DBMS PROJECT

ASHA Worker Health Record Management System

By :

Vyshnav Reddy Pinreddy

Rishith Thommandru

About :

The ASHA Worker Health Record Management System is a community health-focused database solution designed to streamline and digitize the critical health-related data maintained by ASHA (Accredited Social Health Activist) workers in rural India. Traditionally, ASHA workers manage vital information such as pregnancy tracking, immunizations, chronic diseases, and medicine distribution through handwritten records, making data retrieval and analysis time-consuming and error-prone.

To address this challenge, we are developing a robust Database Management System (DBMS) that allows for the structured storage, easy retrieval, and meaningful analysis of healthcare data at the village level. This system captures information on households, individuals, health events (such as births, deaths, and pregnancies), immunization history, chronic conditions, and medicine distribution. It also tracks ASHA activities such as village visits.

The objective is to empower ASHA workers and public health officials with accurate, up-to-date information that enhances health service delivery, facilitates proactive interventions, and improves maternal and child health outcomes.

The system will also:

- Monitor vaccine eligibility based on age groups.
- Track pregnancy records and link newborn data.
- Maintain a medical history for chronic illnesses.

By converting handwritten records into a normalized digital database, this project supports better health decision-making, resource allocation, and rural health program management.

ENTITIES USED:

- **Village**: This entity represents each village under the ASHA worker's coverage.

- **Asha**: This entity represents ASHA workers who are each assigned to exactly one village.

- **Family**: This entity represents families residing in a village.

- **Person**: This entity represents each individual in a family (including children, parents, etc.).

- **VillageVisitRecord**: This entity represents each visit an ASHA makes to a village (with date, purpose, notes).

- **PregnancyRecord**: This entity represents a recorded pregnancy—linking a husband (Person) and wife (Person), along with LMP and current status.

- **NewBornRecord**: This entity represents each newborn child resulting from a pregnancy (e.g., twins, triplets).

- **DeathRecord**: This entity represents a death event for a Person, including date and cause.

- **Disease**: This entity represents chronic diseases (e.g., diabetes, hypertension) that can be diagnosed in persons.

- **ChronicDiseaseRecord**: This entity represents each time a Person is diagnosed with a chronic disease (links Person ↔ Disease with notes).

- **Vaccine**: This entity represents each vaccine (e.g., DPT, BCG) that can be administered.

- **ImmunizationRecord**: This entity represents each vaccination event for a Person (links Person ↔ Vaccine with date).

- **Tablet**: This entity represents each tablet/medicine that can be distributed (e.g., iron supplements).

- **MedicineDistributionRecord**: This entity represents each instance of tablet/medicine distribution to a Person (links Person ↔ Tablet with date).

RELATIONSHIPS:

1. Asha ↔ Village

   - One Asha can be assigned to one Village. (One-to-One)

   - One Village can be assigned to one Asha. (One-to-One)

2. Asha ↔ Person

   - Each Asha corresponds to exactly one Person (One-to-One)

   - Each Person (who is an Asha) corresponds to exactly one Asha record. (One-to-One)

3. Village ↔ Family

   - One Village can have multiple Families. (One-to-Many)

   - One Family belongs to one Village. (Many-to-One)

4. Family ↔ Person

   - One Family can have multiple Persons. (One-to-Many)

   - One Person belongs to one Family. (Many-to-One)

5. Person ↔ Person (Parent-Child)

   - One Person (as a parent) can be parent of multiple Persons. (One-to-Many)

   - One Person (as a child) can have at most one father and one mother. (Many-to-One, for FatherID and MotherID separately)

6. Person ↔ PregnancyRecord

   - One Person can be a husband in multiple PregnancyRecords. (One-to-Many)

   - One Person can be a wife in multiple PregnancyRecords. (One-to-Many)

   - One PregnancyRecord links exactly one husband (Person) and one wife (Person). (Many-to-One on both sides)

7. PregnancyRecord ↔ NewBornRecord

   - One PregnancyRecord can result in multiple NewBornRecords (twins, triplets, etc.). (One-to-Many)

   - One NewBornRecord belongs to exactly one PregnancyRecord. (Many-to-One)

8. PregnancyRecord ↔ NewBornRecord
   - Each NewBornRecord corresponds to exactly one Person. (One-to-One)
   - Each Person (if they are a newborn) corresponds to exactly one NewBornRecord. (One-to-One)

9. Person ↔ DeathRecord
   - One Person can have at most one DeathRecord. (One-to-One)
   - One DeathRecord refers to exactly one Person. (One-to-One)

10. Person ↔ ChronicDiseaseRecord
    - One Person can have multiple ChronicDiseaseRecords. (One-to-Many)
    - One ChronicDiseaseRecord refers to exactly one Person. (Many-to-One)

11. Disease ↔ ChronicDiseaseRecord
    - One Disease can appear in multiple ChronicDiseaseRecords. (One-to-Many)
    - One ChronicDiseaseRecord refers to exactly one Disease. (Many-to-One)

12. Person ↔ ImmunizationRecord
    - One Person can have multiple ImmunizationRecords. (One-to-Many)
    - One ImmunizationRecord refers to exactly one Person. (Many-to-One)

13. Vaccine ↔ ImmunizationRecord

- One Vaccine can appear in multiple ImmunizationRecords. (One-to-Many)
- One ImmunizationRecord refers to exactly one Vaccine. (Many-to-One)

14. Person ↔ MedicineDistributionRecord
    - One Person can have multiple MedicineDistributionRecords. (One-to-Many)
    - One MedicineDistributionRecord refers to exactly one Person. (Many-to-One)

15. Tablet ↔ MedicineDistributionRecord
    - One Tablet can appear in multiple MedicineDistributionRecords. (One-to-Many)
    - One MedicineDistributionRecord refers to exactly one Tablet. (Many-to-One)

16. Asha ↔ VillageVisitRecord
    - One Asha can have multiple VillageVisitRecords. (One-to-Many)
    - One VillageVisitRecord refers to exactly one Asha. (Many-to-One)

17. Village ↔ VillageVisitRecord
    - One Village can have multiple VillageVisitRecords. (One-to-Many)
    - One VillageVisitRecord refers to exactly one Village. (Many-to-One)

Functional Dependencies after Normalization

Village Table

- `VillageID` → `VillageName`, `Mandal`, `District`, `Population`, `LastUpdatedDateofPopulation`
  - ***Reason:*** *`VillageID` is the Primary Key.*

Asha Table

- `AshaID` → `PersonID`, `FirstName`, `LastName`, `PhoneNo`, `AssignedVillageID`, `DateofJoining`, `CurrentStatus`
  - ***Reason:*** *`AshaID` is the Primary Key.*
- `PhoneNo` → `AshaID`, `PersonID`, `FirstName`, `LastName`, `AssignedVillageID`, `DateofJoining`, `CurrentStatus`
  - ***Reason:*** *`PhoneNo` has a `UNIQUE` constraint, making it a Candidate Key.*
- `AssignedVillageID` → `AshaID`, `PersonID`, `FirstName`, `LastName`, `PhoneNo`, `DateofJoining`, `CurrentStatus`
  - ***Reason:*** *`AssignedVillageID` has a `UNIQUE` constraint, making it a Candidate Key.*
- `PersonID` → `AshaID`, `FirstName`, `LastName`, `PhoneNo`, `AssignedVillageID`, `DateofJoining`, `CurrentStatus`
  - ***Reason:*** *The described 1-to-1 relationship ("Each Asha corresponds to exactly one Person" and "Each Person (who is an Asha) corresponds to exactly one Asha record") implies `PersonID` is unique within the `Asha` table for individuals who are Asha workers, thus acting as a Candidate Key.*

Family Table

- *`FamilyID` → `HouseNo`, `VillageID`, `HouseType`, `ToiletFacility`*
    - ***Reason:** `FamilyID` is the Primary Key.*

Person Table

- *`PersonID` → `FirstName`, `LastName`, `DateOfBirth`, `Gender`, `FamilyID`, `PhoneNo`, `FatherID`, `MotherID`, `IsDeceased`*
    - ***Reason:** `PersonID` is the Primary Key.*
    - ***Note:** `PhoneNo` in the `Person` table does not have a `UNIQUE` constraint in the schema, so it cannot be listed as a candidate key for `Person` based solely on the DDL.*

