RELATIONAL DATABASES

Assignment 1: Database Design

Student: S244679

Word count: 1095

Contents

1 Introduction

This project will design a relational database to replace a paper-based system used by a hotel. The Hotel Database will hold information about the rooms, reservations, guests, staff, and other information required to effectively manage and promote the hotel.

A relational database is a computer-based system that organises data into tables, consisting of columns and rows. This structure allows for efficient data manipulation and retrieval. It maintains data integrity by applying constraints. (Codd, 1970)

2 Key Requirements

Receptionists need the database to:

- provide room availability
- support guest reservation/check-in/check-out
- log complaints
- support a guest/company relationship

Management requirements:

- report on reservation history and promotion usage
- find patterns in complaints and room preferences
- generate a marketing list
- maintain a room cleaning rota

The main entities required to fulfil the requirements were identified – *see Table 1*.

Entity	Purpose
room	Tracks the rooms, their types, features and pricing
guest	Holds guest details
company account	Links corporate guests to their company accounts for billing
reservation	Manages room bookings, including check-in and check-out information and can be used to gather room booking history
complaint	Records guest complaints for management to address
invoice	Holds the room charges for a reservation and amount paid
staff	Holds the employees' names, roles and link to their manager
promotion	Stores promotional discounts that can be applied to reservations
marketing	Holds information about what type of information a guest would like to receive and by what method
room clean	Holds the cleaning rota for the rooms. The date, time, type of clean, who cleans it and which master key they used

Table 1 *– Main Entities of Hotel Management Database*

3 Conceptual design

An Entity Relationship Diagram (ERD) was created using Visual Paradigm.

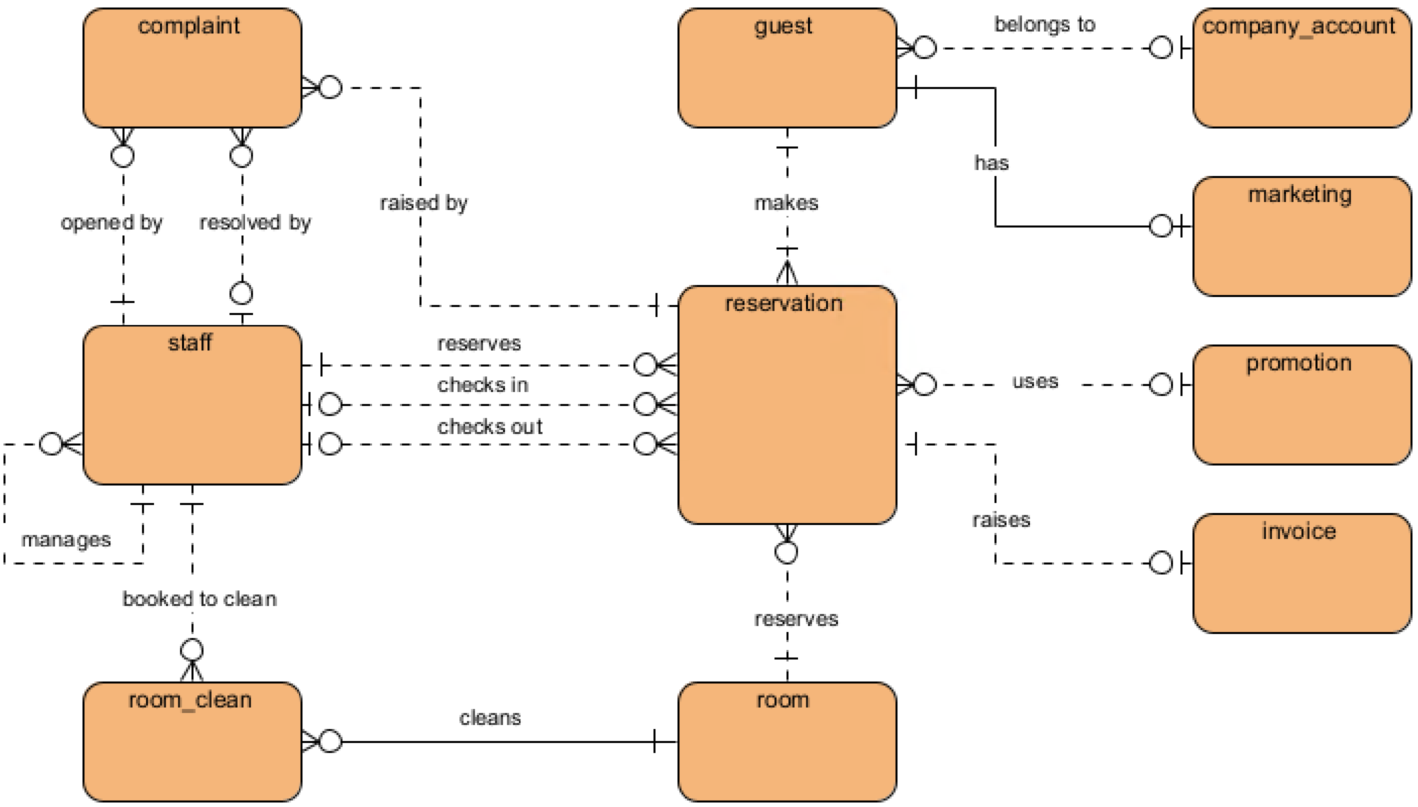


Figure 1 *– Conceptual ERD of the Hotel Database*

See Appendix A for an analysis of relationships of the conceptual design

4 Logical design

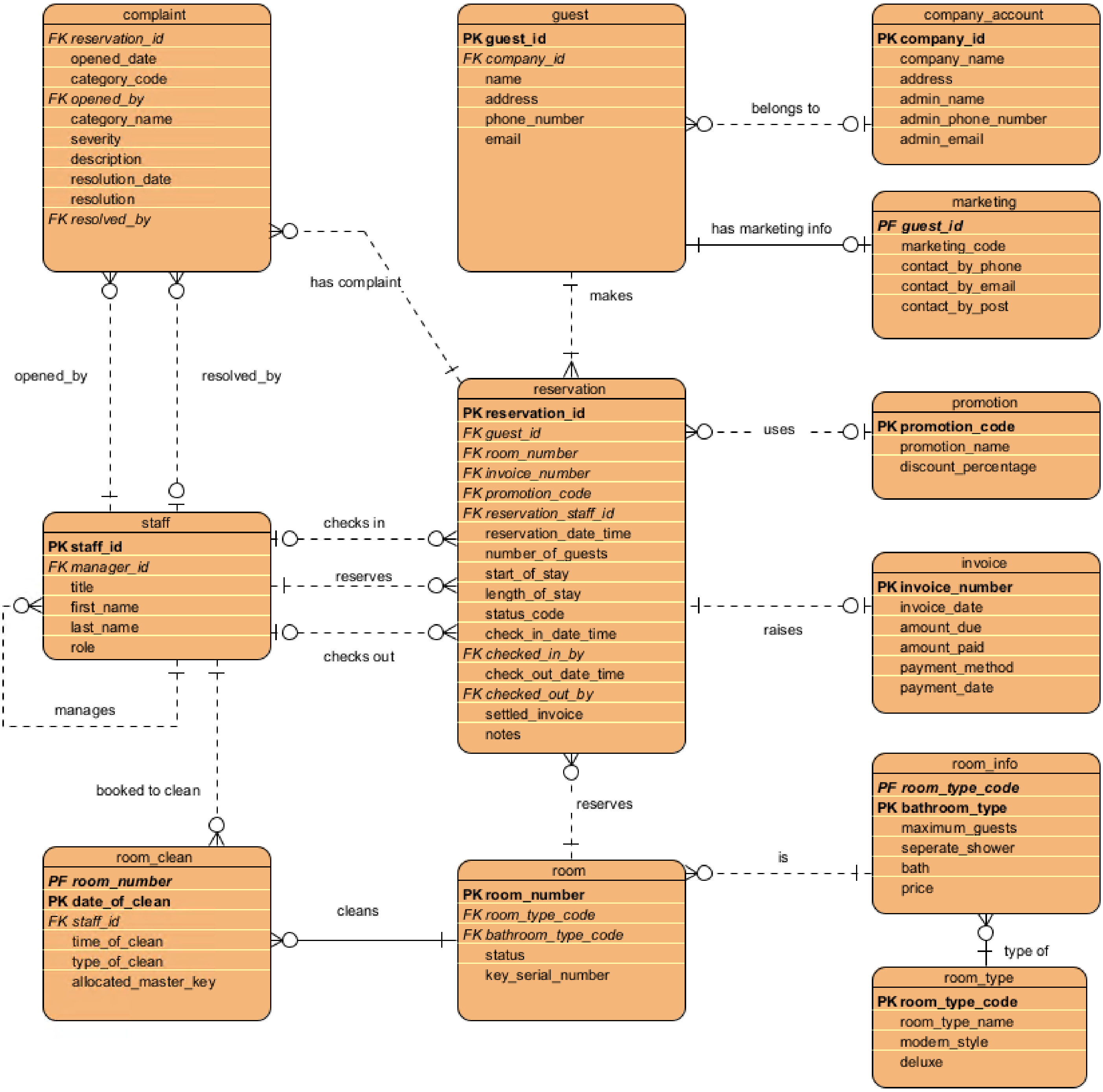


***Figure 2** – Logical Design before Normalisation*

5 Normalisation

Normalisation aims to ensure database efficiency and organisation. It progressively improves the design by eliminating redundancy and reducing risk of insert/update/deletion anomalies. A good design will reduce NULLs in the tuples which improves data integrity. (Date, 2012, Harrington, 2016)

5.1 First Normal Form (1NF)

To comply with 1NF, all tables must only contain atomic values and have a primary key. (Elmasri & Navathe, 2015, Coronel & Morris, 2019)

