Database Systems Development
M30232

CW Group contribution statement

Group 27

	UP2277697	UP2258434	UP2306587
ERD Design	☑	☑	☑
Reflective Analysis	☑	☑	☑
Data Dictionary	☑	☑	☑
Document writing	☑	☑	☑
Brainstorming	☑	☑	☑
SQL Coding	☑	☑	☑
DDT	☑	☑	☑
Percentage Allocation	**33.3%**	**33.3%**	**33.3%**
Signature			

Contribution Description

UP NUMBER	Description (if not equally counted)	Signature
UP1234567		
UP1234567		
UP1234567		

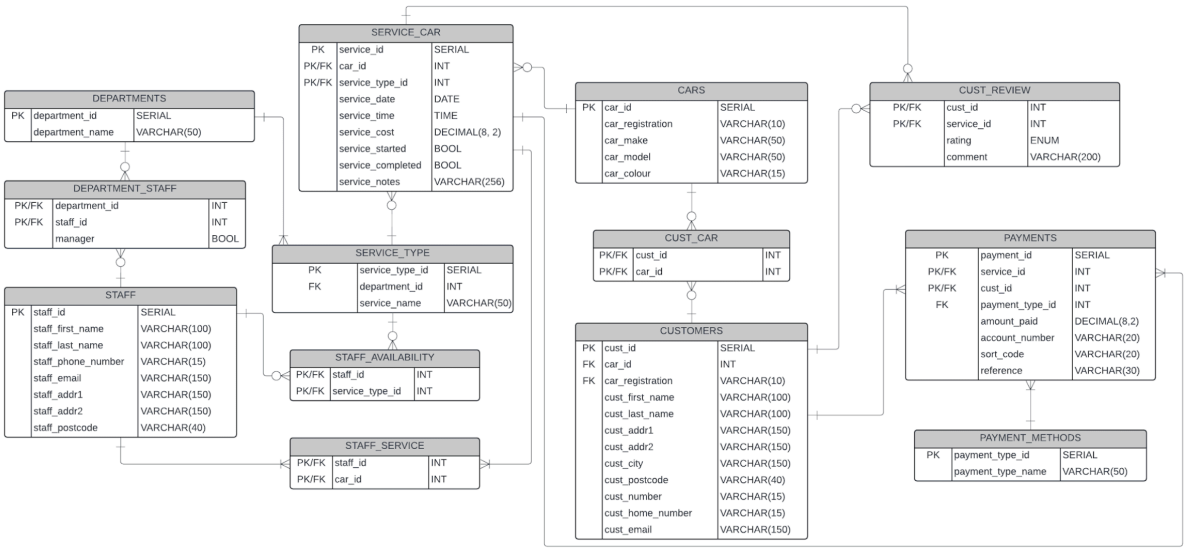
Note[1]: *The contribution is not necessary only "hands on". The contribution can be as research, brainstorming, tests, diagrams etc. For DOCUMENT WRITING one person should write (to maintain the same writing style and general layout) but all members should contribute with ideas, moderation or research.*

Note[2]: *The Percentage Allocation between team members cannot exceed 100%.*

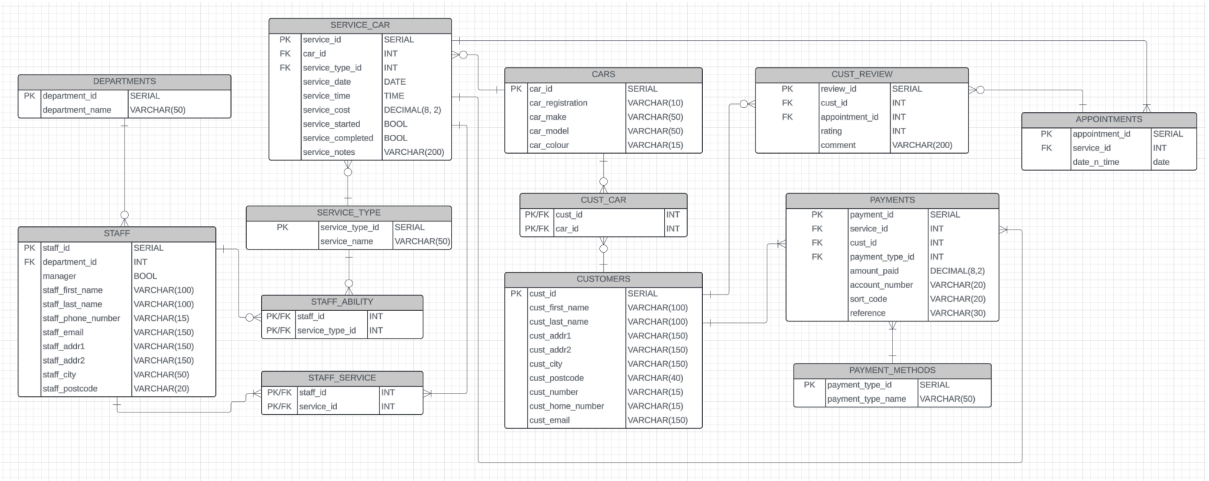
Note[3]: *For submission without Group Contribution Statement (or blank) it will be considered that all members are contributed equally.*

Group 27

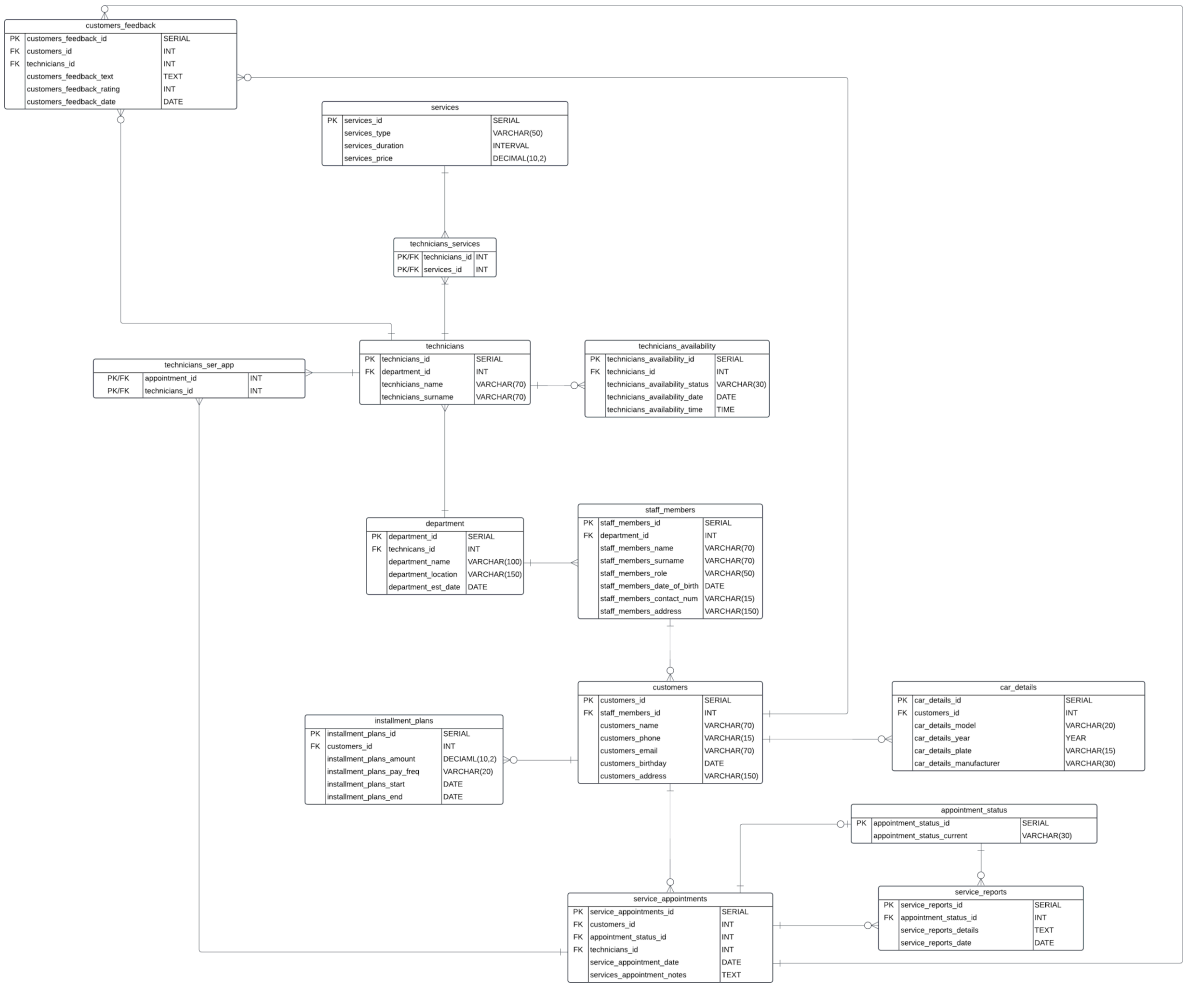
ERD 1 Previous Version



ERD 1 Updated Version

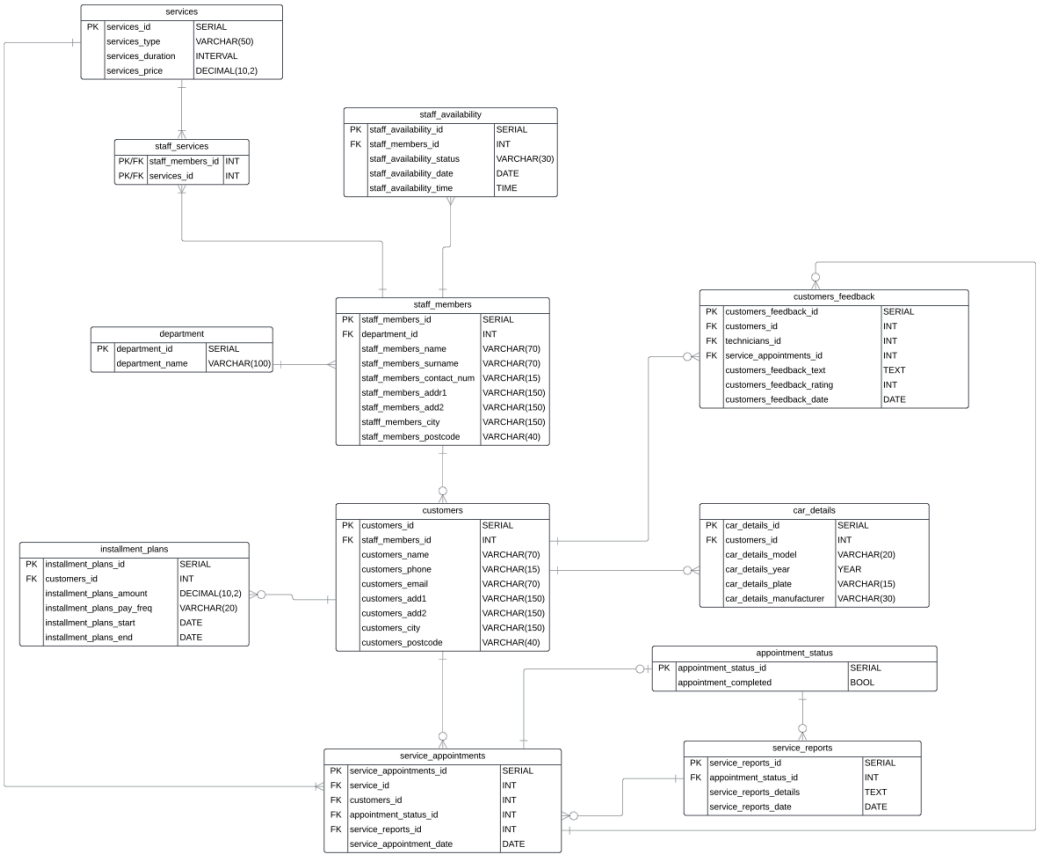


ERD 2 Previous Version



Group 27

ERD 2 Updated Version



Data Dictionary

CARS						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
car_id	PK		SERIAL			
car_registration	AK	Y	VARCHAR(10)	UNIQUE, NOT NULL		Registration plate for customer's car
car_make			VARCHAR(50)	NOT NULL		Make of car, e.g. Ford, BMW
car_model			VARCHAR(50)	NOT NULL		Model of car, e.g. Ford Focus
car_colour			VARCHAR(15)			

CUSTOMERS						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
cust_id	PK		SERIAL			
cust_first_name			VARCHAR(100)	NOT NULL		
cust_last_name		Y	VARCHAR(100)	NOT NULL		
cust_addr1			VARCHAR(150)	NOT NULL		Address of Customer
cust_addr2			VARCHAR(150)			-
cust_city			VARCHAR(100)	NOT NULL		-

cust_postcode			VARCHAR(7)	NOT NULL		-
cust_number	AK		VARCHAR(15)	NOT NULL, UNIQUE		Personal phone number of customer used to contact them
cust_home_number			VARCHAR(15)			Customers home phone number
cust_email	AK		VARCHAR(150)	UNIQUE		Personal email of the customer

STAFF						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
staff_id	PK		SERIAL			
department_id	FK	Y	INT	NOT NULL	departments.department_id	
manager			BOOL	NOT NULL		Determines if the staff member is a manager or not. True if they are a manager, false if they are not
staff_first_name			VARCHAR(100)	NOT NULL		
staff_last_name		Y	VARCHAR(100)	NOT NULL		
staff_phone_number	AK		VARCHAR(15)	NOT NULL, UNIQUE		Personal phone number of staff member
staff_email			VARCHAR(150)	UNIQUE		Personal email of staff member
staff_addr1			VARCHAR(150)	NOT NULL		Staff member's home address
staff_addr2			VARCHAR(150)			-
staff_city			VARCHAR(50)	NOT NULL		-

staff_postcode			VARCHAR(20)	NOT NULL		-

DEPARTMENTS						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
department_id	PK		SERIAL			
department_name			VARCHAR(50)	NOT NULL		Name of department

SERVICE_TYPE						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
service_type_id	PK		SERIAL			
service_name			VARCHAR(50)	NOT NULL		The name of the service type, e.g. oil change

SERVICE_CAR						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
service_id	PK		SERIAL			
car_id	PK/FK		INT	NOT NULL	cars.car_id	
service_type_id	PK/FK	Y	INT	NOT NULL	service.service_type_id	
service_date		Y	DATE	NOT NULL		The date that the service is scheduled for
service_time			TIME	NOT NULL		
service_cost			DECIMAL(8,2)	NOT NULL		
service_started			BOOL	NOT NULL		Whether the service started or not
service_completed			BOOL	NOT NULL		
service_notes			VARCHAR(200)			Any comments about the car from the service, e.g. oil changed, filter replaced etc

STAFF_ABILITY						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
staff_id	PK/FK		INT	NOT NULL	staff.staff_id	
service_type_id	PK/FK		INT	NOT NULL	service_type.service_type_id	

STAFF_SERVICE						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
staff_id	PK/FK		INT	NOT NULL	staff.staff_id	
service_id	PK/FK		INT	NOT NULL	service_car.service_id	

CUST_REVIEW						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
review_id	PK		SERIAL			
cust_id	PK/FK		INT	NOT NULL	customers.cust_id	
appointment_id	PK/FK		INT	NOT NULL	appointments.appointment_id	
rating			INT	CHECK (rating >= 1		Rating of 1-5 from the customer. 1 being the worst and 5

				AND rating <= 5)		being the best
comment			VARCHAR(200)			Any comments the customer may want to leave in the review

APPOINTMENTS						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
appointment_id	PK		SERIAL			
service_id	FK		INT	NOT NULL	service_car.service_id	
date_n_time			DATE	NOT NULL		

PAYMENTS						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
payment_id	PK		SERIAL			
service_id	FK		INT	NOT NULL	service_car.service_id	
payment_type_id	FK		INT	NOT NULL	payment_methods.payment_type_id	
cust_id	FK		INT	NOT NULL	customers.cust_id	
ammount_paid			DECIMAL (8,2)	NOT NULL		How much the customer has paid. The customer does

CUST_CAR						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
cust_id	PK/FK		INT	NOT NULL	customers.cust_id	
car_id	PK/FK		INT	NOT NULL	cars.car_id	

PAYMENT_METHODS						
Attribute_Name	***KEY***	***INDEX***	***Data Type & Size***	***Domains & Constraints***	***FK Reference***	***Description***
payment_type_id	PK		SERIAL			
payment_type_name			VARCHAR(50)	NOT NULL		Name of payment type, e.g. Cash

Reflective Analysis on Database Redesign

The feedback pointed up these problems, and we took important action to fix them. We chose to move on with the final ERD rather than the other, because it was more comprehensive, better standardized, and in line with the CarCare Hub system's needs. Below, we presented the main factors we took into account and our reasoning behind selecting the final design below.

