

## **Maternity Clinic Database**

### **Executive Summary**

Our maternity clinic has decided to store information about the clinic in a database in order to easily track the admissions, visits, scheduling, and treatments including some basic medical history on patients and babies born at the clinic. First, we formulate the data requirements for our database and designed an entity relationship (ER) diagram which shows relationships between each entity involved in the maternity clinic as well as their attributes. We have eight entities, eight relationships and three 'ISA' relationships. There were multiple constraints that could not be captured by our entity relational diagram. The ER diagram was translated into a relational database including table definition (DDL) using MySQL.

Our database will enable viewers to answer various sample queries including the ones listed in this paper. There were several assumptions made regarding attributes and relationships. There were also several shortcomings as our database fails to capture some information relevant to the patients' health history, the babies, and billing.

### **Domain Description**

The application domain we decided on is a maternity clinic. The maternity clinic needs a database to be able to have information on patient-doctor relationships, baby tracker as well as managing room assignment and scheduling visits. As multiple aspects of the patient data must be shared for various tasks, this data is often shared via paper internally in the clinic. This manner of storage often leads to hardships with regards to how the data is managed, updated and shared. In addition, various governmental and professional bodies request periodic surveys to be done on the clinic's patients. Currently, this is done by sifting through hundreds of papers for all the patients and counting them manually. With a centralized database, it could potentially take a single query to get such information.

### **Sample Queries**

- 1) How many visits are scheduled for Doctor A on March 12, 2019?
- 2) How many female babies were born under Doctor Z?
- 3) On which day(s) did we have the highest number of visits in March 2019?
- 4) What are the details of patients D from her latest visit?
- 5) How many patients have had at least two deliveries in the clinic in the past ten years?
- 6) Which nurse(s) cared for patient V during her admission between January 24, 2018 and January 26, 2018?
- 7) Which day had the most number of deliveries in the last five years?
- 8) What is the emergency contact information of the mother of the baby at the nursery ward bed number A1026?
- 9) Which doctor gets the most visits from patients referred to the clinic by a friend?
- 10) What are the most popular reasons for scheduling an appointment? walk-in?

## **Data Requirements**

The data requirements are as follows:

- Each **staff** working at the hospital has a record of a unique staff ID (sid: primary key) assigned to him/her by HR when he/she was hired, a first name, a last name, gender, date of birth, address, a phone number, email, emergency contact phone number, whether they are full-time workers, the date of joining, and the date of exit.
- Doctors, nurses, and admins are staff. In addition to the staff's attributes:
  - Each **nurse** has a unique license number and license expiry date (Since medical licenses need to be renewed every five years)
  - Each **doctor** has a unique license number
  - Each **admin** has a unique admin ID assigned to him/her when he/she is hired/appointed as an admin.
- Each **patient** has a record of a unique patient ID (pid: primary key) assigned to him/her by an admin when he/she first visited the clinic (through admission, walk-in, appointment), a first name, a last name, gender which is set to *female* as default as maternity clinic mainly deals with female patients, date of birth, unique insurance number, blood type, phone number, emergency contact name, emergency contact phone number, emergency contact's relationship to the patient, and who referred the patient to the clinic, which is set to *self* as default.
- Each **baby** has a record of a unique baby ID (bib: primary key) assigned at birth by an admin, patient ID of their birth mother, gender, weight, birthmark, blood type, delivery type, datetime of delivery, condition at delivery, assigned bed number, datetime of discharge, and extra comment (notes).
- Each **visit** has a unique visit number generated through auto-increment (v\_number: primary key), admin ID, patient ID, staff ID (specifically ID of doctor), type of visit (if it is walk-in or appointment), reasons for visit, scheduled datetime, and checked in datetime.
- Each **admission** has a unique admission number generated through auto-increment (a\_number: primary key), admin ID of the admin managing the admission, patient ID (pid), datetime of admission, reasons for patient's admission, room number, datetime of discharge, and extra comment (notes).

