Sql Tasks

**1.1 Select product names:**

SQL

SELECT name FROM products;

**1.2 Select names and prices:**

SQL

SELECT name, price FROM products;

**1.3 Select products with price <= $200:**

SQL

SELECT name FROM products WHERE price <= 200;

**1.4 Select products with price between $60 and $120:**

SQL

SELECT name FROM products WHERE price BETWEEN 60 AND 120;

**1.5 Select name and price in cents:**

SQL

SELECT name, price \* 100 AS price\_in\_cents FROM products;

**1.6 Average price of all products:**

SQL

SELECT AVG(price) AS average\_price FROM products;

**1.7 Average price of products with manufacturer code 2:**

SQL

SELECT AVG(price) AS average\_price FROM products WHERE manufacturer\_code = 2;

**1.8 Count products with price >= $180:**

SQL

SELECT COUNT(\*) AS product\_count FROM products WHERE price >= 180;

**1.9 Sort expensive products:**

SQL

SELECT name, price FROM products WHERE price >= 180

ORDER BY price DESC, name ASC;

**1.10 Select products with manufacturer data:**

This requires joining the products table with the manufacturers table (assuming separate tables).

SQL

SELECT p.name AS product\_name, p.price, m.name AS manufacturer\_name

FROM products p

JOIN manufacturers m ON p.manufacturer\_code = m.code;

**1.11 Select name, price, and manufacturer name:**

SELECT p.name AS product\_name, p.price, m.name AS manufacturer\_name

FROM products p

JOIN manufacturers m ON p.manufacturer\_code = m.code;

**1.12 Average price per manufacturer (code):**

SQL

SELECT manufacturer\_code, AVG(price) AS average\_price

FROM products

GROUP BY manufacturer\_code;

**1.13 Average price per manufacturer (name):**

SQL

SELECT m.name AS manufacturer\_name, AVG(p.price) AS average\_price

FROM products p

JOIN manufacturers m ON p.manufacturer\_code = m.code

GROUP BY m.name;

**1.14 Select manufacturers with average price >= $150:**

SQL

SELECT m.name AS manufacturer\_name

FROM products p

JOIN manufacturers m ON p.manufacturer\_code = m.code

GROUP BY m.name

HAVING AVG(p.price) >= 150;

**1.15 Select cheapest product:**

SQL

SELECT name, price FROM products

ORDER BY price ASC

LIMIT 1;

**1.16 Most expensive product per manufacturer:**

This requires a subquery to find the maximum price per manufacturer.

SQL

SELECT m.name AS manufacturer\_name, p.name AS product\_name, p.price

FROM products p

JOIN manufacturers m ON p.manufacturer\_code = m.code

INNER JOIN (

SELECT manufacturer\_code, MAX(price) AS max\_price

FROM products

GROUP BY manufacturer\_code

) AS max\_prices ON p.manufacturer\_code = max\_prices.manufacturer\_code

AND p.price = max\_prices.max\_price;

**1.17 Add a new product:**

This depends on your database system, but it typically involves an INSERT statement with product details.

**1.18 Update product name:**

This depends on your database system, but it typically involves an UPDATE statement with the new name and product identifier.

**1.19 Apply 10% discount:**

This can be achieved by updating the price column with a formula involving a discount. The specific syntax depends on your database system.

**1.20 Apply 10% discount (price >= $120):**

This requires an UPDATE statement with a conditional clause to filter products based on price. The specific syntax depends on your database system.