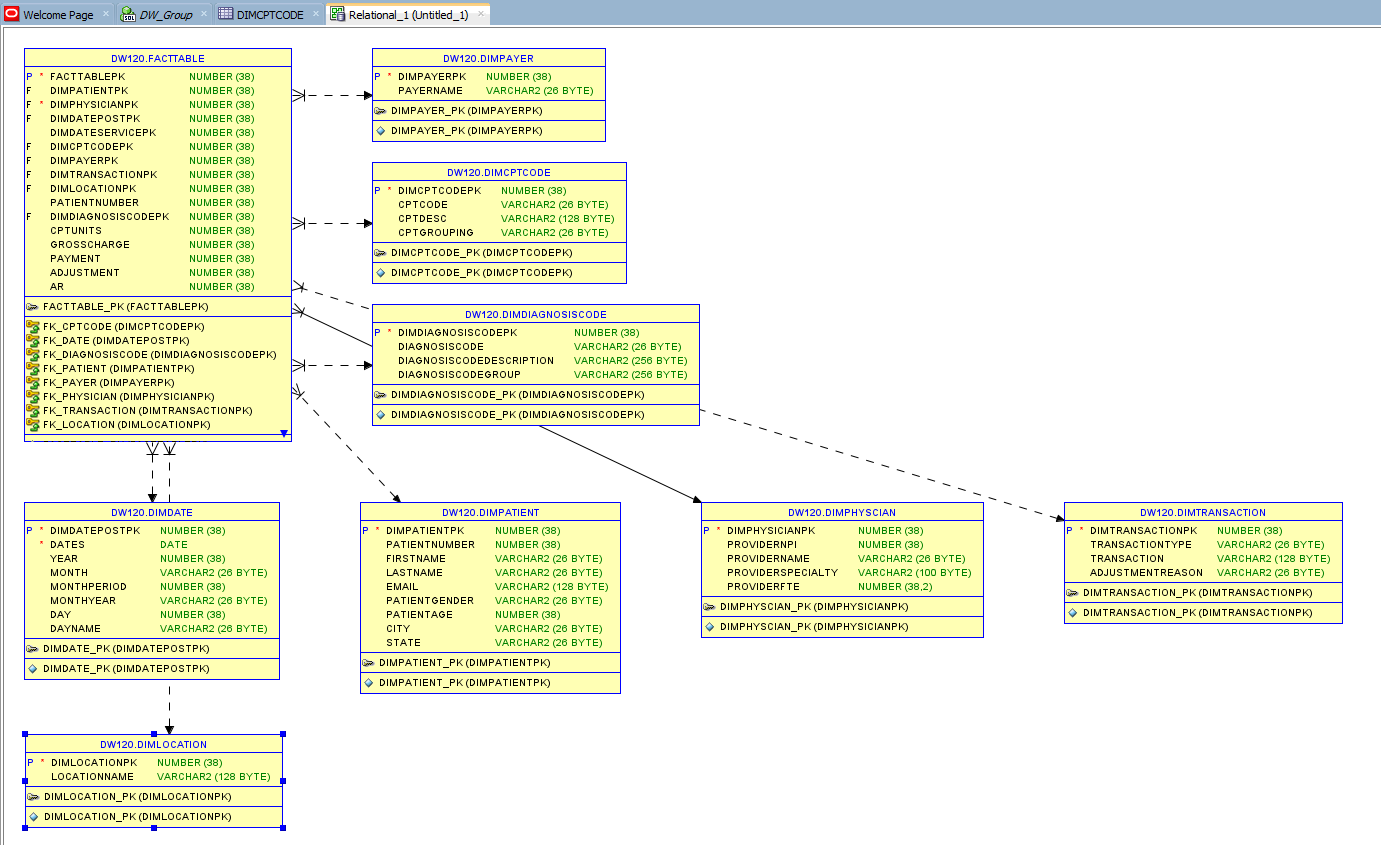
FINAL PROJECT

**Database design:**



The database design follows a star schema, which is typical for data warehouses designed for OLAP (Online Analytical Processing) workloads. Here is a description of the schema components and how they would function within a data warehouse:

**Fact Table**

DW120.FACTTABLE: Central table of the star schema that contains quantitative metrics and keys to connect to dimension tables.

Columns: Include keys to other dimension tables (like DIMPHYSICIANPK, DIMDATEPOSTPK), metrics such as GROSSCHARGE, PAYMENT, ADJUSTMENT, and AR.

**Dimension Tables**

DW120.DIMDATE: Contains date-related attributes that allow analysis over time (time dimension).

DW120.DIMPHYSICIAN: Provides attributes about physicians, which could be used to analyze performance or specialization (provider dimension).

DW120.DIMPATIENT: Holds demographic information about patients which can be used for segmenting and analyzing patient data (patient dimension).

