**Student ID: 33878897**

**Student Name: Muhammad Mubashir Shah**

**Assumptions for Ocean Odyssey Database Design**

This document consolidates all assumptions, design decisions, and constraints discussed throughout the database design process for Ocean Odyssey. It covers conceptual modeling, normalization, logical modeling, and surrogate keys.

**Conceptual Model Assumptions**

**Entities and Relationships**

* Operator → Ship:
  + Cardinality: 1..1 Operator to 1..\* Ship.
  + Assumption: An operator can exist without ships (e.g., newly registered), but ships must belong to one operator.
* Ship → Cabin:
  + Cardinality: 1..1 Ship to 1..\* Cabin.
  + Assumption: Cabins are uniquely identified per ship (e.g., Cabin "D1" on Ship A vs. Ship B).
* Ship → Cruise:
  + Cardinality: 1..1 Ship to 1..\* Cruise.
  + Assumption: A cruise cannot exist without a ship.
* Passenger → Manifest:
  + Cardinality: 0..\* Passenger to 1..1 Manifest.
  + Assumption: Passengers can exist without a booking (e.g., registered but not booked).
* Guardianship (Self-Referential):
  + Cardinality: 0..\* Guardian to 1..1 Minor.
  + Assumption: A minor must have one guardian; adults cannot have guardians.

**Normalization Assumptions**

**Task 2 (Cruise Itinerary)**

* UNF to 3NF:
  + Assumption: The sample itinerary’s cost\_per\_person belongs to CRUISE, not port visits.
  + Dependencies:
    - cruise\_id → cruise\_name, cruise\_cost\_per\_person (direct).
    - port\_code → port\_name, port\_latitude, port\_longitude, country\_code (partial).
    - country\_code → country\_name (transitive).

**Logical Model Surrogate Key Assumptions**

**CABIN Table: cabin\_id**

Primary Key: Composite of cabin\_number and ship\_code

Surrogate Key: cabin\_id

**Assumptions & Justifications**

1. Business Need:
   1. Cabins are reused across ships (e.g., "D1" on Ship A and Ship B)
   2. A surrogate key avoids ambiguity in referencing cabins across different ships
2. Schema Impact:
   1. Foreign Key Simplification: The MANIFEST table references cabin\_id instead of (cabin\_number, ship\_code)
3. Unique Constraint: (cabin\_number, ship\_code) is retained to enforce natural key uniqueness

**CRUISE\_PORT\_VISIT Table: visit\_id**

Primary Key: Composite of cruise\_id, port\_code, and datetime

Surrogate Key: visit\_id

**Assumptions & Justifications**

1. Business Need:
   1. A cruise may visit the same port multiple times (e.g., depart and arrive events)
   2. A surrogate key ensures uniqueness without relying on timestamp precision.
2. Schema Impact:
   1. Simplified Joins: Relationships with CRUISE and PORT use visit\_id instead of composite keys
   2. Avoiding Redundancy: Eliminates redundant storage of cruise\_id, port\_code, and datetime in related tables
3. Constraints:
   1. Unique Constraint: (departure\_arrival, cruise\_id, datetime, port\_code) ensures no duplicate port visits

**MANIFEST Table: manifest\_id**

Primary Key: Composite of cruise\_id, passenger\_id , and cabin\_id

Surrogate Key: manifest\_id

**Assumptions & Justifications**

1. Business Need:
   1. A passenger can book multiple cruises, and cabins are reused across cruises.
   2. A surrogate key simplifies tracking individual bookings.
2. Schema Impact:
   1. Streamlined Relationships: The MANIFEST table references passenger\_id and cabin\_id via single-column FKs.
   2. Query Efficiency: Joins on manifest\_id are faster than composite keys.
3. Constraints:
   1. Unique Constraint: (passenger\_id, cruise\_id) ensures a passenger cannot book the same cruise twice.