Asansol Engineering College

Sub. Name: Database Management System Lab *Sub. Code: PCC-CS691* *Dept- IT*

<table>
<tr><th>Sl. No</th><th>Name of the Experiment</th><th>List of Experiments</th><th>Page No</th></tr>
<tr><td>1</td><td>Database Management and Table Operations in SQL, including creating and deleting databases, creating tables with specific schemas, inserting data, displaying data, and create table from another tables.</td><td>1. Create a database named as aecitdb
2. Create another database named as tempdb
3. Delete the database tempdb
4. Enter into the database named as aecitdb
5. Create a table named as TempPerson with the following Schema
TempPerson (Pid,FirstName,LastName,City,DOB,Salary)
(Data Types of the attributes can be selected by your own choice)
6. Create a table named as Person with the following Schema
Person (Pid,FirstName,LastName,City,DOB,Salary)
(Data Types of the attributes should be as follows:
Pid -- > it should be Integer in nature
FirstName,LastName,City -- > It should be String in nature
DOB -- > It should be Date in nature
Salary -- > It should be Decimal in nature)
7. Insert the following data in the Person table.
101—Amal—Biswas—Asansol—12/08/1972—90000.89
102—Tamal—Choudhury—Kolkata—15/02/1992—40000.58
103—Rita—Sen—Durgapur—12/08/1986—60000.36
8. Show all the data present in the Table, Person.
9. Create a same table named as DummyPerson from the existing table named as Person.
10. Show all the data present in the recently created Table, DummyPerson.
11. Show all the tables present in the currently operating database.
12. Delete the table, DummyPerson.</td><td></td></tr>
<tr><td>2</td><td>Database Management with Table Creation, Modification, and Data Insertion in SQL, Constraints specifications</td><td>1. Create a table named as ITEM with the following specifications
<table><tr><th>Attribute Name</th><th>Data Types</th><th>Constraints</th></tr><tr><td>I_no</td><td>Integer in nature</td><td>Primary Key, values within 1 to 1000</td></tr><tr><td>I_name</td><td>String with maximum 20 characters</td><td>Cannot be null, will be in UPPERCASE</td></tr><tr><td>I_price</td><td>Decimal in nature</td><td>Should be within 1.00 to 25.95</td></tr><tr><td>I_qty</td><td>Integer in nature</td><td>Should not be less than 1</td></tr></table>2. Create a table named as DUMMY_ITEM_1 with the same specifications as of ITEM.
3. Create a table named as DUMMY_ITEM_2 with the same specifications as of ITEM but attributes names will be as follows:
I_no --- > Item_no , I_name ---> Item_name,
I_price --- > Item_price, I_qty ---> Item_quantity
4. Drop the Primary key from ITEM table.</td><td></td></tr>
</table>

5. Add the Primary Key in the ITEM table.
6. Create a table named as **CUSTOMER** with the following specifications

Attribute Name	Data Types	Constraints
Cust_no	Integer in nature	Empty cell value not Possible, unique value
Cust_name	String with maximum 20 characters	not null, uppercase
State	String with maximum character	values must be within 'WB', 'UP', 'AP', default value is 'WB'

7. Insert the following data into the **ITEM** table

I_no	I_name	I_price	I_qty
1	Sword	2.25	50
2	Nut	5.00	110
3	Bolt	3.99	75
4	Hammer	9.99	125
5	Washer	1.99	100
6	Nail	0.99	300
7	Axe	3.55	25
8	Scissor	2.19	15

8. Insert the following data into the **CUSTOMER** table

Cust_no	Cust_name	State
1001	Prakash	UP
1002	Mukesh	AP
1003	Murti	UP
1004	Rajan	WB

9. Using SELECT and INSERT together populate the **DUMMY_ITEM_2** table from **ITEM** table.

3 — **SQL Queries on tables ITEM and CUSTOMER, including displaying item and customer details, filtering based on conditions, searching for specific items, and sorting results.**

