

**DWARAKA DOSS GOVERDHAN DOSS VAISHNAV
COLLEGE(AUTONOMOUS)
Re-Accredited with 'A++' Grade by NAAC
College With Potential For Excellence
Linguistic Minority Institution, Affiliated To University Of Madras
Arumbakkam, Chennai – 600 106.**



**PG AND RESEARCH DEPARTMENT OF MASTER OF COMPUTER
APPLICATIONS (M.C.A.)**

SEMESTER – I

RDBMS LAB

NAME : _____
REGISTER NO : _____
MONTH : **OCTOBER**
YEAR : **2024**

**DWARAKA DOSS GOVERDHAN DOSS VAISHNAV
COLLEGE(AUTONOMOUS)
Re-Accredited with 'A++' Grade by NAAC
College With Potential For Excellence
Linguistic Minority Institution, Affiliated To University Of Madras
Arumbakkam, Chennai – 600 106.**



**PG AND RESEARCH DEPARTMENT OF MASTER OF COMPUTER
APPLICATIONS (M.C.A.)**

SEMESTER I: RDBMS LAB

BONAFIDE CERTIFICATE

Certified to be the Bonafide Record of work done
by **Mr./Ms.** _____
_____ whose **Register No:** _____ of **I M.C.A.**, during the
academic year **2024– 2025**. Submitted for the Practical Examination held on
_____.

Faculty In-Charge

Head of the Department

Internal Examiner

External Examiner

S.NO	CONTENTS	PG.NO:	STAFF SIGN
1.	DDL COMMANDS	04	
2.	DML COMMANDS	06	
3.	A) AGGREGATE FUNCTIONS	08	
	B) STRING FUNCTIONS		
4.	OPERATORS	16	
5.	JOINS	23	
6.	GROUP BY,ORDER BY,INDEX	29	
7.	SUBQUERY AND VIEWS	33	
8.	CONSTRAINTS	42	
9.	TRANSACTION CONTROL LANGUAGE	46	
10.	FIBONACCI SERIES- PLSQL	48	
11.	AMSTRONG NUMBER-PLSQL	50	
12.	EXCEPTION HANDLING -PLSQL	52	

Ex.No:1

DATA DEFINITION LANGUAGES

Aim:

To implement the data definition languages using SQL query.

Create the following relation

Name of the relation: student

Schema: (id, name, gender, course, class, bloodgroup)

Add a new column; DOB to the existing relation. Change the name of the column/field name to Stud_name Modify the column width of the class.

Drop the relation.

a) Schema : (id , name , gender , course , class , bloodgroup)

```
create table students( id int ,
name varchar (10),
gender varchar (10),
course varchar (10), class
varchar (10), bloodgroup
varchar (10));
```

```
SQL> describe students;
Name                               Null?   Type
-----
ID                                NUMBER(38)
STUD_NAME                        VARCHAR2(10)
GENDER                          VARCHAR2(10)
COURSE                          VARCHAR2(10)
BLOODGROUP                      VARCHAR2(10)
```

b) Alter table add anew column ,D.O.B to the existing relation

Alter table students add (dob date);

```
Name                               Null?   Type
-----
ID                                NUMBER(38)
NAME                            VARCHAR2(10)
GENDER                          VARCHAR2(10)
COURSE                          VARCHAR2(10)
BLOODGROUP                      VARCHAR2(10)
CLASS                          VARCHAR2(10)
DOB                             DATE
```

c)Change the name of the field name to stud name. Alter
table students rename column name to stud_name;

```
SQL> describe students;
Name                               Null?    Type
-----
ID                                 NUMBER(38)
STUD_NAME                          VARCHAR2(10)
GENDER                            VARCHAR2(10)
COURSE                            VARCHAR2(10)
BLOODGROUP                        VARCHAR2(10)
CLASS                             VARCHAR2(10)
DOB                                DATE
```

d)Modify the column width of the class

Alter table students modify (class varchar (20));

```
SQL> describe students;
Name                               Null?    Type
-----
ID                                 NUMBER(38)
STUD_NAME                          VARCHAR2(10)
GENDER                            VARCHAR2(10)
COURSE                            VARCHAR2(10)
BLOODGROUP                        VARCHAR2(10)
CLASS                             VARCHAR2(20)
DOB                                DATE
```

e) Drop the relation Drop

table students ;

```
SQL> drop table students;

Table dropped.

SQL> describe students;
ERROR:
ORA-04043: object students does not exist
```

Result:

Data Definition Language commands are implemented using SQL query.

Ex.No:2

DATA MANIPULATION LANGUAGES

Aim:

To Create the table and implement the DML commands in SQL

INSERT
UPDATE
IMPLEMENT

Create a table employee with following

schema(Emp_no,e_name,e_address,e_phno,dept_no,dept_name,designation,salary)

1.Insert at least 10 rows in the table.

Create table employee(Emp_no varchar(10),e_name varchar(10),e_address
varchar(10),e_phno varchar(10), dept_no varchar(10),dept_name
varchar(10),designation varchar(10),salary varchar(10));

```
SQL> Create table employee(Emp_no varchar(10),e_name varchar(10),e_address varchar(10),e_phno varchar(10),dept_no varchar(10),dept_name varchar(10),designation varchar(10),salary varchar(10));
Table created.
```

2.Display all the information of emp table.

Select * from emp;

```
SQL> select * from emp;
EMP_NO EMP_NAME      E_ADDRESS      E_PHNO  DEPT_NO DEPT_NAME      DESIGNATION      SALARY E_EMAIL
-----
101  aparna      tamilnadu      6382204989    2345  sales      staff      67000  aparsamy14@gmail
103  john      kerla      9342204345    2785  sales      staff      53000  john@gmail
104  kishore      kerla      6342204345    2885  marketing  manager      73000  kishore@gmail.com
105  anbu      delhi      6342204905    2675  marketing  manager      73000  anbu@gmail.com
106  aarav      delhi      9432204905    3475  mech      staff      45000  aarav@gmail.com
102  james      tamilnadu      6382204345    4567  sales      staff      63000  james@gmail.com
108  yamini      delhi      9432987905    3475  mech      staff      48000  yamini@gmail.com
107  venkat      kadpadi      6566607905    9475  IT      staff      48000  venkat@gmail.com
109  guna      chennai      9750643940    5625  mech      designer      90000  guna@gmail.com
110  bala      chennai      9626541300    1324  IT      staff      59000  bala@gmail.com
10 rows selected.
```

3.Display the record of each employee who works in department sales along with the and their designation.

select * from emp where dept_name='sales';

```
SQL> select * from emp where dept_name='sales';
```

EMP_NO	EMP_NAME	E_ADDRESS	E_PHNO	DEPT_NO	DEPT_NAME	DESIGNATION	SALARY	E_EMAIL
101	aparna	tamilnadu	6382204989	2345	sales	staff	67000	aparsamy14@gmail
103	john	kerla	9342204345	2785	sales	staff	53000	john@gmail
102	james	tamilnadu	6382204345	4567	sales	staff	63000	james@gmail.com

4.Update the city of emp_no 103 with current city as Nagpur.

update emp set e_address='nagpur' where emp_no='103';

```
SQL> update emp set e_address='nagpur' where emp_no=103;
```

1 row updated.

