**Seminar**

**Hotel Chain**

**Team BHV**

**Members:**

**Trần Gia Bảo 15520044**

**Đào Minh Hiếu 15520219**

**Hoàng Trọng Việt 15521009**

**Distributed Database Schema**

1.1         Global Database Tables

We have six global tables. They are***Hotel, Room, Booking*** *and* ***Customers***. Their conceptual schemas are described below.

Table 1. global tables

|  |
| --- |
| **Global Tables** |
| * ***Hotel* (**hotel\_id (varchar 50), hotel\_name (varchar 100), hotel type (float), hotel\_description (varchar max)). * ***Rooms*** (room\_id (int), hotel\_id (varchar 50), room\_number (varchar 50), room\_type (varchar 50), room\_status (binary), room\_price (float)). * ***Booking***(booking\_id (int), room\_id (int), hotel\_id (varchar 50), customer\_id (int), checkin\_date (date), checkout\_date (date), booking\_type (binary), total\_payment (float)). * ***Customer***(customer\_id (int), customer\_name (varchar 100), customer\_mobile (int), customer\_email (varchar 100)). |

1.2         Table Description

* **Hotel: hotel\_id (varchar 50), hotel\_name (varchar 100), hotel type (float), hotel\_description (varchar max).**

Table 2. Description of Table “**Hotel Attributes**”

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Type** | **Value** |
| **Hotel\_id (Pkey)** | The ID of the hotel | Varchar(50) | 1-100 no duplicate |
| **Hotel\_name** | The name of hotel | Varchar(100) | A string name |
| **Hotel\_type** | The type of the hotel | Float | The number of stars. |
| **Hotel\_description** | The description of the hotel | Varchar(max) | Budget, location and further information. |

Number of records: 100

* **Room: (room\_id (int), hotel\_id (varchar 50), room\_number (varchar 50), room\_type (varchar 50), room\_status (binary), room\_price (float).**

Table 3. Description of Table “Customer”

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Type** | **Value** |
| **Room\_id (Pkey)** | The ID of room in the system. | int | 0001 => 1000, no duplicate |
| **Hotel\_id (Fkey)** | The ID of the hotel | Varchar(50) | 001 => 100, no duplicate |
| **Room\_number** | Number of each room | Varchar(50) |  |
| **Room\_type** | The type of the room | Varchar(50) | * Type 1: normal. * Type 2: business. * Type 3: VIP. |
| **Room\_status** | The status of the room | Binary | * 1: available * 0: not available |
| **Room\_price** | Room’s price depend on type. | Float | * 100$ * 200$ * 300$ |

Number of records: 1000

* **Booking: booking\_id (int), room\_id (int), hotel\_id (varchar 50), customer\_id (int), checkin\_date (date), checkout\_date (date), booking\_type (binary), total\_payment (float).**

Table 4. Description of Table “Book”

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Type** | **Value** |
| **Booking\_id (Pkey)** | The ID of booking service | int | 000001 => 100000, no duplicate |
| **Room\_id (Fkey)** | The ID of the room | int | 0001 => 1000, no duplicate |
| **Hotel\_id (key)** | The ID of the hotel | Varchar(50) | 001 => 100, no duplicate |
| **Customer\_id** | The ID of the customer | int | 0000001 => 1000000, no duplicate |
| **Checkin\_date** | The date of checking-in | Datetime |  |
| **Checkout\_date** | The date of checking-out | Datetime |  |
| **Booking\_type** | The type of booking | Binary | * 1: Online * 0: Offline |
| **Total\_payment** | The amount of payment after checking-out | Float |  |

Number of records: 100000

* **Customers: customer\_id (int), customer\_name (varchar 100), customer\_mobile (int), customer\_email (varchar 100).**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Type** | **Value** |
| **Customer\_id** | The ID of customer | int | 0000001 => 1000000, no duplicate |
| **Customer\_name** | The name of the customer | Varchar(100) |  |
| **Customer\_mobile** | The phone number of the customer | int |  |
| **Customer\_email** | The email of customer | Varchar(100) |  |

Number of records: 1000000

1.3         Fragmentation

* **Hotel: hotel\_id (varchar 50), hotel\_name (varchar 100), hotel type (float), hotel\_description (varchar max).**

Table 6. Vertical Fragmentation of Table “**Hotel**”

|  |  |
| --- | --- |
| **Fragmentation Name** | **Fragmentation Condition** |
| **Hotel.1** | Hotel\_id < 50 and Hotel\_description = “VN” |
| **Hotel.2** | Hotel\_id >= 50 and Hotel\_description = “USA” |
| **Hotel.3** | Hotel\_id > 50 and Hotel\_description = “ENG” |

* **Room: room\_id (int), hotel\_id (varchar 50), room\_number (varchar 50), room\_type (varchar 50), room\_status (binary), room\_price (float).**

Table 7. Horizontal Fragmentation of Table “**Rooms**”

|  |  |
| --- | --- |
| **Fragmentation Name** | **Fragmentation Condition** |
| **Rooms.1** | Room\_id < 600 |
| **Rooms.2** | Room\_id >= 600 AND Room\_id < 900 |
| **Rooms.3** | Room\_id >= 900 |

* **Booking: booking\_id (int), room\_id (int), hotel\_id (varchar 50), customer\_id (int), checkin\_date (date), checkout\_date (date), booking\_type (binary), total\_payment (float).**

Table 8. Horizontal Fragmentation of Table “Customer”

|  |  |
| --- | --- |
| **Fragmentation Name** | **Fragmentation Condition** |
| ***Booking*.1** | Booking\_id < 50.000  AND  Room\_id < 600 |
| ***Booking.*2** | Booking\_id >= 50.000 AND Booking\_id < 90.000 AND  Room\_id >= 600 AND Room\_id < 900 |
| ***Booking*.3** | Booking\_id >= 90.000 AND Room\_id >= 900 |

* **Customers: customer\_id (int), customer\_name (varchar 100), customer\_mobile (int), customer\_email (varchar 100).**

|  |  |
| --- | --- |
| **Fragmentation Name** | **Fragmentation Condition** |
| **Customer.1** |  |
| **Customer.2** |  |
| **Customer.3** |  |

1.4         Allocation

Site Configuration: 3 sites deployed at 3 computers.

* **Scheme1: Basic Fragmentation**

Table 1. Allocation Scheme of Basic Fragmentation

|  |  |
| --- | --- |
| **At Site Name** | **Fragmentation Name** |
| **DB1 at Site 1** | Hotel.1, Room.1, Booking.1, Customer.1 |
| **DB2 at Site 2** | Hotel.2, Room.2, Booking.2, Customer.2 |
| **DB3 at Site 3** | Hotel.3, Room.3, Booking.3, Customer.3 |