



# DATA WAREHOUSE PROJECT

PRESENTATION

IGOR PASQUINI,  
2024218

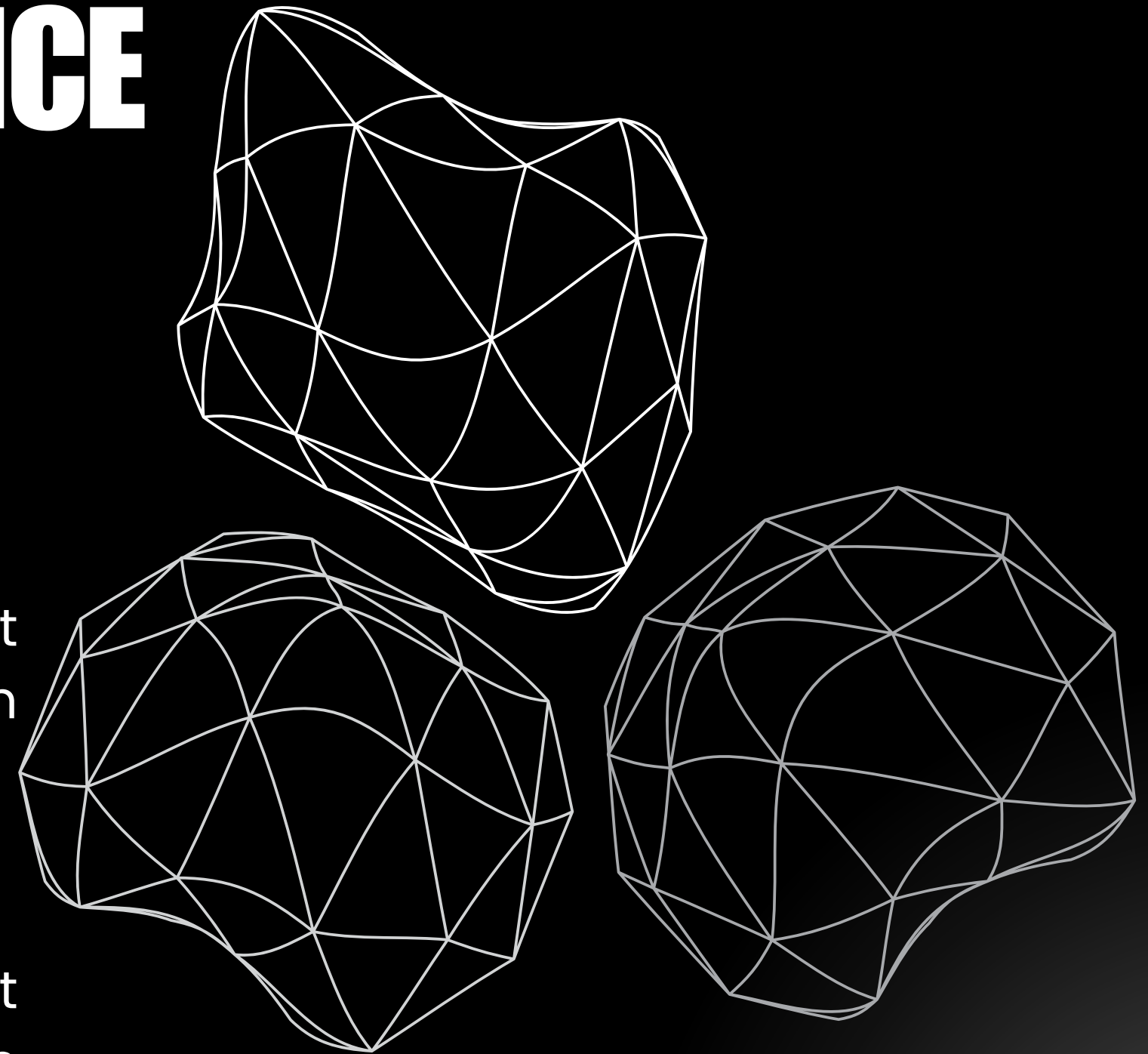
# ENVIRONMENTAL IMPACT ANALYSIS ON ATHLETIC PERFORMANCE

## Problem:

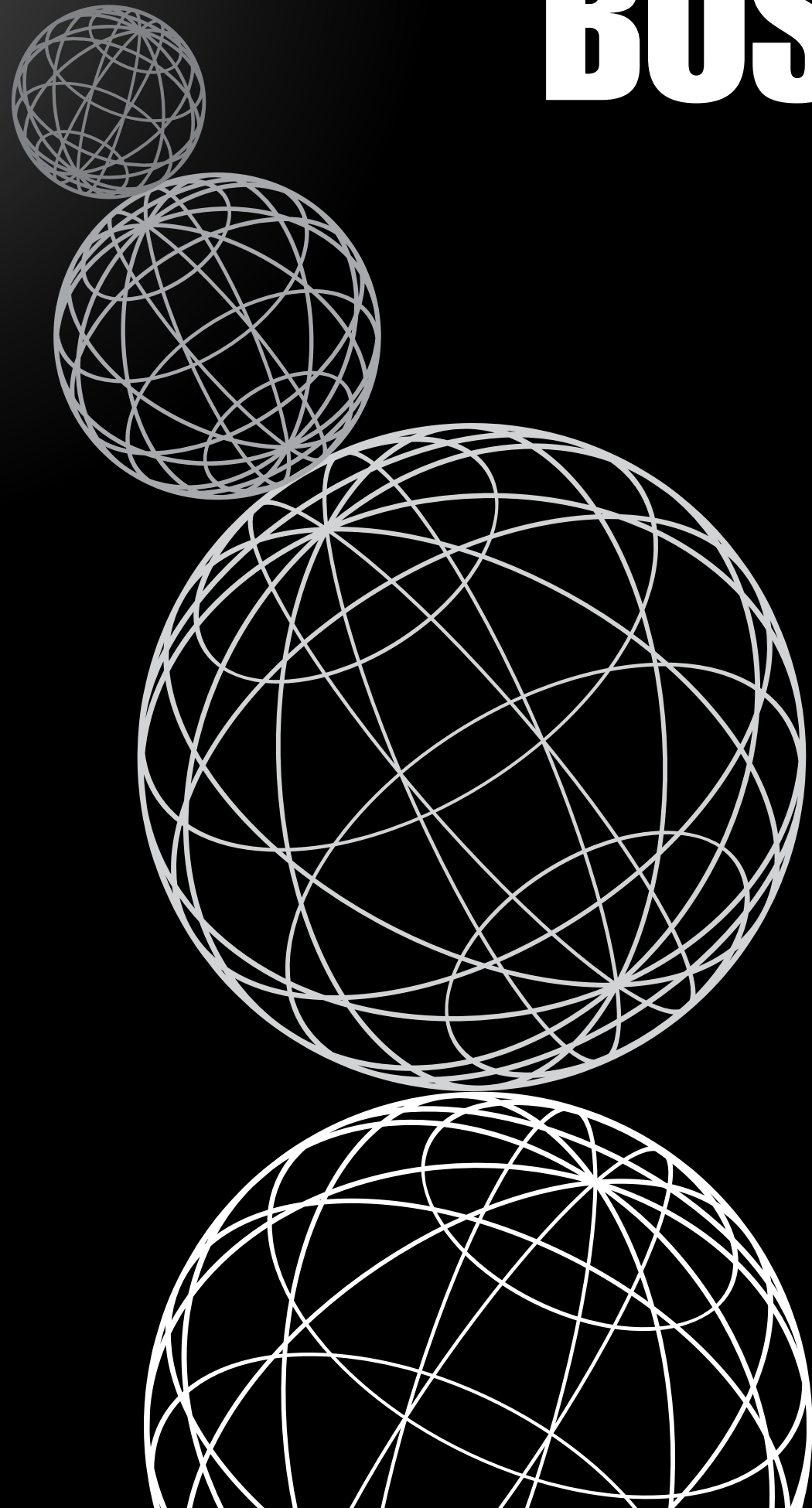
Athletic performance varies significantly under different environmental conditions, but patterns remain unquantified

## Solution:

Integrated DW analyzing altitude/temperature impact on hundreds of thousands performances from professional athletes

