

**DEPARTMENT OF**

**COMPUTER SCIENCE AND ENGINEERING**

**Data Base Management Systems**

**(CS331)**

B. Tech Degree – CSE

**School of Engineering and Technology,**

**CHRIST (Deemed to be University),**

**Kumbalagodu, Bengaluru-560 074**

December 2022



***Certificate***

*This is to certify that ………………………………………………. has successfully completed the record work for Database Management Systems –CS331P in partial fulfillment for the award of Bachelor of Technology in during the year 2021-2022.*

**Dr. K. Balachandran**

**HEAD OF DEPARTMENT FACULTY- IN CHARGE**

**EXAMINER 1:**

**EXAMINER 2:**

Name :

Register No. :

Examination Center :

Date of Examination :

**INDEX**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Exp. No** | **Date** | **Experiment Name** | **Page No** | **Marks** | **Signature** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Ex.No : 1 Date: 21/07/2022**

**Experiment 1**

**PROBLEM GIVEN:**

**DDL Commands**

1. Create a table employee with the following fields: Empno number, Name varchar, Sal Numbers
2. Drop table Employee
3. Create a table student with the following fields: Rollno number, Name varchar
4. Drop table student

**DML Commands**

1. Insert Tuples into employee
2. Display all tuples
3. Display tuples with constraints on EmpNo
4. Delete all tuples
5. Insert Tuples into Student
6. Display tuples
7. Delete a single tuple based on rollno

**QUERY:**

**DDL 1: Create a table employee with the following fields: Empno number, Name varchar, Sal Numbers**

create Table Employee (Empno Number(8), Name varchar(10), Sal Number(10,2));

**DDL 2: Drop table Employee**

drop table employee;

**DDL 3: Create a table student with the following fields: TOllno number, Name varchar**

create table Student (Rollno Number(8), Name varchar2(20));

**DDL 4: Drop table student**

drop table student;

**DML 1: Insert Tuples into employee**

insert into Employee values(1001,'dddd',250);

insert into Employee values(1002,'aaaa',500);

**DML 2: Display all Tuples**

Select \* from Employee;

**DML 3: Display tuples with constrains on EmpNo**

Select \* from Employee where Empno = 1002;

**DML 4: Delete all Tuples**

delete Employee;

**DML 5: Insert Tuples into Student**

insert into Student values(1000,'aa');

insert into Student values(1001,'bb');

insert into Student values(1002,'cc');

insert into Student values(1003,'dd');

insert into Student values(1004,'ee');

**DML 6: Display tuples**

Select \* from Student;

**DML 7: Delete a single tuple based on rollno**

delete student where rollno=1000;

**Spool File Code:**

SQL> spool

currently spooling to C:\2162019\DBMS\Exp1.txt

SQL> create Table Employee (Empno Number(8), Name varchar(10), Sal Number(10,2));

Table created.

SQL> desc Employee

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NUMBER(8)

NAME VARCHAR2(10)

SAL NUMBER(10,2)

SQL> insert into Employee values(1001,'dddd',250);

1 row created.

SQL> desc employee

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NUMBER(8)

NAME VARCHAR2(10)

SAL NUMBER(10,2)

SQL> desc Employee

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NUMBER(8)

NAME VARCHAR2(10)

SAL NUMBER(10,2)

SQL> spool off;

SQL> spool "C:\2162019\DBMS\Exp1.txt" append

SQL> spool

currently spooling to C:\2162019\DBMS\Exp1.txt

SQL> desc Employee

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NUMBER(8)

NAME VARCHAR2(10)

SAL NUMBER(10,2)

SQL> insert into Employee values(1002,'aaaa',500);

1 row created.

SQL> desc Employee

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NUMBER(8)

NAME VARCHAR2(10)

SAL NUMBER(10,2)

SQL> show user

USER is "SYSTEM"

SQL>

SQL> Select \* from Employee

2

SQL> Select \* from Employee where Empno = 1002;

EMPNO NAME SAL

---------- ---------- ----------

1002 aaaa 500

SQL> Select \* from Employee;

EMPNO NAME SAL

---------- ---------- ----------

1001 dddd 250

1002 aaaa 500

SQL> desc Employee

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NUMBER(8)

NAME VARCHAR2(10)

SAL NUMBER(10,2)

SQL> desc Employee

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NUMBER(8)

NAME VARCHAR2(10)

SAL NUMBER(10,2)

SQL> drop table

2

SQL> desc employee

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NUMBER(8)

NAME VARCHAR2(10)

SAL NUMBER(10,2)

SQL> Select \* from Employee;

EMPNO NAME SAL

---------- ---------- ----------

1001 dddd 250

1002 aaaa 500

SQL> delete Employee;

2 rows deleted.

SQL> Select \* from Employee;

no rows selected

SQL> insert into Employee values(2002,'aaaa', 500);

1 row created.

SQL> Select \* from Employee;

EMPNO NAME SAL

---------- ---------- ----------

2002 aaaa 500

SQL> insert into Employee values(2003,'bbbb',500);

1 row created.

SQL> insert into Employee values(2004,'cccc',500);

1 row created.

SQL> Select \* from Employee;

EMPNO NAME SAL

---------- ---------- ----------

2002 aaaa 500

2003 bbbb 500

2004 cccc 500

SQL> delete Employee where empno=2003;

1 row deleted.

SQL> Select \* from Employee;

EMPNO NAME SAL

---------- ---------- ----------

2002 aaaa 500

2004 cccc 500

SQL> insert into Employee values(2003,'dddd',250);

1 row created.

SQL> Select \* from Employee;

EMPNO NAME SAL

---------- ---------- ----------

2002 aaaa 500

2004 cccc 500

2003 dddd 250

SQL> delete Employee;

3 rows deleted.

SQL> desc Employee

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NUMBER(8)

NAME VARCHAR2(10)

SAL NUMBER(10,2)

SQL> select \* from employee;

no rows selected

SQL> drop table employee;

Table dropped.

SQL> spool off;

SQL> create table Student (Rollno Number(8), Name varchar2(20));

Table created.

SQL> insert into Student values(1000,'aa');

1 row created.

SQL> insert into Student values(1001,'bb');

1 row created.

SQL> insert into Student values(1002,'cc');

1 row created.

SQL> insert into Student values(1003,'dd');

1 row created.

SQL> insert into Student values(1004,'ee');

1 row created.

SQL> Select \* from Student;

ROLLNO NAME

---------- --------------------

1000 aa

1001 bb

1002 cc

1003 dd

1004 ee

SQL> delete student where rollno=1000;

1 row deleted.

SQL> Select \* from student;

ROLLNO NAME

---------- --------------------

1001 bb

1002 cc

1003 dd

1004 ee

SQL> drop table student;

Table dropped.

SQL> spool off;

**Ex.No : 2 Date: 28/07/2022**

**Experiment 2**

**QUERY:**

**1. Display all values in Employees**

Select \* from Employees;

**2. Create sample table to use user input**

create table samp(regno number, name varchar(15));

**3. User input**

insert into samp values(&No, '&name');

**4. Display the values inputted into sample table**

select \* from samp;

**5. Display employees table with only attributes first\_name, employee\_id, salary**

select first\_name, employee\_id, salary from employees;

**6. Implement user input search**

select &c1, &c2m, &c3 from employees;

**7. Display employees table with same attributes as ‘5’ but only where salary is greater than 10,000**

select first\_name, employee\_id, salary from employees where salary>10000;

**8. Display the name, id, and salary for all employees whose first name is Michael**

select first\_name, employee\_id, salary from employees where first\_name='Michael';

**9. Display the first name of all the employees and the same in lowercase**

select first\_name, lower(first\_name) from employees;

**10. Display first name, lowercase, uppercase, and first initial capitalized**

select first\_name, lower(first\_name),upper(first\_name),initcap(first\_name) from employees;

**11. Change linesize to a higher value for better readability**

set linesize 1500;

**12. Search for all employees whose first name is Michael, but implement the search without case requirement using the lower function**

select employee\_id, first\_name from employees where lower(first\_name)='michael';

**13. Display the first name, and salary of employees, arranging them using order by only**

select first\_name, salary from employees order by salary;

**14. Display the first name, and salary of employees, arranging them in descending order**

select first\_name, salary from employees order by salary desc;

**15. Display the first name, and salary of employees, arranging them in alphabetical order**

select first\_name, salary from employees order by first\_name;

**16. Display all the employees whose first name starts with an ‘A’**

select first\_name from employees where first\_name like 'A%';

**17. Display all the employees whose first name starts and ends with an ‘A/a’**

select first\_name from employees where first\_name like 'A%a';

**18. Display all the employees whose first names are either starting or ending with ‘A/a’**

select first\_name from employees where first\_name like 'A%' or first\_name like '%a';

**19. Display all the employees whose second letter in their first name is ‘a’**

select first\_name from employees where first\_name like '\_a%';

