



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Programador Web Avanzado

Clase N° 1: Clase de nivelación MySQL

Profesor: Ing. Leandro Rodolfo Gil Carrano

Web: <https://leandrogilcarrano.moodlecloud.com>

Email: leangilutn@gmail.com

Me Presento

- Leandro Gil Carrano
- Ingeniero en Sistemas de Información UTN FRBA
- 3 Años como desarrollador PHP
- Actualmente Analista Funcional
- Proyectos en los que he trabajado
 - Sitios satélites de despegar
 - Cotizador para el mercado de la construcción.
 - Cotizador para el mercado de la compra de medicamentos.

Ahora se presentan ustedes

- Nombre y apellido
- Situación laboral
- Conocimientos previos (en PHP u otro lenguaje)
- ¿Por qué se anotador al curso?
- ¿Qué expectativas tienen del curso?

Contacto

Contacto

- Plataforma online:
 - <https://leandrogilcarrano.moodlecloud.com>
- Email:
 - leangilutn@gmail.com

Solo para casos excepcionales, consultas realizarlas por la plataforma
- Link videos YouTube:
 - <https://www.youtube.com/playlist?list=PLhA16ilUESxkyUFdIIIpmmsr1jAW6Yoxf>

Como inscribirse en la plataforma

- Enviar un email con los siguientes datos:
 - Asunto: “Matricula Nombre y Apellido”
 - Cuerpo del email:
 - Nombre
 - Apellido
- Luego les llegara un email a la cuenta de correo desde donde lo enviaron, con un usuario y contraseña para que puedan acceder.

Temario

Clase 1

- Clase de nivelación
 - Repaso de conocimiento en MySql
 - Repaso de conocimientos en JS
 - Utilización de GIT
 - Utilización de Trello

Clase 2

- Javascript JQuery y JSON
 - Javascript JQuery y JSON
 - Repaso de sintaxis de javascript
 - Qué es el DOM

Clase 3 – Javascript JQuery y JSON

- Ejemplos de manipulación de DOM
- Ejemplos de manipulación de INPUT
- Ejercicios en javascript

Clase 4

- Concepto de AJAX, asincrónico, velocidad, no recarga, tunel/canal.

Clase 5

- Que es NodeJS – Instalación.
- Características fundamentales.
- Módulos de Node, inclusión de módulos, creación de módulos propios
- Como definir un evento en Node.

Clase 6

- Como servir archivos estaticos, css, js, html, etc
- Como recuperar datos enviados por un formulario o de una url por GET.
- Comando npm para instalar paquetes
- Ejemplos de paquetes: nodemailer, nodemon, formidable.

Conexión a Mysql

- Instalación de Express. Sistemas de plantillas JADE, HandleBers, etc
- Manejo de rutas en Express. Ejemplos.

Clase 7

- Uso de cookies y variables de session
- Introducción a MongoDB. Creacion de un API REST con Mongo y Node
- Algunos paquetes y herramientas de utilidad (SocketIO, PassportJS, Auth0)

Clase 8

- Armado de un ecommerce

Clase 9

- Que es Angular 2. Como incluirlo en nuestros sitios. Patron MVC.
- Controladores, modelos, vistas.
- Ejemplos de uso.

Clase 10

- Cuáles son las directivas esenciales en Angular
- Como crear módulos, como crear componentes dentro de un modulo y usarlos en otros módulos

Clase 11

- Enrutador de componentes en Angular el nuevo SPA
- Ejemplos
- Fomularios de datos en Angular 2
- Uso de APIs JSON
- Ejemplos de uso.

Clase 12

- Creación de componentes , directivas y módulos propios

Clase 13

- Armado de ecommerce con Angular + Node js

Clase 14

- Que es React. Características principales.
- Componentes y JSX.
- Ejemplos de uso.

Clase 15

- Propiedades en los componentes.
- Interaccion entre Componentes.
- Ejemplos.

Clase 16

- Eventos y firebase.
- Leer, crear y actualizar datos en firebase

Clase 17

- Armado de ecommerce React + Node js

Clase 18

• Entrega Final

Trabajo Practico

Trabajo Práctico

- Desarrollo de un sitio web o sistema web en la cual se apliquen conceptos vistos en el curso.
- Se habilitará en la plataforma web una tarea para que presenten el desarrollo que quieran realizar.
- Tendrán tiempo para realizar la presentación hasta la clase 8
- Se puede utilizar jquery, javascript, angular, React y/o node.
- NO COMENZAR A DESARROLLAR HASTA QUE LA PROPUESTA ESTE APROBADA.