## **Relationships**

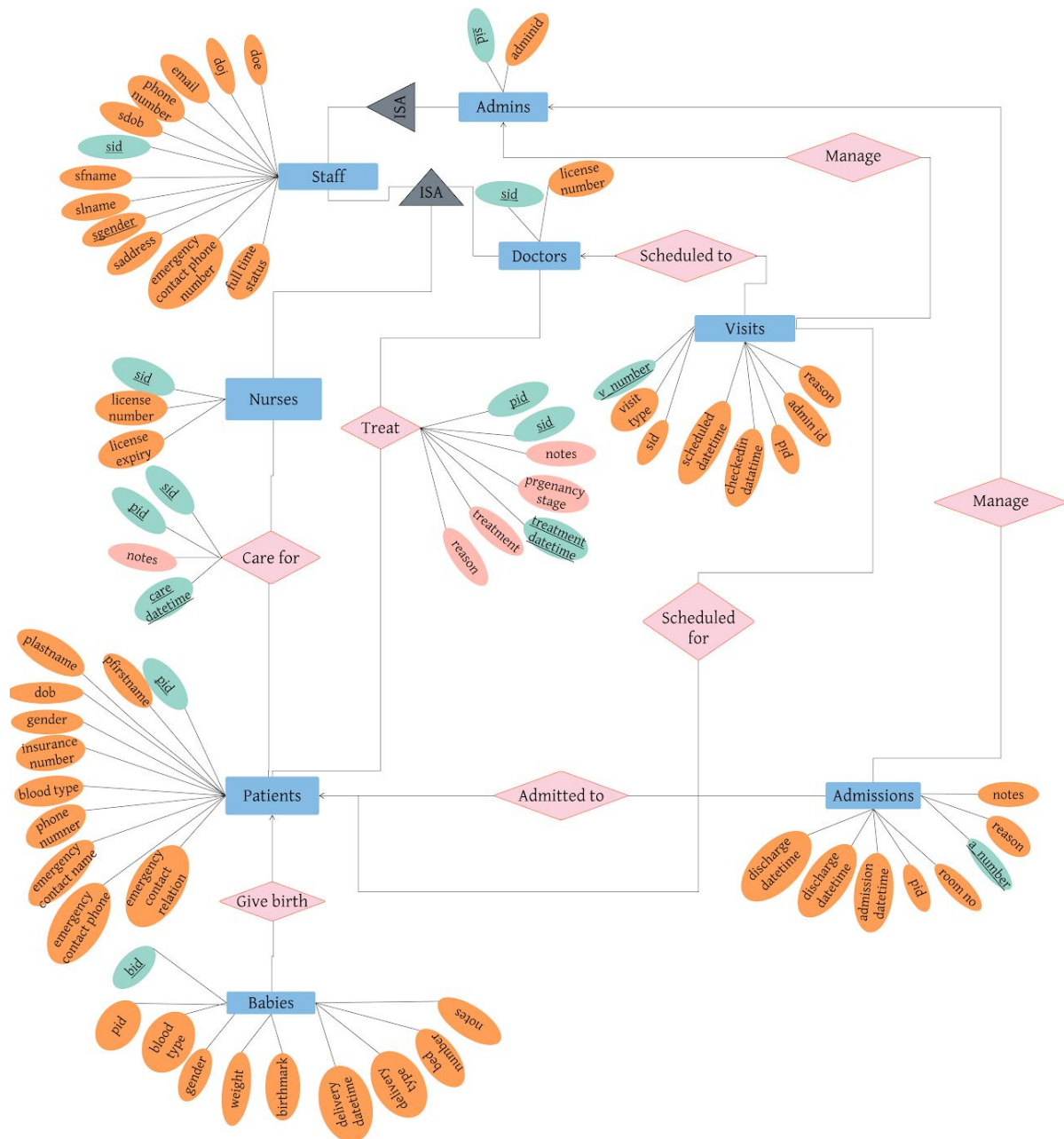
- Each patient may **be cared for** by several nurses, and each nurse may **care for** several patients. Each interaction between a nurse and a patient (**care for** record) has:
  - patient ID (pid), staff ID (for nurse), datetime of care, and extra comment (notes). The primary key is (pid, sid and datetime of care).
- Each patient may be **treated** by several doctors. Each doctor may **treat** at most one patient at a given datetime, and several patients over time. Each interaction between a doctor and a patient (**treat** record) has:
  - patient ID (pid), staff ID (for doctor), stage of pregnancy, treatment, reason (ex. miscarriage, fertility treatment, pregnancy), datetime of visit, and extra comment (notes). The primary key is (pid, sid and datetime of visit).

