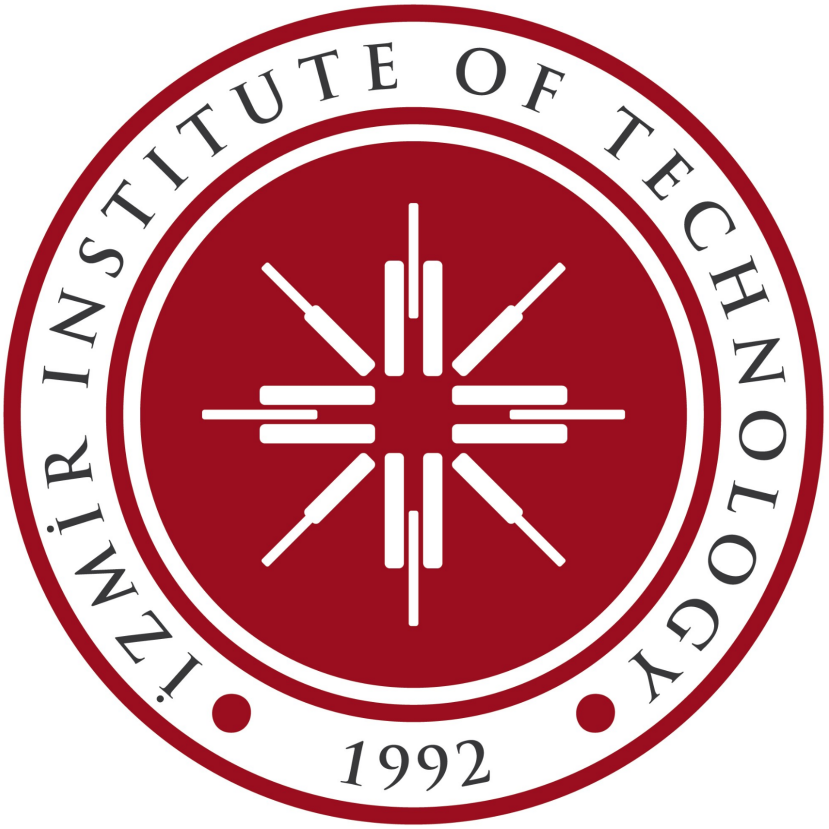


IZMIR INSTITUTE OF TECHNOLOGY

Department of Computer Engineering

Information Management

Hotel Management Database

By

Onur Şahinler
Emir Kaldırımcı
Ramazan Çuhacı
Bora Ersoy
Hasan Çoban

December-2023

IZMIR

We will design a Hotel Management Database project together with our group. This hotel will be a small boutique hotel for our couples to have a romantic time. One of the biggest problems of our married couples with children is that they cannot spend time alone. Our hotel is open for service to give these couples the time they miss. Even if the beds in our rooms are for two, our single guests can also stay in our hotel. Our guests will be able to benefit from two meals: breakfast and dinner. Wish you a beautiful and peaceful holiday…

Assumptions and Business Rules:

-A room can only be booked by our guests if it is available.

-Two people who are not a couple will not be able to use our hotel.

-Our couples who are guests do not have children.

-The consumable items ordered by customers will never be zero, and the stocks will be renewed.

-Each room will have one responsible staff member.

-In our restaurant, our waiters give each of our guests a menu with the food and beverage available in our kitchen.

-Depending on the room number, a table is reserved for our guests in our restaurant.

-The housekeeping employee replenishes all the supplies in the room while cleaning the rooms.

-The fees for the reservation and the orders are calculated separately and our customer must pay all of them.

-The fees of our customers are paid on the date of check out from our hotel by himself/herself.

-People who do not stay at our hotel will not be able to use our restaurant.

-The price of a room in our hotel for one night is 400 TL.

-Not every guest has to benefit from our restaurant.

