Practical Assignment 1

```
create table College(cName varchar2(10), state varchar2(10), enrollment int);

create table Student(sID int, sName varchar2(10), GPA number(2,1), sizeHS int, DoB date);

create table Apply(sID int, cName varchar2(10), major varchar2(20), decision char(1));

insert into Student values (123, 'Amy', 3.9, 1000, '26-JUN-96');
insert into Student values (234, 'Bob', 3.6, 1500, '7-Apr-95');
insert into Student values (345, 'Craig', 3.5, 500, '4-Feb-95');
insert into Student values (456, 'Doris', 3.9, 1000, '24-Jul-97');
insert into Student values (567, 'Edward', 2.9, 2000, '21-Dec-96');
insert into Student values (678, 'Fay', 3.8, 200, '27-Aug-96');
insert into Student values (789, 'Gary', 3.4, 800, '8-Oct-96');
insert into Student values (987, 'Helen', 3.7, 800, '27-Mar-97');
insert into Student values (876, 'Irene', 3.9, 400, '7-Mar-96');
insert into Student values (765, 'Jay', 2.9, 1500, '8-Aug-98');
insert into Student values (654, 'Amy', 3.9, 1000, '26-May-96');
insert into Student values (543, 'Craig', 3.4, 2000, '27-Aug-05');

insert into College values ('Stanford', 'CA', 15000);
insert into College values ('Berkeley', 'CA', 36000);
insert into College values ('MIT', 'MA', 10000);
insert into College values ('Cornell', 'NY', 21000);
insert into College values ('Harvard', 'MA', 50040);

insert into Apply values (123, 'Stanford', 'CS', 'Y');
insert into Apply values (123, 'Stanford', 'EE', 'N');
insert into Apply values (123, 'Berkeley', 'CS', 'Y');
insert into Apply values (123, 'Cornell', 'EE', 'Y');
insert into Apply values (234, 'Berkeley', 'biology', 'N');
insert into Apply values (345, 'MIT', 'bioengineering', 'Y');
insert into Apply values (345, 'Cornell', 'bioengineering', 'N');
insert into Apply values (345, 'Cornell', 'CS', 'Y');
insert into Apply values (345, 'Cornell', 'EE', 'N');
insert into Apply values (678, 'Stanford', 'history', 'Y');
insert into Apply values (987, 'Stanford', 'CS', 'Y');
insert into Apply values (987, 'Berkeley', 'CS', 'Y');
insert into Apply values (876, 'Stanford', 'CS', 'N');
insert into Apply values (876, 'MIT', 'biology', 'Y');
insert into Apply values (876, 'MIT', 'marine biology', 'N');
insert into Apply values (765, 'Stanford', 'history', 'Y');
insert into Apply values (765, 'Cornell', 'history', 'N');
insert into Apply values (765, 'Cornell', 'psychology', 'Y');
insert into Apply values (543, 'MIT', 'CS', 'N');
```

```
1. select sname,dob from student;
2. select sname from student where gpa>3.7;
3. select sname from student where sizehs>=1000 and dob>'31-DEC-1996';
4. select sname from student where gpa between 2.9 and 3.9;
5. select * from college where state='MA';
6. select sname from student where gpa>2.0 and gpa<3.5;
7. select sname from student where dob>'1-jul-96' order by dob;
8. select sid,cname,decision from apply where decision='Y';
9. select sid,cname from apply where cname='Stanford';
10. select cname from college where enrollment>1001;
11. select cname from college where state!='CA';
12. select sname from student where sizehs>1700 and gpa<3.8;
13. desc student;
14. select * from student;
15. select distinct major from apply;
16. select sname from student where sname like '___';
17. select sname from student where sname like 'H____';
18. select sname from student where sname like '__e_e%';
19. select sname from student where sname like '%y';
20. select sname from student order by gpa;
21. select * from student order by gpa,dobdesc;
22. select sid from apply where cname in ('Stanford','MIT','Cornell');
23. delete from apply where cname='Stanford';
24. update student set gpa=gpa+(gpa*0.1);
25. delete from college where cname='Stanford';
26. update student set gpa=gpa+1.5 where gpa<3.5 and sizehs>=1500;
27. delete from student where gpa<3.2;
```

Practical Assignment 2

Create table department (deptno number(3), dname varchar2(20) unique, location varchar2(20) not null check(location in (‘Agra’, ‘Delhi’, ‘Pune’)));

Create table employee (empno varchar2(5) primary key check (empno like ‘E%’), ename varchar2(20) unique, Designationvarchar2(20) not null, salary number(10) check (salary between 15000 and 50000), dob date not null, dno number(3) references department);

Create table candidate (Candidate_ID number(6) primary key, Candidate_Name varchar2(20) not null, Candidate_Email varchar2(30) unique check(Candidate_Email like ‘%_@%_.%_’), Candidate_Dept varchar2(2) default ‘HR’, Manager_ID number(6) references candidate);

1. alter table college add primary key(cname);
2. alter table student Add primary key(sid);
3. alter table apply Add primary key(sid,cname,major);
4. alter table apply Constraint apply_fk foreign key(sid) references student;
5. alter table apply modify(major varchar2(25));
6. alter table apply add(decision varchar2(3) not null);
7. alter table apply modify(decision char(1));
8. alter table apply drop constraint apply_fk;
9. alter table student drop column sizehs;
10. alter table college drop primary key;
11. alter table apply add constraint abc unique(cname,major);
12. alter table apply add constraint foreign key(cname) references college(cname) on delete cascade;
13. alter table apply add constraint fk foreign key(std) references student on delete set null;
14. alter table college rename column enrolment to enroll;

Practical Assignment 3

1. select * from student,apply;
2. select sid,sname,gpa,cname,major from student natural join apply;
3. select * from apply inner join college on apply.cname=college.collegename and state='CA';
4. select sid,sname,gpa,cname from student natural join apply where gpa>3.7 and cname='Stanford';
5. select sid,sname,gpa,sizehs from student natural join apply where major='CS' and decision='N';
6. select sid,sname,gpa,sizehs,cname,major,decision from student natural join apply inner join college on college.collegename=apply.cname where state='NY';
7. select * from student left outer join apply on student.sid=apply.sid where cname is null;
8. select collegename from college left outer join apply on college.collegename=apply.cname where major is null;
9. select a1.sid from apply a1 left outer join apply a2 on a1.sid=a2.sid and (a1.cname!=a2.cname OR a1.major!=a2.major) where a2.major is null;
10. select distinct sname,gpa from student natural join apply inner join college on college.collegename=apply.cname and enrollment<=25000;
11. select a.sid,b.sid from student a inner join student b on a.gpa=b.gpa and a.sid<b.sid;
12. select major from apply natural join college where state = ‘MA’;

Practical Assignment 4

1. select count(sid) from student;
2. select avg(gpa) from student;
3. select min(gpa) as "min_GPA",max(gpa) as"max_GPA" from student;
4. select count(sid) from student where gpa>=3.7;
5. select max(gpa),avg(gpa),min(gpa),sum(gpa) from student;
6. select count(distinct cname) from apply;
7. select count(distinct major) from apply;
8. select count(sid) from apply;
9. select avg(distinct gpa) from student;
10. select count(sid) from apply where decision='Y';
11. select count(sid) from student where gpa>3.4 and sizehs>1000;
12. select count(sid) from apply where major in('marine biology');
13. select count(sid),decision from apply group by decision;
14. select count(sid),major from apply group by major;
15. select count(sid),cname from apply group by cname;
16. select count(sid),cname,major from apply group by cname,major;
17. select cname,major from apply group by cname,major having count(*)>=2;
18. select cname,count(distinct sid) from apply group by cname having count(sid)>=3;
19. select state,count(collegename) from college group by state;
20. select sname from student group by sname having count(sname)>=2;
21. select sname,count(apply.sid) from student left outer join apply on student.sid=apply.sid group by apply.sid,sname ;
22. select sname from student left outer join apply on student.sid=apply.sid group by apply.sid,sname having count(apply.sid)>=3;
23. select sname from student left outer join apply on student.sid=apply.sid group by apply.sid,sname having count(apply.sid)=0;
24. select max(gpa),avg(gpa),min(gpa) from student natural join apply group by cname;
25. select count(sid),gpa from student group by gpa having count(gpa)>=2;

Practical Assignment 5

1. select sid,sname from student where sid in(select sid from apply where major='CS');
2. select sid,sname from student where sizeHS in(select sizeHS from student where sname='Jay');
3. select sid,sname from student where sizeHS in(select sizeHS from student where sname='Jay')and sname not in('Jay');
4. select sid,sname,gpa from student where gpa not in(select gpa from student where sname='Irene');
5. select cname from apply where sid in(select sid from student where snamelike'J%');
6. select distinct major from apply where sid in(select sid from student where sname='Irene');
7. select distinct sid,major from apply where major in(select major from apply where sid in (select sid from student s1 where sname='Irene'));
8. select distinct sid,major from apply where major in(select major from apply where sid in (select sid from student where sname='Irene'))and sid not in(select sid from student where sname='Irene');
9. select count(distinct cname) from apply where sid in(select sid from student where sname='Jay');
10. select sid from apply where cname in(select cname from apply where sid in(select sid from student where sname='Jay'));
11. select * from student where sid in(select sid from apply where major='CS' minus select sid from apply where major='EE');

12. select cname from college c1 where exists(select state from college c2 where c1.state=c2.state and c1.cname!=c2.cname);

13. select cname,enrollment from college c1 where not exists(select enrollment from college c2 where c1.enrollment<c2.enrollment);

14. select sname from student s1 where not exists(select sname from student s2 where s1.gpa>s2.gpa);

15. select major, count(sid) from apply group by major having count(sid)=(select max(count(sid)) from apply group by major);

16. select sid,sizehs,sname from student where sizehs!=(select min(sizehs) from student);

17. select sname,sid from student where not exists (select distinct cname from apply where sid=987 minus select distinct cname from apply where sid=student.sid);

