UNIVERSITY OF MUMBAI
No.UG/191 of 2009

CIRCULAR :-

A reference is invited to the Ordinances, Regulations and syllabi relating to the B.Com. degree courses **vide** pamphlet No.152 and to this office Circular *No.UG/200 of 1999, dated 3rd June, 1999, Circular No.UG/499 of 2006, dated* 8th December, 2006 and No.UG/63 of 2007, dated 28th February, 2007 and the Principals of the affiliated Colleges in Commerce and the Professor-cum-Director, Institute of Distance Education are hereby informed that the recommendation made by the joint Board of Studies in Mathematics and Statistics at its meeting held on 15th April, 2009 has been accepted by the Academic Council at its meeting held on 21st April, 2009 ***vide*** *item No.4.41 and that, in accordance therewith, the revised* syllabus in the subject of Computer System and Applications (Applied Component Paper) at the T.Y.B.Com. examination is as per Appendix and that the same has been brought into force with effect from the academic year 2009-2010.

Further that the practical examination of 20 marks is conducted by the college, on behalf of the University and Theory examination of 80 marks is *conducted by the University.*

MUMBAI-400 032
4th *June, 2009*

PRIN. K. VENKATARAMANI
REGISTRAR

****

To,

The Principals of the affiliated colleges in Commerce, and the Professor-cum-Director, Institute of Distance Education.

A.C/4.41/21.04.2009

****

No.UG/191-A *of 2009,* *MUMBAI-400 032* 4th *June, 2009.*

Copy forwarded with compliments for information to :-

1) The Dean, Faculty of Commerce,
2) The Chairmen, Board of Studies in Mathematics and Statistics,
3) The Controller of Examinations,
4) The Co-ordinator, University Computerization Center.

(D. H. KATE)
DEPUTY REGISTRAR
(U.G./P.G.Section)

Copy to :

The Director, Board of College and University Development, the Deputy Registrar (Eligibility and Migration Section), the Director of Students Welfare, the Personal Assistants to the Vice-Chancellor, the Pro-Vice-Chancellor, the Registrar and the Assistant Registrar, Administrative, Ratnagiri for information.

The controller of examinations (10 copies), the Finance and Accounts officer (2 copies), Record Section (5 copies), Publications Section (5 copies), the Deputy Registrar, Enrollment, Eligibility and Migration Section (3 copies), the Deputy Registrar, Statistical, Affiliation Section (2 copies), the Director, Institute of Distance Education, (10 copies) the Director (2 copies) the Deputy Registrar (Special Cell), the Deputy Registrar,

Enclosure to Item No. 4.41
21.04.2009

UNIVERSITY OF MUMBAI

REVISED SYLLABUS

AT THE

T.Y.B.COM COURSE

IN THE SUBJECT OF

COMPUTER SYSTEMS AND APPLICATION

(Applied Component)

(with effect from the academic year 2010-2011)

T.Y.B.Com Syllabus for Computer Systems and Applications (Applied Component) to be implemented from 2009-2010

Prerequisites

Resume of processors, memory, ports, secondary storage devices, operating systems (no questions to be asked in the examination)

Unit No.	Topic	No. of Lectures	Practicals
Unit I	Data Communication & Networking	18	
Unit II	E-commerce	18	
Unit III	Database and SQL	18	6
Unit IV	Introduction to C Programming Or Visual Basic	18	6
Unit V	Spreadsheets	18	6
	Word Processing		1
	Presentation skills		1

Unit I Data Communication, Networking and Internet (18 L)

(a) Data Communications Components, Data representation, Distributed processing. (Concepts only)

(b) Network Basics and Infrastructure

i. Definition, Types (LAN,MAN,WAN) Advantages.
ii. Network Structures – Server Based, Client server, Peer to Peer.
iii. Topologies – Star, Bus, Ring.
iv. Network Media, Wired and Wireless
v. Network Hardware : Hubs, Bridges, Switches, Routers
vi. Network Protocols – TCP/IP,OSI Model

(c) Internet

I. Definition, Types of connections, Sharing internet connection,
II. Services on net- WWW, Email- Blogs.
III. IP addresses, Domain names, URLs.
IV. Searching Directories, Search engines, Boolean search (AND , OR, NOT), Advanced search, Meta Search engines.
V. Email –POP/SMTP accounts in Email, Different parts of an Email address. Receiving and sending emails with attachments by scanning attachments for viruses.

Unit II E-Commerce (18 Lectures)

a) Definition of E-commerce
b) Features of E-commerce
c) Types of E-commerce (B2C,B2B,C2C,P2P)
d) Business Models in E-Commerce (Advertising, Subscription, Transaction Fee, Sales Revenue, Affiliate Revenue)

e) Major B2C models (Portal, Etailer, Content Provider, Transaction Broker, Market Creator, Service provider, Community provider.
f) E-Commerce Security : Integrity, Non repudiation, Authenticity, Confidentiality, Privacy Availability.
g) Encryption ; Definition, Digital signatures, SSL.
h) Payment Systems : Digital Cash, Online stored value, digital accumulating balance payment, Digital credit accounts, digital checking.
i) How an Online credit card transaction works. SET protocol.
j) Limitations of E-commerce.
k) M-commerce (Definition and Features)

Unit III Database and SQL (18 Lectures)

1)**Introduction** to Databases, Relational and Non relational database systems SQL as a Non-procedural Language. Views of data.