-A new GuestID will not be created for our guests who have used our hotel in previous periods.

o Entity sets

Guest = (GuestID, FirstName, LastName, Gender, DateOfBirth, {Email}, {Phone})

Room = (RoomNumber, Availability, isClear)

Reservation = (ReservationID, CheckInDate, CheckOutDate, ReservationCancellation, BookingStatus, NumberOfPeople)

Staff = (StaffID, FirstName, LastName, Position, {Phone})

Payment = (PaymentID, BillingAmount, Date, PaymentType)

Supply = (SupplyID, Type, Quantity)

Order = (OrderID, TableNumber, TotalPrice)

ConsumableItem = (ItemID, ItemUnitPrice, ItemName, ItemAmount, ItemType)

o Users of the System

Guests: Makes room reservations, pays the bill, and payments and makes food reservations.
Staff: Updates room status, manages reservations, keeps track of supplies and makes food.

o Relationship sets

Booking → *Guest-Reservation*
- *Specifies that our guests make a booking through the system. Each guest can make more than one reservation, but each reservation can belong to one person.*

Check-in → *Reservation-Room*
- *Specifies that the reservation made depends on a room. Each room may have more than one reservation at different times, and it can also be said that each reservation includes different rooms.*

Pays → *Guest-Payment*
- *Specifies the fee that our guest should pay for the service s/he got. Depending on the type of our guests, there could be multiple payments, but each payment belongs to only one of our guests.*

Contains → *Room-Supply*
- *Specifies the supplies contained in our rooms. While there are many supplies that our guests can use in our rooms, these supplies are available in all our rooms.*

Works → *Staff-Reservation*
- *Specifies our staff who are working in the reservation business. While we have one staff member responsible for each reservation, more than one of our staff can work in the reservation department.*

Takes → *Staff-Order*
- *Specifies that our staff will take the order given by our guest. Even if a single staff takes the orders given by our guests, our staff can take multiple orders.*

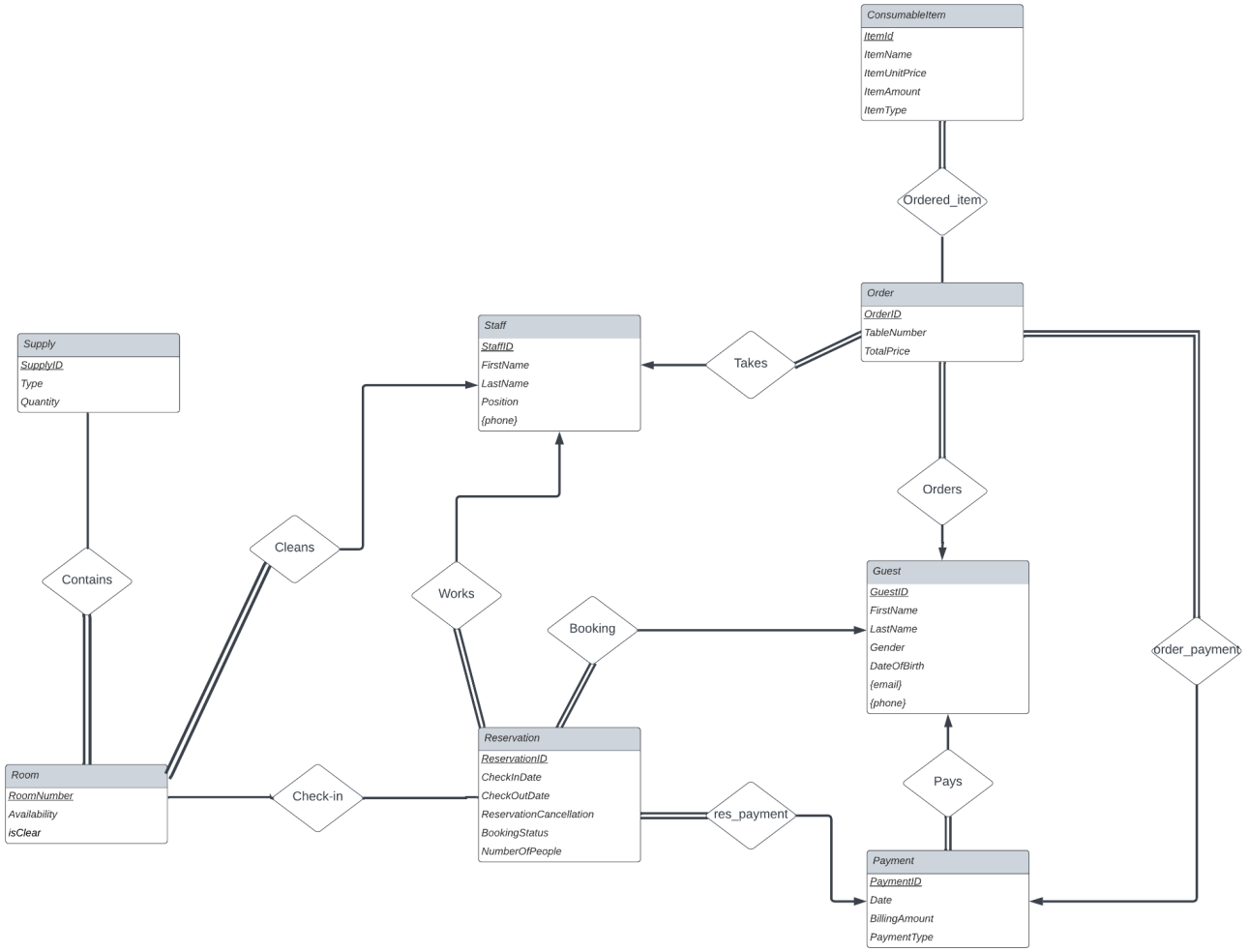
Res_payment → *Reservation-Payment*
- *The relationship that must be paid for the reservation fee. Each reservation must have one payment, and multiple reservation payments can be made using a single payment method.*

Cleans → *Room-Staff*
- *Specifies the relationship of the staff to clean the rooms for the next guests. While there is one staff member who cleans the rooms, one staff member can clean more than one room.*

Orders → *Guest-Order*
- *Specifies that our guests order the food and beverage they want to eat and drink. Our guests can place more than one order, but each order belongs to a customer.*

Ordered_item → *Order-ConsumableItem*
- *Specifies the ordered item by the guest. The consumer items ordered by our customers may be more than one, and the consumer items may belong to more than one order.*

Order_payment → *Order-Payment*
- *Specifies the relationship in which our guests must pay for food and beverage. Although each order should have a payment made by our guests, each payment can cover multiple orders.*



We would like to make explanations about our “Hotel Management Database” project. Firstly, there are guests and staffs as the main users in this system. Guests make reservations based on availability, while some of the staffs provide reservation management. If our guests make a reservation for an available room, that room will not be able to be booked by another guest between the booking dates. Some staff members track the supplies found in the system. Some staff members are responsible for preparing the meals of our guests. Some staff members are responsible for taking orders from our customers.

In addition to the food and beverages ordered by our guests at our restaurant, items that can be used such as towels and slippers are also included in the consumable item. Payments will be created in the reservation and dining sets. Our guests pay a payment to us in exchange for all the services they receive. There are various payments included in this, for example, room fee, invoice fee, restaurant bills, extras, etc.

o Relational Schema

Guest (GuestID, FirstName, LastName, Gender, DateOfBirth)
Guest_phone (GuestID, Phone)
Guest_email (GuestID, Email)
Staff (StaffID, FirstName, LastName, Position)
Staff_phone (StaffID, Phone)
Order (OrderID, TableNumber, TotalPrice, PaymentID, GuestID, StaffID)

- PaymentID referencing the *Payment* relation.
- GuestID referencing the *Guest* relation.
- StaffID referencing the *Staff* relation.

ConsumableItem (ItemID, ItemName, ItemUnitPrice, ItemAmount, ItemType, OrderID)

- OrderID referencing the *Order* relation.

