**Introduction** The Extract, Transform, and Load (ETL) procedures for the data warehouse of Hokie Resort Hotels are the subject of this paper, which outlines the execution of Milestone 2 of Project 2. A complete data warehouse is necessary for Hokie Resort Hotels to assist with strategic decision-making after acquiring two hotel chains with more than 200 sites. This milestone expands upon the dimensional model created in the first milestone by showing how to retrieve data from the two live databases, modify it so it fits the star schema, and then put it into the data warehouse. Revenue optimization, customer satisfaction factors, and cross-selling potential are the three main management problems that the implementation seeks to answer. Data warehouse-to-operational database mapping, SQL ETL processes, and sample data for analyzing hotel-wide business trends are all part of this milestone.  
**Operational Database Analysis**

| **Data Warehouse Table** | **Operational DB Table** | **Aggregation/Summary** |
| --- | --- | --- |
| GuestDim | Corp1: Guest, GuestProfile | No aggregation, each row is an instance in the DW. |
| GuestDim | Corp2: Customer, CustomerAccount | No aggregation, each row is an instance in the DW. |
| HotelDim | Corp1: Property, PropertyDetail | No aggregation, each row is an instance in the DW. |
| HotelDim | Corp2: Hotel, HotelFacilities | No aggregation, each row is an instance in the DW. |
| RoomDim | Corp1: Room, RoomType | No aggregation, each row is an instance in the DW. |
| RoomDim | Corp2: Accommodation, RoomCategory | No aggregation, each row is an instance in the DW. |
| EventDim | Corp1: Event, EventSchedule | No aggregation, each row is an instance in the DW. |
| EventDim | Corp2: Functions, FunctionBooking | No aggregation, each row is an instance in the DW. |
| TimeDim | Generated from date fields in both DBs | No direct source table. Generated from date fields in reservation and booking tables. |
| ReservationFact | Corp1: Booking, BookingDetail, PaymentTransaction | Aggregated. Room revenue summed from BookingDetail. Ancillary revenue summed from PaymentTransaction. |
| ReservationFact | Corp2: Reservation, StayDetail, ServiceCharge | Aggregated. Room charges summed from StayDetail. Ancillary charges summed from ServiceCharge tables. |

The transformation process involves several key operations:

1. **Dimension Loading**: For dimension tables (GuestDim, HotelDim, RoomDim, EventDim), the transformation primarily involves mapping fields from operational tables to the dimension tables without significant aggregation. However, some derived attributes are included:
   * In GuestDim, TotalStays is calculated by counting bookings for each guest
   * In HotelDim, TotalRooms is determined by counting the rooms associated with each hotel
2. **Fact Table Aggregation**: For the ReservationFact table, several aggregations are performed:
   * TotalRoomRevenue is calculated by summing room charges from BookingDetail/StayDetail tables
   * TotalAncillaryRevenue is calculated by summing amounts from PaymentTransaction/ServiceCharge tables with the type 'Ancillary'
3. **Time Dimension**: The TimeDim table is populated based on dates from the operational databases, with additional attributes derived from the date values:
   * DayOfWeek, DayName, MonthNumber, MonthName, Quarter, and Year are all derived from the date
   * IsHighSeason is determined based on the month (June-September and December are marked as high season)
4. **Foreign Key Resolution**: The ETL process resolves relationships between dimension and fact tables by looking up dimension keys based on business keys from the operational systems

**ETL Procedures**Complete SQL code for all ETL procedures has been implemented and saved as separate files:

- Load\_GuestDim\_From\_Corp1.sql (4 KB)

- Load\_HotelDim\_From\_Corp2\_Procedure.sql (3 KB)

- Load\_ReservationFact\_From\_Corp1\_Procedure.sql (7 KB)

- Load\_ReservationFact\_From\_Corp2\_Procedure.sql (8 KB)

These procedure files contain the full implementation details including cursor declarations, variable handling, transformation logic, and data loading operations. Key sections of each procedure are described below, with the complete code available in the reference files. **Load\_GuestDim\_From\_Corp1 Procedure**

This procedure extracts guest data from Corp1's Guest and GuestProfile tables and loads it into the GuestDim table. It uses a cursor to iterate through guest records, looking up information like loyalty tier and calculating total stays. For each guest, it either inserts a new record or updates an existing one based on the GuestID.

**Load\_HotelDim\_From\_Corp2 Procedure**

This procedure extracts hotel data from Corp2's Hotel and HotelFacilities tables and loads it into the HotelDim table. It retrieves information such as hotel details, star ratings, and calculates the total number of rooms. Similar to the guest procedure, it handles both inserts and updates based on the HotelID.

**Load\_ReservationFact\_From\_Corp1 Procedure**

This procedure populates the ReservationFact table with data from Corp1's operational database. It joins several tables (Booking, BookingDetail, Room, PaymentTransaction, EventBooking, and Event) to collect all the necessary information. The procedure:

* Aggregates room revenue and ancillary revenue
* Looks up dimension keys from the dimension tables
* Handles dates by either using existing TimeKey values or creating new ones
* Inserts reservation records with proper foreign keys to dimension tables

**Load\_ReservationFact\_From\_Corp2 Procedure**

Similar to the Corp1 procedure, this one extract reservation data from Corp2's operational database (Reservation, StayDetail, Accommodation, ServiceCharge, and FunctionBooking tables). It follows the same pattern of dimension key lookups, aggregation, and insertion into the fact table, but with mappings appropriate for Corp2's database structure.  
  
Sample Data Explanation   
The sample data was carefully designed to demonstrate how the data warehouse can answer the three management questions:

**Supporting Question 1: Revenue Optimization**  
The sample data covers a wide range of hotel brands, seasons, and room kinds, including numerous booking circumstances. Room rates and auxiliary expenditure are greater for high-season luxury bookings (reservations 1-2) compared to off-season budget bookings (reservations 3-4). This diversity makes it possible to study seasonal trends, ideal pricing tactics, and hotel brand and room type income patterns.

Supporting Question 2: Customer Satisfaction  
Various types of hotels, loyalty tiers, and booking situations (especially bookings 5-7) are represented in the sample data, along with satisfaction values ranging from 5 to 10. For example, we may see whether there is a correlation between reward tier and satisfaction ratings or if guests' experiences are impacted by event bookings. Finding trends that impact visitor pleasure is made easier by comparing the varying satisfaction levels for comparable hotel kinds.

Supporting Question 3: Cross-selling Opportunities  
It is clear from reservations 8–10 in particular that the sample data shows diverse amounts of ancillary revenue across various passenger demographics, hotel types, and booking channels. While some bookings display substantial additional expenditure (reservation 10 with $150) and others display low ratios (reservation 8 with $980 ancillary revenue). With this information, we can better focus our cross-selling activities by identifying visitor demographics and booking channels that deliver higher ancillary expenditure.  
  
All sample data has been implemented through SQL scripts and saved as separate files:

- Data for TimeDim.sql (1 KB)

- Data for GuestDim.sql (1 KB)

- Data for HotelDim.sql (1 KB)

- Data for RoomDim.sql (1 KB)

- Data for EventDim.sql (1 KB)

- Data for ReservationFact.sql (2 KB)

These scripts contain the INSERT statements that populate the data warehouse with sufficient data to demonstrate the analysis capabilities needed for the management questions. Additionally, the test query results have been captured as screenshots and saved as:

- Revenue Optimization.png (123 KB)

- Customer Satisfaction.png (91 KB)

- Cross selling opportunities.png (137 KB)

The results of the test queries show that the management questions can be adequately answered by our data warehouse. There are a number of useful analytical tools available, such as those that show seasonal performance of various hotel brands in the revenue analysis query, patterns of guest happiness across loyalty tiers and hotel types in the satisfaction analysis, and the rooms and booking channels that bring in the most extra cash in the ancillary spending analysis.

**Conclusion**  
  
 For the data warehouse of the Hokie Resort Hotels, milestone 2 has effectively applied the ETL method. While the SQL operations show the extraction, transformation, and loading techniques, the mapping between operational databases and the data warehouse guarantees a clear knowledge of data flow. Addressing the three main management issues found in Milestone 1, the sample data clearly supports study of revenue optimization, customer happiness variables, and cross-selling possibilities. Having this basis in place, the data warehouse is now ready for the reporting and visualization deployment in Milestone 3, which will give management actionable insights to enable strategic decision-making throughout the hotel portfolio.