



Our database consist of 9 tables;

P_Adress ;

AdressID, PatientID, Area, Street, Building, Flat, City

Patient ;

PatientID , Name, Surname, Phone,Birthdate,Gender

Bill ;

BillID, TreatmentID, PatientID , BillDate , Deadline , Amount

Bed;

BedID, TreatmentID , DepartmentID , EntryDate

Department;

DepartmentID,Name,Bed (to store how many available bed in that department)

Rezervation;

RezervationID,StaffId,PatientID,RezervationDate

S_Adress;

AdressID,StaffID,Area,Street,Building,Flat,City

Staff;

Id,DepartmentID,Name,Surname,Phone,Bithdate,Gender,Salary,Position

Treatment;

Id,StaffID,PatientID,BillID,Illness,Explenation,Startdate,Bedstatus

Explanation;

In real life databases can be implemented in many different aspects according to customer's requests. Before starting database design, software developer would contact with customer and can learn which data should be stored and how.

According to information from customer database implementation starts and until it's completed database design can be changed by software developer and customer many times according to technical reasons and business needs.

Since we didn't have customer at the beginning and prepared this database to fulfill course purposes (implementing; triggers,procedures,views,keys,squences,insertion queries) , this aspect is not the best one of course.

2-) KEYS

BED;

-Primary key for Bed ID

-Foreign Key for Department Id referenced from Department table

- Foreign Key for Treatment Id referenced from Treatment table (null is allowed because there are beds without patient)

-Unique values for bed ID

/Patient id would be also useful in this table according to customer's needs.

BILL;

-Primary key for Bill ID

-Foreign Key for Patient Id referenced from Patient table

- Foreign Key for Treatment Id referenced from Treatment table

-Unique values for Bill ID

/Staff id would be also useful in this table according to customer's needs.

DEPARTMENT;

- Primary key for Department ID
- Unique values for Department ID

PATIENT_ADDRESS;

- Primary key for Patient Address ID
- Foreign Key for Patient Id referenced from Patient table
- Unique values for Patient Address ID

PATIENT;

- Primary key for Patient ID
- Unique values for Patient ID

REZERVATION;

- Primary key for Rezervation ID
- Foreign Key for Patient Id referenced from Patient table
- Foreign Key for Staff Id referenced from Staff table
- Unique values for Rezervation ID

STAFF ADDRESS;

- Primary key for Staff Address ID
- Foreign Key for Staff Id referenced from Staff table
- Unique values for Staff Address ID

STAFF;

- Primary key for Staff ID
- Foreign Key for Department Id referenced from Department table

-Unique values for Staff ID

TREATMENT;

-Primary key for Treatment ID

-Foreign Key for Patient Id referenced from Patient table

- Foreign Key for Staff Id referenced from Staff table

- Foreign Key for Bill Id referenced from Bill table (is allowed null because not every treatment has bills yet)

-Unique values for Treatment ID

Explanation;

Some foreign keys would be added to make more flexible database according to business needs.

All of the primary keys have sequences.

3-) INDEXES

All of the primary keys have automatic clustered indexes.

Additional non-clustered indexes in order to maximize searching performance;

- Bill_Treatment_IDX on Bill in order to search bill by treatment id.

-Bill_Patient_IDX on Bill in order to search bill by patient id.

- P_Address_Area_IDX on P_Address in order to search address by area.

- P_Address__IDX on P_Address in order to search address by patient id.

- Rezervation_Staff__IDX on Rezervation in order to search rezervation by staff id.

- Rezervation_Patient__IDX on Rezervation in order to search rezervation by patient id.

-Rezervation_Date__IDX on Rezervation in order to search rezervation by date.

-S_Address__IDX on S_Address in order to search address by staff id.

- S_Dep__IDX on Staff in order to search staffs by department id.

- Treatment_Staff__IDX on Treatment in order to search treatment by staff id.

- Treatment_Patient_IDX on Treatment in order to find treatment by patient id.

Explanation;

9 default indexes from primary keys and 11 extra indexes are implemented.

In total we have 20 index for searching performance, it could be increased, decreased or we can merge multiple columns for an index due to business needs.

4-INSERTION QUERIES

-insertDepartment

Takes variables as; @name varchar(150),@bed int

-insertStaff

Takes variables as; @Dep_Id int,@Name varchar(200),@SName varchar(100),@Phone varchar(20),@BDate date,@Gender varchar(20),@Salary bigint,@Position varchar(100)

-insertPatient

Takes variables as; @Name varchar(200),@SName varchar(100),@Phone varchar(20),@BDate date,@Gender varchar(15)

-insertPAddress

Takes variables as; @Id int,@Area varchar(200),@Street varchar(200),@Building varchar(100),@Flat varchar(50),@City varchar(50)

-insertSAddress

Takes variables as; @Id int,@Area varchar(200),@Street varchar(200),@Building varchar(100),@Flat varchar(50),@City varchar(50)

-insertRez

Takes variables as; @id1 int,@id2 int,@date Date

-insertTreatment

Takes variables as; @id1 int,@id2 int,@Ill varchar(200),@Exp varchar(max),@Bed varchar(50)

Date value declared in the query automatically from current date.

Bill id set as null

-AddBill

Takes variables as; @Tid int,@Deadline date,@Amount bigint

Date value declared in the query automatically from current date.

We update the treatment table with bill id.

If patient was in bed, we also update bed status because if we produce bill, it's mean patient leaves the hospital.

5-) PROCEDURES AND FUNCTIONS

-Find_Pid

Finds Treatment_Id of the patient if the patient in Hospital. Because we don't have patient_id in the bed table we needed this function.

- Doctor_Rezervation

It checks doctor's reservation on the given date. If date given as null, it will find current date's reservations.

- Patient_Treatment

It finds all treatments by the given patient id

-Patient_Bill

It finds all bills by the given patient id

- Profit_By_Doctor_Monthly

It calculates profit of the doctor on specific month.

- Find_Bed

It finds bed id by patient id. In this procedure we have called Find_Pid to locate treatment id.

- Change_Bed_Status

When patient lay in bed or leave hospital , this procedure updates bed status, available bed in the department and bed table with new data.

Explanation;

More procedures and functions would be implemented due to business needs.

In total we have 7 procedures and functions.

6-) VIEWS

- [Yearly_Income]

It brings table of yearly income earned by all staffs

- [Patient_Information]

It brings all of the information belongs to patients

- [Staff_Information]

It brings all of the information belongs to staffs

- [Monthly_Expenses]

It brings table of monthly expenses spent for staffs

-[Yearly_Expenses]

It brings table of yearly expenses spent for staffs

7-) TRIGGERS

- [Add_Bed_TR]

The trigger add beds in the bed table after we insert a department on department table.

- [Set_Bed_To_Patient]

The trigger updates available bed on department table and set a bed for patient when we enter "yes" for the bed status in treatment table.

- [Quarentined_Illness]

The trigger brings table of the people who need to be quarantined after we use words as (corona, quarantine, quarentina, covid) in the illness or explanation columns in the treatment table.

8-) CONCLUSION

how to design relational database, how to alter tables and columns due to business needs and technical reasons.

how to implement triggers,views,functions.

how to use indexes to increase searchin performance.

how to use sequences in 2 different ways.