## CREATE TABLES

(1) -- PATIENT TABLE

```
CREATE TABLE PatientInformation (
    PatientID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    DateOfBirth DATE,
    Gender VARCHAR(10),
    Address VARCHAR(255),
    PhoneNumber VARCHAR(20),
    EmailAddress VARCHAR(100),
    InsurancePlanID INT
);
```

(2) -- Insurance Table

```
CREATE TABLE InsuranceInformation (
    InsurancePlanID INT PRIMARY KEY,
    PatientID INT,
    PolicyNumber VARCHAR(50),
    PayerName VARCHAR(100),
    CoverageStartDate DATE,
    CoverageEndDate DATE,
    CoPaymentAmount DECIMAL(10, 2),
    DeductibleAmount DECIMAL(10, 2),
    OutOfPocketMaximum DECIMAL(10, 2),
    FOREIGN KEY (PatientID) REFERENCES PatientInformation(PatientID)
);
```

(3) -- Provider Info

```
CREATE TABLE ProviderInfo (
    ProviderID INT PRIMARY KEY,
    ProviderName VARCHAR(100),
    ProviderType VARCHAR(50),
    ProviderNPINumber VARCHAR(20),
    TaxIDNumber VARCHAR(20),
    ProviderAddress VARCHAR(255),
    ContactInformation VARCHAR(100)
);
```

(4) -- Encounter Table

```
CREATE TABLE Encounter (
    EncounterID INT PRIMARY KEY,
    PatientID INT,
    ProviderID INT,
    DateOfEncounter DATE,
    TimeOfEncounter TIME,
    ChiefComplaint VARCHAR(255),
    VisitType VARCHAR(50),
    BillingStatus VARCHAR(20),
    FOREIGN KEY (PatientID) REFERENCES PatientInformation(PatientID),
    FOREIGN KEY (ProviderID) REFERENCES ProviderInfo(ProviderID)
);
```

(5) -- Claim Table

```
CREATE TABLE Claim (
    ClaimID INT PRIMARY KEY,
    EncounterID INT,
    PatientID INT,
    InsurancePlanID INT,
    DiagnosisCodes VARCHAR(255),  -- Assuming multiple codes are stored as a
string
    ProcedureCodes VARCHAR(255),  -- Assuming multiple codes are stored as a
string
    ClaimStatus VARCHAR(20),
    ClaimAmount DECIMAL(10, 2),
    SubmissionDate DATE,
    PaymentDate DATE,
    FOREIGN KEY (EncounterID) REFERENCES Encounter(EncounterID),
    FOREIGN KEY (PatientID) REFERENCES PatientInformation(PatientID),
    FOREIGN KEY (InsurancePlanID) REFERENCES
InsuranceInformation(InsurancePlanID)
);
```

(6) -- ClaimEncounter

```
CREATE TABLE ClaimEncounter (
    ClaimID INT,
    EncounterID INT,
    PRIMARY KEY (ClaimID, EncounterID),
    FOREIGN KEY (ClaimID) REFERENCES Claim(ClaimID),
    FOREIGN KEY (EncounterID) REFERENCES Encounter(EncounterID)
);
```

(7) -- Diagnosis Table

```
CREATE TABLE Diagnosis (
    DiagnosisID INT PRIMARY KEY,
    EncounterID INT,
    MappedDiagnosisCode VARCHAR(20),  -- Assuming ICD-10 codes are stored as
strings
    DiagnosisDescription VARCHAR(255),
    DateOfDiagnosis DATE,
    FOREIGN KEY (EncounterID) REFERENCES Encounter(EncounterID)
);
```