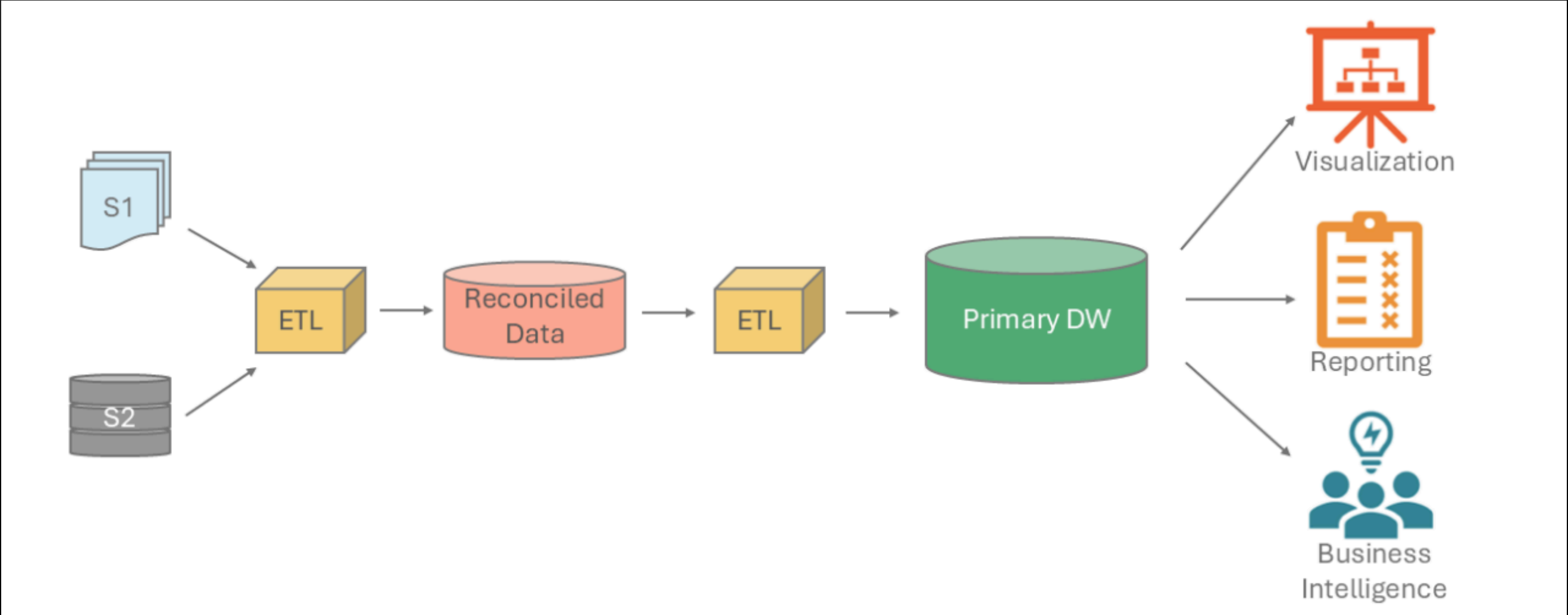


# BUSINESS GOALS



- Quantify altitude effects on sprint vs endurance events
- Identify optimal temperature ranges per event type
- Map geographic performance patterns

# 3-LAYER ARCHITECTURE



## Staging:

Raw CSV ingestion



## Reconciled:

Entity resolution

## DWH:

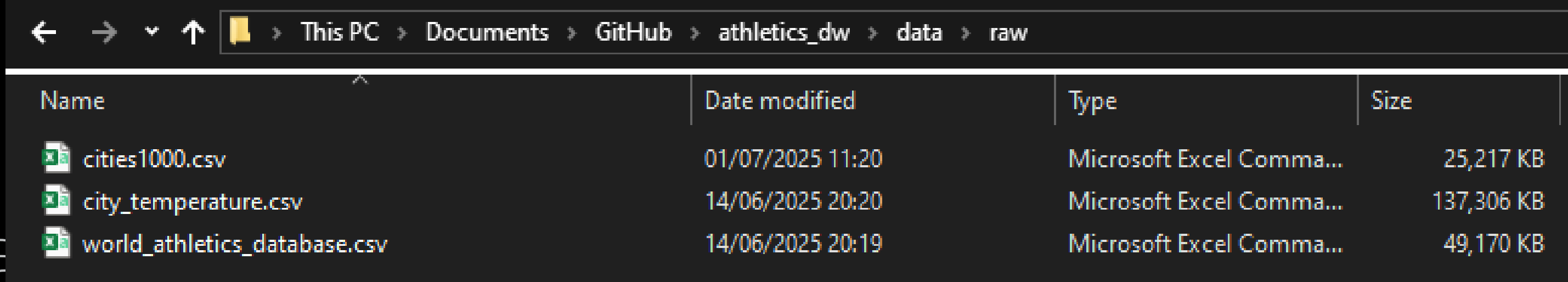
Star schema




The three layers are created by the file create\_layers.sql, that defines 3 schemas representing the different data processing stages:

 01_create_layers.sql	27/06/2025 11:47	SQL File	1 KB
 02a_create_reconciled_schema.sql	11/07/2025 15:26	SQL File	4 KB

# SOURCE LAYER

World Athletics: 461K performances, 35K athletes  
GeoNames: 158K cities (>1000 inhabitants) with elevation  
Temperature DB: 2.9M daily readings (until 2020)




Name	Date modified	Type	Size
 cities1000.csv	01/07/2025 11:20	Microsoft Excel Comma...	25,217 KB
 city_temperature.csv	14/06/2025 20:20	Microsoft Excel Comma...	137,306 KB
 world_athletics_database.csv	14/06/2025 20:19	Microsoft Excel Comma...	49,170 KB






# EXTRACTION

## Data staging

Raw operational data extracted directly from source systems, preserved in original format for initial loading.  
Acts as a buffer zone before transformation





 01_extract.py	15/07/2025 14:55	PY File	7 KB
 02_transform.py	11/07/2025 14:32	PY File	18 KB
 02b_load_reconciled.py	14/07/2025 16:50	PY File	40 KB

```
2025-07-15 14:55:35,118 - INFO - Loading data to staging tables...
2025-07-15 14:55:35,118 - INFO - Loading World Athletics data (461522 rows)...
2025-07-15 15:00:13,037 - INFO - World Athletics data loaded
2025-07-15 15:00:13,047 - INFO - Loading Cities data (158848 rows)...
2025-07-15 15:00:34,027 - INFO - Cities data loaded
2025-07-15 15:00:34,027 - INFO - Loading Temperature data (2906327 rows)...
2025-07-15 15:08:44,778 - INFO - Temperature data loaded
2025-07-15 15:08:44,952 - INFO - Data verification - Athletics: 461522, Cities: 158848, Temperature: 2906327
2025-07-15 15:08:44,952 - INFO - Data extraction completed successfully
PS C:\Users\Igorp\Documents\GitHub\athletics dw> █
```



# RECONCILED DATA LAYER

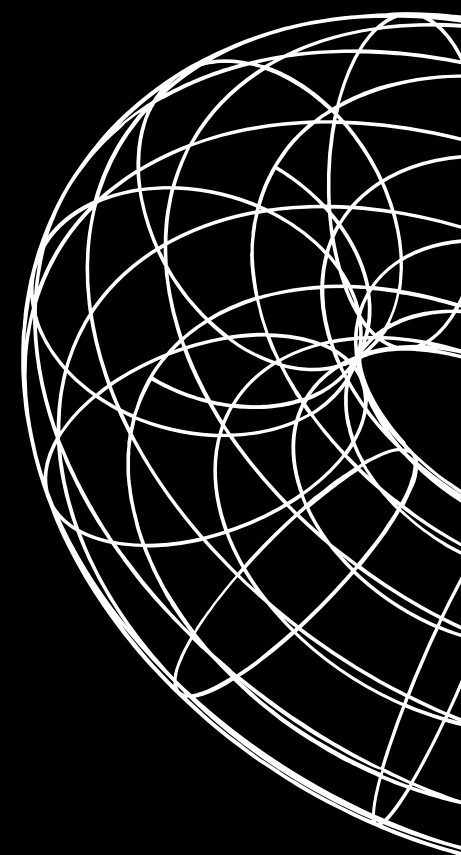
Here the data are transformed, cleaned, and prepared for the loading in the Data Warehouse

 01_create_layers.sql	27/06/2025 11:47	SQL File	1 KB
 02a_create_reconciled_schema.sql	11/07/2025 15:26	SQL File	4 KB




The create\_reconciled\_schema.sql implements the detailed structure of the Reconciled Layer, by creating standardized tables in the reconciled schema:

- Dimension Tables:
  - athletes: Cleaned athlete data (names, nationality, attributes).
  - events: Standardized event definitions (categories, distances).
  - venues: Enriched geographic data (cities, coordinates, climate).
  - weather\_conditions: Environmental metrics (temperature, seasons).
- Fact Table:
  - performances: Core performance records linking all dimensions (results, dates, sources).

Strong points:

- Enforces integrity with **foreign keys** (e.g., performances.athlete\_key → athletes.athlete\_key).
  - Adds **indexes** on key columns (e.g., athlete\_key, event\_key) for query performance.
  - Includes **data quality fields** (e.g., data\_quality\_score, source\_system).
- 

# ETL PIPELINE

 01_extract.py	15/07/2025 14:55	PY File	7 KB
 02_transform.py	11/07/2025 14:32	PY File	18 KB
 02b_load_reconciled.py	14/07/2025 16:50	PY File	40 KB





# TRANSFORMATION

In the photo below, there is the output of the 02\_transform.py file. Its main tasks are:

- Cleaning the raw source data
- Standardizing formats (event names, city names, units)
- Enforcing data quality rules (null handling, value ranges)
- Creating staging tables: clean\_world\_athletics, clean\_cities, clean\_temperature

```
● 2025-07-15 16:36:43,840 - INFO - Starting data transformation process...
2025-07-15 16:36:43,911 - INFO - Cleaning World Athletics data...
2025-07-15 16:36:46,920 - INFO - Original records: 461522
2025-07-15 16:36:46,920 - INFO - Original columns: ['Rank', 'Mark', 'Competitor', 'DOB', 'Nat', 'Pos', 'Venue', 'Date', 'Results Score', 'Mark [meters or seconds]', 'Event', 'Wind', 'Sex']
2025-07-15 16:36:47,207 - INFO - After column selection: ['athlete_name', 'event_name', 'result_value', 'venue_name', 'competition_date', 'nationality', 'gender', 'date_of_birth', 'rank_position', 'wind_reading', 'pos', 'results_score']
2025-07-15 16:36:47,574 - INFO - After removing null essential fields: 461522 records (-0)
2025-07-15 16:36:47,574 - INFO - Cleaning result values...
2025-07-15 16:36:48,215 - INFO - After cleaning result values: 458043 records (-3479 invalid)
2025-07-15 16:36:48,266 - INFO - Standardizing event names...
2025-07-15 16:36:48,392 - INFO - Standardized events: ['10 Kilometres', '10 Kilometres Race Walk', '10 Miles Road', '100 Metres', '100 Metres Hurdles', '1000 Metres', '10000 Metres', '10000 Metres Race Walk', '110 Metres Hurdles', '15 Kilometres', '1500 Metres', '20 Kilometres', '200 Metres', '2000 Metres', '2000 Metres Steeplechase', '20000 Metres Race Walk', '30 Kilometres Race Walk', '300 Metres', '3000 Metres', '3000 Metres Race Walk', '3000 Metres Steeplechase', '35 Kilometres Race Walk', '400 Metres', '400 Metres Hurdles', '5 Kilometres', '5 Kilometres Race Walk', '50 Kilometres Race Walk', '5000 Metres', '5000 Metres Race Walk', '600 Metres', '800 Metres', 'Decathlon', 'Discus Throw', 'Half Marathon', 'Hammer Throw', 'Heptathlon', 'High Jump', 'Javelin Throw', 'Long Jump', 'Marathon', 'One Mile', 'Pole Vault', 'Shot Put', 'Triple Jump', 'Two Miles']
2025-07-15 16:36:48,392 - INFO - Cleaning athlete names...
2025-07-15 16:36:48,785 - INFO - Final columns: ['athlete_name', 'event_name', 'result_value', 'venue_name', 'competition_date', 'nationality', 'gender', 'date_of_birth', 'rank_position', 'wind_reading', 'pos', 'results_score', 'result_numeric', 'event_clean', 'competition_level', 'data_source']
2025-07-15 16:36:48,785 - INFO - Final column count: 16
2025-07-15 16:36:48,785 - INFO - Saving 458043 records to staging.clean_world_athletics in chunks of 10000...
2025-07-15 16:40:07,484 - INFO - clean_world_athletics saved successfully
2025-07-15 16:40:07,487 - INFO - Cleaned World Athletics data: 458043 records saved
2025-07-15 16:40:07,576 - INFO - Integrating geographic data...
2025-07-15 16:40:08,205 - INFO - Original cities data: 158848 records
2025-07-15 16:40:08,205 - INFO - Original columns: ['City', 'Country', 'Latitude', 'Longitude', 'Population', 'Altitude']
2025-07-15 16:40:10,591 - INFO - Clean cities columns: ['city_name', 'country_name', 'latitude', 'longitude', 'altitude', 'altitude_category', 'data_source']
2025-07-15 16:40:10,591 - INFO - Clean cities count: 145457
2025-07-15 16:40:10,592 - INFO - Saving 145457 records to staging.clean_cities in chunks of 10000...
2025-07-15 16:40:38,862 - INFO - clean_cities saved successfully
2025-07-15 16:40:38,863 - INFO - Geographic data integrated: 145457 cities
2025-07-15 16:40:38,902 - INFO - Integrating temperature data...
2025-07-15 16:41:00,083 - INFO - Temperature data after filtering: 2905887 records
2025-07-15 16:41:00,812 - INFO - Monthly averages calculated: 3852 records
2025-07-15 16:41:00,814 - INFO - Converting Fahrenheit to Celsius...
2025-07-15 16:41:00,822 - INFO - Fast saving temperature data...
2025-07-15 16:41:00,823 - INFO - Saving 3852 records to staging.clean_temperature in chunks of 10000...
2025-07-15 16:41:01,499 - INFO - clean_temperature saved successfully
2025-07-15 16:41:01,499 - INFO - Temperature data integrated: 3852 records
2025-07-15 16:41:01,820 - INFO - Data transformation completed successfully!
2025-07-15 16:41:01,820 - INFO - Final record counts:
2025-07-15 16:41:01,820 - INFO -   Athletics: 458043
2025-07-15 16:41:01,821 - INFO -   Cities: 145457
2025-07-15 16:41:01,821 - INFO -   Temperature: 3852
○ PS C:\Users\Igorp\Documents\GitHub\athletics_dw> █
```

# LOADING TO RECONCILED TABLES

The load\_reconciled file manages the Business entity reconciliation. Its main tasks are:

## Entity Resolution:

- Athlete deduplication (reconcile\_athletes())
- Event categorization & standardization (reconcile\_events())
- Venue geolocation matching (reconcile\_venues())
- Weather gap filling (reconcile\_weather())

## Relationship Building:

- Cross-linking performances to athletes/events/venues/weather
- Foreign key assignment

## Advanced Matching:

- Fuzzy matching for weather conditions
- Heuristic-based city extraction from venues

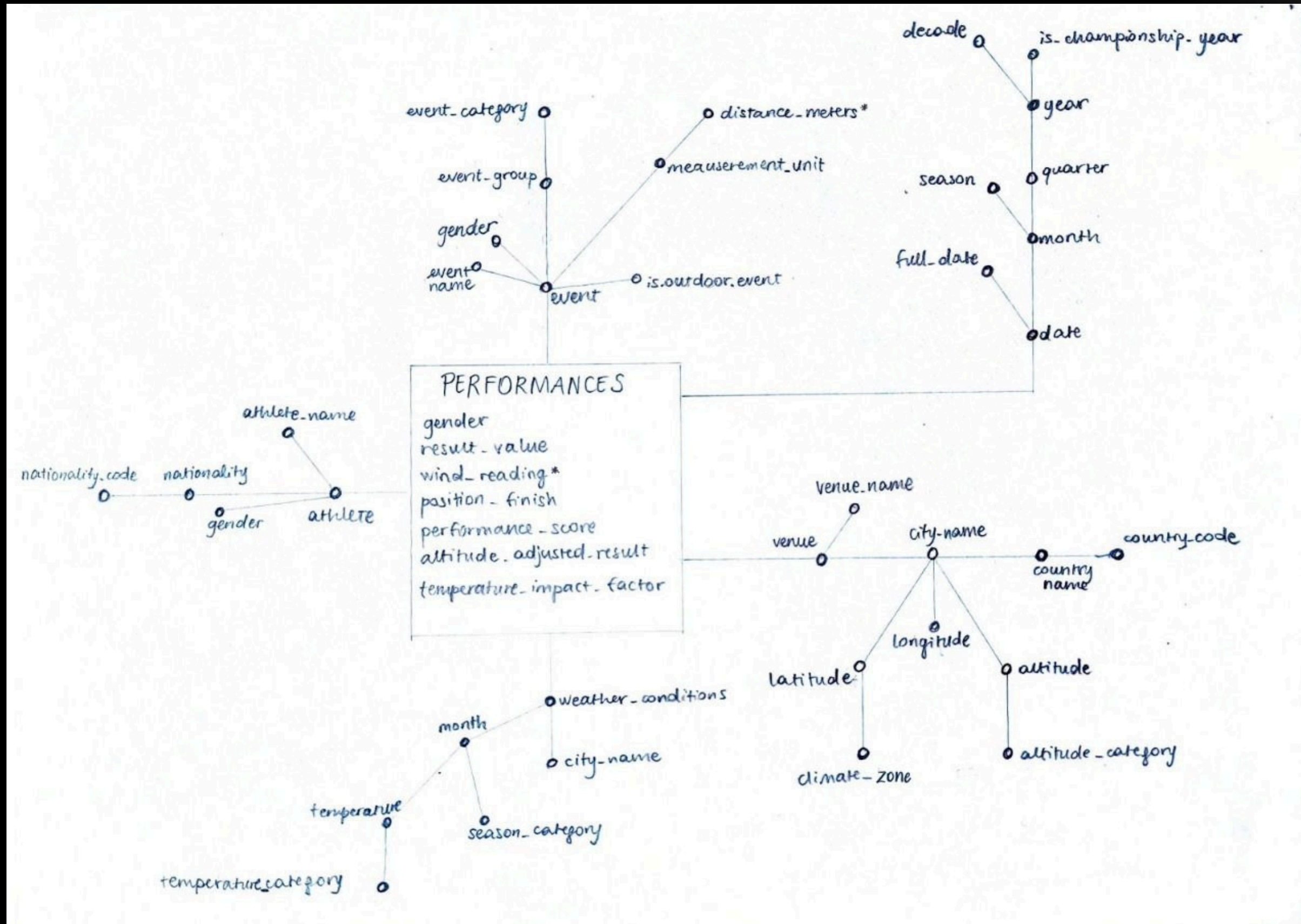
## Output:

- Populates reconciled schema tables

```
INFO: __main__:Reconciling athletes...
INFO: __main__:Deduplication: 34854 → 33785 athletes (1069 duplicates removed)
INFO: __main__:Inserted 33785 athletes.
INFO: __main__:Reconciling events...
INFO: __main__:Filtering out multi-events...
INFO: __main__:Removed 2 multi-event records
INFO: __main__:Inserted 44 events.
INFO: __main__:Reconciling venues with city extraction...
INFO: __main__:Applying city-only matching for 2200 venues...
INFO: __main__:City extraction: 6399/6452 (99.2%)
INFO: __main__:Geographic matching: 4906/6452 (76.0%)
INFO: __main__:Inserted 6452 venue records
INFO: __main__:Reconciling weather conditions...
INFO: __main__:Loaded 3852 temperature records from transform stage
INFO: __main__:Final weather data: 3912 records for 326 cities
INFO: __main__:Inserted 3912 weather records
INFO: __main__:Reconciling performances...
INFO: __main__:STEP 1 - Original performance data: 457454 records
INFO: __main__:Reference table counts - Athletes: 33785, Events: 44, Venues: 6452
INFO: __main__:STEP 2 - After athlete join: 457454 records
INFO: __main__:STEP 3 - After event join: 457454 records
INFO: __main__:Venues after deduplication: 6452 (was 6452)
INFO: __main__:STEP 4 - After venue join: 457454 records
INFO: __main__:Creating weather matching lookup tables...
INFO: __main__:Building exact match lookup table...
INFO: __main__:Exact matches: 205964/457454 (45.0%)
INFO: __main__:Creating similarity matches for 251490 unmatched records...
INFO: __main__:Only 11202 unique city/month combinations need similarity matching
INFO: __main__:Similarity matches: 251490 additional matches
INFO: __main__:Total weather matching success: 457454/457454 (100.0%)
INFO: __main__:STEP 5 - After weather join: 457454 records
INFO: __main__:Removing performances without weather data...
INFO: __main__:Reconciled layer weather filter:
INFO: __main__:  Initial performances: 457,454
INFO: __main__:  Removed (no weather): 185,036 (40.4%)
INFO: __main__:  Reconciled with weather: 272,418
INFO: __main__:Before filtering: 272418 records
INFO: __main__:After requiring athlete/event IDs: 272371 records
INFO: __main__:Filter duplicated performances: 8481 performances removed
INFO: __main__:Final performance records ready for insert: 263890
INFO: __main__:Weather match success: 263890/263890 performances have weather data
INFO: __main__:Cleared existing performance data
INFO: __main__:Starting save of performance data...
INFO: __main__:Appending 263890 records to reconciled.performances...
INFO: __main__:Table reconciled.performances exists and is accessible
INFO: __main__:Executing COPY command for reconciled.performances
INFO: __main__:263890 records appended successfully
INFO: __main__:Inserted 263890 performances.
INFO: __main__:reconciled.athletes: 33785 records
INFO: __main__:reconciled.events: 44 records
INFO: __main__:reconciled.venues: 6452 records
INFO: __main__:reconciled.weather_conditions: 3912 records
INFO: __main__:reconciled.performances: 263890 records
INFO: __main__:Reconciled data layer created successfully.
PS C:\Users\Igorp\Documents\GitHub\athletics_dw> |
```