**20. Display all the employees whose third letter in their first name is ‘a’**

select first\_name from employees where first\_name like '\_\_a%';

**21. Display all the employees whose third letter in their first name is ‘n’**

select first\_name from employees where first\_name like '\_\_n%';

**22. Display all the representatives in the company**

select first\_name from employees where job\_id like '%REP%';

**23. Display the first name and job id of the representatives**

select first\_name,job\_id from employees where job\_id like '%REP%';

**24. Display the sum of the salary of the employees**

select sum(salary) from employees;

**25. Display the sum, the smallest, the largest salary, the number of salaries, and the average salary of the employees**

select sum(salary),min(salary),max(salary),count(salary),avg(salary) from employees;

**26. Display the salaries that each department**

select department\_id, sum(salary) from employees group by department\_id;

**27. Display the job ids, departments, and the salaries that they make**

select department\_id,job\_id, sum(salary) from employees group by department\_id,job\_id;

**28. Display the departments who have salaries more than 50,000 and their salaries**

select department\_id, sum(salary) from employees group by department\_id having sum(salary)>50000 order by sum(salary) desc;

**29. Round any value to two decimal digits**

ROUND(4.93456,2)

**30. Add a ‘date of join’ to the sample table by altering it**

alter table samp add(doj date);

**31. Update the sample table and add joining dates**

update samp set doj = '01-jan-2021';

update samp set doj = '22-mar-2021' where regno = 102;

**32. Display the employee ids, the names, the hire dates, and the time that they were in the company in days**

select employee\_id, first\_name, hire\_date, sysdate-hire\_date as exp from employees;

**33. Display the same as above, but with the time in months**

select employee\_id, first\_name, hire\_date, ((sysdate-hire\_date)/30) as exp from employees;

**34. Display the same as the above, but with the time in months and with no decimal points**

select employee\_id, first\_name, hire\_date, round(((sysdate-hire\_date)/30),0) as exp from employees;

**35. Display the names and departments of employees in certain departments**

select first\_name, department\_id from employees where department\_id in(10,20,30);

**36. Display everybody who weren’t in the above selection**

select first\_name, department\_id from employees where department\_id not in(10,20,30);

**37. Display the commissions that each employee has received**

select employee\_id, commission\_pct from employees;

**38. Display the employees who have a null commission**

select employee\_id, commission\_pct from employees where commission\_pct is null;

**39. Display the employees who don’t have a null commission**

select employee\_id, commission\_pct from employees where commission\_pct is not null;

**Spool File Code:**

SQL> spool

currently spooling to D:/Ora/lab.txt

SQL> @d:/Ora/hr\_main;

SQL> Select \* from Employees;

EMPLOYEE\_ID FIRST\_NAME LAST\_NAME

----------- -------------------- -------------------------

EMAIL PHONE\_NUMBER HIRE\_DATE JOB\_ID SALARY

------------------------- -------------------- --------- ---------- ----------

COMMISSION\_PCT MANAGER\_ID DEPARTMENT\_ID

-------------- ---------- -------------

100 Steven King

SKING 515.123.4567 17-JUN-87 AD\_PRES 24000

90

101 Neena Kochhar

NKOCHHAR 515.123.4568 21-SEP-89 AD\_VP 17000

100 90

102 Lex De Haan

LDEHAAN 515.123.4569 13-JAN-93 AD\_VP 17000

100 90

103 Alexander Hunold

AHUNOLD 590.423.4567 03-JAN-90 IT\_PROG 9000

102 60

104 Bruce Ernst

BERNST 590.423.4568 21-MAY-91 IT\_PROG 6000

103 60

105 David Austin

DAUSTIN 590.423.4569 25-JUN-97 IT\_PROG 4800

103 60

106 Valli Pataballa

VPATABAL 590.423.4560 05-FEB-98 IT\_PROG 4800

103 60

107 Diana Lorentz

DLORENTZ 590.423.5567 07-FEB-99 IT\_PROG 4200

103 60

108 Nancy Greenberg

NGREENBE 515.124.4569 17-AUG-94 FI\_MGR 12000

101 100

109 Daniel Faviet

DFAVIET 515.124.4169 16-AUG-94 FI\_ACCOUNT 9000

108 100

110 John Chen

JCHEN 515.124.4269 28-SEP-97 FI\_ACCOUNT 8200

108 100

111 Ismael Sciarra

ISCIARRA 515.124.4369 30-SEP-97 FI\_ACCOUNT 7700

108 100

112 Jose Manuel Urman

JMURMAN 515.124.4469 07-MAR-98 FI\_ACCOUNT 7800

108 100

113 Luis Popp

LPOPP 515.124.4567 07-DEC-99 FI\_ACCOUNT 6900

108 100

….

107 rows selected.

SQL> create table samp(regno number, name varchar(15));

Table created.

SQL> insert into samp values(&No, '&name');

Enter value for no: 101

Enter value for name: naveen

1 row created.

SQL> /

Enter value for no: 102

Enter value for name: kevin

1 row created.

SQL> select \* from samp;

REGNO NAME

---------- ---------------

101 naveen

102 kevin

2 rows selected.

SQL> desc employees;

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPLOYEE\_ID NOT NULL NUMBER(6)

FIRST\_NAME VARCHAR2(20)

LAST\_NAME NOT NULL VARCHAR2(25)

EMAIL NOT NULL VARCHAR2(25)

PHONE\_NUMBER VARCHAR2(20)

HIRE\_DATE NOT NULL DATE

JOB\_ID NOT NULL VARCHAR2(10)

SALARY NUMBER(8,2)

COMMISSION\_PCT NUMBER(2,2)

MANAGER\_ID NUMBER(6)

DEPARTMENT\_ID NUMBER(4)

SQL> select first\_name, employee\_id, salary from employees;

FIRST\_NAME EMPLOYEE\_ID SALARY

-------------------- ----------- ----------

Steven 100 24000

Neena 101 17000

Lex 102 17000

Alexander 103 9000

Bruce 104 6000

David 105 4800

Valli 106 4800

Diana 107 4200

….

107 rows selected.

SQL> select &c1, &c2m, &c3 from employees;

Enter value for c1: salary

Enter value for c2m: Last\_name

Enter value for c3: job\_id

SALARY LAST\_NAME JOB\_ID

---------- ------------------------- ----------

24000 King AD\_PRES

17000 Kochhar AD\_VP

17000 De Haan AD\_VP

9000 Hunold IT\_PROG

6000 Ernst IT\_PROG

4800 Austin IT\_PROG

4800 Pataballa IT\_PROG

4200 Lorentz IT\_PROG

12000 Greenberg FI\_MGR

9000 Faviet FI\_ACCOUNT

8200 Chen FI\_ACCOUNT

7700 Sciarra FI\_ACCOUNT

7800 Urman FI\_ACCOUNT

6900 Popp FI\_ACCOUNT

….

107 rows selected.

SQL> select first\_name, employee\_id, salary from employees where salary>10000;

FIRST\_NAME EMPLOYEE\_ID SALARY

-------------------- ----------- ----------

Steven 100 24000

Neena 101 17000

Lex 102 17000

Nancy 108 12000

Den 114 11000

John 145 14000

Karen 146 13500

Alberto 147 12000

Gerald 148 11000

Eleni 149 10500

Clara 162 10500

Lisa 168 11500

Ellen 174 11000

Michael 201 13000

Shelley 205 12000

15 rows selected.

SQL> select first\_name, employee\_id, salary from employees where first\_name='Michael';

FIRST\_NAME EMPLOYEE\_ID SALARY

-------------------- ----------- ----------

Michael 134 2900

Michael 201 13000

2 rows selected.

SQL> select first\_name, lower(first\_name) from employees;

FIRST\_NAME LOWER(FIRST\_NAME)

-------------------- --------------------

Ellen ellen

Sundar sundar

Mozhe mozhe

David david

Hermann hermann

Shelli shelli

Amit amit

Elizabeth elizabeth

Sarah sarah

David david

Laura laura

Harrison harrison

Alexis alexis

Anthony anthony

Gerald gerald

Nanette nanette

….

107 rows selected.

SQL> select first\_name, lower(first\_name),upper(first\_name),initcap(first\_name) from employees;

FIRST\_NAME LOWER(FIRST\_NAME) UPPER(FIRST\_NAME)

-------------------- -------------------- --------------------

INITCAP(FIRST\_NAME)

--------------------

Ellen ellen ELLEN

Ellen

Sundar sundar SUNDAR

Sundar

Mozhe mozhe MOZHE

Mozhe

David david DAVID

David

Hermann hermann HERMANN

Hermann

Shelli shelli SHELLI

Shelli

Amit amit AMIT

Amit

Elizabeth elizabeth ELIZABETH

Elizabeth

Sarah sarah SARAH

Sarah

David david DAVID

David

Laura laura LAURA

Laura

Harrison harrison HARRISON

Harrison

….

107 rows selected.