The design was normalised to meet 1NF standards as detailed in *Table 2*

Tables changed	Reason	Figure
guest, company_account	name attribute not atomic. Split into: title, first_name, last_name attributes.	3
guest, company_account	address attribute not atomic. Split into: address_line1, address_line2, city, county, postcode attributes.	3
complaint	No primary key. The reservation_id and the date/time the complaint is opened are a suitable composite primary key to uniquely identify the row.	4
complaint	Resolution information will be captured at a different time so caused the need for excess NULLs. The complaint_resolution table was created to solve this.	4
staff	name attribute not atomic. Split into: title, first_name, last_name attributes.	5
reservation	Check-in and Check-out will be captured at different times requiring NULLs in the reservation table and risking data integrity issues. New check-in and check-out tables were created to solve this.	5

Table 2 *– Design changes required to meet 1NF*

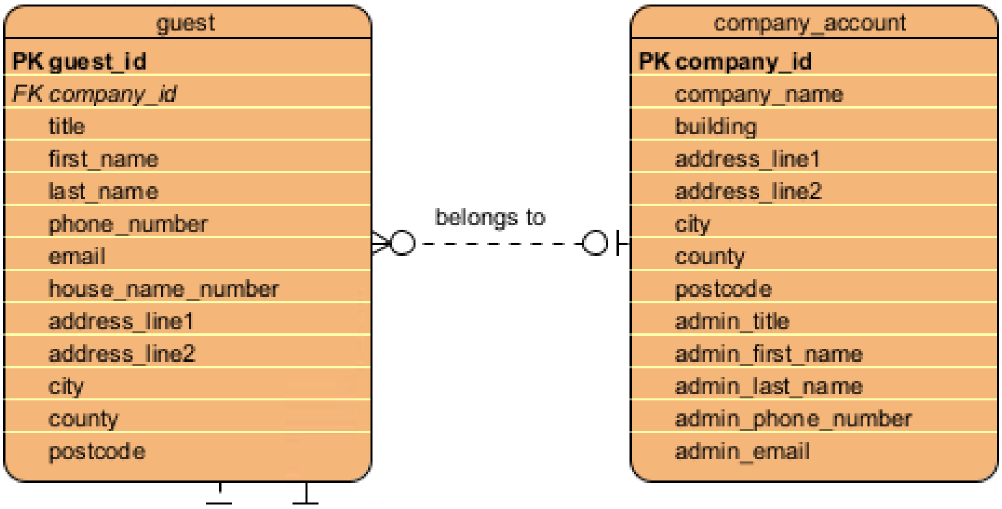


Figure 3 *– Changes to the* ***guest*** *and* ***company_account*** *tables*

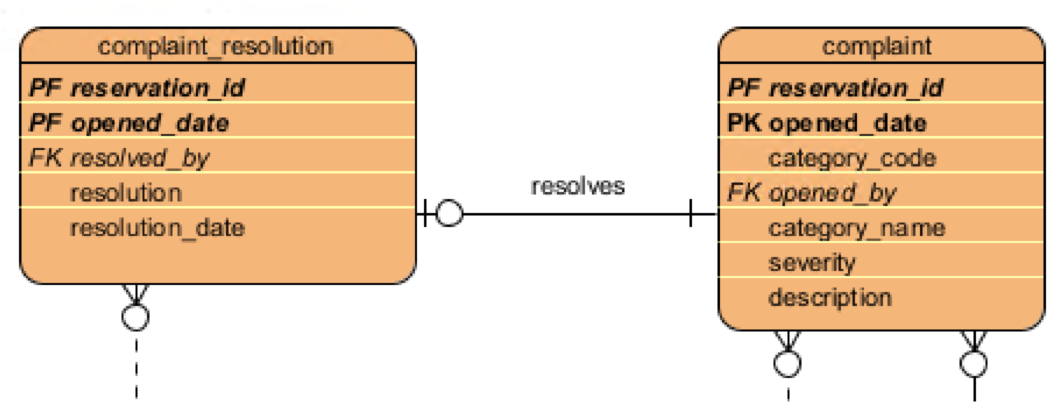


***Figure 4** – Changes to the **complaint** table*

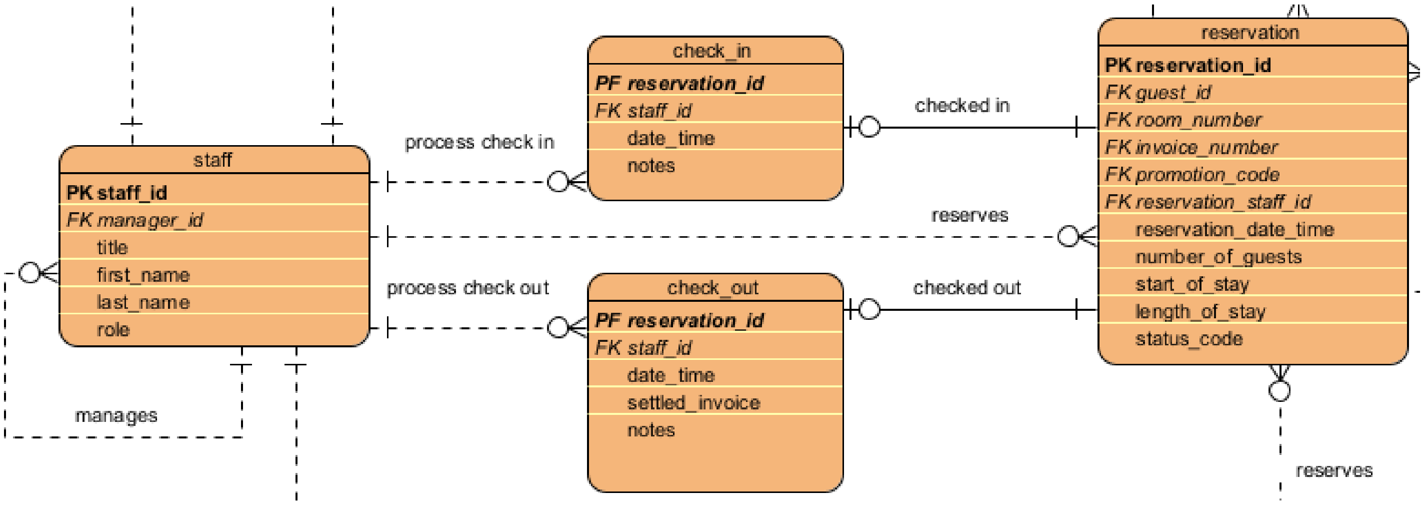


***Figure 5** – Changes to the **staff** and **reservation** tables*

5.2 Second Normal Form (2NF)

2NF focuses on eliminating partial dependencies, these occur when non-prime attributes depend on only part of a composite key. 2NF prevents modification anomalies and improves data integrity, making the design more robust and scalable. (Connolly & Begg, 2014).
The hotel database had three attributes that were dependant on part of a key which failed 2NF – *see Table 3.*

Tables changed	Reason	Figure
room_info	maximum_guests is only dependent on room_type_code and not the bathroom type. The attribute was moved to room_type table to meet 2NF.	6
room_info	separate_shower and bath attributes were moved into a new bathroom_type table. This separation ensured that non-prime attributes only depended on the entire primary key of the table.	6
room_info	Table was renamed as room_price now that its purpose is more defined.	6

***Table 3** – Design changes required to meet 2NF*

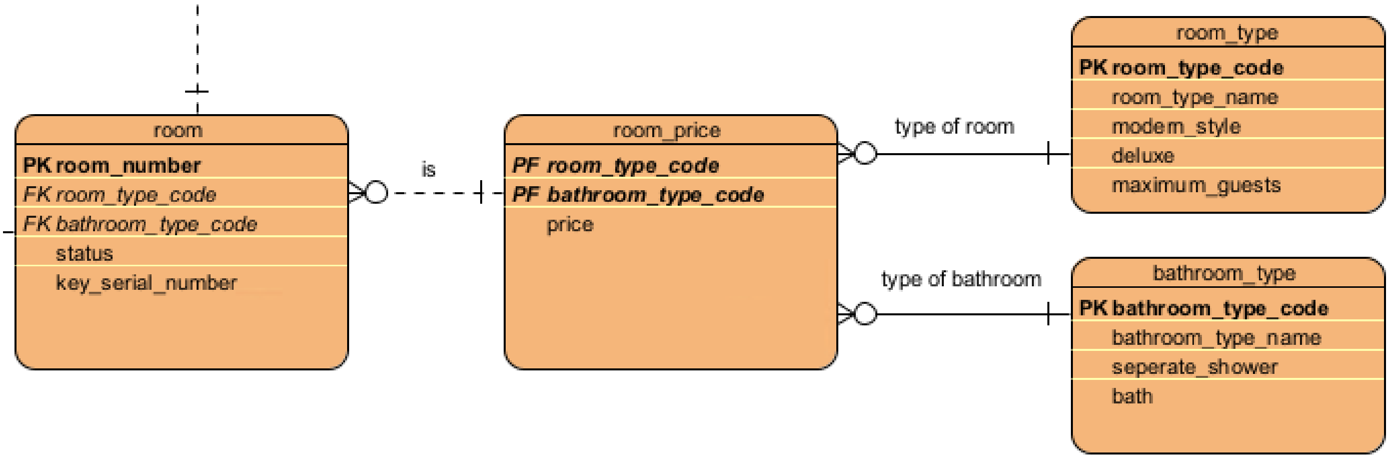


***Figure 6** – Changes to the room information tables to comply with 2NF*

5.3 Third Normal Form (3NF)

3NF aims to eliminate transitive dependencies, they occur when a non-prime attribute is indirectly dependent on the primary key via another non-prime attribute (Silberschatz, Korth, & Sudarshan, 2011).

In the hotel database, transitive dependencies were present in tables containing address information. The street, city and county being dependent on postcode, which was itself dependent on the primary key. The design was changed to prevent redundancy issues and ensure consistent data updates (Garcia-Molina, 2008) – *see Table 4*.