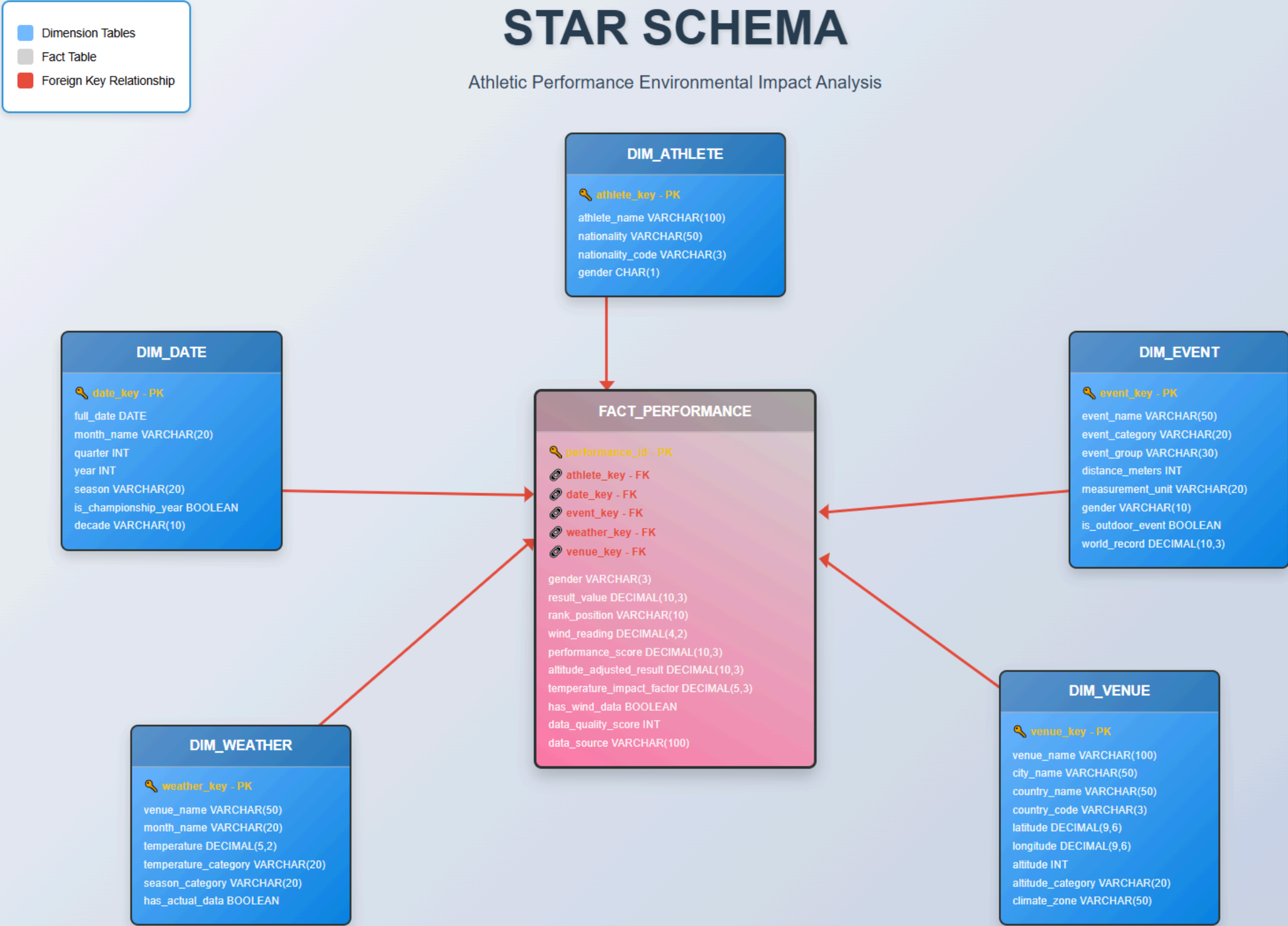
# DIMENSIONAL FACT MODEL



# STAR SCHEMA



## STAR SCHEMA

Athletic Performance Environmental Impact Analysis



# DATA WAREHOUSE LAYER



The Data Warehouse layer is where clean and validated data from RDL are transformed into a star schema optimized for answering business questions.

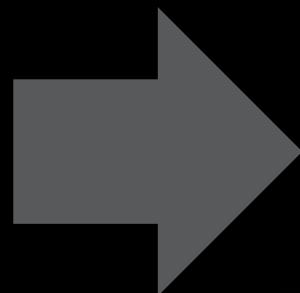
 02b_create_dimensions.sql	11/07/2025 14:38	SQL File	3 KB
 03_create_fact_table.sql	11/07/2025 14:40	SQL File	5 KB

- The first file creates **Dimension tables** in the `dwh` schema:
  - `dim_date`: Temporal attributes (season, decade, championship year).
  - `dim_venue`: Geographic/climate attributes (altitude, climate zone).
  - `dim_event`: Event details (category, distance).
  - `dim_athlete`: Athlete attributes (gender, nationality).
  - `dim_weather`: Environmental conditions (temperature, season).
  - It includes **Indexes** to optimize query performance.
- The second one creates the **fact table** `fact_performance` and defines its structure:
  - **Foreign Keys**: Links to all dimension tables (`athlete_key`, `event_key`, etc.).
  - **Measures**:
    - Raw results (`result_value`, `rank_position`).
    - Standardized metrics (`performance_score`).
    - Environmental adjustments (`altitude_adjusted_result`, `temperature_impact_factor`).
  - It includes **indexes** to support common analytical queries (e.g., `idx_fact_environmental_analysis` for event/venue/weather analysis).
- **Business Focus**: Explicitly designed for environmental impact analysis (e.g., altitude/temperature effects on performance).



# DATA WAREHOUSE LOADING

 03_load_dimensions.py	14/07/2025 11:41	PY File	7 KB
 04_load_facts.py	13/07/2025 15:28	PY File	41 KB



- The first file populates **Dimension tables** from the reconciled layer: It extracts data from reconciled tables, then transforms them (e.g., parses dates, calculates derived attributes like **season**), and finally loads into dwh dimensions.
- **Key Logic:** the `load_date_dimension` derives time attributes from competition dates, and include them in the date dimension.
- The second file populates the **Fact table** `fact_performance` with enriched data.
- **Workflow:**
  - Extract:** Fetches performances from `reconciled.performances`.
  - Join Dimensions:** Links athlete, event, venue, date, and weather data.
  - Calculate Metrics:**
    - `performance_score`: Uses World Athletics coefficients for standardized scoring.
    - `altitude_adjusted_result`: Adjusts results for altitude effects (e.g., sprints benefit at high altitude).
    - `temperature_impact_factor`: Models temperature impact (e.g., distance events suffer in heat).
  - Filter Outliers:** Removes unrealistic performances (e.g., 100m times < 9.57s).
  - Load:** Inserts processed data into `fact_performance`.
- **Environmental Focus:**
  - Altitude/temperature adjustments are central to the calculations.
  - Indexes are optimized for environmental queries (e.g., `idx_fact_weather_impact`).

