



ADVANCED DATABASE PROJECT

INFORMATION ABOUT DOCTOR TABLE

```
CREATE TABLE DOCTORS  
(DOCTOR_ID VARCHAR2(15) PRIMARY KEY,  
FIRST_NAME VARCHAR2(20) NOT NULL,  
LAST_NAME VARCHAR2(20) NOT NULL,  
SPECIALIZATION VARCHAR2(50) NOT NULL);
```

```
DESCRIBE DOCTORS;
```

Name	Null?	Type
DOCTOR_ID	NOT NULL	VARCHAR2 (15)
FIRST_NAME	NOT NULL	VARCHAR2 (20)
LAST_NAME	NOT NULL	VARCHAR2 (20)
SPECIALIZATION	NOT NULL	VARCHAR2 (50)

INFORMATION ABOUT DOCTOR TABLE

INSERT INTO DOCTORS

VALUES(2183783721,'MASOUD','ALGHONIEMY','DERMATOLOGIST')

INSERT INTO DOCTORS

VALUES(2181649758,'NOHA','OTHMAN','FAMILY PHYSICIAN')

INSERT INTO DOCTORS

VALUES(2183671640,'ASHRAF','ELRAGAL','NEUROLOGIST')

INSERT INTO DOCTORS

VALUES(2181167891,'AMR','HAMDY','SURGEON')

INSERT INTO DOCTORS

VALUES(2188734572,'AYMAN','SHERIF','SURGEON')

INSERT INTO DOCTORS

VALUES(2183678014,'MONA','LOTFY','ONCOLOGIST')

Connecting the Python with the DATABASE

```
import cx_Oracle
```

```
# Replace these values with your actual Oracle database credentials
```

```
username = "SYSTEM"
```

```
password = "1234"
```

```
hostname = "localhost"
```

```
port = "1521"
```

```
service_name = "xe"
```

```
# Connection string format for Oracle
```

```
dsn_tns = cx_Oracle.makedsn(hostname, port, service_name=service_name)
```

Connecting the Python with the DATABASE

```
# Establish a connection to the Oracle database
try:
    connection = cx_Oracle.connect(user=username, password=password,
dsn=dsn_tns)
    print("Connected to Oracle Database")
```

```
# Perform database operations here
```

```
# For example, you can create a cursor and execute a query
cursor = connection.cursor()
cursor.execute("SELECT * FROM DOCTORS")
rows = cursor.fetchall()
```

Connecting the Python with the DATABASE

```
# Print the fetched data
for row in rows:
    print(row)

except cx_Oracle.Error as error:
    print("Error connecting to Oracle Database:", error)

finally:
    # Close the cursor and connection in the finally block to ensure they are
    # always closed
    if 'connection' in locals():
        connection.close()
    print("Connection closed")
```

Connecting the Python with the DATABASE

```
Connected to Oracle Database
```

```
('2183783721', 'Masoud', 'alghoniemy', 'Dermatologist')
('2181649758', 'Noha', 'Othman', 'Family physician')
('2183671640', 'Ashraf', 'Elragal', 'Neurologist')
('2181167891', 'Amr', 'Hamdy', 'Surgeon')
('2188734572', 'Ayman', 'Sherif', 'Surgeon')
('2183678014', 'Mona', 'Lotfy', 'Oncologist')
('2183071304', 'Hesham', 'Kotb', 'Dentist')
('2183257901', 'Abdelsalam', 'Khalid', 'Surgeon')
('2186913267', 'Samir', 'Hosny', 'Optometrist')
('2187536249', 'Engy', 'Hasan', 'Psychiatrist')
('2184369725', 'Ahmed', 'Ibrahim', 'Dentist')
```

```
Connection closed
```

GUI for doctor table

```
def fetch_Doctors():
    connection = cx_Oracle.connect(user=username, password=password,
dsn=dsn_tns)
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM DOCTORS")
    Patients = cursor.fetchall()
    connection.close()
    return Patients

def insert_Doctors(d_id, fname, lname, specialization):
    connection = cx_Oracle.connect(user=username, password=password,
```

GUI for doctor table

```
dsn=dsn_tns)

cursor = connection.cursor()
cursor.execute("INSERT INTO DOCTORS VALUES(:1, :2, :3, :4)", (d_id, fname,
lname, specialization))
connection.commit()
connection.close()
```

```
def Delete_Doctors(d_id):
    connection = cx_Oracle.connect(user=username, password=password,
dsn=dsn_tns)
    cursor = connection.cursor()
    cursor.execute("DELETE FROM DOCTORS WHERE DOCTOR_ID = :id", {'id': d_id})
    connection.commit()
```

