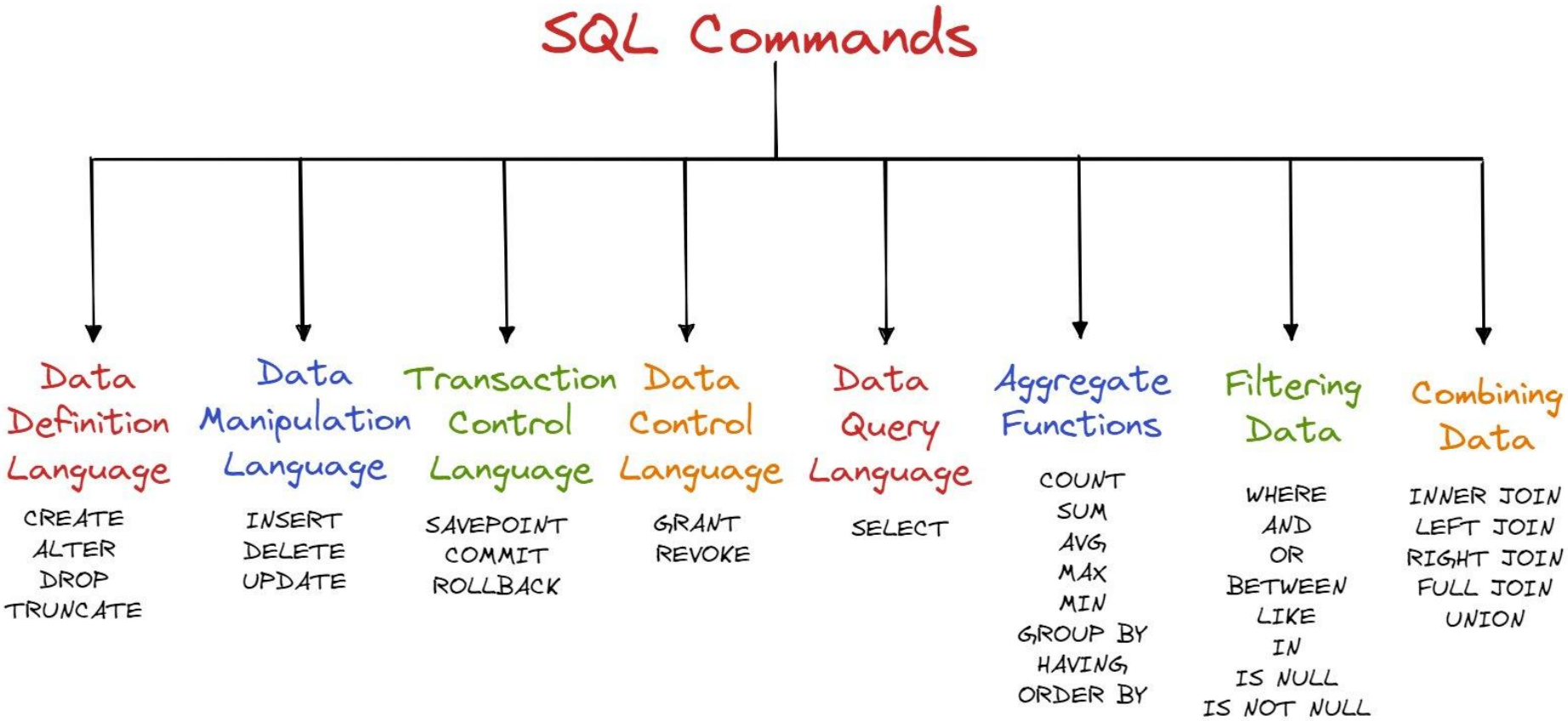
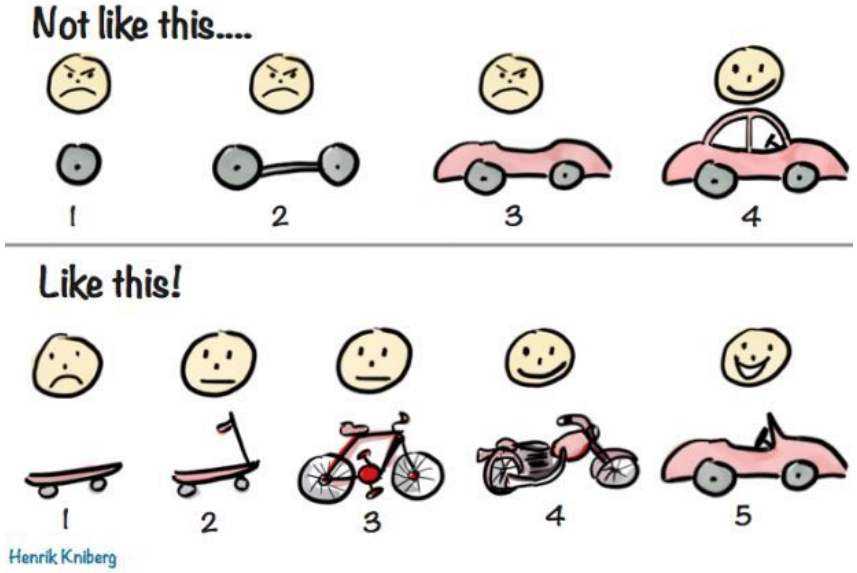
Formas Normales

… criterios para determinar la robustez del modelado de una base de datos

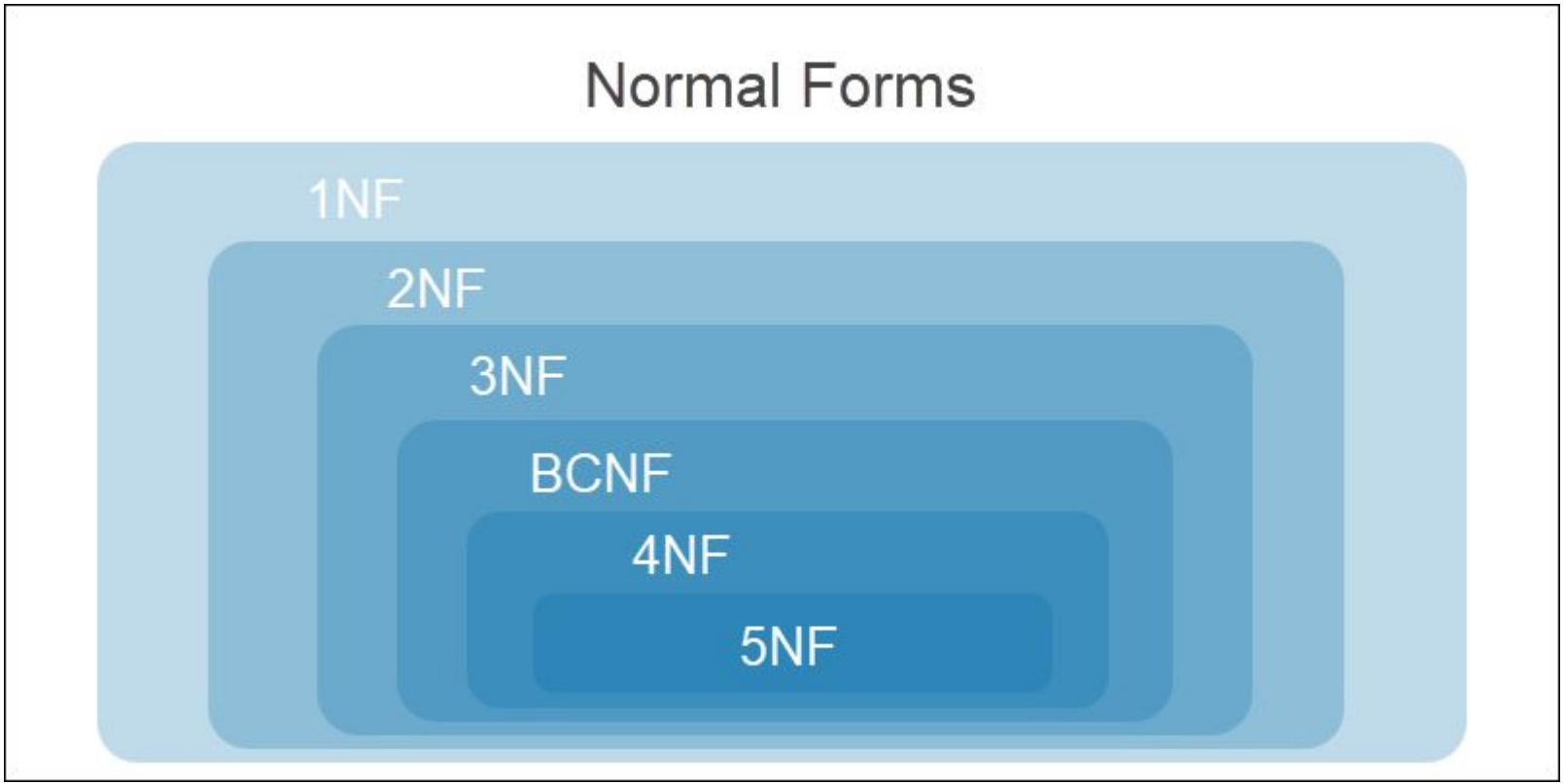

SQL Commands
Data Definition Language
CREATE
ALTER
DROP
TRUNCATE
Data Manipulation Language
INSERT
DELETE
UPDATE
Transaction Control Language
SAVEPOINT
COMMIT
ROLLBACK
Data Control Language
GRANT
REVOKE
Data Query Language
SELECT
Aggregate Functions
COUNT
SUM
AVG
MAX
MIN
GROUP BY
HAVING
ORDER BY
Filtering Data
WHERE
AND
OR
BETWEEN
LIKE
IN
IS NULL
IS NOT NULL
Combining Data
INNER JOIN
LEFT JOIN
RIGHT JOIN
FULL JOIN
UNION

Definición. ¡¡¡Es un Proceso!!!

La normalización de bases de datos es una técnica para crear tablas con columnas y claves adecuadas, mediante la descomposición de una tabla grande en unidades lógicas (tablas) más pequeñas. Es un proceso **iterativo incremental**



Definición. ¡¡¡Es un Proceso!!!



¿Dónde pone el foco?

- Atributos con **múltiples** valores.
- Atributos duplicados o **repetidos**.
- Atributos **no descriptivos**.
- Atributos con **información redundante**.
- Atributos creados **a partir de otras entidades**.

¿Cuál es el objetivo?

- Una estructura de base de datos adecuada para consultas generalizadas.
- Minimizar la redundancia de datos, ergo aumentando la eficiencia de la memoria en un servidor de base de datos.
- Integridad de datos maximizada a través de una reducción de las **anomalías** de inserción, actualización y eliminación.

¿Qué es una anomalía?

employee	manager	sector
Jacob	Adam	IT
Isaac	Adam	IT
David	Joshua	Finance

1. Anomalía a insertar. Al intentar insertar un nuevo empleado en el sector financiero, también debe saber el nombre del gerente. De lo contrario, no puede insertar datos en la tabla.

¿Qué es una anomalía?

employee	manager	sector
Jacob	Adam	IT
Isaac	Adam	IT
David	Joshua	Finance

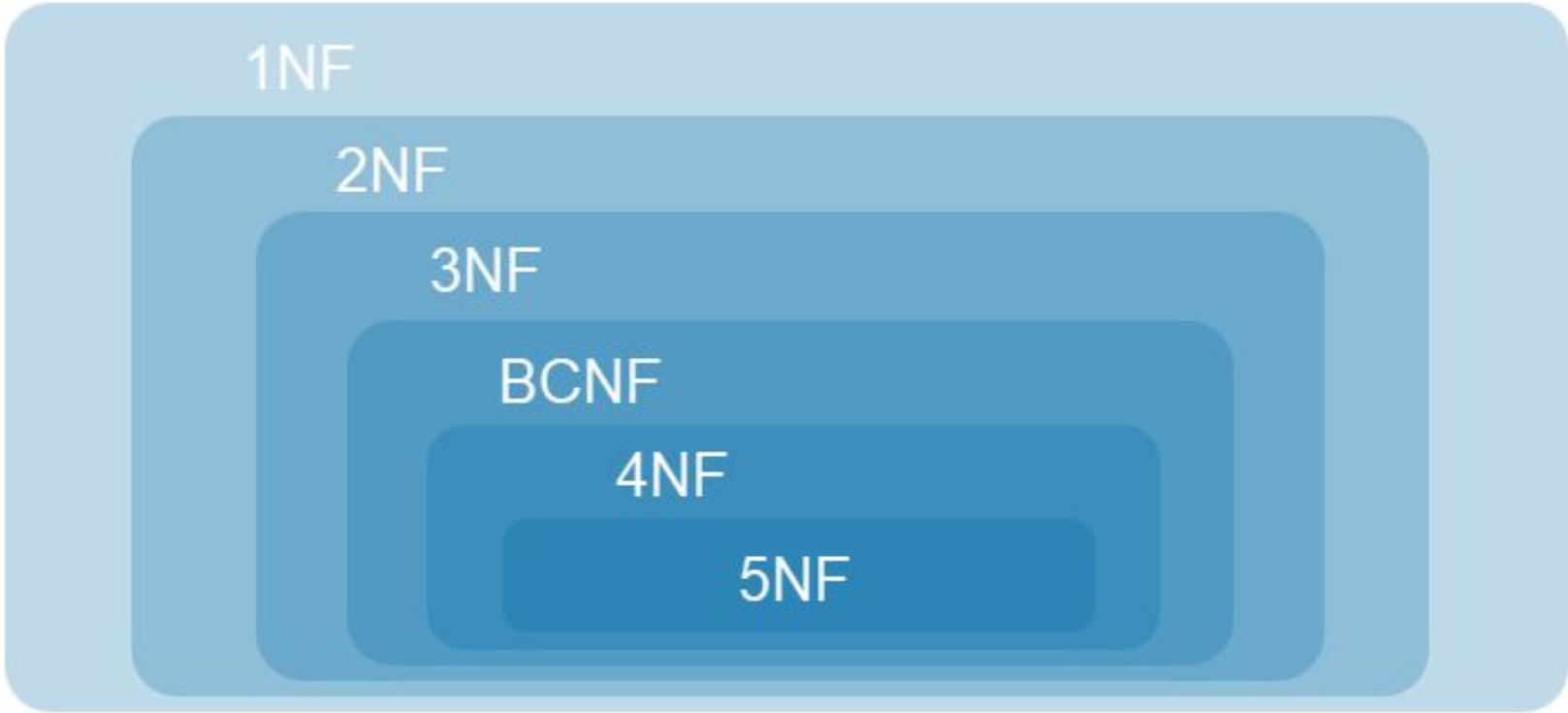
2. Anomalía de actualización. Si un empleado cambia de sector, el nombre del gerente termina siendo incorrecto. Por ejemplo, si Jacob cambia a finanzas, Adam permanece como su gerente.

¿Qué es una anomalía?

employee	manager	sector
Jacob	Adam	IT
Isaac	Adam	IT
David	Joshua	Finance

3. Anomalía al eliminar. Si Joshua decide dejar la empresa, al eliminar la fila también se elimina la información de que existe un sector financiero.

Normal Forms



Las 5 Formas Normales

Forma no normalizada → Antes de cualquier normalización. Existen valores redundantes y complejos.

Primera forma normal → Los valores repetidos y complejos se dividen, haciendo que todas las instancias sean atómicas.

Segunda forma normal → Las dependencias parciales se descomponen en tablas nuevas. Todas las filas dependen funcionalmente de la clave principal.

Tercera forma normal → Las dependencias transitivas se descomponen en tablas nuevas. Los atributos no clave dependen de la clave principal.


THAT'S ENOUGH
makeameme.org

Las 5 Formas Normales

Forma normal de Boyce-Codd → Las dependencias funcionales transitivas y parciales para todas las claves candidatas se descomponen en tablas nuevas.

Cuarta Forma Normal → Eliminación de dependencias multivaluadas.

Quinta forma normal → Eliminación de dependencias JOIN.

Concepto fundamental para la Normalización → Tipos de KEY (Clave)

- **Súper Clave.** Un conjunto de características que definen de forma única cada registro en una tabla.
- **Clave candidata**. Claves seleccionadas del conjunto de superclaves donde el número de campos es mínimo.
- **Clave primaria (PK)**. La elección más adecuada del conjunto de claves candidatas sirve como clave principal de la tabla.
- **Clave externa (FK)**. La clave principal de otra tabla.
- **Clave compuesta**. Dos o más atributos juntos forman una clave única pero no son claves individualmente.

Ejemplo. Identificando posibles claves

employeeID	name	age	email
1	Adam A.	30	adam.a@email.com
2	Jacob J.	27	jacob.j@email.com
3	David D.	35	david.d@email.com

EJEMPLO. De UFN a 3FN

managerID	managerName	area	employeeID	employeeName	sectorID	sectorName
1	Adam A.	East	1	David D.	4	Finance
			2	Eugene E.	3	IT
2	Betty B.	West	3	George G.	2	Security
			4	Henry H.	1	Administration
			5	Ingrid I.	4	Finance
3	Carl C.	North	6	James J.	1	Administration
			7	Katy K.	4	Finance

EJEMPLO. De UFN a 3FN

managerID	managerName	area	employeeID	employeeName	sectorID	sectorName
1	Adam A.	East	1	David D.	4	Finance
1	Adam A.	East	2	Eugene E.	3	IT
2	Betty B.	West	3	George G.	2	Security
2	Betty B.	West	4	Henry H.	1	Administration
2	Betty B.	West	5	Ingrid I.	4	Finance
3	Carl C.	North	6	James J.	1	Administration
3	Carl C.	North	7	Katy K.	4	Finance

EJEMPLO. De UFN a 3FN

managerID	managerName	area
1	Adam A.	East
2	Betty B.	West
3	Carl C.	North

employeeID	employeeName	managerID	sectorID	sectorName
1	David D.	1	4	Finance
2	Eugene E.	1	3	IT
3	George G.	2	2	Security
4	Henry H.	2	1	Administration
5	Ingrid I.	2	4	Finance
6	James J.	3	1	Administration
7	Katy K.	3	4	Finance

EJEMPLO. De UFN a 3FN

Employee

employeeID	employeeName	managerID	sectorID
1	David D.	1	4
2	Eugene E.	1	3
3	George G.	2	2
4	Henry H.	2	1
5	Ingrid I.	2	4
6	James J.	3	1
7	Katy K.	3	4

Sector

sectorID	sectorName
1	Administration
2	Security
3	IT
4	Finance

Manager

managerID	managerName	area
1	Adam A.	East
2	Betty B.	West
3	Carl C.	North

Consejos y recomendaciones. Recursos

- Identificar en qué forma normal está el modelado con el que trabajo en mi proyecto
  - Adquirir contexto y saber de la semántica asociada
- Jerga: “esta tabla no está normalizada” → Ojo NoSQL
- No todo el mundo sabe normalizar.. y no siempre se puede
- Si quieres aprender a modelar, plantéate problemas
- Mira la performance de código con bbdd con anomalías
  - Refactoriza y compara (benchmark)
- Tutorial → https://www.studytonight.com/dbms/database-normalization.php


THANK YOU
FOR YOUR ATTENTION