

clusterAI 2020
ciencia de datos en ingeniería industrial
UTN BA
curso I5521

clase_12_DB



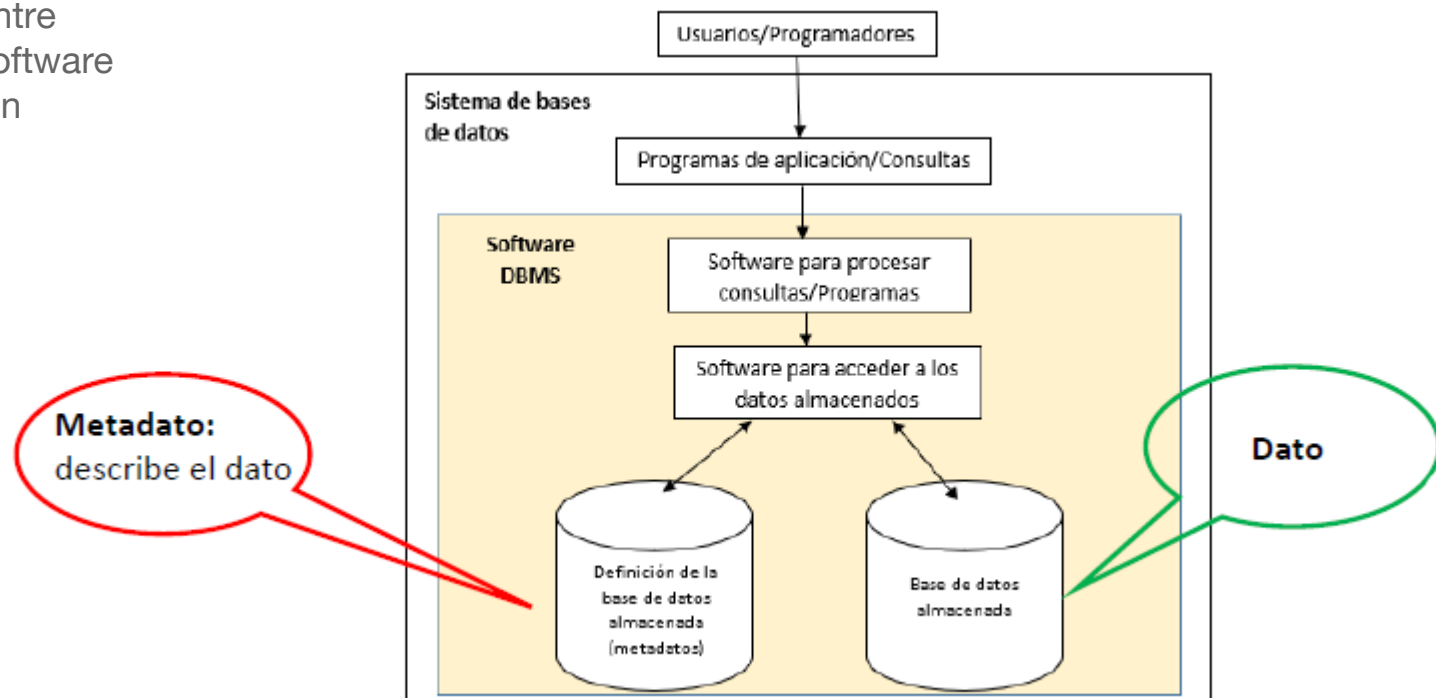
clusterAI

agenda_clase12 Base de Datos

- Arquitectura de Base de Datos (DB)
- ETL
- Cubo de datos
- Esquema relacional de datos
- Comandos básicos SQL
- SQL vs No-SQL
- Ejercicio Integración Python y SQL
- Cloud Computing