2)**SQL Basics**: statements (Schema Statements, Data statements, Transaction statements , names (table & column names), data types (Char, Varchar, Text, Mediumtext, Longtext, Smallint, Bigint, Boolean, Decimal, Float, Double, Date, DateTime, TimeStamp, Year, Time, Creating Database, inserting data, Updating data, Deleting data, expressions , built-in functions, missing data (NULL and NOT NULL , DEFAULT values), CREATE, USE, ALTER (Add, Remove, Change columns) , RENAME, SHOW, DESCRIBE (CREATE TABLE, COLUMNS , STATUS and DATABASES only) and DROP (TABLE, COLUMN, DATABASES statements) ,. PRIMARY KEY, FOREIGN KEY (One and more columns) . Simple Validity checking using CONSTRAINTS

3) **Simple queries:** The SELECT statement (From, Where, Group By, Having, Order B, Distinct , Filtering Data by using conditions. Simple and complex conditions using logical , arithmetic and relational operators (=, !=, <,>, <>,AND, OR, NOT, LIKE, BETWEEN

4) **Multi-table queries** : Simple joins (INNER JOIN), SQL considerations for multi table queries (table aliases, qualified column names, all column selections , self joins).

5) **Nested Queries (Only upto two levels) :** using sub queries, sub query search conditions, sub queries & joins, nested sub queries , correlated sub queries, sub queries in the HAVING clause.

Simple Transaction illustrating START, COMMIT, and ROLLBACK.

UNIT IV Introduction to C Programming (18 lectures)

1) **Introduction :** c- character set, reserved words, identifiers, c-data types, c-type modifiers, constants, variables, expressions, operators(unary, binary, relational, logical), c-assignment statements, operator precedence, concept of header files (stdio.h, math.h, conio.h only).

2) **Simple programming concepts:**
Form of a c program, printf() (%d, %f, %c, %s, %lf, %ld), scanf(),gets(), getchar(),puts(), putchar),getch(),getche(),putch(),pow(),sqr(),clrscr(); storage class specifiers,
Preprocessor directive #include<>

3) **Control statements**
If else, for, while, do while, switch, break, continue.

4) **Arrays:** Concept of one dimensional array (character and integer), sorting array of numbers, printing array of numbers.

OR

Unit IV Visual Basic (18 Lectures)

a) Introduction to Visual Basic, Introduction Graphical User Interface (GUI), Programming Language (Procedural, Object Oriented, Event Driven), Writing VB projects, The Visual Basic Environment, Finding and fixing errors [Chapter 1 of 19]

b) Introduction to VB Controls : Textboxes, Frames, Check Boxes, Option Buttons, Images, Setting a Border & Styles, The Shape Control, The line Control, Working with multiple controls and their properties, Designing the User Interface, Keyboard access keys, Default & Cancel property, tab order, Coding for controls using Text, Caption, Value property and SetFocus method [Chapter 2 of 19]

c) Variables, Constants, and Calculations
Variable and Constant, Data Types (Boolean, Byte, String, Integer, Currency, Single, Double, Date), Naming rules/conventions, Constants (Named & Intrinsic), Declaring variables, Scope of variables, Val Function, Arithmetic Operations, Formatting Data. [Chapter 3 of 19]

d) Decision and Condition
Condition, Comparing numeric variables and constants, Comparing Strings, Comparing Text property of text box, Compound Conditions (And, Or, Not)
If Statement, If .then-else Statement, Nested If Statements, LCase and UCase function, Using If statements with Option Buttons & Check Boxes
MsgBox (Message Box) statement Input Validation: IsNumeric function.
Planning the project. [Chapter 4 of 19]

e) Menus, Sub-Procedures and Sub-functions, Defining / Creating and Modifying a Menu, Using common dialog box, Creating a new sub-procedure, Passing Variables to Procedures, Passing Argument ByVal or ByRef, Writing a Function Procedure [Chapter 5 of 19]

f) Lists, Loops and Printing
List Boxes and Combo Boxes
Do/ Loops, For/Next loops
Functions: MsgBox with multiple buttons, Strings
Sending information to the printer using Printer.Print, Printer.NewPage, Printer.EndDoc methods. [Chapter 7 of 19]

Unit V Spreadsheet (18 Lectures)

a) Creating and Navigating worksheets and adding information to worksheets

i. Types of data, entering different types of data such as texts, numbers, dates, functions.

ii. Quick ways to add data Auto complete, Autocorrect, Auto fill, Auto fit. Undo and Redo.
iii. Moving data, contiguous and non contiguous selections. Selecting with keyboard.
iv. Cut-Copy Paste. Adding and moving columns or rows. Inserting columns and rows. Find and replace values. Spell check.
v. Formatting Cells, Numbers, dates, times, Font, Colors, borders, fills.