- Each baby **is birthed** by one patient. Each patient **gives birth** to at most one baby at a given datetime, and may have several babies over time. We assume that twins, triplets, quadruplets etc... are not born at the exact same time.
- Each visit is/has been **managed by** an admin and each admin **manages/has managed** multiple visits.
- Each visit is **scheduled to** one doctor and **scheduled for** one patient. Each doctor can have at most one visit at a given datetime and several visits over time. Similarly, each patient gets to schedule or check-in at most for one visit at a given datetime and several visits over time.
- Each admission is **managed by** one admin and each admin **manages/has managed** several admissions.
- Each admission **is matched** to one patient. Each patient can have at most one admission at a given datetime and several admissions over time.

#### **Shortcomings of our Maternity Clinic Database:**

- Information about babies' fathers is not captured.
- Both staff and patients' middle names and titles are omitted in our database.
- Babies are not considered as patients at birth. Hence, we are not able to track which nurses are in charge of which babies, as nurses take turns being in charge of the nursery ward.
- After being discharged, if a baby comes back to the clinic, then he/she will be treated as patient and entered in the patients' entity.
- We are not tracking billing information in this project.
- Previous health history or health history at other hospitals or clinics are not captured in our database.
- We are assuming that the patient's room pre and post surgery is the same
- We are not storing images of patients as well as scannings and x-rays
- Discharge datetime for both patients and babies can be NULL (default value) as admin may not know the datetime of discharge when they create an admission record. Similarly exit date for staff can be left at their default value "NULL"

## Entity-Relationship Diagram



## Requirements not Captured by the Entity Relationship Diagram

- Each nurse has a unique license number
- Each doctor has a unique license number
- Each admin has a unique admin ID

- Each patient has a unique insurance number
- Each patient gives birth to at most to one baby at a given datetime
- Each doctor can have at most one scheduled visit at a given datetime
- Each doctor can have at most one checked-in visit at a given datetime
- Each doctor may treat at most one patient at a given datetime
- Each patient gets to schedule or check-in at most for one visit at a given datetime
- Each patient can have at most one admission at a given datetime

### **Relational Model: Set of Relations with all their Constraints in DDL**

#### **Entities:**

```
CREATE TABLE `staff` (
  `sid` int(11) NOT NULL,
  `sfname` varchar(45) NOT NULL,
  `slname` varchar(45) NOT NULL,
  `sgender` varchar(5) NOT NULL,
  `sDoB` date NOT NULL,
  `saddress` varchar(45) NOT NULL,
  `sphone_number` varchar(20) NOT NULL,
  `semail` varchar(45) NOT NULL,
  `semergency_contact_phone_number` varchar(20) NOT NULL,
  `full_time` tinyint(4) NOT NULL DEFAULT '1',
  `sjoin_date` date NOT NULL,
  `sleft_date` date DEFAULT NULL,
  PRIMARY KEY (`sid`))

CREATE TABLE `doctors` (
  `sid` int(11) NOT NULL,
  `license_number` varchar(20) NOT NULL,
  PRIMARY KEY (`sid`),
  UNIQUE KEY `unique_license_number` (`license_number`),
  CONSTRAINT `doctor_is_staff` FOREIGN KEY (`sid`) REFERENCES `staff` (`sid`) ON DELETE
  CASCADE ON UPDATE CASCADE)

CREATE TABLE `nurses` (
  `sid` int(11) NOT NULL,
  `license_number` varchar(20) NOT NULL,
  `license_expiry_date` date NOT NULL,
  PRIMARY KEY (`sid`),
  UNIQUE KEY `unique_license_number` (`license_number`),
  CONSTRAINT `nurse_is_staff` FOREIGN KEY (`sid`) REFERENCES `staff` (`sid`) ON DELETE
  CASCADE ON UPDATE CASCADE)

CREATE TABLE `admins` (
```

```

`sid` int(11) NOT NULL,
`adminid` int(11) NOT NULL,
PRIMARY KEY (`sid`),
UNIQUE KEY `unique_adminid` (`adminid`),
CONSTRAINT `admin_is_staff` FOREIGN KEY (`sid`) REFERENCES `staff` (`sid`) ON DELETE
CASCADE ON UPDATE CASCADE)

