Database Project Documentation

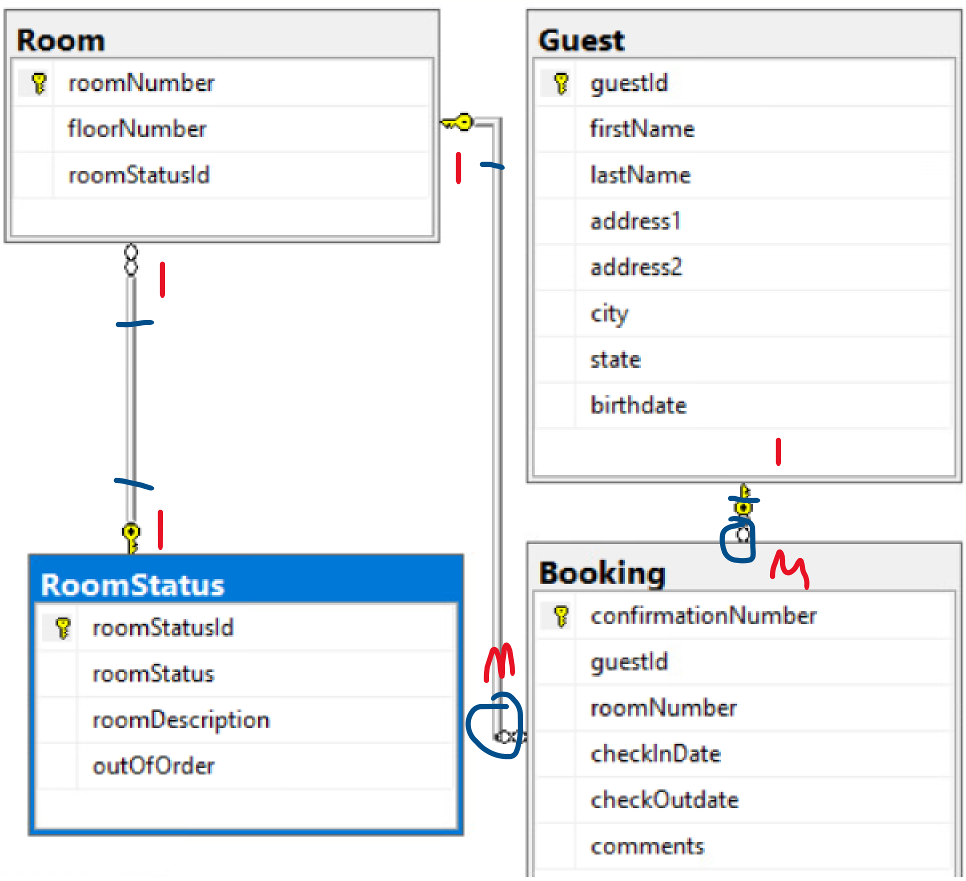
For the SMT_HotelDatabase

By Spencer Thompson

Project Description

The purpose of this database is to examine the relationships between the many parts of hotel operation, and how they can be compartmentalized for effective information gathering. For the sake of simplicity, the database model was simplified to reflect basic states of guest profiles, booking itinerary, and rooms/room status. The guest profiles tables covered basic information, names, addresses, and birthdates. The room table had a room number, floor number, and a status ID. The room status table held status ID, status, description, and whether it was out of order. The booking table held the confirmation number, guest ID, room number, check in date, check out date, and guest requests/comments. The tables covered the basic idea of hotel operations and maintenance. Admittedly, there is room for improvement even in the simplified version, such as the redundance of a room status ID, rather than joining on the room number itself. However, for the sake of illustration this was still effective.

Database Diagram



Data Dictionary

Table Name	Field Name	Data Type	Required (Y/N)	Primary Key (Y/N)	Foreign Key (Y/N)	References
Guest	GuestId	Integer	Y	Yes		
Guest	FirstName	VarChar	Y			
Guest	LastName	VarChar	Y			
Guest	Address1	VarChar	Y			
Guest	Address2	VarChar	Y			
Guest	City	VarChar	Y			
Guest	State	VarChar	Y			
Guest	BirthDate	Date	N			
Booking	ConfirmationNumber	Integer	Y	Yes		
Booking	GuestId	Integer	Y		Yes	Guest
Booking	RoomNumber	Integer	Y		Yes	Room
Booking	CheckInDate	Date	Y			
Booking	CheckOutDate	Date	Y			
Booking	Comments	VarChar	N			
Room	RoomNumber	Integer	Y	Yes		
Room	FloorNumber	Integer	Y			
Room	RoomStatusId	VarChar	Y		Yes	RoomStatus
RoomStatus	RoomStatusId	VarChar	Y	Yes		
RoomStatus	RoomStatus	VarChar	Y			
RoomStatus	RoomDescription	VarChar	Y			
RoomStatus	OutOfOrder	Bit	Y			