5.Delete the email_id of employee james. update emp set
e_email='null' where emp_name='james';

```
SQL> select * from emp where salary>25000 and dept_name='mech';
```

EMP_NO	EMP_NAME	E_ADDRESS	E_PHNO	DEPT_NO	DEPT_NAME	DESIGNATION	SALARY	E_EMAIL
106	aarav	delhi	9432204905	3475	mech	staff	45000	aarav@gmail.com
108	yamini	delhi	9432987905	3475	mech	staff	40000	yamini@gmail.com
109	guna	chennai	9750643940	5625	mech	designer	90000	guna@gmail.com

```
SQL> update emp set e_email=null where e_name='james';
```

```
SQL> select * from emp;
```

EMP_NO	EMP_NAME	E_ADDRESS	E_PHNO	DEPT_NO	DEPT_NAME	DESIGNATION	SALARY	E_EMAIL
101	aparna	tamilnadu	6382204989	2345	sales	staff	67000	aparsamy14@gmail
103	john	nagpur	9342204345	2785	sales	staff	53000	john@gmail
104	kishore	kerla	6342204345	2885	marketing	manager	73000	kishore@gmail.com
105	anbu	delhi	6342204905	2675	marketing	manager	73000	anbu@gmail.com
106	aarav	delhi	9432204905	3475	mech	staff	45000	aarav@gmail.com
102	james	tamilnadu	6382204345	4567	sales	staff	63000	
108	yamini	delhi	9432987905	3475	mech	staff	40000	yamini@gmail.com
107	venkat	kadpadi	0566007905	9475	IT	staff	48000	venkat@gmail.com
109	guna	chennai	9750643940	5625	mech	designer	90000	guna@gmail.com
110	bala	chennai	9626541300	1324	IT	staff	59000	bala@gmail.com

10 rows selected.

6.Display the complete record of employee working in mech department and whose salary is greater than 25000.

Select * from emp where salary>25000 and dept_name='mech';

Result:

Data Manipulation Language commands are implemented using SQL query.

Ex.No:3	TYPES OF FUNCTION
----------------	--------------------------

Aim:

Implementation of different types of functions with SQL query.

a) To create a orders table and execute the queries.

ord_n o	Purchase_amo unt	ord_date	cust_i d	salesman _id
70001	150.5	2022-1005	3005	5002
70009	270.65	2022-1010	3001	5005
70002	65.26	2022-1005	3002	5001
70004	110.5	2022-1017	3009	5003
70007	948.5	2022-1005	3005	5002
70005	2400.6	2022-1005	3007	5001
70010	1983.43	2022-1010	3004	5006

1.Find the total purchase amount.

select sum(purchase_amount)from orders;

```
SUM(PURCHASE_AMOUNT)
-----
5929.44
```


2. Find the mean purchase amount.

```
select avg(PURCHASE_AMOUNT) as mean from orders;
```

```
      MEAN
-----
847.062857
```

3. Find the number of unique salespeople.

```
select count(distinct salesman_id) from orders;
```

```
COUNT(DISTINCTSALESMAN_ID)
-----
                             5
```

4. Find the number of customers.

```
select count(cust_id) from orders;
```

```
COUNT(CUST_ID)
-----
                7
```

5. Find the order number and date which has maximum and minimum purchase.

```
select ord_date, ord_no from order_s where purchase_amount=(select max
(purchase_amount) from order_s);
```

```
ORD_DATE      ORD_NO
-----
2022-10-05    70005
```

```
select ord_date, ord_no from order_s where purchase_amount=(select min
(purchase_amount) from order_s);
```

```
ORD_DATE      ORD_NO
-----
2022-10-05    70002
```

6. Find the nearest maximum value of purchase amount of order 70010.

```
select ceil(PURCHASE_AMOUNT) from orders where ord_no=70010;
```

```
CEIL(PURCHASE_AMOUNT)
-----
1984
```

7. Find the nearest minimum value of purchase amount of order 70007.
 select floor(PURCHASE_AMOUNT) from orders where ord_no=70007;

```
FLOOR(PURCHASE_AMOUNT)
-----
948
```

8. Find the sales of salesmen on 2022-10-05.

select* from orders where ord_date='2022-10-05';

ORD_NO	PURCHASE_AMOUNT	ORD_DATE	CUST_ID	SALESMAN_ID
70001	150.5	2022-10-05	3005	5002
70007	948.5	2022-10-05	3005	5002
70005	2400.6	2022-10-05	3007	5001

b) Create a emp1 table with 20 records:

Emp_id	First_Name	Last_Name	Phone Number	Email	DO J	Designation	Salary	Dept_id	Manager id
100	Neena	Kumar	9840026753	neena	198706-17	AD_PRES	24000.00	90	199
102	Siva	Shankar	9840136985	Siva	198707-18	AD_PRES	17000.00	90	190

103	Sudha	Raghu	9840145268	Sudha	198709-19	IT_Prog	15000.00	60	160
104	Sumitha	Manoj	7325843633	Sumitha	198706-20	IT_Prog	13000.00	61	161
105	Dheena	Ashwin	8690145532	Dheena	198707-22	ST_CLERK	2400.00	80	180

1. Write a query to update the portion of the phone_number in the employees table, within the phone number the substring '401' will be replaced by '411'. update emp1 set ph_no = replace(ph_no, '401','411') where ph_no like '%401%';

EMP_ID	FIRST_NAME	LAST_NAME	PH_NO	EMAIL	D_O_J	DESIGNATION	SALARY	DEPT_ID	MANAGER_ID
100	Naven	Kumar	9840026753	neena	17-JUN-87	ad_pres	24000	90	199
102	Siva	Shankar	9841136985	siva	18-JUL-87	ad_pres	17000	90	190
103	Sudha	Raghu	9841145268	sudha	19-SEP-87	it_prog	15000	60	160
104	Sumitha	Manoj	7325843633	sumitha	20-JUN-87	it_prog	13000	61	161
105	Dheena	Ashwin	8690145532	Dhee	22-JUL-87	st_clerk	2400	80	180
122	Sivan	ramcharan	9841136935	sivan	18-JUN-87	ad_pres	14000	90	199
122	Geetha	chanthuchari	9841136935	geethu	18-JUN-89	ad_pres	14000	90	199
122	VARSHA	surya	9841136935	varsha	18-JUN-89	ad_pres	14000	90	199
122	JANANI	ravi	9841136935	janani	18-JUN-89	ad_pres	24000	90	197
122	MALINI	meera	9841136935	malini	18-JUN-89	ad_pres	24000	90	197

10 rows selected.

