



**INFORMATION AND COMMUNICATION TECHNOLOGIES**

**Report of project**

**Project name: Hotel management system**

**Students: Adilzhan Dusembay, Arailym Meshitbayeva**

**Group: Cs-2006**

**Tutor: Yerasyl Amanbek**

**Date: 29.11.2020**

## **Abstract**

The main purpose of our "Hotel management system" project is to show how data or information in hotels is managed. This has been achieved by dividing the project into various tables. During this process, we analyzed what is related to the management of the hotel and what is needed for it. And based on the results, we identified entities with their relationships. When we created the diagram, we determined which attributes should be in the table with data types to work properly. Also, one of the objectives of this project is to eliminate paperwork, save time and make the process effective.

## **Introduction**

Hospitality services for travellers have been a staple of early civilizations. For example, Japan's Nishiyama Onsen Keiunkan, built-in 705, was officially recognized by the Guinness World Records as the oldest hotel in the world. During the Middle Ages, various religious orders in monasteries and abbeys would provide lodging for travellers on the road. Since the middle of the 17th century, Inns has acted as a sleeping place for travellers in carriages. Inns started to appeal to wealthier clients in the mid-18th century and as a result, increased in size and service levels. Hotels also proliferated in Western Europe and North America throughout the 19th century. Since then, the hotel business has gone up the hill. A new opportunity for handling the hotel system was provided when the DBMS system was created in 1960. From that period the size and scale of the hotel have changed a lot, as well as the management, difficulties in it.

Nowadays, to avoid mistakes, in this case, you need to try hard. That is why creating the right DBMS can become an integral part of any hotel management. Our "Hotel management system" project gives an idea about the management in hotels. Users can make a booking for room and food. The package gives information about the user's full name, gender, contact number and room where he/she decided to stay. Also, when we did our research on hotels, we noticed that many hotels still use paper to save records. The information that is stored in the papers may be damaged, incorrectly filled in or lost/ And this rather slows down the management process. By the help of our project you can easily avoid problems like that. Our program saves data online, and this eliminates paperwork, saves time and makes the process effective. Also find information about room(number of beds, type, rate, price), reservation(number of days), employees(full name, contact number, marital status, working experience and salary on depending on the job), managers(full name, contact number, working experience and salary), job(name, description, salary, required skills for it), restaurant (located floor, number of tables in it) with menu( name, description and price), different services(additional services). The manager controls everything related to the hotel: he can dismiss and hire employees at the request of the hotel; he can analyze the statistics and make decisions for the future of the hotel. All of this will help increase the hotel's revenue.

**Methodology:** To start working on our project we had to have a good background and understand what we are going to work with. It was decided to choose the entities that we will have in our project, define business rules, and come up with some attributes that we will have in a particular table. The idea of bridge tables came to our mind later when we considered how we will collect the data, work with this data. After that we created our ERD with all the entities,relations,attributes. And finally we get started working on our project.

Normalization what is it and why do we need it? We use normalization mostly to reduce the data redundancy and dependency. We were following first three normal forms during the project.

- First normal form - The data in each table must be unique.
- Second normal form - Table must have primary key as an unique attribute, primary key can be only 1 in each table.
- Third normal form - There are transitive dependencies with primary key.

### **Entities**

The first step in the creation of this project was to define the entities that will be stored in the database. We have identified 10 entities for our project:

1. Hotel
2. Room
3. Reservation
4. Users
5. Managers
6. JobDescription
7. Employees
8. Restaurants
9. Menu
10. AdditionalServices

### **Bridge tables**

To handle multivalued dimensions we created a bridge table.

1. DishUser (To let user order same dish several times)
2. hotel\_service (To know in which hotel which service is available)
3. UserService (To let user use same service many times )

### **Business rules**

In order to regulate and influence the conduct of business, we have set up business rules. Below is a list of our business rules:

- A hotel may have 1 or more managers, but manager can work only in 1 hotel.
- A hotel may have 1 or more employees, but employee can work only in 1 hotel.
- A hotel may have 1 or more restaurants, but the restaurant is related to only 1 hotel.
- A hotel may have 1 or more additional services and additional service can be in several hotels .
- A hotel may have 1 or more rooms, but room can be related only to 1 hotel.
- A room may have 0 or reservations, but reservation contains only 1 room
- A user may have only 1 reserved room, this room can be related only to 1 person.
- A restaurant may have 1 or more dishes in it's menu, but this menu is related only to 1 restaurant.

- A manager may manage 1 or more employees, but an employee is working to only 1 manager.
- An employee may have only 1 job, but several persons may have same job.

### **Attributes**

1. Hotel consists id, name, rate, reception number and address.
2. Room has id, number of beds, types, rate, price and hotel id which is connected to hotel.
3. Reservation has id, number of ordered days, hotel id related to Hotel and room id related to Room.
4. Users consists of id, full name, gender, contact number, hotel id connected with Hotel and reservation id connected with Reservation.
5. Managers has id, full name, contact number, working experience, salary and hotel id related to Hotel.
6. JobDescription has id, name, job description, salary range and required skills.
7. Employees has id, full name, contact number, marital status, working experience, salary, job id related to JobDescription, manager id connected Managers and hotel id related to Hotel.
8. Restaurants consists id, floor where it is located, number of tables and hotel id related to Hotel.
9. Menu has dish id, name, description, price and restaurant id connected with Restaurants.
10. DishUser has id, dish id connected Menu and user id related to Users.
11. AdditionalServices has id, name, description, price and age limitation.
12. hotel\_service consists of service id connected to AdditionalServices and hotel id connected to Hotel.
13. UserService has id, service id related to AdditionalServices and user id related to Users.

### **ER diagram**



```
rate DECIMAL(5,2),
```

```
reception_number VARCHAR(60),
```

```
address VARCHAR(255)
```

```
);
```

Then we were inserting values, some values was inserted by mockaroo, some information was written by ourselves:

	employeeid	name	surname	contact	marital_status	working_experience	salary	jobid
1	21001	Frieda	Chaldecott	8052532201	single	4	45000	1
2	21005	Guido	Jinkinson	8242772031	single	3	25000	5
3	21006	Nell	Chatelain	8757706690	married	5	53000	6
4	21010	Rozamond	Tosdevin	8382900769	single	1	32000	10
5	21011	Shelly	Dredge	8543203678	married	3	26000	5
6	21012	Emmerich	Cumberledge	8219291095	single	4	29000	4
7	21016	Jamie	Tunnock	8235705685	married	3	40000	10
8	21002	Hurley	Moakes	8461510059	divorced	9	36000	2
9	21003	Lind	Roon	8998091456	married	9	27600	3
10	21004	Shelby	Kinnen	8340365723	single	7	36000	4
11	21007	Artus	Cunio	8525275975	married	8	78000	7
12	21008	Carney	McDonald	8146742291	married	7	96000	8
13	21009	Levi	Pizey	8988843075	divorced	10	40800	9
14	21013	Elberta	Found	8690326663	married	9	27600	3
15	21014	Ursola	Ormesher	8668972255	widowed	6	30240	2
16	21015	Debor	Cottham	8659008723	married	8	56760	1

Example of insert command:

```
INSERT INTO Employees (EmployeeID, name, surname, contact, working_experience,  
marital_status,salary,JobID,ManagerID,HotelID)
```

```
VALUES
```

```
(21001, 'Frieda', 'Chaldecott', '8052532201', 4, 'single', 45000, 01, 10101, 101),
```

```
(21002, 'Hurley', 'Moakes', '8461510059', 9, 'divorced', 30000, 02, 10101, 101),
```

```
(21003, 'Lind', 'Roon', '8998091456', 9, 'married', 23000, 03, 10102, 102),
```

```
(21004, 'Shelby', 'Kinnen', '8340365723', 7, 'single', 30000, 04, 10103, 103),
```

```
(21005, 'Guido', 'Jinkinson', '8242772031', 3, 'single', 25000, 05, 10104, 104),
```

```
(21006, 'Nell', 'Chatelain', '8757706690', 5, 'married', 53000, 06, 10105, 105),
```

```
(21007, 'Artus', 'Cunio', '8525275975', 8, 'married', 65000, 07, 10106, 106),
```

```
(21008, 'Carney', 'McDonald', '8146742291', 7, 'married', 80000, 08, 10107, 107),
```

```
(21009, 'Levi', 'Pizey', '8988843075', 10, 'divorced', 34000, 09, 10108, 108),
```

```
(21010, 'Rozamond', 'Tosdevin', '8382900769', 1, 'single', 32000, 10, 10109, 109),
```

```
(21011, 'Shelly', 'Dredge', '8543203678', 3, 'married', 26000, 05, 10110, 110),
```

```
(21012, 'Emmerich', 'Cumberledge', '8219291095', 4, 'single', 29000, 04, 10111, 110),
```

```
(21013, 'Elberta', 'Found', '8690326663', 9, 'married', 23000, 03, 10112, 102),
```

```
(21014, 'Ursola', 'Ormesher', '8668972255', 6, 'widowed', 25200, 02, 10111, 110),
```

```
(21015, 'Debor', 'Cottham', '8659008723', 8, 'married', 47300, 01, 10112, 102),
(21016, 'Jamie', 'Tunnock', '8235705685', 3, 'married', 40000, 10, 10106, 106);
```

Then alter table, update, delete commands:

```
--Change name salary to salary_range and change it's datatype
```

```
ALTER TABLE JobDescription RENAME COLUMN salary TO salary_range;
```

```
--Delete services for hotels with age limitation from 18 years old
```

```
DELETE FROM hotel_service
```

```
WHERE serviceid IN (
```

```
SELECT serviceid
```

```
FROM additionalservices
```

```
WHERE age_limitations = 'from the age of 18'
```

```
);
```

```
--Update the name of restaurants--
```

```
UPDATE Restaurants
```

```
SET name = 'Dinner in the Sky'
```

```
WHERE RestaurantID = 1;
```

Example:

Let's imagine that you are a user, when you first come to the hotel you will be asked to tell the data about yourself and for how many days you are going to live, which room will you choose (Users, Reservation and Room tables).

	userid	name	surname	gender	contact_number	hotelid	reservationid
1	1501	Natalee	Jelly	Female	89037672339	101	10
2	1502	Gabriela	Darcy	Female	89811300172	102	11
3	1503	Waneta	Jeffery	Female	8771921203	103	12
4	1504	Bengt	Egiloff	Male	89037672527	101	13
5	1505	Darrell	Mawby	Male	8608428756	104	14
6	1506	Armando	Manuello	Male	8371312457	104	15
7	1507	Faulkner	Harkes	Male	8382359340	105	16
8	1508	Justin	Lampel	Male	8458808919	106	17
9	1509	Matthus	Ivakin	Male	8910767839	107	18
10	1510	Dyane	Lorkins	Female	89811318625	108	19
11	1511	Delcina	Whatman	Female	8251561864	109	20
12	1512	Inger	Bertolaccini	Male	8625334890	110	21
13	1513	Issie	Tomley	Female	8720868383	105	22
14	1514	Muriel	Scholl	Female	8914704992	107	23
15	1515	Lucio	Lennox	Male	87272585444	108	24
16	1516	Frazer	Markos	Male	8232180733	110	25
17	1517	Rene	Redd	Female	89831264444	102	26

You will be a new row in this table with your reservationID.

	reservationid	number_of_day	roomid	hotelid
1	10	3	1	101
2	11	10	2	102
3	12	1	3	103
4	13	7	4	101
5	14	2	5	104
6	15	5	6	104
7	16	11	7	105
8	17	23	8	106
9	18	13	9	107
10	19	14	10	108
11	20	18	11	109
12	21	8	12	110
13	22	3	13	105
14	23	2	14	107
15	24	3	15	108
16	25	1	16	110
17	26	5	17	102

So here will be also a new row with the reservationID and roomID. Also it will contain for how long will you stay in this hotel.

	roomid	number_of_beds	type	rate	price	hotelid
1	1	1	Economy	2.50	50.00	101
2	2	2	Standart	3.50	80.00	102
3	3	1	Economy	2.30	60.00	103
4	4	4	Lux	5.00	150.00	101
5	5	3	Standart+	4.50	120.00	104
6	6	2	Standart	3.80	70.00	104
7	7	3	Standart+	4.50	120.00	105
8	8	1	Economy	2.00	40.00	106
9	9	3	Standart+	4.00	140.00	107
10	10	4	Lux	5.00	200.00	108
11	11	4	Lux	4.00	180.00	109
12	12	3	Standart+	3.00	100.00	110
13	13	2	Standart	3.50	90.00	105
14	14	1	Economy	2.50	50.00	107
15	15	1	Economy	2.50	75.00	108
16	16	4	Lux	5.00	210.00	110
17	17	3	Standart+	4.80	140.00	102

Here will be the info about your room, type, rate, price in which hotel it is placed.



Then you will live in this hotel and decided to go to a restaurant and information about your order will be saved(Restaurants,Menu,DishUser tables).

	id	dishid	userid
1	1	1	1501
2	2	2	1517
3	3	3	1503
4	4	4	1506
5	5	5	1513
6	6	6	1508
7	7	7	1514
8	8	8	1515
9	9	9	1511
10	10	10	1516
11	11	11	1512
12	12	12	1502
13	13	13	1507
14	14	14	1508
15	15	15	1501
16	16	16	1515
17	17	4	1507
18	18	12	1502

This table will contain your ID and DishID.

	dishid	name	description	price	restaurantid
1	1	Steak	A steak is a meat generally sliced across the mus...	50.00	1
2	2	Omelet	An omelette or omelet is a dish made from beaten ...	5.00	2
3	3	Caesar salad	A Caesar salad is a green salad of romaine lettuc...	15.00	3
4	4	French fries	French fries or simply fries or chips, are pieces...	7.00	4
5	5	Okroshka	The classic soup is a mix of mostly raw vegetable...	9.00	5
6	6	Lasagna	Lasagne, or the singular lasagna, is an Italian d...	20.00	6
7	7	Maqluba	It consists of meat, rice, and fried vegetables p...	40.00	7
8	8	Pizza	Pizza, dish of Italian origin consisting of many ...	18.00	8
9	9	Shrimp	Shrimp with the creamy garlic sauce	17.00	9
10	10	Nuggets	A chicken product made from chicken meat that is ...	5.00	10
11	11	Sushi	Sushi is a popular Japanese dish made from season...	15.00	10
12	12	Potatoes with mushrooms	Potatoes with mushrooms in a creamy sauce	25.00	2
13	13	Ice cream with nuts	Ice cream is a sweetened frozen food with nuts.	10.00	5
14	14	Kimchi with rice	a spicy pickled or fermented mixture containing c...	24.00	6
15	15	Dolma	A dish of tomatoes, green peppers, vine leaves, o...	20.00	1
16	16	Pilaf	A steamed rice dish often with meat, shellfish, o...	30.00	8

Here you will find the information about price and description depended on your dishID.  
RestaurantID foreign key will give you information about the restaurant

	restaurantid	floor	number_of_tables	hotelid	name
1		1 1st floor	25	101	Dinner in the Sky
2		2 2nd floor	30	102	The Lockhart
3		3 1st floor	40	103	El Diablo "The Devil"
4		4 2nd floor	35	104	Eternity
5		5 3rd floor	20	105	O.NOIR
6		6 1st floor	25	106	Aurum
7		7 2nd floor	40	107	Kinderkookkafe
8		8 3rd floor	15	108	Steam Plant
9		9 4th floor	25	109	Chill Out
10		10 1st floor	15	110	Villa Escudero

For example in which floor this restaurant is placed, amount of tables and name of the restaurant.

You decided to use some service, the information will be stored(AdditionalServices and UserService tables).

	id	serviceid	userid
1	1	901	1501
2	2	902	1502
3	3	903	1503
4	4	904	1505
5	5	905	1507
6	6	906	1508
7	7	907	1509
8	8	908	1510
9	9	909	1501
10	10	910	1512
11	11	910	1516
12	12	909	1501
13	13	908	1515
14	14	907	1514
15	15	906	1508
16	16	905	1513
17	17	904	1505
18	18	903	1503
19	19	902	1517
20	20	901	1504

Your ID and ServiceID will be recorded in this table.

	serviceid	name	description	price	age_limit
1	901	All-Day Breakfast and Brunch	breakfast available throughout the entire day	\$15	for all
2	902	Pick-Up/Drop-Off Service	Creating a regular schedule for guests	\$9	for all
3	903	Caring	free bottle of water, small pack of mints, or puzz...	free	for all
4	904	Rental Clothing	offer a temporary closet to guests	\$11	from the age
5	905	Morning call	If you want to be woken, ask our staff to call yo...	\$2	for all
6	906	FLOWERS	to order flowers to the room	\$15	for all
7	907	Extra bed	one extra bed(a cot) could be added to the rooms	\$4	for all
8	908	Mini Bar	mini bar with a big collection of alcoholic and ...	with mini bar table	from the age
9	909	Car rental	to rent a car	starts from \$70	from the age
10	910	Luggage storage	to storage baggage	starts from \$10	from the age

The information about service can be found in the additional services table by serviceID foreign key.

At the end of your living time, all this information, prices will be added and you will be given a total invoiced bill.

