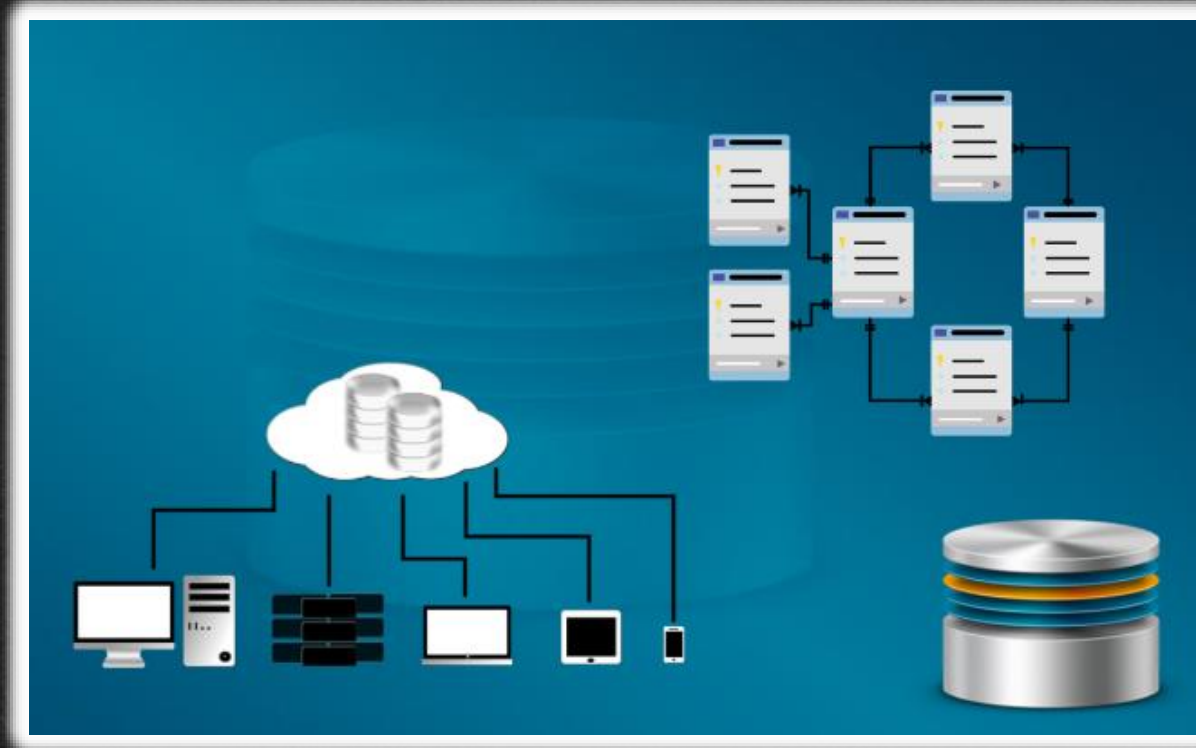




ProfesorenVideo.com

Curso de Python

Diseño de Bases de Datos





ProfesorenVideo.com

Curso de Python

Integridad de los datos: En términos generales, implica asegurarse de que los datos sean correctos y estén libres de errores, redundancias o inconsistencias. Aquí hay algunos aspectos clave de lo que significa mejorar la integridad de los datos:

1. **Exactitud:** Esto significa que los valores almacenados en la base de datos son correctos y están actualizados.
2. **Consistencia:** Los datos deben ser coherentes tanto dentro de una misma tabla como entre diferentes tablas relacionadas. Ejm., Toda clave **fk** debe tener un valor que coincida con la respectiva **pk**.
3. **Validez:** Los datos deben cumplir con las reglas y restricciones definidas para ellos. Por ejemplo, un campo de fecha debe contener una fecha válida, no puede tener letras.



ProfesorenVideo.com

Curso de Python

Integridad de los datos: En términos generales, implica asegurarse de que los datos sean correctos y estén libres de errores, redundancias o inconsistencias. Aquí hay algunos aspectos clave de lo que significa mejorar la integridad de los datos:

4. **Integridad Referencial:** Garantiza que las relaciones entre entidades (tablas) estén correctamente establecidas mediante el uso de claves primarias y claves foráneas. Esto ayuda a prevenir la orfandad de datos.
5. **Seguridad:** Los datos deben estar protegidos contra accesos no autorizados y manipulaciones indebidas. La integridad de los datos también involucra asegurar que sólo usuarios autorizados puedan acceder, modificar o eliminar información de la base de datos.



