



**CHRIST**  
(DEEMED TO BE UNIVERSITY)  
B A N G A L O R E • I N D I A

# **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



**CHRIST**  
(DEEMED TO BE UNIVERSITY)  
B A N G A L O R E • I N D I A

## *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****Write SQL Queries for**

1. Creating tables
2. Display structure of table
3. Insert values into table
4. Display the contents of the table
5. Display the columns from table
6. Demonstrate the WHERE clause using table
7. Demonstrate the DELETE operation on table
8. Demonstrate the DROP operation on table

**Queries:**

1. `create table Marks(Name varchar(20), marks1 number(10), marks2 number(10), marks3 number(10))`
2. `desc Marks`
3. `insert into Marks(Name, Marks1, Marks2, Marks3) values('Avinash', 50, 50, 50)`
4. `select * from Marks`
5. `select Name from Marks;`
6. `select * from Marks where Marks2>60;`
7. `delete Marks where Name='Christina'`
8. `drop table Marks`

**SPOOL FILE:**

```
SQL> spool
currently spooling to C:\Ashwin A 2162015\spool_21_7.txt
SQL> create table Marks(Name varchar(20), marks1 number(10), marks2
number(10), marks3 number(10));
```

Table created.

```
SQL> desc Marks;
```

Name	Null?	Type
-----	-----	-----

NAME	VARCHAR2 (20)
MARKS1	NUMBER (10)
MARKS2	NUMBER (10)
MARKS3	NUMBER (10)

```
SQL> select * from Marks;
```

no rows selected

```
SQL> desc Marks;
```

Name	Null?	Type
-----	-----	-----
NAME		VARCHAR2 (20)
MARKS1		NUMBER (10)
MARKS2		NUMBER (10)
MARKS3		NUMBER (10)

```
SQL> insert into Marks(Name, Marks1, Marks2, Marks3)
values('Avinash', 50, 50, 50);
```

1 row created.

```
SQL> select * from Marks;
```

NAME	MARKS1	MARKS2	MARKS3
-----	-----	-----	-----
Avinash	50	50	50

```
SQL> insert into Marks values('Arun', 80, 80, 80);
```

1 row created.

```
SQL> select * from Marks;
```

NAME	MARKS1	MARKS2	MARKS3
-----	-----	-----	-----
Avinash	50	50	50
Arun	80	80	80

```
SQL> insert into Marks(Name, Marks1, Marks3) values('Christina' ,90,
90);
```

1 row created.

```
SQL> select * from Marks;
```

NAME	MARKS1	MARKS2	MARKS3
Avinash	50	50	50
Arun	80	80	80
Christina	90		90

```
SQL> insert into Marks values('vikram', 100, 100, 100);
```

```
1 row created.
```

```
SQL> select * from Marks;
```

NAME	MARKS1	MARKS2	MARKS3
Avinash	50	50	50
Arun	80	80	80
Christina	90		90
vikram	100	100	100

```
SQL> select * from Marks where Marks2>60;
```

NAME	MARKS1	MARKS2	MARKS3
Arun	80	80	80
vikram	100	100	100

```
SQL> select * from Marks where Marks2>60 and Marks3<90;
```

NAME	MARKS1	MARKS2	MARKS3
Arun	80	80	80

```
SQL> insert into Marks values('ananth', 80, 85, 90);
```

```
1 row created.
```

```
SQL> select * from Marks where Marks2>60 and Marks3<90;
```

NAME	MARKS1	MARKS2	MARKS3
Arun	80	80	80

```
SQL> select * from Marks where Marks2>60 and Marks3<=90;
```

NAME	MARKS1	MARKS2	MARKS3
Arun	80	80	80
ananth	80	85	90

```
SQL> select * from Marks;
```

NAME	MARKS1	MARKS2	MARKS3
Avinash	50	50	50
Arun	80	80	80
Christina	90		90
vikram	100	100	100
ananth	80	85	90

```
SQL> select Name from Marks where Marks2 > 60;
```

NAME
Arun
vikram
ananth

```
SQL> delete Marks where Name='Christina';
```

```
1 row deleted.
```

```
SQL> select * from Marks;
```

NAME	MARKS1	MARKS2	MARKS3
Avinash	50	50	50
Arun	80	80	80
vikram	100	100	100
ananth	80	85	90

```
SQL> insert into Marks values('Christina', 80, 80, 80);
```

```
1 row created.
```

```
SQL> select * from Marks;
```

NAME	MARKS1	MARKS2	MARKS3
Avinash	50	50	50
Arun	80	80	80
vikram	100	100	100
ananth	80	85	90
Christina	80	80	80

```
SQL> drop table Marks;
```

```
Table dropped
```



Ex.No : 2

Date:28-07-22

**EMPLOYEE DATABASE**

1. Case Manipulation LOWER(), UPPER()
2. ORDER BY (ASC, DESC)
3. LIKE Command ('\_a%', '%a%', '%a')
4. Aggregate functions (SUM,MAX,MIN,AVG,COUNT)
5. Order by, Group By
6. Having clause
7. IN and NOT IN Operators
8. IS NULL and IS NOT NULL operator
9. AS command

**QUERIES:**

1. `select first_name, employee_id, salary from employees;`
2. `select first_name, employee_id, salary from employees where salary>10000;`
3. `select first_name, employee_id, salary from employees where first_name='Michael';`
4. `select first_name, lower(first_name), upper(first_name), initcap(first_name) from employees;`
5. `select first_name, salary from employees order by salary asc;`
6. `select first_name, salary from employees order by salary desc;`
7. `select first_name, salary from employees order by first_name;`
8. `select first_name, salary from employees order by first_name;`
9. `select first_name from employees where first_name like 'A%';`
10. `select first_name from employees where first_name like 'A%a';`
11. `select sum(salary), min(salary), max(salary), count(salary), avg(salary) from employees;`
12. `select department_id, sum(salary) from employees group by department_id;`
13. `select department_id,job_id, sum(salary) from employees group by department_id,job_id;`
14. `select department_id, sum(salary) from employees group by department_id having sum(salary)>50000;`
15. `select department_id, sum(salary) from employees group by department_id having sum(salary)>50000 order by sum(salary) desc;`
16. `select employee_id, first_name, hire_date, sysdate-hire_date as exp`

```

from employees;
17. select first_name, department_id from employees where department_id
    in(10,20,30)
18. select first_name, department_id from employees where department_id
    not in(10,20,30)
19. select employee_id, commission_pct from employees where
    commission_pct is null;
20. select employee_id, commission_pct from employees where
    commission_pct is not null
21. select employee_id, first_name, hire_date,
    round(((sysdate-hire_date)/30),0) as exp from employees;

```

## SPOOL FILE

```

SQL> spool
currently spooling to D:/Ora/lab.txt

```

```

SQL> @d:/Ora/hr_main;
***** Creating REGIONS table ....