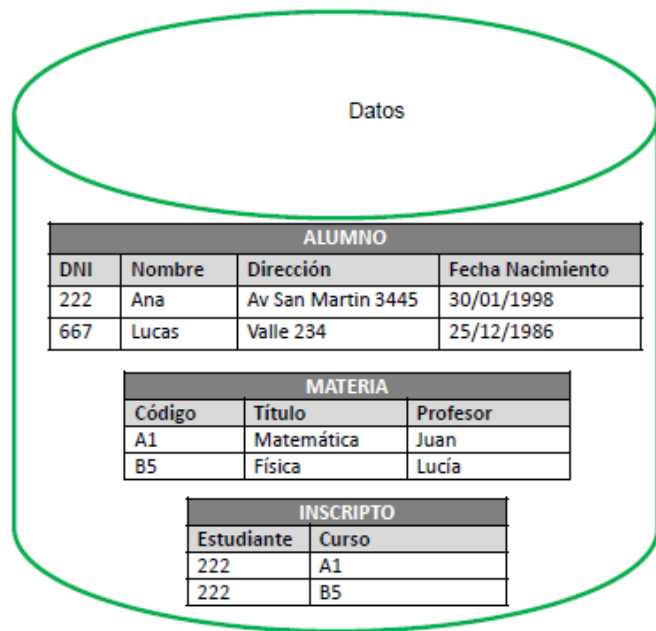
Sistema de Base de Datos

Es la combinación entre base de datos y el software de consulta y gestión



Data y Metadata

Descripción



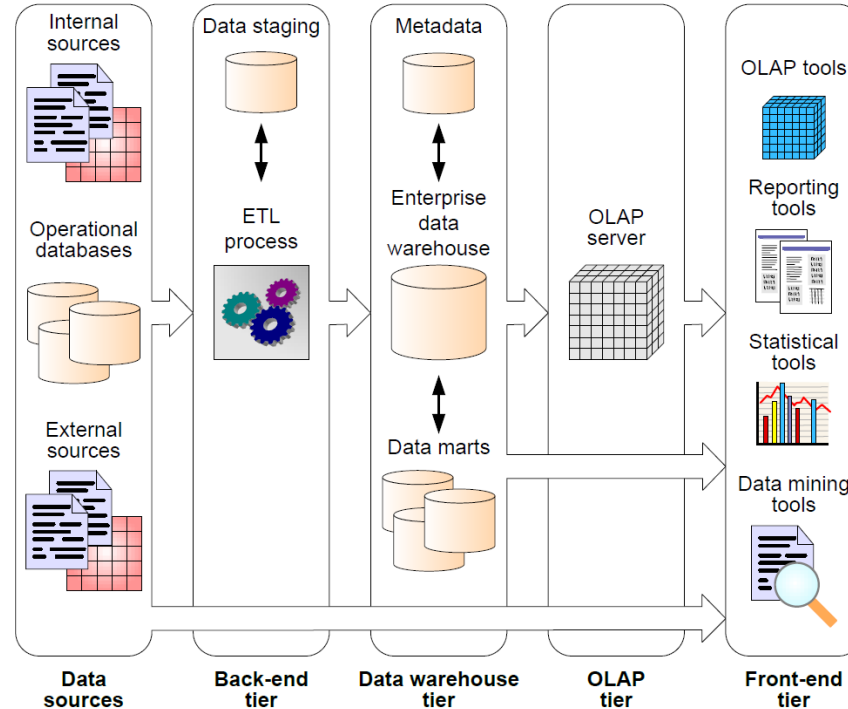
Catálogo (Metadatos)

RELACIONES	
Nombre	CantColumnas
ALUMNO	4
MATERIA	3
INSCRIPTO	2

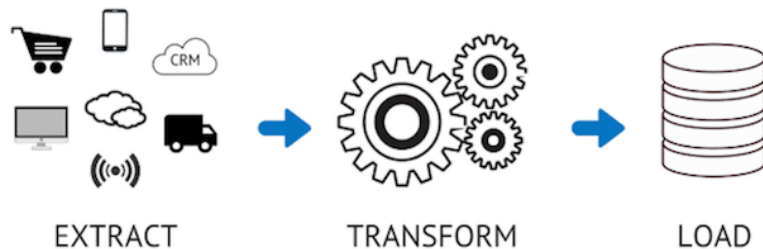
COLUMNAS		
NombreColumna	TipoDato	PerteneceARelación
DNI	Númerico	Alumno
Título	Carácter(100)	Materia
Profesor	Carácter(50)	Materia
...		

