

UNIVERSITY COLLEGE OF ENGINEERING

(OSMANIA UNIVERSITY)



CERTIFICATE

This is to certify that Mr. /Miss _____
is a student of **MCA** _____ year _____ Semester bearing Hall Ticket
No _____ has done the Practical Lab Record in
Database Management System Lab during the academic year 2023-24 .

INTERNAL EXAMINAR

EXTERNAL EXAMINAR

HEAD OF THE DEPARTMENT

DBMS

LAB RECORD

INDEX

S.No	Title	Page No.
1.	Database Schema for a customer-sale scenario Customer(Cust id : integer, cust_name: string) Item(item_id: integer, item_name: string, price: integer) Sale(bill_no: integer, bill_data: date, cust_id: integer, item_id: integer, qty_sold: integer).	1-3
2.	Database Schema for a Student Library scenario.	4-6
3.	Database Schema for a Employee-pay scenario.	7-9
4.	Create the following tables : Student(roll-no, name, date-of-birth, course-id) Course (Course-id, cname, fee, duration).	10-11
5.	Create the following table : Student (roll-no, name, subject-name) Subject(faculty-code, faculty-name, specialization).	12-13
6.	Create the following table : Item (item-code, item-name, qty-in-stock, reorder-level) Supplier (supplier-code, supplier-name, address) Can-supply(supplier-code, item-code) (i)List all those suppliers who can supply the given item. (ii) List all those items which cannot be supplied by given company.	14-17
7.	Create the following tables: Student (roll-no, marks, category, district, state) Student-rank(roll-no, marks, rank) (i) List all those students who have come from Tamilnadu state and secured a rank above 100. (ii) List all those students who come from Andhra Pradesh state and belong to given category who have secured a rank above 100.	18-20
8.	Create the following tables : Branch (branch-id, branch-name, branch-city) Customer (customer-id, customer-name, customer-city, branch-id).	21-22
9.	Create the following tables Branch(branch_id,branch_name,branch_city) Customer(customer_id, customer_name,customer_city,branch_id) a) List all those all customers who live in the same city as the branch in which they have account.	23-25
10.	Create the following tables : Student(roll-no, name, date-of-birth, course-id) Course (Course-id, name, fee, duration) (i) List all those students who are between 18-19 years of age and have opted for MCA course. (ii) List all those courses in which number of students are less than 10.	26-28
11.	Create the following tables : Student (roll-no, name, subject-opted) Subject – rank (subject-code, subject-name, faculty-code) Faculty (faculty-code, faculty-name, specialization) i) Find the number of students who have enrolled for the subject "DBMS" ii) Find all those subjects which are not offered by any faculty members.	29-32
12.	Create the following tables : Student (roll-no, name, subject-opted) Subject – rank (subject-code, subject-name, faculty-code) Faculty (faculty-code, faculty-name, specialization) Find the number of students who have enrolled for the subject "OS" (i) Find all those students who opted for more than 5 subjects.	33-36
13.	Create a table to represent sb-account of a bank consisting of account-no, customername, balance-amount . Write a PL/SQL block to implement deposit and withdraw. Withdraw should not be allowed if the balance goes below RS.100.	37-39

14.	Create the following tables : Item (item-code, item-name, qty-in-stock, reorder-level) Supplier (supplier-code, supplier-name, address, status) Can-supply(supplier-code, item-code) (a) Write PL/SQL procedure to do the following : Generate a report of those items that are supplied by those suppliers whose status is "important".	40-42
15.	Create the following tables for Library Information System : Book : (accession-no, title, publisher, author, status) Status could be issued, present in the library, sent for binding, and cannot be issued. Write a trigger which sets the status of a book to "cannot be issued", if it is published 20 years back.	43-44
16.	Create the following tables : a)Book(accession-no, title, publisher, year, date-of-purchase, status) Member(member-id, name, number-of-books-issued, max-limit) Book-issue(accession-no, member-id, date-of-issue).	45-48
17.	Write a Program to implement a FUNCTION.	49
18.	Write a Program to implement a CURSOR.	50
19.	Write a PL/SQL code to find the Largest of Three Numbers.	51
20.	Write a PL/SQL code to find the Factorial of a given Number.	52
21.	Write a PL/SQL Program for Palindrome Number.	53
22.	Write a PL/SQL Program for Reverse of a Number.	54
23.	Write a PL/SQL Program to Reverse a String.	55
24.	Write a PL/SQL code to display employee Details using cursors.	56
25.	Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values.	57
26.	Write a PL / SQL program to check whether the given number is prime or not.	58

Prog 1. Database Schema for a customer-sale scenario

Customer(Cust id : integer, cust_name: string)

Item(item_id: integer, item_name: string, price: integer)

Sale(bill_no: integer, bill_data: date, cust_id: integer, item_id: integer, qty_sold: integer)

SQL> create table customer1 (cust_id number(5) primary key, cust_name varchar2(15));

Output: Table created.

SQL>desc customer1;

Output:

Name	Null?	Type

CUST_ID	NOT NULL	NUMBER(5)
CUST_NAME		VARCHAR2(15)

b) SQL> insert into customer1 values(&custid,'&custname');

SQL> select * from customer1;

Output:

CUST_ID	CUST_NAME

100	ramu
101	kamal
102	raju
103	rajusundaram
104	lawrence

SQL> create table item(item_id number(4) primary key,

Item_name varchar2(15),price number(6,2));

SQL>dsec item

Output:

Name	Null?	Type

Cust_id	NOT NULL	NUMBER(4)
Item_name		VARCHAR2(15)
PRICE		NUMBER(6,2)

```
SQL>insert into item values(&item_id,'&item_name',&price);
```

```
SQL> select * from item;
```

Output:

ITEM_ID	ITEM_NAME	PRICE
2334	geera	6.25
4532	corn soup	34.65
2124	lays chips	20
4531	setwet	99.99
2319	duracell	45.5

```
SQL>create table sale(bill_no number(5) primary key,bill_date date, cust_id number(5)
references customer(cust_id), item_id number(4) references item(item_id),qty_sold
number(4));
```

Out put: Table Created.

```
SQL>dsc sale
```

Output:

Name	Null?	Type
BILL_NO	NOT NULL	NUMBER(4)
BILL_DATE		DATE
CUST_ID		NUMBER(5)
ITEM_ID		NUMBER(4)
QTY_SOLD		NUMBER(4)

```
SQL>insert into Sale values(&bill_no, '&bill_date',
&cust_id, &item_id, &qty_sold);
```

```
SQL>select * from sale;
```

Output:

BILL_NO	BILL_DATE	CUST_ID	ITEM_ID	QTY_SOLD
1450	04-JAN-06	100	2124	2
1451	04-JAN-06	101	2319	1
1452	04-JAN-06	103	4531	2
1453	04-JAN-06	102	2334	3
1454	04-JAN-06	104	4532	3

c) List all the bills for the current date with the customer names and item numbers

SQL> select c.custname, i.itemid, s.billno from customer c, item I, sale s

Where c.custid=s.custid and

s.billdate=to_char(sysdate);

CUSTNAME	ITEMID	BILLNO
----------	--------	--------

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

John	5001	332
------	------	-----

d) List the total Bill details with the quantity sold, price of the item and the final amount

