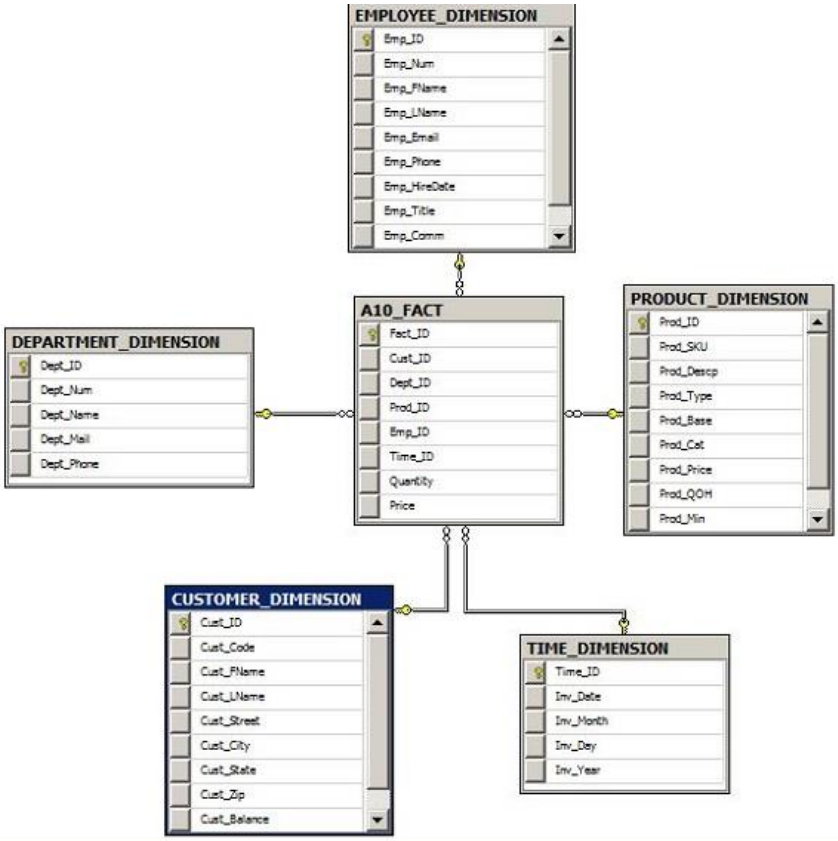
Part 1:



Part 2:

--PART 2

--Create a database to implement the design in Part 1 above.

--You must use Create table statements with Alter table statements to

--add all the constraints. Save the script in a file named A10P2.sql.

```
CREATE TABLE   CUSTOMER_DIMENSION
(
        Cust_ID         int identity,
        Cust_Code       int NOT NULL,
        Cust_FName      nvarchar(20),
        Cust_LName      nvarchar(20),
```

```
	Cust_Street	nvarchar(70),
	Cust_City	nvarchar(50),
	Cust_State	nvarchar(2),
	Cust_Zip	nvarchar(5),
	Cust_Balance decimal(8, 2),
)
	ALTER TABLE CUSTOMER_DIMENSION
	ADD CONSTRAINT PK_CUSTOMER_DIMENSION PRIMARY KEY(Cust_ID)

CREATE TABLE   DEPARTMENT_DIMENSION
(
	Dept_ID		int identity,
	Dept_Num	int not null,
	Dept_Name	nvarchar(50),
	Dept_Mail	nvarchar(3),
	Dept_Phone	nvarchar(9),
)
	ALTER TABLE DEPARTMENT_DIMENSION
	ADD CONSTRAINT PK_DEPARTMENT_DIMENSION PRIMARY KEY(Dept_ID)

CREATE TABLE   PRODUCT_DIMENSION
(
	Prod_ID		int identity,
	Prod_SKU	nvarchar(15) NOT NULL,
	Prod_Descp	nvarchar(255),
	Prod_Type	nvarchar(255),
	Prod_Base	nvarchar(255),
	Prod_Cat	nvarchar(255),
	Prod_Price	decimal(10, 2),
```

```
	Prod_QOH	decimal(10, 0),
	Prod_Min	decimal(10, 0),
)
	ALTER TABLE PRODUCT_DIMENSION
	ADD CONSTRAINT PK_PRODUCT_DIMENSION PRIMARY KEY(Prod_ID)

CREATE TABLE   EMPLOYEE_DIMENSION
(
	Emp_ID		int identity,
	Emp_Num		int NOT NULL,
	Emp_FName	nvarchar(20),
	Emp_LName	nvarchar(25),
	Emp_Email	nvarchar(25),
	Emp_Phone	nvarchar(20),
	Emp_HireDate datetime,
	Emp_Title	nvarchar(45),
	Emp_Comm	decimal(2, 2),
)
	ALTER TABLE EMPLOYEE_DIMENSION
	ADD CONSTRAINT PK_EMPLOYEE_DIMENSION PRIMARY KEY(Emp_ID)

--altering
CREATE TABLE   TIME_DIMENSION
(
	Time_ID		int identity,
	Inv_Date	datetime,
	Inv_Month	varchar(9),
	Inv_Day		int,
	Inv_Year	int,
```

```
)
	ALTER TABLE TIME_DIMENSION
	ADD CONSTRAINT PK_TIME_DIMENSION PRIMARY KEY(Time_ID)


CREATE TABLE   A10_STAGING
(
	Cust_ID		int,
	Dept_ID			int,
	Prod_ID			int,
	Emp_ID		int,
	Time_ID			int,
	CUST_CODE	int not null,
	DEPT_NUM	int not null,
	INV_DATE	datetime not null,
	EMP_NUM			int not null,
	PROD_SKU	nvarchar(15) not null,
	Price		decimal (8, 2),
	Quantity	float,
)

CREATE TABLE   A10_FACT
(
	Fact_ID		int identity,
	Cust_ID		int not null,
	Dept_ID			int not null,
	Prod_ID			int not null,
	Emp_ID		int not null,
	Time_ID			int not null,
```

```
	Quantity	int,
	Price		decimal (8, 2)
)
--Command(s) completed successfully. (For all of the above, including alter statements)

ALTER TABLE A10_FACT
	ADD CONSTRAINT PK_A10_FACT PRIMARY KEY(Fact_ID),
		CONSTRAINT FK_CUSTOMER_DIMENSION FOREIGN KEY(Cust_ID) REFERENCES CUSTOMER_DIMENSION,
		CONSTRAINT FK_DEPARTMENT_DIMENSION FOREIGN KEY(Dept_ID) REFERENCES DEPARTMENT_DIMENSION,
		CONSTRAINT FK_PRODUCT_DIMENSION FOREIGN KEY(Prod_ID) REFERENCES PRODUCT_DIMENSION,
		CONSTRAINT FK_EMPLOYEE_DIMENSION FOREIGN KEY(Emp_ID) REFERENCES EMPLOYEE_DIMENSION,
		CONSTRAINT FK_TIME_DIMENSION FOREIGN KEY(Time_ID) REFERENCES TIME_DIMENSION
-- Command(s) completed successfully


--Test to see that it worked - Yes, it worked
EXEC A10_Stored_Procedure
```

Part 3:

```
-- ============================================
-- Template generated from Template Explorer using:
-- Create Procedure (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- This block of comments will not be included in
-- the definition of the procedure.
-- ============================================
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- ============================================
-- Author:                  Megan Balcom
-- Create date: 12/6/2015
-- Description:    A10
-- ============================================
CREATE PROCEDURE A10_Stored_Procedure

AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;
```

```
--if nothing exists in the dimensions, it should still show all the transaction info
--IF
--BEGIN
--Step 1: Truncate all the tables
ALTER TABLE A10_FACT
                    DROP    CONSTRAINT     FK_CUSTOMER_DIMENSION,
                                CONSTRAINT  FK_DEPARTMENT_DIMENSION,
                                CONSTRAINT     FK_PRODUCT_DIMENSION,
                                CONSTRAINT     FK_EMPLOYEE_DIMENSION,
                                CONSTRAINT  FK_TIME_DIMENSION

        TRUNCATE TABLE  A10_FACT
        TRUNCATE TABLE        CUSTOMER_DIMENSION
        TRUNCATE TABLE        PRODUCT_DIMENSION
        TRUNCATE TABLE        DEPARTMENT_DIMENSION
        TRUNCATE TABLE        EMPLOYEE_DIMENSION
        TRUNCATE TABLE        TIME_DIMENSION
        TRUNCATE TABLE        A10_STAGING
--Command(s) completed successfully. (all of the above)

        --Step 2: Populate the dimensions

            ALTER TABLE A10_FACT
                    ADD CONSTRAINT FK_CUSTOMER_DIMENSION FOREIGN KEY(Cust_ID)
REFERENCES CUSTOMER_DIMENSION,
                    CONSTRAINT FK_DEPARTMENT_DIMENSION FOREIGN KEY(Dept_ID)
REFERENCES DEPARTMENT_DIMENSION,
                    CONSTRAINT FK_PRODUCT_DIMENSION FOREIGN KEY(Prod_ID)
REFERENCES PRODUCT_DIMENSION,
```

```
                    CONSTRAINT FK_EMPLOYEE_DIMENSION FOREIGN KEY(Emp_ID)
REFERENCES EMPLOYEE_DIMENSION,

                    CONSTRAINT FK_TIME_DIMENSION FOREIGN KEY(Time_ID) REFERENCES
TIME_DIMENSION



    --Step 3: Populate staging table

        --3.1: Populate the part of the staging table where columns are from the transaction DB


        INSERT INTO     A10_STAGING (CUST_CODE, DEPT_NUM, PROD_SKU, EMP_NUM,
INV_DATE, Price, Quantity)

        SELECT  C.CUST_CODE, D.DEPT_NUM, P.PROD_SKU, E.EMP_NUM, I.INV_DATE,
L.LINE_PRICE, L.LINE_QTY

        FROM    LGDEPARTMENT D INNER JOIN LGEMPLOYEE E ON D.DEPT_NUM = E.DEPT_NUM

                                INNER JOIN LGINVOICE  I ON I.EMPLOYEE_ID =
E.EMP_NUM

                                INNER JOIN LGCUSTOMER C ON C.CUST_CODE
= I.CUST_CODE

                                INNER JOIN LGLINE     L ON L.INV_NUM =
I.INV_NUM

                                INNER JOIN LGPRODUCT  P ON P.PROD_SKU =
L.PROD_SKU



        INSERT INTO     CUSTOMER_DIMENSION

        SELECT  CUST_CODE, CUST_FNAME, CUST_LNAME, CUST_STREET, CUST_CITY,
CUST_STATE, CUST_ZIP, CUST_BALANCE

        FROM LGCUSTOMER

        --(1362 row(s) affected)


        INSERT INTO     DEPARTMENT_DIMENSION

        SELECT DEPT_NUM, DEPT_NAME, DEPT_MAIL_BOX, DEPT_PHONE
```

```
FROM LGDEPARTMENT

--(8 row(s) affected)


INSERT INTO EMPLOYEE_DIMENSION

SELECT  EMP_NUM, EMP_FNAME, EMP_LNAME, EMP_EMAIL, EMP_PHONE,
EMP_HIREDATE, EMP_TITLE, EMP_COMM

FROM LGEMPLOYEE

--(363 row(s) affected)


INSERT INTO      PRODUCT_DIMENSION

SELECT  PROD_SKU, PROD_DESCRIPT, PROD_TYPE, PROD_BASE, PROD_CATEGORY,
PROD_PRICE, PROD_QOH, PROD_MIN

FROM LGPRODUCT

--(252 row(s) affected)


INSERT INTO TIME_DIMENSION

--SELECT          DISTINCT (INV_DATE), MONTH (INV_DATE) as INV_Month,
DAY(INV_DATE) as INV_Day, YEAR(INV_DATE) as INV_Year

SELECT  DISTINCT INV_DATE,DATENAME(MONTH, INV_DATE),DATEPART(DAY,
INV_DATE),DATEPART(Year, INV_DATE)

FROM    LGINVOICE

--(3351 row(s) affected)



--3.2: Assign DW keys to staging table

Update A10_STAGING

Set             Cust_ID = CD.Cust_ID

From    A10_STAGING S inner join CUSTOMER_DIMENSION CD

                ON      S.CUST_CODE = CD.Cust_Code
```

```
Update A10_STAGING
Set             Dept_ID = DD.Dept_ID
From    A10_STAGING S inner join DEPARTMENT_DIMENSION DD
                ON S.DEPT_NUM = DD.Dept_Num

Update A10_STAGING
Set             Prod_ID = PD.Prod_ID
From    A10_STAGING S inner join PRODUCT_DIMENSION PD
                ON S.PROD_SKU = PD.Prod_SKU

Update A10_STAGING
Set             Emp_ID = ED.Emp_ID
From    A10_STAGING S inner join EMPLOYEE_DIMENSION ED
                ON S.EMP_NUM = ED.Emp_Num

Update A10_STAGING
Set             Time_ID = TD.Time_ID
From    A10_STAGING S inner join TIME_DIMENSION TD
                ON S.INV_DATE = TD.Inv_Date


--3.3: Populate fact table using staging table
INSERT INTO A10_FACT(Cust_ID, Dept_ID, Prod_ID, Emp_ID, Time_ID, Price, Quantity)
SELECT  Cust_ID, Dept_ID, Emp_ID, Prod_ID, Time_ID, Price, Quantity
FROM    A10_STAGING


END
GO
```

Part 4:

-- PART 4

-- Write a query for each of the following:

--1. Show the sales totals by city of customers and product category. Show fields City, Category, Total (quantity * price)

```
Select    CD.Cust_City as City, PD.Prod_Cat as Category, SUM(Quantity * Price) as Total
From      A10_FACT AF inner join CUSTOMER_DIMENSION CD ON AF.Cust_ID = CD.Cust_ID
                                                  inner join PRODUCT_DIMENSION PD ON AF.Prod_ID = PD.Prod_ID
Group BY CD.Cust_City, PD.Prod_Cat
```

--2. What are the top three months in terms of sales (quantity * price)?

```
Select    TOP 3 SUM(AF.Quantity*AF.Price) as Total, Inv_Month
From      A10_FACT AF inner join TIME_DIMENSION TD ON AF.Time_ID = TD.Time_ID
Group BY Inv_Month
Order BY Total
```

--3. Create a question whose answer will require the use of all the dimensions. Write a query to answer your question.

--make it unique

--Write a query to find the customers' last names, product SKUs, department phone numbers, employee hire dates, and

-- invoivce dates associated with the customers who spent more than $1500 in November on a product

-- sold to them by a sales manager in the sales department. The results should be listed by employee hire date.

```
select  Cust_LName, Prod_SKU, Dept_Phone, Emp_HireDate, Inv_Date
```

```
	from	A10_FACT AF inner join CUSTOMER_DIMENSION CD on AF.Cust_ID = CD.Cust_ID
							inner join PRODUCT_DIMENSION PD on AF.Prod_ID =
PD.Prod_ID
							inner join EMPLOYEE_DIMENSION ED on AF.Emp_ID =
ED.Emp_ID
							inner join DEPARTMENT_DIMENSION DD on AF.Dept_ID
= DD.Dept_ID
							inner join TIME_DIMENSION TD on AF.Time_ID =
TD.Time_ID
	where	Cust_Balance > 1500 and Inv_Month like 'November' and Emp_Title like 'SALES
MANAGER' AND Dept_Num = 200
	order by Emp_HireDate


	--NEELY, 9413-EHI, 555-2824, 1995-09-01, 2013-11-01
	--CARNEY, 5437-WBO, 555-2824, 2000-10-14, 2013-11-05
```