SQL> set linesize 1500;

SQL> select first\_name, lower(first\_name),upper(first\_name),initcap(first\_name) from employees;

FIRST\_NAME LOWER(FIRST\_NAME) UPPER(FIRST\_NAME) INITCAP(FIRST\_NAME)

-------------------- -------------------- -------------------- --------------------

Ellen ellen ELLEN Ellen

Sundar sundar SUNDAR Sundar

Mozhe mozhe MOZHE Mozhe

David david DAVID David

Hermann hermann HERMANN Hermann

Shelli shelli SHELLI Shelli

Amit amit AMIT Amit

Elizabeth elizabeth ELIZABETH Elizabeth

Sarah sarah SARAH Sarah

David david DAVID David

Laura laura LAURA Laura

Harrison harrison HARRISON Harrison

Alexis alexis ALEXIS Alexis

Anthony anthony ANTHONY Anthony

Gerald gerald GERALD Gerald

Nanette nanette NANETTE Nanette

….

107 rows selected.

SQL> select employee\_id, first\_name from employees where lower(first\_name)='michael';

EMPLOYEE\_ID FIRST\_NAME

----------- --------------------

134 Michael

201 Michael

2 rows selected.

SQL> select first\_name, salary from employees order by salary;

FIRST\_NAME SALARY

-------------------- ----------

TJ 2100

Steven 2200

Hazel 2200

James 2400

Ki 2400

Karen 2500

James 2500

Joshua 2500

Peter 2500

Martha 2500

Randall 2500

Guy 2600

Randall 2600

Donald 2600

Douglas 2600

Irene 2700

….

107 rows selected.

SQL> select first\_name, salary from employees order by salary asc;

FIRST\_NAME SALARY

-------------------- ----------

TJ 2100

Steven 2200

Hazel 2200

James 2400

Ki 2400

Karen 2500

James 2500

Joshua 2500

Peter 2500

Martha 2500

Randall 2500

Guy 2600

Randall 2600

Donald 2600

Douglas 2600

Irene 2700

….

107 rows selected.

SQL> select first\_name, salary from employees order by salary desc;

FIRST\_NAME SALARY

-------------------- ----------

Steven 24000

Neena 17000

Lex 17000

John 14000

Karen 13500

Michael 13000

Nancy 12000

Alberto 12000

Shelley 12000

Lisa 11500

Ellen 11000

Gerald 11000

Den 11000

Eleni 10500

Clara 10500

Janette 10000

….

107 rows selected.

SQL> select first\_name, salary from employees order by first\_name;

FIRST\_NAME SALARY

-------------------- ----------

Adam 8200

Alana 3100

Alberto 12000

Alexander 9000

Alexander 3100

Alexis 4100

Allan 9000

Alyssa 8800

Amit 6200

Anthony 3000

Britney 3900

Bruce 6000

Charles 6200

Christopher 8000

Clara 10500

Curtis 3100

Daniel 9000

Danielle 9500

David 4800

….

107 rows selected.

SQL> spool

currently spooling to D:/Ora/lab.txt

SQL> select first\_name from employees where first\_name like 'A%';

FIRST\_NAME

--------------------

Amit

Alexis

Anthony

Alberto

Adam

Alexander

Alyssa

Alexander

Allan

Alana

10 rows selected.

SQL> select first\_name from employees where first\_name like 'A%A';

no rows selected

SQL> select first\_name from employees where first\_name like 'A%a';

FIRST\_NAME

--------------------

Alyssa

Alana

2 rows selected.

SQL> select first\_name from employees where first\_name like 'A%'and first\_name like '%a';

FIRST\_NAME

--------------------

Alyssa

Alana

2 rows selected.

SQL> select first\_name from employees where first\_name like 'A%' or first\_name like '%a';

FIRST\_NAME

--------------------

Amit

Laura

Alexis

Anthony

Julia

Alberto

Adam

Alexander

Alyssa

Alexander

Neena

Sundita

Diana

….

24 rows selected.

SQL> select first\_name from employees where first\_name like '\_a%';

FIRST\_NAME

--------------------

David

Sarah

David

Laura

Harrison

Nanette

Karen

Daniel

Pat

Tayler

Nancy

Danielle

Vance

….

32 rows selected.

SQL> select first\_name from employees where first\_name like '\_\_a%';

FIRST\_NAME

--------------------

Jean

Adam

Charles

Diana

Clara

Shanta

Alana

7 rows selected.

SQL> select first\_name from employees where first\_name like '\_\_n%';

FIRST\_NAME

--------------------

Sundar

Nanette

Jennifer

Daniel

Nancy

Danielle

Vance

Janette

Sundita

Renske

Randall

….

19 rows selected.

SQL> select first\_name from employees where job\_id like '%REP%';

FIRST\_NAME

--------------------

Peter

David

Peter

Christopher

Nanette

Oliver

Janette

Patrick

Allan

Lindsey

Louise

Sarath

Clara

Danielle

Mattea

David

….

33 rows selected.

SQL> select first\_name,job\_id from employees where job\_id like '%REP%';

FIRST\_NAME JOB\_ID

-------------------- ----------

Peter SA\_REP

David SA\_REP

Peter SA\_REP

Christopher SA\_REP

Nanette SA\_REP

Oliver SA\_REP

Janette SA\_REP

Patrick SA\_REP

Allan SA\_REP

Lindsey SA\_REP

Louise SA\_REP

Sarath SA\_REP

Clara SA\_REP

Danielle SA\_REP

Mattea SA\_REP

David SA\_REP

….

33 rows selected.

SQL> select sum(salary) from employees;

SUM(SALARY)

-----------

691400

1 row selected.

SQL> select sum(salary),min(salary),max(salary),count(salary),avg(salary) from employees;

SUM(SALARY) MIN(SALARY) MAX(SALARY) COUNT(SALARY) AVG(SALARY)

----------- ----------- ----------- ------------- -----------

691400 2100 24000 107 6461.68224

1 row selected.

SQL> select department\_id, sum(salary) from employees group by department\_id;

DEPARTMENT\_ID SUM(SALARY)

------------- -----------

100 51600

30 24900

7000

90 58000

20 19000

70 10000

110 20300

50 156400

80 304500

40 6500

60 28800

10 4400

12 rows selected.

SQL> select department\_id,job\_id, sum(salary) from employees group by department\_id,job\_id;

DEPARTMENT\_ID JOB\_ID SUM(SALARY)

------------- ---------- -----------

110 AC\_ACCOUNT 8300

90 AD\_VP 34000

50 ST\_CLERK 55700

80 SA\_REP 243500

50 ST\_MAN 36400

80 SA\_MAN 61000

110 AC\_MGR 12000

90 AD\_PRES 24000

60 IT\_PROG 28800

100 FI\_MGR 12000

30 PU\_CLERK 13900

50 SH\_CLERK 64300

20 MK\_MAN 13000

100 FI\_ACCOUNT 39600

SA\_REP 7000

70 PR\_REP 10000

30 PU\_MAN 11000

10 AD\_ASST 4400

20 MK\_REP 6000

40 HR\_REP 6500

20 rows selected.

SQL> select department\_id, sum(salary) from employees group by department\_id having sum(salary)>50000;

DEPARTMENT\_ID SUM(SALARY)

------------- -----------

100 51600

90 58000

50 156400

80 304500

4 rows selected.

SQL> select department\_id, sum(salary) from employees group by department\_id having sum(salary)>50000 order by sum(salary) desc;

DEPARTMENT\_ID SUM(SALARY)

------------- -----------

80 304500

50 156400

90 58000

100 51600

4 rows selected.

ROUND(4.93456,2)

----------------

4.93

1 row selected.

SQL> desc dual

Name Null? Type

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- -------- --------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

DUMMY VARCHAR2(1)

SQL>

SQL>

SQL> alter table samp add(doj date);

Table altered.

SQL> select \* from samp;

REGNO NAME DOJ

---------- --------------- ---------

101 naveen

102 kevin

2 rows selected.

SQL> update samp set doj = '01-jan-2021';

2 rows updated.

SQL> select \* from samp;

REGNO NAME DOJ

---------- --------------- ---------

101 naveen 01-JAN-21

102 kevin 01-JAN-21

2 rows selected.

SQL> update samp set doj = '22-mar-2021' where regno = 102;

1 row updated.

SQL> select sysdate from dual;

SYSDATE

---------

28-JUL-22

1 row selected.

SQL> select employee\_id, first\_name, hire\_date, sysdate-hire\_date as exp from employees;

EMPLOYEE\_ID FIRST\_NAME HIRE\_DATE EXP

----------- -------------------- --------- ----------

100 Steven 17-JUN-87 12825.4862

101 Neena 21-SEP-89 11998.4862

102 Lex 13-JAN-93 10788.4862

103 Alexander 03-JAN-90 11894.4862

104 Bruce 21-MAY-91 11391.4862

105 David 25-JUN-97 9164.48624

106 Valli 05-FEB-98 8939.48624

107 Diana 07-FEB-99 8572.48624

108 Nancy 17-AUG-94 10207.4862

109 Daniel 16-AUG-94 10208.4862

110 John 28-SEP-97 9069.48624

111 Ismael 30-SEP-97 9067.48624

112 Jose Manuel 07-MAR-98 8909.48624

113 Luis 07-DEC-99 8269.48624

114 Den 07-DEC-94 10095.4862

115 Alexander 18-MAY-95 9933.48624

116 Shelli 24-DEC-97 8982.48624

117 Sigal 24-JUL-97 9135.48624

….

