SDM College of Engineering and Technology

Dhavalagiri, Dharwad-580 002. Karnataka State. India.

Email: principal@sdmcet.ac.in, cse.sdmcet@gmail.com

Ph: 0836-2447465/ 2448327 Fax: 0836-2464638 Website: sdmcet.ac.in

Department
of
COMPUTER SCIENCE AND ENGINEERING

LABORATORY REPORT

[22UCSL504 Database Management Systems Lab]

Odd Semester

Course Teacher: Dr. U.P.Kulkarni

2024- 2025

Submitted by
By

Mr. Darshan G Chavan
2SD22CS023
5th Semester A division

Table of Contents

Termwork-1:

Problem Statement:

Prepare Data model diagram for the given business scenario and prepare schema using appropriate SQL statements, Insert data to check/validate the following Integrity constraints.

1.Entity Integrity

2.Row-Integrity

3.Referential Integrity

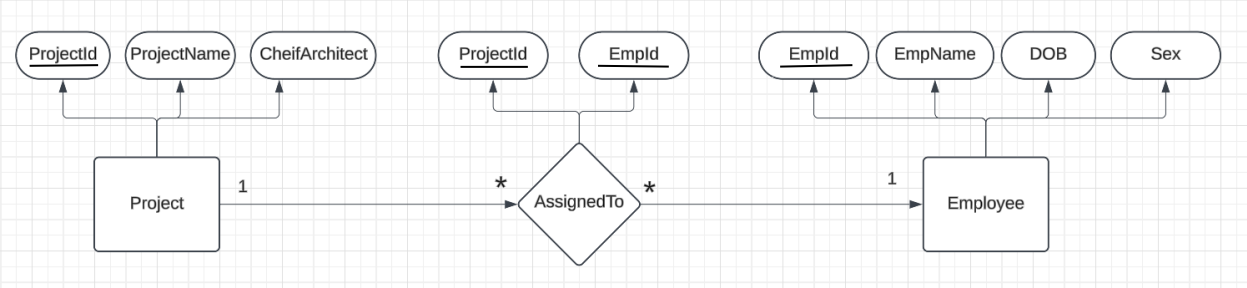
Business Scenario:

Employee (EmpNo#, EmpName, sex, phone, dob)

Project(projNo#, projectName, cheifarchitect)

Assigned_to (EmpNo#, projNo#)

ER Model :



Query and output:

SQL> create table employee (

empno integer not null

constraints EMPLOYEE_PK_VIOLATION

primary key,

empname char(25) not null,

sex char(1) not null

constraints EMPLOYEE_SEX_VIOLAIONcheck(sex in('m','f')),

phone integer null,

dob date default '15-apr-68' not null

);

Table created.

```
SQL> create table project (
projno integer not null,
projectname char(20) not null,
cheifarchitect char(20) default 'UPK' not null,
constraints PROJECT_PK_VIOLATION
primary key(projno)
);
Table created.

SQL> create table assigned_to (
empno integer not null,
projno integer not null,
constraints ASSIGNED_TO_PK_VIOLATION
primary key(empno,projno),
constraints ASSIGNED_TO_FK_EMP_VIOLATION
foreign key(empno)
references employee,
constraints ASSIGNED_TO_FK_PRJ_VIOLATION
foreign key(projno)
references project);
Table created.

SQL> insert into employee values(01,'darshan','m',78945,'07-sep-04');
1 row created.
SQL>  insert into employee values(02,'ankith','m',89445,'04-apr-04');
1 row created.
SQL>  insert into employee values(03,'prateek','m',675445,'23-jun-04');
1 row created.
SQL>  insert into employee values(04,'nivedita','f',82962,'08-jan-03');
1 row created.
```

```
SQL>  insert into employee values(05,'ganga','f',80452,'28-jun-94');
1 row created.
SQL>  insert into employee values(06,'priya','f',67452,'28-dec-04');
1 row created.


SQL> select * from employee;
Empno  empname          sex       phone    dob
1      darshan            m       78945  07-SEP-04
2      ankith             m       89445  04-APR-04
3      prateek            m       675445 23-JUN-04
4      nivedita           f       82962  08-JAN-03
5      ganga              f       80452  28-JUN-94
6      priya              f       67452  28-DEC-04
SQL> insert into project values(01,'DBMS','UPK');
1 row created.
SQL> insert into project values(02,'AIML','PK');
1 row created.
SQL> insert into project values(03,'Software','YS');
1 row created.

SQL> select * from project;
PROJNO PROJECTNAME  CHEIFARCHITECT
---------- -------------------- --------------------
1   DBMS                UPK
2   AIML                 PK
3   Software             YS
4   WebTechnology        IRM
SQL> insert into assigned_to values(1,1);
1 row created.
SQL> insert into assigned_to values(2,1);
```

