Pete Reitmeyer

Home Work 4: Oracle SQL Class

# Research

In your own words give a brief definition of CRUD as it relates to databases.

* CRUD is an acronym for Create, Read, Update, and Delete that then relate directly to Create/Insert, Select, Update, and Delete in SQL. These are the 4 major functions in relational database applications.

In your own words give a brief definition of ACID as it relates to databases.

* ACID is an acronym for Atomicity, Consistency, Isolation, and Durability, which are the properties associated with database transactions (any single logical operation).

In your own words give a brief definition of Atomicity as it relates to databases.

* Atomicity for database transactions is the concept of “all or nothing”, i.e. the entire transaction passes or fails. If any one part fails the entire transaction fails (no partial commits etc.). This must be true even for power failures and crashes etc.

In your own words give a brief definition of Drilling Down as it relates to databases.

* Drilling Down is the concept of accessing information within the database by going from general categories down through lower levels of detail, e.g. from file/table to record to field. This is similar to going from folders to files to actually opening the file on your computer.

In your own words give a brief definition of Granularity as it relates to databases.

* Granularity refers to the number columns or separate fields that the data is separated in to. High granularity would mean multiple columns to encompass yet separate the data allowing for more flexibility in working with the different fields as necessary.

# DECODE

|  |  |
| --- | --- |
| **SCORES** | |
| **Student\_id** | **Score** |
| 2122 | 54 |
| 3211 | 92 |
| 411 | 80 |
| 2771 | 77 |
| 2098 | 64 |
| 7890 | 0 |

Create a table called scores.

Enter the data above.

Write a SQL statement using the decode function from Chapter 6 that gives a result set showing:

Sudent\_id

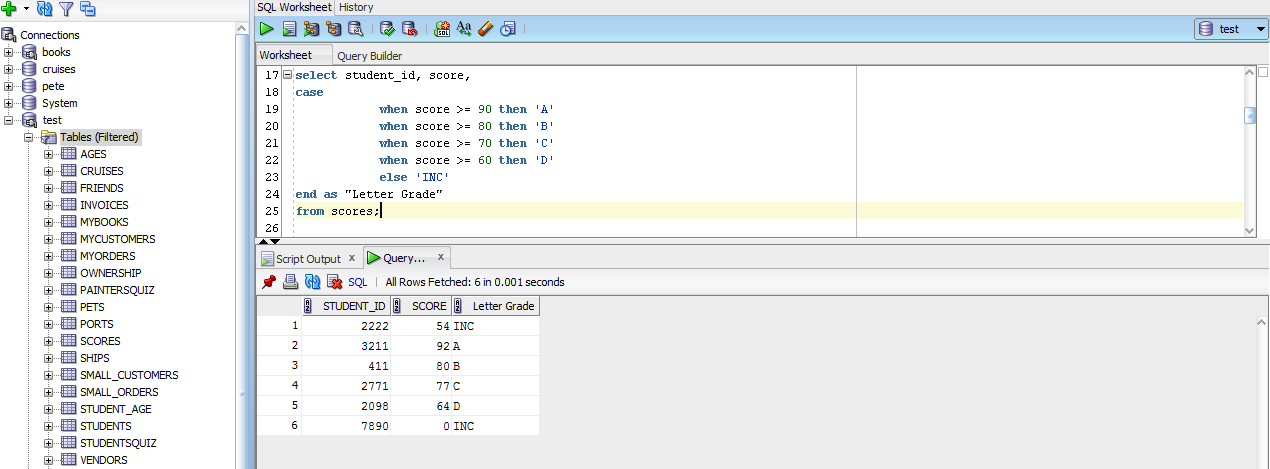
Score

Letter Grade

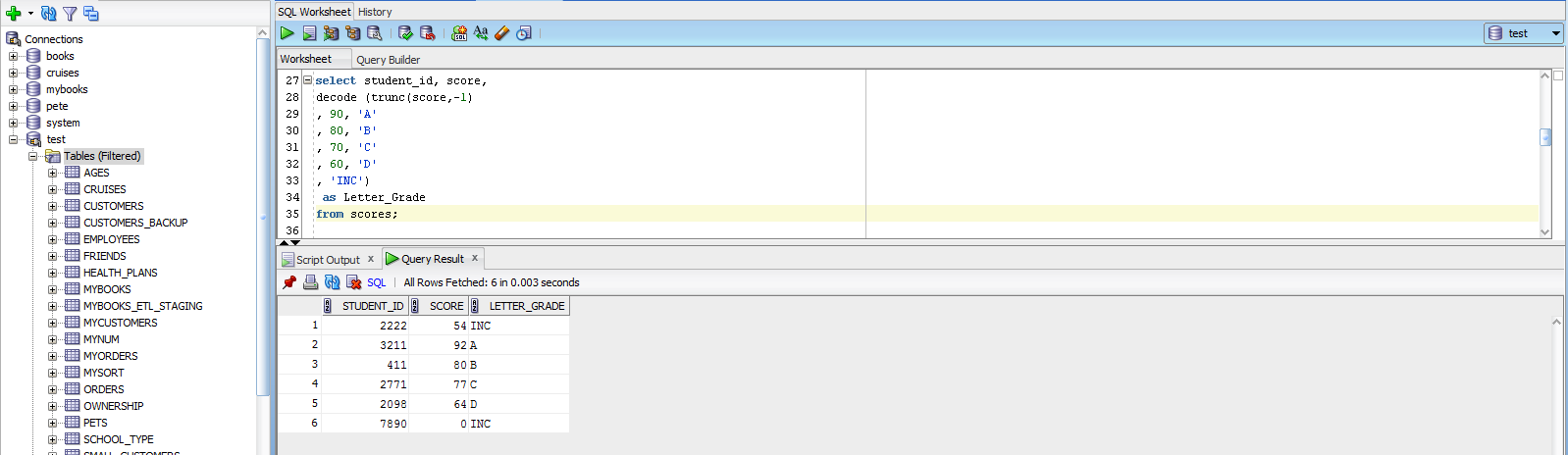
Where the letter grade is A if score 90-100; B if score 80-89;

C if score 70-79; D if score 60-69; INC if score 0-59;

Originally done with case:

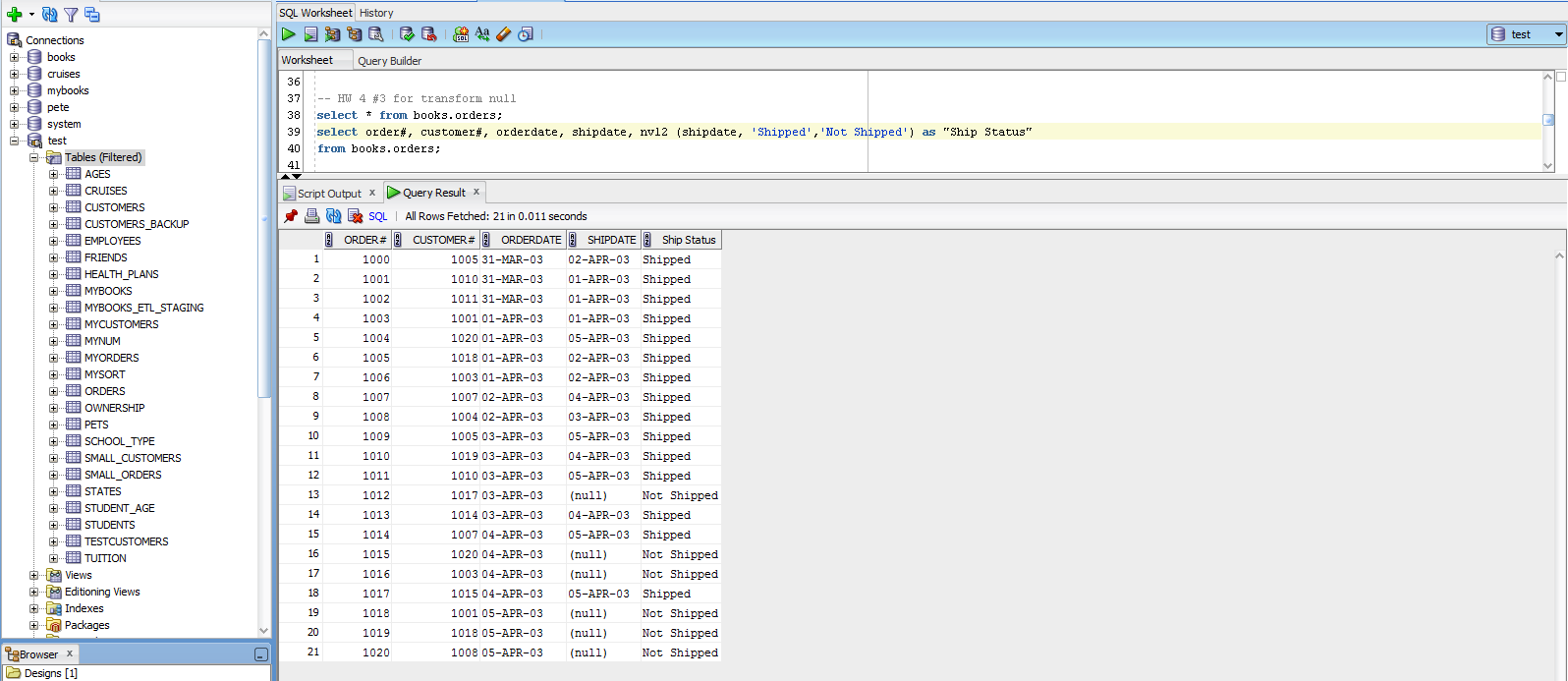


Then I did it with decode after quick chat with Clyde about math expressions not working for decode:



# Transform null

Using Orders table in Books, write a SQL statement that returns “Not Shipped” if the shipdate is null.