b) Multiple spreadsheets

i. Creating and Using templates
ii. Creating and Linking Multiple Spreadsheets.
iii. Adding, removing, hiding and renaming worksheets.
iv. Add headers /Footers to a Workbook. Page breaks, previews.
v. Creating formulas, inserting functions, cell references, Absolute , Relative (within a worksheet, other worksheets and other workbooks), logical operators.
vi. Creating and using named ranges

c) Functions

i. Financial functions : FV,PV,PMT,PPMT,IPMT,NPER,RATE,NPV,IRR
ii. Database Functions LOOKUP, VLOOKUP,HLOOKUP
iii. Conditional Logic functions IF,COUNTIF,SUMIF
iv. Mathematical and statistical functions. ROUND,ROUNDDOWN,ROUNDUP,CEILING,FLOOR,INT,MAX,MIN,MOD, SQRT,
v. ABS,AVERAGE
vi. String functions LEFT,RIGHT,MID,LEN,UPPER,LOWER,PROPER,TRIM,
vii. Date functions. TODAY,NOW,DATE,TIME,DAY,MONTH,YEAR,WEEKDAY,DAYS360

d) Data Analysis

i. Sorting , filter with customized condition, subtotal
ii. The Graphical representation of data
iii. Pivot Tables – Building Pivot Tables, Pivot Table regions, Rearranging Pivot Table.

References:

Unit I Data Communication, Networking and Internet

1. Data Communications and Networking by Behrouz A Forouzan
2. Introduction to Computers – Peter Norton , Tata McGraw Hill

Unit II E-Commerce

3. E-Commerce – Kenneth Laudon Carol Traver , Pearson Education
4. Frontiers of Electronic Commerce: Kalakota & Whinston
5. E Commerce: Rajaraman
6. E- commerce – Whitley
7. E- commerce concept and cases- Rao and Deshpande.

Unit III Database and SQL

8. Fundamentals of Database Systems , Elmasri, Navathe , Somayajulu,Gupta
9. Database Systems and Concepts, Henry F. Korth, Sliberschatz, Sudarshan, McGraw Hill
10. DBMS by Date

11. The Complete Reference My Sql Vikram Vaswani TMH 2004 Seventh Reprint
12. The Complete reference SQL by James R. Groff & Paul N. Weinberg TMG
13. SQL a complete reference by Alexis Leon & Mathews Leon TMG
14. Learning SQL Alan Beaulieu O REILLY.
15. Learning MySQL, Seyed. M.M. and Hugh Williams, OREILLY.

UNIT IV Introduction to C Programming

16. C programming – Schaum series [SS]
17. Let us C – Yashvant Kanitkar [YK]
18. C programming – Rajaraman [R]

Unit IV Visual Basic

19. Programming in VB 6.0 by Julia case Bradley, Anita C. Millspaugh, TMH
20. Visual Basic 6.0 Programming by Content Development Group, TMH
21. The Complete Reference Visual Basic 6 by Noel Jerke, TMH
22. Visual Basic 6 Programming Black Book, Steven Holzner, Dreamtech Press

Note:

a) Theory 3 lectures per week.
b) Practical batch size 20 -25, 1 practical =3 theory lectures
c) 20 practicals are to be completed within 1 year

Topic	Number of Practicals
Word processing	1
Presentation skills	1
Spread sheet	6
SQL	6
Introduction to C Programming/Visual Basic	6

Minimum 12 practicals are to be recorded in journal
[minimum 4 on SQL, 4 on C/VB, 4 on spread sheet]

d) **Scheme of Examination**

Type	Marks	Duration
Theory	80	2 hours 30 minutes
Practicals	20	1 hour per batch of 10

Theory examination Pattern
All questions compulsory

Question No.	Unit No.	Marks	Marks with internal Option
Q1.	I	16	24
Q2.	II	16	24
Q3.	III	16	24
Q4.	IV	16	24
Q5.	V	16	24

Practical examination Pattern

Any two questions out of questions 1, 2, 3 are to be attempted

Viva will be based on Journal only.

Q. No.	Question No.	Marks
1	Spread sheet	5
2	SQL	5
3	Introduction to C / VB Programming	5
4	Journal	5
5	Viva	5

e) Practical examination to be conducted from 2nd or 3rd week of February. Marks out of 20 to be submitted to the University before commencement of theory examination.

f) Software Requirement :
Suitable Spread sheet application, MySQL 5 or 5.1 version, C language and VB [visual studio]

g) Hardware
For a batch of 120 students minimum 10 computers with appropriate hardware and software installed on each computer. During practical hours maximum two student may share one computer.

h) For in house computing facility fee of rupees 1500/- be charged for each student in the existing fee structure against the head of computer fee / computer practical.

Suggested practicals

BASIC SQL STATEMENTS

1.	Show the structure of the DEPARTMENTS table. Select all data from the table.
2.	Show the structure of the EMPLOYEES table. Display the last name, job code, hire date, and employee number for each employee, with employee number appearing first. Provide an alias STARTDATE for the HIRE_DATE column.
3.	Create a query to display unique job codes from the EMPLOYEES table.
4.	Display the last name concatenated with the job ID, separated by a comma and space, and name the column Employee and Title.
5.	Create a query to display all the data from the EMPLOYEES table. Separate each column by a comma. Name the column THE_OUTPUT.

