Introducción y Conceptos básicos de bases de datos

Alex Di Genova

April 5, 2022

Universidad de O'higgins

Outline

Bienvenida curso de Bases de datos

Bases de datos

Planificación curso BD

Bienvenida curso de Bases de datos

Presentación

Alex Di Genova

- 2003–2008 Ingeniero en Bioinformática.
- 2013-2017 Doctor en Sistemas Complejos.
- 2017-2021 Postdoctorado en algoritmos y cáncer (Francia).
- 2022 - Profesor Asistente UOH.
    - Di Genoma Lab
        - Combinamos el desarrollo de nuevos algoritmos, análisis de genomas y tecnologías ómicas de última generación para estudiar sistemas biológicos complejos.

Presentación

Database, Vol. 2011, Article ID bar050, doi:10.1093/database/bar050

Original article

SalmonDB: a bioinformatics resource for *Salmo salar* and *Oncorhynchus mykiss*

Alex Di Génova[1], Andrés Aravena[1,2,*], Luis Zapata[1,3], Mauricio González[1,4], Alejandro Maass[1,2] and Patricia Iturra[3]

[1]Laboratory of Bioinformatics and Mathematics of the Genome, Center for Mathematical Modeling (UMI 2807 CNRS) and Center for Genome Regulation (Fondap 15090007), University of Chile, Santiago, Chile, [2]Department of Mathematical Engineering, Faculty of Physical and Mathematical Sciences, University of Chile, Santiago, Chile, [3]ICBM Human Genetics Program, Faculty of Medicine, University of Chile, Santiago, Chile and [4]Laboratorio de Bioinformática y Expresion Génica, INTA, University of Chile, Santiago, Chile

***Corresponding author:** Tel: +56(2) 978 48 70; Fax: +56(2) 688 97 05; Email: andres.aravena@dim.uchile.cl

Submitted 1 July 2011; Revised 21 September 2011; Accepted 16 October 2011

SalmonDB is a new multiorganism database containing EST sequences from *Salmo salar*, *Oncorhynchus mykiss* and the whole genome sequence of *Danio rerio, Gasterosteus aculeatus, Tetraodon nigroviridis, Oryzias latipes* and *Takifugu rubripes*, built with core components from GMOD project, GOPArc system and the BioMart project. The information provided by this resource includes Gene Ontology terms, metabolic pathways, SNP prediction, CDS prediction, orthologs prediction, several precalculated BLAST searches and domains. It also provides a BLAST server for matching user-provided sequences to any of the databases and an advanced query tool (BioMart) that allows easy browsing of EST databases with user-defined criteria. These tools make SalmonDB database a valuable resource for researchers searching for transcripts and genomic information regarding *S. salar* and other salmonid species. The database is expected to grow in the near feature, particularly with the *S. salar* genome sequencing project.

Database URL: http://genomicasalmones.dim.uchile.cl/

Presentación

Alumnas y Alumnos

Bases de datos

Qué es una base de datos?

- Recopilación organizada de datos interrelacionados que modelan algún aspecto del mundo real.
- Las bases de datos son el componente central de la mayoría de las aplicaciones computacionales (Facebook, Instagram, . . .).

Cuando usamos una base de datos?

Usamos una base de datos, sí:

- Realizamos la inscripción de un curso en la universidad.
- Realizamos una transferencia bancaria.
- Realizamos compras online.
- Utilizamos las redes sociales.
- Utilizamos Spotify o Apple music.
- Otros . . .

Ejemplo de bases de datos

Crear una base de datos que modele el proceso de inscripción de cursos en una universidad para realizar un seguimiento de los estudiantes y los cursos.

Ejemplo de bases de datos

Crear una base de datos que modele el proceso de inscripción de cursos en una universidad para realizar un seguimiento de los estudiantes y los cursos.

Cosas que debemos almacenar:

- Información de los estudiantes (Nombre, Apellido, edad, . . .)
- Que cursos inscribieron los estudiantes (Nombre, semestre, calificación, . . .)

Solución basada en archivos

Almacenaremos nuestros datos de estudiantes y cursos utilizando archivos separados por tabulador, que serán manipulados con código propio.

Solución basada en archivos

Almacenaremos nuestros datos de estudiantes y cursos utilizando archivos separados por tabulador, que serán manipulados con código propio.

Cosas que debemos hacer:

- Para cada entidad crear un archivo (Estudiantes, Cursos, . . .)
- Nuestro código debe soportar operaciones con los archivos y registros (leer, actualizar, buscar, . . .)

Solución basada en archivos

Almacenaremos nuestros datos de estudiantes y cursos utilizando archivos separados por tabulador, que serán manipulados con código propio.

Cosas que debemos hacer:

- Para cada entidad crear un archivo (Estudiantes, Cursos, . . .)
- Nuestro código debe soportar operaciones con los archivos y registros (leer, actualizar, buscar, . . .)

Creamos la base de datos usando los siguientes archivos:

Alumno

Nombre	Edad	Mail
Alicia	21	alicia@gmail.com
Bob	22	bob@gmail.com
Carlos	19	carlos@gmail.com
		. . .

Curso

Nombre	Semestre	Alumno
BD	1	alicia
Programación	4	bob
Química	1	carlos
BD	1	bob
Programación	4	carlos
		. . .

Solución basada en archivos

Ejemplo:

- Obtener los cursos realizados por alicia.

```
file = open("Curso.txt")
for line in file:
 record = parse(line)
  if "alicia" == record[3]
   print record

# DB 1 alicia
```

Problemas de la solución basada en archivos

- Integridad de los datos
  - Como nos aseguramos que el/la alumn@ es él mismo en el archivo de Cursos?
  - Qué pasa si alguien almacena el campo semestre con una palabra (1− > primer)?
  - Como almacenamos múltiples alumnos en un curso?

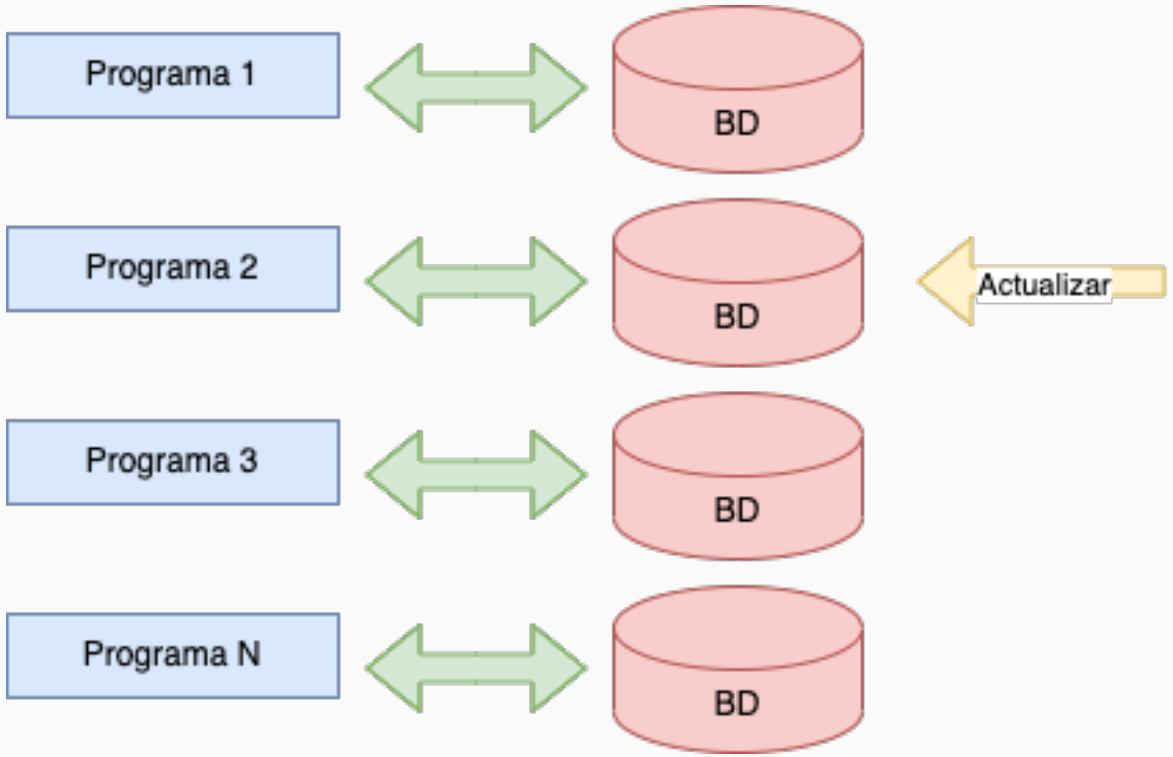
Problemas de la solución basada en archivos

- Integridad de los datos
  - Como nos aseguramos que el/la alumn@ es él mismo en el archivo de Cursos?
  - Qué pasa si alguien almacena el campo semestre con una palabra ($1- >$ primer)?
  - Como almacenamos múltiples alumnos en un curso?
- Implementación
  - Cómo podemos actualizar un o un grupo de registros?
  - Qué pasa si ahora queremos crear una nueva aplicación que usa la misma base de datos?
  - Qué pasa si dos aplicaciones actualizan al mismo tiempo al archivo Curso.txt?

Problemas de la solución basada en archivos

- Integridad de los datos
  - Como nos aseguramos que el/la alumn@ es él mismo en el archivo de Cursos?
  - Qué pasa si alguien almacena el campo semestre con una palabra ($1- > $ primer)?
  - Como almacenamos múltiples alumnos en un curso?
- Implementación
  - Cómo podemos actualizar un o un grupo de registros?
  - Qué pasa si ahora queremos crear una nueva aplicación que usa la misma base de datos?
  - Qué pasa si dos aplicaciones actualizan al mismo tiempo al archivo Curso.txt?
- Fiabilidad
  - ¿Qué pasa si la máquina falla mientras estamos actualizando un registro en alumno o Curso?
  - ¿Qué pasa si queremos replicar la base de datos en varias máquinas para tener alta disponibilidad?

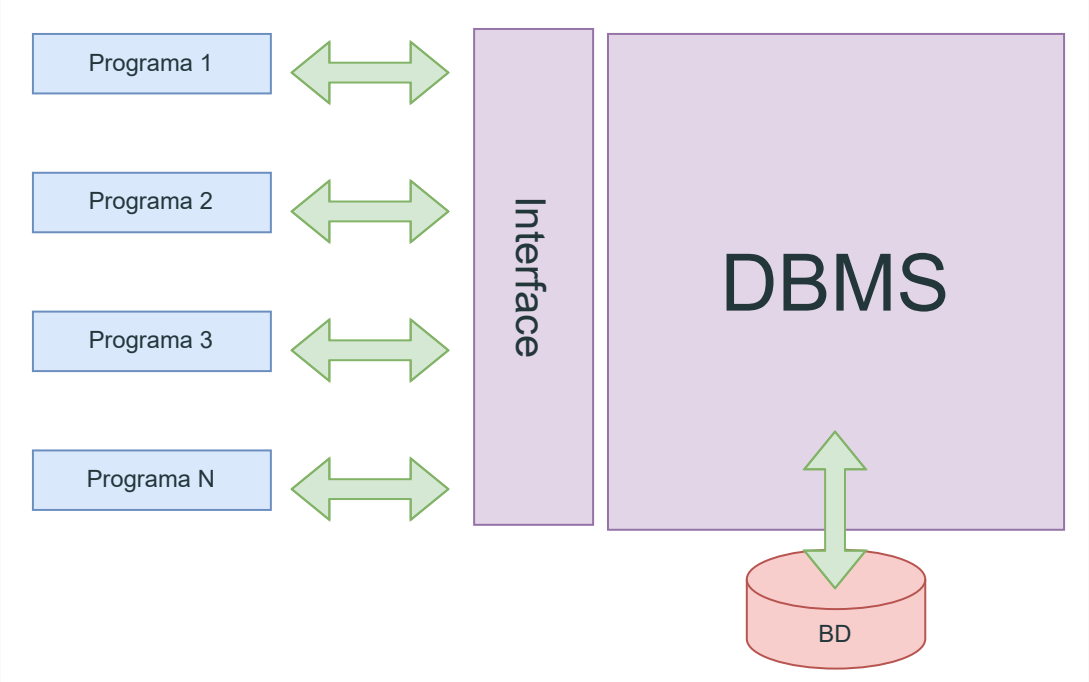
Sistema de administración de bases de datos (DBMS)



- Solución basada en archivos
  - Redundancia de código.
  - Redundancia de datos.
  - Inconsistencia de datos.

Sistema de administración de bases de datos (DBMS)

- Un DBMS es un software que permite a las aplicaciones almacenar y analizar información en una base de datos.
- Un DBMS de propósito general está diseñado para permitir la definición, creación, consultas, actualización y administración de bases de datos.
    - MySQL, PostgreSQL, Microsoft SQLServer, Oracle, **SQLite**
        - Structured Query Language (SQL)



Planificación curso BD

Unidades

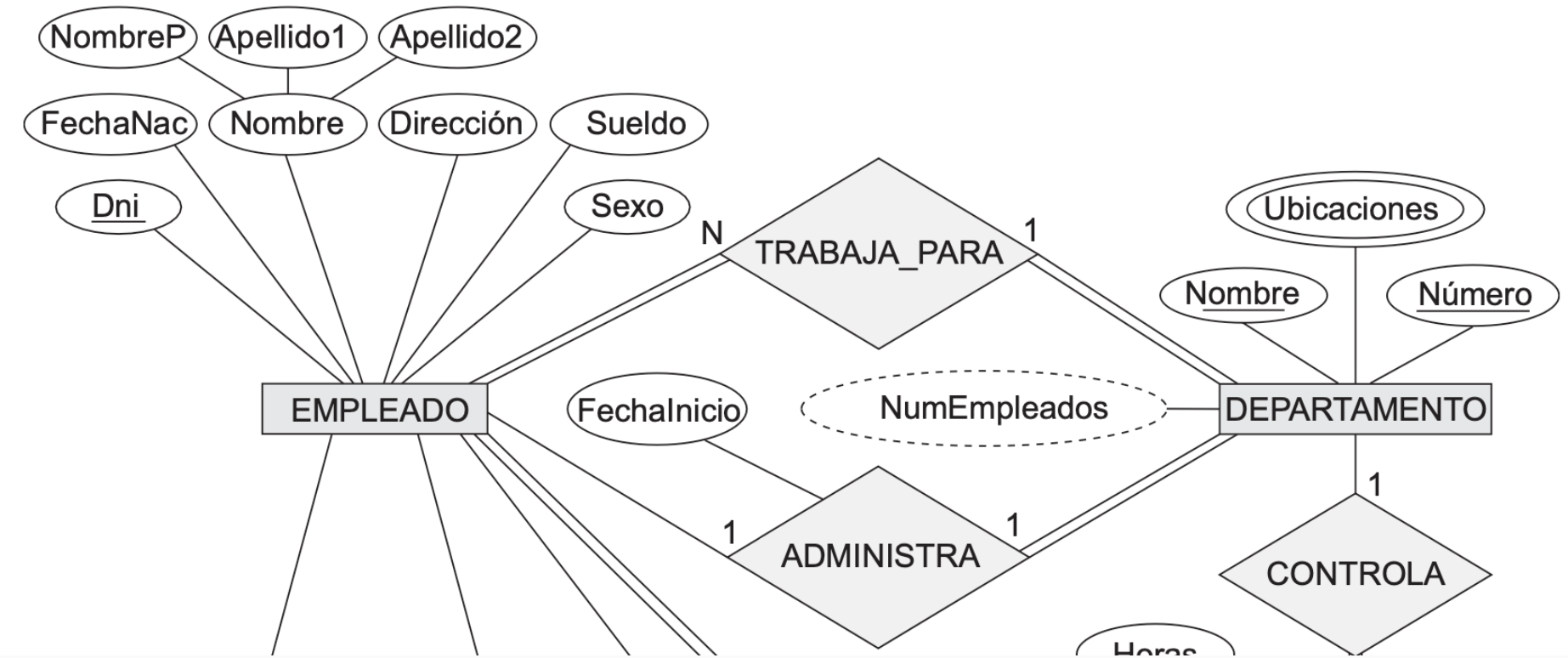
- 4 Unidades (14 semanas)
  - Modelamiento de bases de datos (3 semanas)
  - Modelo Relacional (4 semanas)
  - Lenguaje de Consulta SQL (3 semanas)
  - Transacciones y Bases de datos no relacionales (4 semanas)

Unidades

Modelamiento de bases de datos

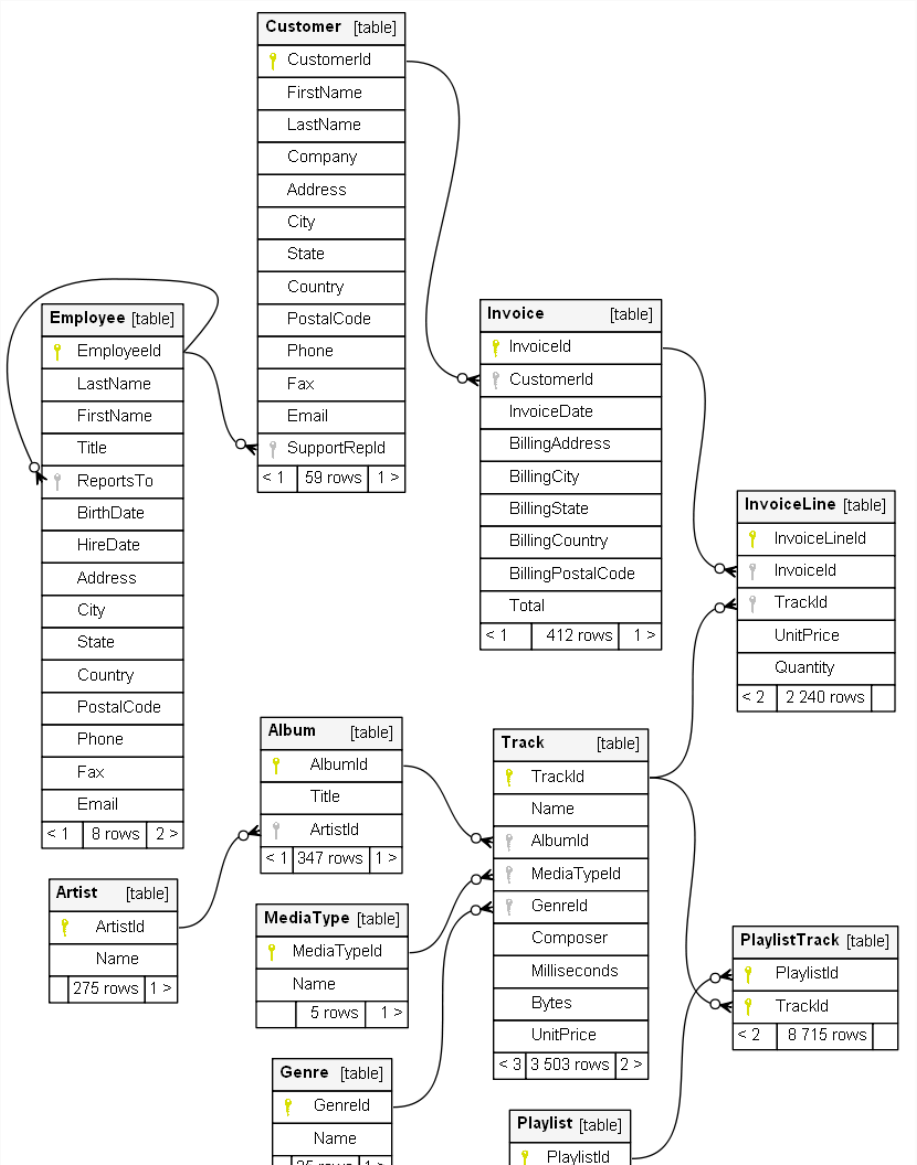
- *Cada* **persona** *puede habitar en solo una* **vivienda** *y estar registrada en solo un* **municipio** *pero pude ser propietaria de varias viviendas.* ...
- La **empresa** está organizada en **departamentos**. Cada uno tiene un nombre único, un número único y un empleado concreto que lo administra. Un departamento controla una cierta cantidad de **proyectos**, cada uno de los cuales tiene un nombre único, un número único y una sola ubicación.

