BASES DE DATOS

Clase 9 – Restricciones de integridad

Profesora: Erika Gutiérrez Beltrán

Tema 2: Bases de datos relacionales

Objetivo

Transformar los modelos conceptuales a lógicos. Construir bases de datos para datos estructurados

Temario

- Utilización de los modelos en el mundo real ✅
- Transformación del modelado a la base de datos ✅
- Clases de entidades
- Modelo relacional, restricciones de integridad y normalización ✅
- Arquitectura de los sistemas gestores de Bases de datos, OLTP, OLAP
- Lenguaje de consulta formal
- Lenguaje de consulta SQL

Logros

Capturar datos desde una pantalla de usuario real y almacenarlos en una base de datos. Manipulación de estos datos.

Restricciones de integridad

Las restricciones de integridad permiten agregar a las bases de datos consistencia, garantizando datos válidos, limpios lo que ayuda a evitar la duplicidad (esto lo enfatizaremos en la normalización). Las restricciones aseguran que las modificaciones a realizar a la base de datos no ocasionen pérdidas de información, la base de datos este protegida ante accidentes y este segura, ya que los cambios serán aplicados por usuarios autorizados y la misma base de datos aplicará bloqueos definidos en la creación de las tablas.

Tipos de restricciones de integridad:

- Declaración de claves (primarias, foráneas)
- Cardinalidad de la relación
- Participación mínima / máxima
- Integridad de dominio
- Integridad referencial
- Dependencias funcionales
- Dependencias multivaloradas

Declaración de claves primarias y foráneas

Las claves primarias permiten establecer un identificador único para cada tupla de la tabla, y consultar de manera eficiente por medio de ese registro. Adicional las claves foráneas establecen la relación entre una tabla padre y una hija.

```
CREATE TABLE "USUARIO" (
    cedula varchar(10) NOT NULL,
    nombre varchar(20),
    apellido varchar(20),
    nombre_usuario varchar(10),
    correo varchar(30),
    clave varchar(8),
    activo boolean,
    CONSTRAINT pk_usuario PRIMARY KEY (cedula)); -- DEFINICIÓN DE CLAVE PRIMARIA
```

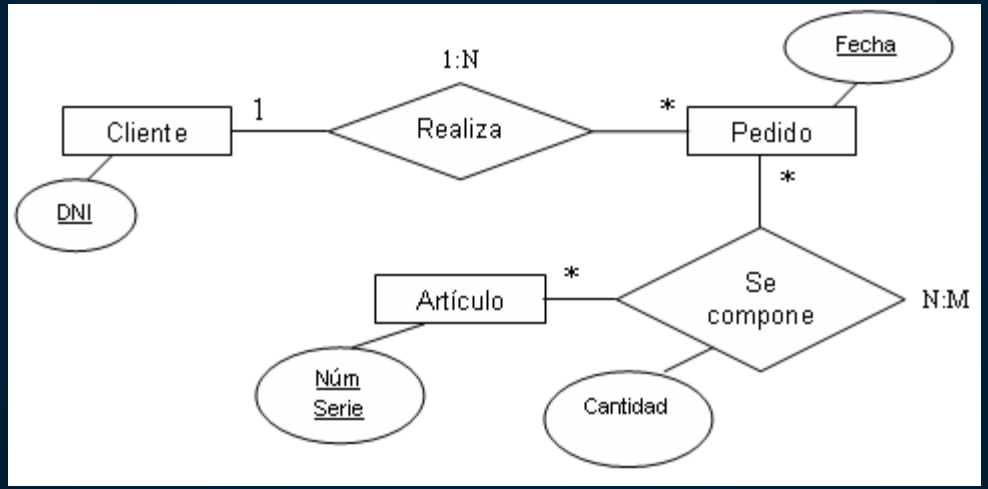
Declaración de claves primarias y foráneas

```
CREATE TABLE “FACTURA” (
    id_factura SERIAL NOT NULL,
    cedula varchar(10),
    precio int(20),
    ciudad varchar(10),
    metodo_pago varchar(30),
    CONSTRAINT pk_factura PRIMARY KEY (id_factura)
    CONSTRAINT fk_cedula FOREIGN KEY (cedula) REFERENCES "USUARIO" (cedula));
-- DEFINICIÓN DE CLAVE FORANEA, RELACION ENTRE ENTIDADES USUARIO Y FACTURA
```

Cardinalidad de la relación
Participación minima y máxima

Define en el modelo entidad relación o modelo relacional, si dos tablas se relacionarán de 1 a muchos, muchos a mucho o uno a muchos.

La participación está relacionada al número de asociaciones que puede tener una entidad (0,1), (1,1), (0,n), (1,n), (m,n)



Integridad de dominio

La integridad de dominio garantiza una forma de restricción simple, donde se establece para cada atributo una regla que permite o no posibles valores. Permite examinar valores inducidos a la base de datos y consultas.

