[**1. Research on Cities/Halls across Countries 1**](#_dqmun1bbwjuv)

[Analysis 1](#_svsr2dcmlu3w)

[Calculations 2](#_saatpi22uzug)

[**2. Database Tables Size 2**](#_ir92qg8g46o6)

[city 2](#_3g067lshkr6d)

[movie 2](#_mo5bj012w4ix)

[movie\_features 3](#_b5d018yri8w2)

[Movie\_language 3](#_c7trgl32q1kp)

[Theatre 3](#_lixm4ijt2anu)

[Theatre\_halls 3](#_fd0u266s4cs4)

[Halls 4](#_tirvy3a57w45)

[Shows 4](#_u669k85ixqd0)

[Theatre\_shows 4](#_30z0c0pmggbk)

[hall\_features 4](#_snlcp36hl0m3)

[seats 5](#_aorwj99tj3e4)

[Show\_seat 5](#_906db22e0xjw)

[Booking 5](#_f4vs381rqnmp)

[Booking\_payment 6](#_fnr854iirdbv)

[Booking\_seats 7](#_jjhx28rwh5zi)

[users 7](#_c6luzmjq6lx7)

[All Tables 8](#_mn1nyk1p68ah)

[**3. HDFS 8**](#_lsmpfjnn44az)

[Theatre Booking Database Sizing Artifact 1](#_5ic1cojp57mt)

# 1. Research on Cities/Halls across Countries

## Analysis

|  |  |  |
| --- | --- | --- |
| S No | Topic | Result |
| 1 | Highest cities available across all the countries. | Russia has max no. of cities **[20,000]** |
| 2 | Highest no. of available Halls(screens) across all the countries. | China has max no. of Halls(screens) **[53,000]** |

## Calculations

|  |  |  |
| --- | --- | --- |
| S No | Topic | Criteria |
| 1 | Data Size | All the Calculations made for a region for the next **[10 yrs]**  considering how the system data grows. |
| 2 | Cities | All the Calculations made for **[20k]** cities |
| 3 | Halls | All the Calculations made for **[1,00,000 Halls(screens)]** keep threshold double for calculation and derive half 53,000 at the end. |
| 4 | Theatre | Assuming 1 Theatre has 2 Halls on an average, and so Total Theatres are **[50k]** |
| 5 | Shows | Assuming 1 Hall has 4 shows per day on an average,so Total no. of shows is **[1.5 billion]** |
| 6 | Data types | Please refer to Database schema design artifacts for Database tables and Data types used in this document. |
| 7 | Database Table type | Used **[UT8MB4]** Character Set wherever required. |

# 2. Database Tables Size

## city

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 8+8+8+1+200 = 300 Bytes | 300 byte \* 20000 cities | 6 MB |

## movie

**Assumption**

Add weekly 1 movie for 1,00,000 halls for next 10 yrs

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 8+8+8+8+1+200 = 300 Bytes | 300 byte \* 5.2 billion | 1.56 TB |

## movie\_features

**Assumption**

1 movie can have average of 5 features

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 8+1 = 10 Bytes | 10 bytes \* 26 billion | 260 GB |

## Movie\_language

**Assumption**

1 movie can have average of 5 Languages

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 8+1 = 10 Bytes | 10 bytes \* 26 billion | 260 GB |

## Theatre

**Assumption**

Total Bytes per row increased little to have some more threshold for varchar fields.

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 1+8+8+8+8+200+400 = 633 bytes = 800 bytes | 800 bytes \* 50k Theatres | 40 MB |

## Theatre\_halls

**Assumption**

Total Bytes per row increased little to have some more threshold for varchar fields.

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 8+8 = 16 bytes | 16 bytes \* 1L Halls | 2 MB |

## Halls

**Assumption**

Total Bytes per row increased little to have some more threshold for varchar fields.

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 8+8+8+1+200 = 225 bytes = 300 bytes | 300 bytes \* 1L | 30 MB |

## Shows

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 8+8+8+1+200 = 225 bytes = 300 bytes | 60 bytes \* 1.5 billion | 90 GB |

## Theatre\_shows

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 8+8 = 16 bytes | 16 bytes \* 1.5 billion | 24 GB |

## hall\_features

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 8+8 = 16 bytes | 8 bytes \* 5 L | 4 MB |

## seats

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 4+4+1+8+8+8+8 = 41 bytes | 41 bytes \* 30 Million | 2 GB |

## Show\_seat

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 1+8+8+8+8+8 = 41 bytes | 41 bytes \* 45,000 trillion | 1.845 billion GB |

**Assumption**

Considering show seats don't need to be in the database for a long time,we can keep only 6 months show seats available and other data beyond that period can be removed. Below is the calculation.

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (6 months) |
| 1 | 1+8+8+8+8+8 = 41 bytes | 41 bytes \* 2,250 trillion | 93 TB |

## Booking

**Assumption**

A booking might average 2 tickets per booking

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 4+8+8+8+8+8+8+1 = 53 bytes | 53 bytes \* 22,500 trillion | 1.1925 exabytes |

**Assumption**

Considering booking doesn't need to be in the database for a long time,so we can keep only 6 months booking available and other data beyond that period can be removed, Below is the calculation.

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (6 months) |
| 1 | 4+8+8+8+8+8+8+1 = 53 bytes | 53 bytes \* 1,125 trillion | 60 PB |

## Booking\_payment

**Assumption**

A booking might average 2 tickets per booking

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 8+8 = 16 bytes | 16 bytes \* 32,500 trillion | 520 PB |

**Assumption**

Considering booking payment doesn't need to be in the database for a long time,so we can keep only 6 months booking available and other data beyond that period can be removed, Below is the calculation.

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (6 months) |
| 1 | 8+8 = 16 bytes | 16 bytes \* 1,625 trillion | 26 PB |

## 

## Booking\_seats

**Assumption**

A booking might average 2 tickets per booking

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 8+8 = 16 bytes | 16 bytes \* 45,000 trillion | 720 PB |

**Assumption**

Considering booking payment doesn't need to be in the database for a long time,so we can keep only 6 months booking available and other data beyond that period can be removed, Below is the calculation.

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (6 months) |
| 1 | 8+8 = 16 bytes | 16 bytes \* 2,250 trillion | 36 PB |

## users

**Assumption**

30 million Active users referenced from existing successfully running systems.

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Total Bytes per row | Calculation | Total Bytes (10 yrs) |
| 1 | 8+8+8+8+200+200+8+8+8+8+200+200+200+200 = 1264 bytes | 1264 bytes \* 30 million | 40 GB |

## All Tables

**Assumption**

* Show seats and Booking related data considered for 6 months since removed after that period.
* All other data for complete 10 yrs.

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Halls(Screens) / Region | Total Bytes (10 yrs) | Shard by City [660 cities] |
| 1 | 1L | 122 PB | 185 TB |

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Halls(Screens) / Region | Total Bytes (10 yrs) | Shard by City [660 cities] |
| 1 | 50k | 61 PB | 93 TB |

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Halls(Screens) / Region | Total Bytes (1 yrs) | Shard by City [660 cities] |
| 1 | 50k | 6 PB | 10 TB |

# 3. HDFS

* Some of the data tables will grow to exabytes of data in the next 10 years, so we need a system to store all this data in a distributed manner.
* Move all the stale data after 6 months or 1 year depends upon the requirement to the HDFS system.