VillageVisitRecord Table

- *`VisitID` → `AshaID`, `VillageID`, `DateOfVisit`, `PurposeofVisit`, `Notes`*
    - ***Reason:** `VisitID` is the Primary Key.*
- *`AshaID` → `VillageID`*
    - ***Reason:** An Asha worker (`AshaID`) is uniquely assigned to one village (`AssignedVillageID` in the `Asha` table, which is `UNIQUE`). Assuming a visit record (`VillageVisitRecord`) is made by an Asha worker for their assigned village, the `AshaID` in `VillageVisitRecord` determines the `VillageID` of the visit. This is supported by the sample data and the 1-to-1 `Asha` to `Village` assignment.*

PregnancyRecord Table

- `PregnancyID` → `HusbandID`, `WifeID`, `LMP`, `CurrentStatus`
    - ***Reason:*** *`PregnancyID` is the Primary Key.*

NewBornRecord Table

- `PersonID` → `PregnancyID`, `Gender`, `BirthWeight`, `DateOfBirth`, `TypeOfDelivery`, `Notes`
    - ***Reason:*** *`PersonID` (referring to the newborn's ID) is the Primary Key. The 1-to-1 relationship "Each NewBornRecord corresponds to exactly one Person" supports this.*

DeathRecord Table

- `PersonID` → `DateOfDeath`, `CauseOfDeath`
    - ***Reason:*** *`PersonID` (referring to the deceased's ID) is the Primary Key, due to the 1-to-1 relationship "One Person can have at most one DeathRecord."*

Disease Table

- `DiseaseID` → `DiseaseName`
    - ***Reason:*** *`DiseaseID` is the Primary Key.*
- `DiseaseName` → `DiseaseID`
    - ***Reason:*** *Assuming `DiseaseName` is unique for each disease (a common convention for lookup tables), it acts as a Candidate Key.*

ChronicDiseaseRecord Table

- `RecordID` → `PersonID`, `DiseaseID`, `Notes`
  - ***Reason:*** `RecordID` *is the Primary Key.*

Vaccine Table

- `VaccineID` → `VaccineName`, `MinimumAge`, `MaximumAge`
  - ***Reason:*** `VaccineID` *is the Primary Key.*
- `VaccineName` → `VaccineID`, `MinimumAge`, `MaximumAge`
  - ***Reason:*** *Assuming* `VaccineName` *is unique for each vaccine (a common convention for lookup tables), it acts as a Candidate Key.*

ImmunizationRecord Table

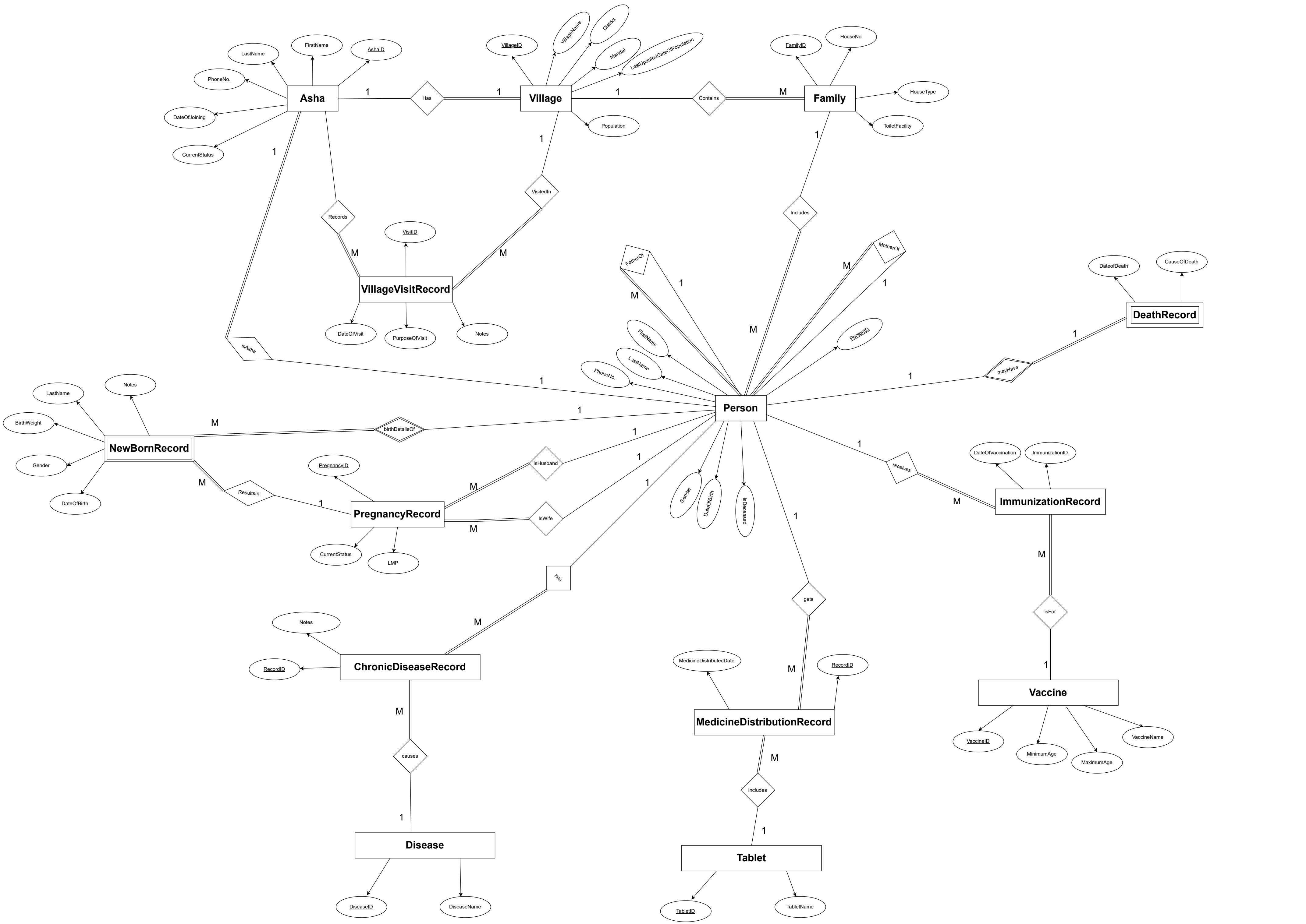
- `ImmunizationID` → `VaccineID`, `PersonID`, `DateOfVaccination`
  - ***Reason:*** `ImmunizationID` *is the Primary Key.*

