TALLER DE PROGRAMACIÓN
UNIDAD 2: BASES DE DATOS

PROFESOR GERMÁN BARRIENTOS

PROFESOR JOEL TORRES

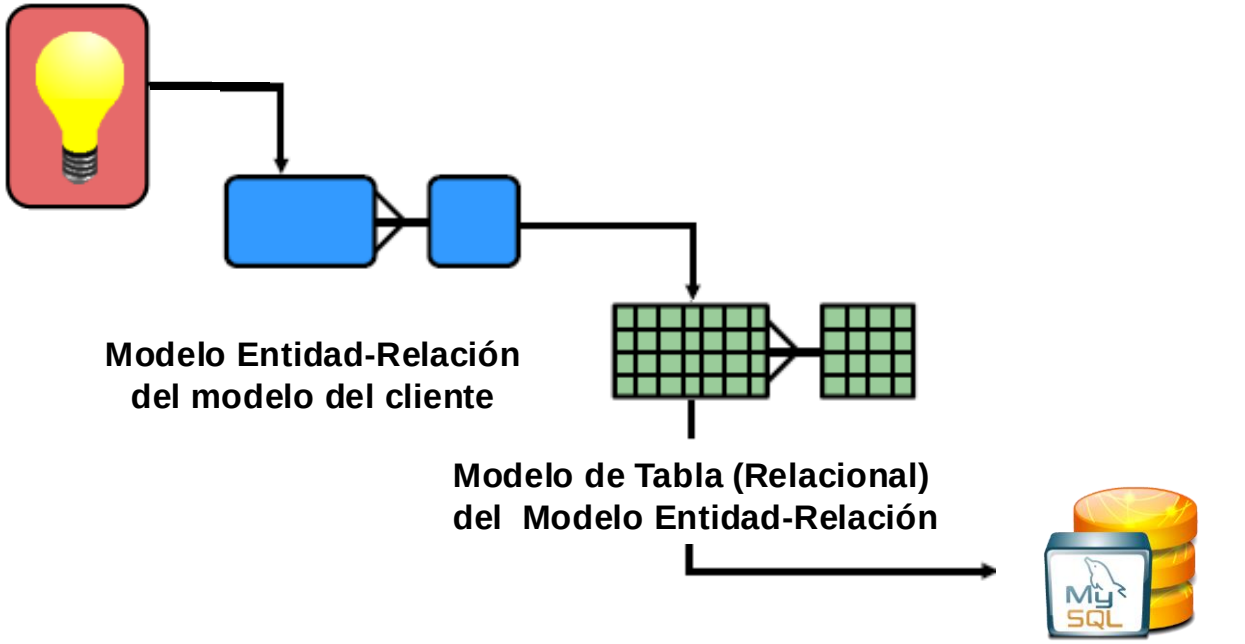
Universidad Austral de Chile

Conocimiento y Naturaleza

ASPECTOS GENERALES DEL SQL

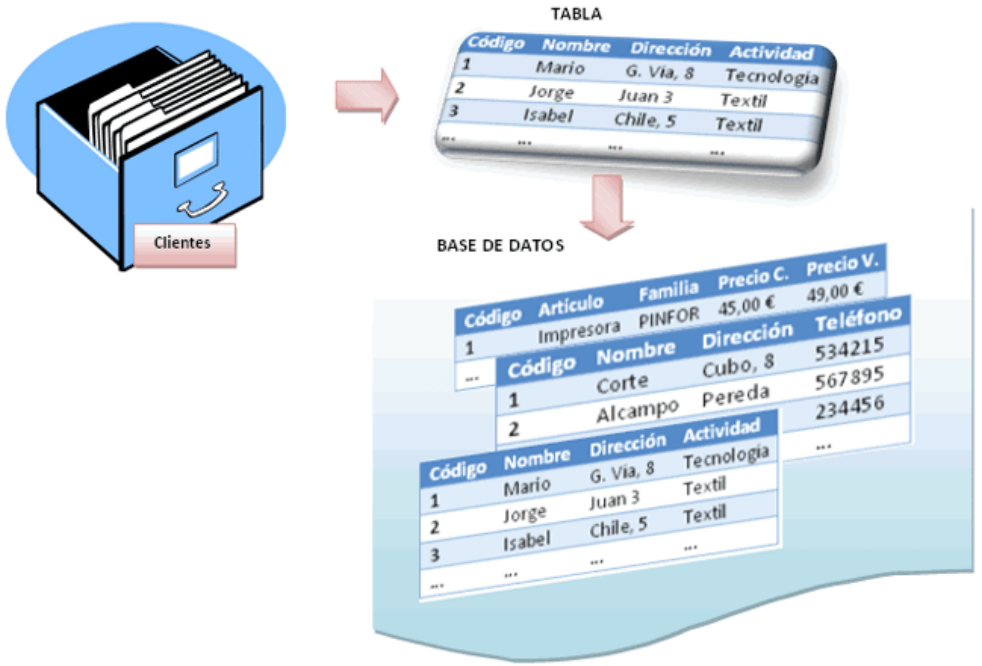
- Conocer los conceptos básicos de una Base de Datos Relacional.
- Conocer los conceptos básicos del lenguaje SQL.
- Conocer el Lenguaje de Consulta Estructurado (SQL).

MODELO DE DATOS



CONCEPTOS DE UNA BASE DE DATOS RELACIONAL

- Ejemplo:



CONCEPTOS DE UNA BASE DE DATOS RELACIONAL

- Ejemplo:

Nombre Tabla: EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
103	Alexander	Hunold	60
104	Bruce	Ernst	60
105	David	Austin	60
106	Valli	Pataballa	60
107	Diana	Lorentz	60
108	Nancy	Greenberg	100
109	Daniel	Faviet	100

Primary key

Foreign key

Nombre Tabla: DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700
100	Finance	108	1700
110	Accounting	205	1700

Primary key

TERMINOLOGÍA DE BASE DE DATOS RELACIONAL

