



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2413 - BASES DE DATOS

Examen

IIC2413 - Bases de Datos

Fecha: 4 de junio de 2024

1º semestre 2024 - Profesores: N. Alvarado - A. Reyes - E. Bustos - C.Álvarez

INSTRUCCIONES

- Antes de comenzar **LEA TODO EL ENUNCIADO.**
- Al finalizar el examen tiene 10 minutos para digitalizarla y subirla a Canvas correctamente. No seguir esta instrucción los expone a un descuento de hasta 5 décimas de la nota.
- Puede utilizar un resumen de la materia de 1 hoja tamaño oficio escrita o impresa por ambas caras.
- Cada hoja debe contener su Nombre Completo y Número de Lista en la esquina superior derecha.
- Entregue todas las hojas de respuesta juntas.
- Se puede llevar el enunciado.
- Debe firmar la lista de asistencia.
- Duración 3 horas.

Pregunta 1: Modelo E/R, Esquema, Formas Normales, SQL

Se le ha encargado modelar el sistema de registro de notas de la carrera de derecho de una universidad. Para ello se le entregan un archivo CSV (valores separados por coma) que contiene, entre otros, los datos de todos los estudiantes y las asignaturas con sus resultados desde el principio de los tiempos hasta la mitad del 1 semestre del 2024 (mayo de 2024-1). Cada línea del archivo contiene la individualización del estudiante, una asignatura con su correspondiente nota de cada vez que lo tomó (por ejemplo, si lo aprobó a la 3a oportunidad aparecen 3 veces en el archivo con cada nota obtenida). Las asignaturas que están en curso en el periodo vigente aparecen sin nota ni calificación.

En el caso de los Estudiantes aparece el año de ingreso, los planes de estudio por los cuales ha pasado y sus datos personales.

Reglas de negocio:

- La carrera dura un total de 10 semestres, luego de lo cual los estudiantes deben realizar el examen de grado de Licenciatura con lo que obtienen el grado de Licenciado.
- Todos los estudiantes ingresan a 1er semestre con último logro = Ingreso y a medida que van completando los semestres (niveles) el logro se incrementa en 1 (es decir 1-10), luego del semestre 10 dan el examen de grado y si aprueban el logro pasa ha estado Licenciado. La secuencia completa sería: Ingreso-1-2-...-10-LIC
- Todas las asignaturas tienen un nombre, código, el periodo en que fue inscrita, el nivel (lvl) que indica el semestre al que corresponde (1 a 10), la nota y calificación obtenida. Si el estudiante reprueba la asignatura, esta aparecerá una vez por cada período que la haya cursado con su respectiva nota y calificación. Si la asignatura tiene nota y calificación vacía, significa que la está cursando en el presente período.
- El número de estudiante es único, si una persona se retira o cae en causal de eliminación y vuelve a ingresar lo hará desde el primer semestre (logro = Ingreso) con un nuevo número de estudiante, en este caso, como ya ingresó una vez tiene 2 números de alumnos diferentes en el archivo y puede tener dos planes de estudio (código de plan y plan).
- Un estudiante puede tener más de un plan de estudios, sin que eso signifique que tiene dos números de estudiante. Por ejemplo, un estudiante que cambia de Derecho diurno Santiago a Derecho vespertino Concepción aparecerá con Código de plan = DE01 y DV02.
- La última carga es el período en que el estudiante inscribió ramos por última vez.