```

```

CREATE TABLE `patients` (
  `pid` int(11) NOT NULL,
  `pfname` varchar(45) NOT NULL,
  `plname` varchar(45) NOT NULL,
  `pgender` varchar(5) NOT NULL DEFAULT 'F',
  `pDoB` date NOT NULL,
  `pinsurance_number` varchar(45) NOT NULL,
  `pblood_type` varchar(2) NOT NULL,
  `pphone_number` varchar(20) NOT NULL,
  `pemergengy_contact_name` varchar(45) NOT NULL,
  `pemergengy_contact_phone_number` varchar(20) NOT NULL,
  `pemergengy_contact_relationship` varchar(45) NOT NULL,
  `referredby` varchar(45) NOT NULL DEFAULT 'self',
  PRIMARY KEY (`pid`),
  UNIQUE KEY `unique_insurance_number` (`pinsurance_number`))

```

```

CREATE TABLE `visits` (
  `v_number` int(11) NOT NULL AUTO_INCREMENT,
  `adminid` int(11) NOT NULL,
  `pid` int(11) NOT NULL,
  `sid` int(11) NOT NULL,
  `vtype` varchar(40) NOT NULL,
  `vreason` varchar(100) NOT NULL,
  `checkedin_datetime` datetime DEFAULT NULL,
  `scheduled_datetime` datetime DEFAULT NULL,
  PRIMARY KEY (`v_number`),
  UNIQUE KEY `unique_sid_checkedin` (`sid`,`checkedin_datetime`),
  UNIQUE KEY `unique_sid_scheduled` (`sid`,`scheduled_datetime`),
  UNIQUE KEY `unique_pid_checkedin` (`pid`,`checkedin_datetime`),
  UNIQUE KEY `unique_pid_scheduled` (`pid`,`scheduled_datetime`),
  KEY `visit_doctor_idx` (`sid`),
  KEY `visit_patient_idx` (`pid`),
  KEY `visit_admin_idx` (`adminid`),
  CONSTRAINT `visit_admin` FOREIGN KEY (`adminid`) REFERENCES `admins` (`adminid`),
  CONSTRAINT `visit_doctor` FOREIGN KEY (`sid`) REFERENCES `doctors` (`sid`),
  CONSTRAINT `visit_patient` FOREIGN KEY (`pid`) REFERENCES `patients` (`pid`))

```

```

CREATE TABLE `admissions` (
  `a_number` int(11) NOT NULL AUTO_INCREMENT,
  `adminid` int(11) NOT NULL,
  `pid` int(11) NOT NULL,
  `admission_datetime` datetime NOT NULL,
  `areason` varchar(45) NOT NULL,
  `aroom_number` varchar(5) NOT NULL,
  `adischarged_datetime` datetime DEFAULT NULL,
  `a_notes` varchar(200) NOT NULL DEFAULT 'None',
  PRIMARY KEY (`a_number`),
  UNIQUE KEY `unique_pid_admission_datetime` (`pid`,`admission_datetime`),
  KEY `admission_admin_idx` (`adminid`),
  KEY `admission_patient_idx` (`pid`),
  CONSTRAINT `admission_admin` FOREIGN KEY (`adminid`) REFERENCES `admins` (`adminid`),
  CONSTRAINT `admission_patient` FOREIGN KEY (`pid`) REFERENCES `patients` (`pid`))