Restricting and Sorting Data

1.	Create a query to display the last name and salary of employees earning more than $12,000.
2.	Create a query to display the employee last name and department number for employee number 176.
3.	Display the last name and salary for all employees whose salary is not in the range of $5,000 and $12,000.
4.	Display the employee last name, job ID, and start date of employees hired between February 20, 1998, and May 1, 1998. Order the query in ascending order by start date.
5.	Display the last name and department number of all employees in departments 20 and 50 in alphabetical order by name.
6.	list the last name and salary of employees who earn between $5,000 and $12,000, and are in department 20 or 50. Label the columns Employee and Monthly Salary, respectively.
7.	Display the last name and hire date of every employee who was hired in 1994.
8.	Display the last name and job title of all employees who do not have a manager.
9.	Display the last name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.
10.	Display the last names of all employees where the third letter of the name is an *a*.
11.	Display the last name of all employees who have an *a* and an *e* in their last name.
12.	Display the last name, job, and salary for all employees whose job is sales representative or stock clerk and whose salary is not equal to $2,500, $3,500, or $7,000.
13.	Display the last name, salary, and commission for all employees whose commission amount is 20%.

Single Row Functions

1.	Write a query to display the current date. Label the column Date.
2.	For each employee, display the employee number, last_name, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary.
3.	For each employee, display the employee number, last_name, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary. Add a column that subtracts the old salary from the new salary. Label the column Increase.
4.	Write a query that displays the employee's last names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.
5.	For each employee, display the employee's last name, and calculate the number of months between today and the date the employee was hired. Label the column MONTHS_WORKED. Order your results by the number of months employed. Round the number of months up to the closest whole number.
6.	Write a query that produces the following for each employee: <employee last name> earns <salary> monthly but wants <3 times salary>. Label the column Dream Salaries.
7.	Create a query to display the last name and salary for all employees. Format the salary to be 15 characters long, left-padded with $. Label the column SALARY.
8.	Display each employee's last name, hire date, and salary review date, which is the first Monday after six months of service. Label the column REVIEW. Format the dates to appear in the format similar to "Monday, the Thirty-First of July, 2000."
9.	Display the last name, hire date, and day of the week on which the employee

	started. Label the column DAY. Order the results by the day of the week starting with Monday.
10.	Create a query that displays the employees' last names and commission amounts. If an employee does not earn commission, put "No Commission." Label the column COMM.
11.	Create a query that displays the employees' last names and indicates the amounts of their annual salaries with asterisks. Each asterisk signifies a thousand dollars. Sort the data in descending order of salary. Label the column EMPLOYEES_AND_THEIR_SALARIES

Displaying Data from Multiple tables

1.	Write a query to display the last name, department number, and department name for all employees.
2.	Create a unique listing of all jobs that are in department 80. Include the location of the department in the output.
3.	Write a query to display the employee last name, department name, location ID, and city of all employees who earn a commission.
4.	Display the employee last name and department name for all employees who have an *a* (lowercase) in their last names.
5.	Write a query to display the last name, job, department number, and department name for all employees who work in Toronto.
6.	Display the employee last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.
7.	Display all employees including King, who has no manager. Order the results by the employee number.
8.	Create a query that displays employee last names, department numbers, and all the employees who work in the same department as a given employee. Give each column an appropriate label.
9.	Show the structure of the JOB_GRADES table. Create a query that displays the name,job, department name, salary, and grade for all employees.
10.	Create a query to display the name and hire date of any employee hired after employee Davies.
11.	Display the names and hire dates for all employees who were hired before their managers, along with their manager's names and hire dates. Label the columns Employee,Emp Hired, Manager, and Mgr Hired, respectively.

Aggregating Data Using Group Functions

1.	Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.
2.	Display the minimum, maximum, sum, and average salary for each job type.
3.	Write a query to display the number of people with the same job.
4.	Determine the number of managers without listing them. Label the column Number of Managers.
5.	Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.
6.	Display the manager number and the salary of the lowest paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is $6,000 or less. Sort the output in descending order of salary.
7.	Write a query to display each department's name, location, number of employees, and

the average salary for all employees in that department. Label the columns Name, Location, Number of People, and Salary, respectively. Round the average salary to two decimal places.

8. Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.

TOTAL	1995	1996	1997	1998
20	1	2	2	3

9. Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.

Job	Dept 20	Dept 50	Dept 80	Dept 90	Total
AC_ACCOUNT					8300
AC_MGR					12000
AD_ASST					4400
AD_PRES				24000	24000
AD_VP				34000	34000
IT_PROG					19200
MK_MAN	13000				13000
MK_REP	6000				6000
SA_MAN			10500		10500
SA_REP			19600		26600
ST_CLERK		11700			11700
ST_MAN		5800			5800

12 rows selected.