Room (RoomNumber, Availability, isClear,StaffID)

- StaffID referencing the *Staff* relation.

Reservation (ReservationID, CheckInDate, CheckOutDate, ReservationCancellation, BookingStatus, NumberOfPeople, StaffID, GuestID, PaymentID, RoomNumber)

- StaffID referencing the *Staff* relation.
- GuestID referencing the *Guest* relation.
- PaymentID referencing the *Payment* relation.
- RoomNumber referencing the Room relation.

Payment (PaymentID, Date, BillingAmount, PaymentType, GuestID)

- GuestID referencing the *Guest* relation.

Supply (SupplyID, Type, Quantity, RoomNumber)

- RoomNumber referencing the Room relation.

Contains (*SupplyID, RoomNumber*)

- *SupplyID* referencing the *Supply* relation.
- *RoomNumber* referencing the *Room* relation.

Check-in (*RoomNumber, ReservationID*)

- *RoomNumber* referencing the *Room* relation.
- *ReservationID* referencing the *Reservation* relation.

o Screenshots of Database

Script Result

Script: hotel_final Status: Complete

View: Detail Summary Rows: 15 Go Create App Edit Script

Number	Elapsed	Statement	Feedback	Rows
1	0.86	DROP TABLE Guest CASCADE CONSTRAINTS	Table dropped.	0
2	0.84	DROP TABLE Guest_Email CASCADE CONSTRAINTS	Table dropped.	0
3	0.77	DROP TABLE Guest_Phone CASCADE CONSTRAINTS	Table dropped.	0
4	0.85	DROP TABLE Room CASCADE CONSTRAINTS	Table dropped.	0
5	0.81	DROP TABLE Reservation CASCADE CONSTRAINTS	Table dropped.	0
6	0.84	DROP TABLE Payment CASCADE CONSTRAINTS	Table dropped.	0
7	0.82	DROP TABLE Staff CASCADE CONSTRAINTS	Table dropped.	0
8	0.84	DROP TABLE Staff_Phone CASCADE CONSTRAINTS	Table dropped.	0
9	0.80	DROP TABLE Supply CASCADE CONSTRAINTS	Table dropped.	0
10	0.87	DROP TABLE Orders CASCADE CONSTRAINTS	Table dropped.	0
11	0.79	DROP TABLE Consumable CASCADE CONSTRAINTS	Table dropped.	0
12	0.02	alter session set nls_language='ENGLISH'	Statement processed.	0
13	0.01	alter session set nls_date_format='DD-MON-YYYY'	Statement processed.	0
14	0.07	CREATE TABLE Guest (guest_id INT NOT NULL PRIMARY KEY,	Table created.	0
15	0.02	CREATE TABLE Guest_Phone (guest_id INT NOT NULL, ph	Table created.	0

Download

row(s) 1 - 15 of 683 Next ▸

683	683	0
Statements Processed	Successful	With Errors

Guest

```
1  select * from guest
```

Results | Explain | Describe | Saved SQL | History

GUEST_ID	FIRST_NAME	LAST_NAME	GENDER	DATE_OF_BIRTH
100001	Hasan	Coban	Male	05/25/1985
100002	Bora	Ersoy	Male	04/15/1985
100003	Ramazan	Cuhaci	Male	05/15/1955
100004	Onur	Sahinler	Male	05/25/1985
100005	Emir	Kaldirimci	Male	06/15/1985
100006	Sebnem	Ferah	Male	05/15/1985
100007	Duke	Ellington	Male	08/22/1990
100008	LeBron	James	Male	12/10/1978
100009	Sarah	Brown	Female	04/03/1982
100010	Jones	Breckie	Male	05/30/2005
100011	Jessica	Miller	Female	09/11/1989
100012	Brian	Moore	Male	02/09/1974
100013	Megan	Taylor	Female	07/14/1987

