

Task 1: Implementation guide

1. Scenario analysis – entities, attributes, relationships

This version adds more operational nuance: tariffs and staff monitoring.

Entities:

- ParkingSite (as before).
- ParkingSlot (as before).
- Customer (as before).
- Vehicle (as before).
- Tariff:
TariffID (PK), SiteID (FK), TariffName, RatePerHour, AppliesFromTime, AppliesToTime, DayType
Relationship: One ParkingSite has many Tariffs; Tariff used to calculate booking charges.
- Booking:
BookingID (PK), VehicleID (FK), SlotID (FK), SiteID (FK), TariffID (FK), StartDateTime, EndDateTime, CalculatedAmount, Status
- Payment: PaymentID (PK), BookingID (FK), PaymentDateTime, AmountPaid, Method, Status
- StaffUser (optional): StaffID (PK), Username, Role, SiteID (FK) Used for audit and site-level management.

2. ERD – textual description

- ParkingSite (1) — (M) ParkingSlot
- ParkingSite (1) — (M) Tariff
- ParkingSite (1) — (M) Booking
- Customer (1) — (M) Vehicle
- Vehicle (1) — (M) Booking
- ParkingSlot (1) — (M) Booking
- Tariff (1) — (M) Booking
- Booking (1) — (M) Payment
- ParkingSite (1) — (M) StaffUser

3. Data structure development – key tables

Tariff

- Attributes: TariffID (PK, AutoNumber), SiteID (FK), TariffName (Text), RatePerHour (Currency), AppliesFromTime (Date/Time – time only), AppliesToTime (Date/Time – time only), DayType (Text: Weekday/Weekend).
- Constraints: RatePerHour > 0; AppliesFromTime < AppliesToTime.

Booking (extended)

- Attributes: BookingID (PK), VehicleID (FK), SlotID (FK), SiteID (FK), TariffID (FK), StartDateTime, EndDateTime, CalculatedAmount (Currency), Status.
- Constraints: CalculatedAmount ≥ 0; TariffID required; referential integrity to Tariff.

StaffUser

- Attributes: StaffID (PK), Username (Text), Role (Text: Admin/Operator), SiteID (FK).
- Constraints: Username unique; SiteID optional if staff roam across sites.

Other tables remain as in Version 1.

4. Normalization (3NF)

- Tariff separates pricing logic from Booking, avoiding repeated rate values.
- StaffUser isolates user credentials and roles from operational data.
- Booking stores only calculated amount and references Tariff; no embedded rate or site details beyond FKs.

5. Sample data (selected tables)

Tariff

TariffID	SiteID	TariffName	RatePerHour	AppliesFromTime	AppliesToTime	DayType
1	1	Weekday Daytime	2.50	08:00	18:00	Weekday
2	1	Weekday Evening	1.50	18:00	23:00	Weekday
3	2	Station Peak	3.00	07:00	10:00	Weekday
4	2	Station Off-peak	1.80	10:00	20:00	Weekday
5	1	Weekend Flat	2.00	08:00	22:00	Weekend

Booking (with Tariff and CalculatedAmount)

BookingID	VehicleID	SlotID	SiteID	TariffID	StartDateTime	EndDateTime	CalculatedAmount	Status
1	1	2	1	1	2026-07-09 09:00	2026-07-09 11:00	5.00	Completed
2	2	4	2	3	2026-07-09 07:30	2026-07-09 09:00	4.50	Completed
3	3	1	1	2	2026-07-09 19:00	2026-07-09 21:00	3.00	Active
4	4	3	1	1	2026-07-09 10:00	2026-07-09 12:00	5.00	Completed
5	5	5	2	4	2026-07-09 11:00	2026-07-09 13:00	3.60	Completed

Payment (partial payments possible)

PaymentID	BookingID	PaymentDateTime	AmountPaid	Method	Status
1	1	2026-07-09 11:05	5.00	Card	Paid
2	2	2026-07-09 09:05	4.50	App	Paid
3	3	2026-07-09 19:10	1.50	Card	Pending
4	4	2026-07-09 12:05	5.00	Cash	Paid
5	5	2026-07-09 13:05	3.60	App	Paid

