



Integridad y modelos de datos



**Activen las cámaras los que puedan y
pasemos asistencia**





Inicio

{**desafío**}

latam_



/ Agregar una clave primaria */*

/ Agregar una clave foránea */*

/ Convertir un modelo de datos a una base de datos*/*

Objetivos

Activación de conceptos

Contesta la pregunta correctamente y gana un punto

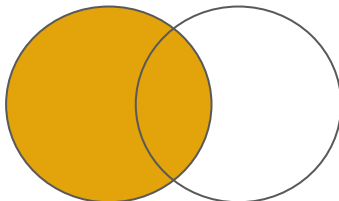
Instrucciones:

- Se realizará una pregunta, el primero en escribir “YO” por el chat, dará su respuesta al resto de la clase.
- El docente validará la respuesta.
- En caso de que no sea correcta, dará la oportunidad a la segunda persona que dijo “Yo”.
- Cada estudiante podrá participar un máximo de 2 veces.
- Al final, el/la docente indicará el 1º, 2º y 3º lugar.
- Esta actividad no es calificada, es solo una dinámica para recordar los conceptos clave para abordar esta sesión.





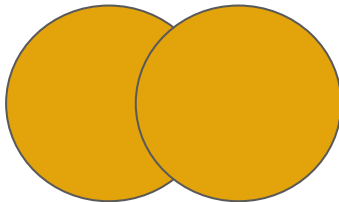
Activación de conceptos



¿A qué tipo de JOIN corresponde el gráfico
mostrado?



Activación de conceptos



¿A qué tipo de JOIN corresponde el gráfico
mostrado?



Activación de conceptos



Si unimos con Inner Join todos los elementos de dos tablas, pero una de las tablas está vacía ¿Qué sucede?



Activación de conceptos



Primer lugar:



Segundo lugar:



Tercer lugar:



Desarrollo

{desafío}
latam_



/* Integridad de datos*/

Integridad de datos

¿De qué se trata?

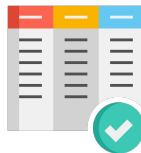
Como administradores de bases de datos debemos considerar 3 aspectos en la integridad de datos:

- **Integridad referencial:** Generada entre datos que se cruzan entre varias tablas.
- **Integridad mediante clave primaria:** Generada a través de un dato que contiene un valor único, por ejemplo la clave primaria (Primary Key, PK)
- **Integridad mediante restricciones:** Generada a partir de la restricción de los tipos de valores que se pueden almacenar en un campo de una tabla.
 - Por ejemplo, si tenemos un campo que es de tipo numérico, entonces la restricción al configurar el campo y la tabla, es definirlo como numérico. En este sentido, si intentamos ingresar texto, obtendremos un error.

Integridad de datos

Motivación

Integridad de datos = correctitud de datos + completitud de los datos



- Problemas de diseño de una base de datos pueden causar problemas de integridad.
- En esta clase aprenderemos a evitar algunos de ellos.

Integridad de datos

Falta de identificador único

Nombre	Edad
Consuelo	27
Consuelo	32
Francisco	27

Queremos aumentar la edad de Consuelo.

¿Cómo lo hacemos?

- Si actualizamos por edad modificaremos 2 registros.
- Si actualizamos por nombre modificaremos 2 registros.
- Es posible modificar por ambos, pero en algunas ocasiones se nos podría olvidar y causar un problema de correctitud de datos.

Integridad de datos

Solución: agregar un valor único

Si agregamos una columna con un valor distinto para cada registro, que no pueda repetirse por ningún motivo y que no pueda ser nulo, entonces tendríamos resuelto el problema.

Id	Nombre	Edad
1	Consuelo	27
2	Consuelo	32
3	Francisco	27

Integridad de datos

Asegurándose que el valor sea único y no nulo

Para asegurarnos que este valor realmente no pueda repetirse, ya que sería una amenaza a la integridad de los datos, especificaremos que esta columna es una **clave primaria** (primary key, abreviado PK).

Id (PK)	Nombre	Edad
1	Consuelo	27
2	Consuelo	32
3	Francisco	27

Integridad de datos

Agregamos un identificador único

Para crear desde cero una tabla con PK	Para alterar una tabla existente
<pre>CREATE TABLE users(id int primary key, nombre varchar, edad int);</pre>	<pre>/* Si ya tiene una columna para la PK */ ALTER TABLE users ADD PRIMARY KEY (id) /* Si no tiene la columna */ ALTER TABLE users ADD COLUMN ID PRIMARY KEY</pre>



Esto no es necesario saberlo de memoria, puedes encontrarlo en Google.

Integridad de datos

Probando la solución

Id	Nombre	Edad
1	Consuelo	27
	Consuelo	32
3	Francisco	27
3	Javier	10

Al establecer que un campo es clave primaria (Primary key, abreviado **PK**) nos aseguramos de que deba ser **único** y además que no pueda ser **nulo**.

Intentemos insertar dos usuarios con el mismo id y un usuario sin id

Veamos los errores.

Ejercicio - Claves primarias e integridad

A continuación, realizaremos un ejercicio donde verificaremos los resultados de ejecución cuando insertamos datos en una tabla que contiene clave primaria.

1. Crea una tabla posts con id, título y contenido, dejando id como clave primaria.
2. Intenta ingresar 2 posts con el mismo id.
3. Intenta ingresar un post sin id.

```
create table posts (id int, title varchar(100), content varchar(100), primary key(id));
```

Ejercicio ¡Manos al teclado!



*/** Agregar una clave primaria **/* 

*/** Agregar una clave foránea **/*

*/** Convertir un modelo de datos a una base de datos**/*

Objetivos

Integridad de datos

Integridad referencial

En otras ocasiones nos preocupará que no se borre un registro relacionado con otro registro.

Usuarios

Id	Nombre	Edad
1	Consuelo	27
2	Consuelo	32
3	Francisco	27

Pago

Id	Monto	Usuario_id
1	1500	1
2	1300	2
3	2700	3

No queremos que se borre un usuario, porque quedaría un pago “zombie”.

Integridad de datos

Agregando una clave foránea

Una clave foránea (Foreign Key y abreviado FK) nos permite protegernos de esta situación. Una clave foránea debe apuntar a una clave única (y de preferencia primaria).

```
CREATE TABLE pagos (  
  id int primary key,  
  monto int,  
  usuario_id int references users(id)  
);
```

- Insertemos un pago asociado al usuario 1 y luego intentemos borrar el usuario 1.

Ejercicio - Claves foráneas

A continuación, realizaremos un ejercicio donde se analizará el comportamiento de las claves foráneas para datos que estén relacionados entre dos tablas. Para ello, realiza las siguientes acciones:

1. Crea la tabla comments con id, contenido y una clave foránea llamada post_id apuntando a la PK de posts.
2. Inserta un comentario asociado al post 1.
3. Intenta borrar el post 1.

```
create table comments (id int, content varchar(100), foreign  
key(id) references posts(id), primary key(id));
```

Ejercicio ¡Manos al teclado!



*/** Agregar una clave primaria **/* ✓

*/** Agregar una clave foránea **/* ✓

*/** Convertir un modelo de datos a una base de datos**/*

Objetivos

Modelos de datos

Introducción a modelos de datos

En algunas ocasiones nos entregarán un modelo de datos en lugar de una tabla.
Dentro de los modelos de entidad relación (ER) existen 3 tipos de modelos famosos:

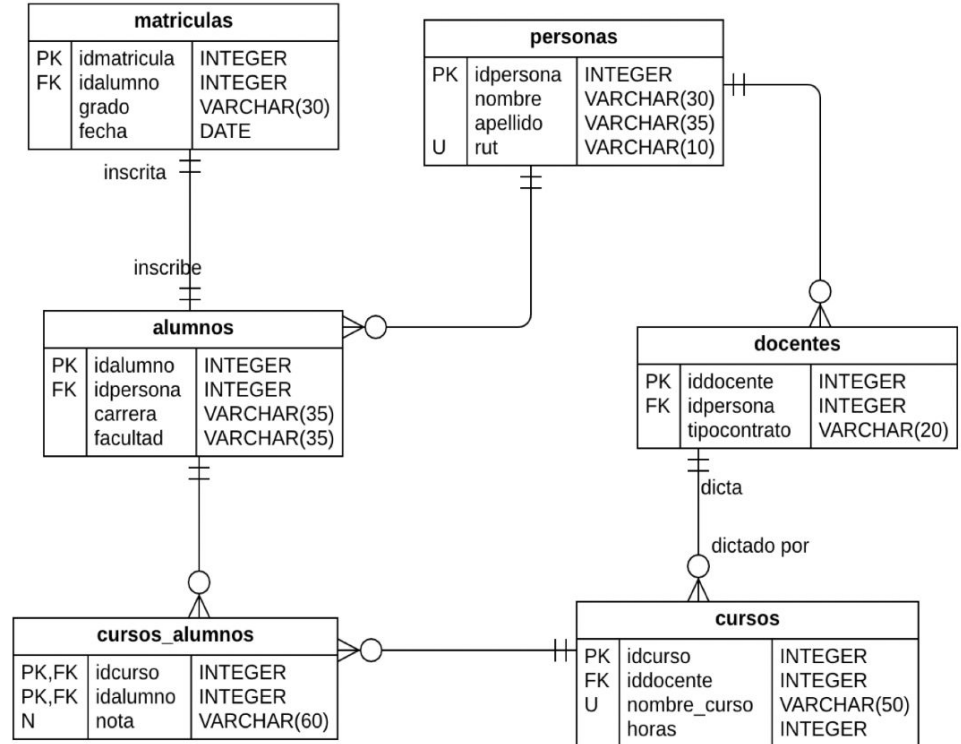
- Modelo conceptual
- Modelo lógico
- Modelo físico

El modelo físico es el que más se asemeja a su implementación final, en esta clase abordaremos este. Por ahora **lo que tenemos que lograr es saber implementar un modelo entregado.**

Modelos de datos

Modelo físico

- El modelo de datos físicos representa cómo se construirá el modelo de la base de datos.
- Aquí se especifican las tablas, columnas, claves primarias y foráneas así como otras restricciones.



Modelos de datos

Relaciones

- Las relaciones entre tablas se indican con una línea cuyas terminaciones indican el tipo de relación.
- Como la relación de muchos tiene esa forma, a este tipo de notación de relaciones se le conoce como crow foot (pies de cuervo).



Uno



Muchos



Uno y solo uno



Cero o uno



Uno o muchos

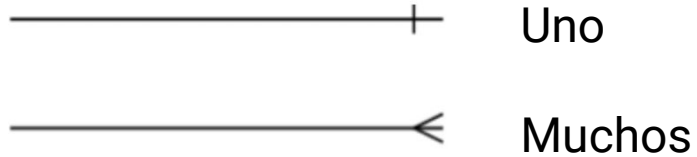


Cero o muchos

Modelos de datos

Relaciones

Por ahora nos enfocaremos en las relaciones de uno y de muchos.

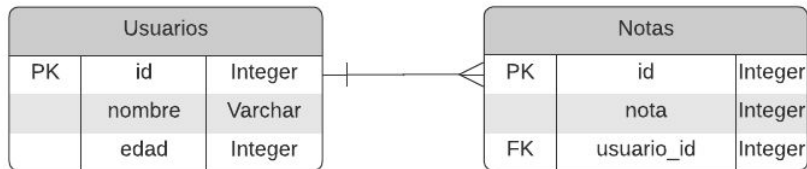


Modelos de datos

Cardinalidad

Se llama cardinalidad al tipo de relación que hay entre dos tablas.
Principalmente, hablaremos de 3 casos:

1. 1 : 1 (Uno a uno)
2. 1 : N (Uno a muchos)
3. N : N (Muchos a muchos)



En nuestro ejemplo, tenemos una tabla de uno a muchos:

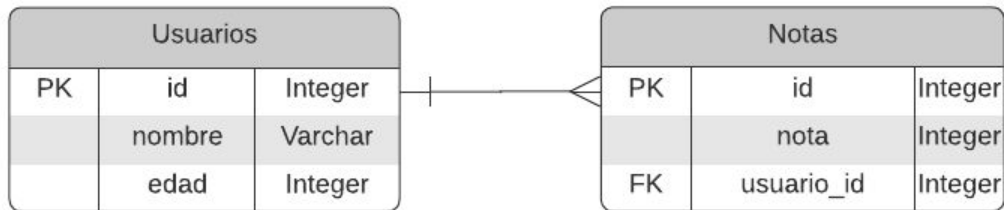
- Un usuario tiene muchas notas y una nota le pertenece a un único usuario.

