UNIVERSITY OF WESTMINSTER

DEPARTMENT OF COMPUTER SCIENCE

MSc Big Data Technologies

MODULE CODE:7BUIS010W

MODULE TITLE :DATA WAREHOUSING AND BUSINESS INTELLIGENCE

MODULE LEADER: DR. PANAGIOTIS CHOUNTAS

COURSEWORK (2020/21)

SUBMISSION DATE: 25 March 2021

STUDENT ID: w1813148

STUDENT NAME: PATEL BHAUMIKKUMAR SHAILESH

1. **Merge the database schemas depicted in Figure-1 and Figure-2 into a single schema (integrated schema) so that can store data from both the original databases. State any assumptions you may have considered while developing the integrated schema.**

- Integrated relational schema from Hotel room booking OLTP system and e-Ticket Data Source
  **The _**primary keys**_ _are underlined and the_ _**foreign keys**_ _are followed by the sharp sign (#) and the name of the referenced table._

Room (RoomID, RoomTypesID#: RoomTypes, RoomBandID#: RoomBand, RoomFacilityID#: RoomFacilities, Price, Floor, AdditionalNotes**)**

RoomTypes (RoomTypeID, TypeDesc**)**

RoomBands (RoomBandID, BrandDesc**)**

RoomFacilities (RoomFacilityID, FacilityDesc**)**

Payments (PaymentID, CustomerID#: Customer, PaymentMethodID#: PaymentMethods, PaymentAmount, PaymentComments**)**

PaymentMethods (PaymentMethodID, PaymentMethod**)**

Bookings (CustomerID#: Customer, DateBookingMade, TimebookingMade, RoomID#: Room, BookedStartDate, BookedEndDate, TotalPaymentDueDate, TotalPaymentDueAmount, BookingComments**)**

Customer (CustomerID,CustomerForenames,CustomerSurnames, CustomerDOB,CustomerHomePhone,CustomerWorkPhone,CustomerMobilePhone, CustomerEmail, CityID#:City**)**

County(CountyID, CountyName**)**

State(StateID, StateName, CountyID#: County**)**

City(CityID, CityName, StateID#:State**)**

Singer(SingerID, SingerForenames, SingerSurnames**)**

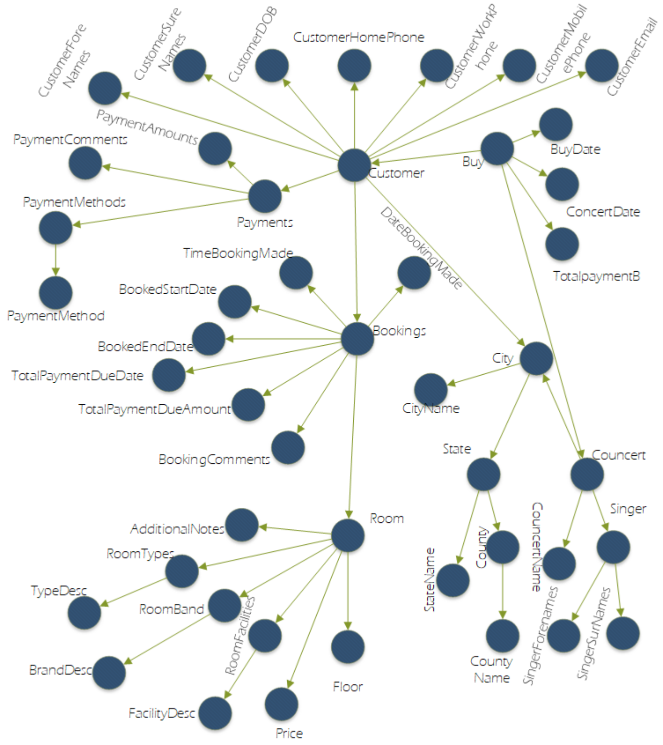
Concert(ConcertID, ConcertName, CityID#:City, SingerID: Singer**)**

Buy(CustomerID#:Customer,ConcertID#:Concert,BuyDate, ConcertDate, TotalPaymentB**)**

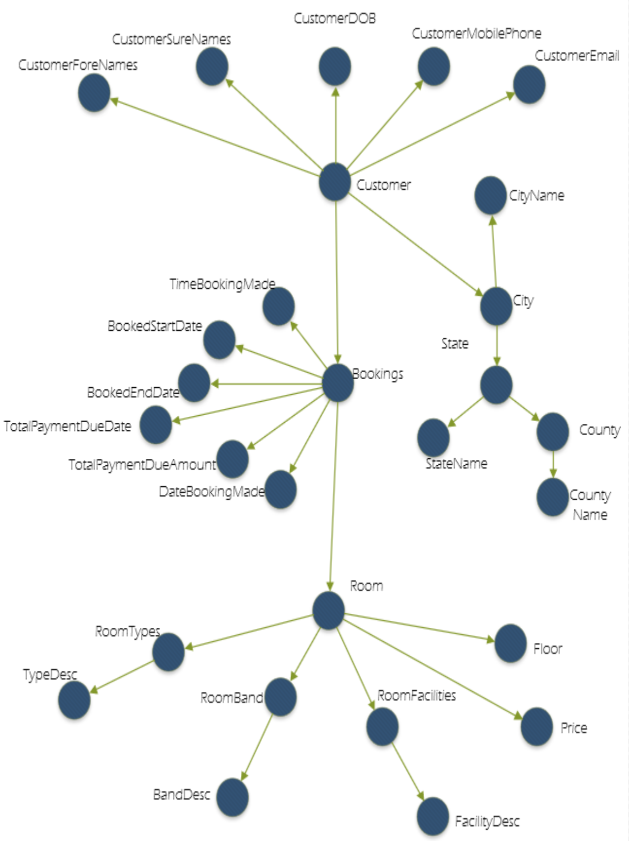
2. **Based on the integrated relational schema, design a data warehouse model (DFM); in particular, the designated data mart must promptly answer to all the frequent queries 1-3.**

I. Build the Attribute Tree from the integrated relational schema

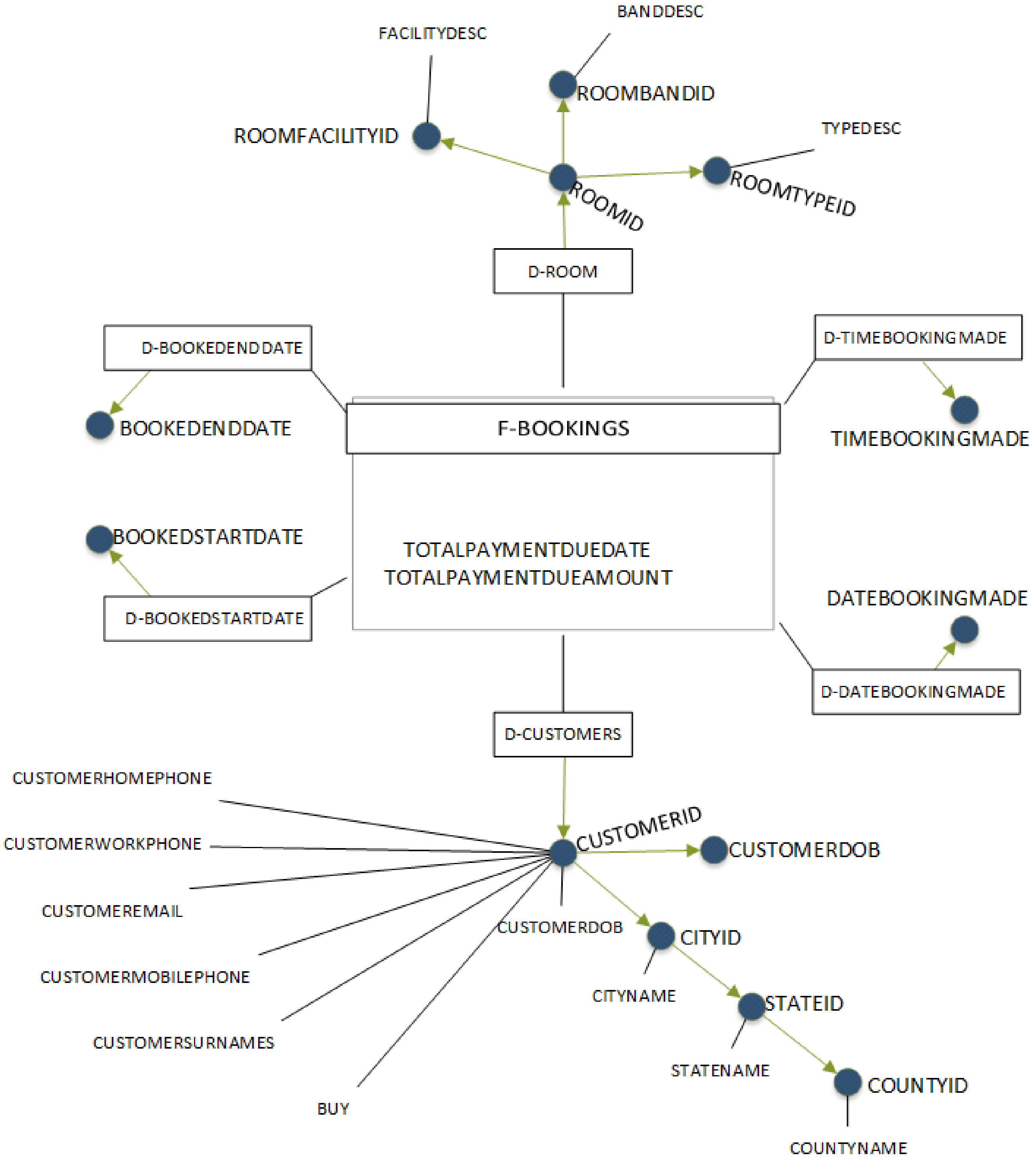
(Construction of the Attribute Tree)



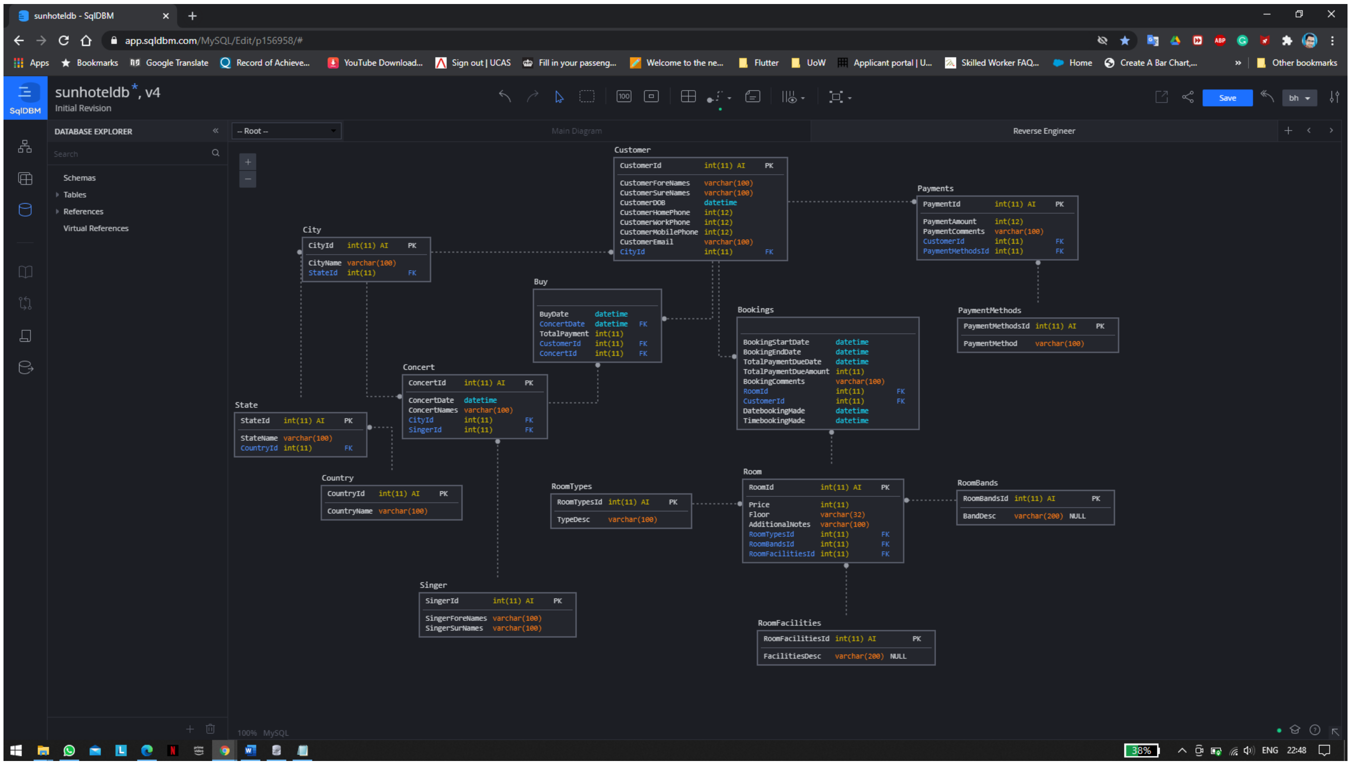
(Pruning of the Attribute Tree)



II. Build the Fact Schema from Attribute Tree



3. Map the DFM model to a logical model (i.e., relational). Clearly display the main fact table(s) and dimensions.



```
Customer
CustomerId            int(11) AI     PK
CustomerForeNames     varchar(100)
CustomerSureNames     varchar(100)
CustomerDOB           datetime
CustomerHomePhone     int(12)
CustomerWorkPhone     int(12)
CustomerMobilePhone   int(12)
CustomerEmail         varchar(100)
CityId                int(11)        FK

Payments
PaymentId             int(11) AI     PK
PaymentAmount         int(12)
PaymentComments       varchar(100)
CustomerId            int(11)        FK
PaymentMethodsId      int(11)        FK

PaymentMethods
PaymentMethodsId      int(11) AI     PK
PaymentMethod         varchar(100)

City
CityId                int(11) AI     PK
CityName              varchar(100)
StateId               int(11)        FK

Buy
BuyDate               datetime
ConcertDate           datetime       FK
TotalPayment          int(11)
CustomerId            int(11)        FK
ConcertId             int(11)        FK

Bookings
BookingStartDate      datetime
BookingEndDate        datetime
TotalPaymentDueDate   datetime
TotalPaymentDueAmount int(11)
BookingComments       varchar(100)
RoomId                int(11)        FK
CustomerId            int(11)        FK
DatebookingMade       datetime
TimebookingMade       datetime

Concert
ConcertId             int(11) AI     PK
ConcertDate           datetime
ConcertNames          varchar(100)
CityId                int(11)        FK
SingerId              int(11)        FK

State
StateId               int(11) AI     PK
StateName             varchar(100)
CountryId             int(11)        FK

Country
CountryId             int(11) AI     PK
CountryName           varchar(100)

RoomTypes
RoomTypesId           int(11) AI     PK
TypeDesc              varchar(100)

Room
RoomId                int(11) AI     PK
Price                 int(11)
Floor                 varchar(32)
AdditionalNotes       varchar(100)
RoomTypesId           int(11)        FK
RoomBandsId           int(11)        FK
RoomFacilitiesId      int(11)        FK

RoomBands
RoomBandsId           int(11) AI     PK
BandDesc              varchar(200)   NULL

Singer
SingerId              int(11) AI     PK
SingerForeNames       varchar(100)
SingerSurNames        varchar(100)

RoomFacilities
RoomFacilitiesId      int(11) AI     PK
FacilitiesDesc        varchar(200)   NULL
```

Table.1 Facts and measures for the integrated schema issued from hotel room booking OLTP system and e-Ticket Data Source

Fact	Relevance level	Measure
F-Room	Second	Price
F-Payments	Second	PaymentAmount
F-Bookings	First	TotalPaymentDueAmount TotalPaymentDueDate
F-Buy	First	TotalPaymentB

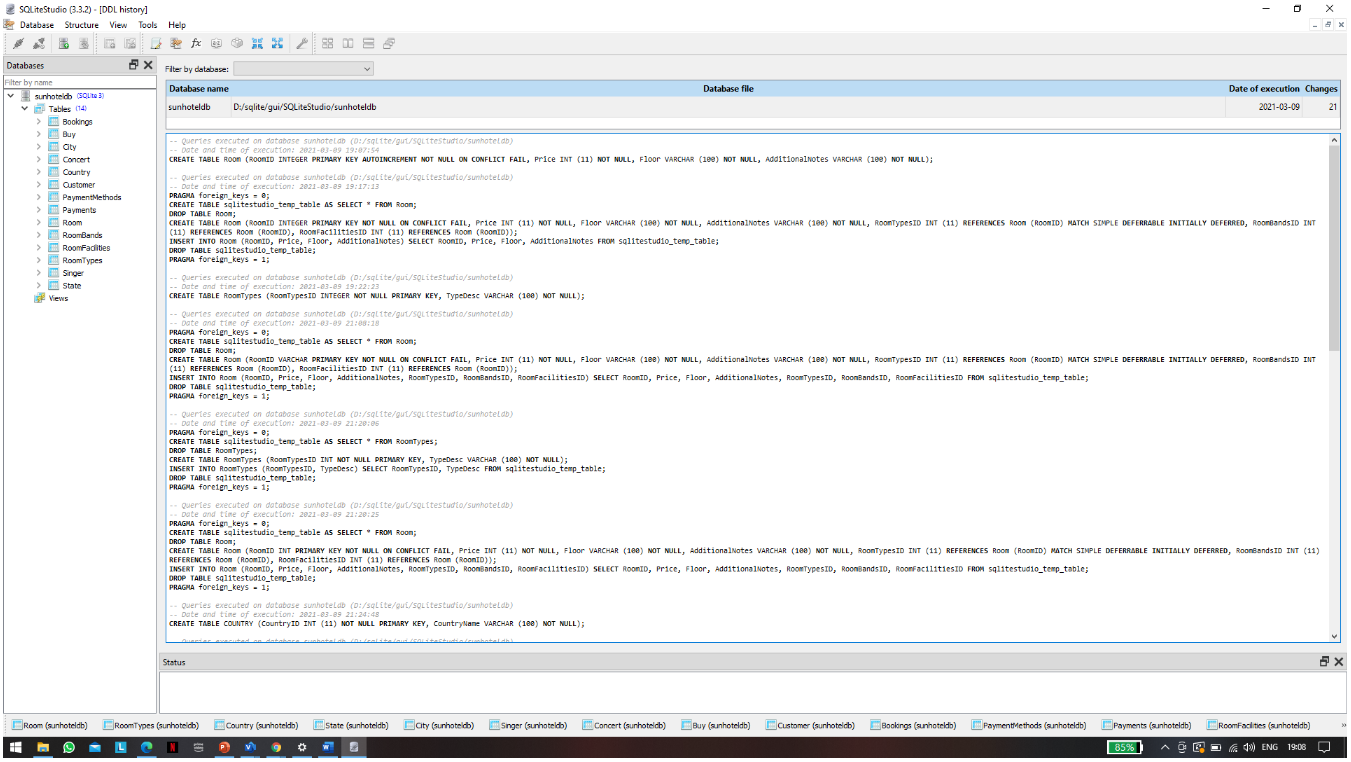
Table.2 Dimensions for the extracted facts

Fact	**Dimension**	**Identifier**
F-Room	D-RoomTypes	RoomTypeID
	D-RoomBands	RoomBandID
	D-Facilities	RoomFacilityID
F-Payments	D-Customer	CustomerID
	D-PaymentMethods	PaymentMethodID
F-Bookings	D-Customer	CustomerID
	D-DateBookingMade	DateBookingMade
	D-TimebookingMade	TimebookingMade
	D-Room	RoomID
	D-BookedStartDate	BookedStartDate
	D-BookedEndDate	BookedEndDate
F-Buy	D-Customer	CustomerID
	D-Concert	ConcertID
	D-BuyDate	BuyDate
	D-ConcertDate	ConcertDate

Table.3 Parameters for the dimensions of table 2

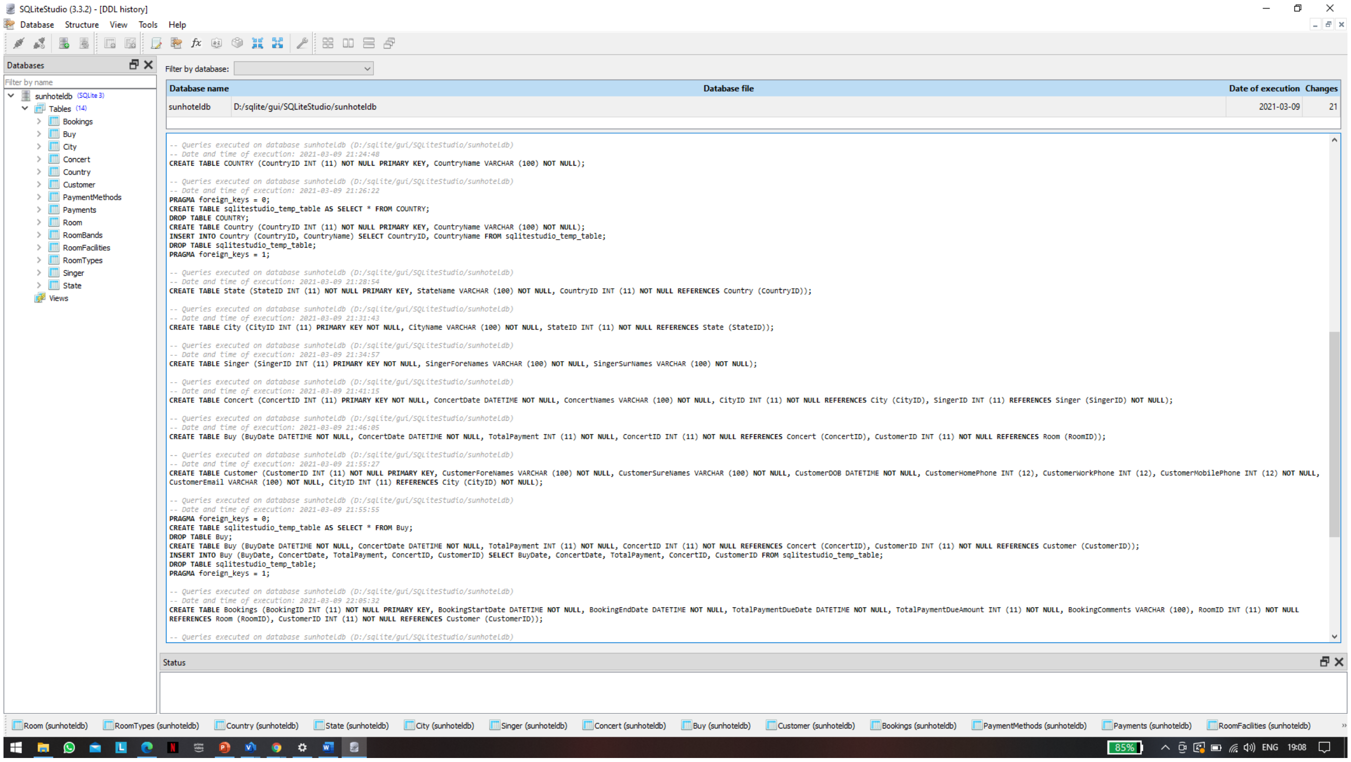
Dimension	**Hierarchy parameters (From finest to coarsest)**		
D-Customer	CityID	StateID	CountyID
	CustomerDOB		
D-Room	RoomTypeID		
	RoomBandID		
	RoomFacilityID		
D-Concert	SingerID		
	CityID	StateID	CountyID

4. **Implement the above logical as a working data warehouse schema, under MySQL/R, or any ither suitable DBMS. Provide the DDL statements to create the proposed data-warehouse schema.**



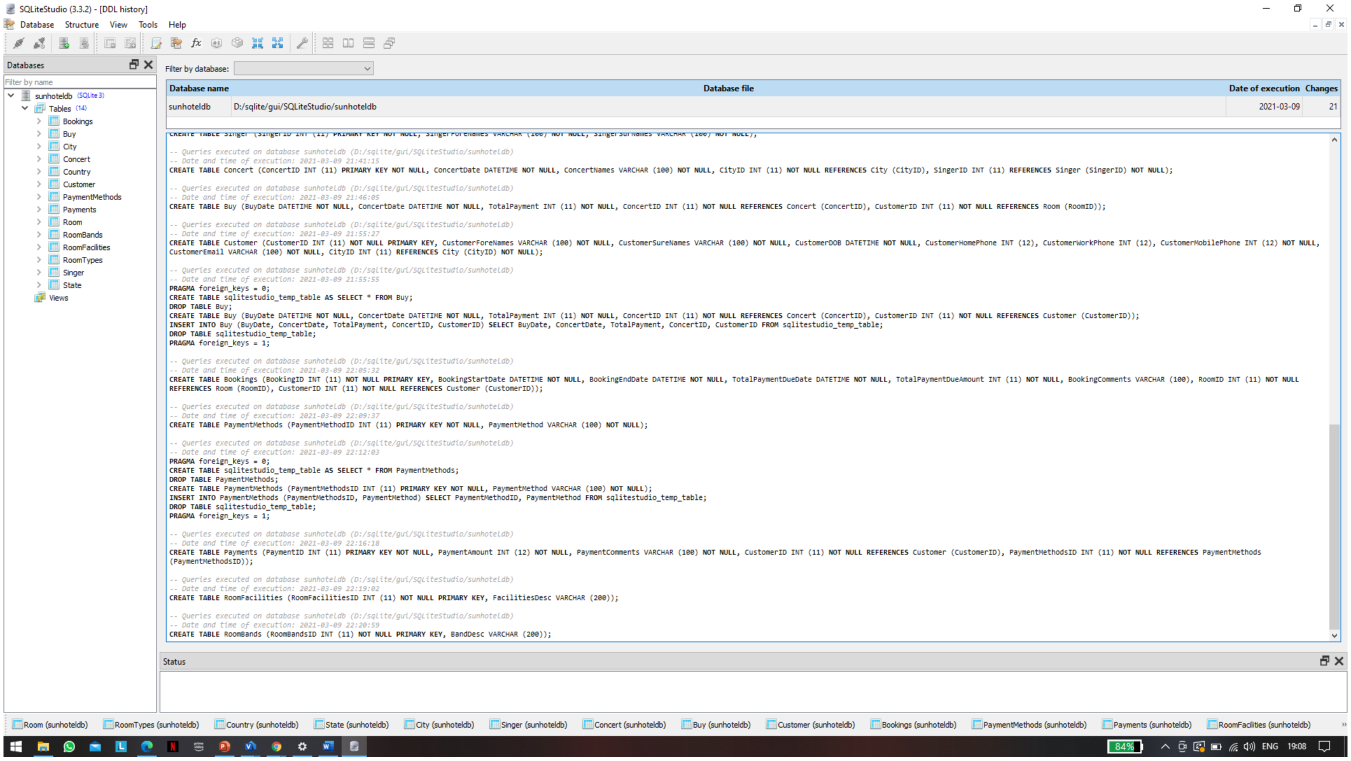
Database name	Database file	Date of execution	Changes
sunhoteldb	D:/sqlite/gui/SQLiteStudio/sunhoteldb	2021-03-09	21

```
-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 19:07:54
CREATE TABLE Room (RoomID INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL ON CONFLICT FAIL, Price INT (11) NOT NULL, Floor VARCHAR (100) NOT NULL, AdditionalNotes VARCHAR (100) NOT NULL);

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 19:17:13
PRAGMA foreign_keys = 0;
CREATE TABLE sqlitestudio_temp_table AS SELECT * FROM Room;
DROP TABLE Room;
CREATE TABLE Room (RoomID INTEGER PRIMARY KEY NOT NULL ON CONFLICT FAIL, Price INT (11) NOT NULL, Floor VARCHAR (100) NOT NULL, AdditionalNotes VARCHAR (100) NOT NULL, RoomTypesID INT (11) REFERENCES Room (RoomID) MATCH SIMPLE DEFERRABLE INITIALLY DEFERRED, RoomBandsID INT
(11) REFERENCES Room (RoomID), RoomFacilitiesID INT (11) REFERENCES Room (RoomID));
INSERT INTO Room (RoomID, Price, Floor, AdditionalNotes) SELECT RoomID, Price, Floor, AdditionalNotes FROM sqlitestudio_temp_table;
DROP TABLE sqlitestudio_temp_table;
PRAGMA foreign_keys = 1;

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 19:22:23
CREATE TABLE RoomTypes (RoomTypesID INTEGER NOT NULL PRIMARY KEY, TypeDesc VARCHAR (100) NOT NULL);

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:08:18
PRAGMA foreign_keys = 0;
CREATE TABLE sqlitestudio_temp_table AS SELECT * FROM Room;
DROP TABLE Room;
CREATE TABLE Room (RoomID VARCHAR PRIMARY KEY NOT NULL ON CONFLICT FAIL, Price INT (11) NOT NULL, Floor VARCHAR (100) NOT NULL, AdditionalNotes VARCHAR (100) NOT NULL, RoomTypesID INT (11) REFERENCES Room (RoomID) MATCH SIMPLE DEFERRABLE INITIALLY DEFERRED, RoomBandsID INT
(11) REFERENCES Room (RoomID), RoomFacilitiesID INT (11) REFERENCES Room (RoomID));
INSERT INTO Room (RoomID, Price, Floor, AdditionalNotes, RoomTypesID, RoomBandsID, RoomFacilitiesID) SELECT RoomID, Price, Floor, AdditionalNotes, RoomTypesID, RoomBandsID, RoomFacilitiesID FROM sqlitestudio_temp_table;
DROP TABLE sqlitestudio_temp_table;
PRAGMA foreign_keys = 1;

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:20:06
PRAGMA foreign_keys = 0;
CREATE TABLE sqlitestudio_temp_table AS SELECT * FROM RoomTypes;
DROP TABLE RoomTypes;
CREATE TABLE RoomTypes (RoomTypesID INT NOT NULL PRIMARY KEY, TypeDesc VARCHAR (100) NOT NULL);
INSERT INTO RoomTypes (RoomTypesID, TypeDesc) SELECT RoomTypesID, TypeDesc FROM sqlitestudio_temp_table;
DROP TABLE sqlitestudio_temp_table;
PRAGMA foreign_keys = 1;

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:20:25
PRAGMA foreign_keys = 0;
CREATE TABLE sqlitestudio_temp_table AS SELECT * FROM Room;
DROP TABLE Room;
CREATE TABLE Room (RoomID INT PRIMARY KEY NOT NULL ON CONFLICT FAIL, Price INT (11) NOT NULL, Floor VARCHAR (100) NOT NULL, AdditionalNotes VARCHAR (100) NOT NULL, RoomTypesID INT (11) REFERENCES Room (RoomID) MATCH SIMPLE DEFERRABLE INITIALLY DEFERRED, RoomBandsID INT (11)
REFERENCES Room (RoomID), RoomFacilitiesID INT (11) REFERENCES Room (RoomID));
INSERT INTO Room (RoomID, Price, Floor, AdditionalNotes, RoomTypesID, RoomBandsID, RoomFacilitiesID) SELECT RoomID, Price, Floor, AdditionalNotes, RoomTypesID, RoomBandsID, RoomFacilitiesID FROM sqlitestudio_temp_table;
DROP TABLE sqlitestudio_temp_table;
PRAGMA foreign_keys = 1;

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:24:48
CREATE TABLE COUNTRY (CountryID INT (11) NOT NULL PRIMARY KEY, CountryName VARCHAR (100) NOT NULL);
```



Database name	Database file	Date of execution	Changes
sunhoteldb	D:/sqlite/gui/SQLiteStudio/sunhoteldb	2021-03-09	21

```
-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:24:48
CREATE TABLE COUNTRY (CountryID INT (11) NOT NULL PRIMARY KEY, CountryName VARCHAR (100) NOT NULL);

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:26:22
PRAGMA foreign_keys = 0;
CREATE TABLE sqlitestudio_temp_table AS SELECT * FROM COUNTRY;
DROP TABLE COUNTRY;
CREATE TABLE Country (CountryID INT (11) NOT NULL PRIMARY KEY, CountryName VARCHAR (100) NOT NULL);
INSERT INTO Country (CountryID, CountryName) SELECT CountryID, CountryName FROM sqlitestudio_temp_table;
DROP TABLE sqlitestudio_temp_table;
PRAGMA foreign_keys = 1;

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:28:54
CREATE TABLE State (StateID INT (11) NOT NULL PRIMARY KEY, StateName VARCHAR (100) NOT NULL, CountryID INT (11) NOT NULL REFERENCES Country (CountryID));

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:31:43
CREATE TABLE City (CityID INT (11) PRIMARY KEY NOT NULL, CityName VARCHAR (100) NOT NULL, StateID INT (11) NOT NULL REFERENCES State (StateID));

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:34:57
CREATE TABLE Singer (SingerID INT (11) PRIMARY KEY NOT NULL, SingerForeNames VARCHAR (100) NOT NULL, SingerSurNames VARCHAR (100) NOT NULL);

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:41:15
CREATE TABLE Concert (ConcertID INT (11) PRIMARY KEY NOT NULL, ConcertDate DATETIME NOT NULL, ConcertNames VARCHAR (100) NOT NULL, CityID INT (11) NOT NULL REFERENCES City (CityID), SingerID INT (11) REFERENCES Singer (SingerID) NOT NULL);

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:46:05
CREATE TABLE Buy (BuyDate DATETIME NOT NULL, ConcertDate DATETIME NOT NULL, TotalPayment INT (11) NOT NULL, ConcertID INT (11) NOT NULL REFERENCES Concert (ConcertID), CustomerID INT (11) NOT NULL REFERENCES Room (RoomID));

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:55:27
CREATE TABLE Customer (CustomerID INT (11) NOT NULL PRIMARY KEY, CustomerForeNames VARCHAR (100) NOT NULL, CustomerSureNames VARCHAR (100) NOT NULL, CustomerDOB DATETIME NOT NULL, CustomerHomePhone INT (12), CustomerWorkPhone INT (12), CustomerMobilePhone INT (12) NOT NULL,
CustomerEmail VARCHAR (100) NOT NULL, CityID INT (11) REFERENCES City (CityID) NOT NULL);

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:55:55
PRAGMA foreign_keys = 0;
CREATE TABLE sqlitestudio_temp_table AS SELECT * FROM Buy;
DROP TABLE Buy;
CREATE TABLE Buy (BuyDate DATETIME NOT NULL, ConcertDate DATETIME NOT NULL, TotalPayment INT (11) NOT NULL, ConcertID INT (11) NOT NULL REFERENCES Concert (ConcertID), CustomerID INT (11) NOT NULL REFERENCES Customer (CustomerID));
INSERT INTO Buy (BuyDate, ConcertDate, TotalPayment, ConcertID, CustomerID) SELECT BuyDate, ConcertDate, TotalPayment, ConcertID, CustomerID FROM sqlitestudio_temp_table;
DROP TABLE sqlitestudio_temp_table;
PRAGMA foreign_keys = 1;

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 22:05:32
CREATE TABLE Bookings (BookingID INT (11) NOT NULL PRIMARY KEY, BookingStartDate DATETIME NOT NULL, BookingEndDate DATETIME NOT NULL, TotalPaymentDueDate DATETIME NOT NULL, TotalPaymentDueAmount INT (11) NOT NULL, BookingComments VARCHAR (100), RoomID INT (11) NOT NULL
REFERENCES Room (RoomID), CustomerID INT (11) NOT NULL REFERENCES Customer (CustomerID));

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
```



Database name	Database file	Date of execution	Changes
sunhoteldb	D:/sqlite/gui/SQLiteStudio/sunhoteldb	2021-03-09	21

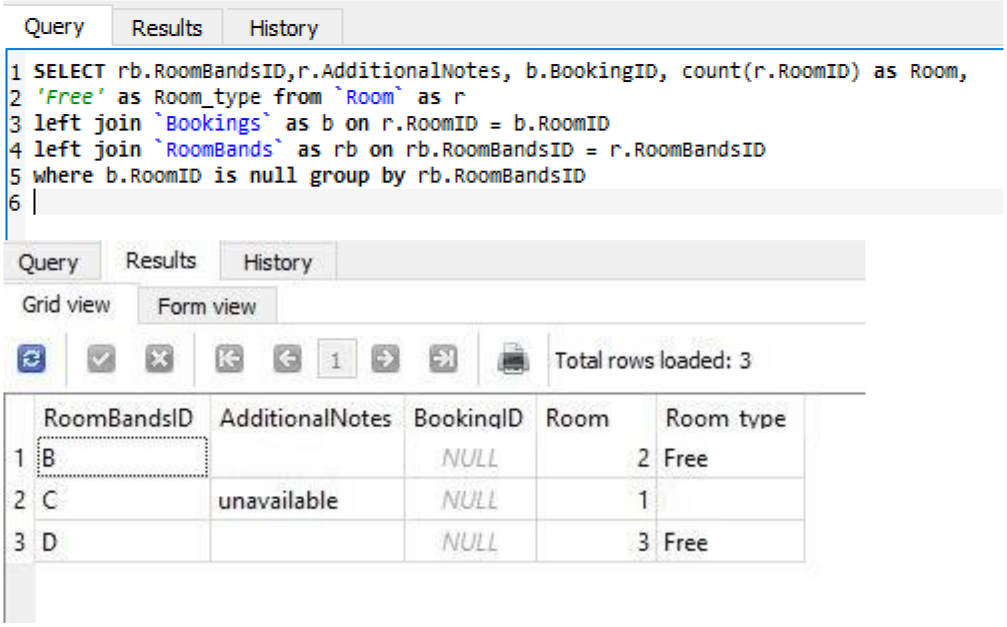
```
CREATE TABLE Singer (SingerID INT (11) PRIMARY KEY NOT NULL, SingerForeNames VARCHAR (100) NOT NULL, SingerSurNames VARCHAR (100) NOT NULL);

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:41:15
CREATE TABLE Concert (ConcertID INT (11) PRIMARY KEY NOT NULL, ConcertDate DATETIME NOT NULL, ConcertNames VARCHAR (100) NOT NULL, CityID INT (11) NOT NULL REFERENCES City (CityID), SingerID INT (11) REFERENCES Singer (SingerID) NOT NULL);

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:46:05
CREATE TABLE Buy (BuyDate DATETIME NOT NULL, ConcertDate DATETIME NOT NULL, TotalPayment INT (11) NOT NULL, ConcertID INT (11) NOT NULL REFERENCES Concert (ConcertID), CustomerID INT (11) NOT NULL REFERENCES Room (RoomID));

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:55:27
CREATE TABLE Customer (CustomerID INT (11) NOT NULL PRIMARY KEY, CustomerForeNames VARCHAR (100) NOT NULL, CustomerSureNames VARCHAR (100) NOT NULL, CustomerDOB DATETIME NOT NULL, CustomerHomePhone INT (12), CustomerWorkPhone INT (12), CustomerMobilePhone INT (12) NOT NULL,
CustomerEmail VARCHAR (100) NOT NULL, CityID INT (11) REFERENCES City (CityID) NOT NULL);

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 21:55:55
PRAGMA foreign_keys = 0;
CREATE TABLE sqlitestudio_temp_table AS SELECT * FROM Buy;
DROP TABLE Buy;
CREATE TABLE Buy (BuyDate DATETIME NOT NULL, ConcertDate DATETIME NOT NULL, TotalPayment INT (11) NOT NULL, ConcertID INT (11) NOT NULL REFERENCES Concert (ConcertID), CustomerID INT (11) NOT NULL REFERENCES Customer (CustomerID));
INSERT INTO Buy (BuyDate, ConcertDate, TotalPayment, ConcertID, CustomerID) SELECT BuyDate, ConcertDate, TotalPayment, ConcertID, CustomerID FROM sqlitestudio_temp_table;
DROP TABLE sqlitestudio_temp_table;
PRAGMA foreign_keys = 1;

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 22:05:32
CREATE TABLE Bookings (BookingID INT (11) NOT NULL PRIMARY KEY, BookingStartDate DATETIME NOT NULL, BookingEndDate DATETIME NOT NULL, TotalPaymentDueDate DATETIME NOT NULL, TotalPaymentDueAmount INT (11) NOT NULL, BookingComments VARCHAR (100), RoomID INT (11) NOT NULL
REFERENCES Room (RoomID), CustomerID INT (11) NOT NULL REFERENCES Customer (CustomerID));

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 22:09:37
CREATE TABLE PaymentMethods (PaymentMethodID INT (11) PRIMARY KEY NOT NULL, PaymentMethod VARCHAR (100) NOT NULL);

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 22:12:03
PRAGMA foreign_keys = 0;
CREATE TABLE sqlitestudio_temp_table AS SELECT * FROM PaymentMethods;
DROP TABLE PaymentMethods;
CREATE TABLE PaymentMethods (PaymentMethodsID INT (11) PRIMARY KEY NOT NULL, PaymentMethod VARCHAR (100) NOT NULL);
