The Gray Inn

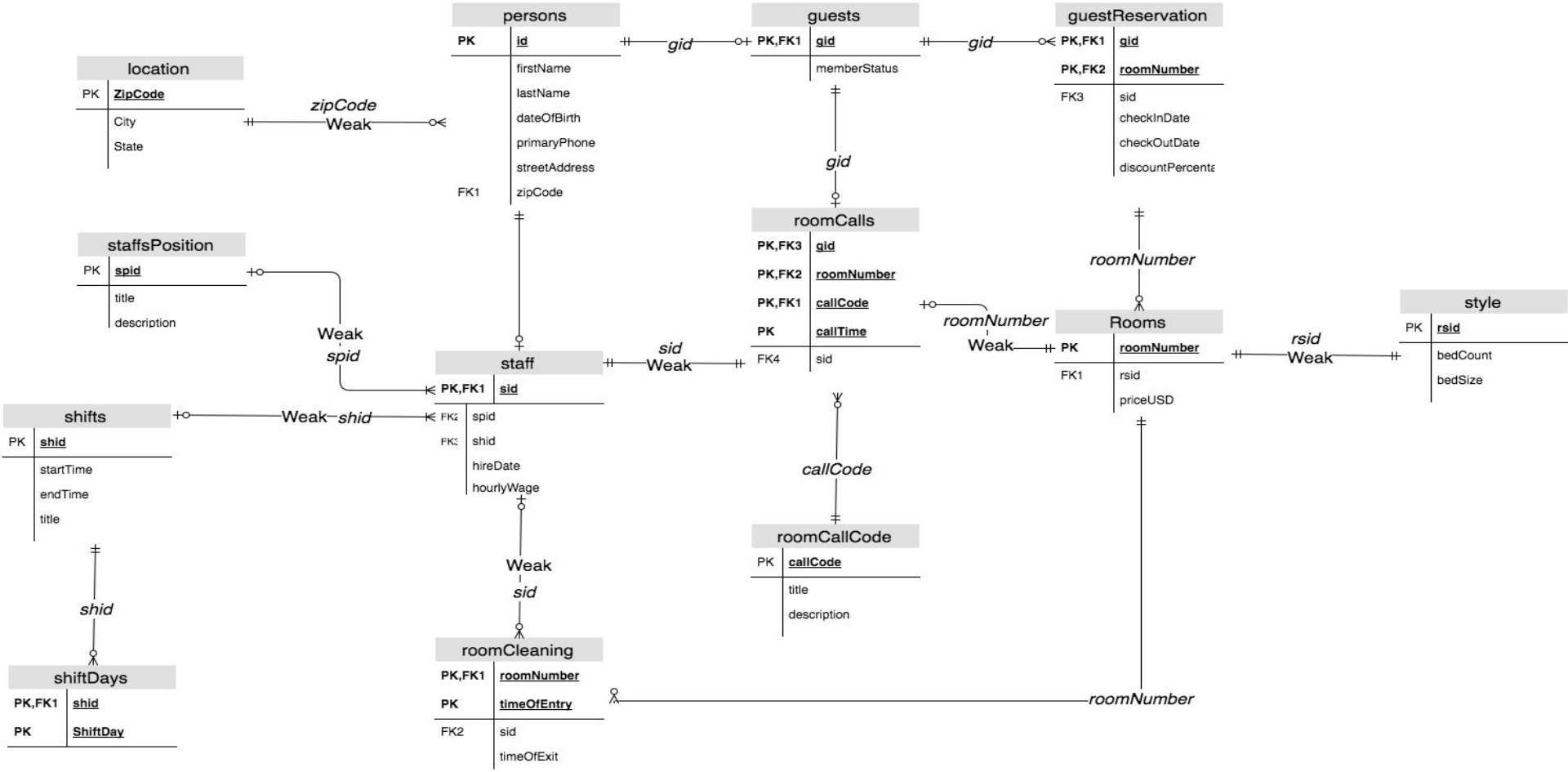
By: Dan Njoku

Table of Content

Executive Summary & Overview

The Gray Inn is a chain of hotels located all throughout the continental United States. This database in particular concentrates solely of just one hotel, however, an expanded database may be soon to come. The hospitality industry is a multi-billion dollar business that grossed an average of $175 billion in the United States in the year 2015. With the average hotel room costing around $120 dollars per night, it is safe to say that the owning a hotel can prove to be a is a fairly lucrative business venture. In the year 2015 the average occupancy rate of hotels across the United States reached a staggering 65.6 percent, an increase of 11 percent dated 6 years prior, during recession of 2009. The purpose of this database is to track the daily operations within the hotel. The data can be used by administration, through the use of queries in order to track the day to day operations that are taking place within the hotel. Through managing The Gray Inn, this database implementation will be able to track the hotels staff and guests in a more accurate and concise manner.