# Executions:

```
● 2025-07-16 14:05:59,156 - INFO - Starting dimension loading from reconciled layer...
2025-07-16 14:05:59,338 - INFO - Cleared dwh.dim_athlete
2025-07-16 14:05:59,408 - INFO - Cleared dwh.dim_date
2025-07-16 14:05:59,479 - INFO - Cleared dwh.dim_event
2025-07-16 14:05:59,548 - INFO - Cleared dwh.dim_venue
2025-07-16 14:05:59,607 - INFO - Cleared dwh.dim_weather
2025-07-16 14:05:59,617 - INFO - Loading date dimension from actual performance dates...
2025-07-16 14:05:59,661 - INFO - Found 9425 unique competition dates
2025-07-16 14:05:59,661 - INFO - Valid dates after parsing: 9425
2025-07-16 14:06:00,463 - INFO - Date dimension loaded: 9425 actual dates
2025-07-16 14:06:00,463 - INFO - Loading athlete dimension from reconciled.athletes...
2025-07-16 14:06:03,199 - INFO - Athlete dimension loaded: 33785 records
2025-07-16 14:06:03,208 - INFO - Loading event dimension from reconciled.events...
2025-07-16 14:06:03,219 - INFO - Event dimension loaded: 44 records
2025-07-16 14:06:03,219 - INFO - Loading venue dimension from reconciled.venues excluding venues without proper altitude value...
2025-07-16 14:06:03,589 - INFO - Venue dimension loaded: 4884 records
2025-07-16 14:06:03,589 - INFO - Loading weather dimension from reconciled.weather_conditions...
2025-07-16 14:06:03,941 - INFO - Weather dimension loaded: 3912 records
2025-07-16 14:06:03,943 - INFO - dwh.dim_athlete: 33785 records
2025-07-16 14:06:03,949 - INFO - dwh.dim_date: 9425 records
2025-07-16 14:06:03,949 - INFO - dwh.dim_event: 44 records
2025-07-16 14:06:03,952 - INFO - dwh.dim_venue: 4884 records
2025-07-16 14:06:03,952 - INFO - dwh.dim_weather: 3912 records
2025-07-16 14:06:03,952 - INFO - All DWH dimensions loaded successfully.
○ PS C:\Users\Igorp\Documents\GitHub\athletics_dw> █
```

```
● 2025-07-16 14:06:35,665 - INFO - Starting fact table loading...
2025-07-16 14:06:35,723 - INFO - FACT LOADING - 5 Essential Dimensions
2025-07-16 14:06:37,459 - INFO - Loaded 260248 performance records from reconciled layer
2025-07-16 14:06:37,561 - INFO - Loaded 5 essential dimensions:
2025-07-16 14:06:37,561 - INFO - Athletes: 33785 records
2025-07-16 14:06:37,561 - INFO - Events: 44 records
2025-07-16 14:06:37,561 - INFO - Venues: 4884 records
2025-07-16 14:06:37,561 - INFO - Weather: 3912 records
2025-07-16 14:06:37,638 - INFO - Dates: 9425 records
2025-07-16 14:06:37,638 - INFO - Joining 5 essential dimensions...
2025-07-16 14:06:38,332 - INFO - After joining 5 dimensions: 240836 records
2025-07-16 14:06:38,338 - INFO - 5-Dimension join success rates:
2025-07-16 14:06:38,338 - INFO - athlete_key: 240836/240836 (100.0%)
2025-07-16 14:06:38,338 - INFO - event_key: 240836/240836 (100.0%)
2025-07-16 14:06:38,338 - INFO - venue_key: 240836/240836 (100.0%)
2025-07-16 14:06:38,338 - INFO - date_key: 240836/240836 (100.0%)
2025-07-16 14:06:38,338 - INFO - weather_key: 240836/240836 (100.0%)
2025-07-16 14:06:38,471 - INFO - Filtered out 0 records missing critical dimensions
2025-07-16 14:06:38,471 - INFO - === FILTERING PERFORMANCE OUTLIERS ===
2025-07-16 14:06:43,038 - INFO - Initial performances: 240,836
2025-07-16 14:06:43,038 - INFO - Realistic performances: 238,918
2025-07-16 14:06:43,038 - INFO - Filtered outliers: 1,918 (0.8%)
2025-07-16 14:06:43,058 - INFO - TOP OUTLIER EVENTS:
2025-07-16 14:06:43,059 - INFO - 10 Kilometres (F): 1.0 outliers, range: 1759.0-1759.0
2025-07-16 14:06:43,059 - INFO - 10 Kilometres Race Walk (F): 169.0 outliers, range: 2476.0-2642.0
2025-07-16 14:06:43,059 - INFO - 10 Kilometres Race Walk (M): 19.0 outliers, range: 2231.0-2311.0
2025-07-16 14:06:43,062 - INFO - 10 Miles Road (F): 8.0 outliers, range: 3032.0-3048.0
2025-07-16 14:06:43,062 - INFO - 10 Miles Road (M): 10.0 outliers, range: 2644.0-2706.0
2025-07-16 14:06:43,062 - INFO - 100 Metres Hurdles (F): 1.0 outliers, range: 12.12-12.12
2025-07-16 14:06:43,063 - INFO - 1000 Metres (F): 1.0 outliers, range: 148.98-148.98
2025-07-16 14:06:43,063 - INFO - 10000 Metres (F): 1.0 outliers, range: 1757.45-1757.45
2025-07-16 14:06:43,063 - INFO - 10000 Metres (M): 1.0 outliers, range: 1571.0-1571.0
2025-07-16 14:06:43,063 - INFO - 10000 Metres Race Walk (F): 50.0 outliers, range: 2516.23-2639.73
2025-07-16 14:06:43,063 - INFO - =====
2025-07-16 14:06:43,148 - INFO - Outlier filtering removed 1918 suspicious performances
2025-07-16 14:06:43,209 - INFO - Filtered out 0 records missing critical dimensions
2025-07-16 14:06:43,227 - INFO - Calculating ALL performance measures...
2025-07-16 14:06:43,228 - INFO - Calculating performance_score
2025-07-16 14:06:47,351 - INFO - Calculating altitude_adjusted_result
2025-07-16 14:06:51,238 - INFO - Calculating temperature_impact_factor
2025-07-16 14:06:54,069 - INFO - Adding performance context flags...
2025-07-16 14:06:54,273 - INFO - SIMPLIFIED FACT TABLE SUMMARY:
2025-07-16 14:06:54,273 - INFO - Total performances: 238,918
2025-07-16 14:06:54,281 - INFO - Unique athletes: 25,887
2025-07-16 14:06:54,281 - INFO - Unique events: 43
2025-07-16 14:06:54,288 - INFO - Unique venues: 1,627
2025-07-16 14:06:54,291 - INFO - Unique dates: 9,010
2025-07-16 14:06:54,298 - INFO - Unique weather conditions: 1,971
2025-07-16 14:06:54,298 - INFO - ALL CALCULATION FUNCTIONS USED:
2025-07-16 14:06:54,299 - INFO - performance_score: avg = 1133.9
2025-07-16 14:06:54,299 - INFO - altitude_adjusted_result: avg = 583.886
2025-07-16 14:06:54,299 - INFO - temperature_impact_factor: avg = 0.984
2025-07-16 14:06:54,299 - INFO - Loading 238918 records to dwh.fact_performance...
2025-07-16 14:08:08,749 - INFO - Total fact records: 238918
○ PS C:\Users\Igorp\Documents\GitHub\athletics_dw> █
```



# DATA VALIDATION

## DATA COMPLETENESS CHECK

- Counts performance records across all layers:

Query Query History

---

```

1 SELECT
2     'Layer 1 (Staging)' as layer,
3     COUNT(*) as total_records,
4     COUNT(DISTINCT athlete_name) as unique_athletes,
5     COUNT(DISTINCT event_clean) as unique_events
6 FROM staging.clean_world_athletics
7 UNION ALL
8 SELECT 'Layer 2 (Reconciled)',
9        COUNT(*), COUNT(DISTINCT athlete_key), COUNT(DISTINCT event_key)
10 FROM reconciled.performances
11 UNION ALL
12 SELECT 'Layer 3 (Star Schema)',
13        COUNT(*), COUNT(DISTINCT athlete_key), COUNT(DISTINCT event_key)
14 FROM dwh.fact_performance
15 ORDER BY layer;
```

Data Output Messages Notifications

---

SQL

	layer text	total_records bigint	unique_athletes bigint	unique_events bigint
1	Layer 1 (Staging)	458043	33787	46
2	Layer 2 (Reconciled)	263890	27024	44
3	Layer 3 (Star Schema)	238918	25887	43

## DATA QUALITY VALIDATION

- Check foreign key integrity in the fact table:

Query

Query History

```

1  SELECT
2      'Missing Athletes' as check_type,
3      COUNT(*) as failed_records,
4      CASE WHEN COUNT(*) = 0 THEN 'PASS' ELSE 'FAIL' END as status
5  FROM dwh.fact_performance f
6  LEFT JOIN dwh.dim_athlete a ON f.athlete_key = a.athlete_key
7  WHERE a.athlete_key IS NULL
8
9  UNION ALL
10
11 SELECT 'Missing Events', COUNT(*),
12      CASE WHEN COUNT(*) = 0 THEN 'PASS' ELSE 'FAIL' END
13 FROM dwh.fact_performance f
14 LEFT JOIN dwh.dim_event e ON f.event_key = e.event_key
15 WHERE e.event_key IS NULL
16
17 UNION ALL
18
19 SELECT 'Missing Venues', COUNT(*),
20      CASE WHEN COUNT(*) = 0 THEN 'PASS' ELSE 'FAIL' END
21 FROM dwh.fact_performance f
22 LEFT JOIN dwh.dim_venue v ON f.venue_key = v.venue_key
23 WHERE v.venue_key IS NULL
24
25 ORDER BY check_type;

```

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	check_type text	failed_records bigint	status text
1	Missing Athletes	0	PASS
2	Missing Events	0	PASS
3	Missing Venues	0	PASS

# PERFORMANCE DATA QUALITY

- Validate performance scores and key measures:

Query

Query History

1

SELECT

2

'Valid Performance Scores' as metric,

3

COUNT(CASE WHEN performance\_score BETWEEN 200 AND 1400 THEN 1 END) as valid\_count,

4

COUNT(\*) as total\_count,

5

ROUND(100.0 \* COUNT(CASE WHEN performance\_score BETWEEN 200 AND 1400 THEN 1 END) / COUNT(\*), 1) as success\_rate

6

FROM dwh.fact\_performance

7

8

UNION ALL

9

10

SELECT 'Valid Result Values',

11

COUNT(CASE WHEN result\_value > 0 AND result\_value < 50000 THEN 1 END),

12

COUNT(\*),

13

ROUND(100.0 \* COUNT(CASE WHEN result\_value > 0 AND result\_value < 50000 THEN 1 END) / COUNT(\*), 1)

14

FROM dwh.fact\_performance

15

16

UNION ALL

17

18

SELECT 'Environmental Data Complete',

19

COUNT(CASE WHEN temperature\_impact\_factor IS NOT NULL AND altitude\_adjusted\_result IS NOT NULL THEN 1 END),

20

COUNT(\*),

21

ROUND(100.0 \* COUNT(CASE WHEN temperature\_impact\_factor IS NOT NULL AND altitude\_adjusted\_result IS NOT NULL THEN 1 END) / COUNT(\*), 1)

22

FROM dwh.fact\_performance;

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	metric text	valid_count bigint	total_count bigint	success_rate numeric
1	Environmental Data Complete	238918	238918	100.0
2	Valid Result Values	238918	238918	100.0
3	Valid Performance Scores	238918	238918	100.0

## SAMPLE DATA VERIFICATION

- Show sample of integrated data to verify joins worked correctly

Query
Query History

---

```

1 SELECT
2     a.athlete_name,
3     a.nationality_code,
4     e.event_name,
5     v.venue_name,
6     v.country_code as venue_country,
7     v.altitude,
8     w.temperature,
9     f.result_value,
10    f.performance_score,
11    d.year
12 FROM dwf.fact_performance f
13 JOIN dwf.dim_athlete a ON f.athlete_key = a.athlete_key
14 JOIN dwf.dim_event e ON f.event_key = e.event_key
15 JOIN dwf.dim_venue v ON f.venue_key = v.venue_key
16 JOIN dwf.dim_weather w ON f.weather_key = w.weather_key
17 JOIN dwf.dim_date d ON f.date_key = d.date_key
18 WHERE f.performance_score > 1000 AND f.performance_score < 1350
19 ORDER BY f.performance_score DESC
20 LIMIT 10;
```

Data Output
Messages
Notifications

	athlete_name <small>character varying (100)</small>	nationality_code <small>character varying (3)</small>	event_name <small>character varying (50)</small>	venue_name <small>character varying (100)</small>	venue_country <small>character varying (3)</small>	altitude <small>integer</small>	temperature <small>numeric (5,2)</small>	result_value <small>numeric (10,3)</small>	performance_score <small>numeric (10,3)</small>	year <small>integer</small>
1	Mike Powell	USA	Long Jump	National Stadium, Tokyo (Jpn)	JP	44	27.04	8.950	1346.033	1991
2	Yohan Blake	JAM	200 Metres	Boudewijnstadion, Bruxelles (Bel)	BE	28	15.02	19.260	1343.072	2011
3	Usain Bolt	JAM	100 Metres	Olympic Stadium, London (Gbr)	GB	25	18.10	9.630	1341.408	2012
4	Karsten Warholm	NOR	400 Metres Hurdles	National Stadium, Tokyo (Jpn)	JP	44	27.04	45.940	1341.105	2021
5	Usain Bolt	JAM	200 Metres	National Stadium, Beijing (Chn)	CN	49	25.53	19.300	1337.887	2008
6	Noah Lyles	USA	200 Metres	Hayward Field, Eugene, Or (usa)	US	132	19.82	19.310	1336.593	2022
7	Zdeňka Šilhavá	TCH	Discus Throw	Nitra (Tch)	TC	144	7.75	74.560	1335.933	1984
8	Michael Johnson	USA	200 Metres	Olympic Stadium, Atlanta, Ga (usa)	US	336	26.02	19.320	1335.298	1996
9	Usain Bolt	JAM	200 Metres	Olympic Stadium, London (Gbr)	GB	25	18.10	19.320	1335.298	2012
10	Gabriele Reinsch	GDR	Discus Throw	Berlin (Ger)	DE	43	15.00	74.440	1333.469	1988

Total rows: 10
Query complete 00:00:00.083

## BUSINESS LOGIC VALIDATION

- Verify event categorization and measurement units

Query

Query History

```

1  SELECT
2      e.event_category,
3      e.measurement_unit,
4      COUNT(*) as performances,
5      ROUND(MIN(f.performance_score), 1) as min_score,
6      ROUND(AVG(f.performance_score), 1) as avg_score,
7      ROUND(MAX(f.performance_score), 1) as max_score,
8      ROUND(STDDEV(f.performance_score), 1) as score_variance
9  FROM dwh.fact_performance f
10 JOIN dwh.dim_event e ON f.event_key = e.event_key
11 GROUP BY e.event_category, e.measurement_unit
12 ORDER BY e.event_category, performances DESC;

```

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	event_category character varying (20) 🔒	measurement_unit character varying (20) 🔒	performances bigint 🔒	min_score numeric 🔒	avg_score numeric 🔒	max_score numeric 🔒	score_variance numeric 🔒
1	Field	meters	82093	856.3	1087.5	1372.1	80.9
2	Road	seconds	7566	979.4	1134.3	1285.0	56.0
3	Track	seconds	149259	801.4	1159.4	1355.9	73.0



# OLAP QUERIES

## 1) Environmental Impact Analysis:

- Cross-sectional analysis of environmental factors
- Operation: DICE
  - It builds a 3-D “cube” over
    - i. Altitude category
    - ii. Temperature category
    - iii. Event category

Query

Query History

1

SELECT

v.altitude\_category,

w.temperature\_category,

e.event\_category,

COUNT(\*) as performances,

ROUND(AVG(f.performance\_score), 1) as avg\_performance\_score,

ROUND(AVG(f.temperature\_impact\_factor), 3) as avg\_temp\_impact,

ROUND(AVG(f.altitude\_adjusted\_result - f.result\_value), 3) as avg\_altitude\_adjustment

FROM dwh.fact\_performance f

JOIN dwh.dim\_venue v ON f.venue\_key = v.venue\_key

JOIN dwh.dim\_weather w ON f.weather\_key = w.weather\_key

JOIN dwh.dim\_event e ON f.event\_key = e.event\_key

WHERE v.altitude\_category IS NOT NULL

AND w.temperature\_category IS NOT NULL

GROUP BY v.altitude\_category, w.temperature\_category, e.event\_category

HAVING COUNT(\*) >= 20

ORDER BY avg\_performance\_score DESC

LIMIT 15;

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🔍

⬇️

📈

SQL

	altitude_category character varying (20) 🔒	temperature_category character varying (20) 🔒	event_category character varying (20) 🔒	performances bigint 🔒	avg_performance_score numeric 🔒	avg_temp_impact numeric 🔒	avg_altitude_adjustment numeric 🔒
1	Sea Level	Hot	Track	2827	1187.0	0.949	-0.015
2	High	Cold	Track	1322	1172.2	0.948	-51.784
3	Sea Level	Moderate	Track	64064	1172.1	0.980	-0.231
4	Sea Level	Warm	Track	25193	1167.0	0.971	-0.092
5	Moderate	Warm	Track	2486	1165.2	0.967	-4.996
6	High	Moderate	Track	188	1158.8	0.984	-10.987
7	Moderate	Moderate	Track	2822	1156.2	0.985	-3.975
8	Moderate	Cool	Track	2829	1152.0	0.993	-6.752
9	Moderate	Cool	Road	209	1150.0	0.991	0.000
10	Sea Level	Cool	Track	41821	1148.4	0.991	-0.155
11	Sea Level	Cold	Road	1542	1143.6	0.979	0.000
12	Moderate	Cold	Track	737	1142.8	0.986	-11.414
13	Sea Level	Cool	Road	2817	1136.9	0.992	0.000
14	High	Cool	Track	76	1136.5	0.995	-5.145
15	Sea Level	Moderate	Road	1916	1130.2	0.981	0.000

Total rows: 15

Query complete 00:00:00.259

# OLAP QUERIES

## 2) Geographic Performance Patterns:

- High-level geographic summary
- Operation: ROLL-UP along the venue  $\rightarrow$  city  $\rightarrow$  country hierarchy
- Aggregates total performances, unique athletes, events covered at country level.

Query

Query History

```

1  SELECT
2      v.country_code,
3      COUNT(*) as total_performances,
4      COUNT(DISTINCT f.athlete_key) as unique_athletes,
5      COUNT(DISTINCT f.event_key) as events_covered,
6      ROUND(AVG(f.performance_score), 1) as avg_performance_score,
7      COUNT(CASE WHEN f.performance_score > 1000 THEN 1 END) as elite_performances,
8      ROUND(AVG(v.altitude), 0) as avg_altitude
9  FROM dwh.fact_performance f
10 JOIN dwh.dim_venue v ON f.venue_key = v.venue_key
11 WHERE v.country_code IS NOT NULL AND v.country_code != 'Unknown'
12 GROUP BY v.country_code
13 HAVING COUNT(*) >= 100
14 ORDER BY avg_performance_score DESC
15 LIMIT 15;

```

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

🔍

📶

📶

📶

SQL

	country_code character varying (3)	total_performances bigint	unique_athletes bigint	events_covered bigint	avg_performance_score numeric	elite_performances bigint	avg_altitude numeric
1	GD	294	155	22	1181.9	294	111
2	BE	5493	2709	28	1180.6	5402	29
3	CH	12637	4151	34	1176.6	12429	431
4	NO	4531	2474	33	1176.2	4441	27
5	PU	144	111	15	1175.5	141	640
6	KE	1357	609	20	1170.6	1305	1697
7	QA	3863	1995	25	1164.8	3763	11
8	TC	1311	584	26	1163.1	1309	186
9	MA	1759	1188	29	1160.6	1666	61
10	BO	209	130	16	1159.1	206	1302
11	IT	10878	4458	36	1156.8	10271	152
12	JM	4784	908	20	1156.2	4759	64
13	HR	2017	1282	24	1150.2	1924	132
14	GR	5603	2503	27	1150.1	5358	75
15	CG	229	175	18	1148.6	225	284

Total rows: 15

Query complete 00:00:00.429

# OLAP QUERIES

### 3) Temporal Performance Trends

- Performance evolution over time and seasons
- Operation: DRILL-DOWN – Breaks down date dimension into multiple levels :
  - 1.Level 1: Decade (2010s, 2020s)
  - 2.Level 2: Season (Indoor, Outdoor)
- The GROUP BY Event Category is used to show how each category performs within those time buckets.