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID
100	King	Steven	24000		90
101	Kochhar	Neena	17000		90
102	De Haan	Lex	17000		90
103	Hunold	Alexander	9000		60
104	Ernst	Bruce	6000		60
107	Lorentz	Diana	4200		
124	Mourgos	Kevin	5800		
141	Rajs	Trenna	3500		50
142	Davies	Curtis	3100		50
143	Matos	Randall	2600		50
144	Vargas	Peter	2500		50
149	Zlotkey	Eleni	10500	.2	80
174	Abel	Ellen	11000	.3	80
176	Taylor	Jonathon	8600	.2	80
178	Grant	Kimberely	7000	.15	
200	Whalen	Jennifer	4400		10
201	Hartstein	Michael	13000		20
202	Fay	Pat	6000		20
205	Higgins	Shelley	12000		110
206	Gietz	William	8300		110

USANDO SQL PARA COMUNICARSE CON UN RDBMS

SELECT INSERT UPDATE DELETE MERGE	DATA MANIPULATION LANGUAJE (DML)
CREATE ALTER DROP RENAME TRUNCATE COMMENT	DATA DEFINITION LANGUAJE (DDL)
GRANT REVOKE	DATA CONTROL LANGUAJE (DCL)
COMMIT ROLLBACK SAVEPOINT	TRANSACTION CONTROL

ELEMENTOS DE UN CÓDIGO DESARROLLADO EN SQL

Comandos SQL: DML, DDL, DCL, etc.

Palabras Reservadas: SELECT, FROM, WHERE, JOIN, etc.