Using the following tables perform the following queries:

ITEM(I_no, I_name,I_price,I_qty)

CUSTOMER(Cust_no,Cust_name,State)

1. Display details of all items
2. Display details of all customers from 'UP'.
3. Display details of the customer 'Prakash'.
4. Display all customers who are either from 'AP' or from 'WB'.
5. Display all customers who are not from 'UP'.
6. Display the details of item 'Nail'.
7. Find all items whose price lies between 2 and 6.
8. Find all items whose price does not lie between 2 and 6.
9. Find all items whose price is greater than 1.00 but quantity is less than 200.
10. Find all items which have 'o' in their names.
11. Find all items which starts with 'A'
12. Find all items which ends with 'r'
13. Sort all customers in descending order of their states.

4 — **Database Modification**

Using the following tables perform the following queries:

	and Table Management in SQL, involving updating data, altering table structure, adding and removing columns, changing column names, displaying constraints, truncating data, and deleting tables.	**ITEM**(I_no, I_name,I_price,I_qty) **CUSTOMER**(Cust_no,Cust_name,State) 1. Change I_qty to 75 of I_no 1. 2. Change the I_price of 'Nut' to 6. 3. Increase the Item price by 10%. 4. Delete all items whose quantity is 500. 5. Delete all items those prices lie between 0.1 and 1.00. 6. Add a column "Phone_no number (10)" to **CUSTOMER** table. 7. Change the size of the newly added column to 15. 8. Change the name of **Phone_no** attribute to **Ph_number** 9. Remove the **Ph_number** attribute from the table 10. Change the name of **CUSTOMER** to **CUSTOMER_YourName**. 11. Change the name of ITEM to ITEM_YourName. 12. Display the all the constraints of ITEM_YourName table. 13. Truncate all data from **CUSTOMER_YourName** 14. Display all data from **CUSTOMER_YourName** 15. Remove the tables **CUSTOMER_YourName, ITEM_YourName**, **DUMMY_ITEM** from the database.	
5	**Database Management and SQL Queries on EMPLOYEE and DEPARTMENT tables, involving table creation, foreign key relationships, data population, and various SELECT queries to retrieve specific information from the tables with Join operations.**	1. Create a table named as **EMPLOYEE** with the following specifications	

Name	Type
EMPNO	NUMBER (4), Primary Key
ENAME	VARCHAR2(10)
JOB	VARCHAR2(9)
MGR	NUMBER(4),
HIREDATE	DATE
SAL	NUMBER(7,2)
COMM	NUMBER(7,2)
DEPTNO	NUMBER(2)

2. Make the **DEPTNO** of EMPLOYEE table as a foreign key of **DEPARTMENT** table.
3. Create a table named as **DEPARTMENT** with the following specifications

Name	Type
DEPTNO	NUMBER(2),
DNAME	VARCHAR2(14), NOT NULL
LOC	VARCHAR2(13), NOT NULL

4. Make the DEPTNO of **DEPARTMENT** table as a Primary key.
5. Populate the **EMPLOYEE** table with the following data

ENO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	Smith	Clerk	7902	17-Dec-88	1000		20
7499	Allen	Salesman	7698	20-Feb-89	1600	300	30
7521	Ward	Salesman	7698	22-Feb-89	1250	500	30
7566	Jones	Manager	7839	02-Apr-89	2975		20
7654	Marti	Salesman	7698	28-Sep-89	1250	1400	30
7698	Blake	Manager	7839	01-May-89	2850		30
7782	Clark	Manager	7839	09-Jun-89	2450		10
7788	Wong	Analyst	7566	19-Apr-87	3000		20
7839	King	President		17-Nov-89	5000		10