(8) -- Procedure Table

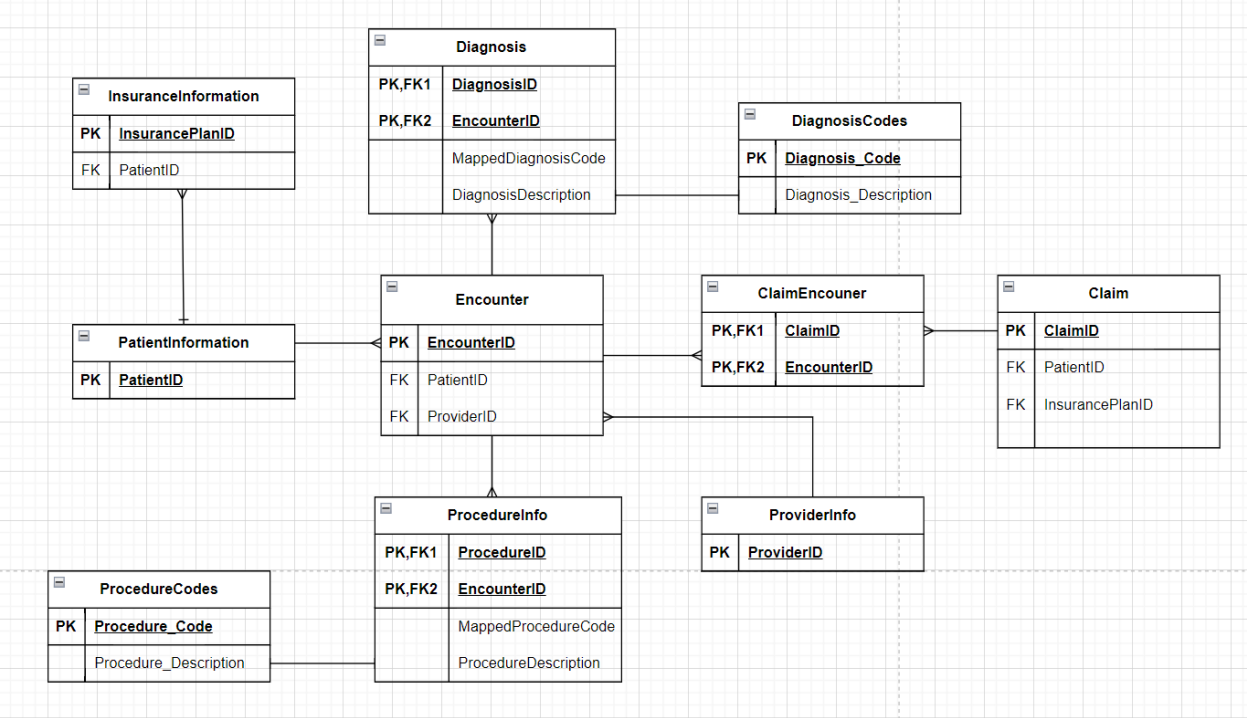
```
CREATE TABLE ProcedureInfo(
    ProcedureID INT PRIMARY KEY,
    EncounterID INT,
    MappedProcedureCode VARCHAR(20),  -- Assuming CPT/HCPCS codes are stored
as strings
    ProcedureDescription VARCHAR(255),
    ProcedureDate DATE,
    ProcedureFee DECIMAL(10, 2),
    FOREIGN KEY (EncounterID) REFERENCES Encounter(EncounterID)
);
```

(9) -- Diagnosis codes

```
CREATE TABLE Diagnosiscodes (
    DiagnosisCode VARCHAR(20) PRIMARY KEY,
    DiagnosisDescription VARCHAR(255)
);
```

(10) -- Procedure codes

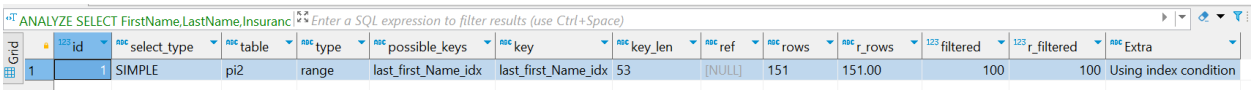
```
CREATE TABLE Procedurecodes (
    ProcedureCode VARCHAR(20) PRIMARY KEY,
    ProcedureDescription VARCHAR(255)
);
```




```
ANALYZE SELECT FirstName,LastName,InsurancePlanID
FROM PatientInformation pi2
WHERE FirstName LIKE "John";
```

```
CREATE INDEX last_first_Name_idx ON Claim(ClaimID);
```

Analyze after index




2nd query

```
SELECT PayerName, CoverageStartDate , CoverageEndDate , CoPaymentAmount ,
DeductibleAmount , OutOfPocketMaximum
FROM InsuranceInformation ii
JOIN PatientInformation pi2
ON ii.PatientID = pi2.PatientID
WHERE PayerName IN ("Home PLC","Cook Group","SMith Inc","Reed Group")
      AND OutOfPocketMaximum > 15000
ORDER BY OutOfPocketMaximum ;
```

