Oracle数据库23c JSON关系二元性

公益讲座11：00准时开始，请大家先浏览云技术微信公众号技术文章。资料会在各群同步发布，已入群客户请勿重复入群！

20-19

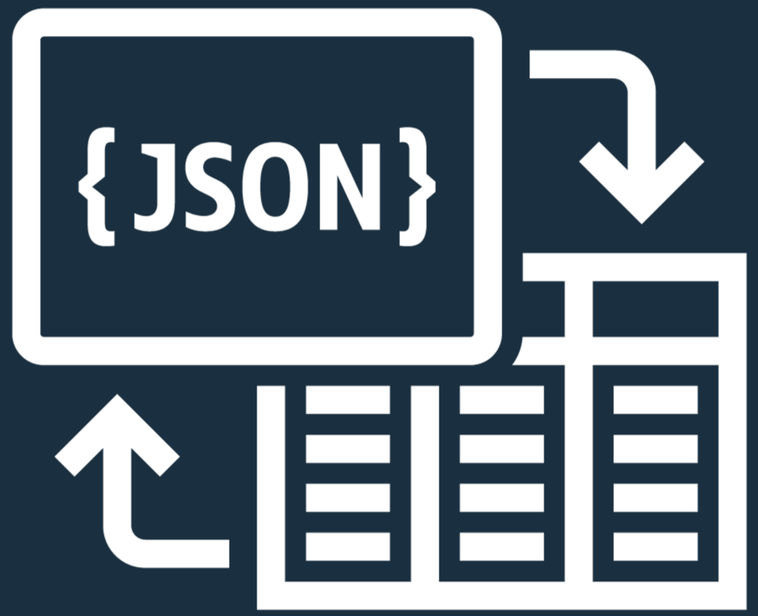
数据库和云讲座群

甲骨文云技术公众号

ORACLE

Oracle数据库23c JSON关系二元性

文档、对象和关系的革命性融合



陈文斌
资深解决方案工程师
Oracle JAPAC SE Hub
Dec, 2022

如何做蛋糕

强大
使用食谱结合基本成分来制作
您想要的任何东西

挑战
做蛋糕需要技巧和努力

简单
所需的蛋糕原料已预先混合
并准备烘烤

挑战
只对做蛋糕有用

应用开发示例：学生时间表

强大

使用声明性 SQL 组合基本规范化数据以创建您想要的任何应用程序

简单

JSON易于表达高级语言中的对象
具有简单的访问方式
减少了所需的技能和工作量

挑战

需要技能和努力将应用程序对象映射到基本数据库数据

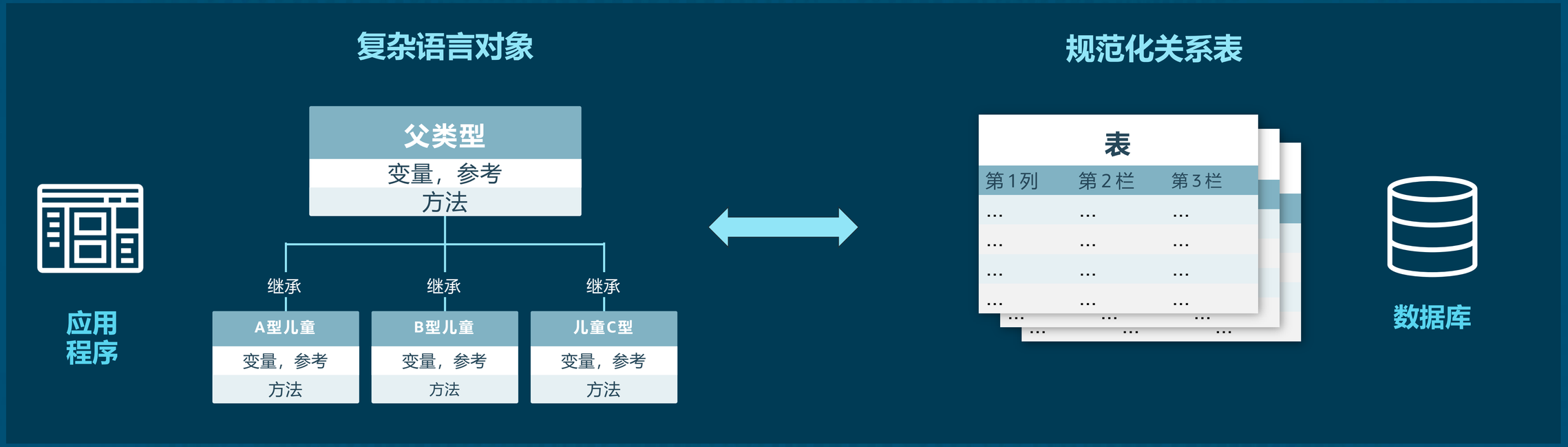
挑战

- 数据冗余
- 难以保持一致性

基于关系挑战：对象与关系不匹配

应用程序希望将数据用作编程语言和用例特定对象（蛋糕）

将语言对象转换为规范化的关系数据需要一些技巧和努力

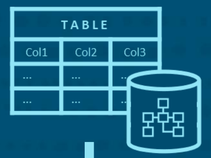


有很多弥合这种不匹配的尝试

对象数据库

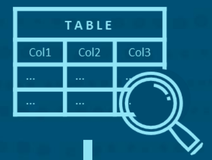
以对象格式本地存储数据，而不是表

对象关系数据库



将用户定义的类型和继承添加到数据库

对象关系映射工具 (ORM)



生成代码以将语言对象映射到表

文档数据库

将数据存储为 JSON 而不是关系表

- 这些取得了有限的成功，因为它们针对特定用例进行了优化，失去了关系模型实现任何用例的灵活性
- ORM 添加了一个层，以便更轻松地将应用程序对象转换为关系对象，但不能从根本上解决不匹配问题

应用程序开发示例 - 学生时间表

此应用程序所需的数据存储在关系模式内的规范化表中：

学生

学生ID	名字	信孚
S3245	吉尔	...
S8524	约翰	...
S1735	简	...
S3409	吉姆	...

老师

老师ID	名称	天福
T123	阿尼卡	...
T543	亚当	...
T789	安妮塔	...
T612	亚历克斯	...

课程

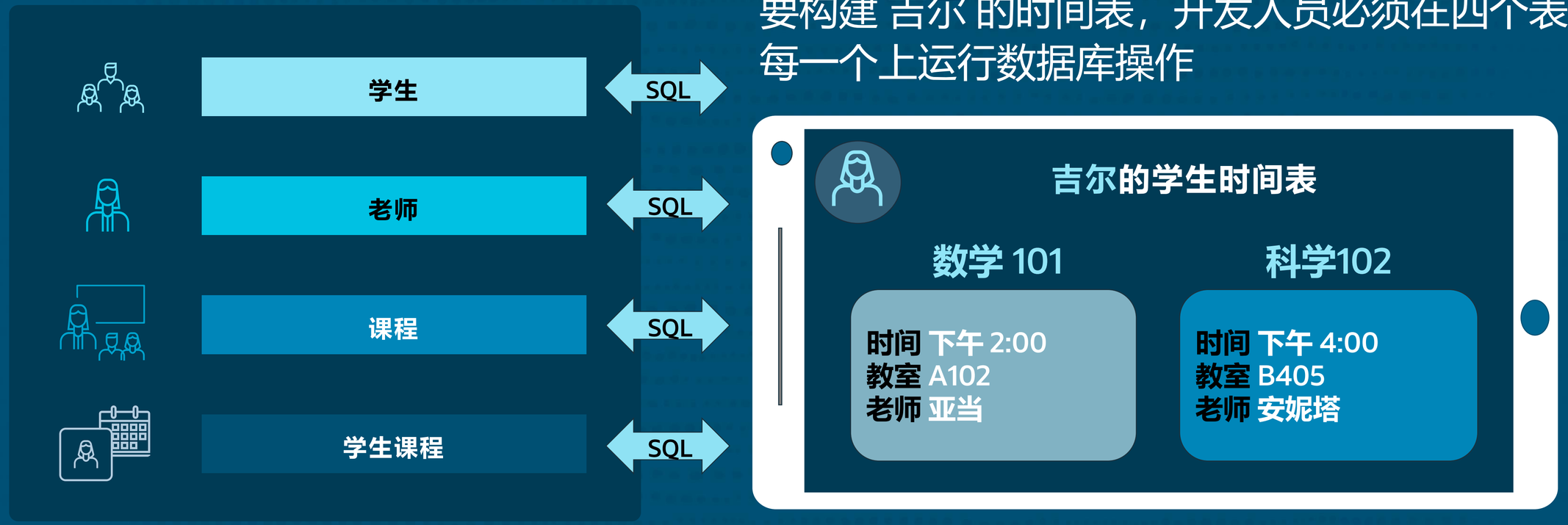
课程ID	课程名	教室	时间	老师
C123	MA_01	A102	14:00	T543
C345	SCI_02	B405	16:00	T789
C567	HIS_02	A102	14:00	T612
C789	LA_01	A256	12:00	T543