INSERT INTO PaymentMethods (PaymentMethodsID, PaymentMethod) SELECT PaymentMethodID, PaymentMethod FROM sqlitestudio_temp_table;
DROP TABLE sqlitestudio_temp_table;
PRAGMA foreign_keys = 1;

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 22:16:18
CREATE TABLE Payments (PaymentID INT (11) PRIMARY KEY NOT NULL, PaymentAmount INT (12) NOT NULL, PaymentComments VARCHAR (100) NOT NULL, CustomerID INT (11) NOT NULL REFERENCES Customer (CustomerID), PaymentMethodsID INT (11) NOT NULL REFERENCES PaymentMethods
(PaymentMethodsID));

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 22:19:02
CREATE TABLE RoomFacilities (RoomFacilitiesID INT (11) NOT NULL PRIMARY KEY, FacilitiesDesc VARCHAR (200));

-- Queries executed on database sunhoteldb (D:/sqlite/gui/SQLiteStudio/sunhoteldb)
-- Date and time of execution: 2021-03-09 22:20:59
CREATE TABLE RoomBands (RoomBandsID INT (11) NOT NULL PRIMARY KEY, BandDesc VARCHAR (200));
```

5. **Considering the designed data warehouse and its cardinalities, decide whether and which materialized views are convenient to improve response time of the frequent queries (consider all the frequent queries). Explain the reasons for your choices.**

   CREATE MATERIALIZED VIEW ROOMID
   BUILD IMMEDIATE
   REFRESH FORCE
   ON COMMIT
   AS
   SELECT rb.RoomBandsID,r.AdditionalNotes, b.BookingID, count(r.RoomID) as Room,
   'Free' as Room_type from `Room` as r
   left join `Bookings` as b on r.RoomID = b.RoomID
   left join `RoomBands` as rb on rb.RoomBandsID = r.RoomBandsID
   where b.RoomID is null group by rb.RoomBandsID ;

6. **Provide and implement a materialized view(s) to answer the director's frequent queries'1-3'**

   I. **For each room band and month, derive the portion of rooms which are reserved, free and unavailable.**



```
SELECT rb.RoomBandsID,r.AdditionalNotes, b.BookingID, count(r.RoomID) as Room,
'Free' as Room_type from `Room` as r
left join `Bookings` as b on r.RoomID = b.RoomID
left join `RoomBands` as rb on rb.RoomBandsID = r.RoomBandsID
where b.RoomID is null group by rb.RoomBandsID

```

Total rows loaded: 3

	RoomBandsID	AdditionalNotes	BookingID	Room	Room type
1	B		NULL	2	Free
2	C	unavailable	NULL	1	
3	D		NULL	3	Free

   II. **For each room band, derive the portion of rooms which are reserved. Associate a rank to each country according to the portion of checkout rooms for that country in a particular year with respect to all reserved rooms for that country. The country with the highest ratio of checkout rooms in a particular year must rank first.**

III. For each room band and concert, produce the cumulative income of 4-star rooms.