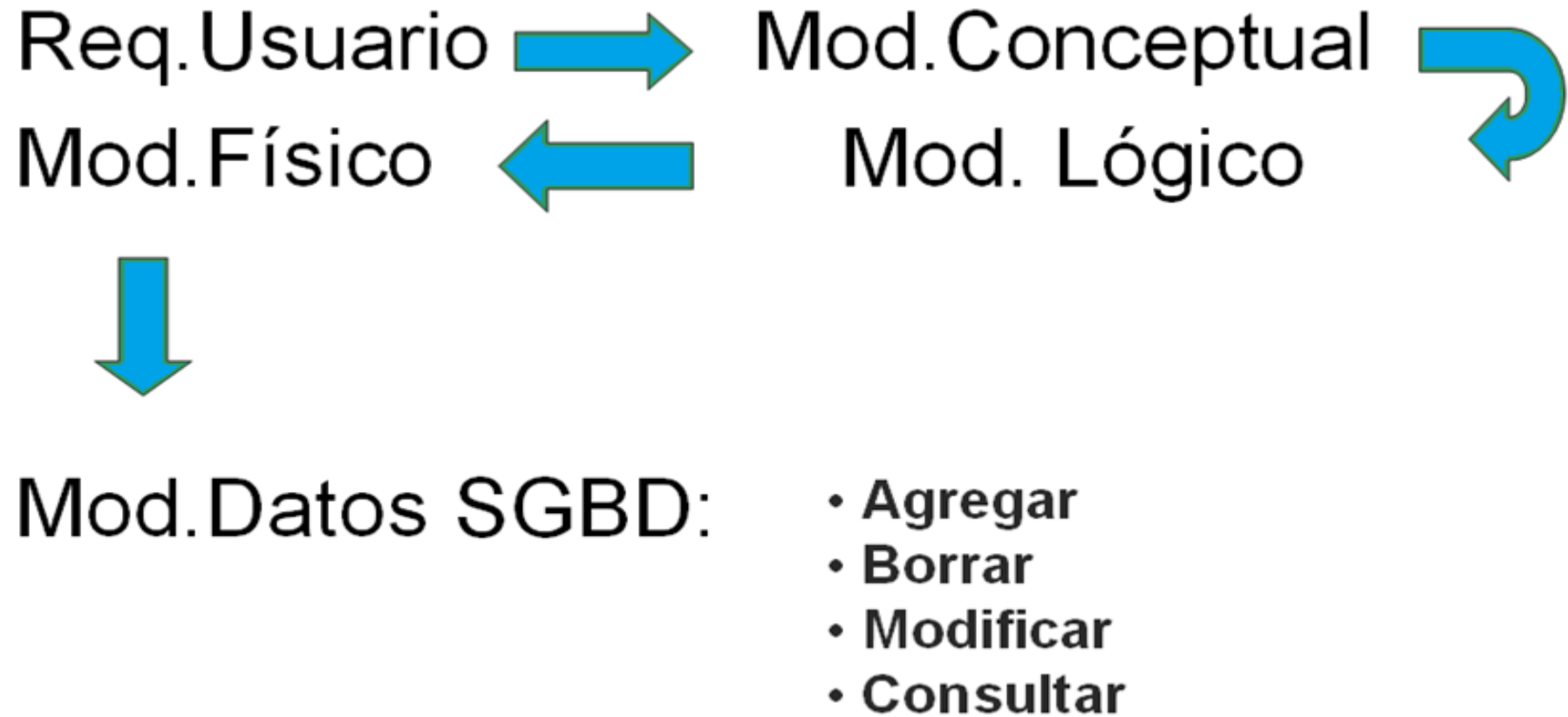


BASES DE DATOS

The background features several overlapping, wavy lines in orange, light blue, and lime green, creating a dynamic and modern aesthetic.

CLASE 5

Lenguajes de consulta



Lenguajes de consulta

- Los **lenguajes de consulta** son utilizados para operar con la base de datos
- Existen dos tipos de lenguajes de consulta:
 - **Procedurales**: dan instrucciones para realizar una secuencia de operaciones → **QUÉ** y **CÓMO**
 - **No procedurales**: solicitan directamente la información deseada → **QUÉ**

Álgebra Relacional

- El **Álgebra Relacional** (AR) es un lenguaje de consulta procedural que fue propuesto originalmente por Edgar F. Codd en el año 1970
- Proporciona una forma teórica y formal de **realizar consultas y manipular datos** en sistemas de gestión de bases de datos relacionales
- Las operaciones del AR trabajan sobre las relaciones, es decir, conjuntos de tuplas