Tipos:

- **NOT NULL**: No permite dejar registros vacíos en una columna, es obligatorio su valor. (Restricción de existencia)
- **UNIQUE**: Los valores de una columna no pueden repetirse dentro de la misma tabla.
- **CHECK**: Verificación del valor antes de ingresarlo, este debe cumplir con el valor o la regla definida (Restricción de unidad)

Integridad de dominio

NOT NULL:

```
CREATE TABLE “PERSON” (
    id int NOT NULL,
    name varchar(10) NOT NULL,
    last_name int(20));
```

UNIQUE:

```
CREATE TABLE “PERSON” (
    id int NOT NULL UNIQUE,
    name varchar(10),
    last_name int(20));
```

CHECK:

```
CREATE TABLE “PERSON” (
    id int NOT NULL,
    name varchar(10) NOT NULL,
    last_name int(20)
    age int
    CHECK (age=>18));
```

Integridad referencial

La integridad referencia se determina cuando las tablas se relacionan, indicando de que manera se verán afectadas las tablas asociadas cuando se elimina o se actualiza un dato.

En le caso de la integridad referencial es posible realizar 3 configuraciones:

- **RESTRICT**: No da permisos al update y al delete de modificar las relaciones (tablas dependientes)
- **CASCADE**: Si un registro es eliminado de la tabla principal, en la tabla dependiente también es eliminado, ocurre lo mismo para el caso de la actualización
- SET NULL: Permite eliminar filas de la tabla principal, poniendo en la tabla dependiente el valor de null para la llave foránea.

Integridad referencial

```
CREATE TABLE “FACTURA” (
    id_factura SERIAL NOT NULL,
    cedula varchar(10),
    precio int(20),
    ciudad varchar(10),
    metodo_pago varchar(30),
    CONSTRAINT pk_factura PRIMARY KEY (id_factura)
    CONSTRAINT fk_cedula FOREIGN KEY (cedula) REFERENCES "USUARIO" (cedula)
    ON DELETE CASCADE ON UPDATE RESTRICT); -- Integridad referencial
```

Integridad referencial

```
CREATE TABLE “FACTURA” (
    id_factura SERIAL NOT NULL,
    cedula varchar(10),
    precio int(20),
    ciudad varchar(10),
    metodo_pago varchar(30),
    CONSTRAINT pk_factura PRIMARY KEY (id_factura)
    CONSTRAINT fk_cedula FOREIGN KEY (cedula) REFERENCES "USUARIO" (cedula)
    ON DELETE SET NULL); -- Integridad referencial
```

Dependencias funcionales

Una dependencia funcional indica que un campo de una base de datos depende de otro para ser asignado su valor. Esto puede generar duplicidad en la información.

En este caso los datos de direcciónC y teléfonoC dependen del campo Centro, ya que según la información ingresada la dirección y el teléfono se repite, según el dato insertado en centro.

Id_empleado	Nombre_apellido	Puesto	Centro	DirecciónC	TelefónoC
12345	Carlos Crespo	Asistente	Informática	c/ Complutense	681993333
12346	Violeta	Profesor	Empresarial	c/ Otro	681994444
12347	Pepe Perez	Profesor	Informática	c/ Complutense	681993333

Dependencias funcionales

En las dependencias funcionales podemos encontrar **la satisfacción de dependencias funcionales**, lo que busca proponer algoritmos que permitan comprobar si algún dato depende de otro, en el caso de que sea necesario migrar las bases de datos.

Y una **dependencia trivial** es que no aporta ninguna restricción al esquema relacional, se puede garantizar a nivel semántico la equivalencia.