Sub Queries

1. Write a query to display the last name and hire date of any employee in the same department as Zlotkey. Exclude Zlotkey.
2. Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary.
3. Write a query that displays the employee numbers and last names of all employees who work in a department with any employee whose last name contains a *u*.
4. Display the last name, department number, and job ID of all employees whose department location ID is 1700.
5. Display the last name and salary of every employee who reports to King.
6. Display the department number, last name, and job ID for every employee in the Executive department.
7. Display the employee numbers, last names, and salaries of all employees who earn more than the average salary and who work in a department with any employee with a *u* in their name.

1. Run the query:
```
CREATE TABLE MY_EMPLOYEE
(ID      NUMBER(4) CONSTRAINT MY_EMPLOYEE_ID_NN  NOT NULL,
LAST_NAME  VARCHAR2(25) ,  FIRST_NAME VARCHAR2(25),
USERID    VARCHAR2(8),  SALARY    NUMBER(9,2) );
```
2. Describe the structure of the MY_EMPLOYEE table to identify the column names.
3. Add the first row of data to the MY_EMPLOYEE table from the following sample data. Do not list the columns in the INSERT clause.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	Rpatel	895
2	Dancs	Betty	Bdancs	860
3	Biri	Ben	Bbiri	1100
4	Newman	Chad	Cnewman	750
5	Ropeburn	Audrey	Aropebur	1550

4. Populate the MY_EMPLOYEE table with the second row of sample data from the preceding list. This time, list the columns explicitly in the INSERT clause. Confirm your addition to the table.
5. Write an insert statement in a text file named loademp.sql to load rows into the MY_EMPLOYEE table. Concatenate the first letter of the first name and the first seven characters of the last name to produce the user ID. Populate the table with the next two rows of sample data by running the insert statement
6. Make the data additions permanent.
7. Change the last name of employee 3 to Drexler.
8. Change the salary to 1000 for all employees with a salary less than 900.
9. Delete Betty Dancs from the MY_EMPLOYEE table.
10. Commit all pending changes.
11. Populate the table with the last row of sample data by modifying the statements in the script that you created in 5. Confirm the addition to the table.
12. Mark an intermediate point in the processing of the transaction.
13. Empty the entire table. Confirm that the table is empty.
14. Discard the most recent DELETE operation without discarding the earlier INSERT operation. Confirm that the new row is still intact.
15. Make the data addition permanent.

Creating and Managing Tables

1. Create the DEPT table based on the following table instance chart. Confirm that the table is created.

Column Name	ID	NAME
Data Type	NUMBER	VARCHAR2
Length	7	25

2. Populate the DEPT table with data from the DEPARTMENTS table. Include only columns that you need.
3. Create the EMP table based on the following table instance chart. Confirm that the table is created.

Column Name	ID	LAST_NAME	FIRST_NAME	DEPT_ID
Data Type	NUMBER	VARCHAR2	VARCHAR2	NUMBER
Length	7	25	25	7

4. Modify the EMP table to allow for longer employee last names up to 50 characters. Confirm your modification.
5. Create the EMPLOYEES2 table based on the structure of the EMPLOYEES table.

	Include only the EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY, and DEPARTMENT_ID columns. Name the columns in your new table ID, FIRST_NAME, LAST_NAME, SALARY , and DEPT_ID, respectively.
6.	Drop the EMP table.
7.	Rename the EMPLOYEES2 table as EMP.
8.	Drop the FIRST_NAME column from the EMP table. Confirm your modification by checking the description of the table.
9.	Add a table-level PRIMARY KEY constraint to the EMP table on the ID column. The constraint should be named at creation. Name the constraint my_emp_id_pk.
10	Create a PRIMARY KEY constraint to the DEPT table using the ID column. The constraint should be named at creation. Name the constraint my_dept_id_pk.
11	Add a column DEPT_ID to the EMP table. Add a foreign key reference on the EMP table that ensures that the employee is not assigned to a nonexistent department. Name the constraint my_emp_dept_id_fk.
12	Modify the EMP table. Add a COMMISSION column of NUMBER data type, precision 2, scale 2. Add a constraint to the commission column that ensures that a commission value is greater than zero.

VIEWS

1.	Create a view called EMPLOYEES_VU based on the employee numbers, employee names , and department numbers from the EMPLOYEES table. Change the heading for the employee name to EMPLOYEE.
2.	Display the contents of the EMPLOYEES_VU view.
3.	Using your EMPLOYEES_VU view, enter a query to display all employee names and department numbers.
4.	Create a view named DEPT50 that contains the employee numbers, employee last names, and department numbers for all employees in department 50. Label the view columns EMPNO, EMPLOYEE, and DEPTNO. Do not allow an employee to be reassigned to another department through the view.
5.	Display the structure and contents of the DEPT50 view.
6.	Attempt to reassign Matos to department 80.
7.	Create a view called SALARY_VU based on the employee last names, department names, salaries, and salary grades for all employees. Use the EMPLOYEES, DEPARTMENTS, and JOB_GRADES tables. Label the columns Employee, Department, Salary, and Grade, respectively.