Álgebra Relacional

- Utiliza operaciones de una o dos relaciones de entrada, que generan una **nueva relación como resultado**
- Las operaciones **fundamentales** son:
 - **Unarias**
 - Selección, Proyección, Renombre
 - **Binarias**
 - Producto Cartesiano, Unión, Diferencia

Álgebra Relacional

- Definición **AR**
 - Una **expresión básica** consta de:
 - Relación (tabla) de una BD ó
 - Relación constante (Ej: en una inserción, la tupla a insertar expresada literalmente)
 - Una **expresión general** se construye a partir de subexpresiones (E_1, E_2, \dots, E_N)

Álgebra Relacional

- Definición **AR**

- Expresiones válidas:**

- $E_1 \cup E_2$

- $E_1 - E_2$

- $E_1 \times E_2$

- $\sigma_P(E_1)$ \rightarrow **P** predicado con atributos en E_1

- $\pi_S(E_1)$ \rightarrow **S** lista de atributos de E_1

- $\rho_X(E_1)$ \rightarrow **X** nuevo nombre de E_1

Álgebra Relacional

PRESTAMO = (idPrestamo, idSucursal, idTitular, monto)

- **Selección** → Operador σ
 - Selecciona tuplas que **satisfacen un predicado** dado
 - El predicado puede tener conectivos lógicos (**and**, **or**, **not**) y operadores de comparación $<>$, $>$, $<$, $>=$, $<=$, $=$
 - **Ej1**: prestamos otorgados por la sucursal con id 526.
 - **Ej2**: prestamos otorgados por la sucursal con id 526 y con monto superior a \$120.000.

Álgebra Relacional

PRESTAMO = (idPrestamo, idSucursal, idTitular, monto)

- **Proyección** \rightarrow Operador π
 - Devuelve la relación argumento con columnas omitidas. Si quedan **tuplas repetidas se excluyen**
 - **Ej3:** id de las sucursales que dieron algún préstamo.
 - **Ej4:** id de las sucursales que dieron algún préstamo cuyo monto sea superior a \$50.000.

Álgebra Relacional

CLIENTE = (idCte, nombreCte, dni, domicilio, idLoc)

LOCALIDAD = (idLoc, nombreLoc)

- **Producto Cartesiano** → Operador **x**
 - Conecta dos entidades de acuerdo a la definición matemática de la operación
 - **Ej5:** cada cliente con sus datos personales y nombre de la localidad en donde vive.
 - **Ej6:** nombre y dni de todos los clientes que vivan en la localidad de “La Plata”.

Álgebra Relacional

CLIENTE = (idCte, nombreCte, dni, domicilio, idLoc)

LOCALIDAD = (idLoc, nombreLoc)

- **Renombre** → Operador ρ
 - Permite utilizar la misma tabla (por ej.) en un producto cartesiano
 - **Ej7**: nombre de los clientes que viven en el mismo domicilio que el cliente con id 654.

Álgebra Relacional

CLIENTE = (idCte, nombreCte, dni, domicilio, idLoc)

SUCURSAL = (idSuc, nombreSuc, domicilio, tel)

CC = (idCte, idSuc, saldo)

CA = (idCte, idSuc, saldo)

- **Unión** → Operador \cup
 - Equivalente a la unión matemática → las instancias repetidas se eliminan automáticamente
 - Las dos tablas deben ser unión-compatibles
 - Igual cantidad de atributos
 - El i-ésimo atributo de 1ª tabla y el i-ésimo atributo de 2ª tabla deben tener el mismo dominio (i:1..n)

Álgebra Relacional

CLIENTE = (idCte, nombreCte, dni, domicilio, idLoc)

SUCURSAL = (idSuc, nombreSuc, domicilio, tel)

CC = (idCte, idSuc, saldo)

CA = (idCte, idSuc, saldo)

- **Unión** → Operador \cup
 - **Ej8:** nombre de clientes que tienen CC o CA en la sucursal con id 721.

