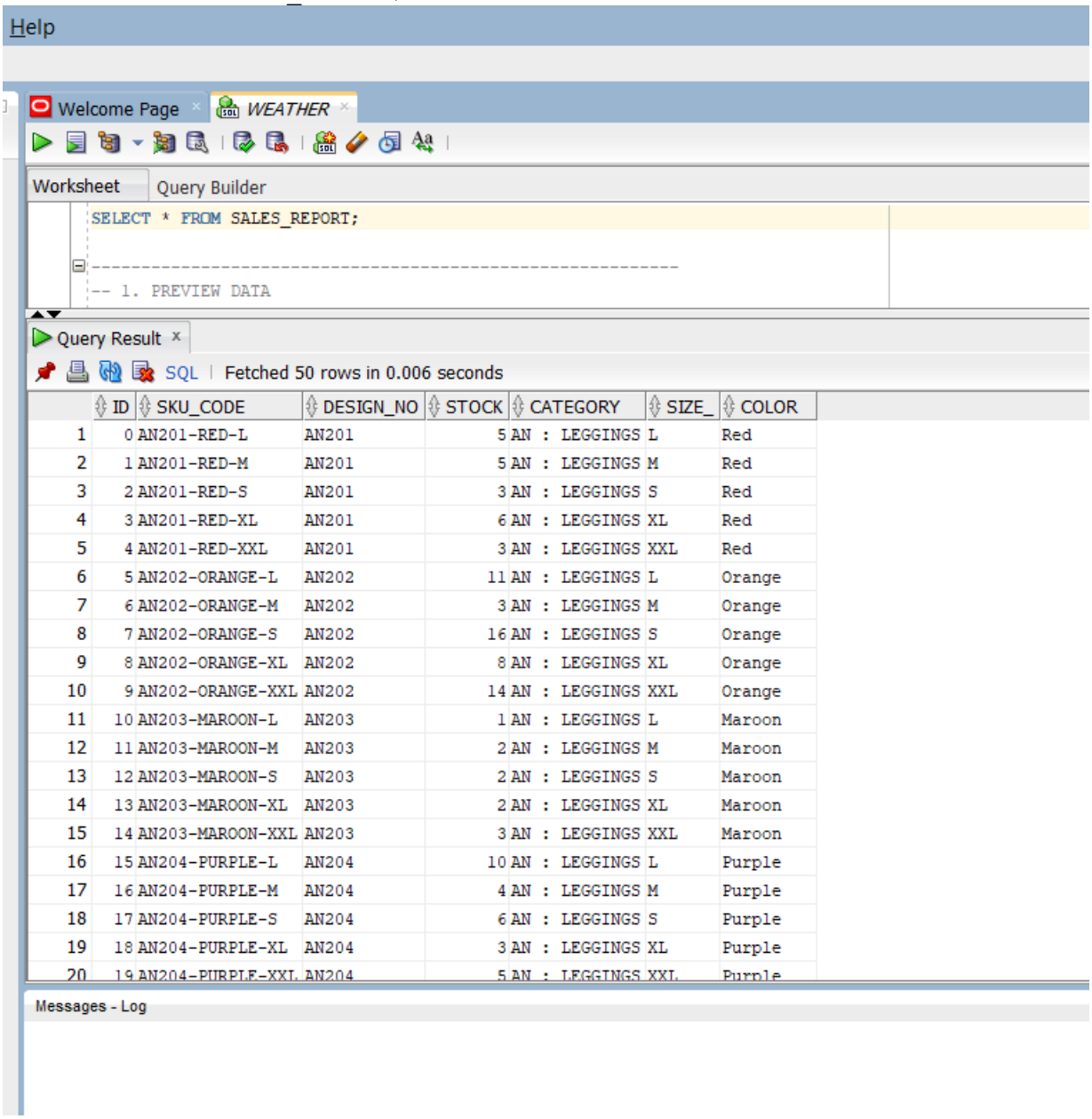
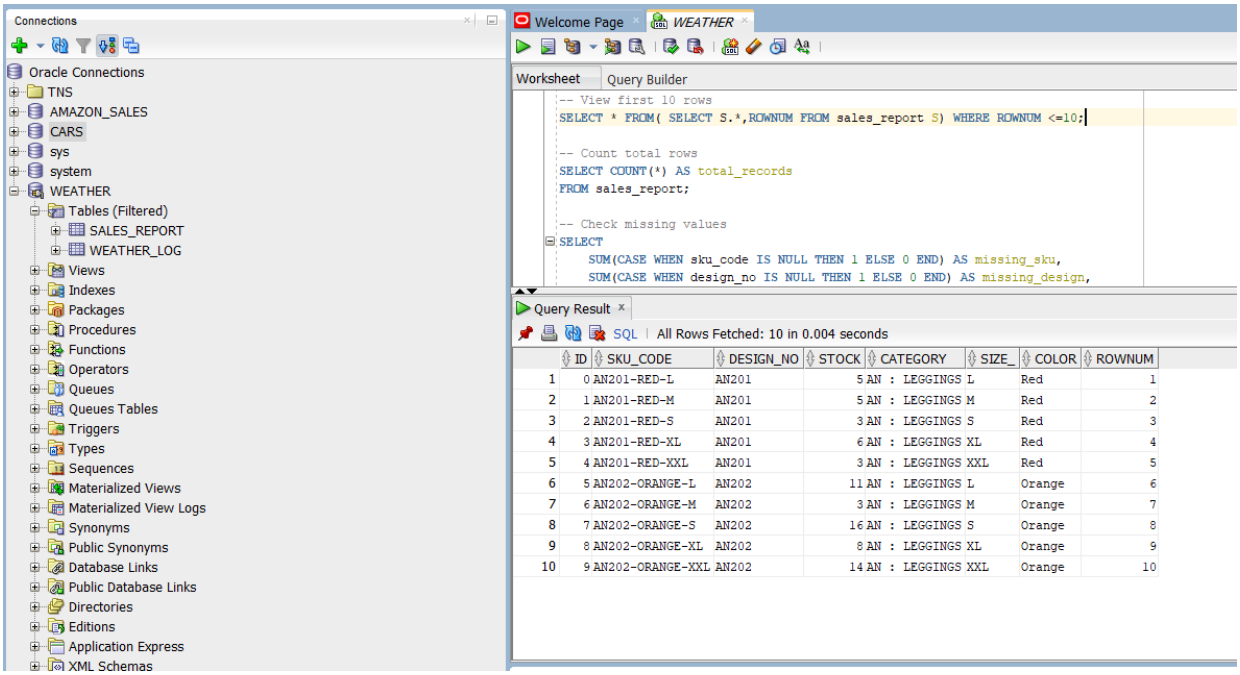
```
SELECT * FROM SALES_REPORT;
```