ProfesorenVideo.com

Curso de Python

Normalización: Es el proceso de organizar una base de datos para reducir la redundancia y mejorar la integridad de los datos.

Formas normales (1NF, 2NF, 3NF):

- **1NF:** Elimina grupos repetitivos, asegura que cada campo contiene solo un valor.
- **2NF:** Elimina la redundancia de datos parciales, asegurando que cada campo depende de la llave primaria completa.
- **3NF:** Elimina la dependencia transitiva, asegurando que los campos no clave no dependen entre sí.



ProfesorenVideo.com

Curso de Python

Formas normales (1NF, 2NF, 3NF):

- 1NF: Elimina grupos repetitivos, asegura que cada campo contiene solo un valor.

TRABAJADOR

DNI	APELLIDOS	NOMBRES	EDAD_SEXO	SALARIO
123456	MENDOZA C.	MARIA L.	35 , FEMENINO	33567



ProfesorenVideo.com

Curso de Python

Formas normales (1NF, 2NF, 3NF):

- **2NF**: Elimina la redundancia de datos parciales, asegurando que cada campo depende de la llave primaria completa.

CLIENTE_PRODUCTO

DNI	APELLIDOS	NOMBRES	EDAD	SEXO	TELEFONO	CODIGO_PRODUCTO	DESCRIPCIÓN_PRODUCTO
1234	MENDOZA C.	LUIS A.	27	F	32567456	BOTMIN_500	Botella Agua mineral 500 cc





ProfesorenVideo.com

Curso de Python

Formas normales (1NF, 2NF, 3NF):

- **3NF**: Elimina la dependencia transitiva, asegurando que los campos no clave no dependen entre sí.

CANTANTES_GENERO

CODIGO_CANTANTE	NOMBRE_CANTANTE	GENERO_MUSICAL
CAN-R22	MICHAEL JACKSON	POP
CAN-R23	FREDDIE MERCURY	ROCK
CAN-R24	POLO G	HIP-HOP





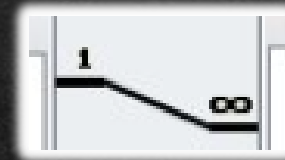
ProfesorenVideo.com

Curso de Python

Simbología para las distintas relaciones:

1. Relación Uno a Uno (**One-to-One**):
2. Relación Uno a Muchos (**One-to-Many**):
3. Relación Muchos a Uno (**Many-to-One**):
4. Relación Muchos a Muchos (**Many-to-Many**):

De uno a uno ———
De uno a muchos ———<
De muchos a muchos >————<



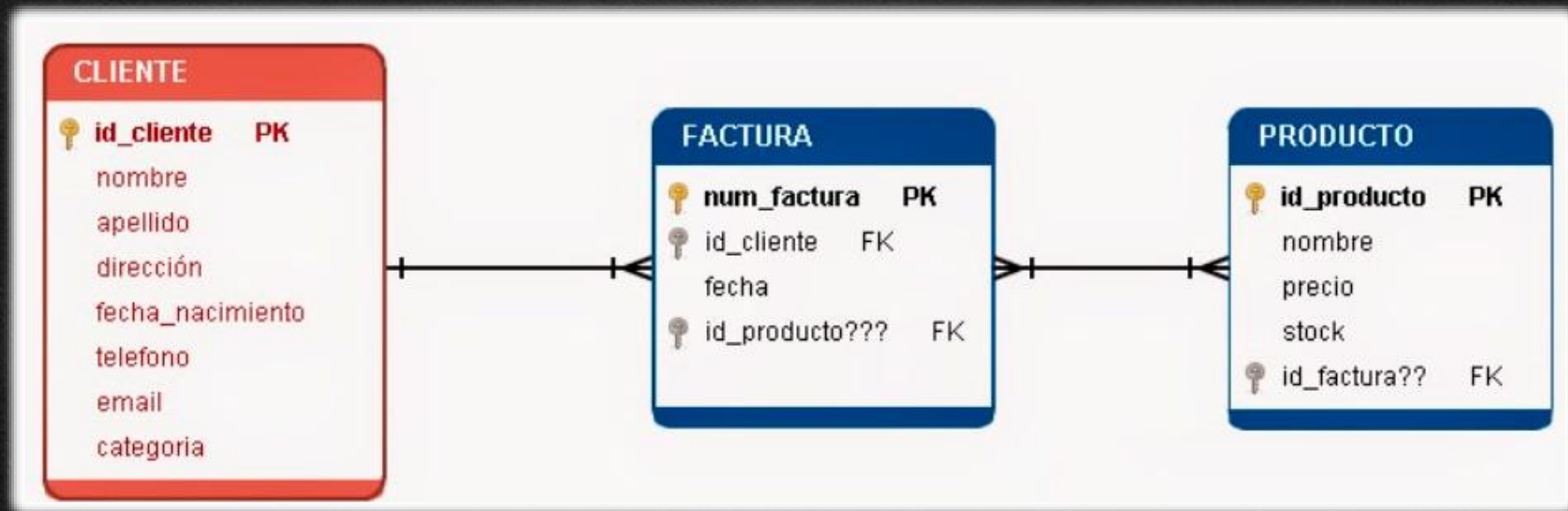
1 : 1
1 : N
N : 1
N : M



ProfesorenVideo.com

Curso de Python

Diagramas de Entidad-Relación (ERD): es una herramienta visual utilizada en el diseño de bases de datos para representar de manera gráfica las entidades del sistema, las relaciones entre esas entidades y los atributos asociados a cada entidad.



Curso de Python

Tipos de relaciones (Relationships) entre entidades: Las relaciones representan las conexiones y asociaciones entre entidades. Existen varios tipos, a saber:

1. Relación Uno a Uno (**One-to-One**): En una relación uno a uno, un registro en la tabla A está asociado con exactamente un registro en la tabla B, y viceversa.
Ejemplo: Tabla Estudiantes y tabla Expedientes personales.
1. Relación Uno a Muchos (**One-to-Many**): En una relación uno a muchos, un registro en la tabla A puede estar asociado con uno o más registros en la tabla B, pero un registro en la tabla B está asociado con solo un registro en la tabla A.
Ejemplo: Tabla Departamento y tabla Empleados.
1. Relación Muchos a Uno (**Many-to-One**): Es el inverso de la relación uno a muchos. En este caso, varios registros en la tabla A pueden estar asociados con un solo registro en la tabla B.
Ejemplo: Tabla Empleados y tabla Departamento.
1. Relación Muchos a Muchos (**Many-to-Many**): En una relación muchos a muchos, varios registros en la tabla A pueden estar relacionados con varios registros en la tabla B. **Para su implementación se requiere una tabla intermedia que conecte a las 2 tablas principales.**
Ejemplo: Tabla Estudiantes y tabla Cursos.



ProfesorenVideo.com

Curso de Python

Tipos de Tablas según su Propósito

Tablas Maestras (Master Tables): Son tablas principales que contienen información central y fundamental para el sistema. No se modifican con mucha frecuencia. Suelen tener claves **pk**. Ejm., "Empleados"

Tablas de Transacción(Transaction Tables): Son tablas donde se registran transacciones individuales o eventos que modifican el estado de los datos en el sistema. Suelen tener claves **pk** y **fk**. Ejm., "Ventas"

Tablas de Detalle (Detail Tables): Estas tablas contienen información detallada que está relacionada con las tablas maestras. Suelen tener claves **pk** y **fk**. Ejm., "Detalles de la Venta"

Tablas Índice(Index Tables): Son tablas utilizadas para mejorar el rendimiento de las consultas al mantener índices de datos importantes de otras tablas. **No contienen los datos en sí mismos, sino que actúan como referencias rápidas a los datos almacenados en otras tablas.**

Tablas de Movimiento(Movement Tables): Estas tablas registran el movimiento o cambios de estado de los datos a lo largo del tiempo. **Se utilizan principalmente para auditoría, seguimiento de cambios históricos o para mantener un registro de todas las modificaciones realizadas en los datos.**



ProfesorenVideo.com

Curso de Python

Por su atención muchas Gracias!

Siguiente clase:

Creando Tablas, Campos y Relaciones