SET Operators

1.	List the department IDs for departments that do not contain the job ID ST_CLERK, using SET operators.
2.	Display the country ID and the name of the countries that have no departments located in them, using SET operators.
3.	Produce a list of jobs for departments 10, 50, and 20, in that order. Display job ID and department ID, using SET operators.
4.	List the employee IDs and job IDs of those employees who currently hold the job title that they held before beginning their tenure with the company.
5.	Write a compound query that lists the following: • Last names and department ID of all the employees from the EMPLOYEES table, regardless of whether or not they belong to any department or not • Department ID and department name of all the departments from the DEPARTMENTS table, regardless of whether or not they have employees working in them

Advanced Sub queries

1.	Write a query to display the last name, department number, and salary of any employee whose department number and salary both match the department number and salary of any employee who earns a commission.
2.	Display the last name, department name, and salary of any employee whose salary and commission match the salary and commission of any employee located in location ID 1700.
3.	Create a query to display the last name, hire date, and salary for all employees who have the same salary and commission as Kochhar.
4.	Create a query to display the employees who earn a salary that is higher than the salary of all of the sales managers (JOB_ID = 'SA_MAN'). Sort the results on salary from highest to lowest.
5.	Display the details of the employee ID, last name, and department ID of those employees who live in cities whose name begins with *T*.
6.	Write a query to find all employees who earn more than the average salary in their departments. Display last name, salary, department ID, and the average salary for the department. Sort by average salary.
7.	Write a query to display the last names of the employees who earn less than the average salary in their departments.
8.	Write a query to display the last names of the employees who have one or more coworkers in their departments with later hire dates but higher salaries.
9.	Write a query to display the employee ID, last names, and department names of all employees.

Spreadsheet Practicals

Q1.
Prepare a worksheet containing the Name and marks in 3 subjects in columns A, B, C and D respectively from columns 1 to column 31, the first row contains the headings and the remaining rows contain data. Use your own data.
Carry out the following operations on this worksheet.
1. Save the worksheet under your name followed by your roll no. , question no. e.g. MONA1231
2. Obtain the total marks of each student in column E.
3. Construct the 3D Pie diagram for the totals.
4. Construct a line chart for subject 1 and subject 3.
4. Display only those students details who have total marks exceeding 180.
5. Display only those students details whose name begins with 'R'.

Q 2.
Prepare a worksheet containing the Name and taxable income in columns A and B respectively from rows 1 to 26, the first row contains the headings and the remaining rows contain the data. Use your own data.
Carry out the following operations on this worksheet.
1. Save the worksheet under your name followed by your roll no. , question no. e.g. MONA1232
2. Obtain the income tax in column C using the following income tax slabs.

Taxable Income(Rs)	Rate
First 1,00,000	Nil
Next 60,000	10%
Next 70,000	20%
Excess	30%

3. Obtain the surcharge in column D, where surcharge is 3% of the income tax for those whose taxable income is above Rs 5,00,000
4. Obtain the total tax in column E, as the sum of income tax and surcharge.

Q 3.
Prepare a worksheet containing the Name, Join Date, Department and Salary in Columns A, B, C, and D respectively from rows 1 to 31 the first row contains the headings and the remaining rows contain data. Use your own data.
Carry out the following operations on this worksheet.
1. Save the worksheet under your name followed by your roll no. , question no. e.g. MONA1233
2. Prepare a Pivot table report containing Join Date in the row area and Sum of Salary and Minimum Salary in the data area.
3. Prepare another Pivot table on the same data taking Department in the row area and Average Salary and Maximum Salary in the data area.

Q 4.
Prepare a worksheet containing the Name, Age, Department and Bonus amount in Column A, B, C and D respectively from rows 1 to 26, the first row contains headings and the remaining rows contain data. Use your own data.
Carry out the following operations on this worksheet.
1. Save the worksheet under your name followed by your roll no. , question no. e.g. MONA1234
2. Sort the data in the ascending order of Department.
3. Obtain subtotals of Bonus for each department.
4. Explain the use of 1 and 2 row level symbol
5. Remove these subtotals and return the worksheet to its previous state.

Q 5.
Prepare a worksheet containing the Name of the salesman and sales for the months of January, February and March in columns A, B, C and D respectively from rows 1 to 26, the first row contains headings and the remaining rows contain data. Use your own data ensuring that the sales entered in a month are minimum 50000 and maximum 250000.
Carry out the following operations on this worksheet.
1. Enter the following Bonus Table in columns H and I.

SALES	BONUS
300000	2000
400000	3000
500000	4000
600000	5000
700000	6000
800000	7000

2. Save the worksheet under your name followed by your roll no. , question no. e.g. MONA1235
3. Obtain the total sales of each salesman in the first quarter in column E.
4. Using VLOOKUP function obtain the bonus amounts to be given to each salesman in column F.(Note if the sales are say 319000, which is above 300000 and below 350000 then the Bonus amount will be 2000)

C programming practicals

(14)

1> Exchange of variables using temporary variable and without using temporary variable
2> assignment and I/O statements
sales bill, electric bill, average marks etc..
3> if else statement
4> for , while, do while loop
5> simple interest and compound interest
6> depreciation calculation
7> sorting of array of integers

VB Programming practicals

0.1 To study VB controls and their properties
To find common properties,methods & events for the control

Present in the following format

Control Name	Property	Value

Property	Description

Method	Description

Events	Description

0.2 Exchange contents of two variables using third temporary variable

0.3 Use assignment statement in the calculation of
a) Gross salaryb) Electricity bill c) Average marks in 3 subjects
d) Sales Bill e) Simple interest
Use your own data