Los datos del archivo son:

Ingreso: año y semestre de ingreso a la carrera <año>-<semestre>(Ej. 2020-1)

Código Plan: código plan de estudios formato: 4 caracteres (Ej. DV02)

Plan: nombre del plan 30 caracteres (Ej. Derecho Vespertino Santiago)

RUN: mantisa del RUN entero máximo 8 cifras (Ej. 12.345.678)

DV: dígito verificador del RUN 1 caracter (Ej. 1)

Nombres: Todos los nombres 50 caracteres (Ej. Juan José)

Apellido Paterno: apellido paterno 50 caracteres (Ej. Soto)

Apellido Materno: apellido materno 50 caracteres (Ej. Contreras)

Número de estudiante: número único que identifica al estudiante (Ej. 9034567)

Periodo Asignatura: período que cursó o está cursando la asignatura
<año>-<semestre>(Ej. 2020-1)

Código Asignatura: código numérico asignado a la asignatura 3 letras y hasta 4 números (Ej. IIC2413)

Asignatura: nombre de la asignatura (Ej. bases de datos)

lvl: nivel de la asignatura, semestre al que pertenece la asignatura (Ej. 5)

Calificación: nota en concepto de acuerdo a {
Aprobados: SO, MB, B, SU, EX
Reprobados: I, M, MM
Otros: NP, P, NA, vacío

Nota: nota obtenida en la asignatura de 1,0 a 7,0 o vacío

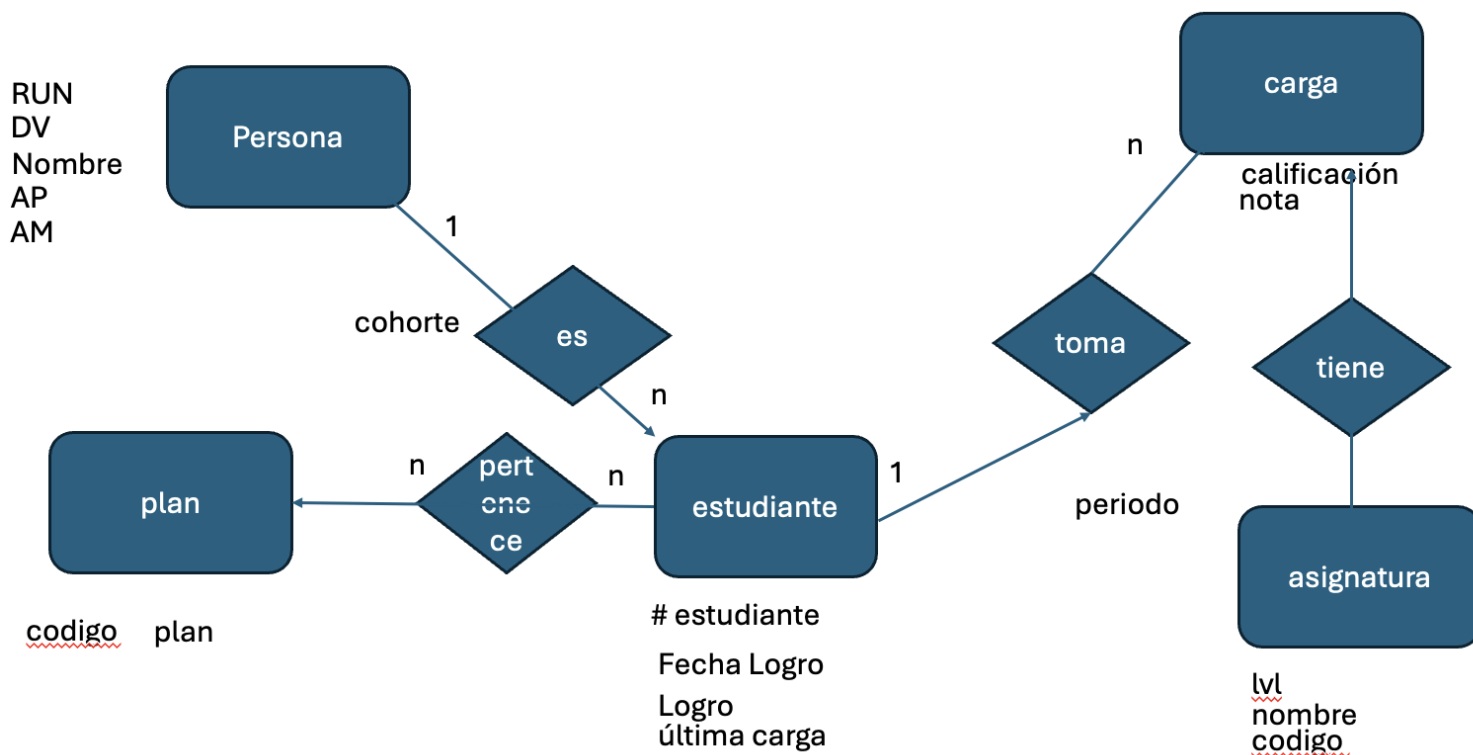
Logro: último semestre de la malla aprobado {
Ingreso
Ej. 3
Licenciado