```

Comment created.

Commit complete.

```

SQL> Select * from Employees;

```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000

```

      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
...
...
...
...

      203 Susan                    Mavris
SMAVRIS                      515.123.7777      07-JUN-94 HR_REP
6500
                                101              40

      204 Hermann                  Baer
HBAER                        515.123.8888      07-JUN-94 PR_REP
10000
                                101              70

      205 Shelley                  Higgins
SHIGGINS                     515.123.8080      07-JUN-94 AC_MGR
12000
                                101              110

      206 William                  Gietz
WGIETZ                       515.123.8181      07-JUN-94 AC_ACCOUNT
8300
                                205              110

```

107 rows selected.

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
.		
.		
.		
.		
Girard	183	2800
Nandita	184	4200
Alexis	185	4100
Julia	186	3400
Anthony	187	3000
Kelly	188	3800
Jennifer	189	3600
Timothy	190	2900
Randall	191	2500
Sarah	192	4000
Britney	193	3900
Samuel	194	3200
Vance	195	2800
Alana	196	3100

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
11000	Raphaely	PU_MAN

3100 Khoo	PU_CLERK
2900 Baida	PU_CLERK
2800 Tobias	PU_CLERK
2600 Himuro	PU_CLERK
2500 Colmenares	PU_CLERK
8000 Weiss	ST_MAN
8200 Fripp	ST_MAN
7900 Kaufling	ST_MAN
6500 Vollman	ST_MAN
5800 Mourgos	ST_MAN
.	
.	
.	
.	
3200 Nayer	ST_CLERK
2700 Mikkilineni	ST_CLERK
2400 Landry	ST_CLERK
2200 Markle	ST_CLERK
3300 Bissot	ST_CLERK
2800 Atkinson	ST_CLERK
2500 Marlow	ST_CLERK
2100 Olson	ST_CLERK
3300 Mallin	ST_CLERK
2900 Rogers	ST_CLERK
2400 Gee	ST_CLERK
2200 Philtanker	ST_CLERK
3600 Ladwig	ST_CLERK
3200 Stiles	ST_CLERK
2700 Seo	ST_CLERK
2500 Patel	ST_CLERK
3500 Rajs	ST_CLERK
3100 Davies	ST_CLERK
2600 Matos	ST_CLERK
2500 Vargas	ST_CLERK
14000 Russell	SA_MAN
13500 Partners	SA_MAN
12000 Errazuriz	SA_MAN
11000 Cambrault	SA_MAN
10500 Zlotkey	SA_MAN
10000 Tucker	SA_REP
9500 Bernstein	SA_REP
9000 Hall	SA_REP
8000 Olsen	SA_REP
10000 Baer	PR_REP
12000 Higgins	AC_MGR

8300 Gietz  
107 rows selected.

AC\_ACCOUNT

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), 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			
John	john	JOHN	John
Sarath	sarath	SARATH	Sarath
.			
.			
.			
.			
Lindsey	lindsey	LINDSEY	
Lindsey			
William	william	WILLIAM	
William			
Stephen	stephen	STEPHEN	
Stephen			
Martha	martha	MARTHA	Martha
Patrick	patrick	PATRICK	
Patrick			
Jonathon	jonathon	JONATHON	
Jonathon			
Winston	winston	WINSTON	
Winston			
Sigal	sigal	SIGAL	Sigal
Peter	peter	PETER	Peter
Oliver	oliver	OLIVER	Oliver
Jose Manuel	jose manuel	JOSE MANUEL	Jose
Manuel			
Peter	peter	PETER	Peter
Clara	clara	CLARA	Clara
Shanta	shanta	SHANTA	Shanta
Alana	alana	ALANA	Alana
Matthew	matthew	MATTHEW	
Matthew			
Jennifer	jennifer	JENNIFER	
Jennifer			
Eleni	eleni	ELENI	Eleni

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 asc;

FIRST_NAME	SALARY
-----	-----
TJ	2100
Steven	2200
Clara	10500
Eleni	10500
Gerald	11000
Den	11000
Ellen	11000
.	
.	
.	
.	
Lisa	11500
Shelley	12000
Nancy	12000
Alberto	12000
Michael	13000
Karen	13500
John	14000
Lex	17000
Neena	17000
Steven	24000

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.
.	
.	
.	

John	2700
Irene	2700
Guy	2600
Douglas	2600
Donald	2600
Randall	2600
Karen	2500
James	2500
Randall	2500
Peter	2500
Martha	2500
Joshua	2500
Ki	2400
James	2400
Hazel	2200
Steven	2200
TJ	2100

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
David	9500
.	
.	
.	
.	
Susan	6500
TJ	2100
Tayler	9600
Timothy	2900
Trenna	3500
Valli	4800
Vance	2800
William	8300
William	7400
Winston	3200

107 rows selected.

```
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';
```

```
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
```

```
Mattea
```

```
Allan
```

```
Julia
```

```
Lisa
```

```
Joshua
```

```
24 rows selected.
```

```
SQL> select first_name from employees where first_name like '_a%';
```

FIRST\_NAME

-----

David

Sarah

David

.

.

.

Valli

Randall

Hazel

Nandita

Sarath

Martha

Patrick

Matthew

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, 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
Sundar	SA_REP
Amit	SA_REP
Lisa	SA_REP
Harrison	SA_REP
Tayler	SA_REP
William	SA_REP
Elizabeth	SA_REP
Sundita	SA_REP
Ellen	SA_REP
Alyssa	SA_REP
Jonathon	SA_REP
Jack	SA_REP
Kimberely	SA_REP
Charles	SA_REP
Pat	MK_REP
Susan	HR_REP
Hermann	PR_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> 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
.			
.			
.			
177	Jack	23-APR-98	8862.48624
178	Kimberely	24-MAY-99	8466.48624
179	Charles	04-JAN-00	8241.48624
180	Winston	24-JAN-98	8951.48624
181	Jean	23-FEB-98	8921.48624
182	Martha	21-JUN-99	8438.48624
183	Girard	03-FEB-00	8211.48624
184	Nandita	27-JAN-96	9679.48624
185	Alexis	20-FEB-97	9289.48624
186	Julia	24-JUN-98	8800.48624
187	Anthony	07-FEB-99	8572.48624
188	Kelly	14-JUN-97	9175.48624
189	Jennifer	13-AUG-97	9115.48624
190	Timothy	11-JUL-98	8783.48624
191	Randall	19-DEC-99	8257.48624
192	Sarah	04-FEB-96	9671.48624
193	Britney	03-MAR-97	9278.48624
194	Samuel	01-JUL-98	8793.48624
195	Vance	17-MAR-99	8534.48624
196	Alana	24-APR-98	8861.48624
197	Kevin	23-MAY-98	8832.48624
198	Donald	21-JUN-99	8438.48624
199	Douglas	13-JAN-00	8232.48624
200	Jennifer	17-SEP-87	12733.4862
201	Michael	17-FEB-96	9658.48624
202	Pat	17-AUG-97	9111.48624
203	Susan	07-JUN-94	10278.4862
204	Hermann	07-JUN-94	10278.4862
205	Shelley	07-JUN-94	10278.4862
206	William	07-JUN-94	10278.4862



