DBMS PROJECT
HOTEL MANAGEMENT SYSTEM

PRANEETH S (PES1UG21CS437)

RAJDEEP JAT(PES1UG21CS475)

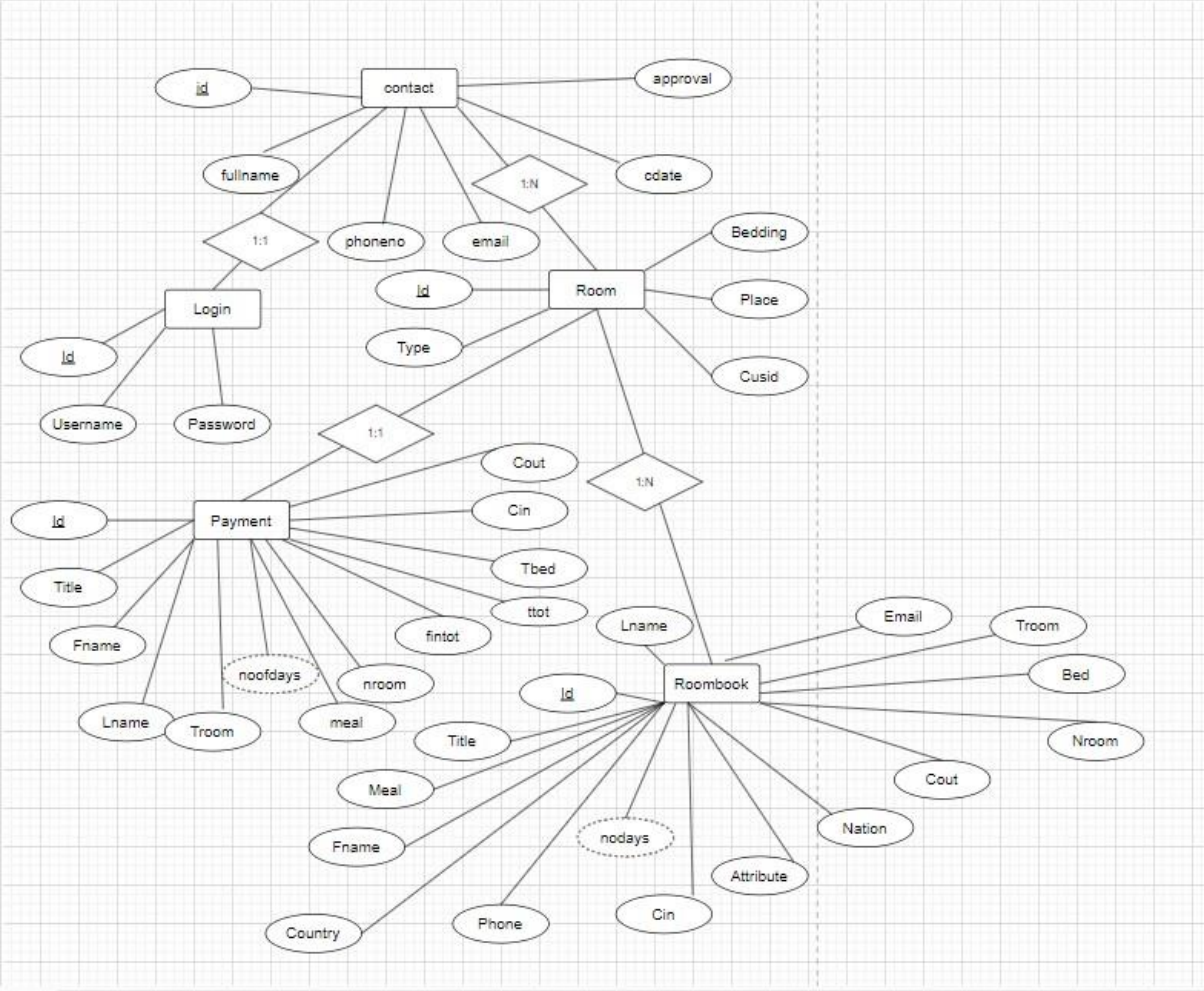
Context:

In the dynamic and competitive hospitality industry, effective management of hotel operations is paramount to ensure smooth customer experiences and efficient business processes. Traditional manual methods for managing hotel tasks, such as room reservations, check-ins, check-outs, and inventory management, are prone to errors and inefficiencies. The need for a reliable and automated solution has become increasingly evident to enhance customer satisfaction, streamline operations, and improve overall business performance.

2. Objectives: The primary objectives of the Hotel Management System project are as follows: Automation of Operations: Implement an

automated system to handle core hotel functions, reducing manual effort and minimizing errors in tasks like room bookings, guest check-ins, and check-outs. Real-time Information Access: Provide a platform that allows hotel staff to access real-time information about room availability, guest details, and other relevant data to facilitate quick decisionmaking. Enhanced Customer Experience: Improve guest satisfaction by ensuring seamless and efficient services, from the booking process to the departure, through features such as online reservations and simplified check-in/check-out procedures. Inventory and Resource Management: Efficiently manage hotel resources, including room inventory, staff assignments, and supplies, to optimize utilization and reduce wastage. Security and Data Integrity: Implement robust security measures to safeguard sensitive guest information and ensure the integrity of the data stored within the system

SYSTEM DESIGN ER DIAGRAM:



RELATIONAL SCHEMA

- **`contact`** table: (id [PK], fullname, phoneno, email, cdate, approval)
- **`login`** table: (id [PK, FK referencing contact.id], usname, pass)
- **`payment`** table: (id [PK, FK referencing room.id], title, fname, lname, troom, tbed, nroom, cin, cout, ttot, fintot, mepr, meal, btot, noofdays)
- **`room`** table: (id [PK], type, bedding, place, cusid [FK referencing contact.id])
- **`roombook`** table: (id [PK, FK referencing room.id], Title, FName, LName, Email, National, Country, Phone, TRoom, Bed, NRoom, Meal, cin, cout, stat, nodays)
- **`audit_table`** table: (id [PK], action, table_name, record_id, timestamp)

FUNCTION AND QUERY USED

Triggers

```
-- Triggers
DELIMITER //


-- Trigger before inserting into payment table
CREATE TRIGGER before_payment_insert
BEFORE INSERT
ON payment FOR EACH ROW
BEGIN
    SET NEW.fintot = NEW.ttot * 1.1; -- Assuming a 10% increase for
demonstration purposes
END;
//


-- Trigger after inserting into payment table
CREATE TRIGGER after_payment_insert
AFTER INSERT
ON payment FOR EACH ROW
BEGIN
    -- Logging the insertion in the audit table
    INSERT INTO audit_table (action, table_name, record_id)
    VALUES ('INSERT', 'payment', NEW.id);
END;
//
DELIMITER ;
```

FUNCTIONS

```
DELIMITER //

CREATE FUNCTION calculate_total_cost(nights INT, room_rate DECIMAL(8,2))
RETURNS DECIMAL(8,2)
DETERMINISTIC
NO SQL
BEGIN
    DECLARE total_cost DECIMAL(8,2);
    SET total_cost = nights * room_rate;
    RETURN total_cost;
END;
//

DELIMITER ;
```

FUNCTIONS WITH NESTED QUERY

```
DELIMITER //

CREATE FUNCTION get_approval_status_from_roombook(roombook_id INT) RETURNS
VARCHAR(20)
DETERMINISTIC
NO SQL
BEGIN
    DECLARE approval_status VARCHAR(20);

    -- Using a nested query to concatenate 'Approval: ' with the approval
status
    SELECT CONCAT('Approval: ', stat)
    INTO approval_status
    FROM roombook
    WHERE id = roombook_id;

    RETURN approval_status;
END;
//

DELIMITER ;
```

JOIN QUERY WITH AGGREGATE FUNCTIONS

```
SELECT
    r.type AS room_type,
    COUNT(p.id) AS payment_count,
    SUM(p.ttot) AS total_amount
FROM
    room r
LEFT JOIN
    payment p ON r.id = p.id
GROUP BY
    r.type;
```

QUERY'S EXEUTED:

```
mysql> select * from audit_table;
+----+--------+------------+-----------+---------------------+
| id | action | table_name | record_id | timestamp           |
+----+--------+------------+-----------+---------------------+
|  1 | INSERT | payment    |         2 | 2023-11-22 15:02:25 |
|  2 | INSERT | payment    |         3 | 2023-11-26 16:34:03 |
|  3 | INSERT | payment    |         4 | 2023-11-26 16:36:21 |
+----+--------+------------+-----------+---------------------+
3 rows in set (0.00 sec)
```

```
mysql> select * from audit_table;
+----+--------+------------+-----------+---------------------+
| id | action | table_name | record_id | timestamp           |
+----+--------+------------+-----------+---------------------+
|  1 | INSERT | payment    |         2 | 2023-11-22 15:02:25 |
|  2 | INSERT | payment    |         3 | 2023-11-26 16:34:03 |
+----+--------+------------+-----------+---------------------+
2 rows in set (0.01 sec)
```

```
mysql> Select get_approval_status_from_roombook(2);
+---------------------------------------+
| get_approval_status_from_roombook(2) |
+---------------------------------------+
| Approval: Conform                     |
+---------------------------------------+
1 row in set (0.01 sec)
```

```
mysql> select calculate_total_cost(5,3200);
+------------------------------+
| calculate_total_cost(5,3200) |
+------------------------------+
|                     16000.00 |
+------------------------------+
1 row in set (0.01 sec)
```

```
mysql> SELECT
    ->     r.type AS room_type,
    ->     COUNT(p.id) AS payment_count,
    ->     SUM(p.ttot) AS total_amount
    -> FROM
    ->     room r
    -> LEFT JOIN
    ->     payment p ON r.id = p.id
    -> GROUP BY
    ->     r.type;
+---------------+---------------+--------------+
| room_type     | payment_count | total_amount |
+---------------+---------------+--------------+
| Superior Room |             2 |      1420.00 |
| Single Room   |             1 |     -1980.00 |
| Deluxe Room   |             0 |         NULL |
| Guest House   |             0 |         NULL |
+---------------+---------------+--------------+
4 rows in set (0.01 sec)
```

SQL FILE USED

```
SET SQL_MODE =
"NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+05:30";


-- Table structure for table `contact`
CREATE TABLE IF NOT EXISTS `contact` (
  `id` int(10) unsigned NOT NULL,
  `fullname` varchar(100) DEFAULT NULL,
  `phoneno` int(10) DEFAULT NULL,
  `email` text,
  `cdate` date DEFAULT NULL,
  `approval` varchar(12) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=1;


-- Dumping data for table `contact`
```

```
INSERT INTO `contact` (`id`, `fullname`,
`phoneno`, `email`, `cdate`, `approval`) VALUES

(1, 'John Doe', 1234567890,
'john.doe@example.com', '2023-01-01', 'Approved');


-- Table structure for table `login`

CREATE TABLE IF NOT EXISTS `login` (

  `id` int(10) unsigned NOT NULL,

  `usname` varchar(30) DEFAULT NULL,

  `pass` varchar(30) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=3;


-- Dumping data for table `login`

INSERT INTO `login` (`id`, `usname`, `pass`)
VALUES

(1, 'admin', '1234');
```

```
-- Table structure for table `payment`
CREATE TABLE IF NOT EXISTS `payment` (
  `id` int(10) ,
  `title` varchar(5) DEFAULT NULL,
  `fname` varchar(30) DEFAULT NULL,
  `lname` varchar(30) DEFAULT NULL,
  `troom` varchar(30) DEFAULT NULL,
  `tbed` varchar(30) DEFAULT NULL,
  `nroom` int(10) DEFAULT NULL,
  `cin` date DEFAULT NULL,
  `cout` date DEFAULT NULL,
  `ttot` double(8,2) DEFAULT NULL,
  `fintot` double(8,2) DEFAULT NULL,
`mepr` double(8,2) DEFAULT NULL,   `meal`
varchar(30) DEFAULT NULL,
  `btot` double(8,2) DEFAULT NULL,
  `noofdays` int(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1; --
Table structure for table `room`
```

```
CREATE TABLE IF NOT EXISTS `room` (
  `id` int(10) unsigned NOT NULL,
  `type` varchar(15) DEFAULT NULL,
  `bedding` varchar(10) DEFAULT NULL,
  `place` varchar(10) DEFAULT NULL,
  `cusid` int(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=16;


-- Dumping data for table `room`
INSERT INTO `room` (`id`, `type`, `bedding`,
`place`, `cusid`) VALUES
(1, 'Superior Room', 'Single', 'Free', NULL),
(2, 'Superior Room', 'Double', 'Free', NULL),
(3, 'Superior Room', 'Triple', 'Free', NULL),
(4, 'Single Room', 'Quad', 'Free', NULL),
(5, 'Superior Room', 'Quad', 'Free', NULL),
(6, 'Deluxe Room', 'Single', 'Free', NULL),
(7, 'Deluxe Room', 'Double', 'Free', NULL),
```

```
(8, 'Deluxe Room', 'Triple', 'Free', NULL),
(9, 'Deluxe Room', 'Quad', 'Free', NULL),
(10, 'Guest House', 'Single', 'Free', NULL),
(11, 'Guest House', 'Double', 'Free', NULL),
(12, 'Guest House', 'Quad', 'Free', NULL),
(13, 'Single Room', 'Single', 'Free', NULL),
(14, 'Single Room', 'Double', 'Free', NULL),
(15, 'Single Room', 'Triple', 'Free', NULL);

-- Table structure for table `roombook`
CREATE TABLE IF NOT EXISTS `roombook` (
  `id` int(10) unsigned NOT NULL,
  `Title` varchar(5) DEFAULT NULL,
  `FName` text,
  `LName` text,
  `Email` varchar(50) DEFAULT NULL,
  `National` varchar(30) DEFAULT NULL,
  `Country` varchar(30) DEFAULT NULL,  `Phone` text,
```

```
  `TRoom` varchar(20) DEFAULT NULL,
  `Bed` varchar(10) DEFAULT NULL,
  `NRoom` varchar(2) DEFAULT NULL,
  `Meal` varchar(15) DEFAULT NULL,
  `cin` date DEFAULT NULL,
  `cout` date DEFAULT NULL,
  `stat` varchar(15) DEFAULT NULL,
  `nodays` int(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=2;



-- Table structure for table `audit_table`
CREATE TABLE IF NOT EXISTS `audit_table` (
  `id` int(10) unsigned NOT NULL
AUTO_INCREMENT,
  `action` varchar(50) DEFAULT NULL,
  `table_name` varchar(50) DEFAULT NULL,
`record_id` int(10) unsigned DEFAULT NULL,
  `timestamp` timestamp DEFAULT
```

```
CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Indexes for dumped tables

-- Indexes for table `contact`
ALTER TABLE `contact`
 ADD PRIMARY KEY (`id`);
-- Indexes for table `login`

ALTER TABLE `login`
 ADD PRIMARY KEY (`id`);

-- Indexes for table `newsletterlog`

-- Indexes for table `room`
ALTER TABLE `room`
 ADD PRIMARY KEY (`id`);
```

-- Indexes for table `roombook`

ALTER TABLE `roombook`

ADD PRIMARY KEY (`id`);

-- Indexes for table `payment`

ALTER TABLE `payment`

ADD PRIMARY KEY (`id`);

-- AUTO_INCREMENT for dumped tables

-- AUTO_INCREMENT for table `contact`

ALTER TABLE `contact`

MODIFY `id` int(10) unsigned NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

-- AUTO_INCREMENT for table `login`

ALTER TABLE `login`

MODIFY `id` int(10) unsigned NOT NULL

AUTO_INCREMENT, AUTO_INCREMENT=3;

-- AUTO_INCREMENT for table `room`

ALTER TABLE `room`

MODIFY `id` int(10) unsigned NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=16;

-- AUTO_INCREMENT for table `roombook`

ALTER TABLE `roombook`

MODIFY `id` int(10) unsigned NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

-- AUTO_INCREMENT for table `payment`

ALTER TABLE `payment`

MODIFY `id` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
ALTER TABLE `room` MODIFY COLUMN `cusid` int(10) unsigned DEFAULT NULL;

-- Add the foreign key constraint

ALTER TABLE `room`

ADD CONSTRAINT `room_ibfk_1` FOREIGN KEY (`cusid`) REFERENCES `contact` (`id`) ON DELETE SET NULL;

-- Alter the column in 'payment' to match the data type and attributes of 'room'

ALTER TABLE `payment` MODIFY COLUMN `id` int(10) unsigned NOT NULL;

-- Add the foreign key constraint between 'payment' and 'room'

ALTER TABLE `payment`

ADD CONSTRAINT `fk_payment_room_id` FOREIGN KEY (`id`) REFERENCES `room` (`id`) ON DELETE CASCADE;

ALTER TABLE `roombook`

ADD CONSTRAINT `fk_roombook_room_id`

FOREIGN KEY (`id`) REFERENCES `room` (`id`) ON DELETE CASCADE;

-- ALTER TABLE `roombook`

-- ADD CONSTRAINT `fk_roombook_payment_id` FOREIGN KEY (`id`) REFERENCES `payment` (`id`) ON DELETE CASCADE;

ALTER TABLE `login`

ADD CONSTRAINT `fk_login_contact_id` FOREIGN KEY (`id`) REFERENCES `contact` (`id`) ON DELETE CASCADE;

-- Triggers

DELIMITER //

-- Trigger before inserting into payment table
CREATE TRIGGER before_payment_insert

BEFORE INSERT

```
ON payment FOR EACH ROW
BEGIN
    SET NEW.fintot = NEW.ttot * 1.1; -- Assuming a 10% increase for demonstration purposes
END;
//


-- Trigger after inserting into payment table
CREATE TRIGGER after_payment_insert
AFTER INSERT
ON payment FOR EACH ROW
BEGIN
    -- Logging the insertion in the audit table
    INSERT INTO audit_table (action, table_name, record_id)
    VALUES ('INSERT', 'payment', NEW.id);
END;
//
```

```
DELIMITER ;



DELIMITER //

CREATE FUNCTION calculate_total_cost(nights
INT, room_rate DECIMAL(8,2)) RETURNS
DECIMAL(8,2)
DETERMINISTIC
NO SQL
BEGIN
    DECLARE total_cost DECIMAL(8,2);
    SET total_cost = nights * room_rate;
    RETURN total_cost;
END;
//

DELIMITER ;

DELIMITER //
```

```
CREATE FUNCTION
get_approval_status_from_roombook(roombook_id
INT) RETURNS VARCHAR(20)

DETERMINISTIC

NO SQL

BEGIN

    DECLARE approval_status VARCHAR(20);


    -- Using a nested query to concatenate 'Approval: '
with the approval status

    SELECT CONCAT('Approval: ', stat)

    INTO approval_status

    FROM roombook

    WHERE id = roombook_id;


    RETURN approval_status;

END;

//
```

DELIMITER ;