



MCI



MAGIC CODE INSTITUTE

BUỔI 9

**TỔNG KẾT KHÓA HỌC-GIẢI ĐÁP THẮC MẮC
BÀI TẬP THỰC HÀNH**





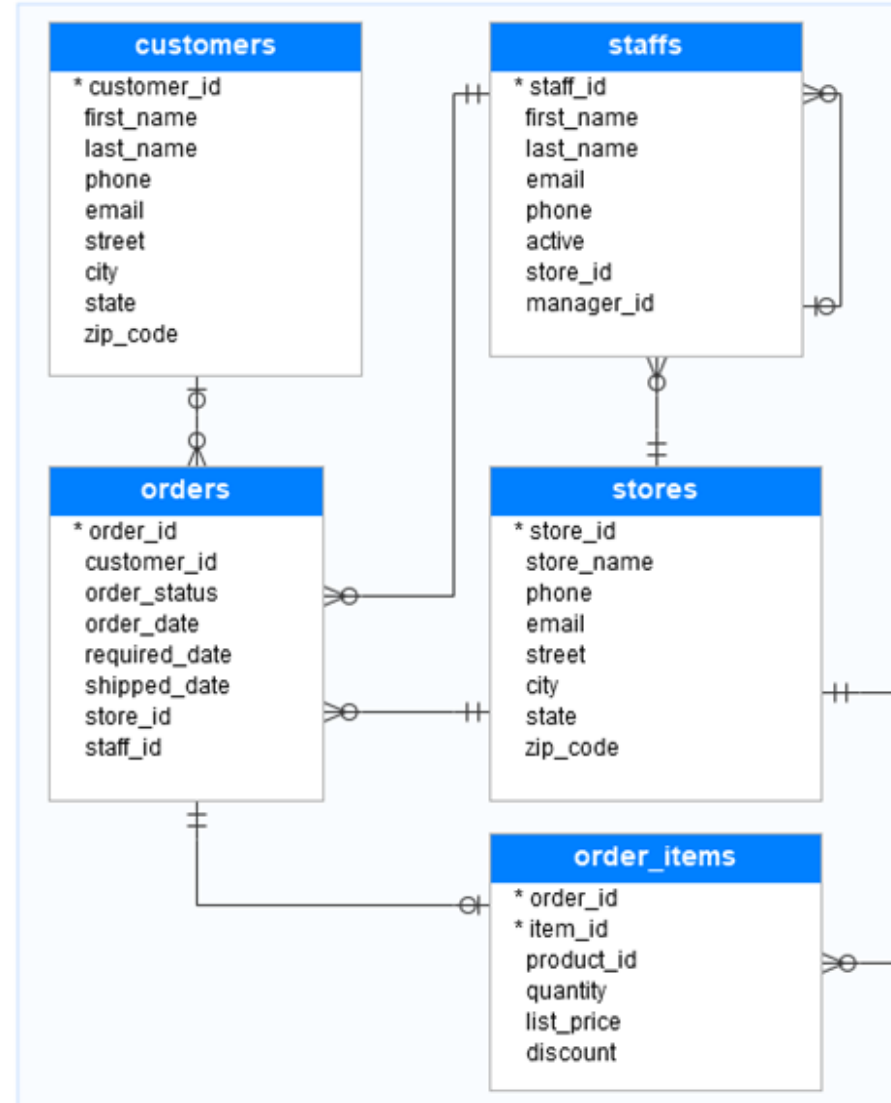
MCI



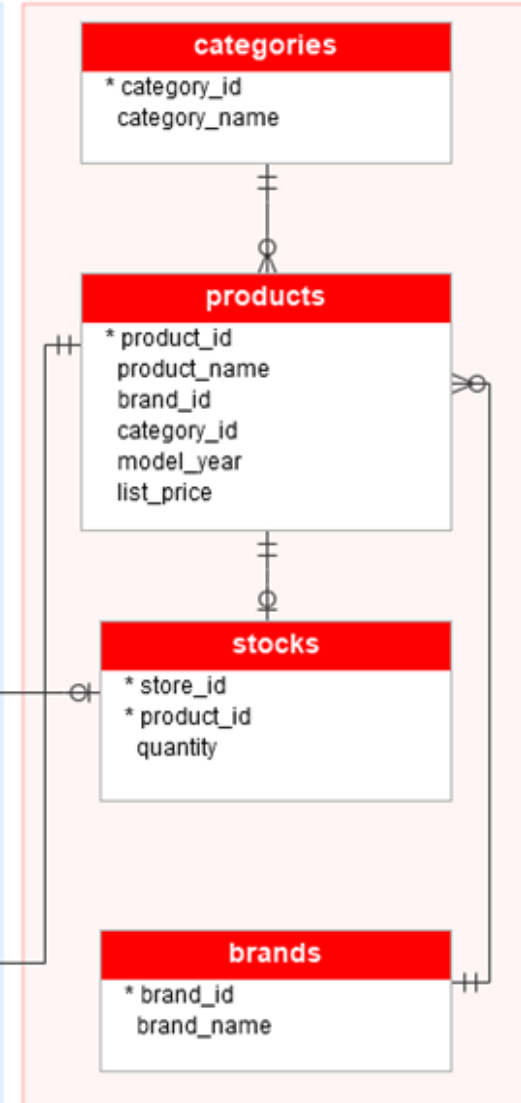
MAGIC CODE INSTITUTE

Database Diagrams-bikestore

Sales



Production



GIỚI THIỆU VỀ DATA ANALYTICS

CRISP – DM - Cross Industry Standard Process for Data Mining, là “quy trình tiêu chuẩn cho khai thác dữ liệu trong nhiều lĩnh vực”. Quy trình này được hình thành từ năm 1996, nhưng cho đến nay, là