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
146	Karen	05-JAN-97	311
147	Alberto	10-MAR-97	309
148	Gerald	15-OCT-99	277
149	Eleni	29-JAN-00	274
150	Peter	30-JAN-97	310
151	David	24-MAR-97	309
152	Peter	20-AUG-97	304
153	Christopher	30-MAR-98	296
.			
.			
.			
.			
.			
197	Kevin	23-MAY-98	294
198	Donald	21-JUN-99	281
199	Douglas	13-JAN-00	274
200	Jennifer	17-SEP-87	424
201	Michael	17-FEB-96	322
202	Pat	17-AUG-97	304
203	Susan	07-JUN-94	343
204	Hermann	07-JUN-94	343
205	Shelley	07-JUN-94	343
206	William	07-JUN-94	343

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
.	
.	
.	
.	
.	
Girard	50
Nandita	50
Alexis	50
Julia	50
Anthony	50
Kelly	50
Jennifer	50
Timothy	50
Randall	50
Sarah	50
Britney	50

Samuel	50
Vance	50
Alana	50
Kevin	50
Donald	50
Douglas	50
Susan	40
Hermann	70
Shelley	110
William	110

97 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	
114	
115	
116	

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

149	.2
150	.3
151	.25
152	.25
153	.2
154	.2
155	.15
158	.35
159	.3
160	.3
161	.25
162	.25
163	.15
164	.1
165	.1
166	.1
167	.1
168	.25
169	.2
170	.2
171	.15
173	.1
174	.3
175	.25
176	.2
177	.2
178	.15
179	.1

31 rows selected.

**Ex.No : 3**

**Date: 11 - 08 - 22**

## **LIBRARY DATABASE**

**Consider the following schema for a Library Database:**

BOOK (Book\_id, Title, Publisher\_Name, Pub\_Year)

BOOK\_AUTHORS (Book\_id, Author\_Name)

PUBLISHER (Name, Address, Phone)

BOOK\_COPIES (Book\_id, Branch\_id, No-of\_Copies)

CARD (Card\_No)

BOOK\_LENDING (Book\_id, Branch\_id, Card\_No, Date\_Out, Due\_Date)

LIBRARY\_BRANCH (Branch\_id, Branch\_Name, Address)

### **Problems Given:**

1. Retrieve details of all books in the library id, book title, name of publisher , author, no of copies in each branch
2. Get the particulars of borrowers who have borrowed more than 3 books between jan 2017 and june 2017
3. Delete a book in the book table . update the contents of other tables reflect to the data manipulation operation
4. Partition the book table based on the year of publication. Demonstrate the working with a simple query
5. create a view of all books and its number of copies that are currently available in the library

### **Queries:**

1. select A.book\_id, A.title, A.pub\_name, B.author\_name, C.no\_of\_copies, D.Branch\_id from Books A, Book\_author 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;
2. select card\_no from Book\_lending where date\_out between '01-Jan-2017' and '30-Jun-2017' group by card\_no having count(\*) > 3;
3. delete from Books where book\_id = 4;
4. create view book\_copies as select A.title, A.Book\_id, B.no\_of\_copies, C.Branch\_id, from Books A, Book\_authors B, Library\_branch C where A.Book\_id = B.Book\_id and B.Branch\_id = C.Branch\_id;
5. create view view\_Publication as select pub\_year, count(pub\_year) from books;

**SPOOL FILE:**

```
SQL> create table publisher(pub_name varchar(20), phone number(10),
address varchar(20));
```

Table created.

```
SQL> create table book_authors(book_id int, author_name varchar(20));
```

Table created.

```
SQL> create table book_copies(book_id int, branch_id int,
no_of_copies int);
```

Table created.

```
SQL> create table book_lending(date_out date, due_date date, book_id
int, branch_id int, card_no int)
2 ;
```

Table created.

```
SQL> create table library_branch(branch_id int, branch_name
varchar(20), address varchar(20));
```

Table created.

```
SQL> create table card(card_no int);
```

Table Created.

```
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, 'HYDERABAD');
```

1 row created.

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

1 row created.

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

1 row created.

```
SQL> INSERT INTO LIBRARY_BRANCH VALUES (10, 'RR NAGAR', 'BANGALORE');
```

1 row created.

```
SQL> INSERT INTO LIBRARY_BRANCH VALUES (11, 'RNSIT', 'BANGALORE');
```

1 row created.

```
SQL> INSERT INTO LIBRARY_BRANCH VALUES (12, 'RAJAJI NAGAR',  
'BANGALORE');
```

1 row created.

```
SQL> INSERT INTO LIBRARY_BRANCH VALUES (13, 'NITTE', 'MANGALORE');
```

1 row created.

```
SQL> INSERT INTO LIBRARY_BRANCH VALUES (14, 'MANIPAL', 'UDUPI');
```

1 row created.

```
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 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 (4, 'CG', 2015, 'GRUPO PLANETA');
```

1 row created.

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

1 row created.



```
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> 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> 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 A.book_id, A.title, A.pub_name, B.author_name,
C.no_of_copies, D.branch_id from books 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	MCGRRAW-HILL	NAVATHE
10	10		
1	DBMS	MCGRRAW-HILL	NAVATHE
5	11		
2	ADBMS	MCGRRAW-HILL	NAVATHE
2	12		
2	ADBMS	MCGRRAW-HILL	NAVATHE
5	13		
3	CN	PEARSON	TANENBAUM
7	14		
5	OS	PEARSON	GALVIN
1	10		

```
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> create view V_Publications as select pub_year from book;
```

View created.

