数据库系统原理简答题汇总

有两个表 R(A)和 S(B),R(A)={1，3},S(B)={5，6}。定义如下触发器：CREATE TRIGGER my_tri AFTER INSERT ON R
FOR EACH ROW INSERT INTO S VALUES (NEW.A+2)
当执行完 INSERT INTO R(SELECT*FROM S)之后,分别写出 R 和 S 的结果。

答案：
R(A)={1，3，5，6},S(B)={5，6，7，8}

解析：
R(A)={1，3，5，6},S(B)={5，6，7，8}

数据库管理系统提供哪些对数据的统一管理和控制功能?
答案：
数据库管理系统具有对数据的统一管理和控制功能，主要包括数据的安全性、完整性、并发控制与故障恢复等，即数据库保护

解析：
数据库管理系统具有对数据的统一管理和控制功能，主要包括数据的安全性、完整性、并发控制与故障恢复等，即数据库保护

主要的逻辑数据模型有哪些?
答案：
主要的逻辑数据模型有层次模型、网状模型、关系模型、面向对象模型

解析：
主要的逻辑数据模型有层次模型、网状模型、关系模型、面向对象模型。
1.层次模型：数据库最早使用的数据模型。特点：有且仅有一个结点没有父结点，它称作根结点；其他结点有且仅有一个父结点。
2.网状模型：以网状结构表示实体与实体之间的联系。
3.关系模型：用二维表结构来表示实体及实体间联系的模型，并以二维表格的形式组织数据库中的数据。优点如下：关系模型是建立在严格的数学概念基础上的；关系模型的概念单一，统一用关系来表示实体以及实体之间的联系，对数据的检索和更新结果同样也是用关系来表示；关系模型的存取路径对用户透明，从而具有更高的数据独立性、更好的安全保密性，也简化了程序员的工作和数据库开发建立的工作。
4.面向对象模型：与数据库相结合所构成的数据模型称为面向对象模型

请列出 MySQL 中和表定义相关的四个 SQL 语句。

答案：
1.创建表：在 MySQL 中，可以使用 CREATE TABLE 语句创建表。
2.更新表：在 MySQL 中，可以使用 ALTER TABLE 语句来更改原有表的结构。
3.重命名表：除了 ALTER TABLE 语句，还可以直接用语句 RENAME TABLE 来更改表名，并可同时命名多个表。
4.删除表：如若需要删除数据库中已存在的表，可以通过使用 DROP TABLE 语句来实现。
5.查看表：（1）显示表的名称：在 MySQL 中，可以使用 SHOW TABIES 语句来显示指定数据库中存放的所有表名；（2）显示表的结构：在 MySQL 中，可以使用 SHOW COLUMNS 语句来显示指定数据表的结构

在 MySQL 中,定义外键时需要指定参照完整性的实现策略,除了 RESTRICT 外，还有其他哪两种含义不同的实现策略？
答案：
关键字“CASCADE”表示级联策略,即从被参照表中删除或更新记录行时,自动删除或更新参照表中匹配的记录行;关键字“SET NULL”表示置空策略,即当从被参照表中删除或更新记录行时,设置参照表中与之对应的外键列的值为 NULL,这个策略需要被参照表中的外键列没有声明限定词 NOT NULL

常见的 NoSQL 数据存储模型有哪些？
答案：
NoSQL 系统支持的数据存储模型通常有键值（Key-Value）模型、文档（Document）模型、列（Column）模型和图（Graph）模型等

解析：
NoSQL 系统支持的数据存储模型通常有键值（Key-Value）模型、文档（Document）模型、列（Column）模型和图（Graph）模型等。
（1）键值（Key-Value）存储。NoSQL 数据库采用最多的数据存储方式。适合通过主键进行查询或遍历。
（2）文档存储。适合存储系统日志等非结构化数据。可以通过复杂的查询条件来获取数据。
（3）列存储。比较适合对某一列进行随机查询处理。主要应用于需要处理大量数据的情况。
（4）图存储。图存储数据库是基于图理论构建的，使用结点、属性和边的概念

简述主属性和非主属性的区别。
答案：
关系中包含在任何一个候选码中的属性称为主属性或码属性，不包含在任何一个候选码中的属性称为非主属性或非码属性。
解析：
关系中包含在任何一个候选码中的属性称为主属性或码属性，不包含在任何一个候选码中的属性称为非主属性或非码属性。

请简述数据库应用软件设计与实现的基本步骤。

答案：

数据库应用软件的设计与开发过程可由需求分析、系统功能与数据库的设计、系统功能与数据库的实现、测试与维护等阶段构成。

解析：

以数据库的生命周期为演化主线，数据库应用软件的设计与开发过程可由需求分析、系统功能与数据库的设计、系统功能与数据库的实现、测试与维护等阶段构成。

简述系统测试与维护的作用。

答案：

完成系统的实现工作之后，在正式交付用户使用之前，需要对所开发的系统进行必要的测试，验证其是否满足用户的功能要求，并根据测试的结果，以及用户的反馈意见，对该系统进行进一步的修改、完善和维护工作。

解析：

完成系统的实现工作之后，在正式交付用户使用之前，需要对所开发的系统进行必要的测试，验证其是否满足用户的功能要求，并根据测试的结果，以及用户的反馈意见，对该系统进行进一步的修改、完善和维护工作。

请叙述文章《第三代数据库系统宣言》中指出第三代数据库系统应具有的基本特征。

答案：

（1）第三代数据库系统应支持数据管理、对象管理和知识管理。

（2）第三代数据库系统必须保持或继承第二代数据库系统的技术。

（3）第三代数据库系统必须对其他系统开放。

解析：

1990 年高级 DBMS 功能委员会发表了《第三代数据库系统宣言》的文章，提出了第三代数据库系统应具有的三个特征。

（1）第三代数据库系统应支持数据管理、对象管理和知识管理。

（2）第三代数据库系统必须保持或继承第二代数据库系统的技术。

（3）第三代数据库系统必须对其他系统开放。

简述多表连接查询中的内连接方式。

答案：

内连接是一种最常用的连接类型，它是通过在查询中设置连接条件的方式，来移除查询结果集中某些数据行之后的交叉连接。

解析：

内连接是一种最常用的连接类型，它是通过在查询中设置连接条件的方式，来移除查询结果

集中某些数据行之后的交叉连接。具体而言，内连接就是利用条件判断表达式中的比较运算符来组合两张表中的记录，其目的是为了消除交叉连接中的某些数据行。

数据库的生命周期可分为哪些阶段？

答案：

从数据库演变过程的角度来看，数据库的生命周期可分为两个阶段，分别是数据库分析与设计阶段、数据库实现与操作阶段。

解析：

从数据库演变过程的角度来看，数据库的生命周期可分为两个阶段，分别是数据库分析与设计阶段、数据库实现与操作阶段。 其中，数据库分析与设计阶段包括需求分析、概念设计、逻辑设计和物理设计四个环节；数据库实现与操作阶段包括数据库的实现、操作与监督、修改与调整三个子阶段。

数据库实现与操作阶段包括哪些子阶段？

答案：

数据库实现与操作阶段包括数据库的实现、操作与监督、修改与调整三个子阶段。

解析：

从数据库演变过程的角度来看，数据库的生命周期可分为两个阶段，分别是数据库分析与设计阶段、数据库实现与操作阶段。 其中，数据库分析与设计阶段包括需求分析、概念设计、逻辑设计和物理设计四个环节；数据库实现与操作阶段包括数据库的实现、操作与监督、修改与调整三个子阶段。

简述左外连接和右外连接的区别。

答案：

1.左外连接：也称左连接。以左表为基表，在 FROM 子句中使用关键字“LEFT OUTER JOIN”或关键字“LEFT JOIN”来连接两张表。

2.右外连接：也称右连接。以右表为基表，在 FROM 子句中使用关键字“RIGHT OUTER JOIN”或关键字“RIGHT JOIN”来连接两张表。

解析：

1.左外连接：也称左连接。以左表为基表，在 FROM 子句中使用关键字“LEFT OUTER JOIN”或关键字“LEFT JOIN”来连接两张表。

2.右外连接：也称右连接。以右表为基表，在 FROM 子句中使用关键字“RIGHT OUTER JOIN”或关键字“RIGHT JOIN”来连接两张表。

简述内连接方式中的自连接的含义。

答案：
自连接：在 FROM 子句中使用关键字“INNER JOIN”或“JOIN”连接两张表时，可以将一个表与它自身进行连接，这种连接方式称为自连接。
解析：
自连接：在 FROM 子句中使用关键字“INNER JOIN”或“JOIN”连接两张表时，可以将一个表与它自身进行连接，这种连接方式称为自连接。

简述内连接方式中的非等值连接。
答案：
非等值连接：在 FROM 子句中使用关键字“INNER JOIN”或“JOIN”连接两张表时，如若在 ON 子句的连接条件中使用除运算符“=”之外的其他比较运算符，即进行不相等性测试，则此连接方式称为非等值连接，也称为不等连接。
解析：
非等值连接：在 FROM 子句中使用关键字“INNER JOIN”或“JOIN”连接两张表时，如若在 ON 子句的连接条件中使用除运算符“=”之外的其他比较运算符，即进行不相等性测试，则此连接方式称为非等值连接，也称为不等连接。

简述多表连接查询中内连接的三种使用情形。
答案：
1.等值连接：在 FROM 子句中使用关键字“INNER JOIN”或“JOIN”连接两张表时，如若在 ON 子句的连接条件中使用运算符“=”（即等号），即进行相等性测试，则此连接方式称为等值连接，也称为相等连接。
2.非等值连接：在 FROM 子句中使用关键字“INNER JOIN”或“JOIN”连接两张表时，如若在 ON 子句的连接条件中使用除运算符“=”之外的其他比较运算符，即进行不相等性测试，则此连接方式称为非等值连接，也称为不等连接。
3.自连接：在 FROM 子句中使用关键字“INNER JOIN”或“JOIN”连接两张表时，可以将一个表与它自身进行连接，这种连接方式称为自连接。
解析：
1.等值连接：在 FROM 子句中使用关键字“INNER JOIN”或“JOIN”连接两张表时，如若在 ON 子句的连接条件中使用运算符“=”（即等号），即进行相等性测试，则此连接方式称为等值连接，也称为相等连接。
2.非等值连接：在 FROM 子句中使用关键字“INNER JOIN”或“JOIN”连接两张表时，如若在 ON 子句的连接条件中使用除运算符“=”之外的其他比较运算符，即进行不相等性测试，则此连接方式称为非等值连接，也称为不等连接。
3.自连接：在 FROM 子句中使用关键字“INNER JOIN”或“JOIN”连接两张表时，可以将一个表与它自身进行连接，这种连接方式称为自连接。

