ALMACENAMIENTO DE DATOS

ÍNDICES Y DESNORMALIZACIÓN DE BBDD

Daniel Blanco Calviño

ÍNDICES

user

id	username	email	created_date
U1	daniel	daniel@email.com	1697300359
U2	noelia	noelia@email.com	1695300359
U3	maria	maria@email.com	1695300400
U4	daniel	daniel2@email.com	1695300405

```
SELECT * FROM User WHERE username = 'daniel';
```

ÍNDICES

user

	id	username	email	created_date
✓	U1	daniel	daniel@email.com	1697300359
✗	U2	noelia	noelia@email.com	1695300359
✗	U3	maria	maria@email.com	1695300400
✓	U4	daniel	daniel2@email.com	1695300405

```
SELECT * FROM User WHERE username = 'daniel';
```

```
CREATE index user_username_index ON user (username);
```

user

id	username	email	created_date
U1	daniel	daniel@email.com	1697300359
U2	noelia	noelia@email.com	1695300359
U3	maria	maria@email.com	1695300400
U4	daniel	daniel2@email.com	1695300405

user_username_index

id	val
daniel	U1, U4
noelia	U2
maria	U3

ÍNDICES

- **Estructuras muy eficientes para las búsquedas.**
  - Tablas Hash.
    - Complejidad O(1) en lecturas.
  - B-Tree
    - Complejidad O(log N) en lecturas.
  - Full table scan
    - Complejidad O(N) en lecturas.

CONSIDERACIONES EN EL USO DE ÍNDICES

- Se mitiga la baja eficiencia en lecturas haciendo un **compromiso en las escrituras.**
  - Las inserciones, eliminaciones y actualizaciones serán más lentas.

- Se hace **uso de mayor espacio de almacenamiento.**

- Indexar los campos **estrictamente necesarios.**
  - Lecturas >>> escrituras y necesidad de lecturas eficientes.

FORMAS NORMALES

Reglas que se aplican para **minimizar la redundancia y evitar inconsistencias.**

- **1FN - Primera Forma Normal.**
  - Cada celda debe contener un único valor, no un conjunto de valores.

- **2FN - Segunda Forma Normal.**
  - Cumplir 1FN.
  - Todos los atributos no clave son totalmente dependientes de la clave primaria.

- **3FN - Tercera Forma Normal.**
  - Cumplir 1FN y 2FN
  - Un atributo no clave no debe depender de otro atributo no clave.

employee - PK {id, job_code}

id	job_code	job_name	name	country_code	country
E1	J1	Software Engineer	Daniel	ES	Spain
E1	J2	Teacher	Daniel	ES	Spain
E2	J3	Accountant	María	US	United States

- Esta tabla está en 1FN ya que no tiene campos con múltiples valores.

- Para 2FN todos los atributos no clave deben depender totalmente de la clave primaria
  - Con el id del empleado podemos saber el nombre, país y código del país.
  - No necesitamos job_code para ello -> **no cumple con 2FN ya que no dependen totalmente de la clave primaria.**

employee - PK {id}

id	name	country_code	country
E1	Daniel	ES	Spain
E1	Daniel	ES	Spain
E2	María	US	United States

job - PK {id}

id	name
J1	Software Engineer
J2	Teacher
J3	Accountant

employee_jobs - PK {id}

employee_id	job_id
E1	J1
E1	J2
E2	J3

- Ahora todos los campos dependen totalmente de sus claves. **Cumple 2FN.**

- Para 3FN los atributos no clave no pueden depender de otros atributos no clave.
  - **Dado un código de un país podemos identificar el país asociado. No cumple 3FN**

employee - PK {id}

id	name	country_id
E1	Daniel	ES
E1	Daniel	ES
E2	María	US

job - PK {id}

id	name
J1	Software Engineer
J2	Teacher
J3	Accountant

employee_jobs - PK {id}

employee_id	job_id
E1	J1
E1	J2
E2	J3

country - PK {id}

id	name
ES	Spain
US	United States

NORMALIZACIÓN

- La normalización ayuda a **mantener la consistencia y minimiza la redundancia.**
  - Si modificamos el nombre de una profesión en la tabla original, debemos modificarlo en todas las filas donde se utilice.

- La contrapartida es un **menor rendimiento en las lecturas.**
  - En la tabla original teníamos toda la información de interés.
  - Ahora debemos realizar múltiples joins.

EJEMPLO DESNORMALIZACIÓN

book

id	name	author_id	year_published
B1	Book 1	A1	2015
B2	Book 2	A2	2024
B3	Book 3	A1	2001

author

id	name
A1	Author 1
A2	Author 2

EJEMPLO DESNORMALIZACIÓN

book

id	name	author_id	year_published
B1	Book 1	A1	2015
B2	Book 2	A2	2024
B3	Book 3	A1	2001

author

id	name
A1	Author 1
A2	Author 2

book_denormalized

book_id	book_name	author_id	author_name	year_published
B1	Book 1	A1	Author 1	2015
B2	Book 2	A2	Author 2	2024

DESNORMALIZACIÓN

- Se consigue un **mayor rendimiento sacrificando consistencia.**
  - Tenemos que actualizar los datos en múltiples ubicaciones.
    - Mayor riesgo de sufrir inconsistencias.

- Además se hace uso de un **mayor espacio de almacenamiento.**

- Utilizar cuando se necesite mejor eficiencia en las lecturas y **no podamos solucionarlo optimizando las consultas existentes o añadiendo índices.**