# 人工智能赋能关系型数据库优化技术:现状与展望

乔少杰[1] 李 洲[1] 韩 楠[2] 徐泉清[3] 吴 涛[4] 袁 冠[5] 吴信东[6]

[1](成都信息工程大学软件工程学院 成都 610225)

[2](成都信息工程大学管理学院 成都 610225)

[3](蚂蚁集团 OceanBase 杭州 310013)

[4](重庆邮电大学网络空间安全与信息法学院 重庆 400065)

[5](中国矿业大学计算机与信息工程学院 江苏 徐州 221116)

[6](合肥工业大学计算机与信息学院 合肥 230009)

**摘 要** 传统的关系型数据库优化技术(如连接顺序选择、节点调整、成本估算、索引和视图选择)已经无法满足大数据时代各种业务的高性能需求,尤其是云上的需求。由于人工智能技术拥有学习能力,所以在数据库领域展现出了巨大潜力以及研究前景。本文首先介绍了人工智能应用于关系型数据库的主流方向;其次,探讨基于学习的数据库优化过程中可能遇到的挑战。进而,综述关系型数据库优化的现状及具体技术,并对数据库优化技术的发展进行了展望。重点综述配置优化与查询优化技术:(1)针对数据库配置优化,主要综述索引推荐、视图推荐以及节点调整。索引推荐包括静态推荐和动态推荐。静态推荐依赖于 DBA(Database Administrator)从查询日志中选取常见的查询作为代表性工作负载,并基于工作负载选择合适的索引;动态推荐可以使用贪心算法或者动态规划(Dynamic Programming,DP),根据工作负载的变化动态更新索引方案,也可以配合 DBA 的反馈进行动态调整,基于学习的动态推荐方法可以自动从历史数据中学习,而不依赖于 DBA 的反馈。视图推荐主要有两个任务,候选视图生成和视图选择,候选视图生成通过分析历史工作负载或重写子查询生成高质量候选视图;视图选择在资源限制下优化子集物化,最小化查询成本。节点调整包括基于搜索的方法、传统机器学习法以及强化学习法。基于搜索的方法可以得到一个较好的节点组合,但可能无法在有限的时间内找到最优的节点值;传统机器学习法可以自动优化节点,但需要优质样本;强化学习可以与环境的持续交互来提高泛化能力,仅需要少量样本进行自动调参;(2)针对查询优化,主要综述基数/代价估计以及连接顺序选择。基数/代价估计分为传统方法与基于学习的方法。传统方法包括直方图、数据画像以及索引采样,现有方法很难支持涉及多表/多列的连接查询,且需要额外空间存放样本;基于学习的方法可以更好地获取表与表、列与列之间的高维关系,并且可以适当地与采样方法结合,达到更好的效果。连接顺序选择包括传统方法、静态学习法以及动态学习法。传统方法通过穷举法、贪心算法或者动态规划来选择一个较好的顺序,但是开销大,无法在短时间选择一个最佳计划;静态学习法可以从历史的查询中学习,以提高未来查询的性能;动态学习法侧重于使用自适应查询处理来学习连接顺序,即使在执行查询时也可以更改连接顺序。

**关键词** 数据库;人工智能;深度学习;强化学习;查询优化

**中图法分类号** TP311 **DOI 号** 10.11897/SP.J.1016.2025.01639

# Artificial Intelligence Enabled Relational Database Optimization Techniques: The State of the Art and Prospects

QIAO Shao-Jie[1] LI Zhou[1] HAN Nan[2] XU Quan-Qing[3] WU Tao[4] YUAN Guan[5] WU Xin-Dong[6]

[1](School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225)

[2](School of Management, Chengdu University of Information Technology, Chengdu 610225)

收稿日期:2024-07-25;在线发布日期:2025-03-27。本课题得到国家自然科学基金(62272066)、四川省科技计划(2025ZNSFSC0044,2025YFHZ0194)、教育部人文社会科学研究规划基金(22YJAZH088)、成都市技术创新研发项目(重点项目)(2024-YF08-00029-GX)、成都市区域科技创新合作项目(2025-YF11-00050-HZ)、成都市技术创新研发项目(2024-YF05-01217-SN)、CCF-蚂蚁科研基金项目(CCF-AFSG RF20240106)、网络空间安全教育部重点实验室及河南省网络空间态势感知重点实验室开放基金课题(KLCS20240106)资助。

**乔少杰**,博士,教授,博士生导师,中国计算机学会(CCF)杰出会员,主要研究领域为 AI for DB、查询优化、移动对象数据库。E-mail:sjqiao@cuit.edu.cn。**李 洲**,硕士研究生,主要研究方向为 AI for DB。**韩 楠**(通信作者),博士,副教授,主要研究领域为数据库、数据挖掘。E-mail:hannan@cuit.edu.cn。**徐泉清**,博士,正高级工程师,中国计算机学会(CCF)杰出会员,主要研究领域为数据库系统、分布式系统。**吴 涛**,博士,教授,博士生导师,主要研究领域为智能安全和隐私保护。**袁 冠**,博士,教授,博士生导师,主要研究领域为人工智能和大数据技术。**吴信东**,博士,教授,博士生导师,主要研究领域为数据挖掘、大数据分析和知识工程。

[3](OceanBase, Ant Group, Hangzhou 310013)
[4](School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065)
[5](School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116)
[6](School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009)

**Abstract** Traditional relational database optimization techniques (such as join order selection, node adjustment, cost estimation, index and view selection) can no longer meet the high-performance demands of various businesses in the era of big data, especially those in the cloud. Due to the artificial intelligence technologies have the capability of learning, it has shown great potentiality and research prospects in the database field. This survey firstly outlines the advanced research directions in which artificial intelligence has be applied to relational databases. Secondly, the challenges that may be faced in the phase of learning-based database optimization are discussed. Then, the research status and specific technology of relational database optimization are reviewed, and the future development of database optimization technology is presented. This survey focuses on configuration optimization and query optimization: (1) for database configuration optimization, it mainly summarizes index recommendation, View recommendation and node adjustment. Index recommendation includes static and the dynamic methods. Static methods rely on the DBA (Database Administrator) to select the frequent queries from the query logs as representative workloads, and use the workloads to select indexes; dynamic methods can use greedy methods or dynamic planning, dynamically update the index scheme according to changes in workload, and can also dynamically adjust based on DBA's feedback. The learning-based dynamic recommendation method can automatically learn experience from historical data instead of DBA's feedback. View recommendation primarily involves two tasks: candidate view generation and view selection. Candidate view generation produces high-quality candidate views by analyzing historical workloads or rewriting subqueries; view selection optimizes the materialization of a subset of views under resource constraints to minimize query costs. Node adjustment includes search-based methods, traditional machine learning methods, and reinforcement learning methods. Search-based methods can get a better knob combination, but it may not be able to find the optimal node value in a limited time; traditional machine learning methods can automatically optimize knobs, but high-quality samples are needed; reinforcement learning can continuously interact with the environment to improve the generalization ability, only a small number of samples are needed for automatic parameter adjustment. (2) For query optimization, it mainly summarizes cardinality/cost estimation and join order selection. First, cardinality/cost estimation is divided into traditional methods and learning-based methods. Traditional methods include histograms, data sketch, and index sampling. The existing methods are difficult to support join queries involving multiple tables/multiple columns, and require additional space to store samples. Learning-based methods can better obtain high-dimensional relationship between tables and between columns, and can be appropriately combined with sampling methods to achieve better results. Second, the selection of connection sequence includes traditional methods, static learning methods and dynamic learning methods. The traditional methods select a better order through enumerate method, greedy method or dynamic programming, but it is expensive and cannot choose an optimal plan in a short period of time; static learning method can learn from historical queries to improve future query performance; dynamic learning method focuses on using adaptive query processing to learn the connection order, even when the query is executed, the connection order can be changed.

**Keywords** database; artificial intelligence; deep learning; reinforcement learning; query optimization

# 1　引　言

## 1.1　研究背景

在过去几十年中，人工智能(Artificial Intelligence，AI)和数据库(DataBase，DB)一直都是关注的热点，并且有许多研究者就这两个相结合的方向上进行了深入地研究。一方面，数据库系统提供了简单的查询范式便于用户理解和使用；另一方面，人工智能技术也在近几年内有了爆炸式的进步，这让人工智能技术突破了固有的三大瓶颈：大数据处理能力、新算法和高计算能力。此外，人工智能也使得数据库变得更加智能化(AI for DB，AI4DB)[1]。在大数据时代，传统的关系型数据库技术已经难以有效地处理不断增长的数据量和应对数据多样化的问题，并且，硬件环境的差异以及用户使用水平的参差不齐也是巨大挑战。然而，基于学习的技术在解决上述问题上有很大的发展潜力。

机器学习可以对过去的数据和行为进行分析和学习，从而找到一个更好的设计方案，人类烦琐的手工计算可以被机器所代替[2]。机器学习技术日渐成熟，将应用于各个领域过去的数据作为训练数据，那么机器可以从这些数据中学到知识。对于数据库优化来说，训练数据就是表中的各种记录样本、查询语句、执行计划以及数据库中的参数等。传统的机器学习模型，例如线性回归[3]、随机森林[4]、支持向量机[5]以及集成学习[6]等，这些模型可以学习历史数据中的知识，进而完成相对简单的分类任务。但是，数据库是一个复杂的系统，数据库优化涉及的数据具有多样性和复杂性，比如在进行基数估计时，模型要同时学习表中的样本分布以及查询语句信息或执行计划信息。此时，如果再使用传统的机器学习模型，将无法达到好的效果。例如，使用简单的线性回归模型时，无法解决诸如连接顺序选择这样的复杂问题，因为它无法处理高维参数和连续空间的问题[7]。

随着人工智能技术三大瓶颈的突破，在数据库优化上使用深度学习和强化学习成为现实。深度学习主要是模拟人类的神经元输送信息，从而进行学习，本质上是调整神经元之间的连接，通过以损失函数达到最小值为目标进行梯度下降，目的是拟合复杂的高维映射关系[8]。传统方法通常难以解决NP-hard问题，而深度学习能够提供解决方案，但是这通常需要大量的训练数据来训练模型。强化学习通过与环境的连续交互来获得奖励信号，并据此指导行为。这种方法通过“试错”的方式进行学习，逐渐改进策略以最大化长期奖励，它不关心过程中某一次的奖励大小，只关心最终的奖励是否最大。最后，深度融合深度学习的特征提取能力和强化学习的决策制定过程，深度强化学习模型得以在复杂多变的工业应用环境中展现出更高的适应性和效能[9-11]。

## 1.2　研究问题

传统的数据库设计不断发展，逐渐能够提供可靠的数据存储和管理解决方案，但其高度依赖人工干预，可能导致操作复杂且易出错[12-13]。人工智能技术被用来减少这些限制，探索在人类经验外更多的设计空间，并取代启发式算法解决难题。本文将使用人工智能优化数据库的现有技术分类为配置优化、查询优化、设计优化、异常优化以及安全优化五个方面，如图1所示。

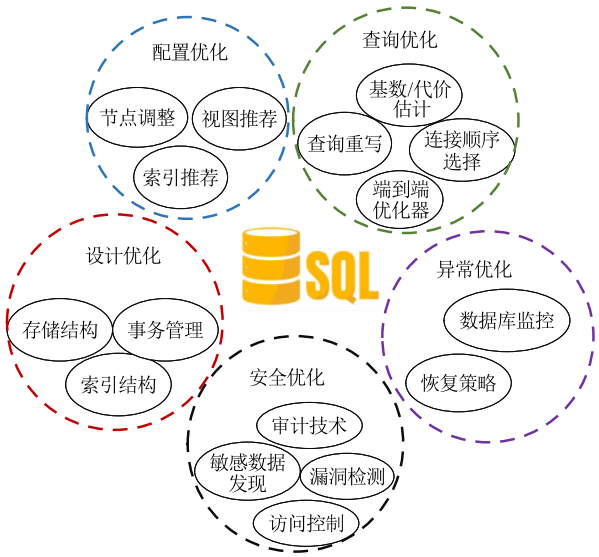


图1　AI for DB优化技术的分类与应用框架

### 1.2.1　基于学习的数据库查询优化

对于数据库查询的优化，最重要的两个组件是优化器与执行器。图2展示了优化器生成执行计划的过程。

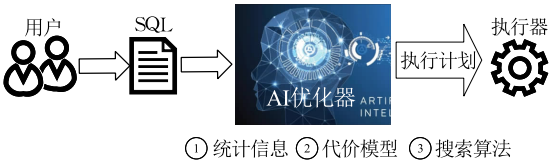


图2　基于AI的查询优化器执行计划生成流程

这方面的研究工作主要包括基于学习的基数/代价估计、基于学习的连接顺序选择、基于学习的查

询重写和基于学习的端到端的优化器四个方面，下面对这四个方面进行概述。

(1)基于学习的基数/代价估计：数据库优化器依赖基数估计来选择最佳执行计划，但传统技术无法有效捕捉不同列或表之间的关联性，导致代价估计的质量较低。目前，各企业所用的数据库的基数估计器在单表等简单业务下精确度较高，但是这种简单的业务场景所占的比例越来越小。对于复杂的查询业务，基数估计的不准确性可能会高达几个数量级。对于代价模型，它参考基数估计的结果来进行代价估计，基数估计的质量越好，代价估计的质量也就越好。但是，代价估计也会受其他因素的影响，如硬件开销、缓存情况等因素给出操作代价。最近，基于深度学习的技术[10,14]被提出，通过使用深度神经网络捕捉相关性估计成本和基数，从而可以获得更好的结果。

(2)基于学习的连接顺序选择：在复杂业务场景下，一个 SQL 查询可能有数百万甚至数十亿个备选的计划，如何高效地找到一个好的计划变得至关重要。传统的数据库优化器使用静态计划选择，使用静态计划选择的缺点很明显，它无法为几十个甚至几百个表找到好的计划，因为探索巨大的查询计划空间的代价是相当大的，这是一个 NP-hard 问题。因此，一些基于迁移学习的方法[15]利用迁移学习将知识从一个数据丰富域转移到另一个数据稀疏域以来解决数据不足的问题。

(3)基于学习的查询重写：SQL 重写是一种通过改写 SQL 查询语句结构的技术，来辅助数据库优化器识别并选择更有效的执行计划。某些程序员仅关心 SQL 查询的结果是否正确，但无法充分考虑 SQL 的质量，因此需要重写 SQL 查询来提高性能。例如，嵌套查询将重写为多表连接查询，进而启用 SQL 优化。现有的方法采用基于规则的策略，使用预定义的规则重写 SQL 查询。然而，基于规则的方法依赖于规则本身的好坏，不可能靠人工穷举所有规则。如果重写器能够自动化的对 SQL 结构进行深入的分析和优化，将极大地提升数据库的性能和效率。

(4)基于学习的端到端优化器：一个优秀的优化器不仅能够解决代价/需要进行基数估算和连接顺序选择，并考虑索引和视图的使用，设计端到端优化器是很重要的。基于学习的优化器[16-17]使用深度神经网络优化 SQL 查询。

### 1.2.2 基于学习的数据库配置优化

数据库配置优化的目标是利用机器学习技术自动化配置。首先，基于学习的数据库可以自动适应不同粒度的查询，合理分配系统资源，并通过历史数据学习工作负载特性，配置自适应参数。其次，它能够分析使用模式和性能数据，智能调整配置参数[12]。传统关系型数据库包含大量参数，手动调整耗时且难以找到最优配置。通过人工智能，可自动优化数据库配置，接下来简要介绍基于学习的索引推荐、视图推荐和节点调整。

(1)基于学习的索引推荐：为了提升数据的查询性能，可以为基本表创建索引。索引相当于字典中的目录，在处理事务型任务时，在列上建立索引可以显著提升处理效率[18]。然而，传统数据库缺乏对数据分布的分析及利用，也没有使用深度学习等人工智能技术去自动解决这些 NP-hard 问题。此外，DBA 要提前预判数据的增长会给索引带来什么样的影响。在一个数据库中，可以在一个表上建立多种索引，让这些索引能够配合使用。但是，人类只能靠经验去判断不同的索引组合在一起的效果，如何能通过机器来获得好的索引推荐是亟待解决的问题。

(2)基于学习的视图推荐：数据库包含三种模式，分别为：① 基于查询频率的模式，该模式首先需要统计和分析数据库中的查询频率，找出重复率高的子查询，针对高频子查询来创建物化视图；② 基于业务需求的模式，该模式根据不同的业务需求和查询模式来评估关键的数据和查询，进而优化物化视图；③ 基于性能和资源使用的模式，DBA 会评估数据库的性能瓶颈，分析查询操作消耗的资源，并以此为基础创建物化视图来达到缓解性能瓶颈的目的。基于学习的视图推荐，可以通过神经网络识别等价子查询，并建立物化视图候选集，然后根据现实情况的一些约束特征来进行视图推荐。

(3)基于学习的节点调整策略：图 3 展示了数据库节点网络。如今，大型互联网公司的数据库包含数百个节点，DBA 需要不断调整节点以适应不同场景。然而，DBA 难以管理云数据库上数百万个实例。不同节点需协同分担同一 SQL 的任务，协调至关重要。对于大多数关系型数据库和非关系型数据库[19]，DBA 依赖经验通过配置文件调整节点，但这一方式有局限，无法同时考虑网络和计算开销等多重因素。系统性能下降的主要原因是数据分布不

均，导致负载不平衡。为此，智能数据划分策略尤为重要，它可自动根据硬件状态进行优化。近期，数据库社区应用基于学习的技术自动调整节点，从而探索更多节点组合并获得优于 DBA 的效果。

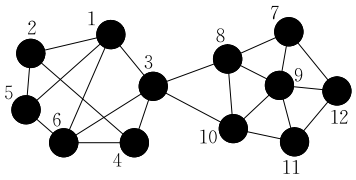


图 3　数据库节点网络架构

### 1.2.3　基于学习的数据库设计优化

传统数据库依赖于数据库架构师的个人经验来完成数据库设计，但由于操作人员经验的限制，仅能探索有限的设计空间。最近，比较流行的方法是在数据库设计中引入了基于学习的方法，通过自动化来提升数据库管理的效率，包括基于学习的索引结构、基于学习的数据存储结构以及基于学习的事务管理，依次进行概述。

(1) 基于学习的索引结构：索引对于数据量巨大的查询业务极其重要，在进行查询之前，索引对基本表中一列或多列数据进行预处理。众所周知，传统关系型数据库中常用的索引都是通用树型结构，因为通常树型结构比线性结构遍历更快。

以图 4 为例，创建 B 树(Balance Tree)索引[20]是对数据排序后，按顺序建立基于磁盘的 B 树，以提升访问效率。创建聚集索引后，每次查询该表时，系统首先检查索引再决定查询策略。索引以数据页形式存储，组织为 B 树结构，每个数据页通过指针连接。B 树从底向上建立，叶节点存储实际数据，中间层节点生成索引页，逐层合成根节点。传统索引方法如 B 树在数量、速度、价值方面能满足需求，但在大数据和人工智能时代，用户体验更为重要，传统索引难以有效应对未知用户行为和大数据。随着数据量增长，索引大小激增，导致磁盘空间膨胀。利用机器学习预测排序数组中特定键位置的模型，是一种高效替代传统索引结构的方法。