Esquema Relacional



Unidades

Diagrama Entidad-Relacion (Modelo UML)



Unidades

SQL (SQLite3)

```
[10]  1 %%sql
      2 PRAGMA table_info([Artist]);
```

```
 * sqlite:////content/chinook-database/ChinookDatabase/DataSources/Chinook_Sqlite.sqlite
Done.
```

cid	name	type	notnull	dflt_value	pk
0	ArtistId	INTEGER	1	None	1
1	Name	NVARCHAR(120)	0	None	0

```
1 %%sql
2 select * from Artist limit 10;
```

```
 * sqlite:////content/chinook-database/ChinookDatabase/DataSources/Chinook_Sqlite.sqlite
Done.
```

ArtistId	Name
1	AC/DC
2	Accept
3	Aerosmith
4	Alanis Morissette
5	Alice In Chains
6	Antônio Carlos Jobim
7	Apocalyptica
8	Audioslave
9	BackBeat
10	Billy Cobham

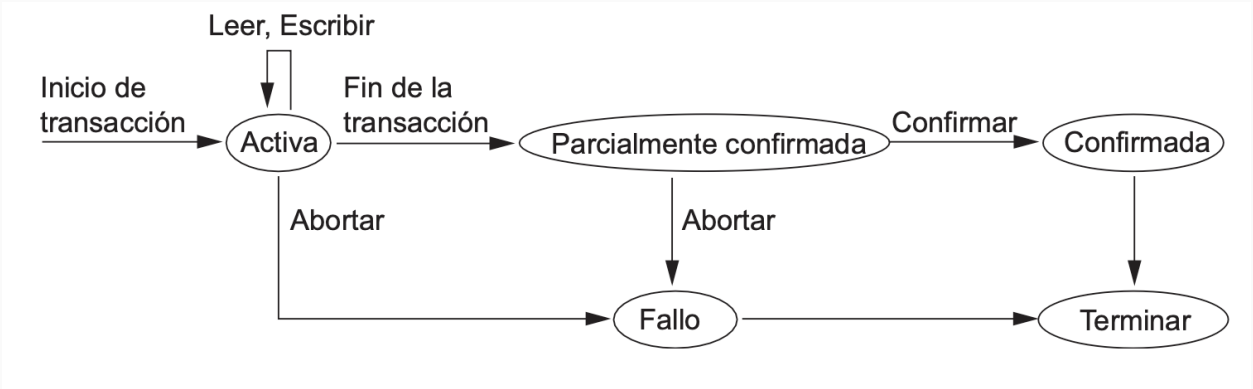
GoogleColab – `https://colab.research.google.com/`

Video descriptivo de GoogleColab

Unidades

Transacciones

- Atomicidad
- Conservación de la consistencia.
- Aislamiento
- Durabilidad



No-SQL

- levelDB, rocksDB . . .

Evaluaciones

- Controles (85%):
  - Control 1: Semana del 9 Mayo.
  - Control 2: Semana del 13 Junio.
  - Control 3: Semana del 11 Julio.

Evaluaciones

- Controles (85%):
  - Control 1: Semana del 9 Mayo.
  - Control 2: Semana del 13 Junio.
  - Control 3: Semana del 11 Julio.
- Tareas (15%):
  - Tarea 1: Semana del 9 Mayo.
  - Tarea 2: Semana del 6 Junio.

Condiciones y Políticas de Evaluación

- El promedio de actividades complementarias se considerará como un cuarto control (control IV) y tendrá una ponderación de 15%. El promedio de controles I,II,III y IV con sus respectivas ponderaciones corresponderán a la nota final del curso. El curso será aprobado con una nota promedio igual o superior a 4,0.

Condiciones y Políticas de Evaluación

- El promedio de actividades complementarias se considerará como un cuarto control (control IV) y tendrá una ponderación de 15%. El promedio de controles I,II,III y IV con sus respectivas ponderaciones corresponderán a la nota final del curso. El curso será aprobado con una nota promedio igual o superior a 4,0.
- Estudiantes que se ausenten a un control tendrán la oportunidad de recuperarlo durante el periodo correspondiente al final del semestre. El control recuperativo es de carácter **acumulativo**, por lo tanto, contendrá contenido de las cuatro unidades del curso.
  Adicionalmente, alumnos que quieran remplazar una calificación en un control o actividades complementarias, también podrán rendir el control recuperativo.