1. Addition of Appointments Table

The addition of an **appointments** table, which was absent from the initial design, was one notable modification. Without it, there would have been no organized method to monitor client service reservations. In order to guarantee that services are appropriately planned and that customer feedback is connected to completed appointments rather than services directly, the new table ties directly to **service_car** and **cust_review.**

2. Refinement of Staff and Department Relationships

In the past, staff members were selectively assigned to several departments. In order to handle this, we added a **department_id** to the **staff** table, guaranteeing that each employee is assigned to a single department while preserving flexibility through the **staff_ability** table, which keeps track of the particular services that staff are capable of performing. This ensures appropriate specialisation while avoiding needless complexity.

3. Clarification of Customer-Car Relationships

Customers used to be limited to being directly associated with a single car in the **customers** table. In order to ensure that vehicle ownership is controlled through the **cars** table, we created connections in the **service_car** and **appointments** columns. This simplifies customer records while preserving the ability for **customers** to own multiple vehicles. This change eliminates redundancy and increases scalability.

4. Correction of Foreign Key Issues

The original schema contained a number of foreign keys that were either incorrectly or poorly referenced. In **cust_review**, feedback was directly linked to **service_car** rather than **appointments**, and in **staff_service, service_id** was incorrectly linked. These were changed to provide for more traceable and logical linkages between entities while maintaining appropriate relational integrity.

5. Improvement of Review and Rating Mechanism

An **ENUM** was used to define the rating characteristic at first, which restricted flexibility and made it unsuitable for scalability. To ensure that the numbers stay inside a valid range and allow for extension and modification if necessary, we substituted an **INT** data type and applied a **CHECK** constraint **(1 to 5).**

6. Indexing for Performance Optimization

We added indexes to frequently used columns including **car_registration, staff_department, staff_last_name, service_type, service_date and cust_last_name** in order to improve query efficiency. By accelerating searches and filters, it enhances database performance, especially when working with big datasets.

7. Choosing the Final Design

We examined both ERDs before deciding to move forward with the final one for a number of reasons. First of all, it was more detailed and fixed a number of problems that other updated designs still had. For instance, it made a more obvious separation between **technicians** and **staff members**, which helped to better address the problem of overlapping jobs. It also featured more comprehensive foreign key references and better-defined relationships, both of which enhanced the database's overall integrity.

Second, the finished design better matched the CarCare Hub system's specifications. In contrast to prior designs, it featured a more thorough breakdown of client information, including address and contact information. Additionally, it had a more reliable system for tracking **service appointments** and **client feedback**, both of which were essential to the database's operation.

Ultimately, there were less redundancies and a better normalisation in the finished design. It made sure that all relationships were consistent and logical and got rid of any tables that weren't needed. As a result, the schema became easier to maintain and more scalable over time. We chose to stick with this design for the project's coding phase because of these benefits.

Database Systems Development
M30232

Group Number: [27]

Database Development Tracker

Date	Task Description	Member ID	Task Details	Time	Signature
05/02/2025	Initial meeting	UP2258434 UP2277697 UP2306587	We reviewed two ERDs (CW1 and an improved version), compared their structures, and selected the best one for implementation. Both versions will be included in the document for reference. After finalizing the ERD, we set up a live server in Visual Studio Code for collaboration. Each team member worked on creating tables for different entities, ensuring correct attributes, data types, and constraints. In our next meeting, we will complete the data dictionary and begin inserting sample data.	3H	
12/02/2025	Updated data dictionary and inserted data into our code	UP2258434 UP2277697 UP2306587	In this meeting, we worked on finalising key database components. UP2277697 and UP2258434 collaborated to create the updated Data Dictionary, aligning it with our improved ERD. UP2306587 generated and inserted sample data using Mockaroo, ensuring	3H	

			realistic test data. Meanwhile, UP2258434 completed the Database Development Tracker, documenting our progress. The SQL code includes all necessary `CREATE` and `INSERT` statements, structured in the correct execution order with appropriate comments. This ensures it can be copied directly into the VM for testing. In our next meeting, we will review the implementation and test database functionality.		
15/02/2025	Indexing, Query Testing, Debugging, and Data Dictionary Update	UP2258434 UP2277697 UP2306587	In order to evaluate performance, the team worked together to add indexes to the database and test different queries. To make sure the code worked properly after indexing, debugging was done. To verify optimization gains, various queries were attempted. The recently created indexes were later added to the data dictionary. Additionally, with the group's assistance, UP2258434 finished the revision assignment. Combining all of the duties into a comprehensive submission document is the last step left.	3H	