2. Write a query to get the details of the employees where the length of the first name greater than or equal to 5.

Select * from emp where length(first_name)>=5;

EMP_ID	FIRST_NAME	LAST_NAME	PH_NO	EMAIL	D_O_J	DESIGNATION	SALARY	DEPT_ID	MANAGER_ID
100	Naven	Kumar	984026753	neena@gmail.com	17-JUN-87	ad_pres	24000	90	199
103	Sudha	Raghu	9841145268	sudha@gmail.com	19-SEP-87	it_prog	15000	60	160
104	Sumitha	Manoj	7325843633	sumitha@gmail.com	20-JUN-87	it_prog	13000	61	161
105	Dheena	Ashwin	8690145532	Dhee@gmail.com	22-JUL-87	st_clerk	2400	80	180

3. Write a query to append '@gmail.com' to the email field.

UPDATE emp1 SET email = CONCAT(email, '@gmail.com');

EMP_ID	FIRST_NAME	LAST_NAME	PH_NO	EMAIL	D_O_J	DESIGNATION	SALARY	DEPT_ID	MANAGER_ID
100	Naven	Kumar	9840026753	neena@gmail.com	17-JUN-87	ad_pres	24000	90	199
102	Siva	Shankar	9841136985	siva@gmail.com	18-JUL-87	ad_pres	17000	90	190
103	Sudha	Raghu	9841145268	sudha@gmail.com	19-SEP-87	it_prog	15000	60	160
104	Sumitha	Manoj	7325843633	sumitha@gmail.com	20-JUN-87	it_prog	13000	61	161
105	Dheena	Ashwin	8690145532	Dhee@gmail.com	22-JUL-87	st_clerk	2400	80	180
122	Sivan	ramcharan	9841136935	sivan@gmail.com	18-JUN-87	ad_pres	14000	90	199
122	Geetha	chanthuchari	9841136935	geethu@gmail.com	18-JUN-89	ad_pres	14000	90	199
122	VARSHA	surya	9841136935	varsha@gmail.com	18-JUN-89	ad_pres	14000	90	199
122	JANANI	ravi	9841136935	janani@gmail.com	18-JUN-89	ad_pres	24000	90	197
122	MALINI	meera	9841136935	malini@gmail.com	18-JUN-89	ad_pres	24000	90	197

10 rows selected.

4. Write a query to get the employee id, first name and DOJ month. select
emp_id,first_name,extract(month from d_o_j) as month_of_job from emp1;

EMP_ID	FIRST_NAME	MONTH_OF_JOB
100	Naven	6
102	Siva	7
103	Sudha	9
104	Sumitha	6
105	Dheena	7
122	Sivan	6
122	Geetha	6
122	VARSHA	6
122	JANANI	6
122	MALINI	6

10 rows selected.

5. Write a query to get the employee id, email id (discard the last three characters).
select emp_id, reverse(substr(reverse(email),4)) from emp1;

EMP_ID	REVERSE(SUBSTR(RE
100	neena@gmail.
102	siva@gmail.
103	sudha@gmail.
104	sumitha@gmail.
105	Dhee@gmail.
122	sivan@gmail.
122	geethu@gmail.
122	varsha@gmail.
122	janani@gmail.
122	malini@gmail.

10 rows selected.

6. Write a query to find all employees where first names are in upper case.
select*from emp1 where first_name=upper(first_name);

EMP_ID	FIRST_NAME	LAST_NAME	PH_NO	EMAIL	D_O_J	DESIGNATION	SALARY	DEPT_ID	MANAGER_ID
122	VARSHA	surya	9841136935	varsha@gmail.com	18-JUN-89	ad_pres	14000	90	199
122	JANANI	ravi	9841136935	janani@gmail.com	18-JUN-89	ad_pres	24000	90	197
122	MALINI	meera	9841136935	malini@gmail.com	18-JUN-89	ad_pres	24000	90	197

7. Write a query to extract the last 4 character of phone numbers.
Select substr(ph_no,7) from emp1;

```

SUBSTR(PH_NO,7)
-----
6753
6985
5268
3633
5532
6935
6935
6935
6935
6935
10 rows selected.

```

8. Write a query to display the length of first name for employees where the last name contains character 'c' after 2nd position.

```

select length(first_name) as length_of_name from emp1 where last_name
like '%c%';

```

```

LENGTH_OF_NAME
-----
5
6

```

9. Write a query that displays the first name and the length of the first name for all employees whose name starts with the letters 'N', 'G' or 'S'. Give each column an appropriate label. Sort the results by the Employees first names.

```

select first_name, length(first_name) as length from emp1 where first_name like
'A%' or first_name like 'J%' or first_name like 'M%' order by (first_name);

```

```

FIRST_NAME      LENGTH
-----
JANANI          6
MALINI          6

```

10. 10. Write a query to get the years in which more than 5 employees joined.

```

Select extract(year from d_o_j) as year from emp1 group by extract (year from
d_o_j) having count(emp_id)>5;

```

```

YEAR
-----
1987

```

11. Write a query to get the first name and hire date from employees table where hire date between '1987-06-01' and '1987-07-30'.

```

select first_name, d_o_j as hire_date from emp1 where d_o_j between '01-jun1987' and '30-jul-1987';

```

```
SQL> select first_name,d_o_j as hire_date from emp1 where d_o_j between '01-jun-1987'and '30-jul-1987';
```

FIRST_NAME	HIRE_DATE
Naven	17-JUN-87
Siva	18-JUL-87
Sumitha	20-JUN-87
Dheena	22-JUL-87
Sivan	18-JUN-87

12. Write a query to display the current date in the following format

(05/09/2014). select to_char(sysdate,'dd-mm-yyyy') today_date from dual;

TODAY_DATE
27-11-2022

13. . Write a query to display the current date in the following format.Thursday
4th September 2014 00:00:00.

select to_char(sysdate,'day ddth month yyyy hh:mm:ss') from dual;

TO_CHAR(SYSDATE, 'DAYDDMONTHYYYYHHMMIS
sunday 27 november 2022 09 19 17

14. Write a query to display the current date in the following format 12:00 AM
Sep 5, 2014.

select to_char(sysdate,'HH:HH AM MONTH DD,YYYY') AS "DATE" FROM
DUAL ;

DATE
09:09 PM NOVEMBER 27,2022

15. Write a query to get first name of employees who joined in 1987. select

first_name from emp1 where extract (year from d_o_j)=1987;

FIRST_NAME
Naven
Siva
Sudha
Sumitha
Dheena
Sivan

6 rows selected.

16. Write a query to get employee ID, last name, and date of first salary of the employees. `select emp_id,last_name,d_o_j,add_months(d_o_j,+1) from emp1;`

EMP_ID	LAST_NAME	D_O_J	ADD_MONTH
100	Kumar	17-JUN-87	17-JUL-87
102	Shankar	18-JUL-87	18-AUG-87
103	Raghu	19-SEP-87	19-OCT-87
104	Manoj	20-JUN-87	20-JUL-87
105	Ashwin	22-JUL-87	22-AUG-87
122	ramcharan	18-JUN-87	18-JUL-87
122	chanthuchari	18-JUN-89	18-JUL-89
122	surya	18-JUN-89	18-JUL-89
122	ravi	18-JUN-89	18-JUL-89
122	meera	18-JUN-89	18-JUL-89

10 rows selected.

