**1. Tables Mapping**

|  |  |  |
| --- | --- | --- |
| **DatawarehouseTable** | **Operational DB Table** | **Aggregation/Sum** |
| RoomDim | Corp1: RoomID | No aggregation, each row is an instance in the DW |
| RoomDim | Corp2: Room | No aggregation, each row is an instance in the DW |
| RoomTypeDim | Corp1: RoomRate | No aggregation, each row is an instance in the DW |
| RoomTypeDim | Corp2: Room\_Type | No aggregation, each row is an instance in the DW |
| RoomReservationDim | Corp1: ResRoomID | No aggregation, each row is an instance in the DW |
| RoomReservationDim | Corp2: Room\_Reservation | No aggregation, each row is an instance in the DW |
| FloorDim | Corp1: Floor table needs to be defined and populated | No aggregation, each row is an instance in the DW |
| FloorDim | Corp2: Floor | No aggregation, each row is an instance in the DW |
| WingDim | Corp1: Wings | No aggregation, each row is an instance in the DW |
| WingDim | Corp2: Wing | No aggregation, each row is an instance in the DW |
| BuildingDim | Corp1: Building | No aggregation, each row is an instance in the DW |
| BuildingDim | Corp2: Building | No aggregation, each row is an instance in the DW |
| ResortDim | Corp1: Resort | No aggregation, each row is an instance in the DW |
| ResortDim | Corp2: Resort | No aggregation, each row is an instance in the DW |
| CityDim | Corp1: Resort | No aggregation, each row is an instance in the DW |
| CityDim | Corp2: City | No aggregation, each row is an instance in the DW |
| RegionDim | Corp1: Region table needs to be defined and populated | No aggregation, each row is an instance in the DW |
| RegionDim | Corp2: Region | No aggregation, each row is an instance in the DW |
| CountryDim | Corp1: Country | No aggregation, each row is an instance in the DW |
| CountryDim | Corp2: Country | No aggregation, each row is an instance in the DW |
| EventDim | Corp1: Funtivity | No aggregation, each row is an instance in the DW |
| EventDim | Corp2: Event | No aggregation, each row is an instance in the DW |
| CustomerDim | Corp1: Person but MiddleName attribute needs to be added | No aggregation, each row is an instance in the DW |
| CustomerDim | Corp2: Customer | No aggregation, each row is an instance in the DW |
| CustomerAffiliationDim | Corp1: Funtivity-Customer | No aggregation, each row is an instance in the DW |
| CustomerAffiliationDim | Corp2: Customer\_Affiliation | No aggregation, each row is an instance in the DW |
| RoomTypeReservationsFact | Corp1: Reservation | Total number and price of stays given are created |
| RoomTypeReservationsFact | Corp2: Room\_Reservation but ResortID attribute needs to be added | Total number and price of stays given are created |
| CustomerReservationsFact | Corp1: Reservation | Total number and price of stays given are created |
| CustomerReservationsFact | Corp2: Room\_Reservation | Total number and price of stays given are created |
| EventReservationsFact | Corp1: FuntivityRes | Total number and price of stays given are created |
| EventReservationsFact | Corp2: Customer\_Affiliation | Total number and price of stays given are created |

**2. Data Extract SQL Procedures**

**a. Customer Dimension**

/\* CustomerDim is a dimension table.\*/

CREATE TABLE CustomerDim

(

C\_CustomerID\_CU *INT* NOT NULL PRIMARY KEY,

C\_Corp1CustomerID\_CU *INT*,

C\_Corp2CustomerID\_CU *INT*,

T\_CustomerFirstName\_CU *VARCHAR*(50) NOT NULL,

T\_CustomerLastName\_CU *VARCHAR*(50) NOT NULL,

T\_CustomerMiddleName\_CU *VARCHAR*(50),

T\_CustomerAddress\_CU *VARCHAR*(255),

I\_CustomerPhoneNumber\_CU *VARCHAR*(50) NOT NULL

);

CREATE OR REPLACE PROCEDURE Get\_CustomerDim\_Dimension

/\* There are two resort operational DBs to merge into the DW.

/ \*------------------------------------------------------------\*/

/\* This procedure creates the Customer dimension table for the

/\* data warehouse by extracting data from both operational databases.

/\*------------------------------------------------------------\*/

/\* Customer Dimension process.

/\* Get customers for the CustomerDim dimension table of the data warehouse from operational database one.

/\* Declare CustomerDim Table Population variables and cursors for DB1 and DB2 \*/

declare @Corp1CustomerID *int*

declare @Corp2CustomerID *int*

declare @existingCustomerID *int*

declare @FirstName *nvarchar*(50)

declare @MiddleName *nvarchar*(50)

declare @LastName *nvarchar*(50)

declare @Address *nvarchar*(255)

declare @PhoneNumber *nvarchar*(50)

declare @getCustomerDimDB1 CURSOR

declare @getCustomerDimDB2 CURSOR

/\* Define the SQL statement to query the first operational DB \*/

set @getCustomerDimDB1 = CURSOR for

Select C\_PersonID\_PR, T\_PersonFName\_PR, T\_PersonMName\_PR, T\_PersonLName\_PR, T\_StreetAddress\_PR, I\_PhoneNumber\_PR from Person

/\* Opening the cursor returns a result set based on the query \*/

open @getCustomerDimDB1;

/\* get the first record\*/

fetch next

from @getCustomerDimDB1 into @Corp1CustomerID, @FirstName, @MiddleName, @LastName, @Address, @PhoneNumber

while @@FETCH\_STATUS = 0

begin

/\* check to see if the customer is already in the table \*/

@existingCustomerID = 0

select C\_CustomerID\_CU into @existingCustomerID from CustomerDim where C\_Corp1CustomerID\_CU = @Corp1CustomerID;

If @existingCustomerID == 0

/\* customer does not exist in dimension table so insert new customer into CustomerDim table \*/

begin try

insert into CustomerDim (C\_CustomerID\_CU, C\_Corp1CustomerID\_CU, T\_CustomerFirstName\_CU,

T\_CustomerMiddleName\_CU, T\_CustomerLastName\_CU, T\_CustomerAddress\_CU, I\_CustomerPhoneNumber\_CU)

values (CustomerDimSequence, @Corp1CustomerID, @FirstName, @MiddleName, @LastName, @Address, @PhoneNumber);

end try

begin catch

print 'Error in insert for CustomerDim DB1';

end catch

else

print 'Customer customer from DB1 exists in dimension table CustomerDim, id = ' + @existingCustomerID;

end

fetch next

from @getCustomerDimDB1 into @Corp1CustomerID, @FirstName, @MiddleName, @LastName, @Address, @PhoneNumber

end

close @getCustomerDimDB1

deallocate @getCustomerDimDB1

/\* Get customers from the second operational DB\*/

/\* Define the SQL statement to query the second operational DB \*/

set @getCustomerDimDB2 = CURSOR for

Select C\_CustomerID\_Cu, T\_CustomerFirstName\_Cu, T\_CustomerMiddleName\_Cu, T\_CustomerLastName\_Cu, T\_CustomerAddress\_Cu, I\_CustomerPhoneNumber\_Cu from Customer

/\* Opening the cursor returns a result set based on the query \*/

open @getCustomerDimDB2;

/\* get the first record\*/

fetch next

from @getCustomerDimDB2 into @Corp2CustomerID, @FirstName, @MiddleName, @LastName, @Address, @PhoneNumber

while @@FETCH\_STATUS = 0

begin

/\* check to see if the customer is already in the table \*/

@existingCustomerID = 0

select C\_CustomerID\_CU into @existingCustomerID from CustomerDim where C\_Corp2CustomerID\_CU = @Corp2CustomerID;

If @existingCustomerID == 0

/\* customer does not exist in dimension table so insert new customer into CustomerDim table \*/

begin try

insert into CustomerDim (C\_CustomerID\_CU, C\_Corp2CustomerID\_CU, T\_CustomerFirstName\_CU,

T\_CustomerMiddleName\_CU, T\_CustomerLastName\_CU, T\_CustomerAddress\_CU, I\_CustomerPhoneNumber\_CU)

values (CustomerDimSequence, @Corp2CustomerID, @FirstName, @MiddleName, @LastName, @Address, @PhoneNumber);

end try

begin catch

print 'Error in insert for CustomerDim DB2';

end catch

else

print 'Customer customer from DB2 exists in dimension table CustomerDim, id = ' + @existingCustomerID;

end

fetch next

from @getCustomerDimDB2 into @Corp2CustomerID, @FirstName, @MiddleName, @LastName, @Address, @PhoneNumber

end

close @getCustomerDimDB2

deallocate @getCustomerDimDB2

DBMS\_OUTPUT.PUT\_LINE ("CustomerDim Dimension Table Loaded at: " || *sysdate*

END;

/

**b. Resort Dimension**

/\* ResortDim is a dimension table.\*/

CREATE TABLE ResortDim

(

C\_ResortID\_RS INT PRIMARY KEY,

C\_Corp1ResortID\_CU INT,

C\_Corp2ResortID\_CU INT,

T\_ResortName\_RS VARCHAR(50) NOT NULL,

T\_StreetAddress\_RS VARCHAR(255) NOT NULL,

C\_CityName\_RS VARCHAR(50) NOT NULL,

FOREIGN KEY (C\_CityName\_RS) REFERENCES CityDim(C\_CityName\_CI)

);

CREATE OR REPLACE PROCEDURE Get\_ResortDim\_Dimension

/\* There are two resort operational DBs to merge into the DW.

/ \*------------------------------------------------------------\*/

/\* This procedure creates the Resort dimension table for the

/\* data warehouse by extracting data from both operational databases.

/\*------------------------------------------------------------\*/

/\* Resort Dimension process.

/\* Get resorts for the ResortDim dimension table of the data warehouse from operational database one.

/\* Declare ResortDim Table Population variables and cursors for DB1 and DB2 \*/

declare @Corp1ResortID int

declare @Corp2ResortID int

declare @existingResortID int

declare @ResortName nvarchar(50)

declare @StreetAddress nvarchar(255)

declare @CityName nvarchar(50)

declare @getResortDimDB1 CURSOR

declare @getResortDimDB2 CURSOR

/\* Define the SQL statement to query the first operational DB \*/

set @getResortDimDB1 = CURSOR for

Select C\_ResortID\_RS, T\_ResortName\_RS, T\_StreetAddress\_CR, T\_City\_CR from Resort

/\* Opening the cursor returns a result set based on the query \*/

open @getResortDimDB1;

/\* get the first record\*/

fetch next

from @getResortDimDB1 into @Corp1ResortID, @ResortName, @StreetAddress, @CityName

while @@FETCH\_STATUS = 0

begin

/\* check to see if the resort is already in the table \*/

@existingResortID = 0

select C\_ResortID\_CU into @existingResortID from ResortDim where C\_Corp1ResortID\_CU = @Corp1ResortID;

If @existingResortID == 0

/\* resort does not exist in dimension table so insert new resort into ResortDim table \*/

begin try

insert into ResortDim (C\_ResortID\_CU, C\_Corp1ResortID\_CU, T\_ResortName\_RS, T\_StreetAddress\_RS, C\_CityName\_RS)

values (ResortDimSequence, @Corp1ResortID, @ResortName, @StreetAddress, @CityName);

end try

begin catch

print 'Error in insert for ResortDim DB1';

end catch

else

print 'Resort resort from DB1 exists in dimension table ResortDim, id = ' + @existingResortID;

end

fetch next

from @getResortDimDB1 into @Corp1ResortID, @ResortName, @StreetAddress, @CityName

end

close @getResortDimDB1

deallocate @getResortDimDB1

/\* Get resorts from the second operational DB\*/

/\* Define the SQL statement to query the second operational DB \*/

set @getResortDimDB2 = CURSOR for

Select C\_ResortCode\_Rs, T\_ResortName\_Rs, T\_StreetAddress\_Rs, C\_CityName\_Rs from Resort

/\* Opening the cursor returns a result set based on the query \*/

open @getResortDimDB2;

/\* get the first record\*/

fetch next

from @getResortDimDB2 into @Corp2ResortID, @ResortName, @StreetAddress, @CityName

while @@FETCH\_STATUS = 0

begin

/\* check to see if the resort is already in the table \*/

@existingResortID = 0

select C\_ResortID\_CU into @existingResortID from ResortDim where C\_Corp2ResortID\_CU = @Corp2ResortID;

If @existingResortID == 0

/\* resort does not exist in dimension table so insert new resort into ResortDim table \*/

begin try

insert into ResortDim (C\_ResortID\_CU, C\_Corp2ResortID\_CU, T\_ResortName\_RS, T\_StreetAddress\_RS, C\_CityName\_RS)

values (ResortDimSequence, @Corp2ResortID, @ResortName, @StreetAddress, @CityName);

end try

begin catch

print 'Error in insert for ResortDim DB2';

end catch

else

print 'Resort resort from DB2 exists in dimension table ResortDim, id = ' + @existingResortID;

end

fetch next

from @getResortDimDB2 into @Corp2ResortID, @ResortName, @StreetAddress, @CityName

end

close @getResortDimDB2

deallocate @getResortDimDB2

DBMS\_OUTPUT.PUT\_LINE ("ResortDim Dimension Table Loaded at: " || sysdate

END;

/

**c. Room Type Reservations Fact**

CREATE TABLE RoomTypeReservationsFact

(

C\_TimeID\_TR INT NOT NULL,

T\_RoomType\_TR VARCHAR(50) NOT NULL,

C\_ResortID\_TR INT NOT NULL,

N\_TotalNumOfStays\_TR DECIMAL,

N\_TotalPrice\_TR DECIMAL,

FOREIGN KEY (C\_TimeID\_TR) REFERENCES TimeDim(C\_TimeID\_TI),

FOREIGN KEY (T\_RoomType\_TR) REFERENCES RoomTypeDim(C\_RoomType\_RT),

FOREIGN KEY (C\_ResortID\_TR) REFERENCES ResortDim(C\_ResortID\_RS),

PRIMARY KEY(C\_TimeID\_TR, T\_RoomType\_TR, C\_ResortID\_TR)

);

CREATE TABLE TimeDim

(

C\_TimeID\_TI INT PRIMARY KEY,

N\_TimeYear\_TI INT NOT NULL,

N\_TimeMonth\_TI INT NOT NULL,

T\_TimeDay\_TI VARCHAR(50) NOT NULL,

);

/\* Populate RoomTypeReservationsFact Fact table from the first operational DB \*/

CREATE OR REPLACE PROCEDURE Get\_RoomTypeReservationsFact\_Fact

declare @TimeId int

declare @Month int

declare @Day int

declare @Year int

declare @OriginReservationId

declare @ResortId

declare @RoomType

declare @ReservationDate

declare @TotalPrice

declare @TotalStays

/\* Define the SQL statement to query the first operational DB \*/

set @getRoomTypeReservationsFactDB1 = CURSOR for

Select C\_ReservationID\_RN, C\_ResortID\_RN, C\_RoomRateID\_RN, T\_ReservationDate\_RN FROM Reservation

