Oracle SQL Capstone Project Exam (17/07/2024)

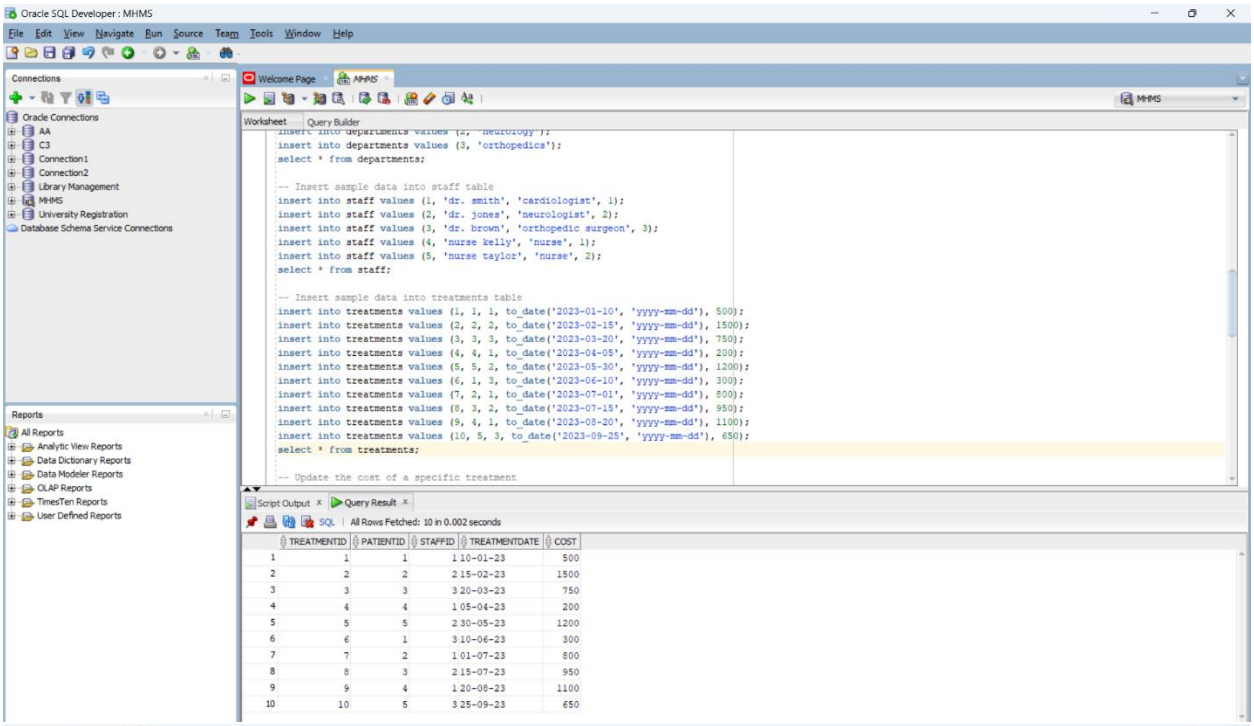
```
-- Create Tables: patients, departments, staff, treatments

-- Patients table
create table patients (
    patientid number primary key,
    name varchar2(100),
    dob date,
    email varchar2(100) unique,
    phone varchar2(20)
);

-- Departments table
create table departments (
    departmentid number primary key,
    departmentname varchar2(100)
);

-- Staff table
create table staff (
    staffid number primary key,
    name varchar2(100),
    role varchar2(50),
    departmentid number,
    constraint fk_staff_department foreign key (departmentid) references
departments(departmentid)
);

-- Treatments table
create table treatments (
    treatmentid number primary key,
    patientid number,
    staffid number,
    treatmentdate date,
    cost number(10, 2),
    constraint fk_treatments_patients foreign key (patientid) references
patients(patientid) on delete cascade,
    constraint fk_treatments_staff foreign key (staffid) references
staff(staffid),
    constraint chk_cost_nonnegative check (cost >= 0)
```

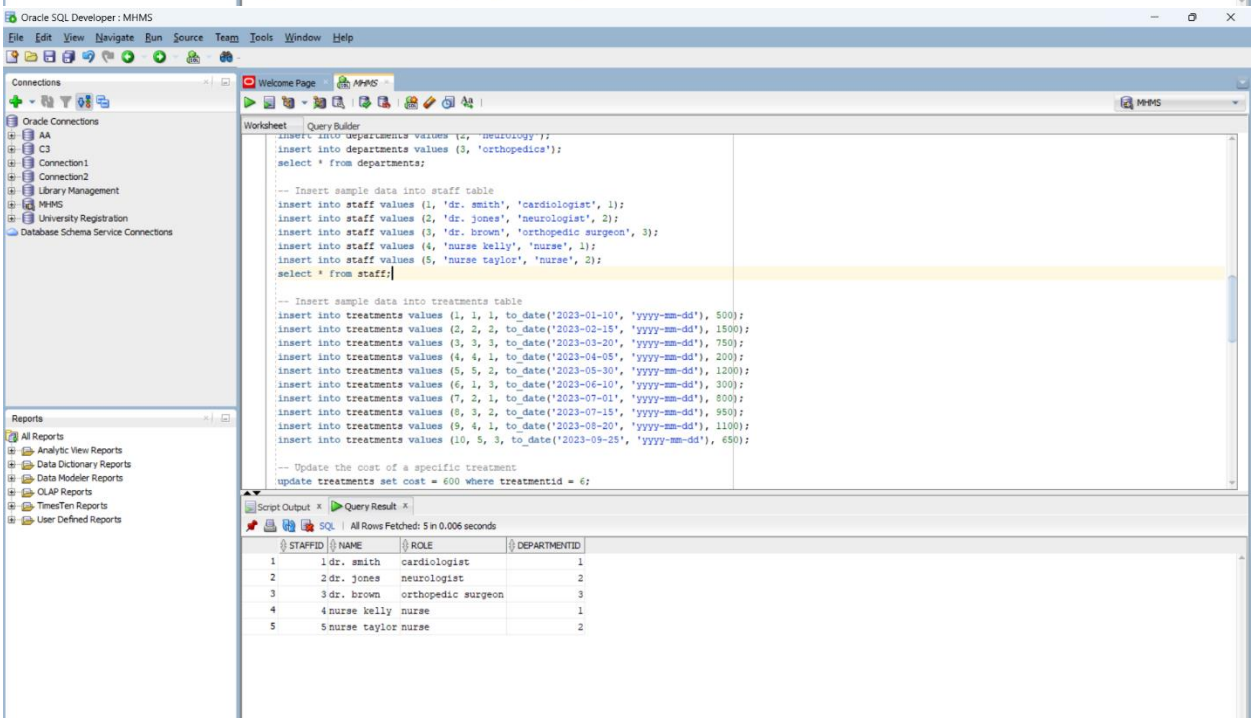
```
);

-- Insert sample data into patients table
insert into patients values (1, 'john doe', to_date('1985-05-15', 'yyyy-mm-
dd'), 'john.doe@example.com', '555-0101');
insert into patients values (2, 'jane smith', to_date('1990-06-20', 'yyyy-mm-
dd'), 'jane.smith@example.com', '555-0102');
insert into patients values (3, 'alice johnson', to_date('1975-09-25', 'yyyy-
mm-dd'), 'alice.j@example.com', '555-0103');
insert into patients values (4, 'bob brown', to_date('1982-01-30', 'yyyy-mm-
dd'), 'bob.brown@example.com', '555-0104');
insert into patients values (5, 'eve white', to_date('1995-12-12', 'yyyy-mm-
dd'), 'eve.white@example.com', '555-0105');

-- Insert sample data into departments table
insert into departments values (1, 'cardiology');
insert into departments values (2, 'neurology');
insert into departments values (3, 'orthopedics');

-- Insert sample data into staff table
insert into staff values (1, 'dr. smith', 'cardiologist', 1);
insert into staff values (2, 'dr. jones', 'neurologist', 2);
insert into staff values (3, 'dr. brown', 'orthopedic surgeon', 3);
insert into staff values (4, 'nurse kelly', 'nurse', 1);
insert into staff values (5, 'nurse taylor', 'nurse', 2);

-- Insert sample data into treatments table
insert into treatments values (1, 1, 1, to_date('2023-01-10', 'yyyy-mm-dd'),
500);
insert into treatments values (2, 2, 2, to_date('2023-02-15', 'yyyy-mm-dd'),
1500);
insert into treatments values (3, 3, 3, to_date('2023-03-20', 'yyyy-mm-dd'),
750);
insert into treatments values (4, 4, 1, to_date('2023-04-05', 'yyyy-mm-dd'),
200);
insert into treatments values (5, 5, 2, to_date('2023-05-30', 'yyyy-mm-dd'),
1200);
insert into treatments values (6, 1, 3, to_date('2023-06-10', 'yyyy-mm-dd'),
300);
insert into treatments values (7, 2, 1, to_date('2023-07-01', 'yyyy-mm-dd'),
800);
insert into treatments values (8, 3, 2, to_date('2023-07-15', 'yyyy-mm-dd'),
950);
```