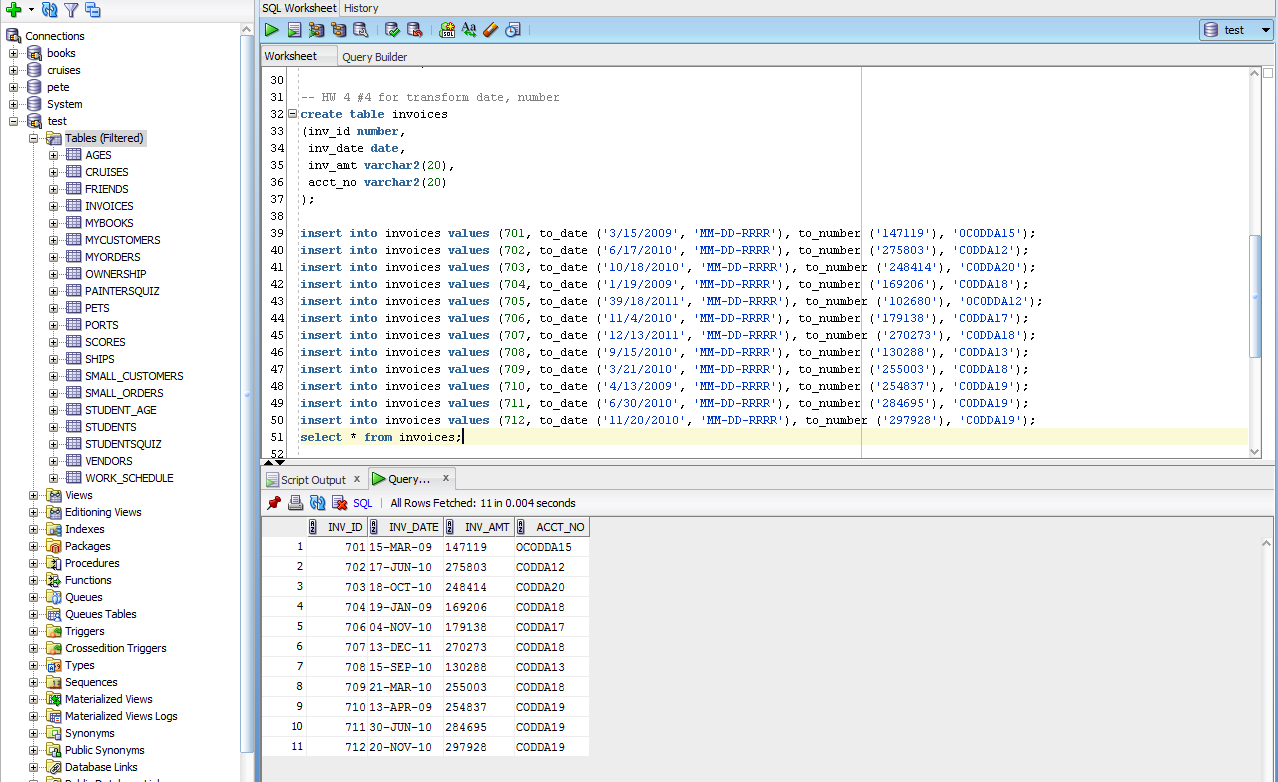
# Transform date, number

Create a table called invoices and insert the following rows. In the insert statement use

to\_date to change string ‘3/15/2009’ to a date

to\_number to change string ‘147119’ to a number

|  |  |  |  |
| --- | --- | --- | --- |
| INVOICES | | | |
| Inv\_id | inv\_date | inv\_amt | Acct\_no |
| 701 | 3/15/2009 | 147119 | 0CODDA15 |
| 702 | 6/17/2010 | 275803 | CODDA12 |
| 703 | 10/18/2010 | 248414 | CODDA20 |
| 704 | 1/19/2009 | 169206 | CODDA18 |
| 705 | 9/18/2011 | 102680 | CODDA12 |
| 706 | 11/4/2010 | 179138 | CODDA17 |
| 707 | 12/13/2011 | 270723 | CODDA18 |
| 708 | 9/15/2010 | 130288 | CODDA13 |
| 709 | 3/21/2010 | 255003 | CODDA18 |
| 710 | 4/13/2009 | 254837 | CODDA19 |
| 711 | 6/30/2010 | 284695 | CODDA19 |
| 712 | 11/20/2010 | 297928 | CODDA19 |



# Case Statement

Create a SQL case statement that returns all table columns with one additional column that states

**Add Invoice to 2009** if the invoice date is in 2009

**Add Invoice to 2010** if the invoice date is in 2010

**Add Invoice to 2011** if the invoice date is in 2011