SQL> select i.price, s.qty,(i.price*s.qty) total from item I, sale s where i.itemid=s.itemid;

PRICE	QTY	TOTAL
-------	-----	-------

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

120	2	240
-----	---	-----

20	3	60
----	---	----

5	2	10
---	---	----

10	1	10
----	---	----

350	4	1400
-----	---	------

e) List the details of the customer who have bought a product which has a price>200

SQL> select c.custid, c.custname from customer c, sale s, item i where i.price>200 and

c.custid=s.custid and i.itemid=s.itemid;

CUSTID	CUSTNAME
--------	----------

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

Duffy	
-------	--

2. Database Schema for a Student Library scenario

```
SQL>create table student(stud_no number(5) primary key,stud_name varchar2(15));
```

```
SQL>desc student;
```

Name	Null?	Type
STUD_NO	NOT NULL	NUMBER(5)
STUD_NAME		VARCAHR2(15)

```
SQL>insert into student values(&stud_no,&stud_name');
```

```
SQL>select * from student;
```

STUD_NO	STUD_NAME
508	HARISH
513	BALAJI
518	RAKESH
524	PAVAN
534	JOYCE

```
SQL>create table membership(mem_no number(5) primary key,stud_no number(5) references student(stud_no));
```

```
SQL>dsec membership;
```

Name	Null?	Type
MEM_NO	NOT NULL	NUMBER(5)
STUD_NO		NUMBER(5)

```
SQL>insert into membership values(&mem_no,&stud_no);
```

Enter value for mem_no:5440

Enter value for stud_no:510

Old 1:insert into membership values(&mem_no,&stud_no)

New 1:insert into membership values(5440,510)

Insert into membership values(5440,510)

*

Errors Observed:

ERROR at line 1:

ORA-02291:integrity constraint(HARISH.SYS_C002724)violated-primary key not found

```
SQL>select * from membership;
```

MEM_NO	STUD_NO
--------	---------

.....

513	
-----	--

508	
-----	--

518	
-----	--

534	
-----	--

524	
-----	--

```
SQL>create table book(book_no number(5) primary key,book_name varchar2(20),author  
varchar2(20));
```

```
SQL>desc book;
```

Name	Null?	Type
BOOK_NO	NOT NULL	NUMBER(5)
BOOK_NAME		VARCHAR2(20)
AUTHOR		VARCHAR2(20)

```
SQL>insert into book values(&book_no,'&book_name','&author');
```

```
SQL>select * from book;
```

BOOK_NO	BOOK_NAME	AUTHOR
9123	DBMS	Rama Krishna
2342	JAVA	Robettwilkins
4523	Fearless tales	Alfred
8723	my ambition	Harish
7821	Harry Potter	JK Rowling

```
SQL>create table lss_rec(iss_no number primary key,iss_datedate,mem_no number(5)  
references membership(mem_no),book_no number(5) references book(book_no));
```

```
SQL>desc lss_rec;
```

Name	Null?	Type
ISS_NO	NOT NULL	NUMBER
ISS_DATE		DATE
MEM_NO		NUMBER(5)
BOOK_NO		NUMBER(5)

SQL>select * from iss_rec;

ISS_NO	ISS_DATE	MEM_NO	BOOK_NO
43	05-JAN-06	5443	4523
81	28-DEC-05	5441	8723
22	08-DEC-05	5440	7821
53	07-JAN-06	5442	9123
35	06-JAN-06	5444	2342

B) List all the student names with their membership numbers

SQL> select s.studname, m.memno from student s, membership m

Where m.studno=s.studno;

STUDNAME MEMNO

STUDNAME	MEMNO
Abhijeet	1001
Arun	1002
Arvind	1003
Ashish	1004
Ashwin	1005

C) List all the issues for the current date with student and Book names

SQL> select i.issno, s.studname, b.bookname from iss_rec i, membership

M, student s, book b where i.memno=m.memno and

m.studno=s.studno and i.issdate=to_char(sysdate);

ISSNO STUDNAME BOOKNAME

ISSNO	STUDNAME	BOOKNAME
13	arvind	P&S

D) Give a count of how many books have been bought by each student

SQL> select s.studno, count(i.bookno) from student s, membership m, book b, iss_rec i
where s.studno=m.studno and b.bookno=i.bookno group by s.studno;

STUDNO COUNT(I.BOOKNO)

STUDNO	COUNT(I.BOOKNO)
501	5
502	5
503	5
504	5
505	5

3. Database Schema for a Employee-pay scenario

Create table employee(emp_idint(5) primary key,emp_name varchar2(25));

SQL>desc employee;

Name	Null?	Type
.....		
EMP_ID	NOT NULL	NUMBER(5)
EMP_NAME		VARCHAR2(25)

SQL>insert into employee values(&emp_id,&emp_name');

SQL>select * from employee;

EMP_ID	EMP_NAME
.....	
10	Robert
21	Coulthard
30	Fernando Alonso
39	Kartikeyan
87	Kimmi

Create table department(dept_idint(5) primary key,dept_name varchar2(20));

SQL>desc department;

Name	Null?	Type
.....		
DEPT_ID	NOT NULL	NUMBER(5)
DEPT_NAME		VARCHAR2(20)

SQL>insert into department values(&dept_id,&dept_name');

SQL>select * from department;

DEPT_ID	DEPT_NAME
.....	
	Sales
	Accounts
	Administration
	Production
	Supervisor

```
SQL>create table paydetails(emp_idint(5) references employee(emp_id),dept_idint(5)
reerences department(dept_id),basic int(7,2),deductions int(5,2),Additions int(5,2),doj
date);
```

```
SQL>descpaydetails;
```

Name	Null?	Type
.....		
EMP_ID		NUMBER(5)
DEPT_ID		NUMBER(5)
BASIC		NUMBER(7,2)
DEDUCTIONS		NUMBER(5,2)
ADDITIONS		NUMBER(5,2)
DOJ		DATE

```
SQL>insert into paydeatils values(&emp_id,&dept_id,
&basic,&deductions,&additions,&doj);
```

```
SQL>select * from paydeatils;
```

EMP_ID	DEPT_ID	BASIC	DEDUCTIONS	ADDITIONS	DOJ
.....					
10	101	25023.12	43.09	71.23	08-JAN-93
21	100	10500.29	23.98	40.9	01-JAN-06
30	102	6500.5	30.54	15	06-JUL-97
39	103	9700.45	32.78	65.09	08-AUG-03
87	104	15000	97.66	154.8	24-SEP-04

```
Create table payroll(emp_idint(5)references employee(emp_id),pay_date date);
```

```
SQL>desc payroll;
```

Name	Null?	Type
.....		
EMP_ID		NUMBER(5)
PAY_DATE		DATE

```
SQL>insert into payroll values(&emp_id,&date');
```

```
SQL>select * from payroll;
```

EMP_ID	PAY_DATE
--------	----------

.....

10	31-JAN-06
----	-----------

21	03-FEB-06
----	-----------

30	15-JAN-06
----	-----------

39	27-JAN-06
----	-----------

87	04-FEB-06
----	-----------

List all the employee names who joined after particular date

```
SQL>select e,empname from employee e,paydet p where e.empid=p.empid
```

```
And p.doj>='05-mar-06';
```

EMPNAME

.....

