

填空题

- 数据模型通常由 数据结构、数据操作 和完整性约束三部分组成
- 关系数据模型由关系数据结构、关系操作集合和关系完整性约束三大要素组成。
- 数据库系统一般由数据库、数据库管理系统DBMS、应用系统、数据库管理员DBA 和用户构成
- 数据管理技术经历了 人工管理、文件系统(file-processing system)、数据库系统(database management system) 三个阶段
- 数据模型根据应用目的不同分为 概念模型 和 逻辑模型
- 实体所具有的某一特性称为 属性(attribute)，唯一标识实体的属性集称为 码(key)
- 两个实体间的联系可以分为四种：一对一(one-to-one)、一对多(one-to-many)、多对一(many-to-one)、多对多(many-to-many) 联系
- 数据库系统由 外模式(view level)、逻辑模式(logical level) 和 内模式(physical level) 三级结构组成，DBMS在这三级之间提供了 外模式/逻辑模式映像(physical(internal)/logical(conceptual) mapping) 和 逻辑模式/内模式(view(extern)/logical(conceptual) mapping) 映像
- 关系模型中有 实体完整性约束(integrity constraint)、参照完整性约束(referential integrity constraint) 和 用户自定义的完整性约束 这三类完整性约束
- 自然连接是一种特殊的等值连接，他要求 两个关系中比较的分量必须是相同的属性组
- SQL语言的
 - 数据查询功能的核心动词是 SELECT
 - 数据定义功能的核心动词是 CREATE, DROP, ALTER
 - 数据操纵功能的核心动词是 UPDATE, DELETER, INSERT
 - 数据控制功能的核心动词是 GRANT, REVOKE
- SQL概念中，视图，基本表，存储文件 分别对应数据库三级模式中的 外模式、模式 和 内模式
- 数据依赖是一个关系内部属性与属性之间的一种约束关系，最重要的数据依赖是 函数依赖 和 多值依赖
- 数据的物理独立性：当数据库的存储结构发生变化，有数据库管理员对（逻辑模式/内模式）映像作相应的改变，可以使逻辑模式保持不变，从而应用程序也可以不变，这样保证了数据的物理独立性
Physical Data Independence:the ability to modify the physical schema without changing the logical schema or the views(The physical / logical mapping)
- 逻辑数据独立性(Logical Data Independence(逻辑数据独立性): the ability to modify the logical schema without changing the logical views(The view / logical mapping)
- 在关系模型中，表的行称为 元组，列称为 属性
- 逻辑模式/内模式映像(The physical/logical mapping) 为数据库提供了物理数据独立性
- 用树型结构表示实体类型及实体间联系的数据模型称为 层次模型
- 用有向图结构表示实体类型及实体间联系的数据模型称为 网状模型
- 若用户编写程序时，只需要指出做什么，不需指出怎么做，这类DML是 非过程化(non-procedural) DML，关系的DML属于这一类
- 在SQL中，用 UPDATE 命令可以修改表中的数据，用 ALTER 命令可以修改表的结构

- 在ER图中，菱形框表示 联系
- 在关系中，内模式是 存储 模式的集合
- 事务 (transaction) 的特性包括：原子性(atomicity)，一般性(consistency)，隔离性(isolation) 和持久性(durability)
- SQL数据库中的表，可以是基本表，也可以是 视图(view)
- E-R方法的三要素是：实体(entity)、属性(attribute)、联系(relationship)
- 关系代数中，连接是由 笛卡尔积(cartesian product) 操作与 选择(select) 操作组合而成的
- 在数据库中，所有应用程序都通过 DBMS 访问数据库
- 在SQL语言中，允许获得某种权限的用户把这种权限再传授给其他用户，可以在 GRANT 语句中指定 WITH GRANT OPTION 子句
- 一个事务成功完成后，他对数据库的改变必须是永久的，这一特性称为事务的 持久性(duration)
- 实体完整性规则规定：关系中的元组在组成 主键(primary key) 的属性上不能为空值
- 根据参照完整性规则，外码的值或者等于以此外码为主码的关系中的某个元组主码的值，或者取 空(NULL)
- 在一个关系中，任何一个候选码中所包含的属性都称为 主属性或码属性
- 数据库系统中构成单一逻辑工作单元的操作集合称为 事务(transaction)
- 关系代数是集合代数为基础发展起来的，它是一种关系操纵语言，它的操作对象和操作结果都是 关系
- A data model defines the specification of managing data item in database, it is a collection of conceptual tools for describing data structure, texts, web pages, voices, images and videos.
- As human-machine interface, the database language consists of two parts, i.e the database definition language and database manipulation language
- DBMS can be divided into two main parts, that is query processor and transaction manager
- There are three types of pure query languages related to the relational model, that is, relational algebra(关系代数), tuple relational calculus (元组关系演算), and domain relational calculus (域关系演算) .

