IIC2413

AYUDANTÍA DE REPASO

Repaso 11

ÁLGEBRA RELACIONAL

Proyección, selección, conjuntos y joins

PROYECCIÓN

Selecciona atributos (columnas)

π $a_1,\ldots,a_n$ (R) conserva los atributos indicados.

En el ejemplo, π nombre (Actores) entrega solo los nombres. Las columnas id y nacimiento quedan fuera.

actores

id	nombre	nacimiento
1	Leonardo DiCaprio	74
2	Matthew McConaughey	69
3	Daniel Radcliffe	89
4	Jessica Chastain	77

Liste el nombre de todos los actores:

$$\pi_{nombre}(actores)$$

nombre
Leonardo DiCaprio
Matthew McConaughey
Daniel Radcliffe
Jessica Chastain

SELECCIÓN

Filtra tuplas
(filas)

σ condición (R) conserva las tuplas que cumplen la condición.

Al filtrar por director = 'C. Nolan', quedan sus películas, con todos sus atributos.

peliculas

id	nombre	año	categoria	calificacion	director
1	Interstellar	2014	SciFi	8.6	C. Nolan
2	The Revenant	2015	Drama	8.1	A. Iñárritu
3	Harry Potter	2011	Fantasía	8.1	D. Yates
4	The Theory of Everything	2014	Biografía	7.7	J. Marsh
5	Inception	2010	Adventure	8.8	C. Nolan
...	...	...	...	...	

Liste todas las películas de C. Nolan:

$$\sigma_{director="C.Nolan"}(peliculas)$$

id	nombre	año	categoria	calificacion	director
1	Interstellar	2014	SciFi	8.6	C. Nolan
5	Inception	2010	Adventure	8.8	C. Nolan

OPERACIONES DE CONJUNTOS

Las relaciones deben tener esquemas compatibles

Un mismo esquema, tres resultados

R = {1, 2} S = {2, 3}

R ∪ S	R ∩ S	R − S
{1, 2, 3}	{2}	{1}

UNIÓN · R ∪ S
Tuplas que están en R, en S o en ambas.

INTERSECCIÓN · R ∩ S
Tuplas que están en ambas relaciones.

DIFERENCIA · R − S
Tuplas de R que no están en S.

Misma cantidad de atributos y tipos compatibles.

RENOMBRAR

Cambia nombres de atributos o relaciones

Renombrar conserva los datos

nombre	año
Interstellar	2014

→

name	year
Interstellar	2014

Cambian los encabezados; la fila se mantiene.

Permite cambiar nombres para usar una relación o sus atributos en otra expresión.

ATRIBUTOS
Por ejemplo: nombre → name,
año → year.

RELACIONES
Podemos nombrar Actores_jovenes al resultado de seleccionar actores menores de 30 años y consultarlo después.

PRODUCTO CRUZ Y JOIN

El join combina tuplas según una condición

El JOIN conserva las coincidencias

Película	id
Interstellar	1
Argo	2

id_película
1

→

Resultado
Interstellar

2 × 1 = 2 pares; con id = id_película queda 1.

PRODUCTO CRUZ · R × S
Combina cada tupla de R con cada tupla de S.

JOIN CON CONDICIÓN
R ⋈ condición S = σ condición (R × S)

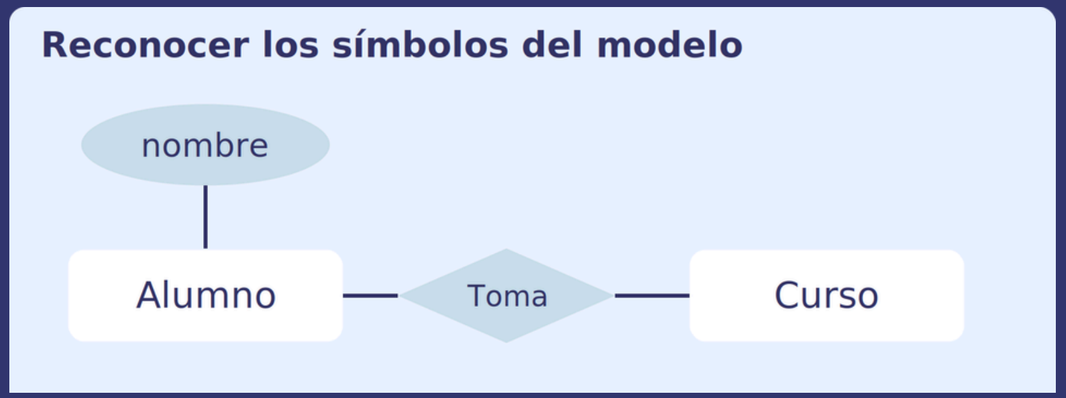
NATURAL JOIN
Usa los atributos con el mismo nombre para realizar la combinación.

MODELO E/R

Entidades, llaves y paso a tablas

MODELO E/R

Entidades, atributos
y relaciones



ENTIDADES → Rectángulos
Agrupan objetos con características comunes.

ATRIBUTOS → Óvalos
Describen las características de las entidades.

RELACIONES → Rombos
Representan asociaciones entre entidades.

La llave primaria se subraya; la parcial se representa con subrayado discontinuo.

TIPOS DE LLAVES

Identifican entidades
y conectan tablas

Primaria y foránea conectan tablas



La FK apunta a una llave de la tabla referida.

SUPERLLAVE
Atributos que determinan el resto de los datos.

CANDIDATA
Superllave minimal: no se le pueden quitar atributos y conservar esa propiedad.

PRIMARIA
Llave candidata elegida como identificador.

FORÁNEA · Referencia a otra tabla.
SURROGATE KEY · Identificador artificial, como un ID.

ENTIDADES A TABLAS

Cada entidad se transforma en una tabla

ENTIDAD FUERTE
Sus atributos pasan a ser columnas de la tabla.

ENTIDAD DÉBIL
Incluye la llave de la entidad de la que depende.

De entidades a esquema relacional

Entidad	Atributos del esquema
Cursos	Id_Curso (PK), Descripción
Tareas	Id_Curso (PK, FK), Id_Tarea (PK), ...

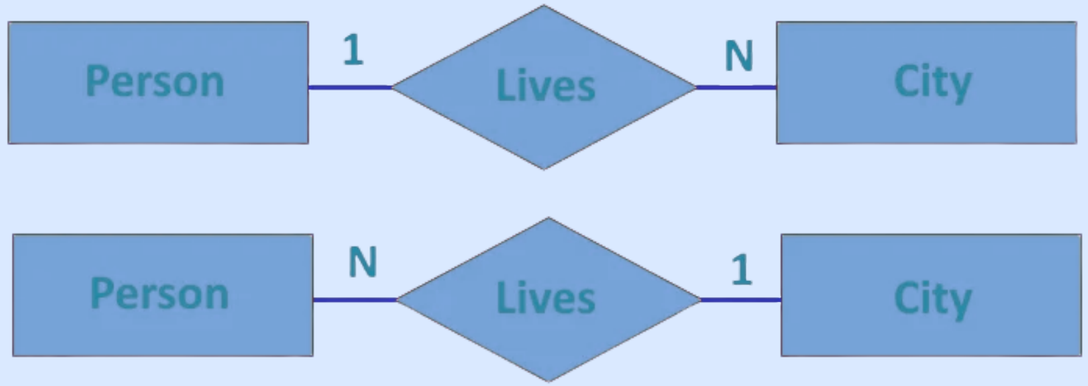
La llave de la entidad débil incorpora la del padre.

RELACIONES 1:N Y 1:1

La cardinalidad indica dónde ubicar la llave foránea

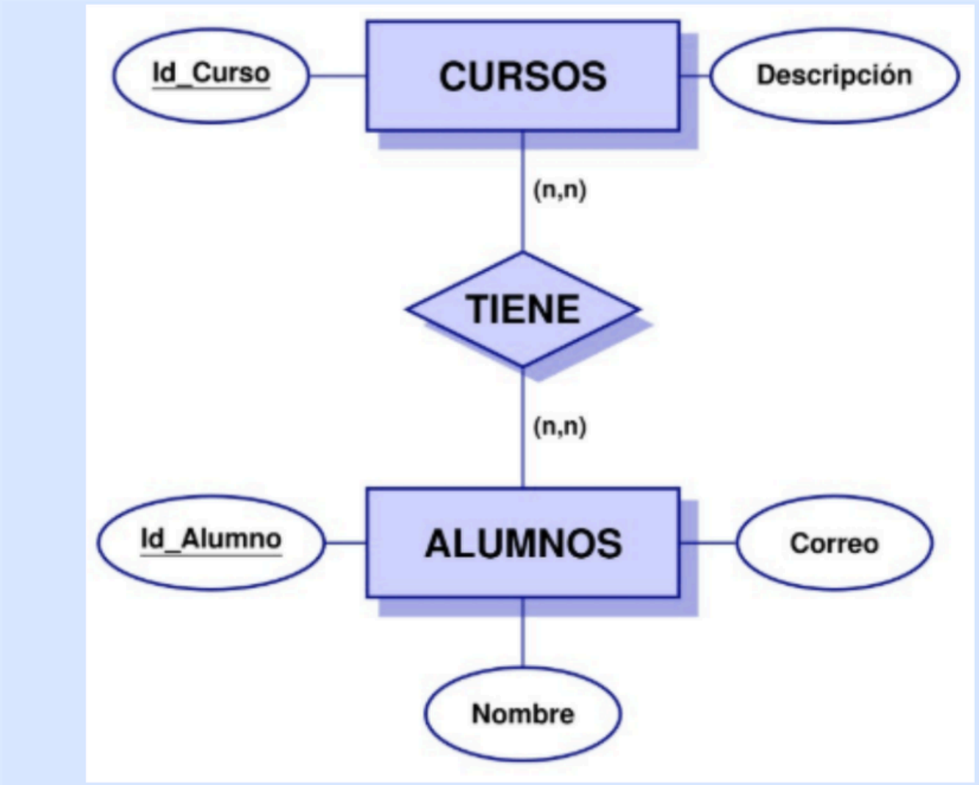
1:N SIN ATRIBUTOS
La llave foránea queda en el lado «muchos».

