

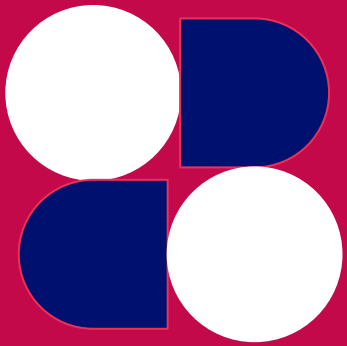


Programa PUCP DATA DRIVEN

Formación tecnológica

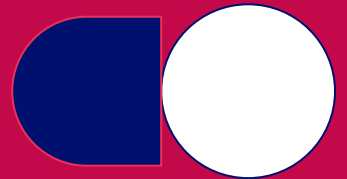
Dirección Académica de Planeamiento y Evaluación
Dirección de Gestión del Talento Humano





SESIÓN 4:

Introducción a SQL y Modelo Relacional





Al finalizar la sesión, lograrás:

- Entender los conceptos tecnológicos detrás de la gestión de datos
- Entender los fundamentos de un modelo relacional
- Diferenciar entre llaves primarias y foráneas en un modelo relacional.
- Primeros pasos con SQL



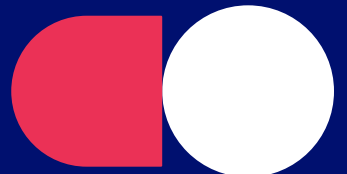
Contenido de la sesión:



- 1 Vista de datos
- 2 Modelo Relacional
- 3 Tarea
- 4 Tour por PowerBi Services
- 5 Lenguajes de Consulta Relacional
- 6 Estructura Básica de Consulta SQL



1. Vista de Datos





Gestión de Datos

- Un sistema de base de datos es una colección de datos interrelacionados y un conjunto de programas que permiten a los usuarios acceder y modificar estos datos.





Gestión de Datos

- Uno de los propósitos principales de un sistema de base de datos es proporcionar a los usuarios una visión abstracta de los datos.
- **Modelos de datos**
 - Una colección de herramientas conceptuales para describir datos, relaciones de datos, semántica de datos y restricciones de coherencia.
- **Abstracción de datos**
 - Oculta la complejidad de las estructuras de datos para representar los datos en la base de datos de los usuarios a través de varios niveles de abstracción de datos.





Gestión de Datos

- Modelos de datos

- Modelo Relacional
- Modelo de datos entidad-relación (principalmente para el diseño de bases de datos)
- Modelos de datos basados en objetos (orientados a objetos y relacionales a objetos)
- Modelo de datos semiestructurados (XML)
- Otros modelos más antiguos:
 - modelo de red
 - modelo jerárquico





Gestion de Datos

- **Niveles de Abstracción**

- **Nivel físico:** describe cómo se almacena un registro (p. ej., instructor).
- **Nivel lógico:** describe los datos almacenados en la base de datos y las relaciones entre los datos.

```
type docentes = record
  id_docente : integer;
  nombres_docente : text;
  departamento_docente: text;
  categoría_docente : text;
end;
```

- **Nivel de vista:** los programas de aplicación ocultan detalles de los tipos de datos. Las vistas también pueden ocultar información (como el salario de un empleado) por motivos de seguridad.





Instancias y Esquemas de Datos

Similar a tipos y variables en lenguajes de programación.

- **Esquema lógico:** la estructura lógica **general** de la base de datos (Análogo a definir el nombre de una variable en un programa)
 - Ejemplo: La base de datos consta de información sobre un conjunto de clientes y cuentas en un banco y la relación entre ellos.
- **Esquema físico:** la estructura física general de la base de datos adaptado al motor de base de datos. (Análogo a escribir el tipo de dato de una variable en un programa de acuerdo al lenguaje de programación)
- **Instancia:** el contenido real de la base de datos en un momento determinado (Análogo al valor contenido dentro de una variable)



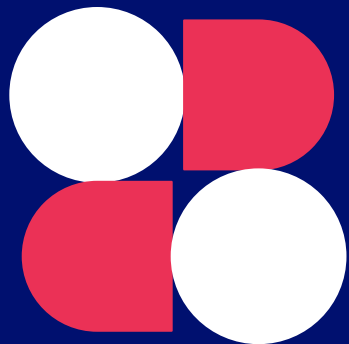


Instancias y Esquemas de Datos

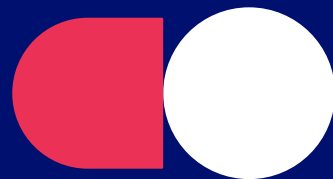
El beneficio más grande:

- ***Independencia de datos físicos:*** la capacidad de modificar el esquema físico sin cambiar el esquema lógico
 - Las aplicaciones dependen del esquema lógico.
 - En general, las interfaces entre los distintos niveles y componentes deben estar bien definidas para que los cambios en algunas partes no influyan seriamente en otras.





2. Motor de Bases de Datos





Características

- Un sistema de base de datos (DBMS) se divide en módulos que se ocupan de cada una de las responsabilidades del sistema general.
- Los componentes funcionales de un sistema de base de datos se pueden dividir en:
 - El administrador de almacenamiento,
 - El componente del procesador de consultas,
 - El componente de gestión de transacciones.





Características

Administrador de Almacenamiento: Un módulo que proporciona la interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación y las consultas enviadas al sistema.

El administrador de almacenamiento es responsable de las siguientes tareas:

- Interacción con el administrador de archivos del sistema operativo
- Almacenamiento, recuperación y actualización eficiente de datos





Características

Procesador de Consultas: Un módulo que traduce y optimiza las consultas para recuperar los datos.

Los componentes del procesador de consultas incluyen:

- Intérprete de definición
- Compilador y optimizador de modificación
- Motor de evaluación de consultas





Características

Gestión de tracciones: Un módulo que administra la concurrencia y las acciones a aplicar cuando una consulta o instrucción de modificación no se puede completar. Responde a las políticas configuradas en el motor.





Administrador de BDD (DBA)

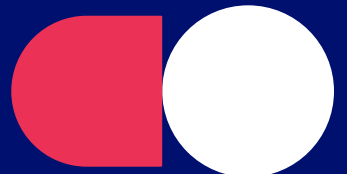
Una persona que tiene control central sobre el sistema completo de bases de datos se denomina **administrador de base de datos (DBA)**.

Es el mejor amigo de cualquier perfil de consulta de datos.





2. Modelo Relacional





Definición

- El modelo relacional, para el modelado y la gestión de bases de datos, es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos.
- Su idea fundamental es el uso de relaciones. Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados tuplas.
- La mayoría de las veces se conceptualiza pensando en cada relación como si fuese una tabla que está compuesta por registros (cada fila de la tabla sería un registro o “**tupla**”) y columnas (también llamadas “**campos**” o “**atributos**”).





Definición

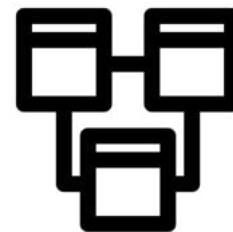
- El modelo relacional está basado en que todos los datos están almacenados en tablas (entidades/relaciones) y cada una de estas es un conjunto de datos, por tanto una base de datos es un conjunto de entidades.
- La jerarquía sería: tabla -> fila (**tupla**) -> campo (**atributo**)





Definición – Atributos

- Características de un objeto, lo que lo describe.
- El conjunto de valores permitidos para cada atributo se denomina dominio del atributo.
- Se requiere (normalmente) que los valores de los atributos sean atómicos; es decir, indivisible
- El valor especial nulo es un miembro de cada dominio. Indica que el valor es “desconocido”. El valor nulo provoca complicaciones en la definición de muchas operaciones.





MODELO RELACIONAL



Ejemplo de entidad/tabla de docentes

attributes
(or columns)

id_do cente	codigo_ docente	nombres_ docente	apellidos_ docente	dedicacion_ docente	categoria_ docente	departamento_ docente	correo_ docente	dni	sexo
1	D001	Juan Carlos	PÃ©rez GÃ³mez	Tiempo Completo	Titular	MatemÃ¡ticas	juan.perez@unive rsidad.edu	12345678	M
2	D002	MarÃ­a Fernanda	RodrÃ­guez Flores	Medio Tiempo	Asociado	FÃ­sica	maria.rodriguez@ universidad.edu	87654321	F
3	D003	Luis Alberto	GarcÃ­a MÃ©ndez	Tiempo Completo	Auxiliar	QuÃ­mica	luis.garcia@unive rsidad.edu	23456789	M
4	D004	Ana LucÃ­a	LÃ³pez DÃ­az	Medio Tiempo	Titular	BiologÃ­a	ana.lopez@univer sidad.edu	98765432	F
5	D005	Carlos Eduardo	MartÃ­nez Soto	Tiempo Parcial	Asociado	IngenierÃ­a	carlos.martinez@ universidad.edu	34567890	M
6	D006	SofÃ­a	RamÃ­rez Paredes	Tiempo Completo	Titular	Historia	sofia.ramirez@uni versidad.edu	54321678	F
7	D007	Fernando	GÃ³mez Carrillo	Tiempo Completo	Auxiliar	FilosofÃ­a	fernando.gomez @universidad.edu	67890543	M

tuples
(or rows)



Formalmente

- $A_1, A_2, \dots, A_n$ son atributos
- $R = (A_1, A_2, \dots, A_n)$ es un esquema(plantilla) de entidad
Ejemplo:
docentes = (id_docente, código_docente,)
- Una tabla de entidad r instanciada sobre el esquema R se denota por $r(R)$.
- Un elemento t de la tabla r se llama *tupla* y se representa por una fila en la tabla





MODELO RELACIONAL



OJO

- **Las tablas están desordenadas:** El orden de las tuplas es irrelevante (las tuplas pueden almacenarse en un orden arbitrario)

id_do	codigo_	nombres_	apellidos_	dedicacion_	categoria_	departamento			
cente	docente	docente	docente	docente	docente	_docente	correo_docente	dni	sexo
80D001		Juan Carlos	PÃ©rez GÃ³mez	Tiempo Completo	Titular	MatemÃ¡ticas	juan.perez@unive rsidad.edu	12345678	M
104D002		MarÃ­a Fernanda	RodrÃ­guez Flores	Medio Tiempo	Asociado	FÃ­sica	maria.rodriguez@ universidad.edu	87654321	F
3D003		Luis Alberto	GarcÃ­a MÃ©ndez	Tiempo Completo	Auxiliar	QuÃ­mica	luis.garcia@unive rsidad.edu	23456789	M
25D004		Ana LucÃ­a	LÃ³pez DÃ­az	Medio Tiempo	Titular	BiologÃ­a	ana.lopez@univer sidad.edu	98765432	F

OJO X2
Se refiere al
orden de las
tuplas, no de los
atributos



txd-faci-datamart



inscripciones		
♂ ♦ #	Id_Inscripcion	Int8
○	Id_estudiante	Int8
○	Id_horario	Int8

estudiantes		
♂ ♦ #	Id_estudiante	Int8
♦	codigo_estudiante	Text
○	Id_especialidad	Int8
♦	apellidos_estudiante	Text
♦	nombres_estudiante	Text
○	correo_estudiante	Text
○	dni	Text
○	sexo	Text

especialidades		
♂ ♦ #	Id_especialidad	Int8
♦	nombre_especialidad	Text
♦	codigo_especialidad	Text
○	rama	Text
○	seccion	Text

cursos		
♂ ♦ #	Id_curso	Int8
♦	nombre_curso	Text
○	creditaje	Int4
○	ciclo	Int4

docentes		
♂ ♦ #	Id_docente	Int8
♦	codigo_docente	Text
♦	nombres_docente	Text
♦	apellidos_docente	Text
○	dedicacion_docente	Text
○	categoria_docente	Text
○	departamento_doce...	Text
○	correo_docente	Text
○	dni	Text
○	sexo	Text

horarios		
♂ ♦ #	Id_horario	Int8
○	Id_curso	Int8
○	Id_docente	Int8
○	horario	Text
○	clave_curso	Text
○	tipo_horario	Text
○	dia	Text
○	hora_inicio	Time
○	hora_fin	Time



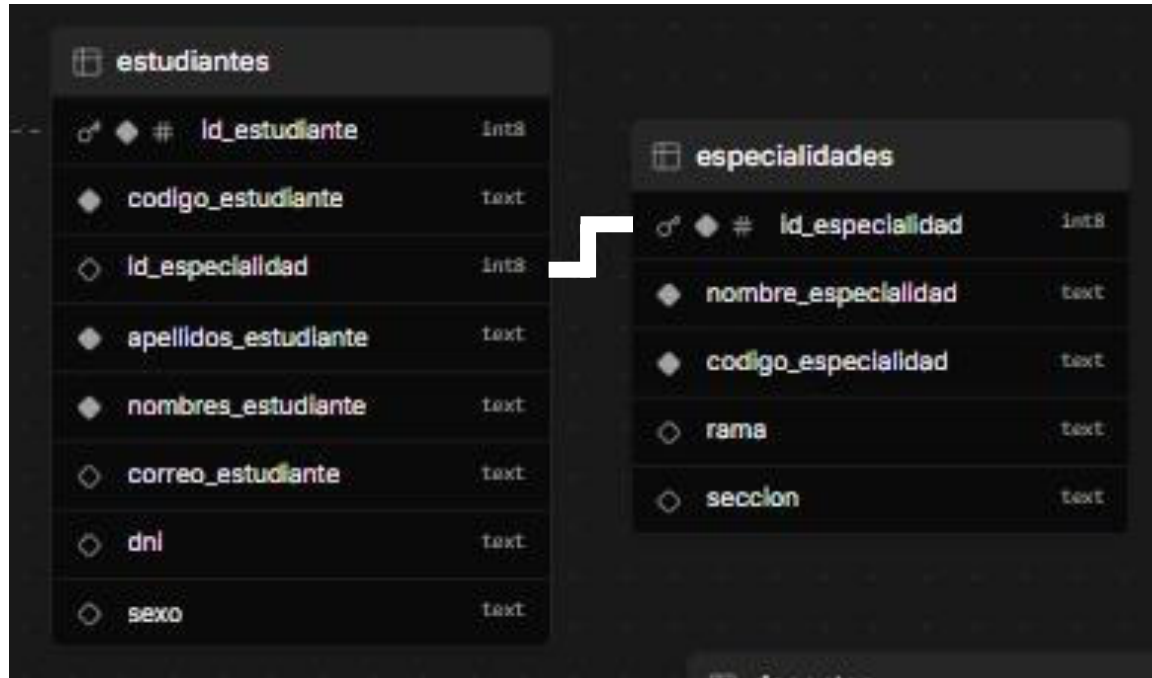
Llaves

- Una **llave primaria** es un campo (o conjunto de campos) en una tabla de una base de datos que identifica de manera única a cada registro de esa tabla. No puede haber valores duplicados ni nulos en la llave primaria.
- Una **llave foránea** es un campo en una tabla que hace referencia a la llave primaria de otra tabla. Sirve para crear una relación entre las dos tablas, asegurando la integridad referencial.





MODELO RELACIONAL



Ejemplo:

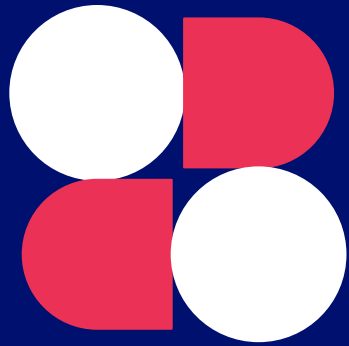
id_especialidad dentro de la entidad *estudiantes* es una **foreign key**, la cual referencia a la entidad de *especialidades*



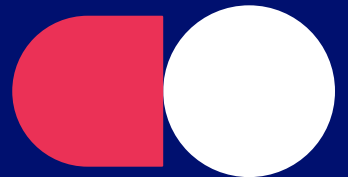
TAREA (YA!)



- Acceso a a Power Bi Services



5. Lenguajes de Consulta Relacional





LENGUAJES DE CONSULTA RELACIONAL



Lenguajes “puros”

- Álgebra relacional
 - Cálculo relacional de tuplas
 - Cálculo relacional de dominio
- Los 3 lenguajes puros son equivalentes en potencia

Lenguajes No Procedimentales

- SQL (Structured Query Language)
 - DDL
 - DML
 - DCL
 - TCL
 - PL/SQL





LENGUAJES DE CONSULTA RELACIONAL



SQL (Structured Query Language)

- ***DDL (Data Definition Language)***: Usado para definir la estructura de la base de datos, como crear, modificar o eliminar tablas. Ejemplos: **CREATE, ALTER, DROP**.
- ***DCL: (Data Control Language)***: Usado para el control de acceso a la base de datos. Ejemplos: **GRANT, REVOKE**.
- ***TCL (Transaction Control Language)***: Usado para gestionar las transacciones en una base de datos. Ejemplos: **COMMIT, ROLLBACK**.
- ***Procedural SQL (PL/SQL)***: Son lenguajes que permiten escribir bloques de código con procedimientos, bucles y condiciones dentro de la base de datos, para tareas más complejas.