107 rows selected.

SQL> select employee\_id, first\_name, hire\_date, ((sysdate-hire\_date)/30) as exp from employees;

EMPLOYEE\_ID FIRST\_NAME HIRE\_DATE EXP

----------- -------------------- --------- ----------

100 Steven 17-JUN-87 427.516266

101 Neena 21-SEP-89 399.949599

102 Lex 13-JAN-93 359.616266

103 Alexander 03-JAN-90 396.482932

104 Bruce 21-MAY-91 379.716266

105 David 25-JUN-97 305.482932

106 Valli 05-FEB-98 297.982932

107 Diana 07-FEB-99 285.749599

108 Nancy 17-AUG-94 340.249599

109 Daniel 16-AUG-94 340.282932

110 John 28-SEP-97 302.316266

111 Ismael 30-SEP-97 302.249599

112 Jose Manuel 07-MAR-98 296.982932

113 Luis 07-DEC-99 275.649599

114 Den 07-DEC-94 336.516266

115 Alexander 18-MAY-95 331.116266

116 Shelli 24-DEC-97 299.416266

117 Sigal 24-JUL-97 304.516266

….

107 rows selected.

SQL> select employee\_id, first\_name, hire\_date, round(((sysdate-hire\_date)/30),0) as exp from employees;

EMPLOYEE\_ID FIRST\_NAME HIRE\_DATE EXP

----------- -------------------- --------- ----------

100 Steven 17-JUN-87 428

101 Neena 21-SEP-89 400

102 Lex 13-JAN-93 360

103 Alexander 03-JAN-90 396

104 Bruce 21-MAY-91 380

105 David 25-JUN-97 305

106 Valli 05-FEB-98 298

107 Diana 07-FEB-99 286

108 Nancy 17-AUG-94 340

109 Daniel 16-AUG-94 340

110 John 28-SEP-97 302

111 Ismael 30-SEP-97 302

112 Jose Manuel 07-MAR-98 297

113 Luis 07-DEC-99 276

114 Den 07-DEC-94 337

115 Alexander 18-MAY-95 331

116 Shelli 24-DEC-97 299

117 Sigal 24-JUL-97 305

118 Guy 15-NOV-98 289

119 Karen 10-AUG-99 280

….

107 rows selected.

SQL> select first\_name, department\_id from employees where department\_id in(10,20,30);

FIRST\_NAME DEPARTMENT\_ID

-------------------- -------------

Jennifer 10

Michael 20

Pat 20

Den 30

Alexander 30

Shelli 30

Sigal 30

Guy 30

Karen 30

9 rows selected.

SQL> select first\_name, department\_id from employees where department\_id not in(10,20,30);

FIRST\_NAME DEPARTMENT\_ID

-------------------- -------------

Steven 90

Neena 90

Lex 90

Alexander 60

Bruce 60

David 60

Valli 60

Diana 60

Nancy 100

Daniel 100

John 100

Ismael 100

Jose Manuel 100

Luis 100

Matthew 50

Adam 50

….

97 rows selected.

SQL> select employee\_id, commission\_pct from employees;

EMPLOYEE\_ID COMMISSION\_PCT

----------- --------------

….

143

144

145 .4

146 .3

147 .3

148 .3

149 .2

150 .3

151 .25

152 .25

153 .2

….

107 rows selected.

SQL> select employee\_id, commission\_pct from employees where commission\_pct is null;

EMPLOYEE\_ID COMMISSION\_PCT

----------- --------------

100

101

102

103

104

105

106

107

108

109

110

111

112

113

….

72 rows selected.

SQL> select employee\_id, commission\_pct from employees where commission\_pct is not null;

EMPLOYEE\_ID COMMISSION\_PCT

----------- --------------

145 .4

146 .3

147 .3

148 .3

149 .2

150 .3

151 .25

152 .25

153 .2

….

35 rows selected.

SQL> spool off

**Ex.No : 3 Date: 04/08/2022**

**Experiment 3**

**QUERY:**

**1. Create the Publisher, Book, Book\_Authors, Library Branch, Book\_Copies, Card, and Book Lending tables in the respective orders. Input the fixed values.**

**2. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch etc.**

Select A.Book\_id, A.Title, A.pub\_name, B.author\_name, C.no\_of\_copies, D.Branch\_id from Book A, Book\_Authors B, Book\_copies C, Library\_Branch D where A.book\_id = B.Book\_id and A.book\_id=C.Book\_id and C.Branch\_id=D.Branch\_id;

**3. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017**

Select card\_no from book\_lending where date\_out between '01-JAN-2017' and '01-JUN-2020' group by card\_no having count(\*)>3;

**4. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.**

delete from book where book\_id = 4;

SELECT \* FROM BOOK;

SELECT \* FROM BOOK\_AUTHORS;

SELECT \* FROM LIBRARY\_BRANCH;

SELECT \* FROM BOOK\_COPIES;

SELECT \* FROM CARD;

SELECT \* FROM BOOK\_LENDING;

**5. Partition the BOOK table based on the year of publication. Demonstrate its working with a simple query.**

create view V\_Publications as select pub\_year from book;

**6. Create a view of all books and its number of copies that are currently available in the Library**

create view bcop as select A.Title, A.Book\_id, B.no\_of\_copies, C.Branch\_id from Book A, Book\_copies B, Library\_Branch C where A.Book\_id = B.Book\_id and B.Branch\_id = C.Branch\_id;

**Spool File Code:**

SQL> spool

currently spooling to C:\2162019\DBMS\Exp4.txt

SQL> create table book\_authors(Author\_name varchar(20), Book\_id references book(Book\_id) on delete cascade, primary key(Book\_id,author\_name));

Table created.

SQL> desc book\_authors;

Name Null? Type

----------------------------------------- -------- ----------------------------

AUTHOR\_NAME NOT NULL VARCHAR2(20)

BOOK\_ID NOT NULL NUMBER(38)

SQL> create table library\_branch(Branch\_id integer primary key, Address varchar(20), Branch\_name varchar(20));

Table created.

SQL> create table book\_copies(no\_of\_copies integer, Book\_id references book(book\_id) on delete cascade, Branch\_id references library\_branch(Branch\_id) on delete cascade);

Table created.

SQL> create table card(Card\_no integer primary key);

Table created.

SQL> create table book\_lending(Date\_Out date, Due\_date date, Branch\_id references library\_branch(Branch\_id) on delete cascade, Book\_id references book(book\_id) on delete cascade, Card\_no references card(card\_no) on delete cascade, primary key(Branch\_id, Book\_id, Card\_no));

Table created.

SQL> desc publisher;

Name Null? Type

----------------------------------------- -------- ----------------------------

NAME NOT NULL VARCHAR2(20)

PHONE NUMBER(38)

ADDRESS VARCHAR2(20)

SQL> INSERT INTO PUBLISHER VALUES ('MCGRAW-HILL', 9989076587,'BANGALORE');

1 row created.

SQL> INSERT INTO PUBLISHER VALUES ('PEARSON', 9889076565, 'NEWDELHI');

1 row created.

SQL> INSERT INTO PUBLISHER VALUES ('RANDOM HOUSE', 7455679345,'HYDRABAD');

1 row created.

SQL> INSERT INTO PUBLISHER VALUES ('HACHETTE LIVRE', 8970862340, 'CHENAI');

1 row created.

SQL> INSERT INTO PUBLISHER VALUES ('GRUPO PLANETA', 7756120238, 'BANGALORE');

1 row created.

SQL> desc book;

Name Null? Type

----------------------------------------- -------- ----------------------------

BOOK\_ID NOT NULL NUMBER(38)

TITLE VARCHAR2(20)

PUB\_YEAR NUMBER(38)

PUB\_NAME VARCHAR2(20)

SQL> INSERT INTO BOOK VALUES (1,'DBMS',2017, 'MCGRAW-HILL');

1 row created.

SQL> INSERT INTO BOOK VALUES (2,'ADBMS',2016,'MCGRAW-HILL');

1 row created.

SQL> INSERT INTO BOOK VALUES (3,'CN',2016,'PEARSON');

1 row created.

SQL> INSERT INTO BOOK VALUES (5,'OS',2016,'PEARSON');

1 row created.