17. Write a query to get first name, hire date and experience of the employees. `select emp_id,first_name,d_o_j,(sysdate-d_o_j)/365 as emp_experi from emp1;`

EMP_ID	FIRST_NAME	D_O_J	EMP_EXPERI
100	Naven	17-JUN-87	35.4763949
102	Siva	18-JUL-87	35.3914633
103	Sudha	19-SEP-87	35.2188606
104	Sumitha	20-JUN-87	35.4681757
105	Dheena	22-JUL-87	35.3805044
122	Sivan	18-JUN-87	35.4736551
122	Geetha	18-JUN-89	33.4709154
122	VARSHA	18-JUN-89	33.4709154
122	JANANI	18-JUN-89	33.4709154
122	MALINI	18-JUN-89	33.4709154

10 rows selected.

Result:

Types of function implemented successfully using SQL query.

Ex.No:4

TYPES OF OPERATORS

Aim:

To implement types of operators using SQL query.

Consider the following Products Table and OrderDetails Table

ProductID	ProductName	SupplierID	CotegoryID	Price
1	Chais	1	1	18
2	Chang	1	1	19
3	Aniseed Syrup	1	2	10
4	Chef Anton's Cajun Seasoning	2	2	22
5	Chef Anton's Gumbo Mix	2	2	21
6	Boysenberry Spread	3	2	25
7	Organic Dried Pears	3	7	30
8	Northwoods Cranberry Sauce	3	2	40
9	Mishi Kobe Niku	4	6	97

OrderDetails:

OrderDetailsID	OrderID	ProductID	Quantity
1	10248	1	12
2	10248	2	10
3	10248	3	15
4	10249	1	8
5	10249	4	4
6	10249	5	6
7	10250	3	5
8	10250	4	18
9	10251	5	2
10	10251	6	8
11	10252	7	9
12	10252	8	9
13	10250	9	20
14	10249	9	4

1. Write a SQL query to find the details of those product names supplied by supplier 2 or 3. `select * from product where sid=2 or sid=3;`

PID	PNAME	SID	CID	PRICE
4	chef antons cajun seasoning	2	2	22
5	chef antons gumbo mix	2	2	21
6	boysenberry spread	3	2	25
7	organic dried pears	3	7	30
8	northwoods cranberry sauce	3	2	40

2. Write a SQL query to find the details of those product names supplied by other than supplier 2 and 3. `select * from product where sid!=2 and sid!=3;`

PID	PNAME	SID	CID	PRICE
1	chais	1	1	18
2	chang	1	1	19
3	aniseed syrup	1	2	10
9	mishi kobe niku	4	6	97

3. Write a SQL query to retrieve the details of the product names whose names begin with any letter between 'A' and 'C' along with product id.

`select * from product where pname between 'b' and 'c';`

PID	PNAME	SID	CID	PRICE
6	boysenberry spread	3	2	25

4. Write a SQL query to find product names that have price ranges from 15 to 40 along with product id and rate .

`select pid,pname,price from product where price between 15 and 40;`

PID	PNAME	PRICE
1	chais	18
2	chang	19
4	chef antons cajun seasoning	22
5	chef antons gumbo mix	21
6	boysenberry spread	25
7	organic dried pears	30
8	northwoods cranberry sauce	40

7 rows selected.

5. Write a SQL query to find the product name except those whose names begin with any letter between 'A' and 'C' along with product id.

`select pid,pname from product where pname not between 'b' and 'c';`

PID	PNAME
1	chais
2	chang
3	aniseed syrup
4	chef antons cajun seasoning
5	chef antons gumbo mix
7	organic dried pears
8	northwoods cranberry sauce
9	mishi kobe niku

8 rows selected.

6. Write a SQL query to retrieve the details of the product names begins with the letter 'C'.

`select * from product where pname like 'c%';`

PID	PNAME	SID	CID	PRICE
1	chais	1	1	18
2	chang	1	1	19
4	chef antons cajun seasoning	2	2	22
5	chef antons gumbo mix	2	2	21

7. Write a SQL query to find the details of those products whose names begin with 'C' and the sixth character is 'a'. Rests may be any character along with product id, supplier id and rate. select pid, sid, price from product where pname like 'c_____a%';

PID	SID	PRICE
1	1	18
2	1	19
4	2	22
5	2	21

8. Find the name of all the products. select
pname from product;

```
PNAME
-----
chais
chang
aniseed syrup
chef antons cajun seasoning
chef antons gumbo mix
boysenberry spread
organic dried pears
northwoods cranberry sauce
mishi kobe niku
9 rows selected.
```

9. Find the name of the product if all the records in the OrderDetails have Quantity either equal to 6 or 2.

select pname from product where pid=all(select pid from order_details where quality=6 or quality=2);

```
PNAME
-----
chef antons gumbo mix
```

10. Find the OrderID whose maximum Quantity among all products of that OrderID is greater than the average quantity of all OrderID.

```
select oid from order_details group by oid having max(quantity)>all(select
avg(quantity) from order_details group by oid);
```

```
OID
-----
10250
10248
```

11. write a query to find the productId that is common in product and order details table. select pid from product intersect select pid from order_details;

```
PID
-----
1
2
3
4
5
6
7
8
9

9 rows selected.
```

12.write a query that returns multiple prodid common in product and order details table.

Select pid from product union all select pid from order_details;

PID	PID
1	3
2	1
3	4
4	5
5	3
6	4
7	5
8	6
9	7
1	8
2	9
	PID
	9
	23 rows selected.

Result:

Types of operators using SQL query implemented successfully.

Ex.No:5	JOINS
----------------	--------------

Aim:

To implement the different types of joins using SQL query.

