
SUMMARY

My work was based on 3 tables: Phone_specs, Supplier and Stores.
My submission consists of 3 files:

Phone_specs, Supplier and Stores are part of the tables I've created for the group assignment.
I started my script by granting my user account access to Insert, Update and Delete on corresponding tables. I then populated the tables with values.
Below are a list of the main contents of my submission:

TRANSACTION
My main focus point of my transaction around the user of variables and error handling.

- I start off by setting a transaction name and then declared 3 variables.
- The transaction was set up using the BEGIN and END keywords as a container.
- DBMS_OUTPUT was enabled output the content of my variables via the console.
- I created some save points in order to rollback changes or in the event of an error.
- The initial group of performance was based on the PHONE_SPECS table – I used a variable to view table information
- Using the SUPPLIER table – I used 2 variables to change table information.
- Using the STORE table – I demonstrated error handling by inserting an existing entity into a table column with a unique constraint.

a) PROJECTION
Selected the phone number for Blackrock store

b) SELECTION (Restriction)
Selected the phone brand, type and price of phones

c) **SORTING**
Selected the phone type and price of phones less than 800 from phone specs and sorted them in ascending order.

d) **NON-AGGREGATE FUNCTION IN THE SELECT LIST**
Selected store addresses with less with a store id number less than 5 in length

e) **TESTING FOR NULLs**
Get all phone brands and types with NULL colour input.

f) **Aggregation with GROUP BY and HAVING**
Select phone brand from phone specs and group by phone brand with a max price less than 750.

g) **MINUS, INTERSECTION and UNION**

MINUS - select supplier name from suppliers table, remove any phone brand from phone spec table that's in supplier's name

INTERSECTION - select phone brands from phone specs table and supplier name from supplier table. Show what the same in both tables is

UNION - select phone brands from phone specs table and supplier name from supplier table. Show what's DISTINCT in both tables.

h) **Use of JOIN USING, JOIN ON and JOINS implemented through WHERE clause**

JOIN USING - select all from stores table and join values in customer contract table that's common to stores table (left table) using store id, where the store id is

*the USING keyword allows comparison only of same column names (store_id).

JOIN ON - select all from phone specs table and join all from suppliers table using phone brand on phone specs table that equals supplier name on suppliers table, where the supplier names are Doro or iPhone

*the ON keyword allows comparison of different column names (phone_brand & supplier_name)

i) **USE OF OUTER JOINS**

Select all from phone specs table and join all from suppliers table using phone brand on phone specs table that equals supplier name on suppliers table, where price from phone specs table is greater than 0, order the price in descending order

j) **USE OF SUB-QUERIES**

SUB QUERY - select all from stores where there is a store id sub query select store id of contract no: 1234568

k) **An appropriate query to the role**

Using a serial number to update the colour of a handset.