**WolfHospital Database Management System**

For a hospital in North Carolina

CSC 540 Database Systems

Project Report #2

Project Team Member:

Xiaoting Fu, Yifan Zhao, Xiao Ma, Xuejiao Fang

# Assumptions and Problem Statement

Assumptions

1. One ward can only be assigned to one patient, no matter how many beds the ward has (similar to hotel booking schema, 1-bed-room and 2-beds rooms can only be booked by one customer).
2. Each patient needs to check-in and be assigned a ward no matter if the patient going to stay overnight or not
3. One nurse can be responsible for multiple wards and when assigning a nurse to a ward the system will select a nurse who has the least number ward that he/she is in charge of.
4. The ward can be used for patient to rest or doctor to give treatment
5. Staffs can be subdivided into Doctors, Nurses and Operators
6. Only Doctors have professional title
7. One check-in can accept many treatments, each treatment generates a medical record.
8. When a patient arrives at the hospital, he will provide his requirement for a ward, if the ward exists, he will create an account otherwise his account is not created.
9. When creating a new patient account, his billing account created at the same time.
10. A billing record is created in the following three cases: when a new medical record is created, when a registration fee is generated and when the accommodation fee is generated.

# **1) Database Schema**

* Patients(PID, SSN, Name, DOB, Gender, Age, Status, Address, PhoneNum)
* Wards(WNum, Capacity, ChargePerDay)
* Staff(SID, Name, Age, Gender, JobTitle, ProfTitle, Department, Address, PhoneNum, Type)
* BillingAccounts(PID, BillingAddr, CreatedDate, Balance)
* BillingRecords(RID, RecordType, CreatedDate)
* PaymentMethods(PMID, PMName, CardNum, CardValidDate)
* MedicalRecord(MID, StartDate, EndDate, Prescription, Diagnosis)
* MedRecordOf(PID,MID)
* ResponsibleFor(MID, SID)
* CheckIn(PID, WNum, StartDate, EndDate, BedNum)
* AssignedTo(SID, WNum)
* Select(PMID, PID)
* Add(RID, PID)

The schemas are all in 3NF because all entities have unique IDs associated with them which makes it easier to identify a specific tuple.

All functional dependencies are obvious because of the IDs associated with all entities.

*Patients*(PID, SSN, Name, DOB, Gender, Age, Status, Address, PhoneNum)

The only FD is **PID->SSN, Name, DOB, Gender, Age, Status, Address, PhoneNum** which holds because each individual patient, identified by a unique PID, has an SSN, name, date of birth, gender, age, status, address and phone number.

This relation is in BCNF because PID is a key.

*Wards*(WNum, Capacity, ChargePerDay)

The only FD is **WNum->Capacity, ChargePerDay** holds because each ward, identified by a unique ward number, has capacity and the charge per day.

This relation is in BCNF because WNum is a key.

*Staff*(SID, Name, Age, Gender, JobTitle, ProfTitle, Department, Address, PhoneNum, Type)

The only FD is **SID -> Name, Age, Gender, JobTitle, ProfTitle, Department, Address, PhoneNum, Type** holds because each staff, identified by a unique SID, has a name, age, gender, job title, professional title, department, address, phone number, type.

This relation is in BCNF because SID is a key.

*BillingAccounts*(PID, BillingAddr, CreatedDate, Balance)

The only FD is **PID -> BillingAddr, CreatedDate, Balance** holds because each patient is identified by the unique PID, has a billingAddr, CreatedDate and balance.

This relation is in BCNF because PID is a key.

*BillingRecords*(RID, RecordType, CreatedDate)

The only FD is **RID->RecordType, CreatedDate** holds because each billing record is identified by the unique RID, and has a record type and the date the record created.

This relation is in BCNF because RID is a key.

*PaymentMethods*(PMID, PMName, CardNum, CardValidDate)

**PMID -> PMName, CardNum, CardValidDate** holds because each payment method is identified by both PMID, and has PMName, CardNum, and CardValidDate.

This relation is in BCNF because PMID and PID are keys.

*MedicalRecord*(MID, StartDate, EndDate, Prescription, Diagnosis)

**MID ->StartDate, EndDate, Prescription, Diagnosis** holds because each medical record is identified by MID, and has StartDate, EndDate, Prescription, Diagnosis.

This relation is in BCNF because MID is a key.

*CheckIn*(PID, WNum, StartDate, EndDate, BedNum)

**PID, WNum -> StartDate, EndDate, BedNum**

This relation is in BCNF because PID and WNum are keys.

*AssignedTo*(SID, WNum)

**SID, WNum -> SID, WNum**

This relation is in BCNF (and thus 3NF) because it contains two key attributes.

*ResponsibleFor*(MID, SID)

**MID, SID -> MID, SID**

This relation is in BCNF (and thus 3NF) because it contains three key attributes.

*Select*(PMID, PID)

**PMID, PID-> PMID, PID**

This relation is in BCNF (and thus 3NF) because it contains one key attribute.

*Add*(RID, PID)

**RID, PID -> RID, PID**

This relation is in BCNF (and thus 3NF) because it contains two key attributes.

