Data Warehousing Agenda

• **Overview**
  – Benefits to Warehousing
  – Defining an Approach
  – Define Key Terms

• **Example**
  – Planning to Warehouse
  – Designing a Warehouse
  – Loading the Warehouse
  – Reporting from the Warehouse
Scope

- Investigate data warehousing, and its applicability to the healthcare industry.
- Identify problems that data warehouses can be designed to solve.
- Build a data warehouse from a simplified EHR data model.

- **Disclaimer: This presentation**
  - will be technical
  - was designed to be applicable to a broad audience
  - will utilize an over-simplified, contrived example
Poll Question 1
Data Warehousing

OVERVIEW
What is Data Warehouse?

• A database
  – Used for reporting / data analytics.
  – Central repository for data
  – Created by integrating data from one or more disparate sources.

• Warehouse as a base
  – Executive dashboards
  – Auditing tools
  – Marketing analysis tools

• ETL describes the process of loading a data warehouse:
  – E: Extract data from outside sources
  – T: Transform (cleanse, normalize, translate) data to fit operational needs
  – L: Load data into the target database
Key Data Warehousing Terms:

- **Aggregation:** Values of multiple rows are grouped together as input on certain criteria to form a single value of more significant meaning
  - *Examples:* Average, Count, Maximum, Minimum, Sum, etc.

- **Normalization:** The process of organizing the fields and tables of a relational database to minimize redundancy and dependency.
  - Divide large table(s) into smaller (and less redundant) tables
  - Define relationships between these tables
  - De-normalize: purposefully don’t do or undo this work for performance or other reasons

- **Contention:** Database resource (row/table/page) locking resulting from one operation inhibits a second, concurrent, operation from completing, at least until the first operation is successful.
  - Deadlock occurs when contention for resources cannot be resolved
EHR Data Models

• **EHR databases were designed for transaction processing**
  – Designed to meet the EHR application’s needs
  – Not designed for reporting

• **Don’t be intimidated by EHR data volume**
  – Many of our clients have very large EHR databases
  – Collapse/summarize data before storage
    • Save space
    • Enable faster reporting
Benefits of Implementing a Data Warehouse

**Business Reasons:**
- Enable high-level (aggregate/summarized) reporting
- Enable ad-hoc (custom, end user) reporting
- Integrate data from multiple sources (multiple EHRs or EHR+PM)
- Improve data quality

**Technical Reasons:**
- Archival of data
- Maintain historic views of operational data
- Restructure data to make sense for business users (enable ad-hoc reporting)
- Restructure data model for performance, without impacting operational systems
- Alleviate contention – don’t use live transaction processing systems for analytics
A Step-by-Step Approach to Warehousing

• Present the approach

• Define a simple Electronic Health Record (EHR) operational data model

• Ask questions the model doesn’t answer directly
  – Assume aggregation would have performance implications for EHR users
  – Assume contention/deadlock are not acceptable in operational environment

• Build a data warehouse to answer these questions
  – Model warehouse
  – Load warehouse
  – Template reports off the warehouse
Step-by-Step Approach to Data Warehousing (Diagram)

1. Reporting Requirements

2. Source Data Requirements

3. Conceptually, Logically, & Physically Model Source Data

4. Develop ETL Processes To Load Warehouse

5. Develop Reports
EHR Data Model

Patient
- PatientId PK
- MedicalRecordNumber (MRN)

Person
- PersonId PK
- Name
- DateOfBirth
- Gender

Provider
- ProviderId PK
- NationalProviderIdentifier (NPI)

VitalSignReading
- VitalSignReadingId PK
- BloodPressureSystolic
- BloodPressureDiastolic
- PulseRatePerMinute
- TemperatureFahrenheit

Appointment
- AppointmentId PK
- PatientId FK
- ProviderId FK
- Date
- ActualStartTime
- ScheduledStartTime

ChargeDictionary
- ChargeDictionaryId PK
- ChargeDescription
- DollarAmount
- ChargeDictionaryId FK
- VitalSignReadingId FK
Data Warehousing

PRACTICE

REPORTING REQUIREMENTS
Reporting Requirements

- Assume the following questions have recently been asked:
  - Which patients have seen a rise in both their average blood pressure, and average pulse rate, since last year?
  - Which providers are habitually late to appointments?
  - Enable top performer awards – determine total revenue, per provider, per year.
  - Determine advertising campaign target audience:
    - Which gender provided more revenue last year?
      - Were these patients generally under or over 50 years of age?
Reporting Requirements

• **The data exists in the EHR data model to answer these questions**
  – Reporting against the EHR model would be very inefficient
  – Aggregation could cause performance implications for EHR users
    • Overall system latency
    • Errors attempting to update/save patient information
    • Inability to view vital patient information

• **Solution:**
  – Decide to develop a data warehouse to fulfill these reporting requirements
  – Reporting requirements defined
    • next step is to identify the data necessary to fulfill these requirements
Poll Question 2
**Data Warehousing**

**PRACTICE**

**SOURCE DATA REQUIREMENTS**

1. Reporting Requirements
2. Source Data Requirements
3. Conceptually, Logically, & Physically Model Source Data
4. Develop ETL Processes To Load Warehouse
5. Develop Reports
Data Requirements

• **Associated Reporting Requirement:**
  – Which patients have seen a rise in both their average blood pressure, and average pulse rate, since last year?