Job_id	Job_title	Min_salary	Max_salary
Ad_press	President	20080	40000
Ad_Vp	Administration Vice President	15000	30000
IT_Prog	Programmer	4000	10000
FI_MGR	Finance Manager	8200	16000
FI_Acct	Accountant	4200	90000

1. Write a SQL query to find the employee name, department number, and department name for each employee.

select a.name,a.department_id,b.job_title from employees a, jobid b where
a.dep_id=b.dep_id;

```
SQL> select b.name,b.d_id,a.dep_name from dept a ,emp b where a.dep_id = b.d_id;
```

NAME	D_ID	DEP_NAME
david	60	IT
bruce	60	IT
alex	60	IT

```
SQL>
```

2. Write a SQL query to find the employee name, department, city, and state for each employee.

select a.name,d.dept_name,c.city,c.state from employees a,department d,location c
where d.location_id=c.location_id and a.dep_id=d.dep_id;

NAME	DEP_NAME	CITY	STATE
alex	IT	kochi	kerala
bruce	IT	kochi	kerala
david	IT	kochi	kerala

3. Write a SQL query to find the Employee name, salary, and job grade for all employees.

```
select e.name,e.salary,j.grade from emp e join grade j on e.salary between j.lowest and j.highest;
```

DEPT_ID	DEPT_NAME	COUNT(*)
10	Administration	1
40	Hr	1
60	IT	3
20	Marketing	1
30	Purchasing	1
50	Shipping	1

4. Write a SQL query to find all those employees who work in department ID 60 or 80. Return employee name,, department number and department name.

```
select e.name,e.d_id,dep_name.d from emp e join dept d on e.dep_id=d.dep_id and e.d_id in(60,80);
```

EMPLOYEE_ID	NAME	JOB_ID	DEPARTMENT_ID
103	Alex	IT_Prog	60
104	Bruce	IT_Prog	60
105	David	IT_Prog	60

5. Write a SQL query to find the employees who earn less than the employee of ID 103. Return employee name and salary.

```
select e.name,e.salary from emp e join emp s on e.salary < s.salary and s.emp_id=103; ( less 9000 )
```

NAME	SALARY
-----	-----
steveking	24000
lex	24000
bruce	6000
david	4800

6. Write a SQL query to find the employees and their managers. Return the name of the employee and manager.

```
select e1.name as manager,e2.name as employee from emp e1 join emp e2 on
e1.emp_id=e2.m_id;
```

MANAGER	EMPLOYEE
-----	-----
lex	alex
alex	bruce
alex	david

7. Write a SQL query to display the department name, city, and state for each department. select d.dep_name,l.city,l.state from dept d join location l on d.loc=l.loc_id;

DEPT_NAME	CITY	STATE
-----	-----	-----
Purchasing	Chennai	TN
Administration	Chennai	TN
Marketing	Mumbai	MH
Purchasing	Hyderabad	AP
Administration	Hyderabad	AP
Hr	Trivandrum	Kerala
Shipping	Bangalore	KK
IT	Kochi	Kerala

8. Write a SQL query to find the employees and their manager. these managers do not work under any manager.

```
Select e1.name as manager,e2.name as employee from emp e1 join emp2 on
e1.emp_id=e2.m_id where e1.m_id=0;
```


MANAGER	EMPLOYEE
Steve King	Lex

9. Write a SQL query to calculate the difference between the maximum salary and the salary of all the employees who work in the department of ID 80. Return job title, employee name and salary difference. Dep_id = 80 not found dep_id=60

Select j.job_title,e.name,j.max_sal-e.salary from emp e natural join j where e.d_id=60;

JOB_TITILE	NAME	J.MAX_SAL-E.SALARY
Programmer	david	5200
Programmer	bruce	4000
Programmer	alex	1000

10. write a SQL query to calculate the average salary of employees for each job title

select j.job_title,avg(e.salary) from emp e join job j on e.job_id = j.job_id group by j.job_title;

JOB_TITLE	AVG(E.SALARY)
Programmer	6600
president	24000
Administration vp	24000

11. write a SQL query to find the department name, name of the manager and their city.

select d.dep_name,e1.name,l.city from emp e1 join emp e2 on e1.emp_id=e2.m_id join dept d on d.dep_id=e1.d_id join location l on l.loc_id=d.loc;

DEP_NAME	NAME	CITY
IT	alex	kochi
IT	alex	kochi

12 . Write a SQL query to calculate the number of days worked by employees in a department of ID 60. Return employee ID, job title, number of days worked.

`select empid,jobid as jobtitle,hiredate from emp1 where deptid=60;`

```

SQL> select empid,jobid as jobtitle,hiredate from emp1 where deptid=60;

  EMPID JOBTITLE    HIREDATE
-----
    103 IT-Prod      03-JAN-06
    104 IT-Prod      21-MAY-07
    105 IT-Prod      25-JUN-05

SQL> |

```

13. Write a SQL query to find the name and salary of all employees working in any department in the city of Bangalore.

`select name,salary from emp1 e join dept d on e.deptid=d.deptid join loc l on d.locid=l.locid where l.city='Bangalore';`

```

SQL> select name,salary from emp1 e join dept d on e.deptid=d.deptid join lo
c l on d.locid=l.locid where l.city='Bangalore';

no rows selected

```

14. Write a SQL query to find the department name, department ID, and number of employees in each department.

`select d.deptid,d.deptname,count(*) as no_of_employees from dept d left join emp1 e on d.deptid=e.deptid group by d.deptid,d.deptname order by d.deptname;`

DEPT_ID	DEPT_NAME	COUNT(*)
10	Administration	1
40	Hr	1
60	IT	3
20	Marketing	1
30	Purchasing	1
50	Shipping	1

Result:

Different types of joins using SQL query was implemented successfully.

Ex.No:6

GROUP BY,ORDER BY,INDEXES

Aim:

To Create the following tables and apply GROUP BY, ORDER BY AND INDEX as follows.

I GROUP BY AND ORDER BY CLAUSE:

Create an order table

create table order1(customerid varchar2(10), order_date varchar2(26), item varchar2(20), quantity varchar2(20), price varchar2(10));

```
SQL> desc order1;
Name                               Null?    Type
-----
CUSTOMERID                         VARCHAR2(10)
ORDER_DATE                         VARCHAR2(26)
ITEM                               VARCHAR2(20)
QUANTITY                           VARCHAR2(20)
PRICE                              VARCHAR2(10)
SQL>
```

1. Find the number of items available based on the quantity.

Select quantity,count(item)as no_of_items from ord group by quantity;

```
SQL> select quantity,count(item) as no_of_items from ord group by quantity;

 QUANTITY NO_OF_ITEMS
-----
         1             4
         4             1
SQL>
```

2. How many customers placed orders in a day.

Select orddate, count(*) as no_of_customers from ord group by orddate;

```
SQL> select orddate, count(*) as no_of_customers from ord group by orddate;
```

ORDDATE	NO_OF_CUSTOMERS
01-JUL-99	2
06-JUL-99	1
30-JUN-99	2

3. Find the customer and order date who placed the highest order.

Select custid, orddate from ord group by custid, orddate, quantity having max(quantity) > 1

```
SQL> select orddate, custid from ord group by custid, orddate having max(quantity) > 1
```

ORDDATE	CUSTID
01-JUL-99	10101

4. Find the customer and order date who placed the least order.

Select custid, orddate from ord group by custid, orddate, quantity having min(quantity) > 1;

```
SQL> select orddate, custid from ord group by custid, orddate having min(quantity) <= 1;
```

ORDDATE	CUSTID
30-JUN-99	10330
30-JUN-99	10101
01-JUL-99	10298
06-JUL-99	10299

5. Find the price of 5 skateboard

Select item, price * 5 as price_of_5_skateboards from ord where item = 'skateboard';

```
SQL> select item,price *5 as price_of_5_skateboards from ord where item ='skateboard';
```

