DAMA-DMBOK

数据管理知识体系指南CDGA/CDGP认证

第5章 数据建模和设计（完整课程视频请扫描二维码）

第5章 数据建模和设计

01 引言
02 活动
03 工具
04 方法
05 治理

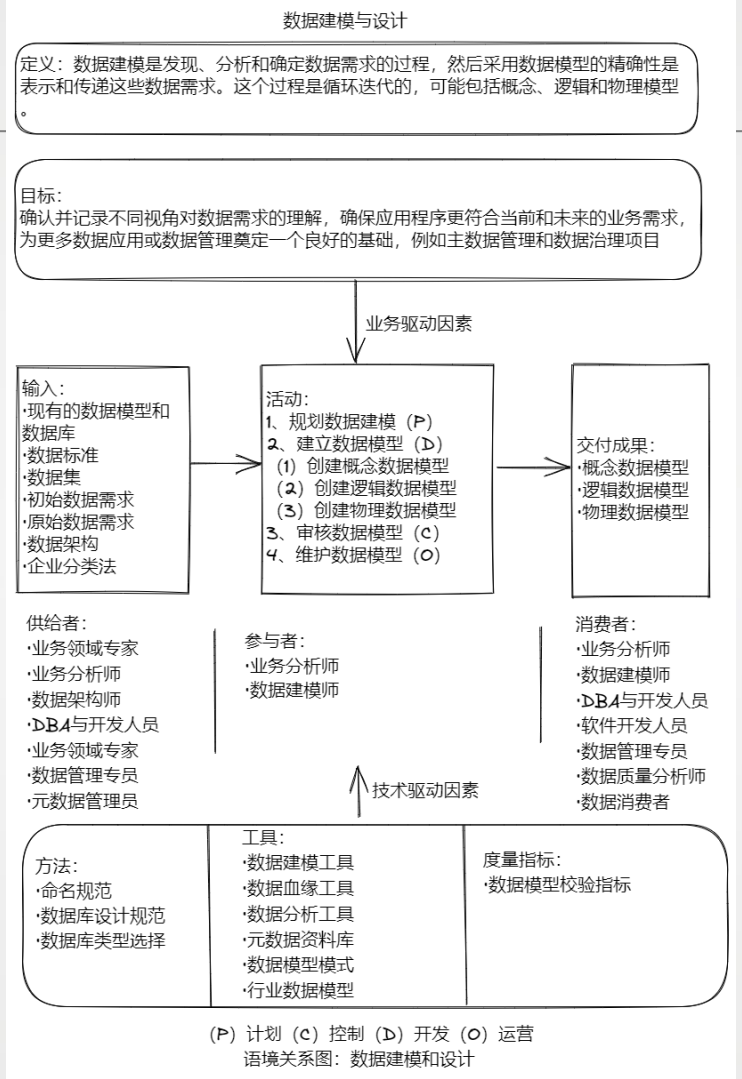
01

引言

数据建模的定义、业务驱动因素、目标和原则、基本概念

数据建模的语境图

每个语境图中：
输入、活动、供给者消费者参与者不会考。
常考：定义、工具、度量指标记清楚每个语境图的定义和工具



数据建模的定义

数据建模是发现、分析和确定数据需求的过程，用一种称为数据模型的精确形式表示和传递这些数据需求。数据建模是数据管理的一个重要组成部分。建模过程中要求组织发现并记录数据组合的方式。在建模过程本身，设计了数据组合的方式。

业务驱动因素

1）提供有关数据的通用词汇表
2）获取、记录组织内数据和系统的详细信息
3）在项目中作为主要的交流沟通工具
4）提供了应用定制、整合，甚至替换的起点

目标和原则

目标：

确认和记录不同视角对数据需求的理解，确保应用程序更符合当前和未来的业务需求，为更多数据应用或数据管理奠定一个良好的基础，例如主数据管理和数据治理项目。

确认和记录有助于：

1）格式化
2）范围定义
3）知识保留记录

1、数据建模和数据模型

数据建模最常用在系统开发与系统维护的工作环境中，也称为系统开发生命周期（SDLC）。数据模型描述了组织已经理解或者未来需要的数据。数据模型包含一组带有文本标签的符号，这些符号试图以可视化方式展现数据需求并将其传递给数据建模人员，以获得一组特别的数据。

2、建模的数据类型

1）类别信息

用于对事物进行分类和分配事物类型的数据

2）资源信息

实施操作流程所需资源的基本数据

3）业务事件信息

在操作过程中创建的数据

4）详细交易信息

详细的交易信息通常通过销售系统生成。

3、数据建模组件

（1）实体

1）实体的别名
2）实体的图形表示：矩形代表实体
3）实体的定义：清晰、准确、完整

（2）关系

1）关系的别名：导航路径、边界、链接
2）关系的图形表示：显示为线条
3）关系基数：在两个实体间的关系中，基数说明了一个实体（实体实例）和其他实体参与建立关系的数量。
4）关系元数
①一元关系　递归关系，或自我引用关系
②二元关系　涉及两个实体
③三元关系　涉及三个实体
5）外键

（3）属性：　属性是一种定义、描述或度量实体某方面的性质。属性可能包含域，这将在后面展开讨论。

1）属性的图形表示：通常在实体矩形内的列表中描述
2）标识符：　也称为键，是唯一标识实体实例的一个或多个属性的集合。
键的结构类型：①单一键　②组合键　③复合键
键的功能类型：主键、备用键
标识关系与非标识关系：**独立实体**是指其主键仅包含只属于该实体的属性；非独立实体是指其主键至少包含一个来自其他实体的属性；非独立实体至少含有一个标识关系；**标识关系是指父实体的主键作为外键被集成到子实体主键的一部分，正如学生和注册之间、课程和注册之间的关系。在非标识关系中，父实体的主键仅被继承为子实体的非主外键属性。**

（4）域：　代表某一属性可被赋予的全部可能取值

基本概念——4、数据建模方法

（1）关系建模

表示方法：

信息工程（IE）： 采用三叉线俗称鸭掌模型来表示基数
信息建模集成定义（IDEF1X）
巴克符号
陈氏符号

考察接下来的六种建模方法，包括关系建模在内。记住哪六种

数据建模方法——4、数据建模方法

（2）维度建模

1）事实表

2）维度表：高度反范式的

维度属性以不同速率变化，3种主要的变化类型，被称为ORC

①覆盖：新值覆盖旧值（缓慢变化slowly changing维如何操作？）

②新行：新值写在新行中，旧行被标记为非当前值

③新列：一个值的多个实例列在同一行的不同列中，而一个新值意味着将系列中的值向下一点写入，以便在前面为新值留出空间，最后一个值被丢弃。

3）雪花模型：是将星型模式中的平面、单表、维度结构规范为相应的组件层次结构或网络结构。

4）粒度：是指事实表中的单行数据的含义或者描述，这是每行都有的最详细信息。定义一个事实表中的粒度是维度建模的关键步骤之一。

5）一致性维度：基于整个组织考虑构建的，而不是基于某个特定的项目。

6）一致性事实：使用跨多个数据集市的标准化术语。

（3）UML统一建模语言

统一建模语言是一种图形风格的建模语言。UML根据数据库的不同有着不同种类的表示法（类模型）。UML规定了类（实体类型）和它们之间关系类型。特点有：

1）与ER图相似，但ER图中没有操作（Operation）或方法部分。

2）在ER图中，与操作最为接近概念的是存储过程。

3）属性类型（如日期、分钟）是用程序编程语言的数据类型表示的，而不是物理数据库数据类型来表示。

4）默认值可以在符号中有选择的显示

5）访问数据是通过类的公开接口。

类操作可以是：

1）公开的

2）内部可见的

3）私密的

数据建模方法——4、数据建模方法

（4）基于事实的建模

一个广泛而强大的约束系统依赖于流畅的自动语言和对具体实例的自动检查。

基于事实的建模是一种概念建模语言，通常基于Fact-Based Modeling对象的特征，以及每个对象在每个事实中所扮演的角色来描述世界。

不使用属性，通过表示对象（实体和值）之间的精确关系来减少直观或专家判断的需求。

1）对象角色建模（ORM）

使用最广；是一种模型驱动的工程方法，以典型的需求信息或查询的实例开始，这些实例在用户熟悉的外部环境中呈现，然后在概念层次上用受控的自然语言所表达的简单事实来描述这些实例。受控自然语言是受限制的无歧义的自然语言版本，因此所表达的语义很容易被人理解，也是形式化的语言。

2）完全面向通信的建模

在注释和方法上与ORM相似。

考察两种类别：面向对象和面向通信

（5）基于时间的数据模型（Timed-Based）

1）数据拱顶（Data Vault）

是一组支持一个或多个业务功能领域，面向细节、基于时间且唯一链接的规范化表。数据拱顶模型是一种混合方式，综合了三范式和星型模型的优点。

有三种类型的实体：中心表、链接表和卫星表。

中心表代表业务主键，链接表定义了中心表之间的事务集成，卫星表定义了中心表主键的语境信息。

2）锚建模

锚模型适合信息的结构和内容都随时间发生变化的情况。它提供用于概念建模的图形语言，能够扩展处理临时数据。

四个基本概念：锚、属性、连接、节点。锚模拟的是实体，属性模拟了锚的特征，连接表示锚之间的关系，节点用来模拟共享的属性。

数据建模方法——4、数据建模方法

（6）非关系型数据库：基于非关系技术构建的数据库的统称。有四类NoSQL：

1）文档数据库

通常将业务主题存储在一个文档结构中，而不是将其分解为多个关系结构。

2）键值数据库

只在两列中存储数据，键和值，特性是可以在值列同时存储简单和复杂的信息

3）列数据库

最接近关系型数据库。将数据视为行和值，不同的是，关系型数据库使用预定义的结构和简单的数据类型，列数据库如Cassandra可以使用复杂的数据类型，包括未格式化的文本和图形；此外列数据库将每列存储在自己的结构中。

4）图数据库

是为哪些使用一组节点就可以很好地表示它们之间的关系的数据而设计的。这些节点之间的连接数不确定。最大功能是寻找最短路径或最近邻居。这些功能在传统的关系型数据库中实现是极其复杂的。包括Neo4J、Allegro、Virtuoso

数据建模方法——5、数据模型级别

数据库管理的三重模式

1）概念模式

体现了正在数据库中建模企业的“真实世界”视图，代表了企业当前的“最佳模式”或“经营方式”。

2）外模式

是数据库管理系统的各个用户操作与特定需求相关企业模型的子集。这些子集称为外模式

3）内模式

数据的“机器视图”由内模式描述，描述了企业信息的存储表示形式。

数据建模方法——5、数据模型级别

（1）概念数据模型（CDM）

是用一系列相关主题域的集合来描述概要数据需求。概念数据模型仅包括给定的领域和职能中基础和关键的业务实体，同时也给出实体和实体之间关系的描述。例如，要对学生和学校之间的关系进行建模，采用信息工程（IE）语法描绘的关系型概念数据模型。

概念模型不受技术约束

数据建模方法——5、数据模型级别

（2）逻辑数据模型（LDM）

是对数据需求的详细描述，通常用于支持特定用法的语境。逻辑数据模型不受任何技术或特定实施条件的约束。逻辑数据模型通常是从概念数据模型扩展而来。

通过添加属性扩展概念数据模型

（3）物理数据模型

描述了一种详细的技术解决方案，通常以逻辑数据模型为基础，与某一类系统硬件、软件和网络工具相匹配。物理数据模型与特定技术相关。关系型数据库管理系统应被设计成具有特定功能的数据库管理系统。

维度模型的物理数据模型：

1）规范模型

规范模型是物理模型的一个变种，用于描述系统之间的数据移动。该模型描述了在系统之间作为数据报或消息传递的数据结构。

2）视图

3）分区

4）逆规范化

①提前组合来自多个其他表的数据，以避免代价高昂的运行时连接

②创建更小的、预先过滤的数据副本，以减少昂贵的运行时计算和/或大型表的扫描

③预先计算和存储昂贵的数据计算结果，以避免运行时系统资源竞争。

6、规范化

是运用规则将复杂的业务转化为规范的数据结构的过程。规范化的基本目标是保证每个属性只在一个位置出现，以消除冗余或冗余导致的不一致性。整个过程需要深入理解每个属性，以及每个属性与主键的关系。

1）第一范式：确保每个实体都有一个主键，一范式和二范式是子集吗？

2）第二范式：确保每个实体都有最小的主键，每个属性都依赖于完整的主键

3）第三范式：确保每个实体都没有隐藏的主键，每个属性都不依赖于键值之外的任何属性

4）Boyce/Codd范式(BCNF)：解决了交叉的复合候选键的问题。

5）第四范式：将所有三元关系分解成二元关系，直到这些关系不能再分解成更小的部分

6）第五范式：将实体内部的依赖关系分解成二元关系，所有联结依赖部分主键。

7、抽象化

泛化

将实体的公共属性和关系分组为超类实体

特化

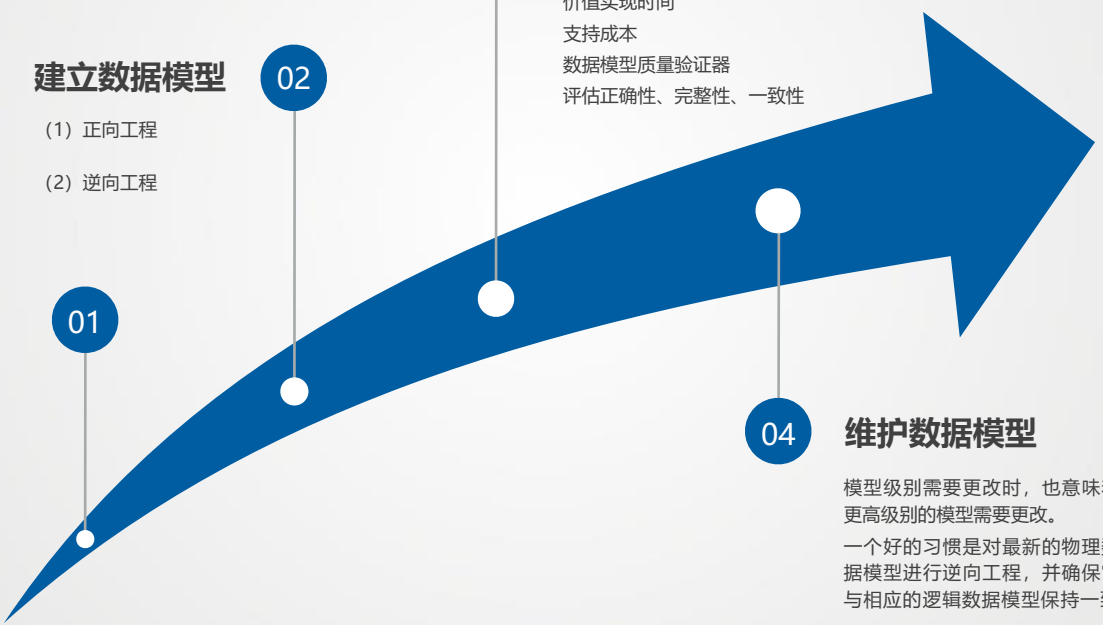
而特化将实体中的区分属性分离为子类实体。这种特化通常基于实体实例中的属性值。

02

活动

规划建模、建立模型、审核模型、维护模型

数据建模活动


03
审核数据模型
价值实现时间
支持成本
数据模型质量验证器
评估正确性、完整性、一致性
建立数据模型
02
（1）正向工程
（2）逆向工程
数据治理
规划数据建模
01
计划包括：
（1）评估组织需求
（2）确定建模标准
（3）明确数据模型存储管理
04
维护数据模型
模型级别需要更改时，也意味着更高级别的模型需要更改。
一个好的习惯是对最新的物理数据模型进行逆向工程，并确保它与相应的逻辑数据模型保持一致。

规划数据建模

计划包括：

1）评估组织需求
2）确定建模标准
3）明确数据模型存储管理

交付成果：

1）图表
2）定义
3）争议和悬而未决的问题
4）血缘关系

建立数据模型——正向工程

1、正向工程：是指从需求开始构建新应用程序的过程。

首先需要通过建立概念模型来理解需求的范围和核心术语；然后建立逻辑模型来详细描述业务过程；最后通过具体的建表语句来实现物理模型。

（1）概念数据模型建模

1）选择模型类型
2）选择表示方法
3）完成初始概念模型
4）收集组织中最高级的概念（名称）
5）收集与这些概念相关的活动
6）合并企业术语
7）获取签署

（2）逻辑数据模型建模

1）分析信息需求

2）分析现有文档

3）添加关联实体

用于描述多对多关系。关联实体从关系涉及的实体获取标识属性，并将它们放入一个新的实体中。该实体只描述实体之间的关系，并允许添加属性来描述这种关系，如有效日期和到期日期。

4）添加属性：属性添加到概念实体中

5）指定域：保证模型属性中格式和数值集的一致性。

6）指定键：分配给实体的属性可以是键属性，也可以是非键属性。键属性有助于从所有实体中识别出唯一的实体实例，可以是单独一个属性成为键，也可以是与其他键元素组合的部分键。

（3）物理数据建模

1）解决逻辑抽象:①子类型吸收　　②超类型分区

2）添加属性细节

3）添加参考数据对象：

①创建匹配的单独代码表

②创建主共享代码表

③将规则或有效代码嵌入到相应对象的定义中。

4）指定代理键：给业务分配不可见的唯一键值，与它们匹配的数据没有任何意义或关系。

5）逆规范化

6）建立索引

7）分区

8）创建视图

2、逆向工程

记录现有数据库的过程
物理数据建模是第一步，以了解现有系统的技术设计
逻辑数据建模是第二步，以记录现有系统满足业务的解决方案
概念数据建模是第三步，用于记录现有系统中的范围和关键术语。

审核数据模型

价值实现时间

支持成本

数据模型质量验证器

评估正确性、完整性、一致性

维护数据模型

模型级别需要更改时，也意味着更高级别的模型需要更改。

一个好的习惯是对最新的物理数据模型进行逆向工程，并确保它与相应的逻辑数据模型保持一致。

03 工具

建模、血缘、数据分析、元数据资料库、数据模型模式、行业模型

数据建模工具

1 数据建模工具

自动实现数据建模功能的软件

2 数据血缘工具

是允许捕获和维护数据模型上的每个属性的源结构变化的工具。实现变更影响分析

3 数据分析工具

帮助探索数据内容，根据当前的元数据进行验证、识别数据质量和现有数据工件（如逻辑和物理模型、DDL和模型描述）的缺陷

4 元数据资料库

存储有关数据模型的描述性信息，包括图标和附带的文本以及通过其他工具和流程导入的元数据

5 数据模型模式

是可重复使用的模型结构，可以在很多场景下被广泛应用，有组件、套件和整合数据模型模式。

6 行业数据模型

04 方法

命名约定、数据库设计

命名约定的最佳实践

元数据注册时一种表示组织中元数据的国际标准，包含与数据标准相关的几个部分，包括命名属性和编写定义

数据建模和数据库设计标准是有效满足业务数据需求的指导原则，它们符合企业架构和数据架构的要求，以确保数据质量标准。

数据库设计中的最佳实践：

1）性能和易用性。确保用户可快速、轻松地访问数据，从而最大限度地提高应用程序和数据的业务价值

2）可重用性。确保数据库结构在适当的情况下，能够被多个应用重复使用，并且可用于多种目的（如业务分析、质量改进、战略规划、客户关系管理和流程改进。避免将数据库、数据结构或数据对象耦合到单个应用程序中。）

3）完整性：无论语境如何，数据应始终具有有效的业务含义和价值，并且应始终反映业务的有效状态。实施尽可能接近数据的数据完整性约束，并理解检测并报告数据完整性约束的违规行为。

4）安全性：应始终及时向授权用户提供真实准确的数据，且仅限授权用户使用。

5）可维护性：确保创建、存储、维护、使用和处置数据的成本不超过其对组织的价值，以能够产生价值的成本方式执行所有数据工作；确保尽可能快速地相应业务流程和新业务需求的变化。

05

数据建模和设计治理

建模和设计质量管理、度量指标

1、开发数据建模和设计标准

1）标准数据建模和数据库设计可交付成果的列表和描述

2）适用于所有数据模型对象的标准名称、可接受的缩写和非常用单词的缩写规则列表

3）所有数据模型对象的标准命名格式列表，包括属性和分词

4）用于创建和维护这些可交付成果的标准方法的列表和说明

5）数据建模和数据库设计角色和职责的列表和描述

6）数据建模和数据库设计中捕获的所有元数据属性的列表和描述，包括业务元数据和技术元数据。

7）元数据质量期望和要求

8）如何使用数据建模工具的指南

9）准备和领导设计评审的指南

10）数据模型版本控制指南

11）禁止或需要避免的事项列表

2、评审数据模型以及数据库设计质量

审查会议议程包括：

- 审查启动模型（如有）的项目
- 对模型所做的更改
- 考虑和拒绝的任何其他选项
- 新模型在多大程度上符合现有的建模或架构标准

3、管理数据模型版本与集成

变更的记录，包括：

1）为什么why项目或情况需要变更

2）变更对象（What）以及如何（How）更改，包括添加了哪些表，修改或删除了哪些列等

3）变更批准的时间（When）以及将此变更应用于模型的时间

4）谁（Who）做出了变更

5）进行变更的位置（Where）在哪些模型中

度量指标

1）各模型多大程度上反映了业务需求：　　要确保数据模型代表需求。

2）模型的完整性如何：

需求的完整性和元数据的完整性

需求完整性意味着已经提出的每个需求都应在模型中得到满足

元数据的完整性是指模型周围的所有描述性信息也要完整

3）模型与模式的匹配度是多少：

确保正在审查模型的具象级别（概念模型、逻辑模型或物理模型）和模式（关系、维度、NoSQL）与该类模型的定义相匹配。

4）模型的结构如何：验证用于构建模型的设计实践，以确保最终可以从数据模型构建数据库。

5）模型的通用性如何：评审模型的扩展性或者抽象程度

6）模型遵循命名标准的情况如何：确保数据模型采用正确且一致的命名标准

7）模型的可读性如何：确保数据模型易于阅读

8）模型的定义如何：确保定义清晰、完整和准确

9）模型与企业数据架构的一致性如何：确认数据模型中的结构能否在更加广泛和一致的环境中应用，以便在组织中可以使用一套统一的术语和模型结构。

10）与元数据的匹配程度如何：确认存储在模型结构中的数据和实际数据是一致的。

本章完结 感谢观看

完整课程视频请扫描二维码咨询