AVINASH

NNITI

PHALGUN

List the details of employees whose basic salary is between 10k and 20k

```
SQL>Select empid,empname from employee where salary between 10kand 20k;
```

EMPID	EMPNAME
-------	---------

.....

AKHILA

Aaaaaaaa

4. Create the following tables :

Student(roll-no, name, date-of-birth, course-id)

Course (Course-id, cname, fee, duration)

```
SQL>create table student(roll_no number(2),name varchar2(20),dob date,course_id number(3));
```

Table Created

```
SQL> insert into student values(1,'Vignesh','15-MAR-90',101);
```

1 row inserted

```
SQL> insert into student values(2,'Ramesh','17-APR-90',101);
```

1 row inserted

```
SQL> insert into student values(3,'Naveen','21-MAY-91',102);
```

1 row inserted

```
SQL> insert into student values(4,'Ashok','01-JAN-91',103);
```

1 row inserted

```
SQL> insert into student values(5,'Sandeep','23-AUG-90',104);
```

1 row inserted

```
SQL>Select * from student;
```

Roll_no	name	dob	Course_id
1	Vignesh	15-MAR-90	101
2	Ramesh	17-APR-90	101
3	Naveen	21-MAY-91	102
4	Ashok	01-JAN-91	103
5	Sandeep	23-AUG-90	104

```
SQL>create table course(course_id number(3), cname varchar2(20),fee number(5),duration number(1));
```

Table Created

```
SQL> insert into course values(101,'MCA',30000,3);
```

1 row inserted

```
SQL> insert into course values(102,'M.Tech',35000,2);
```

1 row inserted

SQL> insert into course values(103,'MBA',33000,2);

1 row inserted

SQL> insert into course values(104,'B.Tech',27000,4);

1 row inserted

SQL>Select * from course;

Course_id	cname	fee	duration
101	MCA	30000	3
102	M.Tech	35000	2
103	MBA	33000	2
104	B.Tech	27000	4

(b) Generate queries to do the following :

(i) List all those students who are greater than 18 years of age and have opted for MCA course.

SQL>select s.roll_no,s.name,s.dob,c.cname "Course" from student s,course c where s.course_id=c.course_id and (sysdate-dob)/365>18 and c.cname='MCA';

SQL>select roll_no, name,dob,cname from student,course where student.course_id=course.course_id and (sysdate-dob)/365>18 and cname='MCA';

Roll_no	name	dob	Cname
1	Vignesh	15-MAR-90	MCA
2	ramesh	17-APR-90	MCA

(ii) List all those courses whose fee is greater than that of MCA course.

SQL> select cname "Course",fee from course where fee>(select fee from course where name='MCA');

SQL>select cname,fee from course where fee>(cname='MCA');

cname	Fee
M.Tech	35000
MBA	33000

5. Create the following table :

Student (roll-no, name, subject-name)

Subject(faculty-code, faculty-name, specialization)

```
SQL>create table student(roll_no number(2),name varchar2(20),subject-name  
varchar2(20));
```

Table Created

```
SQL> insert into student values(1,'Rajesh','SE');
```

1 row inserted

```
SQL> insert into student values(2,'Ashok','DBMS');
```

1 row inserted

```
SQL> insert into student values(3,'Sharma','JAVA');
```

1 row inserted

```
SQL> insert into student values(4,'Naveen','DBMS');
```

1 row inserted

```
SQL> insert into student values(5,'Raju','DAA');
```

1 row inserted

```
SQL>Select * from student;
```

Roll_no	name	Subject-name
1	Rajesh	SE
2	Ashok	DBMS
3	Sharma	JAVA
4	Naveen	DBMS
5	Raju	DAA

```
SQL>create table subject(fcode number(2),fname varchar2(20),specialization varchar2(20));
```

Table Created

```
SQL> insert into subject values(11,'Sai','SE');
```

1 row inserted

```
SQL> insert into subject values(12,'Ramu','DBMS');
```

1 row inserted

SQL> insert into subject values(13,'Narsimha','java');

1 row inserted

SQL> insert into subject values(14,'Vignesh','DAA');

1 row inserted

SQL> insert into subject values(15,'Ramesh','CPP');

1 row inserted

SQL>Select * from subject;

fcode	fname	Specialization
11	Sai	SE
12	Ramu	DBMS
13	Narsimha	JAVA
14	Vignesh	DAA
15	Ramesh	CPP

(a) Generate queries to do the following :

(i) Find the number of students who have enrolled for the subject "DBMS".

SQL>select count(*) "No of Students" from students where subject='DBMS';

No of students
2

(ii) Find all those faculty members who have not offered any subject.

SQL> select fname,specialization,name from subject,student where
subject.specialization(+) = student.subject-name;

fname	Specialization	name
Ramesh	CPP	
Sai	SE	Rajesh
Ramu	DBMS	Ashok
Narsimha	JAVA	Sharma
Ramu	DBMS	Naveen
Vignesh	DAA	Raju

6. Create the following table :

Item (item-code, item-name, qty-in-stock, reorder-level)

Supplier (supplier-code, supplier-name, address)

Can-supply(supplier-code, item-code)

(i) List all those suppliers who can supply the given item.

(ii) List all those items which cannot be supplied by given company.

SQL> create table item

```
( item_code number(4),  
  item_name varchar2(30), qty_in_stock number(5),  
  recorder_level number(5));
```

Table created.

SQL> insert into item values(&item_code, '&item_name', &qty_in_stock, &recorder_level);

Enter value for item_code: 10

Enter value for item_name: harddisk

Enter value for qty_in_stock: 35

Enter value for recorder_level: 25

old 1: insert into item values(&item_code, '&item_name', &qty_in_stock, &recorder_level)

new 1: insert into item values(10, 'harddisk', 35, 25)

1 row created.

SQL> /

Enter value for item_code: 20

Enter value for item_name: cpu

Enter value for qty_in_stock: 45

Enter value for recorder_level: 20

old 1: insert into item values(&item_code, '&item_name', &qty_in_stock, &recorder_level)

new 1: insert into item values(20, 'cpu', 45, 20)

1 row created.

SQL> select * from item;

ITEM_CODE	ITEM_NAME	QTY_IN_STOCK	RECORDER_LEVEL
-----------	-----------	--------------	----------------

10	harddisk	35	25
20	cpu	45	20
30	mouse	50	10
40	keyborad	85	25
50	ups	45	12

SQL> create table supplier

```
( supplier_code number(3),
  supplier_name varchar2(20),
  address varchar2(20));
```

Table created.

SQL> insert into supplier values(&supplier_code,&supplier_name,&address');

Enter value for supplier_code: 10

Enter value for supplier_name: infosys

Enter value for address: hyd

old 1: insert into supplier values(&supplier_code,&supplier_name,&address')

new 1: insert into supplier values(10,'infosys','hyd')

1 row created.

SQL> /

Enter value for supplier_code: 20

Enter value for supplier_name: wipro

Enter value for address: uppal

old 1: insert into supplier values(&supplier_code,&supplier_name,&address')

new 1: insert into supplier values(20,'wipro','uppal')

1 row created.