0.4 Use if statement in the calculation of
a) Income tax on annual salary. Use only two income slabs
b) Electricity bill where rate per unit depends on the units consumed
c) Process result of a student to declare pass or fail

0.5 Use of Loop statement
a) Input Principal, Rate of interest and period in years. Display amount accumulated at the end of each year
b) Display set of integers in the given range as an arithmetic progression

(15)

The following exercises are to be completed as a project in VB [19]

Visual Basic programming practicals are suggestive from [19]

1. 1.2 and 1.3
2. 2.4 and 2.5
3. 3.3 and 3.5
4. 4.3, 4.5 and 4.6
5. 5.1 and 5.3

1.2 Write a project that displays a different greeting, or make it display the name of your school or your company. Include at least three command buttons to display the greeting, print the form, and exit the project. Include a label that holds your name at the bottom of the form and change the Caption property of the form to something meaningful. Follow good naming conventions for object names; Include remarks at the top of every procedure and in the General declarations section of the module. Select a different font name and font for the greeting label. If you wish, you can also select a different color for the font. Select each font attribute from the Font dialog box from the Properties window.

1.3 Write a project that displays four sayings, such as "The early bird gets the worm" or "A penny saved is a penny earned" (You will want to keep the sayings short, as each must be entered on one code statement. However, when the saying displays on your form, long lines will wrap within the label if the label is large enough)

Make a command button for each saying with a descriptive Caption for each, as well as command buttons to print the form and to exit the Project.

Include a label that holds your name at the bottom of the form. Also, make sure to change the Caption property of the form to something meaningful.

You may change the Font properties of the large label to the font and size of your choice. Make sure the label is large enough to display your longest saying and that the command buttons are large enough to hold their entire Captions.
Follow good naming conventions for object names; Include remarks at the top of every procedure and in the General declarations section of the module.

2.4 Write a project that will input the user name and display a message of the day in a label along with the user's name. Include command buttons (with keyboard access keys) for Display, Clear, Print, and Exit. Make the Display button the default button and the Clear button the cancel button. Include ToolTips where appropriate.

Include a group of option buttons for users to select the font size of the message. Give them a choice of three different sizes. Hint: Display the Font dialog box from the Properties window to determine the available sizes. Make sure your message label is large enough to display the longest message in the largest font size. Make your form display a changeable Image. Use the happy face icon files or any other Images you have available.
You may choose to have only one message of the day, or you can have several that the user can select from option buttons. You might want to choose messages that go with the different face icons.

Optional extra: Include check boxes to change the font attributes of the message. Include Bold, Underline, Italic, and Strike Thru. Make sure to Include keyboard access keys.

2.5 Create a project that allows the user to Input information and then display the lines of output for a mailing label.
Remember that fields to be input by the user require text boxes, whereas information to be

displayed belongs in labels. Use text boxes for first name, last name, street address, city, state, and ZIP code; give meaningful names to the text boxes and set the initial Text properties to blank.
Add appropriate labels to each text box to tell the user which data will be entered into each box and also provide Tooltips.
Use command buttons for Display Label Info, Clear, Print Form, and Exit. Make the Display button the default button and the Clear button the cancel button .
Use three labels for displaying the Information for Line 1, Line 2, and Line 3.

A click event on the Display label Info command button will displaythe following

Line 1 -- The first name and last name concatenated together
Line 2 -- The street address.
Line 3 -- The city, state and zip code concatenated together (make sure to concatenate a comma and a space between the city and state using ",", and tow spaces between the state and zip code.

3.3 In retail sales, management needs to know the average inventory figure and the turnover of merchandise. Create a project that allows the user to enter the beginning inventory, the ending inventory, and the cost of goods sold.

Form: Include labeled text boxes for the beginning inventory, the ending inventory, and the cost of goods sold. After calculating the answers, display the average inventory and the turnover formatted in labels.

Include command buttons for Calculate, Clear, Print Form, and Exit.
The formulas for the calculations are

Average Inventory = (Beginning Inventory + Ending Inventory) / 2

Turnover = Cost of goods sold/Average inventory

Note: The average inventory is expressed in dollars; the turnover is the number of times the inventory turns over.
Code: Include procedures for the click event of each command button.
Display the results in labels. Format the average inventory as currency and the turnover as a number with one digit to the right of the decimal.

Test data

Beginning	Ending	Cost of goods sold
58500	47000	400000
75300	13600	515400
3000	19600	48000

Check Figures

Average Inventory	Turn Over
52,750.00	7.58
44,450.00	11.60
11,300.00	4.25

3.5 Create a project that determines the future value of an investment at a given interest rate for a given number of years.

The formula for the calculation is Future value = Investment amount * (1 + Interest rate) ^ Years

Form: Use labeled text boxes for the amount of investment, the interest rate (as a decimal fraction), and the number of years the investment will be held. Display the future value in a label formatted as currency.

Include command buttons for Calculate, Clear, Print Form, and Exit.
Format all dollar amounts.

