TECNOLÓGICO NACIONAL DE MÉXICO®

Ciencia de DATOS

DIPLOMADO “CIENCIA DE DATOS”

Módulo 3

ANÁLISIS Y MANIPULACIÓN DE BASE DE DATOS

1. Introducción
2. **Exploración de datos**

La exploración de datos en una base de datos se refiere al proceso de analizar y examinar los datos almacenados en la base de datos para descubrir patrones, tendencias, relaciones y conocimientos útiles. La exploración de datos busca extraer información significativa de los datos y proporcionar una comprensión más profunda de los mismos.

La exploración de datos implica realizar consultas y análisis de los datos utilizando diferentes técnicas y herramientas. Algunos de los métodos comunes utilizados en la exploración de datos incluyen:

a) **Consultas SQL**: Se utilizan para seleccionar, filtrar y agrupar datos de la base de datos con el fin de obtener información específica. Esto puede incluir la recuperación de datos según ciertos criterios, la combinación de datos de diferentes tablas y la aplicación de funciones de agregación.
b) **Análisis estadístico**: Se utilizan técnicas estadísticas para analizar los datos y obtener medidas descriptivas, como promedios, medianas, desviaciones estándar, correlaciones, etc. Esto ayuda a identificar patrones, tendencias y relaciones entre los datos.
c) **Minería de datos**: Es el proceso de descubrir patrones y conocimientos ocultos en los datos mediante algoritmos y técnicas avanzadas. Esto puede incluir técnicas de clasificación, reglas de asociación, agrupamiento, detección de anomalías, entre otros.
d) **Visualización de datos**: Se utilizan gráficos, diagramas y otras representaciones visuales para presentar los datos de manera más comprensible. Esto ayuda a identificar patrones, tendencias y relaciones de manera intuitiva y rápida.

La exploración de datos es una fase crucial en el proceso de **ciencia de datos** que implica analizar y comprender los datos en bruto antes de aplicar técnicas más avanzadas de modelado y análisis. Esta etapa tiene como objetivo descubrir patrones, tendencias, relaciones y anomalías en los datos, así como obtener información relevante que guiará el resto del proceso de análisis. Aquí hay un desglose detallado de cómo se lleva a cabo la exploración de datos en **ciencia de datos**:

a) **Adquisición de datos**: El primer paso es obtener los datos necesarios para el análisis. Esto puede implicar recopilar datos de diferentes fuentes, como bases

de datos, archivos, servicios web, API, registros de sensores, redes sociales, etc.

b) **Limpieza de datos**: Antes de explorar los datos, es esencial realizar una limpieza de los mismos. Esto implica eliminar datos duplicados, corregir errores, llenar datos faltantes o aplicar técnicas para tratar valores atípicos (outliers).
c) **Exploración inicial**: En esta etapa, se realiza una exploración básica de los datos para tener una idea general de su estructura y contenido. Se pueden utilizar estadísticas descriptivas, como la media, la mediana, la desviación estándar y gráficos básicos para visualizar la distribución de los datos.
d) **Visualización de datos**: La visualización de datos es una parte importante de la exploración. Se utilizan gráficos y diagramas para representar los datos de manera intuitiva y ayudar a identificar patrones y tendencias. Algunas visualizaciones comunes incluyen histogramas, diagramas de dispersión, gráficos de barras, gráficos de líneas y mapas de calor.
e) **Análisis de relaciones**: Se investiga la relación entre diferentes variables o atributos. Esto puede incluir la correlación entre variables numéricas o la comparación de categorías a través de gráficos de barra o diagramas de caja.
f) **Análisis de distribuciones**: Se examina la distribución de los datos y se evalúa si siguen una distribución específica, como la distribución normal. Esto es útil para identificar sesgos o asimetrías en los datos.
g) **Selección de características**: Si se trabaja con un gran número de atributos, se puede realizar una selección de características para identificar cuáles tienen mayor relevancia para el análisis y el modelado posterior.
h) **Identificación de patrones y tendencias**: Se buscan patrones repetitivos y tendencias en los datos que puedan indicar comportamientos significativos interesantes.
i) **Búsqueda de anomalías**: Se investiga la presencia de datos atípicos o anómalos que puedan requerir una atención especial o corrección.
j) **Formulación de hipótesis**: Con base en los descubrimientos realizados en la exploración de datos, se pueden formular hipótesis que guiarán el análisis más detallado y la construcción de modelos.

En general, la exploración de datos es una fase iterativa en el proceso de **ciencia de datos**. A medida que se descubren patrones y relaciones interesantes, se pueden plantear nuevas preguntas y enfoques, lo que lleva a una exploración más profunda. Una vez que se completa la exploración de datos, se pueden diseñar modelos y algoritmos más avanzados para resolver problemas específicos y obtener conocimientos más profundos y útiles.

En ciencia de datos, los datos se pueden almacenar en una variedad de lugares y formatos, dependiendo del tamaño, la estructura y la naturaleza de los datos. Algunos de los lugares más comunes donde se almacenan datos en ciencia de datos son los siguientes:

a) **Bases de Datos Relacionales**: Las bases de datos relacionales son ampliamente utilizadas en ciencia de datos para almacenar datos estructurados en tablas con filas y columnas. Estos sistemas utilizan el lenguaje de consulta SQL para realizar operaciones de inserción, actualización, eliminación y recuperación de datos. Algunos ejemplos populares de bases de datos relacionales son MySQL, PostgreSQL, Oracle y Microsoft SQL Server.

b) **Bases de Datos NoSQL**: Cuando los datos no tienen una estructura rígida o no se ajustan bien a las tablas de una base de datos relacional, se pueden almacenar en bases de datos NoSQL (Not Only SQL). Estos sistemas ofrecen mayor flexibilidad y escalabilidad para manejar datos no estructurados, semiestructurados o no relacionales. Algunos ejemplos de bases de datos NoSQL son MongoDB, Cassandra, Redis y Elasticsearch.

c) **Archivos en Disco**: Los datos también pueden almacenarse en archivos en el disco duro o en sistemas de almacenamiento masivo. Estos archivos pueden ser de diferentes formatos, como CSV, JSON, XML, Excel, etc. Los archivos en disco son útiles para datos que no requieren consultas complejas y se utilizan a menudo para la transferencia y el intercambio de datos.

d) **Almacenamiento en la Nube**: Muchas organizaciones y proyectos de ciencia de datos utilizan servicios de almacenamiento en la nube para almacenar y gestionar sus datos. Los servicios en la nube, como Amazon S3, Google Cloud Storage y Microsoft Azure Blob Storage, ofrecen una forma escalable y segura de almacenar grandes volúmenes de datos.

e) **Sistemas de Almacenamiento Distribuido**: Para proyectos que manejan grandes cantidades de datos, se utilizan sistemas de almacenamiento distribuido y en clúster, como Hadoop Distributed File System (HDFS) y Apache HBase. Estos sistemas permiten el almacenamiento distribuido y paralelo de datos en múltiples nodos para una mayor escalabilidad y rendimiento.

f) **Bases de Datos en Memoria**: Algunas aplicaciones de ciencia de datos requieren un acceso extremadamente rápido a los datos y, por lo tanto, utilizan bases de datos en memoria, como Redis y Memcached. Estas bases de datos almacenan los datos en la memoria RAM, lo que permite una recuperación rápida de la información.

Es importante seleccionar la opción de almacenamiento adecuada para cada proyecto de ciencia de datos, considerando factores como la estructura de los datos, el volumen de datos, los requisitos de rendimiento y escalabilidad, y la disponibilidad de herramientas para el análisis y manipulación de los datos almacenados.

2.1 Recuperación de datos

La recuperación de datos en ciencia de datos implica extraer información específica y relevante de una base de datos o conjunto de datos, con el objetivo de analizarlos, realizar inferencias o generar conocimientos útiles. Es una etapa esencial en el

proceso de ciencia de datos y se puede realizar a través de diferentes técnicas y herramientas. A continuación, se explica a detalle cómo se lleva a cabo la recuperación de datos en **ciencia de datos**:

a) **Definición del problema**: Antes de comenzar con la recuperación de datos, es importante tener una clara comprensión del problema o la pregunta que se quiere responder. Esto implica definir los objetivos del análisis y los requisitos específicos de los datos que se necesitan recuperar.
b) **Selección de la base de datos o fuente de datos**: En esta etapa, se elige la base de datos o fuente de datos adecuada para recuperar la información requerida. Puede ser una base de datos relacional, una base de datos NoSQL, un archivo CSV, un conjunto de datos en línea, entre otras opciones.
c) **Conexión y acceso a la base de datos**: Si se utiliza una base de datos, se establece una conexión a ella a través de lenguajes y herramientas como SQL, Python o R. Esta conexión permite acceder a los datos y realizar consultas.
d) **Consulta de datos**: Se utilizan consultas y sentencias específicas para recuperar los datos necesarios. Las consultas pueden ser simples o complejas, según la complejidad de la información requerida.
e) **Filtrado y procesamiento de datos**: Una vez que se han recuperado los datos, es posible que se requiera un filtrado adicional o procesamiento para obtener los datos específicos necesarios para el análisis. Esto puede incluir filtrar por fechas, categorías o valores específicos.
f) **Pre procesamiento de datos**: Si los datos recuperados no están limpios o tienen datos faltantes, es necesario realizar un pre procesamiento adicional para limpiarlos, imputar valores faltantes y asegurarse de que estén listos para el análisis.
g) **Análisis de datos**: Con los datos recuperados y pre procesados, se realiza el análisis de datos utilizando técnicas estadísticas, minería de datos o aprendizaje automático para obtener conocimientos, patrones y tendencias.
h) **Visualización de datos**: La visualización de datos es una parte esencial del análisis de datos. Se utilizan gráficos y diagramas para representar los resultados de manera visual y comprensible, lo que facilita la identificación de patrones y tendencias.
i) **Interpretación de resultados**: Finalmente, se interpretan los resultados obtenidos del análisis de datos para responder a la pregunta o resolver el problema inicial. Los hallazgos pueden guiar la toma de decisiones o generar nuevos conocimientos que sean útiles para la organización o el proyecto.

La recuperación de datos es un proceso iterativo en ciencia de datos. A medida que se obtienen resultados, pueden surgir nuevas preguntas o enfoques, lo que lleva a la iteración y refinamiento del proceso de recuperación y análisis de datos para obtener una comprensión más profunda y significativa de la información.

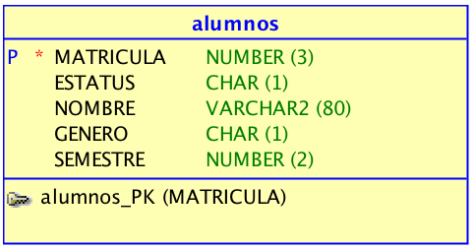
Los comandos de recuperación de datos que se utilizan para obtener información específica y relevante de una base de datos o conjunto de datos almacenad permiten

realizar consultas, filtrar datos, agruparlos y ordenarlos, lo que es esencial para el análisis y la exploración de datos.

Tabla.

La tabla es la estructura fundamental de una base de datos relacional. Se compone de filas y columnas. Cada fila representa una entidad o registro, y cada columna contiene un atributo o característica de ese registro. Por ejemplo, una tabla de alumnos puede tener columnas como “matricula”, ”estatus”, ”nombre”, “genero” y “semestre”.

MATRICULA	E	NOMBRE	GENERO	SEMESTRE
400	R	BALLESTEROS REYES MARIA	F	9
401	R	CISNEROS BAEZA ALEJANDRO	M	9
402	R	ESQUIVEL RUIZ BRANDON	M	9
403	R	FLORES MARTIEZ JORGE	M	8
404	R	FUENTES OLGUIN KARLA	F	8
405	R	GONZALEZ DIAZ ROBERTO	M	8
406	R	GONZALEZ MARQUEZ JOSE ANTONIO	M	7
407	R	GONZALEZ LARA ALEJANDRO	M	7
408	R	GUERRERO TORRES JUAN DIEGO	M	7
409	R	GUERRERO LOPEZ ALMA	F	6
410	R	HERNANDEZ DIAZ ROCIO	F	6
411	R	FERNANDEZ RODRIGUEZ GAEL	M	6



Los comandos de recuperación de datos que se utilizan para obtener información específica y relevante de una base de datos o conjunto de datos almacenad permiten realizar consultas, filtrar datos, agruparlos y ordenarlos, lo que es esencial para el análisis y la exploración de datos

A continuación, se explican los comandos más comunes de recuperación de datos en ciencia de datos utilizando el lenguaje de consulta SQL:

SELECT.

El comando SELECT en SQL es una instrucción fundamental que se utiliza para recuperar datos de una base de datos. Es una de las principales operaciones de consulta que permite seleccionar columnas específicas y/o filas completas de una o varias tablas en una base de datos. Con SELECT, puedes realizar consultas para obtener información específica y relevante de la base de datos según tus necesidades.

La sintaxis básica del comando SELECT es la siguiente:

```
SELECT *|{[DISTINCT] columna|expresión [alias],...}
FROM    tabla;
```

- SELECT identifica las columnas
- FROM Identifica la tabla

Ejemplo 1.

```
SELECT *
FROM   alumnos;
```

MATRICULA	E	NOMBRE	GENERO	SEMESTRE
400	R	BALLESTEROS REYES MARIA	F	9
401	R	CISNEROS BAEZA ALEJANDRO	M	9
402	R	ESQUIVEL RUIZ BRANDON	M	9
403	R	FLORES MARTIEZ JORGE	M	8
404	R	FUENTES OLGUIN KARLA	F	8
405	R	GONZALEZ DIAZ ROBERTO	M	8
406	R	GONZALEZ MARQUEZ JOSE ANTONIO	M	7
407	R	GONZALEZ LARA ALEJANDRO	M	7
408	R	GUERRERO TORRES JUAN DIEGO	M	7
409	R	GUERRERO LOPEZ ALMA	F	6
410	R	HERNANDEZ DIAZ ROCIO	F	6
411	R	FERNANDEZ RODRIGUEZ GAEL	M	6

Ejemplo 2.

```
SELECT matricula, nombre
FROM   alumnos;
```

MATRICULA	NOMBRE
400	BALLESTEROS REYES MARIA
401	CISNEROS BAEZA ALEJANDRO
402	ESQUIVEL RUIZ BRANDON
403	FLORES MARTIEZ JORGE
404	FUENTES OLGUIN KARLA
405	GONZALEZ DIAZ ROBERTO
406	GONZALEZ MARQUEZ JOSE ANTONIO
407	GONZALEZ LARA ALEJANDRO
408	GUERRERO TORRES JUAN DIEGO
409	GUERRERO LOPEZ ALMA
410	HERNANDEZ DIAZ ROCIO
411	FERNANDEZ RODRIGUEZ GAEL

El comando SELECT es esencial para recuperar datos y realizar consultas en SQL. Es una herramienta poderosa que te permite acceder a la información almacenada en una base de datos y obtener los resultados deseados para análisis, informes y otras operaciones relacionadas con datos.

2.2 Expresiones Aritméticas

Los operadores aritméticos en SQL son símbolos especiales que se utilizan para realizar operaciones matemáticas en los datos numéricos almacenados en una base de datos. Estos operadores permiten realizar cálculos y manipulaciones numéricas en las columnas de una tabla durante una consulta SQL. Los operadores aritméticos básicos son los siguientes:

Operador	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División

Suma (+): Se utiliza para sumar dos o más valores numéricos.

```
SELECT columna1 + columna2 AS suma_resultado
FROM tabla;
```

Resta (-): Se utiliza para restar un valor numérico de otro.

```
SELECT columna1 - columna2 AS resta_resultado
FROM tabla;
```

Multiplicación (*): Se utiliza para multiplicar dos o más valores numéricos.

```
SELECT columna1 * columna2 AS multiplicacion
FROM tabla;
```

División (/): Se utiliza para dividir un valor numérico entre otro. Es importante tener en cuenta qué si la división es entre enteros, el resultado será un entero. Si deseas un resultado decimal, asegúrate de que al menos uno de los operandos sea un número decimal o real.

```
SELECT columna1 / columna2 AS division_resultado
FROM tabla;
```

Módulo o Resto (%): Se utiliza para obtener el resto de la división entre dos valores numéricos.

```
SELECT columna1 % columna2 AS modulo_resultado
FROM tabla;
```

Es importante tener en cuenta que los operadores aritméticos siguen las reglas básicas de la aritmética y tienen prioridades. Si tienes múltiples operadores aritméticos en una misma consulta, se evaluarán en el siguiente orden: paréntesis (), multiplicación y división primero, y luego suma y resta.