学生课程

学生ID	课程ID
S3245	C123
S8524	C567
S3245	C345
S3409	C123

应用程序开发示例 - 学生时间表

使用规范化表开发应用程序非常灵活，但对开发人员来说并不总是那么容易



JSON 文档使开发人员能够简化数据库访问

吉尔 的课程表很容易用 JSON 表示

吉尔 的每个班级的数据都嵌入在
吉尔 的 JSON Schedule 文档中

```
{
 "学生" :  "吉尔" ,
 "日程" :
   [  {
     "时间" :  " 14:00 " ,
     "课程" :  "数学101 " ,
     "教室" :  " A102 " ,
     "老师" :  "亚当"
   },
   {
     "时间" :  " 16:00 " ,
     "课程" :  "科学102 " ,
     "教室" :  " B405 " ,
     "老师" :  "安妮塔"
   }
 ]
}
```

文档数据库使对 JSON 的操作变得简单

文档数据库很受欢迎，因为它们允许轻松访问和存储 JSON 文档

使用简单的 GET/PUT API

```
{
 "学生" :  "吉尔" ,
 "日程" :
   [  {
    "时间" :  " 14:00 " ,
    "课程" :  "数学101 " ,
    "教室" :  " A102 " ,
    "老师" :  "亚当"
  },
  {
    "时间" :  " 16:00 " ,
    "课程" :  "科学102 " ,
    "教室" :  " B405 " ,
    "老师" :  "安妮塔"
  }
 ]
}
```

Oracle 数据库已经是一个很棒的文档数据库

Oracle 数据库是一个很棒的文档数据库

Oracle 提供比纯文档数据库更多更好的文档功能

自 1998 年以来支持 XML 文档

自 2014 年起支持 JSON 文档

在每个版本中不断增强

```
{
 "名称1 " :  "字符串值1 " ,
 "名称2 " :
 {
     "姓名3 " :  " 14:00 " ,
     "名称4 " :  1234
 }
}
```

Oracle 数据库是一个很棒的文档数据库

使用文档或 SQL API 可以轻松存储、检索、分析和操作文档

- Oracle 定期为开放 SQL 标准贡献文档的 SQL 增强功能

```
CREATE TABLE
student_schedule ( student_id number),
          schedule_doc JSON );


SELECT schedule_doc
FROM   student_schedule的
WHERE  s.schedule_doc.student = '吉尔' ;
```

Oracle 文档优于文档数据库

1

完整的**文档**API

SODA：简单的 Oracle 文档访问 API - 兼容 MongoDB

2

提供对文档的基于标准的**SQL**访问

可以用 JavaScript、Java 或 PL/SQL 编写存储过程

优于文档数据库

3

跨文档的**完全 ACID 一致性**

分析、空间、图形、机器学习、文档并行化

优于文档数据库

另外，这还是一个自治 JSON 数据库

JSON作为存储格式有局限性

文档数据库的局限性

JSON 文档很容易让应用程序用作数据访问格式

当用作存储格式时，它们会产生数据重复和数据一致性问题。

重复数据造成
- 存储效率低
- 更新成本高
- 难以保持一致

多个用例的相同数据

相同集合的不同文档与不同集合间存在大量重复数据，随着时间推移，重复数据加剧

学生时间表：吉尔

```
{
 "学生" :  " S3245 " ,
 "姓名" :  "吉尔" ,
 "日程" :
  [ {
    "时间" :  " 14:00 " ,
    "课程" :  "数学101 " ,
    "教室" :  " A102 " ,
    "老师" :  "亚当"
   },
   {
    "时间" :  " 16:00 " ,
    "课程" :  "科学102 " ,
    "教室" :  " B105 " ,
    "老师" :  "安妮塔
   }
  ]
}
```

学生时间表：杰克

```
{
 "学生" :  " S4356 " ,
 "姓名" :  "杰克" ,
 "日程" :
[ {
    "时间" :  " 16:00 " ,
    "课程" :  "科学102 " ,
    "教室" :  " B105 " ,
    "老师" :  "安妮塔
}
{
    "时间" :  " 14:00 " ,
    "课程" :  "物理" ,
    "教室" :  " B405 " ,
    "老师" :  "安妮塔
}
   ]
}
```

教师时间表：安妮塔

```
{
 "老师" :  " T9351 " ,
 "名字"        :  "安妮塔" ,
 "日程" :
 [ {
    "时间" :  " 14:00 " ,
    "课程" :  "科学102 " ,
    "教室" :  " A312 "
   },
   {
    "时间" :  " 16:00 " ,
    "课程" :  "科学102 " ,
    "教室" :  " B405 "
   },
   {
    "时间" :  " 18:00 " ,
    "课程" :  "科学102 " ,
    "教室" :  " A115 "
   }
  ]
}
```

JSON 就像用蛋糕粉做蛋糕

蛋糕粉旨在简化单个用例，而不是多个用例

一旦成分混合，就很难将它们用于其他任何用途

```
{
 "名称1 " :  "字符串值1 " ,
 "名称2 " :
 {
    "姓名3 " :  " 14:00 " ,
    "名称4 " :  1234
 }
}
```

文档规范化

文档数据库供应商认识到数据重复是一个问题，并添加了试图解决该问题的功能

1

文档可以引用其他文档

2

使用引用，可以将文档规范化为片段以消除重复

3

文档查询可以将多个文档中的内容聚合到单个结果文档中

文档数据库规范化

```
{
 "学生" :  " S3245 " ,
 "姓名" :  "吉尔" ,
 "日程" :
 [ {
     "时间" :  " 14:00 " ,
     "课程" :  "数学101 " ,
     "教室" :  " A102 " ,
     "老师" :  "亚当"
   }
 ]
}
```

```
{
  "学生":  "S3245" ,
  "日程":
  [ {"课程":  "M101" ,  },
    ... ]
}
```

```
{
 "学生" :  " S3245 " ,
 "姓名" :  "吉尔"
…}
```

```
{
"课程编号":  "M101" ,
"名称":  "数学 101" ,
"老师":  "T543"
…}
```

```
{
 "老师" :  "T543" ,
 "姓名" :  "亚当"
…}
```