```
SQL> select * from V_publications;
```

```
    PUB_YEAR  
-----
```

```
    2017
```

```
    2016
```

```
    2016
```

```
    2016
```

```
SQL> create view book_copies_view as select A.title, A.book_id,  
B.no_of_copies, C.branch_id from books A, book_copies B,  
library_Branch C where A.book_id = B.book_id and B.branch_id =  
C.branch_id;
```

View created.

Ex.No : 4

Date: 18-08-22

### ORDER DATABASE

**Consider the following schema for Order Database:**

SALESMAN (Salesman\_id, Name, City, Commission)

CUSTOMER (Customer\_id, Cust\_Name, City, Grade, Salesman\_id)

ORDERS (Ord\_No, Purchase\_Amt, Ord\_Date, Customer\_id, Salesman\_id)

#### Problems Given:

1. Count the customers with grade above bangalore's average
2. Find the name and number of salesman who had more than one customer
3. List all salesmen and indicate those who have and do not have customers in their cities. (Use 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.  
All his

#### Queries:

1. `select count(*) from customers where grade > (select avg(grade) from customers where city='Bangalore')`
2. `select salesman_id, name from salesman where (select count(*) from customer) > 1`
3. `SQL> select S.salesman_id, S.name, S.commission, C.customer_name from Salesman S, Customers C where S.city=C.City UNION (select salesman_id, name, 'NO MATCH', Commission from salesman where not city = any(select city from`

- ```

customers)) order by 2 desc;
4. create view best_salesman as select O.order_date,
   S.Salesman_id, S.name from Salesman S, Orders O where
   S.salesman_id = O.salesman_id and O.purchase_amt = (select
   max(O.purchase_amt) from Orders I where
   O.order_date=I.order_date);
5. delete from salesman where salesman_id = 1000;

```

**SPOOL FILE:**

```

SQL> create table salesman(
  2  salesman_id int primary key,
  3  Name varchar(10),
  4  City varchar(10),
  5  Commission int);

```

Table created.

```

SQL> create table customers(
  2  customer_id int primary key,
  3  customer_name  varchar(10),
  4  city varchar(10),
  5  grade int,
  6  salesmad_id references salesman(salesman_id) on delete cascade);

```

Table created.

```

SQL> create table orders(
  2  order_no int primary key,
  3  purchase_amt int,
  4  order_date date,
  5  customer_id references customers(customer_id) on delete cascade,
  6  salesman_id references salesman(salesman_id) on delete cascade);

```

Table created.

```

SQL> commit
  2  ;

```

Commit complete.

```

SQL> insert into salesman values(&a, '&b', '&c', &d);
Enter value for a: 1000

```

```

Enter value for b: John
Enter value for c: Bangalore
Enter value for d:
old 1: insert into salesman values(&a, '&b', '&c', &d)
new 1: insert into salesman values(1000, 'John', 'Bangalore', )
insert into salesman values(1000, 'John', 'Bangalore', )
*
```

ERROR at line 1:  
ORA-00936: missing expression

```

SQL> insert into salesman values(&a, '&b', '&c', &d);
Enter value for a: 1000
Enter value for b: John
Enter value for c: Bangalore
Enter value for d: 25
old 1: insert into salesman values(&a, '&b', '&c', &d)
new 1: insert into salesman values(1000, 'John', 'Bangalore', 25)

1 row created.
```

```

SQL> /
Enter value for a: 2000
Enter value for b: Ravi
Enter value for c: Bangalore
Enter value for d: 20
old 1: insert into salesman values(&a, '&b', '&c', &d)
new 1: insert into salesman values(2000, 'Ravi', 'Bangalore', 20)

1 row created.
```

```

SQL> /
Enter value for a: 3000
Enter value for b: Kumar
Enter value for c: Mysore
Enter value for d: 15
old 1: insert into salesman values(&a, '&b', '&c', &d)
new 1: insert into salesman values(3000, 'Kumar', 'Mysore', 15)

1 row created.
```

```

SQL> /
Enter value for a: 4000
Enter value for b: Smith
Enter value for c: Delhi
```

```

Enter value for d: 30
old 1: insert into salesman values(&a, '&b', '&c', &d)
new 1: insert into salesman values(4000, 'Smith', 'Delhi', 30)

```

```
1 row created.
```

```

SQL> /
Enter value for a: 5000
Enter value for b: Harsha Hyderabad
Enter value for c: Hyderabad
Enter value for d: 15
old 1: insert into salesman values(&a, '&b', '&c', &d)
new 1: insert into salesman values(5000, 'Harsha Hyderabad',
'Hyderabad', 15)
insert into salesman values(5000, 'Harsha Hyderabad', 'Hyderabad',
15)

```

\*

```

ERROR at line 1:
ORA-12899: value too large for column "SYSTEM"."SALESMAN"."NAME"
(actual: 16,
maximum: 10)

```

```

SQL> /
Enter value for a: 5000
Enter value for b: Harsha
Enter value for c: Hyderabad
Enter value for d: 15
old 1: insert into salesman values(&a, '&b', '&c', &d)
new 1: insert into salesman values(5000, 'Harsha', 'Hyderabad', 15)

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 | Hyderabad | 15         |

```

SQL> insert into customers values(&a, '&b', '&c', &d, &e);
Enter value for a: 10

```

```
Enter value for b: Preethi
Enter value for c: Bangalore
Enter value for d: 100
Enter value for e: 1000
old 1: insert into customers values(&a, '&b', '&c', &d, &e)
new 1: insert into customers values(10, 'Preethi', 'Bangalore',
100, 1000)
```

1 row created.

```
SQL> /
Enter value for a: 11
Enter value for b: Vivek
Enter value for c: Mangalore
Enter value for d: 300
Enter value for e: 1000
old 1: insert into customers values(&a, '&b', '&c', &d, &e)
new 1: insert into customers values(11, 'Vivek', 'Mangalore', 300,
1000)
```

1 row created.

```
SQL> /
Enter value for a: 12
Enter value for b: Bhaskar
Enter value for c: Chennai
Enter value for d: 400
Enter value for e: 2000
old 1: insert into customers values(&a, '&b', '&c', &d, &e)
new 1: insert into customers values(12, 'Bhaskar', 'Chennai', 400,
2000)
```

1 row created.

```
SQL> /
Enter value for a: 13
Enter value for b: Chethan
Enter value for c: Bangalore
Enter value for d: 200
Enter value for e: 2000
old 1: insert into customers values(&a, '&b', '&c', &d, &e)
new 1: insert into customers values(13, 'Chethan', 'Bangalore',
200, 2000)
```

1 row created.



```
SQL> /
Enter value for a: 14
Enter value for b: Mamatha
Enter value for c: Bangalore
Enter value for d: 400
Enter value for e: 3000
old 1: insert into customers values(&a, '&b', '&c', &d, &e)
new 1: insert into customers values(14, 'Mamatha', 'Bangalore',
400, 3000)
```

1 row created.

```
SQL> select * from customers;
```

| CUSTOMER_ID | CUSTOMER_N | 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> insert into orders values(&a, &b, '&c', &d, &e);
Enter value for a: 50
Enter value for b: 5000
Enter value for c: 04-May-17
Enter value for d: 10
Enter value for e: 1000
old 1: insert into orders values(&a, &b, '&c', &d, &e)
new 1: insert into orders values(50, 5000, '04-May-17', 10, 1000)
```

1 row created.

```
SQL> /
Enter value for a: 51
Enter value for b: 450
Enter value for c: 20-Jan-17
Enter value for d: 10
Enter value for e: 2000
old 1: insert into orders values(&a, &b, '&c', &d, &e)
new 1: insert into orders values(51, 450, '20-Jan-17', 10, 2000)
```

1 row created.

