**DATABASE MANAGEMENT FOR A DENTIST’S POLYCLINIC**

**Procedure Functions & SQL Queries**

**1) Procedures and Functions:**

1. 10% Discount for Patients coming for 10 years and more:

create or replace function discount\_reg\_patient(n1 in number)

return number is ans number;

begin

ans:= n1-0.1\*n1;

return ans;

end discount\_reg\_patient ;

declare

n1 number:=:n1;

ans number;

begin

ans:=discount\_reg\_patient(n1);

dbms\_output.put\_line('Regular Patient(Coming for 10 years)');

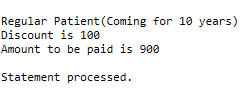
dbms\_output.put\_line('Discount is '||0.1\*n1);

dbms\_output.put\_line('Amount to be paid is '||ans);

end;

**Output:**





1. For New Patients – First Checkup Free

create or replace function ffcheckup\_new\_patient(n1 in number)

return number is ans number;

begin

ans:= n1\*0;

return ans;

end ffcheckup\_new\_patient ;

declare

n1 number:=:n1;

ans number;

begin

ans:=ffcheckup\_new\_patient(n1);

dbms\_output.put\_line('New Patient (First Checkup Free)');

dbms\_output.put\_line('Amount to be paid is '||ans);

end;

**Output:**





1. Procedure for New Patient and Regular Patient

create or replace procedure patient(n1 in number,c in number,ans out number)is

begin

if(c=1) then

ans:=discount\_reg\_patient(n1);

end if;

if(c=2) then

ans:=ffcheckup\_new\_patient(n1);

end if;

end patient;

declare

n1 number:=:n1;

c number:=:c;

ans number;

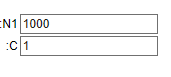
begin

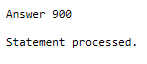
patient(n1,c,ans);

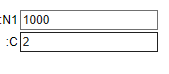
dbms\_output.put\_line('Answer '||ans);

end;

**Output:**









**2) SQL:**

1. create table polyclinic(polyclinic\_name varchar(20) primary key, address varchar(40), timings number(10), phone\_no number(10));

**Output: Table created**

insert into polyclinic values('Dental Polyclinic','Breach Candy',6,9876543210);

**Output: 1 row(s) inserted**

select \* from polyclinic;



1. create table patient1(patient\_id number(7) primary key, polyclinic\_name varchar(20), foreign key(polyclinic\_name) references polyclinic(polyclinic\_name), patient\_name varchar(20), dob date);

**Output: Table created**

insert into patient1 values(1,'Dental Polyclinic','Mr.Smith','03/25/1967');

insert into patient1 values(2,'Dental Polyclinic','Mr.Andrews','02/04/1978');

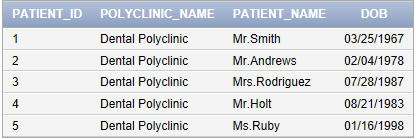
insert into patient1 values(3,'Dental Polyclinic','Mrs.Rodriguez','07/28/1987');

insert into patient1 values(4,'Dental Polyclinic','Mr.Holt','08/21/1983');

insert into patient1 values(5,'Dental Polyclinic','Ms.Ruby','01/16/1998');

**Output: 1 row(s) inserted**

select \* from patient1;



1. create table patient2(patient\_id number(7), foreign key(patient\_id) references patient1(patient\_id), prev\_treatment varchar(50) primary key, last\_date\_visit date);

**Output: Table created**

insert into patient2 values(1,'Root Canal','10/10/2017');

insert into patient2 values(2,'Gums','12/11/2016');

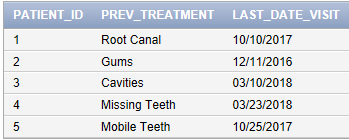
insert into patient2 values(3,'Cavities','03/10/2018');

insert into patient2 values(4,'Missing Teeth','03/23/2018');

insert into patient2 values(5,'Mobile Teeth','10/25/2017');

**Output: 1 row(s) inserted**

select \* from patient2;