Ejemplos

- Carrito de compras
- Ecommerce sin carrito de compras
- Diario digital
- Sistema de cobranzas
- Sistemas de cotizaciones
- Directorio de negocios

Herramientas

Herramientas recomendadas

- IDE
 - Sublime Text
 - Visual Studio Code
- MySql
 - MySql Workbrench
 - Mysql Query Browser
 - PhpMyAdmin
- ClipX
 - Solo para windows

Base de datos

Definición

- Es una colección de datos organizados que permite su rápida búsqueda y recuperación.
- Una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

Clasificación-Según la variabilidad de la base de datos

- **Bases de datos estáticas**

Son bases de datos únicamente de lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones, tomar decisiones y realizar análisis de datos para inteligencia empresarial.

- **Bases de datos dinámicas**

Son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y edición de datos, además de las operaciones fundamentales de consulta.

Clasificación-Según la variabilidad de la base de datos

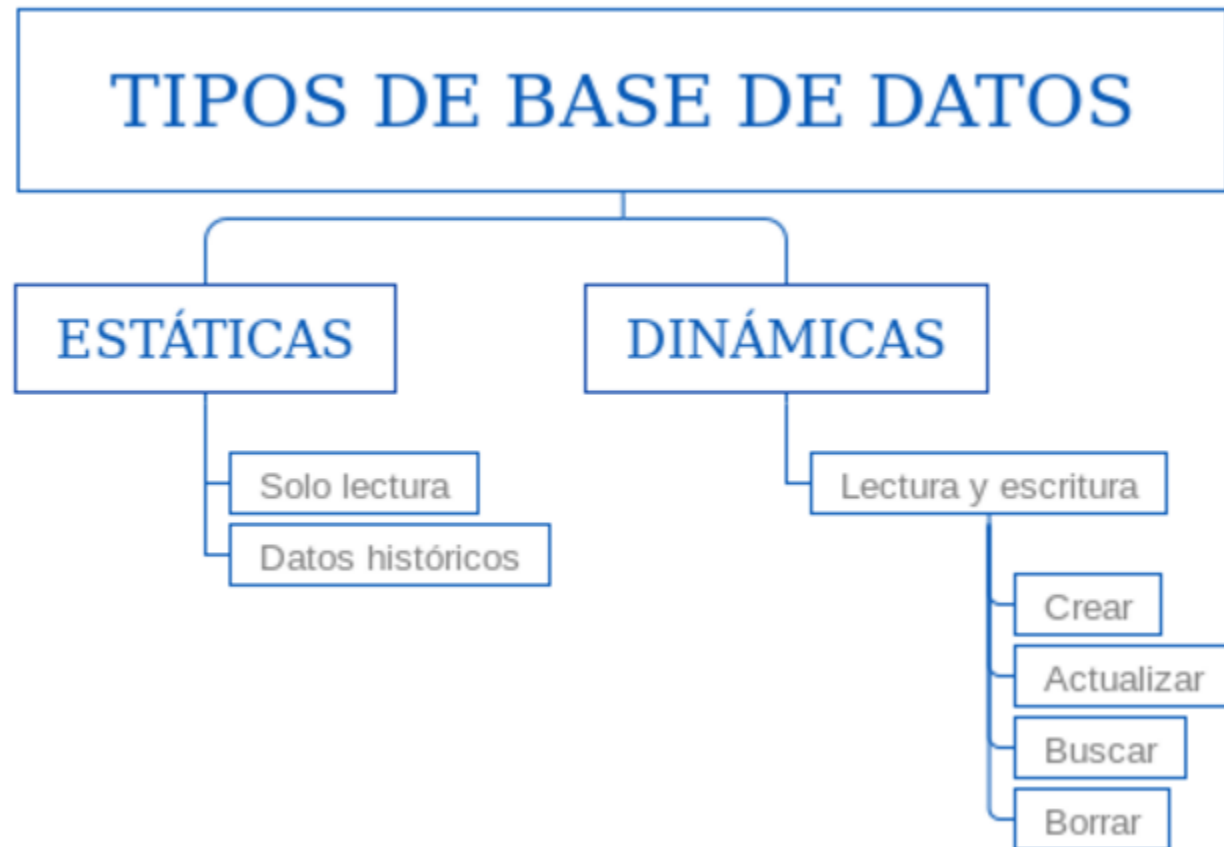
- **Bases de datos estáticas**

Son bases de datos únicamente de lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones, tomar decisiones y realizar análisis de datos para inteligencia empresarial.