1 row created.

SQL> insert into assigned_to values(4,2);

1 row created.

SQL> select * from assigned_to;

EMPNO PROJNO

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

1 1

2 1

4 2

3 1

2 3

2 2

2 4

1.Entity Integrity Checking

SQL> insert into employee values(01,'Aman','f',777,'07-sep-04',50000);

ERROR at line 1:

ORA-00001: unique constraint (SYSTEM.EMPLOYEE_PK_VIOLATION) violated

2.Row integrity Checking

SQL> insert into employee values(01,'Aman','H',777,'07-sep-04',50000);

ERROR at line 1:

ORA-02290: check constraint (SYSTEM.EMPLOYEE_SEX_VIOLAION) violated

3.Referential integrity checking

SQL> insert into assigned_to values(20,5);

ERROR at line 1:

ORA-02291:Integrityconstraint (SYSTEM.ASSIGNED_TO_FK_EMP_VIOLATION) violated-parent key not found.

Learning Outcome:

1. Learning about Database Design models **s**uch as Entity Relationship diagram and Database Schema and creating tables .
2. Understanding about various integrity constraints.

References:

1.Elmasri & Navathe, “Fundamentals of Database Systems”, 6/E, Addison-Wesley, 2012.

2.https://www.geeksforgeeks.org/dbms-integrity-constraints

Termwork-2:

Problem Statement:

a) SQL statement to obtain the empID of all employees working on Project = 1.
b) WSQL to get the details of employee working on project 1.
c) WSQL to get the details of the employee who is working on 'DBMS' project
d) WSQL to get employee details of employee working on both project 1 and 2.
e) WSQL to get the details of the employee who is working on either project 1 or 2.

Sample input and output:

a)

```
SQL>select e.empno from employee e,assigned_to at

where e.empno = at.empno

and projno =1;


EMPNO

----------

1

2

3
```

b)

```
SQL>select e.* from employee e , assigned_to at

where e.empno=at.empno

and projno=1;


EMPNO EMPNAME        S      PHONE DOB
```

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

1 darshan m 78945 07-SEP-04

2 ankith m 89445 04-APR-04

3 prateek m 675445 23-JUN-04

c)

```
SQL> select e.* from employee e,assigned_to at, project pr
where e.empno=at.empno
and pr.projno=at.projno
and projectname='DBMS';
```

EMPNO EMPNAME S PHONE DOB

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

1 darshan m 78945 07-SEP-04

2 ankith m 89445 04-APR-04

3 prateek m 675445 23-JUN-04

d)

```
SQL> select e.* from employee e, assigned_to at
where e.empno=at.empno
and projno=1
intersect
select e.* from employee e, assigned_to at
where e.empno=at.empno
and projno=2;
```

EMPNO EMPNAME S PHONE DOB

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

2 ankith m 89445 04-APR-04

e)

```
SQL> select e.* from employee e,assigned_to at
where e.empno=at.empno
```

and (projno=1 OR projno=2);

Learning Outcome:

1. Understanding about query formation to get required output.
2. Understanding about WHERE and INTERSECT clauses.
3. Understanding AND and OR operators.