SQL> desc book;

Name Null? Type

----------------------------------------- -------- ----------------------------

BOOK\_ID NOT NULL NUMBER(38)

TITLE VARCHAR2(20)

PUB\_YEAR NUMBER(38)

PUB\_NAME VARCHAR2(20)

SQL> select \* from book;

BOOK\_ID TITLE PUB\_YEAR PUB\_NAME

---------- -------------------- ---------- --------------------

1 DBMS 2017 MCGRAW-HILL

2 ADBMS 2016 MCGRAW-HILL

3 CN 2016 PEARSON

5 OS 2016 PEARSON

SQL> INSERT INTO BOOK VALUES (4,'CG',2015, 'GRUPO PLANETA');

1 row created.

SQL> desc book\_authors

Name Null? Type

----------------------------------------- -------- ----------------------------

AUTHOR\_NAME NOT NULL VARCHAR2(20)

BOOK\_ID NOT NULL NUMBER(38)

SQL> INSERT INTO BOOK\_AUTHORS VALUES('NAVATHE', 1);

1 row created.

SQL> INSERT INTO BOOK\_AUTHORS VALUES ('NAVATHE', 2);

1 row created.

SQL> INSERT INTO BOOK\_AUTHORS VALUES ('TANENBAUM', 3);

1 row created.

SQL> INSERT INTO BOOK\_AUTHORS VALUES ('EDWARDANGEL', 4);

1 row created.

SQL> INSERT INTO BOOK\_AUTHORS VALUES('GALVIN', 5);

1 row created.

SQL> desc library\_branch;

Name Null? Type

----------------------------------------- -------- ----------------------------

BRANCH\_ID NOT NULL NUMBER(38)

ADDRESS VARCHAR2(20)

BRANCH\_NAME VARCHAR2(20)

SQL> INSERT INTO LIBRARY\_BRANCH VALUES (10,'BANGALORE','RR NAGAR');

1 row created.

SQL> INSERT INTO LIBRARY\_BRANCH VALUES (11,'BANGALORE','RNSIT');

1 row created.

SQL> INSERT INTO LIBRARY\_BRANCH VALUES (12, 'BANGALORE','RAJAJI NAGAR');

1 row created.

SQL> INSERT INTO LIBRARY\_BRANCH VALUES (13,'MANGALORE','NITTE');

1 row created.

SQL> INSERT INTO LIBRARY\_BRANCH VALUES (14,'UDUPI','MANIPAL');

1 row created.

SQL> desc book\_copies;

Name Null? Type

----------------------------------------- -------- ----------------------------

NO\_OF\_COPIES NUMBER(38)

BOOK\_ID NUMBER(38)

BRANCH\_ID NUMBER(38)

SQL> INSERT INTO BOOK\_COPIES VALUES (10, 1, 10);

1 row created.

SQL> INSERT INTO BOOK\_COPIES VALUES (5, 1, 11);

1 row created.

SQL> INSERT INTO BOOK\_COPIES VALUES (2, 2, 12);

1 row created.

SQL> INSERT INTO BOOK\_COPIES VALUES (5, 2, 13);

1 row created.

SQL> INSERT INTO BOOK\_COPIES VALUES (7, 3, 14);

1 row created.

SQL> INSERT INTO BOOK\_COPIES VALUES (1, 5, 10);

1 row created.

SQL> INSERT INTO BOOK\_COPIES VALUES (3, 4, 11);

1 row created.

SQL> INSERT INTO CARD VALUES (100);

1 row created.

SQL> INSERT INTO CARD VALUES (101);

1 row created.

SQL> INSERT INTO CARD VALUES (102);

1 row created.

SQL> INSERT INTO CARD VALUES (103);

1 row created.

SQL> INSERT INTO CARD VALUES (104);

1 row created.

SQL> desc book\_lending;

Name Null? Type

----------------------------------------- -------- ----------------------------

DATE\_OUT DATE

DUE\_DATE DATE

BRANCH\_ID NOT NULL NUMBER(38)

BOOK\_ID NOT NULL NUMBER(38)

CARD\_NO NOT NULL NUMBER(38)

SQL> INSERT INTO BOOK\_LENDING VALUES ('01-JAN-17','01-JUN-17', 10, 1, 101);

1 row created.

SQL> INSERT INTO BOOK\_LENDING VALUES ('11-JAN-17','11-MAR-17', 14, 3, 101);

1 row created.

SQL> INSERT INTO BOOK\_LENDING VALUES ('21-FEB-17','21-APR-17', 13, 2, 101);

1 row created.

SQL> INSERT INTO BOOK\_LENDING VALUES ('15-MAR-17','15-JUL-17', 11, 4, 101);

1 row created.

SQL> INSERT INTO BOOK\_LENDING VALUES ('12-APR-17','12-MAY-17', 11, 1, 104);

1 row created.

SQL> Select \* from publisher;

NAME PHONE ADDRESS

-------------------- ---------- --------------------

MCGRAW-HILL 9989076587 BANGALORE

PEARSON 9889076565 NEWDELHI

RANDOM HOUSE 7455679345 HYDRABAD

HACHETTE LIVRE 8970862340 CHENAI

GRUPO PLANETA 7756120238 BANGALORE

SQL> Select \* from book;

BOOK\_ID TITLE PUB\_YEAR PUB\_NAME

---------- -------------------- ---------- --------------------

1 DBMS 2017 MCGRAW-HILL

2 ADBMS 2016 MCGRAW-HILL

3 CN 2016 PEARSON

5 OS 2016 PEARSON

4 CG 2015 GRUPO PLANETA

SQL> Select \* from book\_authors;

AUTHOR\_NAME BOOK\_ID

-------------------- ----------

NAVATHE 1

NAVATHE 2

TANENBAUM 3

EDWARDANGEL 4

GALVIN 5

SQL> Select \* from library\_branch;

BRANCH\_ID ADDRESS BRANCH\_NAME

---------- -------------------- --------------------

10 BANGALORE RR NAGAR

11 BANGALORE RNSIT

12 BANGALORE RAJAJI NAGAR

13 MANGALORE NITTE

14 UDUPI MANIPAL

SQL> Select \* from book\_copies;

NO\_OF\_COPIES BOOK\_ID BRANCH\_ID

------------ ---------- ----------

10 1 10

5 1 11

2 2 12

5 2 13

7 3 14

1 5 10

3 4 11

7 rows selected.

SQL> Select \* from card;

CARD\_NO

----------

100

101

102

103

104

SQL> Select \* from book\_lending;

DATE\_OUT DUE\_DATE BRANCH\_ID BOOK\_ID CARD\_NO

--------- --------- ---------- ---------- ----------

01-JAN-17 01-JUN-17 10 1 101

11-JAN-17 11-MAR-17 14 3 101

21-FEB-17 21-APR-17 13 2 101

15-MAR-17 15-JUL-17 11 4 101

12-APR-17 12-MAY-17 11 1 104

SQL> spool off;

SQL> spool

currently spooling to C:\2162019\DBMS\Exp4.txt

SQL> Select A.Book\_id, B.author\_name from book A, Book\_authors B where A.Book\_id = B.Book\_id;

BOOK\_ID AUTHOR\_NAME

---------- --------------------

1 NAVATHE

2 NAVATHE

3 TANENBAUM

4 EDWARDANGEL

5 GALVIN

SQL> Select A.Book\_id, A.Title, A.pub\_name, B.author\_name, C.no\_of\_copies, D.Branch\_id from Book A, Book\_Authors B, Book\_copies C, Library\_Branch D where A.book\_id = B.Book\_id and A.book\_id=C.Book\_id and C.Branch\_id=D.Branch\_id;

BOOK\_ID TITLE PUB\_NAME AUTHOR\_NAME

---------- -------------------- -------------------- --------------------

NO\_OF\_COPIES BRANCH\_ID

------------ ----------

1 DBMS MCGRAW-HILL NAVATHE

10 10

1 DBMS MCGRAW-HILL NAVATHE

5 11

2 ADBMS MCGRAW-HILL NAVATHE

2 12

BOOK\_ID TITLE PUB\_NAME AUTHOR\_NAME

---------- -------------------- -------------------- --------------------

NO\_OF\_COPIES BRANCH\_ID

------------ ----------

2 ADBMS MCGRAW-HILL NAVATHE

5 13

3 CN PEARSON TANENBAUM

7 14

5 OS PEARSON GALVIN

1 10

BOOK\_ID TITLE PUB\_NAME AUTHOR\_NAME

---------- -------------------- -------------------- --------------------

NO\_OF\_COPIES BRANCH\_ID

------------ ----------

4 CG GRUPO PLANETA EDWARDANGEL

3 11

7 rows selected.