ORDER BY C\_ReservationID\_RN group by C\_ReservationID\_RN, C\_ResortID\_RN, C\_RoomRateID\_RN, T\_ReservationDate\_RN;

/\* Opening the cursor returns a result set based on the query \*/

open @getRoomTypeReservationsFactDB1;

fetch next

from @getRoomTypeReservationsFactDB1 into @OriginReservationId, @ResortId, @RoomType, @ReservationDate

while @@FETCH\_STATUS = 0

begin

/\* Extract date fields from timestamp \*/

@Month = Month(@ReservationDate)

@Day = Day(@ReservationDate)

@Year = Year(@ReservationDate)

/\* Find the time dimension row for this fact row (should have \*/

Select TimeID into @TimeID from TimeInfo where time.day = @day and time.month = @month and time.year = @year;

/\* Find the RoomReservationDim dimension joined with RoomTypeDim for this fact row to calculate TotalStays and TotalPrice\*/

Select DATEDIFF(day, D\_ArrivalDate\_RR, D\_DepartureDate\_RR) into @TotalStays,

N\_RoomPrice\_RT \* DATEDIFF(day, D\_ArrivalDate\_RR, D\_DepartureDate\_RR) into @TotalPrice

from RoomReservationDim rr JOIN RoomTypeDim rt on rr.C\_RoomType\_RR = rt.C\_RoomType\_RT

where rr.C\_RoomReservationNum\_RR = @OriginReservationId;

/\* Insert the new fact row into the RoomTypeReservationsFact table \*/

begin try

insert into RoomTypeReservationsFact (C\_TimeID\_TR, T\_RoomType\_TR, C\_ResortID\_TR, N\_TotalNumOfStays\_TR, N\_TotalPrice\_TR)

values (@TimeId, @RoomType, @ResortId, @TotalStays, @TotalPrice);

end try

begin catch

print 'Error in insert for RoomTypeReservationsFactDB1';

end catch

fetch next

from @getRoomTypeReservationsFactDB1 into @OriginReservationId, @ResortId, @RoomType, @ReservationDate

end

close @getRoomTypeReservationsFactDB1

deallocate @getRoomTypeReservationsFactDB1

/\* Define the SQL statement to query the second operational DB \*/

/\* Assumption: ResortID column/attribute has been added to Room\_Reservation table in Corp2 \*/

set @getRoomTypeReservationsFactDB2 = CURSOR for

Select C\_RoomReservationNumber\_Rr, C\_ResortID\_Rr, C\_RoomType\_Rr, D\_ArrivalDate\_Rr FROM Room\_Reservation

ORDER BY C\_RoomReservationNumber\_Rr group by C\_RoomReservationNumber\_Rr, C\_ResortID\_Rr, C\_RoomType\_Rr, D\_ArrivalDate\_Rr;

/\* Opening the cursor returns a result set based on the query \*/

open @getRoomTypeReservationsFactDB2;

fetch next

from @getRoomTypeReservationsFactDB2 into @OriginReservationId, @ResortId, @RoomType, @ReservationDate

while @@FETCH\_STATUS = 0

begin

/\* Extract date fields from timestamp \*/

@Month = Month(@ReservationDate)

@Day = Day(@ReservationDate)

@Year = Year(@ReservationDate)

/\* Find the time dimension row for this fact row (should have \*/

Select TimeID into @TimeID from TimeInfo where time.day = @day and time.month = @month and time.year = @year;

/\* Find the RoomReservationDim dimension joined with RoomTypeDim for this fact row to calculate TotalStays and TotalPrice\*/

Select DATEDIFF(day, D\_ArrivalDate\_RR, D\_DepartureDate\_RR) into @TotalStays,

N\_RoomPrice\_RT \* DATEDIFF(day, D\_ArrivalDate\_RR, D\_DepartureDate\_RR) into @TotalPrice,

from RoomReservationDim rr JOIN RoomTypeDim rt on rr.C\_RoomType\_RR = rt.C\_RoomType\_RT

where rr.C\_RoomReservationNum\_RR = @OriginReservationId;

/\* Insert the new fact row into the RoomTypeReservationsFact table \*/

begin try

insert into RoomTypeReservationsFact (C\_TimeID\_TR, T\_RoomType\_TR, C\_ResortID\_TR, N\_TotalNumOfStays\_TR, N\_TotalPrice\_TR)

values (@TimeId, @RoomType, @ResortId, @TotalStays, @TotalPrice);

end try

begin catch

print 'Error in insert for RoomTypeReservationsFactDB2';

end catch

fetch next

from @getRoomTypeReservationsFactDB2 into @OriginReservationId, @ResortId, @RoomType, @ReservationDate

end

close @getRoomTypeReservationsFactDB2

deallocate @getRoomTypeReservationsFactDB2

DBMS\_OUTPUT.PUT\_LINE ("RoomTypeReservations Fact Table Loaded at: " || sysdate

END;

/

**3. Data Load SQL Statements**

/\* DW-DBINIT.SQL \*/

/\* DATA WAREHOUSE SQL \*/

/\* Script file for MySQL DBMS \*/

/\* This script file creates the following tables: \*/

/\* TimeDim, RoomDim, RoomTypeDim, RoomReservationDim, FloorDim, \*/

/\* WingDim, BuildingDim, ResortDim, CityDim, RegionDim, CountryDim \*/

/\* EventDim, CustomerDim, CustomerAffiliationDim\*/

/\* RoomTypeReservationsFact, CustomerReservationsFact, EventReservationsFact \*/

/\* and loads the default data rows \*/

DROP TABLE RoomTypeReservationsFact;

DROP TABLE CustomerReservationsFact;

DROP TABLE EventReservationsFact;

DROP TABLE TimeDim;

DROP TABLE RoomDim;

DROP TABLE RoomReservationDim;

DROP TABLE RoomTypeDim;

DROP TABLE FloorDim;

DROP TABLE WingDim;

DROP TABLE BuildingDim;

DROP TABLE ResortDim;

DROP TABLE CityDim;

DROP TABLE RegionDim;

DROP TABLE CountryDim;

DROP TABLE CustomerAffiliationDim;

DROP TABLE EventDim;

DROP TABLE CustomerDim;

/\* TimeDim \*/

CREATE TABLE TimeDim

(

C\_TimeID\_TI *INT* PRIMARY KEY,

N\_TimeYear\_TI *INT* NOT NULL,

N\_TimeMonth\_TI *INT* NOT NULL,

T\_TimeDay\_TI *VARCHAR*(50) NOT NULL

);

/\* RoomTypeDim \*/

CREATE TABLE RoomTypeDim

(

C\_RoomType\_RT *VARCHAR*(50) PRIMARY KEY,

N\_RoomPrice\_RT *DECIMAL* NOT NULL

);

/\* CountryDim \*/

CREATE TABLE CountryDim

(

C\_CountryCode\_CO *VARCHAR*(5) PRIMARY KEY,

T\_CountryName\_CO *VARCHAR*(50) NOT NULL

);

/\* RegionDim \*/

CREATE TABLE RegionDim

(

C\_RegionID\_RG *INT* PRIMARY KEY,

T\_RegionName\_RG *VARCHAR*(50) NOT NULL,

C\_CountryCode\_RG *VARCHAR*(5) NOT NULL,

FOREIGN KEY (C\_CountryCode\_RG) REFERENCES CountryDim(C\_CountryCode\_CO)

);

/\* CityDim \*/

CREATE TABLE CityDim

(

C\_CityName\_CI *VARCHAR*(50),

C\_ZipCode\_CI *VARCHAR*(50),

T\_State\_CI *VARCHAR*(50) NOT NULL,

C\_RegionID\_CI *INT* NOT NULL,

FOREIGN KEY (C\_RegionID\_CI) REFERENCES RegionDim(C\_RegionID\_RG),

PRIMARY KEY(C\_CityName\_CI, C\_ZipCode\_CI)

);

/\* ResortDim \*/

CREATE TABLE ResortDim

(

C\_ResortID\_RS *INT* PRIMARY KEY,

T\_ResortName\_RS *VARCHAR*(50) NOT NULL,

T\_StreetAddress\_RS *VARCHAR*(255) NOT NULL,

C\_CityName\_RS *VARCHAR*(50) NOT NULL,

FOREIGN KEY (C\_CityName\_RS) REFERENCES CityDim(C\_CityName\_CI)

);

/\* BuildingDim \*/

CREATE TABLE BuildingDim

(

C\_BuildingID\_BU *INT* PRIMARY KEY,

T\_BuildingName\_BU *VARCHAR*(50) NOT NULL,

T\_BuildingDescription\_BU *VARCHAR*(255) NOT NULL,

C\_ResortID\_BU *INT* NOT NULL,

FOREIGN KEY (C\_ResortID\_BU) REFERENCES ResortDim(C\_ResortID\_RS)

);

/\* WingDim \*/

CREATE TABLE WingDim

(

C\_WingCode\_WG *VARCHAR*(10) NOT NULL,

C\_BuildingID\_WG *INT* NOT NULL,

T\_WingName\_WG *VARCHAR*(50) NOT NULL,

PRIMARY KEY(C\_WingCode\_WG, C\_BuildingID\_WG)

);

/\* FloorDim \*/

CREATE TABLE FloorDim

(

C\_FloorNumber\_FL *INT* NOT NULL,

C\_BuildingID\_FL *INT* NOT NULL,

C\_WingCode\_FL *VARCHAR*(10) NOT NULL,

PRIMARY KEY(C\_FloorNumber\_FL, C\_BuildingID\_FL, C\_WingCode\_FL)

);

/\* RoomDim \*/

CREATE TABLE RoomDim

(

C\_RoomNum\_RM *VARCHAR*(10) NOT NULL,

C\_FloorNumber\_RM *INT* NOT NULL,

C\_WingCode\_RM *VARCHAR*(10) NOT NULL,

C\_RoomType\_RM *VARCHAR*(50) NOT NULL,

PRIMARY KEY(C\_RoomNum\_RM, C\_FloorNumber\_RM, C\_WingCode\_RM),

FOREIGN KEY (C\_RoomType\_RM) REFERENCES RoomTypeDim(C\_RoomType\_RT)

);

/\* EventDim \*/

CREATE TABLE EventDim

(

C\_EventID\_EV *INT* NOT NULL PRIMARY KEY,

T\_EventDescription\_EV *VARCHAR*(255) NOT NULL,

D\_EventStartDate\_EV *DATETIME* NOT NULL,

D\_EventEndDate\_EV *DATETIME* NOT NULL

);

/\* CustomerDim \*/

CREATE TABLE CustomerDim

(

C\_CustomerID\_CU *INT* NOT NULL PRIMARY KEY,

T\_CustomerFirstName\_CU *VARCHAR*(50) NOT NULL,

T\_CustomerLastName\_CU *VARCHAR*(50) NOT NULL,

T\_CustomerMiddleName\_CU *VARCHAR*(50),

T\_CustomerAddress\_CU *VARCHAR*(255),

I\_CustomerPhoneNumber\_CU *VARCHAR*(50) NOT NULL

);

/\* RoomReservationDim \*/

CREATE TABLE RoomReservationDim

(

C\_RoomReservationNum\_RR *VARCHAR*(50),

C\_CustomerID\_RR *INT*,

C\_RoomType\_RR *VARCHAR*(50),

D\_ArrivalDate\_RR *DATETIME* NOT NULL,

D\_DepartureDate\_RR *DATETIME* NOT NULL,

FOREIGN KEY (C\_CustomerID\_RR) REFERENCES CustomerDim(C\_CustomerID\_CU),

FOREIGN KEY (C\_RoomType\_RR) REFERENCES RoomTypeDim(C\_RoomType\_RT),

PRIMARY KEY(C\_RoomReservationNum\_RR, C\_CustomerID\_RR, C\_RoomType\_RR)

);

/\* CustomerAffiliationDim \*/

CREATE TABLE CustomerAffiliationDim

(

C\_CustomerID\_CA *INT* NOT NULL,

C\_EventID\_CA *INT* NOT NULL,

FOREIGN KEY (C\_CustomerID\_CA) REFERENCES CustomerDim(C\_CustomerID\_CU),

FOREIGN KEY (C\_EventID\_CA) REFERENCES EventDim(C\_EventID\_EV),

PRIMARY KEY(C\_CustomerID\_CA, C\_EventID\_CA)

);

/\* RoomTypeReservationsFact \*/

CREATE TABLE RoomTypeReservationsFact

(

C\_TimeID\_TR *INT* NOT NULL,

T\_RoomType\_TR *VARCHAR*(50) NOT NULL,

C\_ResortID\_TR *INT* NOT NULL,

N\_TotalNumOfStays\_TR *DECIMAL*,

N\_TotalPrice\_TR *DECIMAL*,

FOREIGN KEY (C\_TimeID\_TR) REFERENCES TimeDim(C\_TimeID\_TI),

FOREIGN KEY (T\_RoomType\_TR) REFERENCES RoomTypeDim(C\_RoomType\_RT),

FOREIGN KEY (C\_ResortID\_TR) REFERENCES ResortDim(C\_ResortID\_RS),

PRIMARY KEY(C\_TimeID\_TR, T\_RoomType\_TR, C\_ResortID\_TR)

);

/\* CustomerReservationsFact \*/

CREATE TABLE CustomerReservationsFact

(

C\_TimeID\_CR *INT* NOT NULL,

C\_CustomerID\_CR *INT* NOT NULL,

C\_ResortID\_CR *INT* NOT NULL,

N\_TotalNumOfStays\_CR *DECIMAL*,

N\_TotalPrice\_CR *DECIMAL*,

FOREIGN KEY (C\_TimeID\_CR) REFERENCES TimeDim(C\_TimeID\_TI),

FOREIGN KEY (C\_CustomerID\_CR) REFERENCES CustomerDim(C\_CustomerID\_CU),

FOREIGN KEY (C\_ResortID\_CR) REFERENCES ResortDim(C\_ResortID\_RS),

PRIMARY KEY(C\_TimeID\_CR, C\_CustomerID\_CR, C\_ResortID\_CR)

);

/\* EventReservationsFact \*/

CREATE TABLE EventReservationsFact

(

C\_TimeID\_ER *INT* NOT NULL,

C\_EventID\_ER *INT* NOT NULL,

C\_ResortID\_ER *INT* NOT NULL,

N\_TotalNumOfStays\_ER *DECIMAL*,

N\_TotalPrice\_ER *DECIMAL*,

FOREIGN KEY (C\_TimeID\_ER) REFERENCES TimeDim(C\_TimeID\_TI),

FOREIGN KEY (C\_EventID\_ER) REFERENCES EventDim(C\_EventID\_EV),

FOREIGN KEY (C\_ResortID\_ER) REFERENCES ResortDim(C\_ResortID\_RS),

PRIMARY KEY(C\_TimeID\_ER, C\_EventID\_ER, C\_ResortID\_ER)

);

/\* TimeDim rows \*/

INSERT INTO TimeDim VALUES(1, 2010, 09, 29);

INSERT INTO TimeDim VALUES(2, 2010, 09, 30);

INSERT INTO TimeDim VALUES(3, 2010, 09, 31);

INSERT INTO TimeDim VALUES(4, 2010, 10, 03);

INSERT INTO TimeDim VALUES(5, 2010, 10, 04);

INSERT INTO TimeDim VALUES(6, 2010, 10, 05);

INSERT INTO TimeDim VALUES(7, 2010, 10, 06);

INSERT INTO TimeDim VALUES(8, 2010, 10, 07);

INSERT INTO TimeDim VALUES(9, 2010, 10, 08);

INSERT INTO TimeDim VALUES(10, 2010, 10, 09);

INSERT INTO TimeDim VALUES(11, 2010, 10, 10);

/\* RoomTypeDim rows \*/

INSERT INTO RoomTypeDim VALUES('SINGLE', 80.0);

INSERT INTO RoomTypeDim VALUES('DOUBLE', 90.0);

INSERT INTO RoomTypeDim VALUES('TRIPLE', 100.0);

INSERT INTO RoomTypeDim VALUES('MULTI', 105.0);

/\* CountryDim rows \*/

INSERT INTO CountryDim VALUES('US','United States');

INSERT INTO CountryDim VALUES('EG','Egypt');

INSERT INTO CountryDim VALUES('JP','Japan');

/\* RegionDim rows \*/

INSERT INTO RegionDim VALUES(1,'North-West', 'US');

INSERT INTO RegionDim VALUES(2,'Middle-East', 'EG');

INSERT INTO RegionDim VALUES(3,'EAST', 'JP');

/\* CityDim rows \*/

INSERT INTO CityDim VALUES('Blacksburg','24060','Virginia', 1);

INSERT INTO CityDim VALUES('Arlington','76010','Texas', 1);

INSERT INTO CityDim VALUES('Houston','77092','Texas', 1);

/\* ResortDim rows \*/

INSERT INTO ResortDim VALUES(1,'Hokie East', '131 Patrick Dr.','Blacksburg');

INSERT INTO ResortDim VALUES(2,'Hokie Central', '79 Watonga Blvd.','Houston');

INSERT INTO ResortDim VALUES(3,'Hokie DFW', '209 Elm St.','Arlington');

/\* BuildingDim rows \*/

INSERT INTO BuildingDim VALUES(1,'Main','Central Building', 1);

INSERT INTO BuildingDim VALUES(2,'Hokies','Hokies Building', 1);

INSERT INTO BuildingDim VALUES(3,'Innovation','Innovation Building', 1);

INSERT INTO BuildingDim VALUES(4,'Main','Central Building', 2);

INSERT INTO BuildingDim VALUES(5,'Hokies','Hokies Building', 2);

INSERT INTO BuildingDim VALUES(6,'Innovation','Innovation Building', 2);

INSERT INTO BuildingDim VALUES(7,'Main','Central Building', 3);

INSERT INTO BuildingDim VALUES(8,'Hokies','Hokies Building', 3);

INSERT INTO BuildingDim VALUES(9,'Innovation','Innovation Building', 3);

/\* WingDim rows \*/

INSERT INTO WingDim VALUES('E', 1, 'East');

INSERT INTO WingDim VALUES('W', 1, 'West');

INSERT INTO WingDim VALUES('N', 1, 'North');

INSERT INTO WingDim VALUES('S', 1, 'South');

INSERT INTO WingDim VALUES('E', 2, 'East');

INSERT INTO WingDim VALUES('W', 2, 'West');

INSERT INTO WingDim VALUES('N', 2, 'North');

INSERT INTO WingDim VALUES('S', 2, 'South');

INSERT INTO WingDim VALUES('E', 3, 'East');

INSERT INTO WingDim VALUES('W', 3, 'West');

INSERT INTO WingDim VALUES('N', 3, 'North');

INSERT INTO WingDim VALUES('S', 3, 'South');

INSERT INTO WingDim VALUES('E', 4, 'East');

INSERT INTO WingDim VALUES('W', 4, 'West');

INSERT INTO WingDim VALUES('N', 4, 'North');

INSERT INTO WingDim VALUES('S', 4, 'South');

/\* FloorDim rows \*/

INSERT INTO FloorDim VALUES(1, 1, 'W');

INSERT INTO FloorDim VALUES(2, 1, 'W');

INSERT INTO FloorDim VALUES(3, 1, 'W');

INSERT INTO FloorDim VALUES(4, 1, 'W');

INSERT INTO FloorDim VALUES(1, 1, 'E');

INSERT INTO FloorDim VALUES(2, 1, 'E');

INSERT INTO FloorDim VALUES(3, 1, 'E');