DDL Script

```
if OBJECT_ID('Smt_hoteldatabase') is not null

drop database SMT_HotelDatabase;

 GO

Create DATABASE SMT_HotelDatabase;
```

```
GO
Use SMT_HotelDatabase;
GO
Create Table Guest(
   guestId INTEGER not null identity,
   firstName Varchar(25) not null,
   lastName Varchar(25) not null,
   address1 Varchar(255) not null,
   address2 VARCHAR(255) null,
   city varchar(50) not null,
   state VARCHAR(2) not null,
   birthdate date null,
   Constraint guestPK PRIMARY KEY(guestId)
);
GO
Create Table RoomStatus(
   roomStatusId varchar(10) not null,
   roomStatus varchar(50) not null,
   roomDescription varchar(255) not null,
   outOfOrder bit not null,
   CONSTRAINT roomStatusPK PRIMARY KEY(roomStatusId)
);
GO
Create Table Room(
   roomNumber INTEGER not null,
   floorNumber INTEGER not null,
   roomStatusId varchar(10) not null
   Constraint roomPK PRIMARY KEY(roomNumber),
   Constraint roomStatusFK FOREIGN KEY(roomStatusId) references RoomStatus(roomStatusId)
);
GO
Create Table Booking(
   confirmationNumber INTEGER not null identity,
   guestId Integer not null,
   roomNumber Integer not null,
   checkInDate date not null,
   checkOutdate date not null,
   comments varchar(255) null,
   Constraint bookingPK PRIMARY KEY(confirmationNumber),
   Constraint guestFK FOREIGN KEY(guestId) REFERENCES guest(guestId),
   Constraint roomFK FOREIGN KEY(roomNumber) REFERENCES
   room(roomNumber)
);
GO
insert into guest(firstName, lastName, address1, address2, city, state, birthdate)
values('Spencer', 'Thompson', '1111 South Street', 'Apt 2', 'Lincoln', 'NE', '1998-09-22'),
('Grae','VanBonn','2222 North Street',null,'Detroit','MI', null),
('November','Rass','3333 East Court','Suite 5','Omaha','NE','2000-10-19'),
('Preston','Woods','4444 West Street',null,'Salt Lake City','UT','1996-11-14'),
('Barbara','Bread','6666 Nowhere Road',null,'','',''),
('Taelon','Lee','5555 Something Lane','Apt 5','Iowa City','IA','2001-08-22'),
('John','Snow','1234 Lannister Way',null,'Lansing','MI','1986-02-08'),
('Grace','Shoemaker','5432 Link Lane','Estate 7','Glendale','AZ','1969-12-04'),
('Amity','Coral','645 Stamp Road',Null,'St. Paul','MN', null),
('Lucy','Pevensie','2438 Elm Street',null,'Anaheim','CA','1977-09-09'),
('James','Simpson','238 Giants Court','Apt 6','Austin','TX',null),
('Cornelious','Eisenhorn','13 State Road',null,'Anchorage','AK','2003-06-14'),
('Mark','Meyers','4 Bland Place','Suite 9','Oakland','CA',null),
('Amanda','Stark','8437 Crane Street',null,'Henderson','NV','1944-12-20'),
('Jon','Bellion','007 Bond Way',null,'Greensboro','NC','1955-08-24');
GO
insert into booking(guestId, roomNumber, checkInDate, checkOutdate, comments)
values(1, 100, '2022-11-01','2022-11-04','Extra Towels'),
(2, 101, '2022-11-01','2022-11-04',null),
(3, 102, '2022-11-01','2022-11-04','1 Extra pillow'),
(4, 103, '2022-11-01','2022-11-04','Extra Towels'),
(5, 104, '2022-11-01','2022-11-04','Upgrade if available'),
(6, 200, '2022-11-01','2022-11-04','Travelling with pet'),
(7, 201, '2022-11-01','2022-11-04',null),
(8, 202, '2022-11-01','2022-11-04','Extra Towels'),
(9, 203, '2022-11-01','2022-11-04',null),
(10, 204, '2022-11-01','2022-11-04','Rollaway bed'),
(11, 300, '2022-11-01','2022-11-04','Extra room keys'),
(12, 301, '2022-11-01','2022-11-04','Late check in'),
(13, 302, '2022-11-01','2022-11-04',null),
(14, 303, '2022-11-01','2022-11-04',null),
(15, 304, '2022-11-01','2022-11-04',null);
GO
insert into room(roomNumber, floorNumber, roomStatusId)
values(100, 1, '1'),
(101, 1, '2'),
(102, 1, '3'),
(103, 1, '4'),
(104, 1, '5'),
(200, 2, '6'),
(201, 2, '7'),
(202, 2, '8'),
(203, 2, '9'),
(204, 2, '10'),
(300, 3, '11'),
(301, 3, '12'),
(302, 3, '13'),
(303, 3, '14'),
(304, 3, '15');
GO
insert into roomStatus(roomStatusId, roomStatus, roomDescription, OutOfOrder)
values('1', 'Vacant Ready', 'Single King', 0),
('2', 'Vacant Ready', 'Single King', 0),
('3', 'Vacant Ready', 'Double Queen', 0),
('4', 'Vacant Ready', 'Double Queen', 0),
('5', 'Vacant Ready', 'King Suite', 0),
('6', 'Vacant Ready', 'Single King', 0),
('7', 'Vacant Ready', 'Single King', 0),
('8', 'Vacant Ready', 'Double Queen', 0),
('9', 'Vacant Ready', 'Double Queen', 0),
('10', 'Vacant Ready', 'King Suite', 0),
('11', 'Vacant Ready', 'Single King', 0),
('12', 'Vacant Ready', 'Single King', 0),
('13', 'Vacant Ready', 'Double Queen', 0),
('14', 'Vacant Ready', 'Double Queen', 0),
('15', 'Vacant Ready', 'King Suite', 0);
```