SQL> set linesize 1500;

SQL> Select A.Book\_id, A.Title, A.pub\_name, B.author\_name, C.no\_of\_copies, D.Branch\_id from Book A, Book\_Authors B, Book\_copies C, Library\_Branch D where A.book\_id = B.Book\_id and A.book\_id=C.Book\_id and C.Branch\_id=D.Branch\_id;

BOOK\_ID TITLE PUB\_NAME AUTHOR\_NAME NO\_OF\_COPIES BRANCH\_ID

---------- -------------------- -------------------- -------------------- ------------ ----------

1 DBMS MCGRAW-HILL NAVATHE 10 10

1 DBMS MCGRAW-HILL NAVATHE 5 11

2 ADBMS MCGRAW-HILL NAVATHE 2 12

2 ADBMS MCGRAW-HILL NAVATHE 5 13

3 CN PEARSON TANENBAUM 7 14

5 OS PEARSON GALVIN 1 10

4 CG GRUPO PLANETA EDWARDANGEL 3 11

7 rows selected.

SQL> Select card\_no from book\_lending where date\_out between '01-JAN-2017' and '01-JUN-2020' group by card\_no having count(\*)>3;

CARD\_NO

----------

101

SQL> delete from book where book\_id = 4;

1 row deleted.

SQL> SELECT \* FROM PUBLISHER;

NAME PHONE ADDRESS

-------------------- ---------- --------------------

MCGRAW-HILL 9989076587 BANGALORE

PEARSON 9889076565 NEWDELHI

RANDOM HOUSE 7455679345 HYDRABAD

HACHETTE LIVRE 8970862340 CHENAI

GRUPO PLANETA 7756120238 BANGALORE

SQL>

SQL> SELECT \* FROM BOOK;

BOOK\_ID TITLE PUB\_YEAR PUB\_NAME

---------- -------------------- ---------- --------------------

1 DBMS 2017 MCGRAW-HILL

2 ADBMS 2016 MCGRAW-HILL

3 CN 2016 PEARSON

5 OS 2016 PEARSON

SQL>

SQL> SELECT \* FROM BOOK\_AUTHORS;

AUTHOR\_NAME BOOK\_ID

-------------------- ----------

NAVATHE 1

NAVATHE 2

TANENBAUM 3

GALVIN 5

SQL>

SQL> SELECT \* FROM LIBRARY\_BRANCH;

BRANCH\_ID ADDRESS BRANCH\_NAME

---------- -------------------- --------------------

10 BANGALORE RR NAGAR

11 BANGALORE RNSIT

12 BANGALORE RAJAJI NAGAR

13 MANGALORE NITTE

14 UDUPI MANIPAL

SQL>

SQL> SELECT \* FROM BOOK\_COPIES;

NO\_OF\_COPIES BOOK\_ID BRANCH\_ID

------------ ---------- ----------

10 1 10

5 1 11

2 2 12

5 2 13

7 3 14

1 5 10

6 rows selected.

SQL>

SQL> SELECT \* FROM CARD;

CARD\_NO

----------

100

101

102

103

104

SQL>

SQL> SELECT \* FROM BOOK\_LENDING;

DATE\_OUT DUE\_DATE BRANCH\_ID BOOK\_ID CARD\_NO

--------- --------- ---------- ---------- ----------

01-JAN-17 01-JUN-17 10 1 101

11-JAN-17 11-MAR-17 14 3 101

21-FEB-17 21-APR-17 13 2 101

12-APR-17 12-MAY-17 11 1 104

SQL> create view V\_Publications as select pub\_year from book;

View created.

SQL> select \* from V\_publications

2 ;

PUB\_YEAR

----------

2017

2016

2016

2016

SQL> create view bcop as select A.Title, A.Book\_id, B.no\_of\_copies, C.Branch\_id from Book A, Book\_copies B, Library\_Branch C where A.Book\_id = B.Book\_id and B.Branch\_id = C.Branch\_id;

View created.

SQL> Select \* from bcop;

TITLE BOOK\_ID NO\_OF\_COPIES BRANCH\_ID

-------------------- ---------- ------------ ----------

DBMS 1 10 10

DBMS 1 5 11

ADBMS 2 2 12

ADBMS 2 5 13

CN 3 7 14

OS 5 1 10

6 rows selected.

SQL> spool off;

**Ex.No : 4 Date: 18/08/2022**

**Experiment 4**

**PROBLEM GIVEN:**

1. Count the customers with grades above Bangalore’s average.
2. Find the name and numbers of all salesmen who had more than one customer.
3. List all salesman and indicate those who have and don’t have customers in their cities (UNION operation)
4. Create a view that finds the salesman who has the customer with the highest order of a day.
5. Demonstrate the delete operation by removing salesman with id 1000. Ll his orders must also be deleted.

**QUERY:**

1. Select grade, count (distinct customer\_id) from customer1 group by grade having grade>(select avg(grade) from customer1 where city = 'Bangalore');
2. Select salesman\_id, name from salesman A where 1<(select count(\*) from customer1 where salesman\_id=A.salesman\_id);
3. Select salesman.salesman\_id, name, cust\_name, commission from salesman, customer1 where salesman.city = customer1.city union select salesman\_id, name, 'NO MATCH', commission from salesman where not city = any (select city from customer1) order by 2 desc;
4. create view eleitsalesman as select B.ord\_date, A.salesman\_id, A.name from salesman A, orders B where A.salesman\_id = B.salesman\_id and B.purchase\_amt=(select max(purchase\_amt) from orders C where C.ord\_date = B.ord\_date);
5. delete from salesman where salesman\_id = 1000;

**Spool File Code:**

SQL> spool

currently spooling to D:\Uni\Btech AI and ML\Sem 3\DBMS\Spool files\Exp7.txt

SQL> insert into salesman values (100,'John','Bangalore',25);

1 row created.

SQL> insert into salesman values (2000,'Ravi','Bangalore',20);

1 row created.

SQL> update salesman set salesman\_id = 1000 where salesman\_id = 100;

1 row updated.

SQL> insert into salesman values (3000,'Kumar','Mysore',15);

1 row created.

SQL> insert into salesman values (4000,'Smith','Delhi',30);

1 row created.

SQL> insert into salesman values (5000,'Harsha','Hydrabad',15);

1 row created.

SQL> insert into customer1 values (10, 'Preethi', 'Bangalore', 100, 1000);

1 row created.

SQL> insert into customer1 values (11, 'Vivek', 'Mangalore', 300, 1000);

1 row created.

SQL> insert into customer1 values (12, 'Bhaskar', 'Chennai', 400, 2000);

1 row created.

SQL>

SQL> insert into customer1 values (13, 'Chethan', 'Bangalore', 200, 2000);

1 row created.

SQL> insert into customer1 values (14, 'Mamatha', 'Bangalore', 400, 3000);

1 row created.

SQL> insert into orders values (50, 5000, '04-MAY-2017', 10, 1000);

1 row created.

SQL> insert into orders values (51, 450, '20-JAN-2017', 10, 2000);

1 row created.

SQL> insert into orders values (52, 1000, '24-FEB-2017', 13, 2000);

1 row created.

SQL> insert into orders values (53, 3500, '13-APR-2017', 14, 3000);

1 row created.

SQL> insert into orders values (54, 550, '09-MAR-2017', 12, 2000);

1 row created.

SQL> Select \* from salesman;

SALESMAN\_ID NAME CITY COMMISSION

----------- -------------------- -------------------- ----------

1000 John Bangalore 25

2000 Ravi Bangalore 20

3000 Kumar Mysore 15

4000 Smith Delhi 30

5000 Harsha Hydrabad 15

SQL> Select \* from customer1;

CUSTOMER\_ID CUST\_NAME CITY GRADE SALESMAN\_ID

----------- -------------------- -------------------- ---------- -----------

10 Preethi Bangalore 100 1000

11 Vivek Mangalore 300 1000

12 Bhaskar Chennai 400 2000

13 Chethan Bangalore 200 2000

14 Mamatha Bangalore 400 3000

SQL> Select \* from orders;

ORD\_NO PURCHASE\_AMT ORD\_DATE CUSTOMER\_ID SALESMAN\_ID

---------- ------------ --------- ----------- -----------

50 5000 04-MAY-17 10 1000

51 450 20-JAN-17 10 2000

52 1000 24-FEB-17 13 2000

53 3500 13-APR-17 14 3000

54 550 09-MAR-17 12 2000

SQL> Select grade, count (distinct customer\_id) from customer1 group by grade having grade>(select avg(grade) from customer1 where city = 'Bangalore');

GRADE COUNT(DISTINCTCUSTOMER\_ID)

---------- --------------------------

400 2

300 1

SQL> Select salesman\_id, name from salesman A where 1<(select count(\*) from customer1 where salesman\_id=A.salesman\_id);

SALESMAN\_ID NAME

----------- --------------------

1000 John

2000 Ravi

SQL> Select salesman.salesman\_id, name, cust\_name, commission from salesman, customer1 where salesman.city = customer1.city union select salesman\_id, name, 'NO MATCH', commission from salesman where not city = any (select city from customer1) order by 2 desc;

SALESMAN\_ID NAME CUST\_NAME COMMISSION

----------- -------------------- -------------------- ----------

4000 Smith NO MATCH 30

2000 Ravi Chethan 20

2000 Ravi Mamatha 20

2000 Ravi Preethi 20

3000 Kumar NO MATCH 15

1000 John Chethan 25

1000 John Mamatha 25

1000 John Preethi 25

5000 Harsha NO MATCH 15

9 rows selected.