Cláusulas: SELECT employee_id, last_name

Sentencia: SELECT employee_id, last_name
FROM employees;

Operadores: Aritméticos, de Comparación y Lógicos

Funciones de: carácter, números, fecha, conversión y generales

Funciones Agregadas o de Grupo

Constantes: números, textos, caracteres, fechas

SIGNIFICADO DEL FORMATO DE LAS SENTENCIAS SQL

SELECT {* | *columna* | *expresión* [*alias*],...}
FROM *tabla;*

[ ] (corchetes): no es obligatorio en el comando

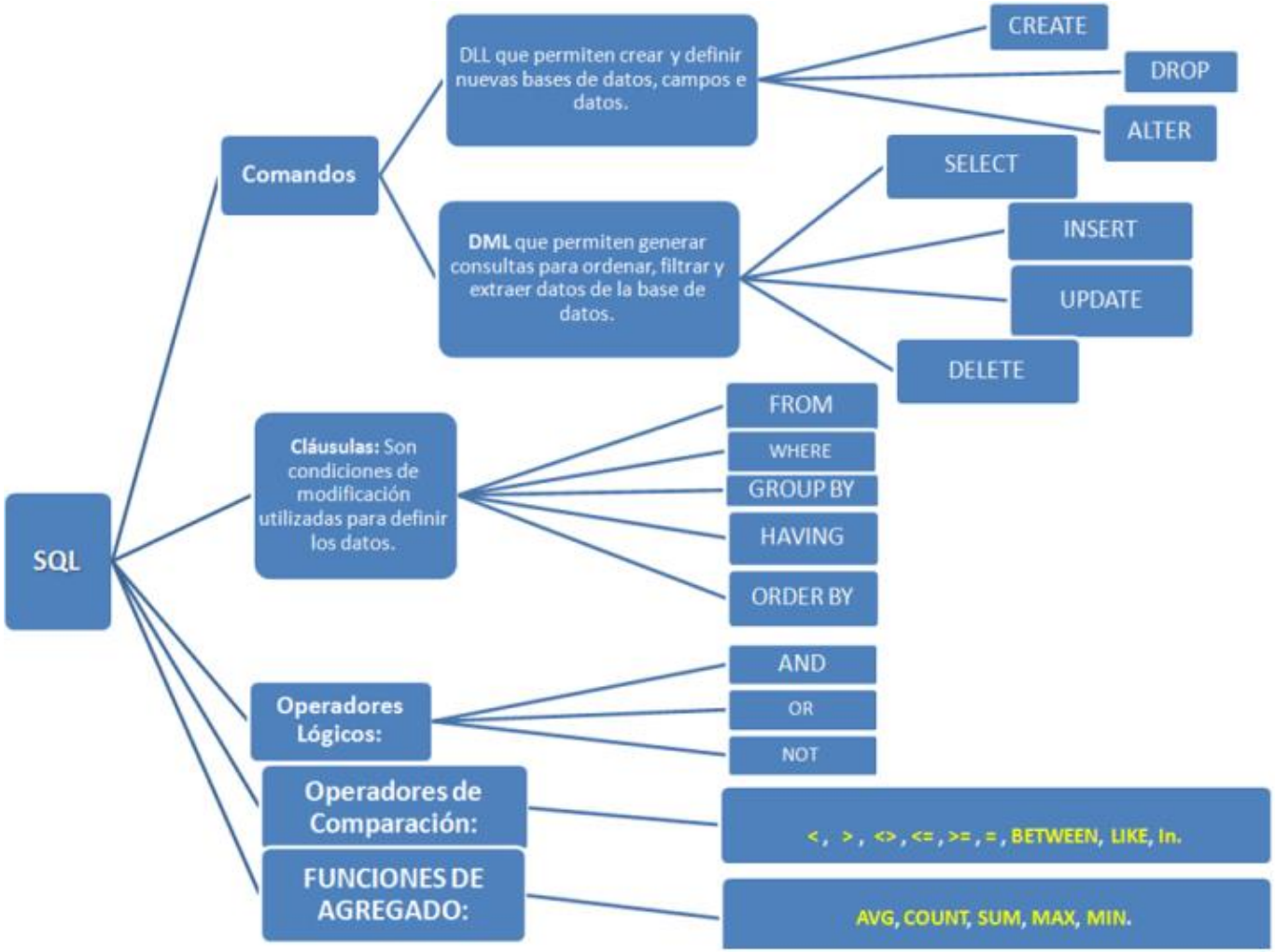
| (barra vertical): es una opción entre varias