Arquitectura de Data Warehouse

Typical Data Warehouse Architecture



Extract, Transform and Load (ETL)



Extract:

- Conectarse a base de datos externas
- Generalmente se transfiere a un esquema “raw” o a un data lake

Transform:

- Normalización de datos
- Eliminación de duplicados
- Cambios de formato
- Filtrados
- Agregaciones
- Estructuración y clasificación
- Construcción de un pipeline de transformación de datos

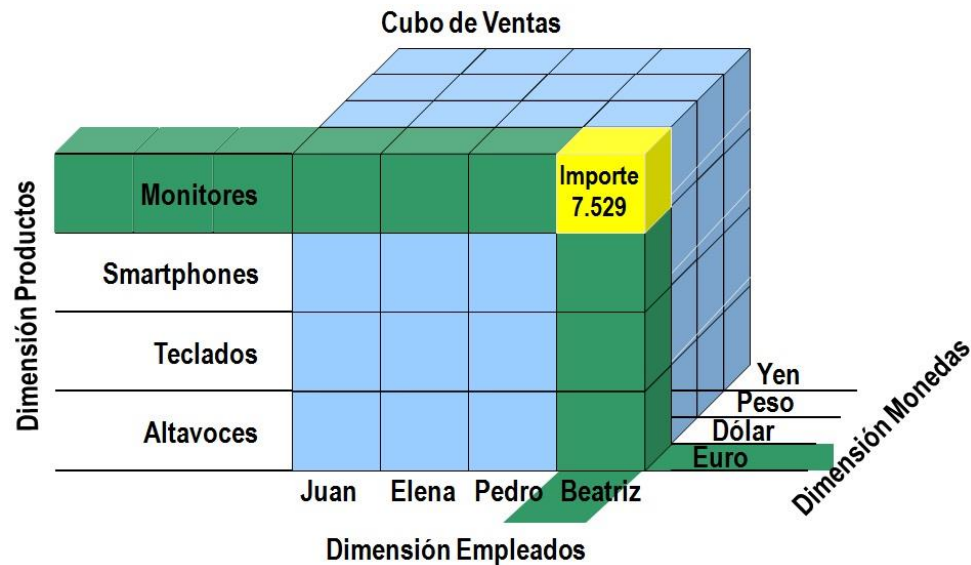
Load:

- Carga de los datos limpios a los lugares de destinos listos para ser procesados por los algoritmos de ML o procesos de BI

Cubo de datos - OLAP

Objetivos:

- Obtención de un resultado numérico (importes de ventas, gastos, cantidad de productos vendidos, etc.).
- Minimizar el tiempo de respuesta a la consulta
- La idea es que sea información lista para consumir en un análisis



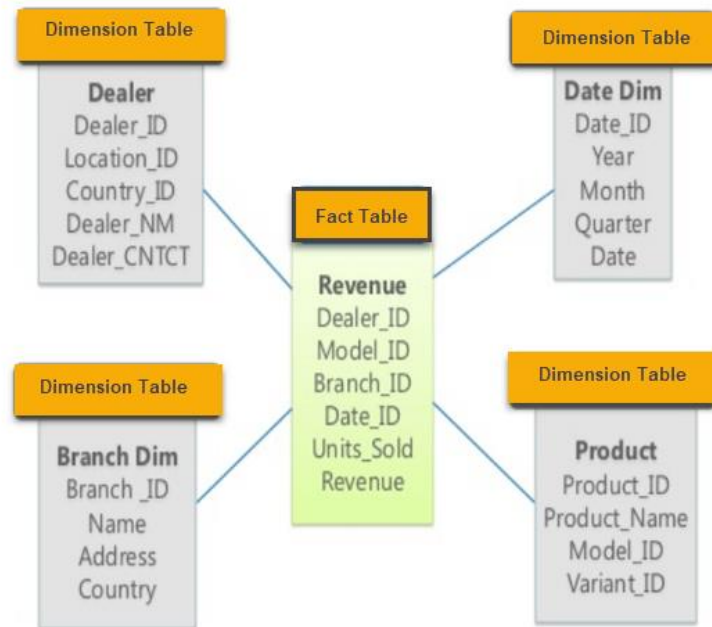
Star vs Snowflake schema

Como organizamos los datos en nuestro Data Warehouse?

Principalmente encontramos el esquema tipo estrella o tipo “copo de nieve”

Star schema

- Facil de entender.
- Queries mas simples y mejor performantes
- Usan mas espacio en memoria
- Mayor redundancia de los datos



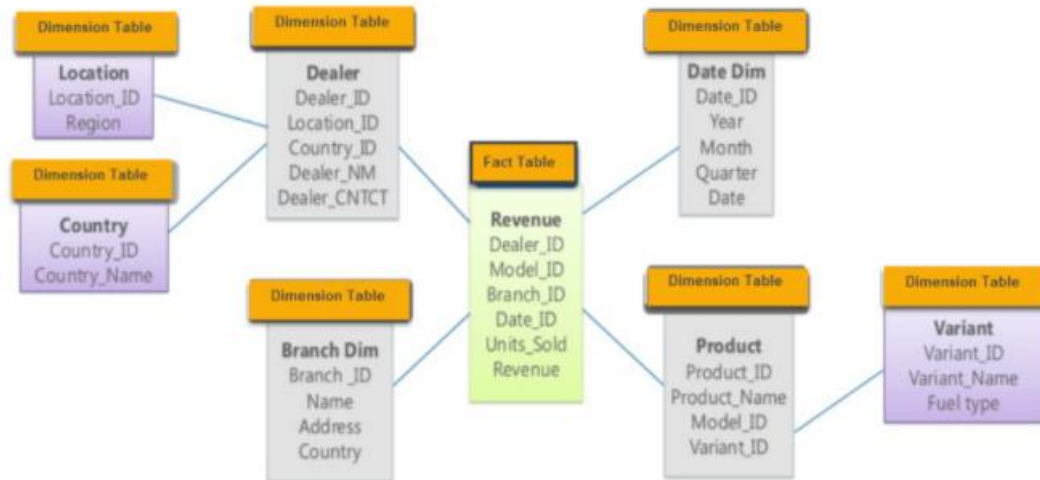
Star vs Snowflake schema

Snowflake schema

- Contiene tablas sub-dimensionales.
- Las queries pueden tomar mas tiempo de ejecución.
- Diseño mas complejo.
- Menor espacio en memoria
- Menor redundancia de los datos

Nota:

- Fact table: Contiene medidas del negocio con claves que refieren a otras tablas dimensionales
- Dimensional table: Contiene atributos descriptivos.



SQL commands basics

SELECT, WHERE, AND

```
SELECT column_name(s)
FROM table_name
WHERE column_1 = value_1
    AND column_2 = value_2;
```

IS NULL / IS NOT NULL

```
SELECT column_name(s)
FROM table_name
WHERE column_name IS NULL;
```

CREATE TABLE

```
CREATE TABLE table_name (
    column_1 datatype,
    column_2 datatype,
    column_3 datatype
);
```

GROUP BY

```
SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name;
```

SELECT DISTINCT

```
SELECT DISTINCT column_name
FROM table_name;
```

ALTER TABLE

```
ALTER TABLE table_name
ADD column_name datatype;
```

LIKE

```
SELECT column_name(s)
FROM table_name
WHERE column_name LIKE pattern;
```

LEFT JOIN

```
SELECT column_name(s)
FROM table_1
LEFT JOIN table_2
    ON table_1.column_name = table_2.column_name;
```

