could be industry application based, internet-based, workshop-based, laboratory-based or field-based. Each micro-project should encompass two or more COs which are in fact, an integration of PrOs, UOs and ADOs. Each student will have to maintain a dated work diary consisting of individual contributions in the project work and give a seminar presentation of it before submission. The total duration of the micro-project should not be less than **16 (sixteen) student engagement hours** during the course. The student ought to submit a micro-project by the end of the semester to develop the industry oriented COs.

A suggestive list of micro-projects is given here. This has to match the competency and the COs. Similar micro-projects could be added by the concerned course teacher:
Students have to perform the following steps for any chosen project.

- a) Choose any topic of your choice and enlist its requirements.
- b) Draw an ER Diagram for your chosen topic and prepare tables, establish relationships between them.
- c) Normalize the database.
- d) Determine the different scenarios and how data will be fetched with queries.
- e) Write the relational algebra queries for queries mentioned.

Sample Project Definitions: Hotel management, Event Management, Hospital Management, Health Monitoring Management System, Airline Management, Bank Management system, Transportation Management System, Library Management System.

14. SUGGESTED LEARNING RESOURCES

| Sr. No | Title of Book | Author | Publication with place, year and ISBN |
|--------|-------------------------------------|--------------|---|
| 1 | Database System Concepts | Henry Korth | Tata McGraw Hill, New Delhi, 978-9390727506 |
| 2 | SQL/ PI-SQL | Ivan Bayross | BPB Publications, Delhi 978-8176569644 |
| 3 | An Introduction to Database Systems | C. J. Date | Pearson Education India 978-0321197849 |

15. SOFTWARE/LEARNING WEBSITES

- a) Latest database trends: <https://cloud.google.com/blog/products/databases>
- b) SQL Basic Concepts: <http://www.w3schools.com/sql/>
- c) SQL Tutorial : <http://beginner-sql-tutorial.com/sql.htm>
- d) DBMS Course: <https://nptel.ac.in/courses/106105175>

16. PO-COMPETENCY-CO MAPPING

| Semester III | Database Management (4331604) | | | | | | |
|--|--|--------------------------|---|--|---|----------------------------|----------------------------|
| | POs and PSOs | | | | | | |
| Competency & Course Outcomes | PO 1 Basic & Discipline specific knowledge | PO 2 Problem Analysis | PO 3 Design/development of solutions | PO 4 Engineering Tools, Experimentation & Testing | PO 5 Engineering practices for society, sustainability & environment | PO 6 Project Management | PO 7 Life-long learning |
| Competency | <i>Design database for a given application & perform various database operations.</i> | | | | | | |
| Course Outcomes | | | | | | | |
| CO a) Explain the basic concepts of databases. | 3 | - | - | - | - | - | - |
| CO b) Design database using Entity relationship approach. | 2 | 3 | 2 | 1 | - | 1 | 2 |
| CO c) Implement Relational algebra in a database. | 2 | 1 | - | - | - | - | - |
| CO d) Apply SQL Commands for creating, manipulating and controlling databases. | 3 | 2 | 3 | 3 | - | 1 | 1 |
| CO e) Apply concepts of normalization to design an optimal database. | 2 | 3 | 2 | - | - | 2 | 2 |
| CO f) Explain transaction management concepts for concurrent use of database. | 2 | 2 | - | - | - | - | - |

Legend: '3' for high, '2' for medium, '1' for low or '-' for the relevant correlation of each competency, CO, with PO/ PSO

17. COURSE CURRICULUM DEVELOPMENT COMMITTEE**GTU Resource Persons**

| Sr. No. | Name and Designation | Institute | Email |
|---------|------------------------|----------------|---------------------------|
| 1 | Miss Priti N. Parikh | GGP, Ahmedabad | priti@priti@gmail.com |
| 2 | Mrs. Chandni H. Shah | GGP, Ahmedabad | chandni.it.engg@gmail.com |
| 3 | Mrs. Nimisha K. Rathod | GGP, Ahmedabad | nimisha.it.dte@gmail.com |