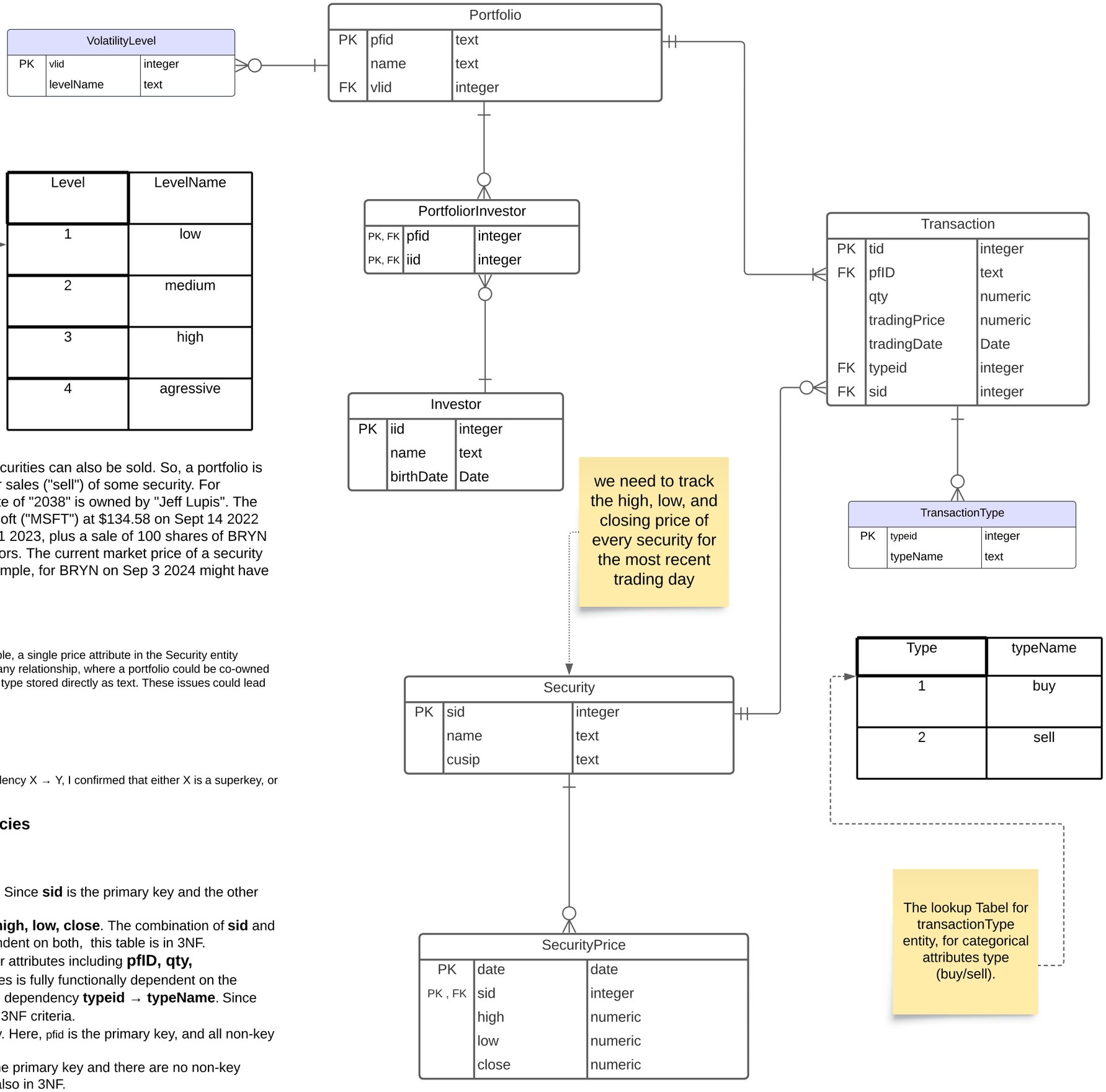
CS5200 - ASSIGNMENT 04.1 - Normaliza a Schema
Shorena K. Anzhilov
Summer 2025



A portfolio is a collection of securities acquired through purchases. Securities can also be sold. So, a portfolio is associated with numerous "transactions" that are purchases ("buy") or sales ("sell") of some security. For example, portfolio "Education" with a volatity of "high" and a target date of "2038" is owned by "Jeff Lupis". The portfolio had several transactions: a purchase of 500 shares of Microsoft ("MSFT") at $134.58 on Sept 14 2022 and 1000 shares of Berkshire Hathaway ("BRYN") at 332.95 on Oct 11 2023, plus a sale of 100 shares of BRYN at 408.17 on Mar 18 2024. A portfolio can be co-owned by two investors. The current market price of a security on the most recent date traded is a triple of high/low/close, so, for example, for BRYN on Sep 3 2024 might have been 431.5/430.25/431.15

My notes:

The original schema was not in 3NF. It had issues like multivalued attributes for example, a single price attribute in the Security entity included high, low, and close values in one field. Another problem was the many-to-many relationship, where a portfolio could be co-owned by two investors. There were also categorical values such as volatility and transaction type stored directly as text. These issues could lead to redundancy, anomalies, or data inconsistency.

To fix this, I decomposed the schema:
1. Created a SecurityPrice table to store daily high, low, close, and date values.
2. Used lookup tables for TransactionType and VolatilityLevel.
3. Added a join table PortfolioInvestor to resolve the many-to-many relationship.
The table was then checked to make sure it satisfies 3NF. For every functional dependency X → Y, I confirmed that either X is a superkey, or Y is part of a candidate key

Proof of 3NF with Functional Dependencies

Functional dependencies for each table are as follows:

In the **Security** table, the functional dependency is **sid → name, cusip.** Since **sid** is the primary key and the other attributes depend only on it, the table satisfies 3NF.
In the **SecurityPrice** table, the functional dependency is **(sid, date) → high, low, close**. The combination of **sid** and **date** forms a composite primary key, and the price values are fully dependent on both, this table is in 3NF.
For the **Transaction** table, **tid** is the primary key and determines all other attributes including **pfID, qty, tradingPrice, tradingDate, typeid,** and **sid.** Each of these attributes is fully functionally dependent on the primary key, the table is in 3NF. The **TransactionType** table contains the dependency **typeid → typeName**. Since **typeid** is the primary key and determines **typeName,** the table meets 3NF criteria.
The **Portfolio** table has **pfid → name, vlid** as its functional dependency. Here, pfid is the primary key, and all non-key attributes depend entirely on it. This confirms that Portfolio is in 3NF.
In the **PortfolioInvestor** table, the composite key **(pfid, iid)** serves as the primary key and there are no non-key attributes. Since no transitive or partial dependencies exist, this table is also in 3NF.
In the **Investor** table, the dependency is **iid → name, birthDate.** With iid as the primary key and both attributes fully dependent on it, this table likewise complies with 3NF.
The **VolatilityLevel** table defines the dependency **vlid → levelName.** As **vlid** is the primary key and **levelName** depends entirely on it, this table is also in 3NF.

In summary, all tables have been analyzed, and their attributes are either fully functionally dependent on their primary keys or are part of a composite key with no transitive or partial dependencies. Based on the analysis above, the entire schema is fully normalized and satisfies the requirements of Third Normal Form (3NF).