

Course Title	Database Management Systems	
Course Code	CE513	
Course Credit	Lecture	: 03
	Practical	: 01
	Tutorial	: 00
	Credits	: 04
Course Learning Outcomes		
<p>At the end of this course, students will be able to:</p> <ul style="list-style-type: none"> • Understand the basic concepts of DBMS • Distinguish file system, DBMS & RDBMS • Experiment various SQL commands to perform data manipulation • Create stored procedure & triggers to improve the performance at database level • Design ER-model for given traditional system • Evaluate the cost of simple and complex query • Understand the underlying concurrency concepts used in database applications i.e. banking, insurance, education, etc. • Develop normalized database design • Apply database constraints to maintain data consistency and to avoid data redundancy. • List recent advancement in the field of DBMS • Understand the concepts of database recovery in case of any failure. 		
Detailed Syllabus		
Sr. No.	Name of chapter & Contents	Hours Allotted
Section – I		
1	An overview of Database Management Introduction of Database and DBMS, Advantages of Database, Data Independence, Database System architecture- levels, Mappings, Database users and DBA, SQL/PL-SQL: (Introduction to SQL, Components of SQL (DDL,DML,DCL,DQL), Data constraints, functions, group by and having clause, sub queries, joins, views, security management commands, PL/SQL block, PL/SQL transactions, database triggers, stored procedures and stored functions)	05

2	Database Design and Entity-Relationship model : Overview of the Design Process. Entity-Relational Model, Constraints, Entity relationship diagram, ER design issues, Extended E-R features: specialization, generalization, aggregation, reduction to relational schemas,	04
3	Relational Model : Structure of relational databases, Fundamental Relational - Algebra Operations, Additional Relational - Algebra operations ,Extended Relational - Algebra Operations, Null values, Modification of the database	08
4	Query Processing & Query Optimization : Overview, phases of query processing, measures of query cost, selection operation, and sorting and joins evaluation of expressions	07
Section – II		
5	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, Multivalued dependency, 4NF, Join dependency and 5NF	03
6	Transaction Management and Concurrency Control: Transaction concepts, transaction state, Concurrent Executions, Serializability, recoverability, testing of serializability , Lock-based protocols, two-phase locking protocol, graph based protocol, timestamp-based protocol, validation-based protocols, Deadlock handling, deadlock detection and recovery,	10
7	Recovery System: Failure classification, Storage Structure, Recovery and Atomicity, Log-based Recovery: Deferred Database, Immediate database, Checkpoints, Recovery with Concurrent Transactions, Buffer Management ,database Security	08
8	Database Technologies: Object based database, Internet database, Digital library, Multimedia database, Mobile Database, Spatial Database, Unstructured data, Overview of Big Data	03

Instructional Method and Pedagogy:

- Small group activities to be conducted for case studies
- Feedback by posing a question, quiz, multiple choice questions.
- Group work assigning real world database application.
- Power point presentations integrated with video lectures.

Note:

- SQL/PLSQL topics of UNIT-1 will be discussed in detail during laboratory hours.
- Few topics of Relational Database Design (UNIT-6) will be discussed in details during LAB hours with case studies.
- Every student has to maintain LAB record book to mention every activities performed during LAB.

Reference Books

- Silberschatz, Korth, Sudarshan ,Database System Concepts, McGraw Hill,Fifth Edition
- C J Date,An introduction to Database Systems, Addition-Wesley, Eighth Edition
- S.K. Singh ,Database Systems: Concepts, Design and Applications, Pearson,2009
- RamezElmasri& Shamkant B. Navathe,Fundamentals of Database Systems, Addison-Wesley, Sixth Edition
- Alexis Leon and Mathews Leon, Database Management Systems, Leon Press, 2003

Additional Resources

- NPTEL Video Lectures of Database Design course of Computer Science & Engineering by Dr. S. Srinath, IIT Bangalore and Dr. S. Janaki Ram, IIT Madras [<http://nptel.ac.in/video.php?subjectId=106106093>]
- Video lectures of Database Management Systems available at <http://www.learnerstv.com/Free-Computer-Science-Video-lectures-ltv048-Page1.htm>

List of Experiments

Tutorial -1

Create the following tables using CREATE (DDL) statement.

Note: understand all the tables' structure as it is going to be used in further experiments.