Select Queries (Output presented below each query)

Aggregate Function -- How mony rooms are there of each type?

```
Select count(roomstatusid) as "Number of Rooms", roomDescription as "Type Of Room"

from roomstatus

group by roomDescription;
```

Number of Rooms	Type Of Room
6	Double Queen
3	King Suite
6	Single King

Inner Join + And -- What Single King Rooms Are Ready For Use?

```
Select roomnumber from room

inner join roomstatus on room.roomstatusid = roomstatus.roomstatusid

where roomDescription like 'Single King' and roomstatus like 'Vacant Ready';
```

RoomNumber
100
101
200
201
300
301

Between + OR -- Guests Eligable for Winter Birthday Promotional

```
select concat(firstname, ' ', lastname) as 'Guest Name', birthdate as 'Winter Birthdays' from guest

where month(birthdate) between 11 and 12 or MONTH(birthdate) between 1 and 2;
```

Guest Name	Birthdays in this Quarter
Preston Woods	1996-11-14

Barbara Bread	1900-01-01
John Snow	1986-02-08
Grace Shoemaker	1969-12-04
Amanda Stark	1944-12-20

Outer Join + Not -- Guest Comments for Rooms Not on the First Floor

```
Select concat(firstname, ' ', lastname) as 'Guest Name',roomnumber, comments from guest

left join booking on guest.guestid = booking.guestid

where roomnumber not like '1%';
```

Guest Name	RoomNumber	Comments
Taelon Lee	200	Travelling with pet
John Snow	201	NULL
Grace Shoemaker	202	Extra Towels
Amity Coral	203	NULL
Lucy Pevensie	204	Rollaway bed
James Simpson	300	Extra room keys
Cornelious Eisenhorn	301	Late check in
Mark Meyers	302	NULL
Amanda Stark	303	NULL
Jon Bellion	304	NULL

TOP + Subquery -- Guest with latest checkout

```
select guestid, concat(firstname, ' ', lastname) as 'Guest Name' from guest

where guestid in (select top 1 b.guestid from booking b order by checkOutdate asc);
```

GuestId	Guest Name
2	Grae VanBonn

Group by + Having-- Room types with more than 3 of that type

```
Select count(roomstatusid) as "Number of Rooms", roomDescription as "Type Of Room"
```

from roomstatus group by roomDescription having count(*) > 3;

Number of Rooms	Type Of Room
6	Double Queen
6	Single King

Calculated -- How many nights is a guest staying?

select DATEDIFF(day, checkindate, checkoutdate) as 'Length of Stay', guestid from booking;

Length of Stay	GuestId
3	1
3	2
3	3
3	4
3	5
3	6
3	7
3	8
3	9
3	10
3	11
3	12
3	13
3	14
3	15