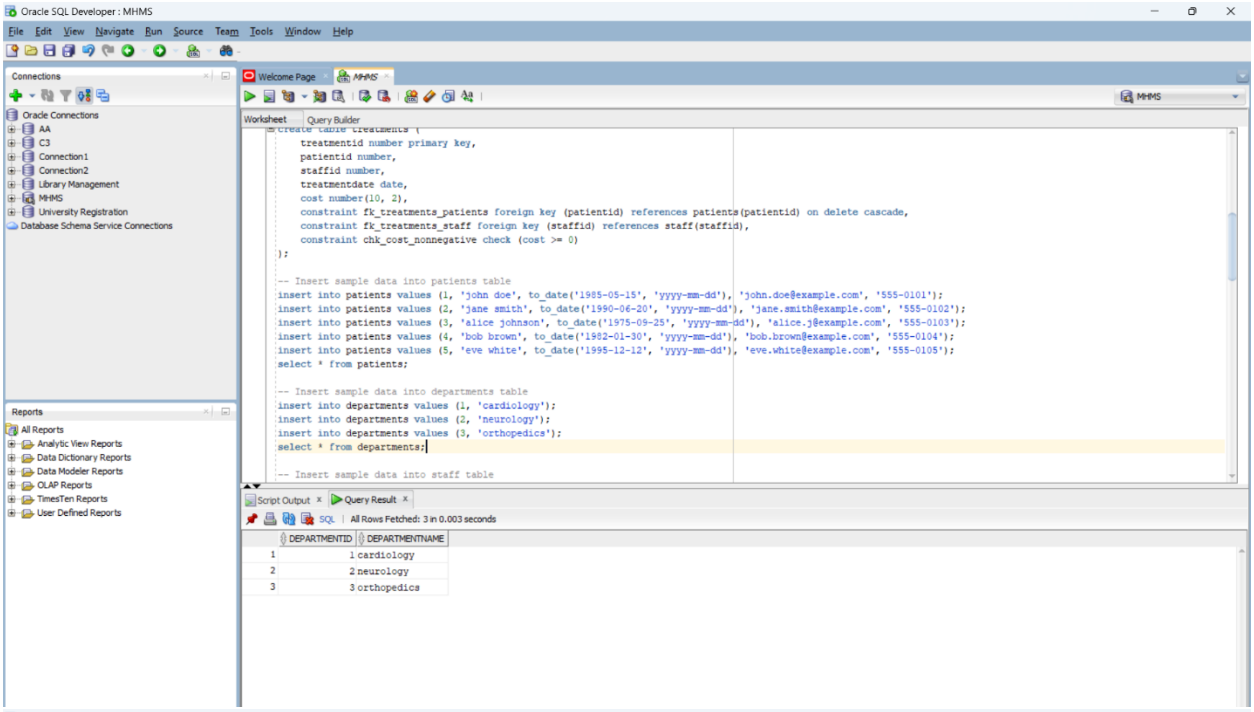
```
insert into treatments values (9, 4, 1, to_date('2023-08-20', 'yyyy-mm-dd'),
1100);
insert into treatments values (10, 5, 3, to_date('2023-09-25', 'yyyy-mm-dd'),
650);

-- Update the cost of a specific treatment
update treatments set cost = 600 where treatmentid = 6;

-- Delete a patient and observe the cascading effect on associated treatments
-- Note: Ensure ON DELETE CASCADE is enabled on the foreign key constraint
(already defined above)

-- Delete patient with patientid = 4
delete from patients where patientid = 4;
select * from treatments;

/*
Data Retrieval using different operations
Write a query to list all patients, their associated treatments, and the
staff who provided those treatments.
*/
select p.name as patient_name, t.treatmentdate, s.name as staff_name
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid;

-- Create a query to display all treatments, including patient names, staff
names, and department names
select p.name as patient_name, s.name as staff_name, d.departmentname,
t.treatmentdate, t.cost
from treatments t
join patients p on t.patientid = p.patientid
join staff s on t.staffid = s.staffid
join departments d on s.departmentid = d.departmentid;

/*
Subqueries and Correlated Subqueries
Use a subquery to find patients who have received treatments from a specific
department (provided DepartmentID).
*/
select distinct p.*
from patients p
join treatments t on p.patientid = t.patientid
```

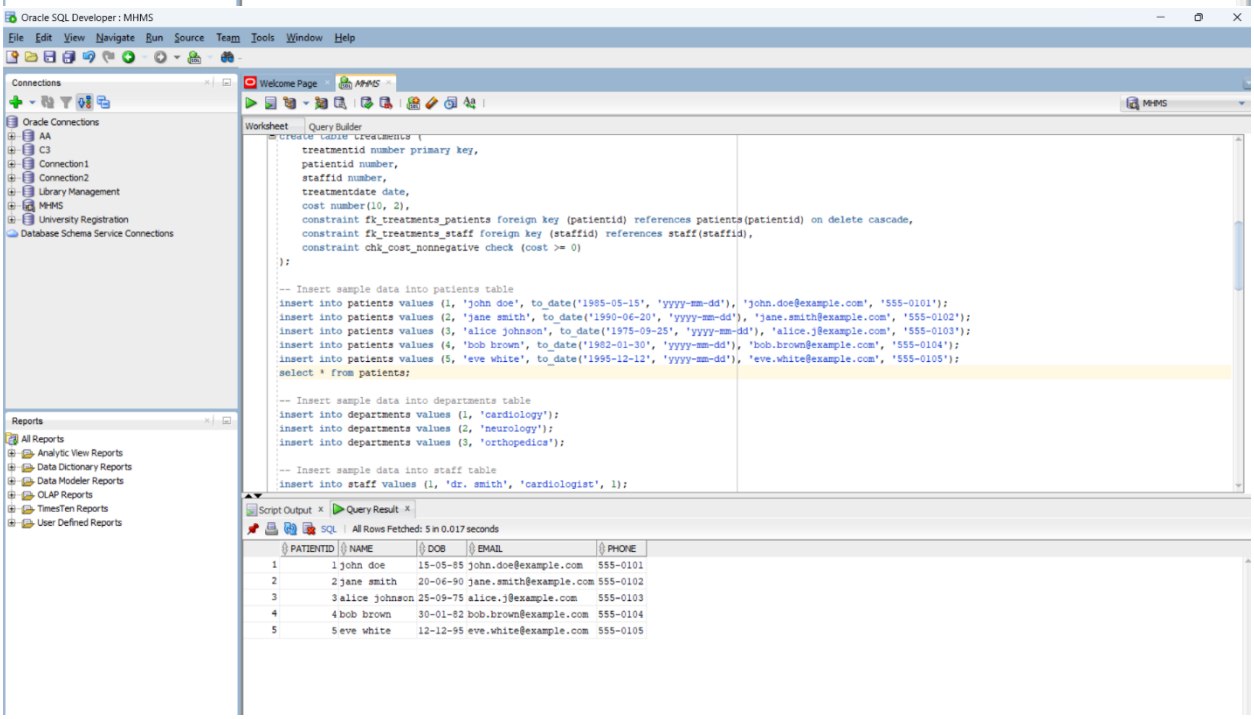
```
join staff s on t.staffid = s.staffid
where s.departmentid = 1;

-- Use a correlated subquery to find staff who have treated more than a
specified number of patients (e.g., 3)
select *
from staff s
where (select count(*) from treatments t where t.staffid = s.staffid) > 2;

/*
Aggregate Functions with GROUP BY and HAVING
Calculate the total cost of treatments per patient and filter out patients
with a total treatment cost less than 1000
*/
select p.name as patient_name, sum(t.cost) as total_cost
from patients p
join treatments t on p.patientid = t.patientid
group by p.name
having sum(t.cost) >= 1000;
```