SELECT PayerName, CoverageStartDate , CoverageEnd  Enter a SQL expression to filter results (use Ctrl+Space)

| | PayerName | CoverageStartDate | CoverageEndDate | CoPaymentAmount | DeductibleAmount | OutOfPocketMaximum |
|---|---|---|---|---|---|---|
| 1 | Smith Inc | 2023-09-13 | 2021-01-05 | 84 | 8,733 | 15,564 |
| 2 | Smith Inc | 2021-09-13 | 2022-07-28 | 53 | 4,604 | 18,485 |
| 3 | Reed Group | 2021-07-22 | 2022-02-15 | 27 | 7,817 | 22,095 |
| 4 | Smith Inc | 2022-07-18 | 2021-11-20 | 48 | 7,123 | 25,399 |
| 5 | Cook Group | 2023-01-18 | 2023-04-25 | 42 | 9,245 | 27,141 |
| 6 | Smith Inc | 2020-04-18 | 2021-01-11 | 95 | 2,584 | 32,423 |
| 7 | Reed Group | 2020-03-31 | 2021-05-04 | 31 | 7,402 | 85,652 |
| 8 | Cook Group | 2021-04-13 | 2023-11-20 | 92 | 2,496 | 87,234 |

Refresh | Save | Cancel | Export data | 200 | 8 | 8 row(s) fetched - 46ms, on 2023-12-12 at 23:10:27

```
CREATE INDEX PayerName_idx ON InsuranceInformation(PayerName);
```

After

ANALYZE SELECT PayerName, CoverageStartDate , Cov  Enter a SQL expression to filter results (use Ctrl+Space)

| | id | select_type | table | type | possible_keys | key | key_len | ref | rows | r_rows | filtered | r_filtered | Extra |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | SIMPLE | ii | range | PayerName_idx | PayerName_idx | 103 | [NULL] | 9 | 8.00 | 100 | 100 | Using index condition; Using where; Using filesort |

SELECT PayerName, CoverageStartDate , CoverageEndDate  Enter a SQL expression to filter results (use Ctrl+Space)

| | PayerName | CoverageStartDate | CoverageEndDate | CoPaymentAmount | DeductibleAmount | OutOfPocketMaximum |
|---|---|---|---|---|---|---|
| 1 | Smith Inc | 2023-09-13 | 2021-01-05 | 84 | 8,733 | 15,564 |
| 2 | Smith Inc | 2021-09-13 | 2022-07-28 | 53 | 4,604 | 18,485 |
| 3 | Reed Group | 2021-07-22 | 2022-02-15 | 27 | 7,817 | 22,095 |
| 4 | Smith Inc | 2022-07-18 | 2021-11-20 | 48 | 7,123 | 25,399 |
| 5 | Cook Group | 2023-01-18 | 2023-04-25 | 42 | 9,245 | 27,141 |
| 6 | Smith Inc | 2020-04-18 | 2021-01-11 | 95 | 2,584 | 32,423 |
| 7 | Reed Group | 2020-03-31 | 2021-05-04 | 31 | 7,402 | 85,652 |
| 8 | Cook Group | 2021-04-13 | 2023-11-20 | 92 | 2,496 | 87,234 |

Refresh | Save | Cancel | Export data | 200 | 8 | 8 row(s) fetched - 21ms, on 2023-12-12 at 23:14:46

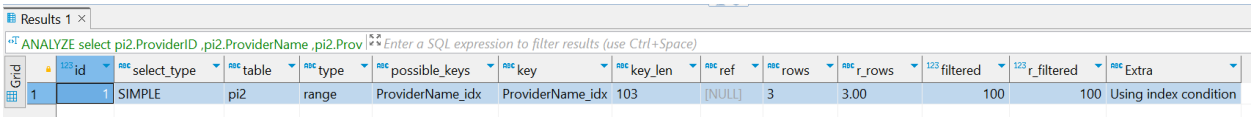
3rd query

```
ANALYZE SELECT
pi2.ProviderID ,pi2.ProviderName ,pi2.ProviderType ,e.DateOfEncounter ,e.Time
OfEncounter ,e.BillingStatus
FROM ProviderInfo pi2
```

```
JOIN Encounter e
ON e.ProviderID =pi2.ProviderID
WHERE ProviderName IN ('Abbott Group','Acosta, Gomez and Bowen','Martin-
Zimmerman')
     AND BillingStatus = 'Not Billed'
```

```
CREATE INDEX ProviderName_idx ON ProviderInfo(ProviderName);
```

After




51ms to 19ms