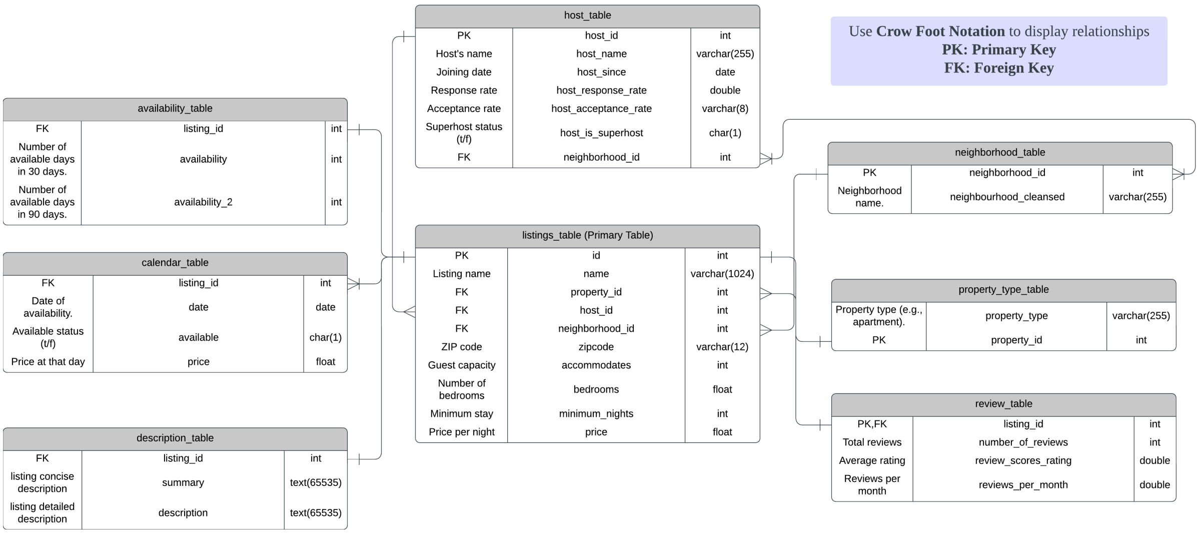
Database Decisions

Overall Structure



host_table

PK	host_id	int
Host's name	host_name	varchar(255)
Joining date	host_since	date
Response rate	host_response_rate	double
Acceptance rate	host_acceptance_rate	varchar(8)
Superhost status (t/f)	host_is_superhost	char(1)
FK	neighborhood_id	int

availability_table

FK	listing_id	int
Number of available days in 30 days.	availability	int
Number of available days in 90 days.	availability_2	int

calendar_table

FK	listing_id	int
Date of availability.	date	date
Available status (t/f)	available	char(1)
Price at that day	price	float

description_table

FK	listing_id	int
listing concise description	summary	text(65535)
listing detailed description	description	text(65535)

listings_table (Primary Table)

PK	id	int
Listing name	name	varchar(1024)
FK	property_id	int
FK	host_id	int
FK	neighborhood_id	int
ZIP code	zipcode	varchar(12)
Guest capacity	accommodates	int
Number of bedrooms	bedrooms	float
Minimum stay	minimum_nights	int
Price per night	price	float

Use **Crow Foot Notation** to display relationships
PK: Primary Key
FK: Foreign Key

neighborhood_table

PK	neighborhood_id	int
Neighborhood name.	neighbourhood_cleansed	varchar(255)

property_type_table

Property type (e.g., apartment).	property_type	varchar(255)
PK	property_id	int

review_table

PK,FK	listing_id	int
Total reviews	number_of_reviews	int
Average rating	review_scores_rating	double
Reviews per month	reviews_per_month	double

1. Indicate the Primary Table in the Database

The `listings_table` serves as the primary table in this database. It is the central table that connects all other tables, as it holds critical information about each listing, such as its ID, name, associated host, neighborhood, and property type.

This table is important because:

- It acts as a **hub** for relationships between key entities like hosts, neighborhoods, and properties.
- Foreign keys from other tables, such as `calendar_table`, `description_table`, `availability_table` and `review_table`, reference `listings_table`, making it the foundation for querying detailed information about listings and their attributes.
- It enables efficient data management and integration, ensuring consistency and easy access to all listing-related data.

2. Discuss Other Table Relationships

- `host_table` → `listings_table`:
    - Relationship: One-to-Many (1 host can have many listings).
    - Importance: Links host-specific data (e.g., response rate, superhost status) to the listings they manage.
- `neighborhood_table` → `listings_table`:

    - Relationship: One-to-Many (1 neighborhood can have many listings).
    - Importance: Associates listings with their geographical areas, enabling location-based analysis.
- `property_type_table` → `listings_table`:
    - Relationship: One-to-Many (1 property type can have many listings).
    - Importance: Classifies listings by property type (e.g., apartment, house), aiding in property-specific queries.
- `availability_table` → `listings_table`:
    - Relationship: One-to-One (One availability records for a single listing).
    - Importance: Tracks availability metrics for each listing.
- `calendar_table` → `listings_table`:
    - Relationship: Many-to-One (Multiple calendar entries for a single listing).
    - Importance: Records daily availability and pricing data, essential for booking systems.
- `description_table` → `listings_table`:
    - Relationship: One-to-One (Each listing has one description).
    - Importance: Provides textual details and summaries for listings, which are essential for evaluating the condition and characteristics of specific listings.
- `review_table` → `listings_table`:
    - Relationship: One-to-One (Each listing has one aggregate review record).
    - Importance: Stores review statistics, such as total reviews and average ratings, which are critical for customer decision-making.

3. Reason for including text data in the database

In this dataset, we utilize the `text` data type in the table `description_table` to store detailed description data. Including this text data is justified for the following reasons:

1. **Essential for Insights and Recommendations**: The original dataset contains lengthy descriptions that are crucial for explaining certain insights and recommendations. This text data provides context and detailed information that cannot be conveyed through structured numerical or categorical data alone.
2. **Efficient Organization**: By isolating the text data in a separate table, we can ensure that it does not negatively impact the performance of lookups and queries in other tables. This approach keeps the database well-structured and optimized.
3. **No Duplication**: Each row in the text data table is unique, corresponding to a specific entry in the primary dataset. This eliminates redundancy and maintains data integrity.

4. Reason for Excluding Specific Tables

Excluded Table: `Reviews` – Contains reviewer IDs and their comments on each listing.

Reason:

- The table is excessively large, which can impact database performance.
- It consists solely of text data, making it difficult to perform broad analyses using SQL without additional tools or preprocessing.

By excluding this table, the database remains more efficient and focused on fields that can be effectively analyzed with SQL.

5. Reason for Excluding Specific Fields

Excluded Fields: `listing_url`, `scrape_id`, `last_scraped`, `experiences_offered`, `neighborhood_overview`, `notes`, `transit`, `access`, `interaction`, `house_rules`, `thumbnail_url`, `medium_url`, `picture_url`, `xl_picture_url`, `host_url`, `host_location`, `host_about`, `host_response_time`, `host_thumbnail_url`, `host_picture_url`, `host_neighbourhood`, `host_verifications`, `host_has_profile_pic`, `host_identity_verified`, `street`, `neighbourhood_group_cleansed`, `city`, `state`, `market`, `smart_location`, `country_code`, `country`

To streamline the dataset for analysis, we excluded certain fields based on the following reasons:

1. **Redundant or Duplicate Information**: Fields like `thumbnail_url`, `medium_url`, and `picture_url` were excluded as they provide unnecessary information.
2. **Single Constant Value**: Fields such as `city`, `country`, and `state` contain only one possible value, adding no meaningful variability to the dataset.
3. **Complex Data Not Easily Analyzed with SQL**: Fields like `amenities` require advanced parsing, which is outside the scope of SQL-based analysis.
4. **Geographical and Policy-Related Fields**: Fields such as `latitude`, `longitude`, and `requires_license` were excluded as they are not essential for the analysis objectives and are challenging to analyze using SQL alone.
5. **Timestamp and Frequency Data**: Fields like `last_scraped` and `calendar_last_scraped` were deemed unnecessary for this analysis.
6. **Overlapping Information**: Fields such as `neighbourhood_group_cleansed` overlap with included columns, making them redundant.
7. **Low Utility Text Data**: Fields like `host_about` were excluded due to their minimal analytical value.