SQL GROUP ASSIGNMENT

28. Write a short essay talking about your understanding of transactions, locks and isolation levels.

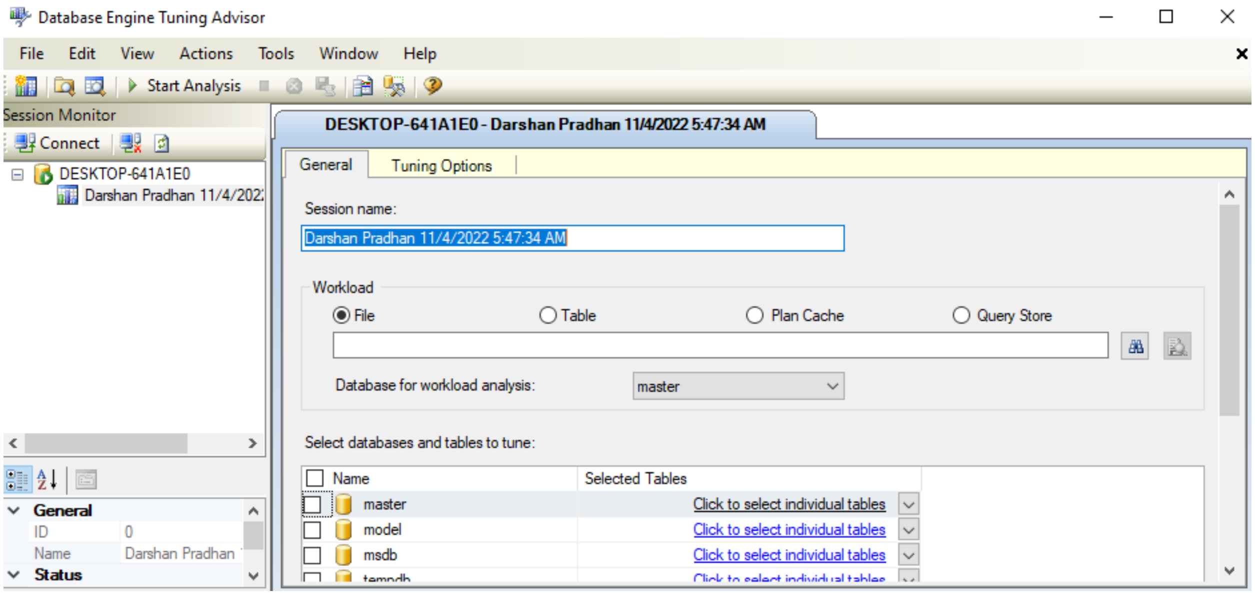
ANS) Transactions in SQL are a single or consecutive set of processes completed in a rational definite flow. If a transaction is successful, all of the data modifications made during the transaction are committed and become a permanent part of the database. If a transaction encounters errors and must be canceled or rolled back, then all of the data modifications are erased.

Locks are held on SQL Server resources, such as rows read or modified during a transaction, to prevent concurrent use of resources by different transactions. For example, if an exclusive (X) lock is held on a row within a table by a transaction, no other transaction can modify that row until the lock is released. Minimizing locks increases concurrency, which can improve performance. Multiple instances of the Locks object can be monitored at the same time, with each instance representing a lock on a resource type.

Isolation levels control the locking and row versioning behavior of Transact-SQL statements issued by a connection to SQL Server. There are 5 different parameters for isolation levels and they are as follows: Read Uncommitted, Read Committed (system default), Repeatable Read, Serializable, Snapshot.

29. Write a short essay, plus screenshots talking about performance tuning in SQL Server. Must include Tuning Advisor, Extended Events, DMV, Logs and Execution Plan.

ANS) The Microsoft Database Engine Tuning Advisor (DTA) analyzes databases and makes recommendations that you can use to optimize query performance. You can use the Database Engine Tuning Advisor to select and create an optimal set of indexes, indexed views, or table partitions without having an expert understanding of the database structure or the internals of SQL Server.



DESKTOP-641A1E0 - Darshan Pradhan 11/4/2022 5:47:34 AM

General: ID 0, Name Darshan Pradhan

Session name: Darshan Pradhan 11/4/2022 5:47:34 AM

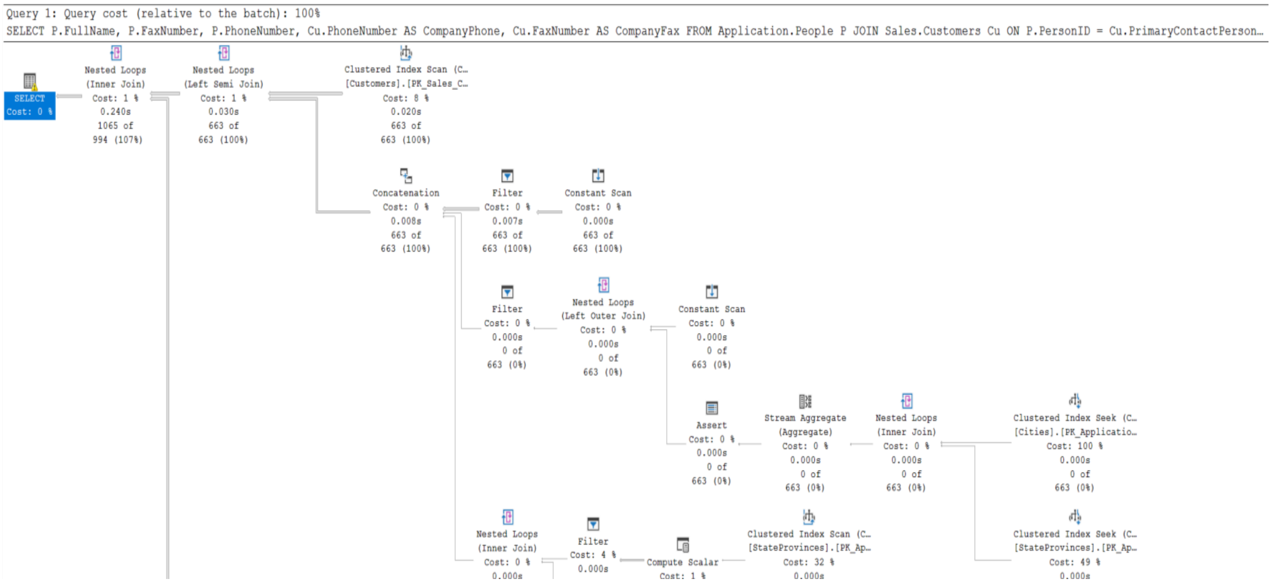
Workload: ● File ○ Table ○ Plan Cache ○ Query Store

Database for workload analysis: master

Select databases and tables to tune:

□ Name	Selected Tables
□ master	Click to select individual tables
□ model	Click to select individual tables
□ msdb	Click to select individual tables
□ tempdb	Click to select individual tables

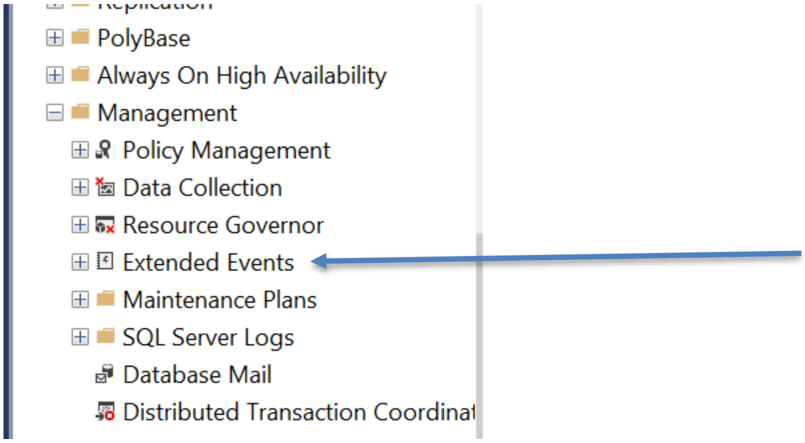
Execution Plans To be able to execute queries, the SQL Server Database Engine must analyze the statement to determine the most efficient way to access the required data. This analysis is handled by a component called the Query Optimizer.



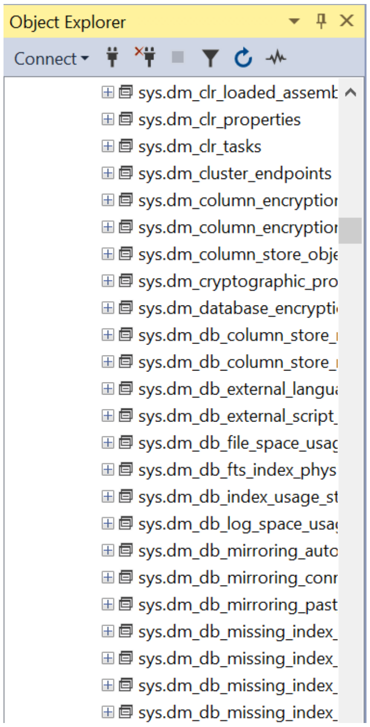
```
Query 1: Query cost (relative to the batch): 100%
SELECT P.FullName, P.FaxNumber, P.PhoneNumber, Cu.PhoneNumber AS CompanyPhone, Cu.FaxNumber AS CompanyFax FROM Application.People P JOIN Sales.Customers Cu ON P.PersonID = Cu.PrimaryContactPerson…
```

SELECT
Cost: 0 %
Nested Loops (Inner Join) Cost: 1 % 0.240s 1065 of 994 (107%)
Nested Loops (Left Semi Join) Cost: 1 % 0.030s 663 of 663 (100%)
Clustered Index Scan (C… [Customers].[PK_Sales_C… Cost: 8 % 0.020s 663 of 663 (100%)
Concatenation Cost: 0 % 0.008s 663 of 663 (100%)
Filter Cost: 0 % 0.007s 663 of 663 (100%)
Constant Scan Cost: 0 % 0.000s 663 of 663 (100%)
Filter Cost: 0 % 0.000s 0 of 663 (0%)
Nested Loops (Left Outer Join) Cost: 0 % 0.000s 0 of 663 (0%)
Constant Scan Cost: 0 % 0.000s 0 of 663 (0%)
Assert Cost: 0 % 0.000s 0 of 663 (0%)
Stream Aggregate (Aggregate) Cost: 0 % 0.000s 0 of 663 (0%)
Nested Loops (Inner Join) Cost: 0 % 0.000s 0 of 663 (0%)
Clustered Index Seek (C… [Cities].[PK_Applicatio… Cost: 100 % 0.000s 0 of 663 (0%)
Nested Loops (Inner Join) Cost: 0 % 0.000s
Filter Cost: 4 % 0.000s
Compute Scalar Cost: 1 %
Clustered Index Scan (C… [StateProvinces].[PK_Ap… Cost: 32 % 0.000s
Clustered Index Seek (C… [StateProvinces].[PK_Ap… Cost: 49 % 0.000s

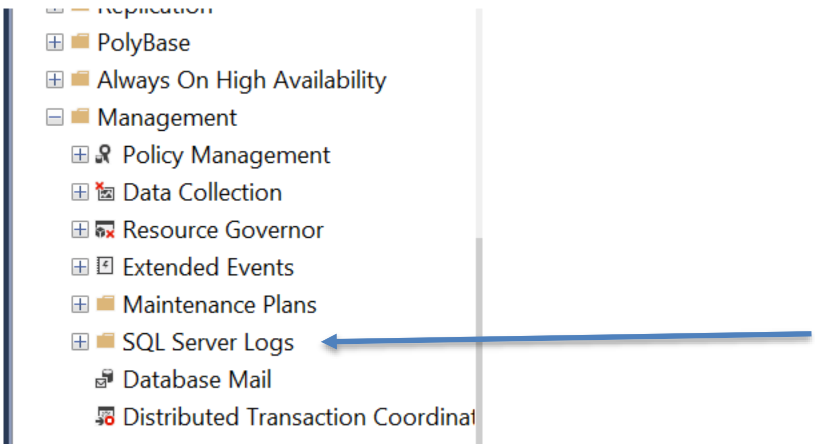
Extended Events is a lightweight performance monitoring system that enables users to collect data needed to monitor and troubleshoot problems in SQL Server.



Dynamic management views and functions return server state information that can be used to monitor the health of a server instance, diagnose problems, and tune performance.

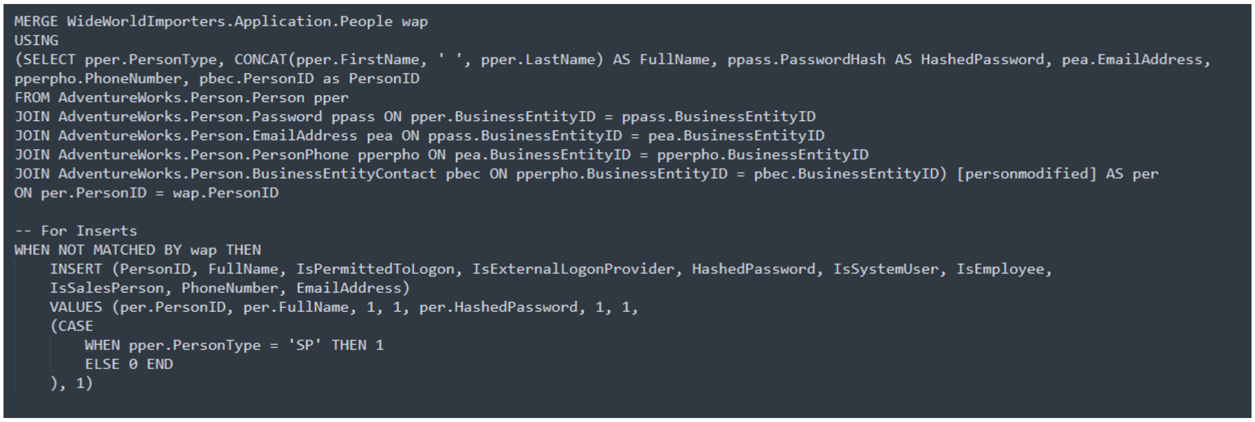


View the SQL Server error logs to ensure that processes have completed successfully (for example, backup and restore operations, batch commands, or other scripts and processes). This can be helpful to detect any current or potential problem areas.



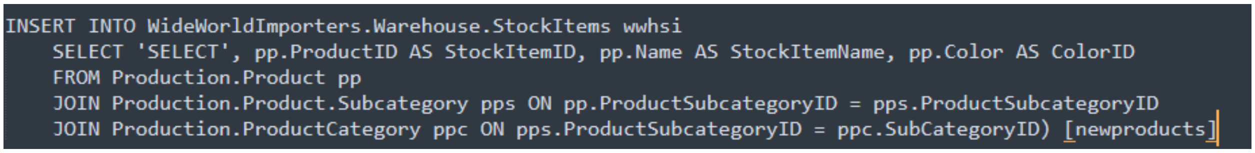
30. Write a short essay talking about a scenario: Good news everyone! We (Wide World Importers) just brought out a small company called “Adventure works”! Now that bike shop is our sub-company. The first thing of all works pending would be to merge the user logon information, person information (including emails, phone numbers) and products (of course, add category, colors) to WWI database. Include screenshot, mapping and query.

ANS) The merge of user logon and personal information can be done with a merge statement, wherein we first create a result table based on joining relevant information from all the Adventureworks tables. Then, following the format outlined in class, we provide relevant values for each instance of insertion of a new employee record.



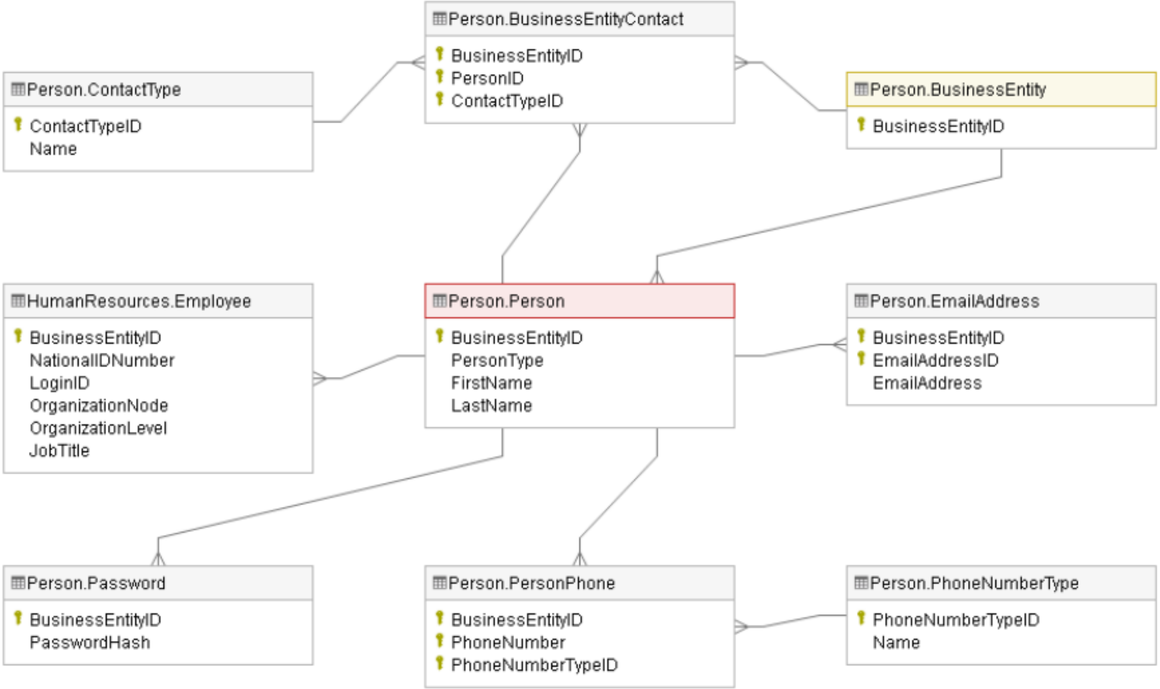
```
MERGE WideWorldImporters.Application.People wap
USING
(SELECT pper.PersonType, CONCAT(pper.FirstName, ' ', pper.LastName) AS FullName, ppass.PasswordHash AS HashedPassword, pea.EmailAddress,
pperpho.PhoneNumber, pbec.PersonID as PersonID
FROM AdventureWorks.Person.Person pper
JOIN AdventureWorks.Person.Password ppass ON pper.BusinessEntityID = ppass.BusinessEntityID
JOIN AdventureWorks.Person.EmailAddress pea ON ppass.BusinessEntityID = pea.BusinessEntityID
JOIN AdventureWorks.Person.PersonPhone pperpho ON pea.BusinessEntityID = pperpho.BusinessEntityID
JOIN AdventureWorks.Person.BusinessEntityContact pbec ON pperpho.BusinessEntityID = pbec.BusinessEntityID) [personmodified] AS per
ON per.PersonID = wap.PersonID

-- For Inserts
WHEN NOT MATCHED BY wap THEN
    INSERT (PersonID, FullName, IsPermittedToLogon, IsExternalLogonProvider, HashedPassword, IsSystemUser, IsEmployee,
    IsSalesPerson, PhoneNumber, EmailAddress)
    VALUES (per.PersonID, per.FullName, 1, 1, per.HashedPassword, 1, 1,
    (CASE
        WHEN pper.PersonType = 'SP' THEN 1
        ELSE 0 END
    ), 1)
```

For the merging of products we take relevant information from the AW tables and do an INSERT

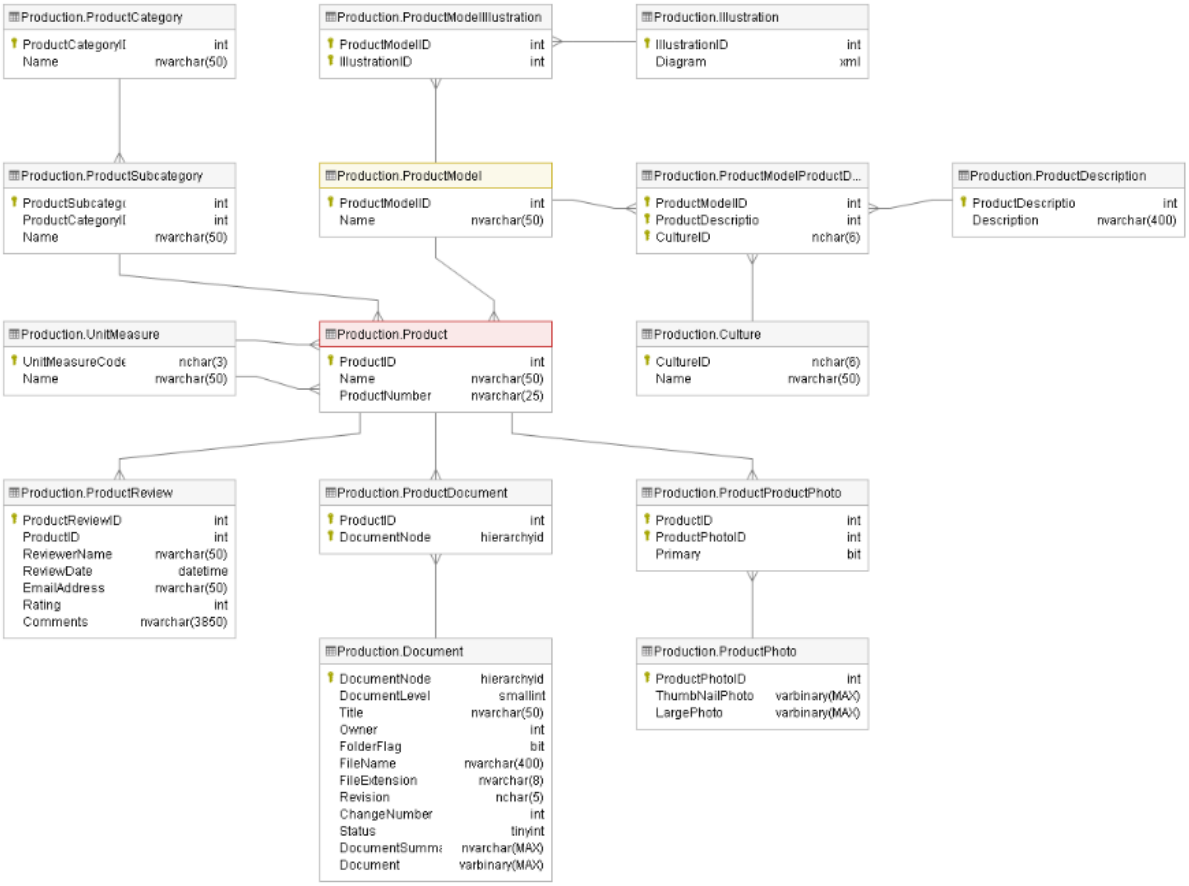


```
INSERT INTO WideWorldImporters.Warehouse.StockItems wwhsi
    SELECT 'SELECT', pp.ProductID AS StockItemID, pp.Name AS StockItemName, pp.Color AS ColorID
    FROM Production.Product pp
    JOIN Production.Product.Subcategory pps ON pp.ProductSubcategoryID = pps.ProductSubcategoryID
    JOIN Production.ProductCategory ppc ON pps.ProductSubcategoryID = ppc.SubCategoryID) [newproducts]
```

3. People

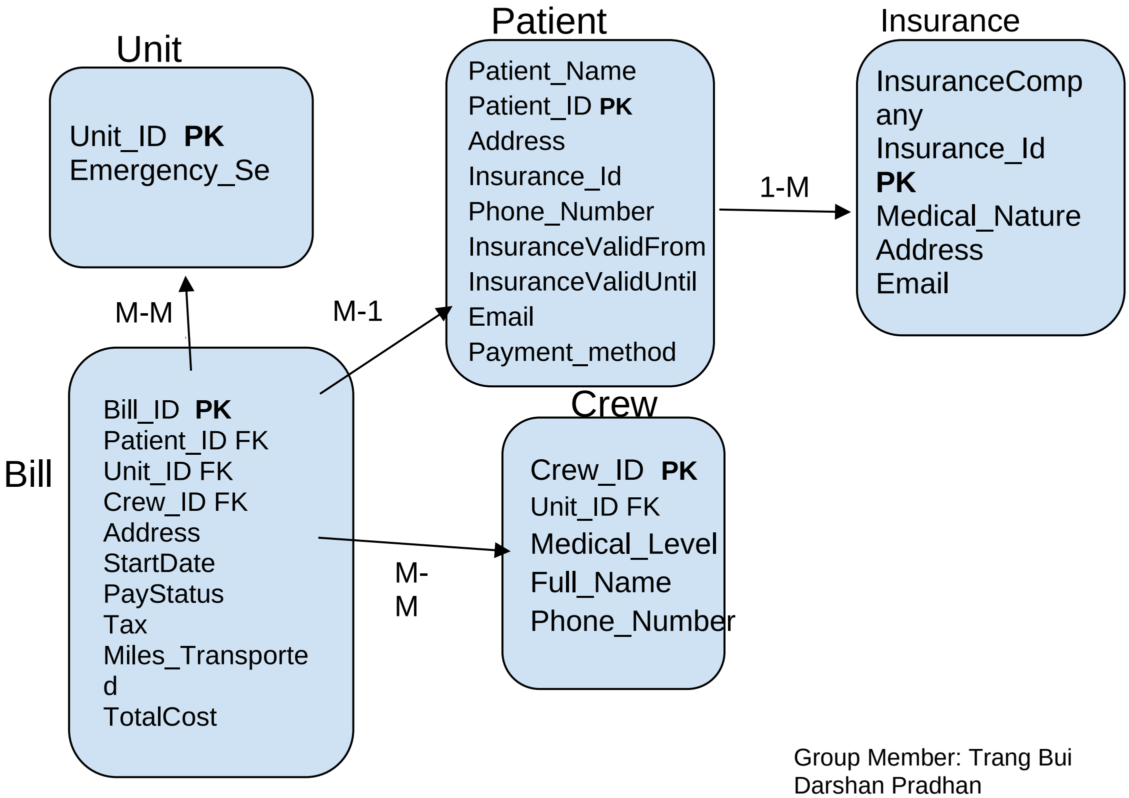

Person.BusinessEntityContact
BusinessEntityID
PersonID
ContactTypeID
Person.ContactType
ContactTypeID
Name
Person.BusinessEntity
BusinessEntityID
HumanResources.Employee
BusinessEntityID
NationalIDNumber
LoginID
OrganizationNode
OrganizationLevel
JobTitle
Person.Person
BusinessEntityID
PersonType
FirstName
LastName
Person.EmailAddress
BusinessEntityID
EmailAddressID
EmailAddress
Person.Password
BusinessEntityID
PasswordHash
Person.PersonPhone
BusinessEntityID
PhoneNumber
PhoneNumberTypeID
Person.PhoneNumberType
PhoneNumberTypeID
Name

5. Products



31. Database Design: OLTP db design request for EMS business: when people call 911 for medical emergency, 911 will dispatch UNITs to the given address. A UNIT means a crew on an apparatus (Fire Engine, Ambulance, Medic Ambulance, Helicopter, EMS supervisor). A crew member would have a medical level (EMR, EMT, A-EMT, Medic). All the treatments provided on scene are free. If the patient needs to be transported, that's where the bill comes in. A bill consists of Units dispatched (Fire Engine and EMS Supervisor are free), crew members provided care (EMRs and EMTs are free), Transported miles from the scene to the hospital (Helicopters have a much higher rate, as you can image) and tax (Tax rate is 6%). Bill should be sent to the patient insurance company first. If there is a deductible, we send the unpaid bill to the patient only. Don't forget about patient information, medical nature and bill paying status.

Table UNIT(***Unit_ID***, Emergency_Service)
Table Crew(*Crew_ID*, Medical_Level, Full_Name, Phone_Number)
Table Patient(***Patient_ID***, PatientName, Address, Insurance_ID, Phone_Number, InsuranceValidFrom, InsuranceValidUntil, Email, Payment_method)
Table Bill(***Bill_ID***, *Patient_ID*, *Unit_ID*, *Crew_ID*, Address, StartDate, PayStatus, Tax, Transported Miles, TotalCost)



Group Member: Trang Bui
Darshan Pradhan
Lawrence Han

32 Remember the discussion about those two databases from the class, also remember, those data models are not perfect. You can always add new columns (but not alter or drop columns) to any tables. Suggesting adding Ingested DateTime and Surrogate Key columns. Study the Wide World Importers DW. Think the integration schema is the ODS. Come up with a TSQL Stored Procedure driven solution to move the data from WWI database to ODS, and then from the ODS to the fact tables and dimension tables. By the way, WWI DW is a galaxy schema db. Requirements:

a. Luckly, we only start with 1 fact: Purchase. Other facts can be ignored for now.
b. Add a new dimension: Country of Manufacture. It should be given on top of Stock Items.
c. Write script(s) and stored procedure(s) for the entire ETL from WWI db to DW.