```

```

CREATE TABLE `babies` (
  `bid` int(11) NOT NULL,
  `pid` int(11) NOT NULL,
  `bgender` varchar(1) NOT NULL,
  `bweight` int(11) NOT NULL,
  `bbirthmark` varchar(45) NOT NULL DEFAULT 'None',
  `bblood_type` varchar(2) NOT NULL,
  `bdelivery_type` varchar(45) NOT NULL,
  `bdelivery_datetime` datetime NOT NULL,
  `bcondition_at_delivery` varchar(45) NOT NULL,
  `bbed_number` varchar(5) NOT NULL,
  `bdischarged_datetime` datetime DEFAULT NULL,
  `bnotes` varchar(100) NOT NULL DEFAULT 'None',
  PRIMARY KEY (`bid`),
  UNIQUE KEY `unique_mother_delivery_datetime` (`pid`,`bdelivery_datetime`),
  KEY `birthed_by_idx` (`pid`),
  CONSTRAINT `birthed_by` FOREIGN KEY (`pid`) REFERENCES `patients` (`pid`))

```

### **Relationships:**

```

CREATE TABLE `treat` (
  `sid` int(11) NOT NULL,
  `pid` int(11) NOT NULL,
  `tdatetime` datetime NOT NULL,
  `treason` varchar(100) NOT NULL,
  `ttreatment` varchar(100) NOT NULL,
  `pregnancy_stage` varchar(45) NOT NULL,

```

```

`tnotes` varchar(100) NOT NULL DEFAULT 'None',
PRIMARY KEY (`sid`,`pid`,`tdatetime`),
UNIQUE KEY `unique_doctor_datetime` (`sid`,`tdatetime`),
KEY `doctor_treat_idx` (`sid`),
KEY `patient_is_treated_by_idx` (`pid`),
CONSTRAINT `doctor_treat` FOREIGN KEY (`sid`) REFERENCES `doctors` (`sid`),
CONSTRAINT `patient_is_treated` FOREIGN KEY (`pid`) REFERENCES `patients` (`pid`))