Dependencias multivaloradas

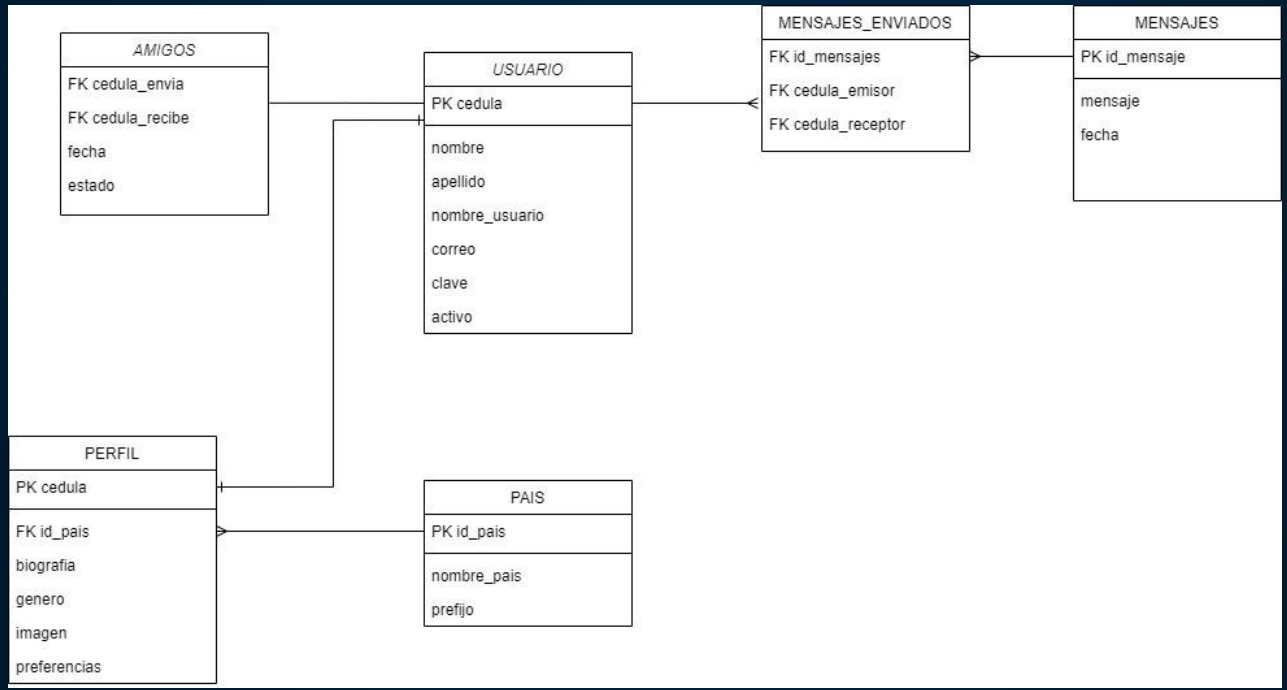
Las dependencias multivaloradas se refiere a relaciones que tienen atributos que no son expresables con las dependencias funcionales. Los atributos son dependientes entre sí pero tiene multiples combinaciones en la misma tabla creando duplicidad para validar todos los posibles escenarios. Ejemplo:

Nombre_apellido	Dirección	Telefóno
Violeta	C / calle1	681993333
Violeta	C / calle1	681993333
Violeta	C / calle2	681994444

La dependencias funcionales impiden que aparezcan ciertas tuplas, en el caso de las multivaloradas, es necesario repetir datos.

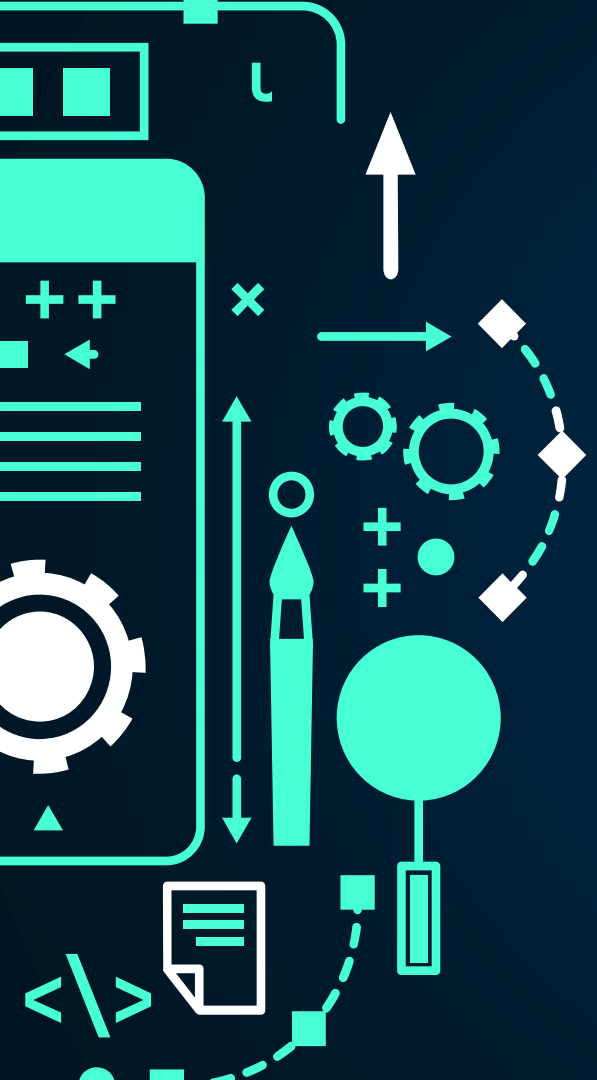
Definamos reglas de integridad

Sobre este modelo que ya conocemos y hemos trabajado en clases anteriores, vamos a listar todas las reglas de integración, si no están muy claras y es necesario crearlas, vamos a hacerlo. Modifica el diagrama como sea necesario, para que cumpla con las reglas de integridad vistas anteriormente.



REFERENCIAS

- Restricciones de integridad: http://gpd.sip.ucm.es/rafa/docencia/bdsi/apuntes/TEMA05.pdf
- Reglas de integridad: https://www.dataprix.com/es/book/export/html/511- El lenguaje SQL, Carme Martín Escofet
- Postgres Tutorial: https://www.postgresqltutorial.com/
- Libro: Fundamentos de base de datos 5 edición, Silberschatz



Gracias!