một trong những mô hình tiêu chuẩn được sử dụng rộng rãi nhất trong khai phá và phân tích dữ liệu. Nó giống như một bộ lan can để giúp chúng ta lên kế hoạch, tổ chức và triển khai các dự án khai phá dữ liệu của mình.

Quy trình gồm 6 bước:

Bước 1: Business understanding – What does the business need?

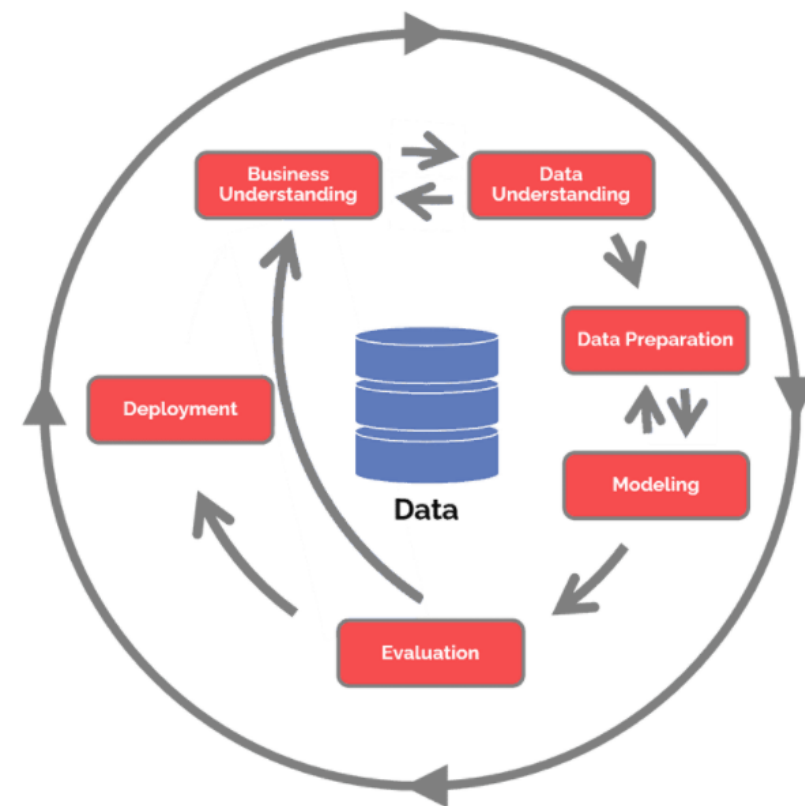
Bước 2: Data understanding – What data do we have / need? Is it clean?

Bước 3: Data preparation – How do we organize the data for modeling?

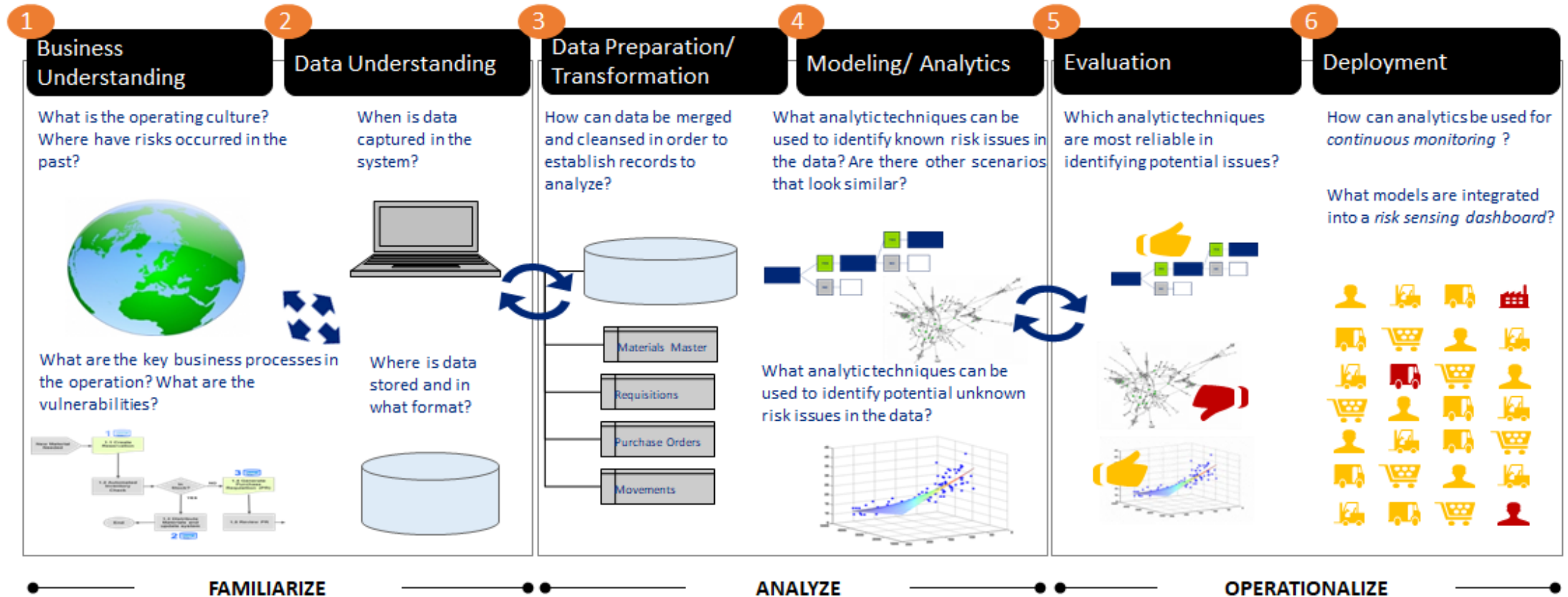
Bước 4: Modeling – What modeling techniques should we apply?

Bước 5: Evaluation – Which model best meets the business objectives?

Bước 6: Deployment – How do stakeholders access the results?



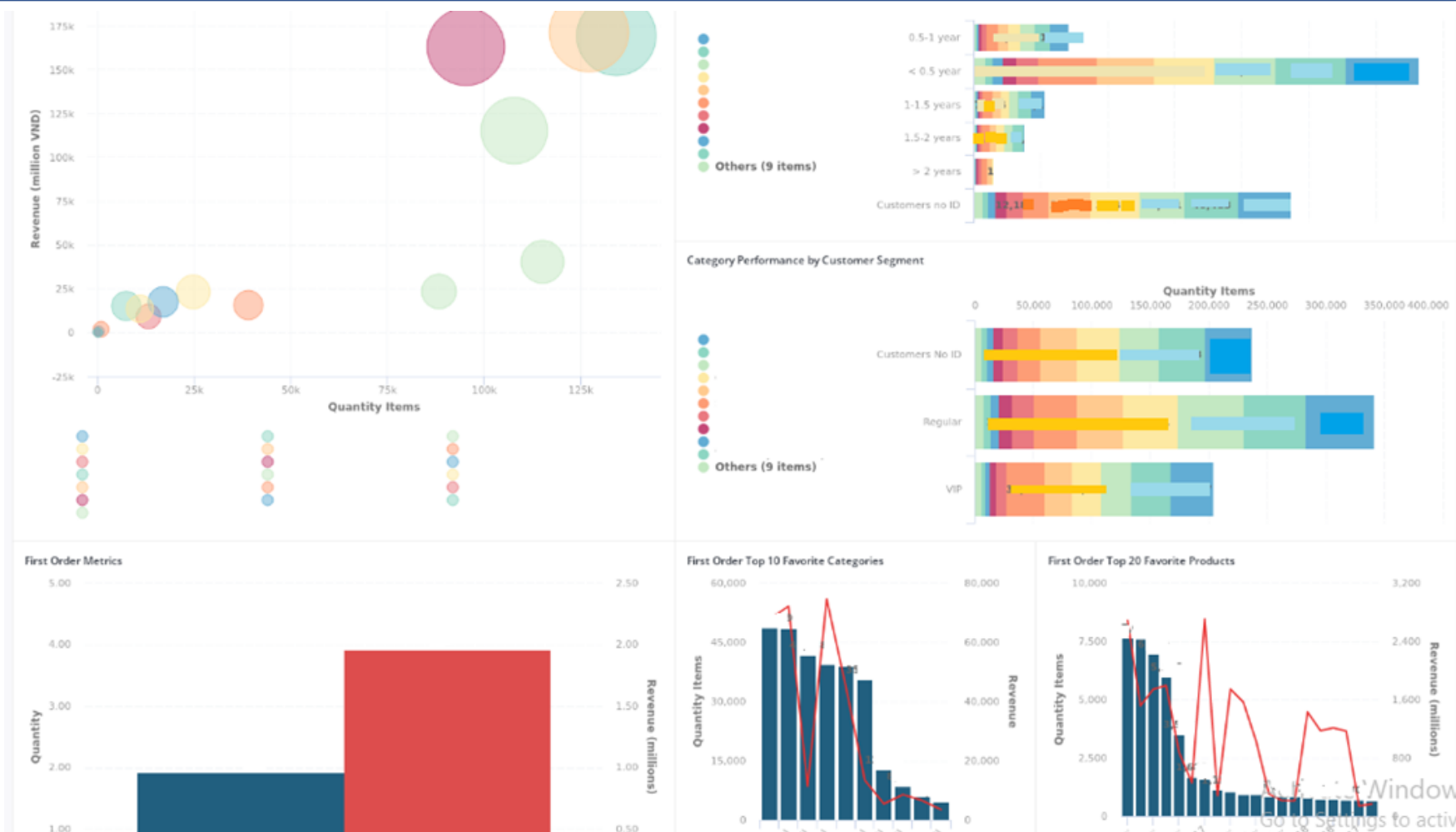
Cross Industry Standard Process for Data Mining/ Analytics (CRISP-DM)