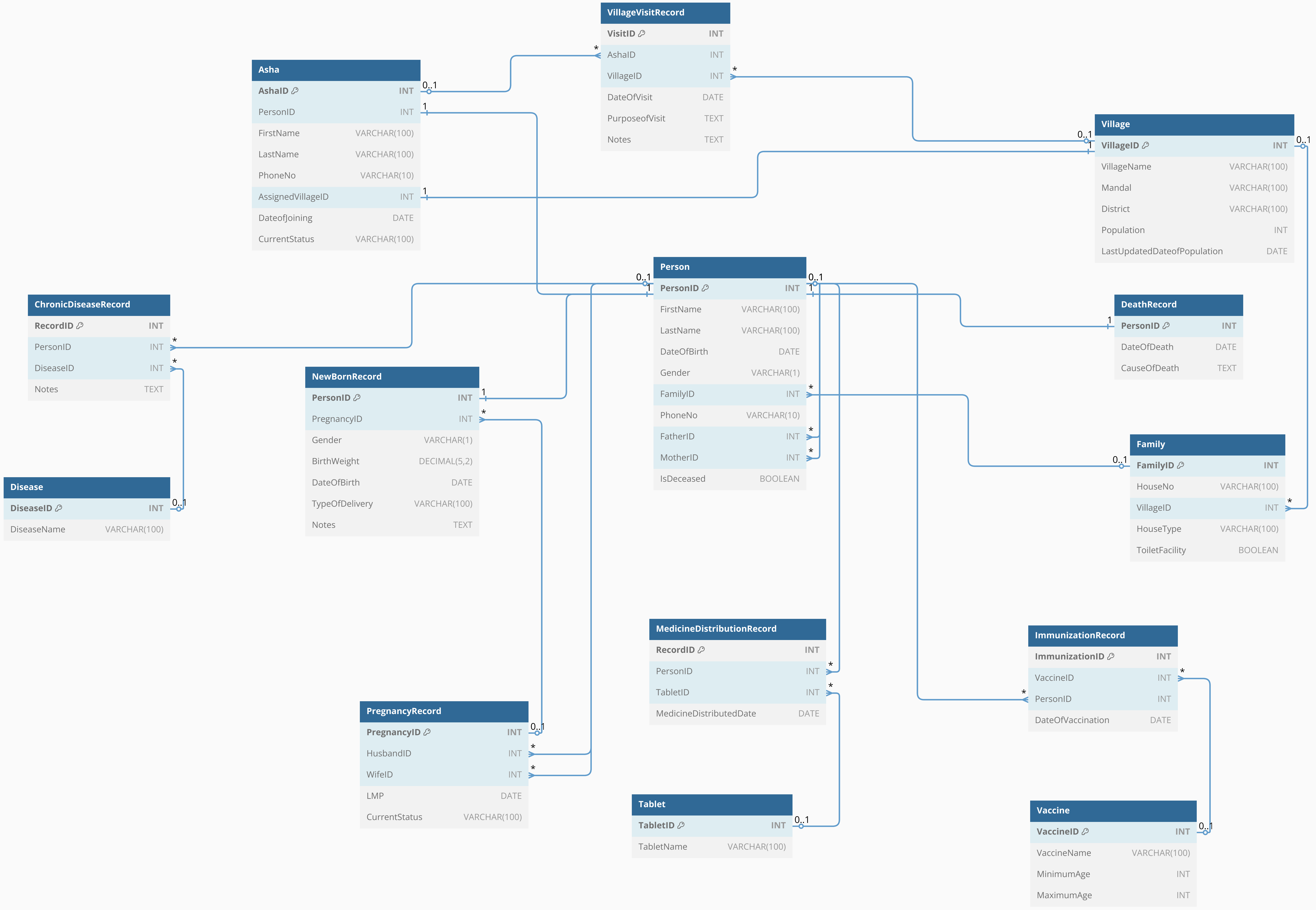
Tablet Table

- `TabletID` → `TabletName`
  - ***Reason:*** `TabletID` *is the Primary Key.*
- `TabletName` → `TabletID`
  - ***Reason:*** *Assuming* `TabletName` *is unique for each tablet (a common convention for lookup tables), it acts as a Candidate Key.*

MedicineDistributionRecord Table

- `RecordID` → `PersonID`, `TabletID`, `MedicineDistributedDate`
  - ***Reason:*** `RecordID` *is the Primary Key.*


Asha
AshaID
FirstName
LastName
PhoneNo.
DateOfJoining
CurrentStatus
Has
Village
VillageID
VillageName
District
Mandal
LastUpdatedDateOfPopulation
Population
Contains
Family
FamilyID
HouseNo
HouseType
ToiletFacility
Records
VisitedIn
VillageVisitRecord
VisitID
DateOfVisit
PurposeOfVIsit
Notes
Includes
FatherOf
MotherOf
isAsha
Person
PersonID
FirstName
LastName
PhoneNo.
Gender
DateOfBirth
IsDeceased
DeathRecord
DateofDeath
CauseOfDeath
mayHave
NewBornRecord
LastName
Notes
BirthWeight
Gender
DateOfBirth
birthDetailsOf
ResultsIn
PregnancyRecord
PregnancyID
CurrentStatus
LMP
IsHusband
IsWife
receives
ImmunizationRecord
DateOfVaccination
ImmunizationID
isFor
Vaccine
VaccineID
MinimumAge
MaximumAge
VaccineName
has
ChronicDiseaseRecord
Notes
RecordID
causes
Disease
DiseaseID
DiseaseName
gets
MedicineDistributionRecord
MedicineDistributedDate
RecordID
includes
Tablet
TabletID
TabletName
1
1
1
M
1
1
1
M
M
M
1
M
M
1
M
1
1
1
1
M
M
1
M
1
1
M
1
M
1
M
M
M
M
1
1
M
M
1
M
1



VillageVisitRecord	
VisitID	INT
AshaID	INT
VillageID	INT
DateOfVisit	DATE
PurposeofVisit	TEXT
Notes	TEXT

Asha	
AshaID	INT
PersonID	INT
FirstName	VARCHAR(100)
LastName	VARCHAR(100)
PhoneNo	VARCHAR(10)
AssignedVillageID	INT
DateofJoining	DATE
CurrentStatus	VARCHAR(100)

Village	
VillageID	INT
VillageName	VARCHAR(100)
Mandal	VARCHAR(100)
District	VARCHAR(100)
Population	INT
LastUpdatedDateofPopulation	DATE

Person	
PersonID	INT
FirstName	VARCHAR(100)
LastName	VARCHAR(100)
DateOfBirth	DATE
Gender	VARCHAR(1)
FamilyID	INT
PhoneNo	VARCHAR(10)
FatherID	INT
MotherID	INT
IsDeceased	BOOLEAN

ChronicDiseaseRecord	
RecordID	INT
PersonID	INT
DiseaseID	INT
Notes	TEXT

DeathRecord	
PersonID	INT
DateOfDeath	DATE
CauseOfDeath	TEXT

NewBornRecord	
PersonID	INT
PregnancyID	INT
Gender	VARCHAR(1)
BirthWeight	DECIMAL(5,2)
DateOfBirth	DATE
TypeOfDelivery	VARCHAR(100)
Notes	TEXT

Family	
FamilyID	INT
HouseNo	VARCHAR(100)
VillageID	INT
HouseType	VARCHAR(100)
ToiletFacility	BOOLEAN

Disease	
DiseaseID	INT
DiseaseName	VARCHAR(100)

MedicineDistributionRecord	
RecordID	INT
PersonID	INT
TabletID	INT
MedicineDistributedDate	DATE

ImmunizationRecord	
ImmunizationID	INT
VaccineID	INT
PersonID	INT
DateOfVaccination	DATE

PregnancyRecord	
PregnancyID	INT
HusbandID	INT
WifeID	INT
LMP	DATE
CurrentStatus	VARCHAR(100)

Tablet	
TabletID	INT
TabletName	VARCHAR(100)

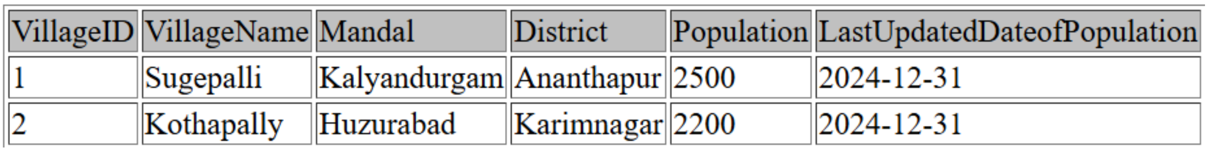
Vaccine	
VaccineID	INT
VaccineName	VARCHAR(100)
MinimumAge	INT
MaximumAge	INT

dbdiagram.io

TABLE CREATION AND INSERTION:-

Village Table

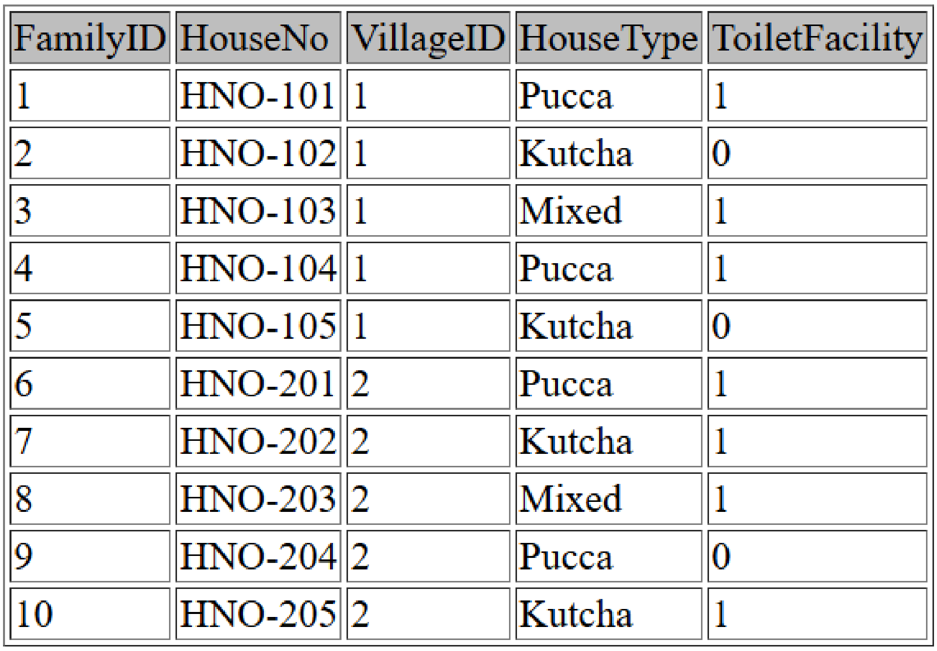
```
CREATE TABLE Village(

 VillageID INT PRIMARY KEY,

    VillageName VARCHAR(100) NOT NULL,

    Mandal VARCHAR(100) NOT NULL,

    District VARCHAR(100) NOT NULL,

    Population INT,

    LastUpdatedDateofPopulation DATE

);


INSERT INTO Village (VillageID, VillageName, Mandal, District, Population,
LastUpdatedDateofPopulation )

 VALUES

(1, 'Sugepalli', 'Kalyandurgam', 'Ananthapur', 2500, '2024-12-31'),

(2, 'Kothapally', 'Huzurabad', 'Karimnagar', 2200, '2024-12-31');
```



VillageID	VillageName	Mandal	District	Population	LastUpdatedDateofPopulation
1	Sugepalli	Kalyandurgam	Ananthapur	2500	2024-12-31
2	Kothapally	Huzurabad	Karimnagar	2200	2024-12-31

Family Table

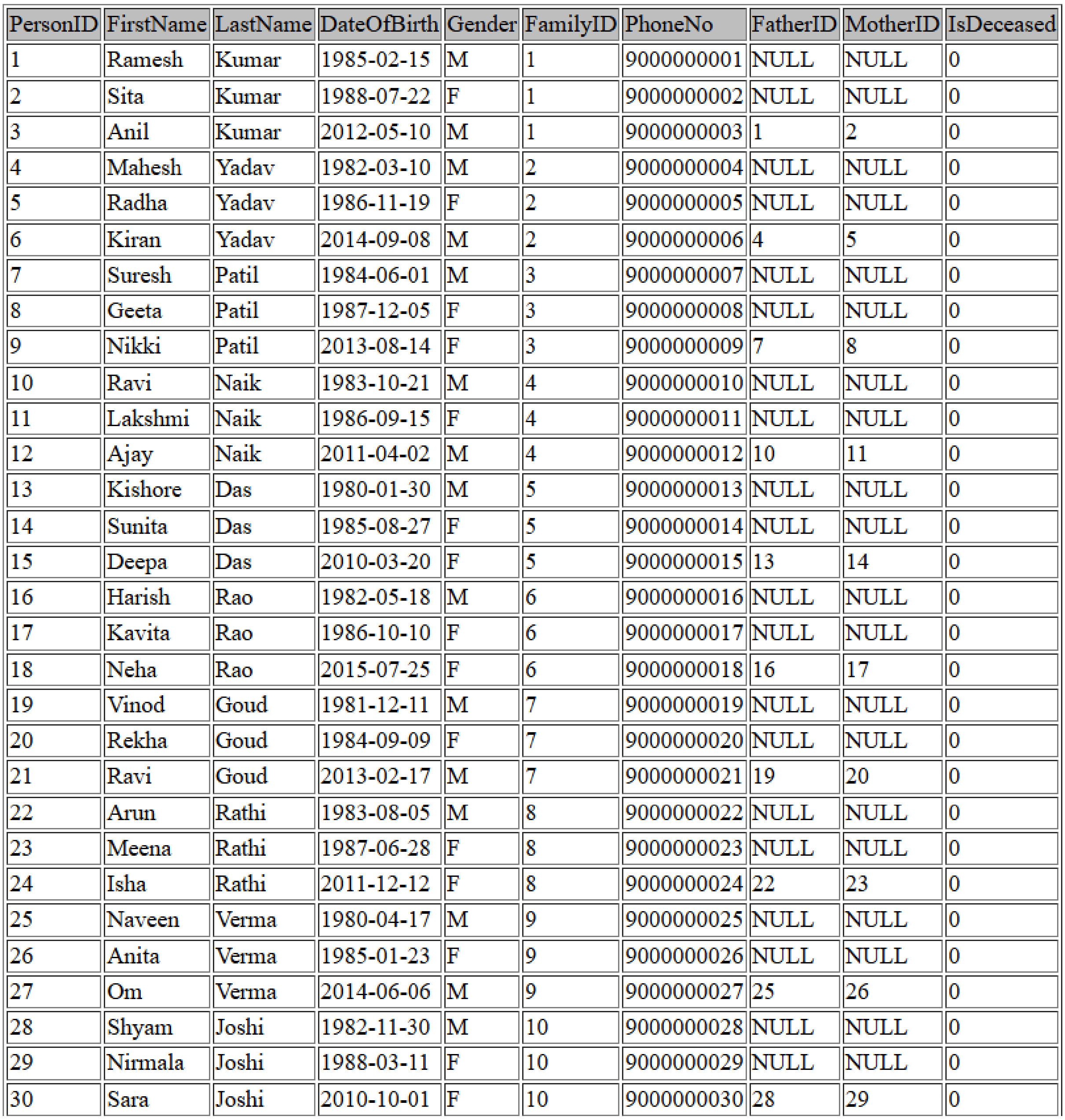
```
CREATE TABLE Family(

   FamilyID INT PRIMARY KEY,

   HouseNo VARCHAR(100),

   VillageID INT NOT NULL,

   HouseType VARCHAR(100),

   ToiletFacility BOOLEAN,

   FOREIGN KEY (VillageID) REFERENCES Village (VillageID)

);


INSERT INTO Family (FamilyID, HouseNo, VillageID, HouseType, ToiletFacility)
VALUES

(1, 'HNO-101', 1, 'Pucca', TRUE),

(2, 'HNO-102', 1, 'Kutcha', FALSE),

(3, 'HNO-103', 1, 'Mixed', TRUE),

(4, 'HNO-104', 1, 'Pucca', TRUE),

(5, 'HNO-105', 1, 'Kutcha', FALSE),

(6, 'HNO-201', 2, 'Pucca', TRUE),

(7, 'HNO-202', 2, 'Kutcha', TRUE),

(8, 'HNO-203', 2, 'Mixed', TRUE),

(9, 'HNO-204', 2, 'Pucca', FALSE),

(10, 'HNO-205', 2, 'Kutcha', TRUE);
```



FamilyID	HouseNo	VillageID	HouseType	ToiletFacility
1	HNO-101	1	Pucca	1
2	HNO-102	1	Kutcha	0
3	HNO-103	1	Mixed	1
4	HNO-104	1	Pucca	1
5	HNO-105	1	Kutcha	0
6	HNO-201	2	Pucca	1
7	HNO-202	2	Kutcha	1
8	HNO-203	2	Mixed	1
9	HNO-204	2	Pucca	0
10	HNO-205	2	Kutcha	1