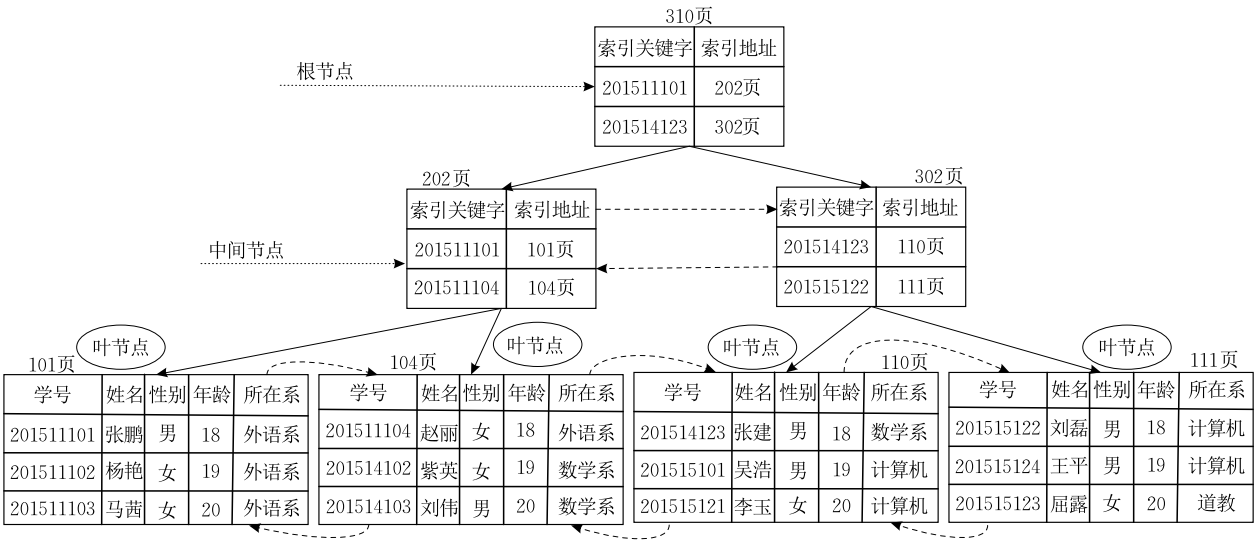


图 4　传统聚集索引 B 树结构与查询优化

(2) 基于学习的数据存储结构：对于关系型数据，存储模型有两种，分别为行式存储和列式存储，其中行式存储的代表数据库是 MySQL、PostgreSQL 等，列式数据库的代表为 HBase；对于非关系型数据，存储模型还包括键值对、图和时间序列等，非关系型数据库的代表是 Redis 和 MongoDB。

如图 5 所示，在复杂的互联网场景下，各类业务会生成多种类型的数据，这些数据往往用高级数据结构表示，统一的关系存储无法满足需求。不同数据类型需要针对具体问题选择合适的存储模型，以加快存储与处理。工作负载类型和数据特性决定存储模型的选择，常见工作负载类型有三类：① 联机事务处理(OLTP)[21]，主要处理日常事务性工作负载，适合使用行式存储来提高效率；② 联机分析处理(OLAP)[22]，用于数据仓库中的多维度数据分析；③ 混合事务/分析处理(HTAP)[23]，将 OLTP 和 OLAP 功能结合，支持同时处理事务性和分析性工作负载。

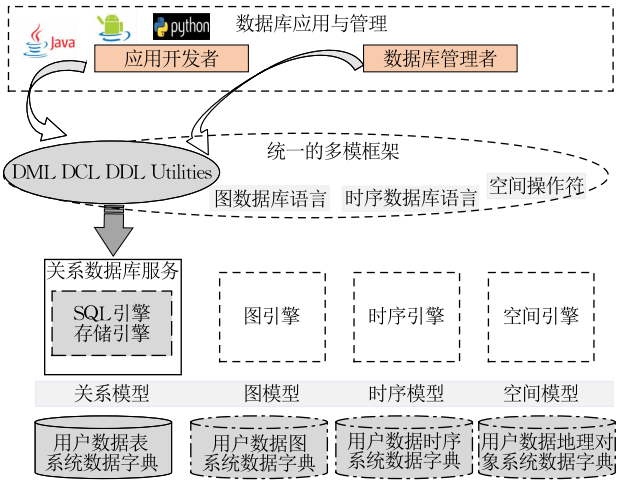


图 5 多模数据库系统架构

(3)基于学习的事务管理:传统的事务管理技术主要集中在事务协议上,如乐观并发控制(OCC)、悲观并发控制(PCC)、多版本并发控制(MVCC)和两阶段提交(2PC)等,近年来,数据库团队试图利用人工智能技术来预测事务和调度事务[24-25],学习现有的数据模式,有效地预测未来的工作负载趋势,并通过平衡冲突率和并发性来有效地调度事务。

### 1.2.4 基于学习的数据库的监控与恢复

数据库自监控可以捕获数据库运行时状态指标,例如读/写延迟、CPU/内存使用率,因此可以在发生异常时提醒管理员(例如性能下降和数据库攻击)。然而,传统的监控方法依赖于数据库管理员对大多数数据库活动进行监控并报告问题,这是不完整和低效的。因此,提出了基于机器学习的技术来优化数据库监控[26-27],这些技术决定了何时以及如何监控哪些数据库性能指标,包括基于学习的数据库监控和基于学习的恢复策略。

(1)基于学习的数据库监控:基于学习的数据库可以自动监控数据库的多方面信息,其中包括:数据库的读写操作,分析数据块的访问模式;并发事务的执行情况,并识别和处理事务死锁、冲突等问题;分析事务执行时间和资源消耗;监控数据的一致性以及数据库的运行状况。如何减少自监控带来的开销是一个热点问题,这里可以采用增量式方法更新监控信息来最小化开销。此外,基于学习的数据库监控方式可以与传统数据监控方法相结合,总结故障转移条件并拟定解决方案。通过学习处理经验,这种监控方式可以帮助人们更好地理解和监督数据库。然而,如果未能充分采集数据库的状态信息,可能会影响数据库的性能。因此,如何在不影响数据库性能的前提下高效监控这些信息,成为当前研究的难点。

(2)基于学习的恢复策略:当数据库遇到未知错误时,能够回退到上一健康状态。首先,在软件工程的整个周期中,数据库随时都有可能发生问题,如锁表、删表、慢查询、非法访问、恶意删库等。需要选择恰当的恢复和备份策略来解决问题并最小化损失。故而数据库的日志系统十分关键,它是数据持久化的保证。基于事务 ID 的多版本管理及历史版本地累积及清除方式,行存储引擎主要以 Redo 日志[28]作为主要持久化手段,配以增量的检查点和日志的并行回放,如图 6 所示,支持数据库实例的快速故障恢复。

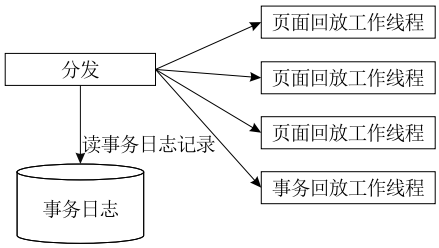


图 6 多线程并行方式回放日志

### 1.2.5 基于学习的数据库安全优化

传统的数据库安全技术,如数据屏蔽和审计,依赖于用户定义的规则,但不能自动检测未知的安全漏洞。因此提出了基于人工智能的算法来发现敏感数据,进行访问控制[29]、数据审计和漏洞检测[30-32]。数据库的安全机制体系如图 7 所示,下面依次从基于学习的敏感数据发现、基于学习的访问控制、基于学习的审计技术以及基于学习的漏洞检测技术进行概述。

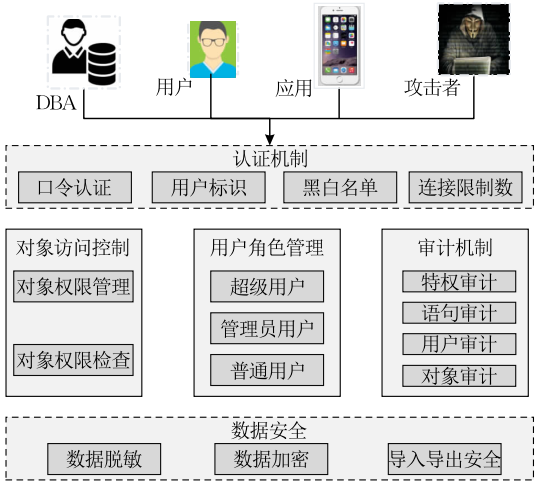


图 7 数据库安全机制体系

(1)基于学习的敏感数据发现：为了保护隐私数据，如身份证号等敏感数据，数据库系统自动识别并学习这些数据特点，然后应用特定策略进行处理。处理敏感数据的方法通常有两种：插入和替换。插入方法涉及在真实数据前添加一些冗余信息，冗余信息可以被接收方分析，以读取出真实数据；而替换方法不涉及增加额外的数据，而是通过改变真实数据的结构，比如打乱字符顺序，来隐藏原始信息。为了恢复数据，接收方必须依据提前设定的规则来解读这些修改后的数据。目前，市面上存在许多数据发现工具，这些工具能够根据发送方设定的数据处理规则来解析出隐藏的数据。可以发现，基于规则的数据隐藏需要统一发送方和接收方的规则，而且不能穷尽所有可能的规则。基于学习的隐藏技术通过深度学习模型对输入数据进行复杂的非线性变化，使得原始数据在高维空间中有效地隐藏和转换。

(2)基于学习的访问控制：传统数据库系统依赖网络和操作系统安全，通常构建在封闭网络环境中。然而，随着云计算的发展，越来越多企业租用云服务器，系统环境变得复杂，安全风险增加。数据库系统通过认证模块限制访问，采用密码、证书认证等机制确保安全，同时通过黑白名单限制 IP 访问。认证详细流程如图 8 所示，用户登录后通过基于角色的访问控制机制获得相应权限。每次访问数据库时，都需通过访问控制验证权限。传统数据库依赖 DBA 手动配置访问策略，基于学习的访问控制则可以自动学习和决策，节省了大量人力。

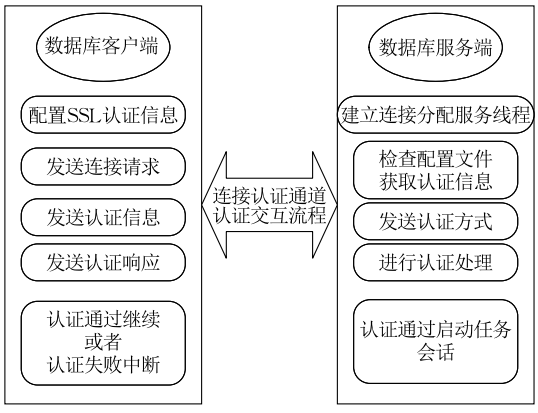


图 8　数据库认证详细流程

(3)基于学习的审计技术：首先，数据预处理是优化审计流程的重要步骤，传统审计中，数据来源于多个异构系统，格式不统一，质量参差不齐。其次，动态分析是提升审计效率和精度的关键。传统审计依靠审计师的经验和静态的规则分析，难以面对复杂的业务环境。而基于学习的审计将任务转移到机器学习模型中，使用机器学习的学习特性来提取信息。采取人机结合的方式，可以节省人力成本，也可以使审计员作出更适当的决定。在基于学习的审计系统中，可以利用机器学习算法来分析用户的行为模式，从而增强数据安全性。

(4)基于学习的漏洞检测：系统漏洞的自动发现是网络安全的关键组成部分。安全扫描工具可以识别已知漏洞，并通过执行各种测试任务来发现未知漏洞。目前，检测未知安全漏洞的方法包括反汇编扫描、源代码分析和环境错误注入分析，但大多依赖于系统已知的安全缺陷。Noller 等人[33]使用深度学习算法，通过分析数据库相关的大量源代码，有效识别国家漏洞数据库中已知的漏洞。此外，还可以对潜在的漏洞进行评级，提高了国家数据库的安全性。

### 1.3　AI4DB 面临的挑战

人工智能技术为数据库领域带来了创新动力。尽管如此，人工智能模型对训练数据以及计算资源和时间的要求较高，这些因素导致人工智能与数据库集成的过程中遭遇诸多挑战。下面将针对数据质量不佳、训练周期长、模型泛化能力差以及人工智能与数据库系统兼容性问题进行概述，并给出针对上述问题的解决方法。

#### 1.3.1　数据较差

人工智能算法对数据的要求有三个方面：首先是数据本身的质量，其次是数据的数量，最后是数据的类别是否丰富。

(1)数据的质量：数据质量对模型训练效果至关重要。为了更好地训练预测模型，确保数据的准确性和唯一性是必要的。但是，无法完全保证数据库中数据的质量。首先，数据库中的信息多且杂，不一定所有信息都是有用的，一些过时的信息和错误的信息可能会误导模型。其次，数据之间往往是有联系的，还要考虑数据之间的分布问题。

(2)数据的数量：高质量数据仅仅是训练优秀模型的一个必要条件，数据的数量是另一个必要条件。以神经网络为例，如果数据量少，那么神经网络探索的解空间就少，很有可能发生欠拟合的现象。

(3)数据的类别：在训练集中，如果数据的类别很少，那么模型不能很好地预测学习过的样本类别，对于那些不在训练集中的数据束手无策。数据的类别影响着模型的泛化性与适应性，然而在实际生产

中,可能由于时间紧迫,无法获得足够类别的数据,只能让模型先学习已有的类别,然后在实际场景中,面对各种情况再逐步优化配置。

针对数据质量较差的问题,Jiménez-Gaona等人[34]提出一种基于生成对抗网络(Generative Adversarial Network,GAN)的数据增强方法,通过生成与真实数据相似的合成数据,扩充训练数据量,同时改善数据的多样性,在一定程度上缓解了因数据量不足和类别单一导致的模型训练问题。Peng等人[35]提出了一种基于深度强化学习的无监督综合数据清洗框架 RLclean,能够有效清洗多种类型的数据错误,无需人工干预,并自适应不同领域,提高数据质量。Gu 等人[23]提出了一个基于深度强化学习的联邦查询优化框架 Coral,用于解决数据库查询中的连接顺序问题,尤其是在数据质量参差不齐的情况下。上述研究表明,解决数据质量问题对于推动 AI 赋能的数据库优化技术发展至关重要。

1.3.2 超长的训练时间

如果现在已经有了大量的可靠训练数据,还需要足够的时间让模型去学习高维空间的知识。因此,当模型很复杂时,往往需要更长的训练时间才能学到有价值的知识,只有那些相对简单,学习速度快,学习能力又好的模型才容易被接受。训练复杂的模型需要很长的时间,现实数据库需要兼顾用户需求、上线压力等问题,很难等到模型收敛。所以,在选择模型时,有两个关键指标需要考虑。首先,模型是否适合特定的业务场景。例如,在调整参数的问题上,一些参数的取值范围内可以取的数值是无限的,如果使用传统的基于表的算法,如 Q-learning[36],则需要从一张庞大的表格中搜索所有可能的参数组合,这会耗费大量的资源。其次,是否可以利用其他高效的人工智能算法诸如确定梯度下降(DDPG)[37]、迁移学习(Transfer Learning)[38]、强化学习中 Actor-Critic 算法[39]等来解决训练时间过长的问题。

例如,Li 等人[40]提出了一种面向大规模云数据库的资源优化系统 Eigen,可以同时最大化资源分配比例和资源可用性,并能够提高用户体验和减少调度延迟。同时,Cai 等人[41]提出了一种改进的多任务二层进化算法(IM-BLEA),用于解决数据密集型科学工作流在云上的执行问题,将工作流调度问题划分为数据放置和任务调度两个子问题,并建立双层最优模型,以优化数据集放置和任务调度并减少训练执行的时间开销。在 Wang 等人[38]探讨了迁移学习在数据库优化中的应用,尤其是解决复杂模型训练时间过长的问题后,提出了一种迁移学习算法,将预训练的模型应用于新的数据库环境中,以减少训练时间。上述研究充分展示了解决训练时间过长问题的重要性。

1.3.3 泛化性弱

训练数据和测试数据是否遵循同一高维空间中的规律决定了人工智能算法的适应性和泛化性。当前数据库系统可能被搭建在 Linux 或者 Windows 等操作系统上,硬件环境不同,用户量不同,用户需求具有多样性,这些因素导致模型的泛化性较弱。

(1)硬件环境不同:硬件环境的差异性主要表现在两个方面:首先,事务处理所涉及的硬件环境正在不断变化,随着新型计算和存储介质如固态硬盘(SSD)、磁盘阵列、非易失性存储器(NVM)以及多核 CPU 架构(如 AMP、SMP)的出现,数据库系统的性能和效率有了显著提升。其次,不同的预算和应用场景要求数据库适应不同的硬件环境。机器学习的性能受到硬件环境的直接影响。因此,机器学习模型在训练时必须考虑硬件环境以及查询的特性,并在迁移到新的硬件环境时,通过增量训练来调整和修正之前学习到的知识,以确保模型的准确性和效率。

(2)用户负载不同:用户负载不同包含两方面。首先,要针对不同的负载特性制定专门的运维策略和结构设计,才能更高效的存储和管理数据。为了进一步提升数据库系统的智能化和自适应能力,引入机器学习模型来辅助运维。其次,混合业务类型增加了数据库管理的复杂性,因为不同的负载类型需要不同的服务支持,数据库实例不能仅针对单一类型的负载提供服务。在当今复杂的业务环境中,除了要具备处理海量的数据存储需求的能力,还要具备灵活的数据管理能力。

(3)用户需求不同:不同类型的用户对数据库的性能有着不同的需求,而批处理业务则更注重数据库的并行处理能力和吞吐量。DBA 在面对不同类型用户的需求时,常常需要在多个性能指标之间进行权衡。在许多情况下,DBA 可以通过降低其他方面性能来提升某一方面的性能。比如,通过增加数据库的并行连接数来提高资源的利用率。然而,这种做法可能会降低某些单个任务的处理效率,因为分配给每个连接的工作区减少了。

为应对硬件环境不同、用户需求不同、用户负载

不同产生的模型泛化性弱的问题，Xiong 等人[42]提出了一种基于强化学习的自适应查询计划优化器 AutoQuo，能够有效生成高质量查询执行计划，尤其针对复杂查询和多样化工作负载表现出良好的泛化能力。Chen 等人[43]提出了一种基于深度强化学习的查询优化方法 GLO，旨在解决现有方法在处理未见过的工作负载时的泛化问题，实现了跨不同查询工作负载的初步泛化能力。Chen 等人[44]提出了一个基于图结构的增量学习方法，应用于复杂的数据库连接顺序选择问题，该方法利用了在线学习和增量学习，特别适用于模型泛化能力差的场景。随着新查询的出现，模型能够不断更新，根据新的环境特征进行优化，极大地提升了模型在不同硬件环境和用户负载下的泛化能力。

1.3.4　AI 匹配数据库问题

数据库优化技术面临许多复杂的挑战，包括环境配置、查询优化和调度机制等关键领域。具体来说，在调参过程中，数据库包含大量参数，DBA 需要依靠经验进行调整，但许多参数的可取值范围极为广泛，人工尝试所有可能的组合几乎是不可能的。此外，参数之间存在复杂的多维关系，确定最佳参数配置成为一个 NP-hard 问题。在优化器方面，从指数级的状态空间中选择最优执行计划也是一个 NP-hard 问题。对于索引技术，传统的位图索引和基于树的索引由于未使用神经网络而受到限制，无法从高维空间的角度处理问题，因此难以有效检测未知用户行为和数据[18]，也未能将用户习惯与索引选择相结合。尽管基于学习的解决方案已有所探索，但要将此类模型成功应用于实际生产环境，仍需克服诸多挑战。在解决 AI 与数据库匹配问题上，Yue 等人[45]使用多任务学习框架，提出了一种基于多任务学习的数据库调优框架，通过感知数据库各功能模块状态，实现功能感知的数据库旋钮调优，并通过学习任务间的关联性来增强调优效果，提高了整体性能及各功能模块指标。Jo 等人[46]提出了 ThalamusDB 系统，一种针对多模态数据执行复杂 SQL 查询的近似查询处理系统，支持整合自然语言谓词的 SQL 查询，通过零样本模型和关系处理相结合的方式回答查询，利用确定性近似查询处理来降低机器学习推断的计算需求。利用多模态学习技术，融合多种数据模态信息，使模型能够更好地适应复杂多变的应用场景。

结合现有工作的优缺点，文献[44]能够有效解决数据质量差、训练时间长、模型泛化性差等问题，基于深度学习和强化学习的数据库查询优化框架如图 9 所示。

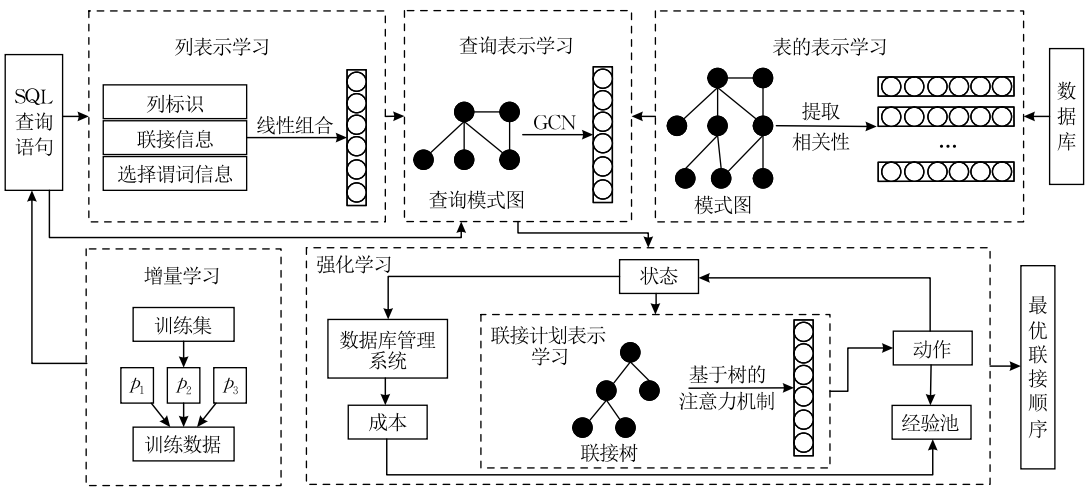


图 9　基于深度强化学习的数据库查询优化框架

图 9 给出了一个基于深度学习和强化学习的数据库查询优化框架。首先，SQL 查询语句经过解析，提取列标识、联接信息和选择谓词等信息，通过线性组合形成列表示学习。同时，表的表示学习模块通过模式图表示表的结构，提取表与列之间的相关性。然后，在列表示学习和表表示学习的基础上，通过 SQL 查询构建一个包含所有表、列及其关系的查询模式图。查询表示学习模块使用图卷积网络对 SQL 查询生成的查询模式图进行学习，以捕捉查询的复杂性。再者，强化学习模块通过状态描述当前查询及其表示，利用数据库管理系统计算查询成本，并结合基于树的注意力机制优化联接计划表示学习。强化学习根据状态选择动作（查询执行顺序），并将结果存入经验池，用于后续优化。最终，框架输

出优化后的联接顺序,从而提高数据库查询的执行效率。此外,借助增量学习技术动态更新和学习历史查询数据,帮助模型持续改进其性能,有助于应对数据库查询多样性和复杂性。

在此基础上,未来的研究方向可以从几个方面进行探索:(1)多模态数据优化:随着多模态数据在数据库中的应用越来越广泛,可以探索如何通过深度学习模型来优化多模态数据的查询和处理效率;(2)自适应查询优化:数据库系统可以进一步引入自适应查询优化技术,通过实时监控查询执行过程中的性能变化,动态调整查询计划;(3)跨平台数据库优化:随着云计算和边缘计算的普及,未来的数据库系统需要在不同的硬件平台和操作系统上高效运行。可以探索如何通过跨平台的机器学习模型来优化数据库性能,确保在不同环境下的泛化能力;(4)自动化数据库管理:数据库系统可以进一步引入自动化管理技术,通过 AI 模型自动调整数据库配置、索引和查询计划,减少 DBA 的手动干预;(5)隐私保护与安全性:随着数据隐私和安全问题的日益突出,数据库系统需要更加注重隐私保护和安全性。可以探索如何通过联邦学习和差分隐私技术来保护用户数据,同时确保数据库的高效查询和优化。通过以上前瞻性研究方向的探索,未来的数据库系统将能够更好地应对大数据时代的挑战,提供更加智能和高效的数据库优化解决方案。

### 1.4 本文贡献

本文主要贡献包括:(1)帮助开发者和研究人员更好地理解当前基于机器学习的数据库优化技术的发展现状,并分析了在实际应用中可能面临的诸多挑战如:模型泛化能力弱、数据库与 AI 算法适配问题等;(2)总结概括了 AI 在数据库优化中的应用,涵盖了查询优化、设计优化、配置优化、监控与恢复以及安全优化五个方面,并针对当前最新问题如动态变化的查询负载和索引方案难以优化、预定义规则的 SQL 重写方法灵活性差,难以适应多变的查询模式等问题,结合人工智能技术给出最新的解决方法如使用强化学习模型自动适应变化的工作负载、动态调整索引方案、使用深度学习和强化学习技术动态生成 SQL 重写规则等;(3)结合实际应用场景,深入探讨数据库与机器学习结合的研究前景如深度学习和强化学习在数据库优化中的应用潜力,并展望了未来的发展趋势。

## 2 基于学习的数据库优化

人工智能技术可以从多个方面对数据库进行优化,包括基于学习的查询优化、基于学习的数据库配置、基于学习的数据库设计、基于学习的数据库监控与保护及基于学习的安全性。本节从如下方面介绍典型工作。

### 2.1 基于学习的查询优化

基于学习的查询优化旨在利用机器学习技术解决查询优化中的难点问题,如基数/代价估计、连接顺序选择、SQL 重写和端到端优化等。

#### 2.1.1 基数/代价估计

基数估计已经被广泛研究了多年,是数据库中最具挑战性的问题之一,它通常被称为现代查询优化器的“致命弱点”[47]。传统的基数估计方法可分为三类:数据画像法[48]、直方图法[49-50]和采样法[51-52]。如表 1 所示,传统方法包含以下优缺点。

**表 1 传统基数/代价估计技术对比**

| 方法 | 优点 | 缺点 | 使用场景 | 表现 | 适应能力 |
|---|---|---|---|---|---|
| 直方图统计 | 简单,广泛使用 | 在估计误差和时空代价上找平衡点 | 被广泛使用 | 中 | 高 |
| 数据画像 | 估计不同数据个数 | 需要额外的位图空间 | 估计单列 | 中 | 低 |
| 索引采样法 | 面向连接 | 需要额外空间存放样本,0-tuple | 内存数据库 | 中 | 中 |

直方图统计方法简单,但是必须在误差与时空代价上综合考虑。数据画像需要额外的位图空间。基于采样的方法可以通过索引来捕获多列和多表之间的关系,但是不能处理复杂查询。基于索引的采样法依赖于索引,如果索引质量不好或者没有建立索引,那么该方法效果就很差了。代价估计的作用是预测物理执行计划的资源使用情况,包括 I/O 和 CPU 使用情况。传统的代价估计是基于基数的代价模型来估计实际的操作。

近年来,数据库研究者提出利用深度学习技术来估计基数和代价。基于学习的基数估计分为有监督方法、无监督方法以及图方法。根据所采用的模型,可以进一步将监督方法分为如表 2 的基于学习的基数/代价估计。

**表 2　基于学习的基数/代价估计技术分类与比较**

| 类别 | 模型 | 核心技术 | 估计 | 编码 | 多列 | 多表 |
|---|---|---|---|---|---|---|
| 监督学习 | 混合模型 | 混合模型 | 选择性/基数 | 谓词 | √ | × |
| | 全连接网络(NN) | NN,XGBoost | 选择性 | 谓词 | √ | × |
| | | LR,PR,NN | 基数 | 任务名,运算符 | √ | √ |
| | | NN | 基数 | 表,谓词 | √ | √ |
| | 卷积神经网络(CNN) | Tree-CNN | 查询延迟 | 查询，部分计划 | √ | √ |
| | | Multi-set CNN | 基数 | 查询 | √ | √ |
| | 循环神经网络(RNN) | Tree-LSTM,Tree-GRU | 基数/代价 | 完整计划 | √ | √ |
| | | Plan-Structure RNN | 代价 | 元数据,操作符 | √ | √ |
| | | RNN | 基数 | 表,谓词 | √ | √ |
| 无监督学习 | 内核密集估计(KDE) | KDE | 选择性 | 数据样本 | √ | × |
| | 深度概率模型 | AutoRegression,DNN | 选择性 | 数据,谓词 | √ | × |
| 图学习 | 知识图谱 | GNN | 基数 | 查询 | √ | √ |

(1) 混合模型：Praciano 等人[53]提出了一种基于查询的混合模型方法。该方法引入了新型基数估计方法，称为鲁棒性基数，旨在提高数据库管理系统(DBMS)中的查询性能。与传统的直方图启发式方法不同，该方法利用机器学习技术，以提高查询操作的基数估计精度。通过这种方式，DBMS 的执行引擎能够有效地避免较差的执行方案进而选择最优查询执行路径。Zhao 等人[54]提出了一种关注不确定性的基数估计模型，该方法不是直接使用复杂的深度学习模型去构建基数估计器，而是使用了贝叶斯深度学习(Bayesian Deep Learning，BDL)作为贝叶斯决策和深度学习之间的桥梁。BDL 的预测分布为测量不确定性的标准提供了原则上的参考。

(2) 全连接神经网络：Ortiz 等人[55]利用全连接神经网络建立基数估计模型，将查询编码为特征向量，通过实验来权衡模型大小和估计误差。Yu 等人[56]提出了一种基于采样的树状长短期记忆神经网络来建模查询，该神经网络使用谓词之间的操作符类型将子表达式构造为树，并通过捕获表之间的连接交叉相关性和谓词之间的顺序依赖关系来提高基数估计的性能和准确性。Chen 等人[17]提出了一种结合操作符级深度神经网络的基数估计方法，该方法利用两个操作符级深度神经网络来处理和连接操作，并使用一个输出网络将中间表示映射到基数估计，相较于其他方法能够有效处理复杂查询计划，并且显著提高基数估计的精度。

(3) 卷积神经网络(Convolutional Neural Network，CNN)：Qiao 等人[1]提出了一种垂直扫描卷积神经网络(Vertical Scanning Convolutional Neural Network，VSCNN)来捕获词向量中词之间的关系，从而生成特征映射，该方法利用基于学习的基数估计器将结构化查询语言从一个句子转化为一个词向量并将表名和样本编码为位图，使得 VSCNN 可以获得 SQL 查询的表、连接和谓词等多样化与语义信息，相较于其他模型的基数估计，该模型显著提高了基数估计的准确性。Marcus 等人[16]提出了一个名为 Neo 的端到端查询优化器。作为强化学习模型的一个重要组成部分，他们提出了一种同时包含查询编码和部分计划编码的神经网络来评估当前查询的最佳学习效果。对于计划编码，它使用 CNN 逐层聚合连接。对于谓词编码时，它利用 word2vec 训练行向量，并利用选定行的平均值去编码每一个谓词。但是，行向量表示对于在线编码是非常耗时的。Yang 等人[57]提出了一种名为 NeuroCard 的模型，该模型使用了残差 CNN[58]中的残差块，重复利用每一列的向量。NeuroCard 可以学习数据库中所有表之间的相关性，而不做任何独立的假设。NeuroCard 从数据中学习，就像经典的数据驱动估计器一样，在概率模型中捕获所有可能的相关性。

(4) 循环神经网络(Recurrent Neural Networks，RNN)：Ortiz 等人[55]提出一种基于 RNN 的基数估计模型计划。在每次迭代中，一个节点将被添加到计划树中，节点序列是 RNN 模型的输入。基于树型的 RNN 估计器如图 10 所示。Wang 等人[59]提出了一个种基于学习的基数估计器，该基数估计器利用归一化流模型来学习关系数据的连续联合分布。可以将连续随机变量上的复杂分布转换为简单分布，并使用概率密度来估计顺序查询和并行查询的基数。Wang 等人[60]提出一种基于学习的渐进式基数估计器，该渐进式基数估计器由初始模型和细化模型组成，初始模型在执行查询之前估计基数，细化模型使用已执行操作符的实际基数逐步改进基数估计并重新优化初始模型。Qiao 等人[61]对树型 LSTM 模型进行改进，使用 GRU 替换 LSTM，加快了训练的

速度,同时使用了基于学习的方式来学习字符串的模式。也有学者使用无监督学习来进行基数/代价估计,工作[62-64]使用无监督密度模型拟合数据集隐藏的分布,但它们很难支持多连接等复杂查询。Heimel等人[62]提出了一种基于核密度估计器(Kernel Density Estimation,KDE)的选择性估计器,这种估计器可以根据数据库的变化进行轻量级的构造和维护,并且通过选择最优的带宽参数对核密度估计器模型进行数值优化,以获得更好的估计质量。Hasan等人[63]和Yang等人[64]利用自回归密度模型来表示列之间的联合数据分布。这类模型返回链式规则中存在的条件密度列表。为了支持范围谓词,该模型采用渐进抽样的方法,并利用学习的密度模型进行有意义的样本选择,甚至适用于倾斜的数据。

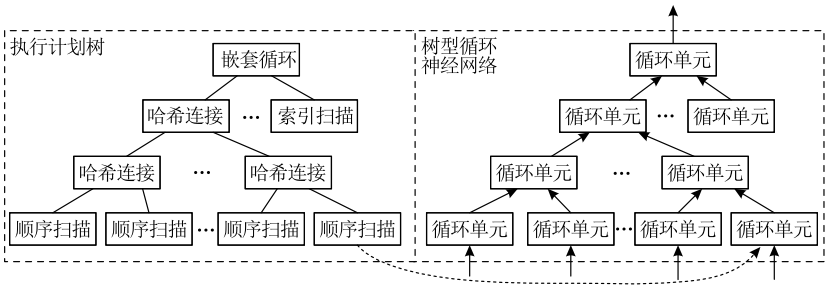


图 10 树型 RNN 估计器示例

最近,有学者采用基于图的学习方法来进行基数估计。Chen等人[65]研究了图数据库中的基数估计器,并总结了两类估计器:(1)作出一致性和条件独立性假设的乐观估计器;(2)使用信息论线性规划的悲观估计器。在基数估计图中选择自下而上的路径来构建基数估计器,其中包含子查询作为节点和子查询之间表示平均度的加权边。Davitkova等人[66]提出了一种基于知识图谱的框架LMKG(Learned Models for Cardinality Estimation)。它采用深度学习方法有效地估计RDF图上查询的基数。同时将监督和无监督方法运用到子图模式,并产生更准确的基数估计。

上述方法虽然有了很大的改进,但它们仅支持简单的/固定的查询,而具有DISTINCT关键字的查询在查询优化器中可能使用不同的技术。因此,Hayek等人[67]使用了两种方法来处理常规查询。首先,他们使用深度学习方案来预测查询结果中的唯一率$R$。对于重复行,其中查询表示为属性、表、连接和谓词的集合。然后,他们用重复结果乘以唯一率进而扩展了现有的基数方法。

### 2.1.2 基于学习的连接顺序选择

连接顺序选择在数据库系统中起着非常重要的作用,已经研究了很多年[68]。传统的方法通常基于基数估计和代价模型,通过一些剪枝技术来搜索所有可能连接顺序的解空间。通常将连接顺序选择的方法分为三类:(1)传统算法;(2)静态学习的方法;(3)动态学习的方法。图11给出了连接顺序选择方法的优缺点。

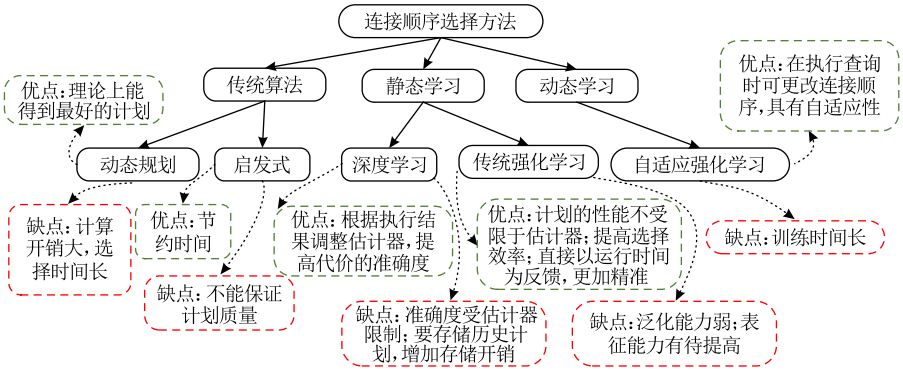


图 11 数据库连接顺序选择方法总结

(1)传统方法包括动态规划算法和启发式算法。传统数据库查询优化方法根据专家经验设定确定的估计器,其中基数估计用于查询结果集的大小,CPU的代价估计用于预估执行查询需要的时间和资源,通过估计器,数据库系统才能对查询执行性能进行预估。这一策略已被PostgreSQL[69]采用。这

种策略基于预先设定的估计器，用于评估每个计划的执行成本。

通过比较这些成本，选出最优执行计划，从而提高查询的性能和效率。基于动态规划的算法[70]通常选择最佳方案，但计算开销很大。另外，DP 算法生成的计划可能由于错误的代价估计而产生较大的代价。

启发式方法 GEQO[71]、QuickPick-1000[72]和 GOO[73]可以更快地生成计划，但可能无法生成好的计划。QuickPick-1000 是一种简单高效的随机搜索算法，它通过随机选择来生成搜索路径，这种随机性使得算法能够探索不同的状态空间，从而找到全局最优解。然而，随机性也带来一定的不确定性，可能会在搜索过程中陷入局部最优解导致无法找到全局最优解。一些研究者结合启发式信息帮助算法在搜索过程中关注可能包含的最优解状态。基因优化器[71]就是一个改进的启发式方法，它也被应用于 PostgreSQL 中。基于贪心做法的搜索算法，如 GOO[73]，根据贪心的思路，设定一个局部评价标准，不断选取目前衡量下最优的计划。

对上述传统方法出现的问题，近年来提出了一些基于机器学习的方法来提高连接顺序选择的性能，这类方法利用机器学习算法来学习已有的连接操作，解决了传统方法的因为基数/代价估计不准确引起的偏差。基于学习的方法可以在较短的时间内有效地选择更好的方案。通常将现有的学习方法分为静态学习法和动态学习法。

(2) 静态学习法：现有研究[47,74-76]从历史的查询中学习，以提高未来查询的性能。DQ 算法[74]和 ReJoin 算法[75]提出利用神经网络和强化学习来优化连接顺序。DQ 算法和 ReJoin 算法均利用历史执行计划的代价作为训练数据对神经网络进行训练，从而对每个连接顺序进行评估。DQ 算法使用一个 one-hot 向量 $\boldsymbol{V}$ 来表示每个连接状态。向量中的每个单元格表示在连接树中每一张表和操作选择，然后以 $\boldsymbol{V}$ 为输入向量，传入到多层感知器(Multilayer Perceptron，MLP)来评估每个连接状态，并用 DQN(Deep Q-Network)来指导连接顺序的选择。一旦生成了一个计划，该计划的代价作为一个反馈来训练神经网络。与 DQ 不同，ReJoin 的输入由深度向量和查询向量组合而成。深度向量表示每张表在连接树中深度信息，查询向量表示查询信息。ReJoin 使用最近策略优化[76]来引导连接顺序选择。结果表明，与 PostgreSQL 的优化器相比，DQ 和 ReJoin 均能以较低的代价生成良好的连接顺序，同时保持了较高的效率。然而，ReJoin 和 DQ 中的神经网络结构简单，不能很好地表示连接树的结构，不能学习查询计划的延迟。此外，上述两种方法不能支持数据库模式更新。为了解决这一问题，RTOS(Reinforcement learning with Tree-LSTM for join Order Selection)[77]把训练分成了两个阶段，通过一个设计良好的神经网络结构在有延迟的情况下生成更好的连接顺序。为了解决 ReJoin 和 DQ 中的图神经网络不能捕捉到连接树结构的问题，RTOS 利用 Tree-LSTM 表示连接状态的模型。RTOS 首先设计了列、表和连接树的向量表示。在得到向量表示后，用深度 Q 网络来引导连接顺序的生成。其次，RTOS 首先利用代价对图神经网络进行预训练，然后利用时延信息对神经网络进行在线训练。结果表明，该方法可以在较低的延迟下生成较好的连接顺序。算法 1 展示了 RTOS 根据连接树动态建立神经网络的过程，*JoinTreeDFS*(*Node*)函数为递归函数，表示通过深度优先去构建神经网络。其为每一个连接树构建一个树型模型，叶子节点可能是表或列。连接树中的一个内部节点对应一个连接，该内部节点由 4 个不同节点($\alpha_0$，$\beta_0$，$\beta_1$，$\alpha_1$)组成，$\alpha_0$ 和 $\alpha_1$ 是两个需要被连接的表节点，$\beta_0$ 和 $\beta_1$ 是该连接中对应的列。$\alpha_0$，$\beta_0$，$\beta_1$，$\alpha_1$ 是位置敏感的，其选用了名为 *N-ary* Tree-LSTM[77]的模型。

**算法 1.**　根据连接树动态建立神经网络

定义函数：*JoinTreeDFS*(*Node*).

输入：节点(包括表节点、列节点以及 *N-ary* 单元[67])

输出：连接树的状态表示

1. IF *Node* is a leaf
2. $h=R(Node)$;
3. $\boldsymbol{m}=Zeros\_init()$;
4. RETURN $h$, $\boldsymbol{m}$;
5. ELSE
6. $h_{\alpha_0}, \boldsymbol{m}_{\alpha_0}=JoinTreeDFS(Node\rightarrow\alpha_0)$;
7. $h_{\alpha_1}, \boldsymbol{m}_{\alpha_1}=JoinTreeDFS(Node\rightarrow\alpha_1)$;
8. $h_{\beta_0}, \boldsymbol{m}_{\beta_0}=JoinTreeDFS(Node\rightarrow\beta_0)$;
9. $h_{\beta_1}, \boldsymbol{m}_{\beta_1}=JoinTreeDFS(Node\rightarrow\beta_1)$;
10. RETURN $N\text{-}aryUnit(h_{\alpha_0}, \boldsymbol{m}_{\alpha_0}, h_{\alpha_1}, \boldsymbol{m}_{\alpha_1}, h_{\beta_0}, h_{\beta_1})$;

对于一个树节点 $j$，使用 $h_j$ 表示该节点的嵌入，$\boldsymbol{m}_j$ 表示记忆单元向量。该算法包含 3 个步骤：

(1) 如果节点 $j$ 是表示单表的叶子节点，则 $h_j=R(j)$，然后 $\boldsymbol{m}_j$ 将被初始化为零向量；

(2) 如果节点 $j$ 是代表单列的叶子节点，则 $h_j=R(j)$，然后 $\boldsymbol{m}_j$ 将被初始化为零向量；

(3) 对于表示连接的节点 $j$，包含 4 个子节点

$(\alpha_{j,0},\beta_{j,0},\beta_{j,1},\alpha_{j,1})$。在 *N-ary* Tree-LSTM 中应用 *N-ary* 单元对这 4 个节点表示,得到 $h_j$ 和 $\boldsymbol{m}_j$。

(3)动态学习法:这类方法[78-79]侧重于使用自适应查询处理来学习连接顺序,即使在执行查询时也可以更改连接顺序。Avnur 等人[78]提出了一种称为 Eddy 的自适应查询处理机制,结合了查询的执行和优化,在执行查询的时候学习并生成连接顺序查询。Eddy 将查询过程拆分为多个操作符,例如,三个关系之间含有两个连接操作符。Eddy 使用两种路由方法来控制这些操作符去处理即将到来的元组的顺序,即 Naive Eddy 和 Lottery Eddy 策略。Naive Eddy 可以以较少的代价路由更多的元组到操作符;同时,Lottery Eddy 能以较小的选择性将更多的元组到操作符。然而,这两种路由方法都是针对特定场景而设计的,需要设计通用的路由算法来处理更复杂的场景。基于强化学习的 Eddy 使用 Q-Learning[80]来解决这一问题。将 $Q$ 函数定义为所有操作符代价之和。通过最小化 $Q$ 函数,指导每次选择哪个操作符。但是,上述优化器不会分析预期执行时间和最佳结果之间的关系,也不会丢弃中间结果。针对上述问题,Trummer 等人[79]提出了名为 Skinner-DB 的强化学习模型。Skinner-DB 使用树的置信上限 UCT(Upper Confidence Bounds Applied to Trees)[81],而不是 Q-Learning,因为 UCT 可以为所有选择的累积结果提供“反悔”功能。Skinner-DB 将查询执行划分为多个时间片,在每个时间片中选择一个连接顺序执行,利用时间片中连接顺序的实际性能,训练 UCT 来指导更好的连接顺序选择。最后,Skinner-DB 合并每个时间片中产生的元组以生成最终结果。

### 2.1.3 基于学习的查询重写

许多数据库用户,尤其是云用户,可能无法编写高质量的 SQL 查询,SQL 重写的目标是将 SQL 查询转换为等价的形式,现有的 SQL 重写方法大多采用基于规则的技术[82-84],在给定一组查询重写规则的情况下,找到可应用于查询的规则,并对其进行修改使用规则重写查询。然而,对重写操作的各种组合进行评估的代价很高,传统的方法往往无法进行优化,而且重写规则与应用程序高度相关,因此,机器学习可以从两个方面来优化 SQL 重写:(1)规则选择:由于有许多重写规则,因此可以使用强化学习模型来做出重写决策。在每个步骤中,代理都会估计不同重写方法的执行成本,并选择代价最低的方法。该模型迭代生成重写的结果,并根据结果更新决策策略。而 Zhou 等人[85]创新性地提出了一种策略树框架,该框架中每个节点代表一个可能的查询重写,并采用 Monte Carlo Tree Search(MCTS)高效地遍历找到最优的重写序列。通过集成学习模型,该系统估计每个查询与其降低的成本,从而引导搜索。该方法能够避免传统基于规则的局部最优问题,找出更高效的重写顺序,对于不同重写方案成本的估算也可以进一步提高搜索效率;(2)规则生成:根据在不同场景的重写规则集合中,使用 LSTM(Long Short-Term Memory networks)模型来学习查询、编译器、硬件特性和相应规则之间的相关性。然后,对于一个新的场景,LSTM 模型捕获门单元内部的特征并预测适当的重写规则。随着大语言模型(Large Language Models,LLMs)的广泛应用,Liu 等人[86]利用 LLMs 迭代修正的特性,提出一种名为 GenRewrite 的系统,该系统可以利用自然语言的形式指导 LLMs 进行查询重写,可以在人类可理解的内容中总结重写内容。此外,引入迭代修正技术,逐步修正重写中的语法和语义错误。与传统基于规则的查询重写方法相比,GenRewrite 表现出更高的灵活性和效率,尤其是在处理复杂查询时。Zhou 等人[87]提出了 DB-GPT 系统,该系统引入自动生成输入提示的方法,包括选择合适的任务说明和演示,并优化生成的提示以提高 LLMs 的性能。同时通过训练数据库相关数据(如 SQL 查询、执行计划等)来学习数据库知识。

可以预见的是未来数据库管理系统中,大型语言模型可以通过语义分析更加准确地理解用户查询的意图,特别是在处理自然语言到 SQL 查询的转换任务时。现有的文本生成能力让大型语言模型能够有效地处理复杂的查询请求,对于非结构化或半结构化查询场景尤其有帮助。

尽管大型语言模型有巨大潜力,但在应用于数据库管理系统时仍面临一些挑战。首先,大型语言模型当前在处理与数据库底层架构相关的物理特性(如数据分布、存储结构等)时尚存在局限。现有的模型主要依赖逻辑语义,难以全面覆盖数据库的物理层优化需求。另一个关键挑战是数据隐私和安全性问题。由于数据库往往包含敏感信息,如何在保护用户隐私的前提下有效利用大型语言模型进行训练和推理仍需要进一步探索。

LLMs 能够更好地理解查询计划中的图结构和物理特性。在研究如何将物理层信息嵌入到大型语言模型的优化流程中,可以生成更高效的查询计划。

此外，可以利用联邦学习等隐私保护技术在确保数据库数据安全的前提下，进一步拓展大型语言模型在数据库任务中的应用场景。

2.1.4　端到端的优化器

尽管许多研究者已经尝试用机器学习方法来解决基数/代价估计和连接顺序选择问题，但在执行计划优化中仍有许多因素需要考虑，如索引和视图的选择。连接顺序选择方法[74-76,79]提供逻辑计划，依靠数据库优化器选择物理运算符和索引，并利用代价模型生成最终的执行计划。最近，Marcus 等人[16]提出了一种端到端优化算法 NEO，它不使用任何代价模型和基数估计来生成最终的物理计划。NEO 也是一种基于 ReJoin 的离线学习方法[75]，与 RTOS[76]类似，NEO 也使用 Tree-CNN 来捕捉结构信息，用行向量来表示谓词，用 one-hot 向量来表示神经网络中每个物理操作符和索引的选择，从而生成执行计划。然后，NEO 执行图神经网络引导的搜索，不断以最小值去扩展状态，进而找到最佳计划。此外，NEO 在没有任何代价模型信息的情况下，采用 PostgreSQL 的计划对神经网络进行预训练，并以时延作为反馈来对神经网络进行训练。这种端到端的方法从时延中学习生成整个执行计划，可以应用于任何环境，并且对估计误差具有鲁棒性，整体的查询优化器示意图如图 12 所示。

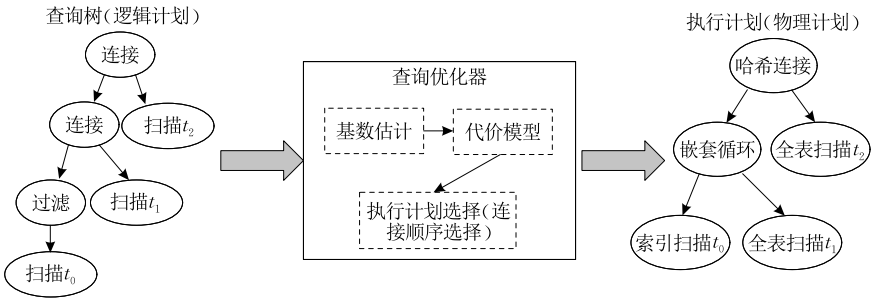


图 12　端到端查询优化器的工作流程

2.1.5　基于云计算的查询器

随着云计算的快速发展，云数据库已成为企业数据管理的主流选择。云数据库的最大特点在于其分布式架构和多节点部署模式，这使得数据存储和计算能够在多个节点上进行，从而提供了更高的可扩展性和可靠性。Kaseb 等人[88]认为相较于集中式数据库，分布式数据库具有更强的可扩展性、可靠性，查询优化技术主要包括：(1) 消除冗余子查询：其主要功能是删除复杂查询中的重复子表达式以减少执行时间；(2) 持续/迭代处理：随着运行时可用数据的增加，利用渐进式查询优化等技术持续监控和调整查询执行计划；(3) 缓存中间结果：其主要功能是缓存和重用先前查询的中间结果，以提高查询性能，尤其是对于重复查询；(4) 具体化：具体化视图存储预先计算的查询结果，加快了未来查询的执行速度，尤其是针对频繁的查询；(5) 流水线：MapReduce Online 等技术在任务之间产生管道中间结果，与传统的 MapReduce 方法相比，减少了存储和网络开销。Wang 等人[89]提出了 ReOptRL 算法，使用强化学习在执行过程中动态调整查询执行计划，避免了与传统的基于成本的方法相关的开销，但依赖于不断更新的数据统计数据。ReOptRL 通过从历史查询执行中学习来提高性能，优化了查询中不同运算符在不同云资源(计算机)上的执行方式。此外，ReOptRL 基于策略梯度来选择最佳操作，通过奖励函数进行评估，同时考虑时间和成本，平衡用户偏好以实现更快的执行速度并获得更低的成本。

## 2.2　基于学习的配置优化

基于学习的数据库配置旨在利用机器学习技术来自动化数据库配置，例如索引推荐、视图推荐和节点调整。

2.2.1　基于学习的索引推荐

在数据库管理系统中，索引是提高查询执行速度的关键，选择合适的索引对提高查询性能至关重要，表 3 展示了索引推荐方法及特征。

**表 3　不同索引推荐方法对比**

| 方法 | 场景 | 质量 | 适应性 |
|---|---|---|---|
| 静态索引推荐 | 静态数据 | 低 | 低 |
| 动态索引推荐 | 动态数据 | 中 | 中 |
| 半自动索引推荐 | 动态数据 | 高 | 中 |
| 基于学习的索引 | 动态数据 | 高 | 高 |

式(1)～(2)展示了索引选择问题的定义[90]，考虑一组表，让 $Col$ 表示这些表中的列集合，$size(col)$ 表示 $col$ 列的索引大小，其中 $col \in Col$，给定一个查询工作负载 $Q$，令 $reward(q,col)$ 表示在 $col$ 列上为查询 $q$ 建立索引的奖励，即在 $col$ 列上有索引和没

有索引的情况下执行查询 $q$ 的奖励。在给定空间预算 $S$ 的情况下，索引选择问题的目标是找到一个列的子集 $Col^*$ 来建立索引，以便在索引总大小保持在 $S$ 以内的情况下使奖励 $R$ 最大化。

$$R=Maximize\sum reward(q\in Q,col\in Col^*) \quad (1)$$

$$\sum size(col\in Col^*)\leq S \quad (2)$$

(1) 静态索引推荐：DBA 需要从查询日志中挑选常见的查询作为代表性的负载，这些负载可以反映数据库系统的日常操作模式，基于这些工作负载，DBA 可以做出决策，选择合适的索引来优化查询性能。Chaudhuri 等人[91]提出了一个名为 AutoAdmin 的索引选择工具，专门为 SQL Server 数据库设计。其核心思想是通过分析查询来选择最优的索引方案，从而优化查询性能，工作流程总结为三个步骤：① 查询分析，对每个查询 $q\in Q$，利用索引选择工具提取需要的索引列，以识别查询中频繁被访问或用于过滤的列；② 索引候选生成，该索引选择工具使用枚举算法生成可能的索引组合，每个组合代表不同索引方案；③ 索引选择，AutoAdmin 在候选集中的所有索引组合中，选择最佳的索引方案。Valentin 等人[92]将索引选择问题建模为背包问题，提出 DB2 Advisor。首先，枚举索引方案及其奖励；然后，将索引选择问题建模为背包问题。更具体地说，它把每个候选索引方案作为一个条目，方案的大小作为条目的权重，方案的奖励作为条目的价值，然后 DB2 使用动态规划来解决这个问题。静态方法的缺点是不够灵活，无法处理工作负载的动态变化。然而，选择一个通用的工作负载将增加 DBA 的负担。针对这些问题，本文将介绍以下几种动态解决方法。

(2) 动态索引推荐：动态索引选择方法很多，可以分为三类。传统的动态索引选择方法、半自动索引选择方法和基于机器学习的索引选择方法。传统的动态索引选择方法持续分析工作负载，并根据工作负载的变化以动态更新索引方案。Yu 等人[93]提出了一种结合规则与强化学习改进索引推荐的方法，该方法设计了三种启发式规则，用于在索引候选者中过滤无效的索引；进而利用强化学习算法根据数据库的动态变化实时调整索引配置，有效提高了该方法在不同数据库规模下的鲁棒性和准确性。Gu 等人[94]提出了使用强化学习模型来决定如何选择插入的子树以及如何在构建 R-Tree 时分隔节点，而不是依赖于启发式规则，该方法使用机器学习模型能够有效提高经典 R-Tree 的查询性能。Qiao 等人[95]提出的 AP-IS 模型则侧重于数据库领域中多模态数据下的索引选择优化问题。该模型通过强化学习算法(APE-X DQN)智能选择适合不同数据模态的索引类型，优化数据库的查询性能。AP-IS 设计了新的索引集编码和 SQL 语句编码方法，使得模型能够识别和理解数据的模态特性，从而选择最适合的数据索引类型，显著减少存储空间占用，同时保证查询效率的提升。此外，AP-IS 利用异步更新和优先经验回放机制，确保在动态工作负载场景下能够快速学习和调整索引配置，以适应不断变化的数据访问模式。实验结果表明，AP-IS 在处理复杂多模态数据集时的性能明显优于传统的索引选择方法，如贪婪算法或基于枚举的方法。Sadri 等人[96]提出了一种基于强化学习的索引选择方法，首先通过提取工作负载特征来反映查询的频率和分布，同时通过列特征来衡量各列在查询中的重要性程度。该方法利用马尔科夫[97]决策过程(Markov Decision Process，MDP)模型，将索引选择视为一个序列决策问题，通过学习查询模式和列特性生成一组创建或删除索引的规则。随着工作负载的变化，MDP 模型能够自适应地调整索引策略，从而动态优化数据库性能。这种方法不仅提升了查询效率，还通过自动化索引管理减轻了人工干预的负担，进一步增强了系统的灵活性和应对复杂查询场景的能力。通过上述研究，可以发现在关系型数据库中，人工智能可以基于查询负载动态优化 MySQL 中的索引结构。例如，可以自动识别哪些列经常被用于查询条件，并为这些列生成多列或联合索引，从而显著提高查询性能。此外，人工智能技术特别是强化学习可以用于自动选择最适合当前查询模式的索引，并动态调整索引配置。

(3) 随着 LLMs 在各种任务中表现出优越性，在索引推荐方面也引入了 LLMs 以提高索引效率。LLMs 可以分析大量历史查询的 SQL 语句和其执行计划，通过学习不同查询的模式和数据库的访问频率来生成最合适的索引。此外，LLMs 能够通过学习不同的查询组合和表间关系，自动推荐哪些列应该被索引。通过捕捉查询的高维特征，LLMs 能够推测哪些列之间存在强关联性，并据此优化索引结构。LLMs 还可以基于实时查询负载变化自动调整索引方案，以应对负载波动和性能需求的变化。例如，Zhou 等人[87]提出了一种基于大型语言模型的数据库框架 DB-GPT，展示了大语言模型在数据库索引调优中的应用，通过生成智能任务指令和示例，DB-GPT 能够自动地生成和优化数据库索引，提

升查询性能。然而，当前 LLMs 在索引调优中并未完全考虑索引的空间消耗，尤其是在存储限制较小的情况下，模型可能会推荐一些不适合的索引。此外，索引调优会对多个查询进行迭代优化，因此需要设计一种机制，允许 LLMs 多次调整索引策略，直到达到最佳性能。为了更好地优化索引选择，未来的研究可能需要改进对查询工作负载和数据特性的建模，特别是在高并发和大规模数据环境中。

2.2.2　基于学习的视图推荐

物化视图是数据库管理系统中的一个重要组成部分，利用视图可以显著提高查询性能。合理地选择物化视图可以在可接受的开销内显著提高查询性能。然而，对于普通用户来说，自动生成物化视图是十分困难的。现有的方法依赖于 DBA 来生成和维护物化视图。不幸的是，即使是 DBA 在处理大型数据库时也面临挑战，尤其是拥有数百万数据库实例并支持数百万用户的云数据库。因此，使用视图推荐器自动为给定的查询工作负载推荐适当的视图是十分必要的。视图推荐器已经研究了很多年，Oracle 数据库和 DB2 数据库等现代关系型数据库提供了支持物化视图的工具[98-100]。视图推荐器有两个主要任务：(1) 候选视图生成。视图推荐器从历史工作负载中发现候选视图；(2) 视图选择。由于系统资源的限制，所有候选视图都不可能具体化，视图推荐器选择候选视图的子集作为具体化视图，这样可以最大程度地提高性能。

(1) 候选视图生成：由于所有可能的视图数量呈指数级增长，因此视图生成的目标是生成一组高质量的候选视图，以便用于将来的查询。主要有两种方法：① 确定经常出现在工作负载中的等效子查询。Dokeroglu 等人[101]提出了一种启发式方法，包括分支定界、遗传、爬山和混合遗传爬山算法，旨在找到一个接近最优的多查询执行计划。将查询分解为子查询，包括选择、投影、连接、排序，和数据传送。为了使这些子查询有更多机会共享公共任务，它们为每个查询生成可选的查询计划。其计划生成器与代价模型交互，该模型通过考虑其子查询的潜在重用能力来估计查询代价。然后，在这些可选计划中检测公共任务，并搜索具有较低代价的全局计划；② 重写子查询以使视图响应更多查询。直接从查询中提取的子查询仍然不具有通用性，因为即使有细微的差别也不能推广到其他查询。为了缓解这一问题，Zilio 等人[98]通过合并相似的视图、更改“select”条件、添加“groupby”子句来概括提取的视图，减少了资源消耗，使更多的查询可以利用视图。

(2) 视图选择：由于系统资源的限制，候选视图不能全部物化。因此，视图推荐器应该选择一个性能良好的视图子集来动态地对其进行物化和维护，视图选择问题的目的是选择一个候选子集作为物化视图。本文将试图选择用形式化的数学符号描述，如式(3)～(4)所示。假设有一个工作负载和一些候选视图，其中 $Q$ 是工作负载中的查询子集合，$V$ 是候选视图集，$V^*$ 是选定要物化的视图子集，$q$ 是查询子集合 $Q$ 中的一个查询，$v$ 是 $V^*$ 中的一个物化视图。$C(q,V^*)$ 表示在一组物化视图 $V^*$ 存在的情况下，查询 $q$ 执行的代价。$M(v)$ 是物化视图 $v$ 的维护代价，$|v|$ 表示视图 $v$ 占用磁盘空间的大小，$r$ 是磁盘空间限制，因此选定视图的总大小不能超过 $r$。视图选择问题的目标是在不超过磁盘空间限制的情况下，选择要物化的视图子集，达到最小的响应工作负载的代价 $Cost$。

$$Cost=\arg\min\Sigma C(q\in Q,V^*\in V)+\Sigma M(v\in V^*) \tag{3}$$

$$\Sigma\,|v\in V^*|\leq r \tag{4}$$

Yuan 等人[102]的研究通过结合 *width-deep* 模型和强化学习技术，将视图选择问题转化为一个动态优化的整数线性规划问题，以寻找最佳视图配置策略。而 Liang 等人[103]提出了基于信用的模型，通过评估视图的未来效用和重建代价来决定在存储限制下的视图排除策略。这两种方法均摒弃了传统的静态规则，转向数据驱动的决策过程，尽管它们在提高决策准确性和适应动态数据环境方面具有潜力，但同时也面临模型可扩展性、信用评估精确度以及在现实应用中适应性等挑战。

2.2.3　基于学习的节点调整

数据库和大数据分析系统包含数百个可调的系统节点[104]，其控制着数据库的内存分配、I/O 控制以及日志记录等方面。传统的方法根据 DBA 的经验手动调整这些节点，但是需要花费大量时间来调整节点，并且无法有效处理云数据库上的数百万个数据库实例。为了解决这一问题，自校正方法被提出来自动调整这些节点，它利用机器学习技术，不仅可以获得更高的性能，而且可以缩短调整时间。现有的自动调参方法大致分为三类：基于搜索的调节方法、基于传统机器学习的调节方法，以及基于强化学习的调节方法。基于搜索的调节方法依靠搜索算法在参数空间中寻找最优配置，而基于传统机器学习的调节方法则利用技术如贝叶斯优化，通过历史数据进行预测和优化。基于强化学习的调节方法，

则通过智能体在系统中进行试探性操作并学习如何在长期内获得最佳性能。如表 4 所示，在固定负载的单机场景下，深度强化学习方法通常表现良好，但在云数据库环境中，随着负载变化，深度强化学习并不能保证总是优于传统机器学习方法，其效果往往依赖于具体的应用场景和系统负载特性。

**表 4 数据库节点调整的方案**

| 方法 | 优点 | 缺点 | 场景 | 表现 | 适应性 |
|---|---|---|---|---|---|
| 专家决策 | 经验丰富，有基本保障 | 耗时耗力，难同步 | 单机，单负载 | 低 | 低 |
| 静态规则 | 基于规则，易于实现 | 只能囊括少量参数 | 单机，单负载 | 低 | 低 |
| 启发式 | 自动调参，速度快 | 资源有限，难以穷举 | 单机，分析型负载 | 低 | 低 |
| 机器学习 | 自动调参，学习经验 | 需要优质样本 | 单机，负载模式固定 | 中 | 高 |
| | 自动调参，学习经验 | 需要更多资源以处理更复杂的负载模式 | 云数据库 | 高 | 中 |
| 深度强化学习 | 自动调参，自动训练 | 依赖高质量反馈信号 | 单机，负载模式固定 | 高 | 高 |
| | 自动调参，自动训练，小样本 | 只负责负载级别调参 | 云数据库 | 高 | 中 |

（1）基于搜索的调整：为了减少人力，Zhu 等人[7]提出了一种递归定界搜索调优方法，该方法在给定查询负载的情况下，从历史数据中找出相似的查询负载，并返回相应的节点值。具体来说，给定 $n$ 个节点，该方法将每个节点的值范围划分为 $k$ 个区间，这些节点区间形成一个有 $k^n$ 个子空间的离散空间。然后，在每次迭代中，从有界空间中随机选择 $k$ 个样本，并从这 $k$ 个样本中选择性能最好的样本，表示为 $S_1$。在下一次迭代中，只从靠近 $S_1$ 的有界空间中获取样本。通过这种方式，可以迭代地减少有界空间，最终得到一个较好的节点组合。但是，该方法可能无法在有限的时间内找到最优的节点值，而且它不能达到高性能，因为其需要搜索整个空间。

（2）基于传统机器学习的调整：为了解决基于搜索的优化方法中存在的问题，传统的机器学习模型可以用来自动优化节点。Van Aken 等人[105]提出了一个基于机器学习的数据库调优系统。使用高斯过程为不同的工作负载推荐合适的节点。首先，选择一些查询模板，其中每个查询模板包含一个查询工作负载及其相应的节点值。其次，提取数据库的读/写的页数、查询缓存的利用率等内部信息来反映工作负载特性。该方法从内部信息特征出发，利用因子分析对不相关的特征进行过滤，然后采用简单的无监督学习方法选择与调整最相关的 $k$ 个特征，并使用这 $k$ 个特性来分析工作负载特性。再者，使用这些选定的特性将当前工作负载映射到最相似的模板。该方法直接推荐该模板的节点配置作为优化配置，并将查询工作量输入到高斯过程模型，学习新的配置来更新模型。

（3）基于强化学习的调整：强化学习通过与环境的持续交互来提高泛化能力。Zhang 等人[106]开发的数据库调优系统利用深度强化学习（Deep Reinforcement Learning，DRL）来优化数据库性能，该系统面临的关键挑战在于重新设计 DRL 模型的五个核心模块。通过将数据库调优抽象为一个强化学习问题，该系统以云数据库实例为交互环境，利用实例的内部指标作为状态信息，调优模型作为智能代理，调整数据库配置参数作为动作，并以性能提升作为奖励信号。代理通过一个称为 Actor 的神经网络来制定调优策略，该网络接收状态指标作为输入，并输出优化的节点值。同时，另一个神经网络 Critic 用于评估 Actor 的决策，结合节点值和内部指标来预测奖励，并据此调整 Actor 的策略。在整个优化过程中，代理根据数据库实例的状态输出调优动作，应用这些动作后，通过执行工作负载并监测性能变化来获得奖励，进而更新 Critic 网络。Critic 的更新进一步指导 Actor 网络，以更好地理解节点值与内部度量之间的复杂关系。该系统的核心在于利用机器学习技术来优化数据库性能，标志着数据库管理从传统的基于规则的静态调整向动态自适应学习的转变。该方法的优势在于能够实时响应数据库工作负载的变化，通过持续学习来优化配置参数，从而提升数据库操作的效率和响应速度。然而，深入分析该技术本身，其仍存在如下缺陷：

首先，深度强化学习模型在训练过程中面临收敛速度慢的问题，特别是在高维度的数据库状态空间中，可能导致在实际应用中模型训练的成本高昂。其次，模型的泛化能力是一个挑战，因为数据库的工作负载模式可能非常多样，模型需要能够在未出现过的场景中也能做出有效的决策。

（4）基于大语言模型的调整：在数据库配置中，LLMs 可以分析数据库中的查询日志、性能数据和硬件资源使用情况，学习如何根据查询负载和硬件特性动态调整数据库参数。LLMs 可以结合强化学习等方法，通过与数据库环境的持续交互优化参数。此外，随着数据库的运行，查询负载和资源使用的变

化会导致最优配置发生变化。LLMs 能够根据实时反馈，自动调整数据库的配置参数。如 Giannakouris 等人[107]提出了一个利用 LLMs 进行自动化数据库系统调优的框架 λ-Tune。该模型通过将调优上下文描述为大型输入文档，使用 LLMs 提取调优提示，并生成多个备选配置，采用基于成本的优化方法确定最佳配置。该框架最小化了重新配置的开销，并确保评估成本随着最佳运行时间的变化而有界限，并有效提升调优的鲁棒性。但 LLMs 的性能通常会受到硬件环境的影响，尤其是在分布式数据库系统中。如何确保 LLMs 在不同硬件环境下的表现一致，是一个需要解决的问题。LLMs 需要大量的多样化数据来进行训练，如果数据库负载和查询模式不具备多样性，模型也可能无法提供准确的调优建议。数据库参数调优需要实时响应工作负载的变化，而 LLMs 的训练和推理过程通常需要较长的时间，因此如何提高 LLMs 的实时性，使其能够在短时间内生成调优策略，也是一项技术挑战。未来可以结合结构化数据和自然语言描述，开发多模态 LLMs 模型，以更好地理解数据库的语义和上下文。并且，开发能够实时学习和适应动态工作负载的 LLMs 模型，自动调整参数调优策略。

### 2.3　基于学习的设计优化

基于学习的设计理念是将机器学习技术融入到数据库组件的设计过程中。通过机器学习技术来提高数据库索引的性能。以下将分别综述基于学习的索引结构、基于学习的数据存储结构以及基于学习的事务管理。

#### 2.3.1　基于学习的索引结构

Kraska 等人[108]提出了“索引就是模型”的概念，这一概念将其中 B+树索引视为是将每个查询键映射到其对应页面的模型。对于排序后的数组，位置 $id$ 越大意味着键值越大，范围索引应该有效地逼近累积分布函数。Ding 等人[109]提出了一种结合机器学习预测能力与传统 B+树存储优势的自适应学习索引结构 ALEX，该索引结构通过引入“间隙数组”，减少插入时的数据移动，并结合自适应扩展与选择性重训练，以适应数据分布的变化，从而在高动态读写工作负载中显著提升插入效率和查询性能。ALEX 在动态环境中表现出色，能够根据数据的变化进行自适应调整，从而在混合读写场景中具备优势。而 Chen 等人[110]则对 ALEX 在非易失性主存上的性能进行研究，着重分析了其在新型存储介质中的表现和优化方法，尽管通过学习数据分布减少了树的高度，提升了查询速度，但在非易失性主存上的插入操作仍需进一步优化以减少写入成本。上述研究介绍了学习索引在动态数据更新和大规模数据管理中的多种解决方案，为数据库系统在不同场景下的应用提供了指导。Li 等人[111]则关注在并发环境下的数据一致性问题，提出一种细粒度的学习索引结构 FINEdex，该结构通过设计更为精细的分层数据结构和独立的模型，降低数据之间的依赖性，减少线程冲突，因此在多线程环境中表现优异。其核心是使用两级排序数组来处理数据插入和修改，这种数据结构不仅保证了插入过程中数据的一致性，还支持低开销的非阻塞重训练机制，使其能够迅速适应动态数据分布的变化，并在高并发场景中保持高效的插入和查询性能。Ma 等人[112]针对超内存数据库的管理，提出了一种完全基于学习的索引方案 FILM，旨在优化内存与磁盘之间的数据管理。通过使用轻量级机器学习模型来近似数据的累积分布函数，FILM 能够有效预测数据的位置，从而减少索引对内存的占用并提升查询效率。FILM 特别关注冷数据管理，通过自适应的 LRU（Least Recently Used，最近最少使用）策略，将不常使用的数据转移到磁盘，降低内存占用，实现了对大规模数据的高效处理，即使在数据量超出内存容量的情况下，依然能保持良好的查询性能。

基于现有对索引结构的研究，展望学习索引在未来数据库系统中的发展，特别是在引入线性回归模型和处理变长字段数据方面的潜力与挑战。学习索引引入线性回归模型的动机在于其简单高效的预测能力，可以近似数据的累积分布函数，从而快速定位数据，正如 ALEX 通过学习数据分布优化查询效率，特别适合动态数据环境。同时，变长字段数据的管理是未来学习索引面临的挑战之一。FILM 针对大规模数据提出了利用轻量级模型进行数据预测的策略，但在面对变长字段时，需要更复杂的模型来捕捉数据间的不规则性，这可能增加模型复杂度。FINEdex 通过细粒度分层结构和非阻塞重训练机制，为处理变长字段和高并发环境提供了参考，保证了索引的高效性和一致性。此外，ALEX 在非易失性主存上的应用展示了学习索引在新型存储介质中的潜力，研究结合 DRAM 优化 NVM 写操作，为新型存储环境中的学习索引设计提供了方向。未来的学习索引可以通过结合线性回归与深度学习模型，优化变长字段处理，并在 NVM 等新型存储技术中发挥优势，同时提升对多核并发环境的支持和自适

应调优能力,从而为数据库系统的查询效率和扩展性带来更高效的解决方案。上述研究为学习索引在实际应用中提供了理论基础,并为下一代数据库系统的优化指明了发展方向。

2.3.2 基于学习的数据存储结构

Idreos 等人[113-114]指出,键值存储系统的数据结构可以通过组合基本组件来构建,并且可以使用基于学习的代价模型来指导这一构建过程。这一模型中,设计空间的概念涵盖了所有可能的数据结构和存储策略,上述内容都是由基本组件构成的。基本组件包括:时间分区,用于分隔和管理基于时间的数据;链接,用于连接和访问不同的数据节点;栅栏指针,用于快速定位和跳过不需要的数据块。设计连续体包含了将多个设计理念和策略连接起来的实际实现路径。在设计数据结构时,首先要识别出总代价的主要瓶颈,可能包括过多的 CPU 计算、内存使用不足或网络延迟等问题。确定可以调整哪些节点以减轻这一瓶颈,通过向某个方向调整节点,直到达到该方向的边界或总代价降至最低。这个过程类似于梯度下降,可以自动执行。

布隆过滤器是一种判断某个值是否属于某个集合的索引。然而,传统的布隆过滤器可能会因为位数组和哈希函数而消耗大量的内存。为了减小布隆过滤器的尺寸,Kraska 等人[108]提出了一种创新的布隆过滤器变体,即基于学习的布隆过滤器。训练一个二分类器模型,用于识别数据集中是否包含特定的查询项。当一个新的查询到达时,它首先被送入这个分类器。如果分类器判断查询可能存在,则进一步处理;如果分类器判断查询不存在(即负例),则需要通过传统的布隆过滤器进行验证,以确保没有错误的否定(即伪负例)。Mitzenmacher 等人[115]提出了一种形式化的数学方法来指导如何改进学习型过滤器的性能。他们提出了一种包含三层的结构,第一层是传统的布隆过滤器,目的是去除大部分不在数据集中的查询;第二层是神经网络,目的是去除伪正例;最后一层是另一个传统的布隆过滤器,目的是保证没有伪负例。其提供了数学和直观的分析,以证明该结构优于双层布隆过滤器。此外,设计了 Bloomier 过滤器,不仅可以确定存在的键,还可以返回该键与数据集中相关联的值。然而,从零开始训练布隆过滤器对于具有高吞吐量的短时输入流是不实际的,所以 Jo 等人[116]提出了一种改进型学习型布隆过滤器来支持数据更新。

2.3.3 基于学习的事务管理

在事务管理中,事务处理是其核心功能,其核心目的是维护数据的一致性。而并发协议则是事务管理的重要组成部分,负责在多事务同时执行时协调它们的行为。事务管理模块通过使用并发控制协议来防止并发事务之间的冲突,确保数据库的正确性和一致性。

而传统的事务管理系统的核心挑战是处理多个事务同时访问共享数据时,如何确保数据的一致性和系统的高效性。事务在执行过程中不可避免地会产生数据访问冲突,从而导致系统性能下降,甚至引发事务失败或数据错误。因此,并发控制协议成为解决事务冲突的关键机制。