References:

1. Elmasri & Navathe, "Fundamentals of Database Systems", 6/E, Addison-Wesley, 2012.
2. https://www.geeksforgeeks.org/sql-intersect-clause

Termwork-3:

Problem Statement:

Modify the schema to store the information about fine to be paid by employees

Query and output:

```
create table employee_fine (

empNo int not null,

fine Number ,

constraints VIOLATION_PK_CONSTRAINTS

primary key(empNo,fine),

foreign key(empNo)

references employee

);
```

Table created.

Learning Outcome:

a) Understanding about the multivalued attributes.

References:

1. Elmasri & Navathe, "Fundamentals of Database Systems", 6/E, Addison-Wesley, 2012
2. https://www.geeksforgeeks.org

Termwork-4:

Problem Statement:

Modify the schema to store the details of dependents for all employees if exixts.

a) List all employees who have 2 dependents

b) List all the employees who have their mother as dependent

Query and output:

```
create table dependents(
empNo int not null,
DepName char(45) not null,
DepRelation char(45) not null,
constraints EMP_NO_FK_VIOLATION
foreign key(empNo)
references employee,
constraints DEPENDENT_PK_VIOLATION
primary key(empNo,DepName)
);
```

Table created.

a)

```
select e.empName
from employee e
Join dependents d on e.empNo = d.empNo
group by e.empNo,e.empName
having count(d.empNo) = 2;
```

```
EMPNAME
------------------------
darshan
```

b)

```
select e.empName from employee e,dependents d
where e.empNo = d.empNo
and d.deprelation='manager';
```

```
EMPNAME
------------------------
```

darshan

Learning Outcomes:

1. Learn how to design a relational database schema to handle one-to-many relationships.
2. Understand how to establish relationships between tables using foreign keys

References:

1. Elmasri & Navathe, “Fundamentals of Database Systems”, 6/E, Addison-Wesley, 2012
2. https://www.geeksforgeeks.org

Termwork-5:

Problem Statement:

Study the following

1.Deleting rows.

2.Deleting table.

3.Updating table.

Query and output:

1.

SQL> delete from student where rollno = 3;

1 row deleted.

2.

SQL> Update student

set name='Rekha'

where rollno=1;

1 row updated.

3.

SQL> drop table student;

Table dropped.

Learning Outcome:

1.Understanding about the deleting and updating the table

References:

1. Elmasri & Navathe, “Fundamentals of Database Systems”, 6/E, Addison-Wesley, 2012
2. https://www.geeksforgeeks.org

Termwork-6:

Problem Statement:

Study the impact of deleting rows and dropping table on foreign key or referential integrity.

Query and output:

SQL> delete from employee where empNo=1;

ERROR at line 1:

ORA-02292: integrity constraint (SYSTEM.ASSIGNED_TO_FK_EMP_VIOLATION) violated

- child record found

SQL> drop table employee;

ERROR at line 1:

ORA-02449: unique/primary keys in table referenced by foreign keys

Learning Outcome:

1. Understand Referential Integrity and foreign key
2. Understanding the impact of deleting a row or dropping a table.

References:

1. Elmasri & Navathe, “Fundamentals of Database Systems”, 6/E, Addison-Wesley, 2012
2. https://www.geeksforgeeks.org

Termwork-7:

Problem Statement:
WSQL statement to display name and DOB of all employees who are on the bench

Query and output:
select e.empName from employee e

MINUS

select e.empName from employee e,assigned_to at

where e.empNo=at.empNo

EMPNAME

---

priya

Learning Outcome:
1. Learning bout the MINUS operator.

References:

1. Elmasri & Navathe, "Fundamentals of Database Systems", 6/E, Addison-Wesley, 2012
2. https://www.geeksforgeeks.org

Termwork-8:

Problem Statement:
WSQL statement to display the name of all employees working on all projects

Query and output:
select e.empName from employee e

join assigned_to at on e.empNo=at.empNo

group by e.empNo,e.empName

having count(at.projNo)=(select count(*) from project);

EMPNAME

---

ankith

Learning Outcome:
1. Understanding about JOIN and GROUP BY clauses.