Person Table

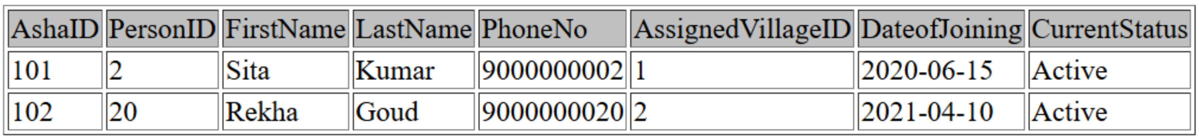
```
CREATE TABLE Person(

  PersonID INT PRIMARY KEY AUTO_INCREMENT,

  FirstName VARCHAR(100) NOT NULL,

  LastName VARCHAR(100),

  DateOfBirth DATE NOT NULL,

  Gender VARCHAR(1) CHECK (Gender IN ('M', 'F', 'O')),

  FamilyID INT NOT NULL,

  PhoneNo VARCHAR(10) NOT NULL CHECK (LENGTH(PhoneNo)=10),

  FatherID INT,

  MotherID INT,
    IsDeceased BOOLEAN DEFAULT FALSE,

    FOREIGN KEY (FamilyID) REFERENCES Family(FamilyID),

    FOREIGN KEY (FatherID) REFERENCES Person(PersonID),

    FOREIGN KEY (MotherID) REFERENCES Person(PersonID)

);


INSERT INTO Person (PersonID, FirstName, LastName, DateOfBirth, Gender,

FamilyID, PhoneNo, FatherID, MotherID, IsDeceased) VALUES

(1, 'Ramesh', 'Kumar', '1985-02-15', 'M', 1, '9000000001', NULL, NULL, FALSE),

(2, 'Sita', 'Kumar', '1988-07-22', 'F', 1, '9000000002', NULL, NULL, FALSE),

(3, 'Anil', 'Kumar', '2012-05-10', 'M', 1, '9000000003', 1, 2, FALSE),

(4, 'Mahesh', 'Yadav', '1982-03-10', 'M', 2, '9000000004', NULL, NULL, FALSE),

(5, 'Radha', 'Yadav', '1986-11-19', 'F', 2, '9000000005', NULL, NULL, FALSE),

(6, 'Kiran', 'Yadav', '2014-09-08', 'M', 2, '9000000006', 4, 5, FALSE),

(7, 'Suresh', 'Patil', '1984-06-01', 'M', 3, '9000000007', NULL, NULL, FALSE),

(8, 'Geeta', 'Patil', '1987-12-05', 'F', 3, '9000000008', NULL, NULL, FALSE),

(9, 'Nikki', 'Patil', '2013-08-14', 'F', 3, '9000000009', 7, 8, FALSE),

(10, 'Ravi', 'Naik', '1983-10-21', 'M', 4, '9000000010', NULL, NULL, FALSE),

(11, 'Lakshmi', 'Naik', '1986-09-15', 'F', 4, '9000000011', NULL, NULL, FALSE),

(12, 'Ajay', 'Naik', '2011-04-02', 'M', 4, '9000000012', 10, 11, FALSE),

(13, 'Kishore', 'Das', '1980-01-30', 'M', 5, '9000000013', NULL, NULL, FALSE),

(14, 'Sunita', 'Das', '1985-08-27', 'F', 5, '9000000014', NULL, NULL, FALSE),

(15, 'Deepa', 'Das', '2010-03-20', 'F', 5, '9000000015', 13, 14, FALSE),
(16, 'Harish', 'Rao', '1982-05-18', 'M', 6, '9000000016', NULL, NULL, FALSE),

(17, 'Kavita', 'Rao', '1986-10-10', 'F', 6, '9000000017', NULL, NULL, FALSE),

(18, 'Neha', 'Rao', '2015-07-25', 'F', 6, '9000000018', 16, 17, FALSE),

(19, 'Vinod', 'Goud', '1981-12-11', 'M', 7, '9000000019', NULL, NULL, FALSE),

(20, 'Rekha', 'Goud', '1984-09-09', 'F', 7, '9000000020', NULL, NULL, FALSE),

(21, 'Ravi', 'Goud', '2013-02-17', 'M', 7, '9000000021', 19, 20, FALSE),

(22, 'Arun', 'Rathi', '1983-08-05', 'M', 8, '9000000022', NULL, NULL, FALSE),

(23, 'Meena', 'Rathi', '1987-06-28', 'F', 8, '9000000023', NULL, NULL, FALSE),

(24, 'Isha', 'Rathi', '2011-12-12', 'F', 8, '9000000024', 22, 23, FALSE),

(25, 'Naveen', 'Verma', '1980-04-17', 'M', 9, '9000000025', NULL, NULL, FALSE),

(26, 'Anita', 'Verma', '1985-01-23', 'F', 9, '9000000026', NULL, NULL, FALSE),

(27, 'Om', 'Verma', '2014-06-06', 'M', 9, '9000000027', 25, 26, FALSE),

(28, 'Shyam', 'Joshi', '1982-11-30', 'M', 10, '9000000028', NULL, NULL, FALSE),

(29, 'Nirmala', 'Joshi', '1988-03-11', 'F', 10, '9000000029', NULL, NULL, FALSE),

(30, 'Sara', 'Joshi', '2010-10-01', 'F', 10, '9000000030', 28, 29, FALSE);
```



PersonID	FirstName	LastName	DateOfBirth	Gender	FamilyID	PhoneNo	FatherID	MotherID	IsDeceased
1	Ramesh	Kumar	1985-02-15	M	1	9000000001	NULL	NULL	0
2	Sita	Kumar	1988-07-22	F	1	9000000002	NULL	NULL	0
3	Anil	Kumar	2012-05-10	M	1	9000000003	1	2	0
4	Mahesh	Yadav	1982-03-10	M	2	9000000004	NULL	NULL	0
5	Radha	Yadav	1986-11-19	F	2	9000000005	NULL	NULL	0
6	Kiran	Yadav	2014-09-08	M	2	9000000006	4	5	0
7	Suresh	Patil	1984-06-01	M	3	9000000007	NULL	NULL	0
8	Geeta	Patil	1987-12-05	F	3	9000000008	NULL	NULL	0
9	Nikki	Patil	2013-08-14	F	3	9000000009	7	8	0
10	Ravi	Naik	1983-10-21	M	4	9000000010	NULL	NULL	0
11	Lakshmi	Naik	1986-09-15	F	4	9000000011	NULL	NULL	0
12	Ajay	Naik	2011-04-02	M	4	9000000012	10	11	0
13	Kishore	Das	1980-01-30	M	5	9000000013	NULL	NULL	0
14	Sunita	Das	1985-08-27	F	5	9000000014	NULL	NULL	0
15	Deepa	Das	2010-03-20	F	5	9000000015	13	14	0
16	Harish	Rao	1982-05-18	M	6	9000000016	NULL	NULL	0
17	Kavita	Rao	1986-10-10	F	6	9000000017	NULL	NULL	0
18	Neha	Rao	2015-07-25	F	6	9000000018	16	17	0
19	Vinod	Goud	1981-12-11	M	7	9000000019	NULL	NULL	0
20	Rekha	Goud	1984-09-09	F	7	9000000020	NULL	NULL	0
21	Ravi	Goud	2013-02-17	M	7	9000000021	19	20	0
22	Arun	Rathi	1983-08-05	M	8	9000000022	NULL	NULL	0
23	Meena	Rathi	1987-06-28	F	8	9000000023	NULL	NULL	0
24	Isha	Rathi	2011-12-12	F	8	9000000024	22	23	0
25	Naveen	Verma	1980-04-17	M	9	9000000025	NULL	NULL	0
26	Anita	Verma	1985-01-23	F	9	9000000026	NULL	NULL	0
27	Om	Verma	2014-06-06	M	9	9000000027	25	26	0
28	Shyam	Joshi	1982-11-30	M	10	9000000028	NULL	NULL	0
29	Nirmala	Joshi	1988-03-11	F	10	9000000029	NULL	NULL	0
30	Sara	Joshi	2010-10-01	F	10	9000000030	28	29	0

Asha Table

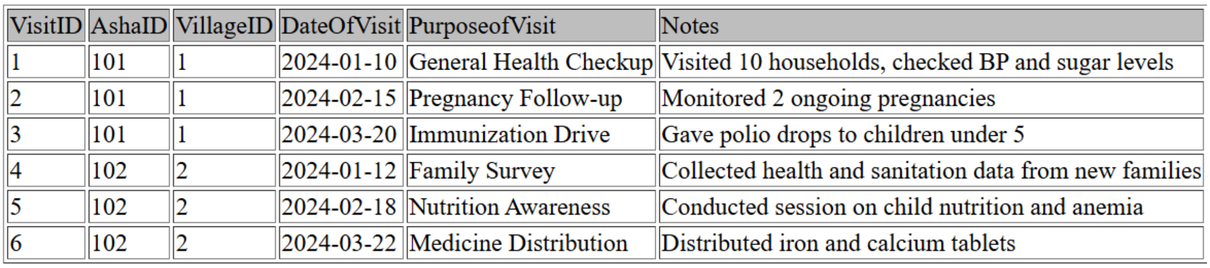
```
CREATE TABLE Asha(
     AshaID INT PRIMARY KEY,
      PersonID INT ,
     FirstName VARCHAR(100) NOT NULL,
    LastName VARCHAR(100),
    PhoneNo VARCHAR(10) UNIQUE NOT NULL CHECK  (LENGTH(PhoneNo)=10),
    AssignedVillageID INT NOT NULL UNIQUE,
    DateofJoining DATE NOT NULL,
    CurrentStatus VARCHAR(100) CHECK (CurrentStatus IN ('Active', 'Inactive', 'Transferred', 'Retired')),
    FOREIGN KEY (AssignedVillageID) REFERENCES Village (VillageID),
     FOREIGN KEY (PersonID) REFERENCES Person (PersonID)
);

INSERT INTO Asha (AshaID, PersonID, FirstName, LastName, PhoneNo, AssignedVillageID, DateofJoining, CurrentStatus)
VALUES
(101,2,'Sita','Kumar',9000000002, 1, '2020-06-15', 'Active'),
(102,20, 'Rekha', 'Goud', 9000000020, 2, '2021-04-10', 'Active');
```



AshaID	PersonID	FirstName	LastName	PhoneNo	AssignedVillageID	DateofJoining	CurrentStatus
101	2	Sita	Kumar	9000000002	1	2020-06-15	Active
102	20	Rekha	Goud	9000000020	2	2021-04-10	Active

VillageVisitRecord Table

```
CREATE TABLE VillageVisitRecord(

   VisitID INT PRIMARY KEY,

   AshaID INT NOT NULL,

   VillageID INT NOT NULL,

   DateOfVisit Date NOT NULL,

   PurposeofVisit TINYTEXT,

   Notes TEXT,

   FOREIGN KEY (AshaID) REFERENCES Asha (AshaID),

   FOREIGN KEY (VillageID) REFERENCES Village (VillageID)

);


INSERT INTO VillageVisitRecord (VisitID, AshaID, VillageID, DateOfVisit, PurposeOfVisit, Notes)

VALUES

(1, 101, 1, '2024-01-10', 'General Health Checkup', 'Visited 10 households, checked BP and sugar levels'),

(2, 101, 1, '2024-02-15', 'Pregnancy Follow-up', 'Monitored 2 ongoing pregnancies'),

(3, 101, 1, '2024-03-20', 'Immunization Drive', 'Gave polio drops to children under 5'),
(4, 102, 2, '2024-01-12', 'Family Survey', 'Collected health and sanitation data from new families'),

(5, 102, 2, '2024-02-18', 'Nutrition Awareness', 'Conducted session on child nutrition and anemia'),

(6, 102, 2, '2024-03-22', 'Medicine Distribution', 'Distributed iron and calcium tablets');
```

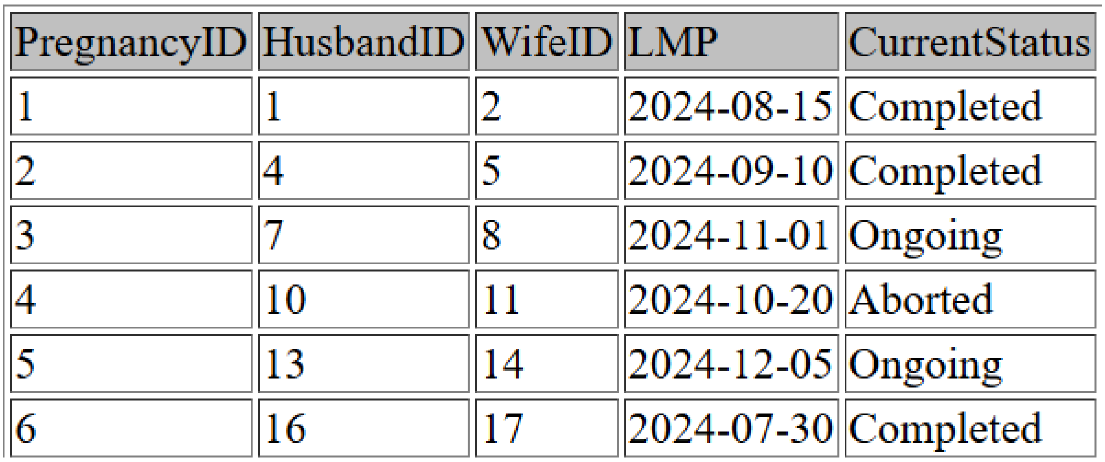


VisitID	AshaID	VillageID	DateOfVisit	PurposeofVisit	Notes
1	101	1	2024-01-10	General Health Checkup	Visited 10 households, checked BP and sugar levels
2	101	1	2024-02-15	Pregnancy Follow-up	Monitored 2 ongoing pregnancies
3	101	1	2024-03-20	Immunization Drive	Gave polio drops to children under 5
4	102	2	2024-01-12	Family Survey	Collected health and sanitation data from new families
5	102	2	2024-02-18	Nutrition Awareness	Conducted session on child nutrition and anemia
6	102	2	2024-03-22	Medicine Distribution	Distributed iron and calcium tablets

PregnancyRecord Table

```
CREATE TABLE PregnancyRecord(

   PregnancyID INT PRIMARY KEY,

   HusbandID INT,

   WifeID INT,

   LMP DATE NOT NULL,

   CurrentStatus VARCHAR(100) CHECK (CurrentStatus IN ('Ongoing', 'Completed', 'Aborted')),

   FOREIGN KEY (HusbandID) REFERENCES Person (PersonID),

   FOREIGN KEY (WifeID) REFERENCES Person (PersonID)

);
```

```
INSERT INTO PregnancyRecord (
    PregnancyID, HusbandID, WifeID, LMP, CurrentStatus
)
VALUES
(1, 1, 2, '2024-08-15', 'Completed'),
(2, 4, 5, '2024-09-10', 'Completed'),
(3, 7, 8, '2024-11-01', 'Ongoing'),
(4, 10, 11, '2024-10-20', 'Aborted'),
(5, 13, 14, '2024-12-05', 'Ongoing'),
(6, 16, 17, '2024-07-30', 'Completed');
```



PregnancyID	HusbandID	WifeID	LMP	CurrentStatus
1	1	2	2024-08-15	Completed
2	4	5	2024-09-10	Completed
3	7	8	2024-11-01	Ongoing
4	10	11	2024-10-20	Aborted
5	13	14	2024-12-05	Ongoing
6	16	17	2024-07-30	Completed

NewBornRecord Table

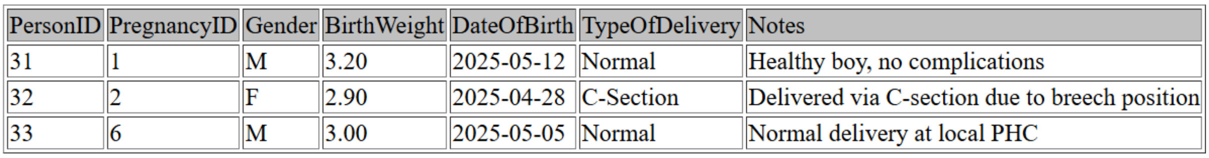
```
CREATE TABLE NewBornRecord(
    PersonID INT,
    PregnancyID INT,
    Gender VARCHAR(1) CHECK (Gender IN ('M', 'F', 'O')),
    BirthWeight DECIMAL(5,2) CHECK (BirthWeight >= 0),
    DateOfBirth DATE NOT NULL,
    TypeOfDelivery VARCHAR(100) NOT NULL CHECK (TypeOfDelivery IN ('Normal', 'C-Section')),
    Notes TEXT,
    PRIMARY KEY (PersonID),
    FOREIGN KEY (PregnancyID) REFERENCES PregnancyRecord(PregnancyID),
    FOREIGN KEY (PersonID) REFERENCES Person(PersonID)
);
```

–TRIGGER

```
DELIMITER $$
CREATE TRIGGER insert_person_for_newborn
BEFORE INSERT ON NewBornRecord
FOR EACH ROW
BEGIN
    DECLARE mother_family_id INT;
    DECLARE mother_id INT;
```

```
   DECLARE father_id INT;
   DECLARE father_phone VARCHAR(15);

   SELECT Mother.FamilyID,Mother.PersonID
   INTO mother_family_id,mother_id
   FROM PregnancyRecord P
   JOIN Person Mother ON P.WifeID = Mother.PersonID
   WHERE P.PregnancyID = NEW.PregnancyID;


   SELECT HusbandID INTO father_id
   FROM PregnancyRecord
   WHERE PregnancyID = NEW.PregnancyID;

   SELECT PhoneNo INTO father_phone
   FROM Person
   WHERE PersonID = father_id;

   INSERT INTO Person (PersonID,FirstName, Gender, DateOfBirth, PhoneNo,
FamilyID,FatherID,MotherID)
   VALUES (New.PersonID,'Baby', NEW.Gender, NEW.DateOfBirth,father_phone,
mother_family_id,father_id,mother_id);
END$$
DELIMITER ;
```

```
INSERT INTO NewBornRecord
(PersonID,PregnancyID,Gender,BirthWeight,DateOfBirth,TypeofDelivery,Notes)

VALUES

(31, 1, 'M', 3.20, '2025-05-12', 'Normal', 'Healthy boy, no complications'),

(32, 2, 'F', 2.90, '2025-04-28', 'C-Section', 'Delivered via C-section due to breech
position'),

(33, 6, 'M', 3.00, '2025-05-05', 'Normal', 'Normal delivery at local PHC');
```