1:1 SIN ATRIBUTOS
Una entidad incorpora la llave de la otra.



RELACIONES N:M E ISA

N:M usa una tabla intermedia;
IsA hereda el identificador



N:M → Tiene(Id_Curso, Id_Alumno).

IsA → la entidad hija hereda el ID de Personal.
Profesores(Id_personal, Departamento)

NORMALIZACIÓN

Dependencias, anomalías y formas normales

ANOMALÍAS

La redundancia dificulta mantener datos consistentes

PedidoResumen(id_pedido, cliente, comuna, costo_envío)

id	cliente	comuna	costo_envío
501	Ana	Ñuñoa	3500
502	Bruno	Ñuñoa	3500
503	Camila	La Florida	4200
504	Diego	Ñuñoa	3500

INSERCIÓN: no se guarda el costo de una comuna sin pedidos.

ACTUALIZACIÓN: un cambio exige modificar varios pedidos.

ELIMINACIÓN: borrar el último pedido de La Florida elimina su costo.

DEPENDENCIAS FUNCIONALES

X → Y significa
«X determina Y»



Ejemplos de la ayudantía:

sku → nombre_producto, precio
patente → marca, modelo
sigla_curso → nombre, facultad

Si X determina todos los atributos, X es una superllave.

DESCOMPOSICIÓN
comuna → costo_envío permite separar:
Pedido(id_pedido, cliente, comuna)
CostoComuna(comuna, costo_envío)

REGLAS DE DEPENDENCIAS

Permiten obtener dependencias a partir de otras

TRANSITIVIDAD

$X \rightarrow Y, Y \rightarrow Z \Rightarrow X \rightarrow Z$

REFLEXIVIDAD

$Y \subseteq X \Rightarrow X \rightarrow Y$

DESCOMPOSICIÓN

$X \rightarrow Y,Z \Rightarrow X \rightarrow Y$

UNIÓN

$X \rightarrow Y$ y $X \rightarrow Z \Rightarrow X \rightarrow Y,Z$

AUMENTO

$X \rightarrow Y \Rightarrow X,Z \rightarrow Y,Z$

1NF Y 2NF

Un valor por celda;
sin dependencias parciales

1NF: un valor por celda.

2NF: mantiene 1NF y evita dependencias parciales.

1NF: separar los valores de una celda

playlist	canciones
10	A, B, C

→

playlist	canción
10	A
10	B
10	C

Una fila por canción; cada celda tiene un valor.

TERCERA FORMA NORMAL · 3NF

Revisa cada dependencia funcional no trivial

La dependencia transitiva del ejemplo



depto no es superllave → aquí se viola 3NF.

Para X → A, debe cumplirse al menos una condición:

- X es una superllave.
- A forma parte de alguna llave candidata.

Ejemplo violación: Empleado(rut, depto, jefe)
rut → depto, jefe
depto → jefe

Si rut es la única llave, depto → jefe viola 3NF: depto no es superllave y jefe no pertenece a una llave candidata.

BCNF

En cada dependencia no trivial, el determinante es superllave



DESCOMPOSICIÓN

1. Buscar una dependencia $X \rightarrow Y$ que viole BCNF.
2. Crear R_1 con $X \cup Y$.
3. Crear R_2 con los atributos restantes, conservando X.
4. Repetir si todavía hay violaciones.

Si A es el conjunto de atributos:
$R_2 = A - (Y - X)$

SIN PÉRDIDA

El join de las relaciones debe reconstruir la relación original.

SQL BÁSICO

Tablas, restricciones y consultas

SQL

Describe qué información queremos obtener

SQL ES DECLARATIVO
El DBMS transforma la consulta en un procedimiento de ejecución.

DDL · Define tablas, atributos y llaves.
DML · Manipula datos almacenados.
DQL · Consulta y recupera información.
DCL · Gestiona permisos de acceso.
TCL · Controla transacciones.

CREAR TABLAS

CREATE TABLE define
los atributos y sus tipos

```
CREATE TABLE Peliculas (
  id int PRIMARY KEY,
  nombre varchar(30),
  año int,
  categoria varchar(30),
  calificacion float,
  director varchar(30)
);
```

char: largo fijo · varchar: largo variable
int, smallint, float: números
time: hora · date: fecha
timestamp: fecha y hora

RESTRICCIONES Y ESTRUCTURA

Definimos llaves, valores por defecto y columnas

PRIMARY KEY (id_pelicula, id_actor)
Define una llave compuesta.

DEFAULT 'Acción'
Establece un valor por defecto.

NOT NULL
Impide almacenar nulos en el atributo.

```
ALTER TABLE Peliculas
ADD COLUMN productor varchar(30);

ALTER TABLE Peliculas
DROP COLUMN director;

DROP TABLE Peliculas;
```

INSERTAR, ACTUALIZAR Y BORRAR

WHERE determina
qué filas se modifican

```
INSERT INTO R(a, b)
VALUES (1, 1);

UPDATE Peliculas
SET calificacion = 0
WHERE nombre = 'Sharknado 6';

DELETE FROM R
WHERE a = 1;
```

Sin WHERE, UPDATE y DELETE afectan todas las filas.

