

PROBLEM SOLVING EXERCISES

Run the script file outdoorDB_v4.sql to load the Outdoor Clubs & Product database to complete the following exercises.

- Ex2B-1. List the PRODUCT_ID, PRODUCT_NAME, and PRICE values from the PRODUCT table.
- Ex2B-2. List the PRODUCT_ID, PRODUCT_NAME, and PRICE values from the PRODUCT table where PRICE is greater than 10.
- Ex2B-3. How many customers are listed in the CUSTOMER table.
- Ex2B-4. List the FIRST_NAME, LAST_NAME for all customers who have placed a product order. (Use subquery).
- Ex2B-5. List the FIRST_NAME, LAST_NAME for all customers who have placed a product order. (Use join).
- Ex2B-6. List the mountain club NAME, DURATION, FIRST_NAME, LAST_NAME for customers who have club membership.
- Ex2B-7. How many product orders are there for each PAYMENT_TYPE.
- Ex2B-8. What is average quantity for each product order.
- Ex2B-9. List all product name which end with word "Pole."
- Ex2B-10. List the customer name (FIRST_NAME, LAST_NAME) along with their club MEMBERSHIP_ID. If a customer does not have club membership, the output should show null.
- Ex2B-11. List all products (PRODUCT_NAME) along with the number of times each product has been re-ordered.
- Ex2B-12. List the SUPPLIER_ID, supplier NAME, and the number of products each supplier has provided.
- Ex2B-13. List the SUPPLIER_ID, supplier NAME, and the number of products each supplier has provided. However, only those suppliers who have supplied more than two products should be listed.
- Ex2B-14. List all products (PRODUCT_ID, PRODUCT_NAME) and their remaining stock (QUANTITY_IN_STOCK - REORDER_POINT).
- Ex2B-15. List all products (PRODUCT_ID, PRODUCT_NAME) and their remaining stock (QUANTITY_IN_STOCK - REORDER_POINT). List only those rows whose remaining stock is more than the average REORDER_QTY value.

- Ex2B-16. Create a database view `PRODUCT_VIEW` containing `PRODUCT_NAME`, `PRICE`, supplier `NAME`, and supplier `CITY` attributes.
- Ex2B-17. Query the previously created database view `PRODUCT_VIEW` to list all `PRODUCT_NAME`, and supplier `NAME` located in Los Angeles.
- Ex2B-18. Create a database view `CUSTOMER_VIEW` containing `FIRST_NAME`, `LAST_NAME`, `ORDER_ID`, `PRODUCT_NAME`, `QUANTITY` attributes.
- Ex2B-19. Query the previously created database view `CUSTOMER_VIEW` to list those customers (`FIRST_NAME`, `LAST_NAME`) who have ordered more than two products.
- Ex2B-20. Query the previously created database view `CUSTOMER_VIEW` to list average quantity per `PRODUCT_NAME`.

Run the script file `superflexDB_v4.sql` to load the Superflex Apartment database to complete the following exercises.

- Ex2B-21. List the name of each child that is in a rental unit.
- Ex2B-22. List the `INVOICE_NO` for each invoice from November 2007.
- Ex2B-23. List the `APT_NO` for each apartment that is currently vacant.
- Ex2B-24. For each apartment, list the apartment number and rental amount. Order this in descending order based on rental amount.
- Ex2B-25. For each `AUTO_MAKE`, list the number (count) for each make.
- Ex2B-26. List the length of the lease for each rental.
- Ex2B-27. For each apartment (`APT_NO`), list the name of each tenant.
- Ex2B-28. List the number of payments made on each credit card, including the credit card type.
- Ex2B-29. List the number of automobiles for each `RENTAL_NO` and `APT_NO`.
- Ex2B-30. List the amount of rent collected per month if all apartments are rented (and the tenants pay their rent).
- Ex2B-31. List the amount of rent collected based on the apartment that are rented (and the tenants pay their rent).
- Ex2B-32. List the `APT_NO` for the most expensive apartments.
- Ex2B-33. List the number of complaints by `RENTAL_NO`.
- Ex2B-34. List the `RENTAL_NO` that has the most complaints.
- Ex2B-35. List the `TENANT_NAME` of the oldest tenant.

Ex2B-36. List the number of tenants in each apartment. Make sure you also count the tenant and include any empty apartments.

Ex2B-37. List the name of any member of the tenant family that is older than the tenant itself.