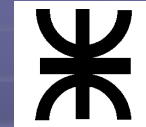


UTN Facultad Regional Avellaneda Departamento Electrónica

Materia: Informática II

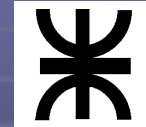
Tema: Introducción a las Bases de Datos



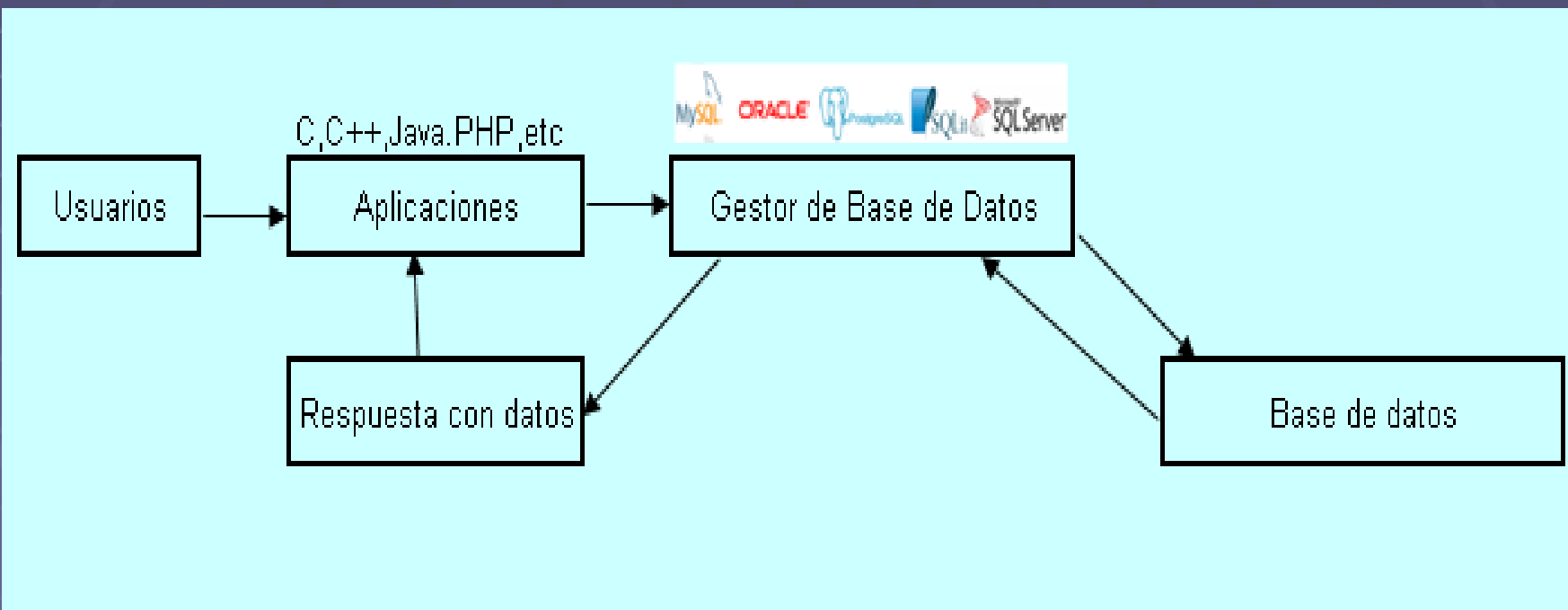
Que es una Bases de Datos.

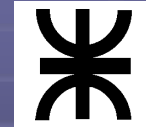
Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.



Enfoque de una base de datos.

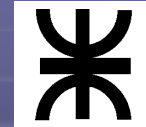




Que es un Gestor de Base de Datos.

Un Sistema Gestor de Bases de Datos (SGBD) es un conjunto de programas no visibles que administran y gestionan la información que contiene una base de datos.

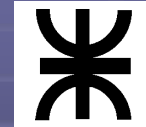
El objetivo de servir de interfaz entre Base de Datos, el usuario y las aplicaciones.