Un punto y coma antes del WHERE termina la sentencia y deja el filtro fuera.

LLAVES FORÁNEAS EN SQL

Las referencias deben mantenerse válidas

Dos acciones al borrar una referencia

Acción	Fila que referencia
CASCADE	Se elimina
SET NULL	Se conserva con FK = NULL

FOREIGN KEY(a) REFERENCES R

No se puede insertar una referencia a una llave que no existe en R.

AL BORRAR UNA FILA REFERENCIADA

Sin una acción alternativa:
se rechaza la eliminación.

ON DELETE CASCADE
Elimina también las filas que la referencian.

ON DELETE SET NULL
Mantiene esas filas y deja su referencia en NULL.

SELECT, FROM Y WHERE

Columnas, relaciones
y condiciones

Qué aporta cada parte de la consulta

Parte	En el ejemplo
SELECT	nombre, calificacion
FROM	Peliculas
WHERE	director = 'C. Nolan'

```
SELECT nombre, calificacion
FROM Peliculas
WHERE director = 'C. Nolan';
```

SELECT * conserva todos los atributos.

WHERE permite comparar y combinar condiciones con AND, OR y NOT.

IN (1971, 1973, 2001)
Comprueba pertenencia a una lista.

BETWEEN 1971 AND 1978
Comprueba un intervalo.

En álgebra: π atributos (σ condición (R × S))

LIKE, DISTINCT Y ORDER BY

Buscar patrones, eliminar repetidos y ordenar

LIKE
% representa una secuencia de caracteres.
_ representa un solo carácter.

```
WHERE nombre LIKE '%Potter%'
```

DISTINCT

```
SELECT DISTINCT nombre
FROM Peliculas;
```

ORDER BY

```
ORDER BY nombre, calificacion
```

Ordena por nombre y deshace empates con la calificación.
DESC cambia a orden descendente.

JOINS Y ALIAS

Relacionamos tablas usando los atributos correspondientes

```
SELECT Peliculas.nombre, Actores.nombre
FROM Peliculas, Actuo_en, Actores
WHERE Peliculas.id = Actuo_en.id_pelicula
AND Actores.id = Actuo_en.id_actor;
```

JOIN EXPLÍCITO

```
FROM Peliculas JOIN Actuo_en
ON Peliculas.id = Actuo_en.id_pelicula
```

ALIAS

AS permite renombrar una tabla o el resultado de una expresión.

Calificar un atributo con su tabla evita ambigüedades.

Conjuntos en SQL

UNION, INTERSECT y EXCEPT

UNION y UNION ALL no son iguales

R = {1, 2} S = {2, 3}

UNION	UNION ALL
1, 2, 3	1, 2, 2, 3

UNION · Une resultados.
INTERSECT · Conserva los comunes.
EXCEPT · Conserva los del primero que no aparecen en el segundo.

UNION ALL conserva duplicados.

Ejemplo de clase:
SELECT nombre FROM Actores
UNION
SELECT director FROM Peliculas;

Las consultas deben entregar columnas compatibles.

CONSULTAS AVANZADAS

Agregación, subconsultas, NULL y joins externos

AGREGACIÓN Y COUNT

Las funciones de agregación resumen varias filas

Compra(producto, fecha, precio, cantidad)

producto	fecha	precio	cantidad
tomates	07/02	100	6
tomates	06/07	150	
zapallo	08/02	800	1
zapallo	09/07	1000	
zapallo	01/01	600	1

AVG, SUM, MIN, MAX y COUNT resumen filas.

En la tabla, los espacios vacíos son NULL.
COUNT(*) = 5
COUNT(cantidad) = 3
COUNT(DISTINCT cantidad) = 2

COUNT(*) cuenta filas; COUNT(atributo) omite sus nulos.

DISTINCT dentro de COUNT elimina duplicados antes de contar. SELECT DISTINCT COUNT(cantidad) devuelve 3.

GROUP BY Y HAVING

WHERE filtra filas;
HAVING filtra grupos

Agrupar y sumar: ejemplo de Compra

producto	Cálculo	ventaTotal
tomates	100×6 + 150×4	1200
zapallo	800×1 + 1000×2	2800

SUM(precio × cantidad), agrupado por producto.

```
SELECT fabricante, COUNT(fabricante)
FROM Productos
WHERE año > 2012
GROUP BY fabricante;
```

Primero se aplica FROM y WHERE. Después se agrupa por fabricante y se calcula el conteo en cada grupo.

HAVING impone una condición a los grupos, por ejemplo:
HAVING SUM(cantidad) > 100

En SELECT se incluyen atributos de agrupación y/o agregados.