1. create table regular\_patient(discount number(5), patient\_id number(7), foreign key(patient\_id) references patient1(patient\_id));

**Output: Table created**

insert into regular\_patient values(null,1);

insert into regular\_patient values(250,2);

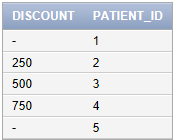
insert into regular\_patient values(500,3);

insert into regular\_patient values(750,4);

insert into regular\_patient values(null,5);

**Output: 1 row(s) inserted**

select \* from regular\_patient;



1. create table new\_patient( first\_free\_checkup varchar(10), date\_visit date, patient\_id number(7), foreign key(patient\_id) references patient1(patient\_id));

**Output: Table created**

insert into new\_patient values('Y','10/10/2017',1);

insert into new\_patient values('N','12/11/2016',2);

insert into new\_patient values('N','03/10/2018',3);

insert into new\_patient values('N','03/23/2018',4);

insert into new\_patient values('Y','10/25/2017',5);

**Output: 1 row(s) inserted**

select \* from new\_patient;



1. create table doc1( doc\_id number(7) primary key, salary\_slipno number(5) unique, doc\_name varchar(20), phone\_no number(10));

**Output: Table created**

insert into doc1 values(100,100,'Dr. Ray',9821054690);

insert into doc1 values(200,101,'Dr. Bing',9821009751);

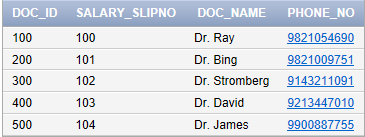
insert into doc1 values(300,102,'Dr. Stromberg',9143211091);

insert into doc1 values(400,103,'Dr. David',9213447010);

insert into doc1 values(500,104,'Dr. James',9900887755);

**Output: 1 row(s) inserted**

select \* from doc1;



1. create table doc2(salary\_slipno number(5) primary key, foreign key(salary\_slipno) references doc1(salary\_slipno), salary number(6));

**Output: Table created**

insert into doc2 values(100,500000);

insert into doc2 values(101,250200);

insert into doc2 values(102,512200);

insert into doc2 values(103,700000);

insert into doc2 values(104,656666);

**Output: 1 row(s) inserted**

select \* from doc2;



1. create table doc3(doc\_id number(7), foreign key(doc\_id) references doc1(doc\_id), dep\_no number(5) , dep\_name varchar(20));

**Output: Table created**

insert into doc3 values(100,1,'Endodontist');

insert into doc3 values(200,1,'Endodontist');

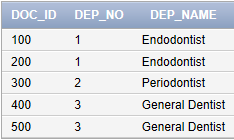
insert into doc3 values(300,2,'Periodontist');

insert into doc3 values(400,3,'General Dentist');

insert into doc3 values(500,3,'General Dentist');

**Output: 1 row(s) inserted**

select \* from doc3;



1. create table endodontist(doc\_id number(7), foreign key(doc\_id) references doc1(doc\_id), root\_canal varchar(20));

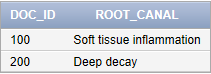
**Output: Table created**

insert into endodontist values(100,'Soft tissue inflammation');

insert into endodontist values(200,'Deep decay');

**Output: 1 row(s) inserted**

select \* from endodontist;



1. create table periodontist(doc\_id number(7), foreign key(doc\_id) references doc1(doc\_id), gums varchar(20));

**Output: Table created**

insert into periodontist values(300,'Gum Disease');

**Output: 1 row(s) inserted**

select \* from periodontist;



1. create table gen\_dentist(doc\_id number(7), foreign key(doc\_id) references doc1(doc\_id), cavities varchar(20), missing\_teeth varchar(20), mobile\_teeth varchar(20));

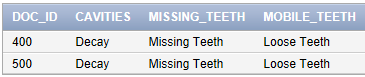
**Output: Table created**

insert into gen\_dentist values(400,'Decay','Missing Teeth','Loose Teeth');

insert into gen\_dentist values(500,'Decay','Missing Teeth','Loose Teeth');

**Output: 1 row(s) inserted**

select \* from gen\_dentist;



