

Johnson Video Store Records Automation

Steven Duchene

MIS407 – Reginald Haseltine

Colorado State University – Global Campus

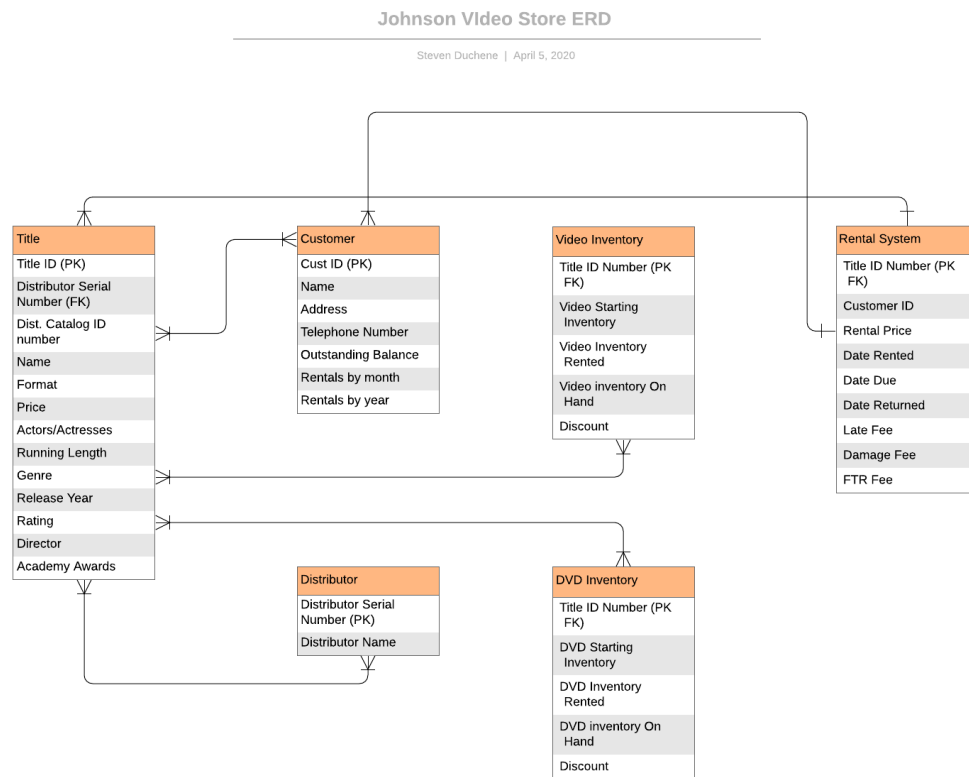
April 5 2020

Johnson Video Store Records Automation

As was discussed in our original meeting, Johnson Video Store has been utilizing a paper invoice and inventory system for several years. In order to develop a relational database to automate the store's record keeping system, a thorough analysis was completed of the organization's operational practices and current method of recordkeeping. The store maintains several copies of each movie that is rented; in the form of both videos and DVDs options.

The previously submitted report detailed the database initial study and database design steps of the database development lifecycle. The following sections detail the steps and results of the implementation/loading, testing/evaluation, operation, and maintenance of the Johnson Video Store Database. Overall, the methodology utilized to develop the database was the Waterfall Model of database development.

Entity Relationship Diagram



Database Table Creation Scripts

The screenshot shows the pgAdmin interface. On the left, the 'Browser' pane displays the 'public' schema containing tables: customer, distributor, dvd_inventory, rental_system, title, and video_inventory. The 'Query Editor' window on the right contains the following SQL scripts:

```

1  CREATE TABLE Title (
2  Title_ID_Number Integer Primary Key,
3  Dist_Serial_Number Integer,
4  Dist_Catalog_Number Text not null,
5  Title_Name text not null,
6  Format text not null,
7  Rental_Price decimal not null unique,
8  Actors_Actresses text not null,
9  Running_Length smallint not null,
10 genre text not null,
11 release_year date not null,
12 rating text null,
13 Director text not null,
14 Academy_Award boolean null);
15
16 CREATE TABLE Customer (
17 Customer_ID Integer Primary Key,
18 Customer_Name text not null,
19 Customer_Address text not null,
20 customer_Phone text not null,
21 Outstanding_Balance Decimal null,
22 Rentals_by_Month integer null,
23 Rentals_By_Year integer null);