SQL> select * from supplier;

SUPPLIER_CODE	SUPPLIER_NAME	ADDRESS
-----	-----	-----
10	infosys	hyd
20	wipro	uppal
30	cmc	ameerpet
40	IBM	delhi
50	ctel	delhi

SQL> create table cansupply

(supplier_code number(4),
itemcode number(4));

Table created.

SQL> insert into cansupply values(&supplier_code,&itemcode);

Enter value for supplier_code: 10

Enter value for itemcode: 10

old 1: insert into cansupply values(&supplier_code,&itemcode)

new 1: insert into cansupply values(10,10)

1 row created.

SQL> /

Enter value for supplier_code: 100

Enter value for itemcode: 20

old 1: insert into cansupply values(&supplier_code,&itemcode)

new 1: insert into cansupply values(100,20)

1 row created.

SQL> select * from cansupply;

SUPPLIER_CODE	ITEMCODE
-----	-----
10	10
100	20
20	50
10	10
30	30

i) List all those suppliers who can supply the given item

SQL> select supplier_name from supplier where supplier_code in (select

2 supplier_code from cansupply

3 where itemcode=(select itemcode from item where item_name='Harddisk'));

no rows selected

ii) List all those items which cannot be supplied by given company.

SQL> select item_name

2 From item I, supplier s, cansupply c

3 Where i.item_code=c.itemcode and s.supplier_code=c.supplier_code and

4 Supplier_name not like '&ssupplier_name';

Enter value for ssupplier_name: wipro

old 4: Supplier_name not like '&ssupplier_name'

new 4: Supplier_name not like 'wipro'

ITEM_NAME

harddisk

harddisk

mouse

7. Create the following tables:

Student (roll-no, marks, category, district, state)

Student-rank(roll-no, marks, rank)

(i) List all those students who have come from Tamilnadu state and secured a rank above 100.

(ii) List all those students who come from Andhra Pradesh state and belong to given category who have secured a rank above 100.

SQL> create table student_s

(rollno integer primary key,

name varchar2(20),

category varchar2(20),

district varchar2(20),

state varchar2(20));

Table created.

SQL> insert into student_s values(&rollo,'&name','&category','&district','&state');

Enter value for rollo: 101

Enter value for name: swpna

Enter value for category: bc

Enter value for district: tamilnadu

Enter value for state: tamilnadu

old 1: insert into student_s values(&rollo,'&name','&category','&district','&state')

new 1: insert into student_s values(101,'swpna','bc','tamilnadu','tamilnadu')

1 row created.

SQL> /

Enter value for rollo: 102

Enter value for name: mahesh

Enter value for category: bc

Enter value for district: tamil

Enter value for state: tamilnadu

old 1: insert into student_s values(&rollo,&name,&category,&district,&state')

new 1: insert into student_s values(102,'mahesh','bc','tamil','tamilnadu')

1 row created.

SQL> select * from student_s;

ROLLNO	NAME	CATEGORY	DISTRICT	STATE
102	mahesh	bc	tamil	Tamilnadu
101	swpna	bc	tamilnadu	Tamilnadu

create table student_rank

(rollno integer,

marks integer,

rank integer,

constraint fk1 foreign key(rollno) references student_s(rollno);

Table created.

SQL> insert into student_rank values(&rollno,&marks,&rank);

Enter value for rollno: 501

Enter value for marks: 548

Enter value for rank: 500

old 1: insert into student_rank values(&rollno,&marks,&rank)

new 1: insert into student_rank values(501,548,500)

1 row created.

SQL> /

Enter value for rollno: 502

Enter value for marks: 365

Enter value for rank: 123

old 1: insert into student_rank values(&rollno,&marks,&rank)

new 1: insert into student_rank values(502,365,123)

1 row created.

SQL> select * from student_rank;

ROLLNO	MARKS	RANK
501	548	500
502	365	123

i) List all those students who come from Tamilnadu state and secured a rank above 100.

SQL> select name from student_s s, student_rank r

2 where s.rollno=r.rollno and rank>100 and state like 'tamilnadu';

NAME

swpna

Mahesh

ii) List all those students who come from Andhrapradesh and belong given category who have secured a rank above 100

SQL> select name

2 from student_s s, student_rank r

3 where s.rollno=r.rollno and rank>100 and state like 'ap'

4 and category like '&category'

5 ;

Enter value for category: bc

old 4: and category like '&category'

new 4: and category like 'bc'

NAME

usha

anusha

8. Create the following tables :

Branch (branch-id, branch-name, branch-city)

Customer (customer-id, customer-name, customer-city, branch-id)

SQL>create table branch(bid number(3),bname varchar2(20),bcity varchar2(20));

Table Created

SQL> insert into branch values(101,'ABC','Hyderabad');

1 row created

SQL> insert into branch values(102,'DEF','Adilabad');

1 row created

SQL> insert into branch values(103,'MNO','Warangal');

1 row created

SQL> insert into branch values(104,'PQR','Tirupathi');

1 row created

SQL> insert into branch values(105,'XYZ','Hyderabad');

1 row created

SQL>Select * from branch;

bid	bname	bcity
101	ABC	Hyderabad
102	DEF	Adilabad
103	MNO	Warangal
104	PQR	Tirupathi
105	XYZ	Hyderabad

SQL>create table customer(cid number(3),cname varchar2(20),
ccity varchar2(20),bid number(3));

SQL> insert into customer values(201,'A','Hyderabad',101);

1 row created

SQL> insert into customer values(202,'A','Hyderabad',101);

1 row created

SQL> insert into customer values(203,'C','Warangal',102);

1 row created

SQL> insert into customer values(204,'D','Warangal',103);

1 row created

SQL> insert into customer values(205,'E','Tirupathi',104);

1 row created

cid	cname	ccity	bid
201	A	Hyderabad	101
202	A	Hyderabad	101
203	C	Warangal	102
204	D	Warangal	103
205	E	Tirupathi	104

(a) Generate queries to do the following :

(i) List all those customers who live in the same city as the branch in which they have account.

SQL>select cname,ccity,bcity from branch b,customer c where b.bid=c.bid and bcity=ccity;

Cname	Ccity	Bcity
A	Hyderabad	Hyderabad
A	Hyderabad	Hyderabad
D	Warangal	Warangal
E	Tirupathi	Tirupathi

(ii)List all those customers who have an account in a given branch city.

SQL>select cname from branch b,customer c where b.bid=c.bid and b.bid=(select bid from branch where bname='ABC');

Cname
A
A

9. Create the following tables

Branch(branch_id,branch_name,branch_city)

Customer(customer_id, customer_name,customer_city,branch_id)

a) List all those all customers who live in the same city as the branch in which they have account.

SQL> create table branch(branch_id number(9)primary key,

branch_name varchar2(25),branch_city varchar2(15));

Table created.

SQL> insert into branch values(&branch_id,&branch_name,&branch_city);

Enter value for branch_id: 101

Enter value for branch_name: uppal

Enter value for branch_city: hyb

old 1: insert into branch values(&branch_id,&branch_name,&branch_city')

new 1: insert into branch values(101,'uppal','hyb')

1 row created.

SQL> /

Enter value for branch_id: 102

Enter value for branch_name: ramnagar

Enter value for branch_city: hyb

old 1: insert into branch values(&branch_id,&branch_name,&branch_city')

new 1: insert into branch values(102,'ramnagar','hyb')

1 row created.