传统的并发控制协议如两阶段锁(Two-Phase Locking,2PL)和乐观并发控制(Optimistic Concurrency Control,OCC)在面对不同类型的工作负载时表现各异。2PL 在高冲突下效果较好,而 OCC 则在低冲突的情况下更为有效。然而,这些协议具有固定的行为模式,缺乏灵活性,难以适应动态、多样化的工作负载环境。为了解决上述问题,越来越多的研究尝试优化事务调度和并发控制,以提高数据库系统的吞吐量和性能。Wang 等人[117]提出一种基于学习的并发控制框架 Polyjuice,该框架通过离线训练寻找最佳的并发控制策略。其核心思想是利用强化学习技术,在细粒度的操作空间中搜索合适的并发控制策略,以最大化事务吞吐量。与传统协议不同,Polyjuice 不会局限于少数固定的并发控制算法,而是通过策略空间的探索,生成能够为特定工作负载优化的新型算法。Polyjuice 的优势在于:(1)适应性强:能够根据特定的工作负载动态调整并发控制策略,从而在高冲突或低冲突环境下均能表现出色;(2)提高系统吞吐量:在 TPC-C 和 TPC-E 等基准测试中,Polyjuice 在中高冲突情况下的吞吐量比传统算法高出 15%至 56%。然而,Polyjuice 的一个局限是需要对工作负载进行离线训练,这使得其难以应对快速变化的动态工作负载场景。此外,虽然 Polyjuice 能够生成新的并发控制策略,但其训练过程可能会带来一定的开销。Cheng 等人[118]则提出了通过优化事务调度来提高性能的策略。其提出的 R-SMF 事务调度系统通过引入 Shortest Makespan First(SMF)调度策略,尽可能减少事务间的冲突,使事务按最低冲突的顺序执行,从而减少总执行时间。与 Polyjuice 专注于并发控制不同,R-SMF 更关注调度层面的优化,尤其是通过优先调度影响最大的热键来减少冲突。R-SMF 的优势在于:(1)SMF 调度策略:通过贪婪算法逐步构建低冲突的调度顺序,从而减少冲突导致的延迟,最大化并发执行的效

率；(2) 并发控制协议：结合多版本时间戳排序协议，R-SMF 能够确保事务按照预定的顺序执行，同时保持高效的并发性，使得 R-SMF 不仅能有效利用调度的优势，还能在执行阶段维持高并发性能。Burke 等人[119]则更专注于重新执行，通过引入事务重新执行方法，探讨了一种在高争用下提高事务系统性能的新方法。其解决的主要挑战是可序列化系统中冲突事务的低效处理，尤其是在高争用工作负载下，乐观并发控制或两阶段锁定等传统方法往往会因死锁或锁抖动而出现高中止率或延迟增加的问题。该系统通过支持部分重新执行冲突事务来克服上述问题，允许系统重新调整事务的序列化窗口，从而有效地减少中止并提高吞吐量。

未来的事务管理系统将会朝着智能化、动态适应性和高效性的方向发展。基于现有的研究，未来的事务管理将向以下方面发展：(1) 学习型和自适应控制：随着人工智能和机器学习技术的发展，类似 Polyjuice 的学习型并发控制将会成为未来事务管理中的重要方向。系统可以根据实时的工作负载特点进行学习和调整，智能地选择最优的并发控制策略，提升系统的吞吐量和性能；(2) 动态事务调度和冲突优化：R-SMF 的调度优化为未来系统提供了一个很好的方向。通过动态调整事务的执行顺序，减少冲突，优化并发性能将是未来事务管理系统的重要研究领域。特别是对于高争用的分布式环境，合理的调度能够显著提高系统的吞吐量；(3) 事务重新执行与细粒度控制：未来的事务管理系统将进一步细化事务操作的粒度，通过局部操作的重新执行来最大化并发性，减少全局回滚带来的开销。这种方法将使得系统能够在处理复杂并发事务时更加高效；(4) 分布式与容错优化：随着分布式系统的广泛应用，事务管理系统需要更多考虑跨地域复制和容错机制。未来系统中，将可能进一步结合多版本时间戳排序协议和其他一致性协议，在保障系统高可用性和一致性的同时，提升分布式环境下的事务处理效率。

### 2.4　基于学习的监控与恢复

数据库监控记录系统的运行状态，检查系统的工作负载，保证数据库的稳定性，对数据库的优化和诊断具有重要意义。例如，节点调整依赖于数据库监控指标。下面依次综述基于学习的数据库监控和基于学习的备份与恢复策略。

#### 2.4.1　基于学习的数据库监控

数据库健康监控记录着与数据库健康相关的指标，例如每秒的查询数、查询延迟，目的是优化数据库或诊断故障。Ma 等人[120]假设带有关键性能指标(Key Performance Indicator，KPI)的间歇性慢查询具有相同的 root cause。因此，采用了两阶段诊断：(1) 离线阶段，从故障记录中提取慢查询，用 KPI 状态进行聚类，并要求 DBA 为每个聚类指定 root cause；(2) 联机阶段，对于传入的慢查询，根据 KPI 状态的相似度分数，将传入的慢查询与集群 $C$ 匹配。如果匹配，则使用集群 $C$ 的根 root cause 通知 DBA；否则，生成一个新的集群，并要求 DBA 分配 root cause。然而，这种方法不能防止潜在的数据库故障，它高度依赖 DBA 的经验，并且监视大量数据库是相当昂贵的，因为监视也会消耗资源。为了解决这一问题，Lahiri 等人[121]提出了一个弹性数据库系统 P-Store，将数据库监控和工作负载预测结合起来，基本思想是主动监视数据库以适应工作负载的变化。

#### 2.4.2　基于学习的数据库备份与恢复

随着信息化时代的到来，各行各业开始重视数据库备份与恢复的应用。当企业的核心数据一旦发生丢失，将给予企业致命打击，这就要求合理应用计算机数据库备份与恢复技术。当数据库遇到未知错误或者天灾人祸时，提前对数据库进行备份是很明智的，事后再进行数据恢复，进而可以挽回一定的损失，为企业的发展提供保障。

传统的关系型数据库都提供了备份与恢复的功能。例如，Oracle 数据备份[122]包括脱机备份、联机备份和逻辑备份；Oracle 数据恢复[123]包括实例恢复和介质恢复。然而，传统方法的备份与恢复速度很慢，已经无法满足大数据应用的需求。即便使用高性能基于闪存的固态驱动器，执行也非常耗时，并且上述方法可能会在正常操作期间对运行时性能产生负面影响。

为此，研究者们开始探索 AI 驱动的备份与恢复策略，例如：Pascal 等人[124]提出了一种使用深度强化学习寻找备份策略的方法。通过用随机过程的语言描述备份策略，可以将寻找最优策略转化为一个强化学习问题。通过训练自主代理，学习如何以最佳方式支持备份过程的规划。为了找到一个备份策略，将问题建模为一个混合离散连续动作空间马尔可夫决策过程，然后使用深层确定性策略梯度进行求解。该方法能够基于实时数据分析来选择最佳的备份策略，从而提高决策的效率和准确性。

通过人工智能驱动的备份与恢复策略的研究尚处于初级阶段，Costa 等人[125]介绍了人工智能在驱动备份与恢复方面的巨大优势，例如：可以利用机器

学习或深度学习算法分析历史数据模式,以预测最佳备份事件,从而优化资源使用和减少系统影响。此外,人工智能还通过自动化的方式选择和应用恢复方法以减少人工干预,加快数据恢复速度。基于此,未来的研究可以进一步探索如何提升数据库管理系统的自主性,使其能够应对不同的故障场景、优化多维数据环境下的资源分配,以及在更加复杂的基础设施中扩展应用范围。

随着人工智能算法的进步,特别是在深度学习和强化学习领域的突破,人工智能驱动的备份与恢复策略有望在未来显著提升数据管理的自动化水平、优化操作效率,并增强系统的韧性。此外,还可以通过智能化故障检测与预警机制,进一步减少数据丢失的风险,提升系统的整体可靠性。

## 2.5 基于学习的安全优化

基于学习的数据库安全旨在利用机器学习技术来保证数据库的机密性、完整性和可用性。本文介绍了最近在敏感数据发现、智能审计、访问控制和漏洞检测方面的工作。

### 2.5.1 基于学习的敏感数据发现

由于敏感数据泄漏会造成巨大的财务和个人信息损失,因此保护数据库中的敏感数据是非常重要的。敏感数据发现旨在自动检测和保护机密数据。传统的方法使用一些用户定义的规则来检测敏感数据[126-127]。例如,Data Sunrise 是一个基于搜索的数据发现框架。它首先定义了一些正则表达式的模式用于表示敏感数据,然后利用这些模式对数据进行模式搜索,以达到检测出敏感数据的目的。然而,这种方法有一些局限性:首先,搜索所有数据的代价高,需要用户指定候选搜索列来修剪搜索空间;其次,当遇到新数据时,不能自动更新规则,因此如果对某些未知的敏感数据没有用户定义的规则,则可能会丢失敏感数据。Bhaskar 等人[128]提出了一种利用机器学习发现敏感数据模式的算法,采用拉普拉斯模型来学习数据记录的真实访问频率,然后将频繁访问的记录作为候选敏感数据。首先,将数据发现描述为一个得分问题:数据集 $D$ 中的每个数据记录 $d$ 被分配一个得分 $S(D,d)$,该得分是数据库中的访问频率。其次,由于抽象的数据集可能很大,递归地对 $k$ 个模式进行采样,对于 $k$ 个模式,用拉普拉斯模型计算频率,拉普拉斯模型根据损失值调整噪声率来拟合真实值。再者,为了简化实现,直接在每个模式的分数中加入独立的拉普拉斯噪声,并选择 $k$ 个具有最高扰动频率的模式,代表了最高的额外风险。通过这种方式,可以在资源有限的情况下提高敏感数据发现的准确性。

### 2.5.2 基于学习的智能审计

传统关系型数据库在部署完成后,实际上会有多个用户参与数据管理。除了管理员用户外,还有更多的普通用户直接进行数据管理。数据库存在一些不可预期的风险,因为用户类别不一样,水平高低也参差不齐。如何快速发现和追溯到这些异常的行为,则需要依赖审计记录机制和审计追踪机制。审计记录的关键在于:(1)定义何种数据库操作行为需要进行日志记录;(2)记录的事件以何种形式展现和存储。只有有效地记录了所关心的行为信息,才能依据这些行为进行问题审计和追溯,实现对系统的有效监督。

人工智能越来越多地应用于各个领域,包括高度监管的领域,例如审计。尽管在审计中使用人工智能表面上看似前景广阔,但迄今为止,有许多因素阻碍了它的广泛应用。Fukas 等人[129]提出了第一个审计人工智能成熟度模型,通过考虑具体的要求,评估了在审计中使用人工智能的优缺点及其推广的可能性。由此产生的模型包含八个不同维度和五个不同的成熟度级别,为进一步使用人工智能技术提供了建议,从而促进审计公司支持人工智能技术。

在法医数据库调查中,审计记录成为重要的证据要素,特别是当某些事件可归因于内部活动时。然而,传统的被动方法可能并不适用,需要采用积极主动的方法来通过审计记录确保问责制,同时满足监管链(Chain of Custody,CoC)对法证的要求。Flores 等人[130]认为角色分离、证据来源、事件及时性和因果关系是 CoC 的要求,以便实现一个可以主动生成、收集和保存数据库审计记录的取证体系结构。Flores 等人建议将触发器和存储过程实现为取证例程,以便构建一个基于时钟向量的时间线,用于解释审计表中记录的事务事件中的因果关系。

### 2.5.3 基于学习的访问控制

为了防止未经授权的用户访问数据,包括表级和记录级访问控制,目前有几种传统的访问控制方法,如基于协议的[30]、基于角色的[131]、基于查询的[132]和基于目的的[133]。然而,上述方法主要基于静态规则,先进的攻击技术可能会伪造访问优先级,传统的方法无法有效地防止这些攻击。近年来,机器学习算法被提出来对访问请求的合法性进行评估。Colombo 等人[134]提出了一种基于目的的访问控制模型,该模型定制了控制策略来调节数据请求。由于不同的行为和数据内容可能导致不同的隐私问题,该方法旨在了解合法访问的目的。

#### 2.5.4　基于学习的漏洞检测

SQL 注入是数据库常见的有害漏洞。攻击者可以通过绕过验证信息或干扰 SQL 语句来修改或查看超出其权限的数据，例如检索隐藏数据、颠倒应用程序逻辑、联合攻击等[135]。例如，应用程序允许用户通过填充类似“SELECT salary, dept from company where dname = '?' and register = 1”的 SQL 语句来访问员工信息。但是攻击者可以通过在 dname 值中添加额外信息来检索未发布产品的隐藏信息，例如“SELECT salary, dept from company where dname='Smith'-- and is_register='yes'”，“--”表示注释符，消除了“released = 1”的限制。然而，传统方法是基于规则的，例如参数化查询，并且有两个限制：首先，扫描非法参数需要很长时间；再者，非法参数的变体很多，无法枚举，传统的特征匹配方法无法识别所有的攻击。目前，利用机器学习的 SQL 注入检测方法主要包括分类树[136-137]和深度学习神经网络[138]两种。Lodeiro-Santiago 等人[137]提出了一种用于检测 SQL 注入的分类算法。由于查询参数中的逻辑错误或过滤错误，经常会发生 SQL 攻击。因此，其基于 SQL 查询提取标记，构建分类器树，以预测可能的 SQL 注入。然而，该方法需要大量的训练数据，不能将知识推广到不同的检测目标。为了解决训练样本有限的问题，Sneha 等人[138]提出了一种基于深度卷积神经网络来解决 SQL 注入危险的方法。其基本思想是使用多个卷积层和全连接层来捕捉并说明 SQL 查询中存在的复杂模式和关联。Dwivedi 等人[139]分析了 NoSQL 数据库，特别是 MongoDB 的安全威胁和缓解策略，揭示了 NoSQL 数据库的潜在安全漏洞，如 NoSQL 注入、缺乏认证和授权、不安全的 REST API 等，并提出了相应的缓解策略，如启用访问控制和认证、查询清理、数据脱敏、使用强加密技术等。遵循上述策略可以有效地保护 NoSQL 数据库的安全，充分发挥其灵活性和可扩展性的优势。

由此可见，在安全优化上，人工智能在 MySQL 中的安全优化可以通过入侵检测系统或异常检测来识别潜在的安全威胁。人工智能模型可以通过分析用户行为模式，发现异常的数据库访问请求，并采取适当的安全措施来保护数据。但在 NoSQL 数据库的安全性更为复杂，因为其分布式特性带来了额外的安全挑战，如数据分片、节点间的通信加密等。人工智能技术可以帮助自动化地检测分布式系统中的安全威胁。通过分析节点间的通信，人工智能可以识别潜在的安全漏洞，如未经授权的访问或数据泄露。

### 2.6　关系型数据库优化技术总结

结合第 2.6 节介绍的关系型数据库优化技术，针对第 1.3 节介绍的 AI4DB 面临的主要挑战，给出解决主要技术问题的文献对照表，如表 5 所示。

**表 5　AI4DB 技术中的典型工作及解决的问题**

| 典型工作 | 所属章节 | 关键技术 | 解决问题 |
|---|---|---|---|
| 文献[55] | 第 2.1 节查询优化 | 通过树型 RNN 估计器建立了基数估计模型 | 解决数据较差和训练时间长问题 |
| 文献[78] | | 通过自适应路由方法和强化学习建立自适应查询处理机制 | 解决训练时间长和 AI 匹配数据库问题 |
| 文献[87] | | 建立 DB-GPT 系统，以优化任务说明和提示 | 解决数据较差和 AI 匹配数据库的问题 |
| 文献[16] | | 利用 Tree-CNN 和图神经网络建立端到端优化算法 | 解决训练时间长和 AI 匹配数据库问题 |
| 文献[43] | 第 2.2 节配置优化 | 提出了一种基于深度强化学习的查询优化方法 GLO | 解决泛化性数据库问题 |
| 文献[106] | | 将数据库调优抽象为强化学习问题建立数据库调优系统， | 解决 AI 匹配数据库和泛化性数据库问题 |
| 文献[109] | 第 2.3 节设计优化 | 利用机器学习与传统 B+树，建立了自适应学习索引结构 | 解决数据较差和 AI 匹配数据库的问题 |
| 文献[111] | | 设计细粒度的分层数据结构，建立了 FINEdex 索引结构 | 主要解决训练时间长和 AI 匹配数据库问题 |
| 文献[117] | | 利用强化学习建立了 Polyjuice 并发控制框架 | 主要解决 AI 匹配数据库和泛化性弱的问题 |
| 文献[125] | 第 2.4 节监控恢复 | 利用机器学习算法分析历史数据，预测最佳备份时机 | 解决数据较差和 AI 匹配数据库的问题 |
| 文献[138] | 第 2.5 节安全优化 | 通过深度卷积神经网络建立了检测 SQL 注入攻击的方法 | 主要解决了 AI 匹配数据库的问题 |
| 文献[139] | | 分析非关系型数据库的安全威胁，提出了缓解策略 | 主要解决 AI 匹配数据库和数据较差问题 |

## 3　未来展望与趋势

本节进一步展望未来机器学习与数据库系统的发展趋势，包括硬件驱动的数据库、云数据库、运维驱动的数据库、应用驱动的数据库、基于自然语言的查询任务以及 DB4AI(DB for AI)。

### 3.1　AI 赋能的数据库优化未来方向

在未来的数据库优化中，人工智能赋能的各类技术将朝着不同的方向发展，主要体现在：(1) 在查询优化方面：基于深度学习的基数估计能够更好地捕捉数据表之间的高维关系，尤其是在处理多表、多列的复杂查询时。因此可预见的是未来将会有更多研究使用深度神经网络、Tree-LSTM 和 CNN

来提高基数估计的精度;而通过强化学习驱动的连接顺序选择将优化连接顺序,尤其是在自适应查询处理和动态计划调整中,这些技术将在实时大数据处理环境中发挥越来越重要的作用。相对而言,基于规则的 SQL 查询重写技术将更加注重实用性,通过引入自然语言处理和大语言模型,数据库系统将更智能地处理和重写 SQL 查询,使其更加适用于不熟悉数据库的用户;(2)在配置优化方面,随着数据库规模和复杂性增加,基于强化学习的自动化节点调整技术将变得更加重要,尤其是在云计算和分布式系统中。强化学习能够通过不断调整系统参数,如内存、缓存、I/O 等,提高系统性能。未来,DRL 在动态负载环境中的自我调节能力将推动这一领域的快速发展。在 AI 动态索引推荐技术中,未来将更多结合 APEX 和 ALEX 等学习型索引技术。数据库可以根据实时工作负载动态生成和调整索引方案,尤其适用于复杂查询和分布式系统。这种结合方式不但能提高索引的自适应性,而且能减少数据库管理员手动调参的工作量,进一步提升数据库管理的智能化水平。而随着应用的成熟,基于 AI 的自动索引系统将逐渐取代 DBA 手动调参的方式,成为数据库管理的标准;(3)在设计优化方面,学习型索引结构的概念将通过深度学习模型优化传统 B+树和布隆过滤器索引结构。由于数据量和查询复杂性逐步增加,使用机器学习进行索引优化将成为研究的热点领域,尤其在高维数据环境下,通过自适应的学习型索引结构提高查询性能的技术将迅速发展;而在事务管理优化方面将会越来越注重实用性,尤其是结合 AI 的事务预测与自动调度系统,将逐渐成为标准应用,减少人为干预,实现事务处理的高效性和准确性;(4)在监控与恢复方面,基于深度学习和强化学习的智能监控与自动恢复技术将在复杂、大规模数据库环境中取得突破性进展,数据库的自我监控、异常检测和故障恢复能力将得到增强,而传统的备份与恢复技术已经开始融入 AI 工具。通过深度学习模型进行备份策略优化,尤其是在云环境中,基于历史数据的恢复策略将更多地向实用性方向发展,逐步成熟并被广泛采用;(5)安全优化方面,AI 将成为数据库安全优化的核心力量,特别是在 SQL 注入检测和防护方面,深度神经网络的异常检测系统能够实时识别和防止复杂攻击,未来的数据库系统将更多依赖 AI 进行自动安全审计和漏洞检测,提高整体安全性。同时,基于 AI 的自动访问控制系统将更具实用性,实时调整权限策略,确保数据安全。总结而言,查询优化、配置优化和安全优化领域的 AI 技术将蓬勃发展,推动更复杂和动态环境下的数据库性能提升,而设计优化、监控与恢复则将逐步成为数据库日常运营和管理的标准工具。

### 3.2 云数据库

随着机器学习和人工智能的广泛应用,智能化数据库开始成为热点,而人工智能在云数据库的索引优化、查询优化等方面有着显著作用。云数据库的最大的特点是多节点部署的模式,即数据和计算分布在多个节点上。这种分布式环境下的索引选择策略需要根据数据的分布方式、查询的类型以及网络延迟等因素来定[140]。例如,常见的方式包括:(1)全局索引:即所有节点上共享同一套索引,这种方法适合全局查询场景,但可能带来同步和一致性开销;(2)局部索引:即每个节点根据本地数据建立各自的索引,适合节点间数据相对独立的情况,但跨节点查询时可能增加复杂性。在某些分布式数据库系统中,每个节点会根据自己存储的数据建立局部索引,但索引的设计可以根据数据分片策略、节点的职责以及查询类型来定。对于热点数据或高频查询的字段,需要在多个节点上建立相同的索引,以提高查询性能;而对于较为稀疏的字段,则可以选择仅在部分节点上建立索引。

但目前云上业务也面临着诸多挑战:(1)索引选择和建立:在云数据库分布式环境下,索引选择策略需要根据数据的分布方式、查询的类型以及网络延迟等因素来确定,不能仅按照固定方式建立索引;(2)在查询优化上:云环境中的数据查询优化比传统数据库更加复杂。业务在云端往往需要处理更大规模的数据集,而数据的物理存储位置分散且动态变化,导致查询的延迟和效率成为瓶颈;(3)在安全优化上:在云上业务中,数据安全性和隐私保护面临更严峻的挑战。由于业务依赖于共享的基础设施,数据泄露和未经授权的访问风险增大,保护敏感数据的难度提升。尽管云上业务面临诸多挑战,但随着人工智能和大数据的深入融合,云平台将为企业提供更智能、更敏捷的解决方案。

### 3.3 运维服务驱动的数据库

随着计算机技术融入各行各业,不少企业如银行已经达到了 IT 即业务的程度。然而,业务越多则系统越复杂,如何管理以及降低成本是重要的问题。在这样的背景下,Gartner 提出了智能运维(Artificial Intelligence for IT Operations,AIOps),AIOps 是让软件和服务工程师通过人工智能和机

器学习技术高效地构建和操作易于支持和维护的服务。AIOps 的意义重大：确保高服务质量和客户满意度，提高工程生产率，降低运营成本。在这个概念基础上，Levin 等人[141]分享了将 AIOps 方法应用于生产云对象存储服务以获得对系统行为和健康状况的可操作说明，描述一个真实的生产云级服务及其操作数据，展示所创建的 AIOps 平台及该平台如何帮助用户解决操作难点，为 AIOps 构建机器学习模型具有其他场景中罕见的挑战。为 AIOps 构建有监督机器学习模型的挑战包括：没有明确的真实标签或者需要巨大的人工努力来获得高质量的标签（不好的标签可能极不平衡、数据量少量、高噪声等）[142]、组件/服务之间的复杂依赖/关系[143]、云服务行为的高度复杂性导致的复杂特征工程、持续的模型更新和在线学习，以及错误的机器模型导致服务中断的风险。在许多 AIOps 场景中，由于难以获得标签数据，只有无监督或半监督机器学习模型是可行的。例如，检测异常服务行为[144]。很难有足够的标签了解一项服务的“什么是不正常的”，因为几乎每个服务是随着客户需求和需求的变化而不断变化的基础设施的变化。高质量的无监督模型的构建难度在于模型的复杂性、服务的内部逻辑和海量的待分析遥测数据[145]。

### 3.4　应用驱动的数据库

从数据库诞生开始，新的应用领域不断为数据库带来新要求，例如巨大的数据量、更短的数据处理时间、更高的可靠性、新的数据类型，数据库也在满足这些新的诉求的同时得到不断的发展与更新。

从历史上来看，通信技术对数据库发展起到了至关重要的作用：1980～1990 年，TCP/IP 网络协议出现，大中型企业内部开始规划部署局域网，甚至通过卫星技术将地域上分散的局域网互联互通，这推动了企业 IT 系统从主机时代走向客户端/服务端（Client/Server，C/S）时代。Oracle 数据库抓住 C/S 架构下数据库系统需要应对更高并发、更多客户端连接的挑战，加大 C/S 架构数据库研发，市场份额直逼当时霸主 IBM 的 DB2 数据库。1990～2000 年，互联网和万维网普及，对运营电子商务和在线购物公司的数据库系统提出更为艰巨的挑战。Oracle 公司抓住市场机会，推出了 RAC（Real Application Clusters）集群数据库，在这一时期最终成为市场霸主。5G 作为最新移动通信技术，其高带宽、极低延迟的特征使其主要应用于增强现实/虚拟现实、云游戏、实时视频通信、无人机、工业互联网等领域。这将对数据库系统带来新的挑战，体现在：（1）终端设备到云端设备网络延迟通常上百毫秒，不利于充分利用 5G 的低延迟特性，考虑在终端设备与云端设备之间部署中小型计算中心，这种部署称为边缘计算。如何将计算和数据在终端-边缘-云之间进行高效的协同，是新型数据库系统的研究方向；（2）5G 网络下，如何解决图像的实时查询和分析等问题，也会成为新型数据库系统的热点话题。

### 3.5　基于自然语言的查询任务

自然语言查询在提升数据库服务的用户体验和效率方面具有重要意义。随着自然语言处理技术（NLP）的快速发展，特别是深度学习模型在处理自然语言问题上的广泛应用，自然语言模型正逐渐被更好地理解和使用[146]。NLP 技术在多个方向上表现出色，包括语言表示、词法句法分析、语义理解以及知识图谱等，这些进步预示着未来用户可能能够通过自然语言进行查询。例如，Pansare 和其他研究人员[147]利用强化学习模型将自然语言转换为结构化查询语言。他们设计了一个序列到序列模型，该模型能够通过强化学习将自然语言翻译成 SQL 语句。为了将这个翻译过程简单化，研究人员在模型中引入了一些其他限制。未来，结合元学习、联邦学习和迁移学习等技术，有望使自然语言翻译成 SQL 的过程更加通用，并能够学习到更复杂的翻译技巧，从而帮助数据库更加普及化。

### 3.6　DB4AI

虽然人工智能可以解决许多现实问题，但由于现有的人工智能系统可复制性差，普通用户难以使用，因此目前还没有一个广泛部署的人工智能系统能够像数据库管理系统那样广泛应用于不同的领域。现有的机器学习平台难以使用，因为用户必须编写代码（例如 Python）才能使用数据发现/清理、建模训练和模型推理的人工智能算法。为了解决这一问题，可以使用数据库技术来降低使用人工智能的障碍。

一方面，数据库系统支持用户在执行数据操作的前提下，利用数据库提供的机器学习服务，以简化其应用过程，该系统的核心关键在于对关系代数的扩充并提供查询接口。另一方面，数据库系统可以利用数据库中诸如哈希表等索引结构来提高机器学习算法的执行速度。此外，系统可以通过集成 AI 基础设施和提供机器学习算法库的方式，来应对数据质量和多样性的高要求，同时这一方式可以减少数据处理和传输的成本进而提高使用效率。

## 4 总 结

本文全面探讨了数据库系统的广泛应用以及传统的数据库系统的困境，并由此引出了数据库系统和人工智能技术的深度结合及目前 AI4DB 的发展动态和面对大数据时代的高性能需求所面临的诸多挑战。文章首先将人工智能优化数据库的现有技术分为了五类：配置优化、异常优化、查询优化、设计优化以及监控与恢复优化。然后，分别概述了这五类数据库优化的基本概念以及不同优化的所有研究方向。进一步地，文章总结了在人工智能结合数据库的过程中普遍遇到的四类挑战，分别为：数据质量较差、训练时间长、泛化性弱以及 AI 与数据库的匹配问题。

进一步，文章对这五类数据库优化的现有方法进行了分析。在查询优化方面，文章概述了目前查询优化的难点问题，如连接顺序选择、SQL 重写、基数估计和端到端优化等。其中在基数估计问题上，文章分析了传统方法如数据画像法、采样法等的优缺点并概述数据库优化组合深度学习之后的方法，如混合模型、全连接模型、深度概率模型、知识图谱等；在连接顺序选择上，文章分析对比了传统方法如动态规划算法，静态学习法以及动态学习法并对比说明这三种方法之间的优劣；在 SQL 重写问题和端到端优化器上，分别阐述了现有的研究动态。在配置优化方面，文章分别概述在索引推荐、视图推荐以及节点调整三个方向上的研究现状并对比介绍机器学习方法和强化学习方法之间的优点与劣势。在设计优化方面，对索引结构、数据存储结构、事务管理三个方向上的现有研究进行了剖析。在监控与恢复方面，文章概述了监控与恢复的重要性及现有研究。在安全优化方面，文章从敏感数据发现、数据审计、访问控制以及漏洞检测四个方向阐述了研究现状。最后，文章针对数据库优化的现有研究，思考了未来可能的研究方向，并展望了未来的 AI4DB 进一步的发展。

**致 谢** 本文得到 2024 年度 CCF-蚂蚁科研基金数据库专项支持，感谢蚂蚁集团科研人员技术上的支持。此外，感谢成都信息工程大学硕士研究生唐榕敏在论文最终修改阶段提出的宝贵建议和修改工作！

## 参 考 文 献

[1] Qiao S, Yang G, Han N, et al. Cardinality estimator: Processing SQL with a vertical scanning convolutional neural network. Journal of Computer Science and Technology, 2021, 36(4): 762-777

[2] Li Guo-Liang, Zhou Xuan-He, Sun Ji, et al. A survey of machine-learning-based database techniques. Chinese Journal of Computers, 2020, 43(11): 2019-2049(in Chinese)
(李国良，周煊赫，孙佶等. 基于机器学习的数据库技术综述. 计算机学报，2020，43(11)：2019-2049)

[3] Wang B, Li W, Li Z, Liao Q. Adaptive linear regression for single-sample face recognition. Neurocomputing, 2013, 115: 186-191

[4] Gallo G, Torrisi A. Random forests based WCE frames classification//Proceedings of the 25th IEEE International Symposium on Computer-Based Medical Systems. Rome, Italy, 2012: 1-6

[5] Jiang H, Ching W K, Chu D. Discriminant analysis in pairwise kernel learning for SVM classification. International Journal of Bioinformatics Research and Applications, 2012, 8(3-4): 305-321

[6] Han M, Liu B. Ensemble of extreme learning machine for remote sensing image classification. Neurocomputing, 2015, 149: 65-70

[7] Zhu Y, Liu J, Guo M, et al. BestConfig: Tapping the performance potential of systems via automatic configuration tuning//Proceedings of the 2017 Symposium on Cloud Computing. Santa Clara, USA, 2017: 338-350

[8] Jiang W, Liang Y, Jiang Z, et al. ABNGrad: Adaptive step size gradient descent for optimizing neural networks. Applied Intelligence, 2024, 54(3): 2361-2378

[9] Gallinucci E, Golfarelli M. SparkTune: Tuning spark SQL through query cost modeling//Proceedings of the Advances in Database Technology-22th International Conference on Extending Database Technology. Lisbon, Portugal, 2019: 546-549

[10] Kobayashi T. Towards deep robot learning with optimizer applicable to non-stationary problems//Proceedings of the 2021 IEEE/SICE International Symposium on System Integration. Iwaki, Japan, 2021: 190-194

[11] Zhang X, Wu H, Li Y, et al. Towards dynamic and safe configuration tuning for cloud databases//Proceedings of the 2022 International Conference on Management of Data. Philadelphia, USA, 2022: 631-645

[12] Li Guo-Liang, Zhou Xuan-He. XuanYuan: An AI-native database systems. Journal of Software, 2020, 31(3): 831-844(in Chinese)
(李国良，周煊赫. 轩辕：AI 原生数据库系统. 软件学报，2020，31(3)：831-844)

[13] Chen J, Chen Y, Chen Z, et al. Data management at Huawei: Recent accomplishments and future challenges// Proceedings of the 35th International Conference on Data Engineering. Macao, China, 2019: 13-24

[14] Zhang X, Chang Z, Wu H, et al. A unified and efficient coordinating framework for autonomous DBMS tuning. ACM on Management of Data, 2023, 1(2): 1-26

[15] Azevedo Santos R, Paes A, Zaverucha G. Transfer learning by mapping and revising boosted relational dependency networks. Machine Learning, 2020, 109: 1435-1463

[16] Marcus R C, Negi P, Mao H, et al. Neo: A learned query optimizer. Proceedings of VLDB Endowment, 2019, 12(11): 1705-1718

[17] Chen L, Huang H, Chen D. Join cardinality estimation by combining operator-level deep neural networks. Information Sciences, 2021, 546: 1047-1062

[18] Gani A, Siddiqa A, Shamshirband S, et al. A survey on indexing techniques for big data: Taxonomy and performance evaluation. Knowledge and Information Systems, 2016, 46(2): 241-284

[19] Leavitt N. Will NoSQL databases live up to their promise. Computer, 2010, 43(2): 12-14

[20] Yokota H, Kanemasa Y, Miyazaki J. Fat-Btree: An update-conscious parallel directory structure//Proceedings of the 15th International Conference on Data Engineering. Sydney, Australia, 1999: 448-457

[21] Zhou W, Peng Q, Zhang Z, et al. GeoGauss: Strongly consistent and light-coordinated OLTP for geo-replicated SQL database. ACM on Management of Data, 2023, 1(1): 1-27

[22] Zhang X, Wu H, Chang Z, et al. ResTune: Resource oriented tuning boosted by meta-learning for cloud databases//Proceedings of the 2021 International Conference on Management of Data. Virtual, China, 2021: 2102-2114

[23] Gu R, Zhang Y, Yin L, et al. Coral: Federated query join order optimization based on deep reinforcement learning. World Wide Web, 2023, 26(5): 3093-3118

[24] Zhou G, Tian W, Buyya R, et al. Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions. Artificial Intelligence Review, 2024, 57(5): 124-140

[25] Sheng Y, Tomasic A, Zhang T, et al. Scheduling OLTP transactions via learned abort prediction//Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management. Amsterdam, The Netherlands, 2019: 1-8

[26] Huang S, Qin Y, Zhang X, et al. Survey on performance optimization for database systems. Science China Information Sciences, 2023, 66(2): 1-19

[27] Grushka-Cohen H, Biller O, Sofer O, et al. Diversifying database activity monitoring with bandits. CoRR abs/1910.10777, 2019

[28] Murillas E G L, Aalst W M P, Reijers H A. Process mining on databases: Unearthing historical data from redo logs// Proceedings of the 13th International Conference on Business Process Management. Rio de Janeiro, Brazil, 2015: 367-385

[29] Iqbal A, Khan S U, Niazi M, et al. Advancing database security: A comprehensive systematic mapping study of potential challenges. Wireless Networks, 2023, 42: 1-28

[30] Ponde S, Kulkarni A, Agarwal R. AI/ML based sensitive data discovery and classification of unstructured data sources// Proceedings of the International Conference on Intelligent Systems and Machine Learning. Hyderabad, India, 2022: 367-377

[31] Sun R, Wang Q, Guo L. Research towards key issues of API security//Proceedings of the China Cyber Security Annual Conference. Beijing, China, 2021: 179-192

[32] Aliero M S, Qureshi K N, Pasha M F, et al. Detection of structure query language injection vulnerability in web driven database application. Concurrency and Computation: Practice and Experience, 2022, 34(13): 1-18

[33] Wartschinski L, Noller Y, Vogel T, et al. VUDENC: Vulnerability detection with deep learning on a natural codebase for Python. Information and Software Technology, 2022, 144: 1-23

[34] Jiménez-Gaona Y, Carrión-Figueroa D, Lakshminarayanan V, et al. Gan-based data augmentation to improve breast ultrasound and mammography mass classification. Biomedical Signal Processing and Control, 2024, 94: 106255

[35] Peng J, Shen D, Nie T, et al. RLclean: An unsupervised integrated data cleaning framework based on deep reinforcement learning. Information Sciences, 2024, 682: 121281

[36] Fulda N, Ventura D A. Dynamic joint action perception for Q-learning agents//Proceedings of the 2003 International Conference on Machine Learning and Applications. Quebec, Canada, 2003: 73-78

[37] Xu L, Qiu S, Yuan B, et al. In-database machine learning with CorgiPile: Stochastic gradient descent without full data shuffle//Proceedings of the 2022 International Conference on Management of Data. Philadelphia, USA, 2022: 1286-1300

[38] Wang Y, Nazir S, Shafiq M. An overview on analyzing deep learning and transfer learning approaches for health monitoring. Computer, 2021, 4: 1-10

[39] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. CoRR abs/1509.02971, 2015

[40] Li J Y, Zhang J, Zhou W, et al. Eigen: End-to-end resource optimization for large-scale databases on the cloud. Proceedings of the VLDB Endowment, 2023, 16(12): 3795-3807

[41] Cai X, Li M, Zhang Y, et al. Multitasking bi-level evolutionary algorithm for data-intensive scientific workflows on clouds. Expert Systems with Applications, 2024, 238: 121833

[42] Xiong X, Yu J, He Z. AutoQuo: An adaptive plan optimizer with reinforcement learning for query plan selection. Knowledge-Based Systems, 2024, 306: 112664

[43] Chen T, Gao J, Tu Y, et al. GLO: Towards generalized learned query optimization//Proceedings of the 2024 IEEE 40th International Conference on Data Engineering. Utrecht, The Netherlands, 2024: 4843-4855

[44] Chen J, Ye G, Zhao Y, et al. Efficient join order selection learning with graph-based representation//Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Washington, USA, 2022: 97-107

[45] Yue Z, Peng S, Cai P, et al. Functionality-aware database tuning via multi-task learning//Proceedings of the 2024 IEEE 40th International Conference on Data Engineering. Utrecht, The Netherlands, 2024: 83-95

[46] Jo S, Trummer I. ThalamusDB: Approximate query processing on multi-modal data//Proceedings of the 2024 ACM SIGMOD International Conference on Management of Data. Santiago, Chile, 2024, 2(3): 1-26

[47] Leis V, Radke B, Gubichev A, et al. Cardinality estimation done right: Index-based join sampling//Proceedings of the Conference on Innovative Data Systems Research. New York, USA, 2017: 8-11

[48] Heule S, Nunkesser M, Hall A. HyperLogLog in practice: Algorithmic engineering of a state of the art cardinality estimation algorithm//Proceedings of the 16th International Conference on Extending Database Technology. Genoa, Italy, 2013: 683-692

[49] Lu X, Guan J. A new approach to building histogram for selectivity estimation in query processing optimization. Computers & Mathematics with Applications, 2009, 57(6): 1037-1047

[50] Poosala V, Haas P J, Ioannidis Y E, et al. Improved histograms for selectivity estimation of range predicates. ACM SIGMOD Record, 1996, 25(2): 294-305

[51] Lipton R J, Naughton J F, Schneider D A. Practical selectivity estimation through adaptive sampling//Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data. New Jersey, USA, 1990: 1-11

[52] Müller M, Moerkotte G, Kolb O. Improved selectivity estimation by combining knowledge from sampling and synopses. The VLDB Endowment, 2018, 11(9): 1016-1028

[53] Praciano F D, Amora P R P, Abreu I C, et al. Robust cardinality: A novel approach for cardinality prediction in SQL queries. Journal of the Brazilian Computer Society, 2021, 27: 1-24

[54] Zhao K, Yu J, He Z, Zhang H. Uncertainty-aware cardinality estimation by neural network Gaussian process. CoRR abs/2107.08706, 2021

[55] Ortiz J, Balazinska M, Gehrke J, Keerthi S S. An empirical analysis of deep learning for cardinality estimation. CoRR abs/1905.06425, 2019

[56] Yu J, He Z. A cardinality estimator in complex database systems based on TreeLSTM. Sensors, 2023, 23(17): 7364:1-7364:19

[57] Yang Z, Kamsetty A, Luan S, et al. NeuroCard: One cardinality estimator for all tables. PVLDB, 2021, 14(1): 61-73

[58] Thongpance N, Dangyai P, Roongprasert K, et al. Exploring ResNet-18 estimation design through multiple implementation iterations and techniques in legacy databases. Journal of Robotics and Control, 2023, 4(5): 650-661

[59] Wang J, Chai C, Liu J, et al. Cardinality estimation using normalizing flow. The VLDB Journal, 2024, 33(2): 323-348

[60] Wang F, Yan X, Yiu M L, et al. Speeding up end-to-end query execution via learning-based progressive cardinality estimation. The ACM on Management of Data, 2023, 1(1): 1-25

[61] Qiao Shao-Jie, Yang Guo-Ping, Han Nan, et al. Cardinality and cost estimator based on tree gated recurrent unit. Journal of Software, 2022, 33(3): 797-813(in Chinese)
(乔少杰, 杨国平, 韩楠等. 基于树型门控循环单元的基数和代价估计器. 软件学报, 2022, 33(3): 797-813)

[62] Heimel M, Kiefer M, Markl V. Self-tuning, GPU-accelerated kernel density models for multidimensional selectivity estimation //Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. Victoria, Australia, 2015: 1477-1492

[63] Hasan S, Thirumuruganathan S, Augustine J, et al. Multi-attribute selectivity estimation using deep learning. CoRR abs/1903.09999, 2019

[64] Yang Z, Liang E, Kamsetty A, et al. Selectivity estimation with deep likelihood models. CoRR abs/1905.04278, 2019

[65] Chen J, Huang Y, Wang M, et al. Accurate summary-based cardinality estimation through the lens of cardinality estimation graphs. ACM SIGMOD Record, 2022, 15(8): 1533-1545

[66] Davitkova A, Gjurovski D, Michel S. LMKG: Learned models for cardinality estimation in knowledge graphs. CoRR abs/2102.10588, 2021

[67] Hayek R, Shmueli O. NN-based transformation of any SQL cardinality estimator for handling distinct. CoRR abs/2004.07009, 2020

[68] Haas L M. Review-access path selection in a relational database management system. ACM SIGMOD Digital Review, 1999, 1: 1-23

[69] Momjian B. PostgreSQL: Introduction and Concepts. New York: Addison-Wesley, 2001: 1-167

[70] Ioannidis Y E, Kang Y C. Left-deep vs. bushy trees: An analysis of strategy spaces and its implications for query optimization//Proceedings of the 1991 ACM SIGMOD International Conference on Management of Data. Denver, USA, 1991: 168-177

[71] Bennett K P, Ferris M C, Ioannidis Y E. A genetic algorithm for database query optimization//Proceedings of the 1st IEEE Conference on Evolutionary Computation. Orlando, USA, 1994: 350-355

[72] Waas F, Pellenkoft A. Join order selection-good enough is easy//Proceedings of the 17th British National Conference on Databases. Exeter, UK, 2000: 51-67

[73] Fegaras L. A new heuristic for optimizing large queries// Proceedings of the 9th International Conference on Database and Expert Systems Applications. Vienna, Austria, 1998, 1460: 726-735

[74] Krishnan S, Yang Z, Goldberg K, et al. Learning to optimize join queries with deep reinforcement learning. CoRR abs/1808.03196, 2018

[75] Marcus R, Papaemmanouil O. Deep reinforcement learning for join order enumeration//Proceedings of the 1st International Workshop on Exploiting Artificial Intelligence Techniques for Data Management. Houston, USA, 2018: 1-4

[76] Kameswaran S, Biegler L T. Simultaneous dynamic optimization strategies: Recent advances and challenges. Computers & Chemical Engineering, 2006, 30(10-12): 1560-1575

[77] Yu X, Li G, Chai C, et al. Reinforcement learning with tree-LSTM for join order selection//Proceedings of the 36th International Conference on Data Engineering. Dallas, USA, 2020: 1297-1308

[78] Avnur R, Hellerstein J M. Eddies: Continuously adaptive query processing//Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. Dallas, USA, 2000: 261-272

[79] Trummer I, Wang J, Wei Z, et al. SkinnerDB: Regret-bounded query evaluation via reinforcement learning. ACM Transactions on Database Systems, 2021, 46(3): 1-45

[80] Hickling T, Zenati A, Aouf N, et al. Explainability in deep reinforcement learning: A review into current methods and applications. ACM Computing Surveys, 2023, 56(5): 1-35

[81] Tony K, Shaji K S, Noble N, et al. NL2SQL: Rule-based model for natural language to SQL//Proceedings of the International Conference on Paradigms of Communication, Computing and Data Analytics. New Delhi, India, 2023: 817-828

[82] Weiner A M, Mathis C, Härder T. Rules for query rewrite in native XML databases//Proceedings of the 2008 EDBT Workshop on Database Technologies for Handling XML Information on the Web. Nantes, France, 2008: 21-26

[83] Giavitto J L, Michel O, Cohen J. Pattern-matching and rewriting rules for group indexed data structures. ACM SIGPLAN Notices, 2002, 37(12): 55-66

[84] Wang Z, Zhou Z, Yang Y, et al. WeTune: Automatic discovery and verification of query rewrite rules//Proceedings of the 2022 International Conference on Management of Data. Philadelphia, USA, 2022: 94-107

[85] Zhou X, Li G, Chai C, et al. A learned query rewrite system using Monte Carlo tree search. Proceedings of the VLDB Endowment, 2021, 15(1): 46-58

[86] Liu J, Mozafari B. Query rewriting via large language models. CoRR abs/2403.09060, 2024

[87] Zhou X, Sun Z, Li G. DB-GPT: Large language model meets database. Data Science and Engineering, 2024, 9(1): 102-111

[88] Kaseb M R, Haytamy S S, Badry R M. Distributed query optimization strategies for cloud environment. Information and Management, 2021, 3(4): 271-279

[89] Wang C, Gruenwald L. Cloud query processing with reinforcement learning-based multi-objective re-optimization//Proceedings of the Model and Data Engineering: 10th International Conference. Tallinn, Estonia, 2021, 12732: 141-160

[90] Zhou X, Chai C, Li G, et al. Database meets artificial intelligence: A survey. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(3): 1096-1116

[91] Chaudhuri S, Narasayya V R. An efficient, cost-driven index selection tool for Microsoft SQL server//Proceedings of the VLDB Conference. Athens, Greece, 1997: 146-155

[92] Valentin G, Zuliani M, Zilio D C, et al. DB2 advisor: An optimizer smart enough to recommend its own indexes//Proceedings of the 16th International Conference on Data Engineering. San Diego, USA, 2000: 101-110

[93] Yu W, You J, Niu X, et al. Rboira: Integrating rules and reinforcement learning to improve index recommendation. EAI Endorsed Transactions on Scalable Information Systems, 2023, 10(6): 96-116

[94] Gu T, Feng K, Cong G, et al. A reinforcement learning based R-tree for spatial data indexing in dynamic environments. IEEE Transactions on Knowledge and Data Engineering, 2021, 30(7): 619-630

[95] Qiao Shao-Jie, Liu Chen-Xu, Han Nan, et al. AP-IS: An intelligent and efficient index selection model for multimodal data. Acta Automatica Sinica, 2025, 51(2): 457-474(in Chinese)
(乔少杰，刘晨旭，韩楠等. AP-IS：面向多模态数据的智能高效索引选择模型. 自动化学报，2025，51(2)：457-474)

[96] Sadri Z, Gruenwald L, Leal E. Online index selection using deep reinforcement learning for a cluster database//Proceedings of the 36th International Conference on Data Engineering Workshops. Dallas, USA, 2020: 158-161

[97] He J, Zhao H, Zhou D, et al. Nearly minimax optimal reinforcement learning for linear Markov decision processes//Proceedings of the International Conference on Machine Learning. Honolulu, USA, 2023: 12790-12822

[98] Zilio D C, Zuzarte C, Lightstone S, et al. Recommending materialized views and indexes with the IBM DB2 design advisor//Proceedings of the International Conference on Autonomic Computing. New York, USA, 2004: 180-187

[99] Xu Z, Kakkar G T, Arulraj J, et al. EVA: A symbolic approach to accelerating exploratory video analytics with materialized views//Proceedings of the 2022 International Conference on Management of Data. Philadelphia, USA, 2022: 602-616

[100] Han Y, Li G, Yuan H, et al. AutoView: An autonomous materialized view management system with encoder-reducer. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(6): 5626-5639

[101] Dokeroglu T, Bayir M A, Cosar A. Robust heuristic algorithms for exploiting the common tasks of relational cloud database queries. Applied Soft Computing, 2015, 30: 72-82

[102] Yuan H, Li G, Feng L, et al. Automatic view generation with deep learning and reinforcement learning//Proceedings of the 36th International Conference on Data Engineering. Dallas, USA, 2020: 1501-1512

[103] Liang X, Elmore A J, Krishnan S. Opportunistic view materialization with deep reinforcement learning. CoRR abs/1903.01363, 2019

[104] Li G, Zhou X, Sun J, et al. OpenGauss: An autonomous database system. The VLDB Endowment, 2021, 14(12): 3028-3042

[105] Van Aken D, Pavlo A, Gordon G J, et al. Automatic database management system tuning through large-scale machine learning //Proceedings of the 2017 ACM International Conference on Management of Data. Chicago, USA, 2017: 1009-1024

[106] Zhang J, Liu Y, Zhou K, et al. An end-to-end automatic cloud database tuning system using deep reinforcement learning//Proceedings of the 2019 International Conference on Management of Data. Amsterdam, The Netherlands, 2019: 415-432

[107] Giannakouris V, Trummer I. λ-Tune: Harnessing large language models for automated database system tuning// Proceedings of the 2025 ACM SIGMOD International Conference on Management of Data. Stockholm, Sweden, 2025, 3(1): 1-26

[108] Kraska T, Beutel A, Chi E H, et al. The case for learned index structures//Proceedings of the 2018 International Conference on Management of Data. Houston, USA, 2018: 489-504

[109] Ding J, Minhas U F, Yu J, et al. ALEX: An updatable adaptive learned index//Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. Portland, USA, 2020: 969-984

[110] Chen L, Chen S. How does updatable learned index perform on non-volatile main memory?//Proceedings of the 2021 IEEE 37th International Conference on Data Engineering Workshops. Chania, Greece, 2021: 66-71

[111] Li P, Hua Y, Jia J, et al. FINEdex: A fine-grained learned index scheme for scalable and concurrent memory systems. The VLDB Endowment, 2021, 15(2): 321-334

[112] Ma C, Yu X, Li Y, et al. FILM: A fully learned index for larger-than-memory databases. The VLDB Endowment, 2022, 16(3): 561-573

[113] Idreos S, Dayan N, Qin W, et al. Design continuums and the path toward self-designing key-value stores that know and learn//Proceedings of the Conference on Innovative Data Systems Research. Asilomar, USA, 2019: 13-16

[114] Idreos S, Dayan N, Qin W, et al. Learning key-value store design. CoRR abs/1907.05443, 2019

[115] Mitzenmacher M. A model for learned bloom filters and optimizing by sandwiching//Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal, Canada, 2018: 384-401

[116] Jo H, Shikata J. Bloomier filters on 3-hypergraphs// Proceedings of the International Conference on Information Security Applications. Jeju Island, Republic of Korea, 2023: 16-26

[117] Wang J, Ding D, Wang H, et al. Polyjuice: High-performance transactions via learned concurrency control//Proceedings of the 15th USENIX Symposium on Operating Systems Design and Implementation. Virtual, USA, 2021: 198-216

[118] Cheng A, Kabcenell A, Chan J, et al. Towards optimal transaction scheduling. The VLDB Endowment, 2024, 17 (11): 2694-2707

[119] Burke M, Suri-Payer F, Helt J, et al. Morty: Scaling concurrency control with re-execution//Proceedings of the 18th European Conference on Computer Systems. Rome, Italy, 2023: 687-702

[120] Ma M, Yin Z, Zhang S, et al. Diagnosing root causes of intermittent slow queries in cloud databases. The VLDB Endowment, 2020, 13(8): 1176-1189

[121] Lahiri T, Ganesh A, Weiss R, et al. Fast-start: Quick fault recovery in oracle//Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data. Santa Barbara, USA, 2001: 593-598

[122] Son Y, Kim M, Kim S, et al. Design and implementation of SSD-assisted backup and recovery for database systems. IEEE Transactions on Knowledge and Data Engineering, 2018, 32(2): 260-274

[123] Wang Q. Cloud data backup and recovery method based on the DELTA compression algorithm//Proceedings of the 2021 IEEE International Conference on Industrial Application of Artificial Intelligence. Harbin, China, 2021: 183-188

[124] Debus P, Müller N, Böttinger K. Deep reinforcement learning for backup strategies against adversaries. CoRR abs/2102.06632, 2021

[125] Costa R. Implementing AI-driven backup and recovery strategies in modern database systems. Journal of Innovative Technologies, 2023, 6(1): 1-21

[126] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. ACM SIGMOD Record, 2000, 29(2): 1-12

[127] Ruggieri S, Pedreschi D, Turini F. DCUBE: Discrimination discovery in databases//Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. Indianapolis, USA, 2010: 102-113

[128] Bhaskar R, Laxman S, Smith A, et al. Discovering frequent patterns in sensitive data//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, USA, 2010: 503-512

[129] Fukas P, Rebstadt J, Remark F, et al. Developing an artificial intelligence maturity model for auditing//Proceedings of the European Conference on Information Systems. Marrakech, Morocco, 2021: 15-17

[130] Flores D A, Jhumka A. Implementing chain of custody requirements in database audit records for forensic purposes// Proceedings of the 2017 IEEE Trustcom/BigDataSE/ICESS. Sydney, Australia, 2017: 675-682

[131] Saxena U R, Alam T. Provisioning trust-oriented role-based access control for maintaining data integrity in cloud. International Journal of System Assurance Engineering and Management, 2023, 14(6): 2559-2578

[132] Mohamed A K Y S, Auer D, Hofer D, et al. A systematic literature review for authorization and access control: Definitions, strategies and models. International Journal of Web Information Systems, 2022, 18(2/3): 156-180

[133] Byun J W, Li N. Purpose based access control for privacy protection in relational database systems. The VLDB Journal, 2008, 17: 603-619

[134] Colombo P, Ferrari E. Efficient enforcement of action-aware purpose-based access control within relational database management systems. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(8): 2134-2147

[135] Alghawazi M, Alghazzawi D, Alarifi S. Detection of SQL injection attack using machine learning techniques: A systematic literature review. Journal of Cybersecurity and Privacy, 2022, 2(4): 764-777

[136] Sheykhkanloo N M. A learning-based neural network model for the detection and classification of SQL injection attacks. International Journal of Cyber Warfare and Terrorism, 2017, 7(2): 16-41

[137] Lodeiro-Santiago M, Caballero-Gil C, Caballero-Gil P. Collaborative SQL-injections detection system with machine learning//Proceedings of the 1st International Conference on Internet of Things and Machine Learning. Liverpool, UK, 2017: 1-5

[138] Sneha K, Singh H. Augmenting SQL injection attack detection via deep convolutional neural network. The VLDB Journal, 2024, 10(2): 120-139

[139] Dwivedi S, Balaji R, Ampatt P, et al. A survey on security threats and mitigation strategies for NoSQL databases: MongoDB as a use case//Proceedings of the International Conference on Information Systems Security. Raipur, India, 2023: 57-76

[140] Gadde H. AI-driven data indexing techniques for accelerated retrieval in cloud databases. Revista de Inteligencia Artificial en Medicina, 2024, 15(1): 583-615

[141] Levin A, Garion S, Kolodner E K, et al. AIOps for a cloud object storage service//Proceedings of the 2019 IEEE International Congress on Big Data. Milan, Italy, 2019: 165-169

[142] Luo C, Zhao P, Qiao B, et al. NTAM: Neighborhood-temporal attention model for disk failure prediction in cloud platforms//Proceedings of the Web Conference 2021. Ljubljana, Slovenia, 2021: 1181-1191

[143] Saxena D, Singh A K. OFP-TM: An online VM failure prediction and tolerance model towards high availability of cloud computing environments. The Journal of Supercomputing, 2022, 78(6): 8003-8024

[144] Girish L, Rao S K N. Anomaly detection in cloud environment using artificial intelligence techniques. Computing Surveys, 2023, 105(3): 675-688

[145] Notaro P, Cardoso J, Gerndt M. A survey of AIOps methods for failure management. ACM Transactions on Intelligent Systems and Technology, 2021, 12(6): 1-45

[146] Patil R, Boit S, Gudivada V, et al. A survey of text representation and embedding techniques in NLP. IEEE Access, 2023, 11: 36120-36146

[147] Pansare S D, et al. Reinforcement learning for improving coherence of multi-turn responses in deep learning-based chatbots//Proceedings of the International Conference on Communication, Circuits, and Systems. Bhubaneswar, India, 2021: 273-279

**QIAO Shao-Jie**, Ph. D., professor, Ph. D. supervisor. His current research interests include artificial intelligence for databases, query optimization, moving objects databases.

**LI Zhou**, M. S. candidate. His current research interests include artificial intelligence for databases.

**HAN Nan**, Ph. D., associate professor. Her current research interests include databases and data mining.

**XU Quan-Qing**, Ph. D., senior engineer. His current research areas include database systems and distributed systems.

**WU Tao**, Ph. D., professor, Ph. D. supervisor. His current research areas include intelligent security and privacy protection.

**YUAN Guan**, Ph. D., professor, Ph. D. supervisor. His current research areas are artificial intelligence and big data technology.

**WU Xin-Dong**, Ph. D., professor, Ph. D. supervisor. His current research areas are data mining, big data analytics, and knowledge engineering.

**Background**

This work is a part of the National Natural Science Foundation of China under Grant No. 62272066 (Research on key technologies of learning autonomous database systems). The project focuses on designing the state-of-the-are techniques of artificial intelligence for database (AI4DB). The difficulties and challenges of AI4DB contain: (1) the data quality is poor, the amount of data is small, and the data diversity cannot be guaranteed; (2) the training time of most AI algorithms is costly; (3) the generalization of the model is too weak to be universal; (4) problems of artificial intelligence matching databases. There are still many difficult and challenging problems that are urgently needed to be solved in database optimization technology, including environment construction, query optimization and scheduling mechanism.

This survey can help database developers and researchers better understand the current development and challenges of database optimization techniques based on machine learning. The AI4DB techniques using artificial intelligence approaches to optimize databases are reviewed, including learning-based configuration tuning, optimizer and index/view recommendation. By combining with practical application scenarios, this survey further discusses the research prospect and development trend of the combination of database and artificial intelligence.