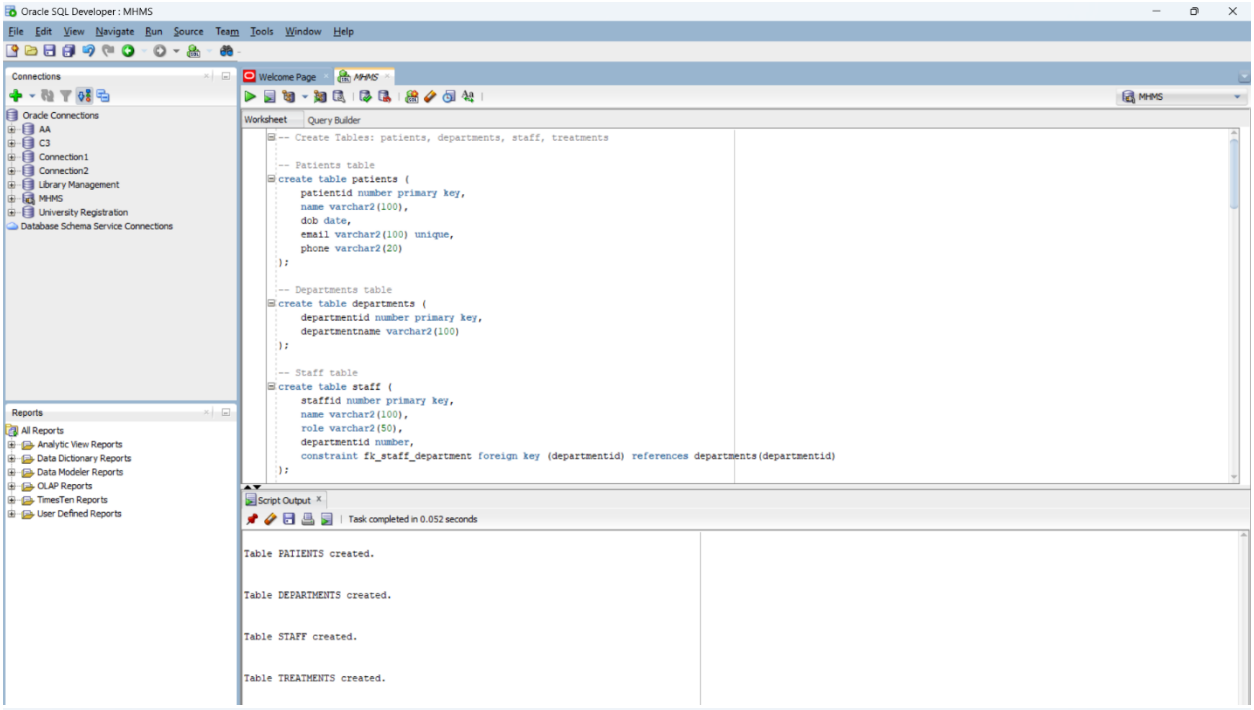
OUTPUT: -

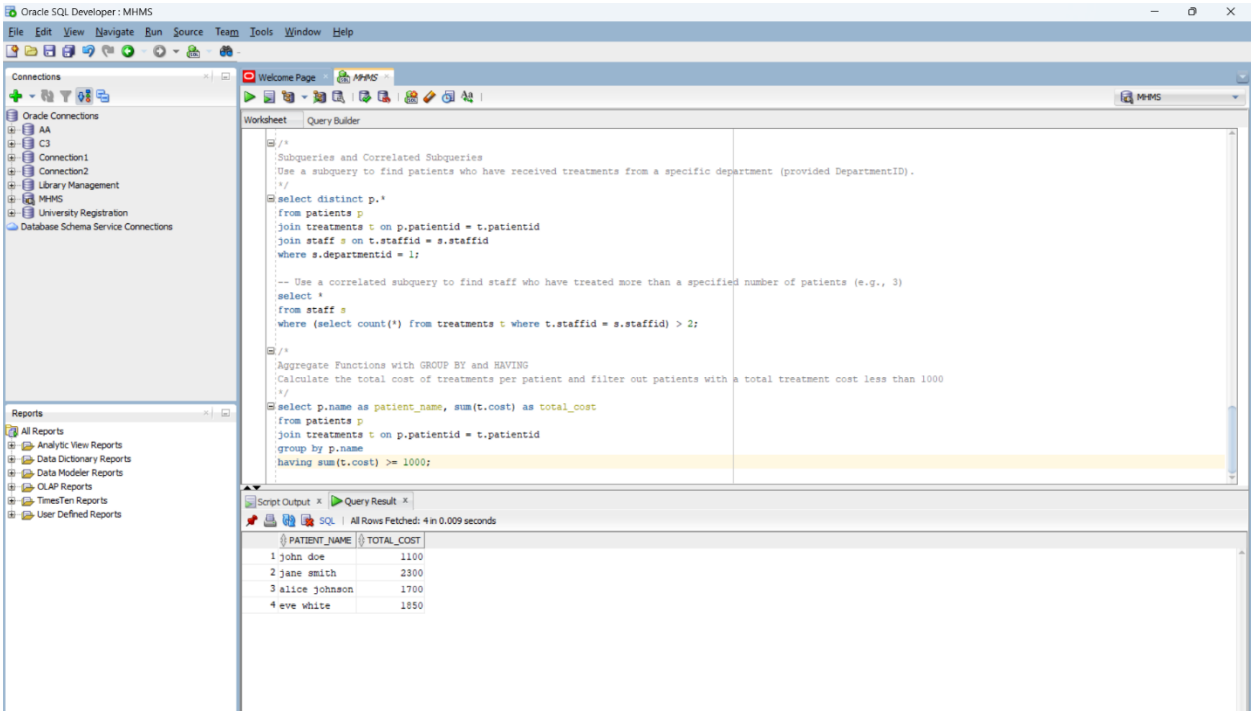

Oracle SQL Developer : MHMS
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
C3
Connection1
Connection2
Library Management
MHMS
University Registration
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
MHMS
Worksheet
Query Builder
insert into departments values (3, 'orthopedics');
select * from departments;
-- Insert sample data into staff table
insert into staff values (1, 'dr. smith', 'cardiologist', 1);
insert into staff values (2, 'dr. jones', 'neurologist', 2);
insert into staff values (3, 'dr. brown', 'orthopedic surgeon', 3);
insert into staff values (4, 'nurse kelly', 'nurse', 1);
insert into staff values (5, 'nurse taylor', 'nurse', 2);
select * from staff;
-- Insert sample data into treatments table
insert into treatments values (1, 1, 1, to_date('2023-01-10', 'yyyy-mm-dd'), 500);
insert into treatments values (2, 2, 2, to_date('2023-02-15', 'yyyy-mm-dd'), 1500);
insert into treatments values (3, 3, 3, to_date('2023-03-20', 'yyyy-mm-dd'), 750);
insert into treatments values (4, 4, 1, to_date('2023-04-05', 'yyyy-mm-dd'), 200);
insert into treatments values (5, 5, 2, to_date('2023-05-30', 'yyyy-mm-dd'), 1200);
insert into treatments values (6, 1, 3, to_date('2023-06-10', 'yyyy-mm-dd'), 300);
insert into treatments values (7, 2, 1, to_date('2023-07-01', 'yyyy-mm-dd'), 800);
insert into treatments values (8, 3, 2, to_date('2023-07-15', 'yyyy-mm-dd'), 950);
insert into treatments values (9, 4, 1, to_date('2023-08-20', 'yyyy-mm-dd'), 1100);
insert into treatments values (10, 5, 3, to_date('2023-09-25', 'yyyy-mm-dd'), 650);
select * from treatments;
-- Update the cost of a specific treatment
Script Output
Query Result
SQL | All Rows Fetched: 10 in 0.002 seconds
TREATMENTID PATIENTID STAFFID TREATMENTDATE COST
1 1 1 1 10-01-23 500
2 2 2 2 15-02-23 1500
3 3 3 3 20-03-23 750
4 4 4 1 05-04-23 200
5 5 5 2 30-05-23 1200
6 6 1 3 10-06-23 300
7 7 2 1 01-07-23 800
8 8 3 2 15-07-23 950
9 9 4 1 20-08-23 1100
10 10 5 3 25-09-23 650

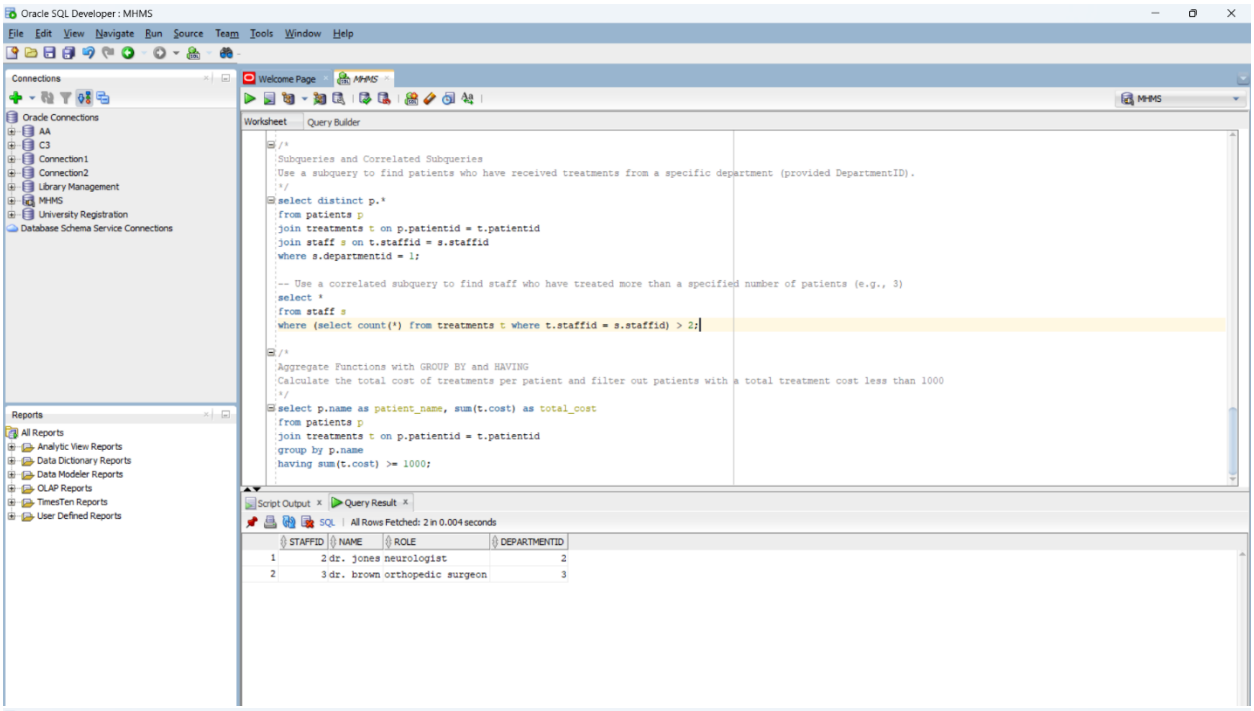

Oracle SQL Developer : MHMS
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
C3
Connection1
Connection2
Library Management
MHMS
University Registration
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
MHMS
Worksheet
Query Builder
insert into departments values (3, 'orthopedics');
select * from departments;
-- Insert sample data into staff table
insert into staff values (1, 'dr. smith', 'cardiologist', 1);
insert into staff values (2, 'dr. jones', 'neurologist', 2);
insert into staff values (3, 'dr. brown', 'orthopedic surgeon', 3);
insert into staff values (4, 'nurse kelly', 'nurse', 1);
insert into staff values (5, 'nurse taylor', 'nurse', 2);
select * from staff;
-- Insert sample data into treatments table
insert into treatments values (1, 1, 1, to_date('2023-01-10', 'yyyy-mm-dd'), 500);
insert into treatments values (2, 2, 2, to_date('2023-02-15', 'yyyy-mm-dd'), 1500);
insert into treatments values (3, 3, 3, to_date('2023-03-20', 'yyyy-mm-dd'), 750);
insert into treatments values (4, 4, 1, to_date('2023-04-05', 'yyyy-mm-dd'), 200);
insert into treatments values (5, 5, 2, to_date('2023-05-30', 'yyyy-mm-dd'), 1200);
insert into treatments values (6, 1, 3, to_date('2023-06-10', 'yyyy-mm-dd'), 300);
insert into treatments values (7, 2, 1, to_date('2023-07-01', 'yyyy-mm-dd'), 800);
insert into treatments values (8, 3, 2, to_date('2023-07-15', 'yyyy-mm-dd'), 950);
insert into treatments values (9, 4, 1, to_date('2023-08-20', 'yyyy-mm-dd'), 1100);
insert into treatments values (10, 5, 3, to_date('2023-09-25', 'yyyy-mm-dd'), 650);
-- Update the cost of a specific treatment
update treatments set cost = 600 where treatmentid = 6;
Script Output
Query Result
SQL | All Rows Fetched: 5 in 0.006 seconds
STAFFID NAME ROLE DEPARTMENTID
1 1 dr. smith cardiologist 1
2 2 dr. jones neurologist 2
3 3 dr. brown orthopedic surgeon 3
4 4 nurse kelly nurse 1
5 5 nurse taylor nurse 2

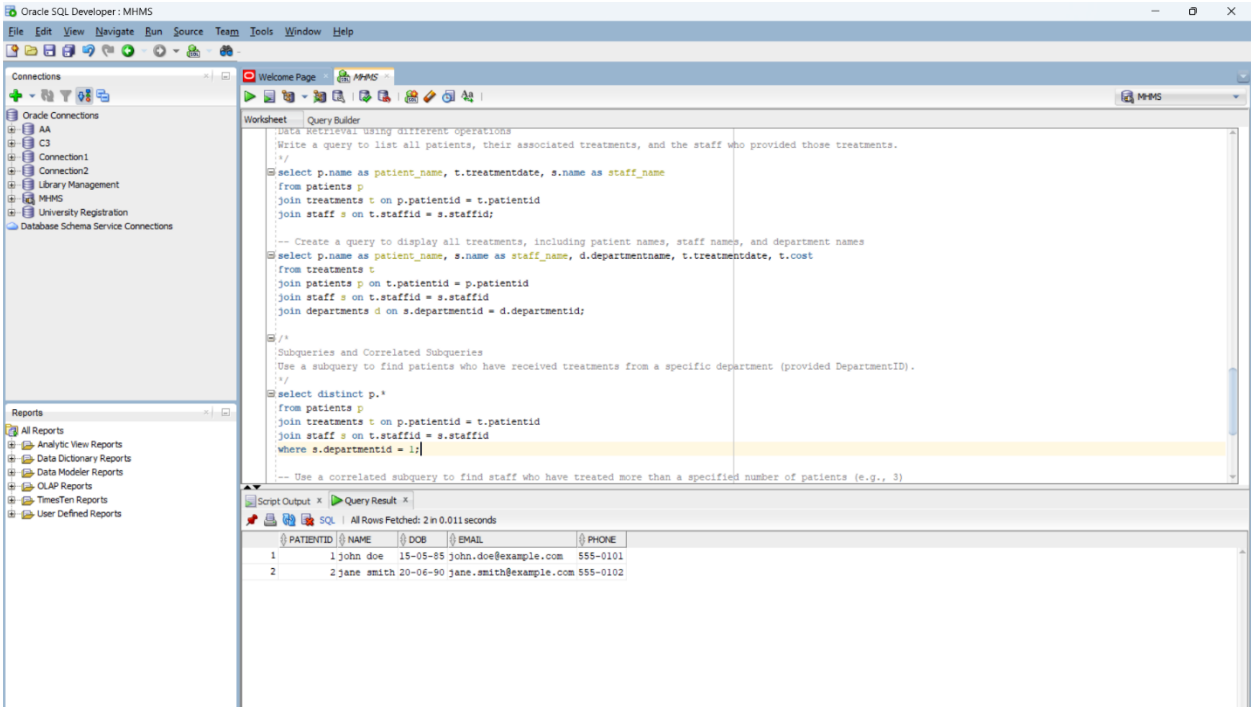

Oracle SQL Developer : MHMS
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
C3
Connection1
Connection2
Library Management
MHMS
University Registration
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
MHMS
Worksheet
Query Builder
create table treatments (
treatmentid number primary key,
patientid number,
staffid number,
treatmentdate date,
cost number(10, 2),
constraint fk_treatments_patients foreign key (patientid) references patients(patientid) on delete cascade,
constraint fk_treatments_staff foreign key (staffid) references staff(staffid),
constraint chk_cost_nonnegative check (cost >= 0)
);
-- Insert sample data into patients table
insert into patients values (1, 'john doe', to_date('1985-05-15', 'yyyy-mm-dd'), 'john.doe@example.com', '555-0101');
insert into patients values (2, 'jane smith', to_date('1990-06-20', 'yyyy-mm-dd'), 'jane.smith@example.com', '555-0102');
insert into patients values (3, 'alice johnson', to_date('1975-09-25', 'yyyy-mm-dd'), 'alice.j@example.com', '555-0103');
insert into patients values (4, 'bob brown', to_date('1982-01-30', 'yyyy-mm-dd'), 'bob.brown@example.com', '555-0104');
insert into patients values (5, 'eve white', to_date('1995-12-12', 'yyyy-mm-dd'), 'eve.white@example.com', '555-0105');
select * from patients;
-- Insert sample data into departments table
insert into departments values (1, 'cardiology');
insert into departments values (2, 'neurology');
insert into departments values (3, 'orthopedics');
select * from departments;
-- Insert sample data into staff table
Script Output
Query Result
SQL | All Rows Fetched: 3 in 0.003 seconds
DEPARTMENTID DEPARTMENTNAME
1 1 cardiology
2 2 neurology
3 3 orthopedics