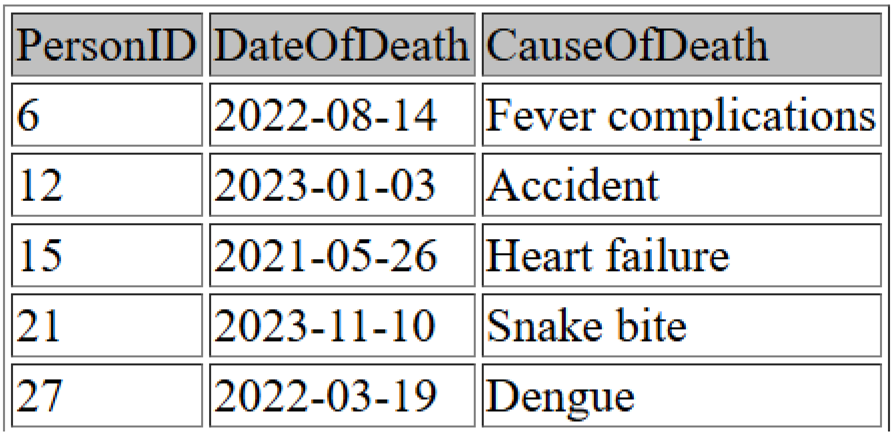
PersonID	PregnancyID	Gender	BirthWeight	DateOfBirth	TypeOfDelivery	Notes
31	1	M	3.20	2025-05-12	Normal	Healthy boy, no complications
32	2	F	2.90	2025-04-28	C-Section	Delivered via C-section due to breech position
33	6	M	3.00	2025-05-05	Normal	Normal delivery at local PHC

DeathRecordTable

```
CREATE TABLE DeathRecord(

    PersonID INT PRIMARY KEY,

    DateOfDeath DATE NOT NULL,

    CauseOfDeath TEXT NOT NULL,

    FOREIGN KEY (PersonID) REFERENCES Person (PersonID)

);
```

–TRIGGER

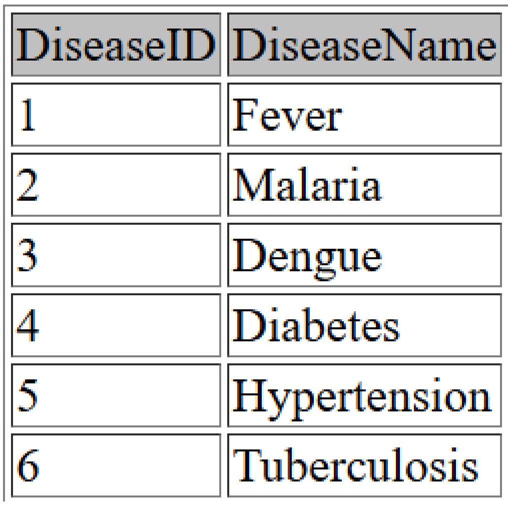
```
DELIMITER $$

CREATE TRIGGER after_insert_death
AFTER INSERT ON DeathRecord
FOR EACH ROW
BEGIN
   UPDATE Person
   SET isDeceased = TRUE
   WHERE PersonID = NEW.PersonID;
END $$

DELIMITER ;

INSERT INTO DeathRecord (PersonID, DateOfDeath, CauseOfDeath) VALUES
(6, '2022-08-14’, 'Fever complications’),
(12, '2023-01-03’, 'Accident’),
(15, '2021-05-26’, 'Heart failure’),
(21, '2023-11-10’, 'Snake bite’),
(27, '2022-03-19’, 'Dengue’);
```



PersonID	DateOfDeath	CauseOfDeath
6	2022-08-14	Fever complications
12	2023-01-03	Accident
15	2021-05-26	Heart failure
21	2023-11-10	Snake bite
27	2022-03-19	Dengue

Disease Table

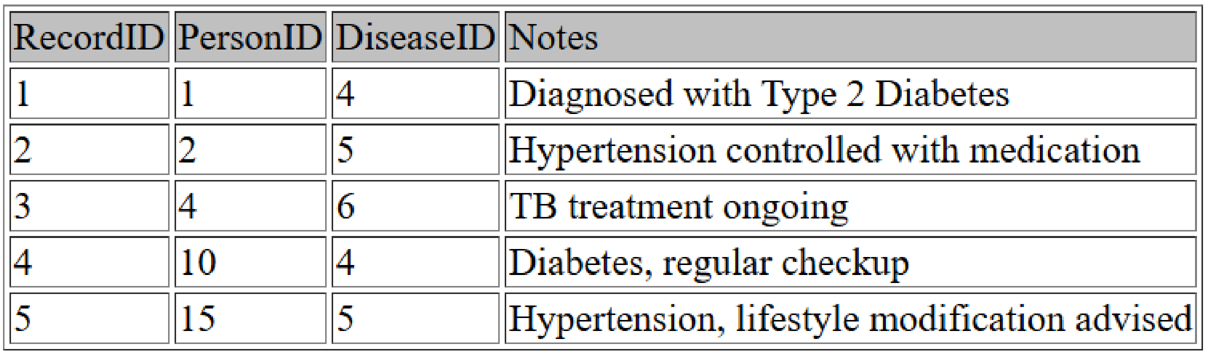
```
CREATE TABLE Disease(

    DiseaseID INT PRIMARY KEY,

    DiseaseName VARCHAR(100) NOT NULL

);


INSERT INTO Disease (DiseaseID, DiseaseName) VALUES

(1, 'Fever'),

(2, 'Malaria'),

(3, 'Dengue'),

(4, 'Diabetes'),

(5, 'Hypertension'),

(6, 'Tuberculosis');
```



DiseaseID	DiseaseName
1	Fever
2	Malaria
3	Dengue
4	Diabetes
5	Hypertension
6	Tuberculosis

ChronicDiseaseRecord Table

```
CREATE TABLE ChronicDiseaseRecord(

 RecordID INT PRIMARY KEY,

    PersonID INT NOT NULL,

    DiseaseID INT NOT NULL,

    Notes TEXT,

    FOREIGN KEY (DiseaseID) REFERENCES Disease (DiseaseID),

    FOREIGN KEY (PersonID) REFERENCES Person (PersonID)

);


INSERT INTO ChronicDiseaseRecord (RecordID, PersonID, DiseaseID, Notes)
VALUES

(1, 1, 4, 'Diagnosed with Type 2 Diabetes'),

(2, 2, 5, 'Hypertension controlled with medication'),
(3, 4, 6, 'TB treatment ongoing'),

(4, 10, 4, 'Diabetes, regular checkup'),

(5, 15, 5, 'Hypertension, lifestyle modification advised');
```



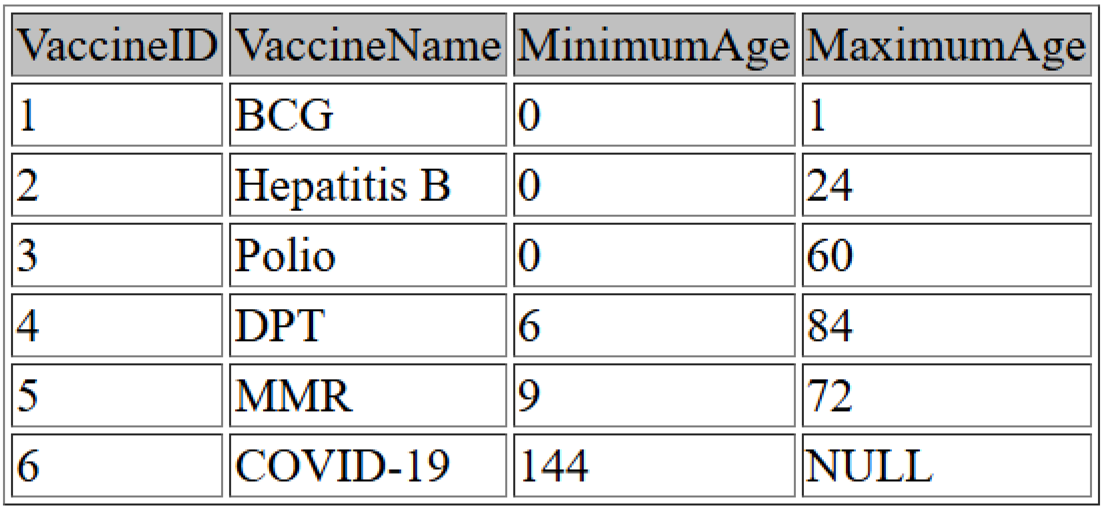
RecordID	PersonID	DiseaseID	Notes
1	1	4	Diagnosed with Type 2 Diabetes
2	2	5	Hypertension controlled with medication
3	4	6	TB treatment ongoing
4	10	4	Diabetes, regular checkup
5	15	5	Hypertension, lifestyle modification advised