1. create table dependents(depen\_name varchar(10) primary key, phone\_no number(10),patient\_id number(7), foreign key(patient\_id) references patient1(patient\_id));

**Output: Table created**

insert into dependents values('Roger',9165625400,1);

insert into dependents values('Fin',9165623880,2);

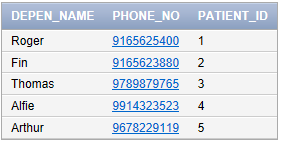
insert into dependents values('Thomas',9789879765,3);

insert into dependents values('Alfie',9914323523,4);

insert into dependents values('Arthur',9678229119,5);

**Output: 1 row(s) inserted**

select \* from dependents;



1. create table medic\_hist(patient\_id number(7), foreign key(patient\_id) references patient1(patient\_id), past\_treatment varchar(20) primary key, allergies varchar(20), pain\_tooth varchar(20), heart\_probs varchar (20), other\_illness varchar(20));

**Output: Table created**

insert into medic\_hist values(1,'Root Canal','Penicillin',null,'High BP','Diabetes');

insert into medic\_hist values(2,'Cavities',null,'Upper Left Tooth',null,'Rhinitis');

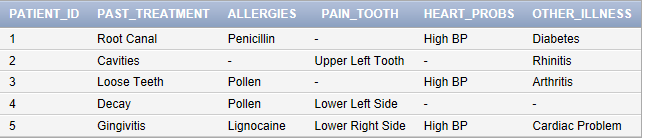
insert into medic\_hist values(3,'Loose Teeth','Pollen',null,'High BP','Arthritis');

insert into medic\_hist values(4,'Decay','Pollen','Lower Left Side',null,null);

insert into medic\_hist values(5,'Gingivitis','Lignocaine','Lower Right Side','High BP','Cardiac Problem');

**Output: 1 row(s) inserted**

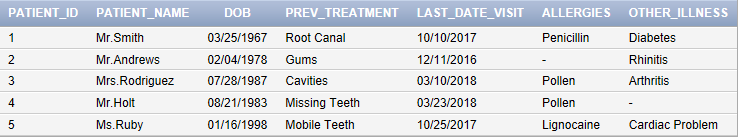
select \* from medic\_hist;



14) To get the Full Patient information:

select patient1.patient\_id, patient1.patient\_name, patient1.dob, patient2.prev\_treatment, patient2.last\_date\_visit, medic\_hist.allergies, medic\_hist.other\_illness from patient1, patient2, medic\_hist where patient1.patient\_id = patient2.patient\_id and patient1.patient\_id = medic\_hist.patient\_id;

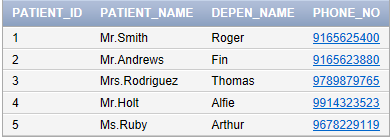
**Output:**



15) To get the Patient’s Dependents information:

select patient1.patient\_id, patient1.patient\_name, dependents.depen\_name, dependents.phone\_no from patient1 inner join dependents on patient1.patient\_id = dependents.patient\_id;

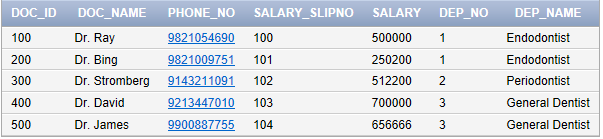
**Output:**



16) To get Full Doctor information:

select doc1.doc\_id, doc1.doc\_name, doc1.phone\_no, doc2.salary\_slipno, doc2.salary, doc3.dep\_no, doc3.dep\_name from doc1, doc2, doc3 where doc1.salary\_slipno = doc2.salary\_slipno and doc1.doc\_id = doc3.doc\_id;

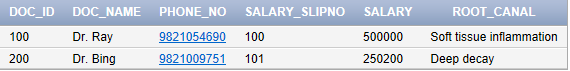
**Output:**



17) To get the Endodontist’s information:

select doc1.doc\_id, doc1.doc\_name,doc1.phone\_no, doc2.salary\_slipno, doc2.salary, endodontist.root\_canal from doc1, doc2, endodontist where doc1.salary\_slipno = doc2.salary\_slipno and doc1.doc\_id = endodontist.doc\_id;