```

The screenshot shows the continuation of the pgAdmin interface. The 'Query Editor' window contains the following SQL scripts:

```

24
25 CREATE TABLE Video_Inventory (
26 Title_ID_Number integer primary key references Title(Title_ID_Number),
27 Video_Start_Inv integer not null,
28 Video_Rent_Inv integer not null,
29 Video_On_Hand_Inv integer not null,
30 Discount Decimal null);
31
32 CREATE TABLE DVD_Inventory (
33 Title_ID_Number integer primary key References Title(Title_ID_Number),
34 DVD_Start_Inv integer not null,
35 DVD_Rent_Inv integer not null,
36 DVD_On_Hand_Inv integer not null,
37 Discount Decimal null);
38
39 CREATE TABLE Distributor (
40 Dist_Serial_Number integer primary key,
41 Dist_Name text not null);

```

```

42
43 CREATE TABLE Rental_System (
44 Title_ID_Number integer primary key,
45 Customer_ID integer references Customer(Customer_ID),
46 Rental_price decimal not null references Title(Rental_Price),
47 Date_Rented date not null,
48 Date_Due date not null,
49 Date_Returned date not null,
50 Late_Fee decimal null,
51 Damaged_Fee decimal null,
52 FTR_Fee decimal null)

```

Insertion of Data Statements (I was unable to go back far enough in query history to see the last two statements, so I had to rework them)

Help ▾

[Dashboard](#)
[Properties](#)
[SQL](#)
[Statistics](#)
[Dependencies](#)
[Dependents](#)
[Johnson Video ...](#)
[Johnson Video Store/postgres@Po](#)

[Query Editor](#)
[Query History](#)

```

INSERT INTO public.rental_system(
    title_id_number, customer_id, rental_price, date_rented, date_due, date_returned, late_fee,
    damaged_fee, ftr_fee)
VALUES (10776, 44912825, 4.20, '2020-04-01', '2020-04-05', NULL, NULL, NULL, NULL),
(83577, 522439973, 9.88, '2020-04-01', '2020-04-05', NULL, NULL, NULL, NULL),
(84852, 565270922, 0.80, '2020-04-01', '2020-04-05', NULL, NULL, NULL, NULL),
(22322, 262970669, 2.33, '2020-04-01', '2020-04-05', NULL, NULL, NULL, NULL),
(77571, 426338224, 8.89, '2020-04-01', '2020-04-05', NULL, NULL, NULL, NULL);

```

s ▾ Help ▾

[Dashboard](#)
[Properties](#)
[SQL](#)
[Statistics](#)
[Dependencies](#)
[Dependents](#)
[Johnson Video ...](#)
[Johnson Video Stor](#)

[Query Editor](#)
[Query History](#)

```

1 INSERT INTO public.dvd_inventory(
2     title_id_number, dvd_start_inv, dvd_rent_inv, dvd_on_hand_inv, discount)
3     VALUES (10776, 5, 0, 5, 0),
4 (83577, 5, 0, 5, 0),
5 (84852, 5, 0, 5, 0),
6 (22322, 5, 0, 5, 0),
7 (77571, 5, 0, 5, 0);

```

Dashboard Properties SQL Statistics Dependencies Dependents Johnson Video ... Johnson Video Store/postgres@f

