Análise de Dados

6.ª Aula Prática Laboratorial

Mestrado Integrado em Engenharia Informática

Ano Letivo 2019/2020

Marisa Esteves

8 de Novembro de 2019

Universidade do Minho

Plano de Aula

1. Finalização da resolução da 4.ª ficha prática laboratorial pelos alunos em grupo;
2. Correção da ficha com os alunos.

Processo ETL

Definição

O processo ETL (*Extract, Transform, Load*) é um conjunto de processos que inclui a extração de dados de fontes de informação internas e externas, podendo estar em diferentes formatos, a transformação dos dados de acordo com as necessidades da organização e, finalmente, o carregamento dos mesmos numa estrutura de dados, como por exemplo um data mart ou um data warehouse.

Processo ETL

Definição

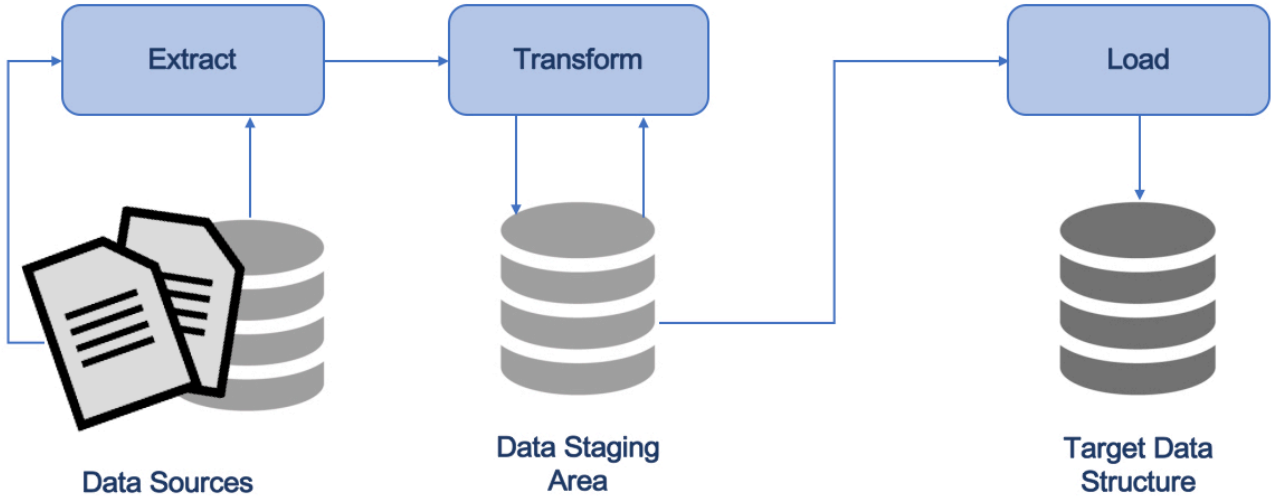

Figura 1 – Esquema do processo ETL.

Processo ETL

Porquê?

Os dados estão espalhados por diferentes localizações

Os dados estão armazenados em diferentes tipos de formato

O volume de dados continua a aumentar

Os dados podem estar estruturados, semi-estruturados ou não estruturados

Data Warehousing

Definição

O processo de data warehousing enfatiza à recolha de dados de diversas fontes através do processo ETL (*Extract, Transform, Load*), correspondendo à construção de data warehouses e/ou data marts, para aceder e analisar a informação de forma útil. Os dados extraídos são processados, formatados e consolidados numa estrutura de dados única para facilitar essencialmente a análise de dados.