1. branch (branch_name varchar2(15),branch_cityvarchar2(10),assets number(10))
2. customer (customer_name varchar2(15),customer_streetvarchar2(20),customer_cityvarchar2(10))
3. account (account_number varchar2(10), branch_namevarchar2(15),balance number(10))
4. loan (loan_number varchar2(10),branch_namevarchar2(15),amount number(10))
5. depositor (account_number varchar2(10),customer_namevarchar2(15))
6. borrower(customer_name varchar2(15),loan_numbervarchar2(10))

Tutorial 2

Apply the INSERT, SELECT(DML) on the created in tutorial-1 statements to satisfy the following

1. Insert relevant records in each table.
2. Retrieve all the records of each tables.
3. Retrieve loan no and amount from the loan table.
4. Retrieve branch name and branch city from the branch table
5. Retrieve account no. and balance from account table.

Tutorial 3

Useaggregate function to retrieve summarized information from table. Also **Experiments** various clauses (WHERE, IN, NOT IN, BETWEEN ...) to filter the information.

1. Find all loans of over 1500

2. Find the loan number for each loan of an amount greater than 1200
3. Display all the customer's name living in city Rajkot.
4. Display all the account no. for the balance 700 to 900.
5. Retrieve all the records of customers in the ascending order of customer name.
6. Retrieve all the records of loans in the Descending order of loan amount.
7. Retrieve all customer name whose name starts from letter 'r'.
8. Retrieve all unique branch name from loan.
9. List all customers who are not staying in city 'amreli' or 'jamnagar' or 'valsad'.
10. List the acc_no of all customers whose branch name is in kalavad road or mavdi road.
11. Count total no of accounts in kalavad road branch.
12. Find the branch name which has the maximum assets.
13. Find the loan which has the minimum balance.
14. Find the total loan amount of padara road branch.
15. Find the average amount of the asset.

Tutorial 4

Analyze the tables and extract information from one table based on another table.

1. Retrieve the details of the customer who have an account.
2. Find the names of all customers who have a loan at the Padara road branch.
3. Find the names of all customers who have a loan, an account, or both, from the bank.
4. Find the Customers with an account but no loan.
5. Find all customers who have an account at all branches located in Rajkot city.
6. Find all customers who have an account from at least the "Mavdi road" and the "Kalavad Road" branches.

Tutorial 5

Experiment DDL, DML Statements on tables created in tutorial-1

1. Delete all the details of acc_no A-101 from table account.
2. Delete all customers whose name starts with letter 'R'.
3. Delete the account having the balance between 300 to 500.
4. Update the city of customer named 'Neha' to 'Rajkot'
5. Update assets of branch gotri road to 8,00,000.
6. Change the size of customer_street column in customer to 30.
7. Add a new column named telephone of data type number and size 10 to customer.
8. Destroy the table borrower along with its data.
9. Change the name of table 'account' to 'account master'.
10. Truncate the depositor table.

Tutorial 6

Identify the limitation of database design created in tutorial-1. Make sure such limitation should not be there in the new database of sales data. To do so **Use** database constraints on the following tables

Customer_Master

COLUMN NAME	DATATYPE(SIZE)	CONSTRAINT
CUST_NO	VARCHAR2(6)	PRIMARY KEY/First character must start with 'C'
PRODUCT_NO	VARCHAR2(6)	Foreign key references PRODUCT_NO of PRODUCT_MASTER Table
CUST_NM	VARCHAR2(10)	UNIQUE
CUST_CITY	VARCHAR2(20)	
CUST_STATE	VARCHAR2(20)	
BALANCE	NUMBER(10)	

CUST_NO	PRODUCT_NO	CUST_NM	CUST_CITY	CUST_STATE	BALANCE
C01	P01	Anil Bhatt	Rajkot	Gujarat	10,000
C02	P01	Dhaval Trivedi	Ajmer	Rajasthan	15000
C03	P02	Vimal Desai	Ahmedabad	Gujarat	50000
C04	P04	Mukesh Das	Junagadh	Gujarat	20000
C05	P05	Venkatesh Rao	Nagpur	Maharashtra	20000
C06	P05	Nirav Ahuja	Bhopal	Madhya Pradesh	15000

Product_Master

COLUMN NAME	DATATYPE(SIZE)	CONSTRAINT
PRODUCT_NO	VARCHAR2(6)	PRIMARY KEY/First character must start with 'P'
NAME	VARCHAR2(20)	NOT NULL
QTY_ON_HAND	NUMBER(8)	
REORDER_LEVEL	NUMBER(3)	NOT NULL
SELL_PRICE	NUMBER(8,2)	NOT NULL
COST_PRICE	NUMBER(8,2)	NOT NULL

PRODUCT_NO	NAME	QTY_ON_HAND	REORDER_LEVEL	SELL_PRICE	COST_PRICE
P01	Shirts	1850	500	1300	1200
P02	Trousers	1900	500	1900	1500
P03	Jeans	2500	200	1200	800
P04	Kurtis	850	220	550	400
P05	T-Shirts	3000	1355	250	150
P06	Chudidar	1000	200	700	500
P07					

Salesman_Master

COLUMN NAME	DATATYPE(SIZE)	CONSTRAINT
SALESMAN_NO	VARCHAR2(6)	PRIMARY KEY / First character must start with 'S'
SALESMAN_NAME	VARCHAR2(20)	NOT NULL
ADDRESS	VARCHAR2(15)	NOT NULL
CITY	VARCHAR2(15)	
STATE	VARCHAR2(15)	
PINCODE	NUMBER(8)	NOT NULL
SAL_AMOUNT	NUMBER(9,2)	NOT NULL

SALESMAN_NO	SALESMAN_NAME	ADDRESS	CITY	STATE	PINCODE	SAL_AMOUNT
S01	Jigar Patel	Kalawad Road	Rajkot	Gujarat	360002	20000
S02	Nikhil Mehta	150 ft Road	Rajkot	Gujarat	360002	50000
S03	Himat Yadav	Tagore Road	Rajkot	Gujarat	360001	25000
S04	Nikunj Bhatt	Sardarnagar	Bhavnagar	Gujarat	364003	45000
S05	Yashpal Singh	Waghavadi Road	Bhavnagar	Gujarat	364004	20000
S06						

Sales_Order

COLUMN NAME	DATATYPE(SIZE)	CONSTRAINT
ORDER_NO	VARCHAR2(6)	PRIMARY KEY/First character must starts with 'OR'
CUST_NO	VARCHAR2(6)	Foreign key references CUST_NO of CUST_MSTR table
ORDER_DATE	DATE	NOT NULL
SALESMAN_NO	VARCHAR2(10)	Foreign key references SALESMAN_NO of SALESMAN_MASTER
DELY_TYPE	CHAR(1)	P=Part,F=Full

BILL	CHAR(1)	Y=if bill issued, N=not issued
DELY_DATE	DATE	NOT NULL
ORDER_STATUS	VARCHAR2(20)	Values('In Progress', 'Fulfilled', 'Cancelled')

ORDER_NO	CUSTOMER_NO	ORDER_DATE	SALESMAN_NO	DELY_TYPE	BILL	DELY_DATE	ORDER_STATUS
OR01	C01	16-JUNE-14	S01	F	Y	18-JUNE-14	Fulfilled
OR02	C02	18-SEP-14	S02	P	N	21-SEP-14	Cancelled
OR03	C03	8-OCT-14	S03	F	Y	11-OCT-09	Fulfilled
OR04	C03	9-NOV-14	S03	P	N	13-NOV-09	In Progress

Sales_order_detail

COLUMN NAME	DATATYPE(SIZE)	CONSTRAINT
ORDER_NO	VARCHAR2(6)	Foreign key references ORDER_NO of SALES_ORDER table
PRODUCT_NO	VARCHAR2(6)	Foreign key references PRODUCT_NO of PRODUCT_MASTER table
QTY_ORDERED	NUMBER(8)	NOT NULL
QTY_DISP	NUMBER(8)	NOT NULL

ORDER_NO	PRODUCT_NO	QTY_ORDERED	QTY_DISP
OR01	P01	50	45
OR02	P02	80	75
OR03	P03	30	29
OR04	P03	24	12
OR05			

Tutorial-7

Analyze the tables and extract information from one table based on another table for tutorial-6

1. Find the product number and description of products not being sold by any salesperson.
2. Display the details of the customers to whom, if the product 'Shirts' has been ordered.
3. Print the description and total quantity sold for each product.
4. Calculating the average quantity sold for each client that has a maximum order value of Rs. 15000.
5. Display the details of the customer whose order has been cancelled.
6. Display the details of the sales person who has sold to more than one customer

Tutorial – 8

Apply various database JOINS to discover information from multiple tables for the followings.

1. Find out the products which have been sold to 'Anil'.
2. Find out the products and their quantities that will have to be delivered in the current month.
3. List the products and orders from customers who have ordered less than 25 units of 'Jeans'.
4. Find the products and their quantities for the orders placed by 'Anil' and 'Mukesh'
5. Find the products and their quantities for the orders placed by customer no 'C01' and 'C02'.

Tutorial-9

Identify critical / secret information in tables. **Create** different views to provide access to other users.

1. Create view on order_no, order_date, order_status of the sales_order table and product_no and quantity ordered of sales_order_details.
2. Create a view from any table and manipulate it. (Update the View Data).
3. Create a view vw_prod (Product_no, Description, Profit_percent, Unit_measure, Sell_price) on the product_master table. Perform the operations such as Insert, Update and Delete on that view.

4. Create a view (Salesman_no, Salesman_name, City, Sal_amt) on the Salesman_master table for the Sales Department.

Tutorial – 10

In order to **improve** performance at database level, **analyze** the tables, **identify** columns and **create** various indexes.

1. Create a simple index on cust_no column of the customer_master table.
2. Create a composite index on the sales_order_details table on columns order_no and product_no.
3. Create a unique index named idx_cust_no on cust_no column of the customer_master table.
4. Remove index idx_cust_no created for the table customer_master.
5. Retrieve first seven rows by using ROWNUM. Sort the data in the ascending order of name.

Tutorial – 11

Apply the concepts of automatic generation of ID on tables using SEQUENCE.

1. Create a sequence by the name order_seq, which will generate numbers from 1 up to 9999 in ascending order with an interval of 1. The sequence must restart from the number 1 after generating number 9999.
2. Insert values for order_no, order_date, client_no in the sales_order table. The order_seq sequence must be used to generate order_no and order_date must be set to system date.
3. Alter a sequence order_seq created on sales order, change the Cache value of the sequence order_seq to 30 and interval between two numbers as 2.

Tutorial – 12 (Normalization - Case Study)

Understand the following data and **apply** normalization concepts to create database design such that it does not create data inconsistency and also prevent redundancy.

Following table contains data about Suppliers, companies' division & parts. Normalize the below mention table:

Company	Company founder	Company logo	Division	Part type	Supplier	Supplier country	Supplier continent
Allied clock and watch	Horace Washington	Sundial	Clocks	Spring, pendulum, Spring, toothed wheel	Tensile globodynamics, Tensile globodynamics, Pieza de Acero, Pieza de Acero	USA, USA, Maxico, Maxico	N.Amer., N.Amer., N.Amer., N.Amer.
Allied clock and watch	Horace Washington	Sundial	Watches	Quartz crystal, tuning fork, battery	Microflux, Microflux, Dakota Electronics	Belgium, Belgium, USA	Europe, Europe, N.Amer.
Global robot	Nils Neumann	gearbox	Industrial robots	Flywheel Axle, Axle, Mechanical, Arm	Wheel 4 less, Wheel 4 less, TransEuropa, TransEuropa	USA, USA, Italy, Italy	N.Amer., N.Amer., Europe, Europe
Global robot	Nils Neumann	gearbox	Domestic robots	Artificial Brain, Artificial Brain, Mental housing, Backplate	Prometheus Labs, Frankenstein labs, Pieza de Acero, Pieza de Acero	Luxembourg, Germany, Maxico, Maxico	Europe, Europe, N.Amer., N.Amer.

Note: In 1st LAB of this tutorial, every group of 2 students has to submit the title of real life system, to create normalized database design & there would be presentation / viva during 5th LAB.

Tutorial -13

Create PL/SQL block for the followings to **understand** its basic programming constructs:

1. Write a PL/SQL block to calculate the area of a circle for a value of radius varying from 3 to 7 store the radius and the corresponding values of calculated area in an empty table named areas consisting of two columns radius and area.
2. write a PL/SQL block of code for inverting the number 5639 to 9365
3. Write a PL/SQL block to find the sum of all odd numbers.
4. Write a PL/SQL block of code that 1 'st insert one new record in account table

Tutorial – 14

Create reusable modules (function and stored procedure) to **perform** specific task as mentioned below:

Functions

1. Write a function that finds whether the input parameters are a prime no.
2. Write a function that accepts the order no from the user and displays the product no and name of the product for that order no.

Stored procedures

1. Write a procedure which updates the field order_status from in progress to fulfilled in the sales_order table.(dely_date + 5 days)

Tutorial– 15

Experiment the followings to **understand** CURSOR in Oracle:

1. Declare cursor on salesman_master with SQL command used to define the cursor with many options like
 1. Declare cursor
 2. Open cursor
 3. Fetch cursor
 4. Close cursor
 5. Delete cursor

Tutorial - 16

Apply mechanism to maintain data consistency using trigger. **Create** trigger that allow to perform the following operations on specified time.

1. Write a database trigger before insert for each row on table sales_order not allowing transactions for the order_date as Sundays.

Tutorial - 17

In client-server architecture, we need to make sure that every user has proper rights to access database tables. **Use** GRANT & REVOKE commands to set proper rights as mention below:

1. Give the user Rahul permission only to view records in the table sales_order and sales_order_details along with an option to further grant permission on these tables to other users.
2. Use the Grant and Revoke Command by giving authority to Other User to update, delete or insert into Table customer_master.
3. Take back all privileges given to the user Rahul on the table customer_master.

Tutorial - 18 (Database Connectivity)

1. **Write** JAVA program to establish database connectivity and **perform** select, insert, delete and update SQL on specific table.
2. **Write** JAVA program to establish database connectivity and also **perform** any stored procedure created earlier.