Dunnes Video Shop Operational Database

This week is some revision on constructing databases. Using entity-relationship diagrams to guide us through building a database. You will construct an ER-diagram based on the description below. Then you will create an SQL script using DDL commands that creates the required data structures including database and tables and sets up the primary key/foreign key connections to support referential integrity where necessary.

Dunnes Video Shop require a system to record the transactions in their Video Store.

## The main requirements are:

### Ability to populate/update the Video Catalogue

### May be a number of copies of a video

### Hold information on Customers/Members

### Renting a video from the store

### Returning a video

## Main Tasks:

### Name the Entities

Video

(Name,Format,Edition)

Customer

(ID,Name,Payment Method)

### Show the Relationships

### Identify Attributes

### Identify PK and FK

### Draw high level conceptual ER data model

### Modify conceptual data model to produce logical relational data model

### Create an SQL script using DDL commands that creates the required data structures including database and tables and sets up the primary key/foreign key connections to support referential integrity where necessary.

# Some preparation for next week’s lab

If you can’t get MySQL working on your laptop or it isn’t available on your lab computer, I advise you have a look at this tutorial on SQLite: <https://docs.python.org/3/library/sqlite3.html>

In fact, I recommend everyone look at this tutorial regardless. Learning how to use both libraries will be very useful for future projects.

Assuming you used MySQL to setup this database, it will be possible to connect to connect to this database using mysql-connector-python. This library allows you to run SQL scripts from a python program. To install this library, type the following command into command prompt (or terminal if using mac):

pip install mysql-connector-python

To use this library in python you will first need to import it:

import mysql.connector as mysqlc

To make a connection to this database you will then need to write the following lines of code:

# connecting to database using configurations

# may need to adjust password/user/database as appropriate

config = {

  'user': 'root',

  'password': 'myPassword',

  'host': '127.0.0.1',

  'database': 'name\_of\_database',

  'raise\_on\_warnings': True

}

connection = mysqlc.connect(\*\*config)

mycursor = connection.cursor()

The above code should only create a connection if the password matches the same password you use for signing in on your MySQL database.

If you want to insert data into an existing table in your database, you can sore the SQL statements in strings and attempt to execute them using the following code:

# constructing an sql statement for the database

sql = "INSERT INTO customer VALUES (%s, %s, %s, %s)"

val = (999, "John", '1990-02-27', "Dublin 12")

#executing the sql statment and committing it to the DB

mycursor.execute(sql, val)

connection.commit()