2. Learning about count() function.

References:

1. Elmasri & Navathe, “Fundamentals of Database Systems”, 6/E, Addison-Wesley, 2012
2. https://www.geeksforgeeks.org

Termwork -09:

Problem Statement:

Display name of all eployees working on atleast all of the project that employee 1 is working

Query and output:

```
select empName from employee where empNo in (

select empNo from assigned_to where projno in (

select projNo from assigned_to where empNo=1));
```

```
EMPNAME

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

darshan

ankith

prateek
```

Learning Outcome:

1. Understanding about the IN clause.

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-10:

Problem Statement:

Display the details of senior most 3 employees.

Query and output:

```
SELECT *

FROM (SELECT * FROM employee ORDER BY dob)

WHERE ROWNUM <= 3;
```

EMPNO	EMPNAME	S	PHONE	DOB
5	ganga	f	80452	28-JUN-94
4	nivedita	f	82962	08-JAN-03
2	ankith	m	89445	04-APR-04

Learning Outcome:

1. Learning about the ORDER BY and ROWNUM clause

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.
2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-11:

Problem Statement:

Find for each employee is penalty incurred

Query and output:

```
select e.empNo,sum(ef.fine)

from employee e

join employee_fine ef on e.empNo = ef.empNo

group by ef.empNo;
```

EMPNO	SUM(EF.FINE)
1	1000
2	500
3	200

Learning Outcome:

a) Understanding about the sum() function

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-12:

Problem Statement:

Display the no of employees working under each project having count>3

Query and output:

```
select COUNT(a.empNo)

FROM assigned_to a

JOIN  project p ON p.projNo = a.projNo

GROUP BY p.projNo, p.projectName

HAVING COUNT(a.empNo) >= 3;
```

```
COUNT(A.EMPNO)

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

3
```

Learning Outcome:

1. Understanding about COUNT() function
2. Understanding about GROUP BY and HAVING clause.

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-13:

Problem Statement:

Study of a)Order By b)Alter clause.

Query and output:

a) SQL>select * from employee

order by dob asc;

b)

SQL>Alter table employee

add email char(30);

Table altered.

EMPNO	EMPNAME	S	PHONE	DOB
5	ganga	f	80452	28-JUN-94
4	nivedita	f	82962	08-JAN-03
2	ankith	m	89445	04-APR-04
3	prateek	m	675445	23-JUN-04
1	darshan	m	78945	07-SEP-04
6	priya	f	67452	28-DEC-04

6 rows selected.

SQL> select * from employee;

EMPNO	EMPNAME	S	PHONE	DOB	EMAIL
1	darshan	m	78945	07-SEP-04	
2	ankith	m	89445	04-APR-04	
3	prateek	m	675445	23-JUN-04	
4	nivedita	f	82962	08-JAN-03	
5	ganga	f	80452	28-JUN-94	
6	priya	f	67452	28-DEC-04	

Learning Outcome:

1) Understanding the ORDER BY clause.
2) Learning about ALTER

References:

1) Elmasri & Navathe, “Fundamentals of Database Systems”, 6/E, Addison-Wesley, 2012
2) https://www.geeksforgeeks.org

Termwork-14:

Problem Statement:

T14. Study of statistical functions

Query and output:

a) Min()

SQL>select Min(salary) from employee;

MIN(SALARY)

14500

SQL> select Max(salary) from employee;

b) Max()

MAX(SALARY)

55000

c) Sum()

SQL> select sum(salary) from employee;

SUM(SALARY)

187500

d) Avg()

SQL> select Avg(salary) from employee;

AVG(SALARY)

31250

e) variance()

SQL> select variance(salary) from employee;

VARIANCE(SALARY)

229975000

f)stddev()

SQL> select stddev(salary) from employee;

STDDEV(SALARY)

15164.9266

Learning Outcome:

1. Understading about different statistical functions in DBMS.

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork:15:

Problem Statement:

T15.Study of clauses

Query and output:

a) between

SQL> select * from employee

where salary between 10000 and 40000;

