Recursos Educacionais / Educational Resources

Como ler (e interpretar) uma base de dados de grandes dimensões

Luís Cavique
DCeT, Univ. Aberta
Luis.Cavique@uab.pt

Lisboa, setembro 2022

Este documento pretende complementar a bibliografia da UC sobre Sistemas de Gestão de Base de dados oferecida no 1º semestre do 3º ano, na Licenciatura em Engenharia Informática.

Apesar da simplicidade do SQL, ler grandes bases de dados com centenas de tabelas pode ser um grande desafio.

Neste trabalho pretendemos detalhar os seguintes assuntos em bases de dados relacionais:

- identificar diferentes tipos de tabelas numa base de dados (inter-tabelas),
- classificar as tabelas de acordo com sua capacidade de agregação de dados (intra-tabelas),
- apresentar um procedimento para extrair o significado de uma grande base de dados.

Um estudo de caso é incluído para exemplificar o procedimento proposta na leitura e interpretação de uma base de dados.

Índice

1. O problema

Nos exemplos didáticos de base de dados são utilizadas geralmente uma dezena de tabelas. Contudo, uma base de dados com 60 tabelas, torna a sua leitura e interpretação bem mais desafiante, tal como apresentado na Figura 1.

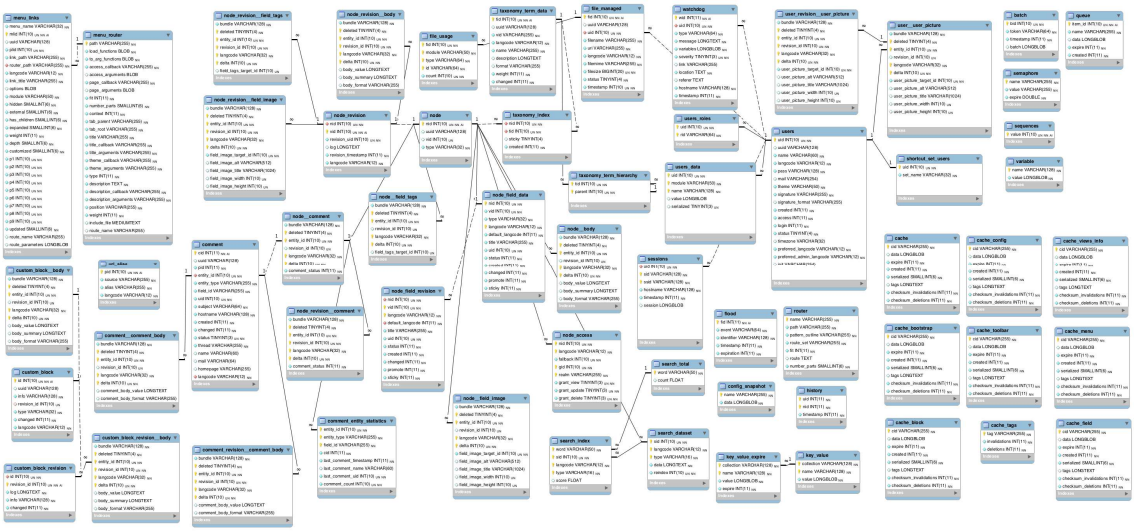

Figura 1. Esquema de uma base de dados com 60 tabelas

As dimensões de bases de dados reais usam centenas de tabelas. Como referência podemos considerar que:

- Pequena base de dados ≤ 100 tabelas
- Base de dados média ≅ 200 tabelas
- Grande base de dados ≥ 500 tabelas

Dada uma base de dados genérica, pretendemos saber que informação útil ela contem. Que informação contem a base de dados? Que dados agregados podem ser extraídos? Que possíveis consultas (*queries*) podem ser colocadas?