- **Bases de datos dinámicas**

Son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y edición de datos, además de las operaciones fundamentales de consulta.

Clasificación-Según la variabilidad de la base de datos



Modelos

- **Bases de datos jerárquicas**

En este modelo los datos se organizan en forma de árbol invertido en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Modelos

- **Bases de datos transaccionales**

Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial, es importante entender que su fin único es recolectar y recuperar los datos a la mayor velocidad posible.

Modelos

- **Bases de datos relacionales**

Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas".

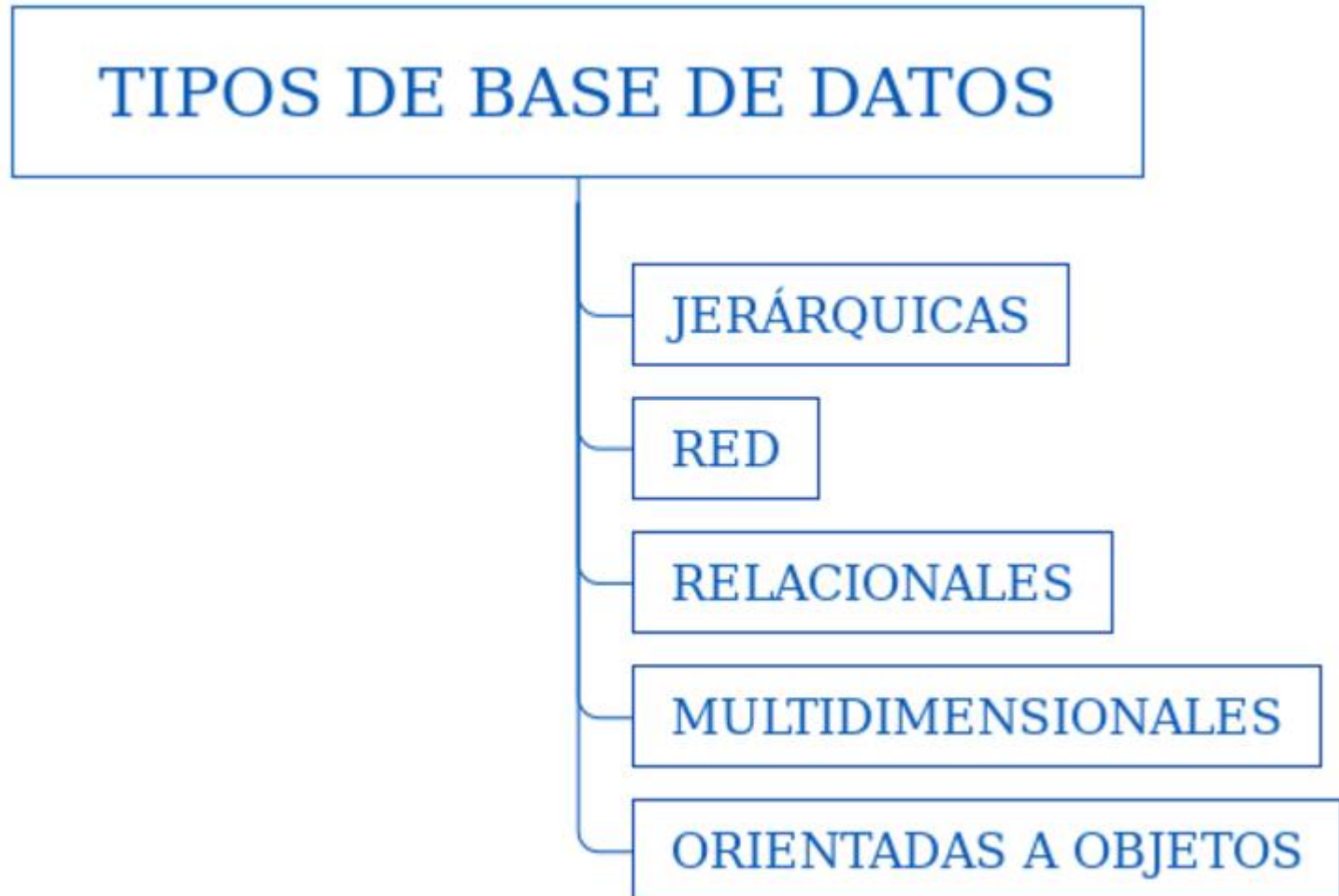
En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

