

```
SELECT * FROM SALES_REPORT;
```

Help

Welcome Page x WEATHER x

Worksheet Query Builder

```
SELECT * FROM SALES_REPORT;
```

-- 1. PREVIEW DATA

Query Result x

SQL | Fetched 50 rows in 0.006 seconds

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

Messages - Log

```
-- 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 is the 'Connections' tree with 'WEATHER' selected. The main window is split into 'Worksheet' and 'Query Builder' tabs. The 'Query Builder' tab 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' window shows the first 10 rows of the query results:

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. On the left is the 'Connections' tree with 'WEATHER' selected. The main window is split into 'Worksheet' and 'Query Builder' tabs. The 'Query Builder' tab contains the following SQL code:

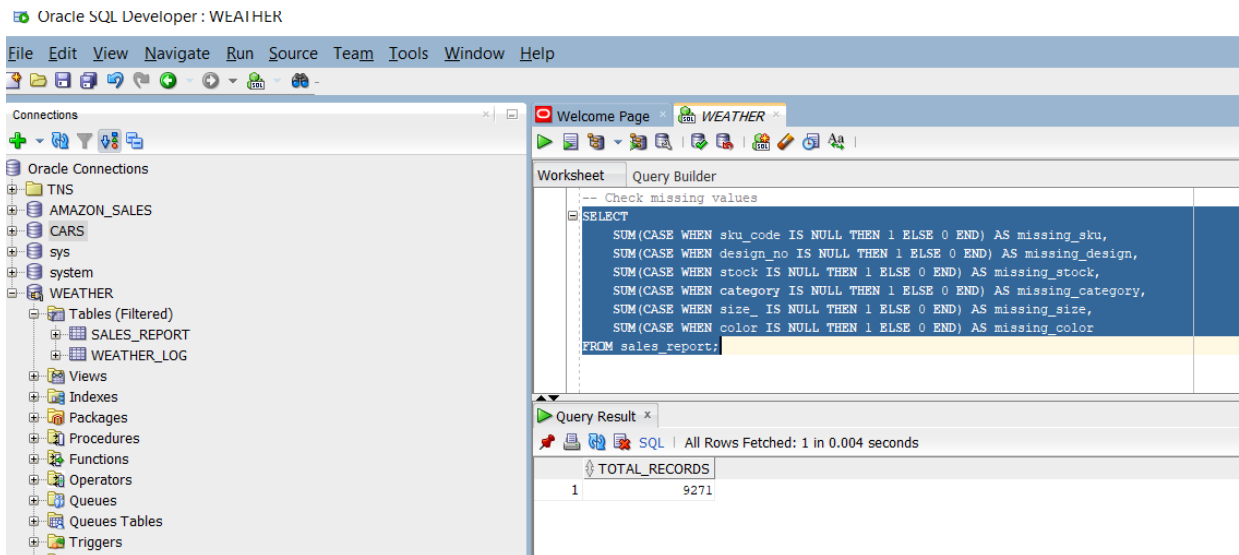
```
-- Count total rows
SELECT COUNT(*) AS total_records
FROM sales_report;

-- 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' window shows the results of the query:

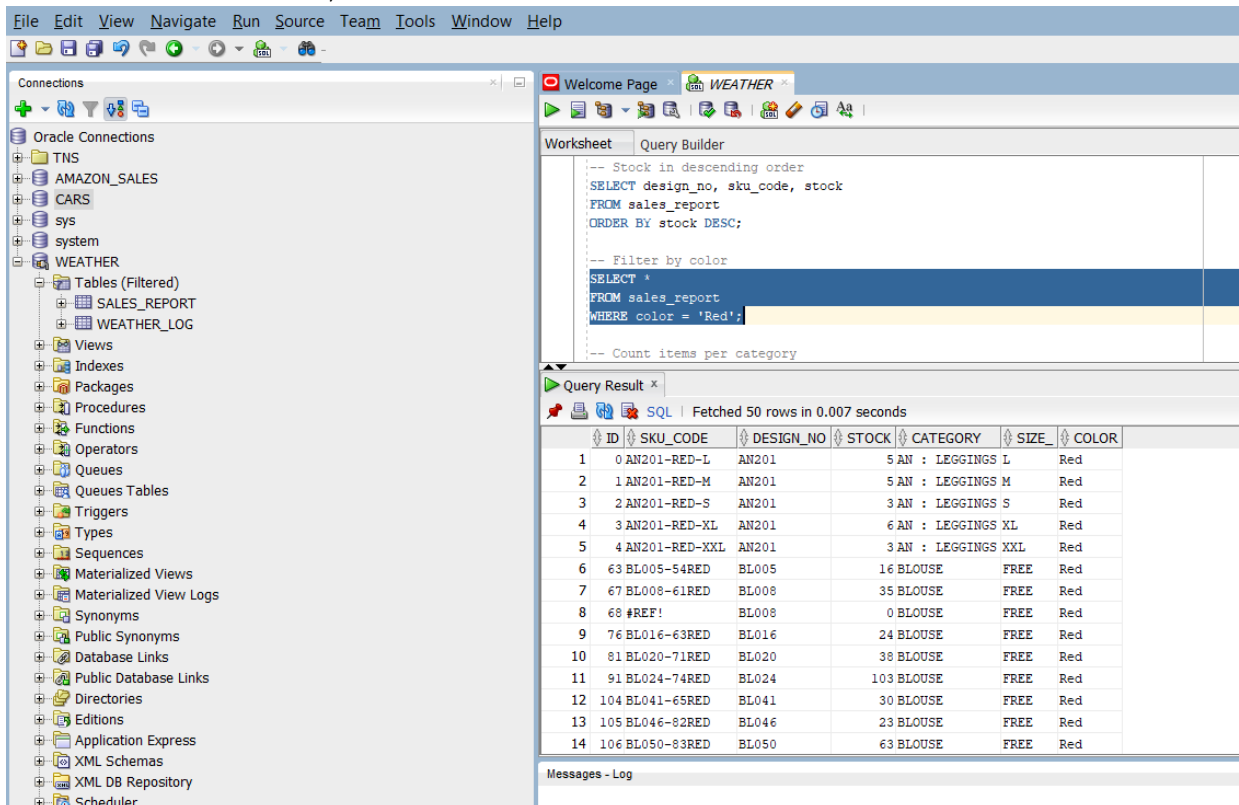
TOTAL_RECORDS
9271

```
-- 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;
```



-- 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;
```

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

```
-- 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
```

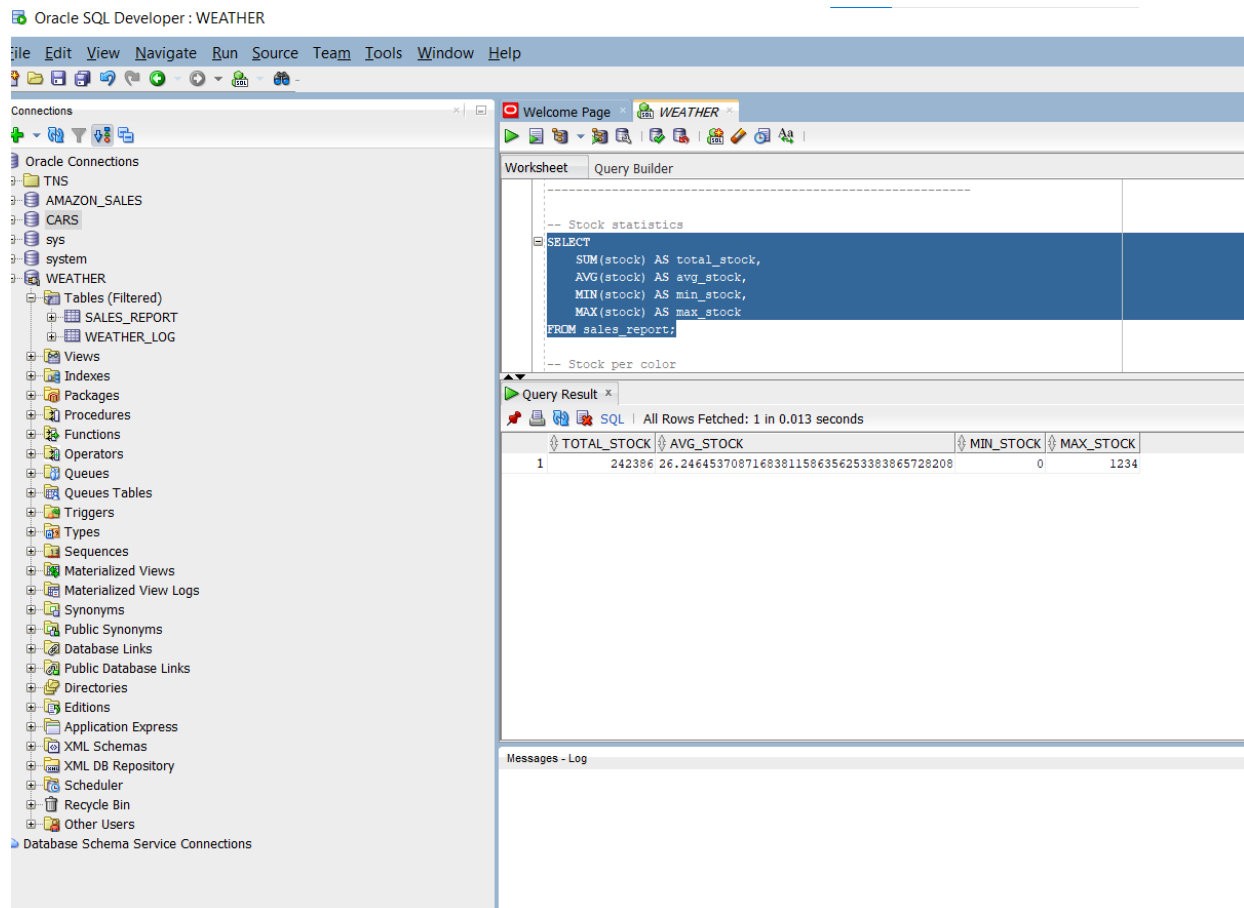
Query Result x

SQL | All Rows Fetched: 22 in 0.015 seconds

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

Messages - Log

```
-- 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;
```

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

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

```

-- 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 with the 'WEATHER' database selected. The 'Connections' pane on the left lists various database objects, including tables like 'SALES_REPORT' and 'WEATHER_LOG'. The main workspace displays a SQL query in the 'Query Builder' tab. The query is designed to find items with above-average stock and colors with more than 10 stock. The 'Query Result' pane shows 50 rows fetched in 0.015 seconds, displaying columns 'SKU_CODE', 'DESIGN_NO', and 'STOCK'.

```
-- 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
```

	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

```
-- 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

- 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

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 = (

```

Query Result

SQL | Fetched 50 rows in 0.007 seconds

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

- 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

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

SQL | Fetched 50 rows in 0.021 seconds

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	83	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 left pane displays the database schema for 'WEATHER', including tables 'SALES_REPORT' and 'WEATHER_LOG'. The main workspace shows a SQL query with three parts: a top 5 SKUs query, a best-selling color query, and a count of SKUs per size query. The 'Query Result' pane shows the results of the first query, displaying 12 rows of category, color, and total stock.

	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;
```

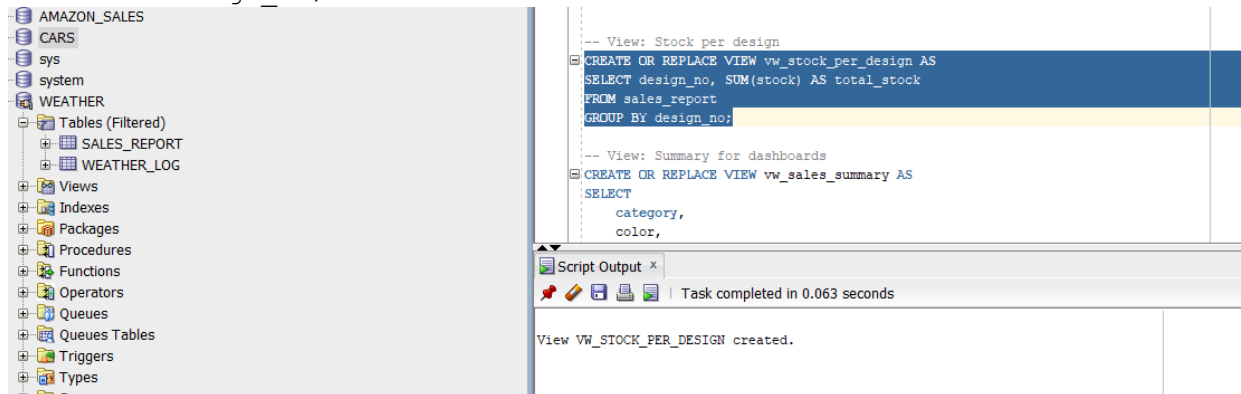
The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema for 'WEATHER'. The main workspace shows a SQL query for counting SKUs per size. The 'Query Result' pane shows the results, displaying 12 rows of size and count.

	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);
```

T
;

;
Logs

nks

```
-- 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