Tables changed	Reason	Figure
complaint	category_name and severity attributes were dependent on category_code rather than the primary key of the complaint table. The attributes were moved to a new complaint_category table and linked via a foreign key.	7
guest & company_account	Address attributes had transitive dependency via postcode. A new address table was created with postcode as the primary key. Tables using an address were updated to use postcode as a foreign key.	8

***Table 4** – Design changes required to meet 3NF*

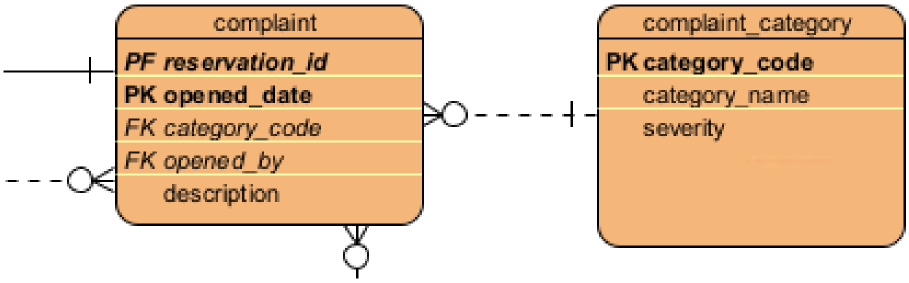


***Figure 7** – Changes to the **complaint** table*

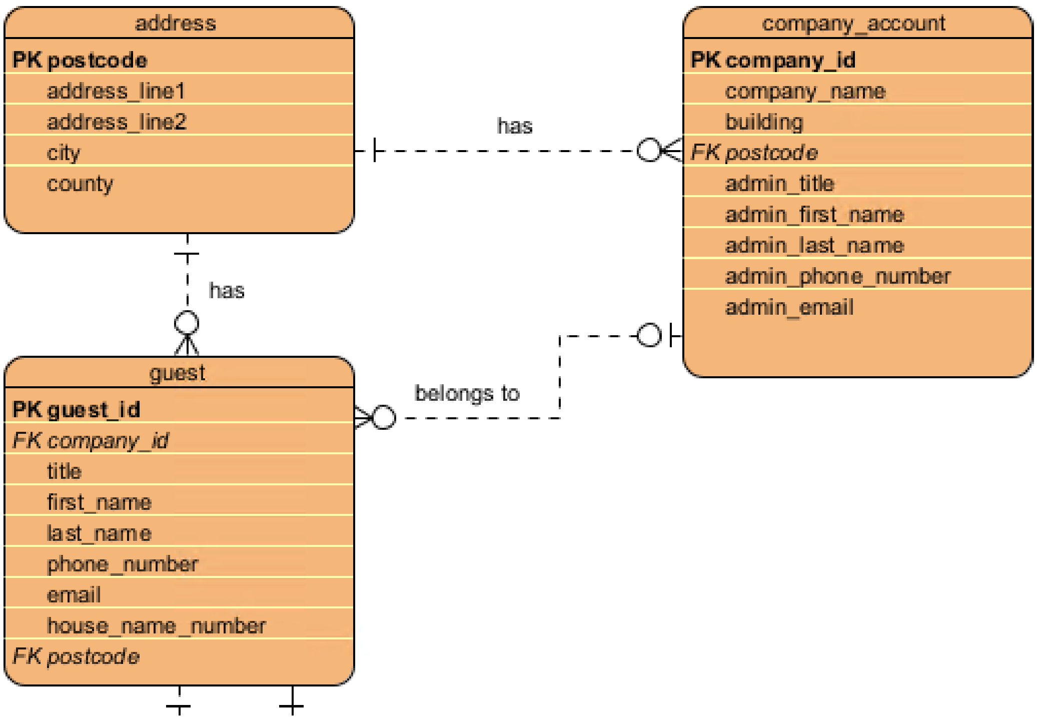


***Figure 8** – Changes to the **guest** and **company_account** tables*

5.4 Boyce-Codd Normal Form (BCNF)

BCNF is an extension of 3NF that addresses more complex forms of functional dependencies. A violation of BCNF occurs when a non-prime attribute is functionally dependent on only **part** of a candidate key rather than an **entire** candidate key. (Wikipedia, 2023. Boyce–Codd normal form)

The *room_clean* table complied with 1NF,2NF & 3NF but violated BCNF (*see Appendix B*)

Tables changed	Reason	Figure
room_clean	allocated_master_key was not dependent on an entire candidate key so was moved into a new cleaning_session table with date_of_clean and staff_id as the primary key	9

***Table 5** – Design changes required to meet BCNF*

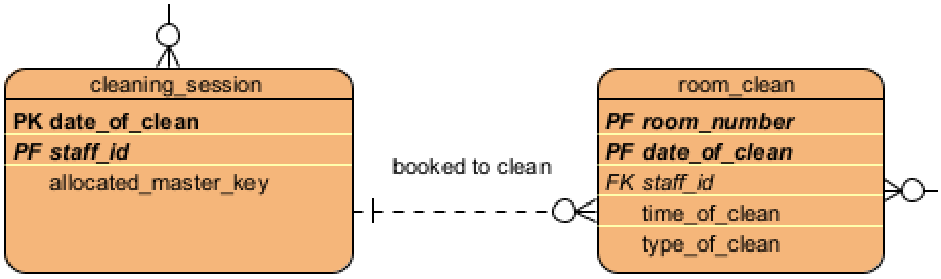


***Figure 9** – Changes to the **room_clean** table*

6 Normalised Logical Design

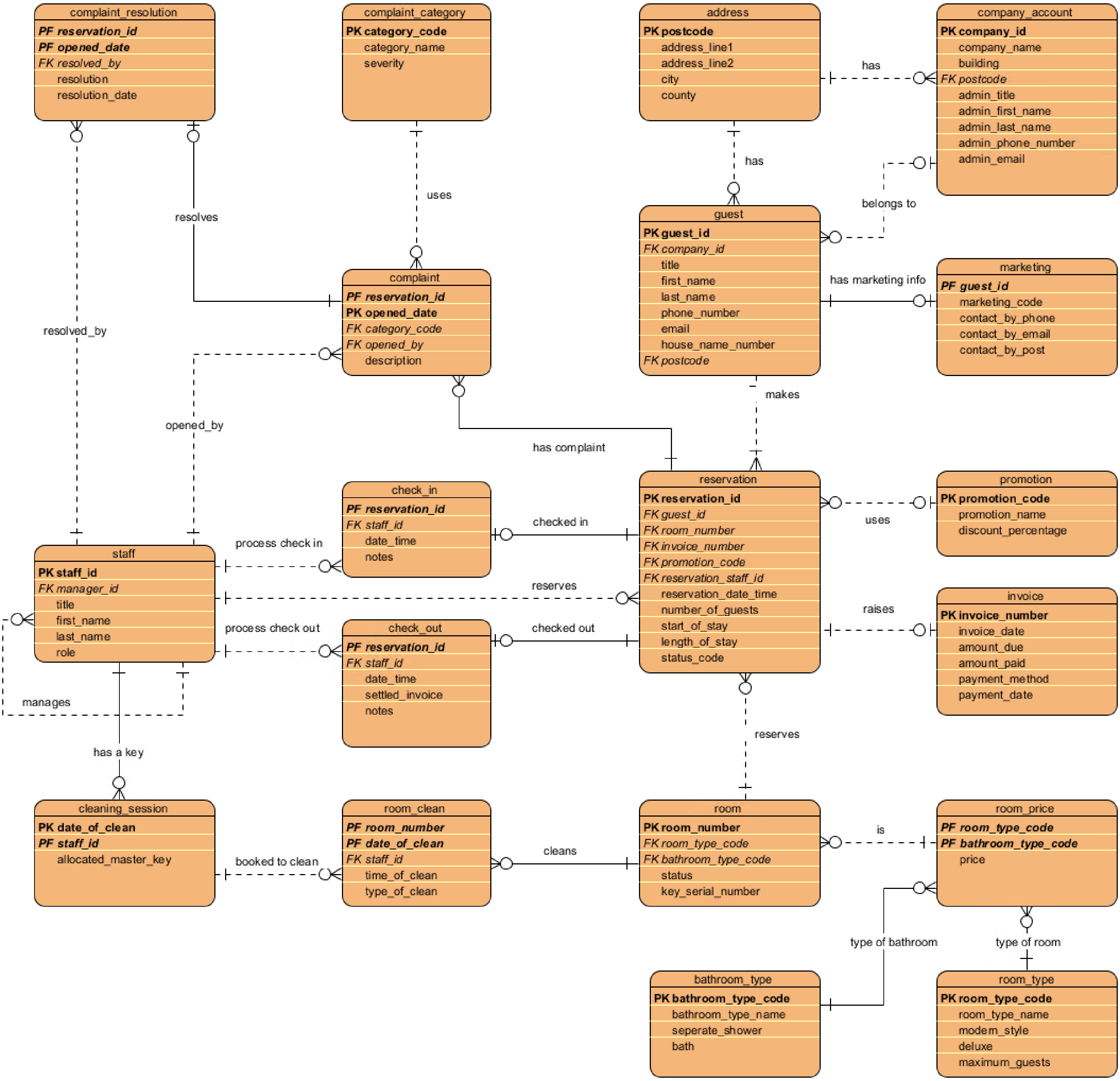
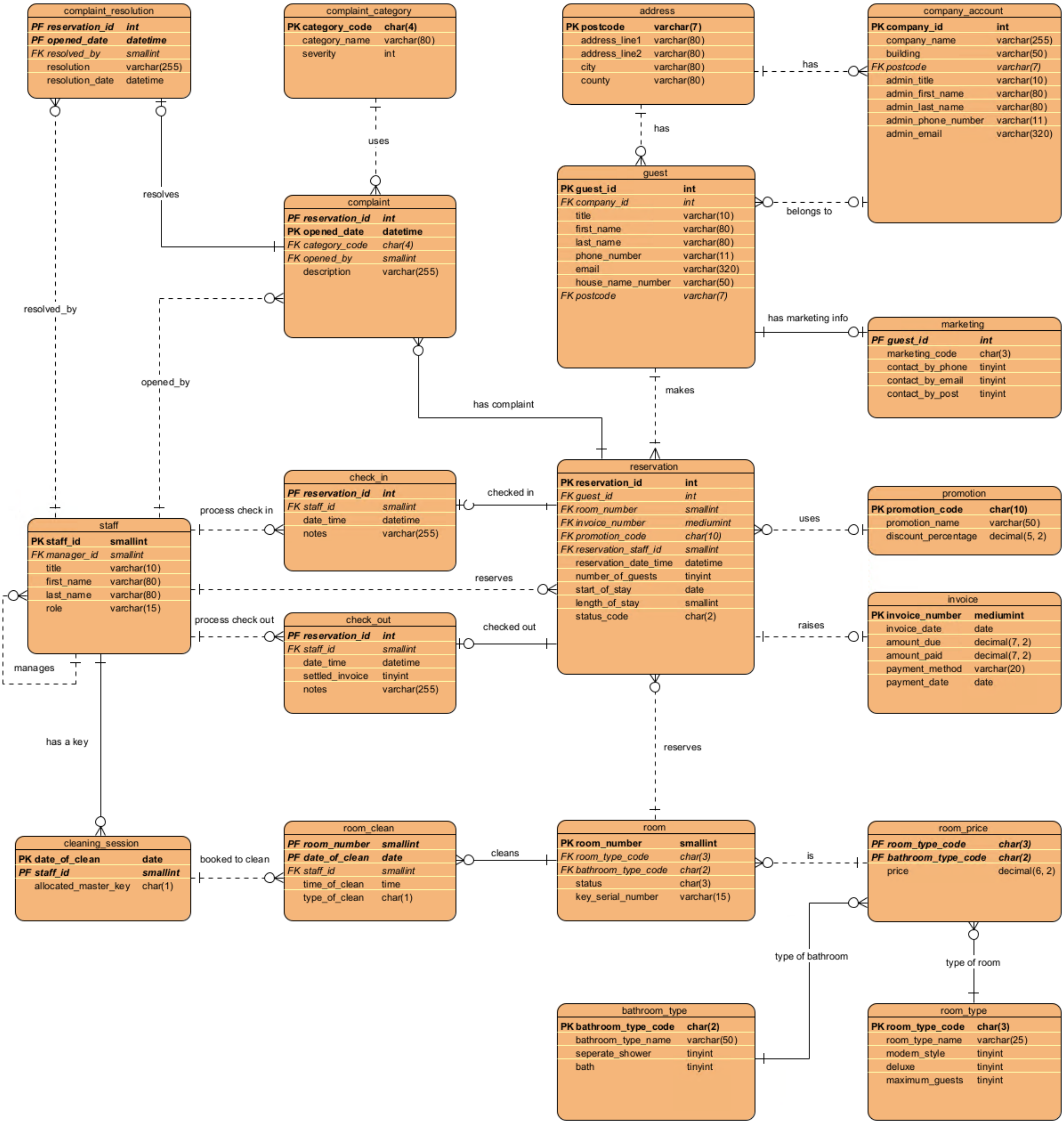