Test data

Amount	Rate	Years
2000.00	.15	5
1234.56	.075	3

Check Figures

Future Value
4,022.71
1,533.69

4.3 Create a project to compute your checking account balance.
Form: Include option buttons to indicate the type of transaction-deposit, check, or service charge. A text box will allow the user to enter the amount of the transaction. Display the new balance in a label. Calculate the balance by adding deposits and subtracting service charges and checks.
Include command buttons for Calculate, Clear and Exit.

4.5 Modify project 4.3 by adding a *Summary* command button that will display the total number of deposits, the total dollar amount of de posits, the number of checks, and the dollar amount of the checks. Do not include checks that were returned for insufficient funds, but do include the service charges. Use a message box to display the Summary information.

4.6 Piecework workers are paid by the piece. Workers who produce a greater quantity of output are often paid at a higher rate.

Form: Use text boxes to obtain the person's name and the number of pieces completed. Include a Calculate command button to display the dollar amount earned. You will need a Summary button to display the total number of pieces, the total pay, and the average pay per person. A clear button should clear the name and the number of pieces for the current employee.

Include validation to check for missing data. If the user clicks on the Calculate button without first entering a name and number of pieces, display a message box. Also, you need to make sure to not display a summary before any data are entered; you cannot calculate an average when no items have been calculated. You can check the number of employees in the Summary event procedure or disable the Summary command button until the first order has been calculated.

Pieces Completed	Price paid per piece for all pieces
1-199	.50
200-399	.55
400-599	.60
600 or more	.65

5.1 Modify project 4.6 (Piecework Pay) to replace command buttons with menus and add a function procedure.
This project will input the number of pieces and calculate the pay for multiple employees. It also must display a summary of the total number of pieces, the total pay, and the average pay for all employees.

Menu: The menu- bar must have these commands:

File
- Calc Pay
- Summary
- Exit

Edit
- Clear
- ———
- Font
- Color

Help
- About

Piecework workers are paid by the piece. Workers who produce a greater quantity of output may be paid at a higher rate.
Use text boxes to obtain the name and the number of pieces completed.
The *Calc Pay* menu command calculates and displays the dollar amount earned. The *Summary* menu command displays the total number of pieces, the total pay, and the average pay per person in a message box.
The *Clear* menu choice clears the name and the number of pieces for the current employee and resets the focus.
The *Color* and *Font* commands should change the color and font of the information displayed in the amount earned label.
Use a message box to display your name for the *About* option on the *Help* menu.
Use a function procedure to find the pay rate and return a value to the proper event procedure.

Pieces Completed	Price paid per piece for all pieces
1-199	.50
200-399	.55
400-599	.60
600 or more	.65

5.3 A salesperson earns a weekly base salary plus a commission when sales are at or above quota. Create a project that allows the user to input the weekly sales and the salesperson's name, calculates the commission, and displays summary information.
Form: The form will have text boxes for the salesperson name and his or her weekly sales.

Menu: The menu- bar must have these commands:

File
- Pay
- Summary
- Exit

Edit
- Clear
- ———
- Font
- Color

Help
- About

Color (19)

Use constants to establish the base pay, the quota, and the commission rate.
The Pay menu command calculates and displays in labels the commission and the total pay for that person. However, if there is no commission, do not display the commission amount (do not display a zero-commission amount).
Use a function procedure to calculate the commission. The function must compare sales to the quota. When the sales are equal to or greater than the quota, calculate the commission by multiplying sales by the commission rate.
Each salesperson receives the base pay plus the commission (If one has been earned).
Format the amounts to two decimal places; do not display a currency sign.
The Summary menu command displays a message box containing total sales, total commissions, and total pay for all salespersons. Display the numbers with two decimal places and currency signs.
The clear menu command clears the name, sales, and pay for the cur- rent employee and then resets the focus.
The Color and Font menu commands should change the color and font of the information displayed in the amount earned label.
Use a message box to display your name as programmer for the About option on the Help menu

Test Data: Quota = 1000; commission rate = .15; and Base pay = 250.

Name	Sales
Ashok	1,000
Gita	999.99
Rajesh	2,000.00

Totals should be

Sales	Rs. 3,999.99
Commission	450.00
Pay	1,200.00

Additional Exercises

1. Enter the amount in dollar and convert it to rupees and display .
2. Enter the marks in two subjects, find the average and generate the message "Distinction" if the average is above 75, "First Class" if it is above 60 and below 75, and "Second Class", if the average is above 45 but below 60.
3. Enter the name and annual taxable income of a person and based on a slab rate calculate the income tax payable by the person.
4. Enter the name and electricity units consumed by a person and based on slab rates calculate the amount payable by the consumer.
5. Enter the name and basic pay of an employee and calculate the allowances and deductions based on a specified criteria and calculate the net pay.

Exercises requiring Loop/ Select Case

1. Enter a number N and find the sum of all numbers up to N.
2. Enter N and find N factorial.
3. Enter the month number and year and display the number of days in the month.
4. Generate a conversion table of temperature in degrees and convert it to Fahrenheit in steps of 5 starting from 0 to 50 degrees.

— End —