• **Necessary Source Data Element(s):**
  – *Patient* and *Person* demographic information
  – *VitalSignReading* metrics
    • BloodPressureSystolic
    • BloodPressureDiastolic
    • PulseRatePerMinute
  – *Appointment*: just to get the *Appointment* date of the *VitalSignReading*
Data Requirements

- **Associated Reporting Requirement:**
  - Which providers are habitually late to appointments?

- **Necessary Source Data Element(s):**
  - *Provider* demographic information
  - *Appointment* Information:
    - ScheduledStartTime
    - ActualStartTime

- **Unnecessary Source Data:**
  - There was a patient who had to wait
  - Bring over the associated *Patient* just in case we want to report on it in the future
Data Requirements

• **Associated Reporting Requirement:**
  – Enable top performer awards – determine total revenue, per provider, per year.

• **Necessary Source Data Element(s):**
  – *Provider* and *Person* demographic information
  – *Appointment*:
    • Get date of the charge
    • Reference the associated charge
  – Get *ChargeDictionary DollarAmounts* for all referenced Charges
Data Requirements

• **Associated Reporting Requirement:**
  – Determine advertising campaign target audience:
    • Which gender provided more revenue last year?
    • Were these patients generally under or over 50 years of age?

• **Necessary Source Data Element(s):**
  – *Patient* and *Person* demographic information (age and gender)
  – *Appointment*:
    • *Get date to determine year*
    • Get the associated *charge*
  – Get *ChargeDictionary DollarAmounts* for all referenced Charges
Data Requirements (Combined)

- **Necessary source data elements for all reporting requirements:**
  - *Patient, Provider* and *Person* demographic information
  - *Appointment*:
    - Date of the *VitalSignReading*
    - *ScheduledStartTime* and *ActualStartTime* (or just the difference between them)
    - Date of the Charge
  - Get *ChargeDictionary DollarAmounts* for all referenced Charges
  - *VitalSignReading* metrics
    - BloodPressureSystolic
    - BloodPressureDiastolic
    - PulseRatePerMinute
Data Warehousing

PRACTICE

CONCEPTUAL, LOGICAL, & PHYSICAL DATA MODELS
Star Schema Database Architecture

- **Simplest data mart schema**
- **One or more fact tables**
  - Metric data and references
- **Reference any number of dimensions**
  - Descriptive data
  - Fewer records, many attributes
- **Preserve inner join**
  - Easier for ad-hoc reporting
- **Add new descriptive data at any time**
  - Easily add “slices”/“snapshotting” to an existing warehouse
Conceptual Data Model

• Define the necessary data constructs:
  – Descriptive/“Dimensional” information:
    • Patient
    • Provider
    • Date
  – Factual Information by Grain:
    • At the Patient and Date Granularity:
      – Vital Sign Metrics
    • At the Patient, Provider, and Date Granularity:
      – Appointment Tardiness
      – Charge
Vital Sign Data Model (Logical, Star Architecture)

**Patient**
- **PK** PatientId
  - MedicalRecordNumber
  - Name
  - Gender
  - OverAgeFiftyFlag

**Date**
- **PK** Dateld
  - Day
  - Month
  - Year

**VitalSignFact**
- **FK** Dateld
- **FK** PatientId
  - BloodPressureSystolic
  - BloodPressureDiastolic
  - PulseRatePerMinute
Charged Appointment Data Model (Logical, Star Architecture)

- **Patient**
  - PK: PatientId
  - MedicalRecordNumber
  - Name
  - Gender
  - OverAgeFiftyFlag

- **Provider**
  - PK: ProviderId
  - NationalProviderIdentifier
  - Name

- **Date**
  - PK: DateId
  - Day
  - Month
  - Year

- **ChargedAppointmentFact**
  - PK: DateId
  - FK: PatientId
  - FK: ProviderId
  - AppointmentChargeAmount
  - AppointmentTardinessInMinutes

This model represents a logical, star architecture, where one central fact table (ChargedAppointmentFact) is linked to three dimension tables (Patient, Provider, Date) via foreign keys (FK).
Logical / (Pseudo) Physical Data Model