SQL> create view eleitsalesman as select B.ord\_date, A.salesman\_id, A.name from salesman A, orders B where A.salesman\_id = B.salesman\_id and B.purchase\_amt=(select max(purchase\_amt) from orders C where C.ord\_date = B.ord\_date);

View created.

SQL> Select \* from eleitsalesman;

ORD\_DATE SALESMAN\_ID NAME

--------- ----------- --------------------

04-MAY-17 1000 John

20-JAN-17 2000 Ravi

24-FEB-17 2000 Ravi

13-APR-17 3000 Kumar

09-MAR-17 2000 Ravi

SQL> delete from salesman where salesman\_id = 1000;

1 row deleted.

SQL> Select \* from salesman;

SALESMAN\_ID NAME CITY COMMISSION

----------- -------------------- -------------------- ----------

2000 Ravi Bangalore 20

3000 Kumar Mysore 15

4000 Smith Delhi 30

5000 Harsha Hydrabad 15

SQL> Select \* from orders;

ORD\_NO PURCHASE\_AMT ORD\_DATE CUSTOMER\_ID SALESMAN\_ID

---------- ------------ --------- ----------- -----------

52 1000 24-FEB-17 13 2000

53 3500 13-APR-17 14 3000

54 550 09-MAR-17 12 2000

SQL> spool off;

**Ex.No : 5 Date: 28/08/2022**

**Experiment 5**

**PROBLEM GIVEN:**

**INNER JOIN**

1. Show book titles along with their authors(ie., the author’s first name and last name)
2. Displaying books along with their translators (i.e. the translator’s last name). Only half of our books have been translated and they have a corresponding translator

**LEFT JOIN**

1. Display information about each books author and translator(i.e. their last names). We also want to keep the basic information about each book (i.e. id, title, type)
2. Show the basic book information (i.e. ID and title) along with the last names of the corresponding editors. Again we want to keep all of the books in the result set.

**RIGHT JOIN**

1. Lets repeat our previous example but this time, our task will be to keep all the records from the editors table.

**FULL JOIN**

1. Let’s again join the books and editors tables, but this time, we’ll be keeping all records from both tables
2. Join all four tables to get information about all the books, authors, editors and translators in one table.

**QUERY:**

**INNER JOIN**

1. Select B.Title, A.First\_name, A.Last\_name from Books B inner join Authors A on B.Author\_id = A.Id;
2. Select B.Title, T.Last\_name from Books B inner join Translators T on B.Translator\_id = T.ID;

**LEFT JOIN**

1. Select B.Id, B.Title, B.Type, A.Last\_name, Translators.Last\_name from Books B left join Authors A on B.Author\_id=A.ID left join Translators on B.Translator\_id = Translators.id;
2. Select B.Id, B.Title, E.Last\_name from Books B left join Editors E on B.Editor\_id = E.Id;

**RIGHT JOIN**

1. Select B.ID, B.Title, E.Last\_name from Editors E right join Books B on E.ID = B.Editor\_ID;

**FULL JOIN**

1. Select \* from Books B full join Editors E on B.ID = E.ID;
2. Select \* from Books B full join Editors on B.Editor\_id = Editors.ID full join Authors on B.Author\_id = Authors.ID full join Translators on B.Translator\_id = Translators.ID;

**Spool File Code:**

SQL> spool

currently spooling to C:\2162019\DBMS\Exp6.txt

SQL> create table Translators( Id number, First\_name varchar(12), Last\_name varchar(12), primary key (Id));

Table created.

SQL> create table Editors(ID number, First\_name varchar(12), Last\_name varchar(12), primary key (Id));

Table created.

SQL> create table Authors(Id number, First\_name varchar(12), Last\_name varchar(12), primary key(Id));

Table created.

SQL> create table books(Id number, Title varchar(30), Type varchar(12), primary key (Id), Author\_id references Authors(ID) on delete set NULL, Editor\_ID references Editors(ID) on delete set NULL, Translator\_id references Translators(ID) on delete set NULL);

Table created.

SQL> insert into Translators values(31,'Ira','Davies');

1 row created.

SQL> insert into Translators values(32,'Ling','Weng');

1 row created.

SQL> insert into Translators values(33,'Kristian','Green');

1 row created.

SQL> insert into Translators values(34,'Roman','Edwards');

1 row created.

SQL> insert into Editors values(21,'Daniel','Brown');

1 row created.

SQL> insert into Editors values(22,'Mark','Johnson');

1 row created.

SQL> insert into Editors values(23,'Maria','Evans');

1 row created.

SQL> insert into Editors values(24,'Catherine','Roberts');

1 row created.

SQL> insert into Editors values(25,'Sebastian','Wright');

1 row created.

SQL> insert into Editors values(26,'Barbara','Jones');

1 row created.

SQL> insert into Editors values(27,'Matthew','Smith');

1 row created.

SQL> insert into Authors values(11,'Ellen','Writer');

1 row created.

SQL> insert into Authors values(12,'Olga','Savalieva');

1 row created.

SQL> insert into Authors values(13,'Jack','Smart');

1 row created.

SQL> insert into Authors values(14,'Donald','Brain');

1 row created.

SQL> insert into Authors values(15,'Yao','Dou');

1 row created.

SQL> insert into Books values(1,'Time to Grow Up!', 'Original', 11, 21, NULL);

1 row created.

SQL> insert into Books values(2,'Your Trip', 'Translated', 15, 22, 32);

1 row created.

SQL> insert into Books values(3,'Lovely Love', 'Original', 14, 24, NULL);

1 row created.

SQL> insert into Books values(4,'Dream Your Life', 'Original', 11, 24, NULL);

1 row created.

SQL> insert into Books values(5,'Oranges', 'Translated', 12, 25, 31);

1 row created.

SQL> insert into Books values(6,'Your Happy Life', 'Translated', 15, 22, 33);

1 row created.

SQL> insert into Books values(7,'Applied AI', 'Translated', 13, 23, 34);

1 row created.

SQL> insert into Books values(8,'My Last Book', 'Translated', 11, 27, NULL);

1 row created.

SQL> spool off;

SQL> spool

currently spooling to C:\2162019\DBMS\Exp6.txt

SQL> Select \* from Books;

ID TITLE TYPE AUTHOR\_ID EDITOR\_ID

---------- ------------------------------ ------------ ---------- ----------

TRANSLATOR\_ID

-------------

1 Time to Grow Up! Original 11 21

2 Your Trip Translated 15 22

32

3 Lovely Love Original 14 24

ID TITLE TYPE AUTHOR\_ID EDITOR\_ID

---------- ------------------------------ ------------ ---------- ----------

TRANSLATOR\_ID

-------------

4 Dream Your Life Original 11 24

5 Oranges Translated 12 25

31

6 Your Happy Life Translated 15 22

33

ID TITLE TYPE AUTHOR\_ID EDITOR\_ID

---------- ------------------------------ ------------ ---------- ----------

TRANSLATOR\_ID

-------------

7 Applied AI Translated 13 23

34

8 My Last Book Translated 11 27

8 rows selected.

SQL> set linesize 150;

SQL> Select \* from Authors;

ID FIRST\_NAME LAST\_NAME

---------- ------------ ------------

11 Ellen Writer

12 Olga Savalieva

13 Jack Smart

14 Donald Brain

15 Yao Dou

SQL> Select \* from Editors;

ID FIRST\_NAME LAST\_NAME

---------- ------------ ------------

21 Daniel Brown

22 Mark Johnson

23 Maria Evans

24 Catherine Roberts

25 Sebastian Wright

26 Barbara Jones

27 Matthew Smith

7 rows selected.

SQL> Select \* from Translators;

ID FIRST\_NAME LAST\_NAME

---------- ------------ ------------

31 Ira Davies

32 Ling Weng

33 Kristian Green

34 Roman Edwards

SQL> Select B.Title, A.First\_name, A.Last\_name from Books B inner join Authors A on B.Author\_id = A.Id;

TITLE FIRST\_NAME LAST\_NAME

------------------------------ ------------ ------------

Time to Grow Up! Ellen Writer

Your Trip Yao Dou

Lovely Love Donald Brain

Dream Your Life Ellen Writer

Oranges Olga Savalieva

Your Happy Life Yao Dou

Applied AI Jack Smart

My Last Book Ellen Writer

8 rows selected.