<u>Vaccine Table</u>

```
CREATE TABLE Vaccine(

    VaccineID INT PRIMARY KEY,

    VaccineName VARCHAR(100) NOT NULL,

    MinimumAge INT NULL,

    MaximumAge INT NULL

);
```

```
INSERT INTO Vaccine (VaccineID, VaccineName, MinimumAge, MaximumAge)
VALUES
```

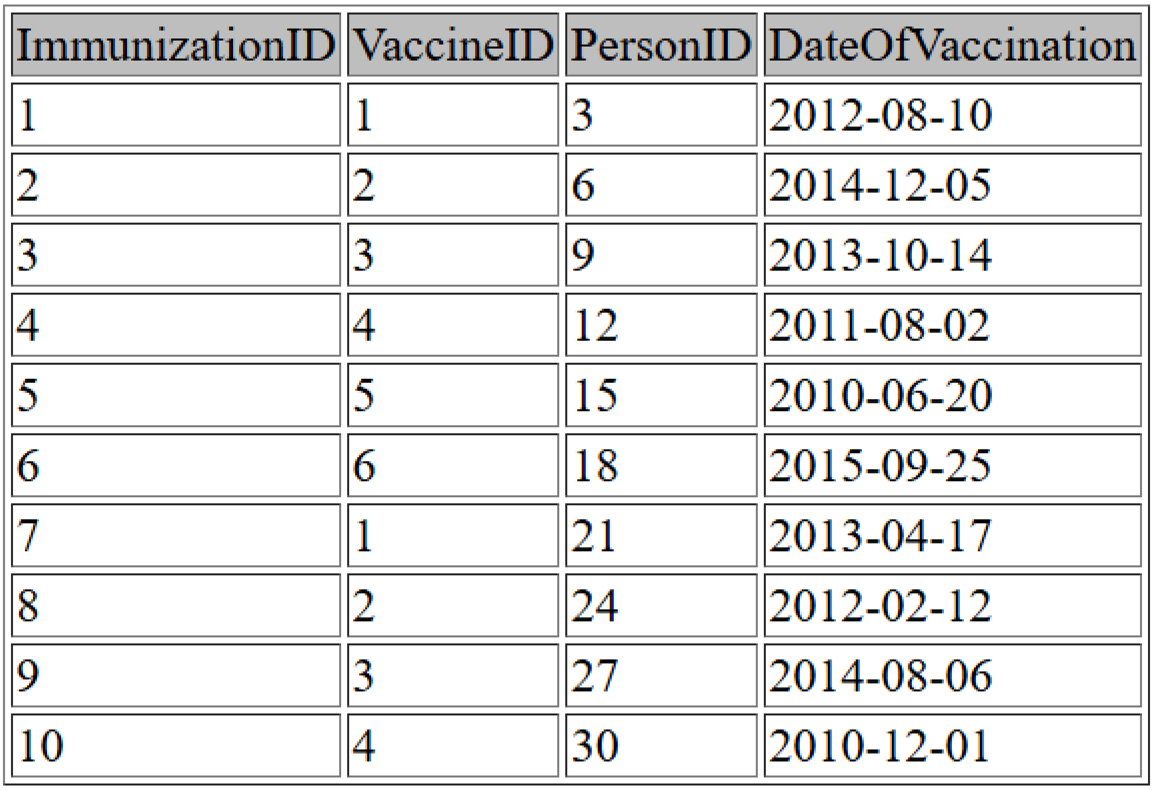
```
(1, 'BCG', 0, 1),

(2, 'Hepatitis B', 0, 24),

(3, 'Polio', 0, 60),

(4, 'DPT', 6, 84),

(5, 'MMR', 9, 72),

(6, 'COVID-19', 144, NULL);
```



VaccineID	VaccineName	MinimumAge	MaximumAge
1	BCG	0	1
2	Hepatitis B	0	24
3	Polio	0	60
4	DPT	6	84
5	MMR	9	72
6	COVID-19	144	NULL

ImmunizationRecord

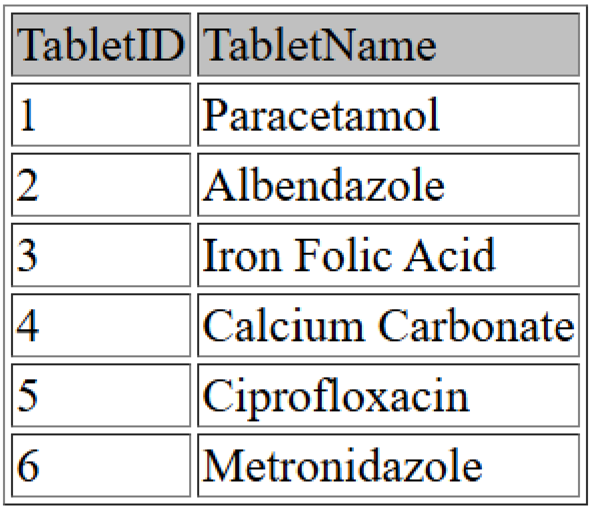
```
CREATE TABLE ImmunizationRecord(
 ImmunizationID INT PRIMARY KEY,
    VaccineID INT NOT NULL,
    PersonID INT NOT NULL,
    DateOfVaccination DATE NOT NULL,
    FOREIGN KEY (PersonID) REFERENCES Person (PersonID),
    FOREIGN KEY (VaccineID) REFERENCES Vaccine (VaccineID)
);

INSERT INTO ImmunizationRecord (ImmunizationID, VaccineID, PersonID,
DateOfVaccination) VALUES
(1, 1, 3, '2012-08-10'),
(2, 2, 6, '2014-12-05'),
(3, 3, 9, '2013-10-14'),
 (4, 4, 12, '2011-08-02'),
(5, 5, 15, '2010-06-20'),
(6, 6, 18, '2015-09-25'),
(7, 1, 21, '2013-04-17'),
(8, 2, 24, '2012-02-12'),
(9, 3, 27, '2014-08-06'),
(10, 4, 30, '2010-12-01');
```



ImmunizationID	VaccineID	PersonID	DateOfVaccination
1	1	3	2012-08-10
2	2	6	2014-12-05
3	3	9	2013-10-14
4	4	12	2011-08-02
5	5	15	2010-06-20
6	6	18	2015-09-25
7	1	21	2013-04-17
8	2	24	2012-02-12
9	3	27	2014-08-06
10	4	30	2010-12-01

Tablet

```sql
CREATE TABLE Tablet(
    TabletID INT PRIMARY KEY,
    TabletName VARCHAR(100) NOT NULL
);

INSERT INTO Tablet (TabletID, TabletName) VALUES
(1, 'Paracetamol'),
(2, 'Albendazole'),
(3, 'Iron Folic Acid'),
(4, 'Calcium Carbonate'),
(5, 'Ciprofloxacin'),
(6, 'Metronidazole');
```



TabletID	TabletName
1	Paracetamol
2	Albendazole
3	Iron Folic Acid
4	Calcium Carbonate
5	Ciprofloxacin
6	Metronidazole

MedicineDistributionRecord

```
CREATE TABLE MedicineDistributionRecord(
 RecordID INT PRIMARY KEY,
    PersonID INT NOT NULL,
    TabletID INT NOT NULL,
    MedicineDistributedDate DATE NOT NULL,
    FOREIGN KEY (TabletID) REFERENCES Tablet (TabletID),
    FOREIGN KEY (PersonID) REFERENCES Person (PersonID)
);

INSERT INTO MedicineDistributionRecord (RecordID, PersonID, TabletID,
MedicineDistributedDate) VALUES
(1, 3, 1, '2024-06-10'),
(2, 6, 2, '2024-06-15'),
(3, 9, 3, '2024-07-01'),
(4, 12, 4, '2024-07-10'),
(5, 15, 5, '2024-07-20'),
(6, 18, 1, '2024-08-05'),
(7, 21, 2, '2024-08-18'),
(8, 24, 3, '2024-09-01'),
(9, 27, 4, '2024-09-14'),
(10, 30, 5, '2024-09-30');
```

RecordID	PersonID	TabletID	MedicineDistributedDate
1	3	1	2024-06-10
2	6	2	2024-06-15
3	9	3	2024-07-01
4	12	4	2024-07-10
5	15	5	2024-07-20
6	18	1	2024-08-05
7	21	2	2024-08-18
8	24	3	2024-09-01
9	27	4	2024-09-14
10	30	5	2024-09-30