Patient
- **PK**: PatientId
- MedicalRecordNumber
- Name
- Gender
- OverAgeFiftyFlag

Date
- **PK**: DateId
- Day
- Month
- Year

Provider
- **PK**: ProviderId
- NationalProviderIdentifier
- Name

VitalSignFact
- **FK**: DateId
- **FK**: PatientId
- BloodPressureSystolic
- BloodPressureDiastolic
- PulseRatePerMinute

ChargedAppointmentFact
- **FK**: PatientId
- **FK**: DateId
- **FK**: ProviderId
- AppointmentChargeAmount
- AppointmentTardinessInMinutes
Poll Question 3
PRACTICE

Data Warehousing

1. Reporting Requirements

2. Source Data Requirements

3. Conceptually, Logically, & Physically Model Source Data

4. Develop ETL Processes To Load Warehouse

5. Develop Reports

ETL PROCESSING
Extract, Transform, Load

• **Extract Stage**
  - Select necessary data from the source system
  • Potential Methods:
    - Mirror source system “a copy of environment” to use as a source system
    - Script export of factual data, and all necessary descriptive information from source system - Denormalized, “flat file” export
      » Avoid aggregate functions
  • Any validation logic is applied here
  • Exported data is loaded into the “stage”
Extract, Transform, Load

- **Transform Stage**
  - Extracted data has been loaded into a staging area
  - Common transformations applied to source data within the stage
    - Aggregate/summarize/collapse/rollup data
    - Derive new data from source data
    - Selectively determine what data to load
    - Join data from multiple sources
    - Normalize free form values
    - Transpose/Pivot data
    - Apply pre-defined mappings to source data
Extract, Transform, Load

- **Load Stage**
  - Select transformed data from the stage
  - Dependent upon the warehouse, the load processes may include
    - Updating reference/descriptive/dimensional data
    - Inserting non-overlapping data
    - Upserting/Merging overlapping data
    - Purge old/obsolete data from the warehouse
    - Clean up staging area
ETL for the Example

- **Extract:**
  - The Vital Extract:
    - Select name, gender, MRN, blood pressure, pulse rate, and date
    - From Person/Patient, VitalSignReading, and Appointment
  
  - The Charge/Appointment Extract:
    - Select provider information, patient information, appointment scheduled and actual start times, dollar amount, and date
    - From Person/Patient, Person/Provider, Appointment, and ChargeDictionary
  
  - Load extracted data into a staging area designed to hold the results of these queries

  - Note: this is a simple, denormalized, “flat-file” extract
ETL for the Example

-- The Vitals Extract

```
SELECT
  p.Name, p.Gender, p.DateOfBirth, pa.MedicalRecordNumber,
  v.BloodPressureSystolic, v.BloodPressureDiastolic, v.PulseRatePerMinute,
  a.Date
FROM Person p
INNER JOIN Patient pa ON pa.PersonId = p.PatientId
INNER JOIN VitalSignReading v ON v.PatientId = pa.PatientId
INNER JOIN Appointment a ON a.VitalSignReadingId = v.VitalSignReadingId
WHERE a.Date > @LastExportDateTime
```

-- The Charge/Appointment Extract

```
SELECT
  p1.Name as ProviderName, pr.NationalProviderIdentifier,
  p2.Name as PatientName, p2.Gender, pa.MedicalRecordNumber,
  a.Date, a.ScheduledStartTime, a.ActualStartTime,
  cd.DollarAmount
FROM Appointment a
INNER JOIN Provider pr ON a.ProviderId = pr.ProviderId
INNER JOIN Person p1 ON pr.ProviderId = p1.PersonId
INNER JOIN Patient pa ON a.PatientId = pa.PatientId
INNER JOIN Person p2 ON pa.PatientId = p2.PersonId
INNER JOIN ChargeDictionary cd ON cd.ChargeDictionaryId = a.ChargeDictionaryId
WHERE a.Date > @LastExportDateTime
```
Transform Extracted Example Data

• **From Vitals Export**
  – Derive OverAgeFiftyFlag from the DateOfBirth and the Date from the Appointment
  – Translate Date to Day, Month, and Year numeric values

• **From Charge/Appointment Export**
  – Derive AppointmentTardiness from ScheduledStartTime and ActualStartTime
  – Translate Date to Day, Month, and Year numeric values
Load Extracted Dimension / Reference Data

• **For Each Record From Vitals Extract:**
  – Insert any new Day/Month/Year combinations to the Date table from staging
    • Update staging record with DateId
  – Update/Insert to Patient table from transformed staging data
    • Update staging record with PatientId
  – Update/Insert the transformed metrics from staging into VitalSignFact
    • Use the DateId and PatientId found above
Load Extracted Dimension / Reference Data