LENGUAJES DE CONSULTA RELACIONAL



Data Manipulation Language (DML)

- Lenguaje para acceder, actualizar y recuperar los datos organizados por el modelo de datos apropiado
- Hay básicamente dos tipos de lenguaje de manipulación de datos
 - **DML de procedimiento:** requiere que un usuario especifique qué datos se necesitan y cómo obtener esos datos.
 - **DML declarativo:** requiere que un usuario especifique qué datos se necesitan sin especificar cómo obtener esos datos.
- Los DML declarativos suelen ser más fáciles de aprender y usar que los DML de procedimiento.
- Los DML declarativos también se denominan DML no procedimentales.





LENGUAJES DE CONSULTA RELACIONAL



Lenguaje de consulta SQL

- El lenguaje de consulta SQL es **no procedimental**. Una consulta toma como entrada varias tablas (posiblemente solo una) y siempre devuelve una sola tabla.
- Ejemplo: Encontrar todos los nombres de instructores que pertenezcan al departamento de “Dpto Acad Ingeniería”

```
select nombres_docente  
from model.[faci-inc_tb_docentes]  
where departamento_docente = 'Dpto Acad Ingeniería'
```





LENGUAJES DE CONSULTA RELACIONAL



Lenguaje de consulta SQL

- Para poder calcular funciones complejas, SQL generalmente está integrado en algún lenguaje de nivel superior
- Los programas de aplicación generalmente acceden a las bases de datos a través de uno de los siguientes métodos:
 - Extensiones de idioma para permitir SQL incorporado
 - Interfaz de programa de aplicación (p. ej., ODBC/JDBC) que permite enviar consultas SQL a una base de datos





LENGUAJES DE CONSULTA RELACIONAL



Lenguaje de consulta SQL

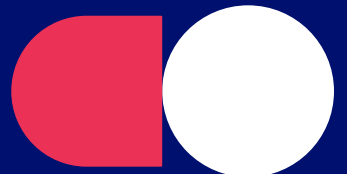
- Limitantes

- Los lenguajes de consulta no procedimentales como SQL no son tan poderosos.
- SQL no admite acciones como la entrada de los usuarios, la salida a las pantallas o la comunicación a través de la red.
- Dichos cálculos y acciones deben escribirse en un lenguaje principal, como C/C++, Java o Python, con consultas SQL integradas que acceden a los datos de la base de datos (Programas de aplicación)





6. Estructura Básica de Consulta SQL





ESTRUCTURA BÁSICA SQL



- Una consulta SQL típica tiene la forma:

```
select A1,A2,...,An  
from r1,r2,...,rm  
where P
```

- A_i representa un atributo
 - r_i representa una relación/tabla
 - P es un predicado.
- El resultado de una consulta SQL es una tabla.





La cláusula “SELECT”

- La cláusula **SELECT** enumera los atributos deseados en el resultado de una consulta
- Ejemplo: Encuentre los nombres de los docentes:

```
SELECT nombres_docente  
from model.[faci-inc_tb_docentes]
```

- **OJO:** Los nombres de los atributos en SQL son **case insensitive**
nombres_docente $\equiv$ NOMBRES_DOCENTE $\equiv$ NoMbReS_dOcEnTe



La cláusula “WHERE”

- La cláusula **WHERE** especifica las condiciones que el resultado debe satisfacer. Generalmente se ubica después del from y se puede interpretar como un filtro.
- Ejemplo: Encontrar todos los nombres de instructores que pertenezcan al departamento de “Dpto Acad Ingeniería”

```
SELECT nombres_docente  
from model.[faci-inc_tb_docentes]  
WHERE departamento_docente = 'Dpto Acad Ingeniería'
```





ESTRUCTURA BÁSICA SQL



La cláusula “WHERE”

```
SELECT nombres_docente  
from model.[faci-inc_tb_docentes]  
WHERE departamento_docente = 'Dpto Acad Ingeniería'
```