GUI for doctor table

```
connection.close()
```

```
def update_Doctors(new_fname,new_lname,new_specialization,d_id):  
    connection = cx_Oracle.connect(user=username, password=password,  
    dsn=dsn_tns)  
    cursor = connection.cursor()  
    cursor.execute("UPDATE DOCTORS SET FIRST_NAME = :1, LAST_NAME = :2,  
    SPECIALIZATION = :3 WHERE DOCTOR_ID = :4",  
    (new_fname, new_lname, new_specialization, d_id))  
    connection.commit()  
    connection.close()
```

GUI for doctor table

```
def ID_exist(d_id):  
    connection = cx_Oracle.connect(user=username, password=password,  
    dsn=dsn_tns)  
    cursor = connection.cursor()  
    cursor.execute('SELECT COUNT(*) FROM DOCTORS WHERE DOCTOR_ID = :id',  
    {'id': d_id})  
    result = cursor.fetchone()  
    connection.close()  
    return result[0] > 0
```

GUI for doctor table

```
def delete():
    selected_item = tree.focus()
    if not selected_item:
        messagebox.showerror('Error','Choose a product to delete')
    else:
        id = id_entry.get()
        Delete_Doctors(id)
        add_to_treeview()
        clear()
        messagebox.showinfo('Success','Data has been deleted.')
```

GUI for doctor table

```
def update():
    selected_item = tree.focus()
    if not selected_item:
        messagebox.showerror('Error', 'Choose a product to update.')
    else:
        id = id_entry.get()
        fname= fname_entry.get()
        lname= lname_entry.get()
        spec = spec_entry.get()
        update_Doctors(fname,lname, spec, id)
        add_to_treeview()
        clear()
        messagebox.showinfo('Success', 'Data has been updated.')
```

GUI for doctor table

```
def display_data(event):
    selected_item = tree.focus()
    if selected_item:
        row = tree.item(selected_item)['values']
        clear()
        id_entry.insert(0,row[0])
        fname_entry.insert(0,row[1])
        lname_entry.insert(0,row[2])
        spec_entry.insert(0,row[3])
    else:
        pass
```

GUI for doctor table

```
def add_to_treeview():
    doctors = fetch_Doctors()
    tree.delete(*tree.get_children())
    for doctor in doctors:
        tree.insert('', END, values=doctor)
def clear(*clicked):
    if clicked:
        tree.selection_remove(tree.focus())
        tree.focus('')
        id_entry.delete(0,END)
        fname_entry.delete(0,END)
        lname_entry.delete(0,END)
        spec_entry.delete(0,END)
```

GUI for doctor table

```
def insert():
    id = id_entry.get()
    fname= fname_entry.get()
    lname= lname_entry.get()
    spec = spec_entry.get()
    if not (id and fname and lname and spec):
        messagebox.showerror('Error', 'Enter all fields.')
    elif ID_exist(id):
        messagebox.showerror('Error', 'ID already exists.')
    else:
        insert_Doctors(id,fname,lname,spec)
        add_to_treeview()
        clear()
        messagebox.showinfo( 'Success','Data has been inserted.)
```

GUI for doctor table

```
app=customtkinter.CTk()  
app.title("Doctor System")  
app.geometry('900x420')  
app.config(bg='#0A0B0C')  
app.resizable(False,False)
```

```
title_label=customtkinter.CTkLabel(app , text ="Doctor info" ,  
text_color="#fff",bg_color='#0A0B0C')  
title_label.place(x=35,y=15)
```

```
frame=customtkinter.CTkFrame(app,bg_color='#0A0B0C',fg_color='#0A0B0C'  
,corner_radius=10 , border_width=2 , border_color='#fff' , width=300,height=300  
frame.place(x=25,y=45)
```

GUI for doctor table

```
id_label=customtkinter.CTkLabel(app , text ="ID" ,  
text_color="#fff",bg_color='#0A0B0C')  
id_label.place(x=60,y=50)
```

```
id_entry=customtkinter.CTkEntry(app,text_color='#000',fg_color='#fff',border_c  
olor="#B2016C",border_width=1 , width=100)  
id_entry.place(x=150,y=50)
```

```
fname=customtkinter.CTkLabel(app , text ="FIRST NAME" ,  
text_color="#fff",bg_color='#0A0B0C')  
fname.place(x=40,y=100)  
fname_entry=customtkinter.CTkEntry(app,text_color='#000',fg_color='#fff',borc  
er_color="#B2016C",border_width=1 , width=100)  
fname_entry.place(x=150,y=100)
```

GUI for doctor table

```
lname=customtkinter.CTkLabel(app , text ="LAST NAME" ,  
text_color="#fff",bg_color='#0A0B0C')  
lname.place(x=40,y=150)
```

```
lname_entry=customtkinter.CTkEntry(app,text_color'#000',fg_color='#fff',bord  
er_color="#B2016C",border_width=1 , width=100)  
lname_entry.place(x=150,y=150)  
spec=customtkinter.CTkLabel(app , text ="SPECIALIZATION" ,  
text_color="#fff",bg_color='#0A0B0C')  
spec.place(x=40,y=200)  
spec_entry=customtkinter.CTkEntry(app,text_color'#000',fg_color='#fff',bord  
er_color="#B2016C",border_width=1 , width=100)  
spec_entry.place(x=150,y=200)
```

GUI for doctor table

```
add_button=customtkinter.CTkButton(frame,text_color='#000',command=inser  
text='Add',fg_color='#047E43',hover_color='#025B30',border_color="#B2016C",  
bg_color='#1B1B2B', cursor='hand2',corner_radius=8, width=120)  
add_button.place(x=20,y=220)
```

```
clear_button=customtkinter.CTkButton(frame,text_color='#000',command=lam  
da:clear(True),text='Clear',fg_color='#047E43',hover_color='#025B30',border_co  
lor="#B2016C",bg_color='#1B1B2B', cursor='hand2',corner_radius=8, width=120)  
clear_button.place(x=150,y=220)
```

```
update_button=customtkinter.CTkButton(frame,command=update,text_color='#  
000',text='Update',fg_color='#047E43',hover_color='#025B30',border_color="#B  
2016C",bg_color='#1B1B2B', cursor='hand2',corner_radius=8, width=120)  
update_button.place(x=20,y=250)
```

GUI for doctor table

```
delet_button=customtkinter.CTkButton(frame,command=delete,text_color="#000",text='Delete',fg_color="#047E43",hover_color="#025B30",border_color="#B2010C",bg_color="#1B1B2B", cursor='hand2',corner_radius=8, width=120)  
delet_button.place(x=150,y=250)
```

```
style=ttk.Style(app)  
style.theme_use('default')  
style.configure("Treeview",foreground='fff',background='#0A0B0C',fieldbackground='#1B1B21')  
style.map("Treeview",background=[('selected','#0A0B0C')])
```

```
tree=ttk.Treeview(app,height=18)
```

GUI for doctor table

```
tree['columns'] = ('ID', 'First name', 'Last name', 'Specialization')
tree.column('#0', width=0, stretch=tk.NO)
tree.column('ID', anchor=tk.CENTER, width=150)
tree.column('First name', anchor=tk.CENTER, width=150)
tree.column('Last name', anchor=tk.CENTER, width=150)
tree.column('Specialization', anchor=tk.CENTER, width=150)
tree.heading('ID', text='ID')
tree.heading('First name', text='First name')
tree.heading('Last name', text='Last name')
tree.heading('Specialization', text='Specialization')
tree.place(x=500,y=49)
tree.bind('<ButtonRelease>',display_data)
add_to_treeview()
app.mainloop()
```

Doctor System

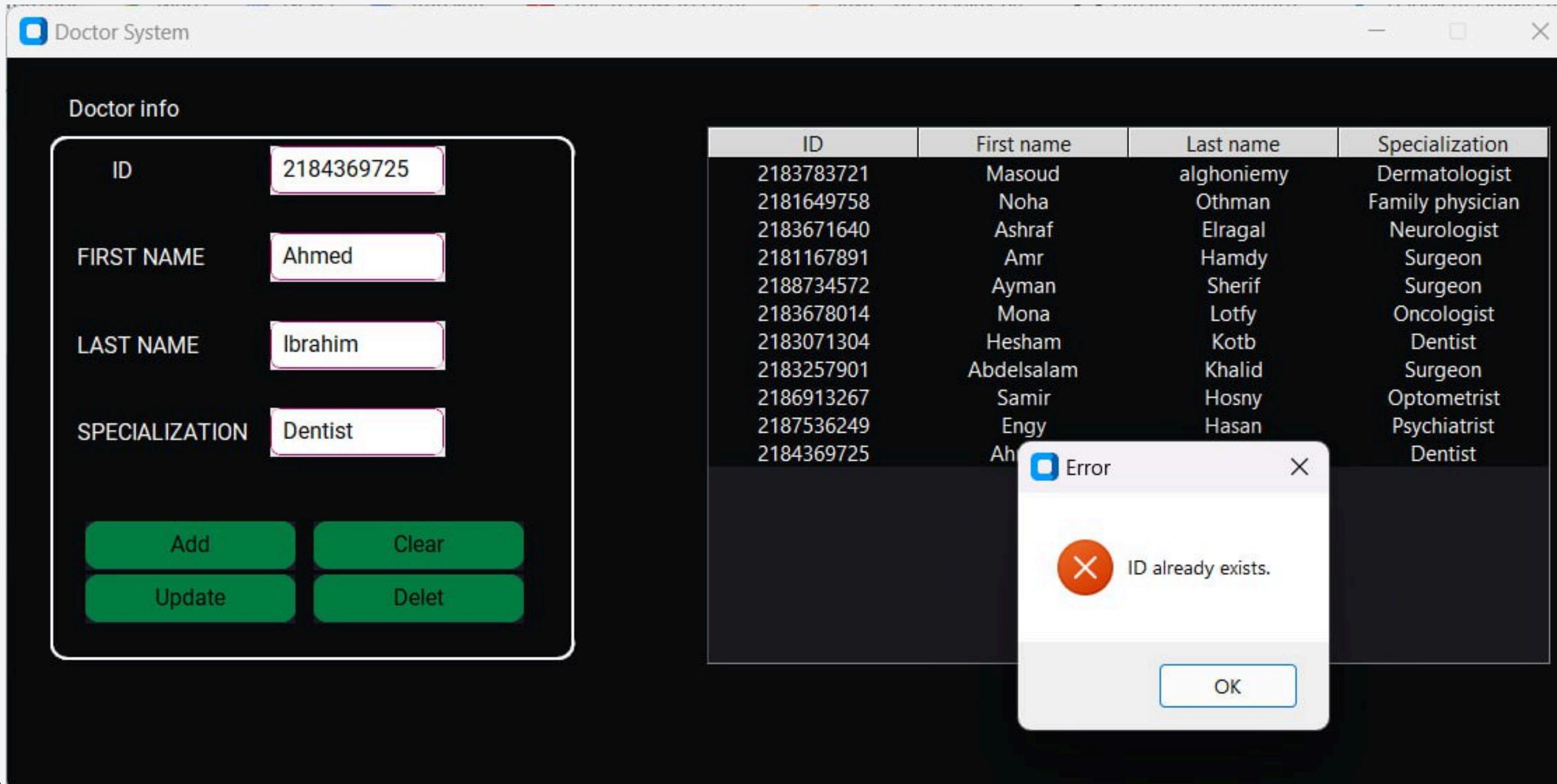
Doctor info			
ID	2184369725	First name	Masoud
FIRST NAME	Ahmed	Last name	alghoniemy
LAST NAME	Ibrahim	Specialization	Dermatologist
SPECIALIZATION	Dentist		
Add	Clear		
Update	Delet		

ID	First name	Last name	Specialization
2183783721	Masoud	alghoniemy	Dermatologist
2181649758	Noha	Othman	Family physician
2183671640	Ashraf	Elragal	Neurologist
2181167891	Amr	Hamdy	Surgeon
2188734572	Ayman	Sherif	Surgeon
2183678014	Mona	Lotfy	Oncologist
2183071304	Hesham	Kotb	Dentist
2183257901	Abdelsalam	Khalid	Surgeon
2186913267	Samir	Hosny	Optometrist
2187536249	Engy	Hasan	Psychiatrist
2184369725	Ahmed	Ibrahim	Dentist

- Add Button: is a button used to insert new data to the doctor table
- Clear Button: is a button used to clear the data we inserted
- Update Button: is a button used to update data in the doctor table
- Delete Button: is a button used to delete data from the doctor table.

GUI for doctor table

We can't insert doctor id that's already exist



Indexing and Optimization

```
CREATE INDEX index_doctor ON DOCTORS(specialization);
```

```
Index INDEX_DOCTOR created.
```

Indexing and Optimization

SQL | All Rows Fetched: 3 in 0.005 seconds

	DOCTOR_ID	FIRST_NAME	LAST_NAME	SPECIALIZATION
1	2181167891	Amr	Hamdy	Surgeon
2	2188734572	Ayman	Sherif	Surgeon
3	2183257901	Abdelsalam	Khalid	Surgeon

SQL | All Rows Fetched: 3 in 0.002 seconds

	DOCTOR_ID	FIRST_NAME	LAST_NAME	SPECIALIZATION
1	2181167891	Amr	Hamdy	Surgeon
2	2188734572	Ayman	Sherif	Surgeon
3	2183257901	Abdelsalam	Khalid	Surgeon

Time Complexity decreases
because of indexing

Indexing and Optimization

```
select patients.first_name,  
       patients.last_name,  
       patients.Age,  
       diseases.Disease,  
       doctors.specialization  
from patients  
join diseases  
    on patients.patients_id =  
diseases.patient_id  
join doctors  
    on patients.doctor_id =  
doctors.doctor_id;
```

#	FIRST_NAME	LAST_NAME	AGE	DISEASE	SPECIALIZATION
1	Omar	Nabil	12	Glaucoma	Optometrist
2	Hesham	Anwar	9	allergy	Dermatologist
3	Mahmoud	Gomaa	41	Appendix Tumors	Surgeon
4	Rania	Waleed	22	Tooth decay	Dentist
5	Atef	Ahmed	36	Schizophrenia	Psychiatrist
6	Gehad	Ali	29	Gallstones	Surgeon
7	Manar	Talal	44	Blepharitis	Optometrist
8	Amira	Saed	44	Brain Tumors	Neurologist
9	Mohamed	Awad	62	Thyroid Cancer	Surgeon
10	Salma	Yousef	32	Depression	Psychiatrist

Indexing and Optimization

FIRST_NAME	LAST_NAME	AGE	DISEASE	SPECIALIZATION
Mahmoud	Gomaa	41	Appendix Tumors	Surgeon
Mohamed	Awad	62	Thyroid Cancer	Surgeon

```
select patients.first_name,  
       patients.last_name,  
       patients.Age,  
       diseases.Disease,  
       doctors.specialization  
  from patients  
  join diseases  
    on patients.patients_id = diseases.patient_id  
  join doctors  
    on patients.doctor_id = doctors.doctor_id  
 where doctors.SPECIALIZATION ='Surgeon' and patients.gender='Male';
```

Indexing and Optimization

FIRST_NAME	LAST_NAME	AGE	DISEASE	SPECIALIZATION
Mahmoud	Gomaa	41	Appendix Tumors	Surgeon
Mohamed	Awad	62	Thyroid Cancer	Surgeon

```
select patients.first_name, patients.last_name , patients.Age,  
      diseases.Disease,  
      doctors.specialization  
from patients  
join diseases  
  on patients.patients_id = diseases.patient_id  
join doctors  
  on patients.doctor_id = doctors.doctor_id  
where doctors.SPECIALIZATION ='Surgeon' and patients.gender='Male';
```

in this query we put doctors.SPECIALIZATION='Surgeon' before patients.gender='Male' because the number of surgeons less than male patients

Indexing and Optimization

```
SELECT PATIENTS.FIRST_NAME,  
       PATIENTS.LAST_NAME,  
       BILLS.BILL,  
       BILLS.NUMBER_OF_DAYS  
FROM PATIENTS  
JOIN PAY  
    ON pay.patients_id = patients.patients_id  
JOIN BILLS  
    ON pay.bill_number = bills.bill_number  
ORDER BY bills.bill;
```

	FIRST_NAME	LAST_NAME	BILL	NUMBER_OF_DAYS
1	Rania	Waleed	100	1
2	Manar	Talal	150	1
3	Atef	Ahmed	160	1
4	Omar	Nabil	400	2
5	Salma	Yousef	450	1
6	Hesham	Anwar	689	1
7	Amira	Saed	790	6
8	Gehad	Ali	2500	8
9	Mohamed	Awad	4000	13
10	Mahmoud	Gomaa	10000	15

THANKS