```
SQL> /
Enter value for a: 52
Enter value for b: 1000
Enter value for c: 24-Feb-17
Enter value for d: 13
Enter value for e: 2000
old 1: insert into orders values(&a, &b, '&c', &d, &e)
new 1: insert into orders values(52, 1000, '24-Feb-17', 13, 2000)

1 row created.
```

```
SQL> /
Enter value for a: 53
Enter value for b: 3500
Enter value for c: 13-Apr-17
Enter value for d: 14
Enter value for e: 3000
old 1: insert into orders values(&a, &b, '&c', &d, &e)
new 1: insert into orders values(53, 3500, '13-Apr-17', 14, 3000)

1 row created.
```

```
SQL> /
Enter value for a: 54
Enter value for b: 550
Enter value for c: 09-Mar-17
Enter value for d: 12
Enter value for e: 2000
old 1: insert into orders values(&a, &b, '&c', &d, &e)
new 1: insert into orders values(54, 550, '09-Mar-17', 12, 2000)

1 row created.
```

```
SQL> select * from orders;
```

| ORDER_NO | PURCHASE_AMT | ORDER_DAT | 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 * from customers where grade > (select avg(grade) from
```

```
customers where city='Bangalore');
```

| CUSTOMER_ID | CUSTOMER_N | CITY      | GRADE | SALESMAN_ID |
|-------------|------------|-----------|-------|-------------|
| 11          | Vivek      | Mangalore | 300   | 1000        |
| 12          | Bhaskar    | Chennai   | 400   | 2000        |
| 14          | Mamatha    | Bangalore | 400   | 3000        |

```
SQL> select count(*) from customers where grade > (select avg(grade)
from customers where city='Bangalore');
```

| COUNT(*) |
|----------|
| 3        |

```
SQL> select salesman_id, name from salesman where (select count(*)
from customer) > 1;
```

| SALESMAN_ID | NAME   |
|-------------|--------|
| 1000        | John   |
| 2000        | Ravi   |
| 3000        | Kumar  |
| 4000        | Smith  |
| 5000        | Harsha |

```
SQL> select S.salesman_id, S.name, S.commission, C.customer_name from
Salesman S, Customers C where S.city=C.City UNION (select
salesman_id, name, 'NO MATCH', Commission from salesman where not
city = any(select city from customers)) order by 2 desc;
```

| SALESMAN_ID | NAME  | CUST_NAME | COMMISSION |
|-------------|-------|-----------|------------|
| 4000        | SMITH | NO MATCH  | 30%        |
| 2000        | RAVI  | CHEETHAN  | 20%        |
| 2000        | RAVI  | MAMATHA   | 20%        |

```
3 rows selected
```

```
SQL> create view best_salesman as select O.order_date, S.Salesman_id,
S.name from Salesman S, Orders O where S.salesman_id = O.salesman_id
and O.purchase_amt = (select max(O.purchase_amt) from Orders I where
O.order_date=I.order_date);
```

View created.

```
SQL> delete from Salesman where salesman_id=1000;  
deleted 1 row.
```

**Ex.No : 5**

**Date: 25-08-22**

## **JOINS**

**Consider the following schema of a database:**

books (id, title, type, author\_id, editor\_id, translator\_id)  
authors (id, first\_name, last\_name)  
editors (id, first\_name, last\_name)  
translators (id, first\_name, last\_name)

**Write Queries for:**

1. Show book titles along with their authors (i.e., the author's first name and last name).
2. Display books along with their translators (i.e., the translator's last name). Only half of our books have been translated and thus have a corresponding translator.
3. Display information about each book's author and translator (i.e., their last names). We also want to keep the basic information about each book (i.e., id, title, and type).
4. 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 the books in the result set.
5. Let us repeat our previous example, but this time, our task will be to keep all the records from the editor's table
6. Let us again join the books and editor's tables, but this time, we will be keeping all records from both tables

7. Join all four tables to get information about all the books, authors, editors, and translators in one table

### Queries:

1. `select Books1.title, Authors1.last_name from Books1 INNER JOIN Authors1 ON Books1.Author_id = Authors1.id`
2. `select Books1.id, Books1.title, Translators1.last_name from Books1 INNER JOIN Translators1 ON Books1.translator_id = Translators1.id`
3. `select Books1.id, Books1.title, Translators1.last_name from Books1 INNER JOIN Translators1 ON Books1.translator_id = Translators1.id order by Books1.id`
4. `select Books1.id, Books1.title, Authors1.last_name, Translators1.last_name from Books1 LEFT JOIN Authors1 on Books1.author_id=Authors1.id LEFT JOIN Translators1 ON Books1.translator_id = Translators1.id order by Books1.id`
5. `select Books1.id, Books1.title, Editors1.last_name from Books1 RIGHT JOIN Editors1 ON Books1.editor_id = Editors1.id order by Books1.id`
6. `select * from Books1 FULL JOIN Editors1 ON Books1.editor_id = Editors1.id order by Books1.id`
7. `select * from Books1 FULL JOIN Authors1 ON Books1.author_id = Authors1.id FULL JOIN Editors1 ON Books1.editor_id=Editors1.id FULL JOIN Translators1 ON Books1.translator_id=Translators1.id`

### SPOOL FILE:

```
SQL> create table Books1(id int, title varchar(20), type varchar(12),
author_id int, editor_id int, translator_id int);
```

Table created.

```
SQL> create table Authors1(id int, first_name varchar(10), last_name
varchar(10));
```

Table created.

```
SQL> create table Editors1(id int, first_name varchar(12), last_name
varchar(10));
```

Table created.

```
SQL> create table Translators1(id int, first_name varchar(12),
```

```
last_name varchar(10));
```

Table created.

```
SQL> insert into Books1 values(1, 'time to grow up', 'original', 11,
21, NULL);
```

1 row created.

```
SQL> insert into Books1 values(&id, '&title', '&type', &ai, &ei,
&ti);
```

Enter value for id: 2

Enter value for title: your trip

Enter value for type: translated

Enter value for ai: 15

Enter value for ei: 22

Enter value for ti: 32

```
old 1: insert into Books1 values(&id, '&title', '&type', &ai, &ei,
&ti)
```

```
new 1: insert into Books1 values(2, 'your trip', 'translated', 15,
22, 32)
```

1 row created.

```
SQL> /
```

Enter value for id: 3

Enter value for title: lovely love

Enter value for type: original

Enter value for ai: 14

Enter value for ei: 24

Enter value for ti: NULL

```
old 1: insert into Books1 values(&id, '&title', '&type', &ai, &ei,
&ti)
```

```
new 1: insert into Books1 values(3, 'lovely love', 'original', 14,
24, NULL)
```