Fecha Logro: fecha del último logro <año>-<semestre>(Ej. 2020-1)

Última Carga: último periodo de toma de ramos <año>-<semestre>(Ej. 2020-1)

Preguntas

I (2pt) Cree un modelo E/R para el problema indicando entidades, relaciones, cardinalidad y llaves.



II (2pt) Indique las dependencias funcionales y normalice a BCNF o 3NF.

Solución

Código asignatura \rightarrow asignatura, lvl

Nota \rightarrow calificación, codigo plan, plan

RUN \rightarrow DV, Nombres, AP, AM

Estudiante \rightarrow RUN, Cohorte, ultimo logro, fecha, ultima carga

Estudiante, asignatura, periodo \rightarrow nota, plan

III (2pt) Escriba el esquema relacional del modelo indicando los dominios, llaves primarias, foráneas y restricciones de integridad.

Solución

```

Asignatura (Codigo string (4)PK NOT NULL UNIQUE,
           asignatura string (30)Not NULL,
           lvl int(1-10)Not NULL
Notas (Nota float(1,0-7,0) PK NOT NULL, UNIQUE permite null
Calificación string(2) (SO, MB, B, SU, EX, I, M, MM, NP, P, NA)
      UNIQUE      permite null
Plan (codigo string(4) PK NOT NULL, plan string(4) NOT NULL
Persona(RUN int NOT NULL PK,DV string(1)NOT NULL, Nombres string(50)NOT NULL,
        AP string(50) NOT NULL, AM string(50) permite null
Estudiante(#Estudiante int NOT NULL UNIQUE PK, RUN int FK(persona.RUN),
           Cohorte string(6) NOT NULL, ultimo logro string(7)NOT NULL,
           fecha Ultimo logro string(6) NOT NULL, ultima carga string(6) NOT NULL
Carga(#Estudiante PK, cod asignatura string(4)PK, periodo string(7)PK,
      nota string(3) FK(Notas.nota, plan string(4) FK (Plan.codigo))

```

IV (2pt) Indique las sentencias de creación de las tablas en SQL-DDL.

Solución

```

CREATE TABLE Asignatura
  Codigo string (4)PK NOT NULL UNIQUE,
  asignatura string (30)Not NULL,
  lvl int(1-10)Not NULL
)
CREATE TABLE Notas (
  Nota float(1,0-7,0) PK NOT NULL, UNIQUE
  Calificación string(2) UNIQUE
)
CREATE TABLE Plan (
  codigo string(4) PK NOT NULL,
  plan string(4) NOT NULL
)
CREATE TABLE Persona(
  RUN int NOT NULL PK,
  DV string(1)NOT NULL (0-9,K),
  Nombres string(50)NOT NULL,
  AP string(50) NOT NULL,
  AM string(50)
)
CREATE TABLE Estudiante(
  #Estudiante int NOT NULL UNIQUE PK,
  RUN int FK(persona.RUN),
  Cohorte string(6) NOT NULL,

```

```

        ultimo logro string(7)NOT NULL,
        fecha Ultimo logro string(6) NOT NULL,
        ultima carga string(6) NOT NULL
    )
CREATE TABLE Carga(
    #Estudiante int PK,
    cod asignatura string(4)PK,
    periodo string(7)PK,
    nota string(3) FK(Notas.nota,
    plan string(4) FK (Plan.codigo)
    PK ( #Estudiante cod asignatura, periodo string)
    FK #Estudiante (Estudiante.#Estudiante)
    FK cod asignatura (asignatura.codigo)
)

```

- V **(2pt)** Escriba el código SQL que entregue el número de estudiantes que están dentro de nivel y fuera de nivel para cada uno de los niveles. Se define como dentro de nivel si $\text{logro} = (\text{periodo actual} - \text{cohort})$, periodo actual es el mayor periodo que se encuentra en el archivo.