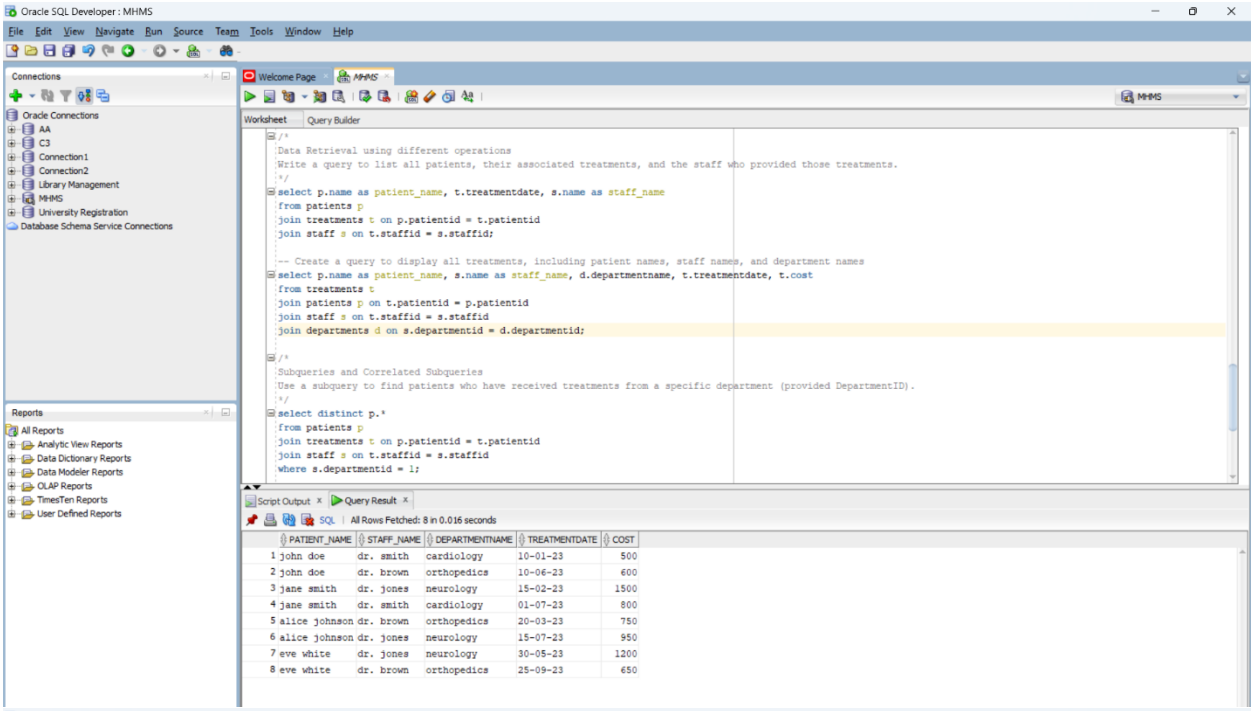

Oracle SQL Developer : MHMS
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
C3
Connection1
Connection2
Library Management
MHMS
University Registration
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
MHMS
Worksheet
Query Builder
create table treatments (
treatmentid number primary key,
patientid number,
staffid number,
treatmentdate date,
cost number(10, 2),
constraint fk_treatments_patients foreign key (patientid) references patients(patientid) on delete cascade,
constraint fk_treatments_staff foreign key (staffid) references staff(staffid),
constraint chk_cost_nonnegative check (cost >= 0)
);
-- Insert sample data into patients table
insert into patients values (1, 'john doe', to_date('1985-05-15', 'yyyy-mm-dd'), 'john.doe@example.com', '555-0101');
insert into patients values (2, 'jane smith', to_date('1990-06-20', 'yyyy-mm-dd'), 'jane.smith@example.com', '555-0102');
insert into patients values (3, 'alice johnson', to_date('1975-09-25', 'yyyy-mm-dd'), 'alice.j@example.com', '555-0103');
insert into patients values (4, 'bob brown', to_date('1982-01-30', 'yyyy-mm-dd'), 'bob.brown@example.com', '555-0104');
insert into patients values (5, 'eve white', to_date('1995-12-12', 'yyyy-mm-dd'), 'eve.white@example.com', '555-0105');
select * from patients;
-- Insert sample data into departments table
insert into departments values (1, 'cardiology');
insert into departments values (2, 'neurology');
insert into departments values (3, 'orthopedics');
-- Insert sample data into staff table
insert into staff values (1, 'dr. smith', 'cardiologist', 1);
Script Output
Query Result
SQL | All Rows Fetched: 5 in 0.017 seconds
PATIENTID NAME DOB EMAIL PHONE
1 1 john doe 15-05-85 john.doe@example.com 555-0101
2 2 jane smith 20-06-90 jane.smith@example.com 555-0102
3 3 alice johnson 25-09-75 alice.j@example.com 555-0103
4 4 bob brown 30-01-82 bob.brown@example.com 555-0104
5 5 eve white 12-12-95 eve.white@example.com 555-0105

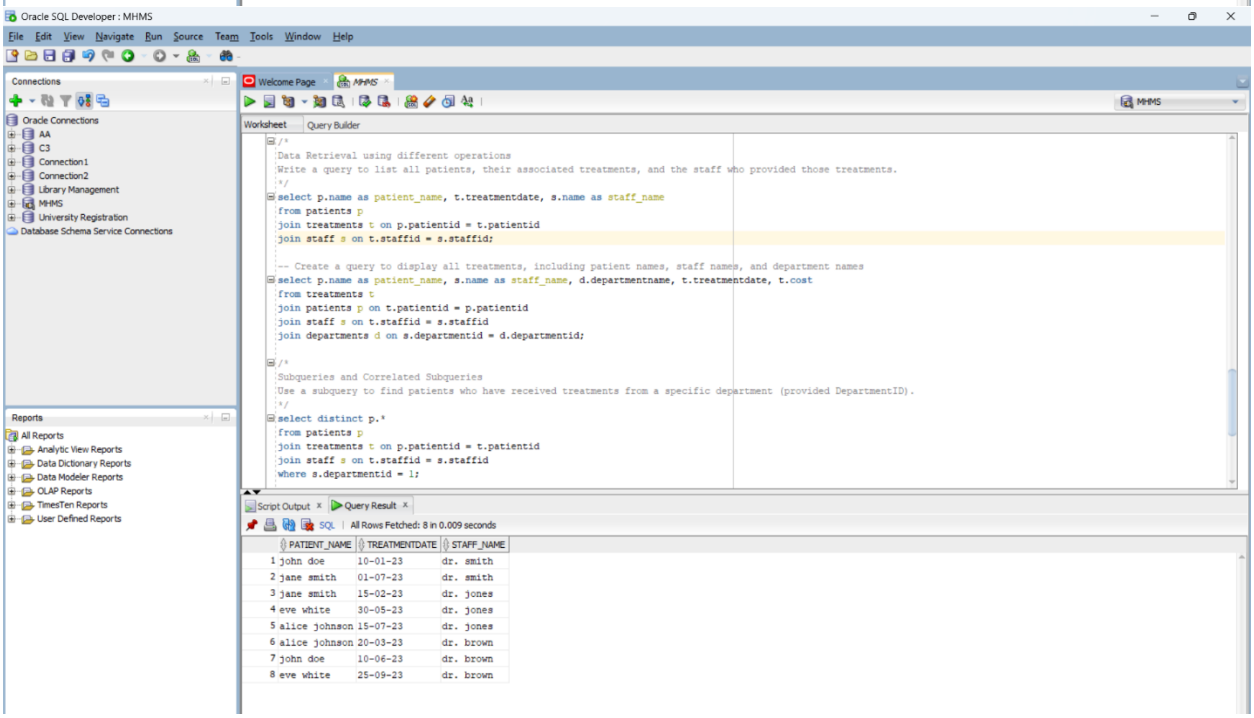

Oracle SQL Developer : MHMS
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
C3
Connection1
Connection2
Library Management
MHMS
University Registration
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
MHMS
Worksheet
Query Builder
-- Create Tables: patients, departments, staff, treatments
-- Patients table
create table patients (
patientid number primary key,
name varchar2(100),
dob date,
email varchar2(100) unique,
phone varchar2(20)
);
-- Departments table
create table departments (
departmentid number primary key,
departmentname varchar2(100)
);
-- Staff table
create table staff (
staffid number primary key,
name varchar2(100),
role varchar2(50),
departmentid number,
constraint fk_staff_department foreign key (departmentid) references departments(departmentid)
);
Script Output
Task completed in 0.052 seconds
Table PATIENTS created.
Table DEPARTMENTS created.
Table STAFF created.
Table TREATMENTS created.