- **For Each Record From Charge/Appointment Extract:**
  - Insert any new Day/Month/Year combinations to the Date table from staging
    - Update staging record with Dateld
  - Update/Insert to Provider table from transformed staging data
    - Update staging record with ProviderId
  - Update/Insert to Patient table from transformed staging data
    - Update staging record with PatientId
  - Update/Insert the transformed metrics from staging into ChargedAppointment
    - Use the Dateld, ProviderId, and PatientId found above
Data Warehousing

PRACTICE

REPORTING

1. Reporting Requirements
2. Source Data Requirements
3. Conceptually, Logically, & Physically Model Source Data
4. Develop ETL Processes To Load Warehouse
5. Develop Reports
Logical / (Pseudo) Physical Data Model

**Patient**
- **PatientId** (PK)
- MedicalRecordNumber
- Name
- Gender
- OverAgeFiftyFlag

**Date**
- **DateId** (PK)
  - Day
  - Month
  - Year

**Provider**
- **ProviderId** (PK)
  - NationalProviderIdentifier
  - Name

**VitalSignFact**
- **DateId** (FK)
- **PatientId** (FK)
  - BloodPressureSystolic
  - BloodPressureDiastolic
  - PulseRatePerMinute

**ChargedAppointmentFact**
- **PatientId** (FK)
- **DateId** (FK)
- **ProviderId** (FK)
  - AppointmentChargeAmount
  - AppointmentTardinessInMinutes
Report On Warehouse

- **Original Reporting Requirement:**
  - Which patients have seen a rise in both their average blood pressure, and average pulse rate, since last year?
  - Or any interval?

- **Report Design:**
  - Base Query
    - Select from VitalSignFact
    - Use Join to Date to collapse or “group by” the Year
    - Aggregate/Average Blood Pressure and Pulse metrics
  - Join Base query to itself on b1.patient=b2.patient and b1.year=b2.year-1
    - Filter where b1 vital signs are greater than b2 vital signs
Report On Warehouse

• **Original Reporting Requirement:**
  – Which providers are habitually late to appointments?
  – Where habitually means over 15 minutes late on average

• **Report Design:**
  – Using ChargedAppointmentFact and Provider
  – Select Provider Name
  – Group By ProviderId
  – Where Average Tardiness > 15

• **Only care about tardiness this month or year?**
  – Join ChargedAppointmentFact to Date, filter on Month/Year
Report On Warehouse

• **Original Reporting Requirement:**
  – Enable top performer awards – determine total revenue, per provider, per year.
  – Make it work on a monthly scale, too…

• **Report Design:**
  – Using ChargedAppointmentFact and Provider
  – Select Provider Name and Row_Number()
  – Group By ProviderId
  – Sort by Sum of AppointmentChargeAmount descending
  – Join ChargedAppointmentFact to Date, filter on Month/Year
Report On Warehouse

• **Original Reporting Requirement:**
  – Determine advertising campaign target audience:
    • Which gender provided more revenue last year?
    • Were these patients generally under or over 50 years of age?

• **Report Design:**
  – Using ChargedAppointmentFact and Patient
  – Select Patient Name and Row_Number()
  – Group By Gender and/or OverAgeFiftyFlag
  – Sort by Sum of AppointmentChargeAmount descending
  – Join ChargedAppointmentFact to Date, filter on Month/Year
Data Warehousing

SUMMARY
Data Warehouse Development Complete

- Using Stakeholder requirements, we’ve successfully used EHR source database to:
  - Identify necessary data
  - Conceptually, and Logically model data
  - Design (physical data model) and develop a data warehouse
  - Develop ETL process to load warehouse from the EHR source database
  - Define flexible reports to fulfill stakeholder reporting requirements
Was This Exercise Necessary?

• Could reporting requirements have been fulfilled against the source EHR system?
  – Yes, but aggregation would cause contention and hinder EHR performance

• So why not just report off a copy of the source system?
  – That alleviates the EHR performance concerns
  – Queries are generally unnecessarily complicated
    • more difficult for end user ad-hoc reporting
  – However, the source system does not efficiently enable the aggregation/summarization of vital sign metrics, nor dollar/revenue
    • Reporting queries would become increasingly slow as tables grow
Does this Contrived Data Warehouse provide Benefit?

**Business Reasons:**
- High-level (aggregate) reporting
- Ad-hoc (custom, end user) reporting

**Technical Reasons:**
- Restructure data to make sense for business users (enable ad-hoc reporting)
- Restructure data for performance, without impacting operational systems
- Alleviated the contention cause by running analytic queries against transaction processing systems
Data Warehousing

• Thank you for joining us today
• You may contact us through our website at:
  – http://www.galenhealthcare.com