EMPNO	EMPNAME	S	PHONE	DOB	SALARY
1	darshan	m	78945	07-SEP-04	10000
2	ankith	m	89445	04-APR-04	30000
5	ganga	f	80452	28-JUN-94	16000
6	priya	f	67452	28-DEC-04	18000

b) LIKE

```
SQL>select *
from tenant
where name LIKE 'G%';

TID NAME                                       ADHARNO    CONTACT
---------- ---------------------------------------------- ---------- ----------
6 Gagan                                    1.2346E+11 9234567876
```

c)ANY

```
SQL> select projectName
from project
where projNo = ANY(
select projNo
from assigned_to
where empNo = 2 OR empNo=4);

PROJECTNAME
--------------------
DBMS
AIML
Software
web technology
```

d) IN

```
'SQL> select * from assigned_to
where ProjNo in 1;

EMPNO     PROJNO
---------- ----------
         1         1
         2         1
         3         1
```

e) EXISTS

```
SQL> select name
from tenant
where exists
(
select *
from holdProperty
where Tenant.TID = HoldProperty.TID);

NAME
--------------------------------------------
David
Virat
Rohit
Smriti

4 rows selected.
```

g)RowNum

```
SQL> select *
from employee
where rownum<=3;

EMPNO EMPNAME                S      PHONE DOB          SALARY
---------- ------------------------ - ---------- --------- ----------
1         darshan                  m        78945        07-SEP-04       10000
2         ankith                   m        89445        04-APR-04        6000
3         prateek                  m        675445        23-JUN-04       18000
```

h)count()

```
SQL> select count(empNo) from employee;
```

COUNT(EMPNO)

6

Learning outcome:

1) Understanding about different statistical functions in DBMS.

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-16:

Problem Statement:

T16.Study of date related functions

Query and output:

SQL>select * from employee where dob = '07-SEP-04';

EMPNO EMPNAME S PHONE DOB SALARY

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

1 darshan m 78945 07-SEP-04 10000

SQL> SELECT SYSDATE AS CurrentDateTime FROM dual;

CURRENTDA

05-NOV-24

SQL> SELECT ADD_MONTHS(SYSDATE, 3) AS DatePlus3Months FROM dual;

DATEPLUS3

05-FEB-25

SQL> SELECT NEXT_DAY(SYSDATE, 'FRIDAY') AS NextFriday FROM dual;

NEXTFRIDA

08-NOV-24

SQL> SELECT LAST_DAY(SYSDATE) AS EndOfMonth FROM dual;

ENDOFMONT

30-NOV-24

SQL> SELECT TRUNC(SYSDATE, 'MM') AS StartOfMonth FROM dual;

STARTOFMO

01-NOV-24

Learning outcome:

1) Understanding the date related functions in DBMS

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-17:

Problem Statement:

Study of views

Query and output:

a)with check option

create view emp1 as

select empNo , dob,salary

from employee

where dob>='1-Jan-68';

View created.

SQL> select * from emp1;

EMPNO	DOB	SALARY
1	07-SEP-04	10000
2	04-APR-04	18000
3	23-JUN-04	36000
4	08-JAN-03	46000
5	28-JUN-94	6000
6	28-DEC-04	26000

6 rows selected.

b)Without check option

```
SQL> create view emp2 as
select empNo , empName , sex, salary
from employee;
```

View created.

SQL> select * from emp2;

EMPNO	EMPNAME	S	SALARY
1	darshan	m	10000
2	ankith	m	18000
3	prateek	m	36000
4	nivedita	f	46000
5	ganga	f	6000
6	priya	f	26000

6 rows selected.

SQL> drop view emp1;

View dropped.

SQL> drop view emp2;

View dropped.

Learning outcome:

1) Understanding about the virtual table with and without check option in DBMS

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-18:

Problem Statement:

Study of a)Copying table b)Synonym

Query and output:

SQL> create table emp_details

AS select * from employee;

Table created.

SQL> select * from emp_details;

EMPNO	EMPNAME	S	PHONE	DOB	SALARY
1	darshan	m	78945	07-SEP-04	10000
2	ankith	m	89445	04-APR-04	18000
3	prateek	m	675445	23-JUN-04	36000
4	nivedita	f	82962	08-JAN-03	46000
5	ganga	f	80452	28-JUN-94	6000
6	priya	f	67452	28-DEC-04	26000

6 rows selected.

b)synonym

SQL> CREATE SYNONYM emp_syn FOR EMPLOYEE;

Synonym created.