7844	Turn	Salesman	7698	08-Sep-89	1500	0	30
7876	Adam	Clerk	7788	23-May-87	1100		20
7900	James	Clerk	7698	03-Dec-89	950		30
7902	Ford	Analyst	7566	03-Dec-89	3000		20
7934	Mille	Clerk	7782	23-Jan-86	1300		10

6. Populate the **DEPARTMENT** table with the following data

DEPTNO	DNAME	LOC
10	HRD	Houston
20	RESEARCH	Dallas
30	SALES	Chicago
40	OPERATION	Boston

7. Display the name, deptno, dname for all employees.
8. Create a unique listing of all jobs that are in deptno 30. Include the location of dept 30 in the output.
9. Display the employee's name, dept name and the location of all employees who earn a commission.
10. Display the employee's name, dept name for all employees who have an 'A' in their name.

6

SQL Queries on DEPARTMENT and EMPLOYEE tables, including retrieving employee details based on location, displaying hierarchical information, modifying queries to include specific employees, comparing hire dates, finding salary-related information such as the second and fifth highest salaries.

Using the following tables perform the following queries:
DEPARTMENT (DEPTNO,DNAME,LOC)
EMPLOYEE (ENO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO)

1. Display the employee name, job, deptno, dept name for all employees who work in DALLAS.
2. Display the employee name, empno along with their manager's name and manager no.
3. Modify the previous query,2, to display all employees including king, who has no manager.
4. Display the employee's name of all the employees that work in the same DEPTNO as a given employee.
5. Display the employee's name, job, dept name, salary, and grade for all employees.
6. Display the employee's name, and hire date of any employee hired after BLAKE.
7. Display the employee's name, and hire date along with their manager's name and hire date of all employees hired before their manager.
8. Find the second highest salary from emp table.
9. Find the fifth highest salary from emp table.

7

SQL Queries on DEPARTMENT and EMPLOYEE tables, including filtering based on job titles, date of

Using the following tables perform the following queries:
DEPARTMENT (DEPTNO, DNAME, LOC)
EMPLOYEE (ENO, ENAME, JOB, MGR,HIREDATE,SAL,COMM,DEPTNO)

1. List the names of analysts and salesmen.
2. List details of employees who have joined before 30 Sep 81.
3. List names of employees who are not managers.

	joining, managerial status, specific employee numbers, department membership, name patterns, and calculating various statistics like the total number of employees, number of designations, total salaries, and specific salary details.	4. List the names of employees whose employee numbers are 7369,7521, 7839,7934, 7788. 5. List employees not belonging to department 30, 40, or 10. 6. List employee names for those who have joined between 30 June and 31 Dec. '81. 7. List the different designations in the company. 8. List the names of employees who are not eligible for commission. 9. List the name and designation of the employee who does not report to anybody. 10. List the employees not assigned to any department. 11. List the employees who are eligible for commission. 12. List employees whose names either start or end with "S". 13. List names of employees whose names have "i" as the second character. 14. List the number of employees working with the company. 15. List the number of designations available in the EMP table. 16. List the total salaries paid to the employees. 17. List the maximum, minimum and average salary in the company. 18. List the maximum salary paid to a salesman.	
8	**Various queries performed on employee and department tables including calculations of employee count, average salary, PF amount, tenure, department-wise statistics, sorting, and additional salary components**	Using the following tables perform the following queries: **DEPARTMENT** (DEPTNO, DNAME, LOC) **EMPLOYEE** (ENO, ENAME, JOB, MGR,HIREDATE,SAL,COMM,DEPTNO) 1. List the number of employees and average salary for employees in department 20. 2. List name, salary and PF amount of all employees. (PF is calculated as 10% of basic salary) 3. List names of employees who are more than 2 years old in the company. 4. List the employee details in the ascending order of their basic salary. 5. List the employee's name and hire date in the descending order of the hire date. 6. List employee name, salary, PF, HRA, DA and gross; order the results in the ascending order of gross. HRA is 50% of the salary and DA is 30% of the salary. 7. List the department numbers and number of employees in each department. 8. List the department number and total salary payable in each department. 9. List the jobs and number of employees in each job. The result should be in the descending order of the number of employees. 10. List the total salary, maximum and minimum salary, and average salary of the employees' job wise. 11. List the total salary, maximum and minimum salary, and average salary of the employees, for department 20. 12. List the total salary, maximum and minimum salary, and average salary of the employees' job wise, for department 20 and display only those rows having an average salary > 1000	
9	**Various queries performed on employee and department tables including filtering, formatting output, salary calculations, date display, aggregate**	Using the following tables perform the following queries: **DEPARTMENT** (DEPTNO, DNAME, LOC) **EMPLOYEE** (ENO, ENAME, JOB, MGR, HIREDATE,SAL,COMM,DEPTNO) 1. Display the details of the employees whose commission is NULL. 2. Display employee's name and sal+comm. And give the heading Total_Salary. 3. Display the name and increased salary (20% increases) of each manager. 4. Display the output in following format for each Salesman: Mr. <Employee's name>'s total earning is (sal+comm). 5. Display today's date in following format: Today's date is <date>. 24th June ,2021	