**Output:**



18) To get the Periodontist’s information:

select doc1.doc\_id, doc1.doc\_name,doc1.phone\_no, doc2.salary\_slipno, doc2.salary, periodontist.gums from doc1, doc2, periodontist where doc1.salary\_slipno = doc2.salary\_slipno and doc1.doc\_id = periodontist.doc\_id;

**Output:**



19) To get the General Dentist’s information:

select doc1.doc\_id, doc1.doc\_name, doc1.phone\_no, doc2.salary\_slipno, doc2.salary, gen\_dentist.cavities, gen\_dentist.missing\_teeth, gen\_dentist.mobile\_teeth from doc1, doc2, gen\_dentist where doc1.salary\_slipno = doc2.salary\_slipno and doc1.doc\_id = gen\_dentist.doc\_id;

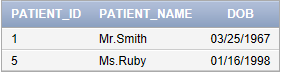
**Output:**



20) To find out which Patients are new (which are in the first check-up free category)

select patient\_id, patient\_name, dob from patient1 where(patient\_id) in (select patient\_id from new\_patient where first\_free\_checkup = 'Y');

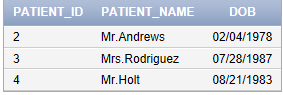
**Output:**



21) To find out which Patients are regular (which get a discount)

select patient\_id, patient\_name, dob from patient1 where(patient\_id) in (select patient\_id from regular\_patient where discount>0);

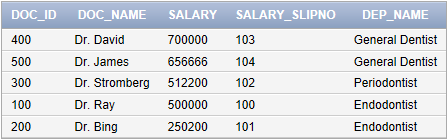
**Output**:



22) To find out which Doctor has the highest salary

select doc1.doc\_id, doc1.doc\_name, doc2.salary ,doc2.salary\_slipno, doc3.dep\_name from doc1,doc2,doc3 where doc1.salary\_slipno = doc2.salary\_slipno and doc1.doc\_id = doc3.doc\_id order by doc2.salary desc;

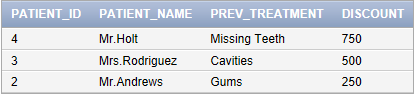
**Output**:



23) To find out which Regular Patients get the most discount

select patient1.patient\_id, patient1.patient\_name, patient2.prev\_treatment, regular\_patient.discount from patient1, patient2, regular\_patient where patient1.patient\_id = patient2.patient\_id and patient1.patient\_id = regular\_patient.patient\_id and regular\_patient.discount>0 order by regular\_patient.discount desc;

**Output**:



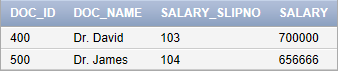
24) To find the Doctors which have salary greater than the average salary of all the Doctors.

select avg(salary) from doc2;

select doc1.doc\_id ,doc1.doc\_name, doc2.salary\_slipno, doc2.salary from doc1,doc2 where doc1.salary\_slipno = doc2.salary\_slipno and doc2.salary >= (select avg(salary) as sal from doc2) ;

**Output:**

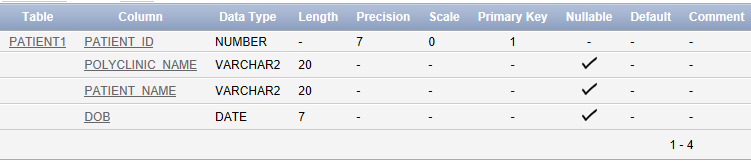




25) The structure of the Patient table

desc patient1;

**Output:**



26) Altering an entry

alter table polyclinic drop (address);

alter table polyclinic add (address varchar(20));

**Output:**



27) Updating a table

update polyclinic set address = 'Malabar Hill';

select \* from polyclinic;

**Output:**





28) To find out the most discount which the Polyclinic is giving at the moment

select max(discount) from regular\_patient;