	ABC nombres_docente ↑	ABC departamento_do
477	Aaron	Dpto Acad Ingeniería
1084	Aaron	Dpto Acad Ingeniería
571	Abigail	Dpto Acad Ingeniería
809	Abigail	Dpto Acad Ingeniería
877	Abigail	Dpto Acad Ingeniería
888	Abigail	Dpto Acad Ingeniería
71	Adam	Dpto Acad Ingeniería
97	Adam	Dpto Acad Ingeniería
105	Adam	Dpto Acad Ingeniería
211	Adam	Dpto Acad Ingeniería
332	Adam	Dpto Acad Ingeniería
422	Adam	Dpto Acad Ingeniería
1005	Adam	Dpto Acad Ingeniería





La cláusula “FROM”

- La cláusula **FROM** enumera las tablas involucradas en la consola (Pueden ser una o muchas)
- Ejemplo Encontrar los nombres de todos los cursos ofrecidos en el ciclo '2024-2' y que tengan un $(\text{creditaje} * 2) > 8$

```
SELECT nombre_curso  
FROM model.[faci-inc_tb_cursos]  
WHERE ciclo = '2024-2' and  $(\text{creditaje} * 2) > 8$ 
```





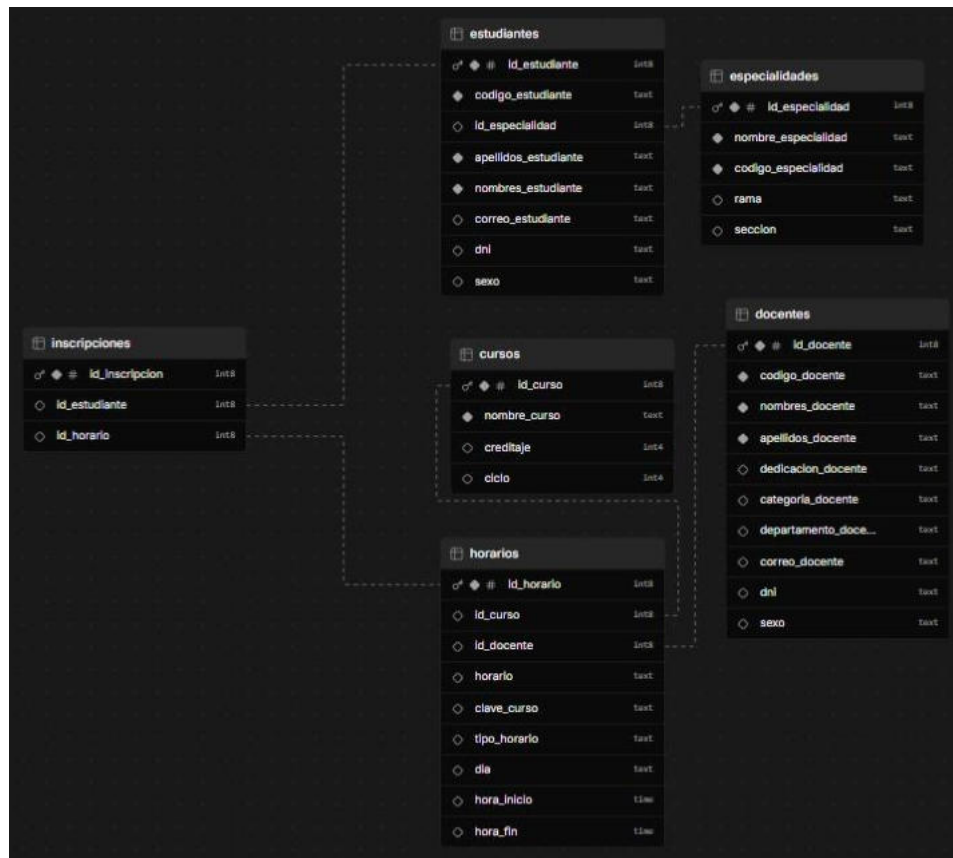
ESTRUCTURA BÁSICA SQL



EJERCICIOS

- ☐ Consulta los nombres de los docentes que tienen dedicación de "DTC".
- ☐ Consulta los nombres de los docentes que sean mujeres.
- ☐ Consulta los nombres y el ciclo de los cursos cuyo creditaje sea 4 .
- ☐ Consulta de todos los datos de los cursos que se llaman 'CREATIVIDAD E INNOVACIÓN'.
- ☐ Consulta el nombre de la sección cuyo código de especialidad sea "IAS".







BIBLIOGRAFÍA



Database system concepts / Abraham Silberschatz, Henry F. Korth and S. Sudarshan.
Seventh Edition. 2019. Cap 1



Gracias