SQL> Select B.Title, T.Last\_name from Books B inner join Translators T on B.Translator\_id = T.ID;

TITLE LAST\_NAME

------------------------------ ------------

Your Trip Weng

Oranges Davies

Your Happy Life Green

Applied AI Edwards

SQL> Select B.Id, B.Title, B.Type, A.Last\_name, Translators.Last\_name from Books B left join Authors A on B.Author\_id=A.ID left join Translators on B.Translator\_id = Translators.id;

ID TITLE TYPE LAST\_NAME LAST\_NAME

---------- ------------------------------ ------------ ------------ ------------

1 Time to Grow Up! Original Writer

4 Dream Your Life Original Writer

8 My Last Book Translated Writer

5 Oranges Translated Savalieva Davies

7 Applied AI Translated Smart Edwards

3 Lovely Love Original Brain

6 Your Happy Life Translated Dou Green

2 Your Trip Translated Dou Weng

8 rows selected.

SQL> Select B.Id, B.Title, E.Last\_name from Books B left join Editors E on B.Editor\_id = E.Id;

ID TITLE LAST\_NAME

---------- ------------------------------ ------------

1 Time to Grow Up! Brown

6 Your Happy Life Johnson

2 Your Trip Johnson

7 Applied AI Evans

4 Dream Your Life Roberts

3 Lovely Love Roberts

5 Oranges Wright

8 My Last Book Smith

8 rows selected.

SQL> Select B.ID, B.Title, E.Last\_name from Editors E left join Books B on E.ID = B.Editor\_ID;

ID TITLE LAST\_NAME

---------- ------------------------------ ------------

1 Time to Grow Up! Brown

2 Your Trip Johnson

3 Lovely Love Roberts

4 Dream Your Life Roberts

5 Oranges Wright

6 Your Happy Life Johnson

7 Applied AI Evans

8 My Last Book Smith

Jones

9 rows selected.

SQL> Select \* from Books B full join Editors E on B.ID = E.ID;

ID TITLE TYPE AUTHOR\_ID EDITOR\_ID TRANSLATOR\_ID ID FIRST\_NAME LAST\_NAME

---------- ------------------------------ ------------ ---------- ---------- ------------- ---------- ------------ ------------

1 Time to Grow Up! Original 11 21

2 Your Trip Translated 15 22 32

3 Lovely Love Original 14 24

4 Dream Your Life Original 11 24

5 Oranges Translated 12 25 31

6 Your Happy Life Translated 15 22 33

7 Applied AI Translated 13 23 34

8 My Last Book Translated 11 27

23 Maria Evans

26 Barbara Jones

22 Mark Johnson

ID TITLE TYPE AUTHOR\_ID EDITOR\_ID TRANSLATOR\_ID ID FIRST\_NAME LAST\_NAME

---------- ------------------------------ ------------ ---------- ---------- ------------- ---------- ------------ ------------

25 Sebastian Wright

21 Daniel Brown

24 Catherine Roberts

27 Matthew Smith

15 rows selected.

SQL> set linesize 500;

SQL> Select \* from Books B full join Editors on B.Editor\_id = Editors.ID full join Authors on B.Author\_id = Authors.ID full join Translators on B.Translator\_id = Translators.ID;

ID TITLE TYPE AUTHOR\_ID EDITOR\_ID TRANSLATOR\_ID ID FIRST\_NAME LAST\_NAME ID FIRST\_NAME LAST\_NAME ID FIRST\_NAME LAST\_NAME

---------- ------------------------------ ------------ ---------- ---------- ------------- ---------- ------------ ------------ ---------- ------------ ------------ ---------- ------------ ------------

1 Time to Grow Up! Original 11 21 21 Daniel Brown 11 Ellen Writer

2 Your Trip Translated 15 22 32 22 Mark Johnson 15 Yao Dou 32 Ling Weng

3 Lovely Love Original 14 24 24 Catherine Roberts 14 Donald Brain

4 Dream Your Life Original 11 24 24 Catherine Roberts 11 Ellen Writer

5 Oranges Translated 12 25 31 25 Sebastian Wright 12 Olga Savalieva 31 Ira Davies

6 Your Happy Life Translated 15 22 33 22 Mark Johnson 15 Yao Dou 33 Kristian Green

7 Applied AI Translated 13 23 34 23 Maria Evans 13 Jack Smart 34 Roman Edwards

8 My Last Book Translated 11 27 27 Matthew Smith 11 Ellen Writer

26 Barbara Jones

9 rows selected.

SQL> spool off;

**Ex.No : 6 Date: 01/09/2022**

**Experiment 6**

**PROBLEM GIVEN:**

1. Create a simple view for the clients table
2. Create a view that will store ID, name, country, and country\_code
3. Insert a new row into a view
4. Update a row in a view
5. Delete a row in a view
6. Drop a view

**QUERY:**

1. create view cl as select Id, Name from clients;
2. create view clco as select L.Id, L.Name, C.Country, C.Country\_code from Clients L, Countries1 C where L.Id = C.Id;
3. insert into cl values(4,'Morrison');
4. update cl set name = 'Morrisons' where Id = 4;
5. delete cl where Id=4;
6. drop view clco;

**Spool File Code:**

SQL> spool

currently spooling to D:\Uni\Btech AI and ML\Sem 3\DBMS\Spool files\Exp6.txt

SQL> create table clients( id number, name varchar(10), email\_id varchar(10), primary key (id));

Table created.

SQL> create table countries1(id number, country varchar(10), country\_code varchar(5), primary key(id));

Table created.

SQL> insert into clients values(1,'George','ge.com');

1 row created.

SQL> insert into clients values(2,'David','da.com');

1 row created.

SQL> insert into clients values(3,'Chirs','ch.com');

1 row created.

SQL> insert into clients values(4,'Morrison','mo.com');

1 row created.

SQL> insert into clients values(5,'Brian','br.com');

1 row created.

SQL> insert into countries1 values(1,'INDIA','IND');

1 row created.

SQL> insert into countries1 values(2,'SPAIN','ESP');

1 row created.

SQL> insert into countries1 values(3,'FRANCE','FRA');

1 row created.

SQL> insert into countries1 values(4,'ENGLAND','ENG');

1 row created.

SQL> insert into countries1 values(5,'POLAND','POL');

1 row created.

SQL> create view cl as select Id, Name from clients;

View created.

SQL> select \* from cl;

ID NAME

---------- ----------

1 George

2 David

3 Chirs

4 Morrison

5 Brian

SQL> create view clco as select L.Id, L.Name, C.Country, C.Country\_code from Clients L, Countries1 C ;

View created.

SQL> select \* from clco;

ID NAME COUNTRY COUNT

---------- ---------- ---------- -----

1 George INDIA IND

2 David INDIA IND

3 Chirs INDIA IND

4 Morrison INDIA IND

5 Brian INDIA IND

1 George SPAIN ESP

2 David SPAIN ESP

3 Chirs SPAIN ESP

4 Morrison SPAIN ESP

5 Brian SPAIN ESP

1 George FRANCE FRA

ID NAME COUNTRY COUNT

---------- ---------- ---------- -----

2 David FRANCE FRA

3 Chirs FRANCE FRA

4 Morrison FRANCE FRA

5 Brian FRANCE FRA

1 George ENGLAND ENG

2 David ENGLAND ENG

3 Chirs ENGLAND ENG

4 Morrison ENGLAND ENG

5 Brian ENGLAND ENG

1 George POLAND POL

2 David POLAND POL

ID NAME COUNTRY COUNT

---------- ---------- ---------- -----

3 Chirs POLAND POL

4 Morrison POLAND POL

5 Brian POLAND POL

25 rows selected.

SQL> drop view clco;

View dropped.

SQL> create view clco as select L.Id, L.Name, C.Country, C.Country\_code from Clients L, Countries1 C where L.Id = C.Id ;

View created.

SQL> select \* from clco;

ID NAME COUNTRY COUNT

---------- ---------- ---------- -----

1 George INDIA IND

2 David SPAIN ESP

3 Chirs FRANCE FRA

4 Morrison ENGLAND ENG

5 Brian POLAND POL

SQL> delete cl where Id=4;

1 row deleted.

SQL> select \* from cl;

ID NAME

---------- ----------

1 George

2 David

3 Chirs

5 Brian

SQL> select \* from clients;

ID NAME EMAIL\_ID

---------- ---------- ----------

1 George ge.com

2 David da.com

3 Chirs ch.com

5 Brian br.com

SQL> insert into cl values(4,'Morrison');

1 row created.

SQL> select \* from cl;

ID NAME

---------- ----------

1 George

2 David

3 Chirs

4 Morrison

5 Brian

SQL> select \* from clients;

ID NAME EMAIL\_ID

---------- ---------- ----------

1 George ge.com

2 David da.com

3 Chirs ch.com

4 Morrison

5 Brian br.com

SQL> update cl set name = 'Morrisons' where Id = 4;

1 row updated.

SQL> select \* from cl;

ID NAME

---------- ----------

1 George

2 David

3 Chirs

4 Morrisons

5 Brian

SQL> select \* from clients;

ID NAME EMAIL\_ID

---------- ---------- ----------

1 George ge.com

2 David da.com

3 Chirs ch.com

4 Morrisons

5 Brian br.com

SQL> spool off;