Guest_Phone

```
1  select * from guest_phone
```

Results | Explain | Describe | Saved SQL | History

GUEST_ID	PHONE_NUMBER
100001	5314234541
100001	5351234567
100002	5329833466
100002	5329876543
100003	5558111322
100003	5558765432
100003	5558767128
100004	5334567890
100005	5362345678
100006	5398765432
100007	5357890141
100007	5457894123
100007	5557891123

Guest_Email

```
1  select * from guest_email
```

Results | Explain | Describe | Saved SQL | History

GUEST_ID	EMAIL
100001	hasan.coban@example.com
100001	hasancoban13579@example.com
100002	bora.ersoy@example.com
100002	boraersoy35@example.com
100003	cuhaciramazan@example.com
100003	ramazan.cuhaci@example.com
100004	onur.sahinler@example.com
100004	onurshnlr@example.com
100005	emir.kaldirimci@example.com
100005	emir_kaldirimci@example.com
100006	john.smith@example.com
100007	emily.johnson@example.com
100007	johnson_emily@example.com

Payment

```
1  select * from payment
```

Results | Explain | Describe | Saved SQL | History

PAYMENT_ID	PAYMENT_DATE	BILLING_AMOUNT	PAYMENT_TYPE	GUEST_ID
209	04/10/2024	2000	Reservation	100001
202	01/22/2024	1600	Reservation	100013
203	02/05/2024	1200	Reservation	100004
204	02/18/2024	2400	Reservation	100005
205	02/28/2024	1200	Reservation	100006
206	03/08/2024	2000	Reservation	100007
207	03/20/2024	2000	Reservation	100018
208	03/25/2024	1200	Reservation	100009
201	01/10/2024	2000	Reservation	100012
210	04/20/2024	2000	Reservation	100015
211	06/18/2024	400	Reservation	100008
212	06/27/2024	2800	Reservation	100010
213	06/29/2024	400	Reservation	100011
214	07/04/2024	1200	Reservation	100002

Results | Explain | Describe | Saved SQL | History

218	05/25/2024	2000	Reservation	100014
219	06/05/2024	1600	Reservation	100020
220	06/15/2024	2000	Reservation	100003
221	04/10/2024	2616	Order	100001
222	07/04/2024	820	Order	100002
223	06/15/2024	849	Order	100003
224	02/05/2024	100	Order	100004
225	02/18/2024	80	Order	100005
226	02/28/2024	100	Order	100006
227	03/08/2024	190	Order	100007
228	06/18/2024	727	Order	100008
229	03/25/2024	240	Order	100009
230	03/25/2024	64	Order	100010
231	06/29/2024	64	Order	100011

Reservation

```
1  select * from reservation
```

Results | Explain | Describe | Saved SQL | History

RESERVATION_ID	CHECK_IN_DATE	CHECK_OUT_DATE	RESERVATION_CANCELLATION	BOOKING_STATUS	NUMBER_OF_PEOPLE	STAFF_ID	GUEST_ID	PAYMENT_ID	ROOM_NUMBER
80001	01/05/2024	01/10/2024	N	B	2	20002	100012	201	101
80002	01/18/2024	01/22/2024	N	P	1	20003	100013	202	201
80003	02/02/2024	02/05/2024	N	B	2	20001	100004	203	202
80004	02/12/2024	02/18/2024	N	P	2	20002	100005	204	203
80005	02/25/2024	02/28/2024	N	B	1	20002	100006	205	401
80006	03/03/2024	03/08/2024	N	P	2	20001	100007	206	402
80007	03/15/2024	03/20/2024	N	B	2	20016	100018	207	403
80008	03/22/2024	03/25/2024	N	P	1	20002	100009	208	301
80009	04/05/2024	04/10/2024	N	B	2	20002	100001	209	302
80010	04/15/2024	04/20/2024	N	P	2	20002	100015	210	501
80011	06/17/2024	06/18/2024	N	P	2	20002	100008	211	501
80012	06/20/2024	06/27/2024	N	P	2	20002	100010	212	501
80013	06/28/2024	06/29/2024	N	P	2	20002	100011	213	501