DELETE TABLE

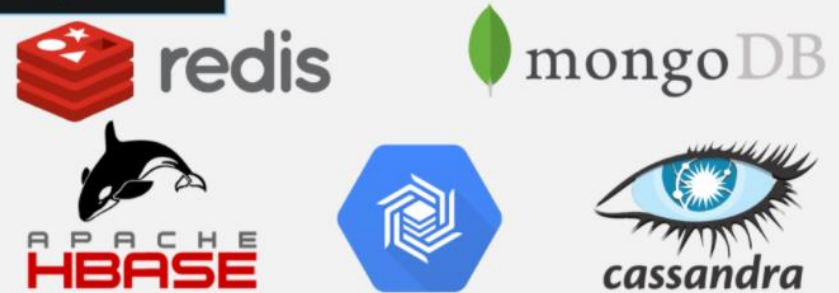
```
DELETE FROM table_name
WHERE some_column = some_value;
```

SQL vs No-SQL

Example of SQL



Example of NoSQL



Cuando usar SQL o No-SQL?

Cuando usar SQL?:

- **Desarrollos web:** para mantener jerarquía de datos
- **Negocios:** inteligencia y análisis de negocios, son temas que requieren el uso de SQL para facilitar el consumo de la información y la identificación de patrones en los datos.
- **Empresarial:** porque tanto el software a la medida y el software empresarial, poseen la característica de mantener información con estructura consistente .

Cuando usar No-SQL?:

- **Redes sociales:** casi obligatorio .
- **Desarrollo Web:** debido a la poca uniformidad de la información que se encuentra en Internet; aun cuando también puede emplearse SQL.
- **Desarrollo Móvil**
- **BigData:** debido a la administración de grandísimas cantidades de información y su evidente heterogeneidad.

SQL vs No-SQL algunas diferencias

SQL

- Las tablas tienen estructuras rígidas, donde cada dato tiene un tipo definido
- SQL tienen buen soporte para escalar verticalmente. Pero es mas costoso.
- Una búsqueda puede tardar minutos y hasta horas debido a la gran cantidad de datos que está revisando.

No-SQL

- Ninguna te exige que definas el tipo de datos que vas a almacenar. NoSQL pone mayor prioridad en cómo acceder dicha data.
- Cuando no tienes la consistencia de datos como prioridad, se distribuye la DB en múltiples máquinas, y por eso se considera que el NoSQL es excelente para bases de datos necesitan escalar horizontalmente.
- NoSQL gane por mucho en velocidad a una SQL, útil para apps donde si no carga en segundos ya piensan en desinstalar / volver a Google.

Python + SQL en código



Cloud computing



De que hablamos con cloud computing?

A cloud provider **can have hundreds of cloud services** that are grouped various types of services. The four most common types of cloud services for Infrastructure as a Service (IaaS) would be:



Compute

Imagine having a virtual computer that can run application, programs and code.



Networking

Image having the a virtual network being able to define internet connections or network isolations



