VARCHAR B. P. PODDAR INSTITUTE OF MANAGEMENT & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

LABORATORY NAME: DONALD E KNUTH LAB(C101)

ACADEMIC YEAR: 2018-2019 EVEN SEMESTER

LIST OF EXPERIMENTS

Database Management System Lab (CS691)

TOPIC	LIST OF EXPERIMENTS	CO	PO/ PSO

Using SQL Create table, Insert values and Use predicates with select and project	1 a) Create the following table : **STUDENT**	CO1 CO2	PO1 PO2 PO3 PO4 PO5 PO8 PO9 PO10 PSO1

Column Name	Data Type	Size	Constraints
RegNo	Varchar2	6	Not null
RollNo	Number	6	Not null
Name	Varchar2	10	Not null
Address	Varchar2	15	Not null
PhoneNo	Number	10	
YearOfAdm	Number	4	Not null
DeptCode	Varchar2	4	Not null
Year	Number	1	Not null
BirthDate	Date		Not null

b) Insert the following data in the student table.

Reg No	Roll No	Name	Address	Phone No	YearOf Adm	DeptCode	Year	Birth Date
012301	123001	Ashish	Jadavpur	24761892	2003	CSE	3	01-Jun-81
012315	123015	Kamal	Kasba	24424987	2003	CSE	3	19-Sep-81
012424	124024	Ipsita	Kaikhali	25739608	2004	CSE	2	15-Aug-82
012250	122050	Anita	Hooghly	36719695	2002	IT	4	22-Dec-80
012344	123044	Biplab	Howrah		2003	IT	3	03-Jan-82
012357	123057	Samik	Barasat	25426742	2003	IT	3	15-Jul-81
012419	124019	Srija	Garia	24755655	2004	EE	2	25-Oct-82
012427	124027	Saibal	Garia	24753306	2004	ECE	2	22-Mar-83
012236	122036	Santanu	DumDum		2002	ECE	4	11-Dec-80
012349	123049	Gita	Kasba	24428682	2003	MCA	3	14-Apr-

									81	

c) Display all records
d) Display name, address and year of admission of each student
e) List the name and year of students who are in Computer Science.
f) List the names and departments of students belonging to 3rd year.
g) Display names of students with 'a' as the second letter in their names.
h) Display names of students in alphabetical order.
i) Display names and addresses of students who took admission in the year 2004.
j) List the names of students who do not have a phone number.

Use of DML - select rows, delete rows and update table operations	Note : Tables created previously in lab exercises may be used if required 2. a) Delete the name of a student whose roll no, year and department code is given. b) Display the number of students in each department. c) Change the address of a student whose roll no and name is given. d) Add the college phone number (25739607) to each of these students. e) Change the size of column Name to 15 characters. f) Add a column MarksObtained (number) to the student table. g) Insert values against marks column. h) Drop column MarksObtained from table student. i) Add constraint primary key to the column RegNo of table student. j) Add check constraints to the column year of student table. (year should be entered within 1,2,3,4).	CO1 CO2	PO1 PO2 PO3 PO4 PO5 PO8 PO9 PO10 PSO1
Use of DDL - Alter Table Statement, Check Constraints, Foreign Key constraints in SQL	Note : Tables created previously in lab exercises may be used if required 3. a. Create table DEPARTMENT	CO1 CO2	PO1 PO2 PO3 PO4 PO5 PO8 PO9 PO10 PSO1 PSO2

Column Name	**Data Type**	**Size**	**Constraints**
DeptCode	Varchar2	4	Not null, Primary key
DeptName	Varchar2	15	Not null
HOD	Varchar2	4	Not null

FACULTY

Column Name	**Data Type**	**Size**	**Constraints**
FacultyCode	Varchar2	4	Not null, Primary key, Starts with 'F'
FacultyName	Varchar2	15	Not null
DateOfJoin	Date		Not null
DeptCode	Varchar2	4	Must be either CSE,IT, CA, CHEM, MTHS, PHYS, HUM, BBA

b. Insert appropriate values in the above table.
c. Add constraint : DeptCode of Faculty is foreign key and references DeptCode in Department

<table>
<tr>
<td></td>
<td>
d. Find the names of faculties of CSE Department.
e. Find the number of faculties in the Computer application department
f. Show the names of the heads of departments with department name.
g. Find the number of faculties who joined in August.
h. Add an extra attribute to the faculty table - Salary Number(8,2)
i. Insert values into the corresponding field Salary Number(8,2).
j. Find the name and salary of the faculty who earn more than 8000.
k. Find the name, department of the faculties who earn between 8000 and 12000.
</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Join Operations Cartesian Product, Natural Join, Outer Join</td>
<td>
Note : Tables created previously in lab exercises may be used if required
4.
a. Create table SUBJECT and insert appropriate values.
<table>
<tr><th>Column Name</th><th>Data Type</th><th>Size</th><th>Constraints</th></tr>
<tr><td>SubjectCode</td><td>Varchar2</td><td>4</td><td>Not null, Primary key</td></tr>
<tr><td>SubjectName</td><td>Varchar2</td><td>15</td><td>Not null</td></tr>
<tr><td>Faculty</td><td>Varchar2</td><td>4</td><td>Foreign key references FacultyCode of table FACULTY</td></tr>
</table>
b. Find the number of faculties in each department with their department name.
c. Increment the salary of each faculty by Rs 500.
d. Find the names of students and faculties whose name start with ‘S’.
e. Find the students who stay in Kaikhali
f. Find the names of faculties who take classes in the IT department.
g. Find the names of all faculties whose HOD is given.
</td>
<td>CO1
CO2</td>
<td>PO1
PO2
PO3
PO4
PO5
PO8
PO9
PO10
PSO1
PSO2</td>
</tr>
<tr>
<td>Queries using aggregate functions (count,sum, avg,max,min) and group by, having</td>
<td>
Note : Tables created previously in lab exercises may be used if required
5.
a. Add extra attribute to the Subject table - department varchar2 (4), year varchar2 (1)
b. Insert values into the fields - department, year.
c. Find the maximum salary among the faculties.
d. Find the names of faculties who earn more than the average of all faculties.
e. List the names of faculties of CSE department who earn more than the average salary of the department.
f. Find the maximum and minimum salaries among faculties.
g. Find the second maximum salary among all faculties.
h. Find the names of faculties who are not the HOD’s of any department.
i. Find the names of subjects for students of CSE 3rd year.
</td>
<td>CO1
CO2</td>
<td>PO1
PO2
PO3
PO4
PO5
PO8
PO9
PO10
PSO1
PSO2</td>
</tr>
<tr>
<td>Creation and</td>
<td>Note : Tables created previously in lab exercises may be used if required
6.</td>
<td>CO1</td>
<td>PO1</td>
</tr>
</table>

Dropping of Views	a. Name the departments having highest number of faculties and display the names of faculties b. Create a view on the STUDENT table named V_STD selecting all the columns. Run the following queries on the view. i. Display all data from the view. ii. Insert a new row into the view with the following data – 012363 123011 Bishakh Salt Lake 23371987 2005 IT iii. Display data from student table to verify that the row has been inserted into the Table. iv. Update the address of Bishakh to "SectorV" & verify the change in the table. c. Create a view on student table snamed V_STD_2 selecting the columns – RegNo, Name, Year, Deptcode. i. Display data from the view. ii. Try to insert data into table through view. iii. Update the Deptcode of 'Kamal' to 'IT' through view. iv. Delete records of students of 4th year through view. d. Create a view named V_FACULTY consisting of columns FacultyName, DeptCode from FACULTY table and HOD from Department table. i. Display data from V_FACULTY ii. Try to insert a new row into this view V_FACULTY. iii. Try to update the DeptCode of a CSE faculty to IT.	CO2	PO2 PO3 PO4 PO5 PO8 PO9 PO10 PSO1 PSO2
Nested Queries using any, all in, exist, not exists, unique, intersect constraints	Note : Tables created previously in lab exercises may be used if required 7. Considering - Branch Schema <branch-name, branch-city, assets> Customer Schema <customer-name, customer-street, customer-city> Loan Schema <loan-number, branch-name, amount> Borrower Schema <customer-name, loan-number> Account Scheme <account-number, branch-name, balance> Depositor Scheme <customer-name, account-number> BRANCH TABLE	CO3	PO1 PO2 PO3 PO4 PO5 PO8 PO9 PO10 PSO1 PSO2

Branch Name	Branch City	Assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	800000

CUSTOMER TABLE

Customer Name	Customer Street	Customer City
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

BORROWER TABLE

Customer Name	Loan Number
Adams	l-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

ACCOUNT TABLE

Account Number	Branch Name	Balance
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900

	A-215	Mianus	700		
	A-217	Brighton	750		
	A-222	Redwood	700		
	A-305	Round Hill	350		

a. To find all customers having a loan, an account or both at the bank, without duplicates.
b. To find all customers having a loan, an account or both at the bank, with duplicates.
c. To find all customers having both a loan and an account at the bank, without duplicates.
d. To find all customers having a loan, an account or both at the bank, with duplicates.
e. To find all customers who have an account but no loan at the bank, without duplicates.
f. To find all customers who have an account but no loan at the bank, with duplicates.
g. Find the number of depositors for each branch where average account balance is more than Rs 1200.
h. Find all customers who have both an account and a loan at the Perryridge branch.
i. Find the names of all branches that have assets greater than that of each branch located in Brooklyn.
j. Find all customers who have an account at all the branches located in Brooklyn.
k. Find all customers who have at most one account at the Perryridge branch.
l. Find all customers who have at least two accounts at the Perryridge branch.
m. Find the all customers who have an account but no loan at the bank.
n. Find the all customers who have either an account or a loan (but not both) at the bank.

DDL DCL TCL Commands	Note : Tables created previously in lab exercises may be used if required 8. Consider the following tables namely "DEPARTMENTS" & "EMPLOYEES" Their schemas are as follows - Departments (dept _no , dept_ name , dept_location); Employees (emp_id , emp_name , emp_salary); a. Develop a query to grant all privileges of employees table into departments table b. Develop a query to grant some privileges of employees table into departments table c. Develop a query to revoke all privileges of employees table from departments table d. Develop a query to revoke some privileges of employees table from departments table e. Write a query to implement the save point f. Write a query to implement the rollback g. Write a query to implement the commit	CO3	PO1 PO2 PO3 PO4 PO5 PO8 PO9 PO10 PSO1 PSO2

PL/Sql Basic	9. a. Write a PL/SQL code, EX_INVNO.SQL, block for inverting a number using all forms of loops. b. Write a PL/SQL code, EX_SUMNO.SQL that prints the sum of 'n' natural numbers. c. Write a PL/SQL program to print all the prime numbers between 100 and 400 d. Write a PL/SQL program to print n terms of fibonacci series. e. Write a PL/SQL program to calculate HCF of two numbers. f. Write a PL/SQL code, EX_AREA.SQL, of block to calculate the area of the circle for the values of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in the table AREA_VALUES.	CO4	PO1 PO2 PO3 PO4 PO5 PO8 PO9 PO10 PSO1 PSO2
Procedures and cursors using PL/SQL	10. a. Create a PL/SQL program using cursors, to retrieve first tuple from the department relation. b. (use table dept(dno, dname, loc)) c. Create a PL/SQL program using cursors, to retrieve each tuple from the department relation. d. (use table dept(dno, dname, loc)) e. Create a PL/SQL program using cursors, to display the number, name, salary of the three highest paid employees. f. (use table emp(empno, ename,sal)) g. Create a PL/SQL program using cursors, to delete the employees whose salary is more than 3000. h. Create a PL/SQL program using cursors, to update the salary of each employee by the avg salary if their salary is less than avg salary. i. Create a PL/SQL program using cursors, to insert into a table, NEWEMP, the record of ALL MANAGERS. Also DISPLAY on the screen the NO, NAME, JOIN_DATE. Handle any user defined exceptions. j. (use table emp(emp_no, emp_name, join_date, desig))	CO4 CO5	PO1 PO2 PO3 PO4 PO5 PO8 PO9 PO10 PSO1 PSO2
Additional Experiments			
Creation and usage of trigger	Note : Tables created previously in lab exercises may be used if required 11. Considering - Empa Schema<id number, name, dname, age, income, expence, savings> Emp Schema<institute name, employee id, salary> Sal <institute name, total employee, total salary> a. For every insert or delete or update in Empa table create trigger to display the message TABLE IS INSERTED or TABLE IS DELETED or TABLE IS UPDATED b. Define trigger to force all department names to uppercase. c. Create a Trigger to check the age valid or not using message after every insert or delete or update in Trig table	CO5	PO1 PO2 PO3 PO4 PO5 PO8 PO9 PO10 PSO1

	d. Create a Trigger to check the age valid and Raise appropriate error code and error message. e. A trigger restricting updates that allows changes to Empa records only on Mondays through Fridays, and only during the hours of 8:00am to 5:00pm. f. Create a Trigger for Emp table it will update another table Sal while inserting values.		PSO2

Prepared By:

Faculty Name: Suvadeep Bhattacharjee

Signature: