**Solution:**

**Part 1:**

**Points to be remembered:**

**Use the provided script in the compiler.**

·         The script needs to be executed in the folder with SQL server.

·         At first, create the database in the computer system that are used.

·         Execute the provided SQL commands.

·         The successful execution of the provided SQL commands can be given in the screenshot.

·         In the execution, each table will be created in a particular order.

·         Refer the screenshot for better understanding of the execution of the SQL commands.

**The tables are created in the below manner:**

**1.** genre\_codes

***Execution of the SQL CODE:***

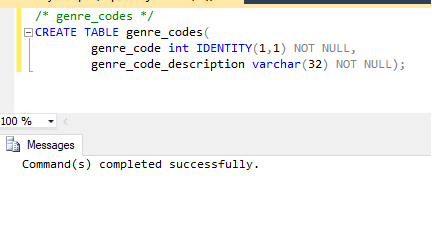
/\* genre\_codes \*/

CREATE TABLE genre\_codes(

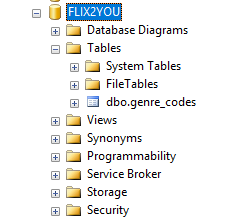
        genre\_code int IDENTITY(1,1) NOT NULL,

        genre\_code\_description varchar(32) NOT NULL);

**The successful completion of the SQL commands will be done:**

****

**The given screenshot shows that the table is created in the database db:**

****

**2.** format\_types

***Execution of the SQL CODE:***

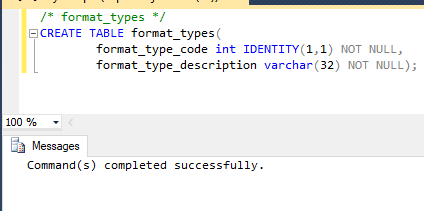
/\* format\_types \*/

CREATE TABLE format\_types(

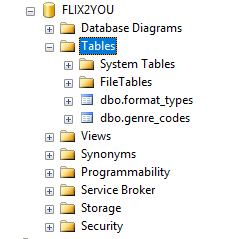
format\_type\_code int IDENTITY(1,1) NOT NULL,

format\_type\_description varchar(32) NOT NULL);

**The successful completion of the SQL commands will be done:**

****

**The given screenshot shows that the table is created in the database db:**



**3.** video\_stores

***Execution of the SQL CODE:***

/\* video\_stores \*/

CREATE TABLE video\_stores(

store\_id int IDENTITY(1,1) NOT NULL,

store\_name varchar(32) NOT NULL,

store\_address varchar(128) NOT NULL,

store\_city varchar(32) NOT NULL,

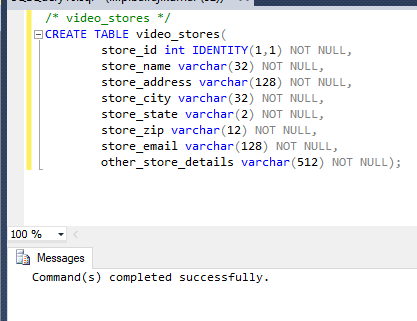
store\_state varchar(2) NOT NULL,

store\_zip varchar(12) NOT NULL,

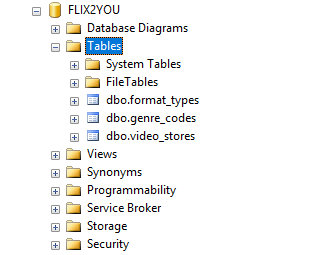
store\_email varchar(128) NOT NULL,

other\_store\_details varchar(512) NOT NULL);

**The successful completion of the SQL commands will be done:**

****

**The given screenshot shows that the table is created in the database db:**

****

**4.** condition\_codes

***Execution of the SQL CODE:***

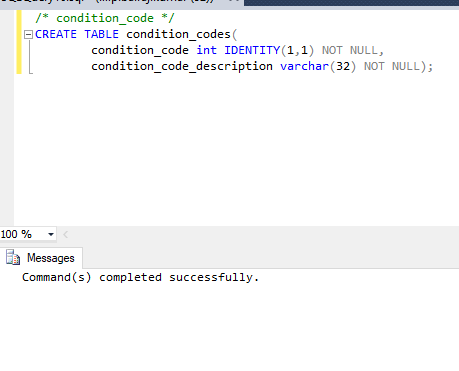
/\* condition\_code \*/

CREATE TABLE condition\_codes(

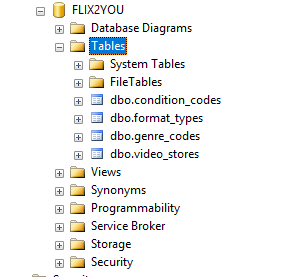
condition\_code int IDENTITY(1,1) NOT NULL,

condition\_code\_description varchar(32) NOT NULL);

**The successful completion of the SQL commands will be done:**

****

**The given screenshot shows that the table is created in the database db:**

****

**5.** actors

***Execution of the SQL CODE:***

/\* actors \*/

CREATE TABLE actors(

actor\_id int IDENTITY(1,1) NOT NULL,

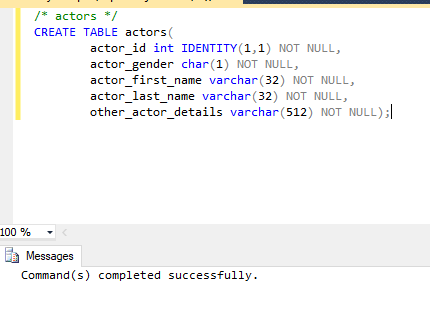
actor\_gender char(1) NOT NULL,

actor\_first\_name varchar(32) NOT NULL,

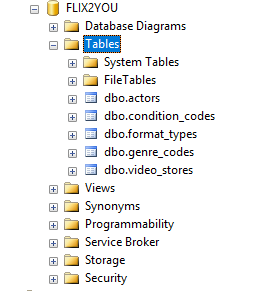
actor\_last\_name varchar(32) NOT NULL,

other\_actor\_details varchar(512) NOT NULL);

**The successful completion of the SQL commands will be done:**

****

**The given screenshot shows that the table is created in the database db:**

****

**6.** movies

***Execution of the SQL CODE:***

/\* movies \*/

CREATE TABLE movies(

movie\_id int IDENTITY(1,1) NOT NULL,

condition\_code int NOT NULL,

format\_type\_code int NOT NULL,

genre\_type\_code int NOT NULL,

store\_id int NOT NULL,

release\_year int NOT NULL,

movie\_title varchar(128) NOT NULL,

movie\_description varchar(1024) NOT NULL,

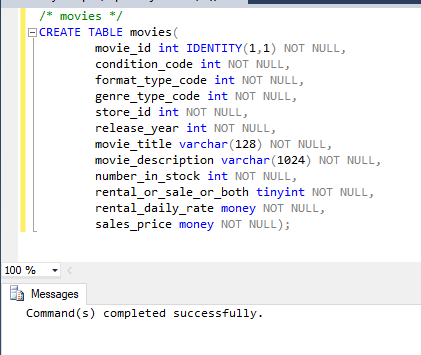
number\_in\_stock int NOT NULL,

rental\_or\_sale\_or\_both tinyint NOT NULL,

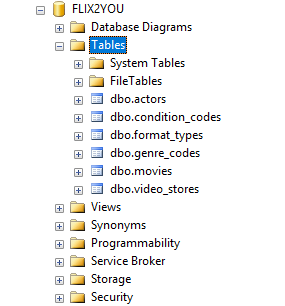
rental\_daily\_rate money NOT NULL,

sales\_price money NOT NULL);

**The successful completion of the SQL commands will be done:**

****

**The given screenshot shows that the table is created in the database db:**



**7.** movie\_cast

***Execution of the SQL CODE:***

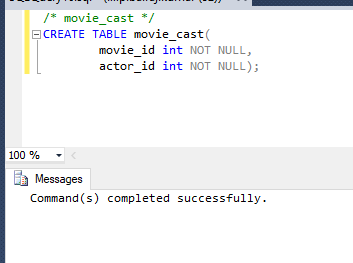
/\* movie\_cast \*/

CREATE TABLE movie\_cast(

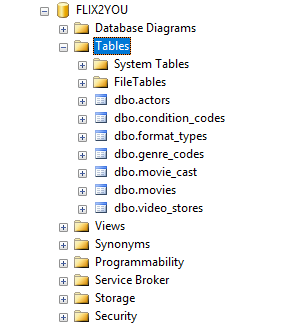
movie\_id int NOT NULL,

actor\_id int NOT NULL);

**The successful completion of the SQL commands will be done:**

****

**The given screenshot shows that the table is created in the database db:**



**8.** Alter script of tables to add the primary key constraints in all tables

***Execution of the SQL CODE:***

/\* create primary keys with ALTER TABLE statement \*/

ALTER TABLE genre\_codes ADD CONSTRAINT pk\_genre\_codes PRIMARY KEY (genre\_code);

ALTER TABLE format\_types ADD CONSTRAINT pk\_format\_types PRIMARY KEY (format\_type\_code);

ALTER TABLE video\_stores ADD CONSTRAINT pk\_video\_stores PRIMARY KEY (store\_id);

ALTER TABLE condition\_codes ADD CONSTRAINT pk\_condition\_codes PRIMARY KEY (condition\_code);

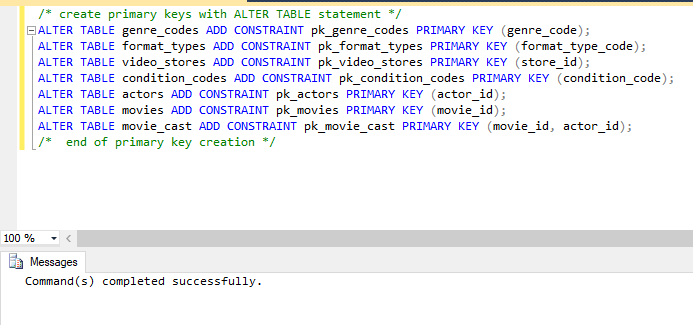
ALTER TABLE actors ADD CONSTRAINT pk\_actors PRIMARY KEY (actor\_id);

ALTER TABLE movies ADD CONSTRAINT pk\_movies PRIMARY KEY (movie\_id);

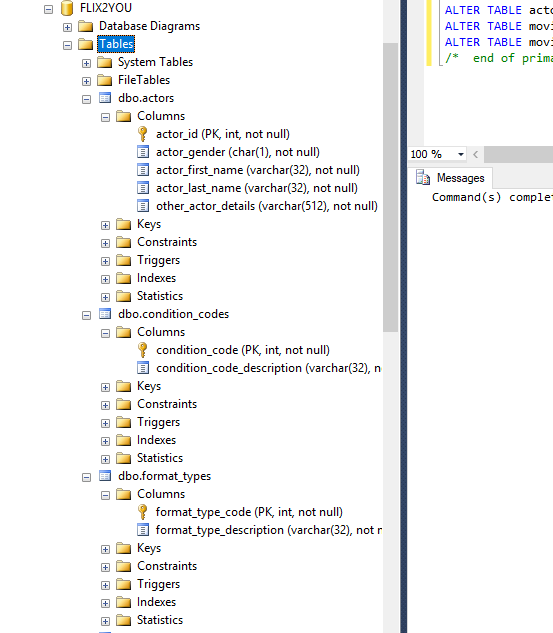
ALTER TABLE movie\_cast ADD CONSTRAINT pk\_movie\_cast PRIMARY KEY (movie\_id, actor\_id);

/\* end of primary key creation \*/

**The successful completion of the SQL commands will be done:**

****

**The given screenshot shows that the resulted table is created in the database db:**

****

**9.** Alter script of tables to add the foreign key constraints in movie\_cast and movies table

***Execution of the SQL CODE:***

/\* create foreign keys \*/

ALTER TABLE movie\_cast ADD CONSTRAINT fk\_Movie\_cast\_actors FOREIGN KEY(actor\_id)

REFERENCES actors (actor\_id);

ALTER TABLE movie\_cast ADD CONSTRAINT fk\_movie\_cast\_movies FOREIGN KEY(movie\_id)

REFERENCES movies (movie\_id);

ALTER TABLE movies ADD CONSTRAINT fk\_movies\_condition\_codes FOREIGN KEY(condition\_code)

REFERENCES condition\_codes (condition\_code);

ALTER TABLE movies ADD CONSTRAINT fk\_movies\_format\_types FOREIGN KEY(format\_type\_code)

REFERENCES format\_types (format\_type\_code);

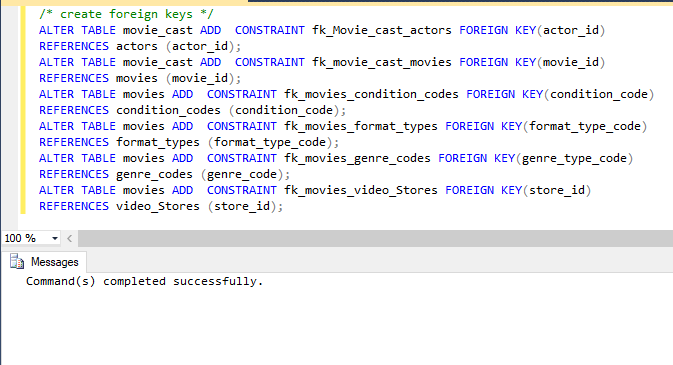
ALTER TABLE movies ADD CONSTRAINT fk\_movies\_genre\_codes FOREIGN KEY(genre\_type\_code)

REFERENCES genre\_codes (genre\_code);

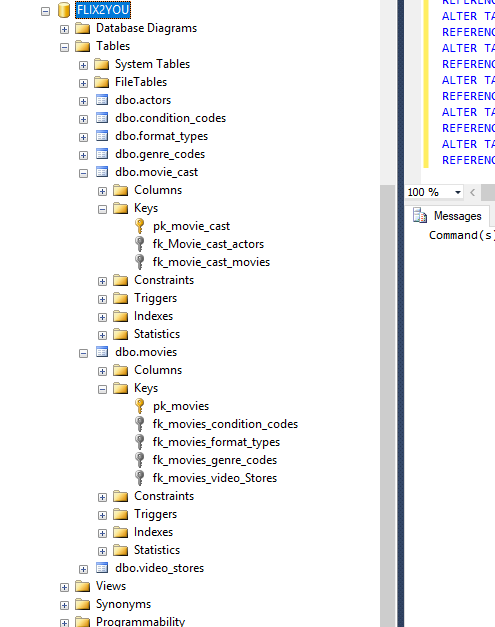
ALTER TABLE movies ADD CONSTRAINT fk\_movies\_video\_Stores FOREIGN KEY(store\_id)

REFERENCES video\_Stores (store\_id);

**The successful completion of the SQL commands will be done:**

****

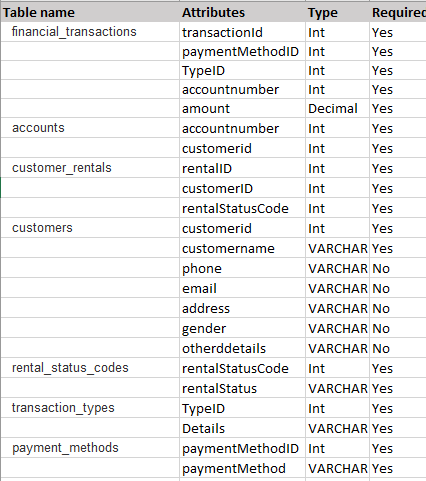
**The given screenshot shows that the resulted table is created in the database db:**

****

**10.** **SQL Data Dictionary:**

**The successful completion of the SQL data dictionary will be done.**

The given screenshot shows that the resulted script is created in the database db which returns the data detailed description of columns:

****

**Part 2:**

·         The data is inserted into the 2nd set of 7 tables.

·         The tables are created, and the SQL code will be written to create the 2nd set of 7 tables.

·         Refer the question to know the values that need to be inserted.

·         The successful execution of the provided SQL commands can be given.

***Execution of the SQL CODE:***

CREATE TABLE Customers(  
    LastName VARCHAR(50) NOT NULL PRIMARY KEY,  
    FirstName VARCHAR(50),  
    Addr\_Street VARCHAR(50),  
    Addr\_City VARCHAR(50),  
    Addr\_State CHAR(2),  
    Addr\_Zip CHAR(5),  
    Email VARCHAR(50)  
);

CREATE TABLE CustomerRentals(  
    Customer VARCHAR(50),  
    Movie VARCHAR(50),  
    Status VARCHAR(10),  
    DateOut DATE,  
    DateReturned DATE,  
    AmountDue REAL(5,2),  
    FOREIGN KEY(Customer) REFERENCES Customers(LastName)  
);

CREATE TABLE FinancialTransactions(  
    Customer VARCHAR(50),  
    Movie VARCHAR(50),  
    Type VARCHAR(10),  
    PayMethod VARCHAR(10),  
    Amount REAL(5,2),  
    `Date` DATE  
    FOREIGN KEY(Customer) REFERENCES Customer(LastName)  
);

INSERT INTO Customers VALUES('Smith', 'Harry', '1206 O Neill', 'Dunmore', 'PA', 18512, 'Fast\_harry AT hotmail.com');  
INSERT INTO Customers VALUES('Jones', 'Sally', '436 7th Ave', 'Pittsburgh', 'PA', 15219, 'jones\_sally AT mail.com');  
INSERT INTO Customers VALUES('Ryder', 'Amy', '4172 William Penn Highway', 'Monroeville', 'PA', 15146, 'crazy\_amy ATyahoo.com');  
INSERT INTO Customers VALUES('Walker', 'Skip', '503 Linden St.', 'Scranthon', 'PA', 18503);  
INSERT INTO Customers VALUES('Murphy', 'Nick','205 S Main St', 'Zelienople', 'PA'. 16063);

INSERT INTO CustomerRentals VALUES('Murphy', 'Spider-Man', 'Active', '2017/8/3', NULL, 5.50);  
INSERT INTO CustomerRentals VALUES('Jones', 'Superbad', 'Reserved', NULL, NULL, NULL);  
INSERT INTO CustomerRentals VALUES('Ryder', 'Gone w/ Wind', 'Overdue', '2017/6/2', NULL, 18.87);  
INSERT INTO CustomerRentals VALUES('Walker', 'Fahr 9/11', 'Returned', '2017/1/5', '2017/5/20', 4.61);  
INSERT INTO CustomerRentals VALUES('Smith', 'Good Will', 'Active', '2012/10/7', NULL, 4.56);

INSERT INTO FinancialTransactions VALUES('Smith', 'Good Will', 'Payment', 'Cash', 4.56, '2017/7/10');  
INSERT INTO FinancialTransactions VALUES('Jones', 'Superbad', 'Deposit', 'Credit card', 10.00, '2017/5/1');  
INSERT INTO FinancialTransactions VALUES('Ryder', 'Gone w/ W', 'Payment', 'Paypal', 18.87, '2017/6/2');  
INSERT INTO FinancialTransactions VALUES('Walker', 'Fahr 9/11', 'Payment', 'Check', 4.61, '2017/5/1');  
INSERT INTO FinancialTransactions VALUES('Walker', 'Fahr 9/11', 'Refund', 'Refund', 'Check', 1.00, '2017/5/30');

**Part 3:**

·         The data is updated into the tables to correct the errors.

·         Whenever the field will be identified as a primary key, that needs to be updated and modified.

·         Refer the question to know the updated values that need to be inserted.

·         The successful execution of the provided SQL commands can be given.

***Execution of the SQL CODE:***

UPDATE Customers SET Addr\_Zip = 18502 WHERE LastName = 'Walker';  
UPDATE FinancialTransactions SET Movie = 'Fahr 9/11' WHERE Customer = 'Smith';