说明 DROP TABLE 语句和 DELETE 语句的联系和区别。

答案：

（1）DROP TABLE 语句和 DELETE 语句都可以从基本表中删除元组。

（2）DROP TABLE 语句不但删除表中的全部元组，还删除整个关系表结构。

（3）DELETE 语句可以根据条件删除表中部分元组。

解析：

DROP TABLE 语句可以同时删除多个表（包括临时表），但操作者必须拥有该命令的权限；当表被删除时，其中存储的数据和分区信息均会被删除，所以使用该语句须格外小心，但操作者在该表上的权限并不会自动被删除。

在 MySQL 中，可以使用 DELETE 语句删除表中的一行或多行数据。其中，可选项 WHERE 子句表示为删除操作限定删除条件，从而删除特定的行，若省略 WHERE 子句，则表示删除该表中的所有行，但表的定义扔在数据字典中，即 DELETE 语句删除的是表中的数据，而不是关于表的定义。

故总结有：

（1）DROP TABLE 语句和 DELETE 语句都可以从基本表中删除元组。

（2）DROP TABLE 语句不但删除表中的全部元组，还删除整个关系表结构。

（3）DELETE 语句可以根据条件删除表中部分元组。

简述逻辑结构设计的主要步骤。

答案：

1）模型转换是指将概念模型等价地转换为特定 DBMS 支持的关系模型、网状模型或层次模型表示。

2)子模式设计的目标是抽取或导出模式的子集，以构造不同用户使用的局部数据逻辑结构。

3）编制应用程序设计说明的目的是为可实际运行的应用程序设计提供依据与指导，并作为设计评价的基础。

4）设计评价的任务是分析并检验模式及子模式的正确性与合理性。

解析：

逻辑结构设计的步骤：

1）模型转换是指将概念模型等价地转换为特定 DBMS 支持的关系模型、网状模型或层次模型表示。

2)子模式设计的目标是抽取或导出模式的子集，以构造不同用户使用的局部数据逻辑结构。

3）编制应用程序设计说明的目的是为可实际运行的应用程序设计提供依据与指导，并作为设计评价的基础。

4）设计评价的任务是分析并检验模式及子模式的正确性与合理性。

简述集合运算中“并”运算的定义。

答案：

并：假设有两个关系 R1 和 R2,R1 和 R2 的并运算产生一个新关系 R3。R3 是由属于关系 R1 或 R2 的所有不同元组所组成，记为 R3=R1∪R2。

解析：

并：假设有两个关系 R1 和 R2,R1 和 R2 的并运算产生一个新关系 R3。R3 是由属于关系 R1 或 R2 的所有不同元组所组成，记为 R3=R1∪R2。

简述用户定义完整性约束的含义。

答案：

用户定义的完整性约束是针对某一应用环境的完整性约束条件，它反映了某一具体应用所涉及的数据应满足的要求。

解析：

用户定义的完整性约束是针对某一应用环境的完整性约束条件，它反映了某一具体应用所涉及的数据应满足的要求。

什么是数据库的完整性?

答案：

数据库完整性是指数据库中数据的正确性和相容性。

解析：

数据库完整性是指数据库中数据的正确性和相容性。例如，人的性别只能是“男”或“女”，人的年龄应该是 0 到 150 岁之间的数字，每个人的身份证号必须唯一等。

请简述数据完整性约束的含义。

答案：

数据完整性约束是为了防止数据库中存在不符合语义的数据，为了维护数据的完整性，DBMS 必须提供一种机制来检查数据库中的数据，以判断其是否满足语义规定的条件。这些加在数据库数据之上的语义约束条件就是数据完整性约束。

解析：

数据完整性约束是为了防止数据库中存在不符合语义的数据，为了维护数据的完整性，DBMS 必须提供一种机制来检查数据库中的数据，以判断其是否满足语义规定的条件。这些加在数据库数据之上的语义约束条件就是数据完整性约束，而 DBMS 检查数据是否满足完整性约束条件的机制就称为完整性检查。

简述完整性约束条件中列级约束包括的内容。

答案：

（1） 对数据类型的约束，其包括数据类型、长度、精度等。

（2） 对数据格式的约束。

（3） 对取值范围或取值集合的约束。

（4） 对空值的约束。

解析：

列级约束主要指对列的类型、取值范围、精度等的约束，具体包括如下内容。

（1） 对数据类型的约束，其包括数据类型、长度、精度等。

（2） 对数据格式的约束。

（3） 对取值范围或取值集合的约束。

（4） 对空值的约束。

数据中外码（或外键）的含义是什么？

答案：

当关系中的某个属性不是这个关系的主码或候选码，而是另一关系的主码时，称该属性为这个关系的外码或外键。

解析：

当关系中的某个属性不是这个关系的主码或候选码，而是另一关系的主码时，称该属性为这个关系的外码或外键。

简述全码（或全键）的含义。

答案：

一个关系模式的所有属性集合是这个关系的主码或主键，则称这样的主码或主键为全码或全键。

解析：

一个关系模式的所有属性集合是这个关系的主码或主键，则称这样的主码或主键为全码或全键。

简述数据库数据完整性的含义。

答案：

数据库的数据完整性是指数据库中数据的正确性、相容性和一致性。这是一种语义概念，包括两个方面：与现实世界中应用需求的数据的正确性、相容性和一致性；数据库内数据之间的正确性、相容性和一致性。

解析：

数据库的数据完整性是指数据库中数据的正确性、相容性和一致性。这是一种语义概念，包括两个方面：与现实世界中应用需求的数据的正确性、相容性和一致性；数据库内数据之间的正确性、相容性和一致性。

简述触发器的概念。

答案：

触发器是用户定义在关系表上的一类由事件驱动的数据库对象，也是一种保证数据完整性的方法。

解析：
触发器是用户定义在关系表上的一类由事件驱动的数据库对象，也是一种保证数据完整性的方法。触发器一旦定义，无须用户调用，任何对表的修改操作均为数据库服务器自动激活相应的触发器。

数据库设计的目标是什么？
答案：
数据库设计的目标：满足应用功能需求和良好的数据库性能。
解析：
数据库设计的目标：满足应用功能需求和良好的数据库性能。

简述存储函数和存储过程的区别。
答案：
1、存储函数不能拥有输出参数，这是因为存储函数自身就是输出参数；而存储过程可以拥有输出参数。
2、可以直接对存储函数进行调用，且不需要使用 CALL 语句；而对存储过程的调用，需要使用 CALL 语句。
3、存储函数中必须包含一条 RETURN 语句，而这条特殊的 SQL 语句不允许包含于存储过程中。
解析：
存储函数和存储过程的区别：
1、存储函数不能拥有输出参数，这是因为存储函数自身就是输出参数；而存储过程可以拥有输出参数。
2、可以直接对存储函数进行调用，且不需要使用 CALL 语句；而对存储过程的调用，需要使用 CALL 语句。
3、存储函数中必须包含一条 RETURN 语句，而这条特殊的 SQL 语句不允许包含于存储过程中。

请说明实体、属性、码或键的概念。
答案：
实体：客观存在并可相互区别的事物称为实体。
属性：实体所具有的某种特性称为实体的属性。
码或键：可唯一标识实体的属性集称为码或键。
解析：
信息世界涉及的基本概念：
实体：客观存在并可相互区别的事物称为实体。实体可以是实际的事物，也可以是抽象的概念或联系。

属性：实体所具有的某种特性称为实体的属性。一个实体可以由多个属性来描述。
码或键：可唯一标识实体的属性集称为码或键。例如，学号是学生实体的码或键。
实体型：具有相同属性的实体必然具有共同的特征和性质。
实体集：同型实体的集合称为实体集。
联系：实体内部的联系通常是指实体各属性之间的联系，实体之间的联系是指不同实体之间的联系。

请说明数据库的定义。
答案：
数据库（DB）是指长期储存在计算机中的有组织的、可共享的数据集合，且数据库中的数据按一定的数据模型组织、描述和存储，具有较小的冗余度、较高的数据独立性，系统易于扩展，并可以被多个用户共享。
解析：
数据库（DB）是指长期储存在计算机中的有组织的、可共享的数据集合，且数据库中的数据按一定的数据模型组织、描述和存储，具有较小的冗余度、较高的数据独立性，系统易于扩展，并可以被多个用户共享。数据库中存储的数据具有永久存储、有组织和可共享三个基本特点。

关系模型中常用的关系操作是什么？
答案：
关系模型中常用的关系操作包括查询（Query）操作和插入(Insert)、删除（Delete）、修改（Update）操作两大部分。
解析：
关系模型中常用的关系操作包括查询（Query）操作和插入(Insert)、删除（Delete）、修改（Update）操作两大部分。

关系操作中的查询操作有哪些？
答案：
查询操作分为：选择、投影、连接、除、并、差、交、笛卡尔积等。
解析：
关系模型中常用的关系操作包括查询（Query）操作和插入(Insert)、删除（Delete）、修改（Update）操作两大部分。
查询操作分为：选择、投影、连接、除、并、差、交、笛卡尔积等。

在关系数据库中，表有哪些部分组成？
答案：

表也称为关系，是一个二维的数据结构，它由表名、构成表的各个列及若干行数据组成。
解析：
表也称为关系，是一个二维的数据结构，它由表名、构成表的各个列及若干行数据组成。每个表有一个唯一的表名，表中每一行数据描述一条具体的记录值。

简述传统的集合运算有哪些？
答案：
传统的集合运算是二目运算，它将关系看成元组的集合，其运算是从关系的“水平”方向，即行的角度来进行，具体有并、差、交、笛卡尔积 4 种运算。
解析：
传统的集合运算是二目运算，它将关系看成元组的集合，其运算是从关系的“水平”方向，即行的角度来进行，具体有并、差、交、笛卡尔积 4 种运算。

请简述关系数据库的基本特征。
答案：
关系数据库的基本特征是使用关系数据模型组织数据，这种思想源于数学。
解析：
关系数据库的基本特征是使用关系数据模型组织数据，这种思想源于数学。

简述第三范式的定义。
答案：
第三范式定义： 设 R 为任一给定关系，若 R 为 2NF，且其每一个非主属性都不传递函数依赖于候选关键字，则 R 为第三范式。
解析：
第三范式定义： 设 R 为任一给定关系，若 R 为 2NF，且其每一个非主属性都不传递函数依赖于候选关键字，则 R 为第三范式。
判分设置：AI 判分
得分点 1（）：设 R 为任一给定关系，若 R 为 2NF，且其每一个非主属性都不传递函数依赖于候选关键字，则 R 为第三范式。

