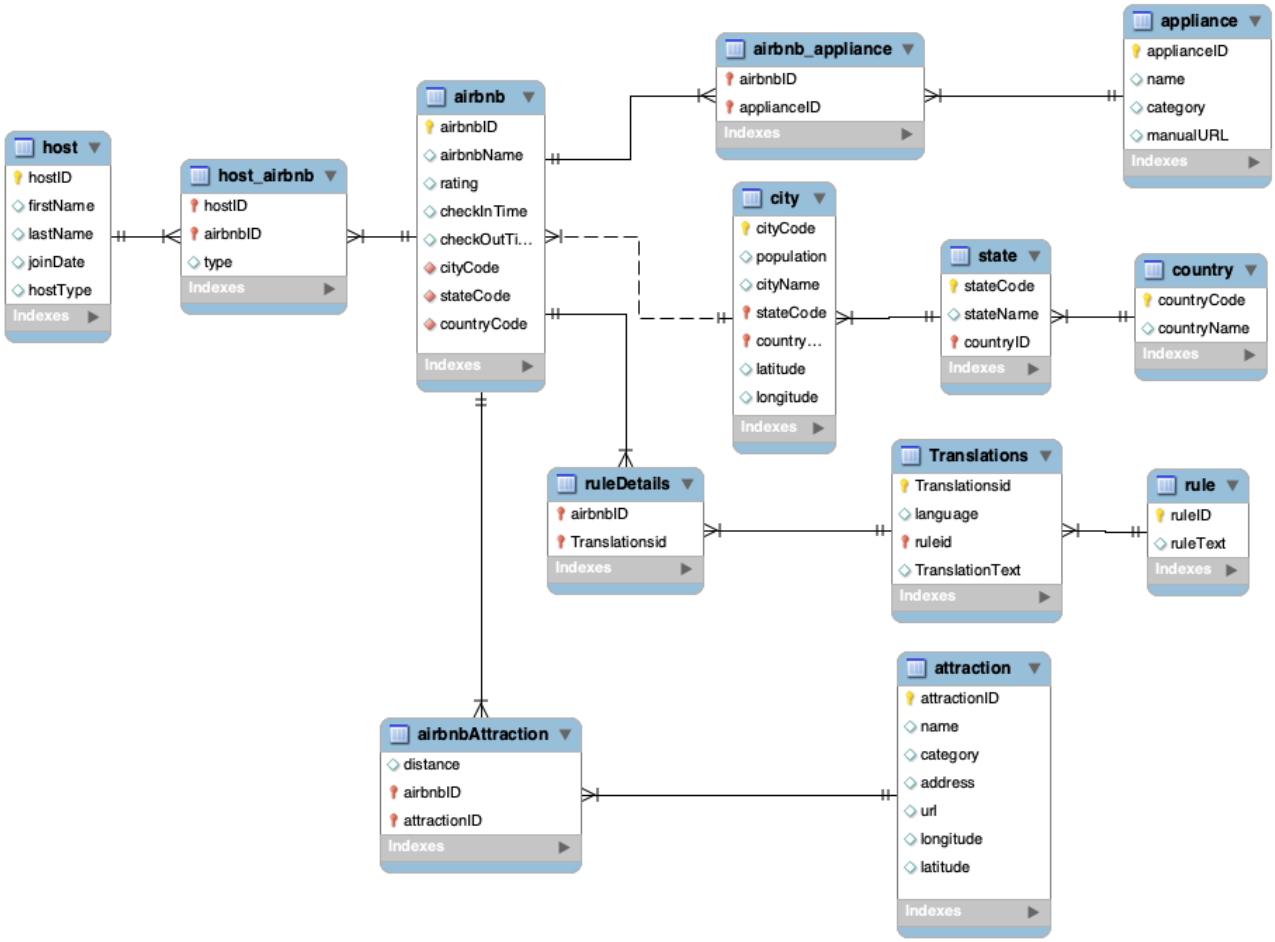
# AirBnB - Data Modeling and SQL Queries

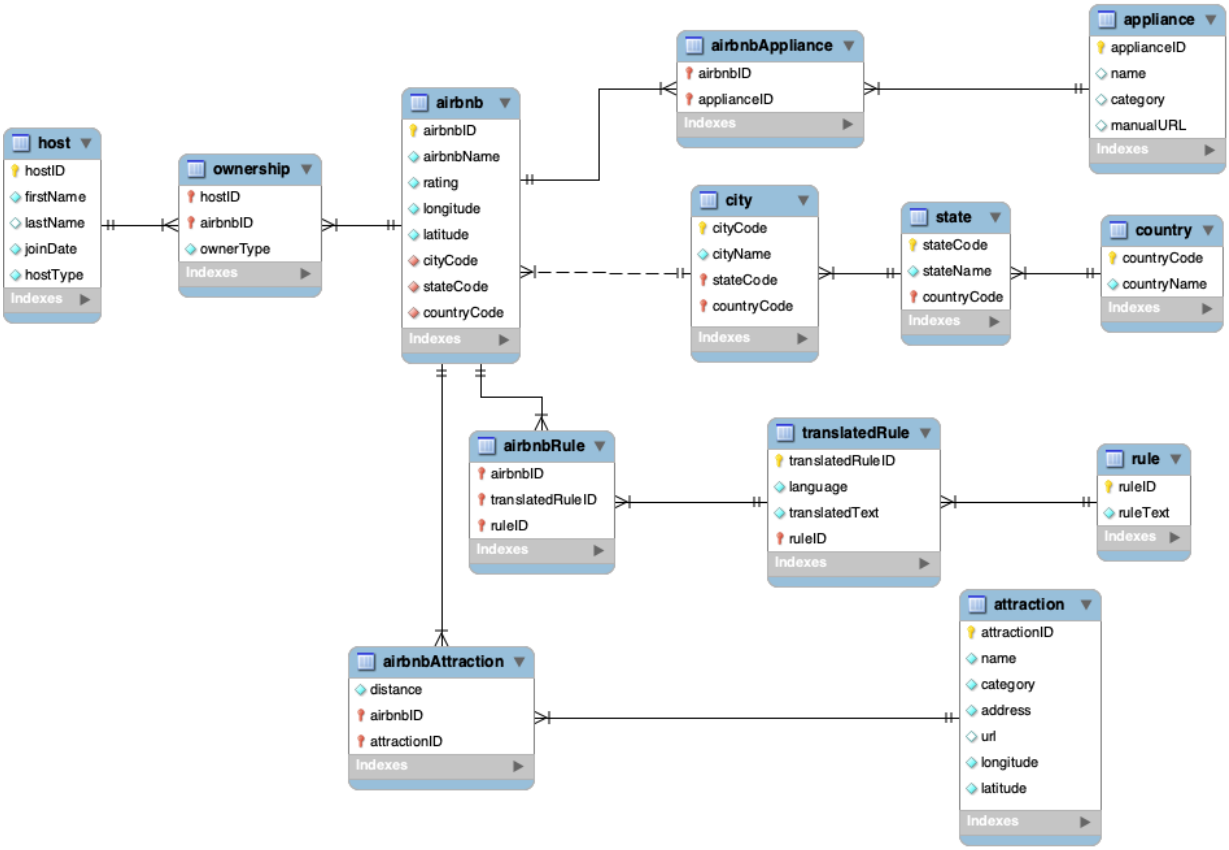
## Description:

AirBnB hosts provide documentation along with their provided units to users. These documents provide various information such as a unit's appliances, a unit's rules, or even what local attractions are close to the AirBnB units. To organize these documentations for different uses, we designed a database and inserted a small amount of data for demonstration purpose. SQL queries were performed against the database to extract relevant information. For example, the distance between a specific Airbnb unit and the nearest museum can be found using the database.

Software: MySQL Workbench

## Data Model




host
hostID
firstName
lastName
joinDate
hostType
Indexes
ownership
hostID
airbnbID
ownerType
Indexes
airbnb
airbnbID
airbnbName
rating
longitude
latitude
cityCode
stateCode
countryCode
Indexes
airbnbAppliance
airbnbID
applianceID
Indexes
appliance
applianceID
name
category
manualURL
Indexes
city
cityCode
cityName
stateCode
countryCode
Indexes
state
stateCode
stateName
countryCode
Indexes
country
countryCode
countryName
Indexes
airbnbRule
airbnbID
translatedRuleID
ruleID
Indexes
translatedRule
translatedRuleID
language
translatedText
ruleID
Indexes
rule
ruleID
ruleText
Indexes
airbnbAttraction
distance
airbnbID
attractionID
Indexes
attraction
attractionID
name
category
address
url
longitude
latitude
Indexes

## Queries:

#2: A host “Walter” wants to check the difference between the rating of his airbnb units that have superhosts as cohosts and the average rating of all airbnbs, also include airbnbID and hostID, sort by airbnb ratings.

>Create View s AS (SELECT avg(rating) AS av FROM host, ownership, airbnb WHERE host.hostType <> "superhost" AND host.hostID = ownership.hostID AND ownership.airbnbID = airbnb.airbnbID);
> SELECT ownership.airbnbID, host.hostID as cohostID, airbnb.rating, (airbnb.rating-s.av) AS difference from s, host, ownership, airbnb where host.hostID = ownership.hostID AND ownership.airbnbID = airbnb.airbnbID AND host.hostType = "superhost" AND ownership.type ="cohost" AND ownership.airbnbID IN (SELECT airbnbID FROM host, ownership WHERE host.hostID = ownership.hostID AND host.firstName = "Walter") Order By rating DESC

```
CREATE VIEW AVE (average) AS
   (SELECT
      AVG(rating)
   FROM
      Airbnb);

SELECT
   ownership.airbnbID,
   host.hostID,
   airbnb.rating,
   (airbnb.rating - AVE.average) AS difference
FROM
   ownership,
   host,
   airbnb,
   calc
WHERE
   host.hostID = ownership.hostID
      AND ownership.airbnbID = airbnb.airbnbID
      AND host.hostType = 'superhost'
      AND ownership.Type = 'cohost'
      AND ownership.airbnbID IN (SELECT
         airbnbID
      FROM
         host,
         ownership
      WHERE
         host.hostID = ownership.hostID
            AND host.firstName = 'Walter')
ORDER BY rating DESC
```

```
+ ------------ + ------------ + ----------- + --------------- +
| airbnbID     | hostID       | rating      | difference      |
+ ------------ + ------------ + ----------- + --------------- +
| 105          | 102          | 5.00        | 0.990909        |
| 104          | 103          | 3.50        | -0.509091       |
| 103          | 102          | 5.00        | 0.990909        |
+ ------------ + ------------ + ----------- + --------------- +
3 rows
```

#3: International health inspectors want to make sure that each rule that has the word “pet” or “smoking” is available in at least 3 different languages since they are the most common rules. So, the number of the translated rules for those is calculated.

```
> SELECT rule.ruleID, ruleText, Count(TranslatedRuleid) AS TranslationCount FROM TranslatedRule, rule WHERE rule.ruleID = TranslatedRule.ruleid AND rule.ruleID IN (SELECT rule.ruleID from rule WHERE ruleText REGEXP "pet|smoking") GROUP BY rule.ruleID
```

```
+ ----------- + ------------- + --------------------- +
| ruleID      | ruleText      | TranslationCount      |
+ ----------- + ------------- + --------------------- +
| 81101       | No pets allowed | 3                   |
| 81102       | No smoking    | 3                     |
+ ----------- + ------------- + --------------------- +
2 rows
```

#4: An Airbnb host is curious about the average ratings of airbnb listings in different U.S. cities, the ratings are arranged in descending order.

Execute:

```
> SELECT
   airbnb.cityCode, ROUND(AVG(rating), 2) AS meanRating
FROM
   airbnb
      JOIN
   country ON airbnb.countryCode = country.countryCode
WHERE
   countryName = 'United States'
GROUP BY airbnb.countryCode , airbnb.cityCode (wrong)
ORDER BY meanrating DESC
```

```
+ ------------- + --------------- +
| cityCode      | meanRating      |
+ ------------- + --------------- +
```

```
| ATH          | 4.83          |
| NYC          | 2.75          |
+ ------------- + ---------------- +
2 rows
```

#5: The host “Daniel” wants to check if there exists any AirBnB that he owns has rates above the average rate 3.0 using EXISTS.

Execute:

```
> SELECT
   firstName, lastName, airbnbName, rating
FROM
   airbnb
      JOIN
   ownership ON airbnb.airbnbID = ownership.airbnbID
      JOIN
   host ON ownership.hostID = host.hostID
WHERE
   EXISTS( SELECT
         *
      FROM
         host
      WHERE
         host.hostID = ownership.hostID
            AND ownership.airbnbID = airbnb.airbnbID
            AND firstName = 'Daniel'
            AND rating > 3.0)
```

```
+ -------------- + ------------- + ---------------- + ------------ +
| firstName      | lastName      | airbnbName       | rating       |
+ -------------- + ------------- + ---------------- + ------------ +
| Daniel         | Doobey        | DelightfulDen    | 3.50         |
+ -------------- + ------------- + ---------------- + ------------ +
1 rows
```

#6: An AirBnB hosts wants to check if any of AirBnBs in Athens, GA has historic attractions nearby, if so, what are the attraction names, Airbnb names and their corresponding distances?

Execute:

```
> SELECT category, name, distance, airbnbName, cityCode
FROM attraction
JOIN airbnbAttraction
ON attraction.attractionID = airbnbAttraction.attractionID
JOIN airbnb
```

```
ON airbnbAttraction.airbnbID = airbnb.airbnbID
GROUP BY category, name, distance, airbnbName, cityCode
HAVING category = "Historic"
AND cityCode = (select cityCode from city where cityName = “Athens”
AND stateCode = (select stateCode from state where stateName = “Georgia”
AND countryCode = (select countryCode from country where countryName = “US”))
AND countryCode = (select countryCode from country where countryName = “US”))
```

```
+ ------------ + --------- + ------------ + --------------- + ------------ +
| category     | name      | distance     | airbnbName      | cityCode     |
+ ------------ + --------- + ------------ + --------------- + ------------ +
| Historic     | Camak House | 5            | BatCaveNY       | ATH          |
| Historic     | Camak House | 12           | SuperHome       | ATH          |
+ ------------ + --------- + ------------ + --------------- + ------------ +
2 rows
```

#7 A airbnb customer wants to know the distances between his/her interested airbnb listings (BatCaveNY , SuperHome, RunAwayFromYourTroubles) and local Museum, Aqurium or Market related attractions if any, the distances are arranged in ascending order.

Execute:

```
> SELECT
   airbnbName, name, distance
FROM
   attraction
      JOIN
   airbnbAttraction ON attraction.attractionID = airbnbAttraction.attractionID
      JOIN
   airbnb ON airbnbAttraction.airbnbID = airbnb.airbnbID
WHERE
   airbnbName IN ('BatCaveNY' , 'SuperHome', 'RunAwayFromYourTroubles')
      AND name REGEXP 'Museum|Aqurium|Market'
ORDER BY distance
```

```
+ --------------- + --------- + ------------ +
| airbnbName      | name      | distance     |
+ --------------- + --------- + ------------ +
| BatCaveNY       | Museum of Modern Art | 5            |
| SuperHome       | Museum of Modern Art | 13           |
| RunAwayFromYourTroubles | Namdaemun Market | 15           |
+ --------------- + --------- + ------------ +
3 rows
```

#8 A airbnb customer wants to find the AirBnB with the most attractions within 5 miles.

Execute:

```
> SELECT
   airbnbName, COUNT(airbnb.airbnbID) AS numberOfAttractions
FROM
   airbnb
      JOIN
   airbnbAttraction ON airbnbAttraction.airbnbID = airbnb.airbnbID
      JOIN
   attraction ON attraction.attractionID = airbnbAttraction.attractionID
WHERE
   distance <= 5
GROUP BY airbnb.airbnbID
ORDER BY COUNT(airbnb.airbnbID) DESC
LIMIT 1
```

```
+ --------------- + ----------------------- +
| airbnbName     | numberOfAttractions   |
+ --------------- + ----------------------- +
| BatCaveNY     | 3                     |
+ --------------- + ----------------------- +
1 rows
```

#9 A airbnb customer wants to find which AirBnBs have ratings greater than 3.0 and more than 2 appliances provided.

Execute:

```
> SELECT
   airbnbName,
   rating,
   COUNT(appliance.applianceID) AS numberOfAppliances
FROM
   airbnb
      JOIN
   airbnbAppliance ON airbnbAppliance.airbnbID = airbnb.airbnbID
      JOIN
   appliance ON appliance.applianceID = airbnbAppliance.applianceID
GROUP BY airbnb.airbnbID
HAVING rating > 3
   AND COUNT(appliance.applianceID) > 2
```

```
+ --------------- + ----------- + ----------------------- +
| airbnbName     | rating              | numberOfAppliances   |
+ --------------- + ----------- + ----------------------- +
```

| CozyCorner | 4.50 | 4 |
| DelightfulDen | 3.50 | 3 |
| BatCaveNY | 5.00 | 4 |
| SuperHome | 5.00 | 3 |
| RunAwayFromYourTroubles | 4.50 | 6 |
| DragonDen | 4.30 | 4 |
| MoonlightVilla | 4.00 | 3 |

+ ---------- + ---------- + ---------- +
7 rows

#10 A airbnb customer wants to find AirBnBs with ratings above average of 3.1. He/she also wants to see how much percentages the corresponding ratings are higher than the average.

Execute:
```
> SELECT airbnbName, ((rating-3.1)/3.1)*100) AS higher_Rating_Percent, rating
FROM airbnb
WHERE rating>3.1
ORDER BY higher_Rating_Percent DESC
```

| airbnbName | higher_Rating_Percent | rating |
| --- | --- | --- |
| BatCaveNY | 61.290323 | 5.00 |
| SuperHome | 61.290323 | 5.00 |
| CozyCorner | 45.161290 | 4.50 |
| RunAwayFromYourTroubles | 45.161290 | 4.50 |
| DragonDen | 38.709677 | 4.30 |
| MoonlightVilla | 29.032258 | 4.00 |
| DelightfulDen | 12.903226 | 3.50 |
| CheeseCastle | 12.903226 | 3.50 |

8 rows

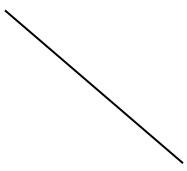
#1: A user wants to find the mandatory rules that appear in every airbnb and read all available translatedRule for those rules together with the individual rule/translationID for better implementation while ordered by languages.

Execute:
```
> SELECT
   language, ruleid, TranslatedRuleid, TranslationText
FROM
   TranslatedRule
```

```
WHERE
   ruleid IN (SELECT
         rule.ruleid
      FROM
         rule
      WHERE
         NOT EXISTS( SELECT
               *
            FROM
               airbnb
            WHERE
               NOT EXISTS( SELECT
                     *
                  FROM
                     airbnbRule,
                     TranslatedRule
                  WHERE
                     airbnb.airbnbID = airbnbRule.airbnbID
                        AND TranslatedRule.ruleid = rule.ruleid
                        AND airbnbRule.TranslatedRuleid = TranslatedRule.TranslatedRuleid)))
ORDER BY language
```

```
+ ------------ + ---------- + ------------------ + -------------------- +
| language     | ruleid     | TranslatedRuleid   | TranslationText      |
+ ------------ + ---------- + ------------------ + -------------------- +
| english      | 81101      | 61101              | No pets allowed      |
| english      | 81102      | 61102              | No smoking           |
| german       | 81101      | 61109              | Keine Haustiere erlaubt. |
| german       | 81102      | 61110              | Rauchen im Apartment ist verboten. |
| spanish      | 81101      | 61105              | No se admiten animales de compañía |
| spanish      | 81102      | 61106              | No fumar             |
+ ------------ + ---------- + ------------------ + -------------------- +
6 rows
```

**Condensed SQL Query Feature Matrix**



| Query<br>SQL Feature | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Multiple Table Join | x | x | x | x | | x | x | x | x | |
| Subquery | x | x | x | | x | x | | | | x |
| Correlated Subquery | | | | | x | | | | | |
| Group by | | | x | x | | | | x | x | x |
| Group by with Having | | | | | | x | | | x | |
| Order by | x | x | | x | | | x | x | | x |
| Divide | x | | | | | | | | | |
| In or Not In | x | x | x | | | | | | | |
| Built In Function/ Calculated Field | | x | x | x | | | | x | x | x |
| Regexp | | | x | | | | x | | | |
| Exists or Not Exist | x | | | | x | | | | | |