***Figure 10** – Logical Design after Normalisation*

7 Physical Design



complaint_resolution	
PF reservation_id	int
PF opened_date	datetime
FK resolved_by	smallint
resolution	varchar(255)
resolution_date	datetime

complaint_category	
PK category_code	char(4)
category_name	varchar(80)
severity	int

address	
PK postcode	varchar(7)
address_line1	varchar(80)
address_line2	varchar(80)
city	varchar(80)
county	varchar(80)

company_account	
PK company_id	int
company_name	varchar(255)
building	varchar(50)
FK postcode	varchar(7)
admin_title	varchar(10)
admin_first_name	varchar(80)
admin_last_name	varchar(80)
admin_phone_number	varchar(11)
admin_email	varchar(320)

complaint	
PF reservation_id	int
PK opened_date	datetime
FK category_code	char(4)
FK opened_by	smallint
description	varchar(255)

guest	
PK guest_id	int
FK company_id	int
title	varchar(10)
first_name	varchar(80)
last_name	varchar(80)
phone_number	varchar(11)
email	varchar(320)
house_name_number	varchar(50)
FK postcode	varchar(7)

marketing	
PF guest_id	int
marketing_code	char(3)
contact_by_phone	tinyint
contact_by_email	tinyint
contact_by_post	tinyint

check_in	
PF reservation_id	int
FK staff_id	smallint
date_time	datetime
notes	varchar(255)

reservation	
PK reservation_id	int
FK guest_id	int
FK room_number	smallint
FK invoice_number	mediumint
FK promotion_code	char(10)
FK reservation_staff_id	smallint
reservation_date_time	datetime
number_of_guests	tinyint
start_of_stay	date
length_of_stay	smallint
status_code	char(2)

promotion	
PK promotion_code	char(10)
promotion_name	varchar(50)
discount_percentage	decimal(5, 2)

staff	
PK staff_id	smallint
FK manager_id	smallint
title	varchar(10)
first_name	varchar(80)
last_name	varchar(80)
role	varchar(15)

check_out	
PF reservation_id	int
FK staff_id	smallint
date_time	datetime
settled_invoice	tinyint
notes	varchar(255)

invoice	
PK invoice_number	mediumint
invoice_date	date
amount_due	decimal(7, 2)
amount_paid	decimal(7, 2)
payment_method	varchar(20)
payment_date	date

cleaning_session	
PK date_of_clean	date
PF staff_id	smallint
allocated_master_key	char(1)

room_clean	
PF room_number	smallint
PF date_of_clean	date
FK staff_id	smallint
time_of_clean	time
type_of_clean	char(1)

room	
PK room_number	smallint
FK room_type_code	char(3)
FK bathroom_type_code	char(2)
status	char(3)
key_serial_number	varchar(15)

room_price	
PF room_type_code	char(3)
PF bathroom_type_code	char(2)
price	decimal(6, 2)

bathroom_type	
PK bathroom_type_code	char(2)
bathroom_type_name	varchar(50)
seperate_shower	tinyint
bath	tinyint

room_type	
PK room_type_code	char(3)
room_type_name	varchar(25)
modern_style	tinyint
deluxe	tinyint
maximum_guests	tinyint

***Figure 11** – Physical Design showing data types with length*

8 Data Dictionary

The data dictionary details the database tables with their attributes, data types, formats, constraints, and relationships.

When choosing data types and lengths, storage requirements were considered.

- *tinyint* and *smallint* were used when the maximum value didn't require *integer*.
- *varchar* was used when text of variable length would be entered
- *char* was used for fixed length codes.

Primary keys of each table are UNIQUE and most attributes can use the NOT NULL constraint to improve data integrity.

Appendix C shows sample data for these attributes.

Table Name	Attribute	Data Type	Format	Constraints	Key (PK or FK)	FK Table	Description
address	postcode	varchar(7)		Not Null, Unique	PK		The postcode for the address
	address_line1	varchar(80)		Not Null			The street name
	address_line2	varchar(80)					The local area (optional)
	city	varchar(80)		Not Null			The city
	county	varchar(80)		Not Null			The county
bathroom_ type	bathroom_type_code	char(2)	XX	Not Null, Unique	PK		Code representing the type of bathroom
	bathroom_type_name	varchar(50)		Not Null			Name that describes the bathroom type
	seperate_shower	tinyint	0 or 1	Not Null			Indicates if the room has a separate shower (or just one over a bath)
	bath	tinyint	0 or 1	Not Null			Indicates if the room has a bath
check_in	reservation_id	int	###########	Not Null, Unique	PK	reservation	The id of the reservation that has checked-in
	staff_id	smallint	######	Not Null	FK	staff	The id of the receptionist that processed the check-in
	date_time	datetime	YYYY-MM-DD hh:mm:ss	Not Null			The date and time of the check-in
	notes	varchar(255)					Any notes that needed to be added during check-in
check_out	reservation_id	int	###########	Not Null, Unique	PK	reservation	The id of the reservation that has checked-out
	staff_id	smallint	######	Not Null	FK	staff	The id of the receptionist that processed the check-out
	date_time	datetime	YYYY-MM-DD hh:mm:ss	Not Null			The date and time of the check-out
	settled_invoice	tinyint	0 or 1	Not Null			Indicates if the invoice was settled at check-out
	notes	varchar(255)					Any notes that needed to be added during check-out
cleaning_ session	date_of_clean	date	YYYY-MM-DD	Not Null	PK		The date of this cleaning session
	staff_id	smallint		Not Null	PK & FK	staff	The id of the cleaner for this cleaning session
	allocated_master_key	char(1)	X	Not Null			Indicates which master key is allocated to the cleaner on this day

***Table 6** – Data Dictionary part 1*

company_ account	company_id	int	###########	Not Null, Unique	PK		Unique identifier for the company account
	company_name	varchar(255)		Not Null			Name of the company
	building	varchar(50)		Not Null			Building name or number
	postcode	varchar(7)		Not Null	FK	address	Postcode of the company's address
	admin_title	varchar(10)		Not Null			Title of the company administrator (e.g., Mr, Mrs)
	admin_first_name	varchar(80)		Not Null			First name of the company administrator
	admin_last_name	varchar(80)		Not Null			Last name of the company administrator
	admin_phone_number	varchar(11)		Not Null			Contact phone number for the company administrator
	admin_email	varchar(320)		Not Null			Email address of the company administrator
complaint	reservation_id	int	###########	Not Null	PK & FK	reservation	Reservation associated with the complaint
	opened_date	datetime	YYYY-MM-DD hh:mm:ss	Not Null	PK		Date & time when the complaint was opened
	category_code	char(4)	XXXX	Not Null	FK	complaint_category	Complaint category code
	opened_by	smallint	######	Not Null	FK	staff	Staff member who opened the complaint
	description	varchar(255)		Not Null			Description of the complaint
complaint_ category	category_code	char(4)	XXXX	Not Null, Unique	PK		Complaint category code (for Noise, Smell, Cleaning etc)
	category_name	varchar(80)		Not Null			Name the describes the complaint code
	severity	tinyint	###	Not Null			Indicates the severity of complaint that this code represents
complaint_ resolution	reservation_id	int	###########	Not Null	PK & FK	complaint	Reservation associated with the complaint
	opened_date	datetime	YYYY-MM-DD hh:mm:ss	Not Null	PK & FK	complaint	Date & time when the complaint was opened
	resolved_by	smallint	######	Not Null	FK	staff	Staff member who resolved the complaint
	resolution	varchar(255)		Not Null			Description of how the complaint was resolved
	resolution_date	datetime	YYYY-MM-DD hh:mm:ss	Not Null			Date & Time when the complaint was resolved
guest	guest_id	int	###########	Not Null, Unique	PK		Unique identifier for each guest
	company_id	int	###########		FK	company_account	Company account, if any, linked to the guest
	title	varchar(10)		Not Null			Title of the guest (e.g., Mr, Mrs)
	first_name	varchar(80)		Not Null			First name of the guest
	last_name	varchar(80)		Not Null			Last name of the guest
	phone_number	varchar(11)		Not Null			Contact phone number for the guest
	email	varchar(320)		Not Null			Email address of the guest
	house_name_number	varchar(50)		Not Null			House name or number of the guest's address
	postcode	varchar(7)		Not Null	FK	address	Postcode of the guest's address
invoice	invoice_number	mediumint	#########	Not Null, Unique	PK		Unique invoice number
	invoice_date	date	YYYY-MM-DD	Not Null			Date the invoice was generated
	amount_due	decimal(7,2)		Not Null			Amount due for the reservation
	amount_paid	decimal(7,2)		Not Null			Amount paid by the guest
	payment_method	varchar(20)					Method of payment used (e.g. Visa, Visa Debit, Cash)
	payment_date	date	YYYY-MM-DD				Date when the payment was made

***Table 7** – Data Dictionary part 2*

marketing	guest_id	int	###########	Not Null, Unique	PK & FK	guest	The id of the guest that wants marketing information
	marketing_code	char(3)	XXX	Not Null			Code representing the type of marketing information the guest requires
	contact_by_phone	tinyint	0 or 1	Not Null			Indicates if the guest wishes to be contacted by phone
	contact_by_email	tinyint	0 or 1	Not Null			Indicates if the guest wishes to be contacted by email
	contact_by_post	tinyint	0 or 1	Not Null			Indicates if the guest wishes to be contacted by post

promotion	promotion_code	char(10)	XXXXXXXXXX	Not Null, Unique	PK		Unique identifier for the promotion
	promotion_name	varchar(50)		Not Null			Name of the promotion
	discount_percentage	decimal(5,2)		Not Null			Discount applied by the promotion

reservation	reservation_id	int	###########	Not Null, Unique	PK		Unique identifier for the reservation
	guest_id	int	###########	Not Null	FK	guest	Guest associated with the reservation
	room_number	smallint	######	Not Null	FK	room	Room assigned for the reservation
	invoice_number	mediumint	#########		FK	invoice	Invoice linked to the reservation
	promotion_code	char(10)	XXXXXXXXXX		FK	promotion	Promotion applied to the reservation
	reservation_staff_id	smallint	######	Not Null	FK	staff	Staff member who handled the reservation
	reservation_date_time	datetime	YYYY-MM-DD hh:mm:ss	Not Null			Date and time when the reservation was made
	number_of_guests	tinyint	###	Not Null			Number of guests staying in the room
	start_of_stay	date	YYYY-MM-DD	Not Null			Start date of the stay
	length_of_stay	smallint	######	Not Null			Length of stay in days
	status_code	char(2)	XX	Not Null			Status of the reservation (e.g., RE = reserved, IN = checked in, OT = checked out)

room	room_number	smallint	######	Not Null, Unique	PK		Unique identifier for each room
	room_type_code	char(3)	XXX	Not Null	FK	room_type	Code representing the type of room (represents: single, double, suite etc)
	bathroom_type_code	char(2)	XX	Not Null	FK	bathroom_type	Code representing the type of bathroom
	status	char(3)	XXX	Not Null			Room status (e.g., ACT = active, CLN = needs cleaning, REP = under repair)
	key_serial_number	varchar(15)		Not Null			Serial number of the key

room_clean	room_number	smallint	######	Not Null	PK & FK	room	The number of the room to be cleaned
	date_of_clean	date	YYYY-MM-DD	Not Null	PK & FK	cleaning_session	The date that the room is scheduled to be cleaned
	staff_id	smallint	######	Not Null	FK	cleaning_session	The id of the cleaner that will clean the room
	time_of_clean	time	hh:mm:ss	Not Null			The time of day the room is scheduled to be cleaned
	type_of_clean	char(1)	X	Not Null			Code to indicate if a Light or Full clean is required

room_price	room_type_code	char(3)	XXX	Not Null	PK & FK	room_type	These two attributes combine to
	bathroom_type_code	char(2)	XX	Not Null	PK & FK	bathroom_type	uniquely identify the price
	price	decimal(6,2)	####.##	Not Null			The price of this category of room

***Table 8** – Data Dictionary part 3*

room_type	room_type_code	char(3)	XXX	Not Null, Unique	PK		Code representing the type of room (represents: single, double, suite etc)
	room_type_name	varchar(25)		Not Null			Name of the room type
	modern_style	tinyint	0 or 1	Not Null			Indicates if the room has a modern or traditional style
	deluxe	tinyint	0 or 1	Not Null			Indicates if the room has a deluxe finish
	maximum_guests	tinyint	###	Not Null			Number of guests the room can accommodate

staff	staff_id	smallint	######	Not Null, Unique	PK		Unique identifier for each staff
	manager_id	smallint	######		FK	staff	Staff ID of the manager
	title	varchar(10)		Not Null			Title of the staff member (e.g., Mr, Mrs)
	first_name	varchar(80)		Not Null			First name of the staff member
	last_name	varchar(80)		Not Null			Last name of the staff member
	role	varchar(15)		Not Null			Role of the staff member (represents: Reception, Cleaning, Leader, Owner)

***Table 9** – Data Dictionary part 4*

9 Keys, relationships of entities, cardinality

Tables use Primary keys to uniquely identify a row. Foreign keys are used to form a relation between one table and another table's primary key (Connolly & Begg, 2014). Most tables in this design use a single attribute as the primary key. Composite keys were used when appropriate – e.g. *room_price* table where a room is priced based on the type of room & type of bathroom combined. The reservation table is the centre of the database; it uses foreign keys to reference other main tables, each which use foreign keys, as required, to reference other supporting tables.

One-to-many cardinality was the most used in the design (Date, 2012). e.g. *guest* has a one-to-many relationship with *reservation*, where each guest may book multiple reservations in a year, but each reservation is linked to only one guest. The concept of room cleaning would have required a many-to-many relationship between *staff* & *room* but the *room_clean* table acts as a link table to avoid this.

Some relationships are optional (zero or more) rather than mandatory (one or more). This is necessary as tables slowly populate with data about a reservation over time. e.g. When a guest reserves a room, only the *guest*, *address* and *reservation* tables would be used. When a price is agreed a row in the *invoice* table would be added. Later *check-in*, then *check-out*, and perhaps *complaint* tables would be used.

10 Example Queries

- Find a guest's room number.
- Find available rooms of a specific type between certain dates.
- Report on guest complaints split by category_code.
- Discover which companies and/or guests have booked the most nights.
- Find reservations that have checked-out without settling the invoice.
- Report on total revenue by type of room and their occupancy percentage.
- Which promotion codes have been effective?
- Report on the number of reservations/check-ins/check-outs processed by each member of staff.

11 Privacy and Security considerations

The database contains guest and company contact information that is sensitive; it must comply with the Data Protection Act (DPA, 2018). Data isn't shared with a third-party without explicit consent. The hotel uses a third-party for payment processing and won't hold credit card information in the database.

Access to data will be determined by staff role. Only managers can run long-term reports. Where possible sensitive data will be anonymised.

Encryption will secure the data; a strict password policy will limit access.

Internet security software will be installed on devices to protect against malware and viruses.

The contents of the database is important for the business, so regular encrypted backups must be taken and kept off-site. (Date, 2004)

To comply with the General Data Protection Regulation (GDPR), guest data will only be stored for as long as necessary. Guests will have the right to request deletion from the database.

12 Conclusion

The design stage was challenging, hotel reservations are complex. As the design expanded, the time constraints of this project became limiting.

Research was required to discover which data types should be used (along with size and format) for common attributes such as phone numbers, postcodes, email addresses and financial values.

The payment/invoice section of the database is limited and would need improvement for real world application. The database would need to provide support for room service and additional chargeable items.

Even with those limitations, this design would create a useful hotel management database. It meets the key requirements and can support the example queries.

References

Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. Communications of the ACM

Connolly, T., & Begg, C. (2014). Database Systems: A Practical Approach to Design, Implementation, and Management (6th ed.). Pearson Education.

Elmasri, R., & Navathe, S. (2015). Fundamentals of Database Systems (7th ed.). Pearson.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). Database System Concepts (6th ed.). McGraw-Hill.

Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). Database Systems: The Complete Book (2nd ed.). Pearson.

Date, C. J. (2004). An Introduction to Database Systems (8th ed.). Addison-Wesley.

Date, C. J. (2012). Database Design and Relational Theory: Normal Forms and All That Jazz. O'Reilly Media.

Harrington, J.L., (2016). Relational Database Design and Implementation: Clearly Explained. 4th ed. Amsterdam: Morgan Kaufmann.

Coronel, C. & Morris, S., (2019). Database System Design, Implementation and Management. 12th ed. Cengage Learning

DPA (2018). Data Protection Act 2018. Retrieved from: https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted

Wikipedia contributors. (2023). Boyce–Codd normal form. In Wikipedia, The Free Encyclopaedia. Retrieved from https://en.wikipedia.org/wiki/Boyce%E2%80%93Codd_normal_form

Appendices

Appendix A - Analysis of the entity relationships of the conceptual design

'From' table	'To' table	'To' end Cardinality	Relationship (From → To)
guest	company_account	0 or 1	A guest may belong to a company
guest	marketing	0 or 1	A guest may have agreed to receive marketing information
guest	reservation	1 or more	A guest makes one or more reservations over time
reservation	complaint	0 or more	A reservation may have one or more complaints raised during a guest's stay
reservation	guest	1	A reservation is made by one guest
reservation	promotion	0 or 1	A reservation may have used a promotional discount code
reservation	invoice	0 or 1	A reservation will have an invoice created once the price is agreed or any payment has been made
reservation	room	1	A reservation books one room
reservation	staff	1	1 member of staff creates the reservation to reserves the room.
reservation	staff	0 or 1	Two separate optional relationships from reservation to staff entity: 1 member of staff processes the check-in if guest arrives. 1 member of staff processes the check-out if guest has previously checked-in.
complaint	reservation	0 or 1	If a complaint is raised during the occupancy of room defined by one reservation.
complaint	staff	1	A complaint will be opened by one member of staff.
complaint	staff	0 or 1	A complaint, if resolved, will be resolved by one member of staff.
staff	staff	0 or more	A member of staff may manage other staff and they themselves will be managed by one other member of staff. Apart from the owner who does not have a manager.
staff	complaint	0 or more	A member of staff may open or resolve a complaint.
staff	room_clean	0 or more	A member of staff may be booked to clean one or more rooms.
staff	reservation	0 or more	A member of staff may reserve a room, check-in a guest, check-out a guest
company_account	guest	0 or more	A company account can be created in advance of guest bookings. It may send multiple guests to the hotel.
marketing	guest	1	A marketing record will be created for a guest that agrees to receive marketing information.
promotion	reservation	0 or more	A promotion code may not have been used yet, but could be used by many reservations.

invoice	reservation	1	Once created, an invoice is linked to a single reservation.
room	reservation	0 or more	A room exists before the first reservation. It is likely to be reserved multiple times over a year.
room	room_clean	0 or more	A room exists before the first cleaning, but a room will be cleaned multiple times over a year.
room_clean	room	1	A room cleaning booking occurring at a specific date and time will clean one room.
room_clean	staff	1	A room cleaning booking occurring at a specific date and time will be cleaned by one member of staff.

Appendix B - Candidate Keys of the room_clean table and the BCNF violation

Background

- A hotel room is cleaned once a day (either a light clean or a full clean).
- A member of staff can clean multiple rooms in a day but will be allocated a master key for the whole day to give them access to all the rooms.
- The room_clean table contains data to show which rooms are to be cleaned (including the type of clean required) at what time on what day by which member of staff, and what key they are using.

Table Attributes

- room_number
- date_of_clean
- staff_id
- time_of_clean
- allocated_master_key
- type_of_clean

Functional dependencies (Fd)

Fd	Attributes making the functional dependency	Candidate Key
1	{ room_number, date_of_clean → time_of_clean, staff_id, allocated_master_key, type_of_clean }	Y
2	{ staff_id, date_of_clean → room_number, time_of_clean, allocated_master_key, type_of_clean }	Y
3	{ allocated_master_key, date_of_clean, time_of_clean → room_number, staff_id, type_of_clean }	Y
4	{ date_of_clean, staff_id → allocated_master_key }	N

Therefore, only type_of_clean is a non-prime attribute.

Normalisation

- The table passes 1NF checks - it has a primary key and has only atomic attributes.
- The table passes 2NF checks - the only non-prime attribute depends on entire candidate keys. (Fd1, Fd2 & Fd3)
- The table passes 3NF checks - there is only one non-prime attribute, so it cannot have transitive dependency on another non-prime attribute.
- **The table violates BCNF** - Fd4, which determines allocated_master_key, is not itself a candidate key because {date_of_clean, staff_id} cannot uniquely identify a row.

Appendix C – Sample Data

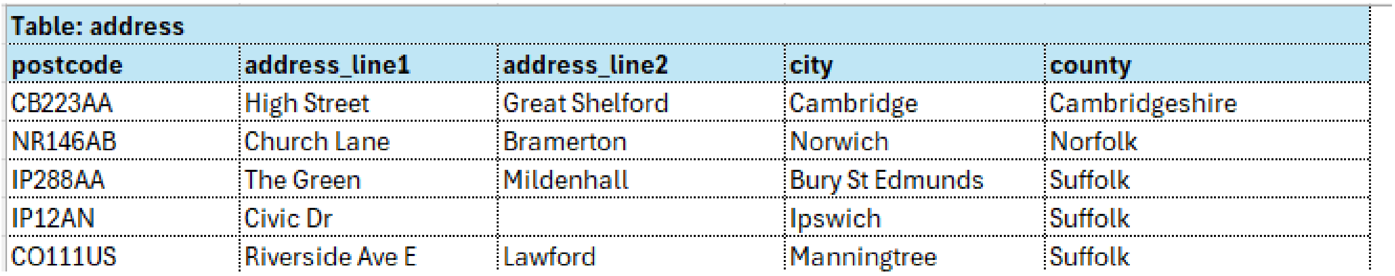


Table: address				
postcode	**address_line1**	**address_line2**	**city**	**county**
CB223AA	High Street	Great Shelford	Cambridge	Cambridgeshire
NR146AB	Church Lane	Bramerton	Norwich	Norfolk
IP288AA	The Green	Mildenhall	Bury St Edmunds	Suffolk
IP12AN	Civic Dr		Ipswich	Suffolk
CO111US	Riverside Ave E	Lawford	Manningtree	Suffolk

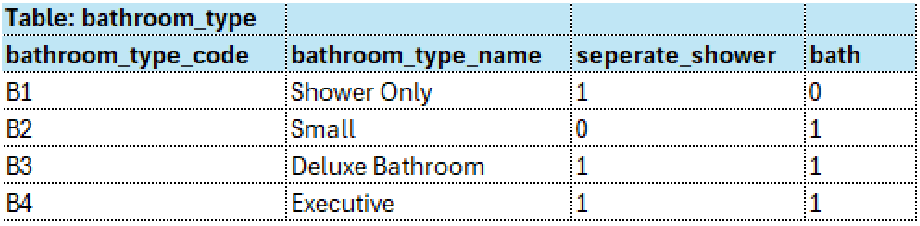


Table: bathroom_type			
bathroom_type_code	**bathroom_type_name**	**seperate_shower**	**bath**
B1	Shower Only	1	0
B2	Small	0	1
B3	Deluxe Bathroom	1	1
B4	Executive	1	1

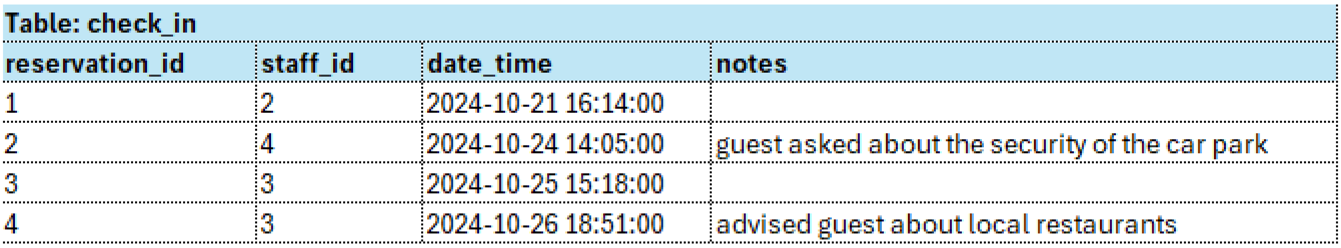


Table: check_in			
reservation_id	**staff_id**	**date_time**	**notes**
1	2	2024-10-21 16:14:00	
2	4	2024-10-24 14:05:00	guest asked about the security of the car park
3	3	2024-10-25 15:18:00	
4	3	2024-10-26 18:51:00	advised guest about local restaurants

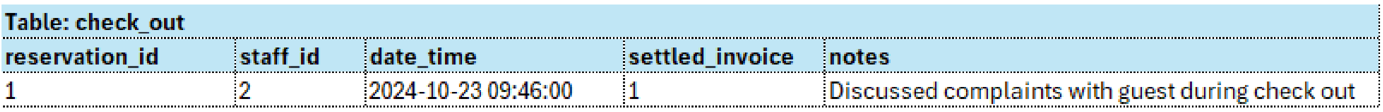


Table: check_out				
reservation_id	**staff_id**	**date_time**	**settled_invoice**	**notes**
1	2	2024-10-23 09:46:00	1	Discussed complaints with guest during check out

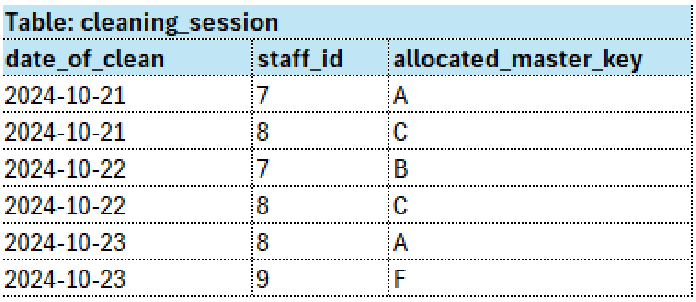


Table: cleaning_session		
date_of_clean	**staff_id**	**allocated_master_key**
2024-10-21	7	A
2024-10-21	8	C
2024-10-22	7	B
2024-10-22	8	C
2024-10-23	8	A
2024-10-23	9	F

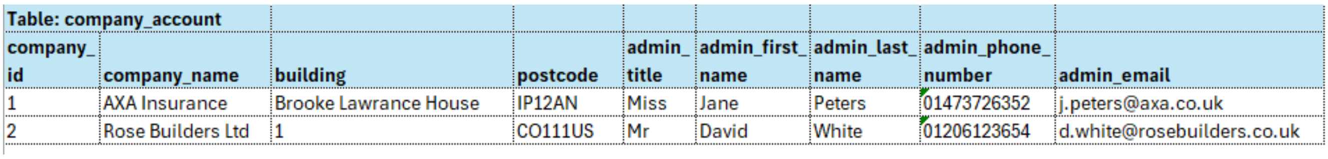


Table: company_account								
company_id	**company_name**	**building**	**postcode**	**admin_title**	**admin_first_name**	**admin_last_name**	**admin_phone_number**	**admin_email**
1	AXA Insurance	Brooke Lawrance House	IP12AN	Miss	Jane	Peters	01473726352	j.peters@axa.co.uk
2	Rose Builders Ltd	1	CO111US	Mr	David	White	01206123654	d.white@rosebuilders.co.uk

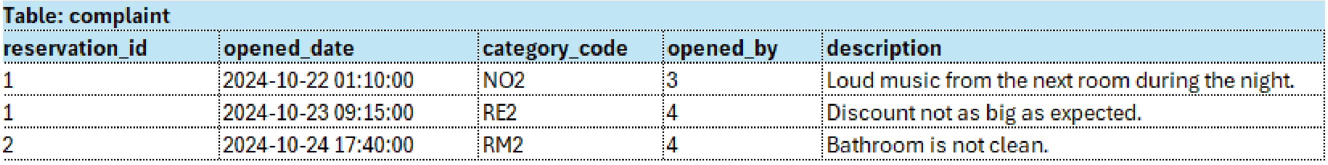


Table: complaint				
reservation_id	**opened_date**	**category_code**	**opened_by**	**description**
1	2024-10-22 01:10:00	NO2	3	Loud music from the next room during the night.
1	2024-10-23 09:15:00	RE2	4	Discount not as big as expected.
2	2024-10-24 17:40:00	RM2	4	Bathroom is not clean.

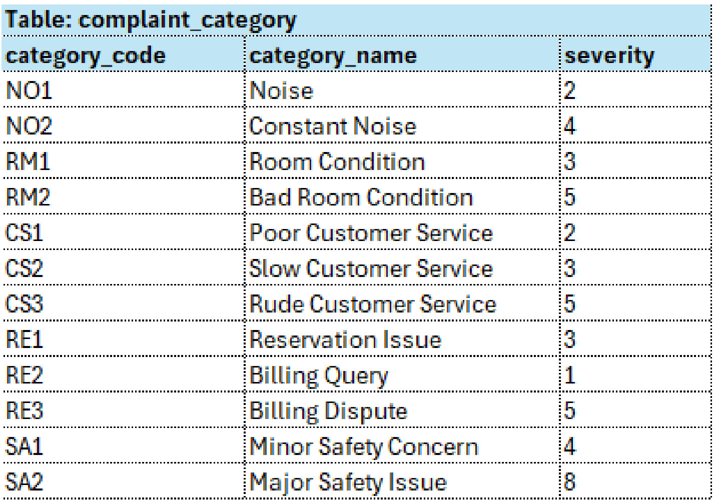


Table: complaint_category

category_code	category_name	severity
NO1	Noise	2
NO2	Constant Noise	4
RM1	Room Condition	3
RM2	Bad Room Condition	5
CS1	Poor Customer Service	2
CS2	Slow Customer Service	3
CS3	Rude Customer Service	5
RE1	Reservation Issue	3
RE2	Billing Query	1
RE3	Billing Dispute	5
SA1	Minor Safety Concern	4
SA2	Major Safety Issue	8

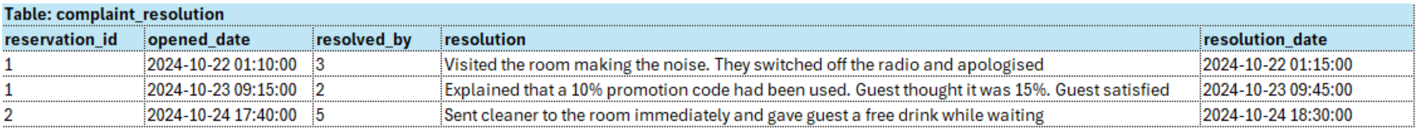


Table: complaint_resolution

reservation_id	opened_date	resolved_by	resolution	resolution_date
1	2024-10-22 01:10:00	3	Visited the room making the noise. They switched off the radio and apologised	2024-10-22 01:15:00
1	2024-10-23 09:15:00	2	Explained that a 10% promotion code had been used. Guest thought it was 15%. Guest satisfied	2024-10-23 09:45:00
2	2024-10-24 17:40:00	5	Sent cleaner to the room immediately and gave guest a free drink while waiting	2024-10-24 18:30:00

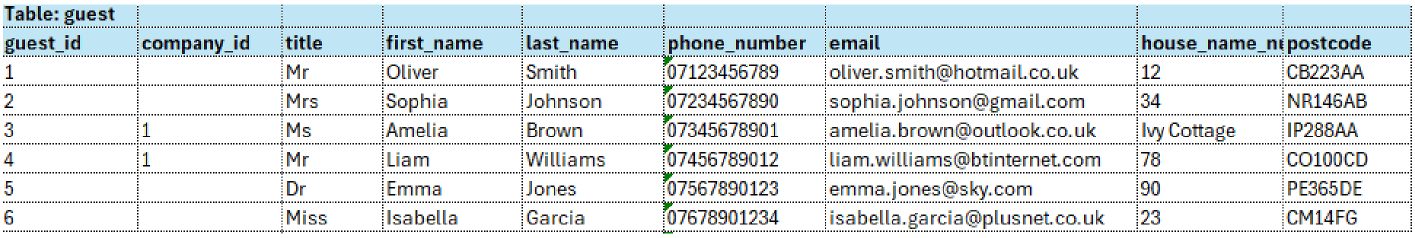


Table: guest

guest_id	company_id	title	first_name	last_name	phone_number	email	house_name_n	postcode
1		Mr	Oliver	Smith	07123456789	oliver.smith@hotmail.co.uk	12	CB223AA
2		Mrs	Sophia	Johnson	07234567890	sophia.johnson@gmail.com	34	NR146AB
3	1	Ms	Amelia	Brown	07345678901	amelia.brown@outlook.co.uk	Ivy Cottage	IP288AA
4	1	Mr	Liam	Williams	07456789012	liam.williams@btinternet.com	78	CO100CD
5		Dr	Emma	Jones	07567890123	emma.jones@sky.com	90	PE365DE
6		Miss	Isabella	Garcia	07678901234	isabella.garcia@plusnet.co.uk	23	CM14FG

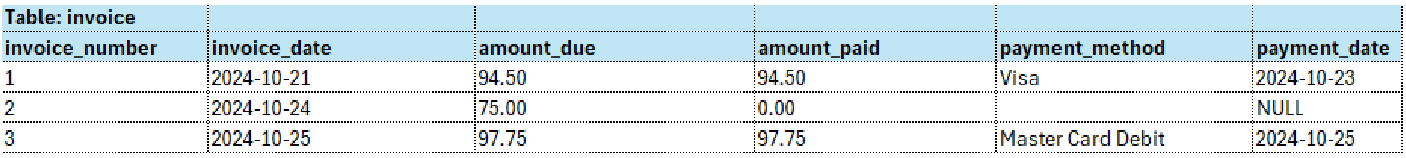


Table: invoice

invoice_number	invoice_date	amount_due	amount_paid	payment_method	payment_date
1	2024-10-21	94.50	94.50	Visa	2024-10-23
2	2024-10-24	75.00	0.00		NULL
3	2024-10-25	97.75	97.75	Master Card Debit	2024-10-25

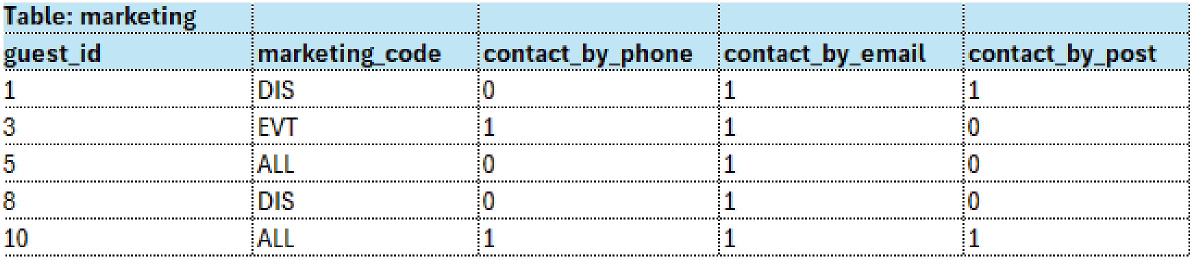


Table: marketing

guest_id	marketing_code	contact_by_phone	contact_by_email	contact_by_post
1	DIS	0	1	1
3	EVT	1	1	0
5	ALL	0	1	0
8	DIS	0	1	0
10	ALL	1	1	1

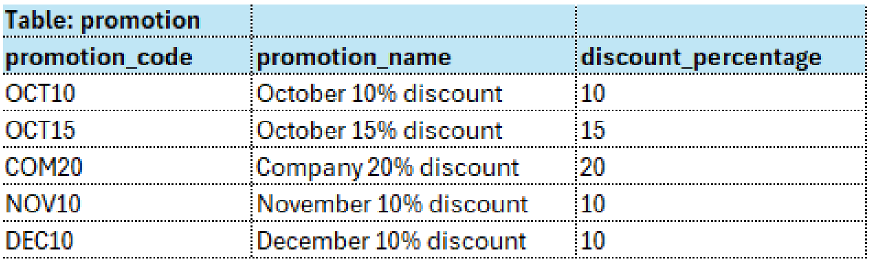


Table: promotion

promotion_code	promotion_name	discount_percentage
OCT10	October 10% discount	10
OCT15	October 15% discount	15
COM20	Company 20% discount	20
NOV10	November 10% discount	10
DEC10	December 10% discount	10