Álgebra Relacional

CLIENTE = (idCte, nombreCte, dni, domicilio, idLoc)

SUCURSAL = (idSuc, nombreSuc, domicilio, tel)

CC = (idCte, idSuc, saldo)

CA = (idCte, idSuc, saldo)

- **Diferencia** → Operador -
 - Equivalente a la diferencia de conjuntos
 - Las dos tablas deben ser **unión-compatibles**
 - **Ej9:** nombre de clientes de la sucursal con id 721 que tienen CC pero no tienen CA

Álgebra Relacional

- Operaciones adicionales de AR
 - No añaden potencia, sino que sólo **simplifican las consultas** realizadas con las operaciones básicas
 - Intersección
 - Producto Theta
 - Producto Natural
 - División
 - Asignación


Álgebra Relacional

CLIENTE = (idCte, nombreCte, dni, domicilio, idLoc)

SUCURSAL = (idSuc, nombreSuc, domicilio, tel)

CC = (idCte, idSuc, saldo)

CA = (idCte, idSuc, saldo)

- **Intersección** → Operador 
 - Equivalente a la intersección matemática
 - Las dos tablas deben ser **unión-compatibles**
 - Se puede definir en función de la **unión** y la **diferencia**
- **Ej10**: nombre de clientes de la sucursal con id 721 que tienen CC y también poseen CA

Álgebra Relacional

- **Producto Theta** \rightarrow Operador \bowtie_{θ}
 - Realiza el producto cartesiano teniendo en cuenta la condición (θ) especificada
 - Combinación de **selección** y **producto cartesiano**
 - Mantiene las columnas repetidas
 - $R \bowtie_{\theta} S = \sigma_{\theta} (R \times S)$

Álgebra Relacional

- **Producto Natural** \rightarrow Operador **|x|**
 - Realiza el producto cartesiano con una selección de tuplas “con sentido” eliminando las columnas repetidas
 - El resultado son las tuplas donde los nombres y valores de los atributos que se repiten en ambas tablas son iguales
 - Si coincide más de un atributo entre las tablas, el |x| se realiza por la coincidencia de todos los atributos comunes a la vez
 - Si **R** y **S** son dos relaciones que no tienen atributos en común $\rightarrow R \times S = R |x| S$

Álgebra Relacional

CLIENTE = (idCte, nombreCte, dni, domicilio, idLoc)

SUCURSAL = (idSuc, nombreSuc, domicilio, tel)

PRESTAMO = (idSuc, idCte, monto)

- **Ej11:** nombre de clientes con préstamos en alguna sucursal
- **Ej12:** nombre de clientes con préstamos en la sucursal con id 53

PERSONA = (dni, nombre, direccion, codCiudad)

CIUDAD = (codCiudad, nombre)

- **Ej13:** personas que viven en la ciudad con nombre 'La Plata'

Álgebra Relacional

ALUMNO = (nroAlumno, apellido, nombre, dni, fechaNac)

CURSO = (codCurso, nombreCurso, añoCurso, fechaInicio)

INSCRIPTO = (nroAlumno, codCurso)

- **División** → Operador %
 - Dado **R** y **S**, el resultado son los valores de atributos de **R**, que se relacionan con **todas las tuplas de S**
 - **R % S** sii Esquema de **S** está incluido en el Esquema de **R**
 - $\text{Esq}(\mathbf{R \% S}) = \text{Esq}(\text{Esq}(\mathbf{R}) - \text{Esq}(\mathbf{S}))$
 - **Ej14**: número de alumno, apellido y nombre de los alumnos que están inscriptos en todos los cursos existentes

Álgebra Relacional

CLIENTE = (idCte, nombreCte, dni, domicilio, idLoc)

SUCURSAL = (idSuc, nombreSuc, domicilio, tel)

CC = (idCte, idSuc, saldo)

CA = (idCte, idSuc, saldo)

- **Asignación** \rightarrow Operador \leftarrow
 - Expresión que asigna a una **variable temporal** el resultado de una operación
 - $\text{varTemporal} \leftarrow (\text{Expresión AR})$
 - **Ej15**: nombre de clientes de la sucursal con id 721 que tienen CC pero no tienen CA

Álgebra Relacional

- Operaciones de actualización
 - Agregar tuplas
 - $R \leftarrow R \cup E$ (R relación, E nueva tupla)
 - Agregar un cliente
 - $CLIENTE \leftarrow CLIENTE \cup (591, \text{"Federico Gomez"}, \text{"45.010.203"}, \text{"Calle 15 Nro 567"}, 17)$

Álgebra Relacional

- Operaciones de actualización
 - Eliminar tuplas
 - $R \leftarrow R - E$ (R relación, E tupla a eliminar)
 - Eliminar un cliente
 - $CLIENTE \leftarrow CLIENTE - (\sigma_{idCte = 591} (CLIENTE))$