CONSULTAS ANIDADAS

Una consulta puede usar el resultado de otra

Bandas cuyo vocalista es estudiante UC y que tocaron en Lollapalooza:

```
SELECT Bandas.nombre
FROM Bandas, Estudiantes_UC
WHERE Bandas.vocalista =
  Estudiantes_UC.nombre
AND Bandas.nombre IN (
  SELECT Toco_en.nombre_banda
  FROM Toco_en
  WHERE Toco_en.nombre_festival =
    'Lollapalooza'
);
```

La subconsulta entrega los nombres de las bandas que tocaron en ese festival.

IN, ALL, ANY Y EXISTS

El operador depende de qué retorna la subconsulta

Comparar con una subconsulta

Resultado de la subconsulta: {2, 5}

Condición	Resultado
3 < ANY (...)	TRUE: 3 < 5
3 < ALL (...)	FALSE: 3 < 2 es falso

IN · El valor pertenece al resultado.
ALL · La comparación se cumple para todos los valores.
ANY · Se cumple para al menos uno.
EXISTS · La subconsulta tiene alguna fila.

precio < ANY (SELECT precio FROM Cervezas)
Busca cervezas que no son las más caras.

Una consulta es monótona si agregar datos no reduce sus resultados.

SUBCONSULTAS CORRELACIONADAS

La consulta interna usa valores de la externa

Nombres de películas que aparecen en años diferentes:

```
SELECT p.nombre
FROM Peliculas AS p
WHERE p.año <> ANY (
  SELECT año
  FROM Peliculas
  WHERE nombre = p.nombre
);
```

p.nombre viene de la consulta externa. Por eso, la subconsulta está relacionada con la película que se está considerando.

SUBCONSULTAS EN FROM

El resultado intermedio
se usa como una relación

```
SELECT Actores.nombre, agg.cuenta
FROM Actores, (
  SELECT id_actor AS id,
    COUNT(*) AS cuenta
  FROM Actuo_en
  GROUP BY id_actor
) AS agg
WHERE Actores.id = agg.id;
```

La subconsulta cuenta las películas por actor.

Luego, el join incorpora el nombre de cada actor al resultado.

NULL Y LÓGICA DE TRES VALORES

SQL evalúa TRUE, FALSE y UNKNOWN

Qué ocurre al evaluar NULL

Expresión	Resultado
NULL = 3	UNKNOWN
NULL IS NULL	TRUE
3 + NULL	NULL

WHERE conserva solo las filas que dan TRUE.

NULL representa información ausente: desconocida, inexistente o cuya existencia no se conoce.

Una comparación como NULL = 3 no es TRUE ni FALSE: es UNKNOWN.

IS NULL e IS NOT NULL permiten revisar si falta un valor.

Una operación aritmética con NULL devuelve NULL.

Los agregados ignoran los nulos; COUNT(*) cuenta las filas completas.

NOT IN con NULL

Un nulo puede cambiar el resultado de la consulta

Ejemplo:
S.A = {1, 2, 3, 4}

```
SELECT S.A FROM S
WHERE S.A NOT IN (SELECT R.A FROM R);
```

Si R.A = {1, 2}, devuelve {3, 4}.
Si R.A = {1, 2, NULL}, devuelve una tabla vacía.

Para 3 y 4, la comparación con NULL produce UNKNOWN; la negación sigue siendo UNKNOWN y la fila no se conserva.

OUTER JOINS

Conservan tuplas
que no tienen pareja

Estudios

nombre	titulo
Warner	Argo
Warner	El Origen
MGM	El Hobbit

Peliculas

titulo	ingreso
Argo	136
El Origen	292
El Artista	44

INNER JOIN: solo coincidencias.
LEFT: conserva también las filas de la izquierda sin pareja.
RIGHT: las de la derecha.
FULL: las de ambos lados.

Los atributos sin correspondencia quedan en NULL.

En el ejemplo, el LEFT JOIN devuelve Warner con 428 y MGM con NULL. Con INNER JOIN, MGM desaparece.

VISTAS

Crear, consultar y eliminar vistas

VISTAS

Una consulta guardada
con un nombre

Del detalle al promedio de la vista

The Prestige · Nolan	eval
The Observer	8.3
Uncut	8.5
Rolling Stone	8.7

→ promedio **8.5**

```
CREATE VIEW PelEval AS
SELECT pelicula, director,
  AVG(eval) AS promedio
FROM Evaluaciones
GROUP BY pelicula, director;
```

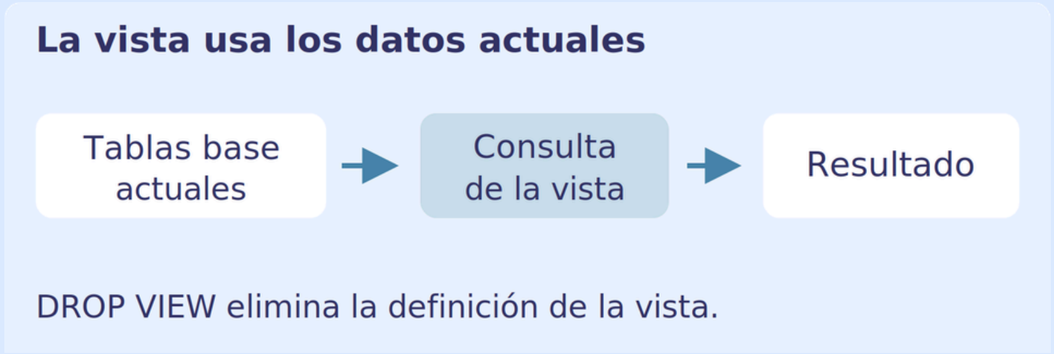
La vista define una tabla virtual.