Data Warehousing

Definição

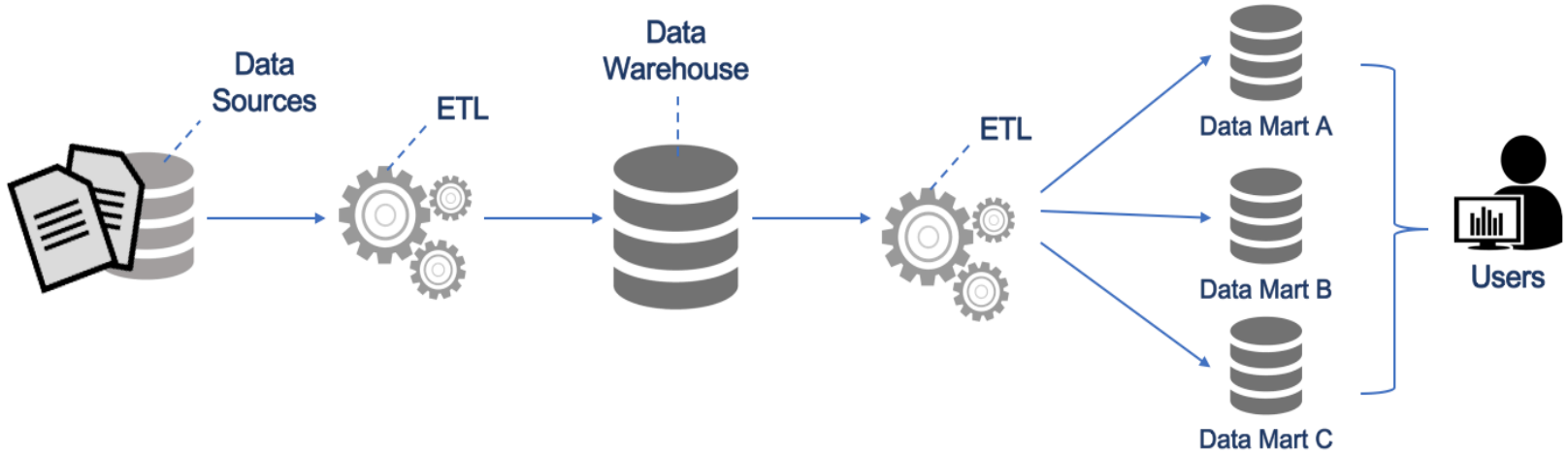

Figura 2 – Esquema do processo de data warehousing.

Data Warehousing

Data Warehouse vs. Data Mart

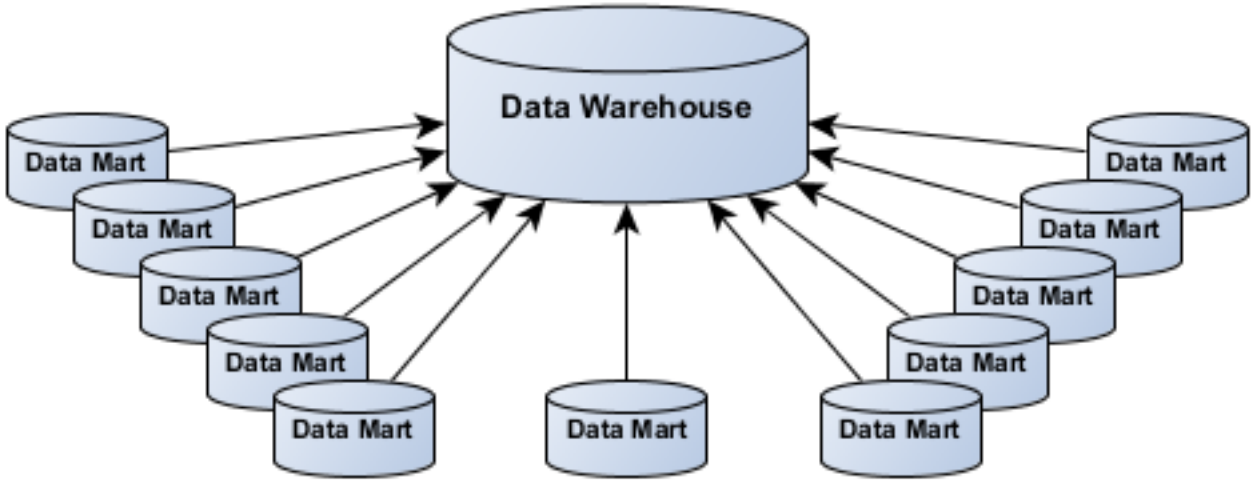

Figura 3 – Data warehouse vs. Data marts.