The queries that can be done with this database:

1. --Show number of "Standart+" rooms in every hotel

**SELECT** Hotel.name,

```

COUNT(Room.*) AS "Standart+ amount"

FROM Hotel

JOIN Room ON Hotel.HotelID = Room.HotelID

WHERE Room.type = 'Standart+'

GROUP BY Hotel.name

ORDER BY "Standart+ amount";

```

Firstly we join Hotel with Room table, then save the amount of Standart+ rooms and make them in ascending order by default.

```

--Show average bill for each restaurant

SELECT Restaurants.name,

      AVG(Menu.price) AS "Average bill"

FROM Restaurants

      JOIN Menu ON Restaurants.RestaurantID = Menu.RestaurantID

GROUP BY Restaurants.name

ORDER BY "Average bill" DESC, Restaurants.name;

```

For example we can show the average price of every restaurant. first we make inner join restaurant and menu tables, then we calculate average for every restaurant and show it in descending order.

```

Show additional services that are available for all in every hotel--

SELECT Hotel.name, AdditionalServices.name, AdditionalServices.description

FROM hotel

      INNER JOIN hotel_service on Hotel.HotelID = hotel_service.HotelID

      INNER JOIN AdditionalServices on hotel_service.ServiceID = AdditionalServices.ServiceID

WHERE AdditionalServices.age_limitations = 'for all';

```

So this one is more complex because we join 3 tables, hotel\_service with hotel and hotel\_service with additional service and show, then we find the age\_limitations for all and display the information on the screen.

Also by using this database we can track employees working experience, who is the manager of a defined worker. Easily update the salary for several persons.

```

Update employee's salary whose working experience more than 5 years--

```

```
update Employees set salary = salary*1.2
```

```
where Employees.working_experience::integer>5;
```

Working experience is a varchar by itself, but we can convert it with ::integer function

**Conclusion and Future work:** Hotel managing systems are essential to work fast and always correct. We made this database, as a result we think we have a good hotel management system, but nothing is perfect and we have to improve it. What can be done to improve this database? Firstly we need to fix our faults, it's very difficult to get everything right. To improve the database we can add an entity called payment to check, whose room is paid and whose not. Make a boolean satisfied to collect the information is user satisfied in hotel or not. Add detailed information about restaurants, users, for example make a discount to frequent customers. The opportunity to book a room at a hotel online by using the website is also will be a good add. It would be good to add medical offices to the database so that a person can safely know where to turn in case of ill health.

## Reflection

**Dusembay Adilzhan:** I was working on the half of insert commands, half of queries, writing subqueries, update. Together we did the idea of this project, ERD, creating tables. What did I learn from this project? For example that we can convert varchar data type to integer by using ::integer command. I also learnt how to use subqueries. For example, let me show what I do.

```
--Show average bill for each restaurant
```

```
SELECT Restaurants.name,
```

```
AVG(Menu.price) AS "Average bill"
```

```
FROM Restaurants
```

```
JOIN Menu ON Restaurants.RestaurantID = Menu.RestaurantID
```

```
GROUP BY Restaurants.name
```

```
ORDER BY "Average bill" DESC, Restaurants.name;
```

I learnt that if we don't write type of join it automatically choose inner join.

```
--Add 5 days to reservation for hotels that have less than 4 stars--
```

```
UPDATE reservation
```

```
SET number_of_day = number_of_day + 5
```

```
WHERE hotelid NOT IN (
```

```
SELECT hotelid
```

```
FROM hotel
```

```
WHERE rate > 4)
```

This update function will increase amount of days in hotel in which the rate is not more than 4.

P.S. Teamwork is a really good practice because you can correct each other. You are not able to make decisions by yourself, everything must be done together. Of course sometimes there might be misunderstandings but it is also a good experience to handle with it.

**Meshitbayeva Arailym:** I was working on the half of insert commands, half of queries, writing subqueries, delete. Together we did the idea of this project, ERD, creating tables. What did I learn from this project? For example to work with multivalued dimensions it is easy to create a bridge table. For example, let me show what I did:

--Show information about worker whose working experience more than 5 years--

```
select
```

```
JobDescription.name,Employees.name,Employees.surname,JobDescription.salary_range,Job  
Description.skills,Employees.working_experience
```

```
from Employees
```

```
inner join JobDescription
```

```
on Employees.JobID = JobDescription.JobID
```

```
where Employees.working_experience::integer>5;
```

In this code I learnt that we can convert varchar data type to integer by using ::integer command.

P.S. I liked working as a team because this way we could understand each other's ideas and work on the project's shortcomings.

## Literature

To create ER diagram used creately.com

Information about Hotel history taken from: wikipedia

<https://en.wikipedia.org/wiki/Hotel>

## Link to github

<https://github.com/Khas3r0nd/ICTFinal>