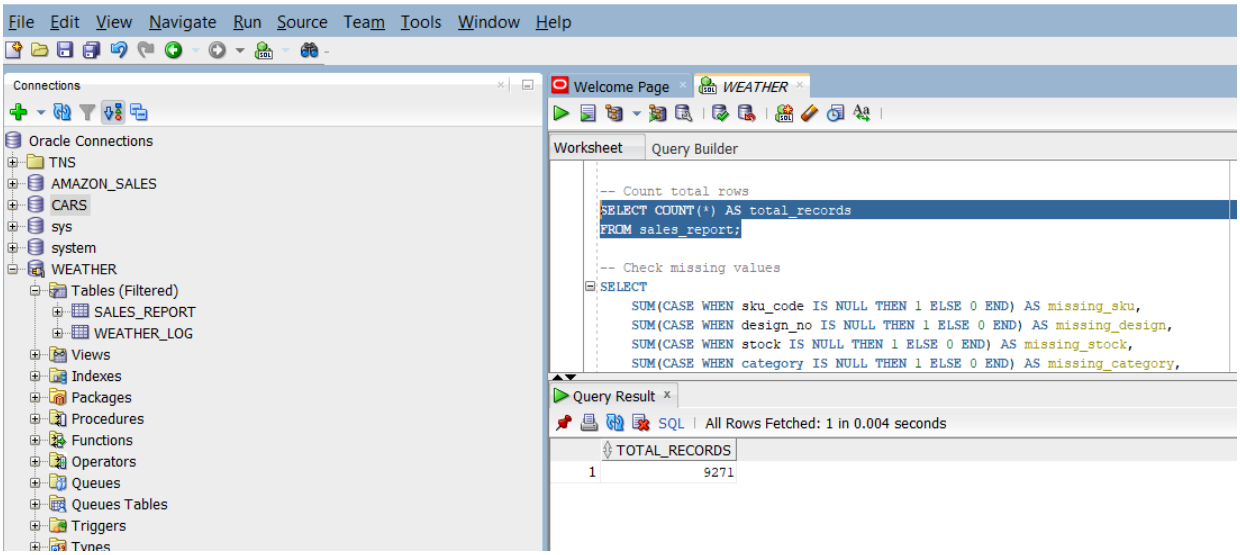


	ID	SKU_CODE	DESIGN_NO	STOCK	CATEGORY	SIZE_	COLOR
1	0	AN201-RED-L	AN201	5	AN : LEGGINGS	L	Red
2	1	AN201-RED-M	AN201	5	AN : LEGGINGS	M	Red
3	2	AN201-RED-S	AN201	3	AN : LEGGINGS	S	Red
4	3	AN201-RED-XL	AN201	6	AN : LEGGINGS	XL	Red
5	4	AN201-RED-XXL	AN201	3	AN : LEGGINGS	XXL	Red
6	5	AN202-ORANGE-L	AN202	11	AN : LEGGINGS	L	Orange
7	6	AN202-ORANGE-M	AN202	3	AN : LEGGINGS	M	Orange
8	7	AN202-ORANGE-S	AN202	16	AN : LEGGINGS	S	Orange
9	8	AN202-ORANGE-XL	AN202	8	AN : LEGGINGS	XL	Orange
10	9	AN202-ORANGE-XXL	AN202	14	AN : LEGGINGS	XXL	Orange
11	10	AN203-MAROON-L	AN203	1	AN : LEGGINGS	L	Maroon
12	11	AN203-MAROON-M	AN203	2	AN : LEGGINGS	M	Maroon
13	12	AN203-MAROON-S	AN203	2	AN : LEGGINGS	S	Maroon
14	13	AN203-MAROON-XL	AN203	2	AN : LEGGINGS	XL	Maroon
15	14	AN203-MAROON-XXL	AN203	3	AN : LEGGINGS	XXL	Maroon
16	15	AN204-PURPLE-L	AN204	10	AN : LEGGINGS	L	Purple
17	16	AN204-PURPLE-M	AN204	4	AN : LEGGINGS	M	Purple
18	17	AN204-PURPLE-S	AN204	6	AN : LEGGINGS	S	Purple
19	18	AN204-PURPLE-XL	AN204	3	AN : LEGGINGS	XL	Purple
20	19	AN204-PURPLE-XXL	AN204	5	AN : LEGGINGS	XXL	Purple

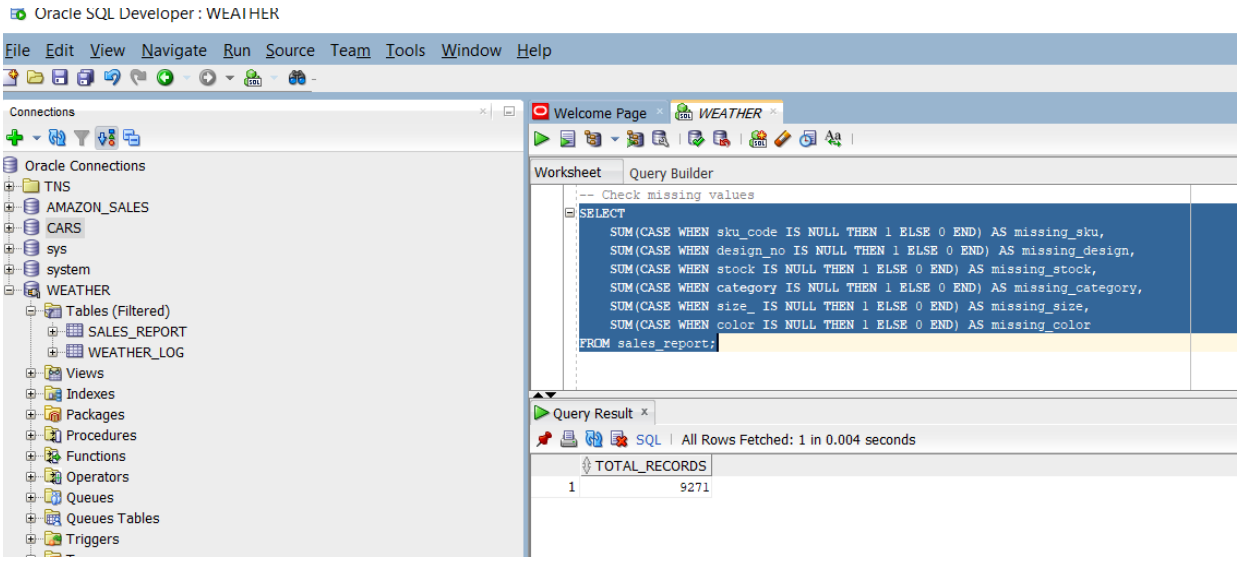
```
------------------------------------------------------------
-- 1. PREVIEW DATA
------------------------------------------------------------

-- View first 10 rows
SELECT * FROM( SELECT S.*,ROWNUM FROM sales_report S) WHERE ROWNUM <=10;
```



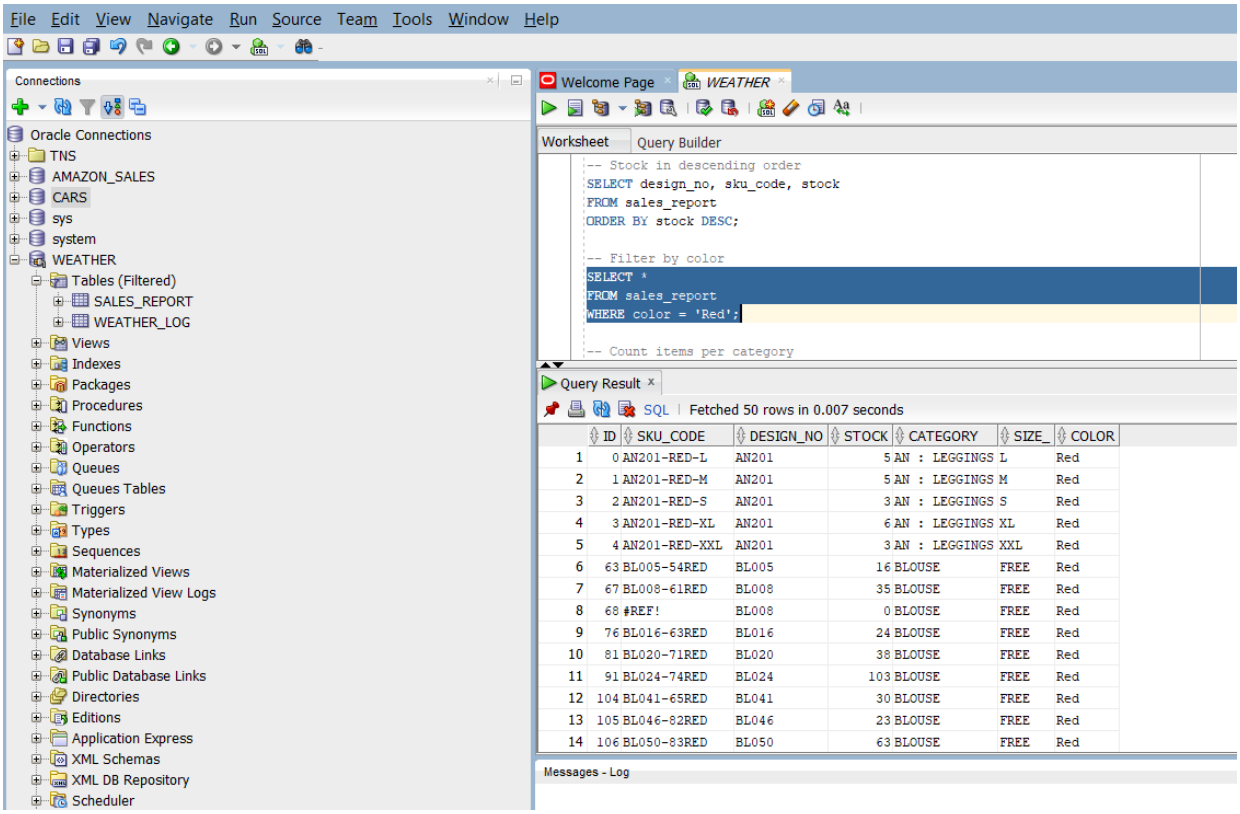

```
-- Count total rows
SELECT COUNT(*) AS total_records
FROM sales_report;
```




```
-- Check missing values
SELECT
    SUM(CASE WHEN sku_code IS NULL THEN 1 ELSE 0 END) AS missing_sku,
    SUM(CASE WHEN design_no IS NULL THEN 1 ELSE 0 END) AS missing_design,
    SUM(CASE WHEN stock IS NULL THEN 1 ELSE 0 END) AS missing_stock,
    SUM(CASE WHEN category IS NULL THEN 1 ELSE 0 END) AS
missing_category,
    SUM(CASE WHEN size_ IS NULL THEN 1 ELSE 0 END) AS missing_size,
    SUM(CASE WHEN color IS NULL THEN 1 ELSE 0 END) AS missing_color
FROM sales_report;
```

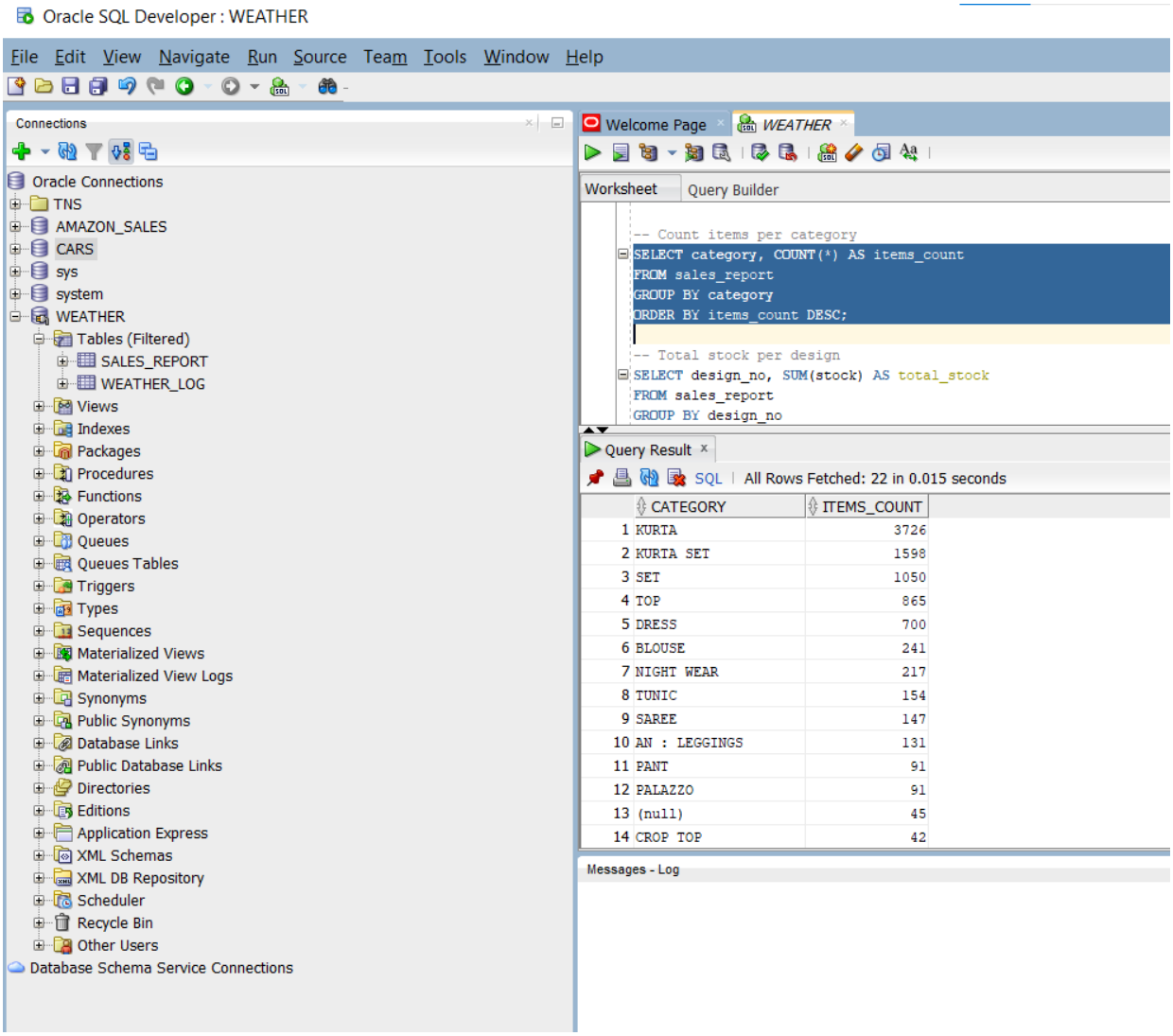



```
-------------------------------------------------------------
-- 2. BASIC SELECT, WHERE, ORDER BY, GROUP BY
-------------------------------------------------------------

-- Stock in descending order
SELECT design_no, sku_code, stock
FROM sales_report
ORDER BY stock DESC;
```

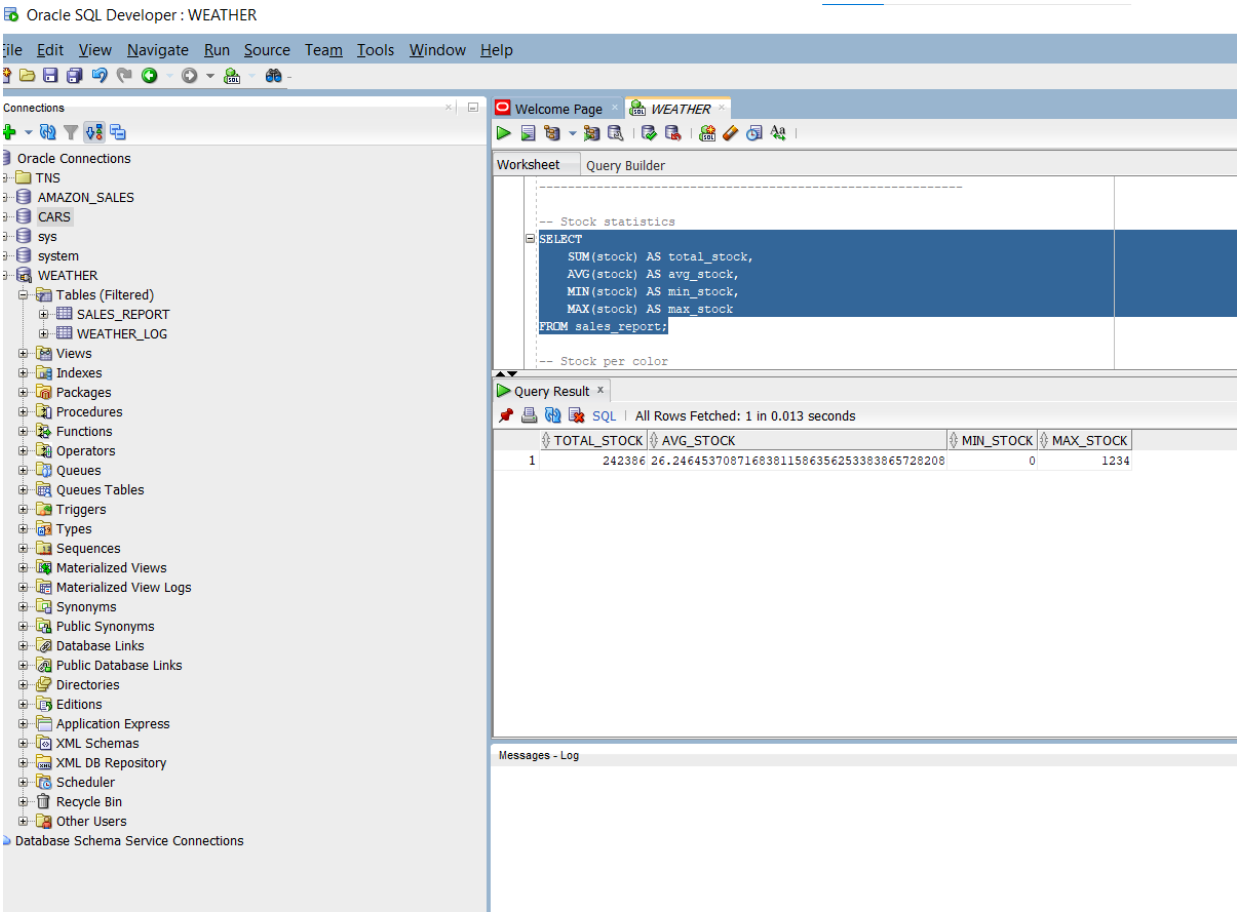



```
-- Filter by color
SELECT *
FROM sales_report
```

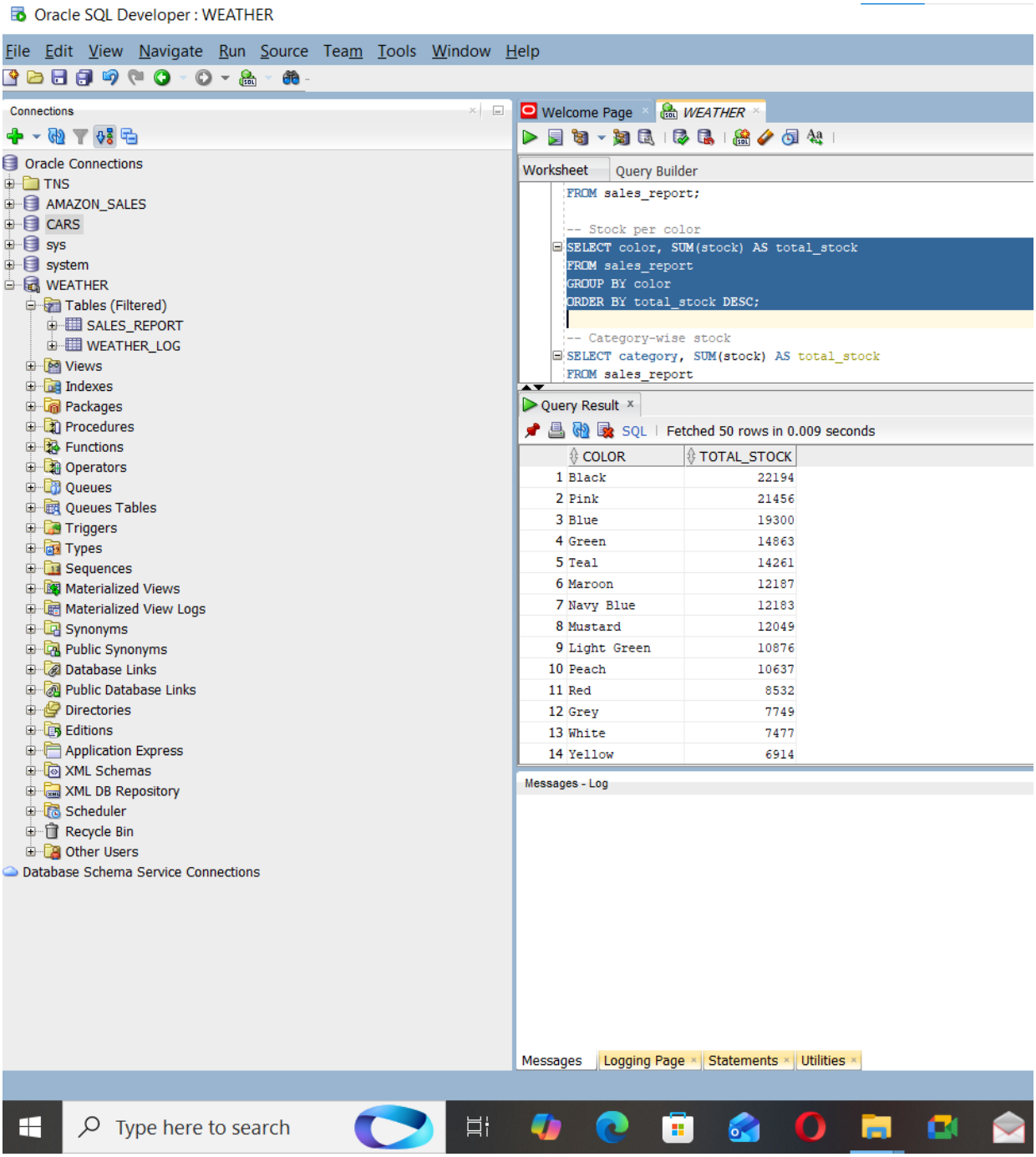
```
WHERE color = 'Red';

-- Count items per category
SELECT category, COUNT(*) AS items_count
FROM sales_report
GROUP BY category
ORDER BY items_count DESC;
```

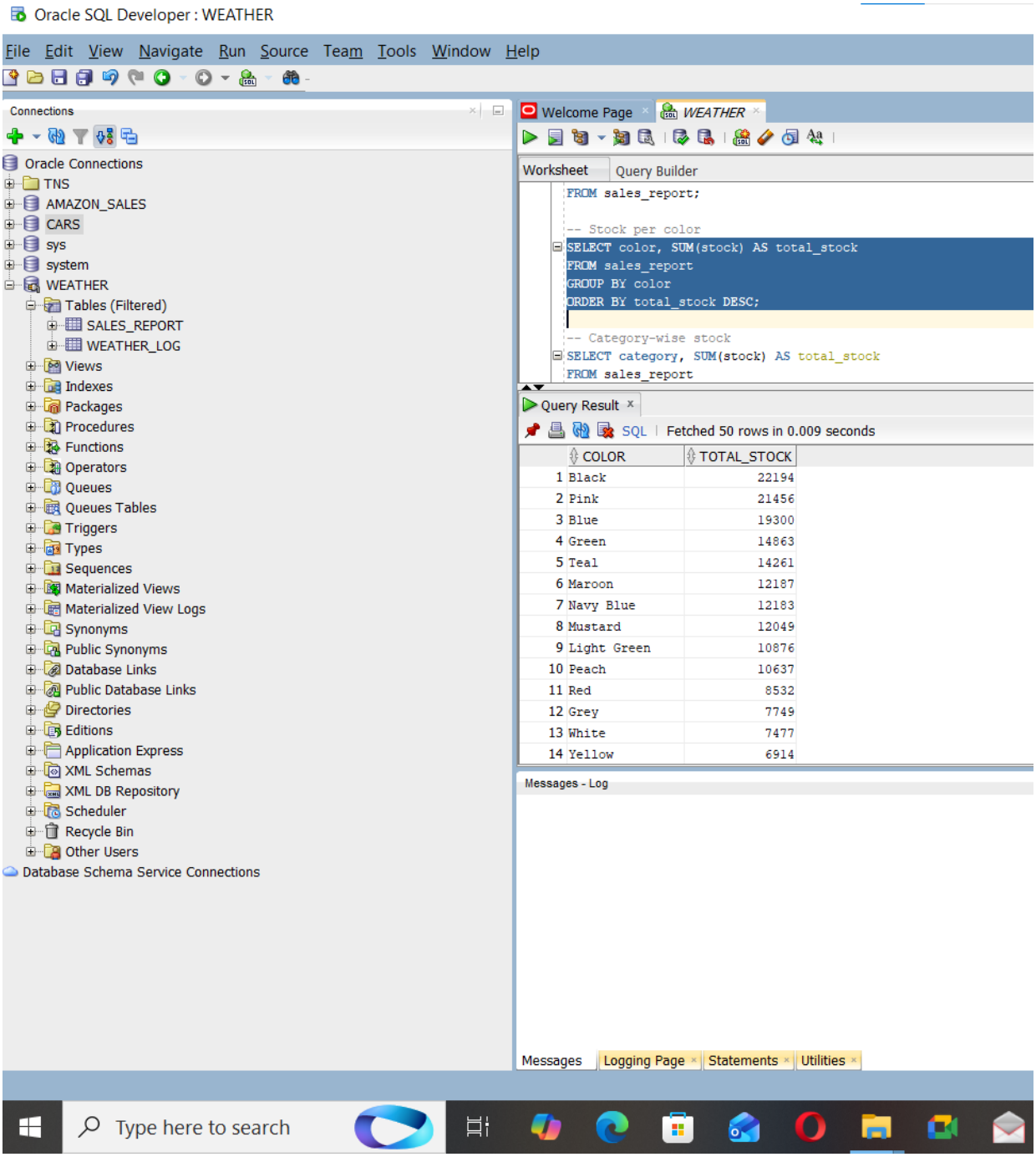



```
-- Total stock per design
SELECT design_no, SUM(stock) AS total_stock
FROM sales_report
GROUP BY design_no
ORDER BY total_stock DESC;
```



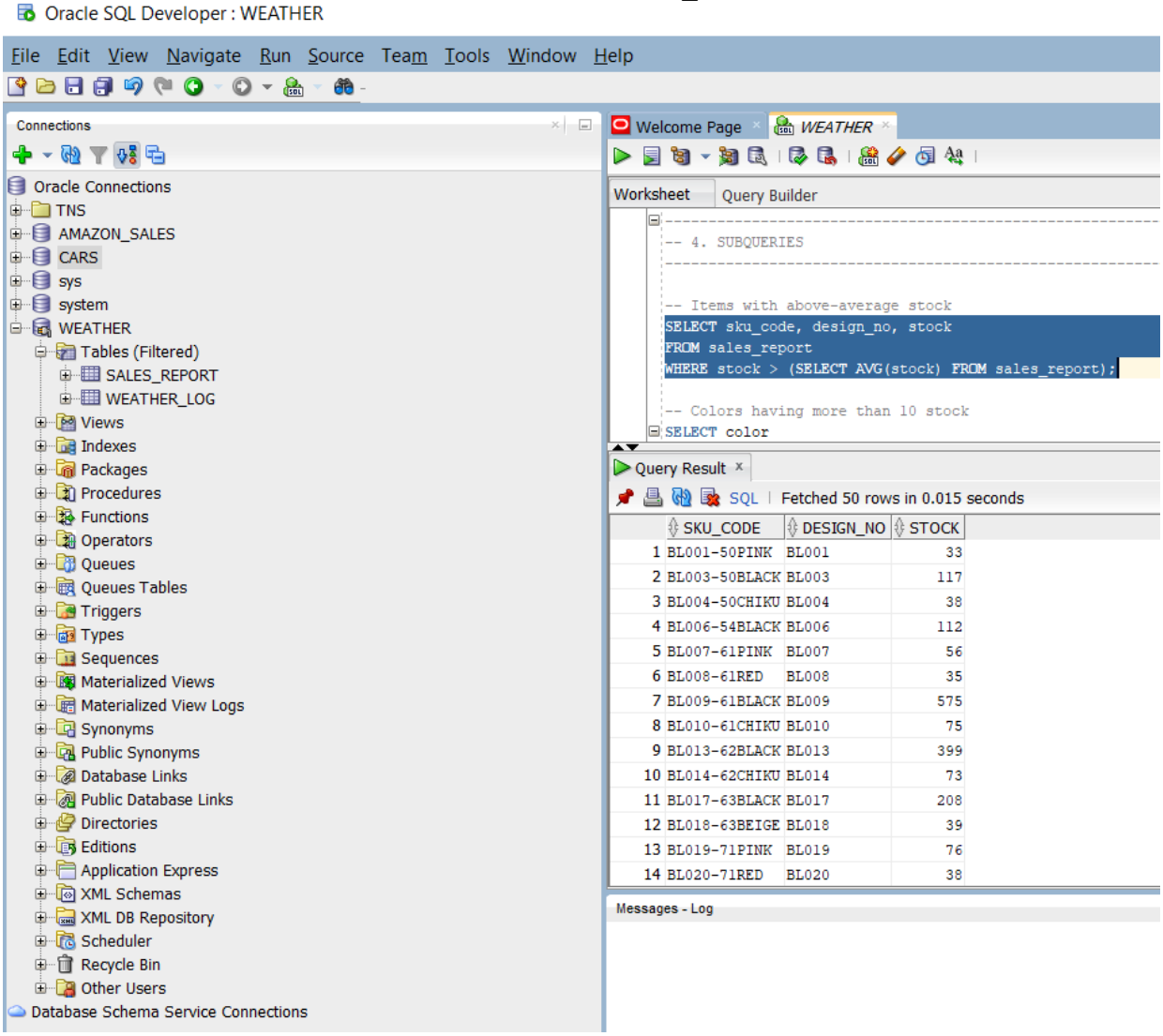

```
-------------------------------------------------------------
-- 3. AGGREGATE ANALYSIS
-------------------------------------------------------------

-- Stock statistics
SELECT
    SUM(stock) AS total_stock,
    AVG(stock) AS avg_stock,
    MIN(stock) AS min_stock,
    MAX(stock) AS max_stock
FROM sales_report;


-- Stock per color
SELECT color, SUM(stock) AS total_stock
FROM sales_report
GROUP BY color
ORDER BY total_stock DESC;
```


Oracle SQL Developer : WEATHER
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
TNS
AMAZON_SALES
CARS
sys
system
WEATHER
Tables (Filtered)
SALES_REPORT
WEATHER_LOG
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Materialized Views
Materialized View Logs
Synonyms
Public Synonyms
Database Links
Public Database Links
Directories
Editions
Application Express
XML Schemas
XML DB Repository
Scheduler
Recycle Bin
Other Users
Database Schema Service Connections
Welcome Page
WEATHER
Worksheet
Query Builder
FROM sales_report;
-- Stock per color
SELECT color, SUM(stock) AS total_stock
FROM sales_report
GROUP BY color
ORDER BY total_stock DESC;
-- Category-wise stock
SELECT category, SUM(stock) AS total_stock
FROM sales_report
Query Result
SQL | Fetched 50 rows in 0.009 seconds
COLOR TOTAL_STOCK
1 Black 22194
2 Pink 21456
3 Blue 19300
4 Green 14863
5 Teal 14261
6 Maroon 12187
7 Navy Blue 12183
8 Mustard 12049
9 Light Green 10876
10 Peach 10637
11 Red 8532
12 Grey 7749
13 White 7477
14 Yellow 6914
Messages - Log
Messages
Logging Page
Statements
Utilities
Type here to search

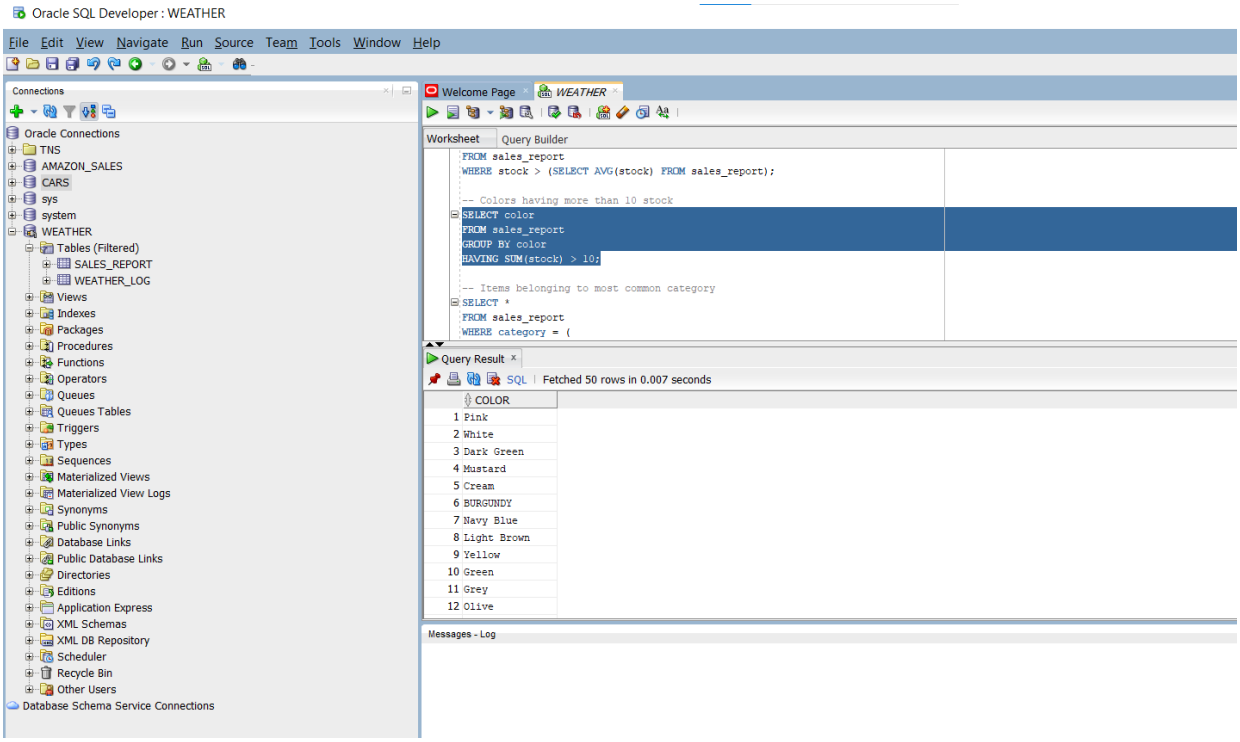



```
-- Category-wise stock
SELECT category, SUM(stock) AS total_stock
FROM sales_report
GROUP BY category
ORDER BY total_stock DESC;


------------------------------------------------------------
-- 4. SUBQUERIES
------------------------------------------------------------

-- Items with above-average stock
SELECT sku_code, design_no, stock
FROM sales_report
```

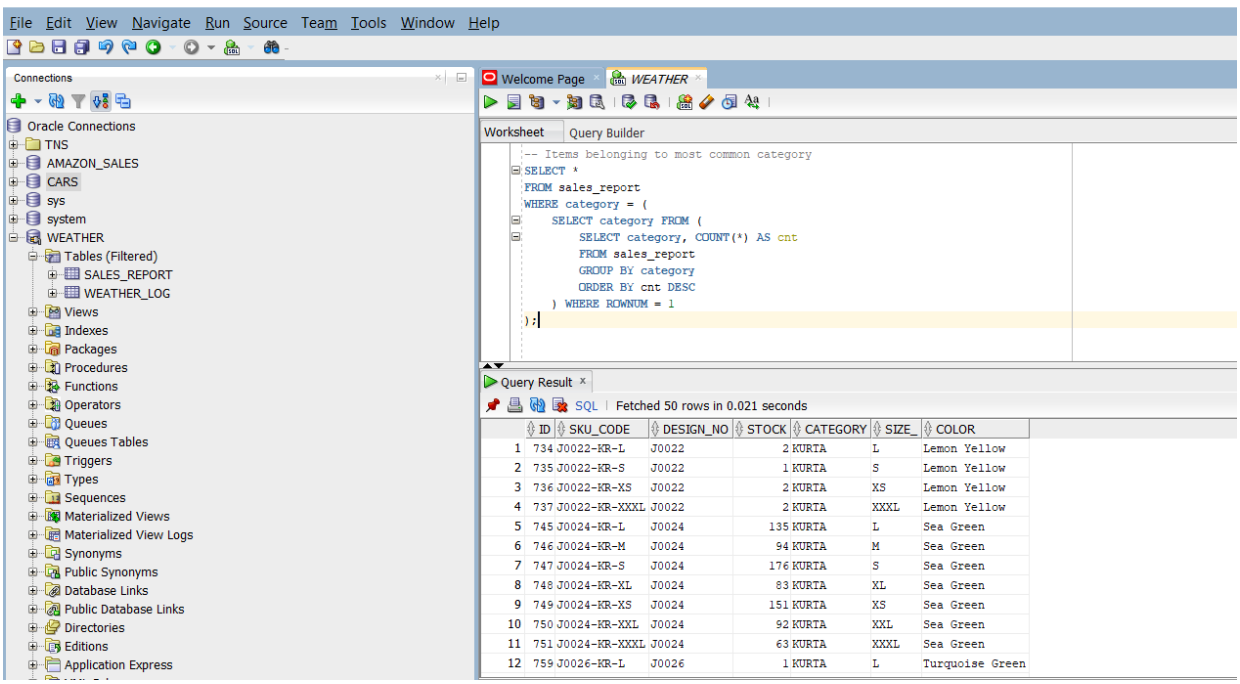
```
WHERE stock > (SELECT AVG(stock) FROM sales_report);
```



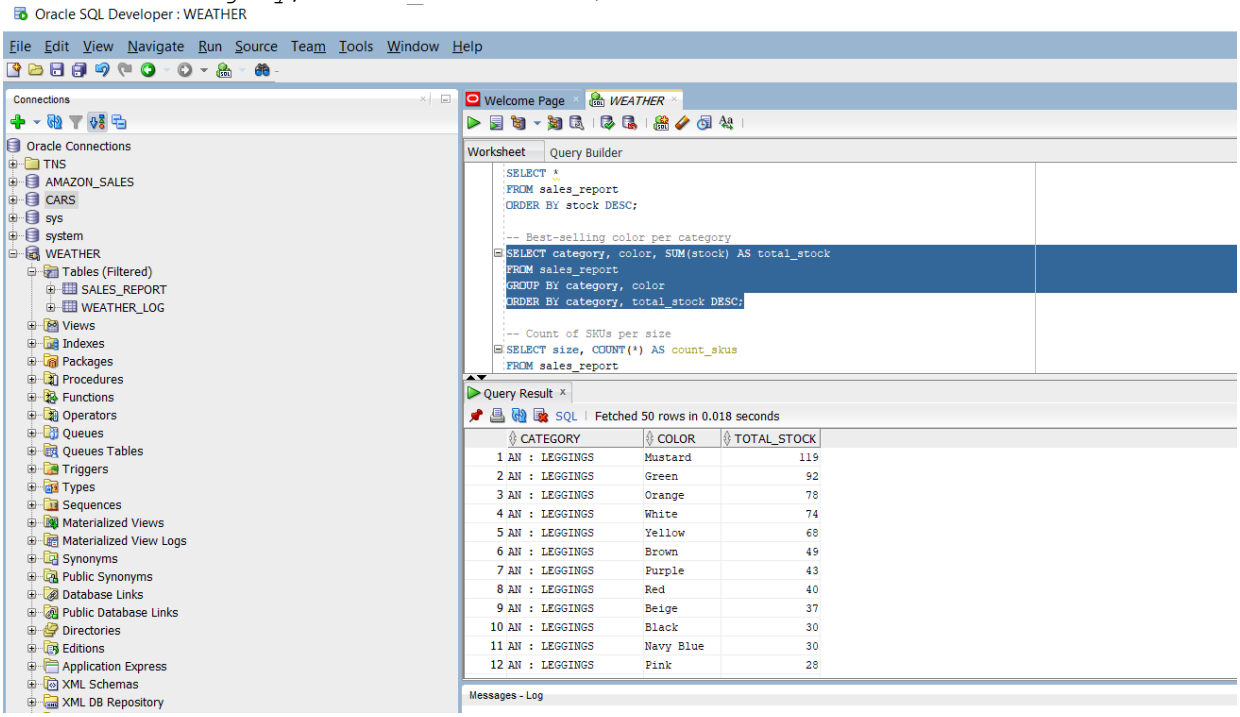

```
-- Colors having more than 10 stock
SELECT color
FROM sales_report
GROUP BY color
HAVING SUM(stock) > 10;
```