Data Warehousing

Modelo Dimensional – Esquema em Estrela vs. Esquema em Floco de Neve

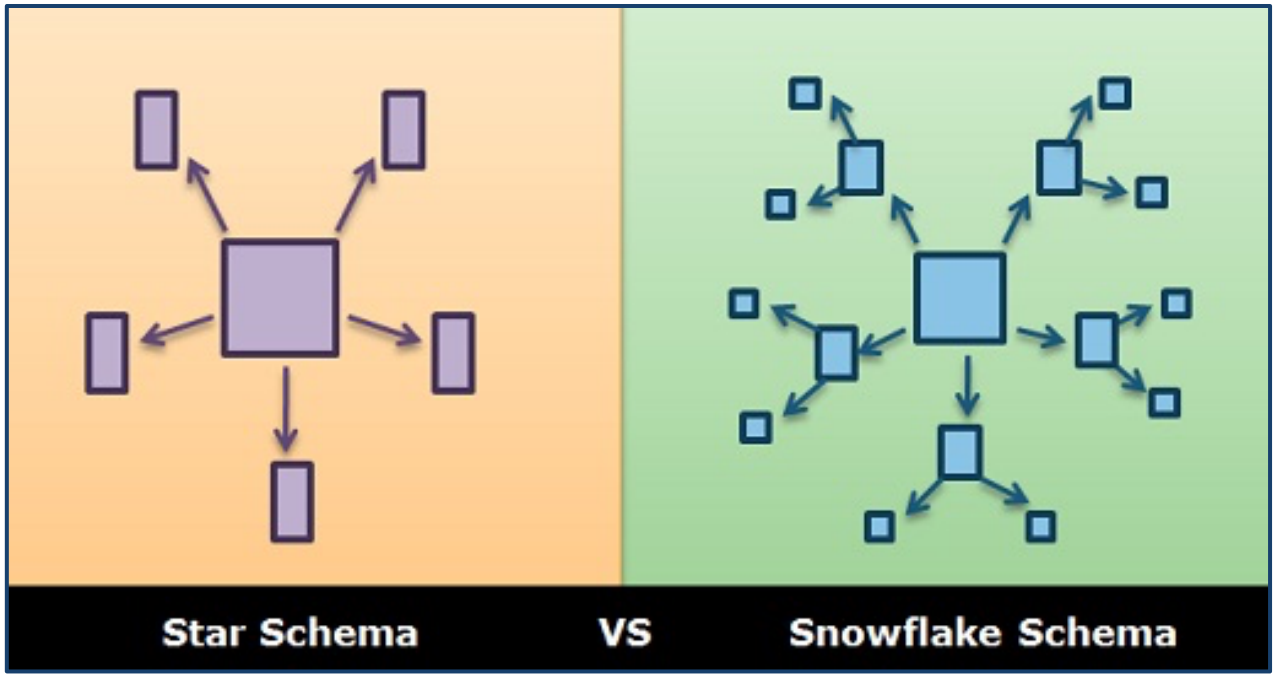

Figura 4 – Esquema em Estrela vs. Esquema em Floco de Neve.

Data Warehousing

Modelo Dimensional – Esquema em Constelação de Factos

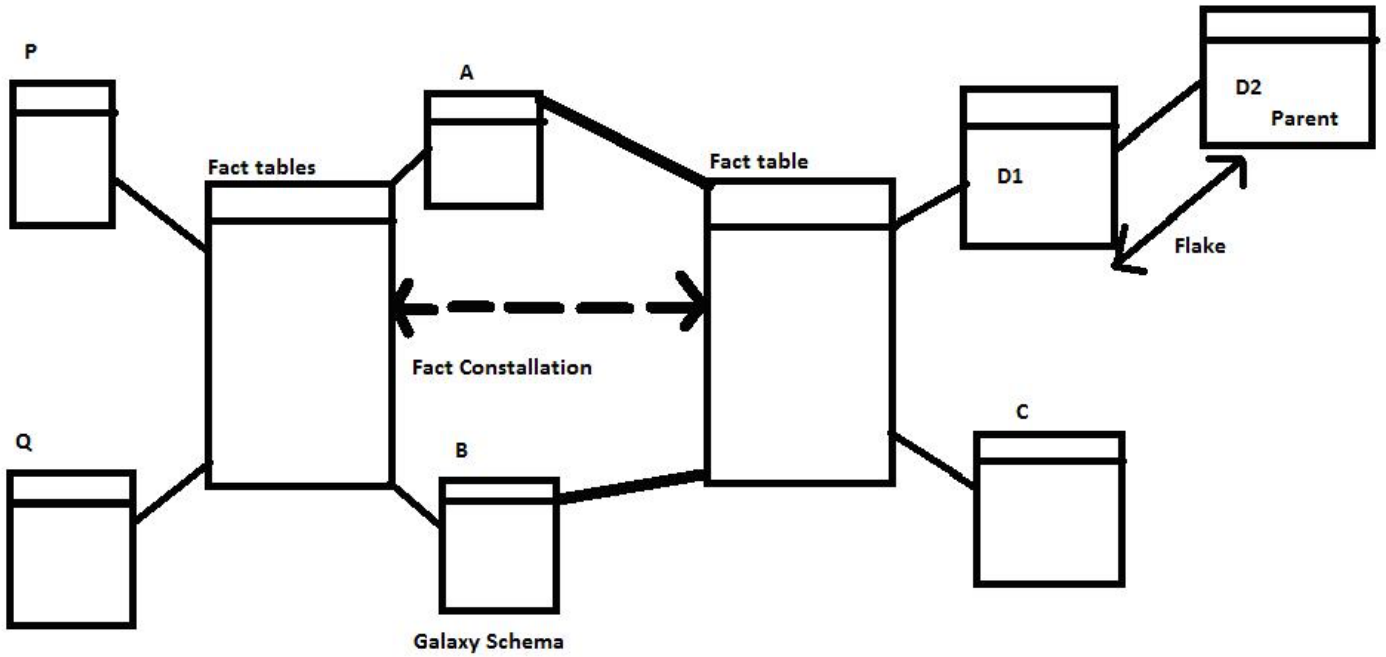

Figura 5 – Esquema em Constelação de Factos.

OLTP vs. OLAP

Definição

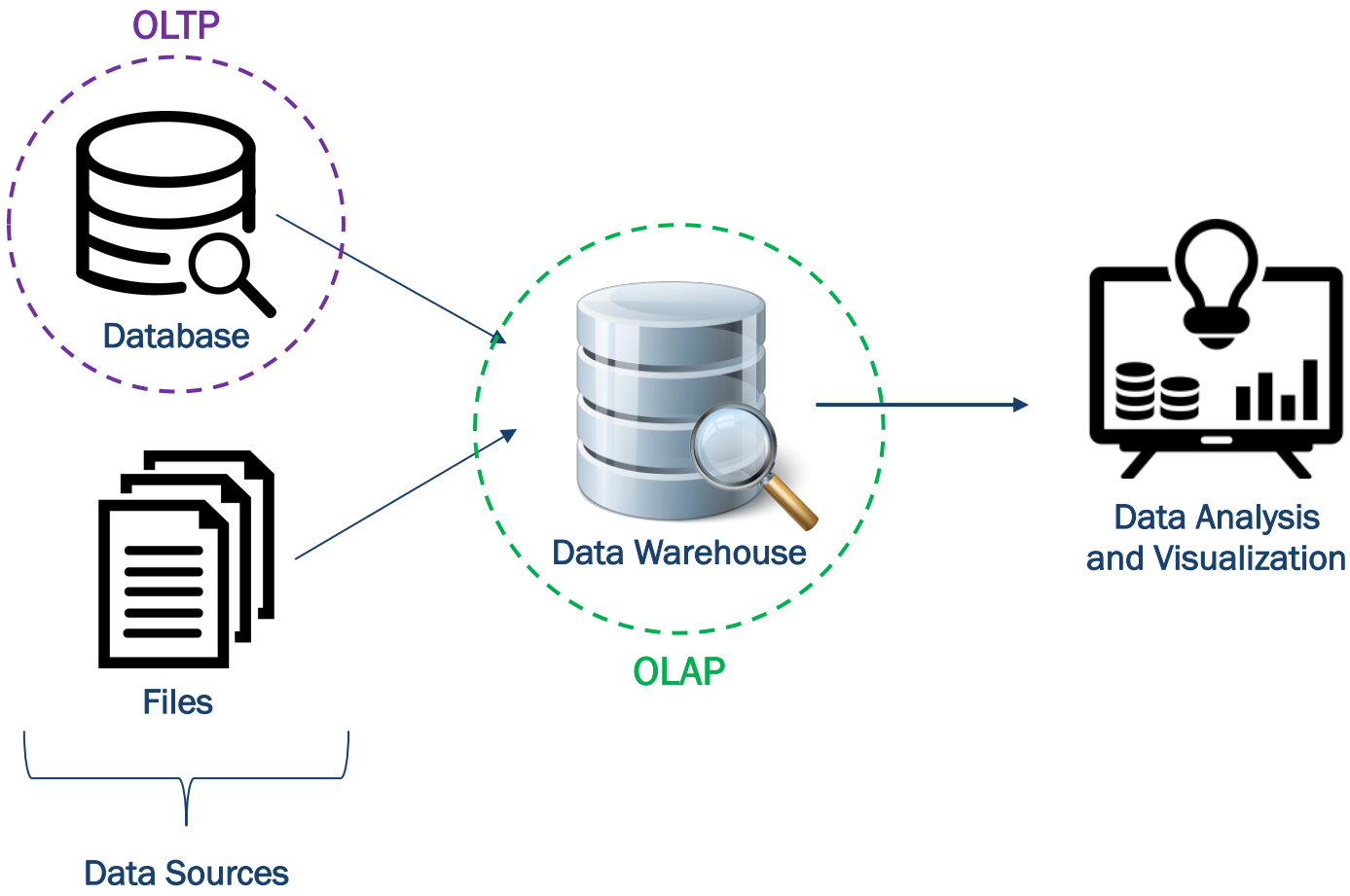

Figura 6 – OLTP (*Online Transaction Processing*) vs. OLAP (*Online Analytical Processing*).

OLTP vs. OLAP

Definição

Relational Database (OLTP)	Analytical Data Warehouse (OLAP)
Contains current data	Contains historical data
Useful in running the business	Useful in analysing the business
Based on Entity Relationship Model	Based on Star, Snowflake or Fact Constellation Schema
Provides primitive and highly detailed data	Provides summarized and consolidated data
Used for writing into the database	Used for reading data from the data warehouse
Database size ranges from 100 MB to 1 GB	Data warehouse ranges from 100 GB to 1 TB
Fast and it provides high performance	Highly flexible but it is not fast
Number of records accessed is in tens	Number of records accessed is in millions
Example: all bank transactions made by a customer	Example: bank transactions made by a customer at a particular time

Figura 7 – Diferenças entre OLTP e OLAP.

MySQL

INSERT INTO SELECT FROM

Permite copiar dados de uma tabela e os inserir noutra tabela.
No entanto, este comando SQL requer que os tipos de dados na tabela de origem (table1) e na tabela destino (table2) sejam iguais.

- **INSERT INTO** *table2* (*column1*, *column2*, *column3*, ...)
  SELECT *column1*, *column2*, *column3*, ...
  FROM *table1*
  WHERE *condition*

MySQL

Cursores

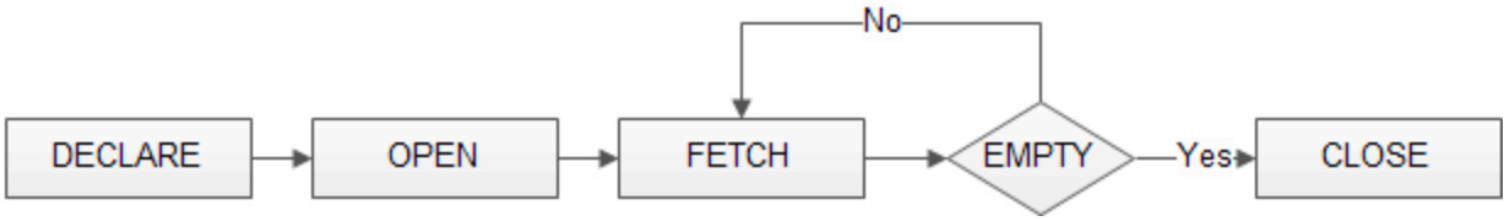

Figura 8 – Modo de funcionamento de cursores em MySQL.

