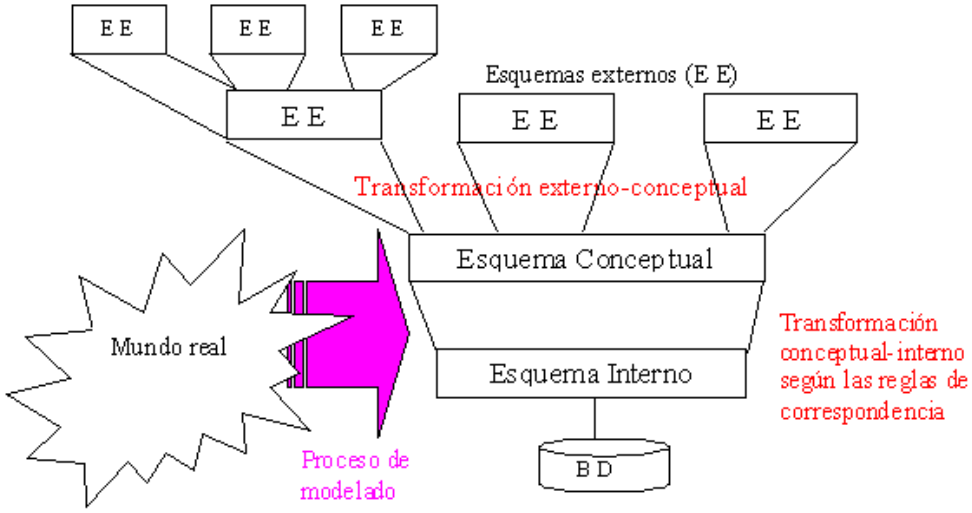


UNIVERSIDAD DE LOS ANDES

Bases de datos Unidad 3

Universidad de Los Andes
Escuela de Ingeniería de Sistemas
Departamento de Computación

Tema 2. Lenguaje de consulta del modelo relacional y objeto-relacional

Tema 2. Lenguajes de consulta del modelo relacional y objeto-relacional

- **Contenido:**
  - SQL3 – LMD y LCD
  - Cálculo relacional de tuplas y de dominio. QBE
  - Catálogo del sistema
- **Objetivo:**
  - Desarrollar habilidades en el uso de los lenguajes de consulta de las bases de datos objeto-relacionales
- **Actividades:**
  - Leer: Elmasri y Navathe, cap. 6, 8 y 9
  - Realizar los ejercicios de este documento
  - Realizar el ejercicio 3

UNIVERSIDAD DE LOS ANDES

Lenguaje de manipulación de datos (LMD)

- **SQL soporta operaciones lógicas basadas en lógica de tres (3) valores, a saber: verdadero, falso y nulo**
- **Verificación de valores nulos con IS NULL o IS NOT NULL**
- **Consulta: conjunto de sentencias SQL cuyo resultado es una tabla temporal**
  - Todas las sentencias SQL deben prepararse con antelación a su ejecución
  - SQL soporta todas las operaciones del algebra relacional
- **BD de ejemplo:**

Producto(codPro, nomPro, cantExistencia, color, ubicacionAlmacen)

Venta(codVen, fechaVen, *rifCli*, *codProVen*, cantVen, montoVen)

Compra(codCom, fechaCom, *codProCom*, cantCom, montoCom, *rifProv*)

Cliente(rif, nomCli, dirCli, ciudad, telCli, celCli)

UNIVERSIDAD DE LOS ANDES

Proyección

select unique listaDeAtributos

from nombreDeLaTabla

Donde: listaDeAtributos contiene los atributos de la proyección separados por coma y la palabra **unique** hace que se eliminen las tuplas duplicadas

Ejemplo: Liste los nombres de los productos, cantidad y ubicación en el almacen

select unique nomPro, cantExistencia, ubicacionAlmacen

from Producto

Liste todas las ventas con código, fecha y monto

Select unique codVen, fechaVen, montoVen

From Venta

UNIVERSIDAD DE LOS ANDES

Restricción

select *

from nombreDeLaTabla

where Q

donde

* indica que se proyecten todos los atributos de la tabla y **Q** es la fórmula de cualificación que se construye con los operadores **<, >, <=, >=, =, <>, and, or, not, between**

Ejemplo: Liste todas las ventas realizadas el 12/5/2015

select *

from Venta

where fechaVen **= '12/5/2015'**

Selección: Proyección y Restricción

select listaDeAtributos

from nombreDeLaTabla

where Q

Ejemplo: Liste el código del producto, la cantidad vendida y la fecha de venta de todas las ventas realizadas cuyos códigos de venta están entre '000100' y '000200'

select codProVen, cantVen, fechaVen

from Venta

where codVen **between** '000100' **and** '000200'

Ordenamiento de los resultados

- **Los resultados de salida en pantalla, papel u otra relación pueden ordenarse ascendente o descendentemente por uno o varios atributos colocando después de la cláusula where la cláusula**

order by atributo1 {asc|desc}, atributo2 {asc|desc}, …

Ejemplo: Liste el código de producto, la cantidad vendida y la fecha de venta de todas las ventas realizadas cuyos código de venta estén entre '000100' y '000200', ordenados por fecha de venta y código de producto

select codProVen, cantVen, fechaVen

from Venta

where codVen **between** '000100' **and** '000200'

order by fechaVen, codProVen **desc**

UNIVERSIDAD DE LOS ANDES

Producto cartesiano

select *

from listaDeTablas

Donde listaDeTablas es una lista de tablas separadas por coma

Ejemplo: Liste la información de la compra de cada producto actualmente en el almacén

select *

from Producto**,** Compra

Esta cláusula es válida si no está declarada codProCom como clave foránea en Compra correspondiente a codPro en Producto, de lo contrario se realiza un producto natural (*join*)

Producto natural (*join*)

select *

from listaDeTablas

where Q'

Donde listaDeTablas es la lista de las tablas involucradas separadas por coma y Q' es la fórmula de cualificación multiatributos

A cada tabla de la listaDeTablas se le puede asociar una variable para utilizarla como su alias

Ejemplo: ¿Cuál es el nombre del proveedor de cada producto actualmente en el almacén?

select P.codPro, CL.nomCli

from Producto P, Compra C, Cliente CL

where P.codPro = C.codProCom **and** C.rifProv=CL.rif

Equiproducto a la izquierda

select *

from tabla1 left join tabla2

on Q'

Donde tabla1 y tabla2 son tablas de la BD y Q' es una fórmula de cualificación multiatributos

Ejemplo: ¿Cuáles son los productos actualmente en almacén comprados al proveedor 'Yoshi Vendo'?

select *

from Cliente CL, Producto P **left join** Compra C

on P.codPro **=** C.codProCom **and** C.rifProv **=** CL.rif **and** CL.nomCli **=** 'Yoshi Vendo'

Equiproducto a la derecha

select *

from tabla1 right join tabla2

on Q'

Donde tabla1 y tabla2 son tablas de la BD y Q' es una fórmula de cualificación multiatributos

Ejemplo: ¿Cuáles son las compras realizadas al proveedor de RIF "J-55055055" del producto de nombre 'Cinta'?

select *

from Producto P **right join** Compra C

on P.codPro **=** C.codProCom **and** C.rifProv **=** 'J-55055055' **and** P.nomPro **=** 'Cinta'

UNIVERSIDAD DE LOS ANDES

Unión

select {*| listaDeAtributos}

from listaDeTablas

where Q1

union

select {*| listaDeAtributos}

from listaDeTablas

where Q2

Solo es equivalente al operador del algebra relacional si los resultados de ambas consultas tienen el mismo esquema

En el resultado se eliminan las tuplas duplicadas

Ejemplo: Liste los productos comprados o vendidos el 12/3/2014

UNIVERSIDAD DE LOS ANDES

Unión

```
select P.codPro, P.nomPro
from Producto P, Compra C
where      C.fechaCom = '12/3/2014' and P.codPro =
           C.codProCom
union
select P.codPro, P.nomPro
from Producto P, Venta V
where      V.fechaVen= '12/3/2014' and P.codPro =
           V.codProVen
```

UNIVERSIDAD DE LOS ANDES

Intersección

select {*| listaDeAtributos}

from listaDeTablas

where Q1

intersect

select {*| listaDeAtributos}

from listaDeTablas

where Q2

Solo es equivalente al operador del algebra relacional si los resultados de ambas consultas tienen el mismo esquema

En el resultado se eliminan las tuplas duplicadas

Ejemplo: Liste los productos comprados y vendidos el 14/3/14

UNIVERSIDAD DE LOS ANDES

Intersección

```
select P.codPro, P.nomPro
from Producto P, Compra C
where      C.fechaCom = '14/3/14' and P.codPro =
           C.codProCom
intersect
select P.codPro, P.nomPro
from Producto P, Venta V
where      V.fechaVen= '14/3/14' and P.codPro =
           V.codProVen
```

UNIVERSIDAD DE LOS ANDES

Diferencia

select {*| listaDeAtributos}

from listaDeTablas

where Q1

except

select {*| listaDeAtributos}

from listaDeTablas

where Q2

Solo es equivalente al operador del algebra relacional menos, si los resultados de ambas consultas tienen el mismo esquema

En el resultado se eliminan las tuplas duplicadas

UNIVERSIDAD DE LOS ANDES

Diferencia

Ejemplo: Liste los productos comprados el 18/3/15 que no fueron vendidos en dicha fecha

```
select P.codPro, P.nomPro
from Producto P, Compra C
where C.fechaCom = '18/3/15' and P.codPro = C.codProCom
except
select P.codPro, P.nomPro
from Producto P, Venta V
where V.fechaVen= '18/3/15' and P.codPro = V.codProVen
```

Sintaxis de la instrucción select

select {*| listaDeAtributos}

from listaDeTablas

[where Q]

[group by listaDeAtributos1]

[having condicion]

[order by listaDeAtributos2]

Donde:

group by agrupa los resultados por los atributos de la **listaDeAtributos1**

having selecciona los resultados que cumplan con la condición especificada

order by ordena ascendente o descendentemente

UNIVERSIDAD DE LOS ANDES

Funciones

Contar valores o tuplas: count(atributo)

Sumar valores numéricos: sum(atributo)

Calcular el promedio de valores numéricos: avg (atributo)

Calcular el valor más pequeño : min (atributo)

Calcular el valor más grande: max (atributo)

Ejemplos:

```
select P.codPro, P.nomPro, min(P.cantExistencia)
from Producto P
```

```
select P.nomPro, avg(V.cantVen)
from Producto P, Venta V
where fechaVen between '1/1/2015' and '31/12/2015' and
        P.codPro = V.codProVen
```

Encadenamiento de consultas

select {*| listaDeAtributos}

from listaDeTablas

where atributo1 {in | [not]exist | operadorDeComparacion {all | any }} (select {*| listaDeAtributos} …)

Donde:

in operador de membrecía equivalente a pertenece

exist cuantificador existencial

all cuantificador universal

any indica que un atributo "es al menos" **>, <, >=, <=, = o <>** que cualquier valor de otro atributo

Encadenamiento de consultas

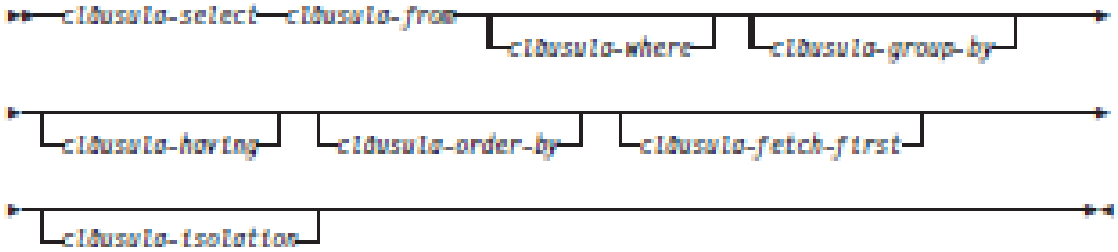
Ejemplo: ¿Cuáles son los nombres de los productos comprados al proveedor de nombre 'Yoshi Vendo' y vendido a por lo menos un cliente?

```
select P.nomPro
from Producto P, Venta V
where P.codPro = V.codProVen and P.codPro in
( select  P.codPro
        from Producto P, Compra C, Cliente CL
        where P.codPro = C.codProCom and
                C.rifProv = CL.rif and
                CL.nomCli = 'Yoshi Vendo' )
```

Sintaxis en PostgreSQL

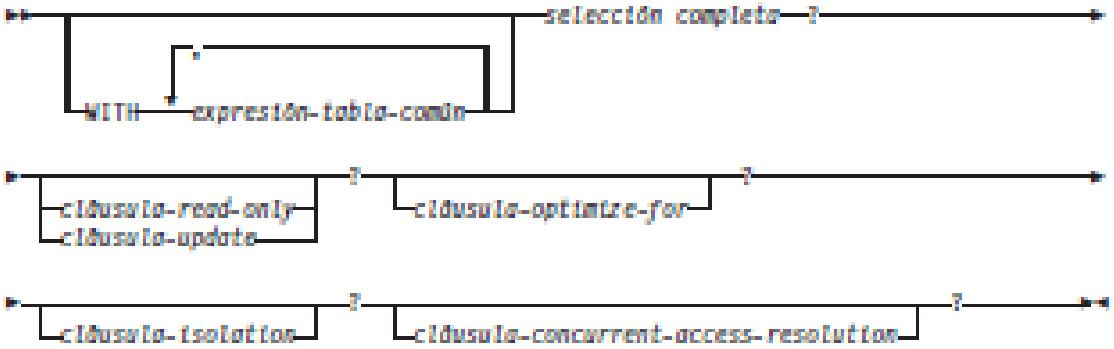
```
[ WITH [ RECURSIVE ] with_query [, ...] ]
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
    * | expression [ [ AS ] output_name ] [, ...]
    [ FROM from_item [, ...] ]
    [ WHERE condition ]
    [ GROUP BY expression [, ...] ]
    [ HAVING condition [, ...] ]
    [ WINDOW window_name AS ( window_definition ) [, ...] ]
    [ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select ]
    [ ORDER BY expression [ ASC | DESC | USING operator ] [ NULLS { FIRST | LAST } ] [, ...]
    [ LIMIT { count | ALL } ]
    [ OFFSET start [ ROW | ROWS ] ]
    [ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } ONLY ]
    [ FOR { UPDATE | SHARE } [ OF table_name [, ...] ] [ NOWAIT ] [...] ]

where from_item can be one of:

    [ ONLY ] table_name [ * ] [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
    ( select ) [ AS ] alias [ ( column_alias [, ...] ) ]
    with_query_name [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
    function_name ( [ argument [, ...] ] ) [ AS ] alias [ ( column_alias [, ...] | column_defini
    function_name ( [ argument [, ...] ] ) AS ( column_definition [, ...] )
    from_item [ NATURAL ] join_type from_item [ ON join_condition | USING ( join_column [, ...]

and with_query is:

    with_query_name [ ( column_name [, ...] ) ] AS ( select | values | insert | update | delete )

TABLE [ ONLY ] table_name [ * ]
```

Sintaxis en DB2



Las cláusulas de la subselección se procesan en el orden siguiente:

1. FROM, cláusula
2. WHERE, cláusula
3. GROUP BY, cláusula
4. HAVING, cláusula
5. SELECT, cláusula
6. ORDER BY, cláusula
7. Cláusula FETCH FIRST



Lenguaje de control de datos (LCD)

- **Privilegio: permiso a un usuario, rol o PUBLIC, para SELECT, INSERT, UPDATE, DELETE, REFERENCES, USAGE, UNDER, TRIGGER, EXECUTE**
- **Rol: colección de 0 o más autorizaciones**
- **Dar privilegios: grant**
- **Quitar privilegios: revoke**

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | TRIGGER }
    [, ...] | ALL [ PRIVILEGES ] }
    ON { [ TABLE ] table_name [, ...]
         | ALL TABLES IN SCHEMA schema_name [, ...] }
    TO { [ GROUP ] role_name | PUBLIC } [, ...] [ WITH GRANT OPTION ]

REVOKE [ GRANT OPTION FOR ]
    { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | TRIGGER }
    [, ...] | ALL [ PRIVILEGES ] }
    ON { [ TABLE ] table_name [, ...]
         | ALL TABLES IN SCHEMA schema_name [, ...] }
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]
```

PostgreSQL

Creación de usuarios

```
CREATE USER name [ [ WITH ] option [ ... ] ]

where option can be:

      SUPERUSER | NOSUPERUSER
    | CREATEDB | NOCREATEDB
    | CREATEROLE | NOCREATEROLE
    | CREATEUSER | NOCREATEUSER
    | INHERIT | NOINHERIT
    | LOGIN | NOLOGIN
    | REPLICATION | NOREPLICATION
    | CONNECTION LIMIT connlimit
    | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
    | VALID UNTIL 'timestamp'
    | IN ROLE role_name [, ...]
    | IN GROUP role_name [, ...]
    | ROLE role_name [, ...]
    | ADMIN role_name [, ...]
    | USER role_name [, ...]
    | SYSID uid
```

PostgreSQL

- GRANT (autorizaciones de bases de datos)

  La autorización ACCESSCTRL no permite a quien la posee conceder autorizaciones ACCESSCTRL, DATAACCESS, DBADM o SECADM. Sólo puede conceder estas autorizaciones un usuario que tiene autorización SECADM.
- GRANT (privilegios de variable global)
- GRANT (privilegios de índice)
- GRANT (privilegios de módulo)
- GRANT (privilegios de paquete)
- GRANT (privilegios de rutina)
- GRANT (privilegios de esquema)
- GRANT (privilegios de secuencia)
- GRANT (privilegios de servidor)
- GRANT (privilegios de tabla, vista o apodo)
- GRANT (privilegios de espacio de tablas)
- GRANT (privilegios de carga de trabajo)
- GRANT (privilegios de objeto XSR)

DB2

Inserción de tuplas en una tabla

insert into nombreDeTabla [(listaDeAtributos)]
values (listaDeValores | consulta)

Donde: **listaDeAtributos** es la lista de los nombres de los atributos que se van a asociar a los valores de **listaDeValores** que es la lista de los valores separados por coma que se insertan en la tabla de nombre dado y consulta es una sentencia **select** que obtiene los valores que se insertan en la tabla

Ejemplo: ingresar el producto nuevo de nombre 'cartulina', cantidad 25, ubicación en 'S01' y color verde

insert into Producto(nomPro, cantExistencia, color, ubicacionAlmacen)
values ('cartulina', 25, 'verde', 'S01')

PostgreSQL

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
INSERT INTO table_name [ ( column_name [, ...] ) ]
    { DEFAULT VALUES | VALUES ( { expression | DEFAULT } [, ...] ) [, ...] | query }
    [ RETURNING * | output_expression [ [ AS ] output_name ] [, ...] ]
```

Eliminación de tuplas de una tabla

delete from nombreDeTabla

[where Q]

donde: Q es una fórmula de cualificación

Ejemplo: Elimine los productos de nombre 'papel bond 16' y color 'beige'

delete from Producto

where nomPro **=** 'papel bond 16' **and** color **=** 'beige'

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
DELETE FROM [ ONLY ] table_name [ * ] [ [ AS ] alias ]
    [ USING using_list ]
    [ WHERE condition | WHERE CURRENT OF cursor_name ]
    [ RETURNING * | output_expression [ [ AS ] output_name ] [, ...] ]
```

PostgreSQL

Actualización de tuplas en una tabla

update nombreTabla

set {listaDeExpresiones}

[where Q]

donde: la **listaDeExpresiones** es una lista de nombre de atributo = expresion separadas por coma

Ejemplo: Actualice la cantidad en existencia con un aumento de 5 para los productos de nombre 'cartulina' y color 'azul' o 'verde'

update Producto **set** cantExistencia **=** cantExistencia+5

where nomPro **=** 'cartulina' **and** color **=** 'azul' **or** color **=** 'verde'

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
UPDATE [ ONLY ] table_name [ * ] [ [ AS ] alias ]
    SET { column_name = { expression | DEFAULT } |
          ( column_name [, ...] ) = ( { expression | DEFAULT } [, ...] ) } [, ...]
    [ FROM from_list ]
    [ WHERE condition | WHERE CURRENT OF cursor_name ]
    [ RETURNING * | output_expression [ [ AS ] output_name ] [, ...] ]
```

PostgreSQL

UNIVERSIDAD DE LOS ANDES

Autoevaluación

1. ¿Cuáles son las instrucciones de manipulación de datos del SQL?
2. ¿Cómo se realiza una proyección y restricción en SQL?
3. ¿Cuáles son las instrucciones de control de datos del SQL?
4. ¿Cómo se realizan las uniones, diferencias e intercepciones en SQL?
5. ¿Cómo se encadenan las sentencias de consulta en SQL?
6. ¿Cuáles son las funciones del SQL3?
7. ¿Cómo se modifican las tuplas de una tabla en SQL3?
8. ¿Cómo se insertan nuevas tuplas en una tabla en SQL3?
9. ¿Cómo se eliminan tuplas de una tabla en SQL3?

UNIVERSIDAD DE LOS ANDES

Ejercicio

Esquema de la base de datos:

Producto(codPro, nomPro, cantExistencia, color, ubicacionAlmacen)

Venta(codVen, fechaVen, *rifCli*, *codProVen*, cantVen, montoVen)

Compra(codCom, fechaCom, *codProCom*, cantCom, montoCom, *rifProv*)

Cliente(rif, nomCli, dirCli, ciudad, telCli, celCli)

Responda las siguientes consultas en SQL estándar:

1. **¿Cuál es el nombre y el color de cada producto en almacén?**
2. **¿Cuál es el nombre y la cantidad en existencia de cada producto de color negro en el almacén?**
3. **¿Cuál es el nombre del proveedor de cada producto en el almacén?**
4. **¿Cuáles son los clientes que han comprado al menos un producto de color blanco?**
5. **Otorgue permisos de modificación de la tabla Producto al usuario XX1**
6. **Revoque los permisos de consulta al usuario METICH**

UNIVERSIDAD DE LOS ANDES

Cálculo relacional de tuplas

- **Es un lenguaje de consulta formal que permite expresar las consultas a partir de fórmulas bien formadas, donde las variables son interpretadas como variantes sobre las tuplas de las tablas**
- **Fue presentado por Codd en 1972 y se deduce del Cálculo de Predicados**
- **Átomos:**

1. **las variables están asociadas a las tuplas de las tablas y se denota como relación(variable)**

   Ejm: Modelo(M)

2. **los valores constantes están asociados a los valores de los dominios de los atributos y las funciones generadoras de los mismos se denotan como variable.atributo**

   Ejm: M.marca

UNIVERSIDAD DE LOS ANDES

Cálculo relacional de tuplas

3. **los predicados utilizados se construyen con los operadores de comparación {<, ≤ , >, ≥ , =, ≠} y constantes**

Ejm: M.marca ≠ 'fiat'

- **Una fórmula bien formada (fbf) se define como:**

1. **Todo átomo es una fórmula bien formada F**
2. **Si F1 y F2 son fbf, entonces F1 and F2, F1 or F2, not F1 o not F2 son fbf**
3. **∃ F1 es una fbf**
4. **∀ F1 es una fbf**

Ejemplos de consultas según el esquema relacional siguiente:

Producto(codPro, nomPro, cantExistencia, color, ubicacionAlmacen)

Venta(codVen, fechaVen, *rifCli*, *codProVen*, cantVen, montoVen)

Compra(codCom, fechaCom, *codProCom*, cantCom, montoCom, *rifProv*)

Cliente(rif, nomCli, dirCli, ciudad, telCli, celCli)

UNIVERSIDAD DE LOS ANDES

Ejemplos CRT

1. **¿Cuál es el nombre y el color de cada producto en almacén?**

{P.nomPro, P.color / Producto(P)}

2. **¿Cuál es el nombre y la cantidad en existencia de cada producto de color rojo en el almacén?**

{P.nomPro, P.cantExistencia / Producto(P) ∧ P.color = 'rojo'}

3. **¿Cuál es el nombre del proveedor de cada producto en el almacén?**

{CL.nomCli, P.nomPro / Producto(P) ∧ Compra(C) ∧ Cliente(CL) ∧ P.codPro = C.codProCom}

4. **¿Cuáles son los clientes que han comprado al menos un producto de color verde?**

{C.nomCli, P.nomPro / ∃ P Venta(V) ∧ Producto(P) ∧ Cliente(C) ∧ V.codProVen = P.codPro ∧ P.color = 'verde' ∧ V.rifCli = C.rif}

UNIVERSIDAD DE LOS ANDES

Ejemplos CRT

5. **¿Cuáles son los nombres de los productos comprados a todos los proveedores y vendidos a por lo menos un cliente?**

{P.nomPro, CL.nomCli / ∀P ∃V ∃ A Venta(V) ∧ Producto(P) ∧ Compra(A) ∧ Compra(C) ∧ Cliente(CL) ∧ P.codPro = V.codProVen ∧ P.codPro = A.codProCom ∧ A.rifProv = C.rifProv ∧ C.rifProv = CL.rif}

Lenguaje QUEL propuesto por Zook, 1977

Range of <listaDeVariables> is <listaDeTablas>

Retrieve [[into] <nombreTabla>] <listaVariableAtributo> [where <condición>]

Append [to] <variable> (at=valor,…) [where <condición>]

Replace <variable> (at=expresión,…) [where <condición>]

Delete <variable> [where <condición>]

UNIVERSIDAD DE LOS ANDES

Cálculo relacional de dominios

- **Lenguaje de consulta formal que permite expresar las consultas a partir de fórmulas bien formadas, donde cada variable se interpreta como variante sobre el dominio del atributo de una relación**
- **Se deduce del cálculo de predicados también, pero:**

1. **Las variables están asociadas a los dominios de los atributos y se denota como relación(at1:v1, at2:v2, …)**

   Ejemplo: ModeloCarro(modelo: m, marca: c)

2. **Los predicados utilizados se construyen igual que para el cálculo relacional de tuplas**

UNIVERSIDAD DE LOS ANDES

Ejemplos CRD

1. **¿Cuál es el nombre y el color de cada producto en almacén?**

{P, C / Producto(nomPro: P, color: C)}

2. **¿Cuál es el nombre y la cantidad en existencia de cada producto de color rojo en el almacén?**

{P, A / Producto(nomPro: P, cantidadExistencia: A, color = 'rojo')}

3. **¿Cuáles son los clientes que han comprado al menos un producto de color verde?**

{V, P / ∃NP Venta(rifCli: V, codProVen: NP) ∧ Producto(codPro: NP, nomPro: P, color = 'verde')}

4. **¿Cuál es el nombre del proveedor de cada producto en el almacén?**

{C, P/ Producto(codPro: PC, nomPro: P) ∧ Compra(codProCom: PC, rifProv: R) ∧ Cliente(rif:R, nomCli:C)}

UNIVERSIDAD DE LOS ANDES

Ejemplos CRD

5. **¿Cuáles son los nombres de los productos comprados a todos los proveedores y vendidos a por lo menos un cliente?**

{P, NC / ∀NC ∃NP Venta(codProVen: NP) ∧ Producto(codPro: NP, nomPro: P) ∧ Compra(codProCom: NP, rifProv: RP) ∧ Cliente(rif:RP, nomCli:NC}

- **Lenguaje QBE (Query By Example)**
  - Presentado por Zloff en 1977 y comercializado desde 1980
  - Es un lenguaje gráfico
  - La idea de su construcción es la formulación de la consulta mediante un ejemplo de la respuesta

UNIVERSIDAD DE LOS ANDES

Lenguaje QBE

- **Las consultas se realizan invocando los esquemas de las tablas objeto de la consulta, las cuales serán desplegadas en forma gráfica en la pantalla**
- **Una vez obtenidas, se posiciona el ratón en la o las columnas deseadas y se indica la operación a realizar**
  - Las variables se indican con el símbolo de subrayado como prefijo, ejemplo: _s, _3, _d5, o se subrayan, ejemplo: s, 3, d5
  - Las constantes se colocan directamente en la columna deseada precedidas por el operador de comparación deseado, si no es =
  - Toda variable desplegable está cuantificada implícitamente por el cuantificador existencial
  - **Todas** las operaciones deben tener como **sufijo** un punto

UNIVERSIDAD DE LOS ANDES

Operaciones de QBE

Operación	QBE
Desplegar o seleccionar	P.
Cuantificador universal	ALL.
Contar	CNT.
Promedio	AVG.
Suma o acumular	SUM.
Calcular el valor mínimo	MIN.
Calcular el valor máximo	MAX.
Agrupar tuplas	G.
Ordenar ascendentemente	AO.
Ordenar descendentemente	DO.
Negación lógica	¬
Disyunción lógica	OR.
Conjunción lógica	AND.

- **Condiciones adicionales se expresan en una ventana aparte, en algunos SGBD**
- **Las funciones CNT, AVG, SUM, MIN y MAX deben aplicarse a variables precedidas con ALL**
- **Si no se desean eliminar las tuplas dobles en una proyección, se coloca P.ALL._v**

UNIVERSIDAD DE LOS ANDES

Ejemplos QBE

1. **¿Cuál es el nombre y el color de cada producto en almacén?**

Producto	codPro	nomPro	cantidadExistencia	color	ubicacionAlmacen
		P.		P.	

2. **¿Cuál es el nombre y la cantidad en existencia de cada producto de color rojo en el almacén?**

Producto	codPro	nomPro	cantidadExistencia	color	ubicacionAlmacen
		P.	P.	rojo	

3. **¿Cuáles son los datos del producto número 100?**

Producto	codPro	nomPro	cantidadExistencia	color	ubicacionAlmacen
	100	P.	P.	P.	P.

Ejemplos en QBE

4. **¿Cuál es el nombre del proveedor de cada producto en el almacén?**

Producto	codPro	nomPro	cantidadExistencia	color	ubicacionAlmacen
	_np	P.			

Compra	codCom	fechaCom	rifProv	codProCom	cantCom	montoCom
			_rp	_np		

Cliente	rif	nomCli	ciudad	dirCli	telCli	celCli
	_rp	P.				

La reunión natural se indica colocando la misma variable en las columnas cuyos valores deben igualarse

Ejemplos en QBE

5. **¿Cuáles son los clientes que han comprado al menos un producto de color verde?**

Producto	codPro	nomPro	cantidadExistencia	color	ubicacionAlmacen
	_np	P.		verde	

Venta	codVen	fechaVen	rifCli	codProVen	cantVen	montoVen
			P.	_np		

Ejemplo de consultas en QBE

- **Desplegar los nombres de los productos de color verde ordenados descendentemente por nombre de producto**

Producto	codPro	nomPro	cantidadExistencia	color	ubicacionAlmacen
		P.ALL.DO._n		verde	

ofrece posibilidades adicionales para el cálculo de la clausura transitiva, que no puede ser expresada en cálculo relacional de tuplas o dominios

- **Desplegar todas las componentes de las componentes de un motor en la tabla**

Prelacion	codMateria	codMateriaPrelada
	'Proyecto de grado'	_cm
	_cm	P._cmp

Manipulación de datos en QBE

- **Inserción de una nueva tupla:** colocar **I.** debajo del nombre de la tabla

Insertar 10 pares de zapatos negros en K23(producto nuevo)

Producto	codPro	nomPro	cantidadExistencia	color	ubicacionAlmacen
I.	MAX.ALL._n+1	zapatos	10	negros	K23

- **Eliminación de tuplas:** colocar **D.** debajo del nombre de la tabla

Eliminar todos los productos de color 'gris'

Producto	codPro	nomPro	cantidadExistencia	color	ubicacionAlmacen
D.				gris	

Manipulación de datos en QBE

- **Actualización de tuplas:** colocar **U.** debajo del nombre de la tabla

Sumar 100 a todos los productos de color verde

Producto	nroPro	nombrePro	cantidadExistencia	color
U.	_np		_ce	'verde'
	_np		_ce+100	

Producto	codPro	nomPro	cantidadExistencia	color	ubicacionAlmacen
U.	_np		_ce	verde	
	_np		_ce+100		

Catálogo del SGBD

- **Contiene la información detallada de cada base de datos en el sistema, denominada metadatos, a saber: tablas, índices, atributos, vistas, restricciones, gatillos, etc.**

RELACIONES

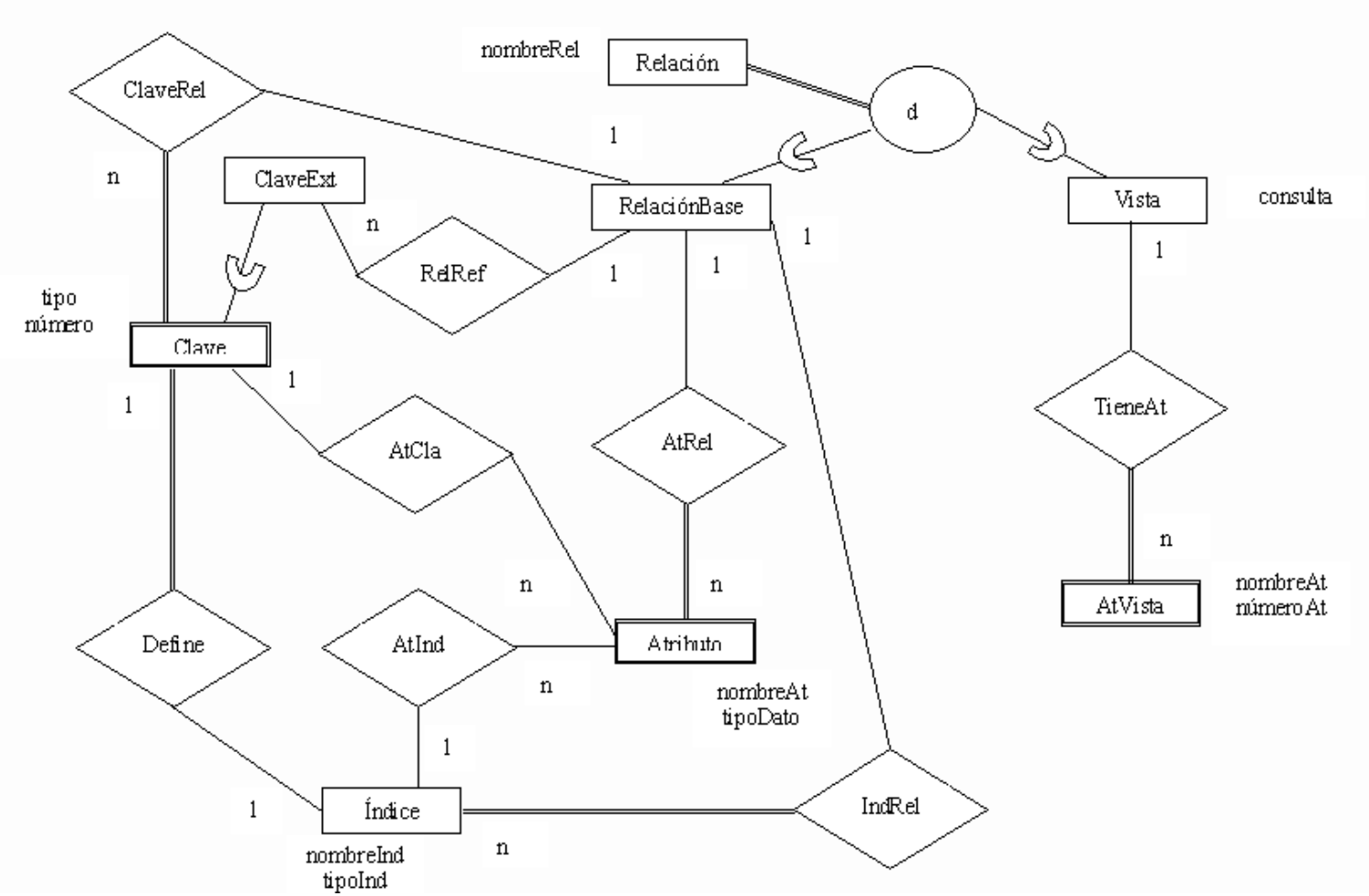
NombreRelacion	NumDeColumnas
ESTUDIANTE	4
CURSO	4
SECCIÓN	5
INFORME_CALIF	3
PRERREQUISITO	2

COLUMNAS

NombreColumna	TipoDatos	PerteneceARelacion
Nombre	Carácter (30)	ESTUDIANTE
NumEstudiante	Carácter (4)	ESTUDIANTE
Clase	Entero (1)	ESTUDIANTE
Especialidad	TipoEspecialidad	ESTUDIANTE
NombreCurso	Carácter (10)	CURSO
NumCurso	XXXXNNNN	CURSO
....	....	.....
....	....	.....
....	....	.....
NumPrerrequisito	XXXXNNNN	PRERREQUISITO

Ejemplo de catálogo del SMBD (tomado de Elmasri y Navathe)

Ejemplo de modelo del catálogo



UNIVERSIDAD DE LOS ANDES

Autoevaluación

1. ¿Qué se entiende por fórmulas bien formadas y por átomos?
2. ¿A qué se asocian las variables en el cálculo relacional de tuplas (CRT) y en el cálculo relacional de dominios (CRD)?
3. ¿Qué se entiende por CRT y por CRD?
4. ¿Cómo se expresan las consultas en el CRT?
5. ¿Cómo se expresan las consultas en el CRD?
6. ¿Cuáles son las características del lenguaje de consulta QBE?
7. ¿Cómo se realiza una consulta en QBE?
8. ¿Cómo se modela el catálogo de un SGBD?

UNIVERSIDAD DE LOS ANDES

Ejercicio

Esquema de la base de datos:

Dpto(#dpto, presupDpto, jefeDpto)

Emp(#emp, #pro, telEmp)

Pro(#pro, #dpto, prePro)

Ofic(#ofi, #dpto, area)

Tlfs(#tel, #ofi)

Responda las siguientes consultas en cálculo relacional de tuplas, cálculo relacional de dominios y QBE:

1. ¿Cuál es el número de oficina del empleado 110, en qué proyecto y en qué departamento trabaja?
2. ¿Cuál es el jefe del departamento 42 y cuál es su número de oficina y de teléfono?
3. ¿Cuáles son los proyectos cuyo presupuesto es mayor que el presupuesto del departamento al que pertenecen y cuáles son sus empleados asignados?
4. ¿Cuáles son las oficinas con área menor a 24 y cuáles son sus teléfonos?
5. ¿Cuáles son los empleados, proyectos, oficinas y teléfonos del departamento cuyo jefe es 101?
6. ¿Cuál es el área promedio de las oficinas del departamento con el presupuesto más alto?
7. Liste la información de todos los departamentos ordenada por su #dpto con sus proyectos, empleados de cada proyecto con sus números de teléfono y de oficina
8. Inserte una nueva oficina para el departamento 28 con un área de 12. Sólo en QBE.
9. Acaba de terminarse el proyecto 14, elimínelo junto con todos sus empleados. Sólo en QBE.
10. Acaba de ser recibida una donación anual para la compañía, por ello actualice los montos de los presupuestos de todos los departamentos para incrementarse en 100.000,00. Sólo en QBE.