1、层次型；网状型

2、实体；参照；用户定义

3、人工管理阶段；文件系统阶段；数据库系统阶段

4、一对一联系；一对多联系；多对多联系

5、整体逻辑结构；局部逻辑结构

6、外模式；模式；内模式

7、选择；投影

8、删除异常；插入异常

9、独立；加锁

10、对查询结果按照一个或多个属性列的升序或降序排列；

将查询结果按某一列或多列的值分组，值相等的为一组

11、子模式

12、3NF

13、外模式/模式映像；模式/内模式映像

14、CREATE VIEW <视图> [(<列名>[,<列名>]……)] AS <子查询> [WITH CHECK OPTION];

15、CREATE TRIGGER <触发器名> {BEFORE | AFTER} <触发事件> ON <表名> REFERENCING NEW|OLD ROW AS <变量> FOR EACH {ROW | STATEMENT} [WHEN <触发条件>] <触发动作体>

GRANT <权限>[,<权限>]…… ON <对象类型><对象名> [,<对象类型><对象名>]…… TO <用户> [,<用户>]…… [WITH GRANT OPTION];

16、统计一列中值的个数；在计算时要取消指定列中的重复值

17、UPDATE <表名> SET <列名>=<表达式>[,<列名>=<表达式>]……[WHERE <条件>];

18、REVOKE <权限>[,<权限>]…… ON <对象类型><对象名> [,<对象类型><对象名>]…… FROM <用户> [,<用户>]…… [CASCADE|RESTRICT];

19、​全码

20、对视图进行UPDATE、INSERT和DELETE操作时要保证更新、插入或删除的行满足视图定义中的谓词条件(即子查询中的条件表达式)

21、CREATE TABLE <表名> (<列名><数据类型>[列级完整性约束条件] [,<列名><数据类型>[列级完整性约束条件]]……[,<表级完整性约束条件>]);

22、​WITH CHECK OPTION

23、CREATE ASSERTION <断言名><CHECK 子句>

24、​3NF

25、​2NF

26、关系数据模型

27、​优点：1、关系模型与格式化模型不同，它是建立在严格的数学概念的基础上的。 2、关系模型的概念单一。无论实体还是实体之间的联系都用关系来表示。对数据的检索和更新结果也是关系(即表)。所以其数据结构简单、清晰，用户易懂易用。3、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。

缺点：由于存取路径对用户是隐蔽的，查询效率往往不如格式化数据模型。为了提高性能，数据库管理系统必须对用户的查询请求进行优化，因此增加了开发数据库管理系统的难度。

三个关系数据库产品的名称：Oracle、MySQL、SQL Server、Access

28、

​1、视图能简化用户操作

2、视图使用户能以多种角度看待同一数据

3、视图对重构数据库提供了一定程度的逻辑独立性

4、视图能够对机密数据提供安全保护

5、适当的利用视图可以更清晰地表达查询

29、

1、综合一体

2、高度非过程化

3、面向集合的操作方式

4、以同一种语法结构提供多种使用方式

​5、语言简洁，易学易用

30、​当多个用户并发地存取数据库时就会产生多个事务同时存取同一数据的情况，对并发操作进行并发控制可以保证存取和存储正确的数据，维护事务的一致性和数据的一致性。

31、

1、需求分析阶段 ；

2、概念结构设计阶段 ；

3、逻辑结构设计阶段 ；

4、物理结构设计阶段 ；

5、数据库实施阶段 ；

​6、数据库运行和维护阶段

32、​因为关系是一个元组的集合,而集合中的元素不允许重复出现,因此在关系模型中对关系作了限制, 关系中的元组不能重复。

33、​审计日志：用于记录用户对数据库的所有操作。

作用：审计员利用审计日志监控数据库中的各种行为，重现导致数据库现有状况的一系列事件，找出非法存取数据的人、时间和内容等。还可以通过对审计日志分析，对潜在的威胁提前采取措施加以防范。

34、1、浪费存储空间及修改麻烦；2、潜在的数据不一致性。

35、含义：角色是被命名的一组与数据库操作相关的权限，角色是权限的集合，可以为一组具有相同权限的用户创建一个角色。

作用：使用角色来管理数据库可以简化授权的过程，使自主授权的执行更加灵活、方便。

36、区别：(1)最大长度：char最大长度是255字符，varchar最大长度是65535个字节；(2)定长： char是定长的，不足的部分用隐藏空格填充，varchar是不定长的；(3)空间使用： char会浪费空间，varchar会更加节省空间；(4)查找效率： char查找效率会很高，varchar查找效率会更低；(5)尾部空格： char插入时可省略，vaechar插入时不会省略，查找时省略。

适用场景：(1)char类型如果存的数据量小于最大长度，剩余的空间会使用空格填充，因此可能会浪费空间，所以char类型适合存储长度固定的数据，这样既不会浪费空间，效率还比varchar略高； (2)varchar类型如果存到数据量小于最大长度，剩余的空间会留给别的数据使用，所以varchar类型适合存储长度不固定的数据，这样虽然没有char存储效率高，但至少不会浪费空间。

37、​用户：应用系统的具体操作者，用户可以自己拥有权限信息，可以归属于0～n个角色，可属于0～n个组。他的权限集是自身具有的权限、所属的各角色具有的权限、所属的各组具有的权限的合集。它与权限、角色、组之间的关系都是n对n的关系。

权限：系统的所有权限信息具有上下级关系，是一个树状的结构。对于每个权限，又存在两种情况，一个是只是可访问，另一种是可授权，例如对于“查看用户”这个权限，如果用户只被授予“可访问”，那么他就不能将他所具有的这个权限分配给其他人。

角色：为了对许多拥有相似权限的用户进行分类管理，定义了角色的概念，例如系统管理员、管理员、用户等角色。角色具有上下级关系，可以形成树状视图，父级角色的权限是自身及它的所有子角色的权限的综合。父级角色的用户、父级角色的组同理可推。

38、​数据库的恢复：数据库管理系统具有把数据库从错误状态恢复到某一已知的正确状态的功能 。

两种恢复数据库的策略或方法：数据转储、登记日志文件

39、物理独立性：用户的应用程序与数据库中数据的物理存储是相互独立的。

逻辑独立性：用户的应用程序与数据库的逻辑结构是相互独立的。

​40、(1)存储过程是一组已创建并存储在数据库中的SQL语句，所以可以一遍又一遍地重用代码。而触发器是一种特殊的不是由用户直接调用存储过程，创建触发器时，会定义在针对特定表或列进行特定类型的数据修改时触发。断言创建以后，任何对断言中涉及关系的操作都会触发关系数据库管理系统对断言的检查。

(2)用户可以使用Execute或Exec语句来直接调用或执行存储过程，而无法直接调用或执行触发器。触发相关事件时，只会自动执行触发器。任何对断言中所涉及关系的操作都会触发断言检查；

(3)存储过程可以采用输入参数，而触发器/断言中不能将参数作为输入传递给触发器/断言；

(4)存储过程可以返回零或n值，触发器无法返回值，断言检查中任何使断言不为真值的操作都会被拒绝执行。

(5)我们可以在存储过程中使用事务，触发器内不允许进行事务处理；

(6)存储过程通常用于执行用户指定的任务，触发器通常用于审计工作，断言用于指定更具一般性的约束。

41、​若属性A是基本关系R的主属性，则A不能取空值，所谓空值就是“不知道”或“不存在”或“无意义”的值。 举例：学生（学号、姓名、性别、专业号、年龄）关系中学号为主码，则学号不能取空值。

42、​不包含在任何候选码中的属性称为非主属性。 例如student(SNO,COURSE,SCORE) (SNO,COURSE) --> SCORE，SCORE就是非主属性。

43、​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中的每个元组在F上的值必须：或者取空值，或者等于S中的某个元组的主码值。

44、提供了表示实体型、属性和联系的方法。(1)实体型用矩形表示，矩形框内写明实体名。(2)属性用椭圆形表示，并用无向边将其与相应的实体型连接起来。(3）联系用菱形表示，并用无向边分别与有关实体型连接起来，同时在无向边旁标上联系的类型。

45、​数据字典是进行详细的数据收集和数据分析所获得的主要成果。它是关于数据库中数据的描述，即元数据，而不是数据本身。数据字典是在需求分析阶段建立，在数据库设计过程中不断修改、充实、完善的。它在数据库设计中占有很重要的地位。

46、θ为“=”的连接运算称为等值连接。它是从关系R与S的广义笛卡尔积中选取A、B属性值相等的那些元组。

47、​当多个用户并发地存取数据库时就会产生多个事务同时存取同一数据的情况。若对并发操作不加控制就可能会存取和存储不正确的数据，破坏事务的一致性和数据库的一致性。

48、自然连接是一种特殊的等值连接，要求两个关系中进行比较的分量必须是同名的属性值，并且在结果中把重复的属性列去掉。

49、在数据库中定义满足下面两个条件的基本层次联系的集合为层次模型：(1)有且只有一个结点没有双亲结点，这个结点称为根节点;(2)根以外的其他结点有且只有一个双亲结点。

50、事务是用户定义的一个数据库操作序列，这些操作要么全做，要么全不做，是一个不可分割的工作单位。

51、完全函数依赖：在R(U)中，如果X -> Y，并且对于X的任何一个真子集X'，都有X’不能函数确定Y，则称Y对X完全函数依赖。

部分函数依赖：如果X -> Y，但Y不完全函数依赖于X，则称Y对X部分函数依赖。

52、概念模型是按用户的观点来对数据和信息建模，主要用于数据库设计。​概念模型实际上是现实世界到机器世界的一个中间层次。概念模型用于信息世界的建模，是现实世界到信息世界的第一层抽象，是数据库设计人员进行数据库设计的有力工具，也是数据库设计人员与用户之间进行交流的语言。

53、​选择：在关系中选择满足给定条件的诸元组。

投影：从关系中选择若干个属性列组成新的关系。

54、​在关系模式R(U)中，如果X->Y(Y不包含于X),Y不函数确定X，Y->Z，Z不包含于Y，则称Z对X传递函数依赖。

55、应用程序和数据结构之间相互独立, 互不影响。它包括数据的物理独立性和逻辑独立性。

56、

(1)码：考号；

作为一个二维表，每一个分量是不可分的数据项，所以满足1NF；且每一个非主属性完全函数依赖于考号，所以满足2NF；毕业中学通讯地址传递依赖于考号，所以不满足3NF；所以该关系最高满足2NF。

(2)码：考号、课程号；

作为一个二维表，每一个分量是不可分的数据项，所以满足1NF；考生姓名部分函数依赖于码，所以不满足2NF；所以该关系最高满足1NF。

(3)码：课程号；

​作为一个二维表，每一个分量是不可分的数据项，所以满足1NF；且每一个非主属性完全函数依赖于课程号，所以满足2NF；满分分值传递依赖于课程号，所以不满足3NF；所以该关系最高满足2NF。

57、

(1)成绩关系SCORE：码：学号、课程号；

作为一个二维表，每一个分量是不可分的数据项，所以满足1NF；姓名部分函数依赖于码，所以不满足2NF；所以该关系最高满足1NF。

(2)学籍信息STUD：码：学号；

作为一个二维表，每一个分量是不可分的数据项，所以满足1NF；且每一个非主属性完全函数依赖于学号，所以满足2NF；没有非主属性传递依赖于码，所以满足3NF；且每一个决定因素都包含码，所以满足BCNF；所以该关系最高满足BCNF。

(3)大学公选课关系T：码：学号、教师号；

作为一个二维表，每一个分量是不可分的数据项，所以满足1NF；课程号部分函数依赖于码，所以不满足2NF；所以该关系最高满足1NF。

58、

​(1)成绩关系SC：码：学号、课程号；

作为一个二维表，每一个分量是不可分的数据项，所以满足1NF；姓名部分函数依赖于码，所以不满足2NF；所以该关系最高满足1NF。

(2)学籍信息STUDENT：码：学号；

作为一个二维表，每一个分量是不可分的数据项，所以满足1NF；且每一个非主属性完全函数依赖于学号，所以满足2NF；没有非主属性传递依赖于码，所以满足3NF；且每一个决定因素都包含码，所以满足BCNF；所以该关系最高满足BCNF。