The Gray Inn - ER Diagram



Persons Table

Due to the fact that staff members may also be guests in the hotel, there is a separate table titled persons, used to store information basic perosnal information regarding the individual.

```
CREATE TABLE persons (
    pid             INTEGER     NOT NULL,
    firstName       VARCHAR(50) NOT NULL,
    lastName        VARCHAR(50) NOT NULL,
    dateOfBirth     DATE        NOT NULL,
    phonePrimary    CHAR(15)    NOT NULL,
    streetAddress   VARCHAR(50) NOT NULL,
    zipCode         INTEGER     NOT NULL    REFERENCES location(zipCode),
PRIMARY KEY(pid)
```

Functional Dependencies

Pid → firstName, lastName, dateOfBirth,
primaryPhone, streetAddress, zipCode

Table is on the following page

Persons Table

pid	firstName	lastName	dateOfBirth	phonePrimary	streetAddress	zipCode
1	Dan	Smith	19681208	231323276	22 Oak Street	37379
2	Will	Atkins	19630830	8996439355	654 Glenwood Drive	31021
3	Bill	Cosby	19520219	8442851051	252 13th Street	9093
4	Amy	Shummer	19580919	8110049929	396 Grove Street	12366
5	Josh	Merrick	19580919	8554809490	227 Devonshire Drive	45431
6	Randy	Queen	19630702	8993496951	138 Devon Court	99959
7	Dante	Matthew	19600529	8553744253	19 Water Street	34234
8	Harry	Johnson	19580109	8555815117	125 Spring Street	11533
9	Marco	Birkshire	19690702	8339754754	440 Canal Street	11533
10	Ruben	Hathaway	19280919	8990479981	736 3rd Street North	11533
11	Mike	Forte	19681208	8446578110	529 Laurel Street	12601
12	Cameron	Nero	19681208	8220221397	595 Parker Street	12601
13	Katie	Carty	19681208	8557264814	602 6th Street North	12601
14	Siobain	Rogers	19681208	8116286109	315 Redwood Drive	12601
15	Scott	Botts	19681208	8226757964	973 Lakeview Drive	12601
16	Sarah	Rogers	19681208	8334068368	92 Beechwood Drive	12601
17	Annie	Dumpty	19681208	8442569423	618 Madison Avenue	12601
18	Lee	Wee	19681208	8444814554	618 Madison Avenue	12601
19	Will	Williams	19681208	8994765763	944 George Street	12601
20	Amanda	Scott	19681208	8441270429	644 Marshall Street	12601

Location The locations table stores the city and state for its given zip code.

```
CREATE TABLE location (
    zipCode INTEGER     NOT NULL,
    city    VARCHAR(50) NOT NULL,
    state   TEXT(2)     NOT NULL,
PRIMARY KEY(zipCode)
```

Function Dependencies

zipCode → city, state

zipCode	city	state
9093	Eugene	OR
11533	Soddy Daisy	TX
12366	Smyra	CA
12433	Campbell	NC
12601	Poughkeepsie	NY
31021	Dublin	GA
34234	Jacksonville	FL
34343	Buffalo	NY
37379	Soddy Daisy	TN
45431	Uniondale	NY
99959	Garden City	NY

Guests

The purpose of the guest table is to identify each guest based on their gid/pid, and also to specify their member status

```
CREATE TABLE guests (
    gid             INTEGER   NOT NULL REFERENCES persons(pid),
    memberStatus    TEXT      NOT NULL,
PRIMARY KEY(gid)
);
```

Function Dependencies

gid → memberStatus

gid	memberStatus
1	Bronze
2	Bronze
3	Bronze
4	Bronze
5	Silver
6	Bronze
7	Bronze
8	Gold
9	Bronze
10	Platinum

Room Calls

The objective of this table is to track each guest's room calls

```
CREATE TABLE roomCall (
    gid          INTEGER      NOT NULL REFERENCES guests(gid),
    roomNumber   VARCHAR(5)   NOT NULL REFERENCES rooms(roomNumber),
    callCode     INTEGER      NOT NULL REFERENCES roomCallCode(callCode),
    callTime     TIME         NOT NULL,
    sid          INTEGER      NOT NULL REFERENCES staff(sid),
PRIMARY KEY(callTime),
);
```

Function Dependencies

gid, roomNumber, callCode, callTime → sid

gid	roomNumber	callCode	callTime	sid
1	101	4	20:19:19	11
1	101	4	18:19:19	11
3	103	2	19:20:19	12

Room Call Codes

This table gives some rudimentary examples to what some call codes can consist of.

```
CREATE TABLE roomCallCode (
    callCode        integer      NOT NULL,
    title           VARCHAR(50)  NOT NULL,
    description     TEXT         NOT NULL,
PRIMARY KEY(callCode)
);
```

callCode	title	description
1	Maintainance	If its broke we fix it
2	Cleaning	New sheets? No problem
3	Room Service	You Hungry?
4	Help Desk	You have a question? Well we have answers

Function Dependencies

callCode → title, description

Staff

This table provides us with the basic information needed in regard to the hotels employees and their job entailment.

```
CREATE TABLE staff (
    sid             INTEGER     NOT NULL REFERENCES persons(pid),
    spid            INTEGER     NOT NULL REFERENCES staffsPosition(pi
    shid            INTEGER     NOT NULL REFERENCES shifts(shid),
    hireDate        DATE        NOT NULL,
    hourlyWageUSD   DECIMAL     NOT NULL,
PRIMARY KEY(sid)
);
```

sid	spid	shid	hireDate	hourlyWageUSD
11	1	3	2002	15.25
12	1	3	1991	10.25
13	1	2	1998	14.25
14	1	2	1995	10.25
15	1	0	1992	10.25
16	1	2	2000	13.25
17	2	0	2002	10.25
18	2	1	1986	11.25
19	2	2	1992	9.25
20	2	3	2002	9.25

Function Dependencies

sid → spid, shid, hireDate, hourlyWage

Shifts This table gives some rudimentary shift hours that the staff could possible work.

```
CREATE TABLE shifts (
    shid        INTEGER     NOT NULL,
    startTime   TIME        NOT NULL,
    endTime     TIME        NOT NULL,
    title       TEXT,
PRIMARY KEY(shid)
);
```

Function Dependencies

shid → startTime, endTime, title

shid	startTime	endTime	title
0	00:00:00	06:00:00	Morning
1	06:00:00	12:00:00	Day
2	12:00:00	18:00:00	Evening
3	18:00:00	00:00:00	Night

Shift Days

Specifies what shifts are available on each specific day of the week

```
CREATE TABLE shiftDays (
    shid        INTEGER      NOT NULL  REFERENCES shifts(shid),
    shiftDay    VARCHAR      NOT NULL    CHECK
                (shiftDay IN ('Sunday', 'Monday','Tuesday', 'Wednesday', 'Thursday','Friday','Saturday')),
PRIMARY KEY(shid, shiftDay)
);
```

Function Dependencies

shid, shiftDay →

Table is located on the following page

ShiftDays Table

shid	shiftDay
0	Monday
0	Tuesday
0	Wednesday
0	Thursday
0	Friday
0	Saturday
0	Sunday
1	Monday
1	Tuesday
1	Wednesday
1	Thursday
1	Friday
1	Saturday
1	Sunday
2	Monday
2	Tuesday
2	Wednesday
2	Thursday
2	Friday
2	Saturday
2	Sunday
3	Monday
3	Tuesday
3	Wednesday
3	Thursday
3	Friday
3	Saturday
3	Sunday

Staff's Position Breaks down the staffs' positions into three main categories

```
CREATE TABLE staffsPosition (
    spid        INTEGER     NOT NULL,
    title       INTEGER     NOT NULL,
    deScription  VARCHAR,
PRIMARY KEY(spid)
);
```

spid	title	deScription
1	Administration	Managerial roll
2	Cleaner	Cleans the hotel room

Function Dependencies

spid → title, description

Rooms Gives the details of each room located in the hotel

```
CREATE TABLE rooms (
    roomNumber   VARCHAR(5)   NOT NULL,
    rsid         INTEGER      NOT NULL,
    priceUSD     DECIMAL      NOT NULL,
PRIMARY KEY(roomNumber),
FOREIGN KEY(rsid) REFERENCES style(rsid)
);
```

Function Dependencies

roomNumber→ rsid, priceUSD

roomNumber	rsid	priceUSD
101	1	200
102	2	150
103	3	120
104	4	250
105	5	200
201	1	200
202	2	150
203	3	120
204	4	250
205	8	350

Style

This table represents the style of each individual room, in regard to bed count and the sizes of the bed(s).

```
CREATE TABLE style (
    rsid        INTEGER     NOT NULL,
    bedCount    CHAR(1)     NOT NULL,
    bedSize     TEXT        NOT NULL,
PRIMARY KEY(rsid)
);
```

rsid	bedCount	bedSize
1	1	King
2	1	Queen
3	1	Full
4	2	Queen
5	2	Full
6	2	Twin
7	2	King

Function Dependencies

rsid → bedCount, bedSize

Guest Reservation

This tables tracks the basic information of each guest through the duration of their stay.

gid	roomNumber	sid	checkInDate	checkOutDate	discountPercentage
1	101	13	1991	1989	0
2	102	13	1990	1989	0
3	103	13	1989	1988	0
4	104	13	1989	1987	0
5	105	13	1992	1989	5
6	201	13	1992	1989	0
7	202	13	1991	1989	0
8	203	13	1990	1986	10
9	204	13	1991	1989	0
10	205	13	1988	1985	12.5

Function Dependencies

gid, roomNumber → sid,checkInDate, checkOutDate, discountPercentage

```
CREATE TABLE guestReservationS (
    gid                 INTEGER      NOT NULL REFERENCES guest(gid),
    roomNumber          VARCHAR(5)   NOT NULL REFERENCES rooms(roomNumber),
    sid                 INTEGER      NOT NULL REFERENCES staff(sid),
    checkInDate         DATE         NOT NULL,
    checkOutDate        DATE         NOT NULL,
    discountPercentage  DECIMAL      NOT NULL,
PRIMARY KEY(gid,roomNumber,sid)
);
```

Room Cleaning

Specifies when each hotel room was last cleaned and which staff member cleaned it

```
CREATE TABLE roomCleaning (
    roomNumber   INTEGER     NOT NULL  REFERENCES rooms(roomNumber),
    timeOfEntry  TIME        NOT NULL,
    sid          INTEGER     NOT NULL  REFERENCES staff(sid),
    timeOfExit   TIME        NOT NULL,
PRIMARY KEY(roomNumber)
);
```

roomNumber	timeOfEntry	sid	timeOfExit
101	12:00:00	17	12:30:00
103	15:00:00	17	15:30:00
105	14:00:00	17	14:30:00
201	16:00:00	17	16:30:00

Function Dependencies

roomNumber, timeOfEntry → sid, timeOfExit

Views

GuestRooms

- The objective of this view is to track what guests have checked out of the hotel.

Create Statement

```
CREATE VIEW guestRooms AS
SELECT per.firstName,
       per.lastName,
FROM   persons per,
       guests  g,
       guestReservation gr,
       rooms r
WHERE    per.pid = g.gid
AND      gr.gid = g.gid
AND      gr.checkOutDate IS NOT NULL
ORDER BY per.lastname ASC;
```

RoomAvailability

- This view provides the hotel with information related to what rooms are currently available

Create statement

```
CREATE VIEW roomAvailability AS
SELECT rooms.roomNumber AS Room,
FROM rooms,
WHERE rooms.roomNumber NOT IN (
                              SELECT roomNumber
                              FROM   guestReservation
                              WHERE  checkOutDate IS NULL)
ORDER BY rooms.roomNumber ASC;
```

Reports

The Average Amount of time it takes to clean a room

- The objective of this query is to track how long it takes for each employee to clean each individual room

Query

```
SELECT staff.sid As Cleaner
    avg(rc.timeOfEntry - rc.timeOfExit) AS AVG Cleaning Time
FROM    staff s
        roomCleaning rc
WHERE rc.timeOfExit IS NOT NULL
AND     s.sid = rc.sid;
GROUP BY s.sid;
```

Average amount of time a guest stays in the hotel

- The objective of the following query is to track the amount of time the average quest stays at the Gray INN