Room

```
1  select * from room
```

Results | Explain | Describe | Saved SQL | History

ROOM_NUMBER	ROOM_AVAILABILITY	ROOM_IS_CLEAR	STAFF_ID
101	Y	T	20005
102	N	T	20015
103	N	T	20005
201	Y	F	20014
202	N	F	20014
203	Y	T	20014
301	N	F	20015
302	Y	F	20015
303	Y	F	20015
401	Y	T	20005
402	N	T	20014
403	N	F	20015
501	N	F	20014

Staff

```
1  select * from staff
```

Results | Explain | Describe | Saved SQL | History

STAFF_ID	STAFF_FIRST_NAME	STAFF_LAST_NAME	STAFF_POSITION
20001	Xavier	Brooks	Receptionist
20002	Jane	Smith	Receptionist
20003	Robert	Johnson	Receptionist
20004	Emily	Williams	Chef
20005	Michael	Jones	Housekeeping
20006	Sophia	Brown	Bartender
20007	William	Taylor	Maintenance
20008	Olivia	Miller	Waiter
20009	James	Davis	Bartender
20010	Emma	Garcia	Kitchen Chef
20011	Daniel	Rodriguez	Waiter
20012	Ava	Martinez	Waitress
20013	Logan	Hernandez	Bellboy

Staff_Phone

```
1  select * from staff_phone
```

Results | Explain | Describe | Saved SQL | History

STAFF_ID	STAFF_PHONE
20002	5123222223
20002	5123456789
20003	5234567890
20003	5634367892
20004	5345678901
20005	5456789012
20006	5567890123
20007	5678901234
20008	5389012344
20008	5489012341
20008	5789012345
20009	5890123456
20010	5901234567

Supply

```
1  select * from supply
```

Results | Explain | Describe | Saved SQL | History

SUPPLY_ID	SUPPLY_TYPE	QUANTITY	ROOM_NUMBER
300001	Pen	3	101
300001	Pen	3	102
300001	Pen	3	103
300001	Pen	3	201
300001	Pen	3	202
300001	Pen	3	203
300001	Pen	3	301
300001	Pen	3	302
300001	Pen	3	303
300001	Pen	3	401
300001	Pen	3	402
300001	Pen	3	403
300001	Pen	3	501

Orders

```
1  select * from orders
```

Results | Explain | Describe | Saved SQL | History

ORDER_ID	TABLE_NUMBER	TOTAL_PRICE	PAYMENT_ID	GUEST_ID	STAFF_ID
101	1	1140	221	100001	20004
102	1	509	221	100001	20004
103	1	325	221	100001	20004
104	1	642	221	100001	20020
105	1	360	222	100002	20006
106	1	340	222	100002	20006
107	1	120	222	100002	20020
108	1	144	223	100003	20020
109	1	480	223	100003	20020
110	1	225	223	100003	20020
111	2	100	224	100004	20004
112	2	80	225	100005	20004
113	2	100	226	100006	20020

ConsumableItem

```
1  select * from consumable
```

Results | Explain | Describe | Saved SQL | History

ITEM_ID	ITEM_NAME	ITEM_PRICE	ITEM_AMOUNT	ITEM_TYPE	ORDER_ID
1	Grilled Salmon	19	20	Main Course	101
2	Chicken Parmesan	16	15	Main Course	102
2	Chicken Parmesan	16	15	Main Course	118
2	Chicken Parmesan	16	15	Main Course	117
3	Vegetarian Pasta	13	25	Main Course	103
3	Vegetarian Pasta	13	25	Main Course	116
4	Caesar Salad	9	18	Appetizer	104
4	Caesar Salad	9	18	Appetizer	115
5	Red Wine Bottle	20	12	Beverage	105
6	White Wine Bottle	19	10	Beverage	106
6	White Wine Bottle	19	10	Beverage	114
7	Garlic Bread	4	30	Bread	107
8	Chocolate Cake	8	8	Dessert	108