A PROJECT REPORT

ON

“PATIENT DOCTOR SCHEDULING”

Submitted By:

VEDSAR KUSHWAHA

Enroll. Id – 53515

HPES – Oracle Summer Training 2012

INDEX

1. PROJECT DESCRIPTION
2. SCOPE OF THE PROJECT
3. ENTITIES AND OBJECTS
4. ENTITY-RELATIONSHIP DIAGRAM
5. CONSTRAINTS
6. DATABASE STRUCTURE
7. REPORT ON NORMAL FORMS
8. DATABASE SCHEMA
9. SAMPLE QUERIES

**PREFACE**

Patient-Doctor Scheduling is a simple interactive PL/SQL developed program that leads you to manage a number of patents. With the help of this program, you not only schedule the doctor’s timing with patient but also help in estimating the required time to deal with a patient, assigning appropriate doctor for the patient. The scheduling engine is so much intelligent that it automatically assigned the patient to the doctor according to the treatment the patient requires. This PL/SQL developed program is designed in such a way that it doesn’t require any front end support. The program is designed with as less as possible interaction with patients so that whole scheduling will done almost automatically. This application supports multiple user interaction and can be accessed from anywhere. Concurrent users are handled efficiently and the respective errors are stored in a separate database table, which can be rectified easily.

**PROJECT DESCRIPTION**

This project is used to schedule the patient with the doctors automatically by considering the disease the patient is suffering from.

The patient arrived at hospital, will be allotted a Patient ID which would be assigned to him unless the disease is procured.

The DBA of the hospital can also see the records of the patient’s information, their timing when they get allotted with the doctor, from which doctor they get treatment etc.

The whole data is saved until or unless the DBA delete it.

This project is designed to run automatically, so only a few management peoples are required to handle the application as well as the database.

**SCOPE OF THE PROJECT**

This project can be used in any hospital to make the job of management much easier. This project only requires an oracle engine to make database and to run the PL/SQL programs.

An elementary knowledge of Database Administration is required to manage this PL/SQL developed application. The application requires only basics configured system. This application can also be accessed through different computers. So anyone from anywhere can schedule his/her timing with doctors.

**ENTITIES AND OBJECTS**

TABLES AND THEIR ATTRIBUTES

1. DOC\_PROFILE
   1. DOC\_ID
   2. F\_NAME
   3. L\_NAME
   4. EMAIL
   5. RESIDENTIAL ADDRESS
   6. PHONE
2. PATIENT\_INFO
   1. P\_ID
   2. F\_NAME
   3. L\_NAME
   4. EMAIL
   5. ADDRESS
   6. PHONE
3. ID\_DISEASE
   1. ID
   2. DISEASE
4. DOC\_ID\_DISEASE
   1. DOC\_ID
   2. DISEASE\_ID
5. T\_TIME
   1. NAME
   2. ID
   3. TIME
   4. P\_QUE
6. DOC\_AVL
   1. BUSY
   2. DOC\_ID
7. DOC\_SCH
   1. DOC\_ID
   2. BSY\_TIME\_START
   3. BSY\_TIME\_END
   4. P\_ID
8. ERR
   1. ERCODE
   2. ERRMSG

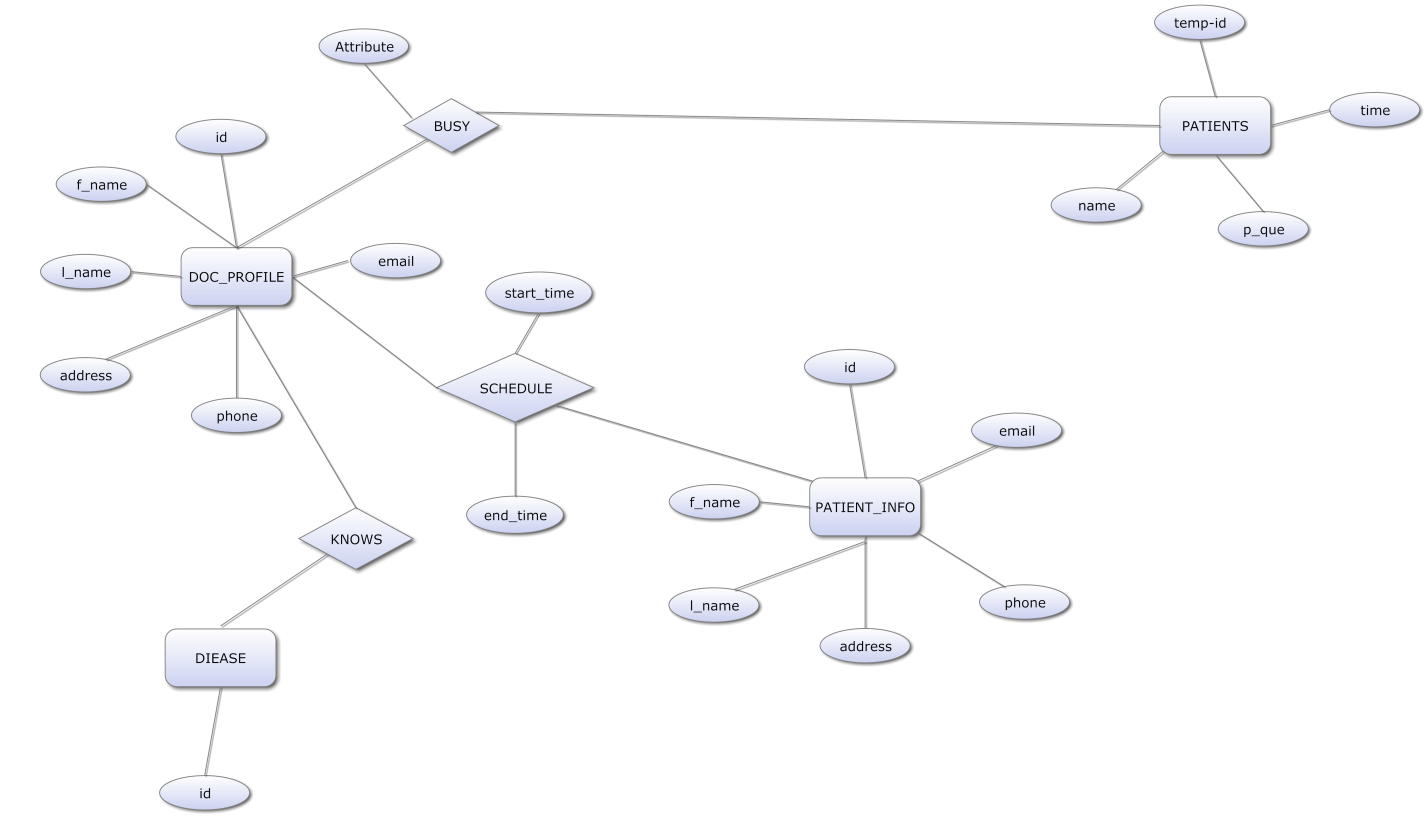
SEQUENCE

1. DISEASE\_ID\_SEQUENCE
2. DOC\_ID\_SEQUENCE
3. PATIENT\_QUE\_SEQUENCE
4. TEMP\_PID\_SEQUENCE

PROCEDURES AND FUNCTIONS

1. GET\_UID\_FUNCTION
2. INSERT\_PATIENT\_INFO\_PROCEDURE

**E-R DIAGRAM**



**CONSTRAINTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CONSTRAINT\_NAME** | **CONSTRAINT\_TYPE** | **TABLE NAME** | **DESCRIPTION** |
| DOC\_AVL\_RF\_DOC\_PROF | FOREIGN KEY | DOC\_AVL | FOREIGN KEY CONSTRAINT TO DOC\_ID COLUMN OF DOC\_PROFILE |
| DOC\_ID\_DISEASE\_RF\_DOC\_PROF | FOREIGN KEY | DOC\_ID\_DISEASE | FOREIGN KEY CONSTRAINT TO DOC\_ID COLUMN OF DOC\_PROFILE |
| DOC\_ID\_DISEASE\_RF\_ID\_DISEASE | FOREIGN KEY | DOC\_ID\_DISEASE | FOREIGN KEY CONSTRAINT TO ID COLUMN OF ID\_DISEASE |
| DOC\_PROFILE\_PK | PRIMARY KEY | DOC\_PROFILE | PRIMARY KEY FOR DOC\_PROF TABLE |
| DOC\_SCH\_RF\_DOC\_PROF | FOREIGN KEY | DOC\_SCH | FOREIGN KEY CONSTRAINT TO DOC\_ID COLUMN OF DOC\_PROFILE |
| DOC\_SCH\_PK | PRIMARY KEY | DOC\_SCH | PRIMARY KEY FOR DOC\_SCH TABLE |
| ID\_DISEASE\_PK | PRIMARY KEY | ID\_DISEASE | PRIMARY KEY FOR ID\_DISEASE TABLE |
| PATIENT\_INO\_PK | PRIMARY KEY | PATIENT\_INFO | PRIMARY KEY FOR PATIENT\_INFO TABLE |
| T\_TIME\_PK | PRIMARY KEY | T\_TIME | PRIMARY KEY FOR T\_TIME TABLE |
| P\_QUE\_UNQ | UNIQUE | T\_TIME | UNIQUE KEY FOR T\_TIME TABLE |

**DATABASE STRUCTURE**

|  |
| --- |
| **DOC\_AVL** |
| BUSY: number |
| DOC\_ID: number(fk) |

|  |
| --- |
| **DOC\_ID\_DISEASE** |
| DISEASE: number(fk) |
| DOC\_ID: number(fk) |

|  |
| --- |
| **DOC\_PROFILE** |
| ID: number(pk) |
| F\_NAME: varchar2(50) |
| L\_NAME: varchar2(50) |
| EMAIL: varchar2(100) |
| ADDRESS: varchar2(150) |
| PHONE: number |

|  |
| --- |
| **DOC\_SCH** |
| DOC\_ID: number(fk)(pk) |
| START\_TIME: date |
| END\_TIME: date |
| P\_ID: number(fk)(pk) |

|  |
| --- |
| **ID\_DISEASE** |
| ID: number(pk) |
| DISEASE: varchar2(50) |

|  |
| --- |
| **PATIENT\_INFO** |
| ID: number(pk) |
| F\_NAME: varchar2(50) |
| L\_NAME: varchar2(50) |
| EMAIL: varchar2(100) |
| ADDRESS: archar2(150) |
| PHONE: number |

|  |
| --- |
| **T\_TIME** |
| NAME: rchar2(50) |
| T\_ID: number(pk) |
| TIME: date |
| P\_QUE: number |

**NORMALIZATION**

Normalization is a process of removing inconsistency, redundancy and data duplicity from a database. This involves breaking up of tables in to smaller forms. A table schema in well proper normal form not only reduces inconsistencies and duplicacy but also provides a way for better management and a high level of security.

The entire table used in these schemas is already normalized to their best level. So there is no need to normalize it further.

**DATABASE SCHEMAS**

|  |
| --- |
| **DOC\_PROFILE** |
| ID: number(pk) |
| F\_NAME: varchar2(50) |
| L\_NAME: varchar2(50) |
| EMAIL: varchar2(100) |
| ADDRESS: varchar2(150) |
| PHONE: number |

|  |
| --- |
| **PATIENT\_INFO** |
| ID: number(pk) |
| F\_NAME: varchar2(50) |
| L\_NAME: varchar2(50) |
| EMAIL: varchar2(100) |
| ADDRESS: varchar2(150) |
| PHONE: number |

SCHEDULED

|  |
| --- |
| **ID\_DISEASE** |
| ID: number(pk) |
| DISEASE: varchar2(50) |

TREATMENT HAS

|  |
| --- |
| **T\_TIME** |
| NAME: rchar2(50) |
| T\_ID: number(pk) |
| TIME: date |
| P\_QUE: number |

HAS

**QUERY SAMPLES**

1. **ENTRY BY PATIENT**

Please Enter Your Name vedsar

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*Welcome Mr.vedsar\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*Complete the Registration Form\*\*\*\*\*\*\*\*\*\*

Please Enter Your First Name vedsar

Please Enter Your Last Name kushwaha

Please Enter Your Email vedsarkushwaha@gmail.com

Please Enter Your Residential Address tyiuot

Please Enter Your Phone 876543

Thank you for the registration

1 d1

2 d2

3 d3

4 d4

5 d5

6 d6

7 d7

8 d8

9 d9

10 d10

Please enter the Code of Disease 8

Doctor Allotted Currently

1. **TO CHECK THE AVAILABLE DCOTORS**

SQL> select \* from doc\_avl;

BUSY DOC\_ID

---------- ----------

0 1

0 2

0 3

0 4

0 5

1 6

0 7

0 8

1. 9
2. 10
3. **TO CHECK THE SCHEDULE ALLOTED**

SQL> select \* from doc\_sch;

DOC\_ID BSY\_TIME\_ BSY\_TIME\_ P\_ID

---------- --------- --------- ----------

6 08-JUL-12 08-JUL-12 186080712

10 08-JUL-12 08-JUL-12 2810080712

1. **FINALLY EXIT QUESRY FROM THE HOSPITAL**

SQL> @d:\exit

Please enter you Unique ID186080712

THANK YOU FOR COMING TO US.

SQL> select \* from doc\_sch;

1. **AGAIN CHECK THE SCHEDULE**

DOC\_ID BSY\_TIME\_ BSY\_TIME\_ P\_ID

---------- --------- --------- ----------

10 08-JUL-12 08-JUL-12 2810080712

1. **CHECK THE AVAILABILITY OF DOCTORS**

SQL> select \* from doc\_avl;

BUSY DOC\_ID

---------- ----------

0 1

0 2

0 3

0 4

0 5

0 6

0 7

0 8

0 9

1 10