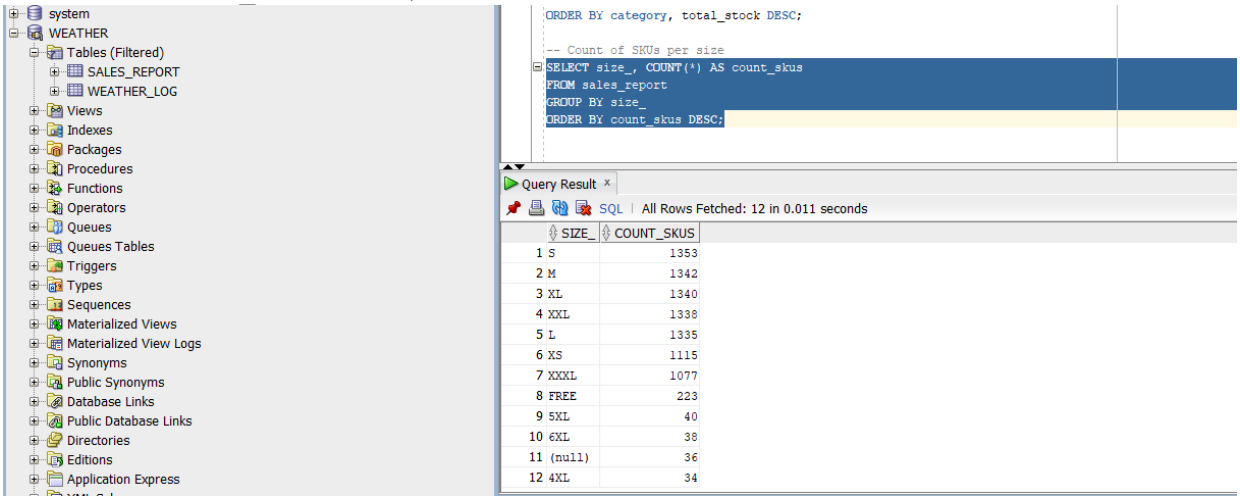

```
-- Items belonging to most common category
SELECT *
FROM sales_report
WHERE category = (
    SELECT category FROM (
        SELECT category, COUNT(*) AS cnt
        FROM sales_report
        GROUP BY category
        ORDER BY cnt DESC
    ) WHERE ROWNUM = 1
);
```



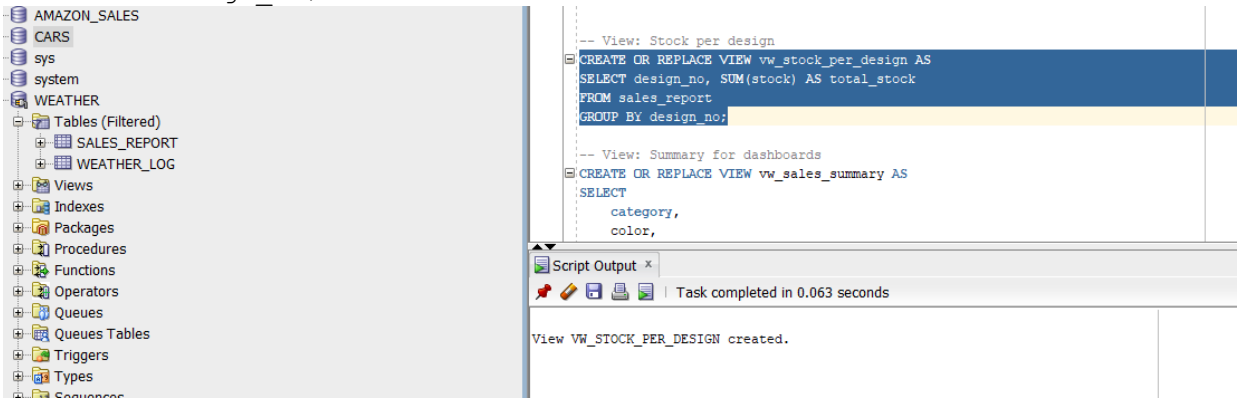
```
-------------------------------------------------------------
-- 6. ADVANCED PATTERNS
-------------------------------------------------------------

-- Top 5 SKUs by stock
SELECT *
FROM sales_report
ORDER BY stock DESC;

-- Best-selling color per category
SELECT category, color, SUM(stock) AS total_stock
FROM sales_report
GROUP BY category, color
ORDER BY category, total_stock DESC;
```



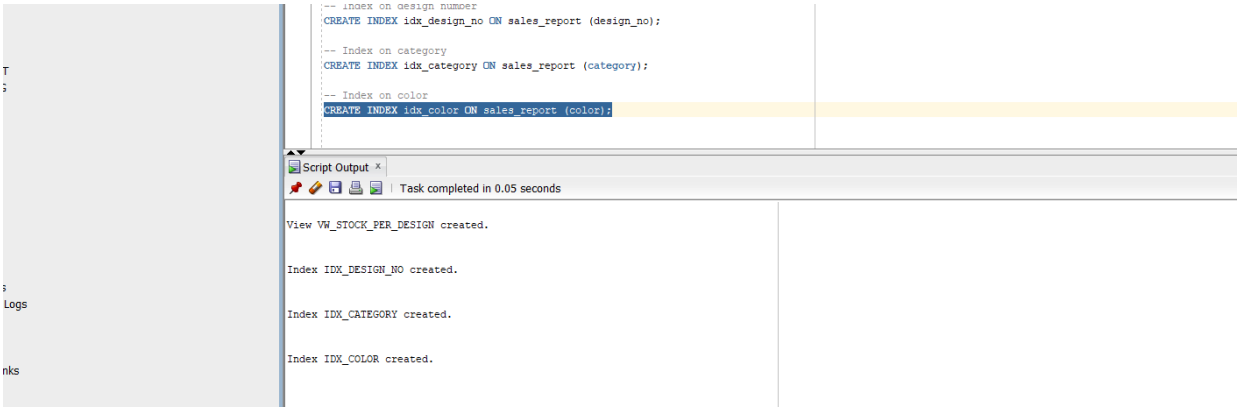

```
-- Count of SKUs per size
SELECT size_, COUNT(*) AS count_skus
FROM sales_report
GROUP BY size_
ORDER BY count_skus DESC;
```



```
-------------------------------------------------------------
-- 7. VIEWS FOR ANALYSIS
-------------------------------------------------------------

-- View: Stock per design
CREATE OR REPLACE VIEW vw_stock_per_design AS
SELECT design_no, SUM(stock) AS total_stock
FROM sales_report
GROUP BY design_no;
```




```
-- View: Summary for dashboards
CREATE OR REPLACE VIEW vw_sales_summary AS
SELECT
    category,
    color,
    size,
    SUM(stock) AS total_stock,
    COUNT(*) AS sku_count
FROM sales_report
GROUP BY category, color, size;


-------------------------------------------------------------
-- 8. PERFORMANCE OPTIMIZATION (INDEXES)
-------------------------------------------------------------

-- Index on design number
CREATE INDEX idx_design_no ON sales_report (design_no);

-- Index on category
CREATE INDEX idx_category ON sales_report (category);

-- Index on color
CREATE INDEX idx_color ON sales_report (color);
```



-- END OF FULL SQL ANALYSIS SCRIPT