Álgebra Relacional

- Operaciones de actualización
 - Modificar datos
 - $\delta_{A \leftarrow E} (R)$
 - Modificar el saldo de la CC de un cliente
 - $\delta_{\text{saldo} \leftarrow \text{saldo} + 100.000} (\sigma_{\text{idCte} = 591} (CC))$

Cálculo Relacional de Tuplas

- El **Cálculo Relacional de Tuplas** (**CRT**) es un lenguaje de consulta no procedural
- Describe la información deseada, como combinarla y filtrarla, pero sin dar un proceso específico para obtenerla
- Utiliza el **cálculo de predicados** para la formulación de consultas
- Expresión de consultas
 - $\{ \mathbf{t} / \mathbf{P}(\mathbf{t}) \} \rightarrow$ el conjunto de tuplas **t** tal que el predicado **P** es verdadero en **t**

Cálculo Relacional de Tuplas

- Definición formal de CRT
 - Dada una expresión de CRT $\{ t / P(t) \}$:
 - P = fórmula donde aparecen una o más variables de tupla
 - t puede ser:
 - Variable límite: $t \in tabla$
 - Variable libre: $\{ t / \exists s / s \in tabla \dots \}$

Cálculo Relacional de Tuplas

- Definición formal de CRT
 - Las fórmulas se componen de átomos:
 - $s \in R$; s = variable de tupla, R = relación (tabla)
 - $s[x] \ominus u[y]$;
 - s, u = variables de tupla
 - x, y = atributos sobre s, u respectivamente
 - \ominus = operador ($>$, $<$, $>=$, $<=$, $=$, $<>$, etc.)
 - $s[x] \ominus c$; c constante

Cálculo Relacional de Tuplas

- Definición formal de CRT
 - Las fórmulas se construyen a partir de la combinación de componentes, utilizando los operadores lógicos:
 - Un **átomo** es una fórmula
 - **P1** fórmula $\Rightarrow \sim$ **P1** fórmula
 - **P1**, **P2** fórmulas \Rightarrow **P1** \vee **P2**, **P1** \wedge **P2** fórmulas
 - **P1**(**s**) fórmula que contiene la variable de tupla libre **s** y **R** es una relación:
 - $\forall \mathbf{s} \in \mathbf{R}(\mathbf{P1}(\mathbf{s}))$ fórmula
 - $\exists \mathbf{s} \in \mathbf{R}(\mathbf{P1}(\mathbf{s}))$ fórmula

Cálculo Relacional de Tuplas

- Definición formal de CRT
 - Seguridad de expresiones
 - Para que una expresión en CRT sea válida, debe ser segura
 - Expresión no segura: $\{ t / \sim (t \in \text{prestamo}) \}$
 - Se está solicitando todas las tuplas que podrían estar definidas en la tabla pero que no lo están → posible número infinito de tuplas

Cálculo Relacional de Tuplas

ALUMNOS = (idAlumno, nombre, dni, idLocalidad, idCarrera)

MATERIAS = (idMateria, nombre, añoCurso, idCarrera)

CARRERAS = (idCarrera, nombre, duración)

INSCRIPCIONES = (idInscripcion, idAlumno, idMateria, año, nota)

LOCALIDADES = (idLocalidad, nombre)

- **Consultas básicas** (sólo con variables de tupla límite)
 - **Ej16:** listar todas las tuplas de ALUMNOS
 - **Ej17:** listar todas las materias de 1° año (de cualquier carrera)

Cálculo Relacional de Tuplas

ALUMNOS = (idAlumno, nombre, dni, idLocalidad, idCarrera)

MATERIAS = (idMateria, nombre, añoCurso, idCarrera)

CARRERAS = (idCarrera, nombre, duración)

INSCRIPCIONES = (idInscripcion, idAlumno, idMateria, año, nota)

LOCALIDADES = (idLocalidad, nombre)

- **Proyección:** $\{t / \exists s \in R / Q(s)\}$
 - **s** es una **variable de tupla limite**, esta ligada a la tabla **R**
 - **t** es una **variable de tupla libre**, no está ligada a una tabla, y se define solo para los atributos que se desea mostrar
 - **Ej18:** presentar el nombre de todas las carreras disponibles
 - **Ej19:** mostrar el nombre de las carreras que duren más de 3 años

Cálculo Relacional de Tuplas

ALUMNOS = (idAlumno, nombre, dni, idLocalidad, idCarrera)

MATERIAS = (idMateria, nombre, añoCurso, idCarrera)

CARRERAS = (idCarrera, nombre, duración)