SQL> select * from emp_syn;

EMPNO	EMPNAME	S	PHONE	DOB	SALARY
1	darshan	m	78945	07-SEP-04	10000
2	ankith	m	89445	04-APR-04	18000
3	prateek	m	675445	23-JUN-04	36000
4	nivedita	f	82962	08-JAN-03	46000
5	ganga	f	80452	28-JUN-94	6000
6	priya	f	67452	28-DEC-04	26000

6 rows selected.

Learning outcome:

1) Understanding table duplication and preserving the data integrity.
2) Learning about the synonyms and how synonym simplify access to database objects and hide schema details

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-19:

Problem Statement:

Study of PL/SQL features

Query and output:

SQL> declare

projectCount integer;

begin

insert into project values(10,'CN','IRU');

select count(projNo) into projectCount from project;

```
delete from project where projNo=10;
end;
/

PL/SQL procedure successfully completed.

SQL> SET SERVEROUTPUT ON;
SQL>DECLARE
projectCount INTEGER;
BEGIN
INSERT INTO project VALUES (6, 'CN', 'IRU');
SELECT COUNT(projNo) INTO projectCount FROM project;
DELETE FROM project WHERE projNo = 6;
DBMS_OUTPUT.PUT_LINE('No of projects = ' || projectCount);
END;/
No of projects = 5

PL/SQL procedure successfully completed.
SQL> declare
projectCount integer;
status char(20);
begin
insert into project values(10,'CN','IRU');
select count(projNo) into projectcount from project;
if projectCount>3 then
status := 'Very few projects';
else
status:='suffiecient project';
END if;
delete from project where projNo=10;
```

```
DBMS_output.put_line('No of projects = ' || projectCount || ' Status = ' || status);
end;
/
No of projects = 5 Status = Very few projects

PL/SQL procedure successfully completed.


SQL> declare
rowNo integer;
pName char(20);
projectCount integer;
begin
rowNo:=1;
select count(projNo) into projectCount from project;

while rowNo<=4 loop
select projectName into pName from project
where projNo=rowNo;
DBMS_output.put_line('Project Name = '||pName);
rowNo := rowNo+1;
End loop;
end;


Project Name = DBMS
Project Name = AIML
Project Name = Software
Project Name = WebTechnology

PL/SQL procedure successfully completed.
```

Learning outcome:

1) Learn how PL/SQL extends SQL with procedural features, allowing for complex operations and create reusable and organized code using procedures, functions

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-20:

Problem Statement:

Study of TRIGGERS

Query and output:

```
CREATE TRIGGER t3_salaryUpdated

BEFORE UPDATE ON employee

FOR EACH ROW

WHEN ((NEW.salary / OLD.salary) > 1.1)

BEGIN

INSERT INTO salaryUpdated VALUES (:NEW.empNo, :OLD.salary,:NEW.salary);

END;

/

Trigger created.

SQL> select * from salaryUpdated;


EMPNO  NEWSALARY

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

4      55000

CREATE OR REPLACE TRIGGER display_salary_changes

BEFORE DELETE OR INSERT OR UPDATE ON employee

FOR EACH ROW

WHEN (NEW.empNo > 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;
SQL> CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON employee
FOR EACH ROW
WHEN (NEW.empNo > 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.
SQL> update employee
set salary = 50000
where empNo=3;
Old salary: 40000
New salary: 50000
Salary difference: 10000
1 row updated.
SQL> select * from salaryUpdated;

EMPNO  OLDSALARY  NEWSALARY
---------- ---------- ----------
```

2 25000 30000

3 40000 50000

Learning outcome:

1) Learning the purpose of triggers as automated database event handlers and understand when and why to use them.

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-21:

Problem Statement:

Study of Stored Procedures

Query and output:

SQL>Create procedure addProjects(ProjectCode IN integer,ProjectTitle IN varchar2,manager varchar2)

AS

BEGIN

insert into project(projNo,ProjectName,cheifArchitect)

values (PRojectCode,ProjectTitle,manager);

END;

Procedure created.