SQL> select * from branch;

BRANCH_ID	BRANCH_NAME	BRANCH_CITY
-----	-----	-----
101	uppal	hyb
102	ramnagar	hyb
103	srinagar	vshp
104	ramnagar	hyb
105	kota	vzm

SQL> create table customer(customer_id number(7)primary key, customer_name
varchar2(25),customer_city varchar2(15),branch_id number(9),foreign
key(branch_id)references branch on delete cascade);

Table created.

SQL> insert into customer
values(&customer_id,&customer_name,&customer_city,&branch_id);

Enter value for customer_id: 4586

Enter value for customer_name: john

Enter value for customer_city: hyb

Enter value for branch_id: 101

old 1: insert into customer
values(&customer_id,&customer_name,&customer_city,&branch_id)

new 1: insert into customer values(4586,'john','hyb',101)

1 row created.

SQL> /

Enter value for customer_id: 789

Enter value for customer_name: jiya

Enter value for customer_city: hyb

Enter value for branch_id: 102

old 1: insert into customer
values(&customer_id,&customer_name,&customer_city,&branch_id)

new 1: insert into customer values(789,'jiya','hyb',102)

1 row created.

SQL> select * from customer;

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_CITY	BRANCH_ID
4586	john	hyb	101
789	jiya	hyb	102
987	alex	hyb	103
485	riya	vskp	104
956	ram	hyb	105

a) List all those all customers who live in the same city as the branch in which they have account.

SQL> select customer_name from branch b, customer c

2 where b.branch_id=c.branch_id and b.branch_city=customer_city;

CUSTOMER_NAME

john

riya

10. Create the following tables :

Student(roll-no, name, date-of-birth, course-id)

Course (Course-id, name, fee, duration)

(i) List all those students who are between 18-19 years of age and have opted for MCA course.

(ii) List all those courses in which number of students are less than 10.

SQL> create table student10

(roll_no number(5),

name varchar2(7),

date_of_birth date,

course_id number(7) primary key,foreign key(course_id) references course1);

SQL> insert into student10 values(&roll_no,&name','&date_of_birth',&course_id);

Enter value for roll_no: 1

Enter value for name: jiya

Enter value for date_of_birth: 12-jan-1989

Enter value for course_id: 1

old 1: insert into student10 values(&roll_no,&name','&date_of_birth',&course_id)

new 1: insert into student10 values(1,'jiya','12-jan-1989',1)

1 row created.

SQL> /

Enter value for roll_no: 13

Enter value for name: riya

Enter value for date_of_birth: 4-sep-1990

Enter value for course_id: 2

old 1: insert into student10 values(&roll_no,&name','&date_of_birth',&course_id)

new 1: insert into student10 values(13,'riya','4-sep-1990',2)

1 row created.

SQL> select * from student10;

ROLL_NO	NAME	DATE_OF_B	COURSE_ID
1	jiya	12-JAN-89	1
13	riya	04-SEP-90	2
50	john	18-APR-89	3
10	yak	03-MAR-88	4

Sql>create table course1

(course_id number(2)primary key,

name varchar2(5),

fee number(7),

duriaton number(7));

SQL> insert into course1 values(&course_id,&name,&fee,&duration);

Enter value for course_id: 1

Enter value for name: mca

Enter value for fee: 2000

Enter value for duration: 3

old 1: insert into course1 values(&course_id,&name,&fee,&duration)

new 1: insert into course1 values(1,'mca',2000,3)

1 row created.

SQL> /

Enter value for course_id: 2

Enter value for name: mba

Enter value for fee: 2500

Enter value for duration: 2

old 1: insert into course1 values(&course_id,&name,&fee,&duration)

new 1: insert into course1 values(2,'mba',2500,2)

1 row created.

SQL> select * from course1;

COURSE_ID	NAME	FEE	DURIATON
-----	-----	-----	-----
1	mca	2000	3
2	mba	2500	2
3	msc	4500	2
4	bsc	6500	3
5	bcom	7500	2

- i) List of all those who are between 18-19 years of age and have opted for MCA course**

SQL> select s.name

From student10 s, course1 c

Where(s.course_id=c.course_id and 18<(to_char(sysdate,'yy')

To_char(date_of_birth,'yy') and c.name like 'mca');

No rows selected

- ii) List all those courses in which number of students are less than 10**

SQL> select c.name from course1 c

where 10>(select count (roll_no) from Student10 s where c.course_id=s.course_id);

NAME

mca

mba

msc

bsc

bcom

11. Create the following tables :

Student (roll-no, name, subject-opts)

Subject –rank (subject-code, subject-name, faculty-code)

Faculty (faculty-code, faculty-name, specialization)

i) Find the number of students who have enrolled for the subject "DBMS"

ii) Find all those subjects which are not offered by any faculty members.

SQL> create table student3

```
(  
2 rollno integer,  
3 name varchar2(30),  
4 sub_opts integer);
```

SQL> insert into student3 values(&rollno,&name,&sub_opts);

Enter value for rollno: 101

Enter value for name: john

Enter value for sub_opts: 201

old 2: values(&rollno,&name,&sub_opts)

new 2: values(101,'john',201)

1 row created.

SQL> /

Enter value for rollno: 102

Enter value for name: ram

Enter value for sub_opts: 202

old 2: values(&rollno,&name,&sub_opts)

new 2: values(102,'ram',202)

1 row created.

SQL> select * from student3;

ROLLNO	NAME	SUB_OPTED
-----	-----	-----
101	john	201
102	ram	202
103	jiya	203
104	riya	204
105	mahesh	205

sql> create table faculty(faculty_code number(4) primary key,

2 faculty_name varchar2(15),

3 specialization varchar2(10));

table created.

sql> insert into faculty values(&faculty_code,&faculty_name,&specialization');

enter value for faculty_code: 1111

enter value for faculty_name: john

enter value for specialization: dbms

old 1: insert into faculty values(&faculty_code,&faculty_name,&specialization')

new 1: insert into faculty values(1111,'john','dbms')

1 row created.

sql> /

enter value for faculty_code: 2222

enter value for faculty_name: jiya

enter value for specialization: java

old 1: insert into faculty values(&faculty_code,&faculty_name,&specialization')

new 1: insert into faculty values(2222,'jiya','java')

1 row created.

sql> select * from faculty;

faculty_code	faculty_name	specialization
1111	john	dbms
2222	jiya	java
3333	ram	net
4444	mahesh	dbms

sql> create table subject_rank(subject_code number(4)

primary key,

2 subject_name varchar2(10),

3 faculty_code number(4)

3 references faculty(faculty_code));

Table created

SQL> insert into subject_rank values (&subject_code,&subect_name,&faculty_code);

Enter value for subject_code: 203

Enter value for subect_name: dbms

Enter value for faculty_code: 1111

old 1: insert into subject_rank values(&subject_code,&subect_name,&faculty_code)

new 1: insert into subject_rank values(203,'dbms',1111)

1 row created.

SQL> /

Enter value for subject_code: 204

Enter value for subect_name: daa

Enter value for faculty_code: 2222

old 1: insert into subject_rank values(&subject_code,&subect_name,&faculty_code)

new 1: insert into subject_rank values(204,'daa',2222)

1 row created.