*MedRecordOf*(PID,MID)

**PID, MID -> PID, MID**

This relation is in BCNF (and thus 3NF) because it contains two key attributes.

# **2) Design Decisions**

The mechanical approach was used to create the global schema with a few exceptions.

* Each entity set was made into a relation with the same set of attributes
* Relationships were replaced by a relation whose attributes are the keys for the connected entity sets

All types of staff including operators, nurses, doctors are distinguished by the “type” attribute.

Many-to-one relationships were combined with other relations. Combining relations in this way makes it more efficient to answer queries that involve attributes of one relation than to answer queries involving attributes of several relations.

* combine *BillingRecords* and *Add* and add *PID* as an attribute in *BillingRecords.*
* combine *PaymentMethods* and *Select* and add *PID* as an attribute in *PaymentMethods.*
* combine *MedicalRecords* and *MedRecordOf* and add *PID* as an attribute in *MedicalRecords.*
* combine *MedicalRecords* and *ResponsibleFor*, Teatxnand add *SID* as an attribute in *MedicalRecords.*
* combine *Wards* and *AssignedTo* and add *SID* as an attribute in *Wards.*

***Patients*(PID, SSN, Name, DOB, Gender, Age, Status, Address, PhoneNum)**

* PID (Primary Key) – unique identifier
* SSN (Can be NULL) - not all patient has a SSN
* Name (NOT NULL) - identification and diagnosis purposes
* DOB (NOT NULL) - identification and diagnosis purposes
* Gender (NOT NULL) - identification and diagnosis purposes
* Age (NOT NULL) - identification and diagnosis purposes
* Status (NOT NULL) - must be one of the followings: “processing treatment plan”, “in ward”, “completing treatment”
* Address (NOT NULL) - contact purpose
* PhoneNum (NOT NULL) - contact purpose

***Wards*(WNum, SID, Capacity, ChargePerDay)**

* WNum (Primary Key) – unique identifier
* SID (Referential Integrity) - refer to other entities within the database (Staff)
* Capacity (NOT NULL) - identification purposes
* ChargePerDay (NOT NULL) - identification purposes, has default value

***Staff*(SID, Name, Age, Gender, JobTitle, ProfTitle, Department, Address, PhoneNum, Type)**

* SID (Primary Key) – unique identifier
* Name (NOT NULL) - identification purpose
* Age (NOT NULL) - identification purpose
* Gender (NOT NULL) - identification purpose
* JobTitle (NOT NULL) - a staff must have a job title (doctor, nurse, billing staff, etc)
* ProfTitle (Can be NULL) – a staff such as a billing staff may not have a professional title
* Department (Can be NULL) - a staff may be not assigned a department
* Address (NOT NULL) - contact purpose
* PhoneNum (NOT NULL) - contact purpose
* Type (Primary Key) - must be one of the followings: “nurse”, “doctor”, “operator”

***BillingAccounts*(PID, BillingAddr, CreatedDate, Balance)**

* PID (Referential Integrity, Primary Key) – unique identifier that refer to other entities within the database (Patients)
* BillingAddr (NOT NULL) - contact purpose
* CreatedDate (NOT NULL) - assigned current date when an account is created
* Balance (NOT NULL) - has default 0.0

***BillingRecords*(RID, PID, RecordType, CreatedDate)**

* RID (Primary Key) – unique identifier
* PID (Referential Integrity) – refer to other entities within the database (Patients)
* RecordType (NOT NULL) - must be one of the followings: “medical”, “registration”, “accommodation”
* CreatedDate (NOT NULL) - assigned current date when a record is created

***PaymentMethods*(PMID, PID, SID, PMName, CardNum, CardValidDate)**

* PMID (Primary Key) – unique identifier
* PID (Referential Integrity) – refer to other entities within the database (Patients)
* SID (Referential Integrity) – that refer to other entities within the database (Staff)
* PMName (NOT NULL) - a payment method must have a name
* CardNum (NOT NULL) - must be a valid card number for successful payments
* CardValidDate (NOT NULL) - must be a valid card valid date for successful payments

***MedicalRecord*(MID, PID,SID, StartDate, EndDate, Prescription, Diagnosis)**

* MID (Primary Key) – unique identifier
* PID (Referential Integrity) – refer to other entities within the database (Patients)
* SID (Referential Integrity) – refer to other entities within the database (Staff)
* StartDate (NOT NULL) - must be assigned a start date when created
* EndDate (NOT NULL) - must be assigned an end date when created
* Prescription (Can be NULL) - can be filled in later
* Diagnosis (Can be NULL) - can be filled in later

***CheckIn*(PID, WNum, StartDate, EndDate, BedNum)**

* PID (Primary Key) – unique identifier
* WNum (Referential Integrity, Primary Key) – unique identifier that refer to other entities within the database (Wards)
* StartDate (NOT NULL) - must be assigned a start date when created
* EndDate (NOT NULL) - must be assigned an end date when created
* BedNum(NOT NULL) - must be assigned the desired bed number

# **3) SQL Statements**

# **4) Interactive SQL Queries**

Note: All result sets from select statements have been formatted for readability.

**4.1 Queries for tasks and operations**

**4.2 Autotrace and Indexes for two tables**

**4.3 Query correctness proofs**