	functions, subqueries, and deletions with associated actions.	6. Display min and max salaries among all employees and rename the fields accordingly. 7. Display details of all employees along with their dept name in the ascending order of hiredate. 8. Display min, max, average, and total salaries of each dept in ascending order of deptname. 9. Display name of the department and number of employees in which at least 4 employees work on. 10. Display emp name whose salary is higher than the average salary of all the employees having 5 letters in their name. 11. Display details of all employees whose deptname is either SALES or HRD (deptname should be supplied by user using sub query). 12. Display name of the location where Wong is working. 13. While deleting RESEARCH department of DEPT, it automatically deletes all the corresponding employees of that dept. 14. Display the details of top 3 earners. 15. Display each employee's name and their corresponding manager's name. 16. Modify previous query,15, to include the record of King in the output.	
10		1. Create the **SALESMAN** table with the following data **salesman_id name city commission** 5001 James Hoog New York 0.15 5002 Nail Knite Paris 0.13 5005 Pit Alex London 0.11 5006 Mc Lyon Paris 0.14 5007 Paul Adam Rome 0.13 5003 Lauson Hen San Jose 0.12 2. Create the **ORDER** table with the following data **ord_no purch_amt ord_date customer_id salesman_id** 70001 150.5 2012-10-05 3005 5002 70009 270.65 2012-09-10 3001 5005 70002 65.26 2012-10-05 3002 5001 70004 110.5 2012-08-17 3009 5003 70007 948.5 2012-09-10 3005 5002 70005 2400.6 2012-07-27 3007 5001 70008 5760 2012-09-10 3002 5001 70010 1983.43 2012-10-10 3004 5006 70003 2480.4 2012-10-10 3009 5003 70012 250.45 2012-06-27 3008 5002 70011 75.29 2012-08-17 3003 5007 70013 3045.6 2012-04-25 3002 5001 3. Create the **CUSTOMER** table with the following data **customer_id cust_name city grade salesman_id** 3002 Nick Rimando New York 100 5001 3007 Brad Davis New York 200 5001 3005 Graham Zusi California 200 5002 3008 Julian Green London 300 5002 3004 Fabian Johnson Paris 300 5006 3009 Geoff Cameron Berlin 100 5003 3003 Jozy Altidor Moscow 200 5007 3001 Brad Guzan London 5005	