```

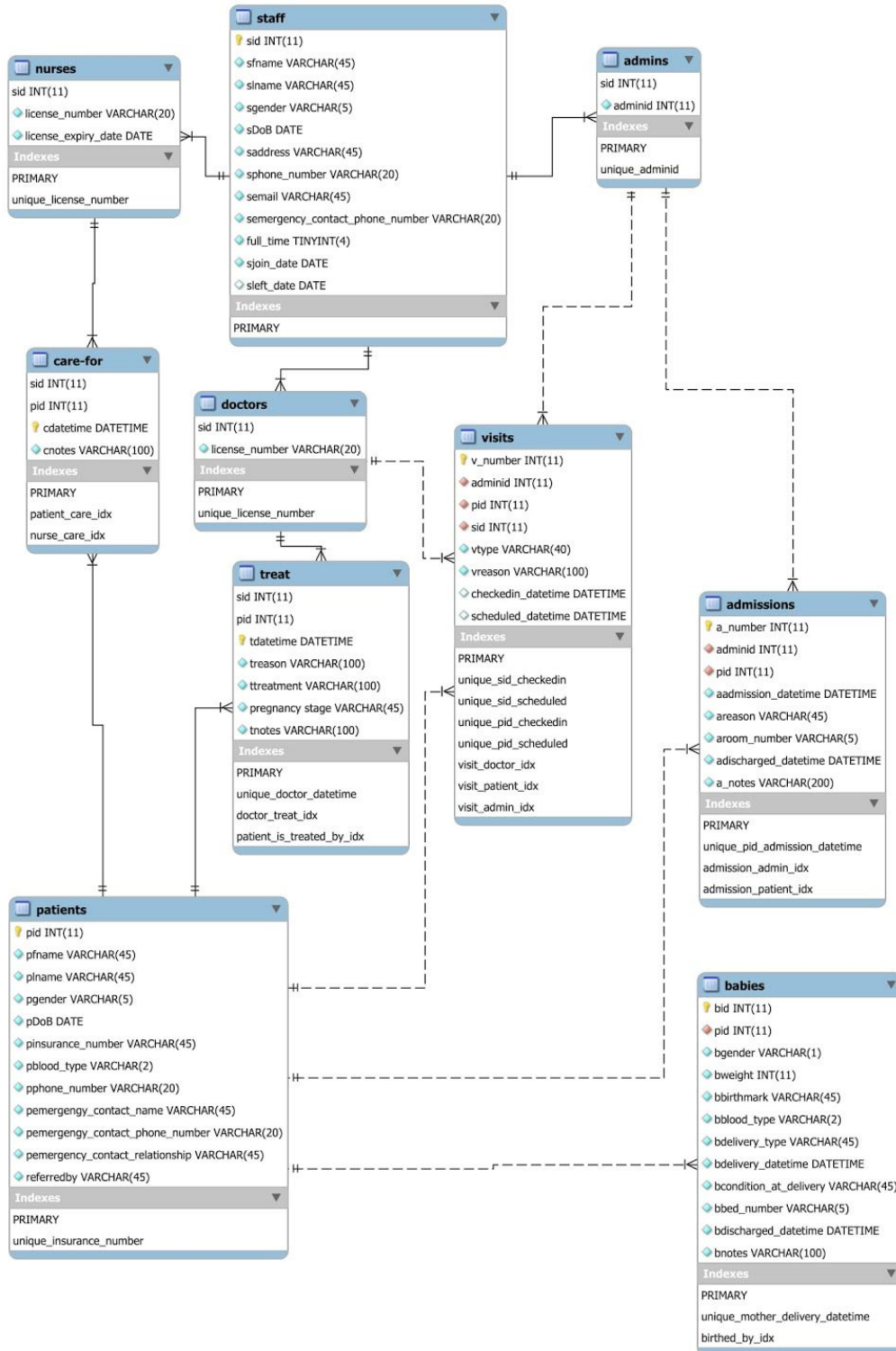
```

CREATE TABLE `care-for` (
  `sid` int(11) NOT NULL,
  `pid` int(11) NOT NULL,
  `cdatetime` datetime NOT NULL,
  `cnotes` varchar(100) NOT NULL DEFAULT 'None',
  PRIMARY KEY (`sid`,`pid`,`cdatetime`),
  KEY `patient_care_idx` (`pid`),
  KEY `nurse_care_idx` (`sid`),
  CONSTRAINT `nurse_care` FOREIGN KEY (`sid`) REFERENCES `nurses` (`sid`),
  CONSTRAINT `patient_care` FOREIGN KEY (`pid`) REFERENCES `patients` (`pid`))

```

### **Enhanced Entity-Relationship Diagram**





## **References**

- (1) Lucidchart. (2020). ER Diagram (ERD) Tool. Retrieved February 18, 2020, from <https://www.lucidchart.com/pages/examples/er-diagram-tool>
- (2) Sam. (n.d.). Hospital Management System Project for Final Year. Retrieved February 19, 2020, from [https://www.lovelycoding.org/2015/10/hospital-management-system-php-java-project.html?fbclid=IwAR1AWeJUP8UR6ehs639oGWFz4QiMd7\\_SK8or-5Z3WmOEeKGbtL3\\_uETBZsw](https://www.lovelycoding.org/2015/10/hospital-management-system-php-java-project.html?fbclid=IwAR1AWeJUP8UR6ehs639oGWFz4QiMd7_SK8or-5Z3WmOEeKGbtL3_uETBZsw)