[Query Editor](#)
[Query History](#)

```

1 INSERT INTO public.video_inventory(
2     title_id_number, video_start_inv, video_rent_inv, video_on_hand_inv, discount)
3     VALUES (10776, 5, 0, 5, 0),
4 (83577, 5, 0, 5, 0),
5 (84852, 5, 0, 5, 0),
6 (22322, 5, 0, 5, 0),
7 (77571, 5, 0, 5, 0);

```

Query Editor Query History

```

1 INSERT INTO public.customer(customer_id, customer_last_name, customer_first_name, customer_address,
2 customer_phone, outstanding_balance, rentals_by_month, rentals_by_year)
3 VALUES (44912825, "Manueli", "Winifield", "6 School Terrace", "643-332-3598", 1.06, 4, 48),
4 (522439973, "Esposito", "Cassie", "4 Hoffman Park", "341-995-1703", 6.54, 2, 11),
5 (565270922, "Polle", "Marilee", "580 Dorton Hill", "795-319-6696", 2.07, 3, 34),
6 (262970669, "ducarme", "Arvy", "51 Shasta Street", "347-844-4726", 9.48, 0, 18),
7 (426338224, "Wolland", "Toby", "1683 Lien Place", "687-470-2624", 7.56, 5, 65);

```

Query Editor Query History

```

1 INSERT INTO public.title(
2 title_id_number, dist_serial_number, dist_catalog_number, title_name, format, rental_price, actors_actresses, running_time,
3 distributor, release_date, genre, mpaa_rating, language, country,
4 (83577, 974894360, "1", "Mystery Science Theater 3000: The Movie", "Video", 9.88, "Marsella Braizier", 103, "Comedy",
5 (84852, 738627997, "3", "Murder, My Sweet", "Video", 0.80, "Calv Hitzschke", 134, "Crime|Film-Noir|Thriller", "2007-07-15",
6 (22322, 738627997, "3", "Cover Girl", "DVD", 2.33, "Meris Lamport", 61, "Comedy|Musical", "2019-07-15", "4", "Spanish",
7 (77571, 175430789, "1", "Elstree Calling", "DVD", 8.89, "Ximenez Lindell", 131, "Comedy|Musical", "2002-06-10", "3")

```

Select Statement to Show the Contents of All Tables

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays the database structure, including tables like 'customer', 'distributor', and 'dvd_inventory'. The main pane shows the 'Query Editor' with the following SQL query:

```

1 select distinct *
2 FROM customer , distributor , dvd_inventory , video_inventory , rental_system , title
3 order by customer_last_name

```

Below the query editor, the 'Messages' pane shows the execution status: 'Successfully run. Total query runtime: 611 msec. 15625 rows affected.'

The 'Data Output' pane displays the results of the query in a table format:

| | customer_id | customer_last_name | customer_first_name | customer_address | customer_phone | outstanding_balance | rentals_by_month | rentals_by_year |
|---|-------------|--------------------|---------------------|------------------|----------------|---------------------|------------------|-----------------|
| 1 | 262970669 | ducarme | Arvy | 51 Shasta Street | 347-844-4726 | 9.48 | 0 | 18 |
| 2 | 262970669 | ducarme | Arvy | 51 Shasta Street | 347-844-4726 | 9.48 | 0 | 18 |

Select Statement to Show Customer Information Sorted by Customer ID Number

The screenshot shows the pgAdmin interface with a SQL query editor. The query is as follows:

```
1 select customer_last_name , customer_first_name , customer_address , customer_id
2 FROM customer
3 order by customer_id
```

The query results are displayed in a table with the following data:

| customer_last_name | customer_first_name | customer_address | customer_id |
|--------------------|---------------------|------------------|-------------|
| Manueli | Winifield | 6 School Terrace | 44912825 |
| ducarme | Arvy | 51 Shasta Street | 262970669 |
| Wolland | Toby | 1683 Lien Place | 426338224 |
| Esposito | Cassie | 4 Hoffman Park | 522439973 |
| Polle | Marilee | 580 Dorton Hill | 565270922 |

Sort by Date Rented

The screenshot shows the pgAdmin interface with a SQL query editor. The query is as follows:

```
1 select title_name , date_rented
2 FROM rental_system , dvd_inventory , title
3 where date_rented BETWEEN '2020-03-06' AND '2020-04-05'
4 order by date_rented
```

The query results are displayed in a table with the following data:

| title_name | date_rented |
|-------------------|-------------|
| Mommie Deare... | 2020-04-01 |
| Mommie Deare... | 2020-04-01 |
| Mommie Deare... | 2020-04-01 |
| Mommie Deare... | 2020-04-01 |
| Mommie Deare... | 2020-04-01 |
| Mystery Scienc... | 2020-04-01 |

Update Customer Last Name

The screenshot shows the pgAdmin 4 interface with the 'customer' table selected in the left sidebar. The Query Editor displays the following SQL query:

```
1 UPDATE public.customer
2 SET customer_last_name = 'Johnson'
3 WHERE customer_last_name = 'Polle';
4
5 select * from public.customer
```

The 'Data Output' tab shows the following data:

| customer_id | customer_last_name | customer_first_name | customer_address | customer_phone | outstanding_balance | rentals_by |
|-------------|--------------------|---------------------|------------------|------------------|---------------------|------------|
| 1 | 44912825 | Manueli | Winifield | 6 School Terrace | 643-332-3598 | 1.06 |
| 2 | 522439973 | Esposito | Cassie | 4 Hoffman Park | 341-995-1703 | 6.54 |
| 3 | 262970669 | ducarme | Arvy | 51 Shasta Street | 347-844-4726 | 9.48 |
| 4 | 426338224 | Wolland | Toby | 1683 Lien Place | 687-470-2624 | 7.56 |
| 5 | 565270922 | Johnson | Marilee | 580 Dorton Hill | 795-319-6696 | 2.07 |

Delete Customer from Database

The screenshot shows the pgAdmin 4 interface with the 'customer' table selected in the left sidebar. The Query Editor displays the following SQL query:

```
1 delete from public.rental_system
2 WHERE customer_id = 44912825;
3
4 DELETE FROM public.customer
5 WHERE customer_id = 44912825;
```

The 'Data Output' tab shows the following data:

| customer_id | customer_last_name | customer_first_name | customer_address | customer_phone | outstanding_balance | rentals_by |
|-------------|--------------------|---------------------|------------------|------------------|---------------------|------------|
| 1 | 522439973 | Esposito | Cassie | 4 Hoffman Park | 341-995-1703 | 6.54 |
| 2 | 262970669 | ducarme | Arvy | 51 Shasta Street | 347-844-4726 | 9.48 |
| 3 | 426338224 | Wolland | Toby | 1683 Lien Place | 687-470-2624 | 7.56 |
| 4 | 565270922 | Johnson | Marilee | 580 Dorton Hill | 795-319-6696 | 2.07 |

Conclusion

In concluding this assignment, there were several things that I learned regarding the use of the PostgreSQL platform as well as about myself. In the current environment we are experiencing, I came down with a still unknown illness; making it that much harder to complete the assignment. Nonetheless, I prevailed. PostgreSQL continues to be an easy platform to work with and is very user friendly. Most, if not all, of my troubles came from having an inadequate understanding of and experience with the SQL language.

Another area where I found a major mistake in my development of the database was regarding the title, DVD, and video tables. I realized towards the end of the assignment that I had failed to include the format when selecting only for DVDs. Looking back, it is likely that my results included both videos and DVDs. If I were to fix that mistake, I would have also included format in my statement as well as selected only for DVD formats. I thoroughly enjoyed this class and it has likely been one of my favorite courses this far. I look forward to utilizing it in the future.

References

Mockaroo. (n.d.). Random Data Generator and API Mocking Tool: JSON / CSV / SQL / Excel.

Retrieved from <https://www.mockaroo.com/>