Guarda la consulta, no una tabla física con el resultado.

En el ejemplo, PelEval permite reutilizar el cálculo del promedio de evaluaciones por película y director.

CONSULTAR Y ELIMINAR VISTAS

La consulta se ejecuta sobre las tablas base




```
SELECT promedio
FROM PelEval
WHERE pelicula = 'The Prestige'
AND director = 'Christopher Nolan';
```

El sistema reemplaza la vista por su definición y ejecuta la consulta resultante.

Si cambian las evaluaciones, la próxima consulta usa esos datos actualizados.

```
DROP VIEW PelEval;
```

Elimina la definición de la vista.

Las vistas reducen la repetición y simplifican consultas frecuentes.

EJERCICIOS

Una página web dedidaca a las recetas, tiene el siguiente modelo de datos que se muestra a continuación. El modelo tiene información sobre **País**, **Plato**, **Ingrediente**, **RecetaIngrediente**, **Trago** y **TragoIngrediente**.

id_pais	nombre
1	Chile
2	México
3	España

Cuadro 1: País

id_plato	nombre	dificultad	porciones	id_pais_origen
1	Cazuela de vacuno	Media	6	1
2	Tacos al pastor	Media	4	2
3	Tortilla de patatas	Baja	4	3

Cuadro 2: Plato

id_ingrediente	nombre
1	Carne de vacuno
2	Papas
3	Zapallo
4	Arroz
5	Cerdo marinado
6	Piña
7	Cebolla
8	Huevo
9	Aceite de oliva
10	Vino pipeño
11	Helado de piña
12	Tequila
13	Triple sec
14	Jugo limón
15	Vino tinto cartoné
16	Naranja
17	Manzana

Cuadro 3: Ingrediente

id_plato	id_ingrediente	cantidad	unidad
1	1	500	gramos
1	2	3	unidades
1	3	200	gramos
1	4	100	gramos
2	5	400	gramos
2	6	100	gramos
2	7	1	unidad
3	2	2	unidades
3	8	4	unidades
3	7	1	unidad
3	9	50	ml

Cuadro 4: PlatoIngrediente

id_trago	nombre	tipo	dificultad	porciones	id_pais_origen
1	Terremoto	Cóctel	Media	2	1
2	Margarita	Cóctel	Media	1	2
3	Sangría	Refrescante	Baja	4	3

Cuadro 5: Trago

id_trago	id_ingrediente	cantidad	unidad
1	10	200	ml
1	11	1	bola
2	12	45	ml
2	13	30	ml
2	14	20	ml
3	15	300	ml
3	16	1	unidad
3	17	1	unidad

Cuadro 6: TragoIngrediente

Escriba las siguientes operaciones en algebra relacional, utilizando la sintaxis vista en clases

Escriba las siguientes operaciones en algebra relacional, utilizando la sintaxis vista en clases

a) ¿Cuáles son los nombres de los platos de dificultad media?

Escriba las siguientes operaciones en algebra relacional, utilizando la sintaxis vista en clases

a) ¿Cuáles son los nombres de los platos de dificultad media?

$\pi_{nombre}(\sigma_{dificultad='Media'}(Plato))$

Escriba las siguientes operaciones en algebra relacional, utilizando la sintaxis vista en clases

b) ¿Que ingredientes se utilizan en platos originarios de México?

Escriba las siguientes operaciones en algebra relacional, utilizando la sintaxis vista en clases

b) ¿Que ingredientes se utilizan en platos originarios de México?

$$\pi_{\text{nombre}}\Big(\text{Ingrediente} \bowtie_{\text{id_ingrediente}=\text{PlatoIngrediente.id_ingrediente}} \Big(\text{PlatoIngrediente} \bowtie_{\text{id_plato}=\text{Plato.id_plato}} \Big(\sigma_{\text{id_país_origen}=2}(\text{Plato}) \Big) \Big) \Big)$$

Escriba las siguientes operaciones en algebra relacional, utilizando la sintaxis vista en clases

c) ¿Que paıses tienen al menos un plato que utiliza el ingrediente cebolla?

Escriba las siguientes operaciones en algebra relacional, utilizando la sintaxis vista en clases

c) ¿Que paises tienen al menos un plato que utiliza el ingrediente cebolla?

$$\pi_{\text{nombre}}\Big(\text{País} \bowtie_{\text{id_país}=\text{Plato.id_país_origen}} \Big(\text{Plato} \bowtie_{\text{id_plato}=\text{PlatoIngrediente.id_plato}} \Big(\text{PlatoIngrediente} \bowtie_{\text{id_ingrediente}=\text{Ingrediente.id_ingrediente}} \Big(\sigma_{\text{nombre}='\text{Cebolla}'}(\text{Ingrediente}) \Big) \Big) \Big) \Big)$$

Escriba las siguientes operaciones en algebra relacional, utilizando la sintaxis vista en clases

d) Liste todos los ingredientes que se usan en recetas platos y tragos de dificultad media.