1 row created.

```
SQL> /
```

Enter value for id: 4

Enter value for title: dream your life

Enter value for type: original

Enter value for ai: 11

Enter value for ei: 24

Enter value for ti: NULL

```
old 1: insert into Books1 values(&id, '&title', '&type', &ai, &ei,
&ti)
new 1: insert into Books1 values(4, 'dream your life', 'original',
11, 24, NULL)
```

1 row created.

```
SQL> /
Enter value for id: 5
Enter value for title: oranges
Enter value for type: translated
Enter value for ai: 12
Enter value for ei: 25
Enter value for ti: 31
old 1: insert into Books1 values(&id, '&title', '&type', &ai, &ei,
&ti)
new 1: insert into Books1 values(5, 'oranges', 'translated', 12,
25, 31)
```

1 row created.

```
SQL> /
Enter value for id: 6
Enter value for title: your happy life
Enter value for type: translated
Enter value for ai: 15
Enter value for ei: 22
Enter value for ti: 33
old 1: insert into Books1 values(&id, '&title', '&type', &ai, &ei,
&ti)
new 1: insert into Books1 values(6, 'your happy life',
'translated', 15, 22, 33)
```

1 row created.

```
SQL> /
Enter value for id: 7
Enter value for title: applied AI
Enter value for type: translated
Enter value for ai: 13
Enter value for ei: 23
Enter value for ti: 34
old 1: insert into Books1 values(&id, '&title', '&type', &ai, &ei,
&ti)
new 1: insert into Books1 values(7, 'applied AI', 'translated', 13,
```

23, 34)

1 row created.

```
SQL> /
Enter value for id: 8
Enter value for title: my last book
Enter value for type: original
Enter value for ai: 11
Enter value for ei: 28
Enter value for ti: NULL
old 1: insert into Books1 values(&id, '&title', '&type', &ai, &ei,
&ti)
new 1: insert into Books1 values(8, 'my last book', 'original', 11,
28, NULL)
```

1 row created.

```
SQL> commit;
```

Commit complete.

```
SQL> insert into Authors1 values(&id, '&fn', '&ln');
Enter value for id: 11
Enter value for fn: ellen
Enter value for ln: writer
old 1: insert into Authors1 values(&id, '&fn', '&ln')
new 1: insert into Authors1 values(11, 'ellen', 'writer')
```

1 row created.

```
SQL> /
Enter value for id: 12
Enter value for fn: olga
Enter value for ln: saveleiva
old 1: insert into Authors1 values(&id, '&fn', '&ln')
new 1: insert into Authors1 values(12, 'olga', 'saveleiva')
```

1 row created.

```
SQL> /
Enter value for id: 13
Enter value for fn: jack
Enter value for ln: smart
old 1: insert into Authors1 values(&id, '&fn', '&ln')
```



```
new    1: insert into Authors1 values(13, 'jack', 'smart')
```

```
1 row created.
```

```
SQL> /
```

```
Enter value for id: 14
```

```
Enter value for fn: donald
```

```
Enter value for ln: brian
```

```
old    1: insert into Authors1 values(&id, '&fn', '&ln')
```

```
new    1: insert into Authors1 values(14, 'donald', 'brian')
```

```
1 row created.
```

```
SQL> /
```

```
Enter value for id: 15
```

```
Enter value for fn: yao
```

```
Enter value for ln: dou
```

```
old    1: insert into Authors1 values(&id, '&fn', '&ln')
```

```
new    1: insert into Authors1 values(15, 'yao', 'dou')
```

```
1 row created.
```

```
SQL> insert into Editors1 values(&id, '&fn', '&ln');
```

```
Enter value for id: 21
```

```
Enter value for fn: daniel
```

```
Enter value for ln: brown
```

```
old    1: insert into Editors1 values(&id, '&fn', '&ln')
```

```
new    1: insert into Editors1 values(21, 'daniel', 'brown')
```

```
1 row created.
```

```
SQL> /
```

```
Enter value for id: 22
```

```
Enter value for fn: mark
```

```
Enter value for ln: johnson
```

```
old    1: insert into Editors1 values(&id, '&fn', '&ln')
```

```
new    1: insert into Editors1 values(22, 'mark', 'johnson')
```

```
1 row created.
```

```
SQL> insert into Editors1 values(&id, '&fn', '&ln');
```

```
Enter value for id: 23
```

```
Enter value for fn: maria
```

```
Enter value for ln: evans
```

```
old 1: insert into Editors1 values(&id, '&fn', '&ln')
new 1: insert into Editors1 values(23, 'maria', 'evans')
```

1 row created.

```
SQL> /
Enter value for id: 24
Enter value for fn: catherine
Enter value for ln: roberts
old 1: insert into Editors1 values(&id, '&fn', '&ln')
new 1: insert into Editors1 values(24, 'catherine', 'roberts')
```

1 row created.

```
SQL> /
Enter value for id: 25
Enter value for fn: sebastian
Enter value for ln: wright
old 1: insert into Editors1 values(&id, '&fn', '&ln')
new 1: insert into Editors1 values(25, 'sebastian', 'wright')
```

1 row created.

```
SQL> /
Enter value for id: 26
Enter value for fn: barbara
Enter value for ln: jones
old 1: insert into Editors1 values(&id, '&fn', '&ln')
new 1: insert into Editors1 values(26, 'barbara', 'jones')
```

1 row created.

```
SQL> /
Enter value for id: 27
Enter value for fn: mathew
Enter value for ln: smith
old 1: insert into Editors1 values(&id, '&fn', '&ln')
new 1: insert into Editors1 values(27, 'mathew', 'smith')
```

1 row created.

```
SQL> insert into Translators1 values(&id, '&fn', '&ln');
Enter value for id: 31
Enter value for fn: ira
Enter value for ln: davies
```

```
old 1: insert into Translators1 values(&id, '&fn', '&ln')
new 1: insert into Translators1 values(31, 'ira', 'davies')
```

1 row created.

SQL> /

Enter value for id: 32

Enter value for fn: ling

Enter value for ln: weng

```
old 1: insert into Translators1 values(&id, '&fn', '&ln')
```

```
new 1: insert into Translators1 values(32, 'ling', 'weng')
```

1 row created.

SQL> /

Enter value for id: 33

Enter value for fn: kristian

Enter value for ln: green

```
old 1: insert into Translators1 values(&id, '&fn', '&ln')
```

```
new 1: insert into Translators1 values(33, 'kristian', 'green')
```

1 row created.