**Output**:



29) To find the least discount which the Polyclinic is giving at the moment

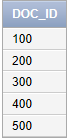
select min(discount) from regular\_patient;

**Output:**



30) Use of union operator

select doc\_id from endodontist union select doc\_id from periodontist union select doc\_id from gen\_dentist;



31) Use of intersect operator

select doc\_id from endodontist intersect select doc\_id from periodontist intersect select doc\_id from gen\_dentist;

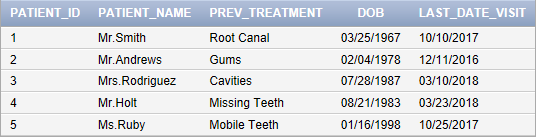
**Output:**

 Since doc\_id is primary key, it wont repeat.

32) Using natural join

select patient\_id, patient\_name, prev\_treatment, dob, last\_date\_visit from patient1 natural join patient2;

Output:



33) Use of null values

select patient\_id from regular\_patient where discount is null;

**Output:**



34) To find number of tuples in a relation

select count(\*) from doc1;

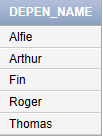
**Output:**



35) Use of distinct

select distinct depen\_name from dependents;

**Output:**



36) Use of minus

select patient\_id from medic\_hist where heart\_probs = 'High BP' minus select patient\_id from medic\_hist where heart\_probs = null;

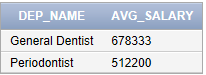
**Output:**



37) Use of group by

select doc3.dep\_name, avg(doc2.salary) as avg\_salary from doc3,doc2,doc1 where doc1.salary\_slipno = doc2.salary\_slipno and doc1.doc\_id = doc3.doc\_id group by doc3.dep\_name having avg(doc2.salary)>500000

**Output:**



38) Use of not in

select distinct doc\_name from doc1 where doc\_name not in('Dr. David', 'Dr. James');

**Output**:

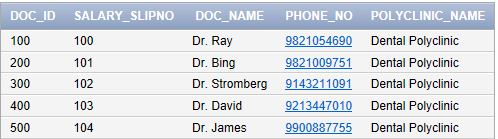


39) alter table doc1 add (polyclinic\_name varchar(20), foreign key(polyclinic\_name) references polyclinic(polyclinic\_name));

update doc1 set polyclinic\_name = 'Dental Polyclinic'

select \* from doc1;

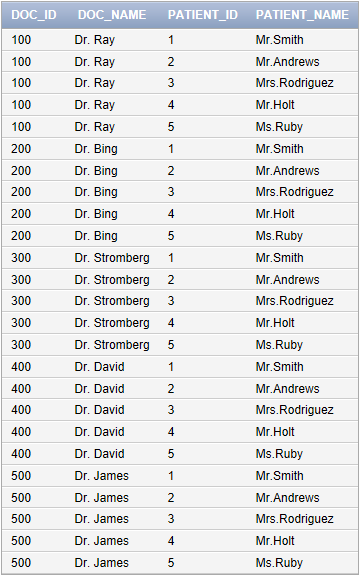
**Output**:



40) Each patient can go to all the doctors in the polyclinic

select doc1.doc\_id, doc1.doc\_name, patient1.patient\_id ,patient1.patient\_name from doc1, patient1 where doc1.polyclinic\_name = patient1.polyclinic\_name order by doc\_id asc, patient\_id asc;

**Output:**



41) create view polyclinic\_view as select

polyclinic\_name, address, timings, phone\_no from polyclinic with read only;

select \* from polyclinic\_view;

**Output:**


42) create view polyclinic\_view2 as select

polyclinic\_name, address, timings, phone\_no from polyclinic ;

insert into polyclinic\_view2 values('Dentists', 'Breach Candy', 2, 9822334519);

select \* from polyclinic\_view2;

**Output**:





**Trigger:**

create or replace trigger print\_salary1

before delete or insert or update on doc2

for each row

when(new.salary\_slipno>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('Difference '|| sal\_diff);

end;

/

**Output:**



update doc2 set salary = salary+500 where salary\_slipno = 100;

**Output:**