Solucion

```

Select distinct count(#Estudiantes) from estudiante
where estudiante.ultimo logro= Max(estudiante.ultima carga)-estudiante.cohorte)
Select distinct count(#Estudiantes) from estudiante

```

- VI **(BONUS 2pt)** Ahora suponga que existe la tabla Prerequisito que contiene el código de la asignatura y el código de su único prerequisito, escriba una sentencia SQL que entregue para cada estudiante de 1er a 9° semestre ($\text{logro} = \text{Ingreso}$, 1..9) una propuesta de asignaturas a inscribir en el próximo período (la próxima toma de ramos). Considere que las asignaturas tienen uno y solo un prerequisito; en la tabla aparecen solo las asignaturas con prerequisitos, otras asignaturas. No considere asignaturas sin prerequisitos.

Tabla Prerequisito

Código Prerequisito: Código de la asignatura que es prerequisito

Código Asignatura: Código de la asignatura

Solucion

```

Select Asignatura from prerequisito where prerequisito= carga.cod asignatura

```

Fracción de archivo CSV

Cohorte	Código Plan	Plan	RUN	DV	Nombres	Apellido Paterno	Apellido Materno	Número Estudiante	Periodo Asignatura	Código Asignatura	Asignatura	Id	Calificación	Nota	último Logro	Fecha Logro	Última Carga
2015-1	DV02	DERECHO VESPERTINO SANTIAGO	19522696	5	NICOLE VALERIA	HENRÍQUEZ	ARENAS	2994324	2023-02	DER7559	EL DERECHO A LA PROTECCION DE LA SALUD	3	SO	7,0	6	2023-2	2024-1
2015-1	DV02	DERECHO VESPERTINO SANTIAGO	21528696	5	GONZALO ESTEBAN	VALLEJOS	VERGARA	2994444	2023-02	DER3486	DERECHO CIVIL VI	7	SU	4,3	6	2023-2	2024-1
2017-1	DV02	DERECHO VESPERTINO SANTIAGO	19339557	6	CRISTIAN FRANCISCO	SALAS	PASTENE	3173894	2023-02	DER3486	DERECHO CIVIL VI	7	Ex		4	2023-2	2024-1
2018-1	DE02	DERECHO PLAN 2019	20077285	6	CRISTIAN DAVID	CORTEZ	APABLAZA	4192398	2023-02	DER3495	SEMINARIO INTEGRATIVO II	9	MB	6,1	7	2023-1	2024-1
2018-1	DE02	DERECHO PLAN 2019	19438711	5	BELISARIO DAVID	GODOY	CAMUS	4080296	2023-02	DER4444	FUNDAMENTOS FILOSOFICOS DEL DERECHO	10	MM	1,0	LIC	2023-2	2024-1
2018-1	DE02	DERECHO PLAN 2019	19438711	5	BELISARIO DAVID	GODOY	CAMUS	4080296	2023-02	DER4444	FUNDAMENTOS FILOSOFICOS DEL DERECHO	10	SU	4,0	LIC	2023-2	2024-1
2019-1	DE02	DERECHO PLAN 2019	20191079	0	CAMILA PAZ	VILLALOBOS	POCH	4345345	2023-02	DER3491	DERECHO DE SOCIEDADES	8	I	3,3	6	2023-1	2024-1
2019-1	DE02	DERECHO PLAN 2019	20191079	0	CAMILA PAZ	VILLALOBOS	POCH	4345292	2023-02	DER3491	DERECHO DE SOCIEDADES	8	SU	4,0	6	2023-1	2024-1
2019-1	DV02	DERECHO VESPERTINO PLAN 2005	20400487	8	LISBETH MELISSA	NAVIA	FLORES	3081498	2023-02	DER3489	DERECHO TRIBUTARIO	8	B	5,3	9	2023-2	2024-1
2019-1	DS03	DERECHO CONCEPCIÓN DIURNO	20761102	7	CAMILA	GODOY	CORTÉS	1521298	2023-02	DER3470	DERECHO PENAL I	4	M	2,8	3	2021-1	2024-1
2024-1	DS03	DERECHO CONCEPCIÓN DIURNO	20761112	7	MELISSA	VERGARA	CORTÉS	1521298	2023-02	DER1111	INTRODUCCIÓN AL DERECHO	1			INGRESO	2024-1	2024-1

Pregunta 2: Álgebra Relacional

El juego EntreNos, consiste en un juego cooperativo donde astronautas deben realizar acciones cooperativas, pero donde uno de los jugadores es un impostor. Los jugadores buscaran llenar una barra de progreso y el impostor buscará asesinar a los astronautas. El juego termina (y ganan los astronautas) si se completa la barra de progreso o si descubren al impostor, y termina (y gana el impostor) si matan a todos los astronautas. Suponga que usted dispone del siguiente modelo relacional para el juego EntreNos:

- Jugador(nick, nombre)
- Sala(código, max_jugadores, Jugador.nick_creador)
- Mapa(nombre, descripción)
- Juego(Sala.código, fecha, Mapa.nombre, gana_impostor)
- Partida(Jugador.nick, Juego.código, Juego.fecha, es_impostor, color, vivo)
- Mensaje(Partida.nick, Partida.código, Partida.fecha, contenido)
- Pertenece(Jugador.nick, Sala.código, fecha_ingreso)

Una sala tiene un creador y alberga varios jugadores. Dentro de cada sala pueden ocurrir muchos juegos. Cada juego ocurre en un mapa. Un jugador puede jugar en muchos juegos, y para cada jugador que juega un juego se sabe si fue impostor o no, que color uso, y si terminó vivo o no. Además cada jugador puede escribir mensajes en los distintos juegos.

La empresa desarrolladora del juego desea evaluar su diseño y tener un recuento más específico de los dato que maneja. Considerando este modelo relacional, escriba usando álgebra relacional las siguientes consultas. Recuerde usar solo los operadores que fueron vistos en clases.

- I (2pt) Los jugadores (nick y nombre) que nunca han creado una sala.
- II (2pt) Los nick de los jugadores que han jugado con todos los colores.
- III (2pt) Todos los jugadores (nick y nombre) que nunca han perdido una partida al jugar como impostor.

Solución. I

$$\pi_{\text{nick, nombre}}(\text{Jugador}) - \pi_{\text{nick, nombre}}(\sigma_{\text{Jugador.nick=Sala.Jugador.nick_creador}}(\text{Jugador} \bowtie \text{Sala})) \quad (1)$$

II

$$C_{\text{colores}} = \pi_{\text{color}}(\text{Partida}) \quad (2)$$

$$R(\text{nick, color}) := \pi_{\text{nick, color}}(\text{Partida}) \quad (3)$$

$$\text{Resultado} := R \div C_{\text{colores}} \quad (4)$$

III

Perdedores_impостor =

$$\pi_{\text{Jugador.nickname, nombre}}(\sigma_{\text{es_impostor}=1 \wedge \text{vivo}=0 \wedge \text{gana_impostor}=0}(\text{Jugador} \bowtie \text{Partida} \bowtie \text{Juego}))$$

$$\pi_{\text{nickname, nombre}}(\text{Jugador}) - \text{Perdedores_impostor}$$

□

Pregunta 3: Lógica en la BD

Usted, ha sido contratado por la Asociación de Gimnasia Rítmica para diseñar un sistema que les permita almacenar la información de las distintas competencias que se desarrollan durante un año. El modelo de la BD del sistema es el siguiente:

- Competencias(id_competencia: int PRIMARY KEY,
nombre: varchar(50),
fecha: date,
lugar: varchar(500))
- Equipos(id_equipo; int PRIMARY KEY,
nobre: varchar(50),
pais: varchar(50))
- Gimnastas(id_gimnasta: int PRIMARY KEY,
nombre; varchar(50),
apellido: varchar(50),
id_equipo: int,
FOREIGN KEY(id_equipo) REFERENCES Equipos(id_equipo))
- Participaciones(id_participacion int PRIMARY KEY,
id_gimnasta: int,
id_competencia: int,
FOREIGN KEY(id_gimnasta) REFERENCES gimnastas(id_gimnasta),
FOREIGN KEY (id_competencia) REFERENCES competencias(id_competencia))
- Puntuaciones(id_puntuacion: int PRIMARY KEY,
id_participacion: int,
modalidad: varchar(50),
puntaje: float,
FOREIGN KEY(id_participacion) REFERENCES participaciones(id_participacion))

Observación: los valores posibles para el campo modalidad de la tabla puntuaciones son: cinta, aro, cuerdas, balón y clavav.

Con respecto al esquema anterior, responda las siguientes preguntas:

- I (1pt) Defina la tabla de **puntuacion_equipo** que permita almacenar el puntaje de cada equipo en cada participación, escribiendo las sentencias SQL-DDL para su creación. Debe especificar los tipo de datos, claves primarias y foráneas.

Solución:

```
CREATE TABLE puntuacion_equipo (  
    id_puntuacion_equipo int PRIMARY KEY,  
    id_participacion: int,  
    modalidad: varchar(50),  
    id_equipo: int,  
    puntaje: float,  
    FOREIGN KEY(id_participacion)  
    REFERENCES participaciones(id_participacion),  
    FOREIGN KEY(id_equipo) REFERENCES equipos(id_equipo))
```

- II (1pt) Genere el código SQL para insertar 2 equipos en la tabla **puntuación_equipo** con puntaje 0.

Solución:

```
insert into puntuacion_equipo(id_puntuacion, id_participacion, id_equipo,  
    modalidad,puntaje)  
VALUES(1,2,1,'clavas',0);  
insert into puntuacion_equipo(id_puntuacion, id_participacion, id_equipo,  
    puntaje)  
VALUES(2,2,2,'balón',0);
```

- III (2pts) Genere el código SQL para crear un trigger que al momento de insertarse un puntaje de una gimnasta en la tabla *puntuaciones*, actualice el puntaje del equipo en la tabla **puntuacion_equipo**.

Solución:

```
CREATE TRIGGER update_equipo  
AFTER CREATE ON puntuaciones  
FOR EACH ROW  
BEGIN  
    UPDATE puntuacion_equipo  
    SET puntaje = puntuacion_equipo.puntaje + NEW.puntaje  
    WHERE puntuacion_equipo.id_participacion = NEW.id_participacion  
    AND puntuacion_equipo.modalidad = NEW.modalidad  
END
```

- IV (2pts) Cree una vista que muestre el puntaje total por equipo, para todas las competencias en que haya participado, en cuantas competencias ha participado, ordenado de mayor a menor puntaje.

Solución:

```
CREATE VIEW PUNTAJE_TOTAL_EQUIPO AS
SELECT pe.id_equipo, eq.nombre, count(par.id_competencia),sum(pe.puntaje)
FROM puntuacion_equipo as pe,
     equipos as eq,
     participaciones as par
WHERE pe.id_equipo = eq.id_equipo
AND pe.id_participacion = par.id_participacion
group by pe.id_equipo, eq.nombre
```

Pregunta 4: Índices, TF-IDF y Transacciones

Responda las siguientes preguntas.

I (2pts) Dado los siguientes textos que se incluyen a continuación, responda cada una de las consultas.