简述存储过程的基本概念。
答案：
存储过程是一组为了完成某项特定功能的 SQL 语句集，其实质上就是一段存储在数据库中的代码，它可以由声明式的 SQL 语句和过程式 SQL 语句组成。
解析：

存储过程是一组为了完成某项特定功能的 SQL 语句集，其实质上就是一段存储在数据库中的代码，它可以由声明式的 SQL 语句（如 CREATE、UPDATE 和 SELECT 等语句）和过程式 SQL 语句（如 IF…THEN…ELSE 控制结构语句）组成。

简述索引存在的弊端有哪些。

答案：

索引存在的弊端如下：

1.索引是以文件的形式存储的，DBMS 会将一个表的所有索引保存在同一个索引文件中，索引文件需要占用磁盘空间。

2.索引在提高查询速度的同时，却会降低更新表的速度。

解析：

索引存在的弊端如下：

1.索引是以文件的形式存储的，DBMS 会将一个表的所有索引保存在同一个索引文件中，索引文件需要占用磁盘空间。

2.索引在提高查询速度的同时，却会降低更新表的速度。

请简述在数据库的操作中使用存储过程的优点。

答案：

1、可增强 SQL 语言的功能和灵活性 。

2、良好的封装性 。

3、高性能 。

4、可减少网络流量 。

5、存储过程可作为一种安全机制来确保数据库的安全性和数据的完整性 。

解析：

1、可增强 SQL 语言的功能和灵活性 。

2、良好的封装性 。

3、高性能 。

4、可减少网络流量 。

5、存储过程可作为一种安全机制来确保数据库的安全性和数据的完整性 。

请简述存储过程与存储函数的区别。

答案：

存储函数和存储过程的区别：

1、 存储函数不能拥有输出参数，这是因为存储函数自身就是输出参数；而存储过程可以拥有输出参数。

2、可以直接对存储函数进行调用，且不需要使用 CALL 语句；而对存储过程的调用，需要使用 CALL 语句。

3、 存储函数中必须包含一条 RETURN 语句，而这条特殊的 SQL 语句不允许包含于存储过程中。

解析：

存储函数和存储过程的区别：

1、 存储函数不能拥有输出参数，这是因为存储函数自身就是输出参数；而存储过程可以拥有输出参数。

2、可以直接对存储函数进行调用，且不需要使用 CALL 语句；而对存储过程的调用，需要使用 CALL 语句。

3、 存储函数中必须包含一条 RETURN 语句，而这条特殊的 SQL 语句不允许包含于存储过程中。

简述使用视图的优点。

答案：

1）集中分散数据；

2）简化查询语句；

3）重用 SQL 语句；

4）保护数据安全；

5）共享所需数据；

6）更改数据格式。

解析：

使用视图有如下优点；

1）集中分散数据。当用户所需的数据分散在数据库多个表中时，通过定义视图可以将这些数据集中在一起，以方便用户对分散数据的集中查询与处理。

2）简化查询语句。通过定义视图可为用户屏蔽数据库的复杂性，使其不必详细了解数据库中复杂的表结构和表连接，因而能简化用户对数据库的查询语句。

3）重用 SQL 语句。视图提供的是一种对查询操作的封装，它本身不包含数据，其所呈现的数据是根据视图的定义从基本表中检索出来的，如若基本表中的数据被新增或更改，视图所呈现的则是更新后的数据。

4）保护数据安全。通过只授予用户使用视图的权限，而不是具体指定使用表的权限，来保护基础数据的安全性。

5）共享所需数据。通过使用视图，每个用户不必都定义和存储自己所需的数据，可以共享数据库中的数据，从而同样的数据只需存储一次。

6）更改数据格式。通过使用视图，可以重新格式化检索出的数据，并组织输出到其他应用程序中去。

简述 MySQL 中候选键与主键之间的区别。

答案：

（1）一个表中只能创建一个主键，但可以定义若干个候选键。

（2）定义主键约束时，系统会自动产生 PRIMARY KEY 索引，而定义候选键约束时，系统自动产生 UNIQUE 索引。

解析：

MySQL 中候选键与主键之间存在以下几点区别。

（1）一个表中只能创建一个主键，但可以定义若干个候选键。

（2）定义主键约束时，系统会自动产生 PRIMARY KEY 索引，而定义候选键约束时，系统自

动产生 UNIQUE 索引。

简述外键声明的两种方式。

答案：

（1） 在表中某个列的属性定义后直接加上“reference_definition”语法项。

（2） 在表中所有列的属性定义后添加“FOREIGN KEY（index_col_name，…）reference_definition”子句的语法项。

解析：

在 MySQL 中，参照完整性是通过在创建表（CREATE TABLE）或更新表（ALTER TABLE）的同时定义一个外键声明来实现的。其中，外键声明有下列两种方式。

（1） 在表中某个列的属性定义后直接加上“reference_definition”语法项。

（2） 在表中所有列的属性定义后添加“FOREIGN KEY（index_col_name，…）reference_definition”子句的语法项。

简述主键约束在 CREATE TABLE 或 ALTER TABLE 语句中实现的两种方式。

答案：

（1）一种是作为列的完整性约束，此时只需在表中某个列的属性定义后加上关键字“PRIMARY KEY”即可。

（2）一种是作为表的完整性约束，需要在表中所有列的属性定义后添加一条 PRIMARY KEY（index_col_name，…）格式的句子。

解析：

主键约束可以在 CREATE TABLE 或 ALTER TABLE 语句中使用关键字“PRIMARY KEY”来实现，其方式有两种。

（1）一种是作为列的完整性约束，此时只需在表中某个列的属性定义后加上关键字“PRIMARY KEY”即可。

（2）一种是作为表的完整性约束，需要在表中所有列的属性定义后添加一条 PRIMARY KEY（index_col_name，…）格式的句子。

简述关系数据语言的分类以及其共同特点。

答案：

关系数据语言可以分为三类：关系代数语言。关系演算语言以及兼具两者双重特点的语言。共同特点是：语言具有完备的表达能力，是非过程化的集合操作语言，功能强，能够独立使用也可以嵌入高级语言中使用。

解析：

关系数据语言可以分为三类：关系代数语言。关系演算语言以及兼具两者双重特点的语言。共同特点是：语言具有完备的表达能力，是非过程化的集合操作语言，功能强，能够独立使用也可以嵌入高级语言中使用。

关系数据库对关系的限定有哪些具体要求？

答案：

关系数据库对关系是有限定的，具体要求如下：

1.每一个属性都是不可分解的。

2.每一个关系仅仅有一种关系模式。

3.每一个关系模式中的属性必须命名，在同一个关系模式中，属性名必须是不同的。

4.同一个关系中不允许出现候选码或候选键值完全相同的元组。

5.在关系中元组的顺序是无关紧要的，可以任意交换。

6.在关系中属性的顺序是无关紧要的，可以任意交换。

解析：

关系数据库对关系是有限定的，具体要求如下：

1.每一个属性都是不可分解的。

2.每一个关系仅仅有一种关系模式。

3.每一个关系模式中的属性必须命名，在同一个关系模式中，属性名必须是不同的。

4.同一个关系中不允许出现候选码或候选键值完全相同的元组。

5.在关系中元组的顺序是无关紧要的，可以任意交换。

6.在关系中属性的顺序是无关紧要的，可以任意交换。

简述关系数据库中关系的三种类型以及其含义。

答案：

关系可以有三种类型，即基本关系、查询表和视图表。

基本关系通常又称为基本表或基表，是实际存在的表，它是实际存储数据的逻辑表示；查询表是查询结果对应的表；视图表是由基本表或其他视图表导出的表，是虚表，不对应实际存储的数据。

解析：

关系可以有三种类型，即基本关系、查询表和视图表。

基本关系通常又称为基本表或基表，是实际存在的表，它是实际存储数据的逻辑表示；查询表是查询结果对应的表；视图表是由基本表或其他视图表导出的表，是虚表，不对应实际存储的数据。

请简述什么是参照完整性约束。

答案：

参照完整性约束就是定义外码和主码之间的引用规则，它是对关系间引用数据的一种限制。描述参照完整性定义：若属性 F 是基本关系 R 的外码，它与基本关系 S 的主码 K 相对应，则对于 R 中每个元组在 F 上的值只允许两种可能，即要么取空值，要么等于 S 中某个元组的主码值。

解析：

参照完整性约束就是定义外码和主码之间的引用规则，它是对关系间引用数据的一种限制。描述参照完整性定义：若属性 F 是基本关系 R 的外码，它与基本关系 S 的主码 K 相对应，则

对于 R 中每个元组在 F 上的值只允许两种可能，即要么取空值，要么等于 S 中某个元组的主码值。

数据库的并发操作会带来哪些问题?

答案：

数据库的并发操作会带来的问题包括：丢失更新，读“脏”数据，不可重复读。

解析：

当多个事务交错执行时，可能出现不一致问题，这也称为并发操作问题，典型的有如下三种：丢失更新、不可重复读和读“脏”数据。

封锁可能引起哪些问题?