可以更改学生计划文档以引用单独的学生文档而不是包含它

并参考单独的课程文档

其中引用了单独的教师文档

当文档被规范化了，
它们的简单性消失了

文档数据库碎片化

规范化文档会产生**两败俱伤的结果**

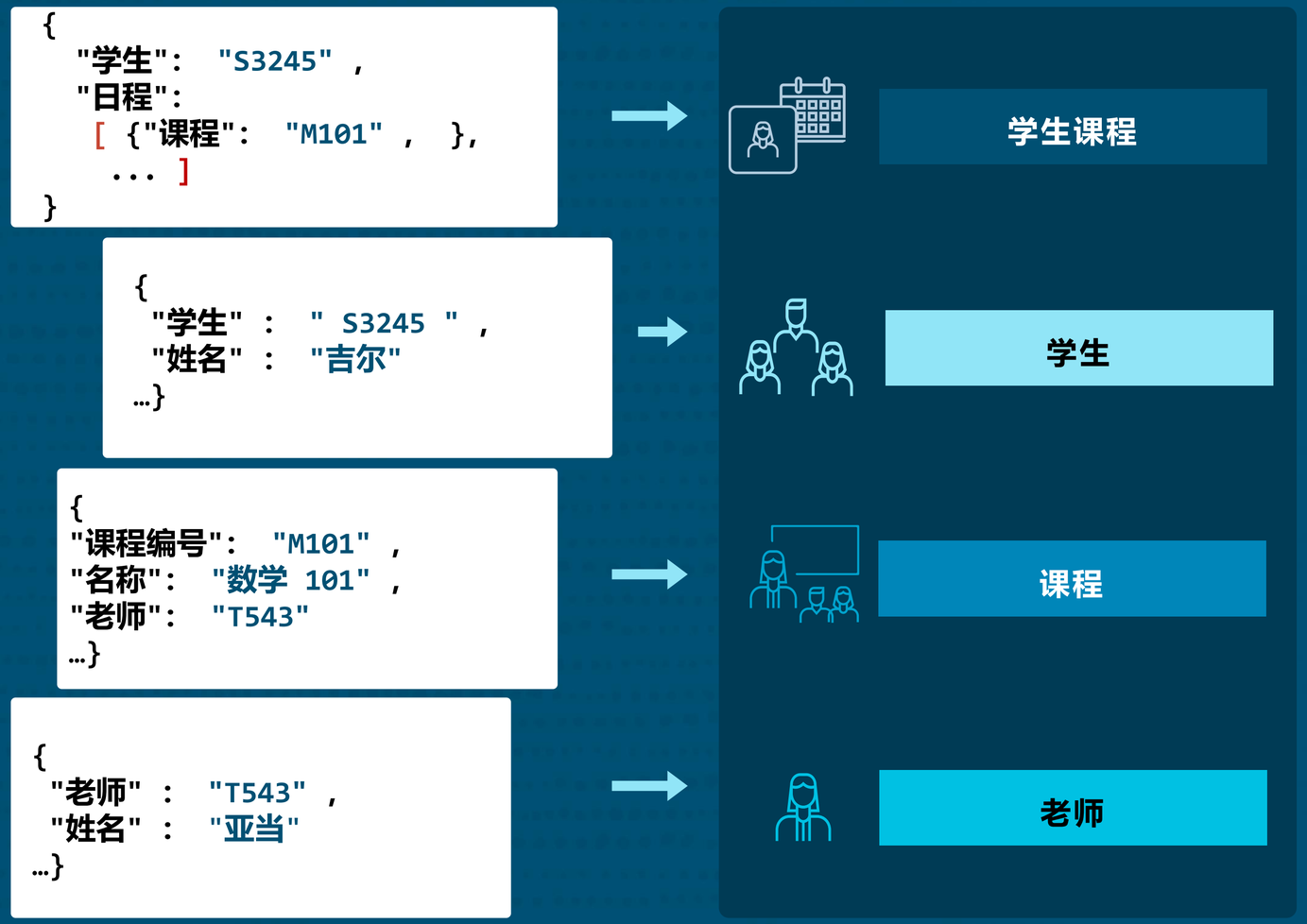
- 文档结构现在反映了规范化的表模式

失去了应用程序级别的文档的简单性

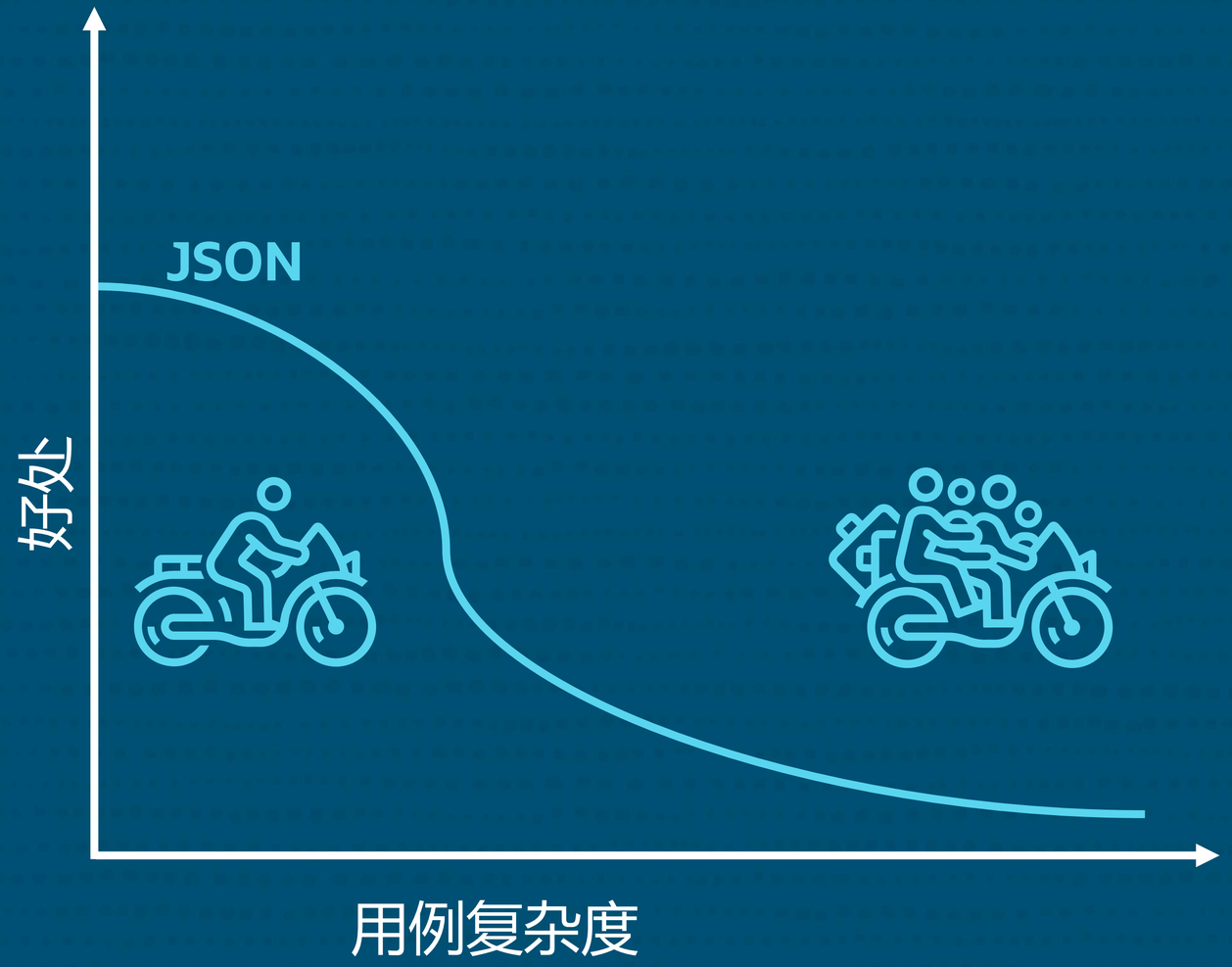
- 在数据库级别

由于引用递归和分片位置丢失，性能受到影响

- 每个应用程序都必须强制执行引用完整性



复杂度与收益 – 文档

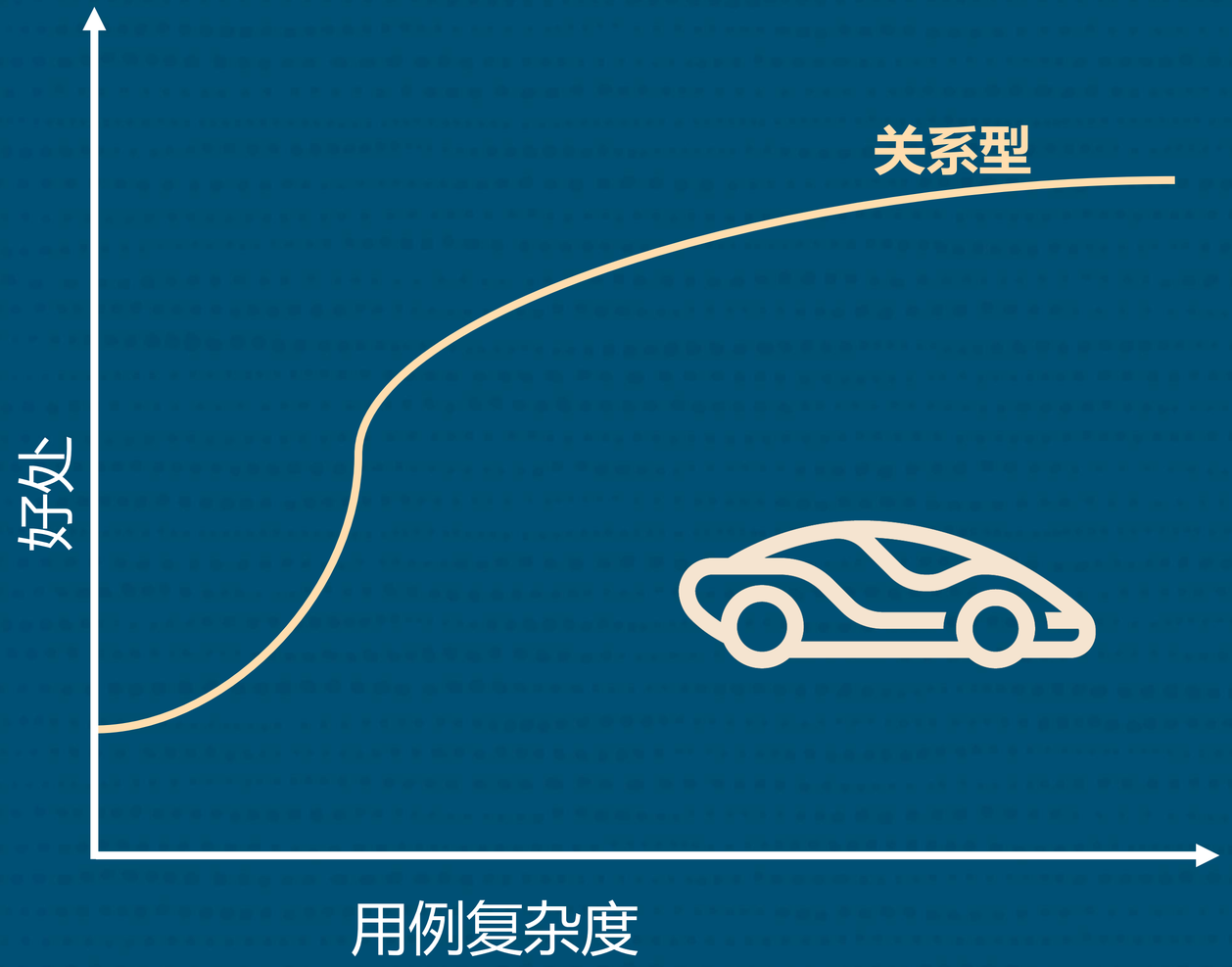


文档非常适合简单的应用程序

随着应用程序变得越来越复杂，变得危险

因为这，
许多数据专家认为纯文档数据库是一种**反模式**

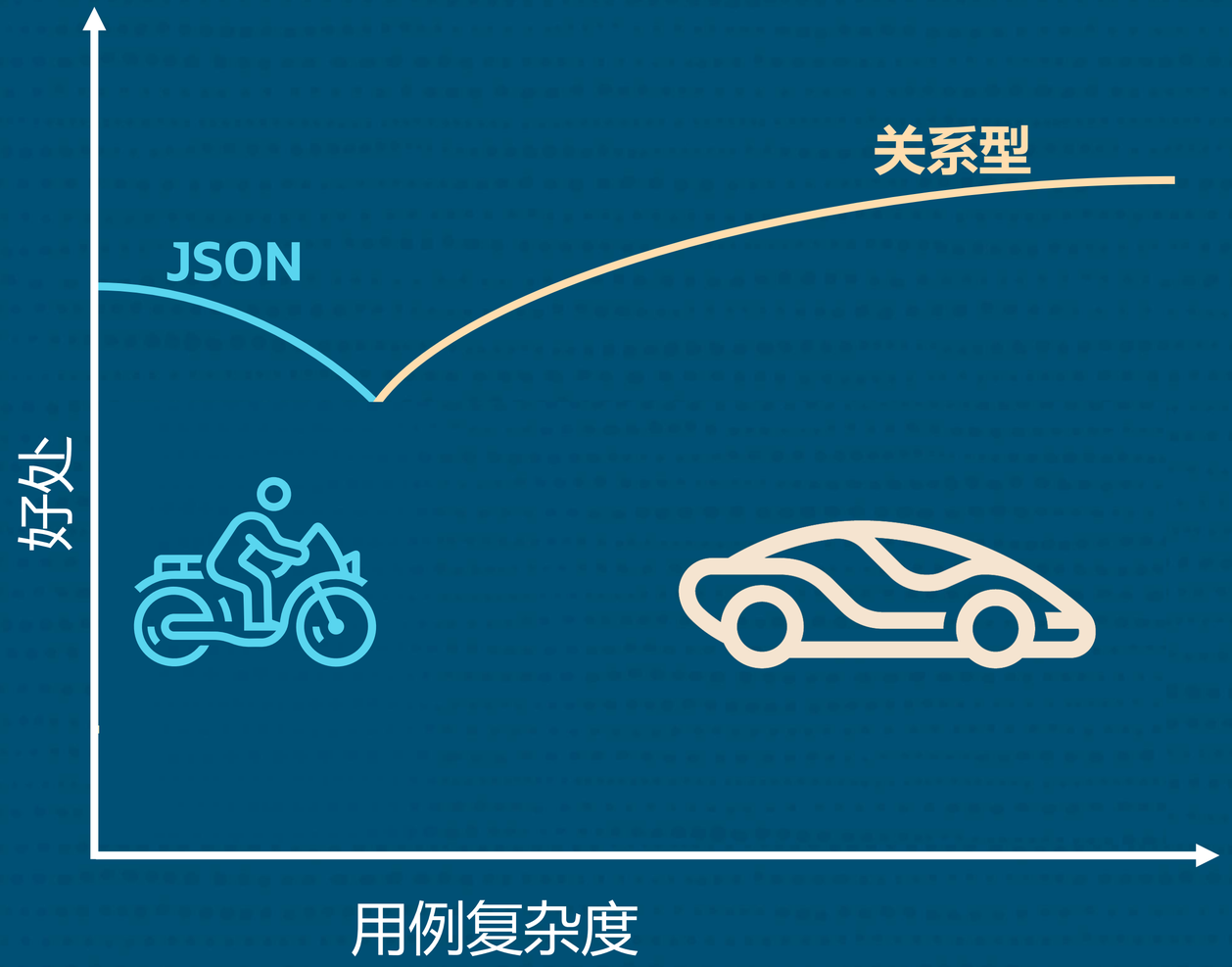
复杂度与收益 - 关系



对于简单的应用程序来说，关系并不容易

随着应用程序复杂性的增加，它的功能变得至关重要

Oracle 使开发人员能够实现两者的结合



使用 Oracle，开发人员可以选择能够更好利用每个用例收益的数据格式

这很棒

我们还能做得更好吗？

而不是选择
关系或文档

我们能否获得 关系 + 文档 的好处?

对于每个用例，我们能否获得两者的所有好处?

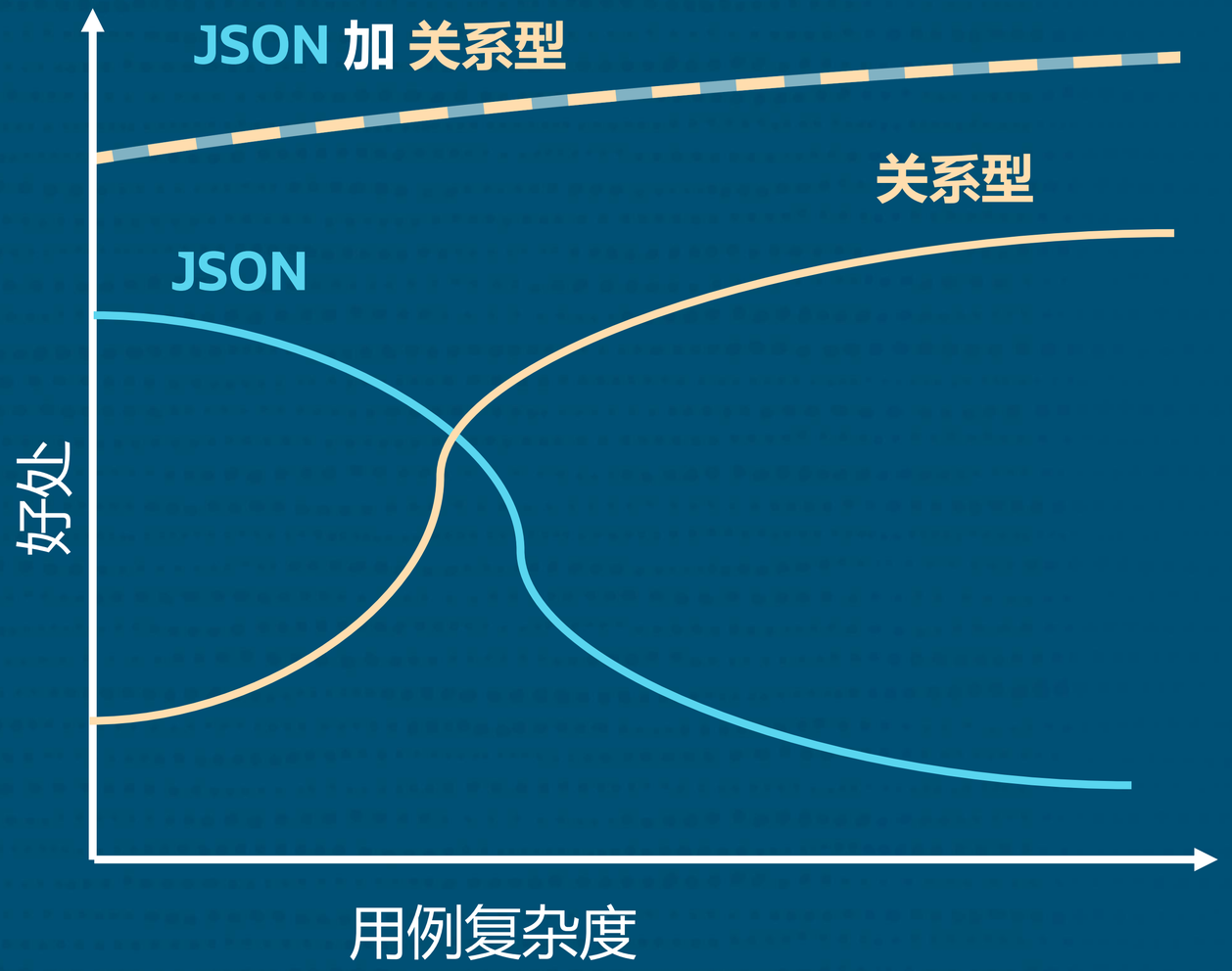
关系型

- 用例灵活性
- 可查询性
- 一致性
- 空间效率

+ 加

文档

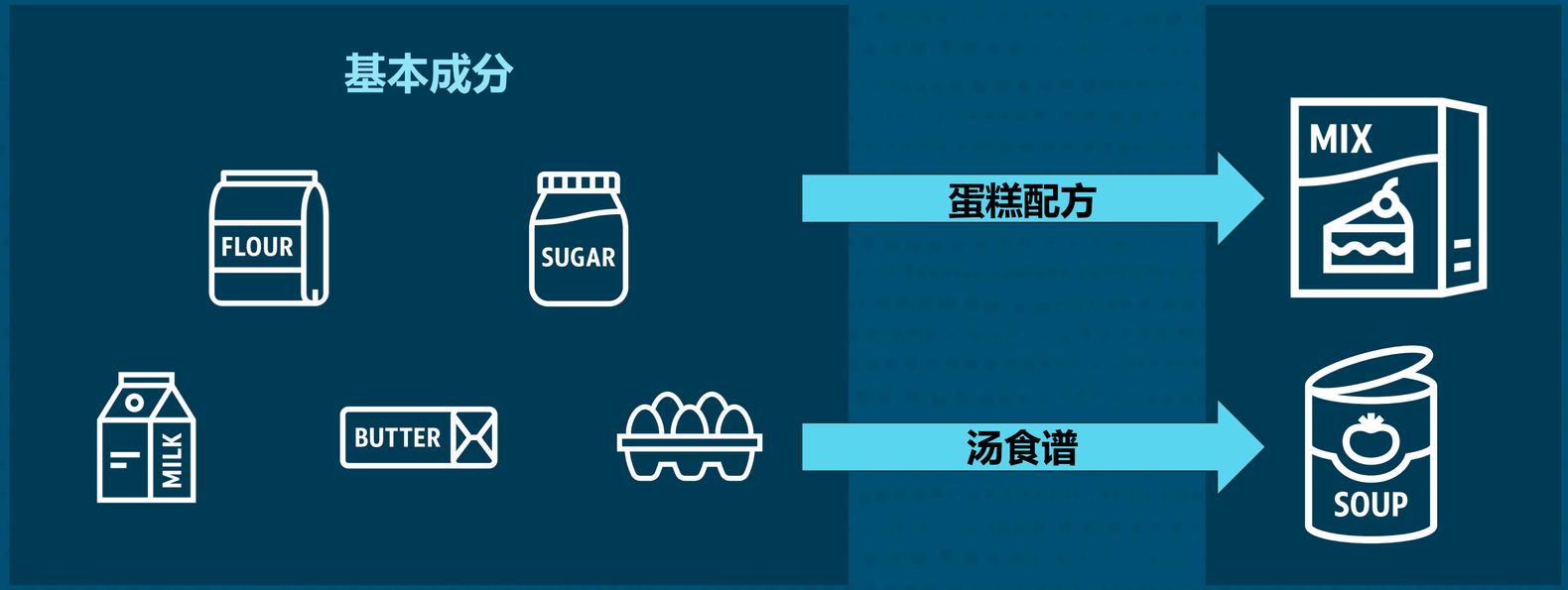
- 轻松映射到语言类型
- 敏捷无模式开发
- 分层数据格式
- 标准交换格式



我们能否获得基本成分的力量加上预制食品的简单性?

我们可以!

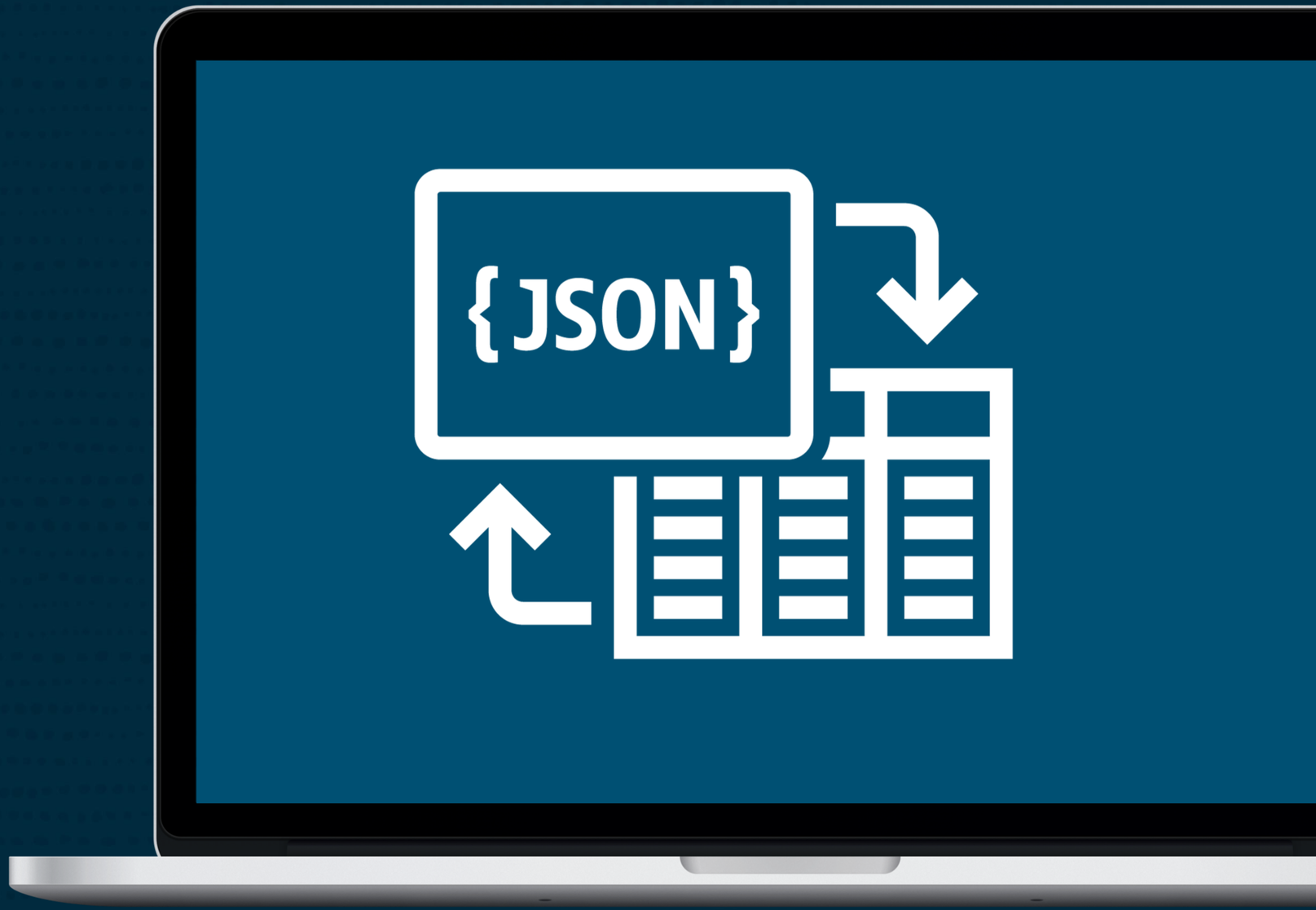
想象一个世界，您只需提供食谱就可以立即用基本成分制作任何食物



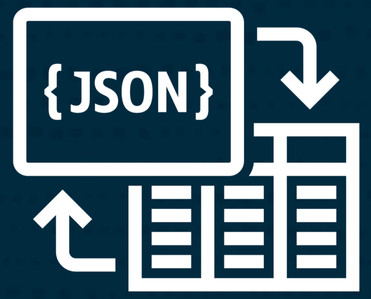
介绍

JSON 关系二元性

甲骨文数据库 23c



在架构上提供
JSON 的用例简单性



具有关系的面向多用例灵活性

JSON 文档关系二元性

数据被**存储为**表中的行

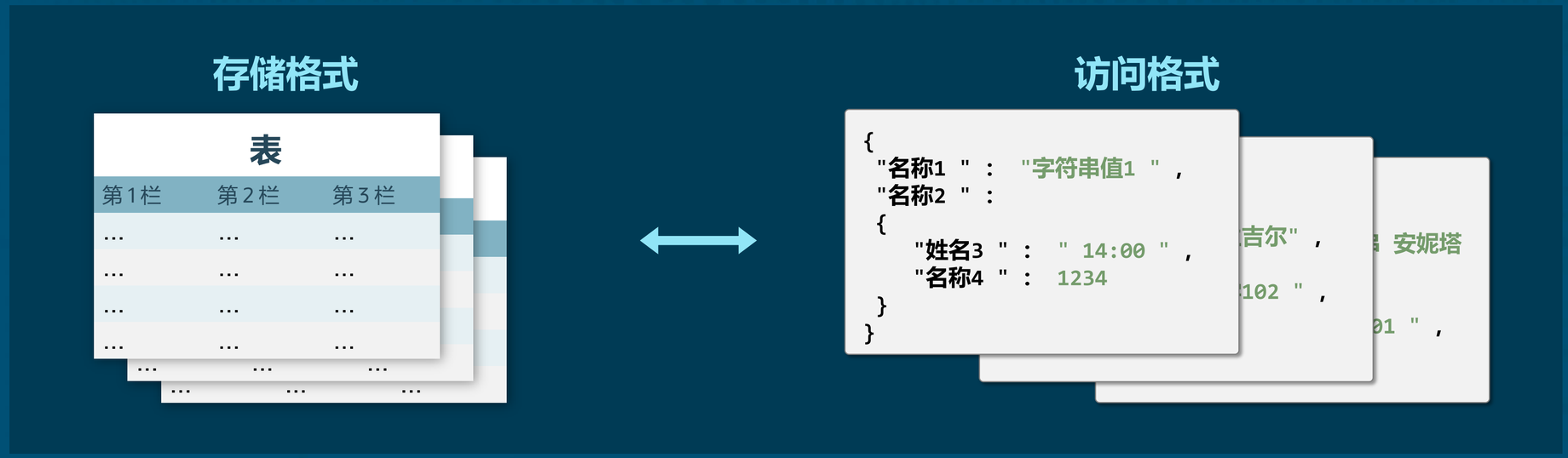
来提供关系模型和 SQL 访问的好处

行可以包含 JSON 列来存储其模式是动态或不断变化的数据

JSON 文档关系二元性

数据被**存储为**表中的行，
提供关系模型和 SQL 访问的好处

可以**访问数据 为 JSON 文档**，
以提供文档的应用程序简单性



JSON Relational Duality 完善了Oracle 对数据库内文档的所有用例的支持

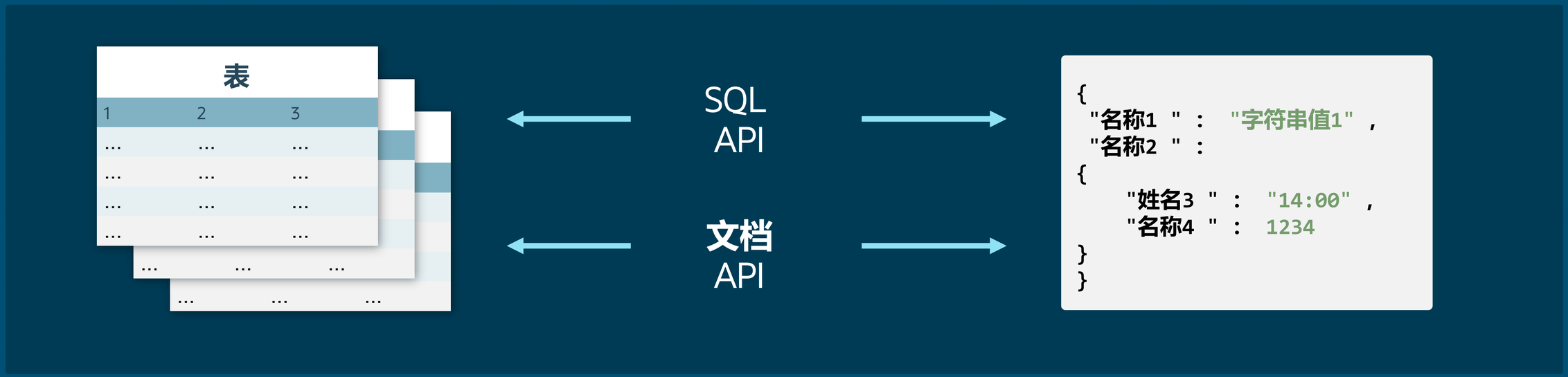
数据库内文档的所有用例

开发人员可以将数据存储为关系

- 使用标准SQL作为关系访问
- **JSON 二元性视图**

开发人员可以将数据存储为文档

- 使用JSON 数据指南作为关系访问以在 JSON 上创建关系视图
- 使用 SQL、SODA 或 MongoDB API 的 ANSII JSON 扩展作为文档访问



定义 JSON 二元性视图

JSON 二元性视图是将规范化的行组装成 JSON 文档的秘诀

```
CREATE OR REPLACE JSON DUALITY VIEW FROM Student AS
student_schedule
{
   name:        sname
   student_id: stuid
   schedule:    student_courses @delete @insert @update
      {
         course: course @unnest
         {
              time
              course:     cname
              course_id: cid
              room
              teacher:    teacher @unnest
              {
                   teacher:     tname
                   teacher_id: tid
              }
      } }
};
```

指定包含 JSON 文档数据的表

指定更新规则，能更新学生课程，不能更新课程、老师

指定嵌套对象中的属性何时应取消嵌套到父对象中

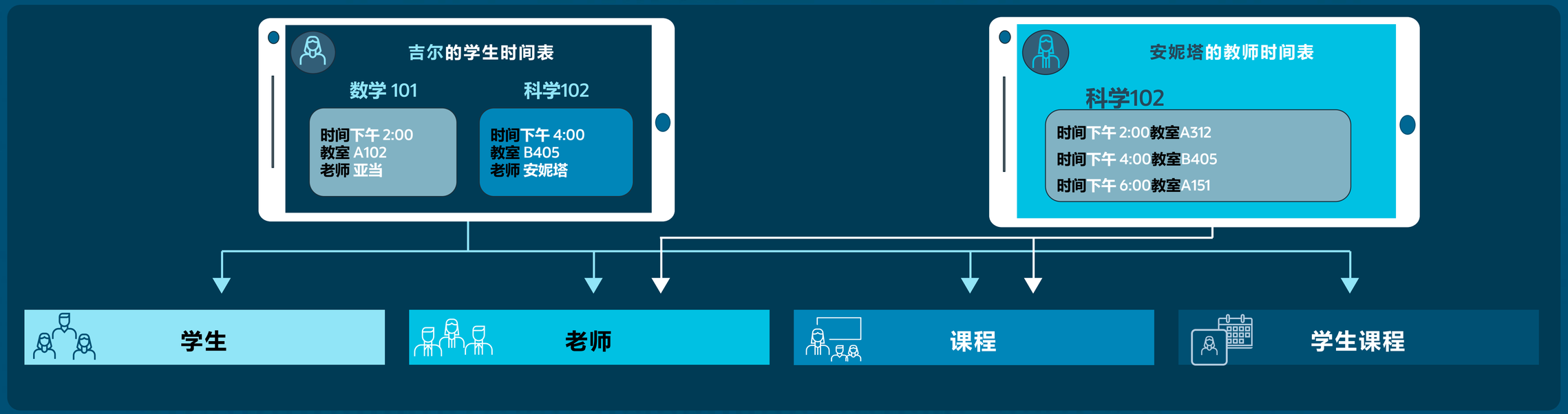
绑定表中的列到JSON属性

按用例划分的改变游戏规则的文档专业化

JSON关系二元性可以改变文档专业化游戏规则

- 可以为每个应用程序用例创建二元性视图，为该用例生成专门的 JSON 文档
- 可以轻松为相同数据添加新用例，而不会产生重复或一致性问题

例如，**相同的基础行数据**可用于创建学生和教师计划文档



使用二元性视图的示例

您可以使用 SQL 或文档 API 访问二元性视图

```
SELECT data
FROM   student_schedule s
WHERE  s.data.name = '吉尔';
```

```
student_schedule.find({"name":"吉尔"})
```

学生时间表：吉尔


```
{
 "student_id" :  " S3245 " ,
 "名字"       :  "吉尔" ,
 "日程" :
 [ {
    "时间" :  " 14:00 " ,
    "课程" :  "数学 101 " ,  "课程 ID " :  "C123" ,
    "教室" :  " A102 " ,
    "老师" :  "亚当" ,      " teacher_id " :  "T543" ,
  },
  {
   "时间" :  " 16:00 " ,
   "课程" :  "科学 102 " 、 "课程 ID " :  "C345" ,
   "教室" :  " B405 " ,
   "老师" :  "安妮塔" ,     " teacher_id " :  "T789" ,
  }
 ]
}
```

为开发人员提供极致的简单性

JSON Duality Views使用 REST 访问

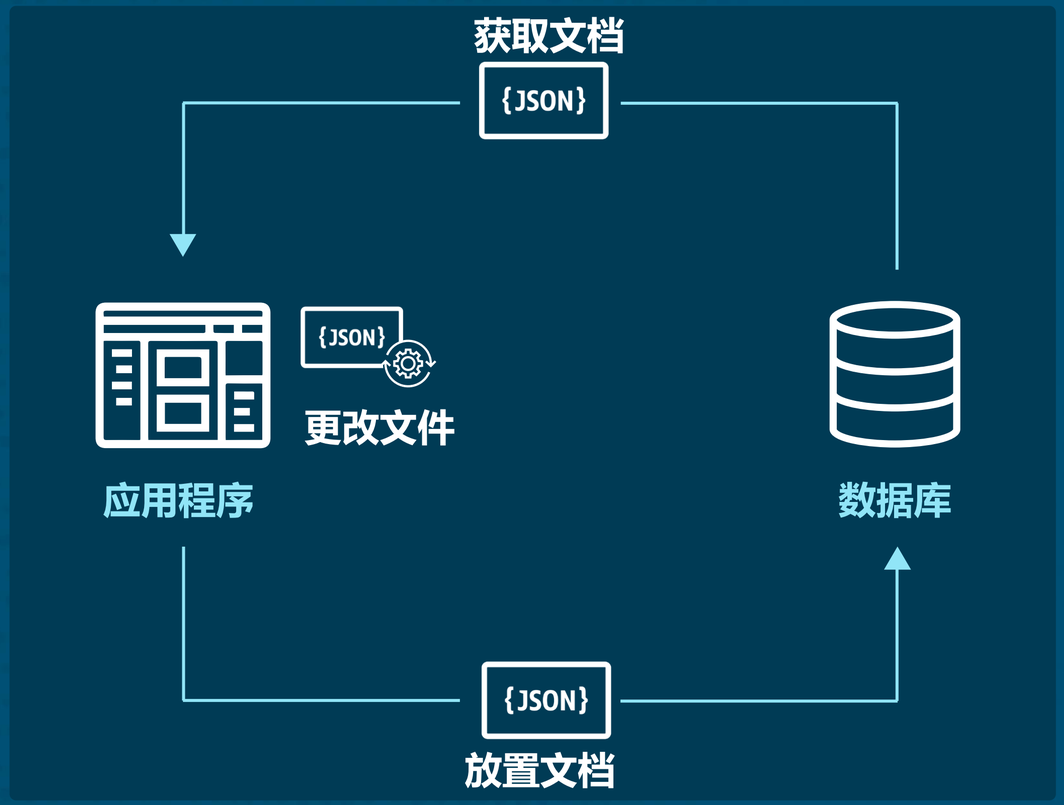
- 从视图中获取文档，一次性获取所有数据
- 对文档进行所需的任何更改
- 将文档放回视图中

喜欢 API 而不是 REST 的应用可以使用

- 简单的 Oracle 文档访问 API (SODA)
- Oracle MongoDB 兼容 API

数据库自动检测新文档中的更改并修改底层行

- 共享相同数据的所有二元性视图立即反映了这种变化
- 开发者不再需要担心不一致




```
GET school.edu/student_schedule?q={"student":"吉尔"}
```

为开发人员提供极大的灵活性

开发人员可以轻松构建以文档为中心的应用程序

- 将新文档存储为关系数据
- 将现有的关系数据作为文档进行操作
- 在关系数据库之上创建 JSON 微服务

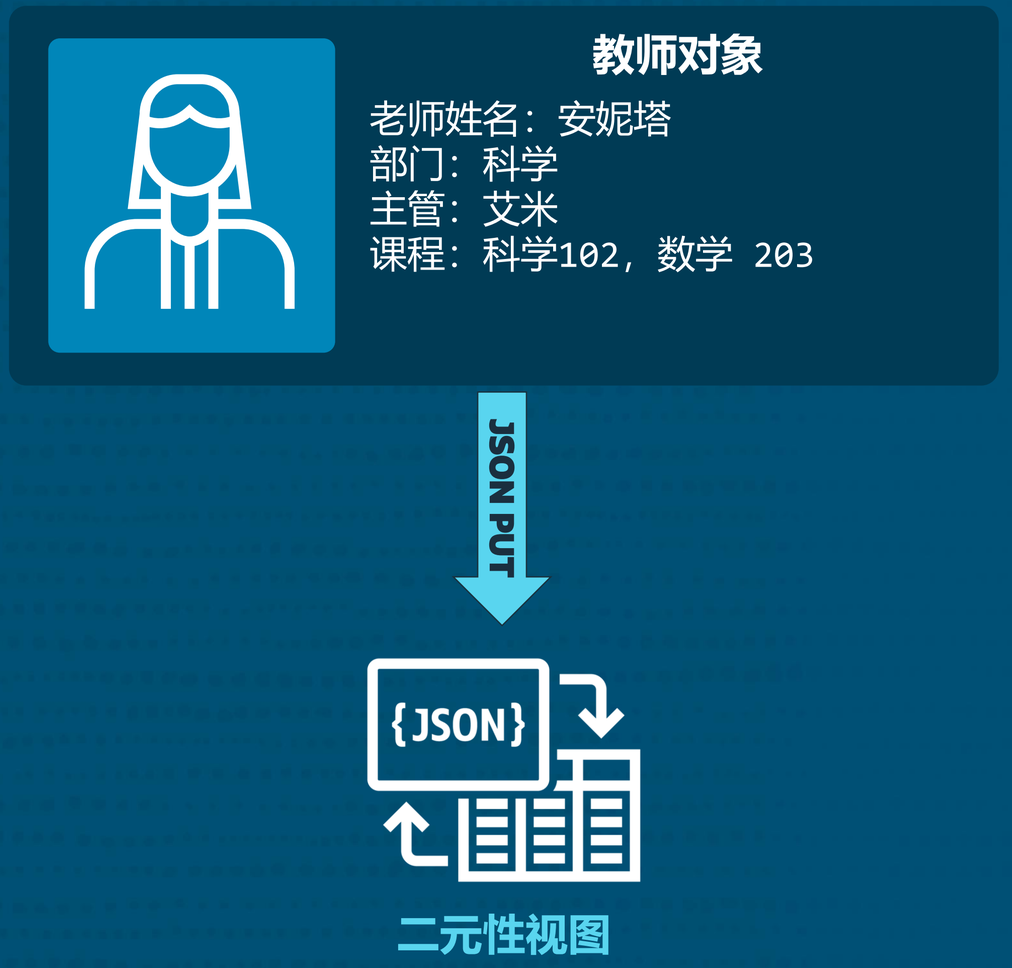
其他便利

- 声明式文档，告诉数据库你要什么，而不必告诉它如何做
- 与客户端语言或框架无关，没有ORM层
- 透明模式演进，隐藏数据库变化和应用变化
- 提供API结构定义接口，便于与低代码、IDE的集成

应用程序对象的双重优势

除了简单和标准化之外，数据库现在可以对应用对象（而不是行）实现丰富的操作

- 分析
- ETL（提取、转换、加载）
- 复制
- 文本索引
- 触发器
- 验证
- 安全规则
- 商业规则



文档和表的颠覆性特性

无锁并发控制

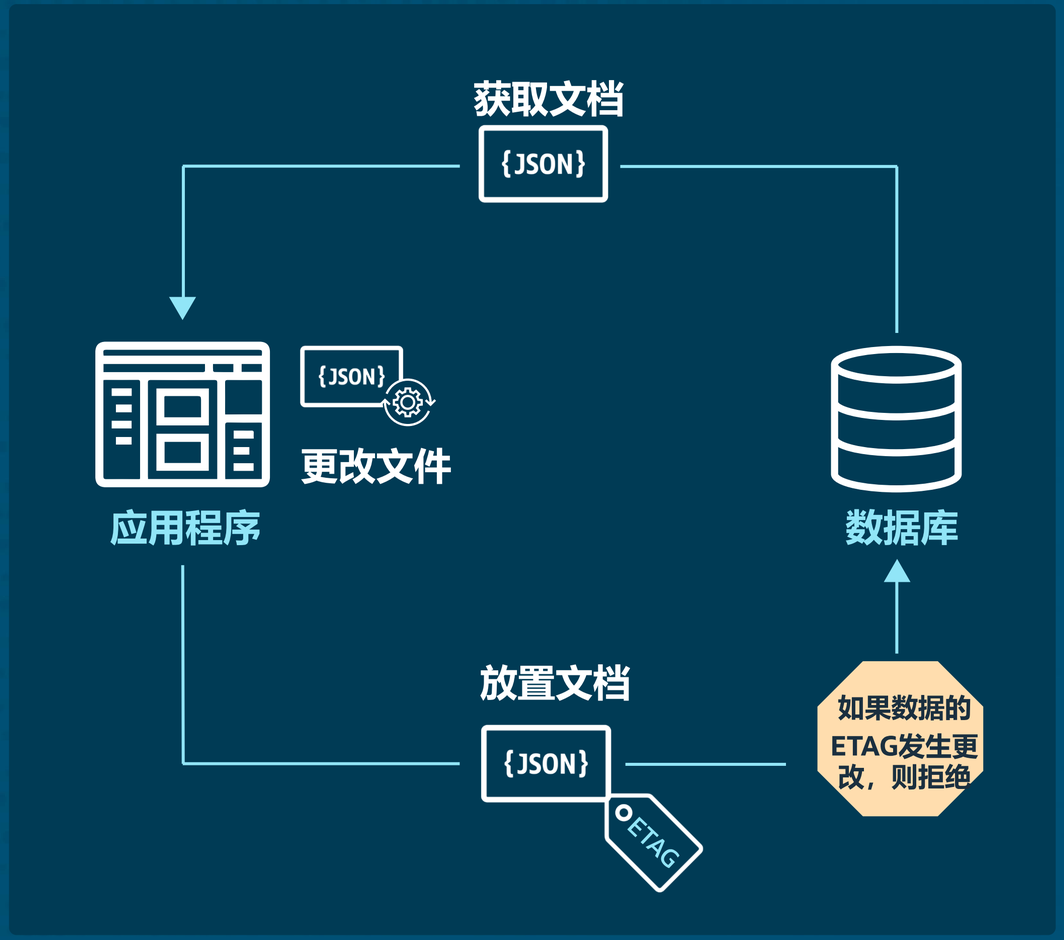
基于值的无锁（乐观）并发控制

Oracle 独特地将 HTTP ETAG 扩展至核心数据库，实现无锁并发控制

- ETAG是基于**数据值**计算而来，而不是添加锁或版本来检测更新冲突
- GET 时自动计算 ETAG 并将其插入返回的文档本身
- PUT修改后的文档时，验证文档的ETAG是否仍然与 GET 的匹配

如果发生更改，写入操作会自动被拒绝并返回错误。

然后，应用程序可以根据更改的数据重新发出写入



相同数据在不同视图中基于值的工作方式

因为它们是基于值的，所以 ETAG 会自动同步更新由许多不同文档共享的数据

- 例如，当许多学生文档共享课程数据时
- 或由具有不同文档根的文档共享，例如教师和学生文档

跨文档和表的基于值的工作方式

- 基于值的 ETAG 还自动确保直接行更新和文档更新之间的一致性

传统的锁或基于版本的并发控制在这些情况下效果不佳

基于值的允许忽略不重要的更改

ETAG 签名可以排除更新不应导致 PUT 被拒绝的数据

这就像在表中锁定列，或在文档中锁定元素

- 关系和文档数据库期望已久的功能

```
CREATE OR REPLACE JSON DUALITY VIEW FROM Student AS
student_schedule {
   name:         sname
   student_id: stuid
  schedule:    student_courses {
        course: course {
           time
           course:     cname
           course_id: cid
           room @nocheck
           teacher:    teacher {
                teacher:     tname
                 teacher_id: tid
             }
}    }   };
```

如果班级位置发生变化，从 ETAG 中排除教室可避免学生应用程序失败

纯关系数据的基于值的工作方式

ETAG 不仅仅用于文档

ETAG 现在也可以用于纯关系数据，以使用 SQL 进行无锁行更新

- 非常适合直接访问表格的交互式或移动应用程序

```
-- Select the ETAG along with data columns
SELECT stid, name, sinfo, SYS_ROW_ETAG(stid, sinfo)
  FROM student
 WHERE name = '吉尔';
```

人类思考时间

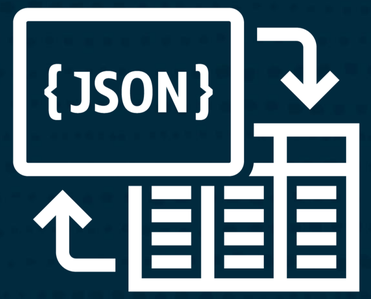
```
-- Validate the ETAG before updating
UDPATE student SET sinfo = 'New Data’
 WHERE  name = '吉尔'
   AND  SYS_ROW_ETAG(stid, sinfo) = 'xxxx';
```

无锁（乐观）并发控制的好处

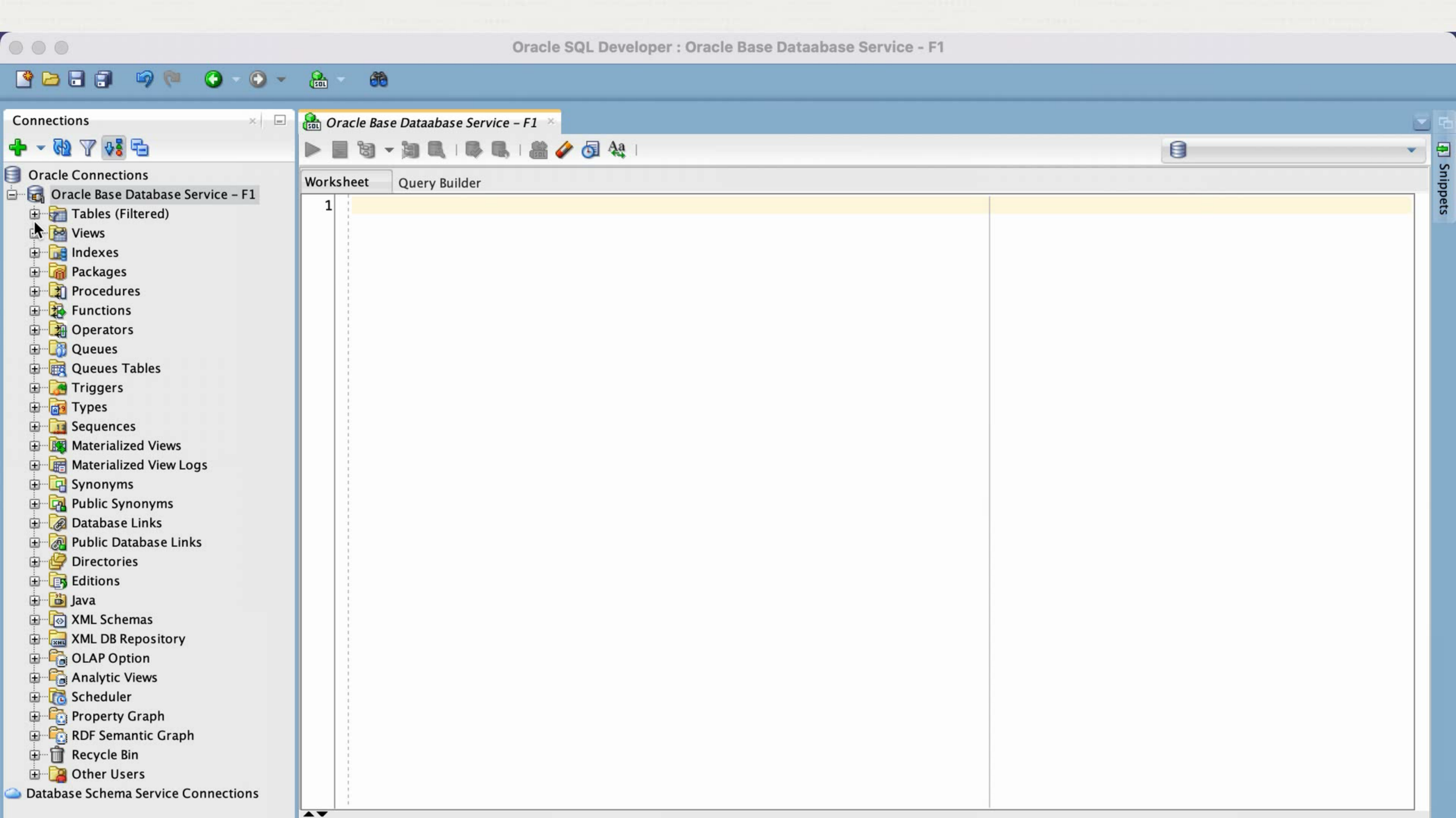
1. 非常适合交互式应用程序，因为在人类思考期间数据不会被锁定
2. 非常适合移动断开连接的应用程序，因为过时文档的写入被拒绝
3. 非常适合文档缓存，因为过时缓存文档的写入被拒绝

一个演示

JSON 关系二元性在架构上
提供了 JSON 的用例简单性



具有关系的多用例灵活性

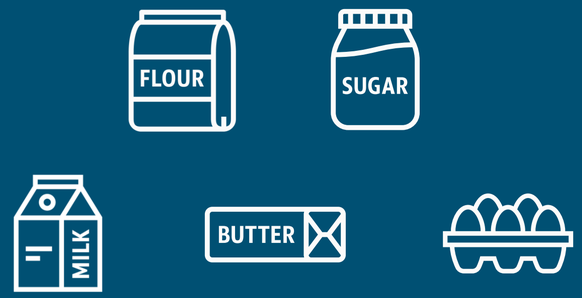

Oracle SQL Developer : Oracle Base Dataabase Service - F1
Connections
Oracle Connections
Oracle Base Database Service – F1
Tables (Filtered)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Materialized Views
Materialized View Logs
Synonyms
Public Synonyms
Database Links
Public Database Links
Directories
Editions
Java
XML Schemas
XML DB Repository
OLAP Option
Analytic Views
Scheduler
Property Graph
RDF Semantic Graph
Recycle Bin
Other Users
Database Schema Service Connections
Oracle Base Dataabase Service – F1
Worksheet
Query Builder
1
Snippets

关键要点

就像提供蛋糕粉的简单性

单一用例

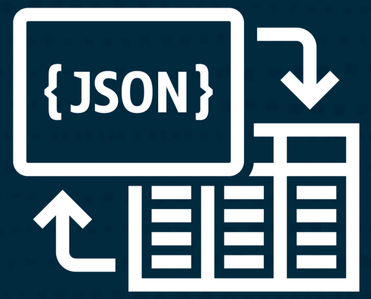
多用例



具有基本成分的多用例功能

关键要点

JSON 关系二元性在架构上
提供了 JSON 的用例简单性

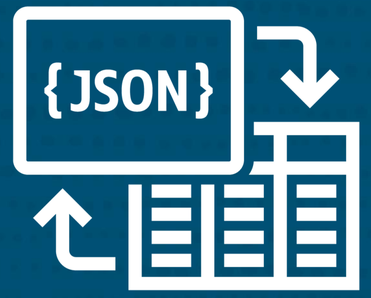


具有关系的多用例灵活性

关键要点

JSON Duality 可以轻松地将应用程序对象作为 JSON 进行持久化和操作，从而消除对象关系不匹配

JSON Duality Views 使数据能够以文档或表的形式透明地读取和写入



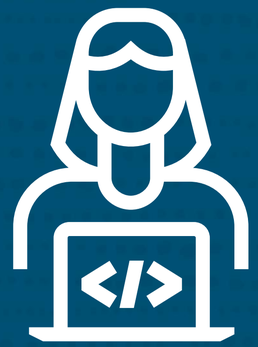
基于值的并发控制确保文档和表之间的一致性

- 启用无状态 API、交互式用例和移动断开客户端

文档开发人员的关键要点

JSON Duality 提供 JSON 的开发简单性以及关系的用例可扩展性

文档模式可以针对每个用例进行



开发人员可以轻松地在现有关系数据上构建以文档为中心的应用程序，并使用 Oracle 融合数据库的所有高级特性

关系开发人员的关键要点

JSON Duality比 ORM 更高效、更集中、更一致

允许在一次调用和一次往返中轻松访问应用程序用例所需的所有行

开发人员可以轻松创建以 SQL 为中心的应用程序或对面向文档的应用程序创建的数据进行分析

Wikibon高级分析师Marc Staimer说：

“Oracle 数据库 23c 结束了长期存在的‘关系与文档’的争论，JSON 关系二元性提供了两全其美的效果"

ORACLE 甲骨文

Oracle数据库闪回技术（Flashback）

实战演练工作坊系列(四)

王旭

- 高级解决方案工程师
- 专注数据库技术领域十余载

内容简介

通过工作坊动手实践demo深入体验数据库闪回技术的诸多特性，了解如何使用数据库闪回机制修复或追溯数据库的数据逻辑错误。

实验内容包括：闪回查询被误删数据、闪回表和恢复误删数据、闪回数据库等。

直播时间：1月6日 11:00 - 12:00

扫描二维码注册并安装手机Zoom进入直播

Zoom ID: 976 6962 5763 密码: 98039717

数据库和云讲座群

20-19

甲骨文云技术公众号

技术专家1V1深入交流