SQL> execute addProjects(10,'Pervasive','UPK');

PL/SQL procedure successfully completed.

SQL> select * from project;

PROJNO PROJECTNAME CHEIFARCHITECT

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

1 DBMS UPK

2 AIML PK

3 Software YS

4 WebTechnology IRM

4 rows selected.

Learning outcome:

1) Learn the concept of stored procedures as SQL code that can be executed as a single unit.

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-22:

Problem Statement :

Study of stored function

Query and output:

```
CREATE OR REPLACE FUNCTION noOfProjects(eID IN assigned_to.empNO%TYPE)

RETURN INTEGER

IS

cnt INTEGER;

BEGIN

SELECT COUNT(projNo)

INTO cnt

FROM assigned_to

WHERE empNo = eID;

RETURN cnt;

END;
```

Function created.

```
declare

projectCount integer;

begin
```

```
projectCount := noOfProjects(1);

DBMS_output.put_line('Project count of employee 1 is :'||projectCount)

end;

/

Project count of employee 1 is :1
```

Learning outcome:

1) Learn the concept of stored functions as reusable SQL objects that return a single value or result after execution.

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-23:

Problem Statement:

Study of Cursors

Query and output:

```
declare

manager project.cheifArchitect%Type;

cursor proj is

select project.cheifArchitect from project where projNo<4;

BEGIN

OPEN proj;

FETCH proj into manager;

while proj%Found LOOP

DBMS_output.put_line('Chief is = '||manager);

FETCH proj into manager;

END LOOP;

CLOSE proj;

END;

/

Chief is = UPK

Chief is = PK
```

```
Chief is = YS


PL/SQL procedure successfully completed.

declare

manager project.cheifArchitect%Type;

projName project.projectName%Type;


cursor proj is

select projName , cheifArchitect from project where projNo<4;

BEGIN

OPEN proj;

FETCH proj into projName,manager;

while proj%Found LOOP

DBMS_OUTPUT.put_line('Name = '||projName||'Chief is = '||manager);

FETCH proj INTO projName,manager;

END LOOP;

CLOSE proj;

END;


Name = Chief is = UPK

Name = Chief is = PK

Name = Chief is = YS

PL/SQL procedure successfully completed.
```

Learning outcome:

1) Understanding cursors and their purpose in handling and processing multiple rows of query results one at a time.

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.

Termwork-24:

Problem Statement:

Study of savepoints,rollback and commit feature of SQL support for transaction with description

Query and output:

SQL> set autocommit off;

SQL> select * from student;

ROLLNO	NAME	PHONE
1	Ananya	777
2	Aditi	666
3	Gagan	333

SQL> update student set phone=111 where rollno=1;

1 row updated.

SQL> select * from student;

ROLLNO	NAME	PHONE
1	Ananya	111
2	Aditi	666
3	Gagan	333

SQL> rollback;

Rollback complete.

SQL> select * from student;

ROLLNO	NAME	PHONE
1	Ananya	777
2	Aditi	666

3 Gagan 333

SQL> commit;

Commit complete.

SQL> update student set phone=111 where rollno=1;

1 row updated.

SQL> savepoint s1;

Savepoint created.

SQL> select * from student;

ROLLNO	NAME	PHONE
1	Ananya	111
2	Aditi	666
3	Gagan	333

s

SQL> update student set phone=222 where rollno =1;

1 row updated.

SQL> select * from student;

ROLLNO	NAME	PHONE
1	Ananya	222
2	Aditi	666
3	Gagan	333

SQL> rollback to s1;

Rollback complete.

SQL> select * from student;

ROLLNO NAME PHONE

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

1 Ananya 111

2 Aditi 666

3 Gagan

Learning outcome:

1) Learn the concept of transactions and their role in maintaining data consistency and integrity in a multi-user database environment.

References:

1) Herbert Schildt, Java-The Complete Reference, 9th Edition, Tata McGraw Hill, 2014.

2) Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.