

SQL 相关面试问题

1. 什么是数据库？数据库的作用是什么？

数据库 (Database, DB) 是一个用于存储、管理和组织数据的系统。它的主要作用包括：

- **数据存储**：结构化地存储大量数据。
- **数据管理**：支持数据的插入、更新、删除和查询。
- **数据共享**：多个用户或系统可以同时访问数据。
- **数据安全**：提供访问控制和数据备份。
- **数据一致性**：保证数据的完整性和一致性。

2. SQL 和 NoSQL 的区别是什么？

特性	SQL (关系型数据库)	NoSQL (非关系型数据库)
数据结构	表 (行、列)	文档、键值对、图、列族
模式 (Schema)	需要预定义表结构	结构灵活，无需固定模式
查询语言	SQL (结构化查询语言)	NoSQL (如 MongoDB 的 JSON 查询)
扩展性	垂直扩展 (scale-up)	水平扩展 (scale-out)
事务支持	强事务支持 (ACID)	一般支持最终一致性 (BASE)
适用场景	传统业务系统 (银行、电商)	海量数据、大数据 (社交媒体、日志存储)

3. RDBMS 和 NoSQL 数据库的主要类型及使用场景

RDBMS (关系型数据库)

- MySQL、PostgreSQL、SQL Server
- 适用场景：财务系统、企业 ERP、电商系统

NoSQL 数据库的主要类型

类型	示例	适用场景
键值存储 (Key-Value)	Redis、DynamoDB	缓存、会话管理
文档数据库 (Document)	MongoDB、CouchDB	内容管理系统、日志存储
列存储 (Column-Family)	HBase、Cassandra	大数据分析、分布式存储
图数据库 (Graph)	Neo4j、JanusGraph	社交网络、推荐系统

4. 常见的 SQL 数据类型有哪些？

数据类型	说明	示例
INT	整数类型	10, 255
DECIMAL(m, n)	定点数 (m 总位数, n 小数位)	99.99
CHAR(n)	固定长度字符串	'ABC'
VARCHAR(n)	变长字符串	'Hello'
TEXT	大文本数据	'This is a long text'
DATE	仅存日期	'2024-03-05'
DATETIME	日期 + 时间	'2024-03-05 10:00:00'
BOOLEAN	布尔值	TRUE, FALSE

5. 什么是主键 (Primary Key) 和外键 (Foreign Key)?

- **主键 (Primary Key)**: 唯一标识表中每一行的列, 不能重复且不能为空。
- **外键 (Foreign Key)**: 用于引用另一个表的主键, 建立表与表之间的关系。

sql

 Copy

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    Name VARCHAR(50)  
);  
  
CREATE TABLE Enrollments (  
    EnrollmentID INT PRIMARY KEY,  
    StudentID INT,  
    CourseID INT,  
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID)  
);
```

SQL 语法

7. 如何创建数据库和数据表?

sql

```
CREATE DATABASE School;  
USE School;
```

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY AUTO_INCREMENT,  
    Name VARCHAR(50) NOT NULL,  
    Age INT NOT NULL  
);
```


8. 如何使用 CREATE TABLE 创建一个包含主键和外键的表?

sql

```
CREATE TABLE Courses (  
    CourseID INT PRIMARY KEY AUTO_INCREMENT,  
    CourseName VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE Enrollments (  
    EnrollmentID INT PRIMARY KEY AUTO_INCREMENT,  
    StudentID INT,  
    CourseID INT,  
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID),  
    FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)  
);
```


9. 如何用 INSERT INTO 插入数据?

sql

```
INSERT INTO Students (Name, Age) VALUES ('Alice', 20);  
INSERT INTO Students (Name, Age) VALUES ('Bob', 22);
```

10. 如何查询某个表的数据?

sql

```
SELECT * FROM Students;  
SELECT Name, Age FROM Students;
```

11. 如何使用 WHERE 过滤查询结果?

sql

```
SELECT * FROM Students WHERE Age > 20;  
SELECT * FROM Students WHERE Name = 'Alice';
```


12. 如何使用 ORDER BY 对查询结果排序?

sql

```
SELECT * FROM Students ORDER BY Age ASC;  -- 按年龄升序  
SELECT * FROM Students ORDER BY Name DESC; -- 按姓名降序
```

13. 如何使用 GROUP BY 和 HAVING 进行数据聚合?

sql

 Copy

```
SELECT Age, COUNT(*) AS StudentCount FROM Students GROUP BY Age;  
SELECT Age, COUNT(*) AS StudentCount FROM Students GROUP BY Age HAVING COUNT(*) > 1;
```

14. 如何更新和删除数据?

sql



Cop

-- 更新

```
UPDATE Students SET Age = 23 WHERE Name = 'Alice';
```

-- 删除

```
DELETE FROM Students WHERE Name = 'Bob';
```


15. 如何使用 JOIN 连接多张表？ LEFT JOIN 和 INNER JOIN 的区别？

sql

-- INNER JOIN 只返回匹配的行

SELECT Students.Name, Courses.CourseName

FROM Students

INNER JOIN Enrollments ON Students.StudentID = Enrollments.StudentID

INNER JOIN Courses ON Enrollments.CourseID = Courses.CourseID;

-- LEFT JOIN 返回左表所有数据，右表匹配的行（若无匹配，则 NULL）

SELECT Students.Name, Courses.CourseName

FROM Students

LEFT JOIN Enrollments ON Students.StudentID = Enrollments.StudentID

LEFT JOIN Courses ON Enrollments.CourseID = Courses.CourseID;

16. 如何查询表中的重复数据?

sql

```
SELECT Name, COUNT(*)  
FROM Students  
GROUP BY Name  
HAVING COUNT(*) > 1;
```

24. 在 MySQL 中，AUTO_INCREMENT 和 UUID 作为主键的优缺点是什么？

类型	AUTO_INCREMENT	UUID
唯一性	自增，容易重复	全局唯一
性能	较快	计算开销大
存储	数值类型，占用空间小	字符串，占用空间大
适用场景	本地系统，单数据库	分布式系统，多数据库