18. insert into Apply select s1.sID, 'Berkeley', 'CSE', 'Y' from student s1where s1.sID IN (select s.sID from student s MINUS select a.sID from apply a where a.cName = 'Berkeley');

select cname from college c where not exists((select sid from student s)minus (select sid from apply a where a.cname=c.cname));

19. select sid from student minus(select distinct sid from apply where cname='Stanford');

20. (select sid from apply where cname='Stanford')intersect(select sid from apply where cname='Berkeley');

21. select sname from student union select cname from college;

22. create table ApplicationInfo(sid int,sname varchar2(10),number_of_applications number(2));

insert into applicationinfo select s.sid,s.sname,nvl(a.no_of_application,0)from ((select sid,sname from student) s left outer join(select sid,count(*) as no_of_application from apply group by sid) a   on s.sid=a.sid);

select * from applicationinfo;

23. create table ApplicationData as (select distinct sid,cname,state from apply natural join college);

24. update apply set decision='N' where cname='Stanford' and sid in ((select sid from apply where cname='Stanford')intersect(select sid from apply where cname='Berkeley'));

25. delete from apply where cname in(select cname from college where state = 'NY');

Practical Assignment 6

1. alter table student add dob date;
2. update student SET DOB = '26-JUN-96' WHERE SID=123;

update student SET DOB = '7-APR-95' WHERE SID=234;

3. select round(avg(gpa),2) from student;
4. select sid , extract (year from dob) from student where sizehs<1000;
5. select sid,extract(year from sysdate)-extract (year from dob) as age from student;
6. select upper(s.sname),lower(a.cname) from student s natural join apply a;
7. select sid, substr(sname,4,1) from student;
8. select sid,sname from student where length(sname)>3;
9. select floor(avg (gpa)),ceil(avg(gpa)),trunc(avg(gpa),1) from student;
10. select * from student where mod(sid,2)=0;
11. select sqrt(900),sqrt(power(24,7)) from dual;
12. select instr('peter piper picked a peck of pickled peppers. a peck of pickled peppers peter piper picked. if peter piper pickled peppers,where the peck of pickled peppers peter piper picked?' ,'pick',1,6) as pick from dual;
13. select translate('Satya Nadella','adelNSty','12345678') from dual;
14. select sid,sname ,to_char(dob,'Monthdd,yyyy') from student;
15. select to_date('26/02/2014','DD/MM/YYYY') from dual;
16. select sysdate, last_day(sysdate) ,next_day(sysdate,'Saturday') from dual;

Practical Assignment 7

1. Create or replace view WeakStudent as select sID, sName, GPA, sizeHS, DoB from student where GPA > 3.7;
2. Create or replace view cView(collegeName, seats, state) as select cName, enrollment, state from student where GPA > 3.7;
3. Create or replace view CSaccept as select sID,cName from apply where major = 'CS' and decision = 'Y';
4. Create view CSberkeley as select sID,sName,GPA,sizeHS,DoB from student natural join CSaccept where cName = 'Berkeley' and major = 'CS' and sizeHS> 500;
5. Select * from CSberkeley where GPA > 3.8;
6. Drop view CSberkeley;
7. Select sID, sName from CSberkeley;
8. Update WeakStudent set GPA = GPA + 0.8 where sizeHS> 1000;
9. Create or replace view AppCount as select sID, count(major) as 'No_of_App' from student group by sID;
10. Update AppCount set No_of_App = 4 where sID = 234;
11. Create view StuName as select sName, GPA from student; It cannot be updated because the view does not contain the primary key.
12. Create or replace view studentHS as select * from student where sizeHS> 1000 with check option;
13. Insert into studentHS values (999,'Ram',9.9,9999); It can be inserted if DoB do not contain any constraint as NOT NULL, if it contains then the above command will give error.
14. Update view studentHS set sizeHS = 999 where sName = 'Ramu'; This is an updatable view so can be updated.
15. Insert into studentHS values (998,'Ramu',9.8,989); generate error because sizeHS is less than 1000

Practical Assignment 8

1. Create index DoBindex on Student(DoB);
2. Create bitmap index MajorIndex on Apply(major);
3. Drop index DoBindex;
4. Create unique index ENRindex on College(enrollment);
5. Create index SCMindex on Apply(sID,cName,major);
6. Create index CMapplyINDX on Apply(cName,major);
7. Create sequence sID_seq   increment by 3   start with 988   maxvalue 999   minvalue 988   cycle   cache 4;
8. sID_seq.nextval;
9. Insert into Student values (sID_seq.nextval, 'Eric', 9.9, 9999, '23-Apr-98');
10. Insert into Student values (sID_seq.nextval, 'Troy', 9.8, 989, '25-Nov-99');
11. Select * from Student;
12. Create sequence GPA_seq   start with 5   maxvalue 5   minvalue 3   cycle   nocache ;
13. Update Student set GPA = GPA_seq.nextval where sName = 'Eric';
14. Insert into Student values (sID_seq.nextval, 'Jack', GPA_seq.nextval, 1500, '22-Oct-97');
15. Select * from Student;