Modelos

- **Bases de datos relacionales**

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Modelos

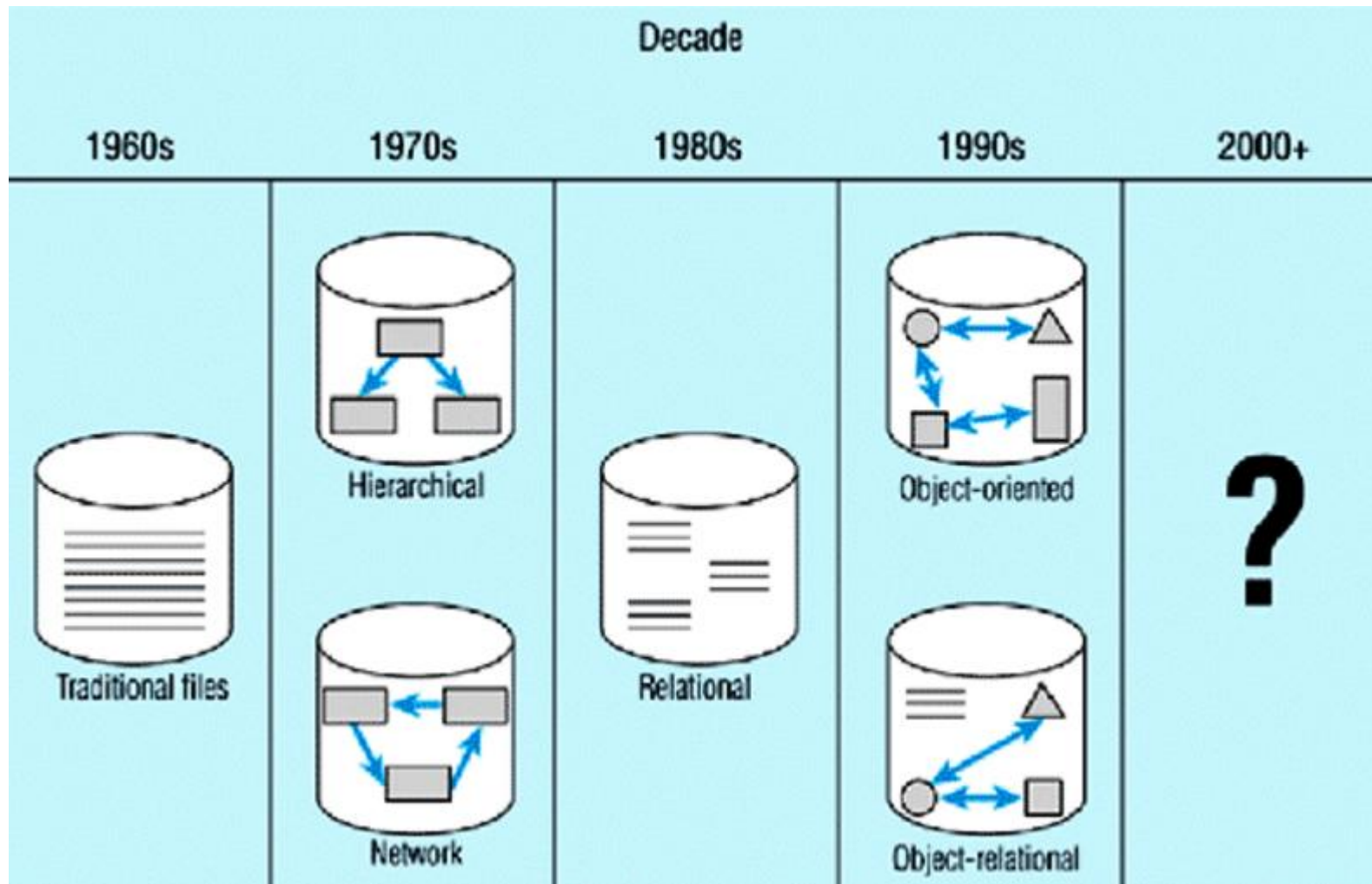




UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Historia



Definiciones Importantes

- **Tabla:** Es una colección de datos relacionados que se mantienen en un formato estructurado dentro de la base de datos (Similar a una hoja en Excel)
- **Campo (columna):** conjunto de valores de un mismo tipo (Similar a una columna en Excel)
- **Registro (fila):** los valores que toma cada campo (Similar a una fila en Excel)
- **Clave primaria (Primary Key):** Es el campo que nos permite identificar unívocamente un registro de la misma tabla
- **Clave foránea:** Es el campo que nos permiten identificar unívocamente un registro de otra tabla

Definiciones Importantes

Curso	Alumno
nombre	nombre
torno	edad



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Definiciones Importantes

Untitled 1 - LibreOffice Calc

File Edit View Insert Format Tools Data Window Help

Liberation Sans 10

F15 f(x) Σ =

	A	B	C	D	E
1	<u>Nombre</u>	<u>Turno</u>			
2	Professional Webmaster	Noche			
3	Experto universitario en PHP	Tarde			
4	Professional Webmaster	Noche			
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Sheet 1 of 2 Default Sum=0 100%

ejemplo.ods - LibreOffice Calc

File Edit View Insert Format Tools Data Window Help

Liberation Sans 10

B11 f(x) Σ =

	A	B	C	D	E	F	G
1	<u>Nombre</u>	<u>Edad</u>					
2	Juan Perez	25					
3	Jorge Gonzalez	30					
4	Maria Lopez	28					
5	Julian Sanchez	34					
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Sheet 2 of 2 Default Sum=0 100%

Definiciones Importantes

- SQL (Foreign Key) Structured Query Language: Es un tipo de lenguaje declarativo para el acceso a bases de datos relacionales
 - Permite crear, modificar, y eliminar tablas, campos y relaciones
 - Permite guardar, recuperar, buscar, actualizar, y borrar registros
 - Permite administrar permisos de acceso a los datos

MySQL

Definición

MySQL es un "gestor de bases de datos", es decir, una aplicación informática que se usa para almacenar, obtener y modificar datos (realmente, se les exige una serie de cosas más, como controlar la integridad de los datos o el acceso por parte de los usuarios, pero por ahora nos basta con eso).

InnoDB vs MyIsam

MyIsam

- Bloqueo de tabla
- Aumento del rendimiento si nuestra aplicación realiza un elevado número de consultas “Select”.
- Las tablas pueden llegar a dar problemas en la recuperación de datos.
- Permite hacer búsquedas full-text
- Menor consumo memoria RAM
- Mayor velocidad en general a la hora de recuperar datos.
- Recomendable para aplicaciones en las que dominan las sentencias SELECT ante los INSERT / UPDATE.
- Ausencia de características de atomicidad ya que no tiene que hacer comprobaciones de la integridad referencial, ni bloquear las tablas para realizar las operaciones.

InnoDB vs MyIsam

InnoDB

- Bloqueo de registros
- Soporte de transacciones
- Rendimiento
- Concurrencia
- Confiabilidad
- Nos permite tener las características ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad en español), garantizando la integridad de nuestras tablas.
- Si nuestra aplicación hace un uso elevado de INSERT y UPDATE notaremos un aumento de rendimiento con respecto a MyISAM.

Crear tabla

```
CREATE TABLE <nombre_tabla> (  
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    <nombre_campo> <tipo_campo>,  
    ...  
    <nombre_campo> <tipo_campo>  
);
```

Ejemplo

```
CREATE TABLE curso (  
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100),  
    turno VARCHAR(100)  
);
```

Modificar tabla

```
ALTER TABLE <nombre_tabla> DROP COLUMN <nombre_columna>;
```

Eliminar tabla

```
DROP TABLE alumno
```

```
ALTER TABLE curso DROP COLUMN codigo;
```

Eliminar todos los registros de una tabla

```
TRUNCATE alumno;
```

Insertar Datos

Sintaxis

```
INSERT INTO <nombre_tabla>  
    (<columna_1>, <columna_2>, ..., <columna_n>)  
VALUES (<valor_col_1>, <valor_col_2>, ..., <valor_col_n>);
```

