

```
SELECT * FROM SALES_REPORT;
```

Help

The screenshot shows the Oracle SQL Developer interface. At the top, there's a toolbar with various icons. Below it is a menu bar with 'Worksheet' and 'Query Builder' tabs. The main area has two panes: one for the query and one for the results.

In the 'Worksheet' pane, the query `SELECT * FROM SALES_REPORT;` is entered. Below the query, a note says `-- 1. PREVIEW DATA`.

In the 'Query Result' pane, the output is displayed in a grid format. The columns are labeled `ID`, `SKU_CODE`, `DESIGN_NO`, `STOCK`, `CATEGORY`, `SIZE_`, and `COLOR`. The data consists of 20 rows, each representing a different product variant:

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

Below the result grid, there's a 'Messages - Log' section which is currently empty.

-- 1. PREVIEW DATA

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

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows several database connections, including TNS, AMAZON_SALES, CARS, sys, system, and WEATHER. The WEATHER connection is expanded, showing Tables (Filtered), 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, and XML Schemas.

In the center, the Worksheet pane contains the following SQL code:

```
-- 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,
```

The Query Result pane shows the output of the first query:

ID	SKU_CODE	DESIGN_NO	STOCK	CATEGORY	SIZE_	COLOR	ROWNUM
1	0 AN201-RED-L	AN201	5 AN :	LEGGINGS L	Red		1
2	1 AN201-RED-M	AN201	5 AN :	LEGGINGS M	Red		2
3	2 AN201-RED-S	AN201	3 AN :	LEGGINGS S	Red		3
4	3 AN201-RED-XL	AN201	6 AN :	LEGGINGS XL	Red		4
5	4 AN201-RED-XXL	AN201	3 AN :	LEGGINGS XXL	Red		5
6	5 AN202-ORANGE-L	AN202	11 AN :	LEGGINGS L	Orange		6
7	6 AN202-ORANGE-M	AN202	3 AN :	LEGGINGS M	Orange		7
8	7 AN202-ORANGE-S	AN202	16 AN :	LEGGINGS S	Orange		8
9	8 AN202-ORANGE-XL	AN202	8 AN :	LEGGINGS XL	Orange		9
10	9 AN202-ORANGE-XXL	AN202	14 AN :	LEGGINGS XXL	Orange		10

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

The screenshot shows the Oracle SQL Developer interface. The Connections tree is identical to the previous screenshot, showing the WEATHER connection expanded.

In the center, the Worksheet pane contains the following SQL code:

```
-- 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,
```

The Query Result pane shows the output of the second query:

TOTAL_RECORDS	9271
1	9271

```
-- 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;
```

Oracle SQL Developer : WEATHER

```
-- 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;
```

TOTAL_RECORDS	9271
---------------	------

-- 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;

File Edit View Navigate Run Source Team Tools Window Help

```
-- 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
```

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	63 BL005-54RED	BL005	16	BLOUSE	FREE	Red
7	67 BL008-61RED	BL008	35	BLOUSE	FREE	Red
8	68 #REF!	BL008	0	BLOUSE	FREE	Red
9	76 BL016-63RED	BL016	24	BLOUSE	FREE	Red
10	81 BL020-71RED	BL020	38	BLOUSE	FREE	Red
11	91 BL024-74RED	BL024	103	BLOUSE	FREE	Red
12	104 BL041-65RED	BL041	30	BLOUSE	FREE	Red
13	105 BL046-82RED	BL046	23	BLOUSE	FREE	Red
14	106 BL050-83RED	BL050	63	BLOUSE	FREE	Red

-- 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;

```

Oracle SQL Developer : WEATHER

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The Connections sidebar on the left lists various database connections, including TNS, AMAZON_SALES, CARS, sys, system, and WEATHER, which is currently selected. The WEATHER connection details show Tables (Filtered), 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, and Other Users. The Database Schema Service Connections section is also present. The central area contains two panes: the Worksheet pane and the Query Result pane. The Worksheet pane displays the following SQL code:

```

-- 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;

```

The Query Result pane shows the output of the first query:

CATEGORY	ITEMS_COUNT
1 KURTA	3726
2 KURTA SET	1598
3 SET	1050
4 TOP	865
5 DRESS	700
6 BLOUSE	241
7 NIGHT WEAR	217
8 TUNIC	154
9 SAREE	147
10 AN : LEGGINGS	131
11 PANT	91
12 PALAZZO	91
13 (null)	45
14 CROP TOP	42

```

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

```

The screenshot shows the Oracle SQL Developer interface. On the left is the Connections sidebar with various database entries like TNS, AMAZON_SALES, CARS, sys, system, and WEATHER. The WEATHER entry is expanded, showing Tables (Filtered), 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, and Other Users. Below this is the Database Schema Service Connections section. The main area has tabs for Worksheet and Query Builder. The Worksheet tab contains a query:

```
-- 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;
```

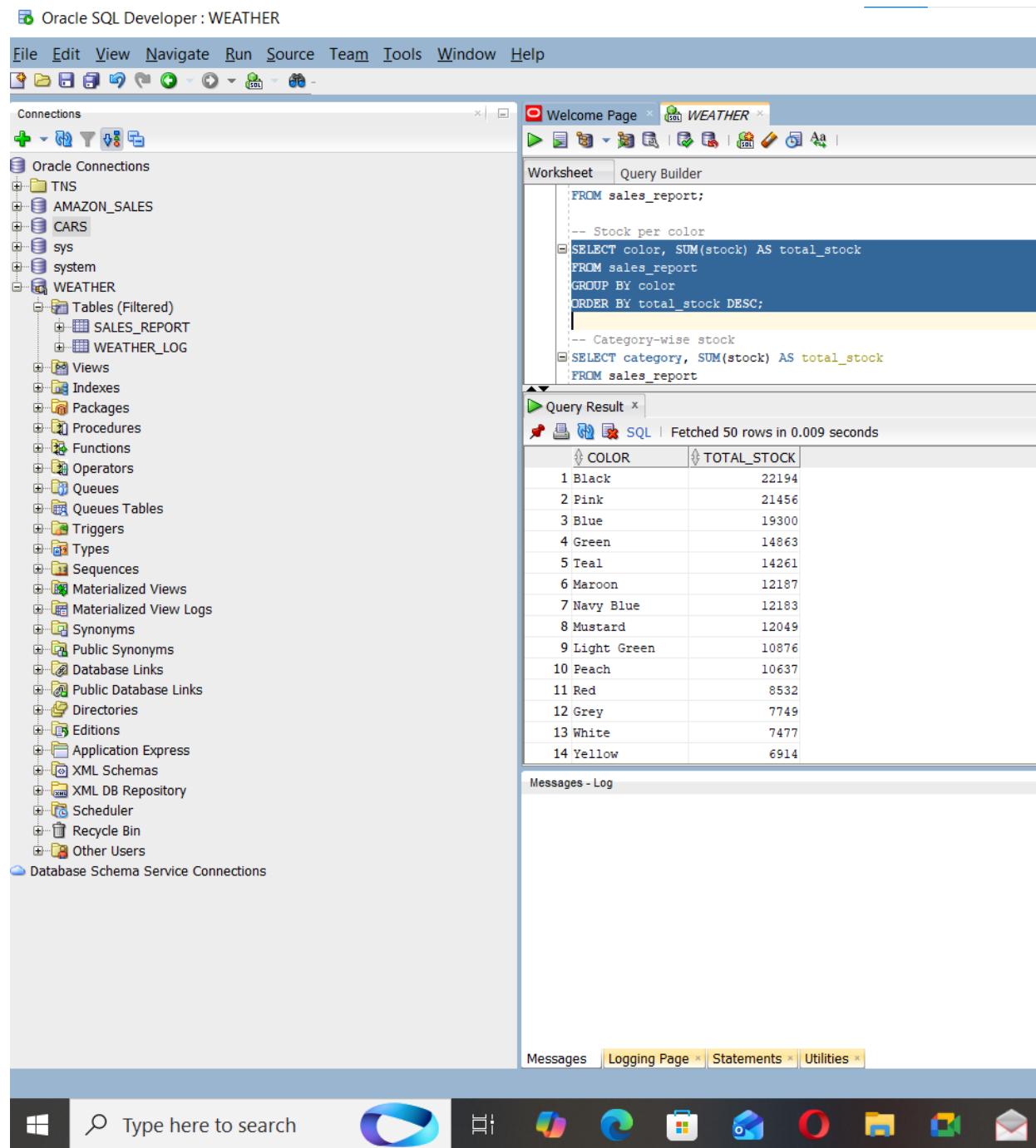
The Query Result tab shows the output of the query:

	TOTAL_STOCK	AVG_STOCK	MIN_STOCK	MAX_STOCK
1	242386	26.24645370871683811586356253383865728208	0	1234

--- 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 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