... (puntos suspensivos): que lo indicado se puede repetir

{ } (llaves): opciones obligatorias pero se debe elegir sólo una

CONSIDERACIONES PARA CONSTRUIR UNA SENTENCIA SQL

ELEMENTOS DE UN CÓDIGO DESARROLLADO EN SQL



INSERT

- Permite INSERTAR una nueva fila a una o más tablas definidas:

```
INSERT INTO nombre_tabla (columna1, columna2, … )
VALUES (valor1, valor2, … );
```

- Las columnas establecidas definen el orden y columnas necesarias.
- Se puede resumir si se inserta toda la información requerida por defecto:

```
INSERT INTO nombre_tabla
VALUES (valor1, valor2, … , valorN);
```

- Se pueden insertar varias filas a la vez:

```
INSERT INTO nombre_tabla
VALUES (valor1, valor2, … , valorN),
       (valor1, valor2, … , valorN),
       …
       (valor1, valor2, … , valorN);
```

SELECT

- Permite recuperar información desde una o más tablas

```
SELECT nombre_columna
FROM nombre_tabla
WHERE condicion_logica;
```

- SELECT: indica los atributos que desea mostrar
  - Opcionalmente puede ocupar DISTINCT para evitar repeticiones de resultados
- FROM: indica las tablas involucradas en la consulta
- WHERE: indica el listado de condiciones que se deben cumplir para efectuar la selección y mostrar la información esperada.
- La sentencia WHERE puede ser opcional.

OPERADORES LÓGICOS

- Una condición es una comparación representada por:

```
Nombre_columna operador_relacional valor_columna
Ej: precio > 1000000
```

- Los operadores relacionales son <, <=, >, >=, =, <>
- Los operadores Lógicos permiten unir dos o más condiciones en la clausula WHERE:
  - AND se usa cuando se requiere que ambas condiciones resulten verdaderas
  - OR se usa cuando se requiere que alguna de las condiciones resulte verdadera
  - XOR se usa cuando se requiere que solamente una condición resulte verdadera
  - NOT se usa cuando se desea el resultado contrario de una condición

```
Consulta1 operador_logico consulta2
Ej: precio > 1000000 AND precio < 5000000
```

UPDATE

- Permite actualizar la información desde una o más tablas

```
UPDATE nombre_tabla
SET nombre_columna = valor, ...
WHERE condicion_logica;
```

- UPDATE: indica la tabla a actualizar
- SET: indica las columna y valores a cambiar
- WHERE: indica el listado de condiciones que se deben cumplir para efectuar la actualización y cambiar la información esperada, generalmente usada para individualizar a las filas que deben aplicar el cambio.
- La sentencia WHERE puede ser opcional.