INSCRIPCIONES = (idInscripcion, idAlumno, idMateria, año, nota)

LOCALIDADES = (idLocalidad, nombre)

- **Producto**

- Una **variable de tupla límite** por cada tabla involucrada
- A la **variable libre** es posible asignarle más de un valor
- **Ej20:** presentar el nombre de cada materia junto al nombre de la carrera que le corresponde

Cálculo Relacional de Tuplas

ALUMNOS = (idAlumno, nombre, dni, idLocalidad, idCarrera)

MATERIAS = (idMateria, nombre, añoCurso, idCarrera)

CARRERAS = (idCarrera, nombre, duración)

INSCRIPCIONES = (idInscripcion, idAlumno, idMateria, año, nota)

LOCALIDADES = (idLocalidad, nombre)

- **Cuantificador Universal (\forall)**

- $s / \forall t \in R(Q(t)) /$ el predicado $Q(t)$ sea verdadero en R

- **Ej21:** nombre de los alumnos que hayan cursado todas las materias de la carrera “Informática”

Cálculo Relacional de Dominios

- El **Cálculo Relacional de Dominios** (CRD) es también un lenguaje no procedural, similar al cálculo relacional de tuplas
- Sin embargo, existen algunas diferencias entre ellos:
 - Las consultas se expresan en términos de **valores individuales** en lugar de tuplas completas
 - Operar a nivel de atributos individuales puede proporcionar una **granularidad más fina** para expresar consultas

Cálculo Relacional de Dominios

- Se utilizan **variables de dominio** que toman valores del dominio de un atributo (y no de tuplas completas)
- Definición formal
 - Expresión: $\{ \langle x_1, \dots, x_N \rangle / P \langle x_1, \dots, x_N \rangle \}$;
con $\langle x_1, \dots, x_N \rangle$ variables de dominio y P fórmula
 - Átomos definidos como el **CRT**
 - Fórmulas definidas como el **CRT**
 - Seguridad de expresiones: similar a **CRT**

Cálculo Relacional de Dominios

ALUMNOS = (idAlumno, nombre, dni, idLocalidad, idCarrera)

MATERIAS = (idMateria, nombre, añoCurso, idCarrera)

CARRERAS = (idCarrera, nombre, duración)

INSCRIPCIONES = (idInscripcion, idAlumno, idMateria, año, nota)

LOCALIDADES = (idLocalidad, nombre)

- **Ej22:** listar todas las tuplas de ALUMNOS
- **Ej23:** listar todas las materias de 1° año (de cualquier carrera)
- **Ej24:** presentar el nombre de las carreras disponibles

Cálculo Relacional de Dominios

ALUMNOS = (idAlumno, nombre, dni, idLocalidad, idCarrera)

MATERIAS = (idMateria, nombre, añoCurso, idCarrera)

CARRERAS = (idCarrera, nombre, duración)

INSCRIPCIONES = (idInscripcion, idAlumno, idMateria, año, nota)

LOCALIDADES = (idLocalidad, nombre)

- **Ej25:** mostrar el nombre de las carreras que duren más de 3 años
- **Ej26:** presentar el nombre de la materia y el de la carrera que le corresponde

Leng. de consulta

Ej. adicionales

VIVE = (nombrePersona, calle, ciudad)

TRABAJA = (nombrePersona, nombreCompañía, salario)

SITUADA_EN = (nombreCompañía, ciudad)

DIRIGE = (nombrePersona, nombreDirector)

1. Nombre de los empleados que trabajan en la compañía X
2. Nombre de los empleados, que no trabajan en la compañía X
3. Nombre y ciudad en la que trabajan los empleados de la compañía X
4. Nombre, calle y ciudad de aquellos que trabajen para X y cobren más de 1000
5. Empleados que viven en la misma ciudad en la que está la compañía en la que trabajan

Leng. de consulta

Ej. adicionales

VIVE = (nombrePersona, calle, ciudad)

TRABAJA = (nombrePersona, nombreCompañía, salario)

SITUADA_EN = (nombreCompañía, ciudad)

DIRIGE = (nombrePersona, nombreDirector)

6. Modificar la dirección del empleado Y
7. Eliminar todos los empleados de la compañía X
8. Dar un aumento del 10% a los empleados
9. Dar un aumento del 20% a los directores

Nota: la tabla **DIRIGE** contiene dos atributos "nombrePersona". Para diferenciar al director del dirigido, el nombre del director es el atributo llamado nombreDirector. Un empleado puede aparecer como director y/o como dirigido.