答案：

封锁带来的一个重要问题是可能引起“活锁”和“死锁”。

在并发事务处理过程中，由于锁会使一事务处于等待状态而调度其他事务处理，因而该事务可能会因优先级低而永远等待下去，这种现象称为“活锁”。活锁问题的解决与调度算法有关，一种最简单的办法是“先来先服务”。

两个以上事务循环等待被同组中另一事务锁住的数据单元的情形，称为“死锁”。

解析：

封锁带来的一个重要问题是可能引起活锁和死锁。

在并发事务处理过程中，由于锁会使一事务处于等待状态而调度其他事务处理，因而该事务可能会因优先级低而永远等待下去，这种现象称为“活锁”。活锁问题的解决与调度算法有关，一种最简单的办法是“先来先服务”。

两个以上事务循环等待被同组中另一事务锁住的数据单元的情形，称为“死锁”。

需求分析的目标是什么？

答案：

目标是了解与分析用户的信息及应用处理的要求，并将结果按一定格式整理而形成需求分析报告。

解析：

需求分析的目标是了解与分析用户的信息及应用处理的要求，并将结果按一定格式整理而形成需求分析报告。

请简述什么是数据库行为设计？

答案：

数据库行为设计：是确定数据库用户的行为和动作，而用户的行为和动作是对数据库的操作，它们通常是通过应用程序来实现的。

解析：

数据库行为设计是确定数据库用户的行为和动作，而用户的行为和动作是对数据库的操作，它们通常是通过应用程序来实现的。

什么是良好的数据库性能？

答案：

良好的数据库性能：主要是指对数据的高效率存取和空间的节省，并具有良好的数据共享性、完整性、一致性及安全保密性。

解析：

良好的数据库性能：主要是指对数据的高效率存取和空间的节省，并具有良好的数据共享性、完整性、一致性及安全保密性。

简述封锁的基本思想。

答案：

需要时，事务通过向系统请求对它所希望的数据对象加锁，以确保它不被非预期改变。

解析：

封锁是最常用的并发控制技术，它的基本思想是：需要时，事务通过向系统请求对它所希望的数据对象（如数据库中的记录）加锁，以确保它不被非预期改变。

请列举说明会造成数据库运行事务异常中断的因素。（至少 4 个）

答案：

（1） 计算机硬件故障
（2） 计算机软件故障
（3） 病毒
（4） 人为误操作
（5） 自然灾害
（6） 盗窃

解析：

会造成数据库运行事务异常中断的因素可能是：

（1） 计算机硬件故障
（2） 计算机软件故障
（3） 病毒
（4） 人为误操作
（5） 自然灾害
（6） 盗窃

请简述逻辑结构设计的任务是什么？

答案：

逻辑结构设计的任务是把在概念结构设计产生的概念模型转换为具体的 DBMS 所支持的逻辑数据模型，也就是导出特定的 DBMS 可以处理的数据库逻辑结构。

解析：

逻辑结构设计的任务是把在概念结构设计产生的概念模型转换为具体的 DBMS 所支持的逻

辑数据模型，也就是导出特定的 DBMS 可以处理的数据库逻辑结构。

请简述局部变量与用户变量之间的区别。

答案：

两者的区别是：局部变量声明时，在其前面没有使用@符号，并且它只能被声明它的 BEGIN…END 语句块中的语句所使用；而用户变量在声明时，会在其名称前面使用@符号，同时已声明的用户变量存在于整个会话之中。

解析：

局部变量不同于用户变量，两者的区别是：局部变量声明时，在其前面没有使用@符号，并且它只能被声明它的 BEGIN…END 语句块中的语句所使用；而用户变量在声明时，会在其名称前面使用@符号，同时已声明的用户变量存在于整个会话之中。

简述主键的含义。

答案：

主键是一种唯一性索引。创建主键时，必须指定关键字 PRIMARY KEY，且不能有空值。

解析：

主键是一种唯一性索引。创建主键时，必须指定关键字 PRIMARY KEY，且不能有空值。

数据更新操作有哪几种，在 SQL 中分别对应哪三类语句？

答案：

数据更新操作：向表中添加若干行数据、修改表中的数据和删除表中的若干行数据。

对应三类语句：插入数据、修改数据和删除数据

解析：

数据更新操作：向表中添加若干行数据、修改表中的数据和删除表中的若干行数据。

对应三类语句：插入数据（INSERT）、修改数据（UPDATE）和删除数据（DELETE）

简述关系数据结构中，参照关系和被参照关系的含义。

答案：

参照关系也称为从关系，被参照关系也称为主关系，它们是指以外码相关联的两个关系。以外码作为主码的关系称为被参照关系；外码所在的关系称为参照关系。被参照关系与参照关系是通过外码相联系的，这种联系通常是一对多的联系。

解析：

参照关系也称为从关系，被参照关系也称为主关系，它们是指以外码相关联的两个关系。以外码作为主码的关系称为被参照关系；外码所在的关系称为参照关系。被参照关系与参照关系是通过外码相联系的，这种联系通常是一对多的联系。

简述数据库中候选码或候选键的含义。

答案：
如果在关系的一个码或键中，不能从中移去任何一个属性，否则它就不是这个关系的码或键。则称这样的码或键为该关系的候选码或候选键。
解析：
如果在关系的一个码或键中，不能从中移去任何一个属性，否则它就不是这个关系的码或键。则称这样的码或键为该关系的候选码或候选键。一个关系的候选码或候选键是这个关系的最小超码或超键。

什么是数据冗余?
答案：
数据冗余是指同一数据被反复存储的情况。
解析：
数据冗余是指同一数据被反复存储的情况。例如，在这个供应商关系模式中，一个供应商每供应一种货物，其地址就要重复一次，若该供应商可供应 1000 种货物，则其地址就要被反复存储 1000 次。

什么是完全函数依赖?
答案：
设 R 为任一给定关系，X、Y 为其属性集，若 X→Y，且对 X 中的任何真子集 X′ 都有 X′ Y，则称 Y 完全函数依赖于 X。
解析：
完全函数依赖 ：
定义 2.2 设 R 为任一给定关系，X、Y 为其属性集，若 X→Y，且对 X 中的任何真子集 X′ 都有 X′ Y，则称 Y 完全函数依赖于 X。

请简述局部信息结构设计的步骤?
答案：
局部信息结构设计的步骤包括：确定局部范围；选择实体；选择实体关键字；确定实体间联系；确定实体的属性。
解析：
局部信息结构设计的步骤包括：确定局部范围；选择实体；选择实体关键字；确定实体间联系；确定实体的属性。

简述码或键的含义。
答案：
如果在一个关系中，存在这样的属性，使得在该关系的任何一个关系状态中的两个元组，在该属性上值的组合都不相同，即这些属性的值都能用来唯一标识该关系的元组，则称这些属性为该关系的码或键。

解析：
如果在一个关系中，存在这样的属性（或属性组），使得在该关系的任何一个关系状态中的两个元组，在该属性（或属性组）上值的组合都不相同，即这些属性（或属性组）的值都能用来唯一标识该关系的元组，则称这些属性（或属性组）为该关系的码或键

什么是数据仓库？
答案：
数据仓库是面向主题的、集成的、稳定的、随时间变化的数据集合，用以支持管理决策的过程。
解析：
1992 年数据仓库概念的创始人 W.H.Inmon 在其《Building the Data Warehouse》一书中定义了数据仓库的概念：数据仓库是面向主题的、集成的、稳定的、随时间变化的数据集合，用以支持管理决策的过程。

在数据仓库中，分割是什么意思？
答案：
分割是将数据分散到各自的物理单元中，以便能分别处理，以提高数据处理的效率。数据分割后的单元称为切片。
解析：
数据仓库三个常用的重要概念，即粒度、分割和维。
分割：是将数据分散到各自的物理单元中，以便能分别处理，以提高数据处理的效率。数据分割后的单元称为切片。

什么是数据挖掘？
答案：
数据挖掘是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中发现并提取隐藏在其中的、人们事先不知道的、但又是潜在有用的信息和知识的一种技术。
解析：
数据挖掘是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中发现并提取隐藏在其中的、人们事先不知道的、但又是潜在有用的信息和知识的一种技术。它又被称为数据库中的知识发现，其与数据库、数据统计、机器学习、模式识别、模糊教学等诸多技术相关。

简述数据挖掘具备的功能。
主知识点： 8.2.2、二、数据挖掘技术
答案：
（1）概念描述

（2）关联分析
（3）分类与预测
（4）聚类
（5）孤立点检测
（6）趋势和演变分析
解析：
数据挖掘具备的几种功能。
（1）概念描述
（2）关联分析
（3）分类与预测
（4）聚类
（5）孤立点检测
（6）趋势和演变分析

简述在实际使用中，数据挖掘的过程。
答案：
（1）确定业务对象
（2）数据的选择
（3）数据的预处理
（4）建模
（5）模型评估
（6）模型部署
解析：
在实际使用中，数据挖掘的过程通常由以下六个步骤构成。
（1）确定业务对象
（2）数据的选择
（3）数据的预处理
（4）建模
（5）模型评估
（6）模型部署

简述大数据的特征。
答案：
（1）数据量巨大，即大量化。
（2）数据种类繁多，即多样化。
（3）处理速度快，即快速化。
（4）价值密度低。
解析：
大数据的特征。
（1）数据量巨大，即大量化。

（2）数据种类繁多，即多样化。
（3）处理速度快，即快速化。
（4）价值密度低。

在数据仓库中，粒度是什么意思？

答案：
粒度是指数据仓库的数据单位中保存数据的细化或综合程度的级别，细化程度越高，粒度级就越小，相反地，细化程度越低，粒度级就越大。
解析：
数据仓库三个常用的重要概念，即粒度、分割和维。
粒度：指数据仓库的数据单位中保存数据的细化或综合程度的级别，细化程度越高，粒度级就越小，相反地，细化程度越低，粒度级就越大。

简述在数据仓库中，维的含义。

答案：
维是人们观察数据的特定角度，是考虑问题时的一类属性。此类属性的集合构成了一个维度，例如时间维、产品维等。
解析：
数据仓库三个常用的重要概念，即粒度、分割和维。
维：是人们观察数据的特定角度，是考虑问题时的一类属性。此类属性的集合构成了一个维度，例如时间维、产品维等。

简述常规文件系统与 HDFS 的不同。

答案：
HDFS 与常规文件系统不同的是，它以大粒度数据块的方式存储文件，从而减少了元数据的数量，这些数据块则通过随机方式选择不同的节点并存储在各个地方。
解析：
HDFS 与常规文件系统不同的是，它以大粒度数据块的方式存储文件，从而减少了元数据的数量，这些数据块则通过随机方式选择不同的节点并存储在各个地方。

请说明概念模型的表示方法。

答案：
概念模型的表示方法：用 E-R 图来描述现实世界的概念模型，实体用矩形表示；属性用椭圆形表示；联系用菱形表示。
解析：
概念模型的表示方法：用 E-R 图来描述现实世界的概念模型，实体用矩形表示；属性用椭圆形表示；联系用菱形表示。

主要的逻辑数据模型有哪些？

答案：

主要的逻辑数据模型有层次模型、网状模型、关系模型、面向对象模型。

解析：

主要的逻辑数据模型有层次模型、网状模型、关系模型、面向对象模型。

1.层次模型：数据库最早使用的数据模型。特点：有且仅有一个结点没有父结点，它称作根结点；其他结点有且仅有一个父结点。

2.网状模型：以网状结构表示实体与实体之间的联系。

3.关系模型：用二维表结构来表示实体及实体间联系的模型，并以二维表格的形式组织数据库中的数据。优点如下：关系模型是建立在严格的数学概念基础上的；关系模型的概念单一，统一用关系来表示实体以及实体之间的联系，对数据的检索和更新结果同样也是用关系来表示；关系模型的存取路径对用户透明，从而具有更高的数据独立性、更好的安全保密性，也简化了程序员的工作和数据库开发建立的工作。