INSERT INTO FloorDim VALUES(4, 1, 'E');

INSERT INTO FloorDim VALUES(1, 1, 'N');

INSERT INTO FloorDim VALUES(2, 1, 'N');

INSERT INTO FloorDim VALUES(3, 1, 'N');

INSERT INTO FloorDim VALUES(4, 1, 'N');

INSERT INTO FloorDim VALUES(1, 1, 'S');

INSERT INTO FloorDim VALUES(2, 1, 'S');

INSERT INTO FloorDim VALUES(3, 1, 'S');

INSERT INTO FloorDim VALUES(4, 1, 'S');

/\* RoomDim rows \*/

INSERT INTO RoomDim VALUES('W1-100', 1, 'W', 'SINGLE');

INSERT INTO RoomDim VALUES('N2-203', 2, 'N', 'MULTI');

INSERT INTO RoomDim VALUES('S4-404', 4, 'S', 'SINGLE');

INSERT INTO RoomDim VALUES('E3-310', 3, 'E', 'DOUBLE');

INSERT INTO RoomDim VALUES('W3-305', 3, 'W', 'TRIPLE');

INSERT INTO RoomDim VALUES('E2-201', 2, 'E', 'MULTI');

INSERT INTO RoomDim VALUES('S1-107', 1, 'S', 'DOUBLE');

INSERT INTO RoomDim VALUES('N4-409', 4, 'N', 'SINGLE');

/\* EventDim rows \*/

INSERT INTO EventDim VALUES(1,'ColdPlay Concert', '2010-04-25 07:00:00','2010-04-30 10:15:00');

INSERT INTO EventDim VALUES(2,'Hokies Reunion', '2011-07-15 12:00:00','2011-07-15 18:00:00');

/\* CustomerDim rows \*/

INSERT INTO CustomerDim VALUES(1, 'John', 'Smith', '', '513 Main St.', '205-814-2233');

INSERT INTO CustomerDim VALUES(2, 'Sarah', 'Kent', 'T.', '20 South St.', '205-600-1024');

INSERT INTO CustomerDim VALUES(3, 'Mariam', 'Hafez', '', '19 Roanoke St.', '817-245-4000');

INSERT INTO CustomerDim VALUES(4, 'Jordan', 'Saba', '', '37 N St.', '205-112-1134');

INSERT INTO CustomerDim VALUES(5, 'Zack', 'Nolan', '', '1700 Pratt St.', '205-406-7780');

/\* RoomReservationDim rows \*/

INSERT INTO RoomReservationDim VALUES(1, 1, 'MULTI', '2010-04-25 15:00:00', '2010-04-26 11:01:00');

INSERT INTO RoomReservationDim VALUES(2, 2, 'SINGLE', '2010-04-25 15:30:00', '2010-04-26 10:30:00');

INSERT INTO RoomReservationDim VALUES(3, 3, 'DOUBLE', '2010-04-25 18:00:00', '2010-04-26 11:05:00');

INSERT INTO RoomReservationDim VALUES(4, 4, 'SINGLE', '2010-04-25 16:09:00', '2010-04-26 09:03:00');

/\* CustomerAffiliationDim rows \*/

INSERT INTO CustomerAffiliationDim VALUES(1, 1);

INSERT INTO CustomerAffiliationDim VALUES(2, 1);

INSERT INTO CustomerAffiliationDim VALUES(3, 1);

INSERT INTO CustomerAffiliationDim VALUES(4, 1);

INSERT INTO CustomerAffiliationDim VALUES(5, 1);

INSERT INTO CustomerAffiliationDim VALUES(3, 2);

INSERT INTO CustomerAffiliationDim VALUES(4, 2);

/\* RoomTypeReservationsFact rows \*/

INSERT INTO RoomTypeReservationsFact VALUES(1, 'SINGLE', 1, 2, 160.0);

INSERT INTO RoomTypeReservationsFact VALUES(2, 'DOUBLE', 1, 1, 90.0);

INSERT INTO RoomTypeReservationsFact VALUES(3, 'MULTI', 1, 1, 105.0);

INSERT INTO RoomTypeReservationsFact VALUES(4, 'SINGLE', 1, 1, 80.0);

/\* CustomerReservationsFact rows \*/

INSERT INTO CustomerReservationsFact VALUES(4, 1, 1, 1, 105.0);

INSERT INTO CustomerReservationsFact VALUES(5, 2, 1, 1, 80.0);

INSERT INTO CustomerReservationsFact VALUES(6, 3, 1, 2, 180.0);

INSERT INTO CustomerReservationsFact VALUES(7, 4, 1, 1, 80.0);

/\* EventReservationsFact rows \*/

INSERT INTO EventReservationsFact VALUES(8, 1, 1, 1, 90.0);

INSERT INTO EventReservationsFact VALUES(9, 2, 1, 1, 80.0);

INSERT INTO EventReservationsFact VALUES(10, 2, 1, 1, 105.0);

INSERT INTO EventReservationsFact VALUES(11, 1, 1, 2, 160.0);

COMMIT;