



Practical Assignment

Aims

- Implement some basic to intermediate SQL queries for the database schema.
- Create database views.

The Scenario

In this assignment you will be constructing an SQL script that builds a series of database views from the DBMS you created in Practical Assignment 1.

You have been tasked by the owners of MovieDirect, a small retailer and online streaming platform, to re-develop their orders and shipments database system. Currently, the system uses four separate spreadsheets to keep track of customers, movies, stock that is currently available and shipments out to the customers. This system is starting to become difficult to use and will prevent MovieDirect from effectively managing its long-term operations in its current state.

Your task was to develop a robust and scalable database solution for the information system that will effectively store MovieDirect's information and provide the capabilities for extracting information to improve sales and management of customers. You should have completed this as part of assignment 1, your next task is to create a set of user views to extract meaningful information from the live database.

For this assignment, transaction data and a user view template will be provided, make sure to utilise these to check your solution.

Your first task was to develop a database using the PostgreSQL DBMS, which will contain four tables (Customers, Shipments, Movies and stock) linked together.

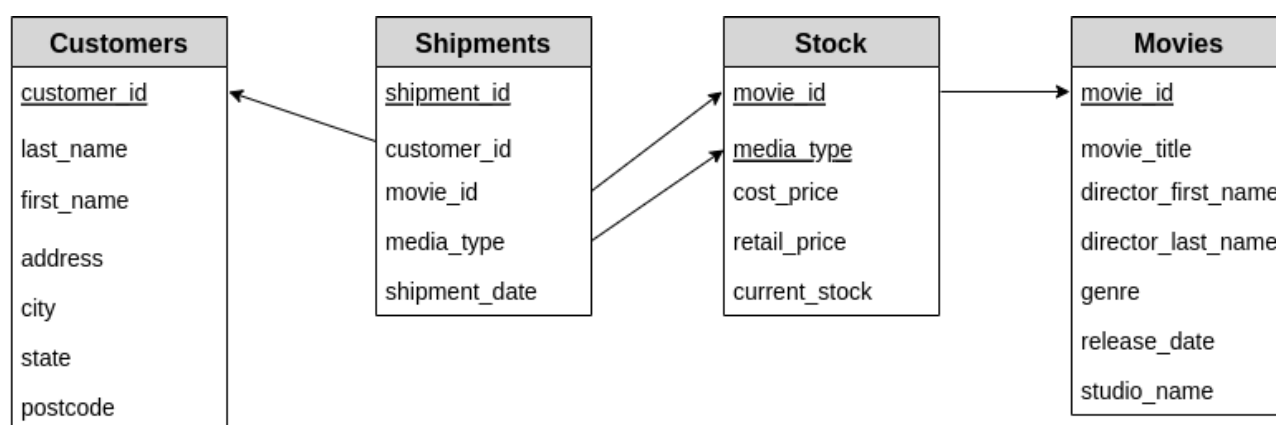


Figure 1. The database schema for MovieDirect.

The SQL script to import data into your completed database is available here: http://turing.une.edu.au/~cosc210/assignments/a4/MovieDirect_Data.sql

Exercise 2 – A set of user views with SQL queries.

Please make sure that you complete your user views using the template that is provided. Your assignment will be marked with the assistance of automated tools and if the names of the views and their attributes are

not correct, you may lose marks.

The assignment template is available here: http://turing.une.edu.au/~cosc210/assignments/a4/p_template.sql

Question Set

Construct an SQL script (i.e. a file with the .sql extension) that contains the definitions of the following views.

1. Create a view called 'movie_summary' which returns the movie_title, release_date, media_type and retail_price for all movies in the database. This will contain some duplicates for media_type. **(10 Mark)**
2. Create a view called 'old_shipments' that lists the customer first_name and last_name, movie_id, shipment_id and shipment_date for every shipment before 2010. **(10 Mark)**
3. Create a view called 'trilogy' that Returns a list of the titles of all movies with the words 'Rings' or 'Wars' in the title. **(10 Mark)**
4. Create a view called 'retail_price_hike' that returns the movie_id , retail_price and a final column that contains the retail price increased by 25%. **(10 Mark)**
5. Create a view called 'value_summary' that returns the total cost value (cost*stock) and total retail value (retail*stock) across all stock. **(10 Mark)**
 - Note: this question includes infinite values for Stream-media, this is not a requirement, but it may be good to remove stream-media from the results.
6. Create a view called 'profits_from_movie' that returns the movie_id and movie_title for each movie along with the difference between the sum of the cost and retail values across all shipments for each movie. The results should be grouped by movie_title. **(10 Mark)**
7. Create a view called 'followers_of_melkor' that returns the first name, last name of any customer who has **not** purchased any media_type of the movie named 'The Lord of the Rings: The Fellowship of the Ring'. **(15 Marks)**
8. Create a view called 'sole_angry_watcher' that returns the first_name and last_name of any customer (If one exists) who is the only customer to buy '12 Angry Men' (Note the customer may buy other movies as well, but if anyone else buys the movie, no records should be returned). **(15 Marks)**

Make sure to test your scripts on turing.une.edu.au and utilise the template and import script provided.

Assignment Submission

You should submit one file for this assignment:

- An sql file named in the format: exercise_2_<username>.sql (e.g. exercise_2_esadgro2.sql)
- Submit your assignment via turing.une.edu.au's submit program.
- **You should submit your assignment to the p2 assignment.**
- The instructions for submit are available [here](#)

Marking

Item	Marks
Admin	- /10
Files named correctly	- /4
SQL script runs without errors (Ex2)	- /6
Exercise 2	- /90
PSQL User View Question 1	- /10
PSQL USer View Question 2	- /10

Item	Marks
PSQL User View Question 3	- /10
PSQL User View Question 4	- /10
PSQL User View Question 5	- /10
PSQL User View Question 6	- /10
PSQL User View Question 7	- /15
PSQL User View Question 8	- /15