Que es SQL.

SQL : Structured Query Language(Lenguaje de consulta estructurada)

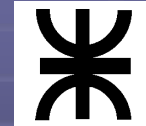
Es un lenguaje específico, utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de base de datos relacionales.



Que es una Tabla.

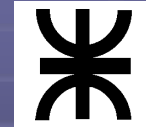
En una Base de Datos, los datos se organizan en tablas.

- Una base de datos puede contener varias tablas, y cada tabla almacena información.
- Las tablas son objetos compuestos por una estructura que almacenan información interrelacionada (filas) acerca de algún objeto en general.
- Las tablas tienen un solo nombre y es único en toda la base de datos.
- Están compuestas por registros (filas) y campos (columnas).



Que es una Tabla.

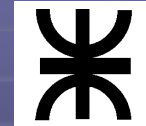
Legajo	Apellido	Nombre	Dni
100300	Gomez	Alejandro	24825667
100403	Alfaro	Juan	25001587
101008	Moreno	Marcelo	28251234
102008	Kepler	Pedro	29250113
101003	Gauss	Jordan	30001233



Restricciones de las Tablas.

Restricciones

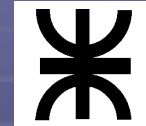
- Los nombres de las tablas deben ser únicos en la base de datos.
- Los nombres de las columnas debe ser únicos en la tabla.
- No pueden haber dos registros con el mismo valor de la clave primaria.-



Clave Primaria (PRIMARY KEY)

La clave primaria (PRIMARY KEY) es una columna o una combinación de columnas cuyos valores identifican de forma única a cada fila de la tabla.

Una tabla sólo puede tener una restricción PRIMARY KEY y ninguna columna a la que se aplique una restricción PRIMARY KEY puede aceptar valores NULL.

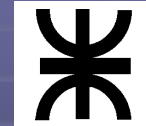


Tipo de datos primitivos.

Cuando se diseña las tablas tenemos que especificar el tipo de datos y tamaño que podrá almacenar cada campo.

Se debe tratar que la tabla no se quede corta en su capacidad, que destine un tamaño apropiado a la longitud de los datos, y la máxima velocidad de ejecución.

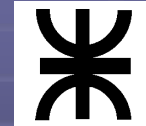
Para este ejemplo usaremos los tipos de datos del gestor **mysql**. Este admite dos tipos de datos: números y cadenas de caracteres. Hay otros tipos de datos especiales se que admiten especiales por ejemplo: formatos de fecha, etc.



Tipo de datos primitivos: Numéricos.

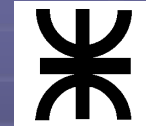
En este tipo de campos sólo pueden almacenarse números, positivos o negativos, enteros o decimales, en notación hexadecimal, científica o decimal.

Tipo de dato	Atributos	Tamaño
<i>integer</i>	SIGNED/UNSIGNED ZEROFILL	INT(5)
BIT o BOOL	Pueden ser 0 o 1	-
TINYINT	UNSIGNED	0 - 255 ó -128 -127
SMALLINT	UNSIGNED	-32768 a 32767 ó 0 a 65535
MEDIUMINT	UNSIGNED	-8388608 a 8388607 ó 0 a 16777215
INT	UNSIGNED	-2147463848 a 2147483647 ó 0 a 4294967295
BIGINT	UNSIGNED	-9223372036854775808 a 9223372036854775807 0 a 18446744073709551615
FLOAT (m,d)		-3.402823466E+38 a -1.175494351E-38 1.175494351E-38 a 3.402823466E+38.
DOUBLE		-1.7976931348623157E+308 a -2.2250738585072014E-308 2.2250738585072014E-308 a 1.7976931348623157E+308
DECIMAL	Número en coma flotante desempquetado. El número se almacena como una cadena	



Tipo de datos primitivos: Caracteres.