Ejemplo

```
INSERT INTO curso (nombre, turno) VALUES ('Webmaster 1', 'Noche');
```

Actualizar Datos

Sintaxis

```
UPDATE <nombre de la tabla>  
    SET <nombre de campo>=<nuevo valor>, <nombre de campo>=<nuevo valor>, ...  
    WHERE <listado de condiciones>
```

Ejemplo

```
UPDATE curso  
    SET nombre='Webmaster introductorio'  
    WHERE nombre='Webmaster 1';
```

Eliminar Datos

Sintaxis

```
DELETE FROM <nombre de la tabla>  
    WHERE <listado de condiciones>
```

Ejemplo

```
DELETE FROM curso  
    WHERE nombre like '%introductorio%';
```



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Consultar datos

Sintaxis

```
SELECT <listado de columnas separadas por coma o *>  
FROM <nombre de la tabla>  
WHERE <listado de condiciones>;
```

Ejemplo

```
SELECT nombre, id  
FROM curso  
WHERE id>3;
```

Ejemplo promedio de edades

```
SELECT AVG(edad)  
FROM alumno;
```

Ejemplo ordenacion (nombre del curso descendente)

```
SELECT id,nombre  
FROM curso  
ORDER BY nombre DESC;
```

Ejemplo contar registros

```
SELECT count(*)  
FROM curso;
```

Ejemplo sumar edades

```
SELECT SUM(edad)  
FROM alumno;
```

Ejemplo buscar cursos que contengan la palabra web

```
SELECT *  
FROM curso  
WHERE nombre like '%web%';
```

Ejemplo buscar cursos que contengan la palabra web y tengan ID mayor a 2

```
SELECT *  
FROM curso  
WHERE nombre like '%web%' and id>2;
```

Limitar la cantidad de resultados (3 registros comenzando con el primer registro)

```
SELECT *  
FROM alumno  
LIMIT 3  
OFFSET 0
```

Tipos de datos

Enteros

- **TINYINT**: Entero muy pequeño (puede tomar el valor desde -128 a 127)
- **SMALLINT**: Entero pequeño (puede tomar el valor desde -32.728 a 32.727)
- **MEDIUMINT**: Entero medio (puede tomar el valor desde -8.388.608 a 8.388.607)
- **INT**: Entero (puede tomar el valor desde -2.147.483.648 a 2.147.483.647)
- **BIGINT**: Entero grande (puede tomar el valor desde -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807)
- **BIT**: Booleano (puede tomar el valor 0 o 1)

Tipos de datos

Reales

- **DECIMAL:** Parte entera + decimal (Se debe especificar la cantidad total de dígitos y cuantos son decimales) Ejemplo

DECIMAL(5,2) 123,45

- **FLOAT:** Para operaciones matemáticas (se debe especificar la cantidad total de dígitos y decimales).

Tipos de datos

Cadenas

- CHAR:** Cadena con una longitud fija que se especifica (longitud máxima 255)
- VARCHAR:** Cadena hasta la longitud que se especifica (longitud máxima 65.535)
- TEXT:** Cadena de texto muy larga
- BLOB:** Datos binarios

Tipos de datos

Fecha y Hora

- **DATE:** Para guardar una fecha en formato AAAA-MM-DD
- **TIME:** Para guardar una hora en formato HH:MM:SS
- **DATETIME:** Para guardar una fecha y hora en formato AAAA-MM-DD HH:MM:SS

Operadores

Aritméticos

- +: Suma
- : Resta
- *: Multiplicación
- /: División
- %: Resto de la división

Operadores

Comparación

- =**: Verifica que los valores sean iguales. $5 = 3 \Rightarrow \text{False}$
- !=** o **<>**: Verifica que los valores NO sean iguales. $5 \neq 3 \Rightarrow \text{True}$
- <**: Verifica que el valor de la izquierda sea menor al de la derecha. $5 < 3 \Rightarrow \text{False}$
- >**: Verifica que el valor de la izquierda sea mayor al de la derecha. $5 > 3 \Rightarrow \text{True}$
- <=**: Verifica que el valor de la izquierda sea menor o igual al de la derecha. $5 \leq 3 \Rightarrow \text{False}$
- >=**: Verifica que el valor de la izquierda sea mayor o igual al de la derecha. $5 \geq 3 \Rightarrow \text{True}$

