-- Timur Naimov

-- Professor Jose

-- CIS 395-1900

-- April 7, 2022

-- Database Project 1 - Enhancement of the Hotel Database

**Project Deliverables**:

1. **UML Diagram or Table layout of the new tables. (10 points)**

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, table

Description automatically generated

**2.  DDL statements for each of items 1, 2, 3. (10 points each, 30 points)**

-- 1. Enhance the primary keys of each base tables (relations) to utilize automatic generation of the primary keys.

use hoteldb;

alter table booking modify column hotelNo int;

alter table booking modify column guestNo int;

alter table booking modify column roomNo int;

alter table hotel modify column hotelNo int auto\_increment;

alter table room modify column roomNo int;

alter table room modify column hotelNo int;

alter table guest modify column guestNo int auto\_increment;

alter table booking modify column hotelNo int primary key auto\_increment;

-- 2. Enhance the tables guest, hotel which store address information. Include real-world address fields as applicable.

alter table hotel add column stateCd char(2);

alter table hotel add column zip varchar(20);

alter table hotel add column countryCd varchar(200);

alter table guest add column stateCd char(2);

alter table guest add column zip varchar(20);

alter table guest add column countryCd varchar(200);

-- 3. Enhance the booking table to use a unique column as primary key. Apply a uniqueness constraint to the booking table using the columns: guestNo, hotelNo, roomNo, startDate. Lastly, add a datetime column to the booking table to keep a record of the latest insert or update of a row in the booking table.

alter table booking add column bookingNo int not null primary key auto\_increment unique;

alter table booking add unique index booking\_idx (guestNo, hotelNo, roomNo, dateFrom);

alter table booking add column lastMod timestamp default current\_timestamp;

**3.  Word document or txt file of the process described in item 4.  You can include SQL, or a flow chart, pseudocode and any additional tables. Images of your test and results using your local MySQL database can be pasted in the document. (30 points)**

-- 4. In a short paragraph, describe a process to determine if a new booking will have an intersection of rows with a previous booking of the same guestNo, hotelNo, and roomNo.

The process to determine if a guest making a new booking has overlapping days with a prior booking is

by enhancing the hoteldb schema to add a calendar of days of the years in scope. Then, SQL can be used to determine

if dates from one booking row intersect with dates of another booking row by counting the rows for each date

within each booking record or the SQL can be selecting using the guestNo for similar results.

To check two booking records on whether days overlap, I will use this code:

use hoteldb;  
select t1.tdate, count(\*)  
from tcalendar t1

, booking b1  
where (t1.tdate >= b1.dateFrom and t1.tdate <= b1.dateTo) and b1.guestNo = 1  
group by t1.tdate  
having count(\*) > 1

order by t1.tdate;

**4.  Word or txt file of the design, DDL and DML for the requirements of item 5. You can include the CREATE TABLE, INSERT, and SELECT statements in the document.  Images of your test and results using your local MySQL database can be pasted in the document (30 points).**

-- 5. In a short paragraph, describe how we use SQL to avoid hard-coding tax rates in the SQL.

To avoid hard-coding tax rates in the SQL, I will design a table called taxes that will have stateCd column and countryCD column, and have a tax rate column called taxrate. Then, I will INSERT records and write SQL to calculate cost of stay using the tax rate from the table.

I will create the table using this code:

create table taxes (countryCD varchar(2) not null primary key, stateCd varchar(2) not null, taxrate dec(12,4) not null default 0);

To check if avoiding hard-coding works, I will use this code:

select r.price, r.hotelNo, h.hotelNo, h.stateCd, t.state,

t.taxrate, b.guestNo, g.guestNo, r.price \* t.taxrate

as final\_price

from hotel h

join room r on h.hotelNo = r.hotelNo

join taxes t on h.stateCd = t.stateCd

join booking b on g.guestNo = b.guestNo;

select r.price, r.hotelNo, h.hotelNo, h.stateCd, t.taxRate, t.stateCd,

r.price + (r.price \* t.taxrate) as final\_price

from hotel h

join room r on h.hotelNo = r.hotelNo

join taxes t on h.stateCd = t.stateCd;

use hoteldb;

create table codeName

(code varchar(20) not null

,codeType varchar(20) not null

,name varchar(60)