Review: Visualization Tools & SQL

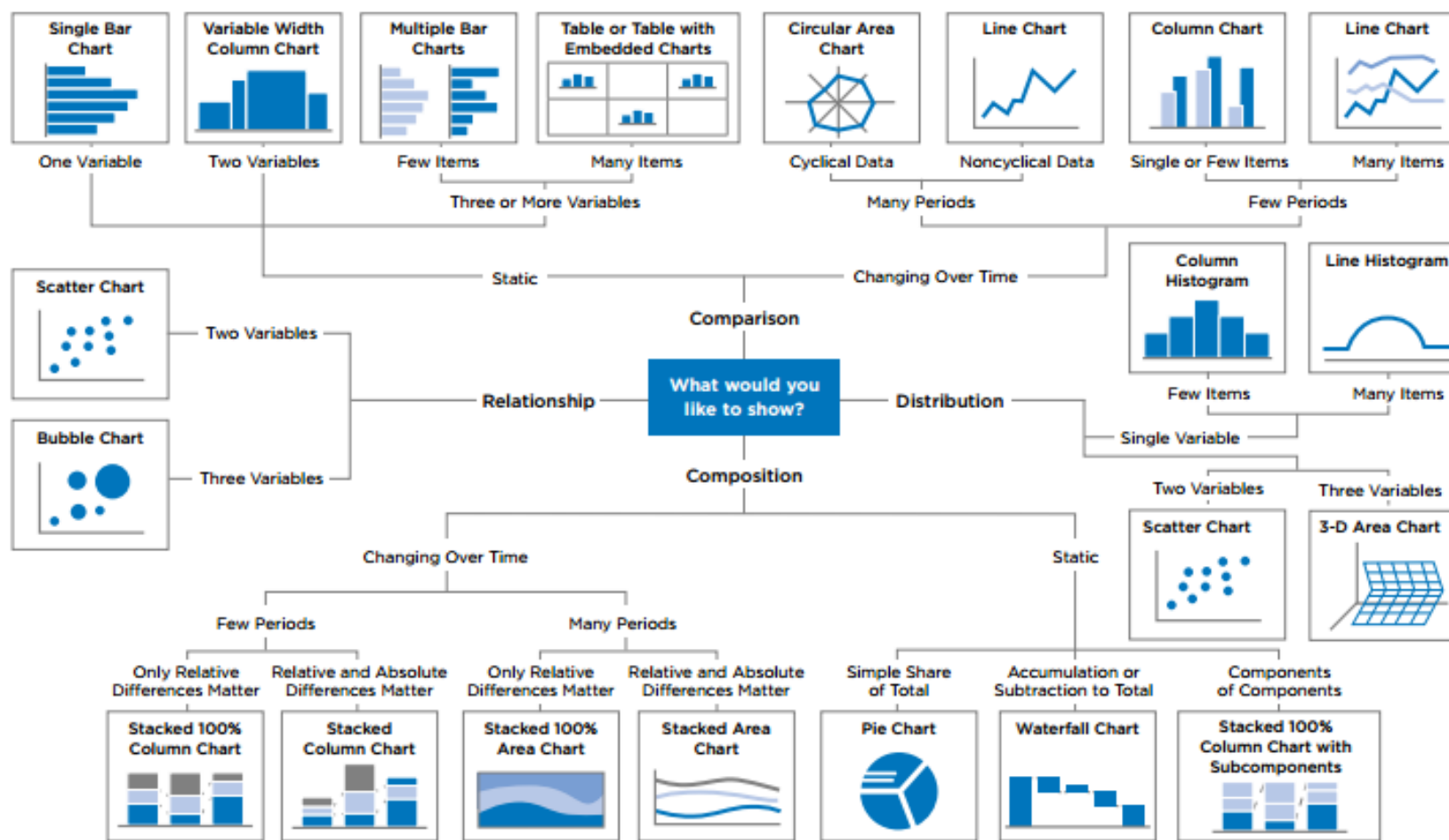
Power BI
Tableau
Google Data Studio
Qlik
Holistics
Metabase

...



HOW TO PICK THE RIGHT CHART TYPE?

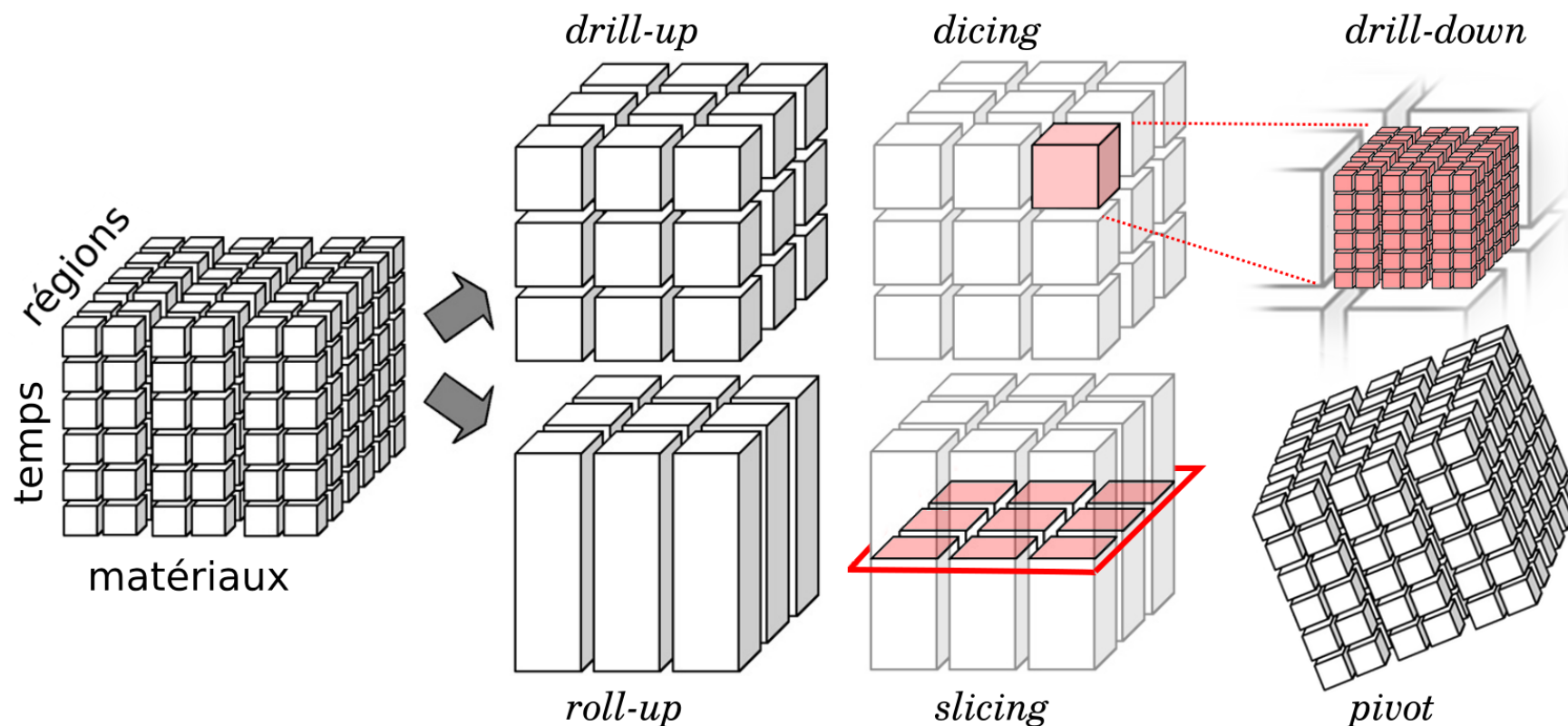
SELECTING THE APPROPRIATE CHART FOR STRATEGY PRESENTATIONS



MÔ HÌNH DỮ LIỆU ĐA CHIỀU

Multi-dimensional model

Dữ liệu trong kho dữ liệu được thể hiện dưới dạng đa chiều (Multi Dimension) gọi là khối (cube). Mỗi chiều mô tả một đặc trưng nào đó của dữ liệu. (Nếu số chiều dữ liệu lớn hơn 3, gọi là **Hyper Cube**)

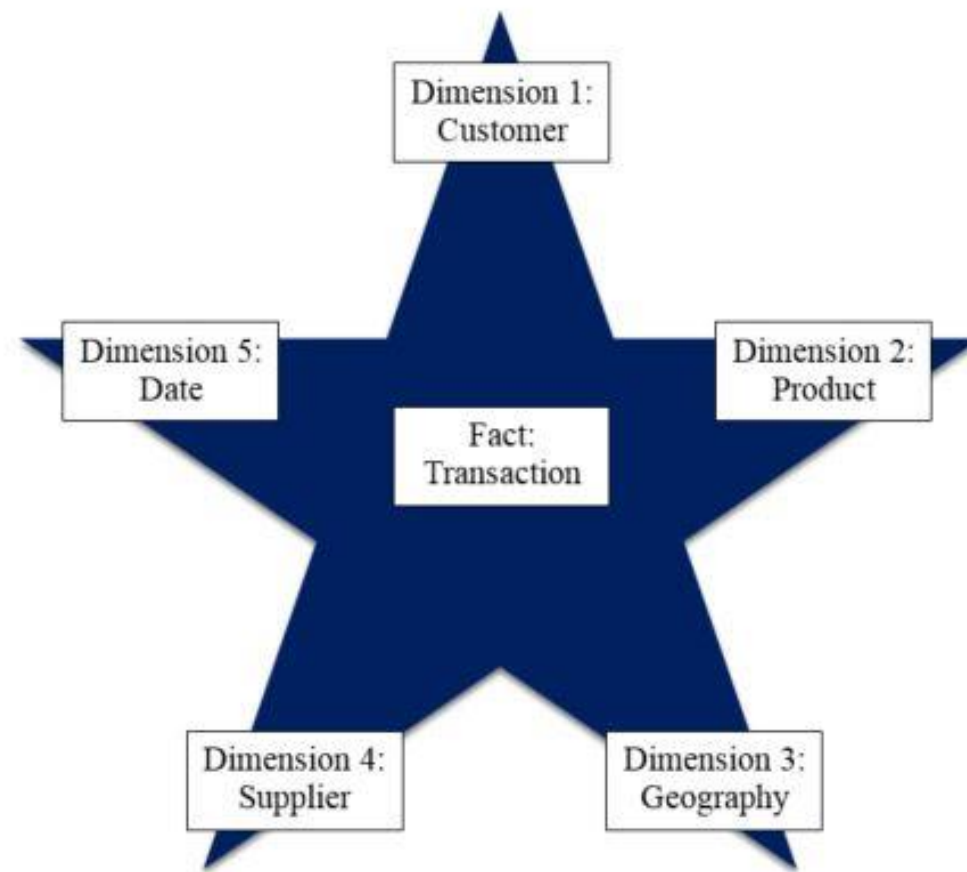


MÔ HÌNH NGÔI SAO

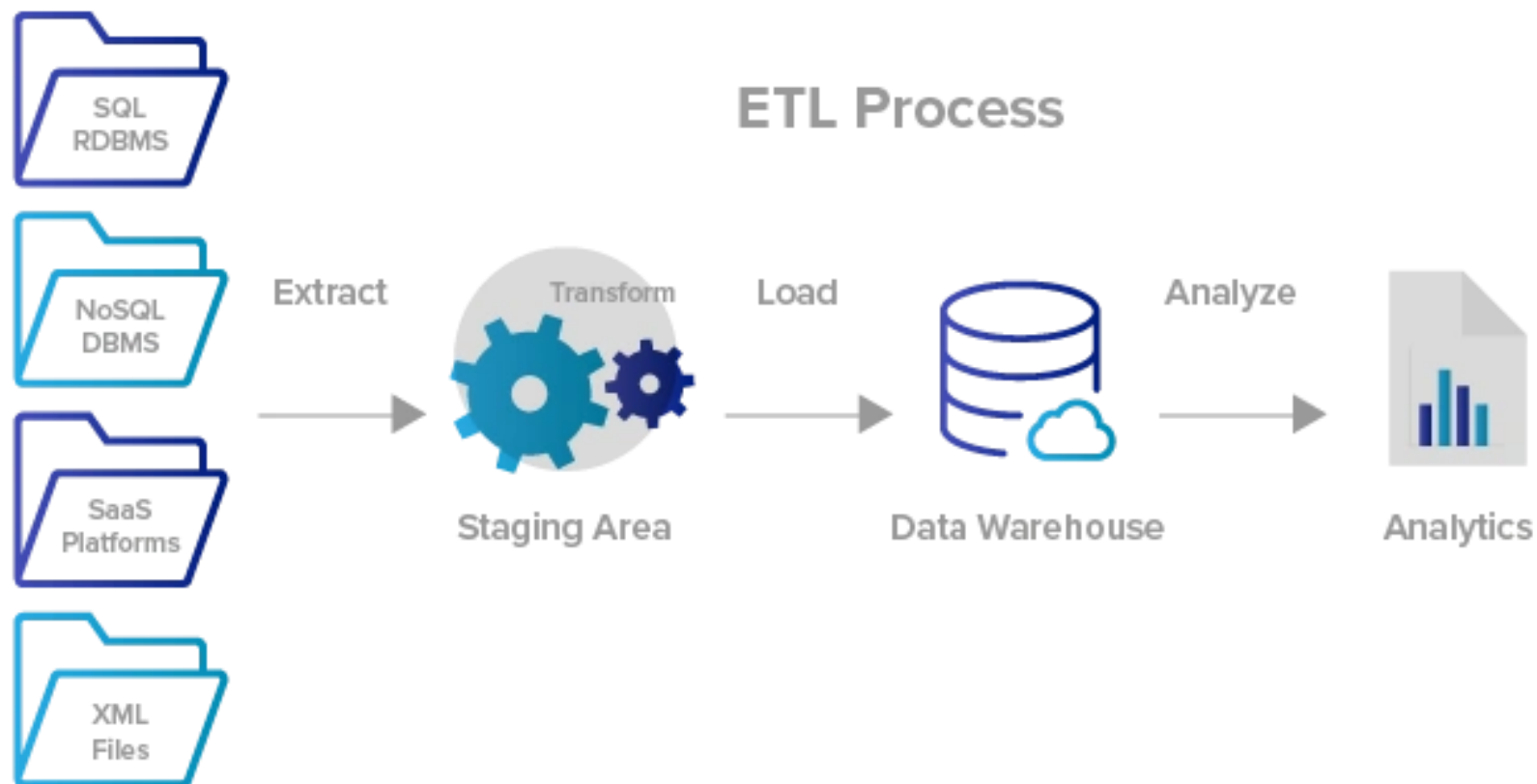
Star Schema

Mô hình ngôi sao (**star schema**) là một cách tiếp cận để tổ chức **data warehouse**. Nó rất đơn giản và thường dùng trong data mart.

Data Mart là một dạng thu nhỏ, là tập hợp con của **Data Warehouse** (Kho dữ liệu) trong đó tập trung vào một đơn vị kinh doanh cụ thể.



ETL & Data Warehousing



MCI YouTube Channel

<https://www.youtube.com/channel/UCB5K6Vzq1VL8bwIk-LVqbiw/videos>

The screenshot displays the MCI YouTube Channel page with the following video thumbnails:

Thumbnail	Title	Views	Upload Time
	Hành trình tự học SQL cùng MCI - Buổi 4 - Tạo databas...	7 views	3 days ago
	Hành trình tự học VBA cùng MCI - Buổi 2 - Sử dụng...	9 views	4 days ago
	Hành trình tự học Python - Buổi 3 - Cài đặt Python trê...	25 views	6 days ago
	Hướng dẫn Import, export dữ liệu trong SQL Server - ...	57 views	1 week ago
	Hướng dẫn cài đặt các thư viện (pandas, numpy,...	27 views	1 week ago
	[QC] Giới thiệu khoá học PYTHON TRACK	1.7K views	1 week ago
	Hướng dẫn cài đặt Power BI cùng MCI - Power BI Cơ Bả...	39 views	1 week ago
	Hướng dẫn cài đặt MS SQL Server và SQL Server...	44 views	1 week ago
	Hướng dẫn cài đặt Python cùng MCI - Python Cơ Bản ...	186 views	1 week ago
	Hướng dẫn cài đặt VBA cùng MCI - VBA Cơ Bản - ...	148 views	1 week ago
	Hướng dẫn cài đặt MySQL cùng MCI - SQL Cơ Bản - ...		
	08 BLOG DẪN KHOA HỌC DỮ LIỆU KHÔNG THỂ BỎ...		
	MCI - TOP1 đào tạo về Python, SQL, R, Power BI, I...		
	Khóa học R cho người mới bắt đầu - buổi 1		
	Khóa học SQL Chuyên nghiệp cho người mới bắt...		

QUERYING DATA FROM A TABLE	QUERYING FROM MULTIPLE TABLES	USING SQL OPERATORS
<p>SELECT c1, c2 FROM t; Query data in columns c1, c2 from a table</p> <p>SELECT * FROM t; Query all rows and columns from a table</p> <p>SELECT c1, c2 FROM t WHERE condition; Query data and filter rows with a condition</p> <p>SELECT DISTINCT c1 FROM t WHERE condition; Query distinct rows from a table</p> <p>SELECT c1, aggregate(c2) FROM t GROUP BY c1; Group rows using an aggregate function</p> <p>SELECT c1, aggregate(c2) FROM t GROUP BY c1 HAVING condition; Filter groups using HAVING clause</p> <p>SQL AGGREGATE FUNCTIONS</p> <p>AVG returns the average of a list</p> <p>COUNT returns the number of elements of a list</p> <p>SUM returns the total of a list</p> <p>MAX returns the maximum value in a list</p> <p>MIN returns the minimum value in a list</p>	<p>SELECT c1, c2 FROM t1 INNER JOIN t2 ON condition; Inner join t1 and t2</p> <p>SELECT c1, c2 FROM t1 LEFT JOIN t2 ON condition; Left join t1 and t2</p> <p>SELECT c1, c2 FROM t1 RIGHT JOIN t2 ON condition; Right join t1 and t2</p> <p>SELECT c1, c2 FROM t1 FULL OUTER JOIN t2 ON condition; Perform full outer join</p> <p>SELECT c1, c2 FROM t1 CROSS JOIN t2; Produce a Cartesian product of rows in tables</p> <p>SELECT c1, c2 FROM t1 A SEFT JOIN t2 B ON condition; Join t1 to itself using SEFT JOIN clause</p>	<p>SELECT c1, c2 FROM t1 UNION [ALL] SELECT c1, c2 FROM t2; Combine rows from two queries</p> <p>SELECT c1, c2 FROM t1 WHERE c1 [NOT] LIKE pattern; Query rows using pattern matching %, _</p> <p>SELECT c1, c2 FROM t WHERE c1 [NOT] IN value_list; Query rows in a list</p> <p>SELECT c1, c2 FROM t WHERE c1 BETWEEN low AND high; Query rows between two values</p> <p>SELECT c1, c2 FROM t WHERE c1 IS [NOT] NULL; Check if values in a table is NULL or not</p>