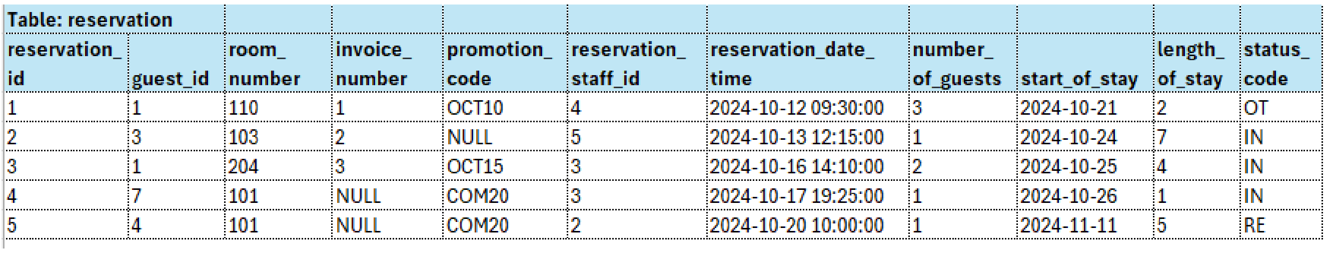


Table: reservation

reservation_		room_	invoice_	promotion_	reservation_	reservation_date_	number_		length_	status_
id	guest_id	number	number	code	staff_id	time	of_guests	start_of_stay	of_stay	code
1	1	110	1	OCT10	4	2024-10-12 09:30:00	3	2024-10-21	2	OT
2	3	103	2	NULL	5	2024-10-13 12:15:00	1	2024-10-24	7	IN
3	1	204	3	OCT15	3	2024-10-16 14:10:00	2	2024-10-25	4	IN
4	7	101	NULL	COM20	3	2024-10-17 19:25:00	1	2024-10-26	1	IN
5	4	101	NULL	COM20	2	2024-10-20 10:00:00	1	2024-11-11	5	RE

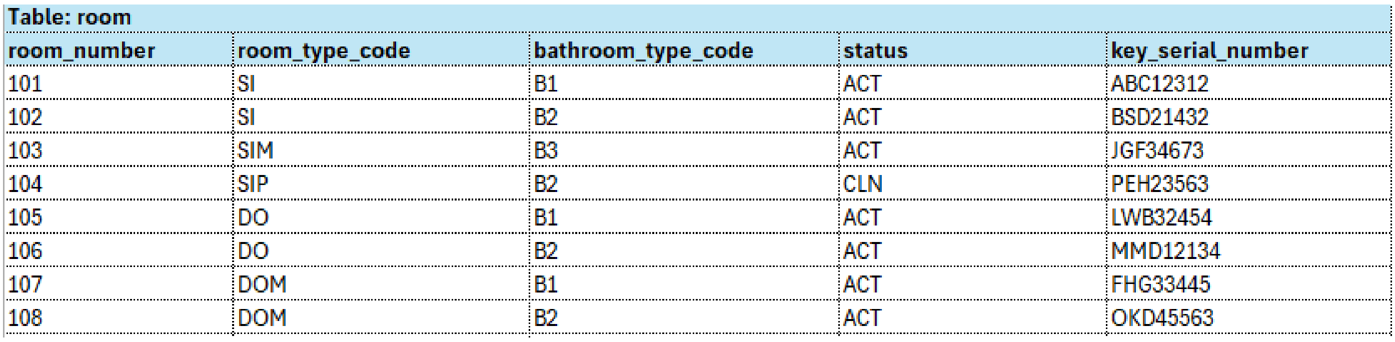


Table: room

room_number	room_type_code	bathroom_type_code	status	key_serial_number
101	SI	B1	ACT	ABC12312
102	SI	B2	ACT	BSD21432
103	SIM	B3	ACT	JGF34673
104	SIP	B2	CLN	PEH23563
105	DO	B1	ACT	LWB32454
106	DO	B2	ACT	MMD12134
107	DOM	B1	ACT	FHG33445
108	DOM	B2	ACT	OKD45563

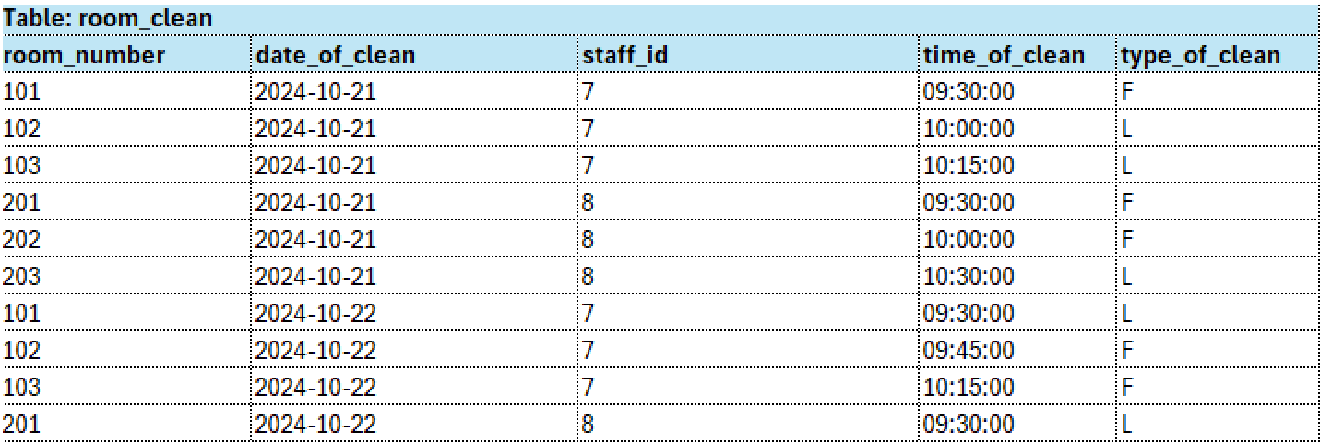


Table: room_clean

room_number	date_of_clean	staff_id	time_of_clean	type_of_clean
101	2024-10-21	7	09:30:00	F
102	2024-10-21	7	10:00:00	L
103	2024-10-21	7	10:15:00	L
201	2024-10-21	8	09:30:00	F
202	2024-10-21	8	10:00:00	F
203	2024-10-21	8	10:30:00	L
101	2024-10-22	7	09:30:00	L
102	2024-10-22	7	09:45:00	F
103	2024-10-22	7	10:15:00	F
201	2024-10-22	8	09:30:00	L

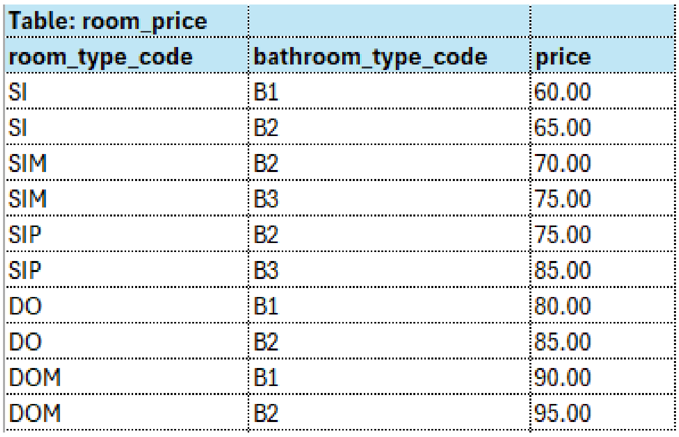


Table: room_price

room_type_code	bathroom_type_code	price
SI	B1	60.00
SI	B2	65.00
SIM	B2	70.00
SIM	B3	75.00
SIP	B2	75.00
SIP	B3	85.00
DO	B1	80.00
DO	B2	85.00
DOM	B1	90.00
DOM	B2	95.00

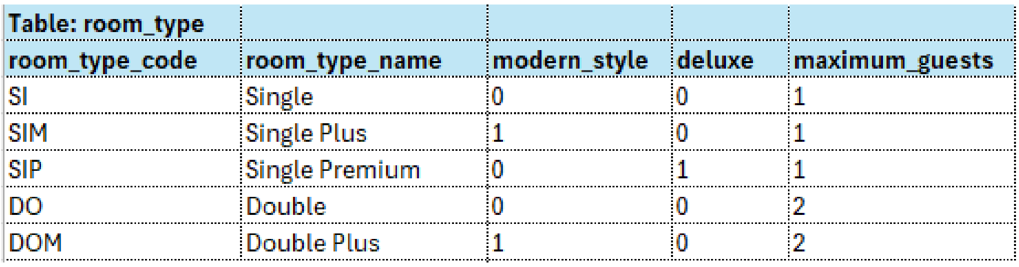


Table: room_type

room_type_code	room_type_name	modern_style	deluxe	maximum_guests
SI	Single	0	0	1
SIM	Single Plus	1	0	1
SIP	Single Premium	0	1	1
DO	Double	0	0	2
DOM	Double Plus	1	0	2

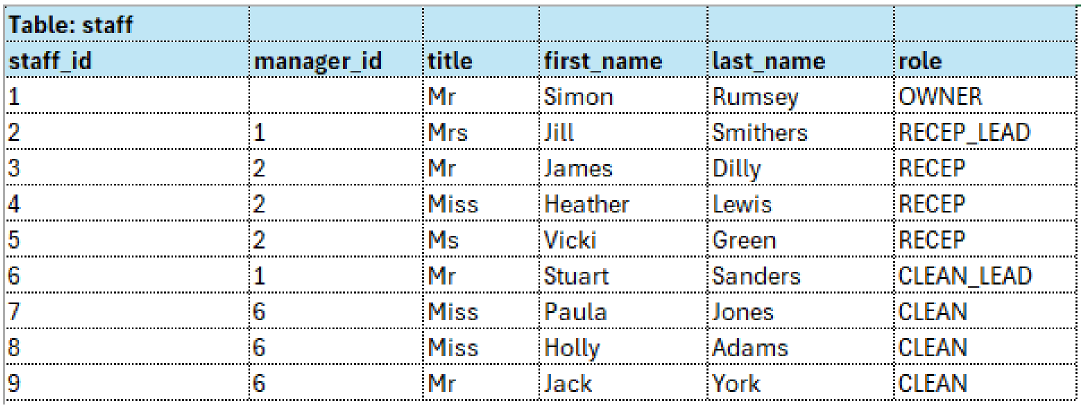


Table: staff

staff_id	manager_id	title	first_name	last_name	role
1		Mr	Simon	Rumsey	OWNER
2	1	Mrs	Jill	Smithers	RECEP_LEAD
3	2	Mr	James	Dilly	RECEP
4	2	Miss	Heather	Lewis	RECEP
5	2	Ms	Vicki	Green	RECEP
6	1	Mr	Stuart	Sanders	CLEAN_LEAD
7	6	Miss	Paula	Jones	CLEAN
8	6	Miss	Holly	Adams	CLEAN
9	6	Mr	Jack	York	CLEAN