ITEM	PRICE_OF_5_SKATEBOARDS
skateboard	165

6. Display the item which is having the highest rate.

Select item from ord where price>1000;

```
SQL> select item from ord where price>1000;
```

ITEM
parachute

II INDEX:

Cust_code	Cust_name	City	Country
C00013	Sam	London	UK
C00001	Micheal	New york	USA
C00020	Albert	New York	USA
C00025	Ravindran	Bangalore	India

1. Create an index on 'custcity' column of the table 'customer'

Create index custrindx on custr(city);

```
SQL> create index custrindx on custr(city);  
Index created.
```

2. Create an index on a combination of 'custcity' and 'cust_country' columns of the table 'customer'.

Create index custrindx1 on custr(city,country);

```
SQL> create index custrindx1 on custr(city,country);  
Index created.
```

3. Create a unique index on 'cust_code' column in the table 'customer'

Create unique index custrunix on custr(cust_code);

```
SQL> create unique index custrunix on custr(cust_code);  
Index created.
```

Result:

Thus by using the GROUP BY , ORDER BY and INDEX the program has been successfully created and executed.

Ex.No:7

SUBQUERY AND VIEWS

Aim:

To implement the subqueries and to create views in SQL

Create the following table

Table Name :actor

Actor_id	Name	Gender
101	James Stewart	M
102	Peter	M
103	Robert DeNiro	M
104	Murray Abraham	M
105	Deborah	F
106	Harrison	M
107	Nicole Kidman	F
108	Jack	M
109	Mark	M
110	Claire	F

Table Name : Movie_cast:

Actor_id	Movie_id	Role
101	901	John Scottie Ferguson
102	902	T.E. Lawrance
103	903	Michael
104	904	Antony
101	905	Rick
106	906	Alice Stefen
107	907	Alvy Singer
108	908	Alice Harford
102	909	McManus
110	910	Eddir Adams

Table Name : Movie

Movie_id	Title	Movie_year	Movie_time	Movie_date_release	Movie_release_country
901	Vertigo	1958	128	24-08-1958	UK
902	The Innocent	1961	100	19-0-1962	UK

903	Lawrence of Arabia	1962	216	11-12-1962	UK
904	The Deer Hunter	1978	183	08-03-1979	UK
905	Amadeus	1984	160	07-01-1987	SW
906	Blade Runner	1982	117	09-09-1982	UK
907	Eyes wide shut	1999	159	08-08-1999	USA
908	The usual suspects	1995	106	25-08-1995	USA
909	Chinatown	1974	130	09-08-1974	UK
910	Bhogie nights	1997	155	16-02-1998	USA

Table Name: Rating

Movie_id	Rev_id	Rec_stars	No_of_ratings
901	9001	8	20000
908	9002	7	35162
903	9003	6	48525
904	9004	4	58469

909	9005	9	34899
906	9006	8	23527
907	9007	10	10348
908	9008	5	25632
909	9009	6	78953
910	9010	10	48599

Table name : Reviewer

Rev_id	Rev_name
9001	Righty Sock
9002	Jack Malvern
9003	Flowrence
9004	Alex
9005	
9006	Victor
9007	Simon
9008	
9009	Neal
9010	

1. Write a SQL query to find the actors who played a role in the movie 'The Deer Hunter'. Return all the fields of actor table.

Select * from actor where actorid =all(select actorid from moviecast where movieid =(select movieid from movie where title ='The Deer Hunter'));

```
SQL> select * from actor where actorid =all(select actorid from moviecast where movieid =(select movieid from movie where title ='The Deer Hunter'));
```

ACTORID	NAME	G
104	Murray Abraham	M

2. Write a SQL query to find those movies that have been released in countries other than the United Kingdom. Return movie title, movie year, movie time, and date of release, releasing country.

Select title,movieyear,movietime,moviedatereleased,movierelcountry from movie where movierelcountry!='uk';

```
SQL> select title,movieyear,movietime,moviedatereleased,movierelcountry from movie where movierelcountry!='uk';
```

TITLE	MOVIEYEAR	MOVETIME	MOVIEDATE	MOVI
Amadeus	1984	160	07-JAN-87	sw
Eyes Wide Shut	1999	159	08-AUG-99	usa
The Usual Suspects	1995	106	25-AUG-95	usa
Bhogie nights	1997	155	16-FEB-98	usa

3. Write a SQL query to find movies that have been reviewed by a reviewer and received a rating. Group the result set on reviewer's name, movie title. Return reviewer's name, movie title.

Select revname,title from reviewer r,rating r1,movie m where r.revid=r1.revid and m.movieid=r1.movieid and r.revname IS NOT NULL group by revname,title;

```
SQL> select revname,title from reviewer r,rating r1,movie m where r.revid=r1
.revid and m.movieid=r1.movieid and r.revname IS NOT NULL group by revname,t
itle;
```

REVNAME	TITLE
Righty Sock	Vertigo
simon	Eyes Wide Shut
alex	The Deer Hunter
Flowerence	Lawrence of Arabia
jack Malvern	The Usual Suspects
Neal	Chinatown
victor	Blade Runner

7 rows selected.

4. write a SQL query to find movies that have been reviewed by a reviewer and received a rating. Sort the result-set in ascending order by reviewer name, movie title, review Stars. Return reviewer name, movie title, review Stars.

Select revname,title,revstars from reviewer r,rating r1,movie m where r.revid=r1.revid and m.movieid=r1.movieid and r.revname IS NOT NULL order by revname,title,revstars;

REVNAME	TITLE	REVSTARS
Flowerence	Lawrence of Arabia	6
Neal	Chinatown	6
Righty Sock	Vertigo	8
alex	The Deer Hunter	4
jack Malvern	The Usual Suspects	7
simon	Eyes Wide Shut	10
victor	Blade Runner	8

7 rows selected.

5. Write a SQL query to find movies that have been reviewed by a reviewer and received a rating. Group the result set on reviewer's name, movie title. Return reviewer's name, movie title.

Select revname,title from reviewer r,rating r1,movie m where r.revid=r1.revid and m.movieid=r1.movieid and r.revname IS NOT NULL group by revname,title;

```
SQL> select revname,title from reviewer r,rating r1,movie m where r.revid=r1
.revid and m.movieid=r1.movieid and r.revname IS NOT NULL group by revname,t
itle;
```

REVNAME	TITLE
Righty Sock	Vertigo
simon	Eyes Wide Shut
alex	The Deer Hunter
Flowerence	Lawrence of Arabia
jack Malvern	The Usual Suspects
Neal	Chinatown
victor	Blade Runner

7 rows selected.

6. Write a SQL query to find those movies, which have received highest number of stars. Group the result set on movie title and sorts the result-set in ascending order by movie title.

Return movie title and maximum number of review stars.