Query

Query History

```

1  SELECT
2      d.decade,
3      d.season,
4      e.event_category,
5      COUNT(*) as performances,
6      ROUND(AVG(f.performance_score), 1) as avg_score,
7      ROUND(AVG(f.temperature_impact_factor), 3) as avg_temp_impact,
8      COUNT(CASE WHEN f.performance_score > 1000 THEN 1 END) as elite_count
9  FROM dwh.fact_performance f
10 JOIN dwh.dim_date d ON f.date_key = d.date_key
11 JOIN dwh.dim_event e ON f.event_key = e.event_key
12 WHERE d.year >= 2010
13 GROUP BY d.decade, d.season, e.event_category
14 HAVING COUNT(*) >= 50
15 ORDER BY d.decade, e.event_category, avg_score DESC;

```

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	decade character varying (10) 🔒	season character varying (20) 🔒	event_category character varying (20) 🔒	performances bigint 🔒	avg_score numeric 🔒	avg_temp_impact numeric 🔒	elite_count bigint 🔒
1	2010s	Outdoor	Field	24771	1073.7	0.991	18825
2	2010s	Indoor	Field	2712	1038.1	0.992	1625
3	2010s	Outdoor	Road	2581	1132.3	0.985	2581
4	2010s	Indoor	Road	636	1127.3	0.986	636
5	2010s	Outdoor	Track	48042	1158.2	0.981	46417
6	2010s	Indoor	Track	3305	1052.3	0.981	2103
7	2020s	Outdoor	Field	7807	1072.1	0.991	5818
8	2020s	Indoor	Field	991	1043.9	0.991	613
9	2020s	Outdoor	Road	967	1140.0	0.987	958
10	2020s	Indoor	Road	418	1131.6	0.986	409
11	2020s	Outdoor	Track	17309	1160.4	0.981	16750
12	2020s	Indoor	Track	2283	1076.9	0.981	1649

Total rows: 12

Query complete 00:00:00.167

# OLAP QUERIES

## 4) Venue Specialization Analysis

- Comparison of venue strengths across event types
- Operation: PIVOT - Rotates event\_group dimension from rows to columns

Query

Query History

```

1 SELECT
2     v.venue_name,
3     v.country_code,
4     v.altitude_category,
5     COUNT(*) as total_performances,
6
7     ROUND(AVG(CASE WHEN e.event_group = 'Sprint' THEN f.performance_score END), 1) as sprint_avg,
8     ROUND(AVG(CASE WHEN e.event_group = 'Distance' THEN f.performance_score END), 1) as distance_avg,
9     ROUND(AVG(CASE WHEN e.event_group = 'Jumps' THEN f.performance_score END), 1) as jumps_avg,
10    ROUND(AVG(CASE WHEN e.event_group = 'Throws' THEN f.performance_score END), 1) as throws_avg,
11
12    ROUND(AVG(f.performance_score), 1) as overall_avg
13 FROM dwh.fact_performance f
14 JOIN dwh.dim_venue v ON f.venue_key = v.venue_key
15 JOIN dwh.dim_event e ON f.event_key = e.event_key
16 WHERE v.venue_name != 'Unknown'
17 GROUP BY v.venue_name, v.country_code, v.altitude_category
18 HAVING COUNT(*) >= 50
19        AND COUNT(DISTINCT e.event_group) >= 2
20 ORDER BY overall_avg DESC
21 LIMIT 15;

```

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🔍

📥

📈

SQL

	venue_name character varying (100)	country_code character varying (3)	altitude_category character varying (20)	total_performances bigint	sprint_avg numeric	distance_avg numeric	jumps_avg numeric	throws_avg numeric	overall_avg numeric
1	Jsera Catholic Hs, San Juan Capistrano, Ca (Usa)	US	Sea Level	234	1162.4	1203.7	1134.8	[null]	1201.5
2	Parliament Hill Fields Athletics Track, London (Gbr)	GB	Sea Level	51	[null]	1205.0	[null]	915.3	1199.3
3	Herzogenaurach (Ger)	DE	Sea Level	110	1170.3	1217.8	[null]	955.5	1196.4
4	Zürich (Sui)	CH	Sea Level	3436	1179.9	1211.9	1154.4	1129.2	1185.4
5	Berlin (Gdr)	GD	Sea Level	105	1179.1	1207.9	1207.6	1179.2	1185.2
6	Sechseläutenplatz, Zürich (Sui)	CH	Sea Level	74	[null]	[null]	1169.6	1161.0	1185.1
7	Oslo (Nor)	NO	Sea Level	2157	1168.9	1214.0	1152.7	1106.5	1184.8
8	Bruxelles (Bel)	BE	Sea Level	2661	1169.1	1219.2	1155.7	1117.7	1183.9
9	Boudewijnstadion, Bruxelles (Bel)	BE	Sea Level	2539	1179.5	1218.6	1151.8	1114.3	1183.6
10	Tilburg (Ned)	NL	Sea Level	77	1129.6	1207.0	[null]	1077.8	1183.1
11	Jena (Gdr)	GD	Sea Level	189	1169.9	1203.6	1178.4	1177.9	1180.1
12	Suhaim Bin Hamad Stadium, Doha (Qat)	QA	Sea Level	1336	1184.2	1215.2	1153.9	1118.5	1179.4
13	Stade Delort, Marseille (Fra)	FR	Sea Level	100	1124.4	1206.0	1119.8	1045.1	1178.8
14	Stade Olympique De La Pontaise, Lausanne (Sui)	CH	Sea Level	4050	1178.9	1211.7	1153.3	1122.0	1177.8
15	Lausanne (Sui)	CH	Sea Level	667	1176.6	1206.4	1142.7	1110.2	1177.6

Total rows: 15

Query complete 00:00:00.515

	venue_name character varying (100) 🔒	country_code character varying (3) 🔒	altitude_category character varying (20) 🔒	total_performances bigint 🔒	sprint_avg numeric 🔒	distance_avg numeric 🔒	jumps_avg numeric 🔒	throws_avg numeric 🔒	overall_avg numeric 🔒
1	Jserra Catholic Hs, San Juan Capistrano, Ca (Usa)	US	Sea Level	234	1162.4	1203.7	1134.8	[null]	1201.5
2	Parliament Hill Fields Athletics Track, London (Gbr)	GB	Sea Level	51	[null]	1205.0	[null]	915.3	1199.3
3	Herzogenaurach (Ger)	DE	Sea Level	110	1170.3	1217.8	[null]	955.5	1196.4
4	Zürich (Sui)	CH	Sea Level	3436	1179.9	1211.9	1154.4	1129.2	1185.4
5	Berlin (Gdr)	GD	Sea Level	105	1179.1	1207.9	1207.6	1179.2	1185.2
6	Sechseläutenplatz, Zürich (Sui)	CH	Sea Level	74	[null]	[null]	1169.6	1161.0	1185.1
7	Oslo (Nor)	NO	Sea Level	2157	1168.9	1214.0	1152.7	1106.5	1184.8
8	Bruxelles (Bel)	BE	Sea Level	2661	1169.1	1219.2	1155.7	1117.7	1183.9
9	Boudewijnstadion, Bruxelles (Bel)	BE	Sea Level	2539	1179.5	1218.6	1151.8	1114.3	1183.6
10	Tilburg (Ned)	NL	Sea Level	77	1129.6	1207.0	[null]	1077.8	1183.1
11	Jena (Gdr)	GD	Sea Level	189	1169.9	1203.6	1178.4	1177.9	1180.1
12	Suhaim Bin Hamad Stadium, Doha (Qat)	QA	Sea Level	1336	1184.2	1215.2	1153.9	1118.5	1179.4
13	Stade Delort, Marseille (Fra)	FR	Sea Level	100	1124.4	1206.0	1119.8	1045.1	1178.8
14	Stade Olympique De La Pontaise, Lausanne (Sui)	CH	Sea Level	4050	1178.9	1211.7	1153.3	1122.0	1177.8
15	Lausanne (Sui)	CH	Sea Level	667	1176.6	1206.4	1142.7	1110.2	1177.6



# OLAP QUERIES

## 5) Athlete Performance Ranking

- Top performing athletes with contextual environmental data
- Operation: DRILL-DOWN - From event category → specific event name

```
Query  Query History
1  WITH top_performers AS (
2      SELECT
3          a.athlete_name,
4          a.nationality_code,
5          e.event_name,
6          f.performance_score,
7          v.venue_name,
8          v.altitude,
9          w.temperature,
10         d.year,
11         ROW_NUMBER() OVER (PARTITION BY e.event_name ORDER BY f.performance_score DESC) as rank_in_event
12     FROM dwh.fact_performance f
13     JOIN dwh.dim_athlete a ON f.athlete_key = a.athlete_key
14     JOIN dwh.dim_event e ON f.event_key = e.event_key
15     JOIN dwh.dim_venue v ON f.venue_key = v.venue_key
16     JOIN dwh.dim_weather w ON f.weather_key = w.weather_key
17     JOIN dwh.dim_date d ON f.date_key = d.date_key
18     WHERE f.performance_score > 1000
19 )
20 SELECT
21     event_name,
22     athlete_name,
23     nationality_code,
24     performance_score,
25     venue_name,
26     altitude,
27     temperature,
28     year,
29     rank_in_event
30 FROM top_performers
31 WHERE rank_in_event <= 3 -- Top 3 per event
32 ORDER BY event_name, rank_in_event;
```



