

Practical 1: SQL Fundamentals

1. SELECT Statement

1. SELECT * FROM retailsales;
2. SELECT
Transaction_ID,
date,
Customer_id,
FROM retailsales;

2. SELECT DISTINCT statement

3. SELECT DISTINCT product_category AS category
FROM retailsales;
4. SELECT DISTINCT gender
FROM retailsales;

3. WHERE Clause

5. SELECT * FROM retailsales
WHERE age > 40;
6. SELECT * FROM retailsales
WHERE price_per_unit BETWEEN 100 AND 500;
7. SELECT * FROM retailsales
WHERE product_category = 'Clothing' OR
product_category = 'Electronics';
8. SELECT * FROM retailsales
WHERE NOT product_category = 'Clothing';
9. SELECT * FROM retailsales
WHERE quantity >= 3;

4. Aggregate Functions

10. SELECT COUNT(*) AS total_transactions
FROM retailsales;
11. SELECT AVG(age) FROM retailsales; /
SELECT ROUND(AVG(age), 0) Average age
FROM retailsales;

12. SELECT SUM (quantity) AS total-quantity
FROM retailsales;

13. SELECT MAX (total-amount)
FROM retailsales;

14. SELECT MIN (price-per-unit)
FROM retailsales;

5. GROUP BY statement

15. SELECT COUNT (DISTINCT transaction_id) AS ~~transaction_id~~
Product-category
FROM retailsales GROUP BY 2;

16. SELECT SUM (total-amount) total-revenue,
gender
FROM retailsales GROUP BY gender;

17. SELECT SUM (price-per-unit) average-price,
Product-category
FROM retailsales GROUP BY gender;

6. HAVING clause

18. SELECT SUM (total-amount) AS total revenue,
Product-category
FROM retailsales
GROUP BY Product-category
HAVING SUM (total-amount) > 10000;

~~18. SELECT ROUND (~~

19. SELECT ROUND (AVG (quantity), 0) AS Average-quantity,
Product-category
FROM retailsales
GROUP BY Product-category
HAVING ROUND (AVG (quantity), 0) > 2;

7. CASE Statement.

```
20. SELECT Transaction-id,  
       total-amount,  
       CASE WHEN total-amount > 10000 THEN 'High'  
       ELSE 'Low'  
       END AS spending-level  
       FROM retailsales;
```

```
21. SELECT customer-id,  
       age,  
       CASE WHEN age > 30 THEN 'Youth'  
       WHEN age BETWEEN 50 AND 59 THEN 'Adult'  
       ELSE 'Senior'  
       END AS Age-group  
       FROM retailsales;
```