Modelos de datos

Cardinalidad

Las tablas más utilizadas son las de 1 a N y las de N a N. Una tabla de 1 a N se implementa directamente tal como hemos hecho hasta ahora.

1. 1 : N (Uno a muchos)

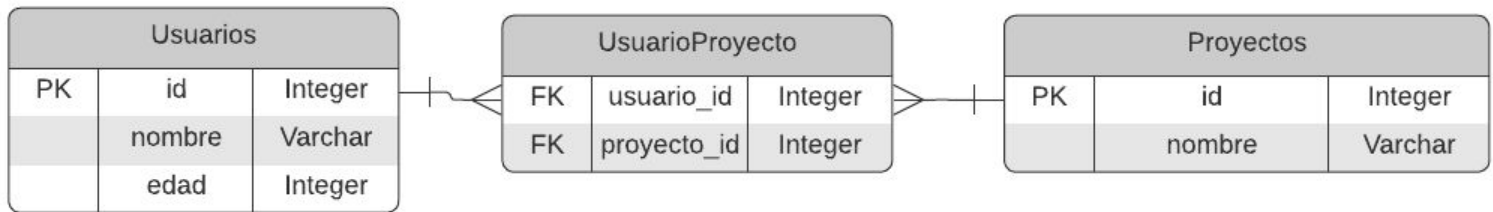
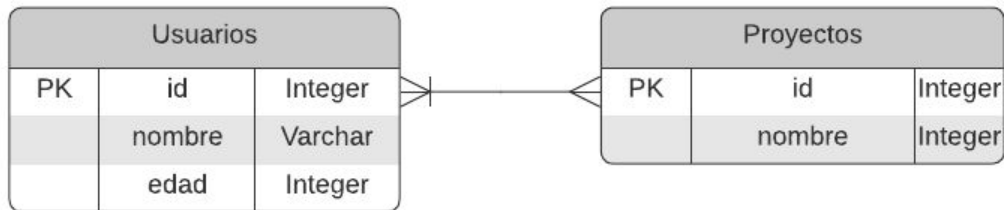


- La línea vertical del lado de la tabla usuario nos indica que necesariamente una nota le debe pertenecer a un usuario, ya que no puede haber una nota sin usuario.

Modelos de datos

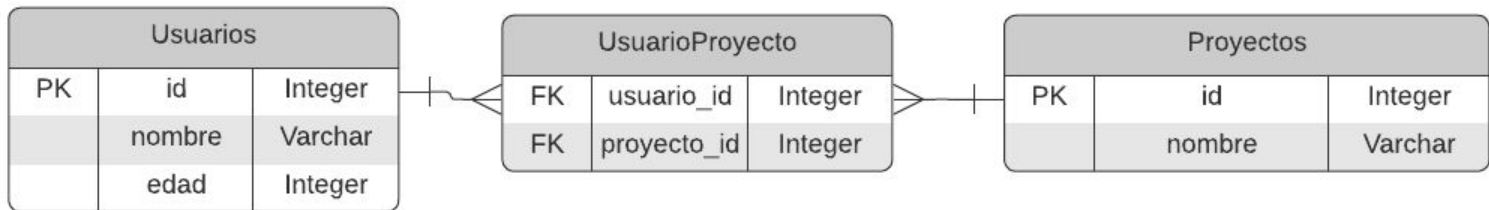
Cardinalidad

Las relaciones N a N no pueden implementarse directamente, requieren de crear una tabla intermedia.



