

Homework Question 4

Wide Column Database Design for Flightapp

Keyspace and Column Families

1. Keyspace

- **Primary Key Structure:** Composite keys are used to uniquely identify rows.

2. Column Families

- **Flights:** Stores flight information.
- **Users:** Stores user information.
- **Itineraries:** Stores itinerary information.
- **Reservations:** Stores reservation details.

Table and Column Family Structure

1. Flights Table

- **Row Key:** `flight_id`
- **Column Family:** `details`
 - **Columns:** `carrier_id`, `origin`, `destination`, `departure_time`, `arrival_time`, `capacity`, `reservable_capacity`
- **Example:**
 - Row Key: FL123
 - Column Family: details
 - Columns:
 - `carrier_id`: C1
 - `origin`: City A
 - `destination`: City B
 - `departure_time`: 2024-05-25T08:00:00Z
 - `arrival_time`: 2024-05-25T10:00:00Z
 - `capacity`: 100
 - `reservable_capacity`: 90

2. Users Table

- **Row Key:** `user_id`
- **Column Family:** `profile`
 - **Columns:** `username`, `password`, `balance`
- **Example:**
 - Row Key: U123
 - Column Family: profile
 - Columns:
 - `username`: john_doe

- password: hashed_password
- balance: 200.00

3. Itineraries Table

- **Row Key:** `itinerary_id`
- **Column Family:** `details`
 - **Columns:** `origin`, `destination`, `is_direct`, `flights` (serialized array of flight ids)
- **Example:**
 - Row Key: I123
 - Column Family: details
 - Columns:
 - origin: City A
 - destination: City C
 - is_direct: false
 - flights: [FL123, FL456]

4. Reservations Table

- **Row Key:** `reservation_id`
- **Column Family:** `details`
 - **Columns:** `user_id`, `is_paid`, `reservation_date`, `flights` (serialized array of flight ids)
- **Example:**
 - Row Key: R123
 - Column Family: details
 - Columns:
 - user_id: U123
 - is_paid: false
 - reservation_date: 2024-05-25T09:00:00Z
 - flights: [FL123]

Operations and Queries

1. Create User

- **Operation:** Insert into `Users` table
- **Command:**

```
cbt set user_id:U123 profile:username=john_doe
profile:password=hashed_password profile:balance=200.00
```

- **Rowkey and Columns Used:**
 - Rowkey: U123
 - Columns: username, password, balance

2. Search Itineraries

- **Operation:** Query **Itineraries** table
- **Command:**

```
cbt read Itineraries prefix=CityA#CityB
```

- **Rowkey and Columns Used:**
 - Rowkey: CityA#CityB
 - Columns: origin, destination, is_direct, flights

3. Reserve Flights

- **Operation:** Insert into **Reservations** table, update **Flights** table to decrement **reservable_capacity**
- **Command:**

```
cbt set reservation_id:R123 details:user_id=U123
details:is_paid=false details:reservation_date=2024-05-
25T09:00:00Z details:flights=[FL123]
cbt set flight_id:FL123 details:reservable_capacity=89
```

- **Rowkey and Columns Used:**
 - Reservation Rowkey: R123, Columns: user_id, is_paid, reservation_date, flights
 - Flight Rowkey: FL123, Columns: reservable_capacity

4. Pay for Reservation

- **Operation:** Update **Reservations** table to mark as paid, update **Users** table to decrement balance
- **Command:**

```
cbt set reservation_id:R123 details:is_paid=true
cbt set user_id:U123 profile:balance=180.00
```

- **Rowkey and Columns Used:**
 - Reservation Rowkey: R123, Columns: is_paid
 - User Rowkey: U123, Columns: balance

5. List Reservations

- **Operation:** Query **Reservations** table by **user_id**
- **Command:**

```
cbt read Reservations prefix=U123
```

- **Rowkey and Columns Used:**

- Rowkey: U123
- Columns: reservation details

Explicit Timestamps

- Explicit timestamps are typically used for versioning data. For Flightapp, we assume the default timestamp behavior provided by Bigtable is sufficient, unless specific version control is required for operations like auditing or historical data analysis.

Implementation Notes

- **Reservable Capacity Management:** Updated directly in the **Flights** table during reservation operations.
- **Serialized Arrays:** Flights in itineraries and reservations are stored as serialized arrays to simplify data retrieval and updates.
- **Indexing:** Use composite row keys to support efficient queries, such as **origin#destination** for itineraries.

This wide column database design leverages the flexibility and scalability of Google Cloud Bigtable to handle the diverse and high-frequency data operations required by Flightapp.