,primary key (code, codeType));

use hoteldb;

insert into codeName (code, codeType, name) values ('NJ','STATE','New Jersey');

insert into codeName (code, codeType, name) values ('NY','STATE','New York');

insert into codeName (code, codeType, name) values ('CA','STATE','California');

insert into codeName (code, codeType, name) values ('IL','STATE','Illinois');

insert into codeName (code, codeType, name) values ('PA','STATE','Pennsylvania');

insert into codeName (code, codeType, name) values ('AZ','STATE','Arizona');

insert into codeName (code, codeType, name) values ('CO','STATE','Colorado');

insert into codeName (code, codeType, name) values ('CT','STATE','Connecticut');

insert into codeName (code, codeType, name) values ('MA','STATE','Massachusetts');

insert into codeName (code, codeType, name) values ('VA','STATE','Virginia');

insert into codeName (code, codeType, name) values ('DC','STATE','District of Columbia');

insert into codeName (code, codeType, name) values ('FL','STATE','Florida');

insert into codeName (code, codeType, name) values ('GA','STATE','Giorgia');

insert into codeName (code, codeType, name) values ('SC','STATE','South Carolina');

insert into codeName (code, codeType, name) values ('NC','STATE','North Carolina');

insert into codeName (code, codeType, name) values ('DE','STATE','Delaware');

insert into codeName (code, codeType, name) values ('WV','STATE','West Virginia');

insert into codeName (code, codeType, name) values ('OR','STATE','Oregon');

insert into codeName (code, codeType, name) values ('WA','STATE','Washington');

insert into codeName (code, codeType, name) values ('MN','STATE','Minnesota');

insert into codeName (code, codeType, name) values ('ME','STATE','Maine');

insert into codeName (code, codeType, name) values ('VT','STATE','Vermont');

insert into codeName (code, codeType, name) values ('IN','STATE','Indiana');

insert into codeName (code, codeType, name) values ('MO','STATE','Missouri');

insert into codeName (code, codeType, name) values ('OK','STATE','Oklahoma');

insert into codeName (code, codeType, name) values ('TN','STATE','Tennessee');

insert into codeName (code, codeType, name) values ('AL','STATE','Alabama');

insert into codeName (code, codeType, name) values ('OH','STATE','Ohio');

insert into codeName (code, codeType, name) values ('TX','STATE','Texas');

insert into codeName (code, codeType, name) values ('NM','STATE','New Mexico');

insert into codeName (code, codeType, name) values ('NV','STATE','Nevada');

insert into codeName (code, codeType, name) values ('MT','STATE','Montana');

insert into codeName (code, codeType, name) values ('UT','STATE','Utah');

insert into codeName (code, codeType, name) values ('HI','STATE','Hawaii');

insert into codeName (code, codeType, name) values ('AK','STATE','Alaska');

insert into codeName (code, codeType, name) values ('AR','STATE','Arkansas');

insert into codeName (code, codeType, name) values ('ND','STATE','North Dakota');

insert into codeName (code, codeType, name) values ('SD','STATE','South Dakota');

insert into codeName (code, codeType, name) values ('ID','STATE','Idaho');

insert into codeName (code, codeType, name) values ('WY','STATE','Wyoming');

insert into codeName (code, codeType, name) values ('IO','STATE','IOWA');

insert into codeName (code, codeType, name) values ('KS','STATE','Kansas');

insert into codeName (code, codeType, name) values ('WI','STATE','Wisconsin');

insert into codeName (code, codeType, name) values ('KN','STATE','Kentucky');

insert into codeName (code, codeType, name) values ('NH','STATE','New Hampshire');

insert into codeName (code, codeType, name) values ('LA','STATE','Louisiana');

insert into codeName (code, codeType, name) values ('MD','STATE','Maryland');

insert into codeName (code, codeType, name) values ('MS','STATE','Mississippi');

insert into codeName (code, codeType, name) values ('NB','STATE','Nebraska');

insert into codeName (code, codeType, name) values ('RI','STATE','Rhode Island');