SQL> /

Enter value for subject_code: 205

Enter value for subect_name: dbms

Enter value for faculty_code: 3333

old 1: insert into subject_rank values(&subject_code,&subect_name,&faculty_code)

new 1: insert into subject_rank values(205,'dbms',3333)

1 row created.

SQL> select * from subject_rank;

SUBJECT_CODE	SUBJECT_NAME	FACULTY_CODE
-----	-----	-----
203	dbms	1111
204	daa	2222
205	dbms	3333

i) Find the no.of students who have enrolled for the subject "DBMS".

SQL> select name from student3,subject_rank where sub_opted=subject_code and subject_name like 'dbms';

name

jiya

ii) Find all those faculty members who have not offered any subject.

SQL> select subject_name from subject_rank where subject_code not

2 in (Select subject_code from faculty);

no rows selected

12. Create the following tables :

Student (roll-no, name, subject-opted)

Subject –rank (subject-code, subject-name, faculty-code)

Faculty (faculty-code, faculty-name, specialization)

Find the number of students who have enrolled for the subject "OS"

(i) Find all those students who opted for more than 5 subjects.

```
SQL>create table subject_rank(
```

```
2 sub_code integer,
```

```
3 sub_name varchar2(20),
```

```
4 faculty_code integer);
```

```
SQL> insert into subject_rank values(&subject_code,&subect_name,&faculty_code);
```

```
Enter value for subject_code: 203
```

```
Enter value for subect_name: dbms
```

```
Enter value for faculty_code: 1111
```

```
old 1: insert into subject_rank values(&subject_code,&subect_name,&faculty_code)
```

```
new 1: insert into subject_rank values(203,'dbms',1111)
```

```
1 row created.
```

```
SQL> /
```

```
Enter value for subject_code: 204
```

```
Enter value for subect_name: daa
```

```
Enter value for faculty_code: 2222
```

```
old 1: insert into subject_rank values(&subject_code,&subect_name,&faculty_code)
```

```
new 1: insert into subject_rank values(204,'daa',2222)
```

```
1 row created.
```

```
SQL> select * from subject_rank;
```

SUBJECT_CODE	SUBJECT_NAME	FACULTY_CODE
-----	-----	-----
203	dbms	1111
204	daa	2222
205	dbms	3333

```
Sql> create table faculty(faculty_code number(4) primary key,  
2 faculty_name varchar2(15),specialization varchar2(10));
```

Table created.

```
SQL> insert into faculty
```

```
2 values(&faculty_code,&faculty_name,&specilization');
```

Enter value for faculty_code: 601

Enter value for faculty_name: abc

Enter value for specilization: daa

old 2: values(&faculty_code,&faculty_name,&specilization')

new 2: values(601,'abc','daa')

1 row created.

```
SQL> /
```

Enter value for faculty_code: 602

Enter value for faculty_name: or

Enter value for specilization: or

old 2: values(&faculty_code,&faculty_name,&specilization')

new 2: values(602,'or','or')

1 row created.

SQL> select * from faculty;

FACULTY_CODE	FACULTY_NAME	SPECIALIZATION
-----	-----	-----
601	abc	daa
602	or	or
603	pqr	os
405	dbms	dbms

SQL> create table student3(

2 rollno integer,

3 name varchar2(30),

4 sub_opted integer);

SQL> insert into student3

2 values(&rollno,&name,&sub_opted);

Enter value for rollno: 101

Enter value for name: john

Enter value for sub_opted: 201

old 2: values(&rollno,&name,&sub_opted)

new 2: values(101,'john',201)

1 row created.

SQL> /

Enter value for rollno: 102

Enter value for name: ram

Enter value for sub_opted: 202

old 2: values(&rollno,&name,&sub_opted)

new 2: values(102,'ram',202)

1 row created.

SQL> select * from student3;

ROLLNO	NAME	SUB_OPTED
-----	-----	-----
101	john	201
102	ram	202
103	jiya	203
104	riya	204
105	mahesh	205

i) Find the no of students who have enrolled for this subject"os"

SQL> select name from student3,subject_rank

2 where sub_opted=subject_code

3 and subject_name like 'os';

no rows selected

ii)Find all those student who opted for more than five subjects.

SQL> select name from student3,subject_rank

2 Where sub_opted>5;

NAME

john

john

john

ram

ram

ram

jiya

jiya

jiya

riya

riya

13. Create a table to represent sb-account of a bank consisting of account-no, customer-name, balance-amount .

Write a PL/SQL block to implement deposit and withdraw. Withdraw should not be allowed if the balance goes below RS.100.

```
SQL> create table sb_account(  
2  acno integer primary key,  
3  custname varchar2(20),  
4  bal integer);
```

Table created.

```
SQL> insert into sb_account values(&acno,'&custname',&bal);
```

Enter value for acno: 201

Enter value for custname: john

Enter value for bal: 7500

```
old 2:  values(&acno,'&custname',&bal)
```

```
new 2:  values(201,'john',7500)
```

1 row created.

```
SQL> /
```

Enter value for acno: 202

Enter value for custname: ram

Enter value for bal: 6500

```
old 2:  values(&acno,'&custname',&bal)
```

```
new 2:  values(202,'ram',6500)
```

1 row created.

```
SQL> /
```

Enter value for acno: 203

Enter value for custname: swapna

Enter value for bal: 8500

old 2: values(&acno,'&custname',&bal)

new 2: values(203,'swapna',8500)

1 row created.

SQL> /

Enter value for acno: 204

Enter value for custname: ramu

Enter value for bal: 9500

old 2: values(&acno,'&custname',&bal)

new 2: values(204,'ramu',9500)

1 row created.

SQL> select * from sb_account;

ACNO	CUSTNAME	BAL
201	john	7500
202	ram	6500
203	swapna	8500
204	ramu	9500

PL/SQL PROCEDURE FOR WITHDRAWS AN AMOUNT:

```
SQL> declare
2  balance sb_account.bal%type;
3  withdraw number:=&withdraw;
4  ano number:=&acno;
5  begin
6  select bal into balance from sb_account where
7  acno=ano;
8  dbms_output.put_line('balance=' || balance);
9  if(balance<1000) then
10 dbms_output.put_line('withdraw fails');
11 else
12 update sb_account set bal=bal-withdraw where
13 acno=ano;
14 end if;
15 end;
16 /
```

Enter value for withdraw: 1500

old 3: withdraw number:=&withdraw;

new 3: withdraw number:=1500;

Enter value for acno: 201

old 4: ano number:=&acno;

new 4: ano number:=201;

PL/SQL procedure successfully completed.

14. Create the following tables :

Item (item-code, item-name, qty-in-stock, reorder-level)

Supplier (supplier-code, supplier-name, address, status)

Can-supply(supplier-code, item-code)

(a) Write PL/SQL procedure to do the following :

Generate a report of those items that are supplied by those suppliers whose status is "important".

Sql> create table item1

```
2 (  
3 item_code number(4),  
4 item_name varchar2(30),qty_in_stock number(5),  
5 recorder_level number(5));
```

Table created.