E-R图向关系模型的转换

关系的描述称为关系模式。可以形式化地表示为 $R(U, D, DOM, F)$ ， R 为关系名， U 为组成该关系的属性名集合， D 为 U 中属性所来自的域， DOM 为属性向域的映射集合， F 为属性间数据的以来关系集合

E-R图向关系模型的转换要解决的问题是，如何将实体型和实体间的联系转换为关系模型，如何确定这些关系的属性和码

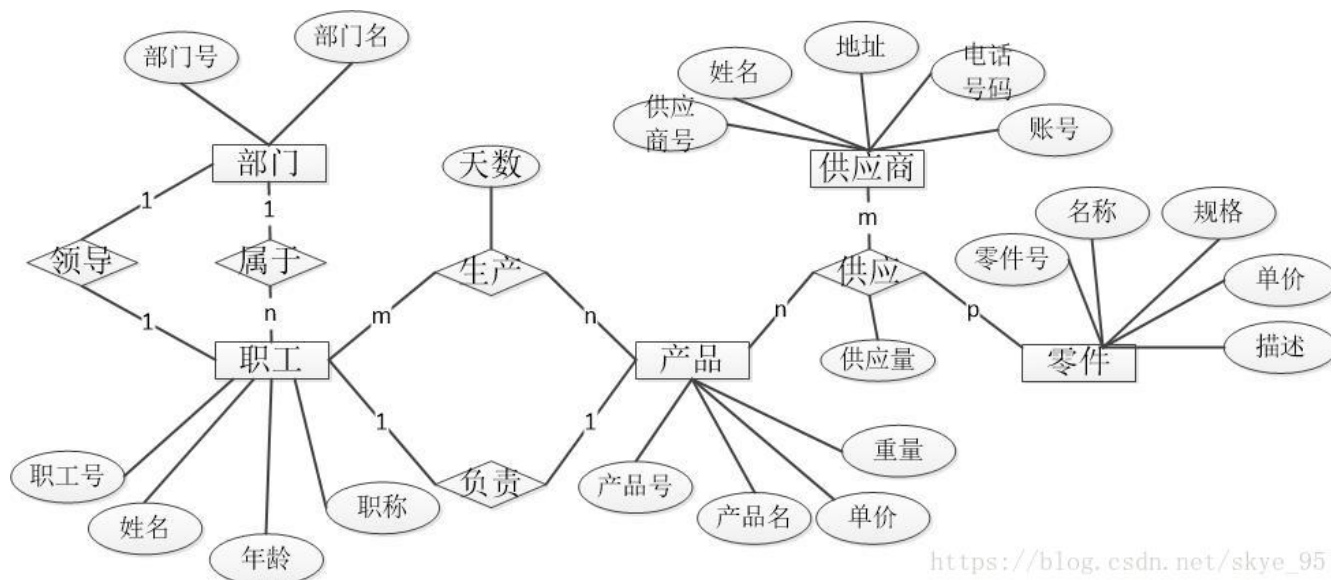
一般的转换原则为：一个实体型转换为一个关系模式，关系的属性就是实体的属性，关系的码就是实体的码

对于实体间的联系有以下不同的情况：

- 一个 1:1 联系可以转换成一个独立的关系模型，也可以与任意一端对应的关系模式合并。如果转换为一个独立的关系模型，则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，那个实体的码均是该关系的候选码。如果与某一端实体对应的关系合并，则需要在该关系模式的属性中加入另一个关系模式的码和联系本身的属性。
- 一个 1: n 联系可以转换为一个独立的关系模式，也可以与n端对应的关系模式合并。如果转换为一个独立的关系模式，则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为n端实体的码。

- 一个m:n联系转换为一个关系模式，与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，各实体的码组成关系的码或关系码的一部分。
- 三个或三个以上实体间的一个多元联系可以转换为一个关系模式。与该多元联系相连的各实体的码以及联系本身的属性均转换为关系的属性，各实体的码组成关系的码或关系码的一部分。
- 具有相同码的关系可以合并。

示例：将下面的E-R模型转换为关系模式。关系的码用下划线标出



https://blog.csdn.net/skye_95

部门 (部门号, 部门名, 经理的职工号)

此为部门实体对应的关系模式，该关系模式已包含了联系“领导”所对应的关系模式，经理的职工号是此关系的候选码

职工 (职工号, 部门号, 姓名, 年龄, 职称, 产品)

此为职工实体对应的关系模式，该关系模式已包含了联系“属于”所对应的关系模式

产品 (产品号, 产品名, 单价, 重量, 产品组长的职工号)

供应商 (供应商号, 姓名, 地址, 电话号码, 账号)

零件 (零件号, 名称, 规格, 单价, 描述)

生产 (职工号, 产品号, 工作天数)

此为联系“生产”所对应的关系模式

供应 (供应商号, 产品号, 零件号, 供应量)

此为联系“供应”所对应的生产模型

Chapter1

- Database-management system (DBMS)
- Database-system applications
- File-processing systems
- Data inconsistency
- Consistency constraints
- Data abstraction
- Instance
- Schema
 - Physical schema
 - Logical schema
- Physical data independence
- Data models
 - Entity-relationship model
 - Relational data model
 - Object-based data model
 - Semistructured data model
- Database languages
 - Data-definition language
 - Data-manipulation language
 - Query language
- Metadata
- Application program
- Normalization
- Data dictionary
- Storage manager
- Query processor
- Transactions
 - Atomicity
 - Failure recovery
 - Concurrency control
- Two- and three-tier database architectures
- Data mining
- Database administrator (DBA)

Chapter2

- Table
- Relation
- Tuple
- Attribute
- Domain
- Atomic domain

- Null value
- Database schema
- Database instance
- Relation schema
- Relation instance
- Keys
 - Superkey
 - Candidate key
 - Primary key
- Foreign key
 - Referencing relation
 - Referenced relation
- Referential integrity constraint
- Schema diagram
- Query language
 - Procedural language
 - Nonprocedural language
- Operations on relations
 - Selection of tuples
 - Selection of attributes
 - Natural join
 - Cartesian product
 - Set operations
- Relational algebra

Chapter3

- Data-definition language
- Data-manipulation language
- Database schema
- Database instance
- Relation schema
- Relation instance
- Primary key
- Foreign key
 - Referencing relation
 - Referenced relation

- Null value
- Query language
- SQL query structure
 - **select** clause
 - **from** clause
 - **where** clause
- Natural join operation
- **as** clause
- **order** by clause
- Correlation name (correlation variable, tuple variable)
- Set operations
 - **union**
 - **intersect**
 - **except**
- Null values
 - Truth value “unknown”
- Aggregate functions
 - **avg, min, max, sum, count**
 - **group by**
 - **having**
- Nested subqueries
- Set comparisons
 - { <, <=, >, >= } { **some, all** }
 - **exists**
 - **unique**
- **lateral** clause
- **with** clause
- Scalar subquery
- Database modification
 - Deletion
 - Insertion
 - Updating

