# Table of Contents

[Table of Contents 1](#_Toc405034401)

[Project Proposal 2](#_Toc405034402)

[Executive Summary 5](#_Toc405034403)

[System Definition 6](#_Toc405034404)

[System Requirements Specifications 7](#_Toc405034405)

[Entity-Relationship Diagram 8](#_Toc405034406)

[Physical Layout 9](#_Toc405034407)

[Data Dictionary 11](#_Toc405034408)

[Architecture 12](#_Toc405034409)

[Project Plan / Schedule 13](#_Toc405034410)

[Test Cases 14](#_Toc405034411)

[Sample Data 15](#_Toc405034412)

# Project Proposal

## Team Members

David Aquino

Tim Bibo

Rohan Dangi

## Proposed Client

In the year 1988, Dan Gordon and Dean Biersch co-founded Gordon Biersch (GB) Company with the goal of creating most authentic German-style larger. With the experience of 25 years, GB has doubled its annual production and increased its capacity to 4 million gallon of beer which made them the largest craft brewery in San Francisco Bay Area (Pulse, 2014). According to Brewers Association, GB ranks in top 49th Breweries in 2013 (Association, n.d.). Currently, there are 34 GB locations around the States that brews 40 different beers (Advocate, n.d). GB uses Ctuit software (also used by its parent company-Craftwork) that combines the restaurant POS data with inventory, accounting, and other tools to form consistent system throughout its brewery (Biersch, n.d.).

## Project Description

Our group would like to apply an existing concept to a new industry. Our idea resembles a stock exchange; however, instead of buying stock in a company, customers would be purchasing products to be consumed. In this particular case, customers would be purchasing alcoholic drinks such as beer or cocktails as they normally do in a bar or restaurant. The twist in this scenario is that the prices of these drinks (and possibly food in the future) will fluctuate according to demand and inventory levels. For example, if a group of people were to order 6 Miller Lites, the result after this purchase would be an increase (~ +5%) in the price of Miller Lites and a decrease in another drink/beer such as Bud Light (~ -5%).

We would be introducing a new POS system which would handle sales and relay them to our system. The system would be smart enough to make decisions such as price changes for the drinks. The business need we are addressing is alcohol sales. We predict that with our system, we can help track sales and introduce methods to increase sales. We intend to build a system which will be incorporated with inventory and track sales. In (almost) real-time we would like to adjust the prices of alcohol based on demand. The system will be smart enough to acknowledge trends as they occur and attempt to capitalize on these trends. It will be able to produce on the fly analysis regarding daily highs/lows compared to original price and current price (based on demand). There will also be periodic (daily/weekly/monthly/yearly) snapshots which can be used to analyze sales trends and affect purchasing decisions.

## Project Justification

Throughout the planning, design, and implementation of the project, the group members will demonstrate many of the skills that they acquired in their completed coursework, and they will acquire new skills that they will attain through independent methods.

The overall design and implementation of the project will follow the SDLC, a theory of design that has been reinforced in every course of the AIT-MS program at Towson University. Beginning with the project proposal and culminating in the operational maintenance of the project, the group members will adhere to the SDLC’s pillars of planning, designing, building, testing, and delivering a database system that meets the needs of a hypothetical customer. In addition to following the SDLC, the project will incorporate a number of skills learned in AIT-632 and AIT-732, the two prerequisite courses for AIT -735.

AIT-632 and AIT- 732 introduced the group members to a number of database design elements which the group members will incorporate into the project. For example, data modifications will be driven by stored procedures, while considering transaction controls and methods of reinforcing business rules. Also, data will be validated, and all code will be commented in such a way that the instructor (and hypothetically future developers) will be able to quickly and thoroughly understand it. In addition to the aforementioned topics, the group members will broaden their skill-set by learning and developing other database management techniques.

While the project is nascent, the group members have identified a number of skills that they wish to develop as part of this project. At the most basic level, group members will create a new database on an existing Microsoft SQL Server. (There has also been discussion of learning to install and configure SQL Server as a piece of this project). Also, the group members will learn about and implement sufficiently robust user access and permissions. Additionally, the group members will learn about different methods for backing up and restoring SQL Server databases. The group intends to implement a backup strategy for the project.

## Project Plan / Schedule

Our group has broken down the project into following tasks:

1. Project Planning: During this phase, our team will provide a system proposal to the client.
2. Requirement Gathering: This section will analyze and determine the needs and expectation of end users of newly developed system. We have decided to utilize the functions available in the existing system within the industry for our initial requirement gathering phase.
3. Design for System: This phase will define the architectural component of the system to satisfy the requirements gathered in the previous stage. Our team will create ER Diagram, Physical Table Layout, and Data Dictionary during this phase.
4. Implementation: This phase includes writing SQL queries, procedure, and triggers.
5. Data Conversion and Loading: Our team will upload a sample data in the system.
6. Testing: During this phase, our team will test if the queries, procedure, and triggers are working as intended or not.
7. Operational Maintenance: During this phase, our team will examine the available solutions for Backup/recovery option during the system failure.

The following table highlights the above mentioned phase with its estimated timeline.

|  |  |  |
| --- | --- | --- |
| TASKS | Estimated Timeline | TOTAL HOURS |
| Task 1. Project Planning | August 29, 2014 – September 01, 2014 |  |
| David Aquino | 3 |
| Tim Bibo | 3 |
| Rohan Dangi | 3 |
| Task 2. Requirement gathering | September 08, 2014– September 16, 2014 |  |
| David Aquino | 20 |
| Tim Bibo | 20 |
| Rohan Dangi | 20 |
| Task 3. Design for system | September 18, 2014– October 20, 2014 |  |
| David Aquino | 85 |
| Tim Bibo | 85 |
| Rohan Dangi | 85 |
| Task 4. Implementation | October 22, 2014– October 27, 2014 |  |
| David Aquino | 15 |
| Tim Bibo | 15 |
| Rohan Dangi | 15 |
| Task 5. Data Conversion & Loading | October 29, 2014– October 31, 2014 |  |
| David Aquino | 4 |
| Tim Bibo | 4 |
| Rohan Dangi | 4 |
| Task 6. Testing | November 03, 2014– November 14, 2014 |  |
| David Aquino | 10 |
| Tim Bibo | 10 |
| Rohan Dangi | 10 |
| Task 7. Operational Maintenance | November 17, 2014– November 19, 2014 |  |
| David Aquino | 3 |
| Tim Bibo | 3 |
| Rohan Dangi | 3 |
| **TOTAL HOURS TO COMPLETE PROJECT** | | **420 Hours** |
| **TOTAL HOURS TO COMPLETE PROJECT PER PARTICIPANT** | | **140 Hours** |

## Work Cited

* Advocate. (n.d.). Gordon Biersch Brewery Restaurant | United States | Beers.BeerAdvocate. Retrieved September 1, 2014, from <http://www.beeradvocate.com/beer/profile/1551/>.
* Association. (n.d.). Brewers Association Lists Top 50 Breweries of 2013.brewersassociation.org. Retrieved August 31, 2014, from <http://www.brewersassociation.org/attachments/0001/4525/CBP13\_Top\_50.pdf/>
* Biersch. (n.d.). Careers. Growth. Retrieved August 31, 2014, from http://www.gordonbiersch.com/careers/growth
* Pulse. (2014, May 20). Gordon Biersch Dunkles release marks brewery's 25th anniversary. BeerPulse. Retrieved August 31, 2014, from <http://beerpulse.com/2014/05/gordon-biersch-dunkles-release-marks-brewerys-25th-anniversary-3108/>.

# 

# Executive Summary

Text

# 

# System Definition

Text

# 

# System Requirements Specifications

Text

# Entity-Relationship Diagram



# Physical Layout

| **TABLE** | **VARIABLE** | **DATA\_TYPE** | **NOT\_NULL** | **PK** | **FK** | **IDENTITY** | **CHECK** | **UNIQUE** | **DESCRIPTION** | **SAMPLE** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **T\_VENDOR** | VEN\_ID | INT | Y | Y |  | Y |  | Y | UNIQUE ID FOR EACH VENDOR | 1 |
| VEN\_NAME | VARCHAR(30) | Y |  |  |  |  | Y | VENDOR NAME | WM BREWERY |
| VEN\_STREET1 | VARCHAR(30) | Y |  |  |  |  |  | VENDOR ADDRESS ONE | 123 FAKE STREET |
| VEN\_STREET2 | VARCHAR(30) |  |  |  |  |  |  | VENDOR ADDRESS TWO | UNIT X |
| VEN\_CITY | VARCHAR(30) | Y |  |  |  |  |  | CITY | WHITE MARSH |
| VEN\_STATE | CHAR(2) | Y |  |  |  |  |  | STATE | MD |
| VEN\_ZIP | NUMERIC(5,0) | Y |  |  |  | >=01000<=99999 |  | U.S. ZIP CODE | 21245 |
| VEN\_PHONE | VARCHAR(12) | Y |  |  |  |  |  | PHONE NUMBER OF VENDOR | 410-111-1111 |
| VEN\_EMAIL | VARCHAR(50) | Y |  |  |  |  |  | EMAIL OF CONTACT AT VENDOR | [JJ@WMBREW.COM](mailto:JJ@WMBREW.COM) |
| VEN\_CONTACT | VARCHAR(50) | Y |  |  |  |  |  | NAME OF CONTACT AT VENDOR | JANET JONES |
|  |  |  |  |  |  |  |  |  |  |  |
| **T\_PRODUCT** | PRO\_ID | INT | Y | Y |  | Y |  | Y | UNIQUE ID FOR EACH PRODUCT | 1 |
| PRO\_NAME | VARCHAR(30) | Y |  |  |  |  | Y | DESCRIPTIVE NAME OF PRODUCT | SPICY SUMMER ALE |
| TY\_ID | INT | Y |  | Y |  |  |  | REFERENCE TYPE (TY\_ID) | 5 |
| PRO\_BASE | NUMERIC(5,2) | Y |  |  |  | >0 |  | BASE PRICE OF THE PRODUCT | $5.00 |
| PRO\_INSTOCK | INT | Y |  |  |  | >=0 |  | NUMBER OF PRODUCT IN STOCK |  |
| VEN\_ID | INT | Y |  | Y |  |  |  | REFERENCE VENDOR (VEN\_ID) | 546 |
|  |  |  |  |  |  |  |  |  |  |  |
| **T\_TYPE** | TY\_ID | INT | Y | Y |  | Y |  | Y | UNIQUE ID FOR EACH PRODUCT CATEGORY | 1 |
| TY\_DESCRIPTION | VARCHAR(50) | Y |  |  |  |  | Y | TYPE OF PRODUCT | 16 OZ BEER BOTTLE |
| TY\_AGERESTRICTED | INT | Y |  |  |  | IN(1,2) |  | MUST CUSTOMER BE 21 TO PURCHASE? | 1=YES / 2=NO |
|  |  |  |  |  |  |  |  |  |  |  |
| **T\_PURCHASE** | PUR\_ID | INT | Y | Y |  | Y |  | Y | UNIQUE ID FOR EACH PURCHASE | 1 |
| PRO\_ID | INT | Y |  | Y |  |  |  | REFERENCE PRODUCT (PRO\_ID) | 33 |
| VEN\_ID | INT | Y |  | Y |  |  |  | REFERENCE VENDOR (VEN\_ID) | 12 |
| PUR\_QTY | INT | Y |  |  |  | >0 |  | QUANTITY PURCHASED | 480 |
| PUR\_UNT\_PRICE | NUMERIC(5,2) | Y |  |  |  | >0 |  | PRICE PAID PER UNIT | $1.25 |
| PUR\_DATE | DATETIME | Y |  |  |  |  |  | DATE AND TIME OF PURCHASE | 8/15/2015:15:32:42 |
|  |  |  |  |  |  |  |  |  |  |  |
| **T\_ACCT\_SALES** | ACCT\_ID | INT | Y | Y |  | Y |  | Y | UNIQUE ID FOR EACH ITEM IN THE CART | 1 |
| ACCT\_DATETIME | DATETIME | Y |  |  |  |  |  | DATE AND TIME OF PURCHASE | 8/15/2015:15:32:42 |
| ACCT\_PRICE | NUMERIC(5,2) | Y |  |  |  | >0 |  | PRICE AT WHICH PRODUCT WAS SOLD | $1.87 |
| CUS\_ID | INT | Y |  | Y |  | >0 |  | REFERENCE CUSTOMER (CUS\_ID) | 133 |
| ACCT\_QTY | INT | Y |  |  |  | >0 |  | QTY SOLD |  |
| PRO\_ID | INT | Y |  | Y |  |  |  | REFERENCE PRODUCT (CUS\_ID) | 2 |
|  |  |  |  |  |  |  |  |  |  |  |
| **T\_PRICE** | PRI\_ID | INT | Y | Y |  | Y |  | Y | UNIQUE ID FOR EACH PRICE CHANGE | 4876 |
| PRO\_ID | INT | Y |  |  |  |  |  | REFERENCE PRODUCT (PRO\_ID) | 33 |
| PRO\_PRI | NUMERIC(5,2) | Y |  |  |  | >0 |  | CURRENT SELLING PRICE OF PRODUCT | $1.87 |
|  |  |  |  |  |  |  |  |  |  |  |
| **T\_CUSTOMER** | CUS\_ID | INT | Y | Y |  | Y |  |  | UNIQUE ID FOR EACH CUSTOMER | 21 |
| CUS\_DOB | DATE | Y |  |  |  |  |  | CUSTOMER DOB | 9/12/1990 |
| CUS\_STREET1 | VARCHAR(30) |  |  |  |  |  |  | CUSTOMER ADDRESS ONE | 123 FAKE STREET |
| CUS\_STREET2 | VARCHAR(30) |  |  |  |  |  |  | CUSTOMER ADDRESS TWO | UNIT X |
| CUS\_CITY | VARCHAR(30) |  |  |  |  |  |  | CITY | WHITE MARSH |
| CUS\_STATE | CHAR(2) |  |  |  |  |  |  | STATE | MD |
| CUS\_ZIP | NUMERIC(5,0) |  |  |  |  | >=01000<=99999 |  | U.S. ZIP CODE | 21245 |
| CUS\_EMAIL | VARCHAR(50) |  |  |  |  |  |  | CUSTOMER EMAIL ADDRESS | [JAKE@OLDMAN.COM](mailto:JAKE@OLDMAN.COM) |
| CUS\_FNAME | VARCHAR(25) |  |  |  |  |  |  | CUSTOMER FIRST NAME | JAKE |
| CUS\_LNAME | VARCHAR(30) |  |  |  |  |  |  | CUSTOMER LAST NAME | OLDMAN |
| CUS\_MI | VARCHAR(1) |  |  |  |  |  |  | CUSTOMER MIDDLE INITIAL | R |
| CUS\_SUFFIX | VARCHAR(5) |  |  |  |  |  |  | CUSTOMER NAME SUFFIX | III |
|  |  |  |  |  |  |  |  |  |  |  |
| **T\_SALES\_INFO** | SI\_ID | INT | Y | Y |  | Y |  | Y | UNIQUE ID USED FOR SETTING PRICES | 3 |
| PRO\_ID | INT | Y |  | Y |  |  |  | REFERENCES PRO\_ID IN PRODUCT TABLE | 9 |
| QTY\_SOLD | INT | Y |  |  |  |  |  | QTY SOLD | 12 |
|  |  |  |  |  |  |  |  |  |  |  |
| **T\_SALES\_PERC** | SP\_ID | INT | Y | Y |  | Y |  | Y | UNIQUE ID FOR EACH SALES PERCENTAGE | 34 |
| PRO\_ID | INT | Y |  | Y |  |  |  | REFERENCES PRO\_ID IN PRODUCT TABLE | 9 |
| PERC\_SALES | INT | Y |  | Y |  |  |  | PERCENT \* 100 | 44 |
|  |  |  |  |  |  |  |  |  |  |  |
| **T\_POS\_SALES** | POS\_ID | INT | Y | Y |  | Y |  | Y | UNIQUE ID FOR EACH ITEM IN THE CART | 1 |
| POS\_DATETIME | DATETIME | Y |  |  |  |  |  | DATE AND TIME OF PURCHASE | 8/15/2015:15:32:42 |
| PRO\_PRICE | NUMERIC(5,2) | Y |  |  |  | >0 |  | PRICE AT WHICH PRODUCT WAS SOLD | $1.87 |
| CUS\_ID | INT | Y |  | Y |  | >0 |  | REFERENCE CUSTOMER (CUS\_ID) | 133 |
| POS\_QTY | INT | Y |  |  |  | >0 |  | QTY SOLD |  |
| PRO\_ID | INT | Y |  | Y |  |  |  | REFERENCE PRODUCT (CUS\_ID) | 2 |