4.面向对象模型：与数据库相结合所构成的数据模型称为面向对象模型。

简述事务的特征包括哪几方面？

答案：

事务主要有以下特征：

1、原子性

2、一致性

3、隔离性

4、持续性

解析：

事务的特征：

1、 原子性：事务的原子性保证事务包含的一组更新操作是原子不可分的，即事务是不可分割的最小工作单位，所包含的这些操作是一个整体。

2、 一致性：一致性要求事务必须满足数据库的完整性约束，且事务执行完毕后将数据库由一个一致性状态转变到另一个一致性状态。其中，数据库的一致性状态是一种以一致性规则为基础的逻辑属性。

3、 隔离性：隔离性要求事务是彼此独立的、隔离的，即一个事务的执行不能被其他事务所干扰，一个事务对数据库变更的结果必须在它 COMMIT 后，另一个事务才能存取。

4、 持续性：持续性也称为永久性，是指一个事务一旦提交，它对数据库中数据的改变就应该是永久性的，且接下来的其他操作或故障不应该对其执行结果有任何影响。

简述什么是“并发控制”？

答案：

DBMS 必须对并发操作提供一定的控制，以防止它们彼此干扰，从而保证数据库的正确性不被破坏，避免数据库的不一致性，这种机制就称为“并发控制”。

解析：

DBMS 必须对并发操作提供一定的控制，以防止它们彼此干扰，从而保证数据库的正确性不被破坏，避免数据库的不一致性，这种机制就称为“并发控制”。其中，事务就是为保证数

据的一致性而产生的一个概念和基本手段。

请写出在 MySQL 中，修改数据库的语法格式。
答案：
在 MySQL 中，修改数据库的语法格式：

解析：
在 MySQL 中，修改数据库的语法格式：

请简述什么是子查询。
答案：
可嵌套在其他 SELECT 查询中的 SELECT 查询。
解析：
通常，可以使用 SELECT 语句创建子查询，即可嵌套在其他 SELECT 查询中的 SELECT 查询。
在 MySQL 中，区分如下四类子查询： 表子查询、行子查询、列子查询和标量子查询。

简述 LAMP 的构架方式。
答案：
LAMP：即使用 Linux 作为操作系统，Apache 作为 Web 服务器，MySQL 作为数据库管理系统，PHP、Perl 或 Python 语言作为服务器端脚本解释器。
解析：
LAMP：即使用 Linux 作为操作系统，Apache 作为 Web 服务器，MySQL 作为数据库管理系统，PHP、Perl 或 Python 语言作为服务器端脚本解释器。

简述 WAMP 的构架方式。
答案：
WAMP：即使用 Windows 作为操作系统，Apache 作为 Web 服务器，MySQL 作为数据库管理系统，PHP、Perl 或 Python 语言作为服务器端脚本解释器。
解析：
WAMP：即使用 Windows 作为操作系统，Apache 作为 Web 服务器，MySQL 作为数据库管理系统，PHP、Perl 或 Python 语言作为服务器端脚本解释器。

简述数据库中主码（或主键）的含义。
答案：
在一个关系的若干个候选码或候选键中指定一个用来唯一标识关系的元组，则称这个被指定

的候选码或候选键为该关系的主码或主键。
解析：
在一个关系的若干个候选码或候选键中指定一个用来唯一标识关系的元组，则称这个被指定的候选码或候选键为该关系的主码或主键。

数据库模式的定义包括哪些操作？
答案：
数据库模式的定义包含数据库的创建、选择、修改、删除、查看等操作。
解析：
数据库模式的定义包含数据库的创建、选择、修改、删除、查看等操作。

简述关系代数中“交”运算的定义。
答案：
交：假设有两个关系 R1 和 R2，R1 和 R2 的交运算产生一个新关系 R3。R3 是由既属于关系 R1，同时又属于 R2 的元组组成，记为 R3=R1∩R2。
解析：
交：假设有两个关系 R1 和 R2，R1 和 R2 的交运算产生一个新关系 R3。R3 是由既属于关系 R1，同时又属于 R2 的元组组成，记为 R3=R1∩R2。

简述使用游标的四个步骤。
答案：
1、声明游标
2、打开游标
3、读取数据
4、关闭游标
解析：
在 MySQL 中，使用游标的具体步骤如下：
1、声明游标，在能够使用游标之前，必须声明（定义）它。
2、打开游标，在定义游标之后，必须打开该游标，才能使用。
3、读取数据，对于填有数据的游标，可根据需要取出数据。
4、关闭游标，在结束游标使用时，必须关闭游标。

简述需求分析的四个步骤。
答案：
1、确定数据库范围
2、分析数据应用过程
3、收集与分析数据
4、编写需求分析报告
解析：

需求分析的四个步骤：即确定数据库范围、分析数据应用过程、收集与分析数据和编写需求分析报告。

简述实体完整性约束的含义。

答案：

实体完整性约束是指关系的主属性，即主码的组成不能为空，也就是关系的主属性不能是空值 NULL。

解析：

实体完整性约束是指关系的主属性，即主码的组成不能为空，也就是关系的主属性不能是空值 NULL。

简述锁的定义。

答案：

一个锁实际上就是允许或组织一个事务对一个数据对象的存取特权。

解析：

一个锁实际上就是允许或组织一个事务对一个数据对象的存取特权。一个事务对一个对象加锁的结果是将别的事务“封锁”在该对象之外，特别是防止了其他事务对该对象的变更，而加锁的事务可执行它所希望的处理并维持该对象的正确状态。

封锁粒度的大小对并发系统有什么影响？

答案：

封锁粒度越大，并发度也就越小；封锁的粒度越小，并发度越高。

解析：

由最底层的数据元素到最高层的整个数据库，粒度越细，并发性就越大，但软件复杂性和系统开销也就越大。反之，粒度越大，并发性就越小。

简述关系数据库的含义。

答案：

关系数据库是以关系模式作为数据的逻辑模型，并采用关系作为数据组织方式的一类数据库，其数据库操作建立在关系代数的基础上。在一个给定的应用领域中，所有关系的集合构成一个关系数据库。

解析：

关系数据库是以关系模式作为数据的逻辑模型，并采用关系作为数据组织方式的一类数据库，其数据库操作建立在关系代数的基础上。在一个给定的应用领域中，所有关系的集合构成一个关系数据库。

简述关系数据库有哪些优点？

答案：
关系数据库的优点：包括高级的非过程语言接口、较好的数据独立性等，为商品化的关系数据库管理系统的研制做好了技术上的准备。
解析：
关系数据库的优点：包括高级的非过程语言接口、较好的数据独立性等，为商品化的关系数据库管理系统的研制做好了技术上的准备。

简述超码或超键的含义。
答案：
如果在关系的一个码中移去某个属性，它仍然是这个关系的码，则称这样的码或键为该关系的超码或超键。
解析：
如果在关系的一个码中移去某个属性，它仍然是这个关系的码，则称这样的码或键为该关系的超码或超键。一般每个关系至少有一个默认的超码或超键，即该关系的所有属性的集合，也是这个关系的的最大超码或超键。

什么是数据库的安全性？
答案：
数据库的安全性是指保护数据库以防止不合法的使用而造成数据泄露、更改或破坏。
解析：
数据库的安全性是指保护数据库以防止不合法的使用而造成数据泄露、更改或破坏，所以安全性对于任何一个 DBMS 来说都是至关重要的。

什么是数据库的外模式？
答案：
在三级模式结构中，外模式也称为子模式或用户模式，它是数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述，是与某一应用有关的数据的逻辑表示。
解析：
数据库系统的三级模式结构是指数据库系统是由模式（也称概念模式或逻辑模式）、外模式（也称子模式或用户模式）和内模式（存储模式）三级构成的。在三级模式结构中，外模式也称为子模式或用户模式，它是数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述，是与某一应用有关的数据的逻辑表示。

简述数据库系统(DBS)的组成。
答案：
通常，一个完整的数据库系统包括数据库、数据库管理系统以及相关实用工具、应用程序、数据库管理员和用户。
解析：

通常，一个完整的数据库系统包括数据库、数据库管理系统以及相关实用工具、应用程序、数据库管理员和用户。

简述数据库设计中逻辑结构设计的主要步骤。

答案：

1）模型转换

2）子模式设计

3）应用程序设计说明

4）设计评价

解析：

逻辑结构设计的步骤：

1）模型转换是指将概念模型等价地转换为特定 DBMS 支持的关系模型、网状模型或层次模型表示。

2)子模式设计的目标是抽取或导出模式的子集，以构造不同用户使用的局部数据逻辑结构。

3）编制应用程序设计说明的目的是为可实际运行的应用程序设计提供依据与指导，并作为设计评价的基础。

4）设计评价的任务是分析并检验模式及子模式的正确性与合理性。

逻辑设计的目的是什么？

答案：

将概念模型转换为等价的、并为特定 DBMS 所支持数据模型的结构。

解析：

逻辑结构设计的目标是将概念模型转换为等价的、并为特定 DBMS 所支持数据模型的结构。

简述定义表时，数据类型的含义。

答案：

数据类型指系统中所允许的数据的类型。数据库中每个列都应有适当的数据类型，用于限制或允许该列中存储的数据。

解析：

数据类型指系统中所允许的数据的类型。数据库中每个列都应有适当的数据类型，用于限制或允许该列中存储的数据。

请简述关系规范化过程。

答案：

一个低一级范式的关系模式通过模式分解可以转换为若干个高一级范式的关系模式的集合，这种过程就叫规范化。

解析：
一个低一级范式的关系模式通过模式分解可以转换为若干个高一级范式的关系模式的集合，这种过程就叫规范化。

简述专门的关系运算有哪些？
答案：
专门的关系运算不仅涉及行，而且涉及列，它可分为一元专门关系操作和二元专门关系操作。一元专门关系操作包括对单个关系进行垂直分解的投影运算和进行水平分解选择运算；二元专门关系操作则是对两个关系进行操作，包括连接运算和除运算。
解析：
专门的关系运算不仅涉及行，而且涉及列，它可分为一元专门关系操作和二元专门关系操作。一元专门关系操作包括对单个关系进行垂直分解的投影运算和进行水平分解选择运算；二元专门关系操作则是对两个关系进行操作，包括连接运算和除运算。

简述“活锁”的现象。
答案：
在并发事务处理过程中，由于锁会使一事务处于等待状态而调度其他事务处理，因而该事务可能会因优先级低而永远等待下去，这种现象称为“活锁”。
解析：
在并发事务处理过程中，由于锁会使一事务处于等待状态而调度其他事务处理，因而该事务可能会因优先级低而永远等待下去，这种现象称为“活锁”。活锁问题的解决与调度算法有关，一种最简单的办法是“先来先服务”。

请简述数据备份与恢复的定义。
答案：
数据库备份是指通过导出数据或者复制表文件的方式来制作数据库的复本；数据库恢复则是当数据库出现故障或遭到破坏时，将备份的数据库加载到系统，从而使数据库从错误状态恢复到备份时的正确状态。
解析：
数据库备份是指通过导出数据或者复制表文件的方式来制作数据库的复本；数据库恢复则是当数据库出现故障或遭到破坏时，将备份的数据库加载到系统，从而使数据库从错误状态恢复到备份时的正确状态。
数据库的恢复是以备份为基础的，它是与备份相对应的系统维护和管理操作。

简述视图用于查询检索，主要体现在哪些应用？
答案：

利用视图简化复杂的表连接；使用视图重新格式化检索出的数据；使用视图过滤不想要的数据。

解析：

视图用于查询检索，主要体现的应用：利用视图简化复杂的表连接；使用视图重新格式化检索出的数据；使用视图过滤不想要的数据。

简述索引的定义。

答案：

索引是 DBMS 根据表中的一列或若干列按照一定顺序建立的列值与记录行之间的对应关系表，因而索引实质上是一张描述索引列的列值与原表中记录行之间一一对应关系的有序表。

解析：

索引是 DBMS 根据表中的一列或若干列按照一定顺序建立的列值与记录行之间的对应关系表，因而索引实质上是一张描述索引列的列值与原表中记录行之间一一对应关系的有序表。

简述创建索引的三种方式。

答案：

创建索引的三种方式：

1.使用 CREATE INDEX 语句创建索引。

2.使用 CREATE TABLE 语句创建索引。

3.使用 ALTER TABLE 语句创建索引

解析：

创建索引的三种方式：

1.使用 CREATE INDEX 语句创建索引。

2.使用 CREATE TABLE 语句创建索引。

3.使用 ALTER TABLE 语句创建索引。

简述索引的分类有哪些。

答案：

索引在逻辑上通常包含以下几类：

1.普通索引：最基本的索引类型，没有任何限制。

2.唯一性索引：索引列中的所有值都只能出现一次，必须是唯一的。

3.主键：一种唯一性索引。

解析：

索引在逻辑上通常包含以下几类：

1.普通索引：最基本的索引类型，没有任何限制。

2.唯一性索引：索引列中的所有值都只能出现一次，必须是唯一的。

3.主键：一种唯一性索引。

什么是唯一性索引？

主知识点： 4.3.3.0、索引定义

答案：

唯一性索引和普通索引基本相同，只是有一点区别，即索引列中的所有值都只能出现一次，必须是唯一的。创建唯一性索引时，通常使用的关键字 UNIQUE。

解析：

唯一性索引和普通索引基本相同，只是有一点区别，即索引列中的所有值都只能出现一次，必须是唯一的。创建唯一性索引时，通常使用的关键字 UNIQUE。

简述普通索引的含义。

答案：

普通索引是最基本的索引类型，没有任何限制。创建普通索引时通常使用的关键字是 INDEX 或 KEY。

解析：

普通索引是最基本的索引类型，没有任何限制。创建普通索引时通常使用的关键字是 INDEX 或 KEY。

索引的删除有哪两种方法？

答案：

1.使用 DROP INDEX 删除索引。

2.使用 ALTER TABLE 语句删除索引

解析：

1.使用 DROP INDEX，语法格式如下：

DROP INDEX index_name ON tb1_name

2.使用 ALTER TABLE 语句删除索引。

简述 INSERT 语句的三种语法形式。

答案：

INSERT 语句有三种语法形式，分别对应的是 INSERT…VALUES 语句、INSERT…SET 语句和 INSERT…SELECT 语句。

解析：

INSERT 语句有三种语法形式，分别对应的是 INSERT…VALUES 语句、INSERT…SET 语句和 INSERT…SELECT 语句。

写出使用 INSERT…SELECT 语句插入子查询数据的语法格式。

答案：

使用 INSERT…SELECT 语句插入子查询数据的语法格式如下：

INSERT[INTO]tb1_name[(col_name,…)]

SELECT…
解析：
使用 INSERT…SELECT 语句插入子查询数据的语法格式如下：
INSERT[INTO]tb1_name[(col_name,…)]
SELECT…

HAVING 子句与 WHERE 子句两者之间存在哪些差异？
答案：
HAVING 子句与 WHERE 子句非常相似，HAVING 子句支持 WHERE 子句中所有的操作符和句法，但两者之间仍存在以下几点差异：
1.WHERE 子句主要用于过滤数据行，而 HAVING 子句主要用于过滤分组，即 HAVING 子句可基于分组的聚合值而不是特定行的值来过滤数据。
2.HAVING 子句中的条件可以包含聚合函数，而 WHERE 子句中则不可以。
3.WHERE 子句会在数据分组前进行过滤，HAVING 子句则会在数据分组后进行过滤。因而，WHERE 子句排除的行不包含在分组中，这就会可能改变计算值，从而影响 HAVING 子句基于这些值过滤掉的分组。
解析：
HAVING 子句与 WHERE 子句非常相似，HAVING 子句支持 WHERE 子句中所有的操作符和句法，但两者之间仍存在以下几点差异：
1.WHERE 子句主要用于过滤数据行，而 HAVING 子句主要用于过滤分组，即 HAVING 子句可基于分组的聚合值而不是特定行的值来过滤数据。
2.HAVING 子句中的条件可以包含聚合函数，而 WHERE 子句中则不可以。
3.WHERE 子句会在数据分组前进行过滤，HAVING 子句则会在数据分组后进行过滤。因而，WHERE 子句排除的行不包含在分组中，这就会可能改变计算值，从而影响 HAVING 子句基于这些值过滤掉的分组。

对于 GROUP BY 子句的使用，需要注意哪些内容？
答案：
对于 GROUP BY 子句的使用，需要注意以下几点：
1.GROUP BY 子句可以包含任意数目的列，使得其可对分组进行嵌套，为数据分组提供更加细致的控制。
2.如果在 GROUP BY 子句中嵌套了分组，那么将按 GROUP BY 子句中列的排列顺序的逆序方式依次进行汇总，并将在最后规定的分组上进行一个完全汇总。
3.GROUP BY 子句中列出的每个列都必须是检索列或有效的表达式，但不能是聚合函数。
4.除聚合函数之外，SELECT 语句中的每个列都必须在 GROUP BY 子句中给出。
5.如果用于分组的列中含有 NULL 值，则 NULL 将作为一个单独的分组返回；如果该列中存在多个 NULL 值，则将这些 NULL 值所在的行分为一组。
解析：
对于 GROUP BY 子句的使用，需要注意以下几点：

1.GROUP BY 子句可以包含任意数目的列，使得其可对分组进行嵌套，为数据分组提供更加细致的控制。
2.如果在 GROUP BY 子句中嵌套了分组，那么将按 GROUP BY 子句中列的排列顺序的逆序方式依次进行汇总，并将在最后规定的分组上进行一个完全汇总。
3.GROUP BY 子句中列出的每个列都必须是检索列或有效的表达式，但不能是聚合函数。
4.除聚合函数之外，SELECT 语句中的每个列都必须在 GROUP BY 子句中给出。
5.如果用于分组的列中含有 NULL 值，则 NULL 将作为一个单独的分组返回；如果该列中存在多个 NULL 值，则将这些 NULL 值所在的行分为一组。

简述在指定外键时，需要遵守的规则。
答案：
（1） 被参照表必须已经用一条 CREATE TABLE 语句创建了，或者必须是当前正在创建的表。如若是后一种情形，则被参照表与参照表是同一个表，这样的表称为自参照表，这种结构称为自参照完整性。
（2） 必须为被参照表定义主键。
（3） 必须在被参照表的表名后面指定列名或列名的组合。这个列或列组合必须是这个被参照表的主键或候选键。
（4） 尽管主键是不能够包含空值的，但允许在外键出现一个空值。这意味着，只要外键的每个非空值出现在指定的主键中，这个外键的内容就是正确的。
（5） 外键中的列的数目必须和被参照表的主键中的列的数目相同。
（6） 外键中的列的数据类型必须和被参照表的主键中的对应列的数据类型相同。
解析：
在指定外键时，需要遵守下列规则。
（1） 被参照表必须已经用一条 CREATE TABLE 语句创建了，或者必须是当前正在创建的表。如若是后一种情形，则被参照表与参照表是同一个表，这样的表称为自参照表，这种结构称为自参照完整性。
（2） 必须为被参照表定义主键。
（3） 必须在被参照表的表名后面指定列名或列名的组合。这个列或列组合必须是这个被参照表的主键或候选键。
（4） 尽管主键是不能够包含空值的，但允许在外键出现一个空值。这意味着，只要外键的每个非空值出现在指定的主键中，这个外键的内容就是正确的。
（5） 外键中的列的数目必须和被参照表的主键中的列的数目相同。
（6） 外键中的列的数据类型必须和被参照表的主键中的对应列的数据类型相同。

什么是函数依赖？
答案：
函数依赖定义：设 R 为任一给定关系，如果对于 R 中属性 X 的每一个值，R 中的属性 Y 只有唯一值与之对应，则称 X 函数决定 Y 或称 Y 函数依赖于 X，记作 X→Y。其中 X 称为决定因素。
解析：
函数依赖定义：设 R 为任一给定关系，如果对于 R 中属性 X 的每一个值，R 中的属性 Y 只有

唯一值与之对应，则称 X 函数决定 Y 或称 Y 函数依赖于 X，记作 X→Y。其中 X 称为决定因素。

写出使用 INSERT…SET 语句插入部分列值数据的语法格式。

答案：

使用 INSERT…SET 语句插入部分列值数据的语法格式如下：

INSERT[INTO]tb1_name

SET col_name={expr|DEFAULT},…

解析：

使用 INSERT…SET 语句插入部分列值数据的语法格式如下：

INSERT[INTO]tb1_name

SET col_name={expr|DEFAULT},…

简述封锁的工作原理。

答案：

（1） 若事务 T 对数据 D 加了 X 锁，则所有别的事务对数据 D 的锁请求都必须等待直到事务 T 释放锁。

（2） 若事务 T 对数据 D 加了 S 锁，则别的事务还可对数据 D 请求 S 锁，而对数据 D 的 X 锁请求必须等待直到事务 T 释放锁。

（3） 事务执行数据库操作时都要先请求相应的锁，即对读请求 S 锁，对更新请求 X 锁。这个过程一般是由 DBMS 在执行操作时自动隐含地进行。

（4） 事务一直占有获得的锁直到结束时释放。

解析：

封锁的工作原理是：

（1） 若事务 T 对数据 D 加了 X 锁，则所有别的事务对数据 D 的锁请求都必须等待直到事务 T 释放锁。

（2） 若事务 T 对数据 D 加了 S 锁，则别的事务还可对数据 D 请求 S 锁，而对数据 D 的 X 锁请求必须等待直到事务 T 释放锁。

（3） 事务执行数据库操作时都要先请求相应的锁，即对读请求 S 锁，对更新（插入、删除、修改）请求 X 锁。这个过程一般是由 DBMS 在执行操作时自动隐含地进行。