# OLAP QUERIES

## 5° Query Result (partial):

Data Output		Messages		Notifications					
<div><div><div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div></div><div>SQL</div></div></div>									
	event_name character varying (50)	athlete_name character varying (100)	nationality_code character varying (3)	performance_score numeric (10,3)	venue_name character varying (100)	altitude integer	temperature numeric (5,2)	year integer	rank_in_event bigint
1	10000 Metres	Kenenisa Bekele	ETH	1300.177	Boudewijnstadion, Bruxelles (Bel)	28	18.16	2005	1
2	10000 Metres	Kenenisa Bekele	ETH	1292.614	Eugene, Or (Usa)	132	15.50	2008	2
3	10000 Metres	Paul Tergat	KEN	1290.931	Bruxelles (Bel)	28	18.16	1997	3
4	10000 Metres Race Walk	Evan Dunfee	CAN	1198.742	Alexander Stadium, Birmingham (Gbr)	149	26.43	2022	1
5	10000 Metres Race Walk	Kevin Campion	FRA	1198.359	Paris (Fra)	42	18.63	2013	2
6	10000 Metres Race Walk	Walter Arena	ITA	1197.841	Catania (Ita)	40	12.57	1990	3
7	1000 Metres	Noah Ngeny	KEN	1250.119	Rieti (Ita)	395	18.15	1999	1
8	1000 Metres	Maria De Lurdes Mutola	MOZ	1248.311	Bruxelles (Bel)	28	18.16	1995	2
9	1000 Metres	Sebastian Coe	GBR	1248.071	Oslo (Nor)	26	16.68	1981	3
10	100 Metres	Usain Bolt	JAM	1355.947	Olympiastadion, Berlin (Ger)	43	19.00	2009	1
11	100 Metres	Usain Bolt	JAM	1341.408	Olympic Stadium, London (Gbr)	25	18.10	2012	2
12	100 Metres	Usain Bolt	JAM	1324.054	National Stadium, Beijing (Chn)	49	25.53	2008	3
13	100 Metres Hurdles	Kendra Harrison	USA	1259.099	Olympic Stadium, London (Gbr)	25	18.71	2016	1
14	100 Metres Hurdles	Kendra Harrison	USA	1252.733	Hayward Field, Eugene, Or (Usa)	132	12.68	2016	2
15	100 Metres Hurdles	Ginka Zagorcheva	BUL	1251.144	Drama (Gre)	122	25.40	1987	3
16	10 Kilometres	Rhonex Kipruto	KEN	1285.018	Valencia (Esp)	10	-2.12	2020	1
17	10 Kilometres	Joyciline Jepkosgei	KEN	1273.179	Praha (Cze)	202	13.70	2017	2
18	10 Kilometres	Joshua Cheptegei	UGA	1272.597	Valencia (Esp)	10	0.38	2019	3
19	10 Kilometres Race Walk	Luke Adams	AUS	1198.576	Beijing (Chn)	49	20.43	2010	1
20	10 Kilometres Race Walk	Chilsung Park	KOR	1197.986	Beijing (Chn)	49	20.43	2010	2
21	10 Kilometres Race Walk	Perseus Karlström	SWE	1197.986	Madrid (Esp)	665	16.67	2022	3
22	10 Miles Road	Charles Waweru Kamathi	KEN	1224.721	Amsterdam (Ned)	13	15.02	2002	1
23	10 Miles Road	Keisuke Morita	JPN	1224.721	Kosa (Jpn)	179	7.30	2022	2
24	10 Miles Road	Samuel Kamau Wanjiru	KEN	1223.734	Kosa (Jpn)	179	7.30	2005	3
25	110 Metres Hurdles	Aries Merritt	USA	1294.108	Boudewijnstadion, Bruxelles (Bel)	28	15.02	2012	1
26	110 Metres Hurdles	Grant Holloway	USA	1292.526	Hayward Field, Eugene, Or (Usa)	132	15.50	2021	2
27	110 Metres Hurdles	Xiang Liu	CHN	1281.477	Stade Olympique De La Pontaise, Lausanne (Sui)	453	22.00	2006	3
28	1500 Metres	Hicham El Guerrouj	MAR	1302.444	Stadio Olimpico, Roma (Ita)	52	24.45	1998	1
29	1500 Metres	Hicham El Guerrouj	MAR	1301.412	Bruxelles (Bel)	28	18.16	2001	2
30	1500 Metres	Bernard Lagat	KEN	1299.520	Bruxelles (Bel)	28	18.16	2001	3
31	15 Kilometres	Paul Tergat	KEN	1208.802	La Courneuve (Fra)	38	12.72	1994	1
32	15 Kilometres	Emmanuel Moi Maru	KEN	1207.641	Valencia (Esp)	10	20.84	2021	2
33	15 Kilometres	Todd Williams	USA	1203.002	Jacksonville. Fl (Usa)	218	16.36	1995	3
Total rows: 129		Query complete 00:00:01.418							

# OLAP QUERIES

## 6) Environmental Optimization Insights

- Best environmental conditions for each event type
- Operation: DRILL-ACROSS – analyzes the same measures (performance\_score) across different dimensional perspectives:
  1. Event Group × Altitude Category
  2. Event Group × Temperature Category

```
Query  Query History
1  SELECT
2      e.event_group,
3      'Optimal Altitude' as condition_type,
4      v.altitude_category as optimal_value,
5      COUNT(*) as sample_size,
6      ROUND(AVG(f.performance_score), 1) as avg_performance
7  FROM dwh.fact_performance f
8  JOIN dwh.dim_event e ON f.event_key = e.event_key
9  JOIN dwh.dim_venue v ON f.venue_key = v.venue_key
10 WHERE v.altitude_category IS NOT NULL
11 GROUP BY e.event_group, v.altitude_category
12 HAVING COUNT(*) >= 100
13
14 UNION ALL
15
16 SELECT
17     e.event_group,
18     'Optimal Temperature',
19     w.temperature_category,
20     COUNT(*),
21     ROUND(AVG(f.performance_score), 1)
22 FROM dwh.fact_performance f
23 JOIN dwh.dim_event e ON f.event_key = e.event_key
24 JOIN dwh.dim_weather w ON f.weather_key = w.weather_key
25 WHERE w.temperature_category IS NOT NULL
26 GROUP BY e.event_group, w.temperature_category
27 HAVING COUNT(*) >= 100
28
29 ORDER BY event_group, avg_performance DESC;
```

## 6° Query Result:

# OLAP QUERIES

Data Output		Messages	Notifications			
	event_group character varying (30)	condition_type text	optimal_value character varying (20)	sample_size bigint	avg_performance numeric	
1	Distance	Optimal Temperature	Hot	1359	1203.7	
2	Distance	Optimal Temperature	Moderate	25875	1184.8	
3	Distance	Optimal Temperature	Warm	8419	1176.9	
4	Distance	Optimal Altitude	High	1101	1172.8	
5	Distance	Optimal Altitude	Moderate	2384	1158.3	
6	Distance	Optimal Altitude	Sea Level	58013	1155.2	
7	Distance	Optimal Temperature	Cool	21082	1135.7	
8	Distance	Optimal Temperature	Cold	4763	1034.4	
9	Distance (Road)	Optimal Temperature	Cold	1643	1141.5	
10	Distance (Road)	Optimal Temperature	Cool	3040	1137.6	
11	Distance (Road)	Optimal Altitude	Sea Level	6988	1134.6	
12	Distance (Road)	Optimal Altitude	Moderate	527	1133.5	
13	Distance (Road)	Optimal Temperature	Moderate	2069	1129.7	
14	Distance (Road)	Optimal Temperature	Warm	776	1121.2	
15	Hurdles	Optimal Temperature	Hot	513	1175.6	
16	Hurdles	Optimal Temperature	Moderate	12622	1174.8	
17	Hurdles	Optimal Altitude	Sea Level	24250	1173.9	
18	Hurdles	Optimal Temperature	Cool	7026	1172.9	
19	Hurdles	Optimal Temperature	Warm	5433	1171.7	
20	Hurdles	Optimal Altitude	Moderate	1733	1167.6	
21	Hurdles	Optimal Temperature	Cold	502	1165.3	
22	Hurdles	Optimal Altitude	High	113	1163.2	
23	Jumps	Optimal Temperature	Hot	387	1147.8	
24	Jumps	Optimal Temperature	Moderate	8067	1144.6	
25	Jumps	Optimal Altitude	Sea Level	15833	1144.0	
26	Jumps	Optimal Temperature	Cool	4831	1143.7	
27	Jumps	Optimal Temperature	Warm	3738	1142.4	
28	Jumps	Optimal Altitude	Moderate	1418	1142.2	
29	Jumps	Optimal Temperature	Cold	288	1136.1	
30	Sprint	Optimal Temperature	Hot	959	1169.2	
31	Sprint	Optimal Temperature	Warm	13904	1158.6	
32	Sprint	Optimal Temperature	Moderate	28577	1157.7	

33	Sprint	Optimal Altitude	Sea Level	56455	1157.6
34	Sprint	Optimal Temperature	Cool	16618	1154.7
35	Sprint	Optimal Altitude	High	449	1153.7
36	Sprint	Optimal Altitude	Moderate	4761	1151.1
37	Sprint	Optimal Temperature	Cold	1607	1150.3
38	Throws	Optimal Temperature	Hot	1073	1104.0
39	Throws	Optimal Temperature	Moderate	26177	1074.9
40	Throws	Optimal Altitude	Moderate	5743	1074.1
41	Throws	Optimal Altitude	Sea Level	58913	1072.4
42	Throws	Optimal Temperature	Cool	22983	1071.9
43	Throws	Optimal Temperature	Warm	11429	1071.5
44	Throws	Optimal Temperature	Cold	3120	1049.3
45	Throws	Optimal Altitude	High	126	1031.4
Total rows: 45		Query complete 00:00:00.285			

The background is a dark gradient with white, wavy, grid-like lines that create a sense of depth and movement. These lines are concentrated in the corners and edges, leaving the center relatively clear for the text.

**THANK YOU**