SQL> /

Enter value for id: 34

Enter value for fn: roman

Enter value for ln: edwards

```
old 1: insert into Translators1 values(&id, '&fn', '&ln')
```

```
new 1: insert into Translators1 values(34, 'roman', 'edwards')
```

1 row created.

```
SQL> select Books1.title, Authors1.last_name from Books1 INNER JOIN
Authors1 ON Books1.Author_id = Authors1.id;
```

| TITLE           | LAST_NAME |
|-----------------|-----------|
| -----           | -----     |
| my last book    | writer    |
| dream your life | writer    |
| time to grow up | writer    |
| oranges         | saveleiva |
| applied AI      | smart     |
| lovely love     | brian     |
| your happy life | dou       |
| your trip       | dou       |

8 rows selected.

```
SQL> select Books1.id, Books1.title, Translators1.last_name from
Books1 INNER JOIN Translators1 ON Books1.translator_id =
Translators1.id;
```

| ID | TITLE           | LAST_NAME |
|----|-----------------|-----------|
| 5  | oranges         | davies    |
| 2  | your trip       | weng      |
| 6  | your happy life | green     |
| 7  | applied AI      | edwards   |

8 rows selected.

```
SQL> select Books1.id, Books1.title, Translators1.last_name from
Books1 INNER JOIN Translators1 ON Books1.translator_id =
Translators1.id order by Books1.id;
```

| ID | TITLE           | LAST_NAME |
|----|-----------------|-----------|
| 2  | your trip       | weng      |
| 5  | oranges         | davies    |
| 6  | your happy life | green     |
| 7  | applied AI      | edwards   |

8 rows selected.

```
SQL> select Books1.id, Books1.title, Authors1.last_name,
Translators1.last_name from Books1 LEFT JOIN Authors1 on
Books1.author_id=Authors1.id LEFT JOIN Translators1 ON
Books1.translator_id = Translators1.id;
```

| ID | TITLE           | LAST_NAME | LAST_NAME |
|----|-----------------|-----------|-----------|
| 1  | time to grow up | writer    |           |
| 4  | dream your life | writer    |           |
| 8  | my last book    | writer    |           |
| 5  | oranges         | saveleiva | davies    |
| 7  | applied AI      | smart     | edwards   |
| 3  | lovely love     | brian     |           |
| 6  | your happy life | dou       | green     |
| 2  | your trip       | dou       | weng      |

8 rows selected.

```
SQL> select Books1.id, Books1.title, Authors1.last_name,
Translators1.last_name from Books1 LEFT JOIN Authors1 on
Books1.author_id=Authors1.id LEFT JOIN Translators1 ON
Books1.translator_id = Translators1.id order by Books1.id;
```

| ID | TITLE           | LAST_NAME | LAST_NAME |
|----|-----------------|-----------|-----------|
| 1  | time to grow up | writer    |           |
| 2  | your trip       | dou       | weng      |
| 3  | lovely love     | brian     |           |
| 4  | dream your life | writer    |           |
| 5  | oranges         | saveleiva | davies    |
| 6  | your happy life | dou       | green     |
| 7  | applied AI      | smart     | edwards   |
| 8  | my last book    | writer    |           |

8 rows selected.

```
SQL> select Books1.id, Books1.title, Editors1.last_name from Books1
LEFT JOIN Editors1 ON Books1.editor_id = Editors1.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    |           |

8 rows selected.

```
SQL> select Books1.id, Books1.title, Editors1.last_name from Books1
LEFT JOIN Editors1 ON Books1.editor_id = Editors1.id order by
Books1.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    |         |

8 rows selected.

```
SQL> select Books1.id, Books1.title, Editors1.last_name from Books1
RIGHT JOIN Editors1 ON Books1.editor_id = Editors1.id order by
Books1.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     |
|    |                 | jones     |
|    |                 | smith     |

9 rows selected.

```
SQL> set linesize 200
```

```
SQL> select * from Books1 FULL JOIN Editors1 ON Books1.editor_id =
Editors1.id order by Books1.id;
```

| ID            | TITLE             | TYPE       | AUTHOR_ID | EDITOR_ID |
|---------------|-------------------|------------|-----------|-----------|
| TRANSLATOR_ID | ID                | FIRST_NAME | LAST_NAME |           |
| 1             | time to grow up   | original   | 11        | 21        |
| 21            | daniel brown      |            |           |           |
| 2             | your trip         | translated | 15        | 22        |
| 32            | 22 mark           | johnson    |           |           |
| 3             | lovely love       | original   | 14        | 24        |
| 24            | catherine roberts |            |           |           |
| 4             | dream your life   | original   | 11        | 24        |
| 24            | catherine roberts |            |           |           |
| 5             | oranges           | translated | 12        | 25        |

```

31          25 sebastian    wright
          6 your happy life      translated      15          22
33          22 mark        johnson
          7 applied AI          translated      13          23
34          23 maria        evans
          8 my last book      original          11          28

```

```
27 mathew      smith
```

```
26 barbara      jones
```

```
10 rows selected.
```

```
SQL> select * from Books1 FULL JOIN Authors1 ON Books1.author_id =
Authors1.id FULL JOIN Editors1 ON Books1.editor_id=Editors1.id FULL
JOIN Translators1 ON Books1.translator_id=Translators1.id;
```

| ID            | TITLE           | TYPE       | AUTHOR_ID | EDITOR_ID |            |           |
|---------------|-----------------|------------|-----------|-----------|------------|-----------|
| TRANSLATOR_ID | ID              | FIRST_NAME | LAST_NAME | ID        | FIRST_NAME | LAST_NAME |
| 5             | oranges         | translated | 12        | 25        |            |           |
| 31            | 12 olga         | saveleiva  | 25        | sebastian | wright     |           |
| 31            | ira             | davies     |           |           |            |           |
| 2             | your trip       | translated | 15        | 22        |            |           |
| 32            | 15 yao          | dou        | 22        | mark      | johnson    |           |
| 32            | ling            | weng       |           |           |            |           |
| 6             | your happy life | translated | 15        | 22        |            |           |
| 33            | 15 yao          | dou        | 22        | mark      | johnson    |           |
| 33            | kristian        | green      |           |           |            |           |
| 7             | applied AI      | translated | 13        | 23        |            |           |
| 34            | 13 jack         | smart      | 23        | maria     | evans      |           |
| 34            | roman           | edwards    |           |           |            |           |
| 27            | mathew          | smith      |           |           |            |           |
| 26            | barbara         | jones      |           |           |            |           |
| 8             | my last book    | original   | 11        | 28        |            |           |
| 11            | ellen           | writer     |           |           |            |           |
| 4             | dream your life | original   | 11        | 24        |            |           |
| 11            | ellen           | writer     | 24        | catherine | roberts    |           |
| 3             | lovely love     | original   | 14        | 24        |            |           |
| 14            | donald          | brian      | 24        | catherine | roberts    |           |

```

          1 time to grow up          original          11          21
11 ellen          writer          21 daniel          brown

```

10 rows selected.

SQL> set linesize 1520

SQL> select \* from Books1 FULL JOIN Authors1 ON Books1.author\_id =  
Authors1.id FULL JOIN Editors1 ON Books1.editor\_id=Editors1.id FULL  
JOIN Translators1 ON Books1.translator\_id=Translators1.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
-----
-----
-----
          5 oranges          translated          12          25
31          12 olga          saveleiva          25 sebastian          wright
31 ira          davies
          2 your trip          translated          15          22
32          15 yao          dou          22 mark          johnson
32 ling          weng
          6 your happy life          translated          15          22
33          15 yao          dou          22 mark          johnson
33 kristian          green
          7 applied AI          translated          13          23
34          13 jack          smart          23 maria          evans
34 roman          edwards

27 mathew          smith
26 barbara          jones
          8 my last book          original          11          28
11 ellen          writer
          4 dream your life          original          11          24
11 ellen          writer          24 catherine          roberts
          3 lovely love          original          14          24
14 donald          brian          24 catherine          roberts
          1 time to grow up          original          11          21
11 ellen          writer          21 daniel          brown

```

10 rows selected.