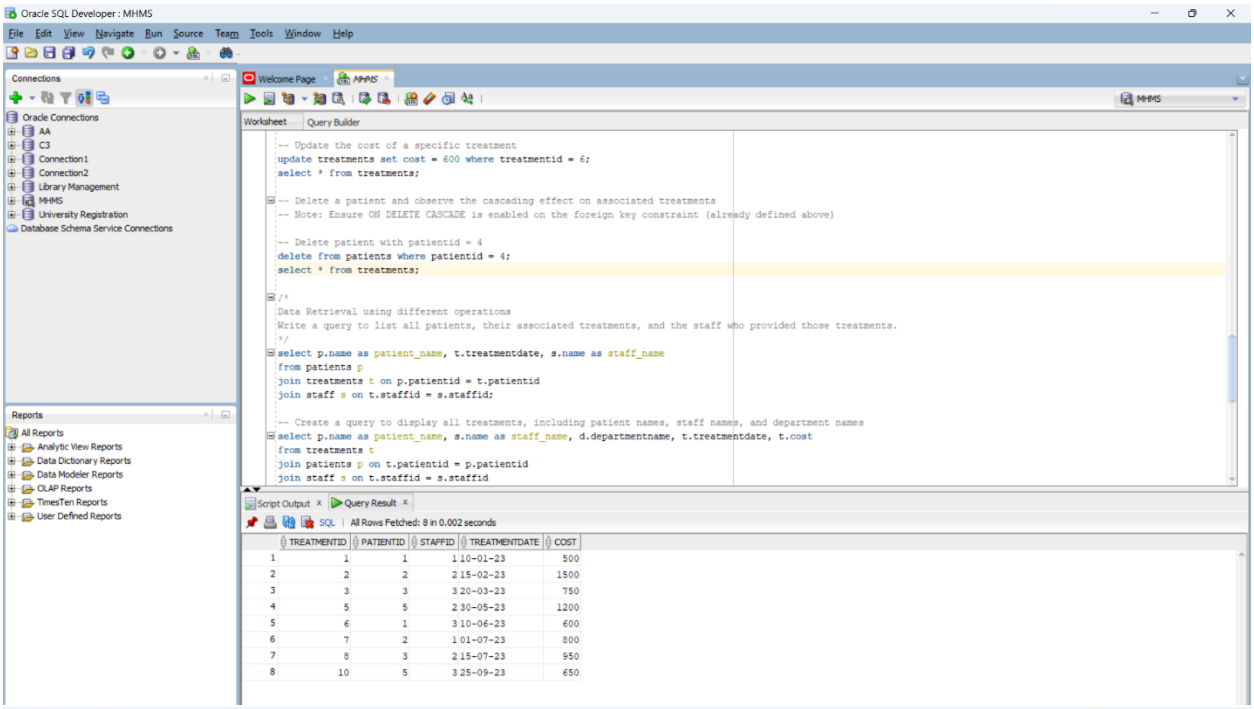

Oracle SQL Developer : MHMS
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
C3
Connection1
Connection2
Library Management
MHMS
University Registration
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
MHMS
Worksheet
Query Builder
/*
Subqueries and Correlated Subqueries
Use a subquery to find patients who have received treatments from a specific department (provided DepartmentID).
*/
select distinct p.*
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid
where s.departmentid = 1;
-- Use a correlated subquery to find staff who have treated more than a specified number of patients (e.g., 3)
select *
from staff s
where (select count(*) from treatments t where t.staffid = s.staffid) > 2;
/*
Aggregate Functions with GROUP BY and HAVING
Calculate the total cost of treatments per patient and filter out patients with a total treatment cost less than 1000
*/
select p.name as patient_name, sum(t.cost) as total_cost
from patients p
join treatments t on p.patientid = t.patientid
group by p.name
having sum(t.cost) >= 1000;
Script Output
Query Result
SQL
All Rows Fetched: 4 in 0.009 seconds
PATIENT_NAME
TOTAL_COST
1 john doe 1100
2 jane smith 2300
3 alice johnson 1700
4 eve white 1850

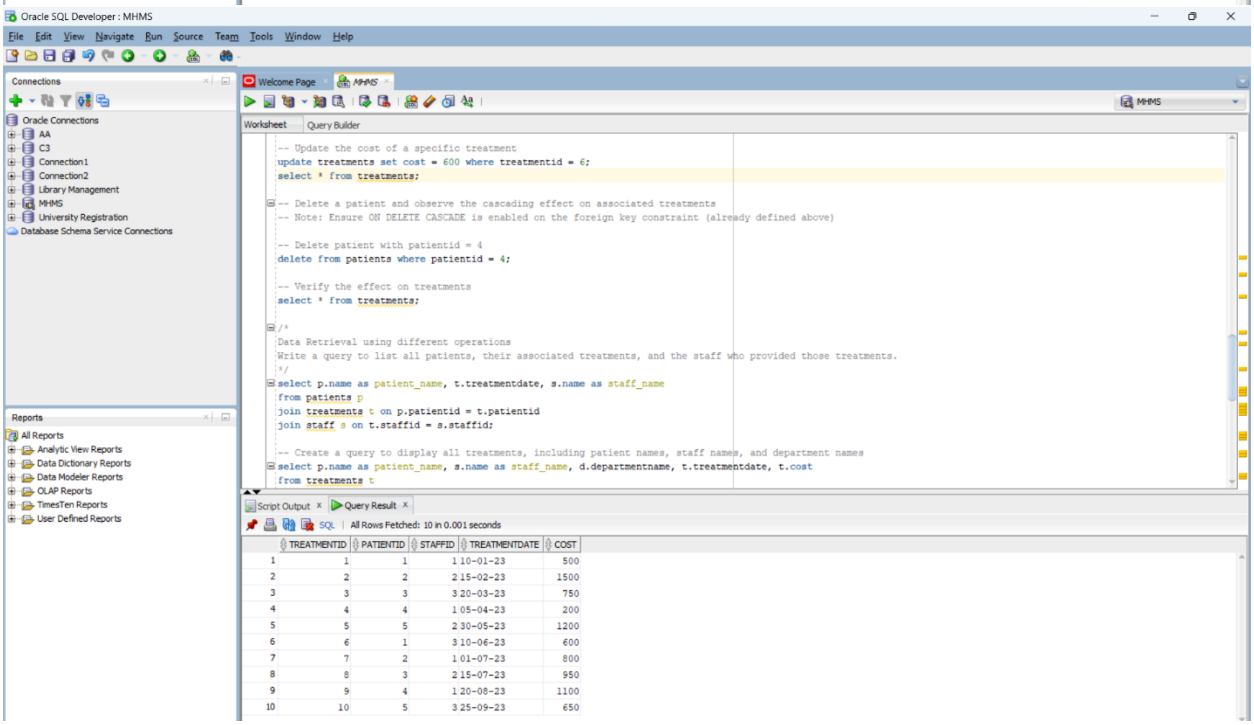

Oracle SQL Developer : MHMS
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
C3
Connection1
Connection2
Library Management
MHMS
University Registration
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
MHMS
Worksheet
Query Builder
/*
Subqueries and Correlated Subqueries
Use a subquery to find patients who have received treatments from a specific department (provided DepartmentID).
*/
select distinct p.*
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid
where s.departmentid = 1;
-- Use a correlated subquery to find staff who have treated more than a specified number of patients (e.g., 3)
select *
from staff s
where (select count(*) from treatments t where t.staffid = s.staffid) > 2;
/*
Aggregate Functions with GROUP BY and HAVING
Calculate the total cost of treatments per patient and filter out patients with a total treatment cost less than 1000
*/
select p.name as patient_name, sum(t.cost) as total_cost
from patients p
join treatments t on p.patientid = t.patientid
group by p.name
having sum(t.cost) >= 1000;
Script Output
Query Result
SQL | All Rows Fetched: 2 in 0.004 seconds
STAFFID NAME ROLE DEPARTMENTID
1 2 dr. jones neurologist 2
2 3 dr. brown orthopedic surgeon 3


Oracle SQL Developer : MHMS
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
C3
Connection1
Connection2
Library Management
MHMS
University Registration
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
MHMS
Worksheet
Query Builder
Data Retrieval using different operations
Write a query to list all patients, their associated treatments, and the staff who provided those treatments.
*/
select p.name as patient_name, t.treatmentdate, s.name as staff_name
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid;
-- Create a query to display all treatments, including patient names, staff names, and department names
select p.name as patient_name, s.name as staff_name, d.departmentname, t.treatmentdate, t.cost
from treatments t
join patients p on t.patientid = p.patientid
join staff s on t.staffid = s.staffid
join departments d on s.departmentid = d.departmentid;
/*
Subqueries and Correlated Subqueries
Use a subquery to find patients who have received treatments from a specific department (provided DepartmentID).
*/
select distinct p.*
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid
where s.departmentid = 1;
-- Use a correlated subquery to find staff who have treated more than a specified number of patients (e.g., 3)
Script Output
Query Result
SQL | All Rows Fetched: 2 in 0.011 seconds
PATIENTID NAME DOB EMAIL PHONE
1 1 john doe 15-05-85 john.doe@example.com 555-0101
2 2 jane smith 20-06-90 jane.smith@example.com 555-0102


Oracle SQL Developer : MHMS
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
C3
Connection1
Connection2
Library Management
MHMS
University Registration
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
MHMS
Worksheet
Query Builder
/*
Data Retrieval using different operations
Write a query to list all patients, their associated treatments, and the staff who provided those treatments.
*/
select p.name as patient_name, t.treatmentdate, s.name as staff_name
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid;
-- Create a query to display all treatments, including patient names, staff names, and department names
select p.name as patient_name, s.name as staff_name, d.departmentname, t.treatmentdate, t.cost
from treatments t
join patients p on t.patientid = p.patientid
join staff s on t.staffid = s.staffid
join departments d on s.departmentid = d.departmentid;
/*
Subqueries and Correlated Subqueries
Use a subquery to find patients who have received treatments from a specific department (provided DepartmentID).
*/
select distinct p.*
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid
where s.departmentid = 1;
Script Output
Query Result
SQL | All Rows Fetched: 8 in 0.016 seconds
PATIENT_NAME STAFF_NAME DEPARTMENTNAME TREATMENTDATE COST
1 john doe dr. smith cardiology 10-01-23 500
2 john doe dr. brown orthopedics 10-06-23 600
3 jane smith dr. jones neurology 15-02-23 1500
4 jane smith dr. smith cardiology 01-07-23 800
5 alice johnson dr. brown orthopedics 20-03-23 750
6 alice johnson dr. jones neurology 15-07-23 950
7 eve white dr. jones neurology 30-05-23 1200
8 eve white dr. brown orthopedics 25-09-23 650


Oracle SQL Developer : MHMS
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
C3
Connection1
Connection2
Library Management
MHMS
University Registration
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
MHMS
Worksheet
Query Builder
/*
Data Retrieval using different operations
Write a query to list all patients, their associated treatments, and the staff who provided those treatments.
*/
select p.name as patient_name, t.treatmentdate, s.name as staff_name
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid;
-- Create a query to display all treatments, including patient names, staff names, and department names
select p.name as patient_name, s.name as staff_name, d.departmentname, t.treatmentdate, t.cost
from treatments t
join patients p on t.patientid = p.patientid
join staff s on t.staffid = s.staffid
join departments d on s.departmentid = d.departmentid;
/*
Subqueries and Correlated Subqueries
Use a subquery to find patients who have received treatments from a specific department (provided DepartmentID).
*/
select distinct p.*
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid
where s.departmentid = 1;
Script Output
Query Result
SQL | All Rows Fetched: 8 in 0.009 seconds
PATIENT_NAME TREATMENTDATE STAFF_NAME
1 john doe 10-01-23 dr. smith
2 jane smith 01-07-23 dr. smith
3 jane smith 15-02-23 dr. jones
4 eve white 30-05-23 dr. jones
5 alice johnson 15-07-23 dr. jones
6 alice johnson 20-03-23 dr. brown
7 john doe 10-06-23 dr. brown
8 eve white 25-09-23 dr. brown


Oracle SQL Developer : MHMS
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
C3
Connection1
Connection2
Library Management
MHMS
University Registration
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
MHMS
Worksheet
Query Builder
-- Update the cost of a specific treatment
update treatments set cost = 600 where treatmentid = 6;
select * from treatments;
-- Delete a patient and observe the cascading effect on associated treatments
-- Note: Ensure ON DELETE CASCADE is enabled on the foreign key constraint (already defined above)
-- Delete patient with patientid = 4
delete from patients where patientid = 4;
select * from treatments;
/*
Data Retrieval using different operations
Write a query to list all patients, their associated treatments, and the staff who provided those treatments.
*/
select p.name as patient_name, t.treatmentdate, s.name as staff_name
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid;
-- Create a query to display all treatments, including patient names, staff names, and department names
select p.name as patient_name, s.name as staff_name, d.departmentname, t.treatmentdate, t.cost
from treatments t
join patients p on t.patientid = p.patientid
join staff s on t.staffid = s.staffid
Script Output
Query Result
SQL | All Rows Fetched: 8 in 0.002 seconds
TREATMENTID PATIENTID STAFFID TREATMENTDATE COST
1 1 1 1 10-01-23 500
2 2 2 2 15-02-23 1500
3 3 3 3 20-03-23 750
4 5 5 2 30-05-23 1200
5 6 1 3 10-06-23 600
6 7 2 1 01-07-23 800
7 8 3 2 15-07-23 950
8 10 5 3 25-09-23 650


Oracle SQL Developer : MHMS
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
C3
Connection1
Connection2
Library Management
MHMS
University Registration
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
MHMS
Worksheet
Query Builder
-- Update the cost of a specific treatment
update treatments set cost = 600 where treatmentid = 6;
select * from treatments;
-- Delete a patient and observe the cascading effect on associated treatments
-- Note: Ensure ON DELETE CASCADE is enabled on the foreign key constraint (already defined above)
-- Delete patient with patientid = 4
delete from patients where patientid = 4;
-- Verify the effect on treatments
select * from treatments;
/*
Data Retrieval using different operations
Write a query to list all patients, their associated treatments, and the staff who provided those treatments.
*/
select p.name as patient_name, t.treatmentdate, s.name as staff_name
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid;
-- Create a query to display all treatments, including patient names, staff names, and department names
select p.name as patient_name, s.name as staff_name, d.departmentname, t.treatmentdate, t.cost
from treatments t
Script Output
Query Result
SQL | All Rows Fetched: 10 in 0.001 seconds
TREATMENTID PATIENTID STAFFID TREATMENTDATE COST
1 1 1 1 10-01-23 500
2 2 2 2 15-02-23 1500
3 3 3 3 20-03-23 750
4 4 4 1 05-04-23 200
5 5 5 2 30-05-23 1200
6 6 1 3 10-06-23 600
7 7 2 1 01-07-23 800
8 8 3 2 15-07-23 950
9 9 4 1 20-08-23 1100
10 10 5 3 25-09-23 650