Databases I Week 5 Functions

The content and layout of this week’s practical class is similar to the lab test in week 8 but the number and type of questions are different.

## Recommendation for doing the lab test

In order to make sure you have something to submit I would recommend you get into the habit of saving your work consistently throughout. I would recommend you write and test the SQL in this order:

* + Create the required table (basic datatypes and primary keys)
    - Test and Save your work.
  + Populate the table with data.
  + Add the column required to the table identified.
  + Update the data.
  + Add the relevant constraint to the table identified.
  + Answer the questions
    - Test and Save your work after each element.
  + Compile one single script.
  + Make sure you clearly indicate which question each SQL statement(s) relate to.
  + Submit via webcourses. (Not needed this week)

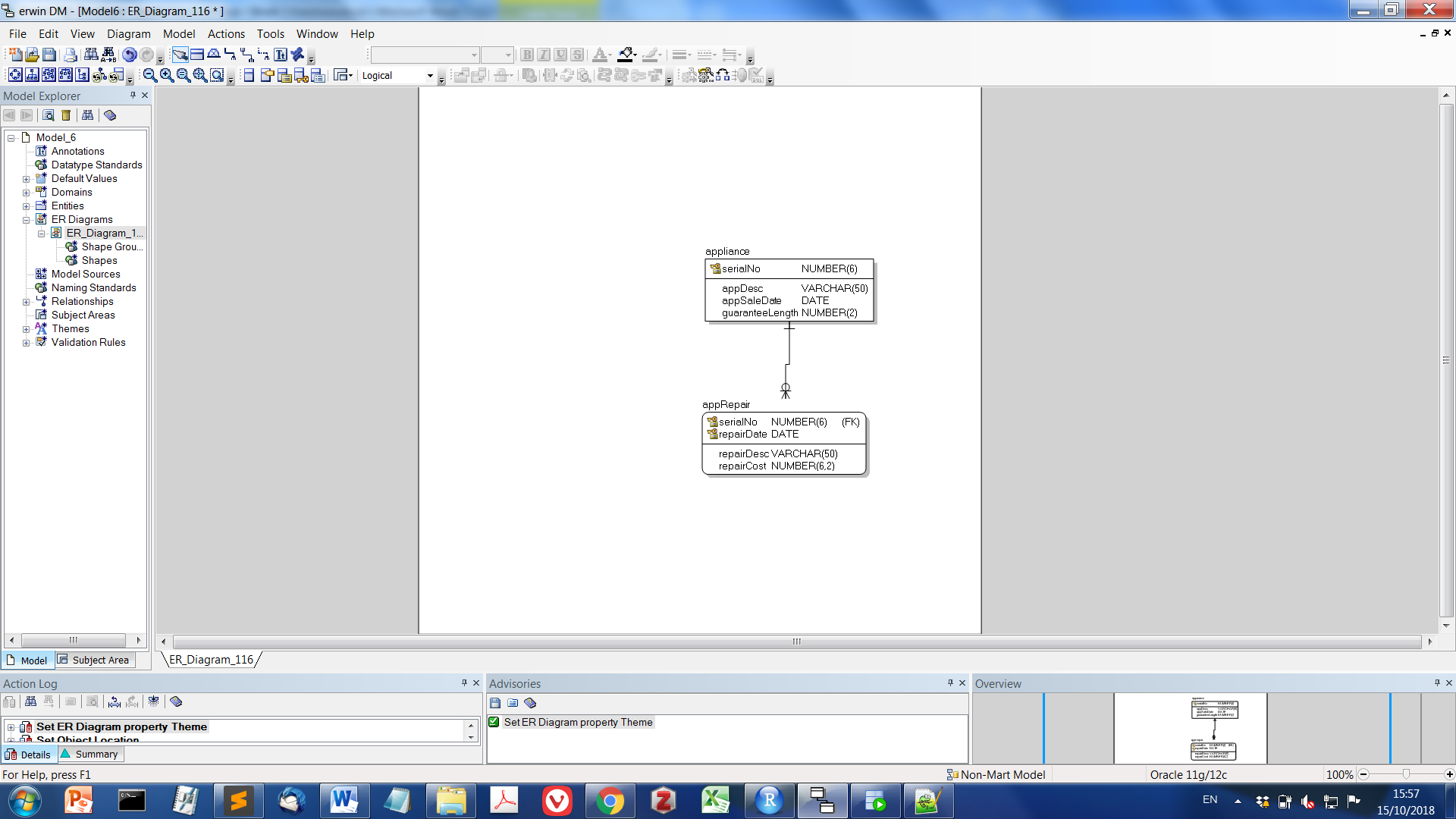
**BE CAREFUL**

* If you are constantly running all your work as a script and not hitting/submitting a commit statement regularly then you will run into problems with your session locking. It WILL NOT BE possible to unlock your sessions for you and you will have to develop some workaround for the lab test.
* If you constantly hit save in the SQLDeveloper window and you have deleted some commands out of the window since you last saved, it will overwrite your file.
  + SQL History – if you do manage to delete some of your work you *may* be able to retrieve it via SQL history (From View Menu -> SQL History). This will open window at the bottom of the screen with all the commands you have input in this session.