- Documento 1:
 Se ha formao' un casamiento
 Todo cubierto de negro
 Negros novios y padrinos
 Negros cuñados y suegros
 Y el cura que lo casó
 Era de los mismos negros
- Documento 2:
 Cuando empezaron la fiesta
 Pusieron un mantel negro
 Luego llegaron al postre
 Se sirvieron higos secos
 Y se fueron a acostar
 Debajo de un cielo negro
- Documento 3:
 Y allí están las dos cabezas
 De la negra con el negro
 Amanecieron con frío
 Tuvieron que prender fuego
 Carbón trajo la negrita
 Carbón que también es negro

a) (1/2 pt) Calcule TF de las palabras: **negro** y **carbón** en cada uno de los 3 documentos.

Solución:

- negro: Documento 1 --> 1
 Documento 2 --> 2
 Documento 3 --> 2
- carbón: Documento 1 --> 0
 Documento 2 --> 0
 Documento 3 --> 2

b) (1/2 pt) Calcule el IDF de las palabras **negro** y **carbón** en los 3 documentos.

Solución:

- negro: $\log(3/3) \rightarrow 0$
- carbón: $\log(3/1) \rightarrow 0.477$

c) (1/2 pt) Calcule TF-IDF de las palabras **negro** y **carbón** para cada uno de los 3 documentos

Solución:

- negro: Documento 1 $\rightarrow 1*0=0$
Documento 2 $\rightarrow 2*0=0$
Documento 3 $\rightarrow 2*0=0$
- carbón: Documento 1 $\rightarrow 0*0.477=0$
Documento 2 $\rightarrow 0*0.477=0$
Documento 3 $\rightarrow 2*0.477=0.954$

d) (1/2 pt) Si buscáramos la palabra **negro**, cuál es el documento que debiéramos responder como primera preferencia.

Solución:

De acuerdo al cálculo realizado anteriormente, todos los cálculos de TF-IDF es 0, lo que significa que no es una diferenciadora. Por lo tanto podríamos responder con cualquiera de los 3 documentos.

II (2pts) Responda las siguientes preguntas sobre Índices

a) (1/2 pt) Defina Índices Clustered y Unclustered.

- Clustered: es un índice que al preguntar por un valor, retorna la tupla asociada al índice.
- Unclustered: es un índice que al preguntar por un valor, retorna un puntero a la página donde se encuentra la tupla asociada al índice.

b) (1/2 pt) ¿Por qué se usan Clustered Index para las Primary Key?

Solución:

Los Clustered Index se usan para las llaves primarias porque se obtiene acceso inmediato a los datos y al ser el índice más usado de la tabla, es más eficiente.

- c) (1 pt) ¿Para qué tipos de consultas son más convenientes los índices Hash y para cuales los índices B+ Tree? Justifique su respuesta.

Solución:

Los índices hash son más usados para consultas de igualdad y los índices B+ Tree para consultas de rango.

III (2 pts) Responda las siguientes preguntas sobre Transacciones

- a) (1/2 pt) Defina Transacción (Transaction).

Solución:

Es una secuencia de 1 o más operaciones que modifican o consultan la base de datos.

- b) (1/4 pt) Defina Atomicidad (Atomicity).

Solución:

Se ejecutan todas las operaciones de la transacción o no se ejecuta ninguna

- c) (1/4 pt) Defina Consistencia (Consistency).

Solución:

La transacción preserva las restricciones de integridad de la BD.

- d) (1/4 pt) Defina Aislamiento (Isolation).

Solución:

Cada transacción se debe ejecutar como si se estuviera ejecutando sola, de forma aislada

- e) (1/4 pt) Defina Durabilidad (Durability).

Solución:

Los cambios que hace cada transacción son permanentes en el tiempo, independiente de cualquier tipo de falla.

f) (1/2 pt) ¿Por qué es necesario preservar las propiedades de ACID?

Solución:

Es necesario preservar ACID para conservar
la integridad y confiabilidad de la BD.