

PL/SQL Extra Assignment

Section 1 : Basics

1. Display the SUBAREA_DESC for the CAT_ID. The CAT_ID values are 1 and 5. The SUBAREA_DESC based on CAT_ID is queried rarely and hence need to be dynamically constructed. Handle the appropriate type of exceptions.
2. A print out of the customer details is required on demand based on the ACCOUNT_NO. The print out should also contain the date and time.
3. The administrator would like to add a new service category. He provides the following information.

COLUMN	VALUE

CAT_TYPE	Installation
CAT_DESC	Installation settings

However, he does not provide the CAT_ID and it has to be automatically generated as a continuation from the previous CAT_ID.

Section 2 : Cursors

4. Display the total number of sub areas of service. Delete the sub areas for the AREA_CODE 102 and display the number of deleted records.
5. Iterate through the Proirity_Range table and transfer information record by record to another table named Priority_Range_Backup.
6. A report has to be generated for the SERVICE_REQUEST with the SR_ID, STATUS and COMMENTS. The status should be displayed as "O" for open, "P" for pending and "C" for closed.

Section 3 : Exceptions

7. Write a PL/SQL block which displays the SUBAREA_DESC from the Service_Subarea for a particular AREA_CODE entered by the user. Ensure that the program does not terminate unexpectedly when there are no records or many records for the given AREA_CODE. The program should handle all errors and display appropriate user defined messages.
8. The administrator is interested in deleting the service owner information from the Service_Owner table when ever required. A module has to be written which allows the deletion and also handles any unnamed exceptions which may arise due to reasons as related records existing in other tables.

Section 4 : Subprograms

9. Write a reusable module named “getOwnerDesignation” which can accept the OWNER_NAME of the service_owner as input and return the OWNER_DESIGNATION. To save memory, the module should have a single parameter to take the OWNER_NAME and return the OWNER_DESIGNATION.
10. Create a module named “getService_RequestInfo” which accepts the SR_ID as input and displays the STATUS, DESIGNATION, SR_DATE, CUST_ID and OWNER_DESIGNATION. The OWNER_DESIGNATION has to be obtained by reusing the “getOwnerDesignation” module.

Section 5 : Packages

11. The developer working on the Pyramid Call Centre project are having several queries on the Service_Request table. They would like to project information by passing different inputs and obtaining information. They would like to develop the “getServiceRequestInfo” which should do the following:
 - a. Accept SR_ID and display SR_DATE, CUST_ID and CAT_ID.
 - b. Accept SR_ID and STATUS to display SR_DATE, CUST_ID and CAT_ID.
 - c. Accept SR_DATE and display the SR_OWNER and SR_PRIORITY

Note : The “getServiceRequestInfo” module should have universal access.

Section 6 : Objects

12. The managers of Pyramid Call Center require the designation for several owners. A reusable module has to be developed which takes in the owner name as parameter and returns the designation of the owner.
13. The managers of Pyramid Call Center require the report of high priority requests that are currently not closed. A reusable module has to be developed which will store the service_request_id, service request_status, sr_owner details of such requests into a table Pending_Requests as below:

Column_name	Column_description
Sr_owner	Primary key
service_request_id and service_request_details	Service request details

A given service owner can be assigned multiple service requests.

a) Write a block that accepts the owner id and display the corresponding service request details from the table Pending_requests.

b) Write a block that accepts the owner_id and service_request_id as parameter and update the status of the corresponding request to 'closed' in the table Pending_requests.

14. Create a table Service_Area_New that will have following columns:

Column_name	Column_description
Cat_id	Primary key
service_area_code and area_desc	Service area details

Under a given category id, there could be multiple service area details.

a) Migrate the data from service_area table into service_area_new table in the required form.

b) Write a module that accepts the cat_id and display the corresponding service area details from the service_area_new table

c) Write a module that accepts cat_id, service_area_code and area_desc as parameter and append the service area details into the service_area_new table for the corresponding existing cat_id.

d) Write a module that accepts cat_id, service_area_code and new_area_desc as parameter. Change the area_desc with new_area_desc of the existing service_area_code for the corresponding existing cat_id of the service_area_new table.

e) Write a module that accepts the cat_id as a parameter and delete the corresponding service_area_details from the service_area_new table.