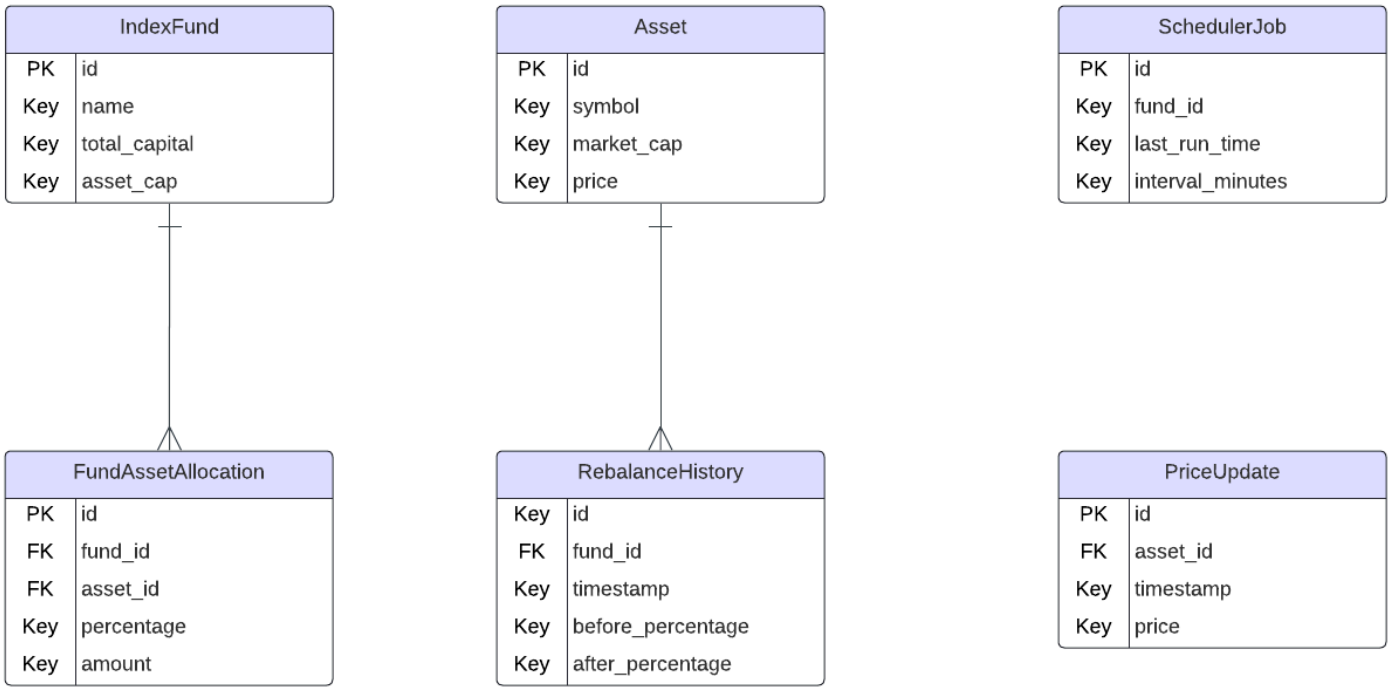

Description of Entities:

- **IndexFund**: Stores the basic information about each index fund.
- **Asset**: Stores information about each asset, including its market cap and current price.
- **FundAssetAllocation**: Represents the allocation of each asset within a specific index fund.
- **RebalanceHistory**: Logs each rebalancing event, showing the percentage and amount of changes before and after the rebalance.
- **SchedulerJob**: Manages the scheduling of the rebalance checks.
- **PriceUpdate**: Logs historical price changes for assets.

2. Flow of Events When the Scheduler Runs

Each time the scheduler runs, the system checks the asset allocations in each index fund to see if a rebalance is required based on price fluctuations. If rebalancing is necessary, the system automatically performs it.

Steps for Rebalancing:

1. **Scheduler Trigger**:
    - The scheduler runs at regular intervals (based on `interval_minutes` in the `SchedulerJob` table).
    - It checks the `IndexFund` entities for any funds requiring rebalancing.
2. **Fetch Current Prices**:
    - The system fetches the current prices of assets from an external source (e.g., Binance, CoinMarketCap).
    - These prices are logged in the `PriceUpdate` table.
3. **Rebalance Check**:
    - For each index fund, the system checks if the current allocation deviates from the target allocation (based on the latest prices).
    - It calculates the current allocation and compares it with the target allocation (using the algorithm designed earlier).
4. **Determine Rebalance Need**:
    - If the percentage deviation of any asset exceeds a predefined threshold (e.g., 2% deviation), a rebalance is triggered.
5. **Rebalance Execution**:
    - The system recalculates the allocation for each asset based on the current market caps and prices.
    - The changes are recorded in the `RebalanceHistory` table, including the before and after allocation percentages and amounts.
6. **Log the Rebalance**:
    - Each time the fund is rebalanced, the details of the rebalance (before and after values) are logged in the `RebalanceHistory` table.
7. **Update Fund Asset Allocations**:
    - The `FundAssetAllocation` table is updated with the new asset allocations after the rebalance.
8. **Scheduler Logs Completion**:
    - The `SchedulerJob` table has been updated with the timestamp for the last run.

Description of the Scheduler Flow:

1. **Scheduler Trigger**: The scheduler runs at predefined intervals (e.g., every 30 minutes) to check for potential rebalancing needs.
2. **Fetch Prices**: Current prices of assets are fetched and logged.
3. **Check Allocations**: The system calculates current allocations and compares them with the target allocations for each index fund.
4. **Needs Rebalance?**: If the deviation exceeds the allowed threshold, a rebalance is triggered. If not, the system simply logs the scheduler run.
5. **Execute Rebalance**: The system recalculates asset allocations and updates the `FundAssetAllocation` table.
6. **Log Rebalance**: The rebalance is logged in the `RebalanceHistory` table for audit purposes.
7. **Update Asset Allocations**: The new allocation percentages and amounts are stored in the `FundAssetAllocation` table.