Ed’s Repair Shop

Edmund owns a domestic appliance shop called Edmund’s Electrics that sells and repairs domestic appliances (kettles, toasters, dishwashers, fridges, and cookers etc.). Ed only repairs appliances bought in the shop. The following is a relational schema which represents the information Ed’s shop need to store about appliances it repairs. Anything underlined in the schema means that that attribute is part of the primary key of the table in which it appears underlined.

## Relational Schema



appliance(serialNo, appDesc, appSaleDate, guaranteeLength)

Stores for each appliance, the serial no which is unique, a description of the appliance, the purchase date and the length of the guarantee in years. Primary key is serialNo.

appRepair(serialNo, repairDate, repairDesc, repairCost)

Stores for each repair completed, the appliance serialNo, the customer identifier (cId), the date the repair was booked into the shop, a description of the repair and the cost of the repair. Primary key is compound serialNo and repairDate. SerialNo is a foreign key to appliance.

## Data

|  |  |  |  |
| --- | --- | --- | --- |
| **Appliance** | | | |
| **serialNo** | **appDesc** | **appSaleDate** | **guaranteeLength** |
| 9001 | DVD Player | 01 Jan 2018 | 2 |
| 9002 | Fridge Freezer | 31 May 2018 | 5 |
| 9003 | 48’ TV | 12 Jun 2018 | Null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **appRepair** | | | |  |
| **serialNo** | **repairDate** | **repairCost** | **Customer** | **Repair Desc** |
| 9001 | 01 Mar2018 | 67.50 | 1001 | DVD Stuck |
| 9002 | 04 Jun 2018 | 60.70 | 1002 | Constantly Defrosting |
| 9003 | 16 Aug 2016 | 102.50 | 1002 | Blurred Output |

**BEFORE you start**

**Download and run DB5Week5-LabSetup.sql**

This will create the appliance and appRepair tables and populate them with data.

**Your Task**

**NOTE: There will be fewer questions on the actual lab test.**

1. Ed now wants to keep track of customers who bring appliances for repair. A Customer table is needed with the columns and constraints described below. Write the SQL needed to create this table with the constraints described.

CustID, a numeric column which can take values up to 999999 (whole numbers only) which will be a unique identifer. This will be the primary key of the table.

CustName to store the customer name, which should be capable of storing up to 50 characters (either numeric or alphabetic)

CustPhone to store the customer phone number, which should be capable of storing up to 30 characters (either numeric or alphabetic)

CustEmail to store the customer email address, which should be capable of storing up to 30 characters (either numeric or alphabetic), and must contain the @ symbol

A value must be provided for CustName and CustPhone (i.e. they cannot be null).

Note: you need to name your columns as outlined above.

1. Write the SQL needed to Insert the following data into the Customer table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Customer Details** | | | |
| **CustID** | **CustName** | **CustPhone** | **Customer Email** |
| 1001 | 1. Green | 01 402 2849 | [agreen@mail.com](mailto:agreen@mail.com) |
| 1002 | J. Keogh | 02 203 7896 | [jkeogh@mail.com](mailto:jkeogh@mail.com) |
| 1003 | K. Jones | 03 333 8888 | Null |

1. Ed wants to be able to find out for each repair, the customer that requested it. Add a column customer\_id to the appRepair table to store the customer ID.
2. Update the appRepair table so that the value of customer\_id is set as follows:

|  |  |
| --- | --- |
| **Appliance Repairs** | |
| **serialNo** | **customer\_id** |
| 9001 | 1001 |
| 9002 | 1002 |
| 9003 | 1002 |

1. Add a constraint to the appRepair table to ensure consistency between the customer and apprepair table (Foreign Key).
2. Write the SQL to output the name of each customer in uppercase. Change the name of the column in the output to be CUSTOMER NAME.
3. Write the SQL to output the sale date of each appliance formatted DDTH MONTH YY (using TO\_CHAR). Change the name of the column in the output to be SALEDATE.
4. Write the SQL to output the price of each appliance repair formatted to include the local currency symbol and formatted as 5 digits followed by two decimal places. Change the name of the column in the output to be REPAIR COST.
5. Write the SQL to output for each customer their email address or ‘No Email provided’ if the email address is null (use NVL). Change the name of the column in the output to be Email Address.
6. Write the SQL to output for each customer 'Email Provided' if the customer has provided and email address or ‘No Email provided’ if the email address is null (use NVL2).
7. Write the SQL to retrieve for each appliance submitted for repair its serial number and the name and email address of the customer who requested the repair. (JOIN required)
8. Write the SQL to retrieve to retrieve for each appliance submitted for repair its serial number, description, date of sale and the name and email address of the customer who requested the repair. (JOIN to two tables required)
9. Write the SQL to retrieve to retrieve for each appliance submitted for repair its serial number, description, date of sale and the name and email address of the customer who requested the repair. (JOIN to two tables required). Format your output so that the customer name is in uppercase and the date of sale is formatted as DDTH MONTH YY.
10. Write the SQL to retrieve to retrieve for each appliance submitted for repair its serial number, description, date of sale, the name of the customer who requested the repair, the description of the repair plus the cost of the repair. (JOIN to two tables required).

Format the output so that it is output in sentences as follows:

"The repair to appliance 9001 DVD Player, sold on 1ST JANUARY 2018 to A.GREEN, to solve the issue DVD Stuck will cost €67.50".

Ensure that zeros are output if no decimal value exists.