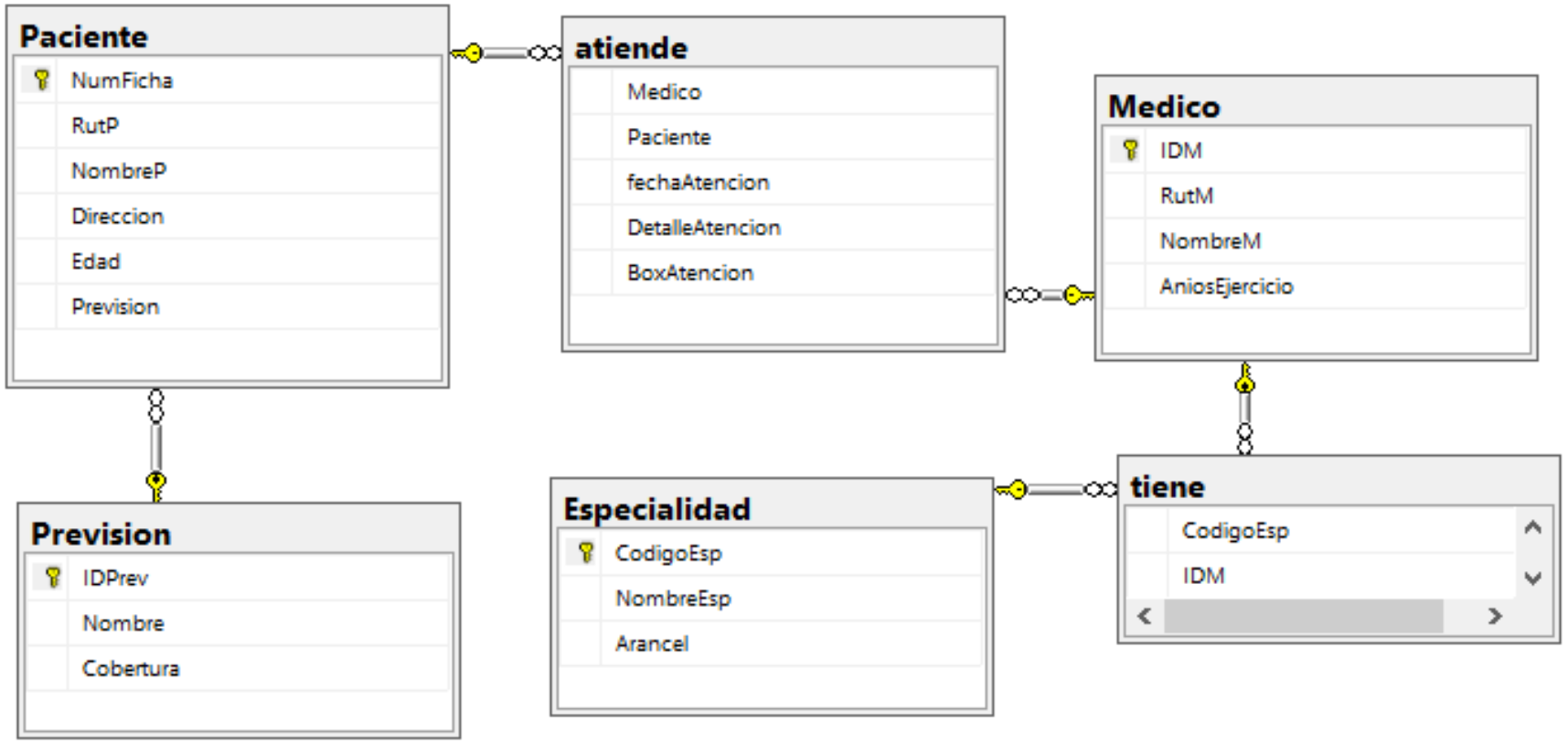
DELETE

- Permite recuperar información desde una o más tablas

```
DELETE TOP numero
FROM nombre_tabla
WHERE condicion_logica;
```

- DELETE: indica las filas a borrar
  - Opcionalmente puede ocupar TOP para indicar la cantidad de filas a borrar
  - Estas filas son escogidas aleatoriamente, usar cuidadosamente.
- FROM: indica la tabla de donde se borrarán las tablas
- WHERE: indica el listado de condiciones que se deben cumplir para efectuar la eliminación y borrar la información esperada.
- La sentencia WHERE puede ser opcional, pero es IMPORTANTE no olvidar usarla, ya que puede traer serios problemas en la base de datos.

EJEMPLO


Paciente
NumFicha
RutP
NombreP
Direccion
Edad
Prevision
atiende
Medico
Paciente
fechaAtencion
DetalleAtencion
BoxAtencion
Medico
IDM
RutM
NombreM
AniosEjercicio
Prevision
IDPrev
Nombre
Cobertura
Especialidad
CodigoEsp
NombreEsp
Arancel
tiene
CodigoEsp
IDM