Ex.No : 6

Date: 01-09-22

## VIEWS

**Consider the following schema of a database:**

Client (ID, Name, E\_ID)

Country (ID, Country, Country\_Code)

**Write Queries for:**

1. Create a view from a single table. Create a simple view for client table.
2. Create a view from multiple tables. Create a view that will display Client\_ID, Name, Country, and Country\_Code columns data for all five clients.
3. Inserting a new roll in a view.
4. Updating a row in a view.
5. Deleting a row in a view.
6. Drop a view in SQL.

**Queries:**

1. create view client\_view as select \* from client
2. create view client\_country as select C1.id, C1.name, C2.country, C2.country\_code from client C1, country C2 where C1.id = C2.id
3. insert into client\_view values(6, 'kevin', 'ke.com')
4. update client\_view set name='john', email\_id='jo.com' where id=6
5. delete from client\_view where id=6
6. drop view client\_country

**SPOOL FILE**

```
SQL> create table client(id int, Name varchar(10), email_id
varchar(7));
```

Table created.

```
SQL> insert into client values(&id, '&name', '&email');
Enter value for id: 1
Enter value for name: george
Enter value for email: ge.com
old 1: insert into client values(&id, '&name', '&email')
new 1: insert into client values(1, 'george', 'ge.com')
```

1 row created.

```
SQL> /
Enter value for id: 2
Enter value for name: david
Enter value for email: da.com
old 1: insert into client values(&id, '&name', '&email')
new 1: insert into client values(2, 'david', 'da.com')
```

1 row created.

```
SQL> /
Enter value for id: 3
Enter value for name: chris
Enter value for email: ch.com
old 1: insert into client values(&id, '&name', '&email')
new 1: insert into client values(3, 'chris', 'ch.com')
```

1 row created.

```
SQL> /
Enter value for id: 4
Enter value for name: morrison
Enter value for email: mo.com
old 1: insert into client values(&id, '&name', '&email')
new 1: insert into client values(4, 'morrison', 'mo.com')
```

1 row created.

```
SQL> /
Enter value for id: 5
Enter value for name: brian
Enter value for email: br.com
old 1: insert into client values(&id, '&name', '&email')
new 1: insert into client values(5, 'brian', 'br.com')

1 row created.

SQL> create table country(id int, country varchar(10), country_code
char(3));

Table created.

SQL> insert into country values(&id, '&name', '&code');
Enter value for id: 1
Enter value for name: India
Enter value for code: IND
old 1: insert into country values(&id, '&name', '&code')
new 1: insert into country values(1, 'India', 'IND')

1 row created.

SQL> /
Enter value for id: 2
Enter value for name: Spain
Enter value for code: ESP
old 1: insert into country values(&id, '&name', '&code')
new 1: insert into country values(2, 'Spain', 'ESP')

1 row created.

SQL> /
Enter value for id: 3
Enter value for name: France
Enter value for code: FRA
old 1: insert into country values(&id, '&name', '&code')
new 1: insert into country values(3, 'France', 'FRA')

1 row created.

SQL> /
Enter value for id: 4
Enter value for name: England
Enter value for code: ENG
```

```
old 1: insert into country values(&id, '&name', '&code')
new 1: insert into country values(4, 'England', 'ENG')
```

1 row created.

```
SQL> /
```

Enter value for id: 5

Enter value for name: Poland

Enter value for code: POL

```
old 1: insert into country values(&id, '&name', '&code')
```

```
new 1: insert into country values(5, 'Poland', 'POL')
```

1 row created.

```
SQL> select * from country;
```

| ID | COUNTRY | COU |
|----|---------|-----|
| 1  | India   | IND |
| 2  | Spain   | ESP |
| 3  | France  | FRA |
| 4  | England | ENG |
| 5  | Poland  | POL |

```
SQL> create view client_view as select * from client;
```

View created.

```
SQL> select * from client_view;
```

| ID | NAME     | EMAIL_I |
|----|----------|---------|
| 1  | george   | ge.com  |
| 2  | david    | da.com  |
| 3  | chris    | ch.com  |
| 4  | morrison | mo.com  |
| 5  | brian    | br.com  |

```
SQL> create view client_country as select C1.id, C1.name, C2.country,
C2.country_code from client C1, country C2 where C1.id = C2.id;
```

View created.

```
SQL> select * from client_country;
```

| ID | NAME     | COUNTRY | COU |
|----|----------|---------|-----|
| 1  | george   | India   | IND |
| 2  | david    | Spain   | ESP |
| 3  | chris    | France  | FRA |
| 4  | morrison | England | ENG |
| 5  | brian    | Poland  | POL |

```
SQL> insert into client_view values(6, 'kevin', 'ke.com');
```

1 row created.

```
SQL> select * from client_view;
```

| ID | NAME     | EMAIL_I |
|----|----------|---------|
| 1  | george   | ge.com  |
| 2  | david    | da.com  |
| 3  | chris    | ch.com  |
| 4  | morrison | mo.com  |
| 5  | brian    | br.com  |
| 6  | kevin    | ke.com  |

6 rows selected.

```
SQL> update client_view set name='john', email_id='jo.com' where id=6;
```

1 row updated.

```
SQL> select * from client_view;
```

| ID | NAME     | EMAIL_I |
|----|----------|---------|
| 1  | george   | ge.com  |
| 2  | david    | da.com  |
| 3  | chris    | ch.com  |
| 4  | morrison | mo.com  |
| 5  | brian    | br.com  |
| 6  | john     | jo.com  |

6 rows selected.

```
SQL> delete from client_view where id=6;
```

1 row deleted.

```
SQL> select * from client_view;
```

| ID | NAME     | EMAIL_I |
|----|----------|---------|
| 1  | george   | ge.com  |
| 2  | david    | da.com  |
| 3  | chris    | ch.com  |
| 4  | morrison | mo.com  |
| 5  | brian    | br.com  |

```
SQL> drop view client_country;
```

View dropped.