Views

Between and OR

```
create view vwWinterBirthdayGuests as
select concat(firstname, ' ', lastname) as 'Guest Name', birthdate as 'Winter Birthdays' from guest
where month(birthdate) between 11 and 12 or MONTH(birthdate) between 1 and 2;
select * from vwWinterBirthdayGuests;
```

```
Guest Name      Winter Birthdays
Preston Woods   1996-11-14
Barbara Bread   1900-01-01
John Snow       1986-02-08
Grace Shoemaker         1969-12-04
Amanda Stark    1944-12-20
```

Inner Join

```
Create View vwAvailableRooms as
Select roomnumber from room
inner join roomstatus on room.roomstatusid = roomstatus.roomstatusid
where roomDescription like 'Single King' and roomstatus like 'Vacant Ready';
select * from vwAvailableRooms
where roomnumber < 200;
```

```
RoomNumber
100
101
```

Subquery

```
create view vwLatestCheckOut as select guestid, concat(firstname, ' ', lastname) as 'Guest Name'

from guest

where guestid in

(select top 1 b.guestid from booking b

order by checkOutdate asc);

select * from vwLatestCheckOut;
```

GuestId	Guest Name
2	Grae VanBonn

Aggregate

```
create view vwCountRoomTypes as Select count(roomstatusid) as "Number of Rooms",

roomDescription as "Type Of Room" from roomstatus

group by roomDescription

having count(roomstatusid) > 3;

select * from vwCountRoomTypes;
```

Number of Rooms	Type Of Room
6	Double Queen
6	Single King

Outer Join

```
create view vwGuestCommentsRequests as Select concat(firstname, ' ', lastname) as

'Guest Name',roomnumber, comments from guest

full outer join booking on guest.guestid = booking.guestid;

select * from vwGuestCommentsRequests;
```

Guest Name	RoomNumber	Comments
Spencer Thompson	100	Extra Towels
Grae VanBonn	101	NULL

November Rass	102	1 Extra pillow
Preston Woods	103	Extra Towels
Barbara Bread	104	Upgrade if available
Taelon Lee	200	Travelling with pet
John Snow	201	NULL
Grace Shoemaker	202	Extra Towels
Amity Coral	203	NULL
Lucy Pevensie	204	Rollaway bed
James Simpson	300	Extra room keys
Cornelious Eisenhorn	301	Late check in
Mark Meyers	302	NULL
Amanda Stark	303	NULL
Jon Bellion	304	NULL

Stored Procedures

Procedure 1

```
Create proc spVacantRoomStatus as
select concat(count(roomnumber), ': ', roomStatus) as "Count of Vacant Room's Status" from Room r
join RoomStatus rs on r.roomStatusId = rs.roomStatusId
group by roomStatus
having roomStatus like 'Vacant%'
GO
Exec spVacantRoomStatus;
```

```
All Vacant Room's Status
100: Vacant Ready
104: Vacant Ready
200: Vacant Ready
201: Vacant Ready
```

```
202: Vacant Ready

203: Vacant Ready

204: Vacant Dirty

300: Vacant Dirty

301: Vacant Dirty

302: Vacant Ready

303: Vacant Ready

304: Vacant Ready
```

Procedure 2

```
Create proc spGuestStayDuration @guestid integer as

declare @days integer;

declare @guestname varchar(40);

select @days =  DATEDIFF(day, checkindate, checkoutdate), @guestname = convert(varchar,
concat(firstname, ' ', lastname)) from booking

join guest on booking.guestId = guest.guestId

where guest.guestId = @guestid;

PRINT @guestname + ' will be staying for ' + convert(varchar(2), @days) + ' nights.';

GO

exec spGuestStayDuration 2;
```

```
Grae VanBonn will be staying for 3 nights.
```

Logins and Users

The creation of a login with public role

```
Create Login User_SMT WITH Password = 'CIT%2411', Default_Database = smt_hoteldatabase

, check_expiration = off

, check_policy = off;

ALTER SERVER ROLE public ADD MEMBER User_SMT;
```

Creating a User and assigning all permissions for a table and view

```
Create User User_SMT for login User_SMT;

GO

GRANT ALTER, CONTROL, DELETE, INSERT, REFERENCES, SELECT, TAKE

OWNERSHIP, UPDATE, VIEW CHANGE TRACKING, VIEW DEFINITION

ON Guest

TO User_SMT;

GO

GRANT ALTER, CONTROL, DELETE, INSERT, REFERENCES, SELECT, TAKE

OWNERSHIP, UPDATE, VIEW CHANGE TRACKING, VIEW DEFINITION

ON vwGuestCommentsRequests

TO User_SMT;
```