Storage

Imagine having a virtual hard-drive that can store files

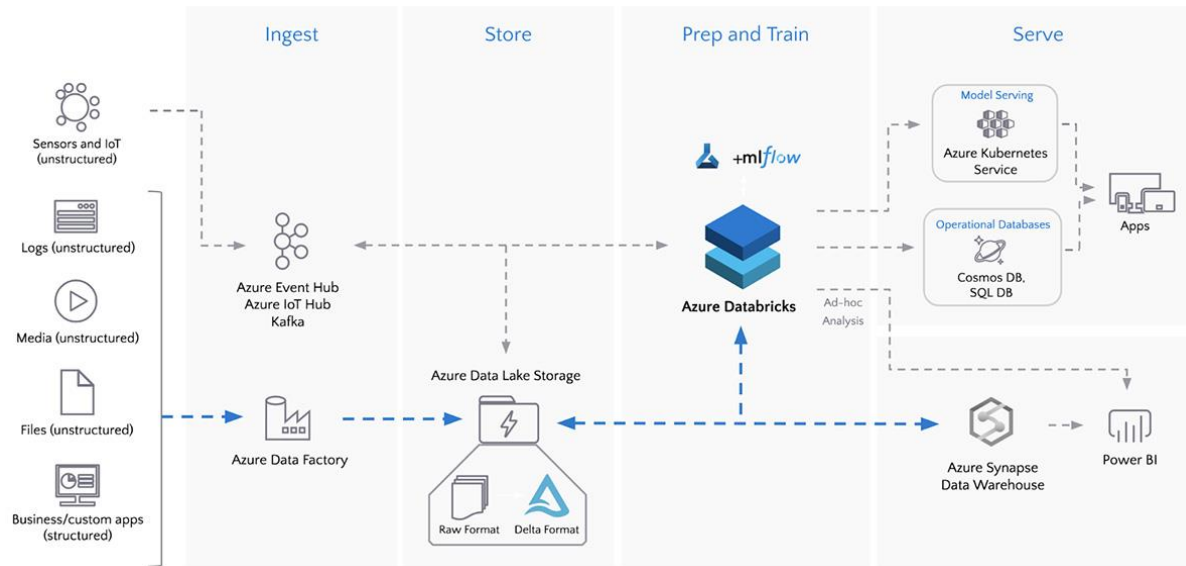
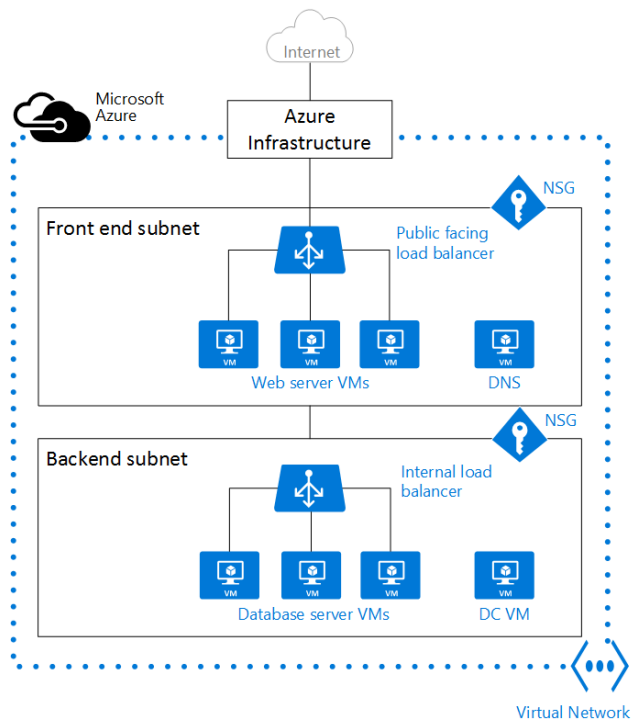


Databases

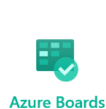
Imagine a virtual database for storing reporting data or a database for general purpose web-application

Todos los servicios ofrecidos por Azure: <https://azure.microsoft.com/es-es/services/>

Estructura alto nivel Azure



Fuente: <https://azure.microsoft.com/>



Spark + Python = PySpark

¿Qué es Pyspark?

Spark es el nombre del motor para realizar la computación en clúster, mientras que PySpark es la biblioteca de Python para usar Spark.

¿Cómo funciona Spark?

Spark se basa en el motor computacional donde cada tarea se realiza en varios equipos de trabajo llamados clúster de computación. Un clúster informático hace referencia a la división de tareas. Una máquina realiza una tarea, mientras que las otras contribuyen a la salida final a través de una tarea diferente.

Pyspark le da al científico de datos una API que se puede usar para resolver los datos paralelos que se han procedido en problemas. Pyspark maneja las complejidades del multiprocessing, como la distribución de los datos, la distribución de código y la recopilación de resultados de los trabajadores en un clúster de máquinas.



- Cargar los datos en el disco
- Importar los datos a la memoria de la máquina
- Procesar/analizar los datos
- Construir el modelo de aprendizaje automático
- Almacenar la predicción en el disco