Bob's Fixit Group

CIS 535 – Management and Design of Databases

Submitted to: Prof. George Lamperti

Bellevue University

Team Members

Madhuri Satturi

Vamshi Bandaru

Bob's Home Repairs

Bob owns a small company called Bob’s Home repairs. He does the small home repair jobs that the large companies pass by. Need some wood work or wood furniture fixed? Call Bob.

Here is how the business works:

Someone calls Bob and asks him to bid on a job. He drives over, looks at the situation, and gives them a bid. Sometimes it is an official looking bid by mail, and sometimes it is scribbled on notebook paper. He decides how long it will take to do the job (he bills by the hour), how much wood will be needed, any odds and ends that are unique to the job, and an overall price. He moves from job to job and bills customers as he finishes the work.

Bob buys items and supplies from a variety of places, but he buys stuff only when it is needed for a particular project. A potential problem: if he gets behind on his payments to various suppliers, then they won't let him order any more. This would stop his business dead in its tracks. His biggest and most crucial supply is lumber (the price rises and falls constantly). So, he must pay all bills within 30 days of receiving them, especially the lumber companies.

Bob is pretty nice to his customers. They don't have to pay until the work is completed and they are satisfied with his work. This has occasionally led to some problems because the cash coming in is sometimes slower than the cash going out, and he would like to have a better idea of when his bills are due and when his customers will be paying.

Currently, all business records are kept in Bob's head and in one file cabinet. Sometimes he forgets which jobs he bid on and how much he bid on them. He doesn't call potential customers to ask about earlier bids, but this could increase business. He wants reports on suppliers that

need to be paid and customers that are slow in paying their bills. So, he wants to computerize these aspects of the business to make things more efficient. Can you help him?

Here is what you need to be able to provide Bob in order to land your first consulting contract:

1. An E-R diagram of the situation.
2. A script that will create the tables and enter a minimum of 3 sets of data for each table (more if necessary for orders, bids, etc.).
3. Documentation of your analysis of Bob's data needs
4. Some sample queries with sample outputs for the reports that he wants.
5. A nice business cover letter that summarizes this package as a prototype/proposal and details how your services will meet his business data needs.

Here is what you don't need to do:

- You don't need to computerize the entire business, just the processes that Bob finds most bothersome.
- You don't need to create forms for data entry. This is something you would do if hired as a consultant.

An outstanding project will have the following characteristics, with lower grades for projects that lack these things:

1. The E-R diagram clearly identifies the entities, relationships, and attributes.
2. The tables are in third normal form, and you used good names for the tables and the fields.

3. The sample data is useful in creating sample reports.
4. The script works with Oracle using the 10g XE interface.
5. The script has comments to make it easy to edit later. (dash dash space)
6. The sample queries that will show Bob how the database will be useful in meeting his data needs.
7. The business cover letter looks professional. A potential customer would be more likely to hire you because of it.
8. The entire package is organized in a Word document for easy viewing and editing, with diagrams and scripts copied into the Word document rather than stapled together from separate files.

Analysis of 'Bob's Fix It' Data Needs:

Our team has carefully analyzed the Bob's case study and finally as per our team analysis, Bob needs to computerize the tasks to keep track of the regular business transactions.

We have gone through the case study and created a database for Bob which has 8 entities. We made sure to follow 3NF for creating the entities. Below are the names of the eight entities.

1. Customers 2. Bids 3. Jobs 4. Items 5.Suppliers 6. Job_Item_Supplies 7. Customer_Invoice

8. Supplier_Invoice

We have identified the primary keys in each entity which are supposed to hold unique values and we have also identified the foreign keys as required. In this document, we have included the ER diagrams and have designed the fields for every entity as required.

Sample data is inserted into each entity of Bob's database to generate sample reports. The document includes sample Queries and their output.

Bob's Fix IT Database has been computerized to simplify Bob's daily business transactions and generate reports as required, which would help bob in tracking his Bids, accepted contracts, number of hours assigned to a particular Job, customer payment information, supplier payment information etc.

In the created database, some sample data is inserted into each entity to generate sample reports. We have mentioned some sample queries which would help Bob in retrieving the reports or any kind of information from the designed tables.

With the help of the tables, Bob's data has been computerized, for making easy retrieval of information. All these changes would help Bob in making his business more effective by making payments to supplier and collecting the payments from clients on time. Thus help Bob's business work effectively and more efficiently.

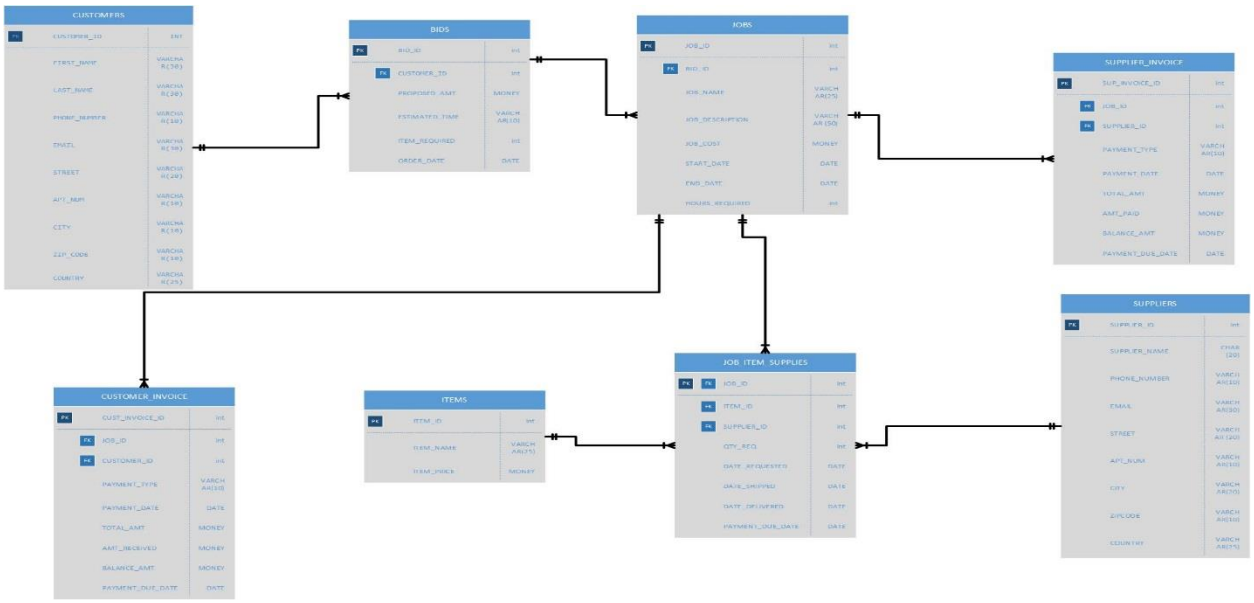
List of Entities:

The final list of entities identified for the Bob's Fixit database are

1. Customers
2. Bids
3. Jobs
4. Items
5. Suppliers
6. Job_Item_Supplies

7. Customer_Invoice
8. Supplier_Invoice

E-R Diagram:



The Visio file for the E-R diagram has been attached below for the reference.

Bobs_FixIt_ERD.vsdx

All the tables are in third normal form with good and understandable names for the table and the fields.

Business Rules:

1. There are eight entities involved in Bob's Fix It database.
2. They are CUSTOMERS, BIDS, JOBS, ITEMS, SUPPLIERS, JOB_ITEM_SUPPLIES, CUSTOMER_INOVICE and SUPPLIER_INVOICE.

3. The entity named CUSTOMERS will have the details related to all the customers who approach Bob.
4. The entity named BIDS will have the details related to all the bids provided by Bob to the orders placed by the customers.
5. The entity named JOBS will have the details related to all the jobs involved with each bid.
6. The entity named ITEMS will have the details related to all the items that are required to complete the jobs.
7. The entity named SUPPLIERS will have the details related to all the suppliers who supplies the items to Bob.
8. The entity named JOB_ITEM_SUPPLIES will have the details related to the jobs, the items required for each job and the supplier information who supplies the required items for the jobs.
9. The entity named CUSTOMER_INVOICE will have the details related to the payments made by the customers to Bob.
10. The entity named SUPPLIER_INVOICE will have the details related to the payments made by Bob to the suppliers.
11. The entities CUSTOMERS and BIDS will have one to many relationship as a single customer can approach Bob for many orders.
12. The entities BIDS and JOBS will have one to many relationship as a single bid can have multiple jobs associated with it.
13. The entities ITEMS, JOBS and SUPPLIERS will have one to many relationship each with the JOB_ITEM_SUPPLIES table.

14. The entities JOBS and CUSTOMER_INVOICE will have one to many relationship as each customer can pay in installments for each job. So, one job can have many payments by the customers.
15. The entities JOBS and SUPPLIER_INVOICE will have one to many relationship as each supplier can be paid in installments for each job by Bob. So, one job can have many payments to the suppliers.

Creating the tables and inserting the values:

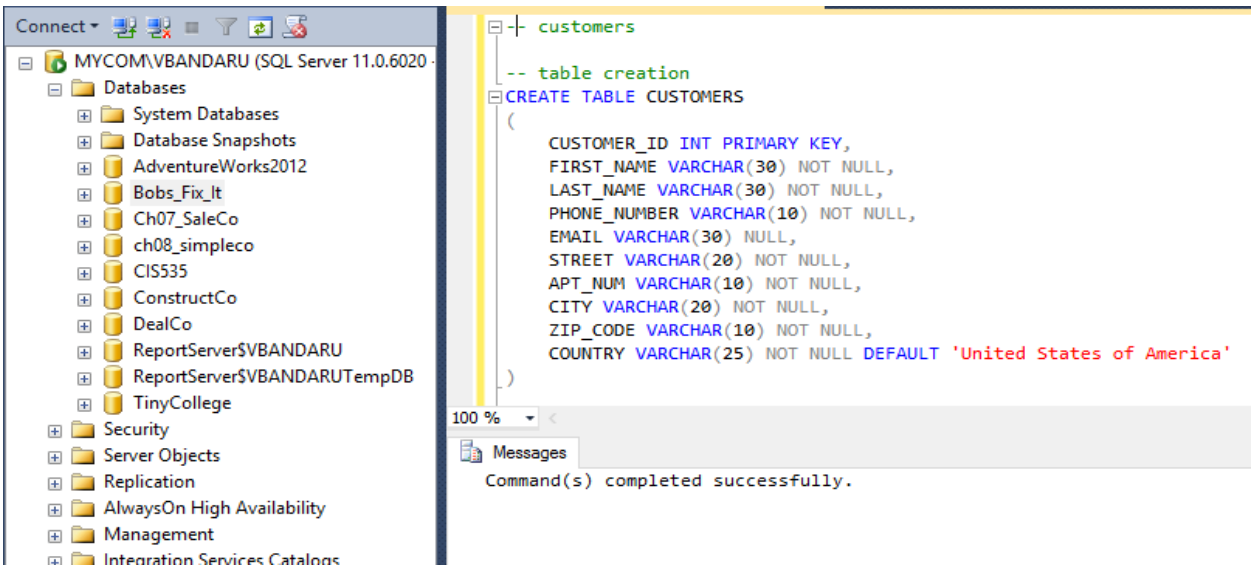
The tables that are created in the Bobs_Fix_It database are based on the above attached ER diagram. Below are screen shot of the database that has been created for the project, with create and insert scripts. There are eight tables created as specified in the earlier sections.

1. **Customers:**

   Table creation script:

```sql
-- Customers

-- Table creation
CREATE TABLE CUSTOMERS
(
       CUSTOMER_ID INT PRIMARY KEY,
       FIRST_NAME VARCHAR(30) NOT NULL,
       LAST_NAME VARCHAR(30) NOT NULL,
       PHONE_NUMBER VARCHAR(10) NOT NULL,
       EMAIL VARCHAR(30) NULL,
       STREET VARCHAR(20) NOT NULL,
       APT_NUM VARCHAR(10) NOT NULL,
       CITY VARCHAR(20) NOT NULL,
       ZIP_CODE VARCHAR(10) NOT NULL,
       COUNTRY VARCHAR(25) NOT NULL DEFAULT 'United States of America'
)

-- adding unique key constraint
ALTER TABLE CUSTOMERS
  ADD CONSTRAINT CUSTOMER_UK
  UNIQUE (FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL)
```

To add a unique key combination for the values in the database, we created a unique key based on the first name, middle name, last name, phone number and the email address. The reason why the address fields are not considered is that two customers can stay in a single apartment. However, their names, phone number and the email address would be different. There might be cases where the names will be same too. But, with the unique key enabled on all the five columns, even if the name matches, the phone number and the email address will differ and this will ensure there is no duplicate data in the table.



Insertion scripts:

```
-- insert statements
INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES (1, 'Vamshi', 'Bandaru', '4024024024', 'vkbandaru@gmail.com', '184th
Street', '18423', 'Omaha' , '68154');

INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES (2, 'Olu', 'Oyesiku', '6146526235', 'oluoyesiku@gmail.com', '67th Street',
'67156', 'Omaha' , '68154');

INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES (3, 'Madhuri', 'Satturi', '4145256355', 'msatturi@gmail.com', '27th
Street', '2751', 'Omaha' , '68154');
```

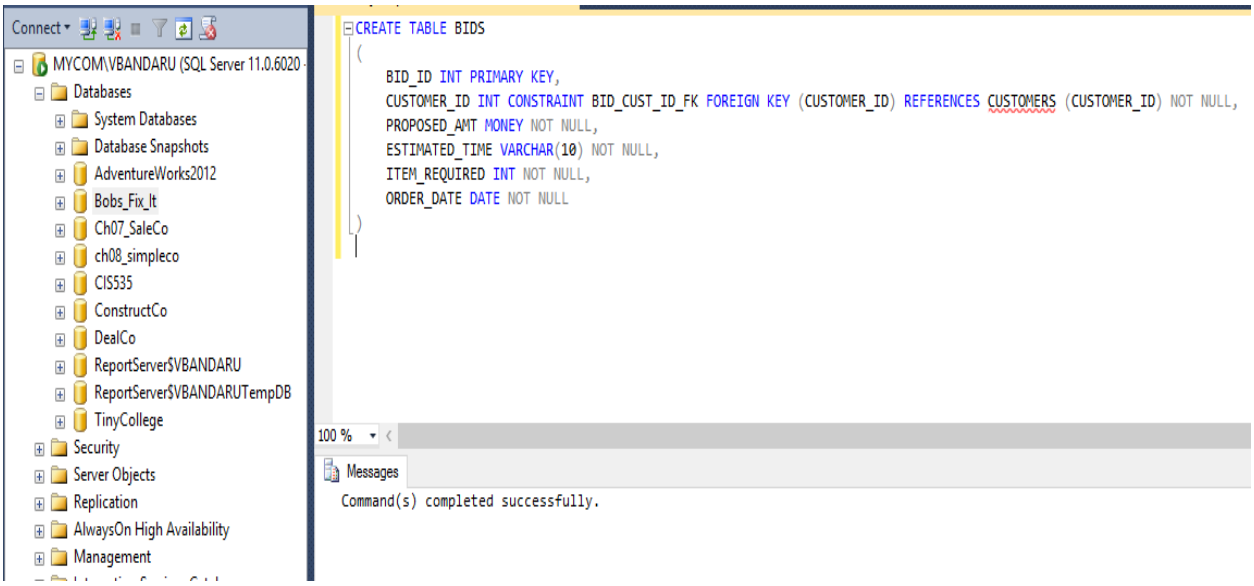
```
INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES (4, 'Paul', 'Marthala', '4325987841', 'psreddy@gmail.com', '108th Steet',
'10823', 'Omaha' , '68154');

INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES (5, 'Shiva', 'Vasa', '2145632598', 'shvasa@gmail.com', '114th Street',
'11485', 'Omaha' , '68154');

INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES (6, 'Isys', 'Ervine', '5989897877', 'iervine@gmail.com', '112th Street',
'11256', 'Omaha' , '68154');

INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES(7,'John','Smith','4024910291','john.smith@gmail.com','Sunridge
Road','101','Lincoln','68505');

INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES(8,'Kevin','Anderson','4024110110','kevin.anderson@aol.com','Faulkner
Drive','209','Lincoln','68516');

INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES(9,'Laura','Baker','4024116102','laura.baker@aol.com','72nd
street','209','Omaha','68114');

INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES(10,'Cicily','Tanner','7575887223','cicily.tanner@aol.com','Pacific
street','706','Omaha','68114');

INSERT INTO CUSTOMERS (CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES(11,'Kim','Fernandez','4024110410','kim.fernandez@aol.com','2nd street',
'Lincoln','68519');


INSERT INTO CUSTOMERS (CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES(12,'Rohit','Malhotra','4024110510','rohit.malhotra@aol.com','19th
street','Lincoln','68520');

INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES(13,'Tom','Joseph','4024110610','Tom.joseph@aol.com','West Drive',
'245','Lincoln','68495');


INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES(14,'Tony','Anderson','4024110710','Tony.anderson@aol.com','Q Road',
'225','Lincoln','68521');
```

```
INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES(15,'Rosy','Pam','4024110810','Rosy.Pam@aol.com','Nebraska
Drive','209','Lincoln','68525');

INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES(16,'Ria','smith','4024110210','ria.smith@aol.com','Ridge Drive',
'211','Lincoln','68518');


INSERT INTO CUSTOMERS(CUSTOMER_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL,
STREET, APT_NUM, CITY, ZIP_CODE)
VALUES(17,'Bob','Anderson','4024110310','bob.anderson@aol.com','113th Street',
'223''Lincoln','68509');
```

2. Bids:

Table creation script:

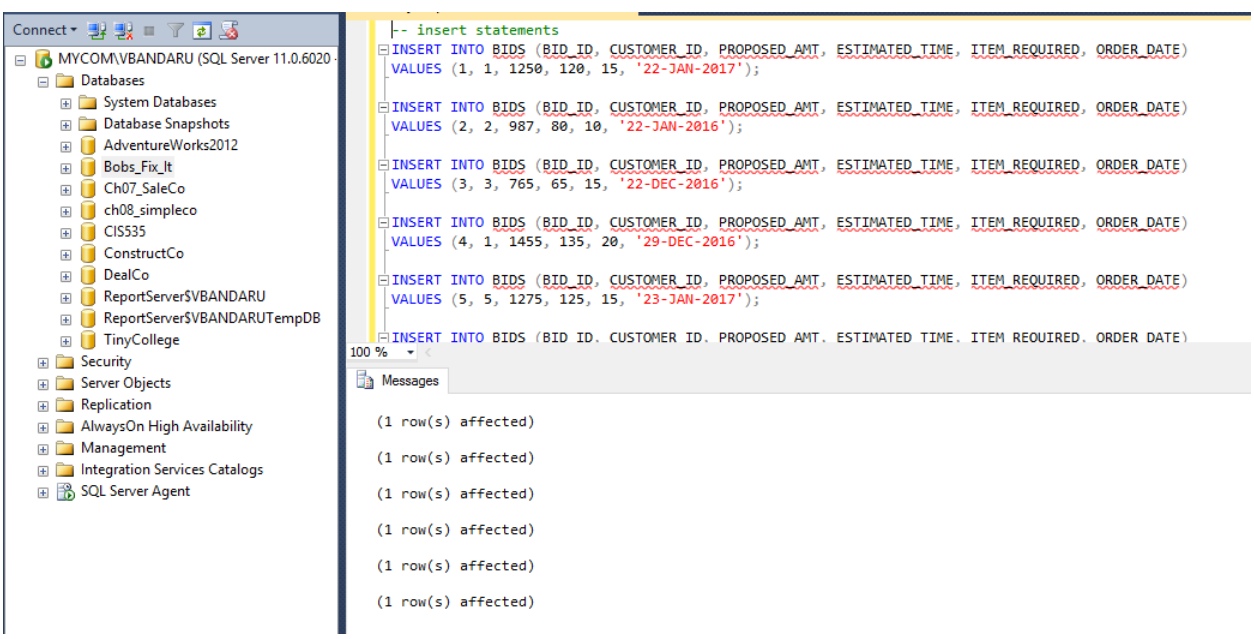
```
CREATE TABLE BIDS
(
	BID_ID INT PRIMARY KEY,
	CUSTOMER_ID INT CONSTRAINT BID_CUST_ID_FK FOREIGN KEY (CUSTOMER_ID)
REFERENCES CUSTOMERS (CUSTOMER_ID) NOT NULL,
	PROPOSED_AMT MONEY NOT NULL,
	ESTIMATED_TIME VARCHAR(10) NOT NULL,
	ITEM_REQUIRED INT NOT NULL,
	ORDER_DATE DATE NOT NULL)
```



Insertion scripts:

```
INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (1, 1, 1250, 120, 15, '22-JAN-2017');

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (2, 2, 987, 80, 10, '22-JAN-2016');

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (3, 3, 765, 65, 15, '22-DEC-2016');

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (4, 1, 1455, 135, 20, '29-DEC-2016');

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (5, 5, 1275, 125, 15, '23-JAN-2017');

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (6, 3, 1600, 150, 25, '21-JAN-2017');

INSERT INTO BIDS(BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME, ITEM_REQUIRED,
ORDER_DATE)
VALUES(7, 7, 4000, 200, 20, '22-FEB-2017');

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (8, 7, 2000, 100, 20, '20-FEB-2017');
```

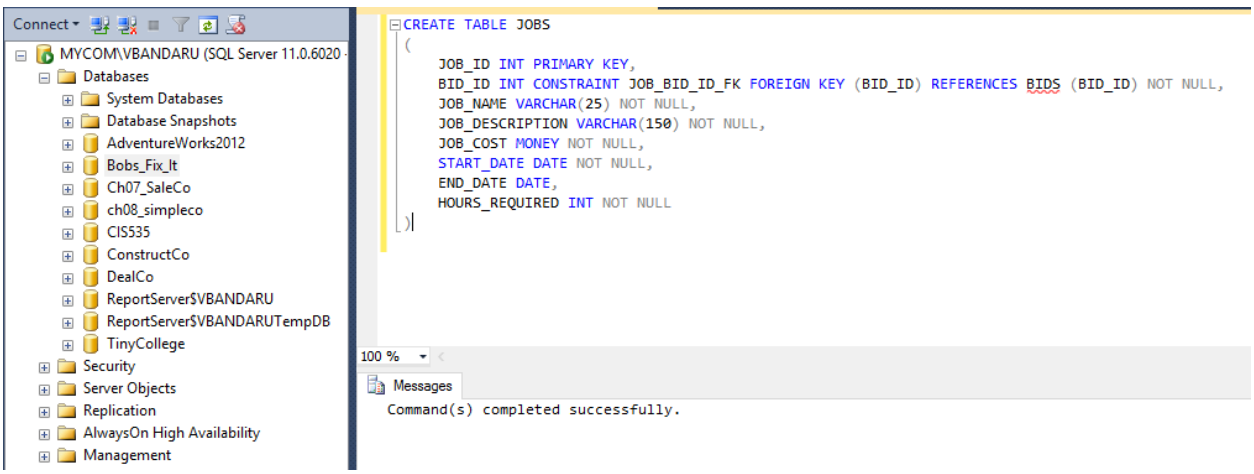
```sql
INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (9, 8, 1000, 100, 10, '22-FEB-2017');

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (10, 8, 3000, 150, 20, '23-FEB-2017');

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (11, 9, 1500, 150, 25, '24-FEB-2017');

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (12, 10, 4000, 250, 15, '23-FEB-2017');

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (13, 11, 3000, 150, 20, '24-FEB-2017');

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (14, 12, 2000, 50, 10, '24-FEB-2017');

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (15, 13, 1000, 100, 10, '24-FEB-2017')

INSERT INTO BIDS (BID_ID, CUSTOMER_ID, PROPOSED_AMT, ESTIMATED_TIME,
ITEM_REQUIRED, ORDER_DATE)
VALUES (16, 9, 2000, 100, 10, '23-FEB-2017');
```



3. Jobs:

Table creation script:

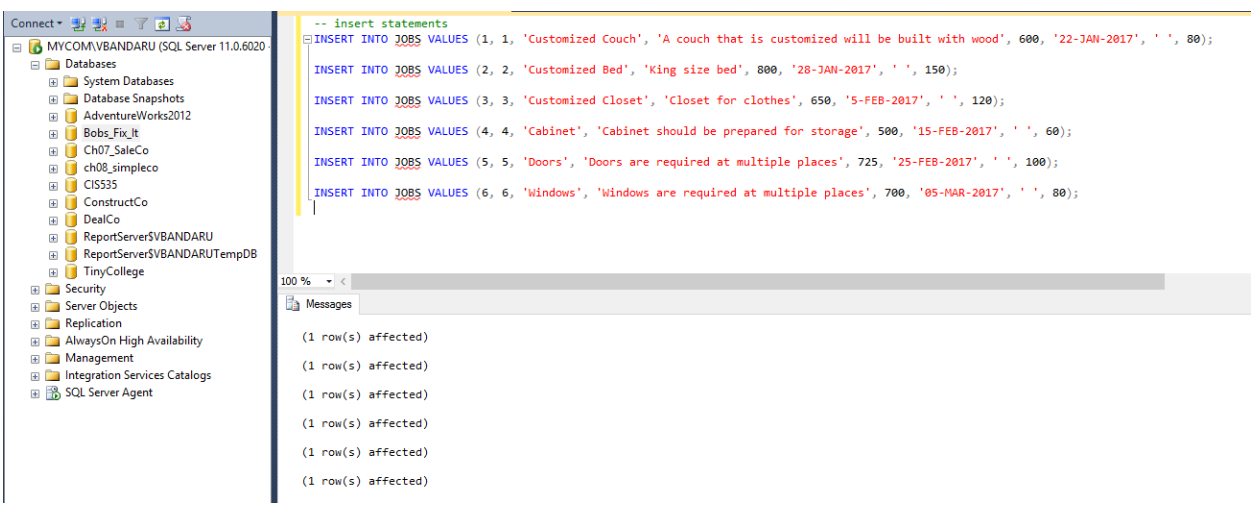
```
CREATE TABLE JOBS
(
        JOB_ID INT PRIMARY KEY,
        BID_ID INT CONSTRAINT JOB_BID_ID_FK FOREIGN KEY (BID_ID) REFERENCES BIDS
        (BID_ID) NOT NULL,
        JOB_NAME VARCHAR(25) NOT NULL,
        JOB_DESCRIPTION VARCHAR(150) NOT NULL,
        JOB_COST MONEY NOT NULL,
        START_DATE DATE NOT NULL,
        END_DATE DATE,
        HOURS_REQUIRED INT NOT NULL
)
```



Insertion scripts:

```
-- insert statements
INSERT INTO JOBS VALUES (1, 1, 'Customized Couch', 'A couch that is customized
will be built with wood', 600, '22-JAN-2017', ' ', 80);

INSERT INTO JOBS VALUES (2, 2, 'Customized Bed', 'King size bed', 800, '28-JAN-
2017', ' ', 150);

INSERT INTO JOBS VALUES (3, 3, 'Customized Closet', 'Closet for clothes', 650, '5-
FEB-2017', ' ', 120);

INSERT INTO JOBS VALUES (4, 4, 'Cabinet', 'Cabinet should be prepared for
storage', 500, '15-FEB-2017', ' ', 60);

INSERT INTO JOBS VALUES (5, 5, 'Doors', 'Doors are required at multiple places',
725, '25-FEB-2017', ' ', 100);

INSERT INTO JOBS VALUES (6, 6, 'Windows', 'Windows are required at multiple
places', 700, '05-MAR-2017', ' ', 80);
```

```
INSERT INTO JOBS VALUES (7, 7, 'Door Job', 'A door knob need to installed for 150
doors',4000, '22-FEB-2017', ' ', 80);

INSERT INTO JOBS VALUES (8, 7, 'Electrical Job', 'Install Electrical outlets in a
house', 2000, '20-FEB-2017', ' ', 16);

INSERT INTO JOBS VALUES (9, 4, 'Cabinet', Cabinet should be prepared for kitchen',
500, '24-FEB-2017', ' ', 60);

INSERT INTO JOBS VALUES (10, 7, 'Electrical Job', 'Install New Switches in
Kitchen', 500, '20-FEB-2017', ' ', 3);

INSERT INTO JOBS VALUES (12, 11, 'Electrical Job', 'Install Electrical outlets in
the patio', 300, '20-FEB-2017', ' ', 16);

INSERT INTO JOBS VALUES (13, 12, 'Cabinet', 'Kitchen Cabinets installation', 4000,
'22-FEB-2017', ' ', 80);

INSERT INTO JOBS VALUES (14, 13, 'Window', 'Window Blinds installation in 5
homes', 4500, '20-FEB-2017', ' ', 16);


INSERT INTO JOBS VALUES (15, 14, 'Door Job', 'A door knob need to installed for
150 doors',4000, '22-FEB-2017', ' ', 80)

INSERT INTO JOBS VALUES (16, 9, 'Patio', 'Patio door installation', 3000, '22-FEB-
2017', ' ', 5);
```

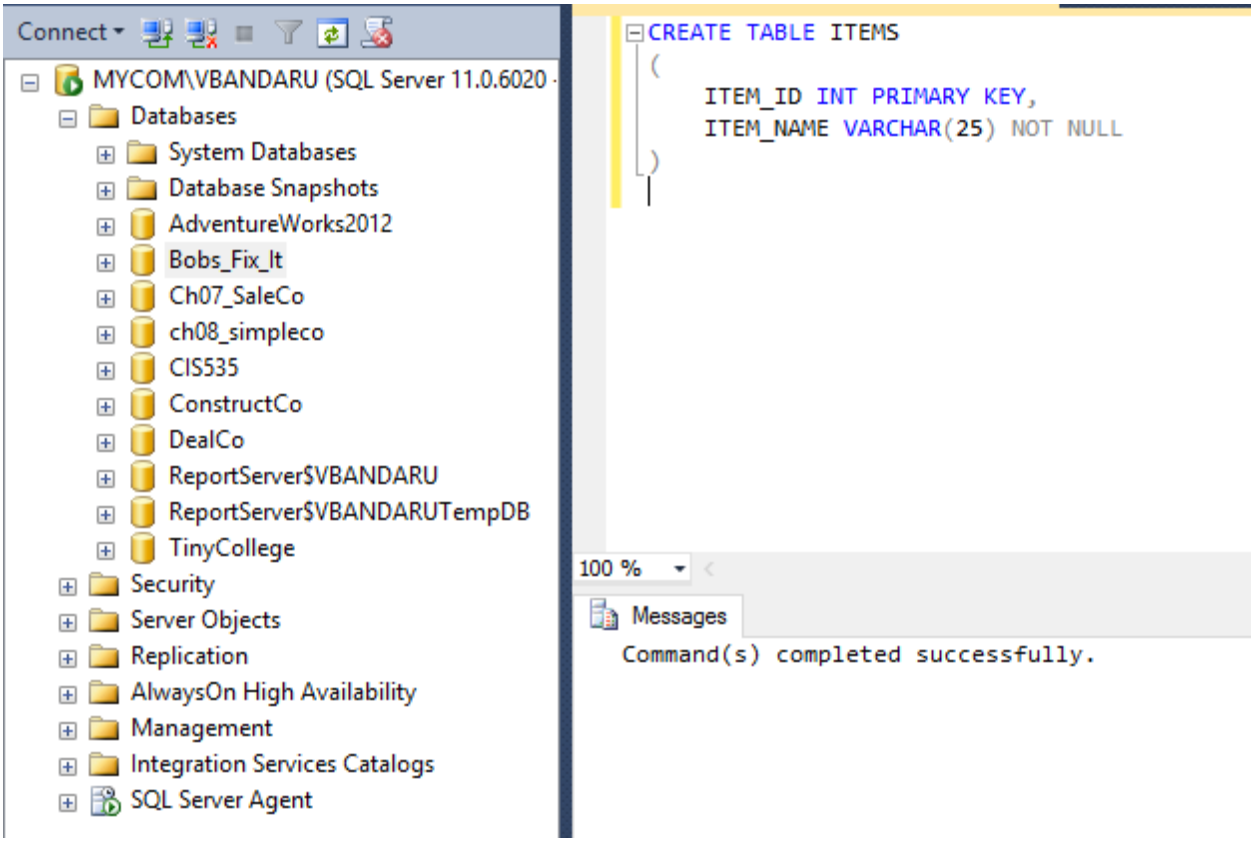


4. Items:

Table creation script:

```
CREATE TABLE ITEMS
(
        ITEM_ID INT PRIMARY KEY,
```

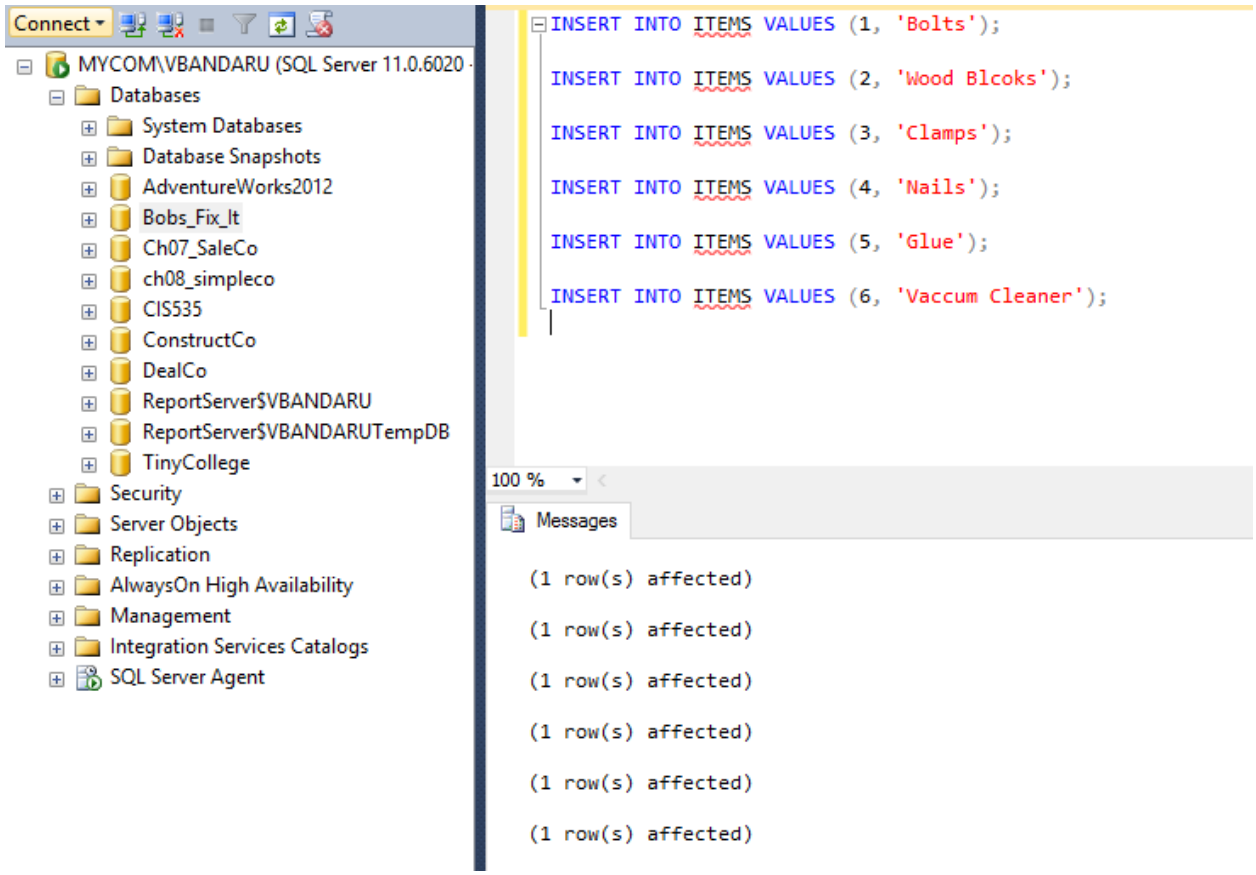
```
        ITEM_NAME VARCHAR(25) NOT NULL
)
```



Insertion scripts:

```
INSERT INTO ITEMS VALUES (1, 'Bolts');

INSERT INTO ITEMS VALUES (2, 'Wood Blcoks');

INSERT INTO ITEMS VALUES (3, 'Clamps');

INSERT INTO ITEMS VALUES (4, 'Nails');

INSERT INTO ITEMS VALUES (5, 'Glue');

INSERT INTO ITEMS VALUES (6, 'Vaccum Cleaner');

INSERT INTO ITEMS VALUES (7, 'Door Knobs');

INSERT INTO ITEMS VALUES (8, 'Switch Board');

INSERT INTO ITEMS VALUES (9, 'Screws');

INSERT INTO ITEMS VALUES (10, 'Hammer');

INSERT INTO ITEMS VALUES (11, 'Cock');
```

```
INSERT INTO ITEMS VALUES (12, 'Door Lock');

INSERT INTO ITEMS VALUES (14, 'Window');

INSERT INTO ITEMS VALUES (13, 'Vaccum Cleaner');

INSERT INTO ITEMS VALUES (15, 'Paint');

INSERT INTO ITEMS VALUES (16, 'Blinds');
```

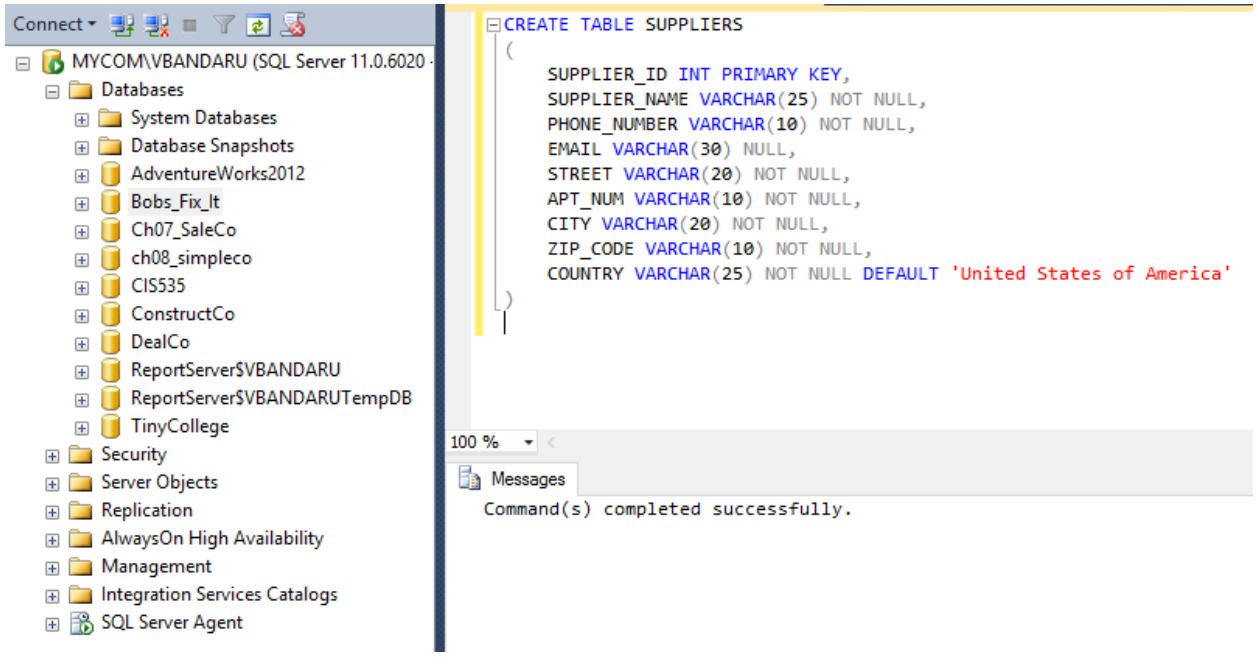


5. Suppliers:

Table creation script:

```
CREATE TABLE SUPPLIERS
(
        SUPPLIER_ID INT PRIMARY KEY,
```

```
        SUPPLIER_NAME VARCHAR(25) NOT NULL,
        PHONE_NUMBER VARCHAR(10) NOT NULL,
        EMAIL VARCHAR(30) NULL,
        STREET VARCHAR(20) NOT NULL,
        APT_NUM VARCHAR(10) NOT NULL,
        CITY VARCHAR(20) NOT NULL,
        ZIP_CODE VARCHAR(10) NOT NULL,
        COUNTRY VARCHAR(25) NOT NULL DEFAULT 'United States of America'
)
```



Insertion scripts:

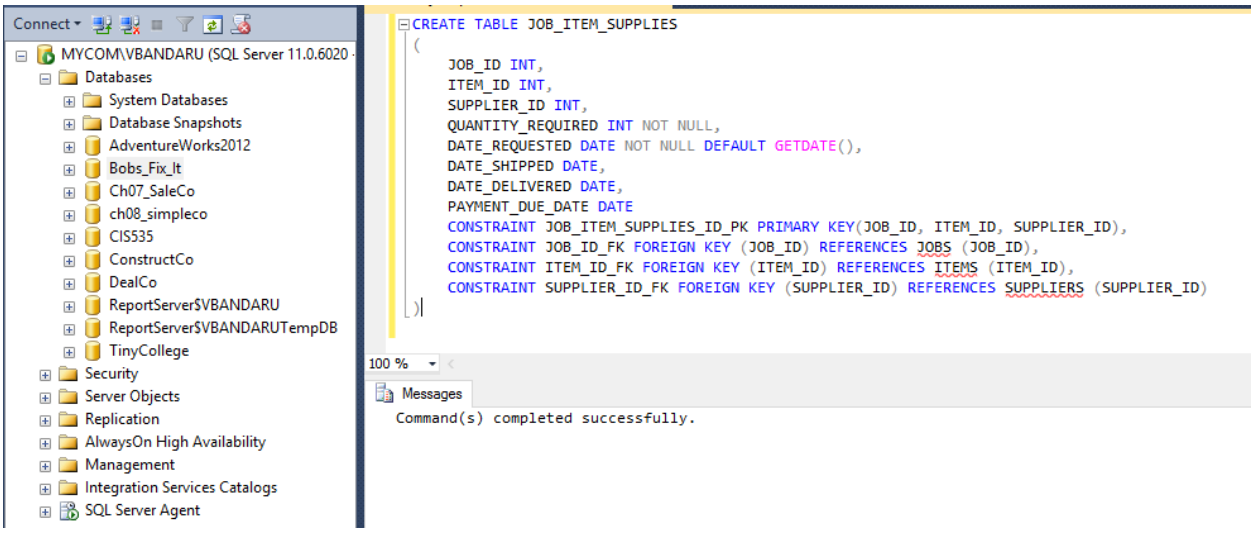
```
INSERT INTO SUPPLIERS VALUES (1, 'ABC Inc.', 4524564571, 'abc@abc.net', '118
Street', '11823', 'Hyderabad','500089', 'India');

INSERT INTO SUPPLIERS VALUES (2, 'XYZ Inc.', 9998887744, 'xyz@xyz.net', '157
Street', '15721', 'Chicago','78987', 'United States of America');

INSERT INTO SUPPLIERS VALUES (3, 'Furniture Bros.', 8889997745, 'fb@fb.net', '102
Street', '10254', 'New York','12456', 'United States of America');

INSERT INTO SUPPLIERS VALUES (4, 'Furniture Unplugged.', 4512121212, 'fu@fu.net',
'19 Street', '1923', 'La vista','68441', 'United States of America');
INSERT INTO SUPPLIERS VALUES (5, 'Josh Bros', 7456981236, 'jb@jb.net', '197
Street', '19754', 'Kearney','67789', 'United States of America');

INSERT INTO SUPPLIERS VALUES (6, 'Love It Inc.', 3653653653, 'li@li.net', '12
Street', '1245', 'Omaha','68154', 'United States of America');

INSERT INTO SUPPLIERS VALUES (7, 'Lowes', 4524564571, 'info@lowes.com', '72nd
Street', '299', 'LINCOLN','68516', 'United States of America');
```

```
INSERT INTO SUPPLIERS VALUES (8, 'Menards', 4524562323, 'info@menards.com', '86th
Street', '100', 'LINCOLN','68516', 'United States of America');

INSERT INTO SUPPLIERS VALUES (9, 'Home Depot', 3475662596, 'info@homedepot.com',
'Kearney Drive', '107', 'San Diego','68516', 'United States of America');
INSERT INTO SUPPLIERS VALUES (10, 'Paints ABC Inc.', 7512121212,
'paintabc@paint.net', '14 Street', '1333', 'La vista','68451', 'United States of
America');

INSERT INTO SUPPLIERS VALUES (11, 'Nuts Inc', 6456981236, 'nutsinc@nut.net', '1967
Street', '19754', 'Kearney','67789', 'United States of America');

INSERT INTO SUPPLIERS VALUES (12, 'Korles Nuts.', 2653653653,
'korlesnuts@nuts.net', '45th Street', '1355', 'Omaha','68344', 'United States of
America');

INSERT INTO SUPPLIERS VALUES (13, 'PQR Ltd',41344564571, 'pqrltd@pqr.com', '63rd
Street', '299', 'LINCOLN','68556', 'United States of America');

INSERT INTO SUPPLIERS VALUES (14, 'L&L Bolts', 3224562323, 'l&lbolts@llbolt.com',
'86th Street', '125', 'LINCOLN','68236', 'United States of America');

INSERT INTO SUPPLIERS VALUES (15, 'Jode Bolts Inc', 243564571,
'jodebolts@jode.com', '85th Street', '299', 'LINCOLN','68516', 'United States of
America');

INSERT INTO SUPPLIERS VALUES (16, 'M & M Nuts and Bolts.', 9889997745,
'mmnuts@mm.net', '112 Street', '10234', 'New York','12476', 'United States of
America');
```

6. Job_Item_Supplies:

Table creation script:

```sql
CREATE TABLE JOB_ITEM_SUPPLIES
(
	JOB_ID INT,
	ITEM_ID INT,
	SUPPLIER_ID INT,
	QUANTITY_REQUIRED INT NOT NULL,
	DATE_REQUESTED DATE NOT NULL DEFAULT GETDATE(),
	DATE_SHIPPED DATE,
	DATE_DELIVERED DATE,
	PAYMENT_DUE_DATE DATE
	CONSTRAINT JOB_ITEM_SUPPLIES_ID_PK PRIMARY KEY(JOB_ID, ITEM_ID,
	SUPPLIER_ID),
	CONSTRAINT JOB_ID_FK FOREIGN KEY (JOB_ID) REFERENCES JOBS (JOB_ID),
	CONSTRAINT ITEM_ID_FK FOREIGN KEY (ITEM_ID) REFERENCES ITEMS (ITEM_ID),
	CONSTRAINT SUPPLIER_ID_FK FOREIGN KEY (SUPPLIER_ID) REFERENCES SUPPLIERS
	(SUPPLIER_ID)
)
```



Insertion scripts:

```sql
INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 1, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 1, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 1, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 2, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 2, 4, 200);
```

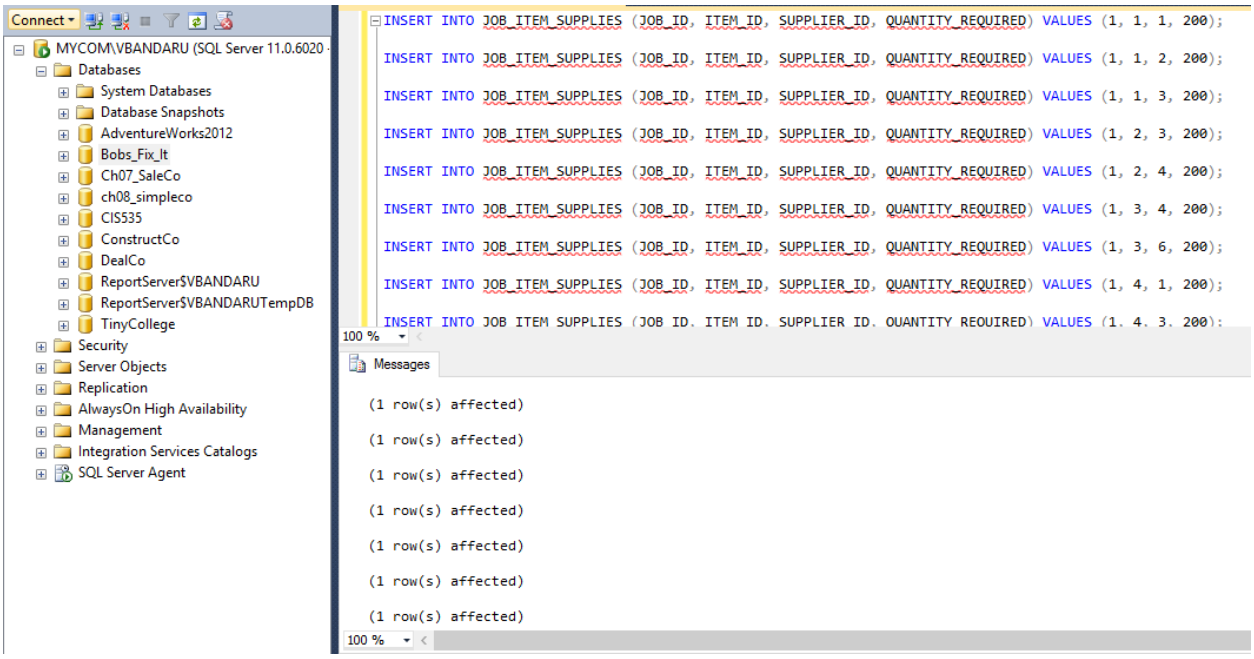
```sql
INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 3, 4, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 3, 6, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 4, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 4, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 5, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 5, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 5, 6, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 5, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (1, 6, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (2, 1, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (2, 1, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (2, 1, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (2, 2, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (2, 2, 4, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (2, 3, 4, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (2, 3, 6, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (2, 4, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (2, 4, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (2, 5, 2, 200);
```

```sql
INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (2, 5, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (2, 1, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (3, 1, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (3, 1, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (3, 2, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (3, 2, 4, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (3, 3, 4, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (3, 3, 6, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (3, 4, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (3, 4, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (3, 5, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (3, 5, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (3, 5, 6, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (3, 6, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 1, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 1, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 1, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 2, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 2, 4, 200);
```

```sql
INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 3, 4, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 3, 6, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 4, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 4, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 5, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 5, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 5, 6, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 5, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (4, 6, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 1, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 1, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 1, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 2, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 2, 4, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 3, 4, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 3, 6, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 4, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 4, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 5, 2, 200);
```

```
INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 5, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 5, 6, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 5, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (5, 6, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 1, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 1, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 1, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 2, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 2, 4, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 3, 4, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 3, 6, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 4, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 4, 3, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 5, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 5, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 5, 6, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 5, 2, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED)
VALUES (6, 6, 1, 200);

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED,
DATE_REQUESTED,DATE_SHIPPED,DATE_DELIVERED
VALUES (7, 7, 7, 150,'04-FEB-2017','07-FEB-2017','09-FEB-2017');
```

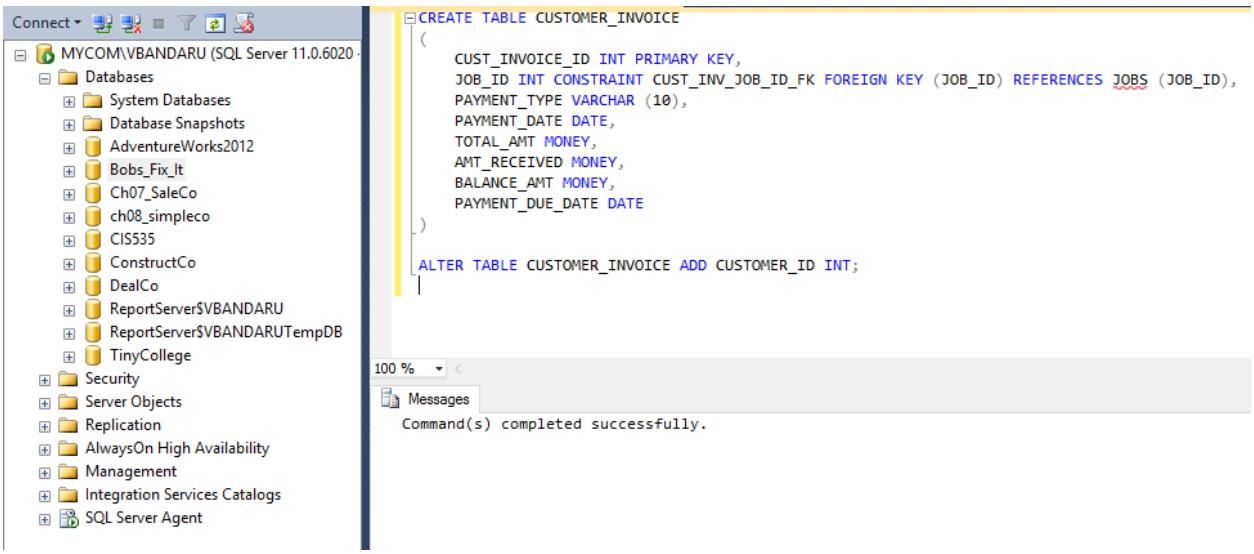
```
INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED,
DATE_REQUESTED,DATE_SHIPPED,DATE_DELIVERED)
VALUES (7, 9, 7, 500,'04-FEB-2017','07-FEB-2017','09-FEB-2017');

INSERT INTO JOB_ITEM_SUPPLIES (JOB_ID, ITEM_ID, SUPPLIER_ID, QUANTITY_REQUIRED,
DATE_REQUESTED,DATE_SHIPPED,DATE_DELIVERED)
VALUES (8, 8, 8, 40,'04-FEB-2017','11-FEB-2017','13-FEB-2017');

UPDATE JOB_ITEM_SUPPLIES SET PAYMENT_DUE_DATE = DATEADD(DAY,30,DATE_DELIVERED)
WHERE DATE_DELIVERED IS NOT NULL;
```



7. Customer_Invoice:

Table creation script:

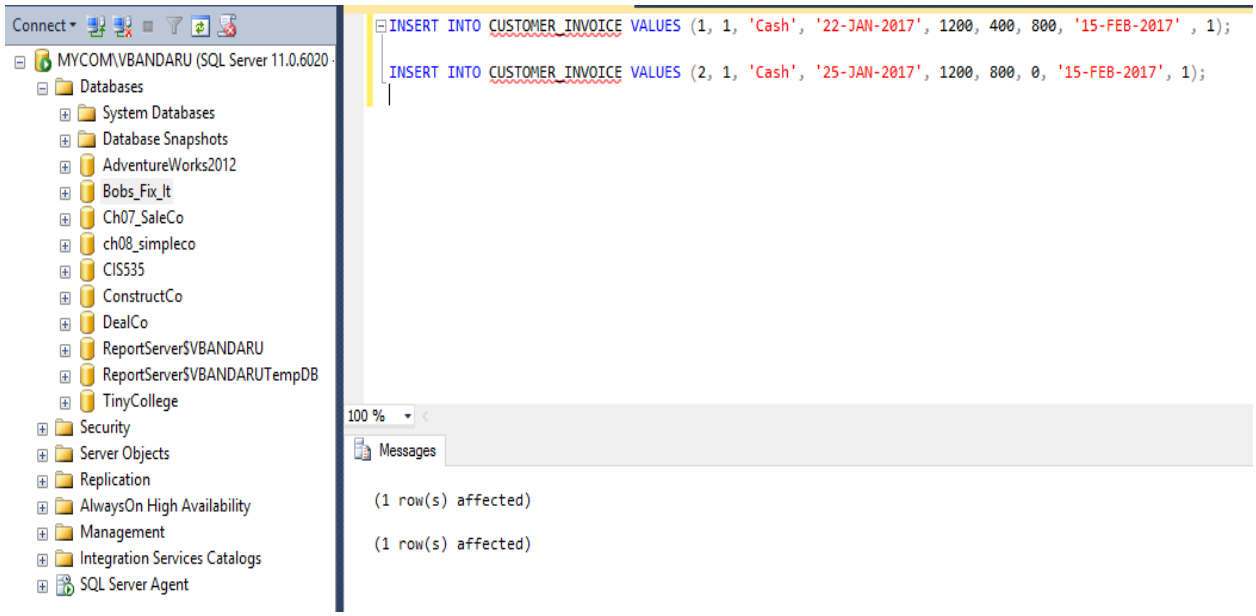
```
CREATE TABLE CUSTOMER_INVOICE
(
        CUST_INVOICE_ID INT PRIMARY KEY,
        JOB_ID INT CONSTRAINT CUST_INV_JOB_ID_FK FOREIGN KEY (JOB_ID) REFERENCES
JOBS (JOB_ID),
        PAYMENT_TYPE VARCHAR (10),
        PAYMENT_DATE DATE,
        TOTAL_AMT MONEY,
        AMT_RECEIVED MONEY,
        BALANCE_AMT MONEY,
        PAYMENT_DUE_DATE DATE
)

ALTER TABLE CUSTOMER_INVOICE ADD CUSTOMER_ID INT;
```



Insertion scripts:

```
INSERT INTO CUSTOMER_INVOICE VALUES (1, 1, 'Cash', '22-JAN-2017', 1200, 400, 800,
'15-FEB-2017', 1);

INSERT INTO CUSTOMER_INVOICE VALUES (2, 1, 'Cash', '25-JAN-2017', 1200, 800, 0,
'15-FEB-2017', 1);

INSERT INTO CUSTOMER_INVOICE VALUES (3, 7, 'Cash', '22-FEB-2017', 4000, 3000,
1000, '15-MAR-2017' , 7);

INSERT INTO CUSTOMER_INVOICE VALUES (4, 8, 'Cash', '20-FEB-2017', 2000, 1200, 800,
'15-MAR-2017', 7);

INSERT INTO CUSTOMER_INVOICE VALUES (5, 5, 'Cash', '24-FEB-2017', 1200, 700, 400,
'15-MAR-2017', 7);


INSERT INTO CUSTOMER_INVOICE VALUES (6, 2, 'Cash', '24-FEB-2017', 5000, 200, 600,
'15-MAR-2017', 7);


INSERT INTO CUSTOMER_INVOICE VALUES (7, 6, 'Cash', '24-FEB-2017', 5000, 100, 300,
'15-MAR-2017', 7);


INSERT INTO CUSTOMER_INVOICE VALUES (8, 7, 'Cash', '20-FEB-2017', 2000, 1200, 800,
'15-MAR-2017', 7);

INSERT INTO CUSTOMER_INVOICE VALUES (9, 5, 'Cash', '22-JAN-2017', 1400, 400, 1000,
'15-FEB-2017', 1);

INSERT INTO CUSTOMER_INVOICE VALUES (10, 1, 'Cash', '25-JAN-2017', 1000, 800, 200,
'15-FEB-2017', 1);

INSERT INTO CUSTOMER_INVOICE VALUES (11, 2, 'Cash', '22-JAN-2017', 1000, 400, 600,
'15-FEB-2017', 1);
```

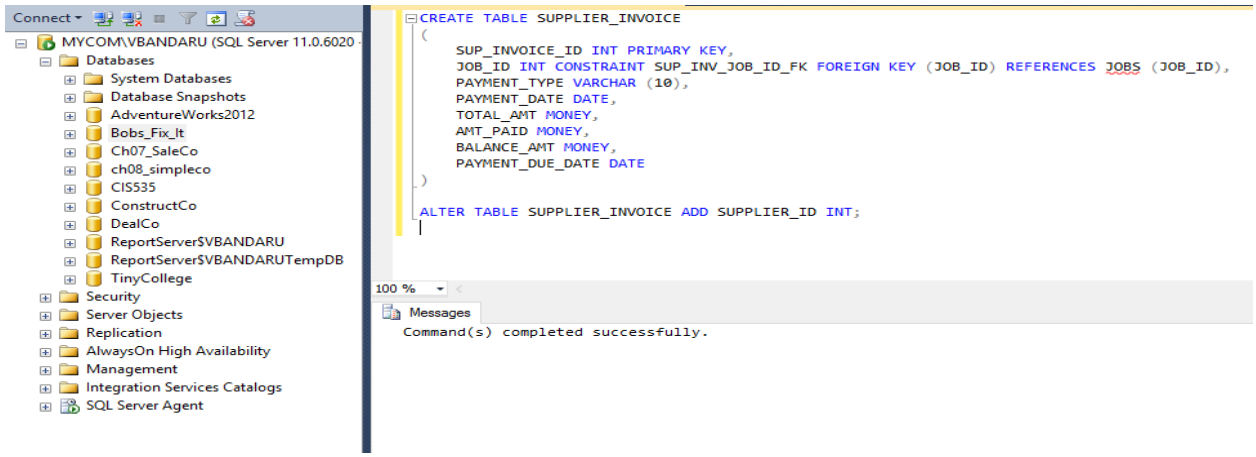
```
INSERT INTO CUSTOMER_INVOICE VALUES (12, 1, 'Cash', '25-JAN-2017', 1200, 800, 0,
'15-FEB-2017', 1);

INSERT INTO CUSTOMER_INVOICE VALUES (13, 8, 'Cash', '22-FEB-2017', 2000, 1000,
1000, '15-MAR-2017', 7);

INSERT INTO CUSTOMER_INVOICE VALUES (14, 5, 'Cash', '20-FEB-2017', 725, 225, 475,
'15-MAR-2017', 7);


INSERT INTO CUSTOMER_INVOICE VALUES (15, 1, 'Cash', '20-FEB-2017', 2000, 1200,800,
'15-MAR-2017', 7)
```



8. Supplier_Invoice:

Table creation script:

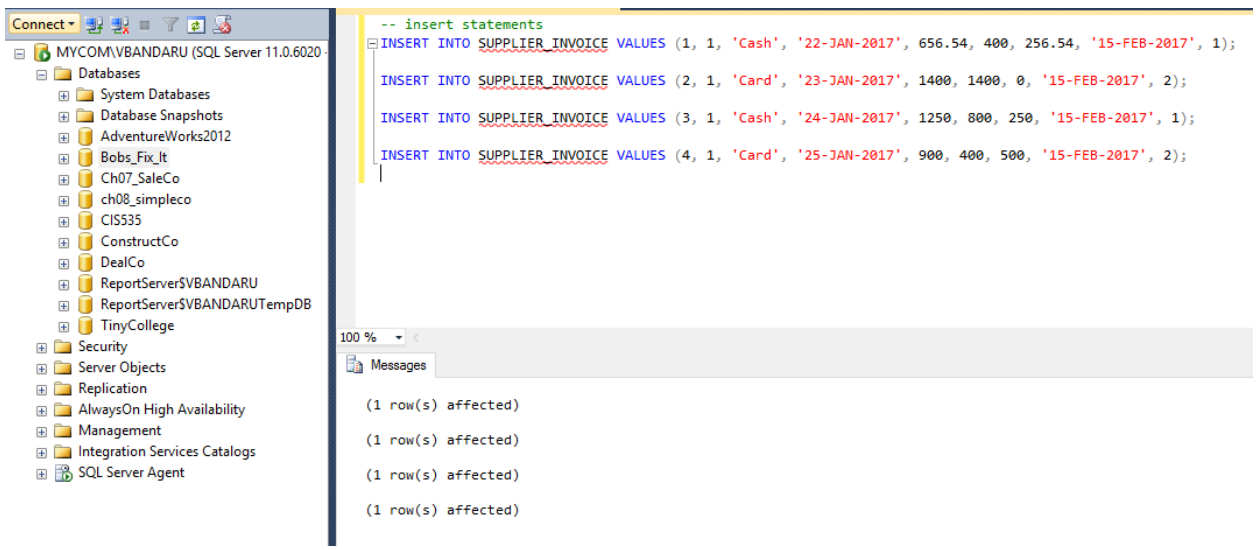
```
CREATE TABLE SUPPLIER_INVOICE
(
	SUP_INVOICE_ID INT PRIMARY KEY,
	JOB_ID INT CONSTRAINT SUP_INV_JOB_ID_FK FOREIGN KEY (JOB_ID) REFERENCES
JOBS (JOB_ID),
	PAYMENT_TYPE VARCHAR (10),
	PAYMENT_DATE DATE,
	TOTAL_AMT MONEY,
	AMT_PAID MONEY,
	BALANCE_AMT MONEY,
	PAYMENT_DUE_DATE DATE
)

ALTER TABLE SUPPLIER_INVOICE ADD SUPPLIER_ID INT;
```



Insertion scripts:

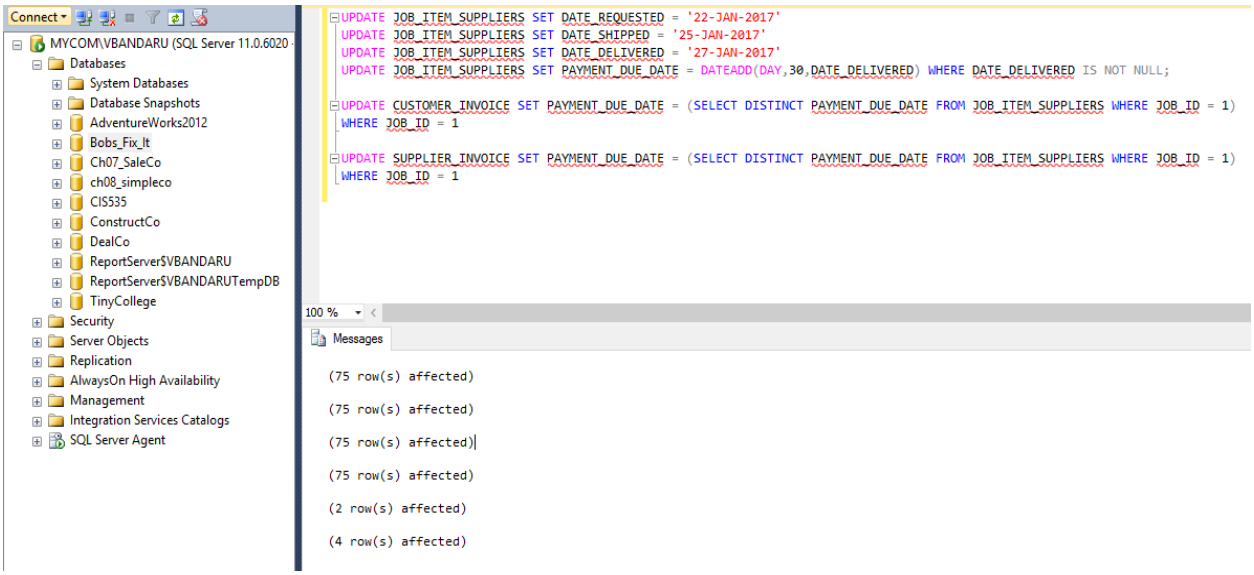
```
-- insert statements
INSERT INTO SUPPLIER_INVOICE VALUES (1, 1, 'Cash', '22-JAN-2017', 656.54, 400,
256.54, '15-FEB-2017', 1);

INSERT INTO SUPPLIER_INVOICE VALUES (2, 1, 'Card', '23-JAN-2017', 1400, 1400, 0,
'15-FEB-2017', 2);

INSERT INTO SUPPLIER_INVOICE VALUES (3, 1, 'Cash', '24-JAN-2017', 1250, 800, 250,
'15-FEB-2017', 1);

INSERT INTO SUPPLIER_INVOICE VALUES (4, 1, 'Card', '25-JAN-2017', 900, 400, 500,
'15-FEB-2017', 2);

INSERT INTO SUPPLIER_INVOICE VALUES (5, 7, 'Cash', '22-FEB-2017', 2000, 1500, 500,
'15-MAR-2017', 7);

INSERT INTO SUPPLIER_INVOICE VALUES (6, 7, 'Cash', '22-FEB-2017', 100,100, 0,
NULL, 7);

INSERT INTO SUPPLIER_INVOICE VALUES (7, 8, 'Cash', '22-FEB-2017', 1000, 800, 200,
'15-MAR-2017', 8);

INSERT INTO SUPPLIER_INVOICE VALUES (8, 7, 'Cash', '22-FEB-2017', 200,200, 0,
NULL, 8);

INSERT INTO SUPPLIER_INVOICE VALUES (9, 7, 'Card', '22-FEB-2017', 400,200, 100,
NULL, 2);

INSERT INTO SUPPLIER_INVOICE VALUES (10, 7, 'Cash', '22-FEB-2017', 2000, 1500,
500, '15-MAR-2017', 8);

INSERT INTO SUPPLIER_INVOICE VALUES (11, 7, 'Cash', '22-FEB-2017', 100,100, 0,
NULL, 7);

INSERT INTO SUPPLIER_INVOICE VALUES (12, 8, 'Cash', '22-FEB-2017', 2000, 1200,
800, '15-MAR-2017', 8);
```

```
INSERT INTO SUPPLIER_INVOICE VALUES (13, 6, 'Cash', '22-FEB-2017', 1400,800, 600,
NULL, 8);

INSERT INTO SUPPLIER_INVOICE VALUES (14, 3, 'Cash', '22-FEB-2017', 2000, 1500,
500, '15-MAR-2017', 8);

INSERT INTO SUPPLIER_INVOICE VALUES (15, 2, 'Cash', '22-FEB-2017', 1000, 1000, 0,
NULL, 4);


INSERT INTO SUPPLIER_INVOICE VALUES (16, 6, 'Card', '25-JAN-2017', 900, 400, 500,
'15-FEB-2017', 2);
```



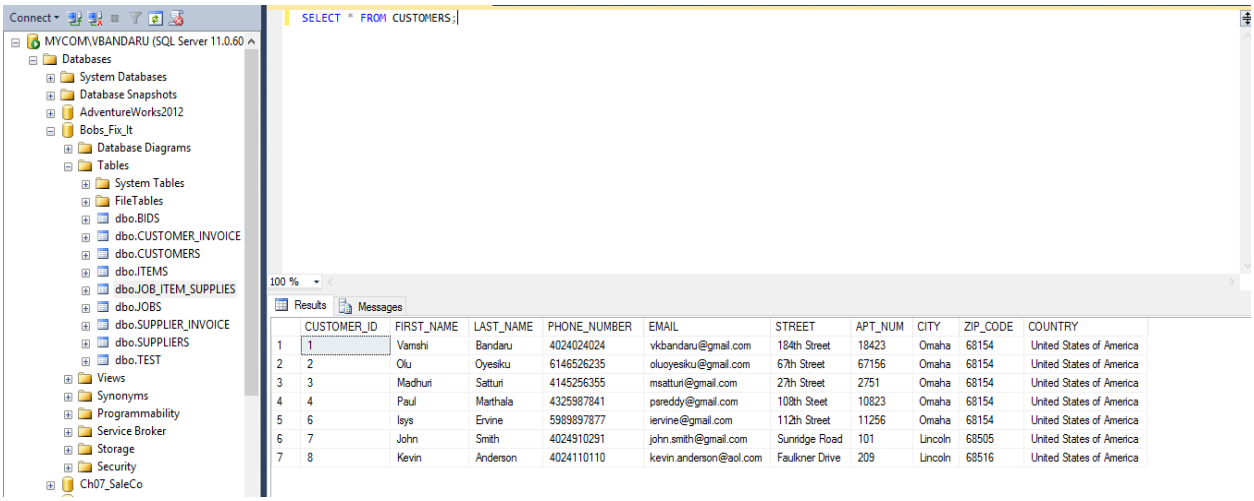
The below mentioned update statements are written to run the queries and get the desired output.

```
UPDATE JOB_ITEM_SUPPLIES SET DATE_REQUESTED = '22-JAN-2017'
UPDATE JOB_ITEM_SUPPLIES SET DATE_SHIPPED = '25-JAN-2017'
UPDATE JOB_ITEM_SUPPLIES SET DATE_DELIVERED = '27-JAN-2017'
UPDATE JOB_ITEM_SUPPLIES SET PAYMENT_DUE_DATE = DATEADD(DAY,30,DATE_DELIVERED) WHERE
DATE_DELIVERED IS NOT NULL;

UPDATE CUSTOMER_INVOICE SET PAYMENT_DUE_DATE = (SELECT DISTINCT PAYMENT_DUE_DATE FROM
JOB_ITEM_SUPPLIES WHERE JOB_ID = 1)
WHERE JOB_ID = 1

UPDATE SUPPLIER_INVOICE SET PAYMENT_DUE_DATE = (SELECT DISTINCT PAYMENT_DUE_DATE FROM
JOB_ITEM_SUPPLIES WHERE JOB_ID = 1)
WHERE JOB_ID = 1
```



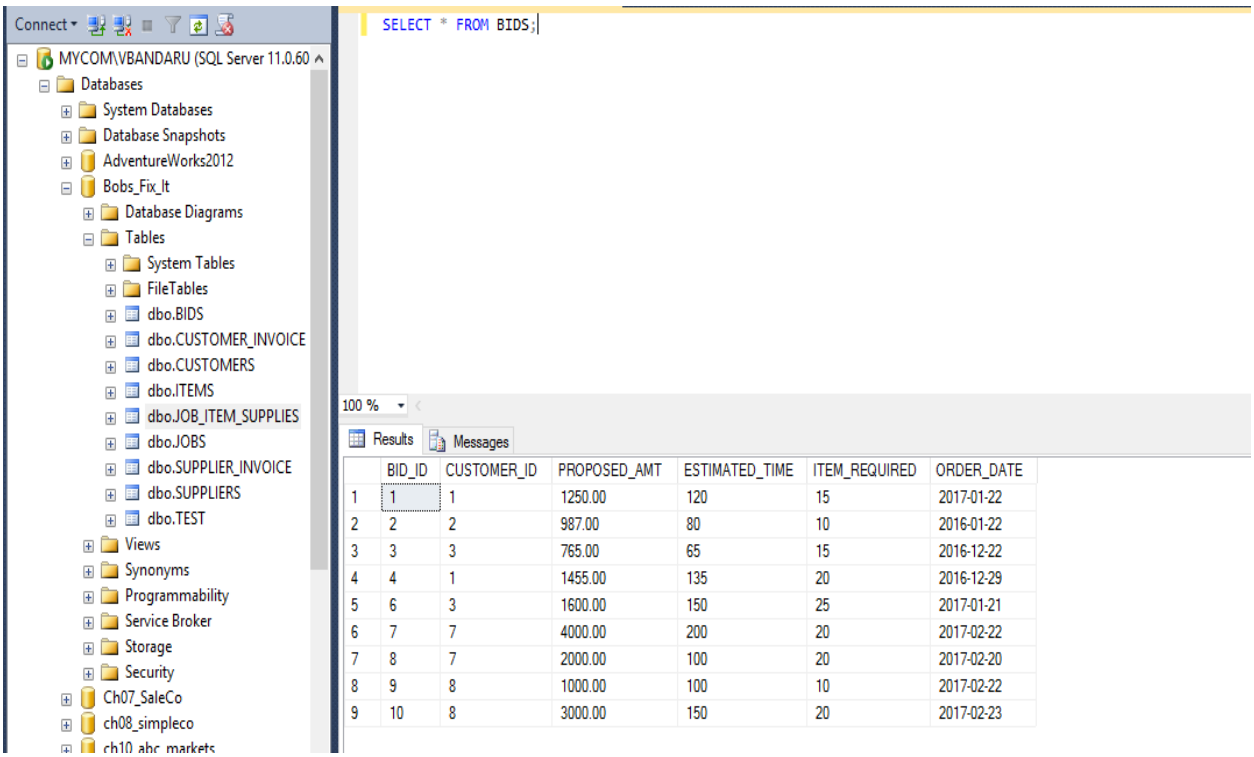
Sample Data Output for all tables:

CUSTOMERS:



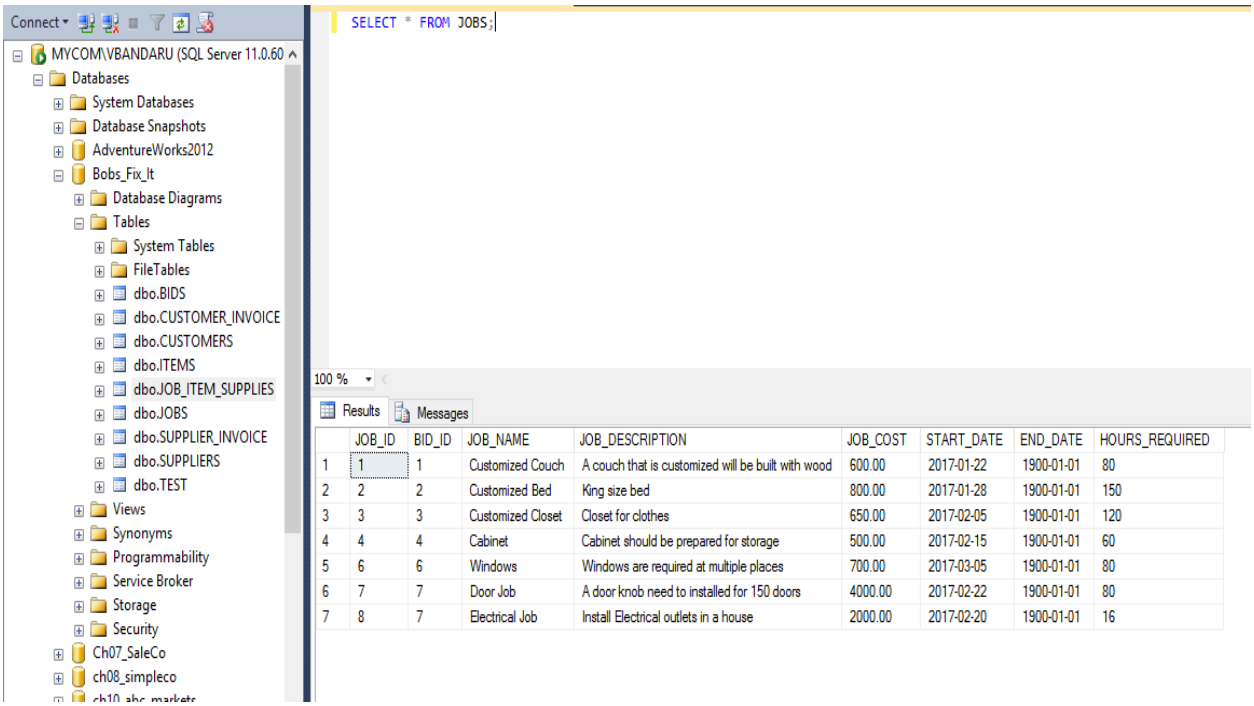
	CUSTOMER_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER	EMAIL	STREET	APT_NUM	CITY	ZIP_CODE	COUNTRY
1	1	Vamshi	Bandaru	4024024024	vkbandaru@gmail.com	184th Street	18423	Omaha	68154	United States of America
2	2	Olu	Oyesiku	6146526235	oluoyesiku@gmail.com	67th Street	67156	Omaha	68154	United States of America
3	3	Madhuri	Satturi	4145256355	msatturi@gmail.com	27th Street	2751	Omaha	68154	United States of America
4	4	Paul	Marthala	4325987841	psreddy@gmail.com	108th Steet	10823	Omaha	68154	United States of America
5	6	Isys	Ervine	5989897877	iervine@gmail.com	112th Street	11256	Omaha	68154	United States of America
6	7	John	Smith	4024910291	john.smith@gmail.com	Sunridge Road	101	Lincoln	68505	United States of America
7	8	Kevin	Anderson	4024110110	kevin.anderson@aol.com	Faulkner Drive	209	Lincoln	68516	United States of America

BIDS:



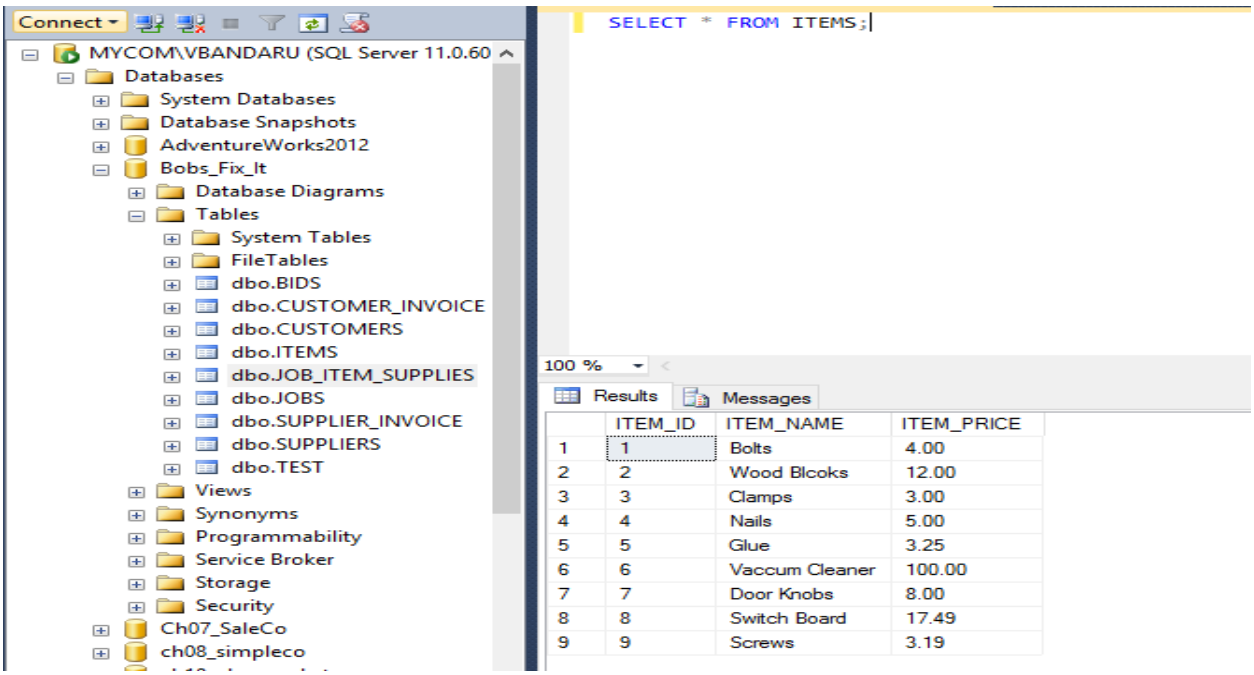
	BID_ID	CUSTOMER_ID	PROPOSED_AMT	ESTIMATED_TIME	ITEM_REQUIRED	ORDER_DATE
1	1	1	1250.00	120	15	2017-01-22
2	2	2	987.00	80	10	2016-01-22
3	3	3	765.00	65	15	2016-12-22
4	4	1	1455.00	135	20	2016-12-29
5	6	3	1600.00	150	25	2017-01-21
6	7	7	4000.00	200	20	2017-02-22
7	8	7	2000.00	100	20	2017-02-20
8	9	8	1000.00	100	10	2017-02-22
9	10	8	3000.00	150	20	2017-02-23

JOBS:



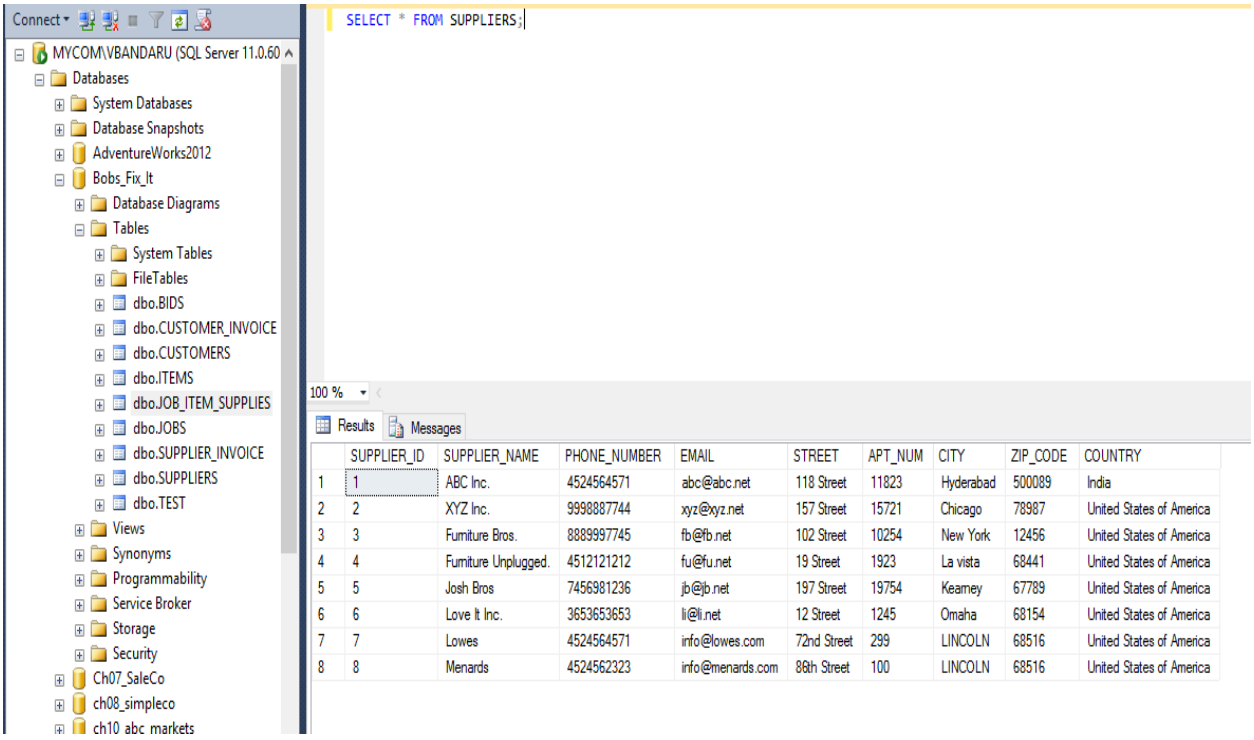
	JOB_ID	BID_ID	JOB_NAME	JOB_DESCRIPTION	JOB_COST	START_DATE	END_DATE	HOURS_REQUIRED
1	1	1	Customized Couch	A couch that is customized will be built with wood	600.00	2017-01-22	1900-01-01	80
2	2	2	Customized Bed	King size bed	800.00	2017-01-28	1900-01-01	150
3	3	3	Customized Closet	Closet for clothes	650.00	2017-02-05	1900-01-01	120
4	4	4	Cabinet	Cabinet should be prepared for storage	500.00	2017-02-15	1900-01-01	60
5	6	6	Windows	Windows are required at multiple places	700.00	2017-03-05	1900-01-01	80
6	7	7	Door Job	A door knob need to installed for 150 doors	4000.00	2017-02-22	1900-01-01	80
7	8	7	Electrical Job	Install Electrical outlets in a house	2000.00	2017-02-20	1900-01-01	16

ITEMS:



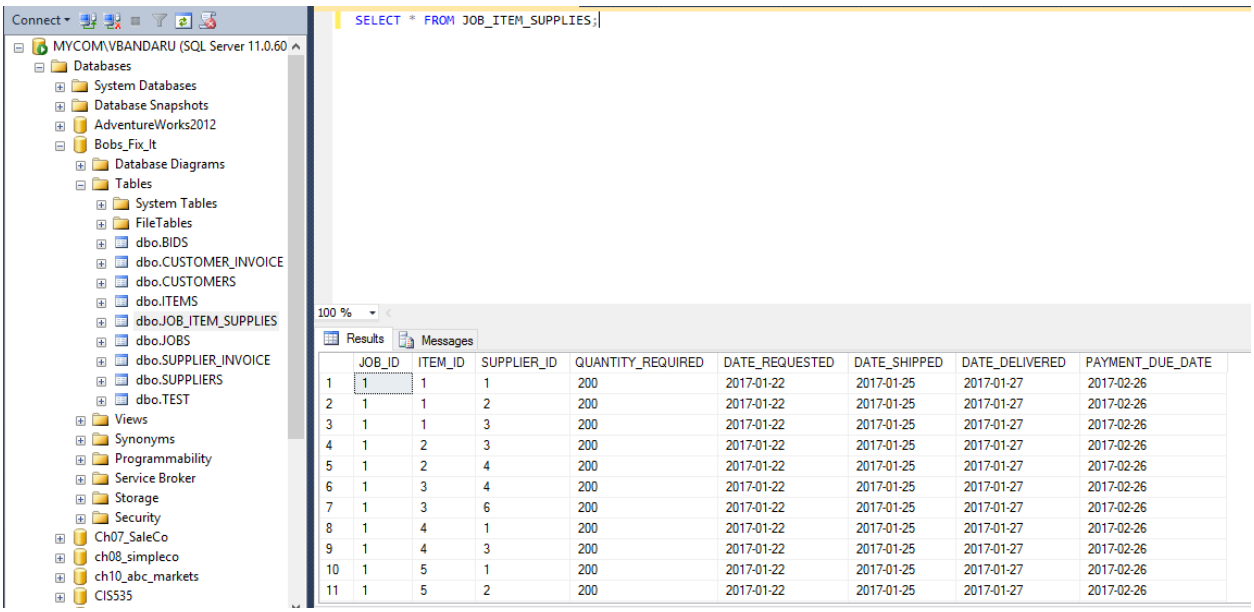
	ITEM_ID	ITEM_NAME	ITEM_PRICE
1	1	Bolts	4.00
2	2	Wood Blcoks	12.00
3	3	Clamps	3.00
4	4	Nails	5.00
5	5	Glue	3.25
6	6	Vaccum Cleaner	100.00
7	7	Door Knobs	8.00
8	8	Switch Board	17.49
9	9	Screws	3.19

SUPPLIERS:



	SUPPLIER_ID	SUPPLIER_NAME	PHONE_NUMBER	EMAIL	STREET	APT_NUM	CITY	ZIP_CODE	COUNTRY
1	1	ABC Inc.	4524564571	abc@abc.net	118 Street	11823	Hyderabad	500089	India
2	2	XYZ Inc.	9998887744	xyz@xyz.net	157 Street	15721	Chicago	78987	United States of America
3	3	Furniture Bros.	8889997745	fb@fb.net	102 Street	10254	New York	12456	United States of America
4	4	Furniture Unplugged.	4512121212	fu@fu.net	19 Street	1923	La vista	68441	United States of America
5	5	Josh Bros	7456981236	jb@jb.net	197 Street	19754	Kearney	67789	United States of America
6	6	Love It Inc.	3653653653	li@li.net	12 Street	1245	Omaha	68154	United States of America
7	7	Lowes	4524564571	info@lowes.com	72nd Street	299	LINCOLN	68516	United States of America
8	8	Menards	4524562323	info@menards.com	86th Street	100	LINCOLN	68516	United States of America

JOB_ITEM_SUPPLIES:

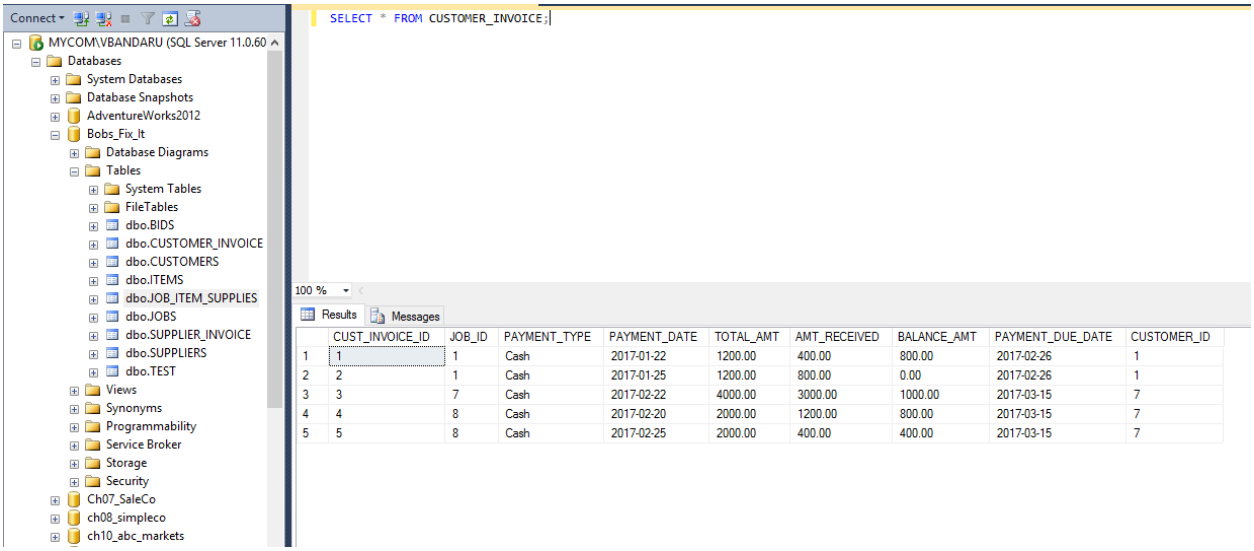


	JOB_ID	ITEM_ID	SUPPLIER_ID	QUANTITY_REQUIRED	DATE_REQUESTED	DATE_SHIPPED	DATE_DELIVERED	PAYMENT_DUE_DATE
1	1	1	1	200	2017-01-22	2017-01-25	2017-01-27	2017-02-26
2	1	1	2	200	2017-01-22	2017-01-25	2017-01-27	2017-02-26
3	1	1	3	200	2017-01-22	2017-01-25	2017-01-27	2017-02-26
4	1	2	3	200	2017-01-22	2017-01-25	2017-01-27	2017-02-26
5	1	2	4	200	2017-01-22	2017-01-25	2017-01-27	2017-02-26
6	1	3	4	200	2017-01-22	2017-01-25	2017-01-27	2017-02-26
7	1	3	6	200	2017-01-22	2017-01-25	2017-01-27	2017-02-26
8	1	4	1	200	2017-01-22	2017-01-25	2017-01-27	2017-02-26
9	1	4	3	200	2017-01-22	2017-01-25	2017-01-27	2017-02-26
10	1	5	1	200	2017-01-22	2017-01-25	2017-01-27	2017-02-26
11	1	5	2	200	2017-01-22	2017-01-25	2017-01-27	2017-02-26

CUSTOMER_INVOICE:

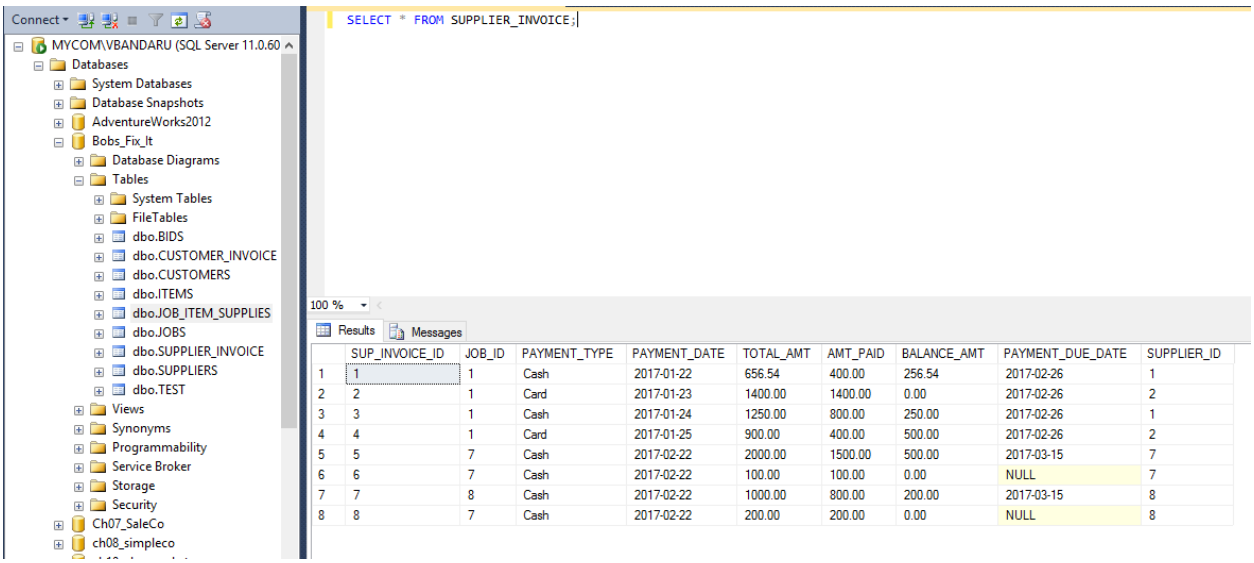
There were only 4 rows inserted into this table. Hence we can use the below insert statement to insert one more row

```
INSERT INTO CUSTOMER_INVOICE VALUES (5, 8, 'Cash', '25-FEB-2017', 2000, 400, 400, '15-MAR-2017', 7);
```



	CUST_INVOICE_ID	JOB_ID	PAYMENT_TYPE	PAYMENT_DATE	TOTAL_AMT	AMT_RECEIVED	BALANCE_AMT	PAYMENT_DUE_DATE	CUSTOMER_ID
1	1	1	Cash	2017-01-22	1200.00	400.00	800.00	2017-02-26	1
2	2	1	Cash	2017-01-25	1200.00	800.00	0.00	2017-02-26	1
3	3	7	Cash	2017-02-22	4000.00	3000.00	1000.00	2017-03-15	7
4	4	8	Cash	2017-02-20	2000.00	1200.00	800.00	2017-03-15	7
5	5	8	Cash	2017-02-25	2000.00	400.00	400.00	2017-03-15	7

SUPPLIER_INVOICE:



	SUP_INVOICE_ID	JOB_ID	PAYMENT_TYPE	PAYMENT_DATE	TOTAL_AMT	AMT_PAID	BALANCE_AMT	PAYMENT_DUE_DATE	SUPPLIER_ID
1	1	1	Cash	2017-01-22	656.54	400.00	256.54	2017-02-26	1
2	2	1	Card	2017-01-23	1400.00	1400.00	0.00	2017-02-26	2
3	3	1	Cash	2017-01-24	1250.00	800.00	250.00	2017-02-26	1
4	4	1	Card	2017-01-25	900.00	400.00	500.00	2017-02-26	2
5	5	7	Cash	2017-02-22	2000.00	1500.00	500.00	2017-03-15	7
6	6	7	Cash	2017-02-22	100.00	100.00	0.00	NULL	7
7	7	8	Cash	2017-02-22	1000.00	800.00	200.00	2017-03-15	8
8	8	7	Cash	2017-02-22	200.00	200.00	0.00	NULL	8

Sample Queries:

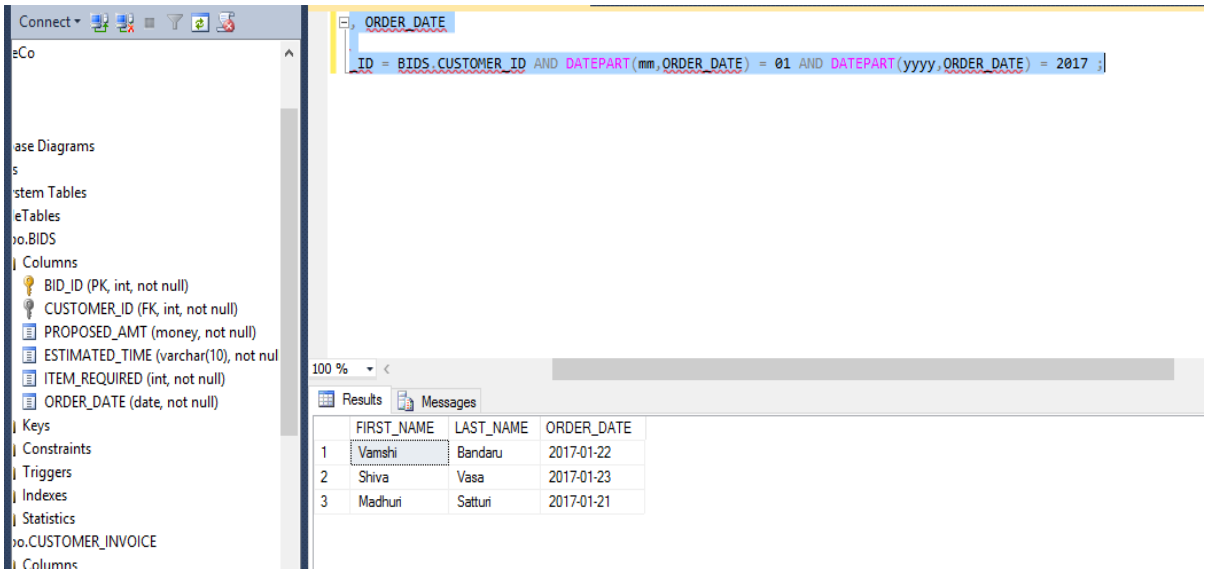
The below query displays, names of the customers who were provided bids last month.

Query:

```
SELECT FIRST_NAME, LAST_NAME, ORDER_DATE
FROM dbo.CUSTOMERS, dbo.BIDS
WHERE dbo.CUSTOMERS.CUSTOMER_ID = BIDS.CUSTOMER_ID AND DATEPART(mm,ORDER_DATE) = 01
AND DATEPART(yyyy,ORDER_DATE) = 2017 ;
```

Output:

```
FIRST_NAME                     LAST_NAME                      ORDER_DATE
------------------------------ ------------------------------ ----------
Vamshi                         Bandaru                        2017-01-22
Shiva                          Vasa                           2017-01-23
Madhuri                        Satturi                        2017-01-21

          (3  row(s) affected)
```



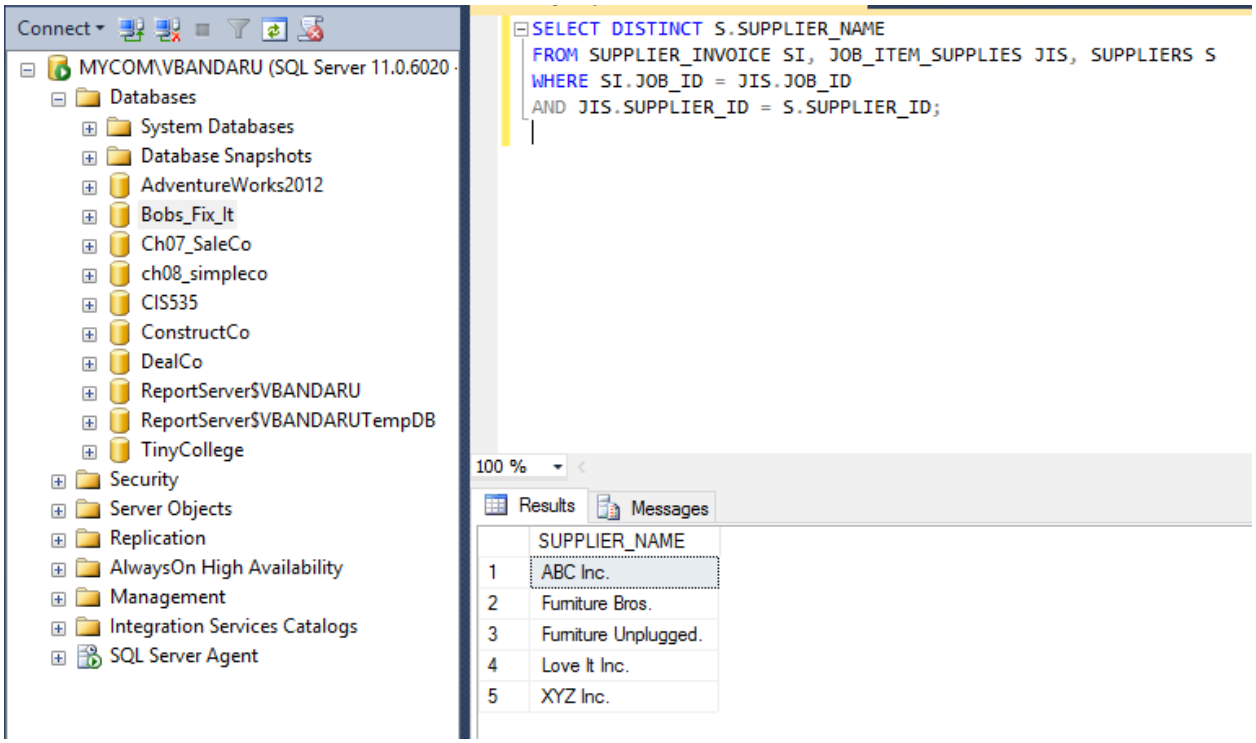
Query displaying unique names of Bob's suppliers.

Query:

```
SELECT DISTINCT S.SUPPLIER_NAME
FROM SUPPLIER_INVOICE SI, JOB_ITEM_SUPPLIES JIS, SUPPLIERS S
WHERE SI.JOB_ID = JIS.JOB_ID
AND JIS.SUPPLIER_ID = S.SUPPLIER_ID;
```

Output:

```
SUPPLIER_NAME
-------------------------
ABC Inc.
Furniture Bros.
Furniture Unplugged.
Love It Inc.
XYZ Inc.

(5 row(s) affected)
```



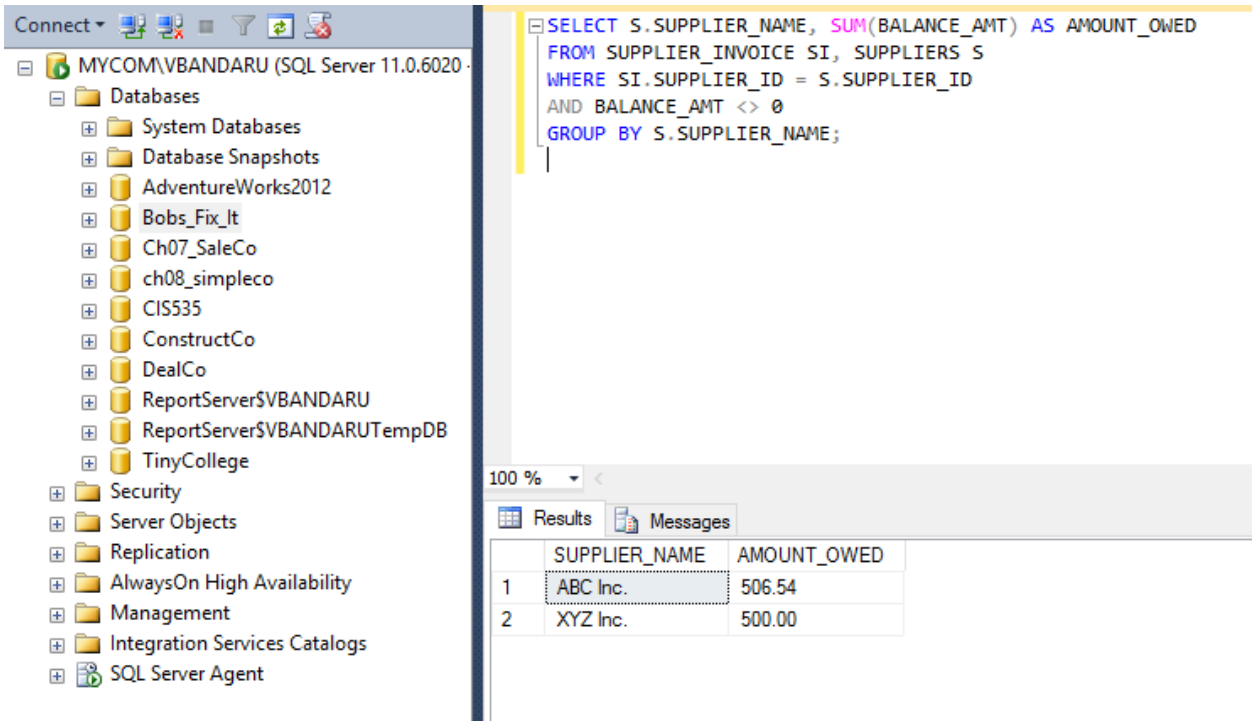
Below query displays the names of the suppliers and the total amount owed to any unpaid suppliers. Then write a separate query to display the days past due for each supplier.

Query:

```
SELECT S.SUPPLIER_NAME, SUM(BALANCE_AMT) AS AMOUNT_OWED
FROM SUPPLIER_INVOICE SI, SUPPLIERS S
WHERE SI.SUPPLIER_ID = S.SUPPLIER_ID
AND BALANCE_AMT <> 0
GROUP BY S.SUPPLIER_NAME;
```

Output:

```
SUPPLIER_NAME              AMOUNT_OWED
------------------------- --------------------
ABC Inc.                   506.54
XYZ Inc.                   500.00

(2 row(s) affected)
```

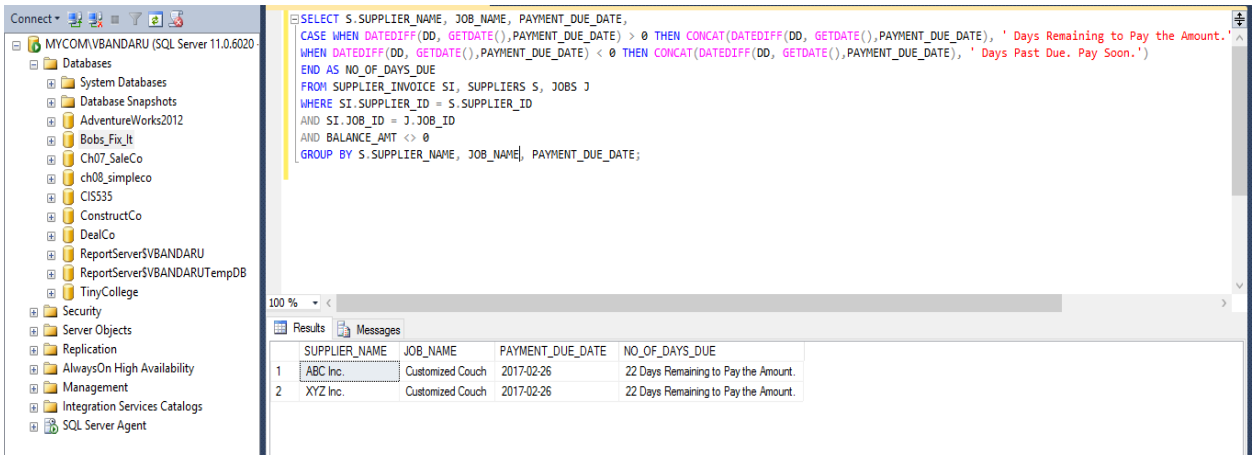


Query:

```
SELECT S.SUPPLIER_NAME, JOB_NAME, PAYMENT_DUE_DATE,
CASE WHEN DATEDIFF(DD, GETDATE(),PAYMENT_DUE_DATE) > 0 THEN CONCAT(DATEDIFF(DD,
GETDATE(),PAYMENT_DUE_DATE), ' Days Remaining to Pay the Amount.')
WHEN DATEDIFF(DD, GETDATE(),PAYMENT_DUE_DATE) < 0 THEN CONCAT(DATEDIFF(DD,
GETDATE(),PAYMENT_DUE_DATE), ' Days Past Due. Pay Soon.')
END AS NO_OF_DAYS_DUE
FROM SUPPLIER_INVOICE SI, SUPPLIERS S, JOBS J
WHERE SI.SUPPLIER_ID = S.SUPPLIER_ID
AND SI.JOB_ID = J.JOB_ID
AND BALANCE_AMT <> 0
GROUP BY S.SUPPLIER_NAME, JOB_NAME, PAYMENT_DUE_DATE;
```

Output:

```
SUPPLIER_NAME              JOB_NAME                   PAYMENT_DUE_DATE NO_OF_DAYS_DUE
-------------------------- -------------------------- ---------------- ---------------------
-------------------------
ABC Inc.                   Customized Couch           2017-02-26       22 Days Remaining to
Pay the Amount.
XYZ Inc.                   Customized Couch           2017-02-26       22 Days Remaining to
Pay the Amount.

               (2  row(s) affected)
```



Below are delete statement(s) to delete a particular customer from the customer table (make sure to delete any corresponding rows in other tables).

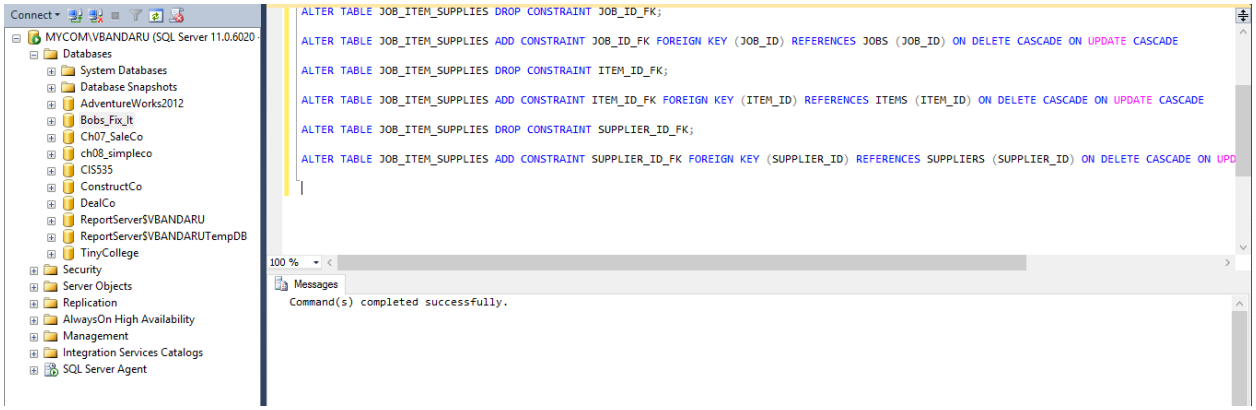
Query:

First, we need to alter the foreign key constraints to add the DELETE CASCADE statement which will ensure that the rows from the child tables will be deleted when a particular row in the parent table (which is being accessed in the child tables) is deleted.

```
ALTER TABLE BIDS DROP CONSTRAINT BID_CUST_ID_FK;

ALTER TABLE BIDS ADD CONSTRAINT BID_CUST_ID_FK FOREIGN KEY (CUSTOMER_ID) REFERENCES
CUSTOMERS (CUSTOMER_ID) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE JOBS DROP CONSTRAINT JOB_BID_ID_FK;

ALTER TABLE JOBS ADD CONSTRAINT JOB_BID_ID_FK FOREIGN KEY (BID_ID) REFERENCES BIDS
(BID_ID) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE JOB_ITEM_SUPPLIES DROP CONSTRAINT JOB_ID_FK;

ALTER TABLE JOB_ITEM_SUPPLIES ADD CONSTRAINT JOB_ID_FK FOREIGN KEY (JOB_ID) REFERENCES
JOBS (JOB_ID) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE JOB_ITEM_SUPPLIES DROP CONSTRAINT ITEM_ID_FK;

ALTER TABLE JOB_ITEM_SUPPLIES ADD CONSTRAINT ITEM_ID_FK FOREIGN KEY (ITEM_ID)
REFERENCES ITEMS (ITEM_ID) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE JOB_ITEM_SUPPLIES DROP CONSTRAINT SUPPLIER_ID_FK;
```

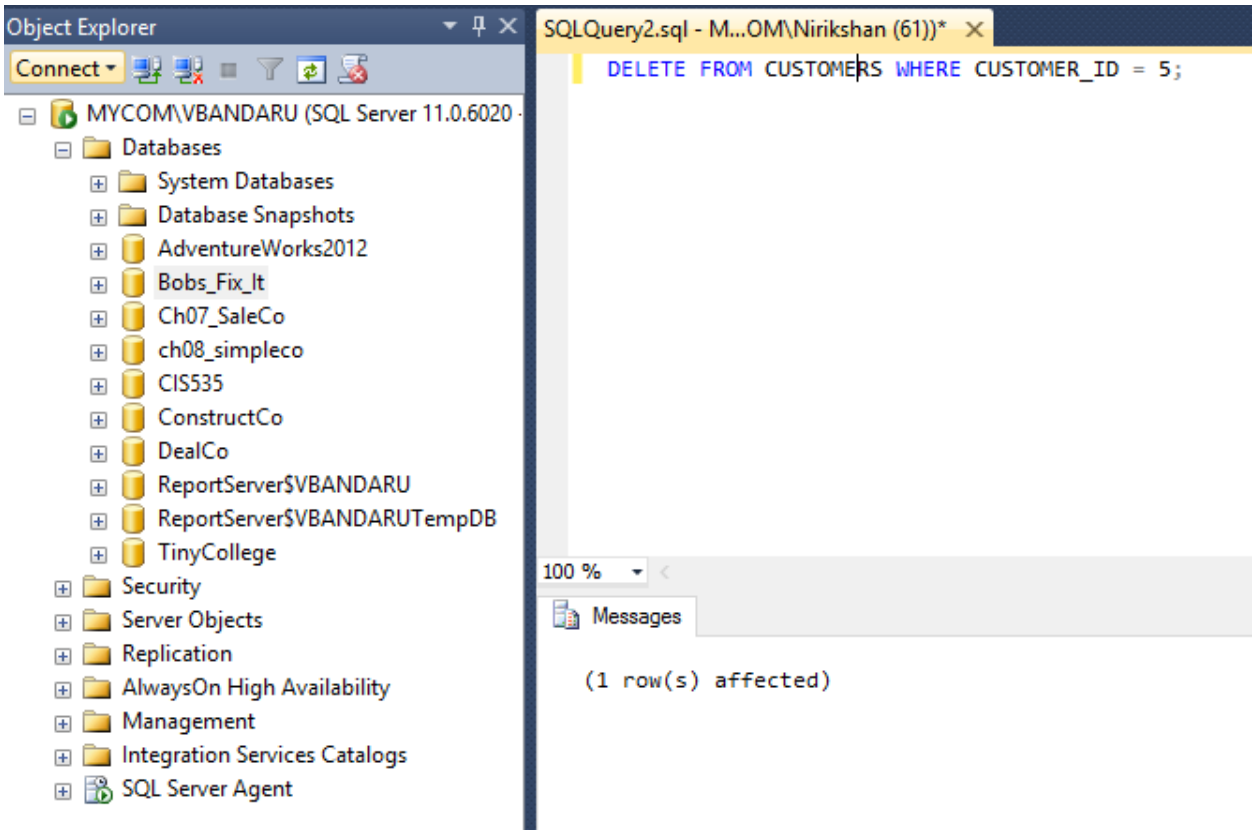
```
ALTER TABLE JOB_ITEM_SUPPLIES ADD CONSTRAINT SUPPLIER_ID_FK FOREIGN KEY (SUPPLIER_ID)
REFERENCES SUPPLIERS (SUPPLIER_ID) ON DELETE CASCADE ON UPDATE CASCADE
```

```
Command(s) completed successfully.
```



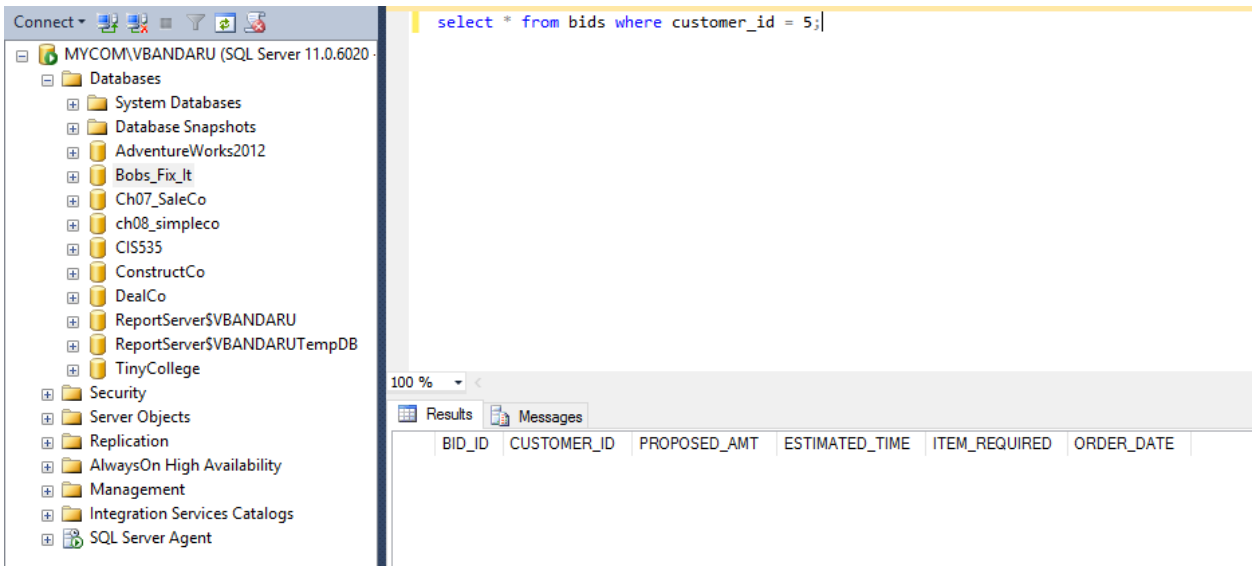

```
DELETE FROM CUSTOMERS WHERE CUSTOMER_ID = 5;
```

```
(1 row(s) affected)
```

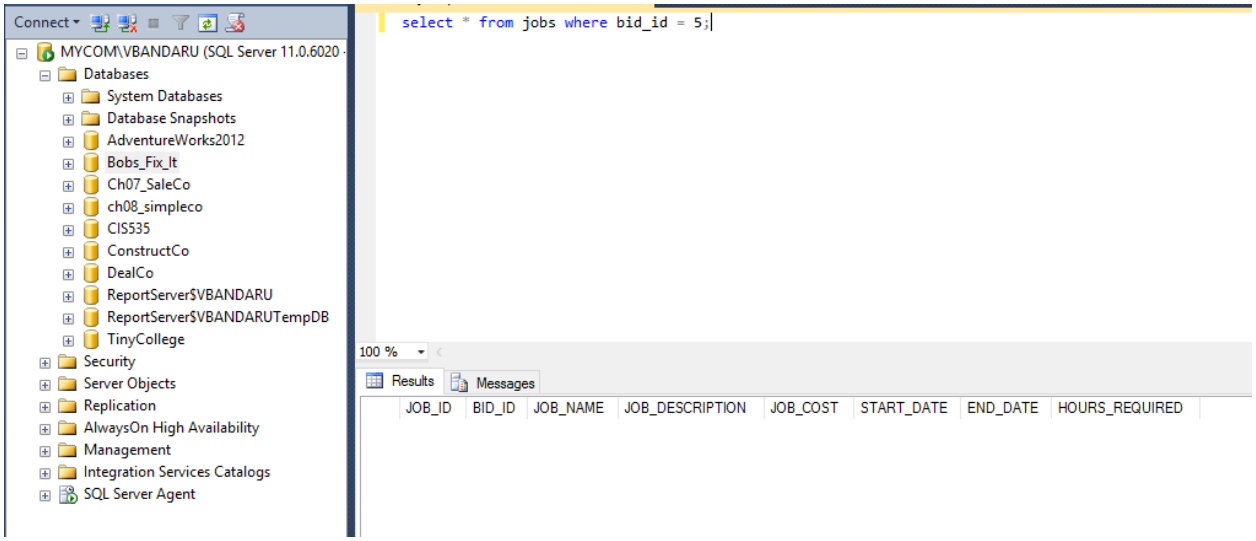



```
select * from bids where customer_id = 5;
```

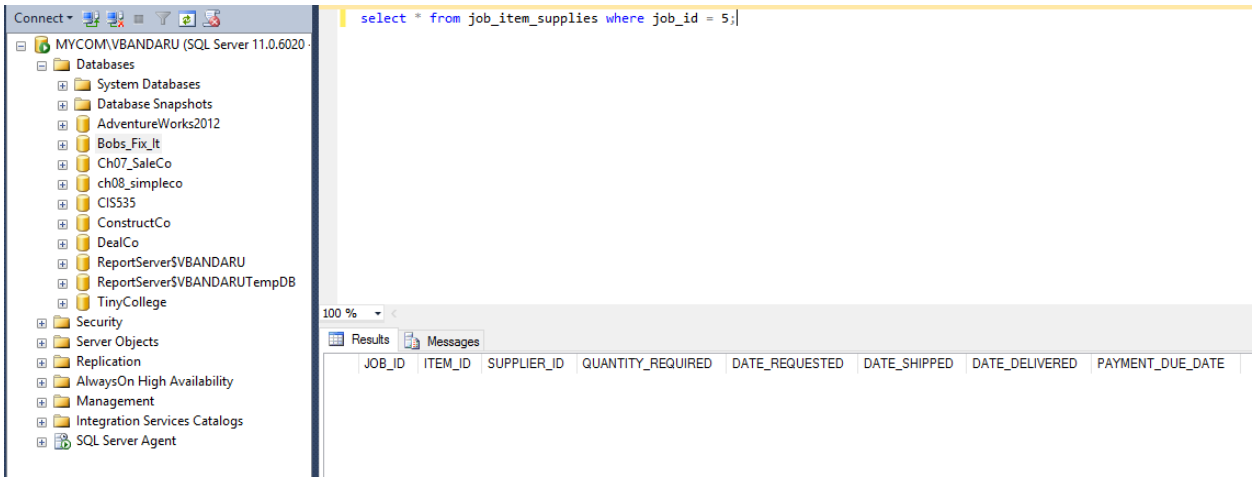
```
BID_ID      CUSTOMER_ID PROPOSED_AMT          ESTIMATED_TIME ITEM_REQUIRED ORDER_DATE
----------- ----------- --------------------- -------------- ------------- ----------

(0 row(s) affected)
```




```
select * from jobs where bid_id = 5;

JOB_ID      BID_ID      JOB_NAME                  JOB_DESCRIPTION
JOB_COST              START_DATE END_DATE   HOURS_REQUIRED
----------- ----------- ------------------------- ---------------------------------------
-----------------------------------------------------------------------------------------
------------------- --------------------- ---------- ---------- --------------

(0 row(s) affected)
```

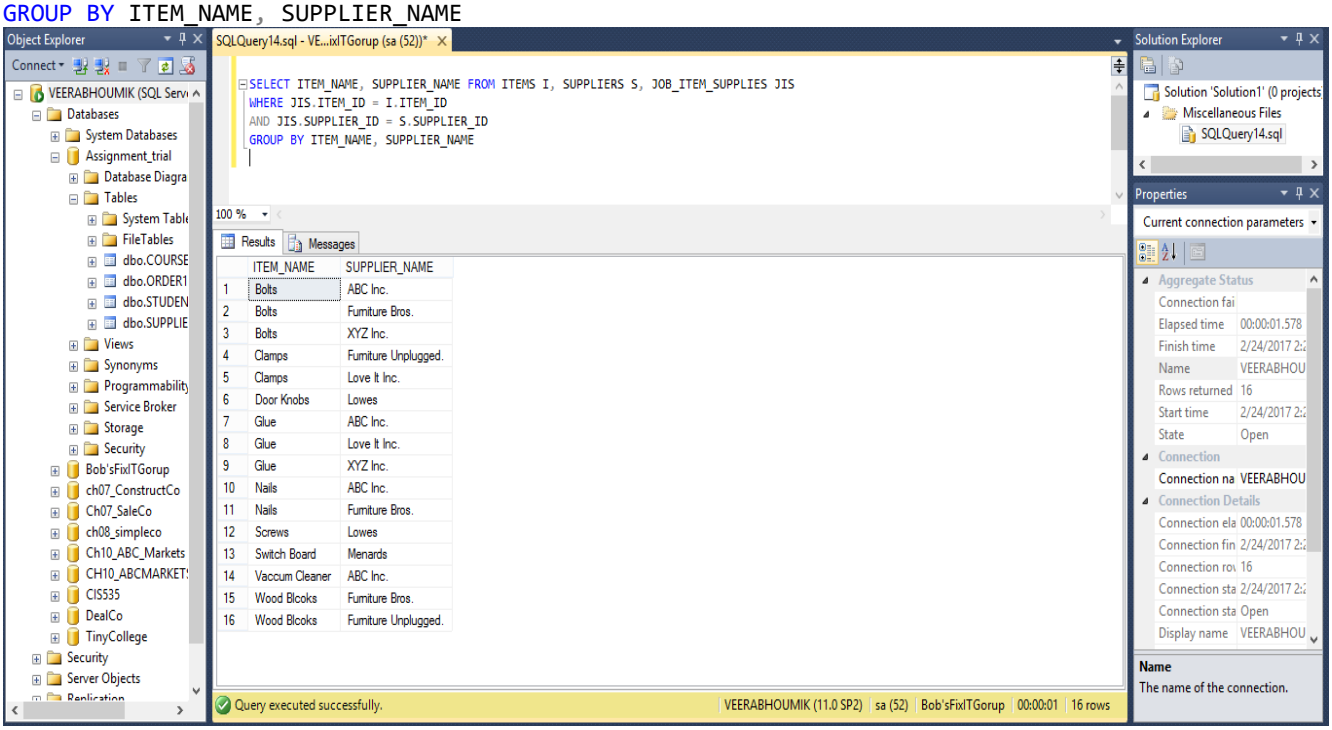


```
select * from job_item_supplies where job_id = 5;

JOB_ID      ITEM_ID     SUPPLIER_ID QUANTITY_REQUIRED DATE_REQUESTED DATE_SHIPPED
DATE_DELIVERED PAYMENT_DUE_DATE
----------- ----------- ----------- ---------------- -------------- ------------ --------
------ ----------------

(0 row(s) affected)
```



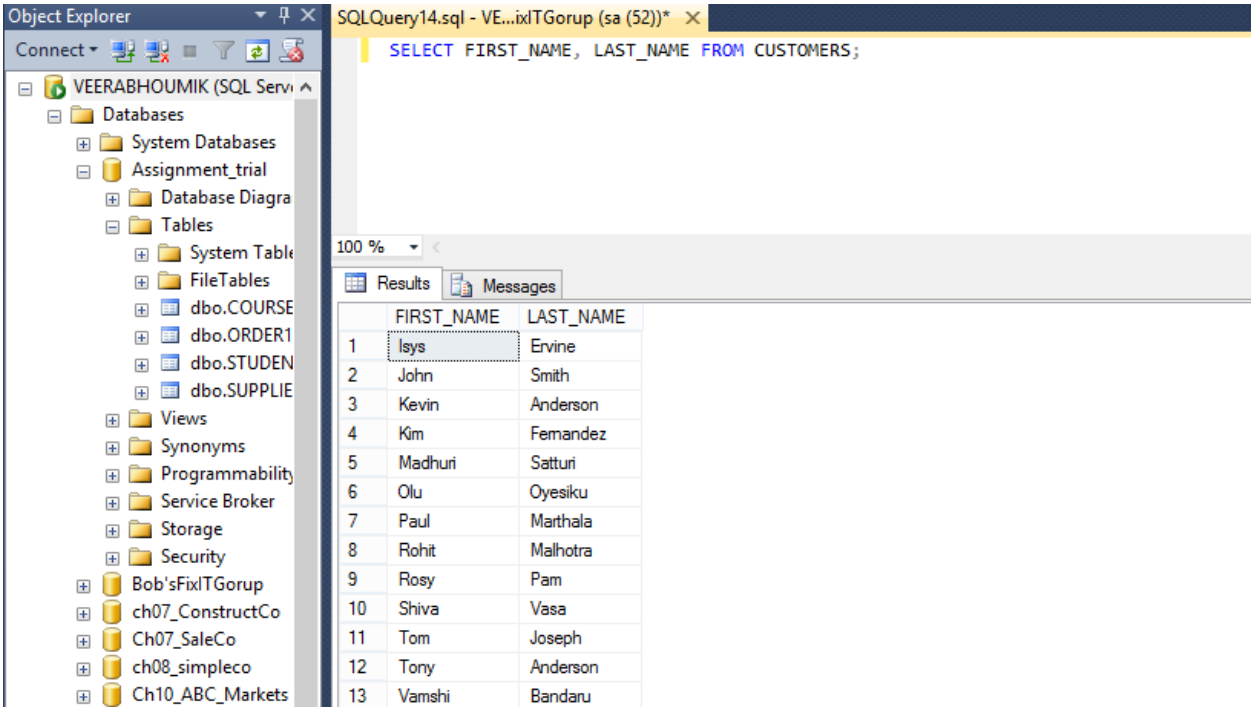
Display the items provided by the suppliers for all the jobs handled by Bob

```
SELECT ITEM_NAME, SUPPLIER_NAME FROM ITEMS I, SUPPLIERS S, JOB_ITEM_SUPPLIES JIS
WHERE JIS.ITEM_ID = I.ITEM_ID
AND JIS.SUPPLIER_ID = S.SUPPLIER_ID
GROUP BY ITEM_NAME, SUPPLIER_NAME
```



Display the names of Bob's customers

```
SELECT FIRST_NAME, LAST_NAME FROM CUSTOMERS;
```

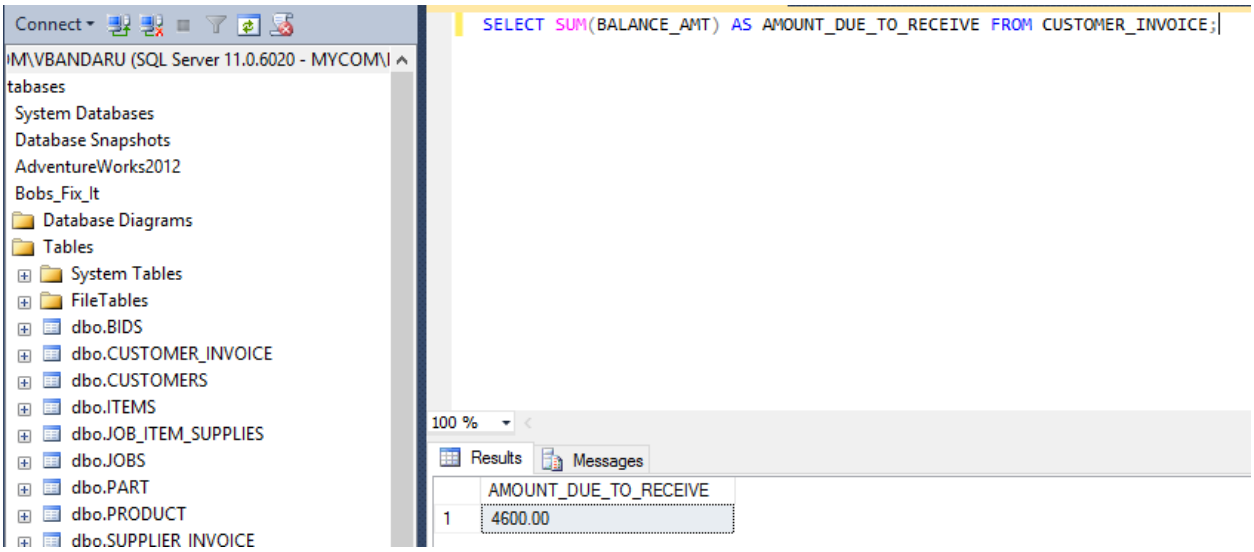


Display the total amount earned by Bob for all the jobs

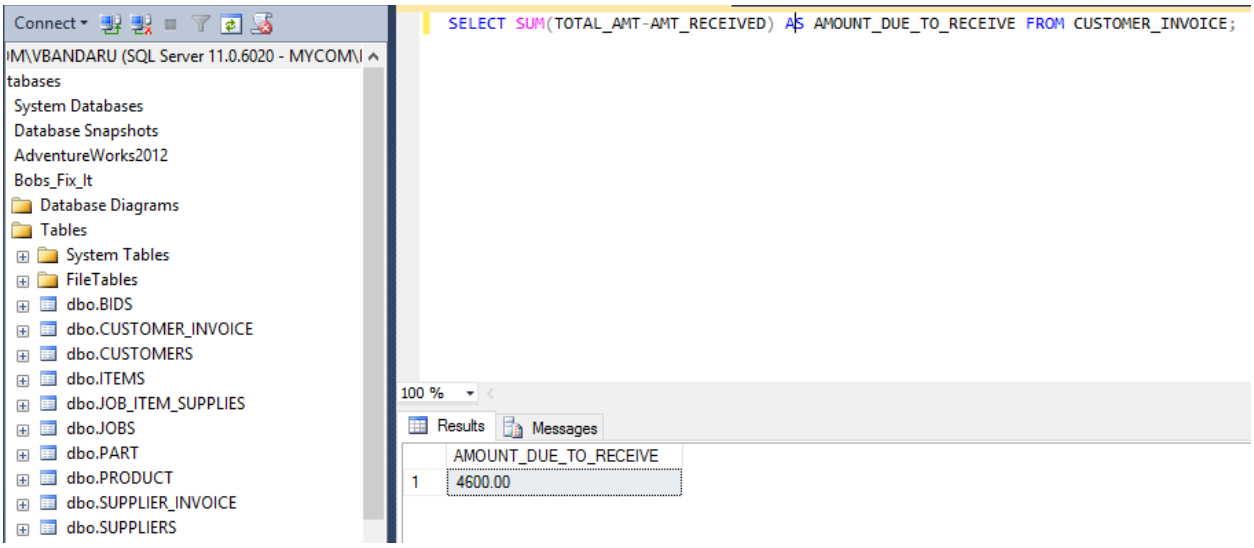
```
SELECT SUM(AMT_RECEIVED) AS AMOUNT_EARNED FROM CUSTOMER_INVOICE;
```

Display the total amount Bob should receive from the customers

```
SELECT SUM (BALANCE_AMT) AS AMOUNT_DUE_TO_RECEIVE FROM CUSTOMER_INVOICE;
```

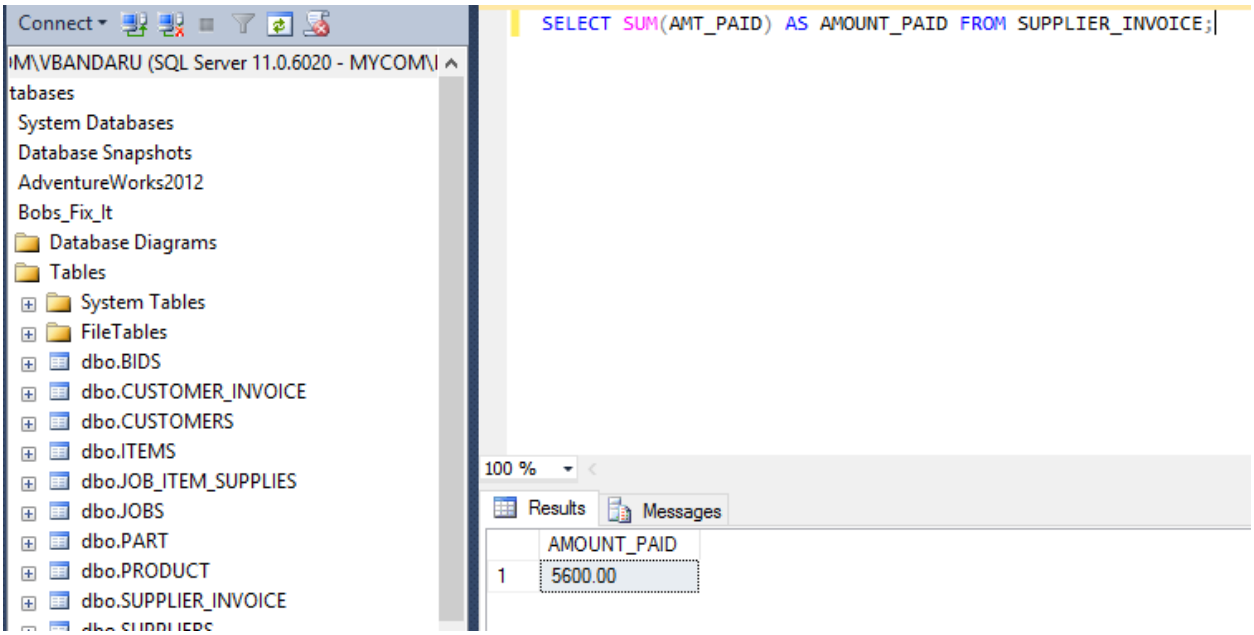



```
SELECT SUM (TOTAL_AMT-AMT_RECEIVED) AS AMOUNT_DUE_TO_RECEIVE FROM
CUSTOMER_INVOICE;
```



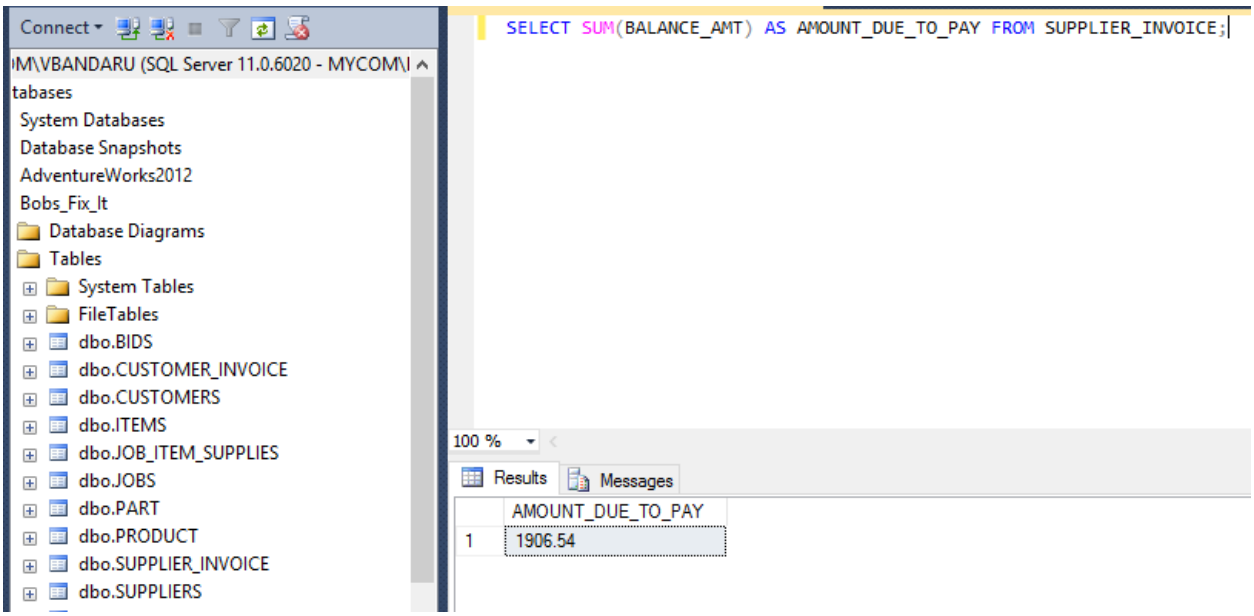
Display the total amount paid by Bob to suppliers

```
SELECT SUM (AMT_PAID) AS AMOUNT_PAID FROM SUPPLIER_INVOICE;
```

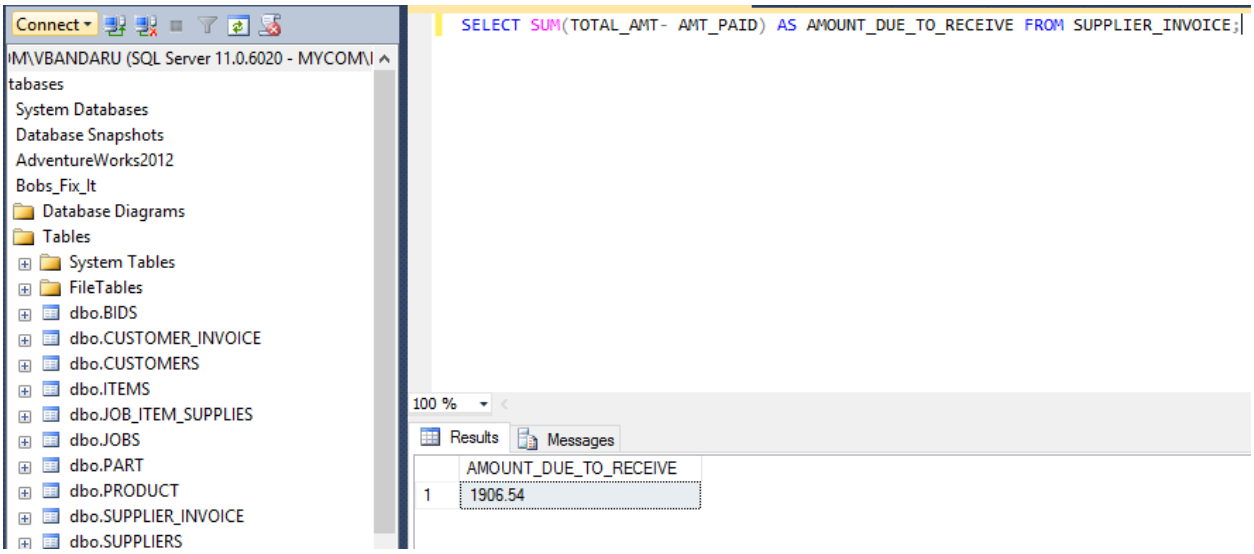


Display the total amount due by Bob to suppliers

```
SELECT SUM (BALANCE_AMT) AS AMOUNT_DUE_TO_PAY FROM SUPPLIER_INVOICE;
```

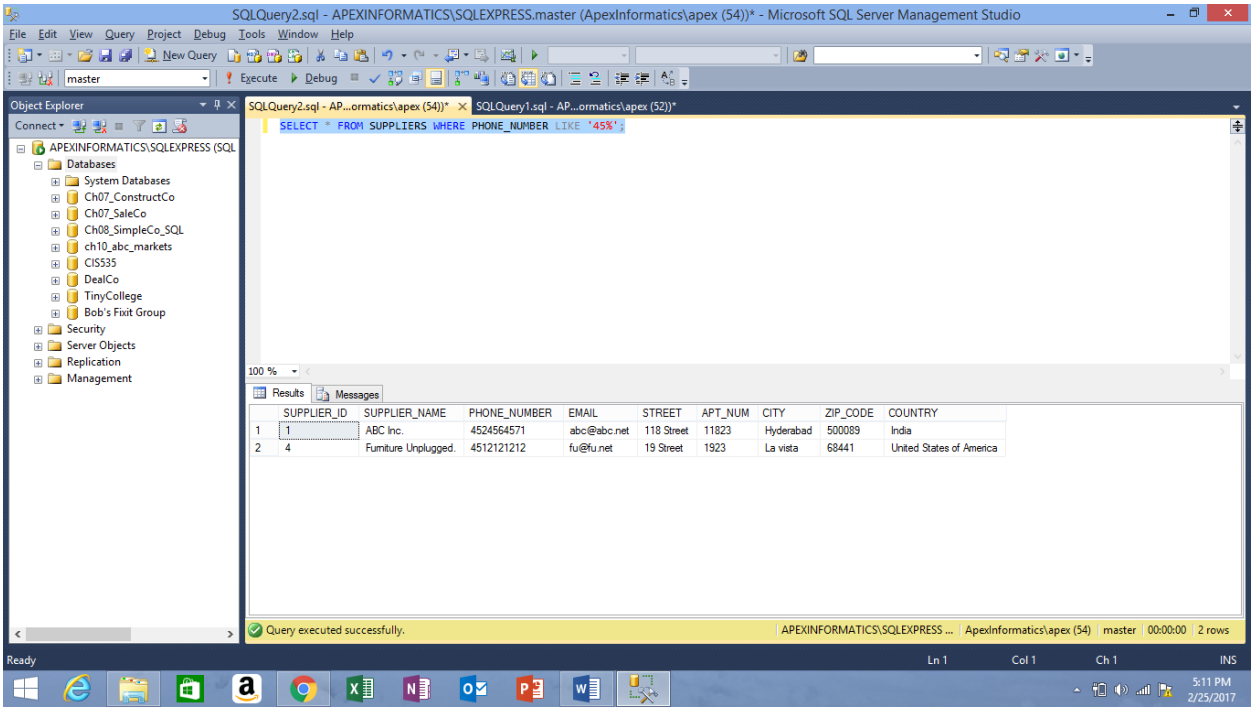



```
SELECT SUM (TOTAL_AMT- AMT_PAID) AS AMOUNT_DUE_TO_RECEIVE FROM SUPPLIER_INVOICE;
```



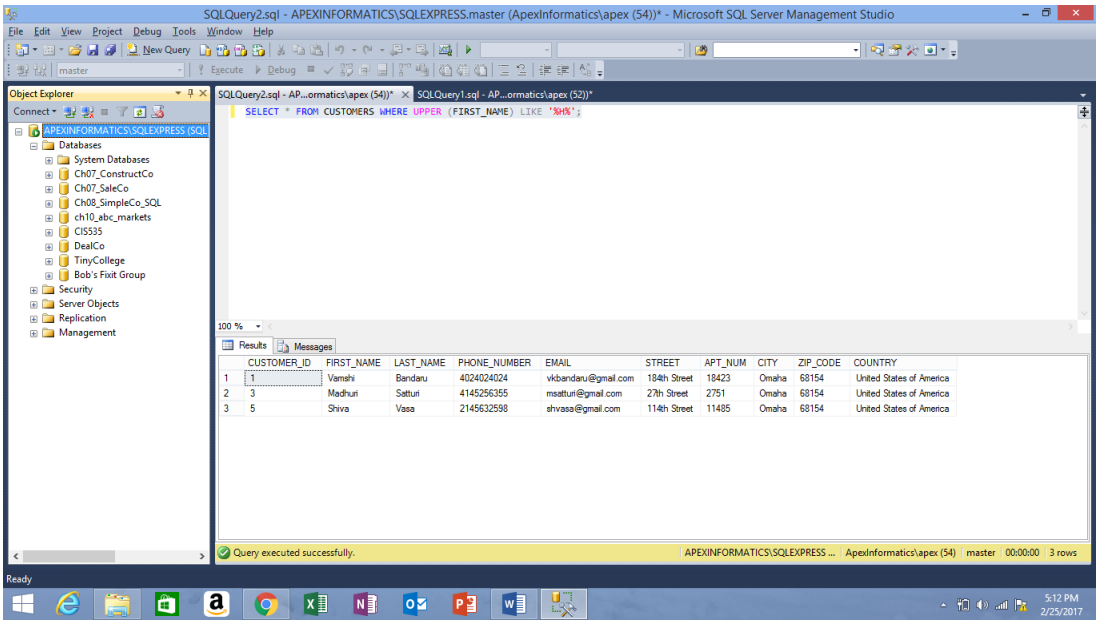
Display the suppliers whose phone number starts with a prefix 45

```
SELECT * FROM SUPPLIERS WHERE PHONE_NUMBER LIKE '45%';
```



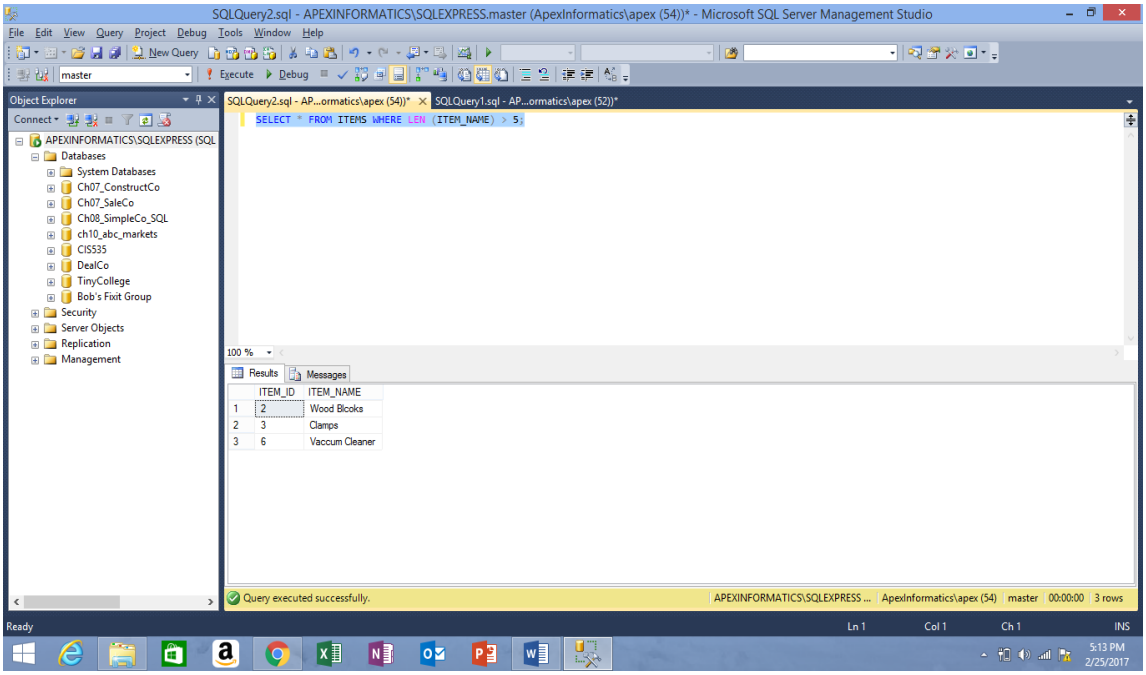
Display the customers whose first name has 'h' in upper or lower case

```
SELECT * FROM CUSTOMERS WHERE UPPER (FIRST_NAME) LIKE '%H%';
```



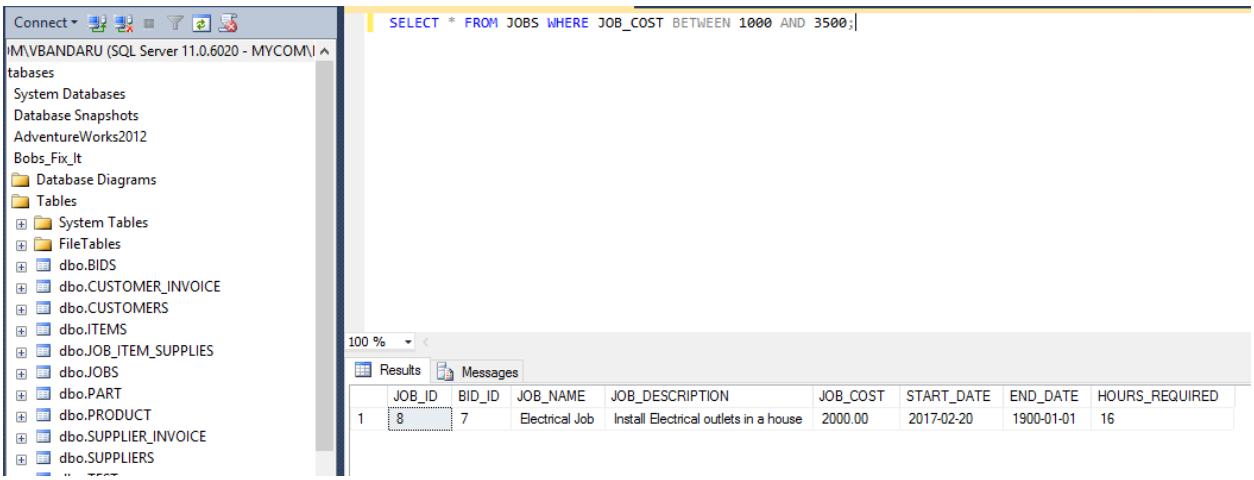
Display the items with the names having more than 5 characters

```
SELECT * FROM ITEMS WHERE LEN (ITEM_NAME) > 5;
```



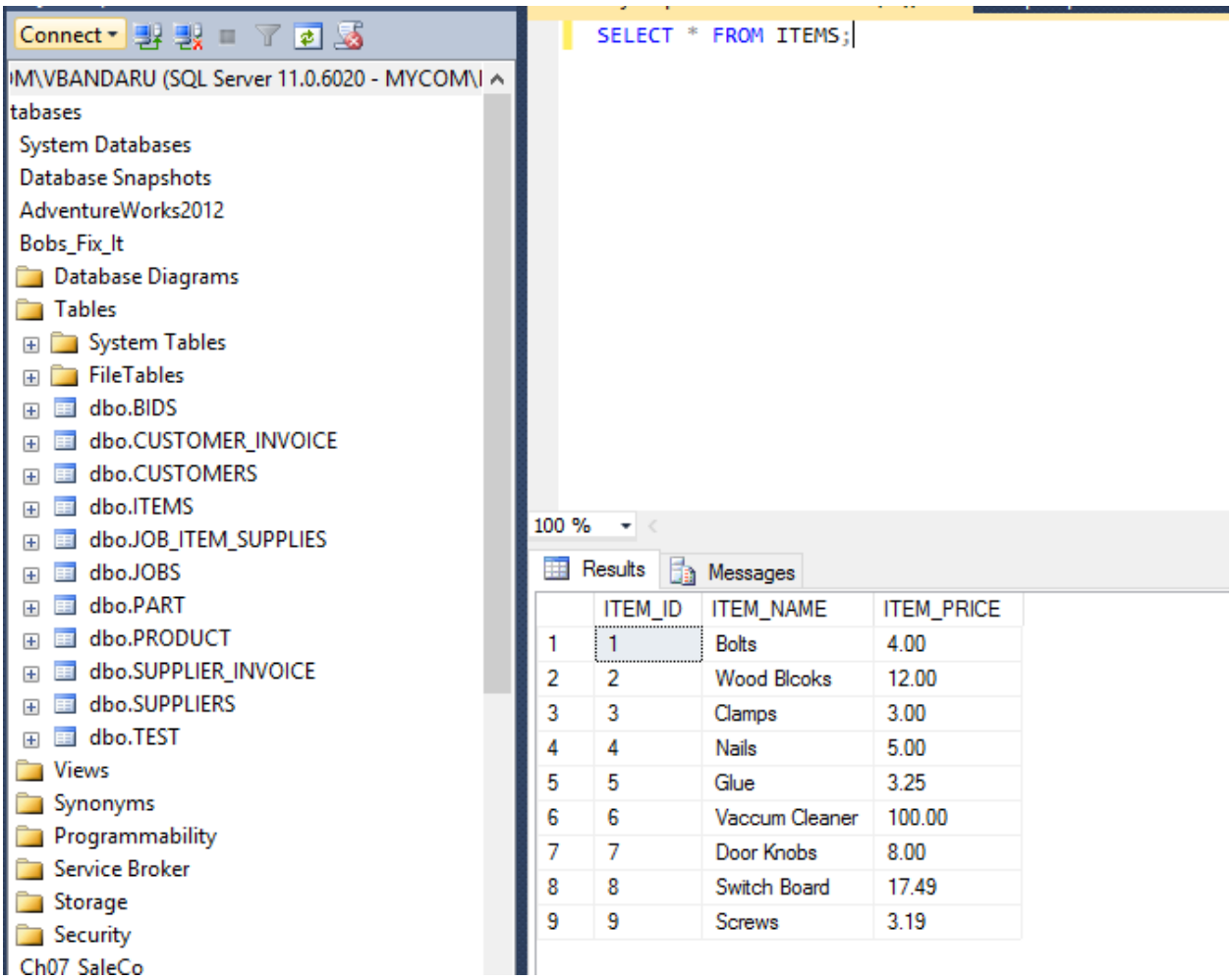
Display the information related to the jobs handled by Bob which costs between $1000 - $3500

```
SELECT * FROM JOBS WHERE JOB_COST BETWEEN 1000 AND 3500;
```



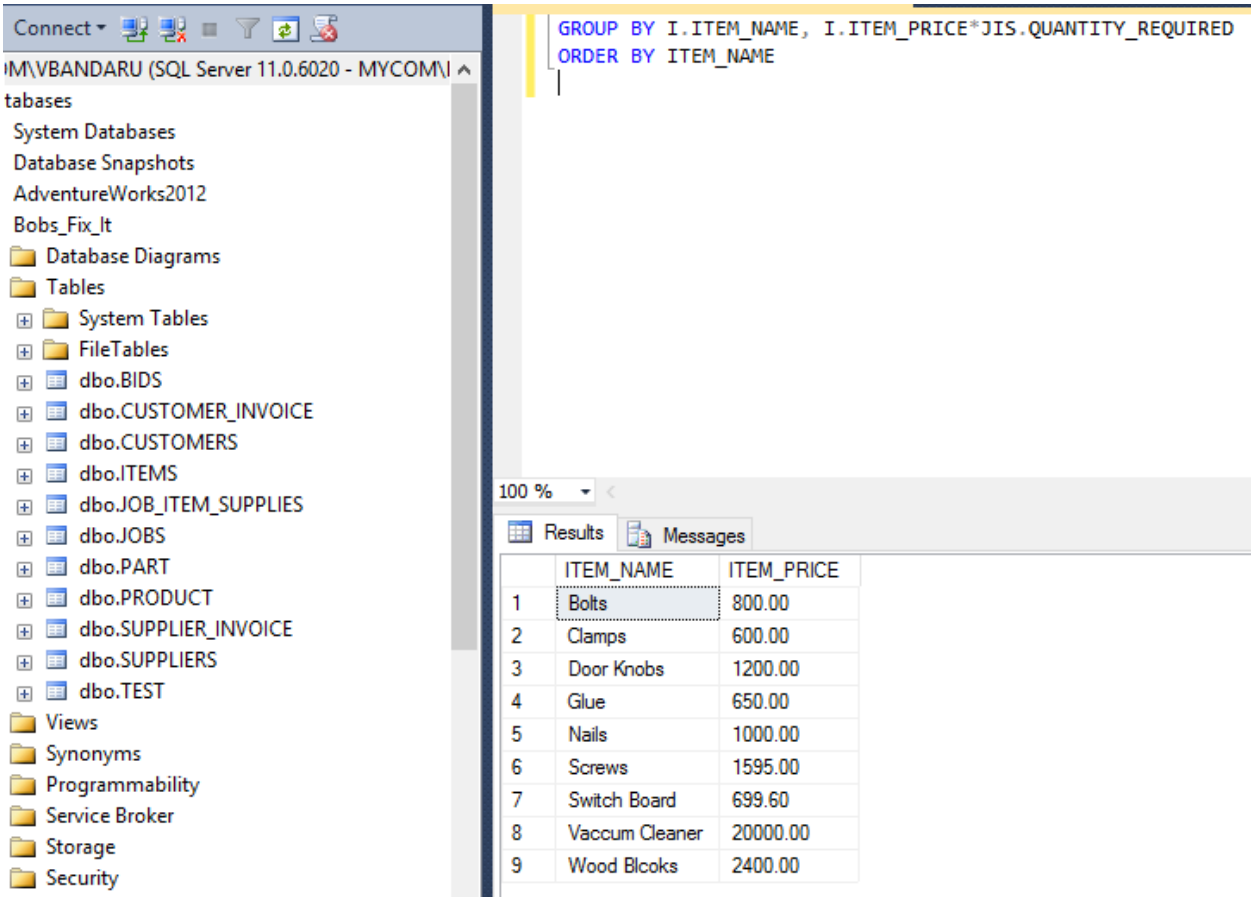
Display the items acquired by Bob for the jobs

```
SELECT * FROM ITEMS;
```



Display the cost of all items required for all jobs sorted by item name

```
SELECT I.ITEM_NAME, I.ITEM_PRICE*JIS.QUANTITY_REQUIRED AS ITEM_PRICE
FROM ITEMS I, JOB_ITEM_SUPPLIES JIS
WHERE JIS.ITEM_ID = I.ITEM_ID
GROUP BY I.ITEM_NAME, I.ITEM_PRICE*JIS.QUANTITY_REQUIRED
ORDER BY ITEM_NAME
```


Connect
M\VBANDARU (SQL Server 11.0.6020 - MYCOM\I
tabases
System Databases
Database Snapshots
AdventureWorks2012
Bobs_Fix_It
Database Diagrams
Tables
System Tables
FileTables
dbo.BIDS
dbo.CUSTOMER_INVOICE
dbo.CUSTOMERS
dbo.ITEMS
dbo.JOB_ITEM_SUPPLIES
dbo.JOBS
dbo.PART
dbo.PRODUCT
dbo.SUPPLIER_INVOICE
dbo.SUPPLIERS
dbo.TEST
Views
Synonyms
Programmability
Service Broker
Storage
Security
GROUP BY I.ITEM_NAME, I.ITEM_PRICE*JIS.QUANTITY_REQUIRED
ORDER BY ITEM_NAME
100 %
Results
Messages
ITEM_NAME ITEM_PRICE
1 Bolts 800.00
2 Clamps 600.00
3 Door Knobs 1200.00
4 Glue 650.00
5 Nails 1000.00
6 Screws 1595.00
7 Switch Board 699.60
8 Vaccum Cleaner 20000.00
9 Wood Blcoks 2400.00

Cover Letter

XYZ Database Systems **Phone: 402-000-0000**
102, North, Omaha, NE, 68000 **E-mail: bob.fix@gmail.com**

March 02, 2017
Bob,
Bob Fixit Group
102 South Omaha
NE-68000
402-000-0000

Dear Mr. Bob,
RE: Computerizing Data

With Reference to the Bob Fixit case study, we are writing to work on the contract of computerizing your data. I believe, we as a team possess the necessary skills and experience you are seeking and would make a valuable addition to your business.

We have reviewed your requirements; we recognize the relevance of our experience in the field of Database Management and Design. We as a team have created some excellent Database designs for many organizations.

We have proposed an initial design document based on your requirements for your review we look forward to speaking with you further regarding the contract.

Sincerely,
XYZ Database Systems.