

“Fitzwilliam Institute Group” Course:
'Diploma in ORACLE PL/SQL
programming'.

This document contains a summary of
information for my final project. My
aim: To design a database for handling
stock information for a business
named 'Double Take'(DT). Double
Take redistributes Fibreglass Models
intended for use primarily as props for
Point of Sale, or for events such as film
and theatre.

Double Take

PI SQL class project

Lecturer: Darren Redmond

Helena Rothwell

Table of Contents

CONTACT DETAILS 1

PROJECT BRIEF 2

 Business Profile:..... 2

 Project Specification:..... 2

LIST OF FILES - FOR SUB-PROJECT to demonstrate users and grants 3

 Users and Roles 3

 Technical Steps are:..... 4

LIST OF FILES - Main Project 5

 Technical Steps are:..... 7

ENTITY RELATIONSHIP DIAGRAM 8

CONTACT DETAILS

Helena Rothwell

HelenaRothwell@gmail.com

083 171 3055

PROJECT BRIEF

Business Profile:

‘Double Take’ redistributes Fibreglass Models intended for use primarily as props for Point of Sale, or for events such as film and theatre.

Project Specification:

To design a database for handling stock information ‘Double Take’ (DT) that can manage its stock in an easily searchable form.

To return stock items based on search criteria given, for example: ‘category’.

To log updates to sale price in a separate table.

To include:

• Primary key constraint	Done
• Foreign key constraint	Done
• A sequence	Done
• A cursor	Done
• A view	Done
• A trigger	Done
• 2 procedures	Done
• 2 functions	Done
• A package containing the subprograms	Done
• Use of %type	Done
• Use of %rowtype	Done
• Use of overloading	Done
• Use of a setup.sql file to run project	Done
• Users & Grants	Done

LIST OF FILES - FOR SUB-PROJECT to demonstrate users and grants

Users and Roles

As users and roles statements are run from different user areas of the db, I have created this set of files as a sub project to demonstrate my understanding of them.

In this folder, there are 6 files numbered in order of execution, and separated out depending on which user environment is being employed. Further instructions are listed below the file list.

In folder named – **User_Roles_Files**

Contents:

1_users_and_grants

2_create_table

3_users_and_grants

4_insert_data

5_test_view_only_user

6_drop_everything

readme - Users and Roles - These instructions are also shown here:

Technical Steps are:

run 1_users_and_grants.sql in sys user area.

create new connection dt, dt, dt for (connection name, Username & password)

run 2_create_tables.sql in dt user area (needed to grant permissions on them later)

run 3_users_and_grants.sql in dt user area

create new connections mgr, mgr, mgr (connection name, Username & password)

create new connections guest, guest, guest (connection name, Username & password)

run 4_insert_data.sql in mgr user area (to test permissions for this role and user)

run 5_test_view_only_user.sql in guest user area (to test permissions for this role and user)

run 6_drop_everything.sql in dt user area (to drop tables and sequences created for above tests)

For the rest of the project I start over with new content with all statements run from 'dt' admin area.

LIST OF FILES - Main Project

1) create_sequences

- 3 * Sequences created for each of the tables that have a primary key to increase their id by one with each new entry
- CREATE SEQUENCE category_id_seq
- CREATE SEQUENCE item_id_seq
- CREATE SEQUENCE supplier_id_seq

2) create_tables

- 5 * Tables with constraints, primary keys, foreign keys & not nulls
- CREATE TABLE category
- CREATE TABLE supplier
- CREATE TABLE item
- CREATE TABLE dimensions
- CREATE TABLE availability

3) create_trigger

- 2 * Triggers
- CREATE OR REPLACE TRIGGER print_action --Trigger using case statement to print text output based on type of action performed on item table
- DROP TABLE item_log & CREATE TABLE item_log
- CREATE OR REPLACE TRIGGER log_sale_price_update --Trigger to log to table item_log instances of updates to sale price on item table

4) create_view

- 1 * View
- CREATE VIEW item_name_and_size -- view to show joined information taken from both 'item' and 'dimensions' tables

5) **cursors**

- 1 * Cursor
- CURSOR show_category_contents -- Cursor to loop through table and print to screen items from category code 2.

6) **drop_tables**

- Drops tables, sequences and views for db set up.

7) **insert_item**

- Many insert statements to insert rows into tables using procedures some of which are overloaded, from the project package_dt
- 1 * update sql statement used to demonstrate use of triggers - both print_action and item_log triggers

8) **package_body**

- 3 procedures named add_supplier -OVERLOADED to take either 1,2 or 3 of each of phone numbers & emails.
- 1 procedure named add_category
- 2 procedures named add_item -OVERLOADED to take all parameters or all except an optional second description.
- 1 procedure named add_dimensions
- 2 procedures named add_availability_info - OVERLOADED to take 2 or 4 parameters. (with or without due dates)
- function to show margin on items. Calculates the difference between selling and purchase price
- function to Return number of items that belong to a particular category.

9) **package_spec**

- Contains specification for Package procedures and functions

10) select_stmts

- `SELECT package_dt.calc_diff -- testing calc_diff function using number inputs`
- `SELECT package_dt.calc_diff -- testing calc_diff function using 'item' table inputs`
- `SELECT * FROM item_log; -- show any updates made to the item table sale-price items are reflected in the item_log table`
- `SELECT * FROM item_name_and_size; -- shows contents of 'view' where name is chosen from item table and sizes are chosen from dimensions table`

11) setup_dt

- DT Database set up file which runs all above mentioned statements

Technical Steps are:

Run setup_dt.sql from dt admin area

ENTITY RELATIONSHIP DIAGRAM