Además de los operadores aritméticos básicos, SQL también admite otras funciones matemáticas incorporadas que pueden usarse en combinación con los operadores aritméticos para realizar cálculos más complejos o específicos en los datos numéricos.

Ejemplo 1.

En este ejemplo, la consulta utiliza el operador de suma (+) para calcular el siguiente.

```
SELECT matricula, nombre, semestre, semestre + 1
FROM   alumnos;
```

MATRICULA	NOMBRE	SEMESTRE	SEMESTRE + 1
400	BALLESTEROS REYES MARIA	9	10
401	CISNEROS BAEZA ALEJANDRO	9	10
402	ESQUIVEL RUIZ BRANDON	9	10
403	FLORES MARTIEZ JORGE	8	9
404	FUENTES OLGUIN KARLA	8	9
405	GONZALEZ DIAZ ROBERTO	8	9
406	GONZALEZ MARQUEZ JOSE ANTONIO	7	8
407	GONZALEZ LARA ALEJANDRO	7	8
408	GUERRERO TORRES JUAN DIEGO	7	8
409	GUERRERO LOPEZ ALMA	6	7
410	HERNANDEZ DIAZ ROCIO	6	7
411	FERNANDEZ RODRIGUEZ GAEL	6	7

Clausula DISTINCT.

La cláusula DISTINCT en SQL se utiliza junto con el comando SELECT y permite eliminar duplicados de los resultados de una consulta. Cuando se aplica la cláusula DISTINCT a una columna o conjunto de columnas en una consulta, el resultado solo mostrará valores únicos para esa columna o combinación de columnas.

```
SELECT DISTINCT semestre
FROM   alumnos;
```

SEMESTRE
9
8
7
6

La cláusula DISTINCT es útil cuando se quiere obtener una lista de valores únicos en una columna o cuando se desean resultados sin duplicados para ciertas operaciones de análisis o agregación. Es importante tener en cuenta que, en algunas ocasiones, el uso de DISTINCT puede hacer que la consulta sea más lenta, especialmente cuando se trabaja con grandes conjuntos de datos. Por lo tanto, es recomendable utilizarlo de manera apropiada y evaluar su impacto en el rendimiento según las necesidades específicas del proyecto.

Renombrando columnas

Los ALIAS de columnas en SQL se utilizan para dar nombres temporales o alternativos a las columnas de los resultados de una consulta. Estos nombres temporales o alias son útiles para mejorar la legibilidad de los resultados y para dar un significado más claro a las columnas, especialmente cuando los nombres originales de las columnas son largos o poco descriptivos. Use la palabra reservada AS para renombrar

```
SELECT matricula AS numero, nombre AS nomble_completo
FROM   alumnos;
```

La palabra reservada AS se puede omitir para renombrar columnas

```
SELECT matricula numero, nombre nomble_completo
FROM   alumnos;
```

NUMERO	NOMBRE_COMPLETO
400	BALLESTEROS REYES MARIA
401	CISNEROS BAEZA ALEJANDRO
402	ESQUIVEL RUIZ BRANDON
403	FLORES MARTIEZ JORGE
404	FUENTES OLGUIN KARLA
405	GONZALEZ DIAZ ROBERTO
406	GONZALEZ MARQUEZ JOSE ANTONIO
407	GONZALEZ LARA ALEJANDRO
408	GUERRERO TORRES JUAN DIEGO
409	GUERRERO LOPEZ ALMA
410	HERNANDEZ DIAZ ROCIO
411	FERNANDEZ RODRIGUEZ GAEL

Extracción de datos desde PYTHON.

Python es un lenguaje de programación interpretado, de alto nivel, de propósito general y de código abierto. Fue creado por Guido van Rossum y lanzado por primera vez en 1991. Python ha ganado una gran popularidad debido a su sintaxis clara y legible, que hace que sea fácil de aprender y usar, así como a su amplia comunidad de desarrolladores que lo respalda y su amplia gama de bibliotecas y frameworks disponibles.