```
-- 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);  
Oracle SQL Developer : WEATHER
```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The left sidebar displays the 'Connections' tree, which includes Oracle Connections (TNS, AMAZON_SALES, CARS, sys, system), and the current connection set to WEATHER, which contains Tables (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, and Other Users. Below the connections tree is a 'Database Schema Service Connections' section. The main workspace has tabs for 'Worksheet' and 'Query Builder'. The 'Worksheet' tab shows a query editor with the following code:

```
-- 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
```

The 'Query Result' tab shows the output of the first part of the query, listing 50 rows of data from the sales_report table. The columns are SKU_CODE, DESIGN_NO, and STOCK. The data is as follows:

SKU_CODE	DESIGN_NO	STOCK	
1	BL001-50PINK	BL001	33
2	BL003-50BLACK	BL003	117
3	BL004-50CHIKU	BL004	38
4	BL006-54BLACK	BL006	112
5	BL007-61PINK	BL007	56
6	BL008-61RED	BL008	35
7	BL009-61BLACK	BL009	575
8	BL010-61CHIKU	BL010	75
9	BL013-62BLACK	BL013	399
10	BL014-62CHIKU	BL014	73
11	BL017-63BLACK	BL017	208
12	BL018-63BEIGE	BL018	39
13	BL019-71PINK	BL019	76
14	BL020-71RED	BL020	38

The 'Messages - Log' tab is empty.

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

Oracle SQL Developer : WEATHER

File Edit View Navigate Run Source Team Tools Window Help

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

Worksheet Query Builder

```

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
);
  
```

Query Result

COLOR
1 Pink
2 White
3 Dark Green
4 Mustard
5 Cream
6 BURGUNDY
7 Navy Blue
8 Light Brown
9 Yellow
10 Green
11 Grey
12 Olive

Messages - Log

```

-- 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
);
  
```

Oracle SQL Developer : WEATHER

File Edit View Navigate Run Source Team Tools Window Help

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

Worksheet Query Builder

```

-- 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
);
  
```

Query Result

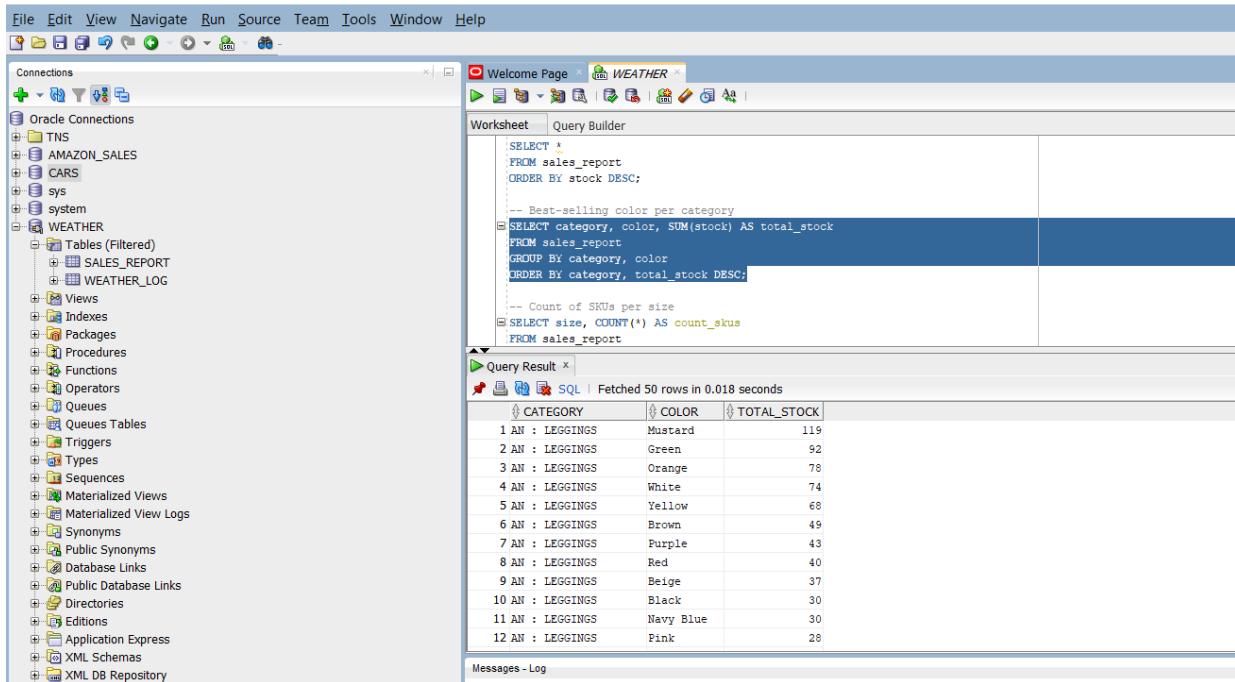
ID	SKU_CODE	DESIGN_NO	STOCK	CATEGORY	SIZE	COLOR
1	734 J0022-KR-L	J0022	2	KURTA	L	Lemon Yellow
2	735 J0022-KR-S	J0022	1	KURTA	S	Lemon Yellow
3	736 J0022-KR-XS	J0022	2	KURTA	XS	Lemon Yellow
4	737 J0022-KR-XXXL	J0022	2	KURTA	XXXL	Lemon Yellow
5	745 J0024-KR-L	J0024	135	KURTA	L	Sea Green
6	746 J0024-KR-M	J0024	94	KURTA	M	Sea Green
7	747 J0024-KR-S	J0024	176	KURTA	S	Sea Green
8	748 J0024-KR-XL	J0024	93	KURTA	XL	Sea Green
9	749 J0024-KR-XS	J0024	151	KURTA	XS	Sea Green
10	750 J0024-KR-XXL	J0024	92	KURTA	XXL	Sea Green
11	751 J0024-KR-XXXL	J0024	63	KURTA	XXXL	Sea Green
12	759 J0026-KR-L	J0026	1	KURTA	L	Turquoise Green