# 

# Data Dictionary

Note: The following format is utilized to illustrate the entities and descriptive information.

|  |
| --- |
| **Entity Name** |
|  |
| Field Designator Type Size |
|  |
| Required/Optional – Definition |

## T\_VENDOR

|  |
| --- |
| Vendor ID (Primary Key) |
|  |
| ven\_id INT (Auto-Generated) |
|  |
| Required – The ven\_id is a unique number that all the vendors should have which is assigned during the first time transaction between the vendor and the organization. |

|  |
| --- |
| Vendor Name |
|  |
| ven\_name VARCHAR 30 |
|  |
| Required – Name of the vendor. For example, Towson Brewery |

|  |
| --- |
| Street1 Name |
|  |
| ven\_street1 VARCHAR 30 |
|  |
| Required – The street name of the Vendor’s address. |

|  |
| --- |
| Street2 Name |
|  |
| ven\_street2 VARCHAR 30 |
|  |
| Optional – If the Vendor’s address requires a second line. |

|  |
| --- |
| City |
|  |
| ven\_city VARCHAR 30 |
|  |
| Required – The city of the Vendor’s address. |

|  |
| --- |
| State |
|  |
| ven\_state CHAR 2 |
|  |
| Required – The two character state code of the vendor’s address |

|  |
| --- |
| Zip Code |
|  |
| ven\_zip NUMERIC 5,0 |
|  |
| Required – The zip code of the vendor. |

|  |
| --- |
| Vendor’s Phone |
|  |
| ven\_phone VARCHAR 12 |
|  |
| Required – Phone number of the Vendor including dash delimiters. For example: “410-458-8774”. |

|  |
| --- |
| Vendor’s Email Address |
|  |
| ven\_email VARCHAR 50 |
|  |
| Required – Email Address of the Vendor |

|  |
| --- |
| Vendor’s Contact Person |
|  |
| ven\_contact VARCHAR 50 |
|  |
| Required – First name & last name of contact person at Vendor |

## T\_PRODUCT

|  |
| --- |
| Product ID (Primary Key) |
|  |
| pro\_id INT(Auto-Generated) |
|  |
| Required- The pro\_id is a unique number for products. |

|  |
| --- |
| Name of Product |
|  |
| pro\_name VARCHAR 30 |
|  |
| Required – Name given to the product |

|  |
| --- |
| Type ID of the Product (Foreign Key-References T\_TYPE Table) |
|  |
| ty\_id INT |
|  |
| Required- This is a type id from the T\_Type table. |

|  |
| --- |
| Price of the Product |
|  |
| pro\_base NUMERIC 5,2 |
|  |
| Required – This is the base price of the product which has to be >0 |

|  |
| --- |
| Vendor ID (Foreign Key- References T\_VENDOR Table) |
|  |
| ven\_id INT 10 |
|  |
| Required – The ven\_id is a unique number that all the vendors should have which is assigned during the first time transaction between the vendor and the organization. See T\_Vendor Table. |

|  |
| --- |
| Products in Stock |
|  |
| pro\_instock INT |
|  |
| Required – Total number of products in stock must be >=0 |

## T\_TYPE

|  |
| --- |
| Type ID (Primary Key) |
|  |
| type\_id INT |
|  |
| Required – This is a unique ID given to a product based on its type |

|  |
| --- |
| Type Description |
|  |
| ty\_description VARCHAR 50 |
|  |
| Required – Type description of the product such as 12/16 Oz Beer can, 8.4/20 Oz Energy Drink, etc. |
| Age Restriction |
|  |
| ty\_restricted Int 1 |
|  |
| Required – Ty\_Restricted can be either “0” or “1” where “1” denotes as Alcohol drink requiring to be 21 to purchase whereas, “0” denotes as non-alcoholic drink. |

## T\_PURCHASE

|  |
| --- |
| Purchase ID (Primary key) |
|  |
| pur\_id INT (Auto-Generated) |
|  |
| Required – This is a unique ID given for each purchase from a vendor. |

|  |
| --- |
| Product ID (Foreign Key- References T\_PRODUCT Table) |
|  |
| pro\_id INT |
|  |
| Required- The pro\_id is a unique number for products. See T\_PRODUCT Table. |

|  |
| --- |
| Vendor ID (Foreign Key- References T\_VENDOR Table) |
|  |
| ven\_id INT |
|  |
| Required – The ven\_id is a unique number that all the vendors should have which is assigned during the first time transaction between the vendor and the organization. See T\_Vendor Table. |

|  |
| --- |
| Purchased Quantity |
|  |
| pur\_qty INT |
|  |
| Required – Total number of quantities of product purchased which has to be >0. |

|  |
| --- |
| Unit Price |
|  |
| Pur\_Unt\_Price Numeric 5,2 |
|  |
| Required – This is price paid per unit of product. |

|  |
| --- |
| Date of Purchase |
|  |
| Pur\_Date Datetime 8 |
|  |
| Required – This is date and time of the day when purchase occurred |

## T\_ACCT\_SALES

|  |
| --- |
| Accounting ID (Primary key) |
|  |
| acct\_ID INT(Auto-Generated) |
|  |
| Required – This is a unique ID for sales. The Accounting Sales table is permanent and is used to track historical data. |

|  |
| --- |
| Date of Sales |
|  |
| acct\_datetime DATETIME |
|  |
| Required – This is date and time of the day when sales occurred. |

|  |
| --- |
| Price of Sold Price |
|  |
| acct\_price NUMERIC 5,2 |
|  |
| Required – This is the price at which the product was sold. |

|  |
| --- |
| Customer ID (Foreign key- Reference T\_CUSTOMER Table) |
|  |
| cus\_ID INT |
|  |
| Required – This is a unique ID given for each customer. See T\_CUSTOMER table |

|  |
| --- |
| Sold Quantity |
|  |
| acct\_qty INT |
|  |
| Required – Total number of quantities of product sold which has to be >0. |

|  |
| --- |
| Product ID (Foreign Key- References T\_PRODUCT Table) |
|  |
| pro\_id INT |
|  |
| Required- The pro\_id is a unique number for products. See T\_PRODUCT Table |

## T\_PRICE

|  |
| --- |
| Price ID (Primary key) |
|  |
| pri\_id INT (Auto-Generated) |
|  |
| Required – This is a unique ID for the price table. |

|  |
| --- |
| Product ID (Foreign Key- References T\_PRODUCT Table) |
|  |
| pro\_id INT |
|  |
| Required- The pro\_id is a unique number for products. See T\_PRODUCT Table |

|  |
| --- |
| Current Price |
|  |
| pro\_price NUMERIC 5,2 |
|  |
| Required – This is the current price of the product. |

## T\_CUSTOMER

|  |
| --- |
| Customer ID (Primary Key) |
|  |
| cus\_id INT |
|  |
| Required – This is a unique ID given for each customer. |

|  |
| --- |
| Customer’s Date of Birth |
|  |
| cus\_dob DATETIME |
|  |
| Required- Date of Birth is needed to calculate customer age and determine eligibility to purchase products. |

|  |
| --- |
| Customer’s Address Street 1 |
|  |
| cus\_street1 VARCHAR 30 |
|  |
| Optional – Customer’s street address line 1. |

|  |
| --- |
| Customer’s Address Street2 |
|  |
| cus\_street2 VARCHAR 30 |
|  |
| Optional – Customer’s street address line 2. |

|  |
| --- |
| Customer’s Address City |
|  |
| cus\_city VARCHAR 30 |
|  |
| Optional – The city of the Customers’s address. |

|  |
| --- |
| Customer’s State |
|  |
| cus\_state CHAR 2 |
|  |
| Optional – The two character state code of the Customer’s address. |

|  |
| --- |
| Zip Code |
|  |
| cus\_zip NUMERIC 5,0 |
|  |
| Optional – The zip code of the Customer’s Address. |

|  |
| --- |
| Customer’s Email Address |
|  |
| cus\_email VARCHAR 50 |
|  |
| Optional – Email Address of the Customer |

|  |
| --- |
| Customer’s First Name |
|  |
| cus\_fname VARCHAR 25 |
|  |
| Optional – First name of the customer |

|  |
| --- |
| Customer’s Last Name |
|  |
| cus\_lname VARCHAR 30 |
|  |
| Optional – Last name of the customer |

|  |
| --- |
| Middle Initial |
|  |
| cus\_mi VARCHAR 1 |
|  |
| Optional – Middle initial of the customer |

|  |
| --- |
| Customer Name Suffix |
|  |
| cus\_suffix VARCHAR 5 |
|  |
| Optional – Customer Name Suffix such Jr., Sr. Ph.D., II, III, IV, V, etc. |

## T\_SALES\_INFO

|  |
| --- |
| Sales ID (Primary Key) |
|  |
| si\_id INT (Auto-generated) |
|  |
| Required – This is a unique ID for this helper table. |

|  |
| --- |
| Product ID (Foreign Key- References T\_PRODUCT Table) |
|  |
| pro\_id INT |
|  |
| Required- The pro\_id is a unique number for products. See T\_PRODUCT Table. |

.

|  |
| --- |
| Quantity |
|  |
| qty\_sold INT |
|  |
| Required – Total number of product sold. |

## T\_SALES\_PERC

|  |
| --- |
| Sales Percentage ID (Primary Key) |
|  |
| sp\_id INT (Auto-Generated) |
|  |
| Required – This is a unique ID for this helper table. |

|  |
| --- |
| Product ID (Foreign Key- Reference T\_PRODUCT Table) |
| pro\_id INT |
|  |
| Required- The pro\_id is a unique number for products. See T\_PRODUCT Table. |

|  |
| --- |
| Percentage Sold |
| pct\_of\_sales INT |
|  |
| Required- Percentage of sales for products sold. |

## T\_POS\_SALES

|  |
| --- |
| POS ID (Primary Key) |
|  |
| pos\_id INT (Auto-generated) |
|  |
| Required – This is a unique ID given for sales at the pos terminal. |

|  |
| --- |
| Date/Time of Sale |
|  |
| pos\_datetime DATETIME |
|  |
| Required – This is date and time of the sale at the pos terminal. |

|  |
| --- |
| Price |
|  |
| pro\_price NUMERIC 5,2 |
|  |
| Required – This is the price at which the product was sold. |

|  |
| --- |
| Customer ID (Foreign key- References T\_CUSTOMER Table) |
|  |
| cus\_id INT |
|  |
| Required – This is a unique ID given for each customer. See T\_CUSTOMER Table |

|  |
| --- |
| POS Paid |
|  |
| pos\_paid INT |
|  |
| Required – 0 for unpaid, 1 for paid |

|  |
| --- |
| Quantity |
|  |
| pos\_qty INT |
|  |
| Required – Number products sold which has to be >0. |

|  |
| --- |
| Product ID (Foreign Key- References T\_PRODUCT Table) |
|  |
| pro\_id INT |
|  |
| Required- The pro\_id is a unique number for products. See T\_PRODUCT Table |

## T\_PRICE\_DIFF

|  |
| --- |
| Price Differential ID |
|  |
| diff\_id INT |
|  |
| Required- The diff\_id is a unique number for this helper table |

|  |
| --- |
| Product ID (Foreign Key- References T\_PRODUCT Table) |
|  |
| pro\_id INT |
|  |
| Required- The pro\_id is a unique number for products. See T\_PRODUCT Table |

|  |
| --- |
| Percentage Difference |
|  |
| diff\_perc DECIMAL 5,1 |
|  |
| Required- The percentage difference between current price of a product and it’s base price. |

## T\_CUS\_AGE

|  |
| --- |
| Customer Age ID |
|  |
| cage\_id INT |
|  |
| Required- The cage\_id is a unique number this helper table. |

|  |
| --- |
| Customer ID (Foreign Key- References T\_Customer Table) |
|  |
| pro\_id INT |
|  |
| Required- The pro\_id is a unique number for products. See T\_PRODUCT Table |

|  |
| --- |
| Age |
|  |
| age INT |
|  |
| Required- Age is determined by a date diff from the customer date of birth. |

# 

# Architecture

Text

# 

# Project Plan / Schedule

Text

# 

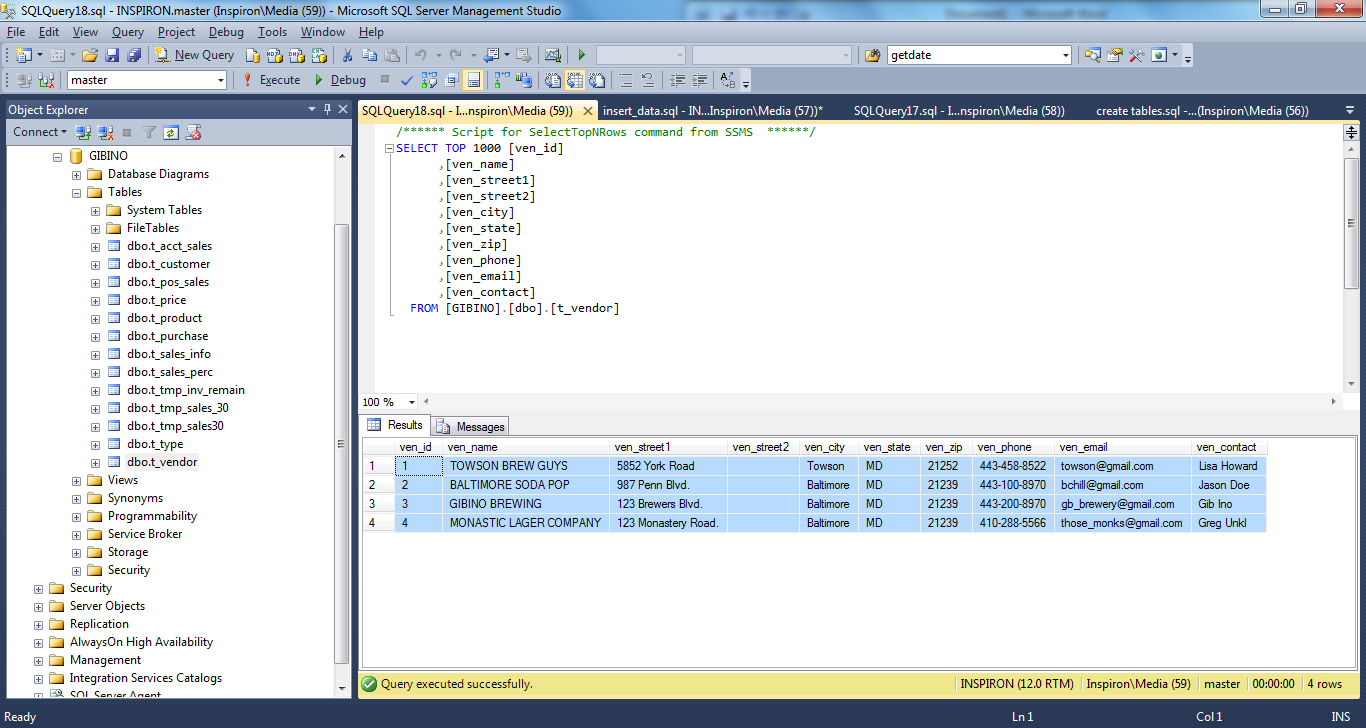
# Test Cases

Text

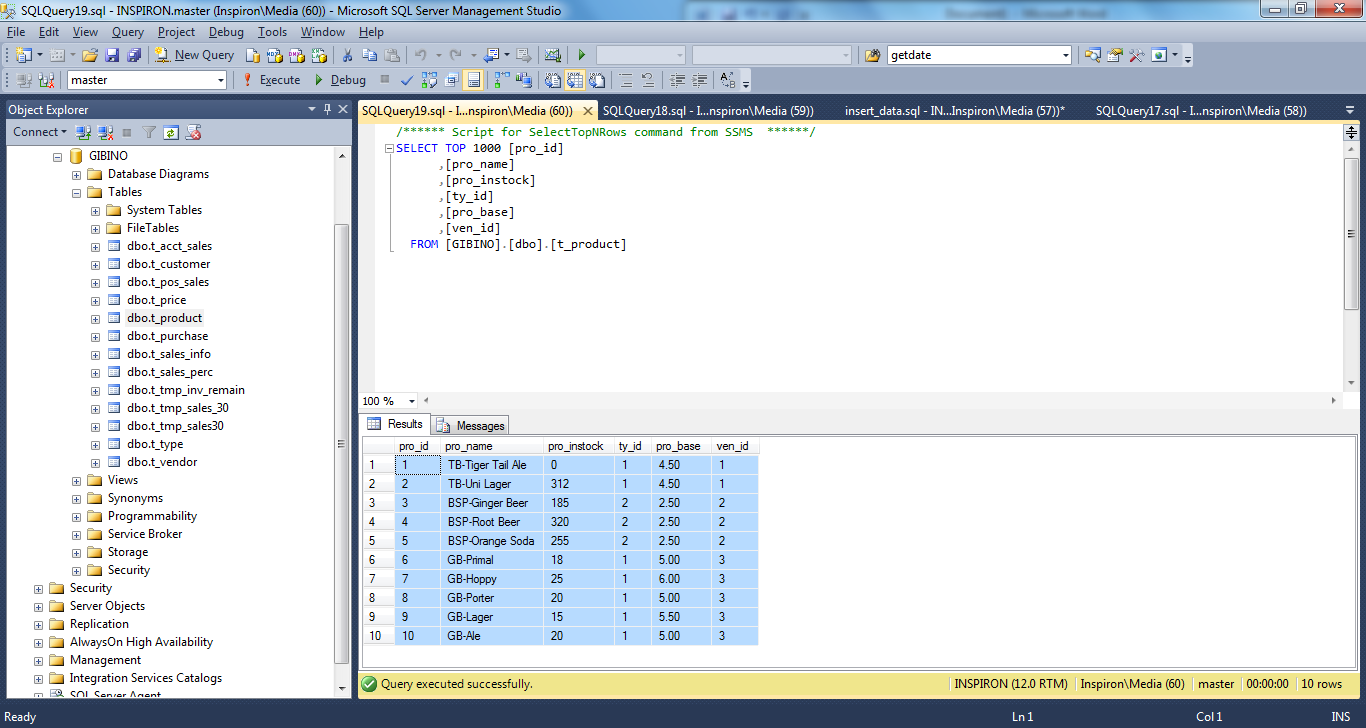
# 

# Sample Data

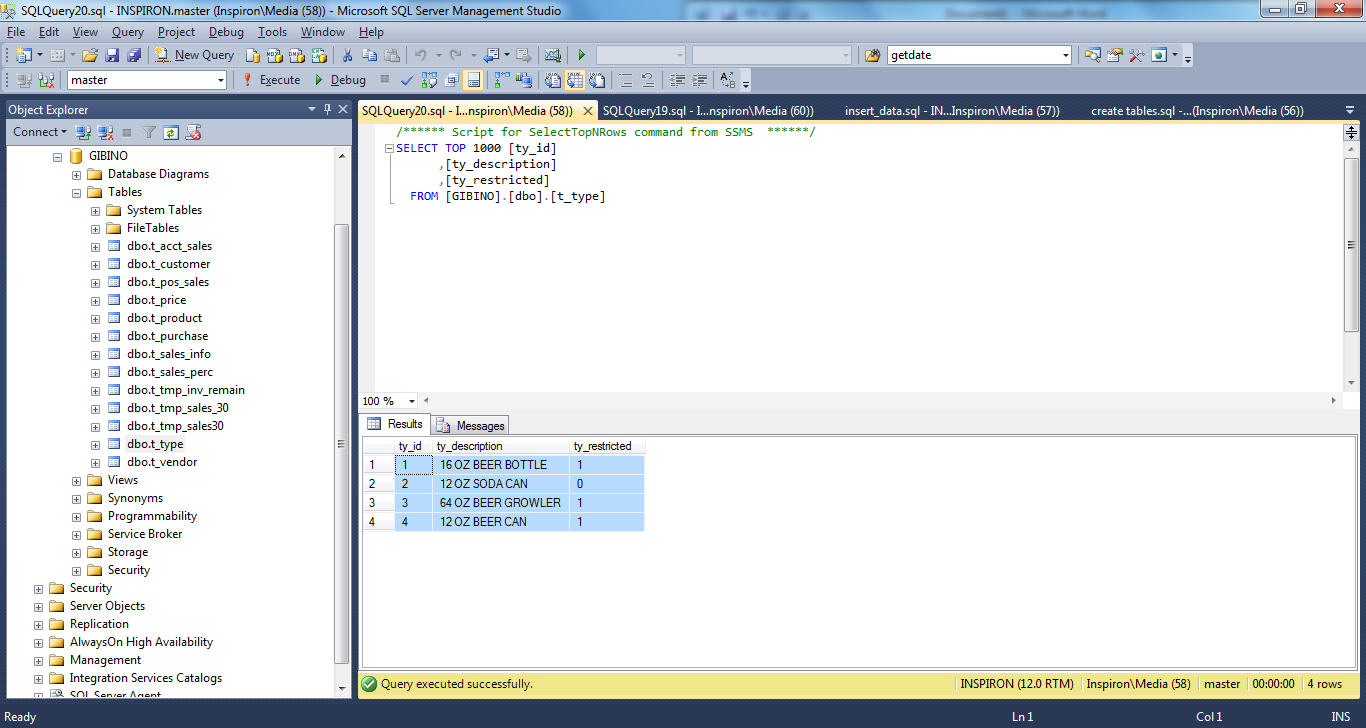
T\_VENDOR



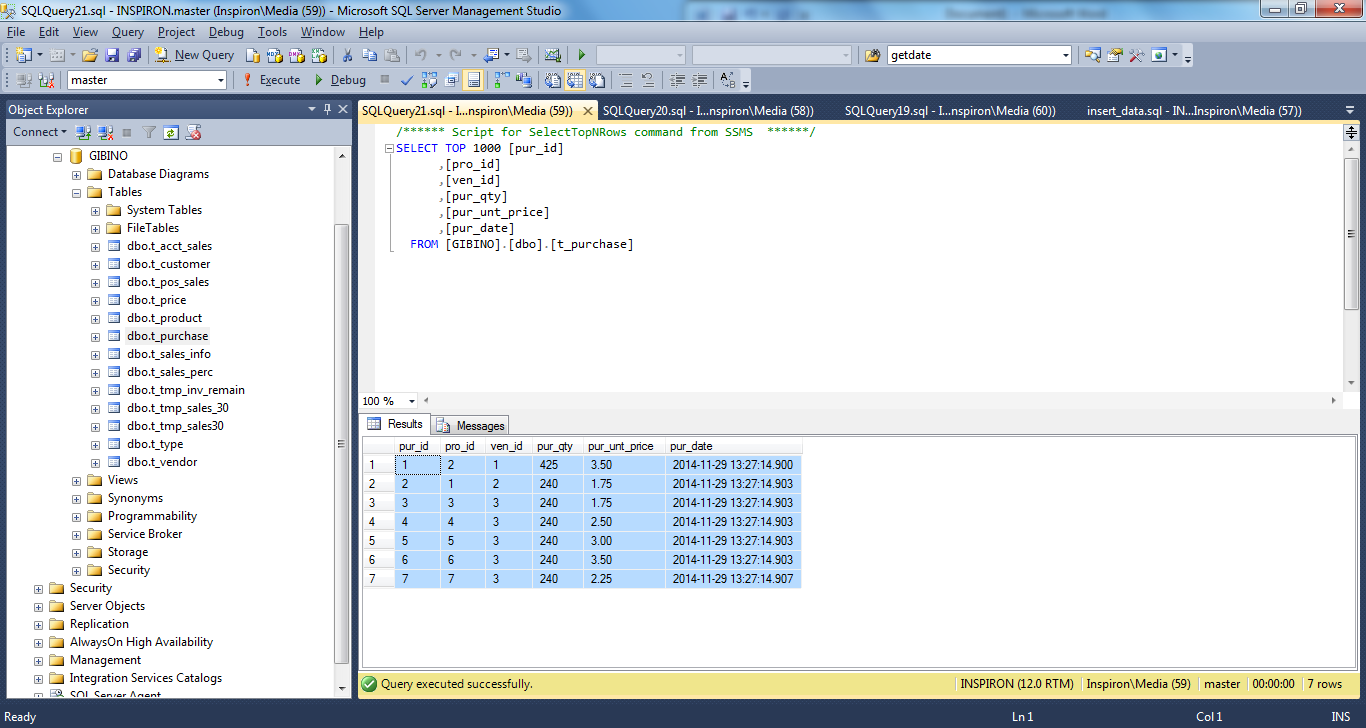
T\_PRODUCT



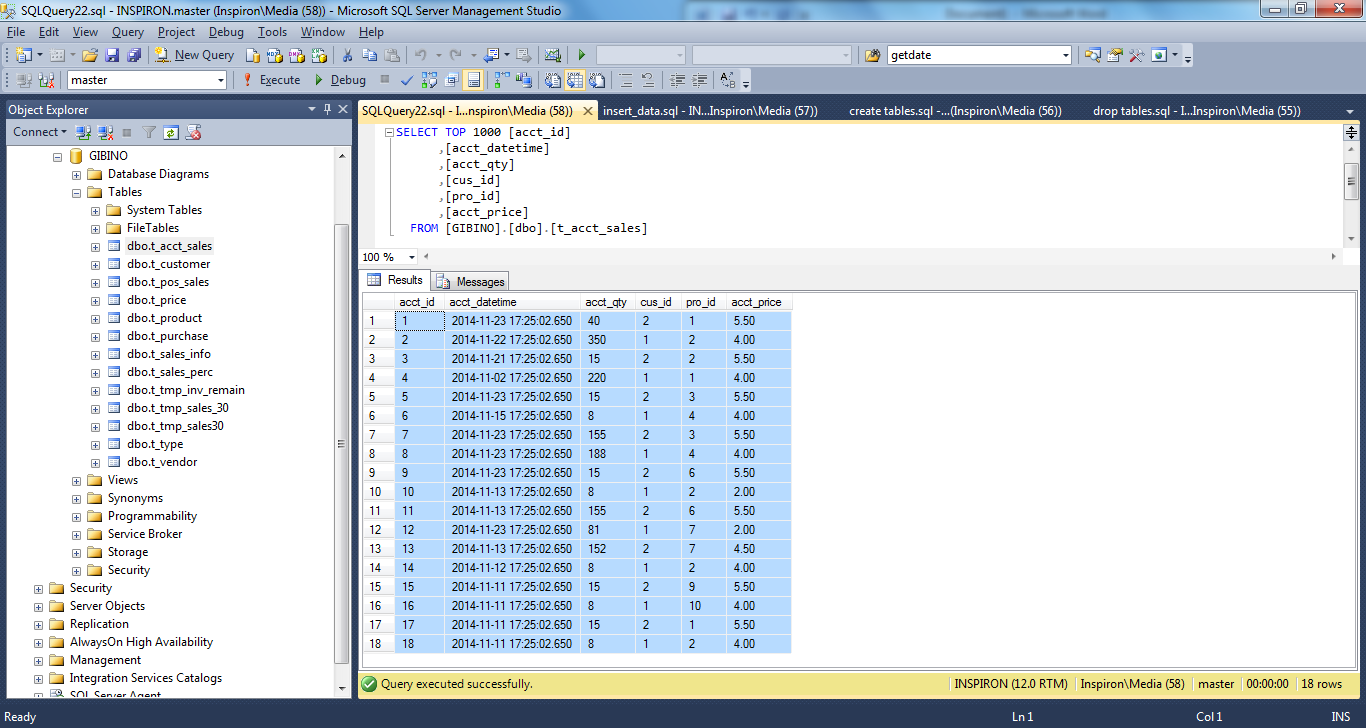
T\_TYPE



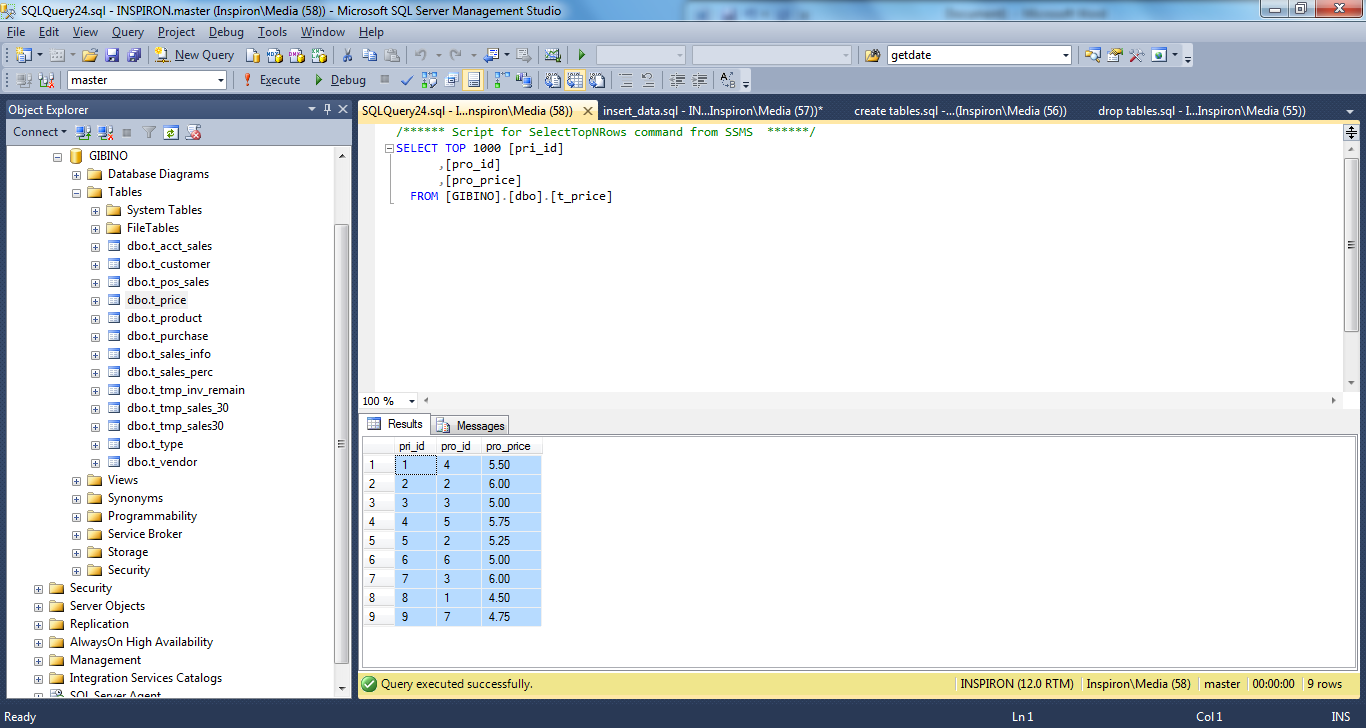
T\_PURCHASE



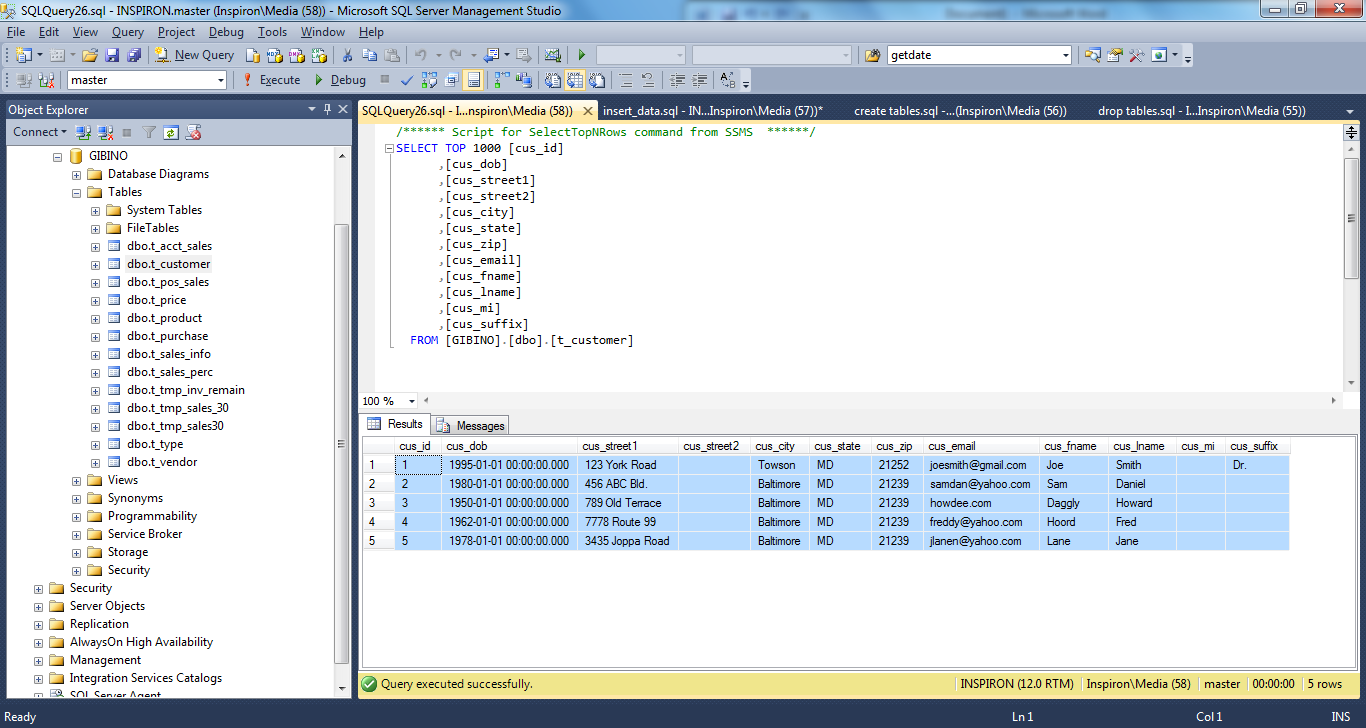
T\_ACCT\_SALES



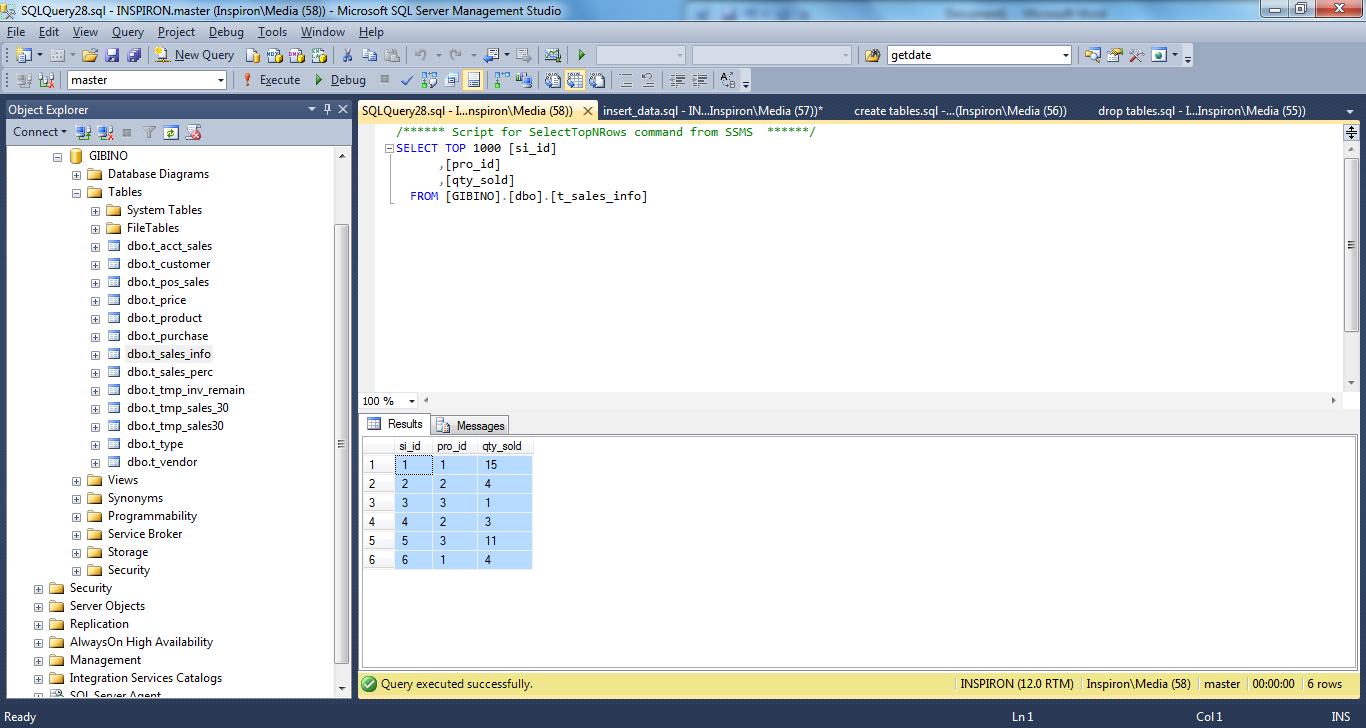
T\_PRICE



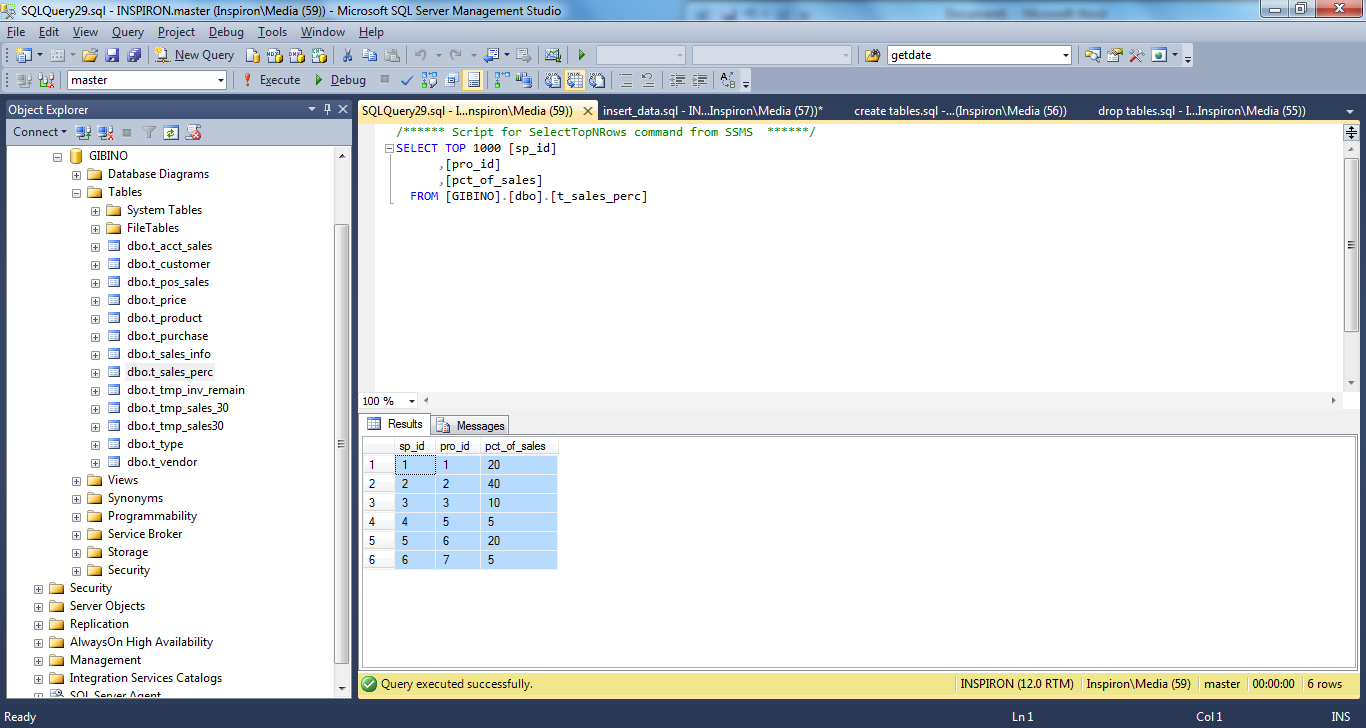
T\_CUSTOMER



T\_SALES\_INFO



T\_SALES\_PERC



T\_POS\_SALES