Escriba las siguientes operaciones en algebra relacional, utilizando la sintaxis vista en clases

d) Liste todos los ingredientes que se usan en recetas platos y tragos de dificultad media.

$$\pi_{\text{nombre}}(\text{Ingrediente} \bowtie_{\text{id_ingrediente}=\text{PlatoIngrediente.id_ingrediente}} (\text{PlatoIngrediente} \bowtie_{\text{id_plato}=\text{Plato.id_plato}} (\sigma_{\text{dificultad}='\text{Media}'}(\text{Plato}))))$$

$$\cup$$

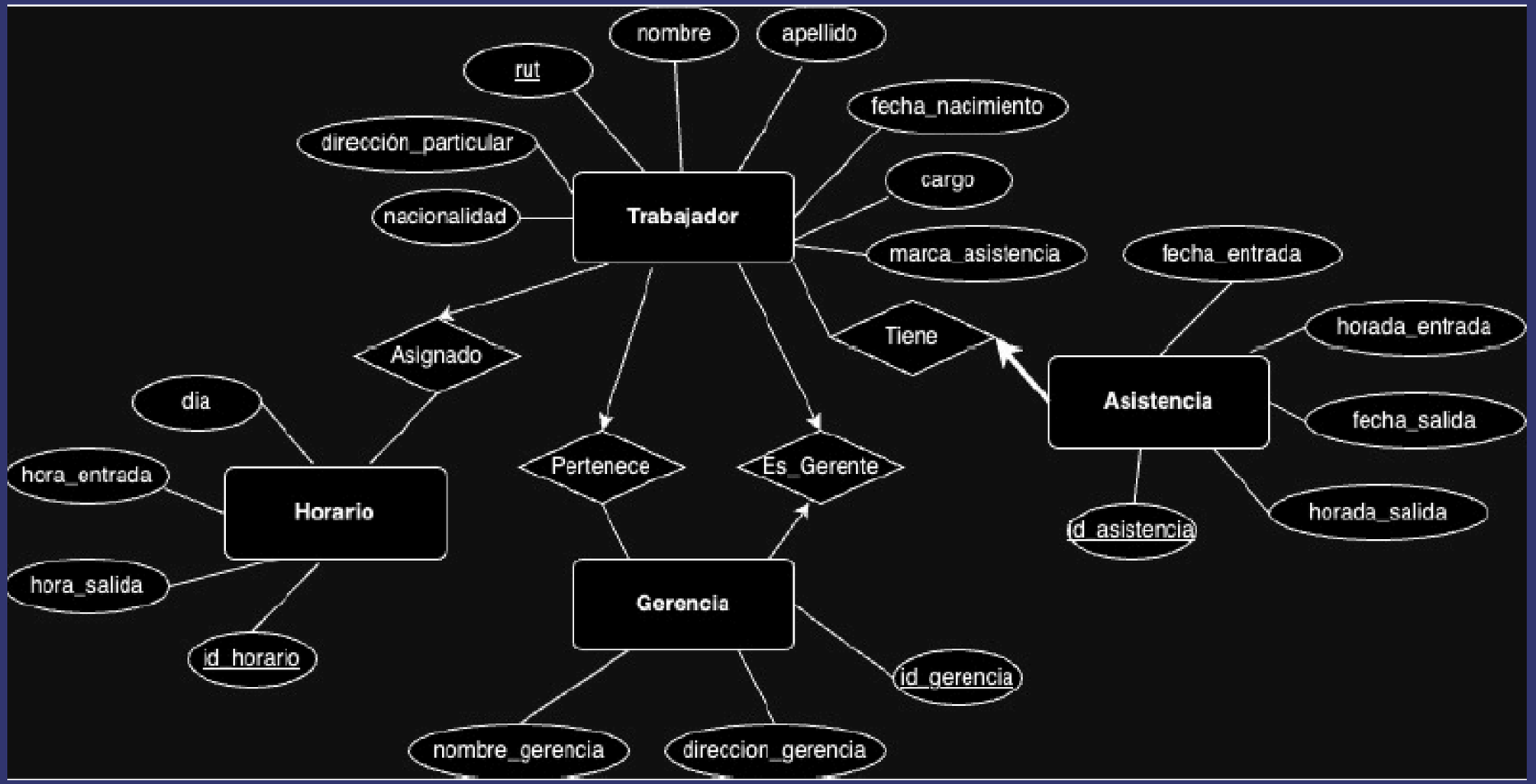
$$\pi_{\text{nombre}}(\text{Ingrediente} \bowtie_{\text{id_ingrediente}=\text{TragoIngrediente.id_ingrediente}} (\text{TragoIngrediente} \bowtie_{\text{id_trago}=\text{Trago.id_trago}} (\sigma_{\text{dificultad}='\text{Media}'}(\text{Trago}))))$$