6. Query development – extended examples

- Revenue by site and day

```
SELECT ParkingSite.SiteName,  
       FORMAT(Payment.PaymentDateTime, "yyyy-mm-dd") AS RevenueDate,  
       SUM(Payment.AmountPaid) AS TotalRevenue  
FROM Payment  
INNER JOIN Booking ON Payment.BookingID = Booking.BookingID  
INNER JOIN ParkingSite ON Booking.SiteID = ParkingSite.SiteID
```

```
GROUP BY ParkingSite.SiteName, FORMAT(Payment.PaymentDateTime, "yyyy-mm-dd")  
ORDER BY RevenueDate, ParkingSite.SiteName;
```

Tariff effectiveness – average stay per tariff

```
SELECT Tariff.TariffName,  
       AVG(DateDiff("n", Booking.StartDateTime, Booking.EndDateTime)) AS AvgMinutes  
FROM Booking  
INNER JOIN Tariff ON Booking.TariffID = Tariff.TariffID  
GROUP BY Tariff.TariffName;
```

7. Validation and testing – extended

- **Tariff validation:** Confirm that bookings use correct tariff based on time and site; manually calculate expected amounts and compare.
- **Revenue reconciliation:** Sum CalculatedAmount per booking and compare with total AmountPaid from Payment.
- **Stress tests:** Create overlapping bookings for the same slot and ensure constraints or business rules prevent double allocation (e.g., via queries or custom forms).
- **User roles (if implemented):** Test that StaffUser roles restrict access to certain forms or reports.

Task 2: Course work written elements

Table of contents:

- 1. Introduction**
 - **1.1 Scenario and business problem**
 - **1.2 Objectives of the AutoPark database**
- 2. Requirements analysis**
 - **2.1 Operational requirements (slots, bookings, payments)**
 - **2.2 Management requirements (reports, revenue, occupancy)**
- 3. Conceptual and logical design**
 - **3.1 Entity and relationship identification**
 - **3.2 ERD with extended entities (Tariff, StaffUser)**
 - **3.3 Normalization and design rationale**
- 4. Physical implementation in Microsoft Access**
 - **4.1 Table definitions and data types**
 - **4.2 Relationships, indexes, and constraints**
 - **4.3 Sample data and initial configuration**
- 5. Queries, forms, and reports**
 - **5.1 Operational queries (current occupancy, vehicle details)**
 - **5.2 Analytical queries (revenue trends, tariff performance)**
 - **5.3 Example forms and reports**
- 6. Testing and validation**
 - **6.1 Test scenarios and expected outcomes**
 - **6.2 Results and issue resolution**
- 7. Reflection and future enhancements**
 - **7.1 Design decisions and trade-offs**
 - **7.2 Challenges and lessons learned**
 - **7.3 Planned improvements (dynamic pricing, integration)**
- 8. Conclusion**
- 9. References (IEEE format)**

Design decisions (Version 2 focus):

- **Tariff separation:** Keeps pricing flexible and maintainable; management can change rates without altering bookings.
- **CalculatedAmount stored:** Supports historical reporting even if tariffs change later.
- **Optional StaffUser:** Prepares the system for multi-user, role-based access in future.

Challenges and improvements:

- **Challenges:**
 - Designing queries that correctly handle time-based tariffs and overlapping periods.
 - Balancing normalization with performance and ease of use in Access.
- **Improvements:**
 - Add integration with payment gateways.
 - Implement automatic tariff selection via Access forms and VBA.
 - Introduce dashboards for occupancy and revenue using Access reports.

References (IEEE style – examples, similar to Version 1):

- **[1]** R. Elmasri and S. Navathe, *Fundamentals of Database Systems*, 7th ed. Pearson, 2016.
- **[2]** A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed. McGraw-Hill, 2019.
- **[3]** Microsoft, “Access database design basics,” Access documentation, 2024.
- **[4]** N. J. Curran, “Designing time-dependent tariffs in relational databases,” *J. Database Pract.*, vol. 12, no. 3, pp. 45–53, 2020.