Chapter4

- Join types
 - Inner and outer join
 - Left, right and full outer join
 - Natural, using, and on
- View definition
- Materialized views
- View update
- Transactions
 - Commit work
 - Rollback work
 - Atomic transaction
- Integrity constraints
- Domain constraints
- Unique constraint
- Check clause
- Referential integrity
 - Cascading deletes
 - Cascading updates

- Assertions
- Date and time types
- Default values
- Indices
- Large objects
- User-defined types
- Domains
- Catalogs
- Schemas
- Authorization
- Privileges
 - **select**
 - **insert**
 - **update**
 - **all privileges**
 - Granting of privileges
 - Revoking of privileges
 - Privilege to grant privileges
 - Grant option
- Roles
- Authorization on views
- Execute authorization
- Invoker privileges
- Row-level authorization

Chapter5

- JDBC
- ODBC
- Prepared statements
- Accessing metadata
- SQL injection
- Embedded SQL
- Cursors
- Updatable cursors
- Dynamic SQL
- SQL functions
- Stored procedures
- Procedural constructs
- External language routines
- Trigger
- Before and after triggers
- Transition variables and tables
- Recursive queries
- Monotonic queries
- Ranking functions
 - Rank
 - Dense rank
 - Partition by
- Windowing
 - Pivoting
 - Data cube
 - Slicing and dicing
 - Rollup and drill down
- Online analytical processing (OLAP)
- Multidimensional data
 - Measure attributes
 - Dimension attributes
- Cross-tabulation

Chapter6

- Relational algebra
- Relational-algebra operations
 - Select σ
 - Project Π
 - Union \cup
 - Set difference $-$
 - Cartesian product \times
 - Rename ρ
- Additional operations
 - Set intersection \cap
 - Natural join \bowtie
 - Assignment operation
 - Outer join
 - ◊ Left outer join \Join
 - ◊ Right outer join \Join
 - ◊ Full outer join \Join
- Multisets
- Grouping
- Null value
- Tuple relational calculus
- Domain relational calculus
- Safety of expressions
- Expressive power of languages

Chapter7

- Entity-relationship data model
 - Entity and entity set
 - Attributes
 - Domain
 - Simple and composite attributes
 - Single-valued and multivalued attributes
 - Null value
 - Derived attribute
 - Superkey, candidate key, and primary key
 - Relationship and relationship set
 - Binary relationship set
 - Degree of relationship set
 - Descriptive attributes
 - Superkey, candidate key, and primary key
 - Role
 - Recursive relationship set
 - E-R diagram
 - Mapping cardinality:
 - One-to-one relationship
 - One-to-many relationship
 - Many-to-one relationship
 - Many-to-many relationship
 - Participation
 - Total participation
 - Partial participation
 - Weak entity sets and strong entity sets
 - Discriminator attributes
 - Identifying relationship
 - Specialization and generalization
 - Superclass and subclass
 - Attribute inheritance
 - Single and multiple inheritance
 - Condition-defined and user-defined membership
 - Disjoint and overlapping generalization
 - Total and partial generalization
 - Aggregation
 - UML
 - UML class diagram
-

Chapter8

- E-R model and normalization
- Decomposition
- Functional dependencies
- Lossless decomposition
- Atomic domains
- First normal form (1NF)
- Legal relations
- Superkey
- R satisfies F
- F holds on R
- Boyce–Codd normal form (BCNF)
- Dependency preservation
- Third normal form (3NF)
- Trivial functional dependencies
- Closure of a set of functional dependencies
- Armstrong's axioms
- Closure of attribute sets
- Restriction of F to R_i
- Canonical cover
- Extraneous attributes
- BCNF decomposition algorithm
- 3NF decomposition algorithm
- Multivalued dependencies
- Fourth normal form (4NF)
- Restriction of a multivalued dependency
- Project-join normal form (PJNF)
- Domain-key normal form (DKNF)
- Universal relation
- Unique-role assumption
- Denormalization