Condiciones y Políticas de Evaluación

- El promedio de actividades complementarias se considerará como un cuarto control (control IV) y tendrá una ponderación de 15%. El promedio de controles I,II,III y IV con sus respectivas ponderaciones corresponderán a la nota final del curso. El curso será aprobado con una nota promedio igual o superior a 4,0.
- Estudiantes que se ausenten a un control tendrán la oportunidad de recuperarlo durante el periodo correspondiente al final del semestre. El control recuperativo es de carácter **acumulativo**, por lo tanto, contendrá contenido de las cuatro unidades del curso.
  Adicionalmente, alumnos que quieran remplazar una calificación en un control o actividades complementarias, también podrán rendir el control recuperativo.
- Un/a estudiante que cometa plagio sobtendrá un 1,0 en la evaluación y el caso será informado a Escuela de Ingeniería.

Materiales

- Repositorio GitHub – `https://github.com/adigenova/uohdb`
  - Clases
  - Notebooks para ejecutar en GoogleColab

Materiales

- Repositorio GitHub – `https://github.com/adigenova/uohdb`
  - Clases
  - Notebooks para ejecutar en GoogleColab
- Ucampus – `https://ucampus.uoh.cl/uoh/2022/1/COM3101`
  - Comunicación (Consultas, noticias, evaluaciones)
  - Planificacion

Materiales

- Repositorio GitHub – `https://github.com/adigenova/uohdb`
  - Clases
  - Notebooks para ejecutar en GoogleColab
- Ucampus – `https://ucampus.uoh.cl/uoh/2022/1/COM3101`
  - Comunicación (Consultas, noticias, evaluaciones)
  - Planificacion
- BIBLIOGRAFÍA Y RECURSOS COMPLEMENTARIOS
  - Ramez A. Elmasri, Shamkant B. Navathe, Fundamentos de Sistemas de Bases de Datos, 5a Edic., Addison Wesley. 2007 (Capitulo 1)
  - Molinaro, Anthony. SQL Cookbook. O'Reilly Media. (2009).

Resultados de Aprendizaje

- Diseñar diagramas de Entidad/Relacional para satisfacer las necesidades de un problema enunciado.

Resultados de Aprendizaje

- Diseñar diagramas de Entidad/Relacional para satisfacer las necesidades de un problema enunciado.
- Realizar a partir de un diagrama Entidad/Relación un diseño relacional.

Resultados de Aprendizaje

- Diseñar diagramas de Entidad/Relacional para satisfacer las necesidades de un problema enunciado.
- Realizar a partir de un diagrama Entidad/Relación un diseño relacional.
- Normalizar un diseño relacional de bases de datos.

Resultados de Aprendizaje

- Diseñar diagramas de Entidad/Relacional para satisfacer las necesidades de un problema enunciado.
- Realizar a partir de un diagrama Entidad/Relación un diseño relacional.
- Normalizar un diseño relacional de bases de datos.
- Formular consultas de distinto tipo en SQL.

Resultados de Aprendizaje

- Diseñar diagramas de Entidad/Relacional para satisfacer las necesidades de un problema enunciado.
- Realizar a partir de un diagrama Entidad/Relación un diseño relacional.
- Normalizar un diseño relacional de bases de datos.
- Formular consultas de distinto tipo en SQL.
- Reconocer la noción de transacción y operar el sistema de recuperación de un sistema de administración de bases de datos.

Resultados de Aprendizaje

- Diseñar diagramas de Entidad/Relacional para satisfacer las necesidades de un problema enunciado.
- Realizar a partir de un diagrama Entidad/Relación un diseño relacional.
- Normalizar un diseño relacional de bases de datos.
- Formular consultas de distinto tipo en SQL.
- Reconocer la noción de transacción y operar el sistema de recuperación de un sistema de administración de bases de datos.
- Conocer sistemas de bases de datos no relacionales.

Consultas?

Consultas o comentarios?

Muchas Gracias.