Características principales de Python:

a) **Fácil de aprender y leer**: Python utiliza una sintaxis clara y legible que se asemeja mucho al lenguaje humano, lo que facilita su aprendizaje y comprensión para programadores principiantes y experimentados.
b) **Versátil**: Python es un lenguaje de propósito general que se puede utilizar para una amplia variedad de tareas, desde desarrollo web y análisis de datos hasta scripting, automatización, inteligencia artificial y mucho más.
c) **Multiplataforma**: Python es compatible con múltiples plataformas, lo que significa que puede ser ejecutado en diversos sistemas operativos, como Windows, macOS, Linux, entre otros.
d) **Interpretado**: Python es un lenguaje interpretado, lo que significa que el código fuente se traduce en tiempo de ejecución a código máquina en lugar de ser compilado antes de ser ejecutado. Esto facilita la depuración y la modificación del código sin necesidad de recompilar.
e) **Orientado a objetos**: Python es un lenguaje orientado a objetos, lo que permite utilizar programación orientada a objetos para crear y estructurar código de manera eficiente y organizada.
f) **Amplia biblioteca estándar**: Python cuenta con una amplia biblioteca estándar que proporciona módulos y funciones predefinidos para realizar tareas comunes, lo que facilita la implementación de funcionalidades avanzadas sin necesidad de desarrollarlas desde cero.
g) **Comunidad activa**: Python tiene una comunidad de desarrolladores muy activa y dedicada que contribuye al desarrollo de nuevas funcionalidades, módulos y bibliotecas, lo que enriquece el ecosistema de Python y proporciona soluciones para diversos problemas.

Python es ampliamente utilizado en diversas áreas, como desarrollo web (con frameworks como Django o Flask), análisis de datos (con bibliotecas como Pandas o NumPy), inteligencia artificial y aprendizaje automático (con bibliotecas como TensorFlow o PyTorch), automatización, scripting, entre otros. Debido a su versatilidad y facilidad de uso, Python se ha convertido en uno de los lenguajes de programación más populares y preferidos por muchos desarrolladores en todo el mundo.

Para extraer datos desde Python en ciencia de datos, puedes utilizar diversas bibliotecas y métodos según el origen y formato de los datos que desees obtener. Algunas de las bibliotecas más comunes para este propósito son:

a) **Pandas**: es una biblioteca de Python ampliamente utilizada para la manipulación y análisis de datos. Puedes usar Pandas para extraer datos desde archivos CSV, Excel, bases de datos SQL y otras fuentes de datos estructurados

```
import pandas as pd
# Leer datos desde un archivo CSV
data = pd.read_csv('ruta/del/archivo.csv')
```

b) **NumPy**: es una biblioteca fundamental para la computación numérica en Python. Puedes usar NumPy para trabajar con matrices y arreglos multidimensionales, lo que es útil para procesar grandes conjuntos de datos numéricos

```
import numpy as np
# Crear un arreglo NumPy
data = np.array([1, 2, 3, 4, 5]);
```

c) **Request**: Si los datos están disponibles en una API web, puedes utilizar la biblioteca Requests para hacer solicitudes HTTP y obtener los datos

```
import requests

# Hacer una solicitud GET a una API
response = requests.get('https://api.example.com/data')
data = response.json()
# Si los datos están en formato JSON;
```

d) **SQLAlchemy**: Si deseas extraer datos desde una base de datos SQL, puedes utilizar SQLAlchemy, que proporciona una capa de abstracción para interactuar con bases de datos relacionales.

```
from sqlalchemy import create_engine

# Conectar a la base de datos
engine = create_engine('sqlite:///ruta/del/archivo.db')
data = pd.read_sql_query('SELECT * FROM tabla', engine);
```

e) **BeautifulSoup y Scrapy**: Si necesitas extraer datos desde páginas web, puedes utilizar BeautifulSoup o Scrapy para realizar web scraping y obtener los datos de manera estructurada.

Estos son solo ejemplos básicos, y el proceso de extracción de datos puede variar dependiendo del formato de los datos y la fuente de los mismos. En ciencia de datos, es común utilizar una combinación de estas bibliotecas y métodos para recolectar y preparar los datos antes de realizar el análisis y modelado.