Estás trabajando como consultor para una cadena de comida rápida, que desea automatizar el proceso de registro de asistencia de sus trabajadores. Tu labor como consultor será diseñar la base de datos de asistencia, la cuál debe cumplir los siguientes requisitos:

- Por exigencia de la Dirección del Trabajo se necesitan los siguientes datos de los trabajadores: rut, nombre, apellido, fecha de nacimiento, cargo, nacionalidad y dirección particular.
- Gerencia a la que pertenece el trabajador, el gerente a cargo y dirección de la gerencia.
- Horario asignado al trabajador, el cual se define por día de la semana, hora de entrada y salida (varios trabajadores pueden tener el mismo horario)
- Para cada trabajador que debe registrar asistencia (no todos los trabajadores marcan asistencia), se deben registrar los siguientes datos de su asistencia: fecha de entrada, hora de entrada, fecha de salida y hora de salida.

Basado en el enunciado anterior, responda las siguientes preguntas:

Desarrolle el modelo entidad-relacion para la aplicacion de asistencia y su esquema,
especificando el tipo de datos.


nombre
apellido
rut
fecha_nacimiento
dirección_particular
cargo
Trabajador
nacionalidad
marca_asistencia
fecha_entrada
Tiene
horada_entrada
Asignado
Asistencia
dia
fecha_salida
Pertenece
Es_Gerente
hora_entrada
Horario
horada_salida
id_asistencia
hora_salida
Gerencia
id_horario
id_gerencia
nombre_gerencia
direccion_gerencia

Genere el codigo sql para crear las tablas del esquema de la base de datos. Debe
especificar las llaves primarias y foraneas.

```
CREATE TABLE trabajador(
rut int,
nombre varchar(50),
apellido varchar(50),
fecha_nacimiento date,
cargo varchar(50),
direccion_particular varchar(100),
nacionalidad varchar(50),
marca_asistencia boolean,
id_horario int,
id_gerencia int,
PRIMARY KEY(rut),
FOREIGN KEY(id_horario) REFERENCES horario(id_horario),
FOREIGN KEY(id_gerencia) REFERENCES gerencia(id_gerencia)
)
```

```
CREATE TABLE gerencia(
id_gerencia int,
nombre_gerencia varchar(30),
direccion_gerencia varchar(100),
rut_gerente int,
PRIMARY KEY(id_gerencia),
FOREIGN KEY(rut_gerente) REFERENCES trabajador(rut)
)
CREATE TABLE horario(
id_horario int,
dia varchar(10),
hora_entrada time,
hora_salida time,
PRIMARY KEY(id_horario, dia)
)
CREATE TABLE asistencia(
id_asistencia int,
fecha_entrada date,
hora_entrada time,
fecha_salida date,
hora_salida time,
rut_trabajador int,
PRIMARY KEY(id_asistencia),
FOREIGN KEY(rut_trabajador) REFERENCES trabajador(rut)
```

Escribir las siguientes consultas SQL

Escribir las siguientes consultas SQL

Lista de todos los trabajadores que tienen que marcar asistencia y tengan un turno de 40 horas

Escribir las siguientes consultas SQL

Lista de todos los trabajadores que tienen que marcar asistencia y tengan un turno de 40 horas

```
SELECT rut, nombre, apellido
FROM trabajador, horario
WHERE trabajador.id_horario = horario.id_horario
AND trabajador.marca_asistencia = TRUE
GROUP BY horario.id_horario, rut, nombre, apellido
HAVING SUM(hora_salida - hora_entrada)=40
```

Escribir las siguientes consultas SQL

Cantidad de trabajadores por cada gerencia, que hayan faltado un dıa o mas durante el ultimo mes

Escribir las siguientes consultas SQL

Cantidad de trabajadores por cada gerencia, que hayan faltado un dıa o mas durante el ultimo mes

```
SELECT nombre_gerencia, COUNT(trabajador.rut)
FROM trabajador as T, gerencia, asistencia
WHERE trabajador.id_gerencia = gerencia.id_gerencia
AND trabajador.rut = asistencia.rut
AND trabajador.marca_asistencia = TRUE
AND fecha_entrada between ’01/03/2024’ and ’31/03/2024’
AND asistencia.hora_entrada IS NULL
GROUP BY gerencia.id_gerencia, gerencia.nombre_gerencia
```

Escribir las siguientes consultas SQL

Listado de todos los trabajadores que hayan trabajado más horas que las especificadas
en su turno (Nota: la validación se realiza por cantidad de horas semanal)

Escribir las siguientes consultas SQL

```
SELECT rut, nombre, apellido
FROM trabajador as T, horario
WHERE trabajador.id_horario = horario.id_horario
AND trabajador.marca_asistencia = TRUE
GROUP BY horario.id_horario, rut, nombre, apellido
HAVING SUM(hora_salida - hora_entrada) < (
SELECT SUM(hora_salida - hora_entrada)
FROM asistencia
WHERE rut_trabajador = T.rut
AND fecha_entrada BETWEEN ’8/04/2024’ AND ’15/4/2024’
GROUP BY rut_trabajador)
```

Mucho éxito en la prueba!