

# UNIVERSITY OF TECHNOLOGY, SYDNEY

Faculty of Engineering and Information Technology

**Object-Relational Databases 31075/42901**

**Spring 2019**

## **DATABASE DESIGN AND IMPLEMENTATION ASSIGNMENT**

**(50 marks)**

This assignment is due in week 11, at **9:00 am on Monday 7 October 2019**. The assignment should be submitted in electronic form in the PDF format via UTSONline Turnitin Assignments before the above deadline. Only a single submission per group will be accepted. The file name should have the format **StudentNo1\_StudentNo2** (i.e. concatenation of the student numbers of both students in the group). The title page of the assignment must contain:

- 1) Subject name and number
- 2) Assignment title
- 3) Names and student numbers of group members
- 4) Oracle username where the assignment was implemented (do not include your password)

Refer to the subject outline for the academic standards statement. General assignment feedback will be provided in the lecture and individual feedback via UTSONline. Ask in the lecture if you need any clarifications of the assignment. **Late assignments will not be accepted.**

This assignment supports the following subject objectives: (4) students should be able to show working knowledge of object-relational features of the SQL standard as implemented in Oracle DBMS, (5) students should be able to explain object-relational database design issues and trade-offs, and (8) students should be able to show ability to communicate in the form of a structured report. The assignment is a team effort; each group will have **2 students**. Please, note that assignments submitted by a group of more than two students will not be accepted. Prior arrangement is needed for anyone who wishes to submit the assignment individually with confirmation via email by the subject coordinator; this will only be granted in special circumstances. The mark of each team member will be weighted according to their individual contribution as assessed by their peers in the team. The individual contribution to this assignment must be at least 20%. If you feel that other group members are not contributing, the subject coordinator should be informed to find a solution. No complaints about group operation will be considered after the assignment has been handed in. The assignment accounts for 50% of the marks for the subject and should represent a substantial effort (approximately 50 hours of work for each group member). Assignments in this subject should be your own original work. The inclusion in assessable work of any material such as code, graphics or essay text obtained from other persons or sources without citation of the source is plagiarism and is a breach of University Rule 16.2.2. **Assignments that contain copied material will be given zero marks.**

### **ASSESSMENT**

Assessment of the assignment will be based on correctness, completeness and conciseness of your design and program code. An important objective of the assignment is to demonstrate your ability to communicate in the form of a structured report. Poor quality of the report will result in a loss of marks. The assignment report should include the following sections:

1. Introduction - brief introduction outlining the objectives and content of the report
  2. Entity-Relationship Model and Relational Design (section 2.1)
  3. OMDB Object-Relational Implementation (section 2.2)
  4. OMDB Methods and Queries (section 2.3)
  5. Discussion
- Appendices – containing scripts and corresponding computer output

## 1. SCENARIO

The assignment involves design and implementation of an Online Music Database (OMDB). OMDB maintains information about music albums available for purchase from an online music store. The database supports a website that allows users to query this information and search for albums; visit <https://music.amazon.com> for an illustration of a similar website. Albums are available as audio CDs, vinyl records, or in MP3 format for download. The price of albums varies depending on the format, e.g. a vinyl record is typically more expensive than an audio CD of the same album. New albums are generally more expensive than used items. Note that albums with the same title can be released on different dates and on different media (i.e. release date, price, etc. of vinyl, audio CD and MP3 can have different values for the same album title).

### 1.1 OMDB Information Requirements

Albums are described by the following attributes:

Album Title	title of the album
Album Playtime	play time in minutes
Album Release Date	the date of release
Album Genre	album genre
Album Price	price of a new album (audio CD, vinyl, or MP3)
Album Tracks	number of tracks on the album
Album Artists	up to 5 artists can be recorded with the following attributes: Artist Name Artist Role
Album Reviews	unlimited number of reviews with the following attributes: Reviewer Name Review Date Review Text Review Score

Audio CDs and Vinyl albums have the following additional attributes:

Media Type	type of media, i.e. Audio CD or Vinyl
Disk Number	number of disks
Disk Used Price	price of used CD or vinyl disks
Disk Delivery Cost	shipping costs

MP3 albums have the following additional attribute:

Download Size	MP3 size in MB
---------------	----------------

## 2. YOUR TASK

This section details your tasks in this assignment; marks are indicated for each sub-section. The individual sections should be presented in the form of a report with corresponding section numbers. Section 1 should be a brief introduction outlining the objectives and content of the report. All computer output should be included in the appendix.

### 2.1 Entity-Relationship Model and Relational Design (10 marks in total)

- Use the Oracle Data Modeler to draw an Entity-Relationship Diagram (ERD) to represent the data requirements of the OMDB database. The ERD should show all entities, relationships, attributes, identifiers and cardinalities. Make use of sub-typing where appropriate. Use the names indicated in the above description for name of entities, relationships and attributes. Ensure that your ERD (Logical Model) uses the Barker notations; this can be specified in Preferences; Data Modeler; Diagram; Logical Model. You can download the Oracle Data Modeler from:

<http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/index.html>.

Data Modeler User's Guide is available on:

<https://docs.oracle.com/en/database/oracle/sql-developer-data-modeler/18.2/dmdug/oracle-sql-developer-data-modeler-users-guide.pdf> (5 marks)

- b) Using the Data Modeler convert the OMDB ERD (Logical Model) into a set of corresponding tables (Engineer to Relational Model). Draw the table diagram (Relational Model) and ensure that the resulting tables are fully normalized (i.e. in BCNF). (5 marks)

*The deliverables for this section are the OMDB ERD and table diagrams.*

## 2.2 OMDB Object-Relational Design and Implementation (5 marks)

The above requirements were addressed by implementing a single typed table Albums that includes two collections: variable array AlbumArtists and a nested table AlbumReviews. To account for different types of albums the album\_type supertype has two subtypes: disk\_type (audio CDs and vinyl disks) and mp3\_type (MP3 downloads). The script used for creating the OMDB database (omdb.sql) is included in the assignment folder. Inspect the omdb.sql script and use it to create the OMDB database. **Include the script and the results of its execution in the appendix.**

Now create a script of INSERT statements to populate the OMDB database using the data supplied in the omdb\_data.txt file and suitable constructor statements. Run the script and ensure that it executes without errors. **Include the script and the results of its execution in the appendix.**

*The deliverables for this section should be an appendix containing the OMDB SQL scripts and the results of their execution.*

## 2.3 OMDB Methods and Queries (30 marks in total)

Your task in this section is to implement methods and SQL queries. **Show both the SQL query and the corresponding query output. Queries without query output will be given zero marks.**

1. Give album title, album release date and album price of all Neil Young's albums released after 1st January 2015. (2 marks)
2. Give album title and artist name for albums released only in MP3 format. Order by album title. (2 marks)
3. Give lowest rated MP3 album (i.e. album with the lowest average review score). Show album title and the average score. Exclude albums with only one review. (3 marks)
4. Are there any albums released on all media, i.e. on MP3, audio CD and vinyl? Show album title and order by album title. (3 marks)
5. Implement the method discountPrice() that returns a discounted price using the following business rule:
  - a. for audio CDs released more than one year ago the discount is 20%
  - b. for vinyl records released more than one year ago the discount is 15%
  - c. for MP3 downloads released more than two years ago the discount is 10%

Note that the signature of the discountPrice method is included in the original OMDB script for both disk\_type and mp3\_type subtypes. (5 marks)

6. Create a view all\_albums that includes the columns: album title, media type ('MP3', 'Vinyl', 'Audio CD'), album price, and discount (album price – discount price). Use this view to find the album that received the largest discount; show all view columns. (5 marks)

7. Now, modify the view `all_albums` to also include the column `album used price` for disks; set `album used price` to zero for MP3 albums. Use this view to find the most expensive used album; show all view columns. (5 marks)
8. Implement the method `containsText(pString1, pString2)` that returns 1 if `pString1` contains `pString`, and 0 if it does not. Use this method to find albums with reviews that contain the word 'Great'. Show album title, review text, review score. Note that the signature of the `containsText` method is included in the original OMDB script. (5 marks)

#### **2.4 Discussion (5 marks)**

Briefly discuss the relative advantages of the relational design in section 2.1 when compared to the object-relational design presented in section 2.2 and implemented using the `omdb.sql` script.