		4. Write a SQL statement to display specific columns like name and commission for all the salesmen. 5. Write a query to display the columns in a specific order like order date, salesman id, order number and purchase amount from for all the orders. 6. Write a query which will retrieve the value of salesman id of all salesmen, getting orders from the customers in orders table without any repeats. 7. Write a SQL statement to display names and city of salesman, who belongs to the city of Paris. 8. Write a SQL statement to display all the information for those customers with a grade of 200. 9. Write a SQL query to display the order number followed by order date and the purchase amount for each order which will be delivered by the salesman who is holding the ID 5001. 10. Write a query to display all customers with a grade above 100. 11. Write a query statement to display all customers in New York who have a grade value above 100. 12. Write a SQL statement to display all customers, who are either belongs to the city New York or had a grade above 100. 13. Write a SQL statement to display all the customers, who are either belongs to the city New York or not had a grade above 100. 14. Write a SQL statement to display all customers, who are either belongs to the city New York or had a grade above 100.	
11		Using the following tables perform the following queries: **SALESMAN** (salesman_id, name, city, commission) **ORDER** (ord_no, purch_amt, ord_date, customer_id,salesman_id) **CUSTOMER** (customer_id, cust_name, city, grade, salesman_id) 1. Write a SQL query to display those customers who are neither belongs to the city New York nor grade value is more than 100. 2. Write a SQL statement to display either those orders which are not issued on date 2012-09-10 and issued by the salesman whose ID is 5005 and below or those orders which purchase amount is 1000.00 and below. 3. Write a SQL statement to display salesman_id, name, city and commission who gets the commission within the range more than 0.10% and less than 0.12%. 4. Write a SQL query to display all orders where purchase amount less than 200 or exclude those orders which order date is on or greater than 10th Feb,2012 and customer id is below 3009. 5. Write a SQL statement to exclude the rows which satisfy 1) order dates are 2012-08-17 and purchase amount is below 1000 2) customer id is greater than 3005 and purchase amount is below 1000. 6. Find those salesmen with all information who come from the city either Paris or Rome. 7. Write a query to produce a list of salesman_id, name, city and commision of each salesman who live in cities other than Paris and Rome. 8. Write a query to sort out those customers with all information whose ID value is within any of 3007, 3008 and 3009 9. Write a SQL statement to find those salesmen with all information who gets the commission within a range of 0.12 and 0.14. 10. Write a query to filter all those orders with all information where purchase amount value is within the range 500 and 4000 except those orders of purchase amount value 948.50 and 1983.43 11. Write a SQL statement to find those salesmen with all other information and name started with any letter within 'A' and 'L.	

		12. Write a SQL statement to find that customer with all information whose name begins with the letter 'B'. 13. Write a SQL statement to find those salesmen with all information whose name containing the 1st character is 'N' and the 4th character is 'l' and rests may be any character. 14. Write a SQL statement to find that customer with all information who does not get any grade except NULL. 15. Write a SQL statement to find that customer with all information who gets a grade except NULL value. 16. Write a query in SQL to display all the data of employees whose last name begins with a 'D'.	
12		Write the **PL/SQL Block** for the followings: 1. To find the largest from the three numbers. 2. To find the factorial value of any number. 3. To print the Fibonacci Series of n numbers. 4. To compute the area of the circle with radius 2,4,6……40 and store the data into a table 'circle' containing attributes 'radius' and 'area'. 5. To accept the marks for three subjects from a student, calculate its average. If average <50 then print FAIL, average is between 50 to 59 then Second Division, average is between 60 to 75 then First Division, average is in between 76 and above then print Distinction.	
13		Write the **PL/SQL Block** for the followings: 1. To accept the empno from EMP table and calculate the tax on salary based on the following – **Basic Salary** — **Tax** Less than 1500 — 5% of Salary 1500-2500 — 7% of Salary 2501-3000 — 9% of Salary 3501 and above — 10%of Salary. 2. 3. To calculate Gross Salary on the basis of basic salary if DA is 40% of basic, HRA is 20% of basic and PF deduction is 12% of basic salary, update all the records in emp table. 4. Input 2 non-negative numbers through the key board & find the GCD & LCM. 5. To check whether a number is Armstrong or not. 6. To display the prime numbers within a range 10 to 100	