（4） 事务一直占有获得的锁直到结束（COMMIT 或 ROLLBACK）时释放。

因此，利用封锁机制可以解决并发操作所带来的三个不一致性问题。

使用 DESCRIBE 语句来显示表的结构，请写出其语法格式。

答案：

使用 DESCRIBE 语句来完成的，语法格式如下：

{DESCRIBE|DESC}tb1_name[col_name|wild]

解析：

使用 DESCRIBE 语句来完成的，语法格式如下：

{DESCRIBE|DESC}tb1_name[col_name|wild]

简述第一代数据库系统，即层次数据库系统和网状数据库系统的共同特点。

答案：

（1）支持三级模式（外模式、模式、内模式）的体系结构。

（2）用存取路径来表示数据之间的联系。

（3）独立的数据定义语言。

（4）导航的数据操纵语言。

解析：

层次数据库系统和网状数据库系统是第一代数据库系统，这两类数据库系统具有以下的共同特点。

（1）支持三级模式（外模式、模式、内模式）的体系结构。

（2）用存取路径来表示数据之间的联系。

（3）独立的数据定义语言。

（4）导航的数据操纵语言。

请说明 OLAP 与数据挖掘的区别。

答案：

在数据仓库技术中，OLAP 是数据汇总/聚集工具，可帮助简化数据分析，而数据挖掘是自动地发现隐藏在大量数据中的隐含模式和有趣知识；OLAP 工具的目标是简化和支持交互式数据分析，而数据挖掘工具的目标是尽可能自动处理。在这种意义下，数据挖掘比传统的联机分析处理前进了一步。

解析：

OLAP 与数据挖掘的区别：

在数据仓库技术中，OLAP 是数据汇总/聚集工具，可帮助简化数据分析，而数据挖掘是自动地发现隐藏在大量数据中的隐含模式和有趣知识；OLAP 工具的目标是简化和支持交互式数据分析，而数据挖掘工具的目标是尽可能自动处理。在这种意义下，数据挖掘比传统的联机分析处理前进了一步。

数据管理的任务是什么？

答案：

数据管理的任务就是进行数据收集、组织、控制、存储、选取、维护，实现在适当的时刻、以适当的形式、给适当的人、提供适当的数据，它是数据处理的中心问题，而数据处理则是指对各种数据进行收集、存储、加工和传播的一系列活动的总和。

解析：

数据管理的任务就是进行数据收集、组织、控制、存储、选取、维护，实现在适当的时刻、以适当的形式、给适当的人、提供适当的数据，它是数据处理的中心问题，而数据处理则是指对各种数据进行收集、存储、加工和传播的一系列活动的总和。

请说明数据库管理系统的功能。

答案：

数据库管理系统的主要功能包括以下几个方面：

1.数据定义功能；

2.数据操纵功能；

3.数据库的运行管理功能；

4.数据库的建立和维护功能；

5.数据组织、存储和管理功能；

6.其他功能：主要包括与其他软件的网络通信功能、不同数据库管理系统之间的数据传输以及相互访问功能等。

解析：

数据库管理系统是专门用于建立和管理数据库的一套软件，介于应用程序和操作系统之间。它负责科学有效地组织和存储数据，并帮助数据库的使用者能够从大量的数据中快速地获取所需数据，以及提供必要的安全性和完整性等统一控制机制，实现对数据有效的管理与维护。

数据库管理系统的主要功能包括以下几个方面：

1.数据定义功能；

2.数据操纵功能；

3.数据库的运行管理功能；

4.数据库的建立和维护功能；

5.数据组织、存储和管理功能；

6.其他功能：主要包括与其他软件的网络通信功能、不同数据库管理系统之间的数据传输以及相互访问功能等。

请说明数据库设计的目标。

答案：

数据库设计具有两个十分重要的目标，即满足应用功能需求和良好的数据库性能。

解析：

数据库设计具有两个十分重要的目标，即满足应用功能需求和良好的数据库性能。 其中，满足应用功能需求，主要是指用户当前与可预知的将来应用所需要的数据及其联系，应全部准确地存储在数据库之中，从而可满足用户应用中所需要的对数据进行的存、取、删、改等操作；良好的数据库性能，主要是指对数据的高效率存取和空间的节省，并具有良好的数据共享性、完整性、一致性及安全保密性。

简述事务与程序的关系。

答案：

事务与程序很相似，但它们是两个彼此相联而又不同的概念：程序是静止的，事务是动态的，是程序的执行而不是程序本身；同一程序的多个独立执行可以同时进行，每一步执行则是一个不同的事务。

解析：

事务与程序很相似，但它们是两个彼此相联而又不同的概念：程序是静止的，事务是动态的，是程序的执行而不是程序本身；同一程序的多个独立执行可以同时进行，每一步执行则是一个不同的事务。

MySQL 支持数据库的三级模式结构么？并做概述。

答案：

MySQL 作为一种关系型数据库管理系统，遵循 SQL 标准，提供了对数据定义语言 DDL、数据操纵语言 DML、数据控制语言 DCL 的支持，同样支持关系数据库的三级模式结构。其外模式包括视图和部分基本表，数据库模式包括若干基本表，内模式则包括若干存储文件。

解析：

MySQL 作为一种关系型数据库管理系统，遵循 SQL 标准，提供了对数据定义语言 DDL、数据操纵语言 DML、数据控制语言 DCL 的支持，同样支持关系数据库的三级模式结构。其外模式包括视图和部分基本表，数据库模式包括若干基本表，内模式则包括若干存储文件。如下图所示：

MySQL 增加的部分扩展的语言要素包括哪些？

答案：

MySQL 在 SQL 标准的基础上增加了部分扩展的语言要素：包括常量、变量、运算符、表达式、函数、流程控制语句和注解等。

解析：

MySQL 在 SQL 标准的基础上增加了部分扩展的语言要素：包括常量、变量、运算符、表达式、函数、流程控制语句和注解等。

简述 MySQL 中常量的含义及分类。

答案：

常量是指在程序运行过程中值不变的量，也称为字面值或标量值。

常量的使用格式取决于值的数据类型，可分为字符串常量、数值常量、十六进制常量、时间日期常量、位字段值、布尔值和 NULL 值。

解析：

常量是指在程序运行过程中值不变的量，也称为字面值或标量值。

常量的使用格式取决于值的数据类型，可分为字符串常量、数值常量、十六进制常量、时间日期常量、位字段值、布尔值和 NULL 值。

简述 MySQL 的内置函数的基本分类。

答案：

MySQL 包含了 100 多个函数，基本分类如下：

数学函数，例如 ABS()函数、SORT()函数；

聚合函数，例如 COUNT()函数；

字符串函数，例如 ASCII()函数、CHAR()函数；

日期和时间函数，例如 NOW()函数、YEAR()函数；
加密函数，例如 ENCODE()函数、ENCRYPT()函数;
控制流程函数，例如 IF()函数、IFNULL()函数；
格式化函数，例如 FORMAT()函数；
类型转换函数，例如 CAST()函数；
系统信息函数，例如 USER()函数、VERSION()函数。
解析：
MySQL 包含了 100 多个函数，基本分类如下：
数学函数，例如 ABS()函数、SORT()函数；
聚合函数，例如 COUNT()函数；
字符串函数，例如 ASCII()函数、CHAR()函数；
日期和时间函数，例如 NOW()函数、YEAR()函数；
加密函数，例如 ENCODE()函数、ENCRYPT()函数;
控制流程函数，例如 IF()函数、IFNULL()函数；
格式化函数，例如 FORMAT()函数；
类型转换函数，例如 CAST()函数；
系统信息函数，例如 USER()函数、VERSION()函数。

SQL 的特点有哪些？
答案：
SQL 具有如下特点：
1.SQL 不是某个特定数据库供应商专有的语言。
2.SQL 简单易学。
3.SQL 尽管看上去很简单，但它实际上是一种强有力的语言，灵活使用其语言元素，可以进行非常复杂和高级的数据库操作。
解析：
SQL 具有如下特点：
1.SQL 不是某个特定数据库供应商专有的语言。
2.SQL 简单易学。
3.SQL 尽管看上去很简单，但它实际上是一种强有力的语言，灵活使用其语言元素，可以进行非常复杂和高级的数据库操作。

简述数据定义语言在数据库中的主要应用。
答案：
数据定义语言主要用于对数据库及数据库中的各种对象进行创建、删除、修改等操作。包括 SQL 语句有：
1.CREATE：用于创建数据库或数据库对象。
2.ALTER：用于对数据库或数据库对象进行修改。
3.DROP：用于删除数据库或数据库对象。
解析：
数据定义语言主要用于对数据库及数据库中的各种对象进行创建、删除、修改等操作。包括

SQL 语句有：
1.CREATE：用于创建数据库或数据库对象。
2.ALTER：用于对数据库或数据库对象进行修改。
3.DROP：用于删除数据库或数据库对象。

简述数据操纵语言在数据库中的主要应用。
答案：
数据操纵语言主要用于操纵数据库中各种对象，特别是检索和修改数据。包括 SQL 语句主要有：
1.SELECT：用于从表或视图中检索数据，使用最为频繁的语句之一。
2.INSERT：用于将数据插入到表或视图中。
3.UPDATE：用于修改表或视图中的数据，其既可修改表或视图中一行数据，也可同时修改多行或全部数据。
4.DELETE：用于从表或视图中删除数据。
解析：
数据操纵语言主要用于操纵数据库中各种对象，特别是检索和修改数据。包括 SQL 语句主要有：
1.SELECT：用于从表或视图中检索数据，使用最为频繁的语句之一。
2.INSERT：用于将数据插入到表或视图中。
3.UPDATE：用于修改表或视图中的数据，其既可修改表或视图中一行数据，也可同时修改多行或全部数据。
4.DELETE：用于从表或视图中删除数据。

简述数据控制语言包括的 SQL 语句以及其功能。
答案：
数据控制语言主要 SQL 语句如下：
1.GRANT：用于授权，把语句许可或对象许可的权限授予其他用户和角色。
2.REVOKE：用于收回权限。
解析：
数据控制语言主要用于安全管理。主要 SQL 语句如下：
1.GRANT：用于授权，把语句许可或对象许可的权限授予其他用户和角色。
2.REVOKE：用于收回权限。

什么是传递函数依赖？
答案：
传递函数依赖定义： 设 R 为任一给定关系，X、Y、Z 为其不同属性子集，若 X→Y，Y 不能决定 X，Y→Z，则有 X→Z，称为 Z 传递函数依赖于 X。
解析：
传递函数依赖定义： 设 R 为任一给定关系，X、Y、Z 为其不同属性子集，若 X→Y，Y 不能决定 X，Y→Z，则有 X→Z，称为 Z 传递函数依赖于 X。

简述关键字的定义。

答案：

关键字的定义：设 R 为任一给定关系，U 为其所含的全部属性集合，X 为 U 的子集，若有完全函数依赖 X→U，则 X 为 R 的一个候选关键字。

解析：

关键字的定义：设 R 为任一给定关系，U 为其所含的全部属性集合，X 为 U 的子集，若有完全函数依赖 X→U，则 X 为 R 的一个候选关键字。

简述第一范式的定义。

答案：

第一范式定义：设 R 为任一给定关系，如果 R 中每个列与行的交点处的取值都是不可再分的基本元素，则 R 为第一范式。

解析：

第一范式定义：设 R 为任一给定关系，如果 R 中每个列与行的交点处的取值都是不可再分的基本元素，则 R 为第一范式。

第一范式是一个不含重复组的关系，其中不存在嵌套结构。不满足第一范式的关系为非规范关系。

简述第二范式的定义。

答案：

第二范式定义：设 R 为任一给定关系，若 R 为 1NF，且其所有非主属性都完全函数依赖于候选关键字，则 R 为第二范式。

解析：

第二范式定义：设 R 为任一给定关系，若 R 为 1NF，且其所有非主属性都完全函数依赖于候选关键字，则 R 为第二范式。

什么是 BCNF?

答案：

BCNF 定义：设 R 为任一给定关系，X、Y 为其属性集，F 为其函数依赖集，若 R 为 3NF，且其 F 中所有函数依赖 X→Y(Y 不属于 X)中的 X 必包含候选关键字，则 R 为 BCNF。

解析：

BCNF 定义：设 R 为任一给定关系，X、Y 为其属性集，F 为其函数依赖集，若 R 为 3NF，且其 F 中所有函数依赖 X→Y(Y 不属于 X)中的 X 必包含候选关键字，则 R 为 BCNF。

简述关系代数中“差”运算的定义。

答案：

差：假设有两个关系 R1 和 R2，R1 和 R2 的差运算产生一个新关系 R3。R3 是由属于关系 R1，

但不属于 R2 的元组组成，记为 R3=R1-R2。

解析：

差：假设有两个关系 R1 和 R2，R1 和 R2 的差运算产生一个新关系 R3。R3 是由属于关系 R1，但不属于 R2 的元组组成，记为 R3=R1-R2。

简述集合运算中“笛卡尔积”的运算规则。

答案：

笛卡尔积：假设有两个关系 R1 和 R2，且 R1 为 m 元关系，R2 为 n 元关系，R1 和 R2 的笛卡尔积产生一个新关系 R3，记作 R3=R1✖R2。R3 是由 R1 和 R2 的所有元组连接而成的具有（m+n）个分量的元组组成。

解析：

笛卡尔积：假设有两个关系 R1 和 R2，且 R1 为 m 元关系，R2 为 n 元关系，R1 和 R2 的笛卡尔积产生一个新关系 R3，记作 R3=R1✖R2。R3 是由 R1 和 R2 的所有元组连接而成的具有（m+n）个分量的元组组成。

什么是完全函数依赖？

答案：

完全函数依赖的定义：设 R 为任一给定关系，X、Y 为其属性集，若 X→Y，且对 X 中的任何真子集 X'，都有 X'不能决定 Y，则称 Y 完全函数依赖于 X。

解析：

完全函数依赖的定义：设 R 为任一给定关系，X、Y 为其属性集，若 X→Y，且对 X 中的任何真子集 X'，都有 X'不能决定 Y，则称 Y 完全函数依赖于 X。

什么是部分函数依赖？

答案：

部分函数依赖定义：设 R 为任一给定关系，X、Y 为其属性集，若 X→Y，且 X 中存在一个真子集 X'满足 X'→Y，则称 Y 部分函数依赖于 X。

解析：

部分函数依赖定义：设 R 为任一给定关系，X、Y 为其属性集，若 X→Y，且 X 中存在一个真子集 X'满足 X'→Y，则称 Y 部分函数依赖于 X。

简述关系语言的特点。

答案：

关系语言的特点是高度非过程化，即：用户不必请求数据库管理员为其建立特殊的存取路径，存取路径的选择由 DBMS 的优化机制来完成；用户也不必求助于循环和递归来完成数据的重复操作。

解析：

关系语言的特点是高度非过程化，即：用户不必请求数据库管理员为其建立特殊的存取路径，存取路径的选择由 DBMS 的优化机制来完成；用户也不必求助于循环和递归来完成数据的重

复操作。

简述在关系数据库中关系模型和关系的含义的区别。

答案：

同数据模型一样，数据库也有型和值之分。在关系数据库中，关系模式是型，关系是值，即关系模式是对关系的描述。

关系模式是静态的、稳定的，而关系是动态的、随时间不断变化的。这是因为关系操作在不断地更新着数据库中的数据。

解析：

同数据模型一样，数据库也有型和值之分。在关系数据库中，关系模式是型，关系是值，即关系模式是对关系的描述。

关系模式是静态的、稳定的，而关系是动态的、随时间不断变化的。这是因为关系操作在不断地更新着数据库中的数据。

数据查询时，指定列名有哪两种方式？

答案：

在数据查询时，列名的指定可以采用直接给出该列的名称的方式，也可以使用完全限定的列名方式，即“tbl_name.col_name"的列名格式。

解析：

在数据查询时，列名的指定可以采用直接给出该列的名称的方式，也可以使用完全限定的列名方式，即“tbl_name.col_name"的列名格式。

什么是交叉连接？

答案：

交叉连接又称笛卡尔积。在 MySQL 中， 它是通过在 FROM 子句中使用关键字“CROSS JOIN”来连接两张表，从而实现一张表的每一行与另一张表的每一行的笛卡尔乘积，并返回两张表的每一行相乘的所有可能的搭配结果，供 SELECT 语句中其他语法元素进行过滤和筛选条件。

解析：

交叉连接又称笛卡尔积。在 MySQL 中， 它是通过在 FROM 子句中使用关键字“CROSS JOIN”来连接两张表，从而实现一张表的每一行与另一张表的每一行的笛卡尔乘积，并返回两张表的每一行相乘的所有可能的搭配结果，供 SELECT 语句中其他语法元素进行过滤和筛选条件。

什么是等值连接？

答案：

在 FROM 子句中使用关键字“INNER JOIN”或“JOIN”连接两张表时，如若在 ON 子句的连接条件中使用运算符“=”（即等号），即进行相等性测试，则此连接方式称为等值连接，也称为相等连接。

解析：

在 FROM 子句中使用关键字“INNER JOIN”或“JOIN”连接两张表时，如若在 ON 子句的连接条件中使用运算符“=”（即等号），即进行相等性测试，则此连接方式称为等值连接，也称为相等连接。

简述在 MySQL 中使用游标的过程。

答案：

（1）声明游标：用 DECLARE CURSOR 语句创建游标。

（2）打开游标：用 OPEN 语句打开游标。

（3）读取数据：用 FETCH...INTO 语句从中读取数据。FETCH 语句是将游标指向的一行数据赋给一些变量，这些变量的数目必须等于声明游标时 SELECT 子句中选择列的数目。游标相当于一个指针，它指向当前的一行数据。

（4）关闭游标：用 CLOSE 语句关闭游标。

解析：

（1）声明游标：在能够使用游标之前，必须先声明它。这个过程实际上没有检索数据，只是定义要使用的 SELECT 语句。用 DECLARE CURSOR 语句创建游标。

（2）打开游标：在定义游标之后，必须打开该游标，才能使用。这个过程实际上是将游标连接到由 SELECT 语句返回的结果集中。用 OPEN 语句打开游标。

（3）读取数据：对于填有数据的游标，可根据需要取出数据。用 FETCH...INTO 语句从中读取数据。FETCH 语句是将游标指向的一行数据赋给一些变量，这些变量的数目必须等于声明游标时 SELECT 子句中选择列的数目。游标相当于一个指针，它指向当前的一行数据。

（4）关闭游标：在结束游标使用时，必须关闭游标。用 CLOSE 语句关闭游标。

简述在 MySQL 中，主要数据类型的分类。

答案：

在 MySQL 中，主要的数据类型包括数值类型、日期和时间类型、字符串类型、空间数据类型等。

解析：

数据类型指系统中所允许的数据的类型。数据库中每个列都应有适当的数据类型，用于限制或允许该列中存储的数据。在 MySQL 中，主要数据类型包括数值类型、日期和时间类型、字符串类型、空间数据类型等。

简述创建表时，默认值的含义。

答案：

默认值是指在向表插入数据时，如果没有明确给出某个表列所对应的值，则 DBMS 此时允许为此表列指定的一个值。

解析：

默认值是指在向表插入数据时，如果没有明确给出某个表列所对应的值，则 DBMS 此时允许为此表列指定的一个值。

MySQL 提供的几类编程语言中的常用运算符有哪些？

答案：

MySQL 提供的几类编程语言中的常用运算符：

算术运算符有：+（加）、—（减）、*（乘）、/（除）和%（求模）

位运算符有：&（位与）、|（位或）、^（位异或）、~（位取反）、>>（位右移）、<<（位左移）。

比较运算符：=（等于）、>（大于）、<（小于）、>=（大于等于）、<=（小于等于）、<>（不等于）、!=（不等于）、<=>（相等或都等于空）。

逻辑运算符：NOT 或！（逻辑非）、AND 或&&（逻辑与）、OR 或||（逻辑或）、XOR（逻辑异或）。

解析：

MySQL 几类编程语言中常用的运算符：

算术运算符有：+（加）、—（减）、*（乘）、/（除）和%（求模）

位运算符有：&（位与）、|（位或）、^（位异或）、~（位取反）、>>（位右移）、<<（位左移）。

比较运算符：=（等于）、>（大于）、<（小于）、>=（大于等于）、<=（小于等于）、<>（不等于）、!=（不等于）、<=>（相等或都等于空）。

逻辑运算符：NOT 或！（逻辑非）、AND 或&&（逻辑与）、OR 或||（逻辑或）、XOR（逻辑异或）。