```
-- 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;
```

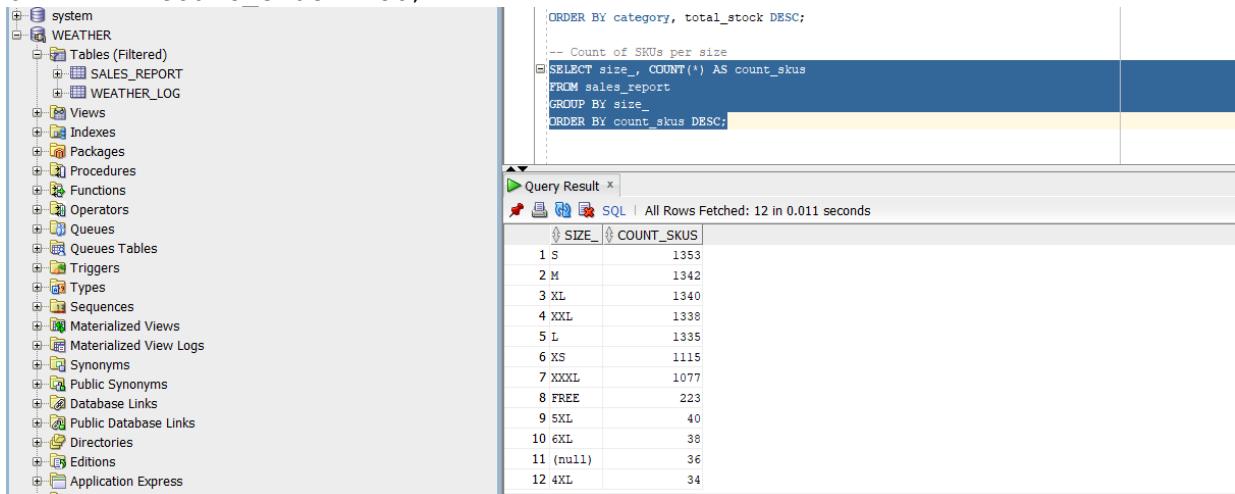
Oracle SQL Developer:WEATHER



The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar on the left lists several database connections, including 'AMAZON_SALES', 'CARS', 'sys', 'system', and 'WEATHER'. The 'WEATHER' connection is selected. The 'Worksheet' tab in the center contains the SQL code for finding top SKUs by stock and the best-selling color per category. The 'Query Result' tab shows the results of the first query, listing 12 different legging colors and their stock counts.

CATEGORY	COLOR	TOTAL_STOCK
1 AN : LEGGINGS	Mustard	119
2 AN : LEGGINGS	Green	92
3 AN : LEGGINGS	Orange	78
4 AN : LEGGINGS	White	74
5 AN : LEGGINGS	Yellow	68
6 AN : LEGGINGS	Brown	49
7 AN : LEGGINGS	Purple	43
8 AN : LEGGINGS	Red	40
9 AN : LEGGINGS	Beige	37
10 AN : LEGGINGS	Black	30
11 AN : LEGGINGS	Navy Blue	30
12 AN : LEGGINGS	Pink	28

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

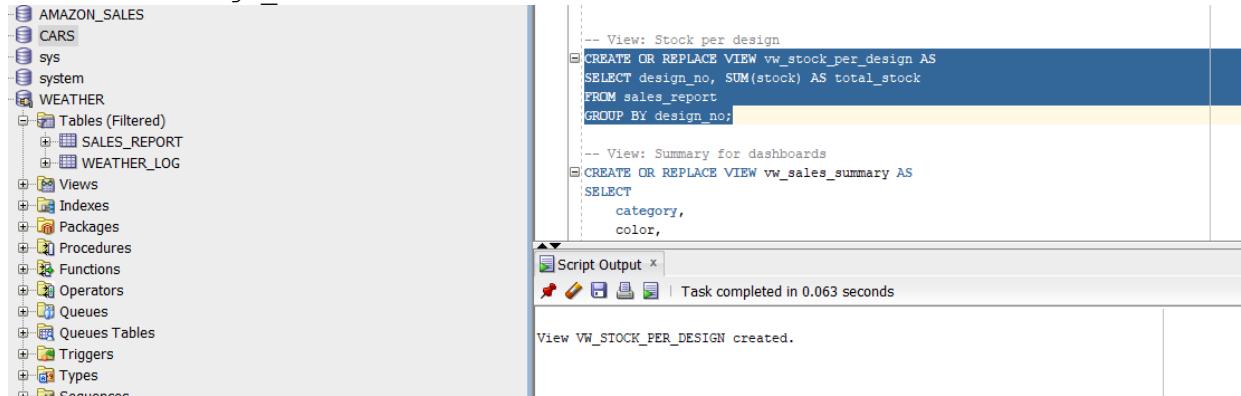


The screenshot shows the Oracle SQL Developer interface again. The 'WEATHER' connection is selected. The 'Worksheet' tab contains the SQL code for counting SKUs per size. The 'Query Result' tab shows the results, listing 12 size categories and their respective counts.

SIZE_	COUNT_SKUS
1 S	1353
2 M	1342
3 XL	1340
4 XXL	1338
5 L	1335
6 XS	1115
7 XXXL	1077
8 FREE	223
9 5XL	40
10 6XL	38
11 (null)	36
12 4XL	34

```
-- 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);
```

```
-- 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);
```

Script Output x | Task completed in 0.05 seconds

View VW_STOCK_PER_DESIGN created.

Index IDX DESIGN NO created.

Index IDX CATEGORY created.

Index IDX COLOR created.

-- END OF FULL SQL ANALYSIS SCRIPT