DW120.DIMDIAGNOSISCODE: Stores information on diagnosis codes which is essential for any medical analysis involving patient diagnoses (diagnosis dimension).

DW120.DIMLOCATION: Contains location data where healthcare services are provided, useful for geographical analysis (location dimension).

DW120.DIMTRANSACTION: Tracks different types of financial transactions, their descriptions, and reasons for adjustments (transaction dimension).

DW120.DIMCPTCODE: Stores Current Procedural Terminology (CPT) codes and descriptions for billing purposes (services dimension).

DW120.DIMPAYER: Likely contains information about the payers, such as insurance companies or other entities responsible for payment (payer dimension).

**Data Cubes**

Based on this schema, data cubes can be developed to facilitate multidimensional analysis. For example, a data cube could allow a hospital administrator to analyze GROSSCHARGE and PAYMENT across different dimensions such as time (DIMDATE), location (DIMLOCATION), or physician (DIMPHYSICIAN). This is typically achieved through OLAP operations like slice, dice, roll-up, and drill-down.

**OLTP Design Comparison**

The current schema is optimized for query performance and not for transaction processing. In an OLTP system, the design would focus on normalizing data to reduce redundancy and optimize for frequent updates or insert operations. The tables would be more numerous and highly interconnected, emphasizing data integrity and update efficiency over query speed.

**Queries and Output:**

**1. Healthcare Service Utilization**

**-- Examine the frequency and types of procedures carried out at all times.**

SELECT dimCptCode.CptDesc, COUNT(\*) as ProcedureCount

FROM FactTable

JOIN dimCptCode ON FactTable.dimCPTCodePK = dimCptCode.dimCPTCodePK

GROUP BY dimCptCode.CptDesc

ORDER BY ProcedureCount DESC;

A screenshot of a computer

Description automatically generated

**2. Patient Demographics Analysis**

**-- Understanding the demographics of the patients can be crucial for managerial decisions related to service provision.**

SELECT dimPatient.PatientAge, dimPatient.PatientGender, COUNT(\*) as PatientCount

FROM FactTable

JOIN dimPatient ON FactTable.dimPatientPK = dimPatient.dimPatientPK

GROUP BY dimPatient.PatientAge, dimPatient.PatientGender

ORDER BY dimPatient.PatientAge ;

A screenshot of a computer

Description automatically generated

**3. Financial Analysis**

**-- Investigate the financial aspects of healthcare services, such as average charges, payments, and adjustments.**

SELECT

AVG(FactTable.GrossCharge) as AverageCharge,

AVG(FactTable.Payment) as AveragePayment,

AVG(FactTable.Adjustment) as AverageAdjustment

FROM FactTable;

A screenshot of a computer

Description automatically generated

**4. Patients with Specific Diagnosis and their Visit Count**

**-- This query retrieves the number of visits for patients diagnosed with diabetes.**

SELECT

p.FirstName,

p.LastName,

COUNT(\*) AS VisitCount

FROM

FactTable f

JOIN

dimPatient p ON f.dimPatientPK = p.dimPatientPK

JOIN

dimDiagnosisCode d ON f.dimDiagnosisCodePK = d.dimDiagnosisCodePK

JOIN

dimDate dt ON f.dimDateServicePK = dt.dimDatePostPK

WHERE

d.DiagnosisCodeDescription LIKE '%diabetes%'

GROUP BY

p.FirstName, p.LastName

ORDER BY

VisitCount DESC;

A screenshot of a computer

Description automatically generated

**5. Top 10 Most Common Procedures and Associated Physicians**

**-- This query lists the top 10 most common procedures and the physicians who performed them.**

SELECT

c.CptDesc,

phy.ProviderName,

COUNT(\*) AS ProcedureCount

FROM

FactTable f

JOIN

dimCptCode c ON f.dimCPTCodePK = c.dimCPTCodePK

JOIN

dimphyscian phy ON f.dimPhysicianPK = phy.dimPhysicianPK

GROUP BY

c.CptDesc, phy.ProviderName

ORDER BY

ProcedureCount DESC

FETCH FIRST 10 ROWS ONLY;

A screenshot of a computer

Description automatically generated

**6. Monthly Revenue from Each Department**

**-- This query calculates the monthly revenue generated by each department, assuming that the location represents different departments within the hospital.**

SELECT

l.LocationName,

TO\_CHAR(dt.Dates, 'YYYY-MM') AS Month,

SUM(f.Payment) AS MonthlyRevenue

FROM

FactTable f

JOIN

dimLocation l ON f.dimLocationPK = l.dimLocationPK

JOIN

dimDate dt ON f.dimDateServicePK = dt.dimDatePostPK

GROUP BY

l.LocationName, TO\_CHAR(dt.Dates, 'YYYY-MM')

ORDER BY

l.LocationName, Month;

A screenshot of a computer

Description automatically generated