Overview

The Healthcare Patient Outcome Warehouse schema illustrates a mixed system that comprises:

- **OLTP Layer**: records of patients, diagnoses, and treatments. It keeps raw transactional data from CSV files.
- **Data Warehouse Layer**: tables such as dim_patient, dim_diagnosis, dim_doctor, dim_department, dim_date, and fact_admissions. It is optimized for analytics queries and business intelligence.

The DW uses a Star Schema, the fact_admissions table is the core fact table with descriptive dimensions around it.

Data Dictionary

Table 1: OLTP Tables

Table	Column	Data Type	Description
diagnoses	diagnosis_id	INT (PK, AUTO_INCREMENT)	Unique identifier for each diagnosis.
	diagnosis_code	VARCHAR(50)	Standard medical code for the diagnosis (e.g., ICD-10).
	description	VARCHAR(255)	Text description of the diagnosis.
patients	patient_id	INT (PK, AUTO_INCREMENT)	Unique patient identifier.
	first_name	VARCHAR(255)	Patient’s first name.
	last_name	VARCHAR(255)	Patient’s last name.
	gender	ENUM('Male','Female','Other')	Patient gender.

	age	INT	Patient's age at admission.
treatments	admission_id	INT (PK, AUTO_INCREMENT)	Unique identifier for each admission/treatment.
	patient_id	INT (FK → patients.patient_id)	References the patient receiving treatment.
	doctor_name	VARCHAR(255)	Name of the attending doctor.
	department	VARCHAR(100)	Department responsible for treatment (e.g., Cardiology).
	diagnosis_id	INT (FK → diagnoses.diagnosis_id)	References the diagnosis related to this treatment.
	admission_date	DATETIME	Timestamp when the patient was admitted.
	treatment_cost	DECIMAL(10,2)	Cost of the treatment/procedure.

Table 2: Data Warehouse Tables

Table	Column	Data Type	Description
dim_patient	patient_id	INT (PK)	Surrogate key (same as OLTP).
	first_name	VARCHAR(255)	Patient first name.

	last_name	VARCHAR(255)	Patient last name.
	gender	ENUM('Male','Female','Other')	Gender.
	age	INT	Age.
	age_group	VARCHAR(20)	Derived attribute categorizing patients: Child, Adult, or Senior.
dim_diagnosis	diagnosis_id	INT (PK)	Surrogate key (same as OLTP).
	diagnosis_code	VARCHAR(50)	Diagnosis code.
	diagnosis_desc	VARCHAR(255)	Text description.
	category	VARCHAR(100)	Derived classification (e.g., Cardiology, Respiratory, General).
dim_doctor	doctor_id	INT (PK, AUTO_INCREMENT)	Unique doctor key.
	doctor_name	VARCHAR(255)	Full name of doctor.
	specialization	VARCHAR(100)	Medical department or field.

dim_department	department_id	INT (PK, AUTO_INCREMENT)	Unique department key.
	department_name	VARCHAR(100)	Name of hospital department.
	location	VARCHAR(100)	Location or floor (optional field, can be null).
dim_date	date_id	VARCHAR(8) (PK)	Surrogate date key formatted as YYYYMMDD.
	full_date	DATE	Calendar date.
	day	INT	Day of month.
	month	INT	Month number (1–12).
	month_name	VARCHAR(20)	Full month name (e.g., January).
	quarter	VARCHAR(2)	Quarter of the year (1–4).
	year	INT	Calendar year.

	weekday	VARCHAR(10)	Name of weekday (e.g., Monday).
fact_admissions	admission_id	INT (PK)	Unique admission record (from OLTP).
	patient_id	INT (FK → dim_patient.patient_id)	Links to patient dimension.
	doctor_id	INT (FK → dim_doctor.doctor_id)	Links to doctor dimension.
	diagnosis_id	INT (FK → dim_diagnosis.diagnosis_id)	Links to diagnosis dimension.
	department_id	INT (FK → dim_department.department_id)	Links to department dimension.
	date_id	VARCHAR(8) (FK → dim_date.date_id)	Links to date dimension.
	treatment_cost	DECIMAL(10,2)	Measure field for analysis (e.g., total revenue, cost per admission).

Design Decisions Explanation

The data warehouse utilizes a Star Schema architecture, which gives prominence to the main fact table (fact_admissions) in the middle capturing the metrics of trading events of the business like hospital admissions together with their cost of the attached treatment. Numerous dimension tables storing the descriptive attributes such as patient demographics, diagnosis information, doctor specialization, department, and time-based data surround the central table. This arrangement provides an opportunity to conduct a flexible, multi-dimensional analysis of the hospital data and to perform OLAP-style analytics such as averaging the treatment cost per department, counting admissions per diagnosis category, analyzing revenue trends by month or quarter, and examining patient demographics by disease type. The Star Schema design helps in breaking down complex queries by reducing joins, thus increasing the query performance, and at the same time, giving a clear and intuitive framework for analytical reporting and decision-making.

By this schema, the separation of data between the OLTP (Online Transaction Processing) and Data Warehousing (DW) layers guarantees both efficiency in operations and performance in analytics. The OLTP tables (patients, diagnoses, and treatments) are subjected to normalization in order to uphold data integrity, minimize repetition, and optimize data entry operations. Conversely, the Data Warehouse tables are denormalized to boost analytical queries performance. This separation permits the transactional systems to operate efficiently in handling the day-to-day activities while the warehouse overlaps with complex reporting and trend analysis without affecting OLTP performance.

Partially derived attributes were applied to certain dimension tables to enhance the analytical potential of the data. The dim_patient table, for instance, holds an age_group column that delineates and categorizes patients into three major groups such as “Child,” “Adult,” and “Senior,” thereby making screening demographic easier during the analysis process. The same way, the dim_diagnosis table offers a category field that provides a classification of diagnoses into groups like “Infectious Disease,” “Respiratory,” and “Cardiology,” thus making reporting on the medical and financial side easier. The dim_date table comes with different attributes such as day, month, quarter, and weekday, which allows for the employing of only basic date functions without a heavy dependency on complex date functions during queries for performing time-series analysis.

A schema with surrogate keys plus foreign key relationships is utilized for the purpose of assuring data consistency and referential integrity throughout all the tables. Each

dimension table possesses a primary key that is unique, and that serves as a stable reference point even when the data from the source undergoes changes. The fact table has these surrogate keys as foreign keys, hence, each admission record is connected to its corresponding patient, doctor, diagnosis, department, and date via this link. By this method, data consistency is guaranteed, and at the same time, join operations between the fact and dimension tables are rendered efficient.

Scalability and extensibility were the primary considerations when designing the schema with the idea that the company might alter its requirements or grow in the future. So, with no need to completely redesign the schema, the new features like the hospital branch, treatment type, or insurance provider, etc., can be added with ease. In addition, the performance of large-scale queries is significantly improved by indexing the fields that are very often joined, such as doctor_name, department_name, and date_id.

The ETL (Extract, Transform, Load) process, in the end, not only makes data preparation easier but also guarantees data quality. Initially, raw data coming from CSV files undergoes cleaning and is subsequently, using Python scripts, loaded onto the OLTP tables. The tables are then used for staging data transformation, where SQL scripts are utilized to populate the Data Warehouse dimension and fact tables using INSERT INTO ... SELECT statements. This procedure guarantees that all derived characteristics are uniformly applied, and the ties between the entities are correctly established. The design, in general, accomplishes a compromise between normalization for data integrity in the OLTP layer and denormalization for analytical efficiency in the Data Warehouse, which in turn results in a resilient and scalable system that caters to both operational and strategic decision-making needs.