SQL> insert into item1 values(&item_code,&item_name,&qty_in_stock,&recorder_level);

Enter value for item_code: 10

Enter value for item_name: mouse

Enter value for qty_in_stock: 45

Enter value for recorder_level: 10

old 1: insert into item1 values(&item_code,&item_name,&qty_in_stock,&recorder_level)

new 1: insert into item1 values(10,'mouse',45,10)

1 row created.

SQL> /

Enter value for item_code: 20

Enter value for item_name: keyboard

Enter value for qty_in_stock: 85

Enter value for recorder_level: 25

old 1: insert into item1 values(&item_code,&item_name,&qty_in_stock,&recorder_level)

new 1: insert into item1 values(20,'keyboard',85,25)

1 row created.

SQL> select * from item1;

ITEM_CODE	ITEM_NAME	QTY_IN_STOCK	RECORDER_LEVEL
10	mouse	45	10
20	keyboard	85	25
30	mouse	45	25
40	ups	65	50
50	cpu	40	10
60	cpu	10	10
70	mouse	20	20

sql> create table cansupply

2 (

3 supplier_code number(4),

4 itemcode number(4));

SQL> insert into supplier values(&supplier_code,&supplier_name,&address');

Enter value for supplier_code: 1

Enter value for supplier_name: john

Enter value for address: ramanthapur

old 1: insert into supplier values(&supplier_code,&supplier_name,&address')

new 1: insert into supplier values(1,'john','ramanthapur')

1 row created.

SQL> /

Enter value for supplier_code: 2

Enter value for supplier_name: ram

Enter value for address: uppal

old 1: insert into supplier values(&supplier_code,&supplier_name,&address')

new 1: insert into supplier values(2,'ram','uppal')

1 row created.

```
SQL> select * from supplier;
```

SUPPLIER_CODE	SUPPLIER_NAME	ADDRESS
-----	-----	-----
10	john	ramanthapur
20	ram	uppal
30	jiya	miyapur
40	riya	rayagiri
50	swapna	golnaka

```
SQL> declare
```

```
2  item2 item1.item_name%type;
3  cursor item_report is select item_name from item1
4  where qty_in_stock<=recorder_level;
5  begin
6  open item_report;
7  loop
8  fetch item_report into item2;
9  exit when item_report%notfound;
10 dbms_output.put_line('item:' || item2);
11 end loop;
12 close item_report;
13 end;
14 /
```

```
item:cpu
```

```
item:mouse
```

PL/SQL procedure successfully completed.

15. Create the following tables for Library Information System :

Book : (accession-no, title, publisher, author, status)

Status could be issued, present in the library, sent for binding, and cannot be issued.

**Write a trigger which sets the status of a book to "cannot be issued",
if it is published 20 years back**

```
sql> create table book(acno integer primary key,
```

```
2 title varchar2(30),
```

```
3 author varchar2(30),
```

```
4 status varchar(30),
```

```
5 dop date);
```

Table created.

```
sql> insert into book values(&acno,&title,&author,&status,&dop);
```

Enter value for acno: 101

Enter value for title: SE

Enter value for author: PANKAG

Enter value for status: PRESENT

Enter value for dop: 10-JAN-1990

```
old 2: values(&acno,&title,&author,&status,&dop')
```

```
new 2: values(101,'se','pankag','present','10-jan-1990')
```

1 row created.

```
SQL> /
```

Enter value for acno: 102

Enter value for title: DBMS

Enter value for author: RAMAKRISHNA

Enter value for status: PRESENT

Enter value for dop: 10-FEB-1990

old 2: values(&acno,&title,&author,&status,&dop')

new 2: values(102,'dbms','ramakrishna','present','10-feb-1990')

1 row created.

SQL> SELECT * FROM BOOK;

ACNO	TITLE	AUTHOR	STATUS	DOP
-----		-----	-----	-----
101	SE	PANKAG	PRESENT	10-JAN-90
102	DBMS	RAMAKRISHNA	PRESENT	10-FEB-90
103	DAA	MALLIKARJUNA	PRESENT	10-MAR-94
104	OS	GALVIN	PRESENT	12-APR-95
105	OR	SHRMA	PRESENT	15-APR-95

Write a trigger which sets the status of a book to "cannot be issued", if it is published 20 years back.

SQL> create or replace trigger update_book after insert on book

2 begin

3 update book set status='cannot be issued'

4 where(to_char(sysdate,'yyyy')-to_char(dop,'yyyy'))>=20;

5 end;

Trigger created.

16. Create the following tables :

a)Book(accession-no, title, publisher, year, date-of-purchase, status)

Member(member-id, name, number-of-books-issued, max-limit)

Book-issue(accession-no, member-id, date-of-issue)

SQL>create table book1(accession_no number(4) primary key,title varchar2(15),publisher varchar2(15),year number(4),dop date,status varchar2(20));

Table Created

SQL>insert into book values(101,'cobol','elite',2008,'10-aug-09','issued');

1 row created

SQL>insert into book values(102,'sucess','sakshi',2009,'22-jul-09','issued');

1 row created

SQL>insert into book values(103,'plsql,sql','bpb',2003,'14-nov-05','issued');

1 row created

SQL>insert into book values(104,'java','james',2002,'20-jun-02','issued');

1 row created

SQL>insert into book values(105,'c,c++','dennis',1991,'07-dec-94','not issued');

1 row created

select * from book;

accession_no	title	Publisher	year	dop	status
101	cobol	Elite	2008	10-aug-09	issued
102	sucess	Sakshi	2009	22-jul-09	issued
103	Plsql,sql	Bpb	2003	14-nov-05	issued
104	Java	James	2002	20-jun-02	issued
105	C,c++	Dennis	1991	07-dec-94	Not issued

SQL>create table member(member_id number(4) primarykey, name varchar2(15),no_of_books_issued number(2),max_limit number(2));

Table Created

SQL>insert into member values(1,'sandeep',2,2);

1 row created

```
SQL>insert into member values(2,'narsimha',1,2);
```

1 row created

```
SQL>insert into member values(3,'sai',2,3);
```

1 row created

```
SQL>insert into member values(4,'bang',3,3);
```

1 row created

```
SQL>insert into member values(5,'kiran',2,2);
```

1 row created

```
select * from member;
```

member_id	name	no_of_books_issued	max_limit
1	Sandeep	2	2
2	Narasimha	1	2
3	Sai	2	3
4	Bang	3	3
5	kiran	2	2

```
SQL>create table book_issue2(accession_no number(5), member_id number(4),  
date_of_issue date,foreign key(accession_no) references book2(accession_no),foreign  
key(member_id) references member2(member_id));
```

Table Created

```
SQL>insert into book_issue2 values(101,2,'04-sep-09');
```

1 row created

```
SQL>insert into book_issue2 values(102,5,'10-jul-11');
```

1 row created

```
SQL>insert into book_issue2 values(101,5,'10-jul-11');
```

1 row created

```
SQL>insert into book_issue2 values(105,3,'22-jan-12');
```

1 row created

```
SQL>insert into book_issue2 values(103,4,'15-feb-10');
```

1 row created

Select * from book_issue2;

accession_no	member_id	date_of_issue
101	2	04-sep-09
102	5	10-jul-11
101	5	10-jul-11
105	3	22-jan-12
103	4	15-feb-10

Write a PL/SQL procedure to issue the book.