Query

```
SELECT g.gid as GuestStay
    avg(gr.checkInDate - gr.checkOutDate)  AS Average Stay
FROM    guestReservation gr,
        guest g,
        rooms r
WHERE   gr.checkOutDate IS NOT NULL
AND     r.roomNumber = gr.roomNumber
GROUP BY g.gid;
```

Stored Procedures

addNewReservation

```
CREATE OR REPLACE FUNCTION insertGuestReservation ()
RETURNS tigger AS
$$
    BEGIN
        IF NEW.roomNumber IS NULL THEN
            RAISE EXPECTATION 'INVALID ROOM NUMBER!';
        END IF;
        IF NEW.checkInTime IS NULL THEN
            RAISE EXCPECTION 'This room is currently available';
        END IF;
        IF old.checkInTime IS NOT NULL THEN
            RAISE EXCPECTION 'This room is already occupied';
        END IF;
        INSERT INTO guestReservation (roomNumber, checkInTime)
                            VALUES    (NEW.roomNumber, 'now');
        Return NEW
    END;
$$ LANGUAGE pipgsql;
```

guestCheckOut

```
CREATE OR REPLACE FUNCTION guestCheckOut
RETURNS trigger AS
$$
    DECLARE
    ID INTEGER
    checkInTime TIMESTAMP
BEGIN
    IF NEW.checkOutTime IS NOT NULL THEN
    RAISE EXPCEPTION ('BY PROCEEDING YOU WILL VOID YOUR RESERVATION!')
END IF;

pid = NEW.gid
checkInTime = NEW.checkInTime

    Update guestReservation
    SET checkOutTime = 'now'
    WHERE pid = gid
        AND checkInTime = checkInTime
        AND checkOutTime IS NULL
    RETURN NEW
END;
$$ LANGUAGE pipgsql;
```

Triggers

AddRoomtoCleaningJob

- The purpose of this trigger in this scenario is to add a room to be cleaned once the guest has checked out of their room. Rooms may still be clean when occupied by the guest

Query

```
CREATE TRIGGER addRoomToRoomCleaning
AFTER UPDATE ON guestReservation
FOR EACH ROW EXECUTE PROCEDURE  insertRoomCleaning();
```

Security

There are three primary users within the hotel. For each role the specified user is allowed certain permissions are only given to certain staff members as shown.

Guests

```
GRANT INSERT ON roomCalls TO guests;
```

Housekeeping

```
GRANT SELECT, UPDATE ON roomCleaning TO cleaner;
GRANT SELECT ON rooms TO cleaner;
```

Administration

```
GRANT SELECT ON location TO Administration;
GRANT SELECT, INSERT, UPDATE ON persons TO Administration;
GRANT SELECT, INSERT, UPDATE ON guests TO Administration;
GRANT SELECT, INSERT, UPDATE ON staff TO Administration;
GRANT SELECT, INSERT, UPDATE, DELETE ON staffPosition TO Administration;
GRANT SELECT, INSERT, UPDATE, DELETE ON shifts TO Administration;
GRANT SELECT, INSERT, UPDATE, DELETE ON shiftDays TO Administration;
GRANT SELECT, INSERT, UPDATE ON patientVisits TO Administration;
GRANT SELECT, INSERT, UPDATE ON bedAssignments TO Administration;
GRANT SELECT, UPDATE ON roomCalls TO Administration;
GRANT SELECT, INSERT, UPDATE, DELETE ON callCodes TO Administration;
GRANT SELECT, UPDATE ON roomCleaning TO Administration;
GRANT SELECT, UPDATE ON rooms TO Administration;
GRANT SELECT, UPDATE ON style TO Administration;
```

Notes - Issues - Future Enhancements

Some future enhancements that will prove to be beneficial to the day to day functionality of the hospital would be to allowing a more diverse staff selection, giving staff members more roles besides housekeeping, administration, and working in the kitchen. Also, there should be a lot more flexibility with the amount of shifts, in regard to the shift hours and shift days. Through the expansion of the hotel database we can potentially add more space for private events in order to generate more revenue, and as well as increase the total number of rooms available.

Perhaps if I implemented more test data, similar to that of a real hotel, I would have been able to create more advanced queries. For future implementation I would like to specify different administrative roles, as well as move into the daily operations of the kitchen and housekeeping staff.