MySQL

Cursores

```
1  CREATE PROCEDURE curdemo()
2  BEGIN
3    DECLARE done INT DEFAULT FALSE;
4    DECLARE a CHAR(16);
5    DECLARE b, c INT;
6    DECLARE cur1 CURSOR FOR SELECT id,data FROM test.t1;
7    DECLARE cur2 CURSOR FOR SELECT i FROM test.t2;
8    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
9
10   OPEN cur1;
11   OPEN cur2;
12
13   read_loop: LOOP
14     FETCH cur1 INTO a, b;
15     FETCH cur2 INTO c;
16     IF done THEN
17       LEAVE read_loop;
18     END IF;
19     IF b < c THEN
20       INSERT INTO test.t3 VALUES (a,b);
21     ELSE
22       INSERT INTO test.t3 VALUES (a,c);
23     END IF;
24   END LOOP;
25
26   CLOSE cur1;
27   CLOSE cur2;
28 END;
```

Figura 9 – Exemplo de um procedimento com cursores em MySQL.

Resolução da 4.ª Ficha Prática Laboratorial

1 Modelação Dimensional em Constelação de Factos

O principal objetivo da resolução da primeira parte deste exercício é analisar a base de dados *sakila*, bem como o ficheiro calendario.xlsx, disponibilizados durante as aulas práticas laboratoriais desta unidade curricular, escolher a informação de interesse para futura análise de dados e, consequentemente, definir um modelo dimensional no formato de constelação de factos.

É de relembrar que um modelo dimensional baseado no formato em constelação de factos é constituído por duas ou mais tabelas de factos que podem partilhar entre elas tabelas de dimensão.

Numa segunda parte, procederá ao povoamento do *data warehouse* definido e implementado, bem como à gestão dos seus processos.

É de notar que pode consultar mais informação de apoio sobre a base de dados *sakila* disponibilizada na seguinte referência: https://dev.mysql.com/doc/sakila/en/.

Resolução da 4.ª Ficha Prática Laboratorial

Com base no caso apresentado, pretende-se que:

1. Implemente a base de dados *sakila* no MySQL Workbench com o ficheiro sakila-schema.sql.
2. Povoe as tabelas da base de dados criada no passo anterior com o ficheiro sakila-data.sql.
3. Defina um modelo dimensional em constelação de factos a partir da base de dados *sakila* (ver o ficheiro sakila.mwb) – *EER Diagram*. No entanto, deverá ter em consideração os seguintes três pontos:
   (a) Deverá definir duas tabelas de factos: FACTS_PAYMENT (para os pagamentos) e FACTS_RENTAL (para os alugueres);
   (b) Deverá definir obrigatoriamente uma tabela de dimensão para o tempo denominada "DIM_TIME". Tenha em atenção à granularidade que definirá para esta tabela de dimensão, uma vez que terá de guardar na mesma pelo menos um identificador único para a data (id), valor, dia, mês, ano, dia da semana, semana do ano, se é dia útil ou não (ver ficheiro calendario.xlsx), se é feriado ou não (ver ficheiro calendario.xlsx), entre outros;
   (c) Adicione uma coluna denominada "etiqueta_DTA" em cada tabela do modelo dimensional de modo a guardar o *datetime* de povoamento de cada linha.
4. Converta o modelo dimensional definido para o respetivo modelo físico numa base de dados denominada data_warehouse (Database > Forward Engineer).
5. Guarde num ficheiro .sql a *script* de criação das tabelas do modelo dimensional definido.