Modelos de datos

N a N y tabla intermedia

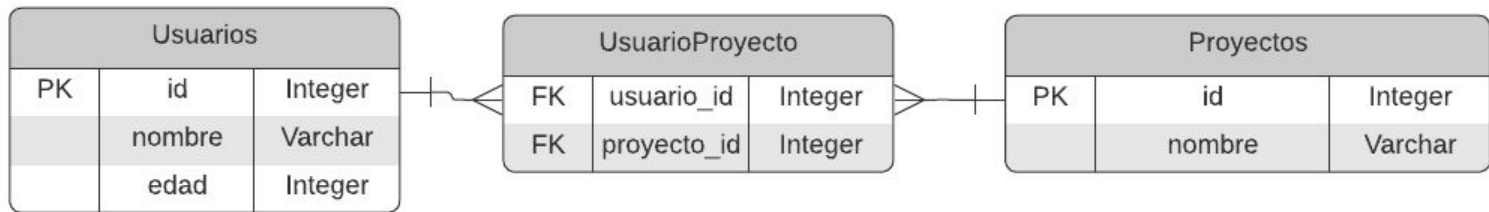


La lógica de la tabla intermedia siempre es igual entre 2 tablas de N a N, contiene las FK de las otras dos tablas y siempre queda del lado de los muchos.

Modelos de datos

N a N y tabla intermedia

Una vez que tenemos el modelo de esta forma podemos llegar e implementarlo con SQL.

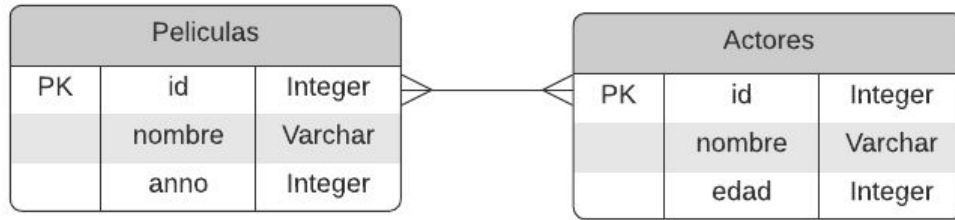


```
CREATE TABLE "Usuarios" (  
  "id" Integer,  
  "nombre" Varchar,  
  "edad" Integer,  
  PRIMARY KEY ("id")  
);
```

```
CREATE TABLE UsuarioProyecto (  
  usuario_id integer,  
  proyecto_id integer,  
  foreign key (usuario_id) references  
    "Usuarios" (id),  
  foreign key (proyecto_id) references  
    "Proyectos" (id)  
);
```

```
CREATE TABLE "Proyectos" (  
  "id" Integer,  
  "nombre" Varchar,  
  PRIMARY KEY ("id")  
);
```

Ejercicio - Relaciones y tablas



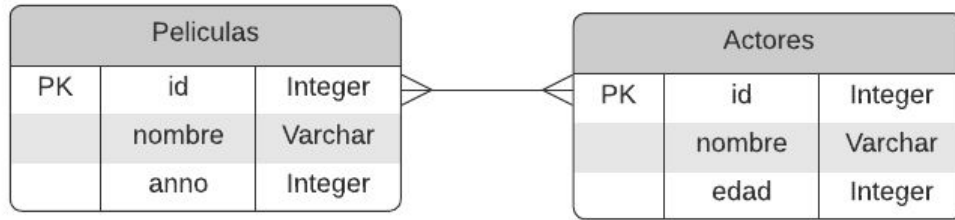
1. ¿Qué tipo de relación es?

Ejercicio

¡Manos al teclado!



Ejercicio



Utilizando como referencia el modelo de datos N:N, dibuja el diagrama insertando una tercera tabla que sea intermediaria entre películas y actores.

Ejercicio

¡Manos al teclado!



*/** Agregar una clave primaria **/* ✓

*/** Agregar una clave foránea **/* ✓

*/** Convertir un modelo de datos a una base de datos**/* ✓

Objetivos



Cierre

{desafío}
latam_



¿Existe algún concepto que no
hayas comprendido?

Reflexionemos

- Revisemos el material de estudio asincrónico.
 - Agregar claves primarias y foráneas a tablas existentes.
 - Restricciones como Not NULL y Unique.
 - Otras restricciones llamadas Checks.
 - Borrado en cascada.
- Revisemos el desafío de la unidad leyendo la descripción y requerimientos.

¿Qué sigue?

¿Tienen alguna duda respecto al Desafío?



*Academia de
talentos digitales*

www.desafiolatam.com



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam