Mazhar Siddique

SQL Assignment

**Section 1 :**

**Table Creation:**

Create table department

(

department\_id number primary key, department\_name varchar(30), department \_block\_number varchar(30), department\_block\_name number);

)

create table student

(student\_id number primary key, student\_name varchar(30), address varchar (40), city varchar(30),department\_id number, constraint fk\_department\_id foreign key (department\_id) references department(department\_id) );

create table staff

(staff\_id number primary key, staff\_name varchar(30), department\_id number,

constraint fk\_staffDep\_id foreign key (department\_id) references department (department\_id) );

create table subject

(subject\_id number primary key , student\_name varchar (30), student\_code varchar(10), staff\_id number,

constraint fk\_staffId foreign key (staff\_id) references staff (staff\_id) )

create table mark

(value\_mark number, subject\_id number,student\_id number,

constraint pk\_marks primary key (subject\_id, student\_id),

constraint fk\_mark\_subjid foreign key (subject\_id) references subject(subject\_id),

constraint fk\_mark\_studid foreign key (student\_id) references student(student\_id));

6. Add a constraint by writing a query to add a not null constraint to the column staff\_name in the staff table.

Alter table staff

modify (staff\_id not null);

7. Alter table student

Add email varchar(20)

8. alter table student

modify (email varchar(50));

9. alter table student

drop column email;

**Section 2**

**Section 3**

12).update subject

set subject\_name='Computer Science',

subject\_code=1919

where subject\_code=1842;

13). delete from subject

where subject\_name = 'Accounting';

14). select department\_name

from department

order by department\_name asc;

15). select department\_name

from department

where department\_block\_number between 3 and 10;

16). select student\_name from

student

order by student\_name asc;

17). select student\_name from

student

where city = 'San Jose' or city= 'Chicago' or city= 'Taylor'

order by student\_name asc;

18). select address || city as address\_student

from student;

19).select student\_name

from student

where LENGTH(student\_name) >6;

**20).** select DEPARTMENT\_BLOCK\_NUMBER, count(department\_name)

from department group by DEPARTMENT\_BLOCK\_NUMBER

order by department\_block\_number;

21). select count(\*) as std\_count

from student;

22).select DEPARTMENT\_NAME, count(\*) AS student\_count

FROM department

JOIN student on student.DEPARTMENT\_ID = department.DEPARTMENT\_ID

GROUP BY department.department\_name

ORDER BY department\_name ASC ;

23). SELECT student\_name, subject\_name

from student

join mark on mark.STUDENT\_ID = student.STUDENT\_ID

join subject on subject.SUBJECT\_ID = mark.SUBJECT\_ID and subject.subject\_code > 1600

order by student\_name asc;

24). SELECT student\_name, subject\_name, mark.value\_mark

from student

join mark on student.STUDENT\_ID = mark.STUDENT\_ID

join subject on mark.SUBJECT\_ID = subject.subject\_id and mark.value\_mark <3;

25).

26). select staff\_name

FROM staff

WHERE staff.staff\_id NOT IN (select distinct subject.STAFF\_ID from subject)

ORDER BY STAFF\_NAME ASC;

**Section 10**

27). create or replace function find\_dept\_name(d\_id int)

return varchar2

is depName varchar2(30);

begin

select department\_name

into depName

from department

where department.department\_id = d\_id;

return (depName);

End find\_dept\_name;

28). create or replace function find\_dept\_block(dept\_id in int )

return number

is deptBlock number;

begin

select department\_block\_number

into deptBlock

from department

where dept\_id = department\_id;

return (deptBlock);

end find\_dept\_block;

29).create or replace function find\_staff\_name (m\_staffid number)

return varchar2

is staffName varchar2(30);

begin

select staff\_name

into staffName

from staff

where m\_staffid = staff\_id;

return staffName;

end find\_staff\_name;

**Section 11**

30). Create or replace trigger ‘trigger\_deparment\_af\_delete’

After update on DEPARTMENT

FOR EACH ROW

Begin

dbms\_output.put\_line(‘DEPARTMENTS table has been updated’);

end;

31). create or replace trigger ‘trigger\_department\_bf\_delete’

before delete on DEPARTMENT

FOR EACH ROW

begin

dbms\_output.put\_line(“A row has been deleted from DEPARTMENT');

end;

32).create index index\_Stud

on student(student\_name);

33). Create view v\_staffNames as (select staff\_name from staff);

**Section 13: Cursors**

36). DECLARE

stud\_row student%ROWTYPE;

CURSOR stud\_cur IS

SELECT \*

FROM student;

Begin

OPEN stud\_cur;

LOOP

FETCH stud\_cur into stud\_row;

EXIT when stud\_cur%NOTFOUND;

dbms\_output.put\_line(stud\_row.student\_name);

dbms\_output.put\_line(stud\_row.student\_id);

dbms\_output.put\_line(stud\_row.address);

dbms\_output.put\_line(stud\_row.city);

END LOOP;

CLOSE stud\_cur;

END;

**Section 14 : Packages**

37).create or replace package College as

procedure select\_departments;

function select\_students (num1 number) return varchar2;

end College;

38). create or replace package body College as

procedure select\_departments

as

depRow department%ROWTYPE;

deptNum varchar2(30);

Cursor dep\_cur is

select \* from department;

begin

dbms\_output.put\_line(dep\_cur.department\_name);

end select\_departments;

function select\_students (num1 number)

return varchar2

is stud\_name varchar2(30);

begin

SELECT student\_name

INTO stud\_name

FROM student

WHERE student.student\_id = num1;

return (stud\_name);

end select\_students;

end College;