(3)大学公选课关系T：码：学号、教师号；

作为一个二维表，每一个分量是不可分的数据项，所以满足1NF；课程号部分函数依赖于码，所以不满足2NF；所以该关系最高满足1NF。

59、

(1)成绩关系S：码：学号、课程号；

作为一个二维表，每一个分量是不可分的数据项，所以满足1NF；姓名部分函数依赖于码，所以不满足2NF；所以该关系最高满足1NF。

(2)学籍信息R：码：学号；

作为一个二维表，每一个分量是不可分的数据项，所以满足1NF；且每一个非主属性完全函数依赖于学号，所以满足2NF；毕业中学的通讯地址传递依赖于学号，所以不满足3NF；所以该关系最高满足2NF。

(3)大学公选课关系T：码：学号、教师号；

作为一个二维表，每一个分量是不可分的数据项，所以满足1NF；课程号部分函数依赖于码，所以不满足2NF；所以该关系最高满足1NF。

60、

(1) select \* from student where S\_NAME='王五';

(2) create view CS\_STUDENT as select S\_ID,S\_NAME from student where DEPT='CS';

(3) select DEPT,COUNT(S\_ID) as NUM from student group by DEPT order by NUM ASC;

(4) select S\_ID,S\_NAME from student where SEX='女' AND DEPT in (select DEPT from student where S\_NAME='赵梅');

(5) select distinct student.S\_ID,S\_NAME from student,SC where student.S\_ID IN (select S\_ID from SC,course where SC.C\_ID=course.C\_ID AND C\_NAME='离散数学') AND DEPT='CS';

61、

(1) select 学号,家庭住址 from STUDENT where 姓名='张三';

(2) create view IS\_STUDENT as select \* from STUDENT where 专业='信息系统';

(3) select 性别,COUNT(学号) as 人数 from STUDENT where 专业='计算机' group by 性别 order by 人数 ASC;

(4) select 学号,姓名 from STUDENT where 学号 IN (select 学号 from SC,COURSE where SC.课程号=COURSE.课程号 AND 成绩>=90 AND 课程名称='数据结构');

(5) select STUDENT.学号,姓名 from STUDENT,SC,COURSE where STUDENT.学号=SC.学号 AND SC.课程号=COURSE.课程号 AND 课程名称='数据库原理' AND 成绩<=ALL(select 成绩 from SC where 学号 IN (select 学号 from STUDENT where 姓名='李四')) AND 专业 IN (select 专业 from STUDENT where 姓名='李四');

62、

(1) SELECT \* FROM student WHERE SNAME='李四';

(2) CREATE VIEW SNO\_SNAME AS SELECT SNO,SNAME FROM student;

(3) SELECT DEPARTMENT,COUNT(SNO) FROM student GROUP BY DEPARTMENT ORDER BY COUNT(SNO) DESC;

(4) SELECT \* FROM student WHERE SEX='男' AND DEPARTMENT IN (SELECT DEPARTMENT FROM student WHERE SNAME='王五');

(5) SELECT SNO,SNAME FROM student WHERE DEPARTMENT='CS' AND SNO NOT IN (SELECT SNO FROM SC WHERE CNO IN (SELECT CNO FROM course WHERE CNAME='数据库原理'));

63、

(1) SELECT 学号,家庭住址 FROM STUDENT WHERE 姓名='李强';

(2) CREATE VIEW IS\_STUDENT AS SELECT 学号,姓名,性别,专业,家庭住址 FROM STUDENT WHERE 专业='信息系统';

(3) SELECT 性别,COUNT(学号) FROM STUDENT WHERE 专业='中文' GROUP BY 性别 ORDER BY COUNT(学号) ASC;

(4) SELECT 学号 FROM 选课 WHERE 成绩>90 AND 课程号 IN (SELECT 课程号 FROM 课程 WHERE 课程名称='数据库原理');

(5) SELECT STUDENT.学号,姓名,性别,专业,家庭住址 FROM STUDENT,选课 WHERE 专业=(SELECT 专业 FROM STUDENT WHERE 姓名='王五') AND STUDENT.学号=选课.学号 AND 课程号=(SELECT 课程号 FROM 课程 WHERE 课程名称='数据库原理') AND 成绩 > (SELECT 成绩 FROM 选课 WHERE 学号=(SELECT 学号 FROM STUDENT WHERE 姓名='王五')) ;