MANAGING TABLES	USING SQL CONSTRAINTS	MODIFYING DATA
<p>CREATE TABLE t (id INT PRIMARY KEY, name VARCHAR NOT NULL, price INT DEFAULT 0); Create a new table with three columns</p> <p>DROP TABLE t ; Delete the table from the database</p> <p>ALTER TABLE t ADD column; Add a new column to the table</p> <p>ALTER TABLE t DROP COLUMN c ; Drop column c from the table</p> <p>ALTER TABLE t1 RENAME TO t2; Rename column t1 to t2</p> <p>ALTER TABLE t1 RENAME c1 TO c2 ; Rename column c1 to c2</p> <p>TRUNCATE TABLE t; Remove all data in a table</p>	<p>CREATE TABLE t (c1 INT, c2 INT, c3 VARCHAR, PRIMARY KEY (c1,c2)); Set c1 and c2 as a primary key</p> <p>CREATE TABLE t1 (c1 INT PRIMARY KEY, c2 INT, FOREIGN KEY (c2) REFERENCES t2(c2)) ; Set c2 column as a foreign key</p> <p>CREATE TABLE t (c1 INT PRIMARY KEY, c2 VARCHAR NOT NULL) ; Set values in c2 column not NULL</p>	<p>INSERT INTO t (column_list) VALUES (value_list); Insert one row into a table</p> <p>INSERT INTO t (column_list) VALUES (value_list), (value_list),; Insert multiple rows into a table</p> <p>INSERT INTO t1 (column_list) SELECT column_list FROM t2; Insert rows from t2 into t1</p> <p>UPDATE t SET c1 = new_value; Update new value in the column c1 for all rows</p> <p>UPDATE t SET c1 = new_value, c2 = new_value WHERE condition; Update values in the column c1, c2 that match the condition</p> <p>DELETE FROM t; Delete all data in a table</p> <p>DELETE FROM t WHERE condition; Delete subset of rows in a table</p>

BÀI 1

Tạo Database (cơ sở dữ liệu) tên QuanLyBanHang bao gồm các tablesau:

VATTU

Tên cột	Kiểu dữ liệu	Ghi chú
MaVTu	Char (4)	Khoá chính
TenVTu	NVarChar (100)	
DvTinh	NVarChar (10)	
PhanTram	Int	

NHACC

Tên cột	Kiểu dữ liệu	Ghi chú
MaNhaCC	Char (3)	Khoá chính
TenNhaCC	NVarChar (100)	
DiaChi	NVarchar (200)	
DienThoai	NVarchar (20)	

DONDH

Tên cột	Kiểu dữ liệu	Ghi chú
SoDH	Char (4)	Khoá chính
NgayDH	DateTime	
MaNhaCC	Char (4)	Khoá ngoại

BÀI 2

Từ bảng Orders:

1. Có bao nhiêu đơn hàng có mức độ ưu tiên (order_priority) = 'Low'
2. Lấy danh sách top 5 tỉnh có nhiều đơn hàng nhất
3. Lấy danh sách 3 nhà quản lý có nhiều đơn hàng nhất
4. Hãy đưa ra danh sách các Loại sản phẩm (Product_category) có lợi nhuận trung bình lớn hơn lợi nhuận trung bình của tất cả các đơn hàng
5. Hãy đưa ra danh sách top 5 sản phẩm (product_name) có total_value lớn nhất của mỗi khu vực (region). Gợi ý sử dụng: row_number()
6. Hãy đưa ra danh sách các sản phẩm (product_name) có total_value bị xếp hạng xếp hạng thấp nhất theo từng tỉnh (province). Gợi ý sử dụng dense_rank()
7. Hãy tạo bảng gồm các cột: year, month, total_revenue, total_revenue_returned, acc_revenue, group_revenue

Trong đó:

Revenue = order_quantity* unit_price*(1- discount)

acc_revenue = total_revenue - total_revenue_returned

Group_revenue là 'Thấp' khi acc_revenue < 10000

Group_revenue là 'Trung bình' khi acc_revenue < 20000

Group_revenue là 'Cao' khi acc_revenue >=20000

CHÚC CÁC BẠN HOÀN
THÀNH TỐT BÀI KIỂM TRA

