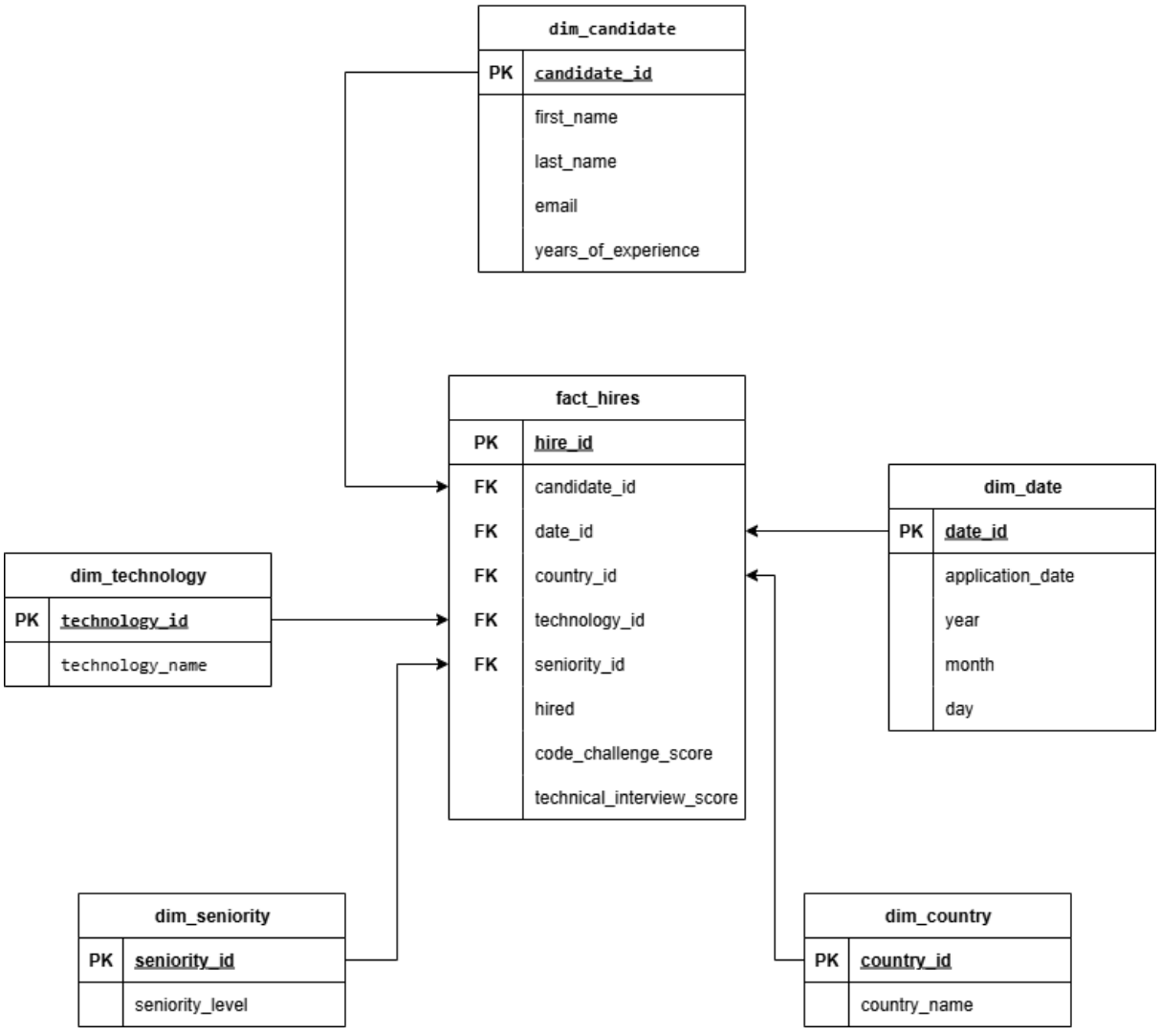
Dimensional Model of Hires: Star Schema

The presented dimensional model follows a **star schema**, widely used in **Data Warehouses (DW)** to optimize the analysis of large datasets and the generation of key performance indicators (KPIs). This design separates quantitative metrics from descriptive attributes, enabling efficient and consistent queries.



Model Components

1. Fact Table: `fact_hires`

The central table records each hiring event and contains:

- **Primary Key**: `hire_id` — unique identifier for each record.
- **Foreign Keys**: `candidate_id`, `date_id`, `country_id`, `technology_id`, `seniority_id`, linking the fact table to the corresponding dimensions.
- **Quantitative metrics**:
    - `hired`: binary indicator reflecting an effective hire based on pre-defined rules (scores ≥ 7 in code challenge and technical interview).
    - `code_challenge_score` and `technical_interview_score`: individual scores for each selection stage.

This structure enables the calculation of strategic KPIs, such as hires by technology, country, seniority, and years of experience, ensuring data consistency and traceability.

2. Candidate Dimension: `dim_candidate`

Provides detailed information about each candidate:

- **PK**: `candidate_id`
- Attributes: `first_name`, `last_name`, `email`, `years_of_experience` (YOE)

Enables the analysis of hires by experience and ensures the unique identification of each candidate in the DW.

3. Date Dimension: `dim_date`

Supports precise temporal analysis:

- **PK**: `date_id`
- Attributes: `application_date`, `year`, `month`, `day`

Fundamental for constructing time-based KPIs, hiring trends per year or month, and historical analyses.

4. Technology Dimension: `dim_technology`

Classifies candidates according to the evaluated technology area:

- **PK**: `technology_id`
- Attribute: `technology_name`

Allows segmentation of hires by technology or specialized roles, such as Python, DevOps, or Game Development.

5. Seniority Dimension: `dim_seniority`

Categorizes candidates' professional levels:

- **PK**: `seniority_id`
- Attribute: `seniority_level` (Intern, Junior, Mid-level, Senior, Lead, Architect)

Enables the evaluation of average performance per seniority level and comparisons across different professional levels.

6. Country Dimension: `dim_country`

Describes the geographical location of candidates:

- **PK**: `country_id`
- Attribute: `country_name`

Essential for geographical analysis of hires and segmentation by strategic countries (U.S., Brazil, Colombia, Ecuador).

Academic Justification of the Design

Criterion	Justification
Controlled denormalization	Dimensions are normalized to reduce redundancy, while the schema remains flat enough for fast queries.
Scalability	The model can grow in data volume without compromising query performance.
Analytical flexibility	Enables multiple slicing, aggregation, and segmentation options for diverse KPIs.

Business rule enforcement	The `hired` logic is applied during transformation, ensuring integrity and consistency in the DW.
Clear separation of metrics and context	Metrics reside in the fact table, while descriptive context is in the dimensions, following best practices in dimensional modeling.

KPIs and Metrics

From this model, the following KPIs can be generated:

- Hires by **technology** (`technology_name`)
- Hires by **year** (`year`)
- Hires by **seniority level** (`seniority_level`)
- Hires by **country and year** (`country_name`, `year`)
- Hires by **years of experience** (`years_of_experience`)
- Average scores by **seniority level** (`AVG(code_challenge_score)`, `AVG(technical_interview_score)`)

This approach allows the creation of comprehensive dashboards, analytical reports, and data-driven decision-making.