Select title,max(revstars) from movie m,rating r where m.movieid=r.movieid group by title order by title;

```
SQL> select title,max(revstars) from movie m,rating r where m.movieid=r.mov
ieid group by title order by title;
```

TITLE	MAX(REVSTARS)
Bhogie nights	10
Blade Runner	8
Chinatown	9
Eyes Wide Shut	10
Lawrence of Arabia	6
The Deer Hunter	4
The Usual Suspects	7
Vertigo	8

8 rows selected.

7. Write a SQL query to find all reviewers who rated the movie 'China Town'. Return reviewer name.

select revname from reviewer r,rating r1,movie m where r.revid=r1.revid and r1.movieid=m.movieid and title='Chinatown';

```
SQL> select revname from reviewer r,rating r1,movie m where r.revid=r1.revid
and r1.movieid=m.movieid and title='Chinatown';
```

REVNAME
Neal

8. write a SQL query to find the movies with the lowest ratings. Return reviewer name, movie title, and number of stars for those movies.

select revname,title,revstars from rating,movie,reviewer where revstars=(select min(revstars) from rating) and rating.revid=reviewer.revid and movie.movieid=rating.movieid;

```
SQL> select revname,title,revstars from rating,movie,reviewer where revstars=(select min(revstars) from rating) and rating.revid=reviewer.revid and movie.movieid=rating.movieid;
```

REVNAME	TITLE	REVSTARS
alex	The Deer Hunter	4

9. Write a query in SQL to find the movies in which one or more actors appeared in more than one film.

Select title from movie where movieid in(select movieid from moviecast where actorid in(select actorid from actor where actorid in (select actorid from moviecast group by actorid having count(actorid)>1)));

```
SQL> select title from movie where movieid in(select movieid from moviecast where actorid in(select actorid from actor where actorid in (select actorid from moviecast group by actorid having count(actorid)>1)));
```

TITLE
Vertigo
The Innocent
Amadeus
Chinatown

10. Write a query to create reviewer_view from reviewer table.

create view reviewer_view as select * from reviewer;

```
SQL> create view reviewer_view as select * from reviewer;
View created.
```

11. Write a query to update the empty values in reviewer_view

update reviewer_view set revname= 'Nelson' where revid=9005;

update reviewer_view set revname= 'Sherin' where revid=9008;

update reviewer_view set revname= 'Michael' where revid=9010;

```
SQL> select * from reviewer_view;
-----
REVID  REVNAME
-----
9001   Righty Sock
9002   Jack Malvern
9003   Flowerence
9004   alex
9005   Nelson
9006   victor
9007   simon
9008   Sherin
9009   Neal
9010   Michael
10 rows selected.
```

12. Write a query to sort the reviewer_view by name.

Select reviewer_view.revname from reviewer_view order by(revname);

```
SQL> select reviewer_view.revname from reviewer_view order by(revname);
REVNAME
-----
Flowerence
Michael
Neal
Nelson
Righty Sock
Sherin
alex
jack Malvern
simon
victor
10 rows selected.
```

Result:

The subqueries and views has been created and executed successfully.

Ex.No:8	CONSTRAINTS
----------------	--------------------

Aim:

To Create the following tables and apply constraints as follows

1. Books Table:

- a. isbn – primary key
- b. title
- c. price
- d. quantity – not null

2. Authors Table:

- a. author_id – primary key

b. email - unique 3. Book_Authors:

- a. isbn – foreign key references books table
- b. author_id – foreign key references authors table Table:

- To create a book table:

Create table book(isbn integer, title varchar(25), price numeric(7,2), quantity integer not null, constraint pk_isbn, primary key(isbn));

```

Run SQL Command Line
SQL> Create table book(isbn integer, title varchar(25), price numeric(7,2), qty integer not null, constraint pk_isbn primary key(isbn));
Table created.

SQL> describe book;
Name                               Null?   Type
-----
ISBN                               NOT NULL NUMBER(38)
TITLE                             VARCHA2(25)
PRICE                             NUMBER(7,2)
QTY                               NOT NULL NUMBER(38)
SQL>

```

- To create author table:
Create table author(author_id integer, email varchar(25), constraint pk_id primary key(author_id), constraint uk_email unique(email));

```

Run SQL Command Line
SQL> Create table author(author_id integer, email varchar(25), constraint pk_id primary key(author_id), constraint uk_email unique(email));
Table created.

SQL> describe author;
Name                               Null?   Type
-----
AUTHOR_ID                         NOT NULL NUMBER(38)
EMAIL                             VARCHA2(25)
SQL>

```

- To create bookauthor table:
Create table bookauthor(isbn integer, author_id integer, constraint fk_isbn foreign key(isbn) references book(isbn), constraint fk_author_id foreign key(author_id) references author(author_id));

```

Run SQL Command Line
SQL> Create table bookauthor(isbn integer, author_id integer, constraint fk_isbn foreign key(isbn) references book(isbn), constraint fk_author_id foreign key(author_id) references author(author_id));
Table created.

SQL> describe bookauthor;
Name                               Null?   Type
-----
ISBN                               NUMBER(38)
AUTHOR_ID                         NUMBER(38)
SQL>

```

Q1. Add unique constraint to title in books table

Alter table book add unique(title);

Q2. Add not null constraint to price in books table

Alter table book modify price numeric not null;

Q3. Alter not null constraint in price attribute in books table and set the check constraint so that value is greater than 0.0

Alter table book add constraint ck_price check(price>0.0);

Q4. Drop not null constraint for quantity in books table

Alter table book modify quantity integer null;

Q5. Set a default value of quantity in books table as 0

Alter table book modify quantity default 0;

Q6. Drop unique constraint for email attribute in authors table

Alter table author drop constraint uk_email;

Q7. Drop any one foreign key constraint.

Alter table bookauthor drop constraint fk_isbn;

Q8. Drop a primary key [after referenced foreign key is dropped]

Alter table book drop constraint pk_isbn;

Q9. Add an attribute for the authors table and set a constraint for it.

Alter table author add(author_name varchar(20) not null);

```
SQL> Run SQL Command Line
SQL> alter table book add unique(title);
Table altered.
SQL> alter table book modify price numeric not null;
Table altered.
SQL> alter table book add constraint ck_price check(price>0.0);
Table altered.
SQL> alter table book modify qty integer null;
Table altered.
SQL> alter table book modify qty default 0;
Table altered.
SQL> alter table author drop constraint uk_email;
Table altered.
SQL> alter table bookauthor drop constraint fk_isbn;
Table altered.
SQL> alter table book drop constraint pk_isbn;
Table altered.
SQL> alter table author add(author_name char(20) not null);
Table altered.
SQL> _
```

Result:

The subqueries and views has been created and executed successfully

Aim:

To implement Transaction control language by using SQL query.

create table emp1 with the following constraints: 1)EmpId number(12),2)Ename varchar(20),3)City varchar(12);

- 1) insert two values and implement commit operation,
- 2) insert one value and implement rollback operation, 3) insert one new values and implement savepoint operation.

Table Format:**Emp1:**

Name	Null?	Type
-----	-----	-----
EMPID		NUMBER(12)
ENAME		VARCHAR2(16)
CITY		VARCHAR2(12)

insert into emp1 values(101,'Ankit','Bangalore'); insert
into emp1 values(102,'rahul','chennai');

```
SQL> create table emp1(empid number(12),ename varchar(16),city varchar(12));
Table created.
SQL> insert into emp1 values(101,'Ankit','Bangalore');
1 row created.
SQL> insert into emp1 values(102,'rahul','chennai');
1 row created.
SQL> commit;
Commit complete.
```

Insert into emp1 values(103,'akshay','Coimbatore');
rollback;

```
SQL> insert into emp1 values(103,'akshay','Coimbatore');
1 row created.
SQL> rollback;
Rollback complete.
SQL> select * from emp1;
```

EMPID	ENAME	CITY
101	Ankit	Bangalore
102	rahul	chennai

savepoint s;
insert into emp1 values(111,'shalini','Chennai'); rollback
to s;

```
SQL> savepoint s;
Savepoint created.
SQL> insert into emp1 values(111,'shalini','Chennai');
1 row created.
SQL> select * from emp1;
```

EMPID	ENAME	CITY
101	Ankit	Bangalore
102	rahul	chennai
111	shalini	Chennai

```
SQL> rollback to s;
Rollback complete.
SQL> select * from emp1;
```

EMPID	ENAME	CITY
101	Ankit	Bangalore
102	rahul	chennai

Result:

Transaction control language is implemented successfully by using SQL query.

EX. NO: 10	FIBONACCI SERIES
-------------------	-------------------------

Aim:

To Create and implement Fibonacci series in Pl-Sql.

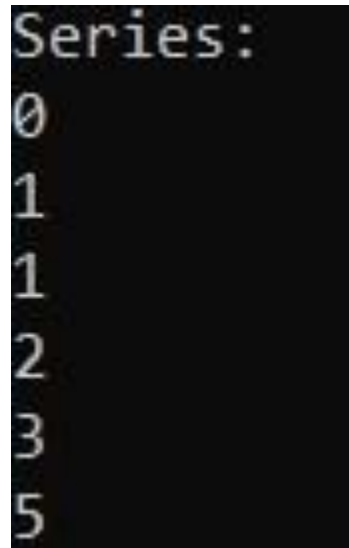
Program Coding:

```
set serveroutput on
set verify off
declare first number
:= 0; second number
:= 1; temp number;
n number := 5; i
number; begin
dbms_output.put_line('Series:');
dbms_output.put_line(first);
dbms_output.put_line(second);
for i in 2..n loop
temp:=first+second; first :=
second; second := temp;
dbms_output.put_line(temp);
end loop;
```

```
end;
```

```
/
```

Output:



```
Series:  
0  
1  
1  
2  
3  
5
```

Result:

Fibonacci series using PL/SQL implemented successfully.

EX. NO: 11	ARMSTRONG NUMBER
------------	-------------------------

Aim:

To create program to check whether the given number is Armstrong number or not.

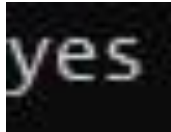
Program:

```
declare n
number:=1634; s
number:=0; r
number; len
number; m
number; begin m
:= n;
len := length(to_char(n));
while n>0 loop r :=
mod(n , 10); s := s +
power(r , len); n :=
trunc(n / 10); end loop; if
m = s then
dbms_output.put_line('y
```



```
es'); else  
dbms_output.put_line('n  
o'); end if; end;  
/
```

Output:



Result:

Armstrong number checked successfully.

EX. NO: 12

EXCEPTION HANDLING

Aim:

To handle various exceptions that may occur in pl sql.

Program:

create table students with following constraints ,1)G_id,2)G_name,3)marks

- 1)use exception block handle no_data_found error,
- 2)use exception block handle too_many_rows error,
- 3)use exception block handle value_error error, 4)use
exception block handle zero_divide error, **Table**

structure:

Students:

Name	Null?	Type
G_ID		NUMBER(12)
G_NAME		VARCHAR2(16)
MARKS		NUMBER(12)

G_ID	G_NAME	MARKS
1	suraj	100
2	praveen	97
3	jessie	99

DECLARE

temp varchar(20);

BEGIN

SELECT g_id into temp from students where g_name='deepthi'; exception

WHEN no_data_found THEN

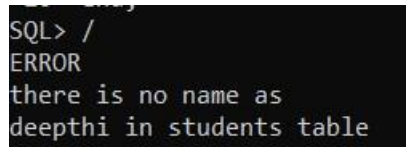
dbms_output.put_line('ERROR');

dbms_output.put_line('there is no name as');

dbms_output.put_line('deepthi in students table'); end;

/

Output:



```
SQL> /  
ERROR  
there is no name as  
deepthi in students table
```

DECLARE

temp varchar(20);

BEGIN

SELECT g_name into temp from students ; dbms_output.put_line(temp);

EXCEPTION

WHEN too_many_rows THEN

dbms_output.put_line('error trying to SELECT too many rows'); end;

/

Output:

```
SQL> /
error trying to SELECT too many rows

PL/SQL procedure successfully completed.
```

DECLARE

temp number(12);

BEGIN

SELECT g_name into temp from students where g_name='suraj';

dbms_output.put_line('the g_name is '||temp);

EXCEPTION

WHEN value_error THEN

dbms_output.put_line('Error');

dbms_output.put_line('Change data type of temp to varchar(20)');

END;

/

Output:

```
SQL> /
Error
Change data type of temp to varchar(20)

PL/SQL procedure successfully completed.

SQL> ed
Wrote file afiedt.buf
```

DECLARE

a int:=10; b

int:=0;

answer int;

BEGIN

```
answer:=a/b;

dbms_output.put_line('the result after division is'||answer); exception
WHEN zero_divide THEN

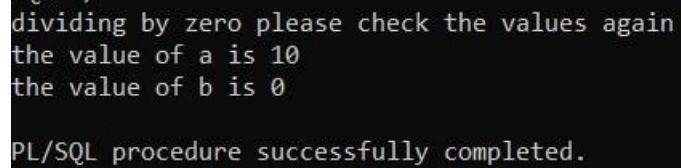
dbms_output.put_line('dividing by zero please check the values again');

dbms_output.put_line('the value of a is '||a); dbms_output.put_line('the
value of b is '||b);

END;

/
```

Output:

A screenshot of a terminal window with a black background and white text. It shows the output of a PL/SQL procedure. The first line is 'dividing by zero please check the values again', followed by 'the value of a is 10' and 'the value of b is 0'. The final line is 'PL/SQL procedure successfully completed.'.

```
dividing by zero please check the values again
the value of a is 10
the value of b is 0

PL/SQL procedure successfully completed.
```

Result:

PL/SQL exception handling executed successfully.