Operadores

Lógicos

- AND**: Verifica que se cumplan las dos condiciones. $2 < 3$ AND $3 < 5$
 \Rightarrow True
- BETWEEN**: Verifica que un valor esté entre otros dos. 3 BETWEEN 2 AND $5 \Rightarrow$ True
- IN**: Verifica que un valor esté dentro de una lista de valores. 2 IN $(1,2,3,4,5) \Rightarrow$ True
- LIKE**: Verifica que un valor esté contenido en otro (string) 'Jorge Luis Borges' like '%Luis%' \Rightarrow True
- NOT** o **!**: Niega el resultado de la siguiente operación. 3 NOT BETWEEN 2 AND $5 \Rightarrow$ False

Funciones

- MAX:** Retorna el valor máximo de los valores. $\text{MAX}(20,100,180) \Rightarrow 180$
- MIN:** Retorna el valor mínimo de los valores. $\text{MIN}(20,100,180) \Rightarrow 20$
- SUM:** Retorna la suma de los valores. $\text{SUM}(20,100,180) \Rightarrow 300$
- COUNT:** Retorna la cantidad de valores. $\text{COUNT}(20,100,180) \Rightarrow 3$
- AVG:** Retorna el promedio entre los valores. $\text{AVG}(20,100,180) \Rightarrow 100$

Joins

Ejercicio 1

Nos permite "unir" los resultados de dos o más tablas relacionadas

Ejemplo: si queremos obtener el nombre del alumno, y el curso que está tomando

Tipo	Descripcion
Inner Join	Los resultados están en las dos tablas relacionadas
Left Join	Los resultados están en las dos tablas relacionadas, o en la tabla izquierda
Right Join	Los resultados están en las dos tablas relacionadas, o en la tabla derecha

Joins – INNER JOIN

Inner Join

Muestra solo los resultados de ambas tablas



Ejemplo

```
select *  
  from alumno  
  INNER JOIN curso ON curso.id=alumno.curso_id
```




UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Joins – INNER JOIN

alumno

id	nombre	edad	curso_id
3	Juan Perez	25	1
4	Jorge Gonzalez	30	2
5	Maria Lopez	28	3
6	Julian Sanchez	34	1
7	Carlos SinCurso	38	NULL

curso

id	nombre	turno
1	Professional Webmaster	Noche
2	Experto Universitario en PHP	Tarde
3	Professional Webmaster	Tarde
4	Professional Webmaster (vacio)	Mañana

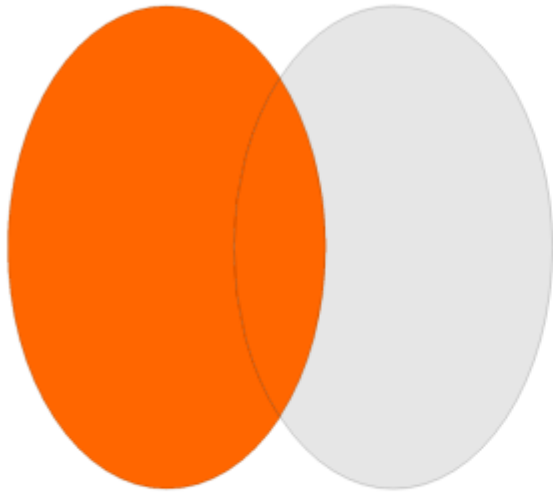
```
select *  
  from alumno  
  INNER JOIN curso ON curso.id=alumno.curso_id
```

id	nombre	edad	curso_id	id	nombre	turno
3	Juan Perez	25	1	1	Professional Webmaster	Noche
6	Julian Sanchez	34	1	1	Professional Webmaster	Noche
4	Jorge Gonzalez	30	2	2	Experto Universitario en PHP	Tarde
5	Maria Lopez	28	3	3	Professional Webmaster	Tarde

Joins – LEFT JOIN

Left Join

Muestra solo los resultados que están en ambas tablas o en la tabla de la izquierda



Ejemplo

```
select *  
  from alumno  
  LEFT JOIN curso ON curso.id=alumno.curso_id
```



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Joins – LEFT JOIN

alumno

id	nombre	edad	curso_id
3	Juan Perez	25	1
4	Jorge Gonzalez	30	2
5	Maria Lopez	28	3
6	Julian Sanchez	34	1
7	Carlos SinCurso	38	NULL

curso

id	nombre	turno
1	Professional Webmaster	Noche
2	Experto Universitario en PHP	Tarde
3	Professional Webmaster	Tarde
4	Professional Webmaster (vacio)	Mañana

