**Database Management Assignment**

**Section A: Introduction to SQL/NoSQL**

**1. You are working on a project where you need to store large amounts of structured and semi-structured data. Which type of database (SQL or NoSQL) would you choose and why? Explain with a practical example.**

SQL is the best choice for structured data as it provides a fixed schema, strong relationships, and ACID compliance. It is ideal for applications like banking and enterprise resource planning, where data integrity and consistency are critical.

**2. A company wants to migrate from a relational database to a NoSQL database for better scalability. What challenges might they face? Discuss with an example.**

Migrating data to SQL requires schema normalization, ensuring ACID compliance, and restructuring queries. Performance optimization is necessary for handling large transactions, and indexing strategies must be refined to improve query efficiency.

**Section B: Advantages and Disadvantages of SQL/NoSQL**  
**3. You are designing an e-commerce website's database. Explain the advantages and disadvantages of using SQL vs. NoSQL in this scenario.**

SQL databases ensure data integrity, ACID compliance, and efficient transaction processing. They are ideal for managing customer orders, payment transactions, and inventory control. MySQL or PostgreSQL is commonly used to maintain structured product and customer records.

**4. A banking system requires high consistency and ACID compliance. Which database system (SQL or NoSQL) would you recommend? Justify your answer with a real-world use case.**

SQL is preferred in banking due to its strong consistency, ACID compliance, and secure transaction handling. It ensures that financial transactions are atomic and reliable. Oracle and MS SQL Server are widely used to maintain secure banking databases.

**Section C: Managing Databases**  
**5. You are a database administrator and need to perform routine maintenance on a production database. Describe at least three essential database management tasks you would perform.**

Regular backups prevent data loss, indexing optimizes query performance, and database normalization reduces redundancy. Performance tuning ensures efficient query execution, and access control secures data from unauthorized modifications.

**6. An online streaming service needs to optimize its database performance. What strategies can be used for effective database management in this case?**

Query optimization, indexing, and partitioning improve performance. Caching frequently accessed data reduces query load, and scheduled maintenance prevents system slowdowns. Using stored procedures and triggers enhances execution efficiency.

**Section D: Identifying System Databases in SQL Server**  
**7. List and describe the system databases in SQL Server. Provide one practical use case for each system database.**

Master stores system configurations, model is a template for new databases, msdb manages SQL jobs and backups, and tempdb handles temporary data storage. Tempdb is essential for managing temporary tables and query sorting.

**8. You have accidentally deleted a user database in SQL Server. Which system database would you use to recover it, and how?**

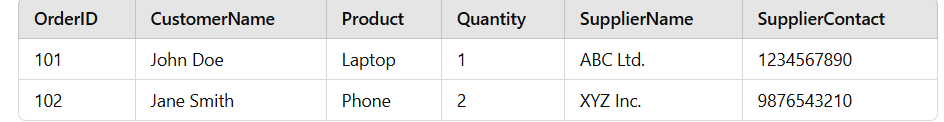
The msdb system database stores backup history and recovery options. Using the RESTORE DATABASE command from the last backup can efficiently recover a deleted database, ensuring minimal data loss.

**Section E: Normalization Forms (1NF, 2NF, 3NF, BCNF)  
9. Given the following unnormalized table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **OrderID** | **CustomerName** | **Product** | **Quantity** | **SupplierName** | **SupplierContact** |
| **101** | **John Doe** | **Laptop** | **1** | **ABC Ltd.** | **1234567890** |
| **102** | **Jane Smith** | **Phone** | **2** | **XYZ Inc.** | **9876543210** |

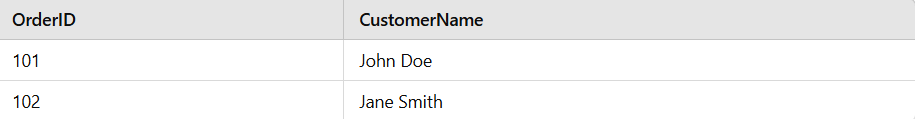
**Convert it to 1NF, 2NF, and 3NF with proper explanations.**

1NF:

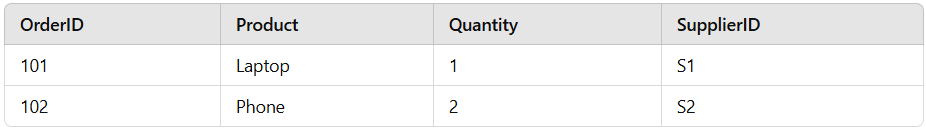


2NF:

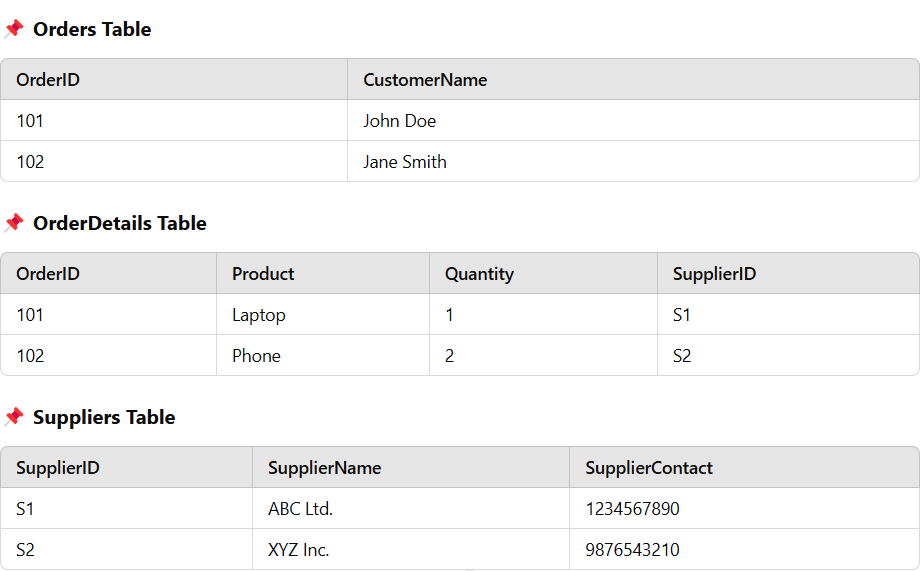
Orders Table



OrderDetails Table



3NF:



**10. A company is facing redundancy issues in their database. How would applying BCNF help reduce redundancy? Explain with an example.**

BCNF ensures that every determinant is a candidate key, eliminating redundancy and ensuring efficient database design. It prevents anomalies in relational databases, improving consistency and scalability.