2. Conceitos essenciais

Nesta secção apresentam-se alguns conceitos essenciais, tais como: os grafos acíclicos diretos, os tipos de tabelas segundo as suas ligações (chaves estrangeiras), e a classificação das tabelas segundo a forma da agregação dos seus dados. A classificação usando as chaves estrangeiras é usada entre várias tabelas ou inter-tabelas. A classificação ao nível da agregação dos dados, é apresentada dentro de cada tabela ou intra-tabela.

2.1. Grafos acíclicos diretos, DAG

Em teoria dos grafos e na ciência da computação, um grafo acíclico direcionado (DAG, direct acyclic graph) é um grafo orientado (com arcos) e sem ciclos. O grafo G(V, A) é constituído por vértices, V, e arestas orientadas, A, (também chamadas de arcos), com cada aresta é direcionada de um vértice a outro. No caso do grafo acíclico as direções nunca formam um ciclo fechado. Um grafo é um DAG se e só se puder ser ordenado topologicamente, i.e., organizando os vértices com uma ordenação linear consistente com todas as direções das arestas [Christofides 1975].

Na Figura 2 é apresentado um DAG com ordenação topológica utilizando as letras de ‘a’ a ‘e’.

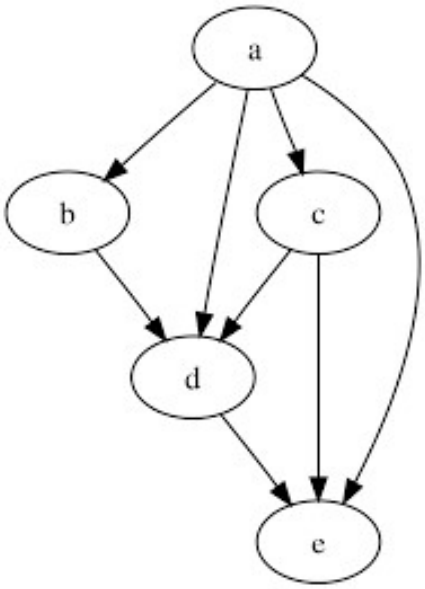

Figura 2. DAG, grafo acíclico direto

Na nossa visualização do DAG as setas (ou mais formalmente, os arcos) são orientadas de cima para baixo (ou da esquerda para a direita).

Na representação das bases de dados em vez de setas (->) vamos utilizar os usuais pés-de-galinha (1 -< N). Na vossa visualização as ligações de 1:N, a tabela com um único identificador (1) é desenhada em cima e a tabela com vários identificadores (N) é desenhada por baixo.

2.2. Os três tipos de tabelas

Reutilizando os trabalhos de Cavique et al. [2019] e Cavique et al. [2020] que identificam três tipos de tabelas, conforme mostra a Figura 3:
• tabelas de consulta (*lookup*) são tabelas com cardinalidade igual a 1,
• tabelas do meio para tabelas com cardinalidade 1 e N, e
• tabelas de eventos são tabelas com cardinalidade igual a N.

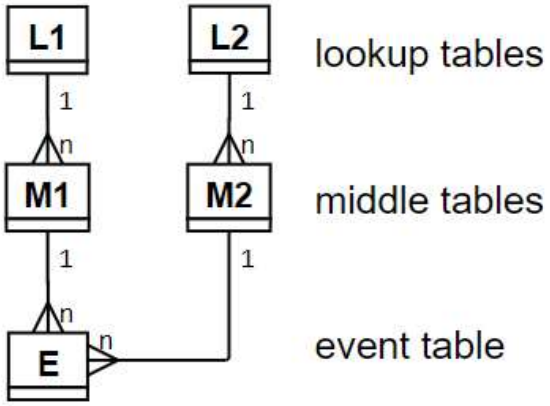

Figura 3. Três tipos de tabelas

As tabelas de consulta são tabelas muito estáveis, com poucas alterações de dados (inserção, alteração e remoção) e com muitas leituras. As tabelas do meio fazem a ligação entre as tabelas de consulta e as tabelas de eventos. As tabelas de eventos são as tabelas com mais alterações de dados (inserção, alteração e remoção), com dados muitos dependentes dos anteriores e onde se guardam os resultados mais relevantes da base de dados.

Nos *data warehouses* as tabelas correspondente às tabelas de eventos tem o nome de tabelas de factos. Neste trabalho houve a preocupação de dar um nome diferente.

Nesta subsecção a classificação é realizada com base nas chaves estrangeiras entre várias tabelas ou inter-tabelas. Na próxima subsecção a classificação realiza-se ao nível da agregação dos dados dentro de cada tabela ou intra-tabela.

2.3. Classificação das tabelas de eventos

Vamos escolher as tabelas de eventos com mais movimentos e alterações, para categorizar relativamente à capacidade de agregação dos dados. Como estamos a trabalhar dentro de uma tabela de cada vez, realizamos uma análise intra-tabela.

Neste trabalho reutilizamos os conceitos de factos aditivos, semi-aditivos e não aditivos dos *data warehouses* [Kimball, Ross 2016], [Santos, Ramos 2017]. Vamos categorizar os eventos em quatro grupos:

- Eventos aditivos: são atributos que podem ser agregados (ou somados) por todas as dimensões, por exemplo, o valor de venda (use sempre a função Sum())
- Eventos semi-aditivos: são atributos que podem ser agregados (ou somados) por algumas dimensões, por exemplo, quantidade (use a função Sum() em condições particulares)
- Eventos não aditivos: são atributos que não podem ser agregados (ou somados), por exemplo, o preço unitário (use a função Average() para encontrar o preço unitário médio)
- Sem eventos: existem apenas identificadores (use a função Count()).

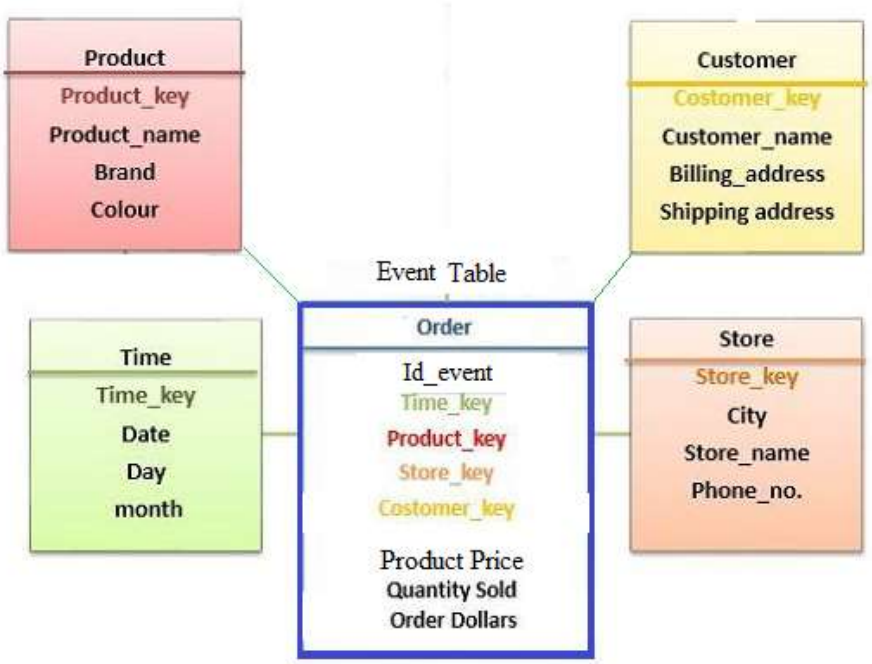

Figura 4. Tabela de eventos com 4 dimensões

Considerando a tabela de eventos da Figura 4, os atributos:

- Order dollars é um atributo aditivo;
- Quantity sold é um atributo semi-aditivo, e aditivo por Product_Key;
- Product Price é um atributo não aditivo;
- Id_event é um atributo que só pode ser contado, mas nunca somado.

2.4. Análise das tríades

Tríade é um conjunto de 3 tabela com 3 relações (ou chaves estrangeiras), tratando-se de um caso particular de um problema de aceso por caminhos múltiplos. O Multiple Access Path Problem (MAPP) é apresentado em [Hall 1986], [Cavique 2021].

Algumas tríades são redundantes nas bases de dados e podem ser removidas. Contudo, deve ser analisada com cuidado o contexto de cada situação [NIST 1993].

Na Figura 5 são apresentados dois casos com exemplos de conteúdos das chaves estrangeiras (FK). No caso A, o funcionário (FK="John Smith") que faz o aluguer é o mesmo que faz o pagamento, pelo que existe redundância de informação nas chaves estrangeiras e a ligação Staff-Payment pode ser retirada. Já no caso B, a loja tem uma morada (FK="Ferry Street") e o funcionário da loja vive numa morada diferente (FK="Market Street"), pelo que não existe redundância e é de manter a tríade.

Sempre que estamos na presença de uma tríade tem de ser garantida que a informação presente nas duas chaves estrangeiras é diferente, e, portanto, não redundante.

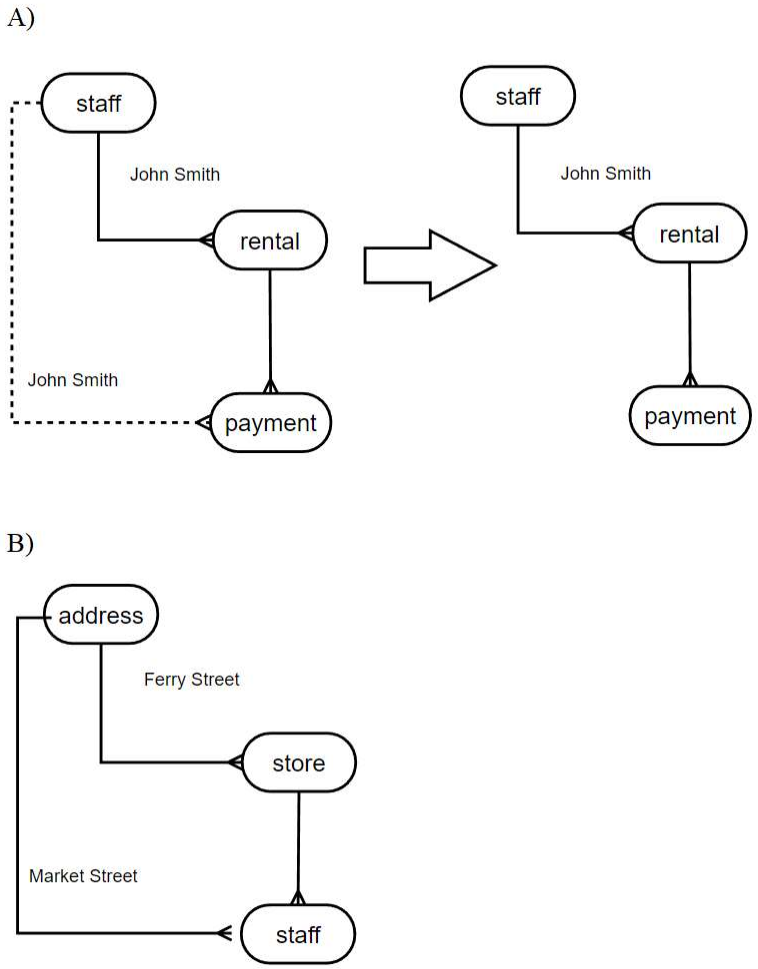

Figura 5. Tríade redundante (caso A) e tríade não redundante (caso B)

3. Método proposto

No método proposto para ler e interpretar uma base de dados, visto que existem centenas de tabelas, teremos de nos cingir ao essencial. O essencial corresponde às tabelas de eventos. Dentro de todas as tabelas de eventos o utilizador pode escolher, por exemplo, só os eventos aditivos. Por fim, a geração do DAG reduzido recorre à análise da redundância das tríades. Em Procedimento 1 detalhamos o processo.

Procedimento 1: Ler_Base_de_Dados

Entrada: Base de dados

Saída: DAGs reduzidos

(i) Encontre todas as tabelas de eventos $E = \{E^1, E^2, \ldots, E^n\}$

(ii) Para cada tabela de eventos E^i

Classifique os atributos de E^i (aditivos .. sem eventos)

Se a eventos aditivos desenhe o DAG^i

Analise das tríades do DAG^i criando um DAG reduzido

FimPara

O duplo filtro (inter e intra-tabelas) e a representação com um DAG reduzido vai simplificar a visualização da base de dados.

4. Estudo de caso

Considere a base de dados de Aluguer de DVD apresentada na Figura 6. Esta base de dados com 15 tabelas tem o nome Sakila dos exemplos do MySQL.

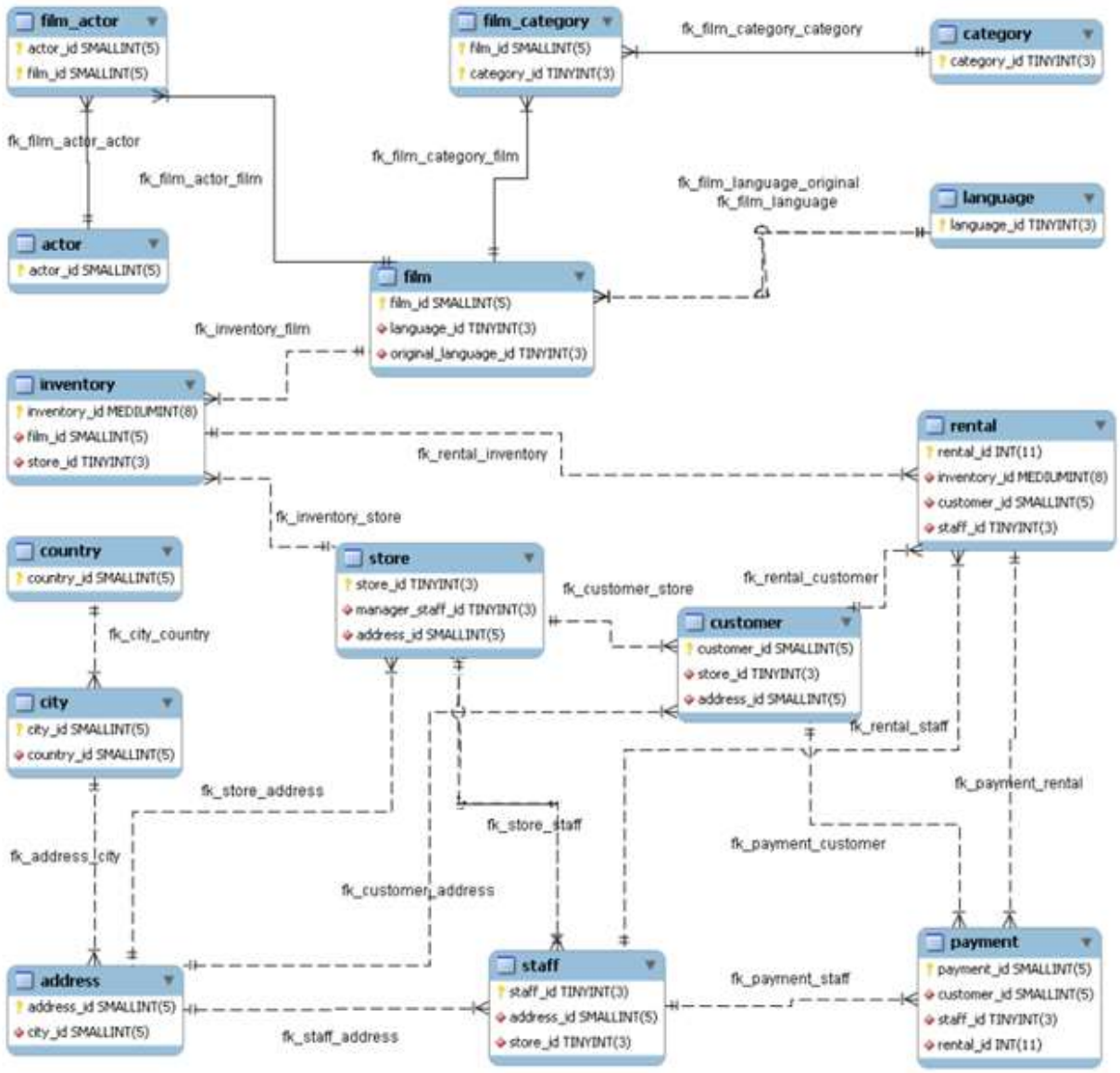

Figura 6. Base de dados Sakila

Aplicando o primeiro passo do Procedimento 1, encontramos E = {film_category, film_actor, payment}.

De seguida, classificamos cada evento segundo a sua capacidade de agregação dos dados, obtendo a Tabela 1.

Tabela 1. Lista das tabelas de eventos e classificação

Tabelas de eventos	**Classificação**
film_category	Sem eventos
film_actor	Sem eventos
payment	Aditivo no atributo *Amount*

A única tabela com eventos aditivos é a tabela Payment. Para a tabela Payment é desenhado um DAG representado na Figura 7.

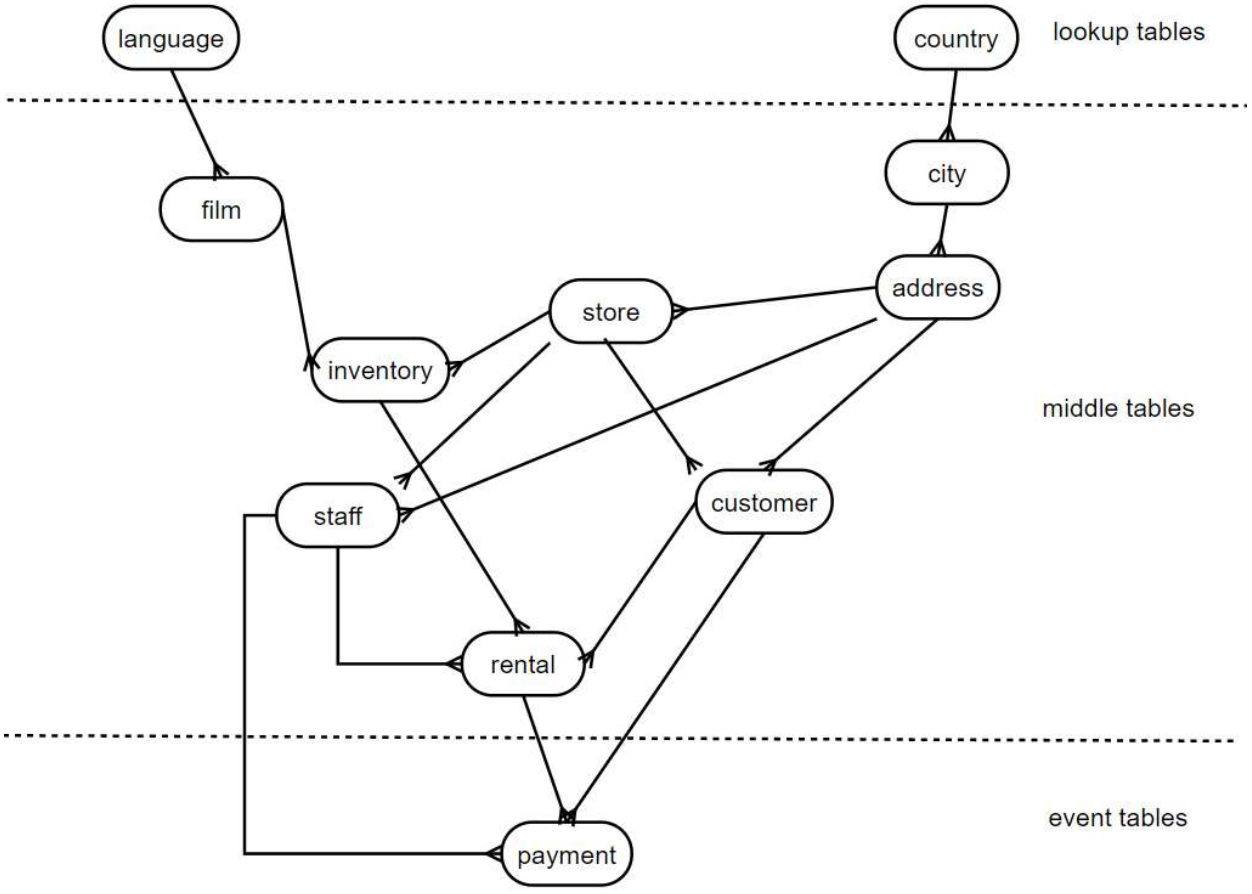

Figura 7. DAG para a tabela de eventos Payment

De seguida é verificada a possibilidade de redundância e posterior remoção de tríades:
- na tríade Staff-Rental-Payment, Staff-Payment é removida
- na tríade Customer-Rental-Payment, Customer-Payment é removida
- na tríade Address-Store-Customer, Store-Customer é removida

O DAG reduzido da tabela de eventos Payment é apresentado na Figura 8.

A leitura da base de dados torna-se agora mais clara. A área de negócio de bases de dados Sakila é o aluguer de DVD. Cada loja (Store) tem um conjunto de DVD (Inventory) e vários funcionários (Staff). Em cada aluguer (Rental) está envolvido um DVD de um filme (Inventory), um cliente (Customer) e um funcionário (Staff). Para cada aluguer corresponde um pagamento (Payment). As lojas, os funcionários e os clientes têm moradas específicas.

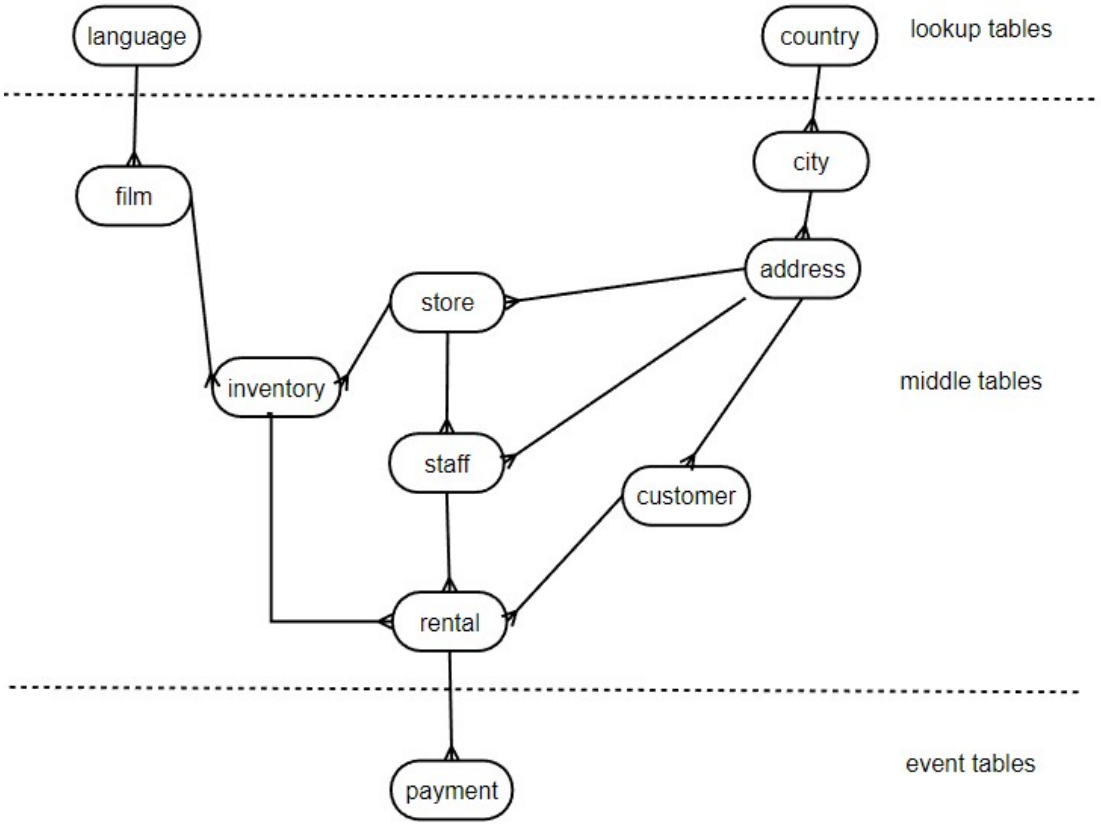

Figura 8. DAG reduzido para a tabela de eventos Payment

5. Conclusões

Ler e interpretar uma base de dados relacional com 200 ou mais tabelas é um grande desafio.

O procedimento proposto é composto por duas fases: um filtro inter-tabela e um segundo filtro intra-tabela. Na primeira fase usamos um filtro inicial às tabelas de eventos (com cardinalidade igual a N) visto que o núcleo do negócio se concentra nessas tabelas.

Conforme exemplificado no estudo de caso, aplica-se um segundo filtro às tabelas de eventos com atributos aditivos, sendo esses os que podem produzir um maior impacto quantitativo. Finalmente, é possível visualizar os DAG das tabelas de eventos com atributos aditivos, e realizar a leitura e interpretação da base de dados.

Referências

Cavique L. (2021), Armadilhas em consultas SQL: em Bases de Dados Relacionais, Recursos Educacionais, Univ. Aberta.

Cavique L., M. Cavique, A. Gonçalves (2019), Extraction of Fact Tables from a Relational Database: An Effort to Establish Rules in Denormalization, in: Rocha Á., Adeli H., Reis L., Costanzo S. (eds), New Knowledge in Information Systems and Technologies, WorldCIST 2019, Advances in Intelligent Systems and Computing, Springer, Cham, vol. 930, pp. 936–945. https://doi.org/10.1007/978-3-030-16181-1_88

Cavique L., M. Cavique, J. Santos (2020), Supply-demand matrix: a process-oriented approach for data warehouses with constellation schemas, in Rocha Á., Adeli H., Reis L., Costanzo S., Orovic I., Moreira F. (eds) Trends and Innovations in Information Systems and Technologies, WorldCIST 2020, Advances in Intelligent Systems and Computing, vol. 1159, Springer, Cham. https://doi.org/10.1007/978-3-030-45688-7_33

Christofides N. (1975), Graph theory: an algorithmic approach, Academic Press.

Hall G.W. (1986), Querying cyclic databases in natural language, Master of Science in the Scholl of Computer Science, Simon Fraser University.

Kimball R., M. Ross (2016), The Kimball Group Reader, Relentlessly Practical Tools for Data Warehousing and Business Intelligence, Remastered Collection, Wiley, ISBN-13: 978-1119216315.

NIST, National Institute of Standards and Technology (1993), Integration Definition for Information Modeling (IDEF1X), Federal Information Processing Standards Publication 184, Gaithersburg, USA.

Santos M.Y., I. Ramos (2017), Business Intelligence, Da Informação ao Conhecimento (3ª Edição Atualizada), coleção Data Science, FCA editora, ISBN 9789727228805.