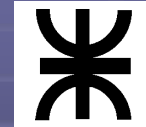
Tipo de dato	Características	Tamaño
CHAR	Su longitud abarca desde 1 a 255 caracteres.	Siempre el tamaño máximo
VARCHAR	rango de 1-255 caracteres de longitud variable.	Solo almacena la longitud del dato



Tipo de datos primitivos: Varios.

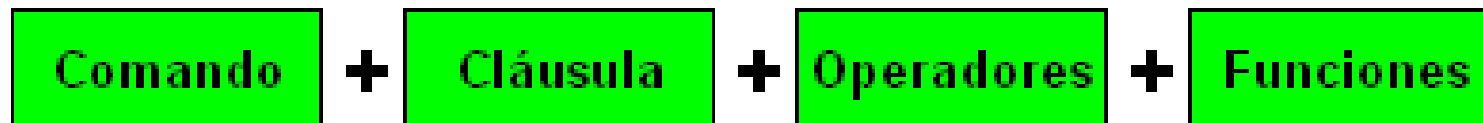
Tipo de dato	Descripción	Formato
DATE	Almacena fechas	YYYY MM DD
DATETIME	Combinación de fecha y hora	año-mes-día horas:minutos:segundos
TIME	Almacena una hora	'HH:MM:SS'
YEAR	almacena un año	Tamaño dos o cuatro dígitos

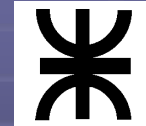
Nota: Existen otros tipos de datos que no utilizaremos en este curso.



Componentes del SQL

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.





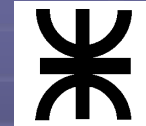
Comandos SQL

DDL (Data Definition Language): Permite crear y modificar la estructura de una base de datos.

DML (Data Manipulation Language): Permite recuperar, almacenar, modificar, eliminar, insertar y actualizar datos de una base de datos.

DCL (Data Control Language): Permite crear roles, permisos e integridad referencial, así como el control al acceso a la base de datos.

TCL (Transactional Control Language): Permite administrar diferentes transacciones que ocurren dentro de una base de datos.



Comandos SQL: DDL

CREATE: Crea nuevas tablas, campos e índices.

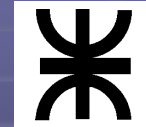
ALTER: Modificar las tablas agregando campos o cambiando la definición de los campos.

DROP: Elimina tablas e índices.

TRUNCATE: Eliminar todos los registros de una tabla.

COMMENT: Agrega comentarios al diccionario de datos.

RENAME: Se utiliza para renombrar objetos.



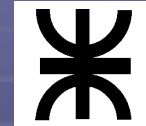
Comandos SQL: DML

SELECT: Consulta registros de la base de datos que satisfagan un criterio determinado.

INSERT: Se usa para cargar de datos en la base de datos en una única operación.

UPDATE: Utilizado para modificar los valores de los campos y registros especificados

DELETE: Utilizado para eliminar registros de una tabla de una base de datos.



Cláusulas

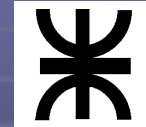
FROM: Utilizada para especificar la tabla de la cual se van a seleccionar los registros

GROUP BY: Utilizada para separar los registros seleccionados en grupos específicos

HAVING: Utilizada para expresar condición que debe satisfacer cada grupo

ORDER BY: Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

WHERE: Utilizada para determinar los registros seleccionados en la cláusula FROM



Operadores

Operadores Lógicos

AND, OR, NOT

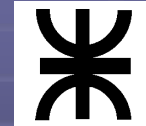
Operadores de comparación

< > <> <= >=

BETWEEN

LIKE

IN



Funciones

Se usan dentro de una cláusula **SELECT** en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

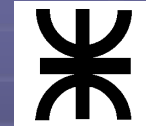
AVG Calcula el promedio de los valores de un campo determinado

COUNT Devuelve el número de registros de la selección

SUM Devuelve la suma de un campo determinado

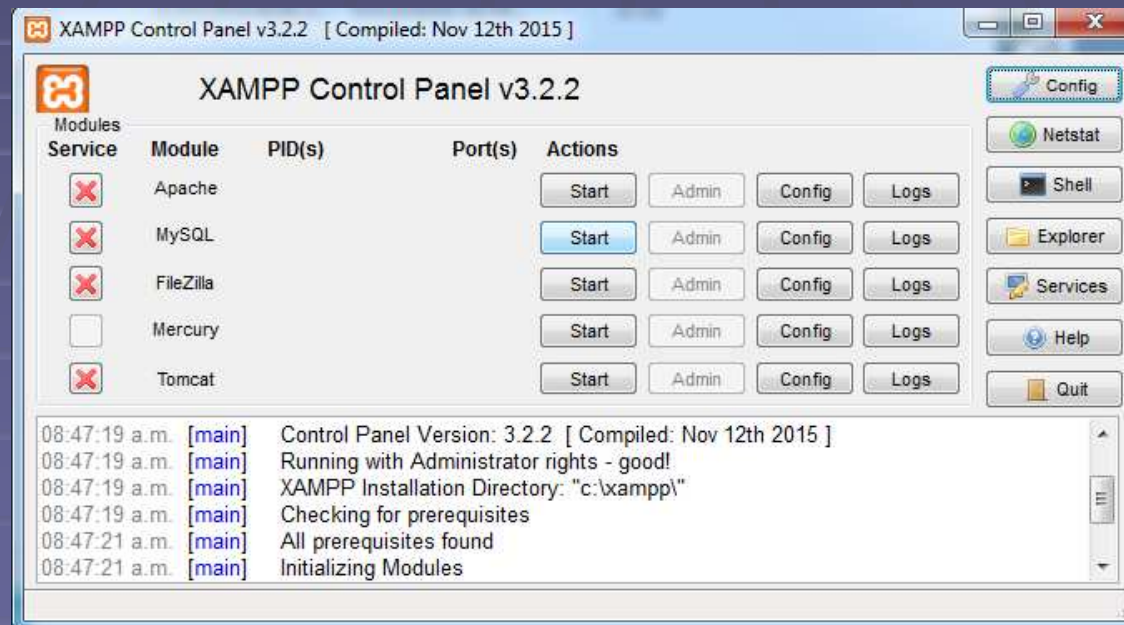
MAX Devuelve el valor más alto de un campo especificado

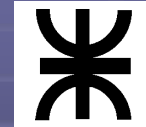
MIN Devuelve el valor más bajo de un campo especificado



Práctica

Lo primero es arrancar el mysql. En este ejemplo usamos en XAMPP.

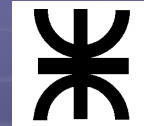




Práctica

También usaremos para la práctica **Workbench** que es una herramienta visual de, Administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL.

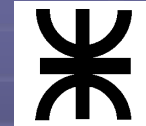




Práctica

```
Query 1 x
1 create database if not exists estudiantes;
2 use estudiantes;
3 |
```


```
CREATE TABLE alumnos (
  legajo int(6) unsigned NOT NULL,
  nombre varchar(20) NOT NULL default '',
  apellido varchar(20) NOT NULL default '',
  fechaNac date NOT NULL default '0000-00-00',
  idcarrera int(6),
  PRIMARY KEY (legajo));
```





Práctica

```
describe alumnos;
```

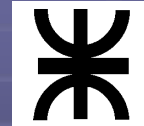
Result Grid

 Filter Rows:

Export: 

Wrap Cell Content: 

	Field	Type	Null	Key	Default	Extra
	legajo	int(6) u...	NO	PRI	NULL	
	nombre	varcha...	NO			
	apellido	varcha...	NO			
	fechaNac	date	NO		0000-00-00	
	idcarrera	int(6)	YES		NULL	

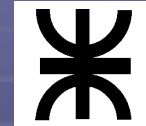


Práctica

```
CREATE TABLE especialidad(  
  idesp int(6) auto_increment  
  PRIMARY KEY,  
  especialidad varchar(60) not null);
```

```
show tables;
```


Result Grid		Filter Rows:
	Tables_in_estudiantes	
	alumnos	
	especialidad	

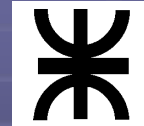


Práctica

```
alter table alumnos add  
CONSTRAINT fk_alumnos_especialidad  
FOREIGN KEY (idcarrera)  
REFERENCES especialidad(idesp)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION
```

```
describe alumnos;
```

Result Grid		Filter Rows:			Export:		Wr
	Field	Type	Null	Key	Default	Extra	
	legajo	int(6) u...	NO	PRI	NULL		
	nombre	varcha...	NO				
	apellido	varcha...	NO				
	fechaNac	date	NO		0000-00-00		
	idcarrera	int(6)	YES	MUL	NULL		

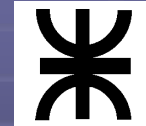


Práctica

*** Listado de Especialidades***

IdEsp.	Especialidad.
0001	Ing. Electronica
0002	Ing. Electrica
0003	Ing. Civil
0004	Ing. quimica
0005	Ing. Industrial

```
insert into especialidad(especialidad) values('Ing. Electrónica');  
insert into especialidad(especialidad) values('Ing. Electrica');  
insert into especialidad(especialidad) values('Ing. Civil');  
insert into especialidad(especialidad) values('Ing. Química');  
insert into especialidad(especialidad) values('Ing. industrial');
```



Práctica

```
select * from especialidad;
```

Result Grid			Filter Rows:
	idesp	especialidad	
	1	Ing. Electrónica	
	2	Ing. Electrica	
	3	Ing. Civil	
	4	Ing. Química	
	5	Ing. industrial	
	NULL	NULL	



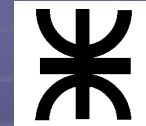
Práctica

*** Listado de Alumnos***

Legajo.	Apellido	Nombre	Fecha Nac	Carrera
1088	Lopez	Juan Carlos	19980422	2
1101	Mascote	Jose Luis	19981219	2
1089	Samudio	Marcelo	19990612	2
1105	Zorzoto	Miguel	19951204	1
1188	Zamulle	Juan Gabriel	19990706	3

Presione una tecla para continuar . . . _

```
insert into alumnos(legajo,apellido,nombre,fechanac,idcarrera)
values(1088,'Lopez','Juan Carlos','1998-04-22',2);
insert into alumnos(legajo,apellido,nombre,fechanac,idcarrera)
values(1104,'Mascote','Jose Luis','1998-12-19',2);
insert into alumnos(legajo,apellido,nombre,fechanac,idcarrera)
values(1089,'Samudio','Marcelo','1999-06-12',2);
insert into alumnos(legajo,apellido,nombre,fechanac,idcarrera)
values(1105,'zorzo', 'Miguel','1995-12-04',1);
insert into alumnos(legajo,apellido,nombre,fechanac,idcarrera)
values(1188,'Zamulle','Juan Gabriel','1999-07-06',3);
```



Práctica

```
select * from alumnos;
```

Result Grid		Filter Rows:		Edit:	
	legajo	nombre	apellido	fechaNac	idcarrera
	1088	Juan Carlos	Lopez	1998-04-22	2
	1089	Marcelo	Samudio	1999-06-12	2
	1104	Jose Luis	Mascote	1998-12-19	2
	1105	Miguel	zorzo	1995-12-04	1
	1188	Juan Gabriel	Zamulle	1999-07-06	3
	NULL	NULL	NULL	NULL	NULL



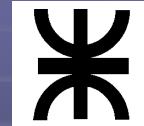
Práctica

1. colocar una especialidad que no existe

```
insert into alumnos(legajo,apellido,nombre,fechanac,idcarrera)
values(1000,'Ejemplo de','Error','1999-09-09',8);
```

Message

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('estudiantes`.`alumnos`, CONSTRAINT `fk_alumnos_especialidad` FOREIGN KEY (`idcarrera`) REFERENCES `esp...

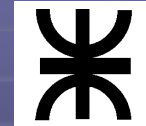


Práctica

2. colocar una clave primaria repetida

```
insert into alumnos(legajo,apellido,nombre,fechanac,idcarrera)  
values(1088,'Perez','Error','1999-09-09',4);
```


❌ 3 10:51:20 i Error Code: 1062. Duplicate entry '1088' for key 'PRIMARY'

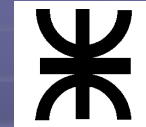


Práctica

```
CREATE TABLE Materias(  
  idmaterias int(6) auto_increment  
  PRIMARY KEY,  
  mat varchar(60) not null);
```

```
describe Materias;
```

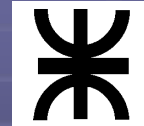
Result Grid						
Filter Rows:						
Export:  Wrap C						
	Field	Type	Null	Key	Default	Extra
	idmate...	int(6)	NO	PRI	NULL	auto_in...
	mat	varcha...	NO		NULL	



Práctica

```
Show tables;
```

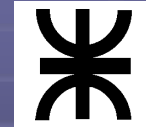
Result Grid		Filter Rows:
	Tables_in_estudiantes	
	alumnos	
	especialidad	
	materias	



Práctica

```
*** Listado de Materias***  
  
IdMaterias.      Materia.  
-----  
0001             Informatica I  
0002             Informatica II  
0003             Ingenieria Electronica  
0004             Algebra  
Presione una tecla para continuar . . . _
```

```
insert into Materias(mat) values('Informática I');  
insert into Materias(mat) values('Informática II');  
insert into Materias(mat) values('Ingeniería Electrónica');  
insert into Materias(mat) values('Algebra');
```



Práctica

```
select * from Materias;
```

Result Grid			Filter Rows:
	idmaterias	mat	
▶	1	Informática I	
	2	Informática II	
	3	Ingeniería Electrónica	
	4	Algebra	
⊙	NULL	NULL	

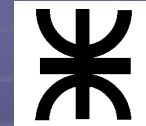


Práctica

```
CREATE TABLE Maprobadas(  
    legajo int(6) unsigned NOT NULL,  
    idmaterias int(6),  
    nota int(2) unsigned NOT NULL,  
    fecha date NOT NULL default '0000-00-00',  
    PRIMARY KEY (legajo,idmaterias),  
    FOREIGN KEY(legajo) REFERENCES Alumnos(legajo),  
    FOREIGN KEY (idmaterias) REFERENCES Materias(idmaterias)  
);
```

```
describe Maprobadas;
```

Result Grid						
		Filter Rows:			Export:	Wrap Cell Content:
	Field	Type	Null	Key	Default	Extra
►	legajo	int(6) unsigned	NO	PRI	NULL	
	idmate...	int(6)	NO	PRI	NULL	
	nota	int(2) unsigned	NO		NULL	
	fecha	date	NO		0000-00-00	



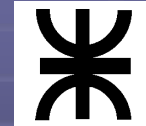
Práctica

*** Listado de Materias Aprobadas***

Legajo.	idMaterias	nota	Fecha Aprob.
1104	0002	8	20190227
1088	0004	4	20181211

Presione una tecla para continuar . . .

```
insert into Maprobadas(legajo,idmaterias,nota,fecha)
values(1104,2,8,'2019-02-27');
insert into Maprobadas(legajo,idmaterias,nota,fecha)
values(1088,4,6,'2018-12-11');
```



Práctica

```
select * from maprobadas;
```

Result Grid					Filter Rows:		Edit:
	legajo	idmaterias	nota	fecha			
▶	1088	4	6	2018-12-11			
	1104	2	8	2019-02-27			
✖	NULL	NULL	NULL	NULL			