Procedure:

```
sql> edit one.sql
```

```
create or replace procedure p11
```

```
is
```

```
ac_no book2.accession_no%type;
```

```
sta book2.status%type;
```

```
begin
```

```
    ac_no:='&ac_no';
```

```
    select status into sta from book2 where accession_no=ac_no;
```

```
    if(sta='not issued') then
```

```
update book2 set status='issued' where accession_no=ac_no;
```

```
    dbms_output.put_line('book issued');
```

```
    else
```

```
        dbms_output.put_line('already book has been issued');
```

```
    end if;
```

```
end;
```

```
sql>@one.sql;
```

```
enter the value for ac_no:105
```

```
procedure created.
```

```
Sql>exec p11;
```

```
Book issued
```

```
Pl/sql procedure successfully completed.
```

Sql> select * from book;

accession_no	title	Publisher	year	dop	status
101	cobol	Elite	2008	10-aug-09	issued
102	sucess	Sakshi	2009	22-jul-09	issued
103	Plsql,sql	Bpb	2003	14-nov-05	issued
104	Java	James	2002	20-jun-02	issued
105	C,c++	Dennis	1991	07-dec-94	issued

(ii)Write a trigger to set the status of students to "back listed" if they have taken book but not returned even after one year.

Trigger:

sql>edit tri2.sql;

```
create or replace trigger t1
before insert or update on book_issue2
for each row
begin
if(:new.date_of_issue<'01-jan-07')then
update book2 set status='back listed';
end if;
end;
```

sql>@tri2.sql;

trigger created.

Sql> select * from book2;

accession_no	title	Publisher	year	dop	status
101	cobol	Elite	2008	10-aug-09	Back listed
102	sucess	Sakshi	2009	22-jul-09	Back listed
103	Plsql,sql	Bpb	2003	14-nov-05	Back listed
104	Java	James	2002	20-jun-02	Back listed
105	C,c++	Dennis	1991	07-dec-94	Back listed

17. Write a Program to implement a FUNCTION

SQL>ed function.sql

Create or replace function fname

return number

is

var number(3);

begin

select count(*) into var from emp;

return var;

end;

/

OUTPUT:

ALT+F4

SQL>@ function.sql

Function Created

SQL>select fname feom emp;

FNAME

14

18. Write a Program to implement a CURSOR

```
SQL>ed cursor.sql;
```

```
Declare
```

```
eid emp.empno%type;
```

```
ename emp.ename%type;
```

```
cursor c is
```

```
select empno,ename into eid,ename from emp;
```

```
begin
```

```
open c;
```

```
loop
```

```
fetch c into eid,ename;
```

```
dbms_output.put_line('Employee id of' || ename || ' is ' || eid);
```

```
exit when c%NOTFOUND;
```

```
end loop;
```

```
end;
```

```
/
```

OUTPUT:

```
SQL>set serveroutput on
```

```
SQL> @ cursor.sql;
```

```
Employee id of SMITH is 7369
```

```
Employee id of ALLEN is 7499
```

```
PL/SQL procedure successfully created.
```

19. Write a PL/SQL code to find the Largest of Three Numbers

```
declare
a number:=10;
b number:=12;
c number:=5;
begin
dbms_output.put_line('a='||a||' b='||b||' c='||c);
if a>b AND a>c
then
dbms_output.put_line('a is greatest');
else
if b>a AND b>c
then
dbms_output.put_line('b is greatest');
else
dbms_output.put_line('c is greatest');
end if;
end if;
end;
/
```

Output

```
a=10 b=12 c=5
b is greatest
```


20. Write a PL/SQL code to find the Factorial of a given Number

Declare

n number;

fac number:=1;

i number;

begin

n:=&n;

for i in 1..n

loop

fac:=fac*i;

end loop;

dbms_output.put_line('factorial=' || fac);

end;

/

Output

Enter value for n: 10

old 7: n:=&n;

new 7: n:=10;

factorial=3628800

21. Write a PL/SQL Program for Palindrome Number

```
declare
    n number;
    m number;
    rev number:=0;
    r number;
begin
    n:=12321;
    m:=n;
    while n>0
    loop
        r:=mod(n,10);
        rev:=(rev*10)+r;
        n:=trunc(n/10);
    end loop;

    if m=rev
    then
        dbms_output.put_line('number is palindrome');
    else
        dbms_output.put_line('number is not palindrome');
    end if;
end;
/
```

Output

number is palindrome

22. Write a PL/SQL Program for Reverse of a Number

```
declare
n number;
i number;
rev number:=0;
r number;

begin
n:=&n;
while n>0
loop
r:=mod(n,10);
rev:=(rev*10)+r;
n:=trunc(n/10);
end loop;

dbms_output.put_line('reverse is ' || rev);

end;
/
```

Output

```
Enter value for n: 4578
old 8: n:=&n;
new 8: n:=4578;
reverse is 8754
```

23. Write a PL/SQL Program to Reverse a String

```
declare
str1 varchar2(50):='&str';
str2 varchar2(50);
len number;
i number;

begin
len:=length(str1);
for i in reverse 1..len
loop
str2:=str2 || substr(str1,i,1);
end loop;
dbms_output.put_line('Reverse of String is:' || str2);
end;
/
```

Output

```
Enter value for str: hello world
old 2: str1 varchar2(50):='&str';
new 2: str1 varchar2(50):='hello world';
Reverse of String is:dlrow olleh
```

24. Write a PL/SQL code to display employee Details using cursors.

```
DECLARE
CURSOR C1 IS SELECT EMPNO,ENAME ,DEPTNO FROM EMP;
EMPNUM EMP.EMPNO%TYPE;
EMPNAME EMP.ENAME%TYPE;
DEPTNUM EMP.DEPTNO%TYPE;
BEGIN
OPEN C1;
LOOP
FETCH C1 INTO EMPNUM,EMPNAME,DEPTNUM;
if c1%notfound then
exit;
else
dbms_output.put_line(EMPNUM || ' ' || EMPNAME || ' ' || DEPTNUM);
end if;
END LOOP;
end;
```

EXPECTED OUTPUT:

PL/SQL Procedure successfully Created

25. Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values:

CUSTOMERS table:

ID	NAME	AGE	ADDRESS	SALARY
1	Alive	24	Khammam	2000
2	Bob	27	Kadappa	3000
3	Catri	25	Guntur	4000
4	Dena	28	Hyderabad	5000
5	Eeshwar	27	Kurnool	6000
6	Farooq	28	Nellur	7000

```
CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON customers
FOR EACH ROW
WHEN (NEW.ID > 0)
DECLARE
sal_diff number;
BEGIN
sal_diff := :NEW.salary - :OLD.salary;
dbms_output.put_line('Old salary: ' || :OLD.salary);
dbms_output.put_line('New salary: ' || :NEW.salary);
dbms_output.put_line('Salary difference: ' || sal_diff);
END;
/
Trigger created.
```

26. Write a PL / SQL program to check whether the given number is prime or not.

```
declare
num number;
i number:=1;
c number:=0;
begin
num:=&num;
for i in 1..num
loop
if((mod(num,i))=0)
then
c:=c+1;
end if;
end loop;
if(c>2)
then
dbms_output.put_line(num || ' not a prime');
else
dbms_output.put_line(num || ' is prime');
end if;
end;
/
```