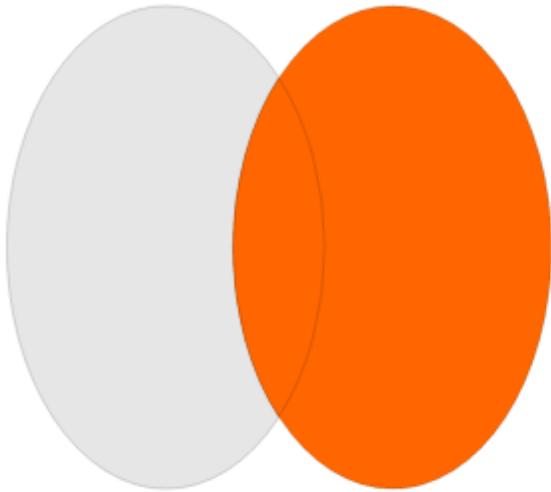
```
select *  
  from alumno  
 left join curso ON curso.id=alumno.curso_id
```

id	nombre	edad	curso_id	id	nombre	turno
1	Juan Perez	25	1	1	Professional Webmaster	Noche
2	Jorge Gonzalez	30	2	2	Experto Universitario en PHP	Tarde
3	Maria Lopez	28	3	3	Professional Webmaster	Tarde
4	Julian Sanchez	34	1	1	Professional Webmaster	Noche
5	Carlos SinCurso	38	NULL	NULL		NULL

Joins – RIGHT JOIN

Right Join

Muestra solo los resultados que están en ambas tablas o en la tabla de la derecha



Ejemplo

```
select *  
  from alumno  
  RIGHT JOIN curso ON curso.id=alumno.curso_id
```



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Joins – RIGHT JOIN

alumno

id	nombre	edad	curso_id
3	Juan Perez	25	1
4	Jorge Gonzalez	30	2
5	Maria Lopez	28	3
6	Julian Sanchez	34	1
7	Carlos SinCurso	38	NULL

curso

id	nombre	turno
1	Professional Webmaster	Noche
2	Experto Universitario en PHP	Tarde
3	Professional Webmaster	Tarde
4	Professional Webmaster (vacio)	Mañana

```
select *  
  from alumno  
 right join curso on curso.id=alumno.curso_id
```

id	nombre	edad	curso_id	id	nombre	turno
1	Juan Perez	25	1	1	Professional Webmaster	Noche
4	Julian Sanchez	34	1	1	Professional Webmaster	Noche
2	Jorge Gonzalez	30	2	2	Experto Universitario en PHP	Tarde
3	Maria Lopez	28	3	3	Professional Webmaster	Tarde
NULL	NULL	NULL	NULL	4	Professional Webmaster (vacio)	Mañana

IF

Permite tomar decisiones respecto a cada valor que se retorna en un SELECT.

Sintaxis

```
IF(<condición>,  
    <valor que retorna si la condición se cumple>,  
    <valor que retorna si la condición NO se cumple>)
```

Ejemplo

```
SELECT nombre, edad, IF(edad>25, 'ADULTO', 'JOVEN')  
FROM alumno
```

Ejercicios

Ejercicio 1

Utilizando phpMyAdmin

- Creemos una base de datos
- Creemos la tabla curso
- Creemos la tabla alumno (recordemos la clave foránea con curso)
- Ingresemos algunos datos
- Realicemos un backup
- Restauremos el [backup](#)

Ejercicio 2

Utilizando phpMyAdmin

- Listar el nombre de cada alumno y el turno (debe tener un curso asignado)
- Listar el nombre de cada alumno y el turno (tenga o no curso asignado)
- Listar el nombre del curso y el nombre de cada alumno de los cursos por la tarde
- Mostrar el alumno con mayor edad
- Mostrar el promedio de edad de los alumnos
- Mostrar la cantidad de cursos

Bibliografía

- <https://dev.mysql.com/doc/refman/5.7/en/>
- <http://www.nachocabanes.com/sql/curso/sql02.php>
- <http://www.tutorialesprogramacionya.com/mysqlya/>
- Fuente: Profesor Orlando Brea