

**HENRY**

Data Science

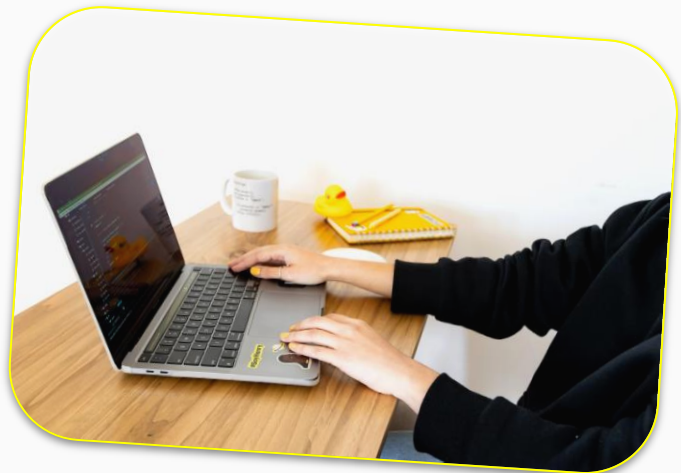


# KPI'S & BUSINESS APPLICATIONS





# Agenda



- KPIs & Business Applications
- Funciones en SQL
- Orden de las sentencias



---

# OBJETIVOS DE LA CLASE

*Al finalizar esta lecture estarás en la capacidad de...*

- **Comprender** el concepto y características de los KPI.
- **Conocer** algunos ejemplos de los KPI's más importantes.

**¿Qué es un KPI?**



**Indicador clave** de rendimiento, es útil para el seguimiento de los resultados en función de un objetivo definido.





# Características

## Alcanzable

Los objetivos deben ser **realistas**.

## Medible

Aunque suene obvio, un KPI debe de poder **medirse**.

## Relevante

No te llenes de datos, selecciona sólo los **más importantes**.

## Periódico


El indicador tiene que ser **analizable** periódicamente.


## Exacto

Elige sólo la parte más **precisa** de toda información recabada.



 Un **KPI** es un número, que puede obtenerse de **manera directa o mediante alguna fórmula de cálculo.**

 Por ejemplo, si el objetivo es incrementar las ventas anuales un 25 %, nuestro **KPI** sería el resultado de **"VENTAS ACTUALES x 1.25"**.

 Ese mismo KPI podría expresarse en N° de unidades, volumen de ventas, etc., depende del **objetivo**.



# Ejemplos de KPI's





# KPI's relevantes en tecnología

Runway

CAC

ROI

Conversion rate

LTV

churn rate



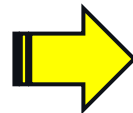


# Runway



Es la cantidad de tiempo de vida que tiene una startup antes de quedarse sin efectivo, medido en número de meses.

$$\text{Runway} = \text{Efectivo} \div \text{Gastos mensuales totales. (Expresado en meses).}$$





**Por ejemplo, si una startup gasta USD \$ 10 mil al mes y tienes USD \$ 50 mil en caja, tiene 5 meses de “vida”.**





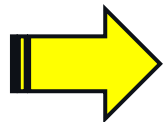
# Conversion rate

---



Permite conocer la capacidad de monetizar un negocio.

La tasa de conversión es la cantidad de clientes que compraron mi producto sobre la cantidad total de clientes **"registrados"** o **"interesados"**.





## Para cada paso se puede medir una conversión:



**Convertir registros**  $\frac{\text{registros}}{\text{visitas al sitio}}$



**Convertir inicios de procesos**  $\frac{\text{inicios de procesos}}{\text{registros}}$



**Convertir compras**  $\frac{\text{Compras}}{\text{Inicios del proceso de compra.}}$



# CAC



Los modelos de atribución permiten fidelizar esta métrica a partir de poder distinguir de qué campañas o anuncios se atribuye un cliente.

**CAC**



Suma de los costos de  
adquisición directos e  
indirectos de cada  
canal



Número total de  
usuarios captados





# LVT



Es el valor neto de ingresos que genera un cliente durante el tiempo que permanece en cartera.

$$\text{LVT} = \text{Compra media} \times \text{Cantidad de compras} \times \text{Tiempo activo en cartera}$$





# ROI



ROI (Return On Investment):

Esta métrica permite medir el retorno de un determinado proyecto o una empresa, puede darse el caso de que sea en base a estimaciones o variables concretas.

$$\text{ROI} = \frac{\text{Ingresos} - \text{Inversión total del proyecto}}{\text{Inversión total del proyecto}}$$







## ROI de una campaña de Facebook Ads

$$\text{ROI} = \left( \text{CI} - \text{IT} \right) \div \text{IT}$$

**CT:** Compras totales de clientes que fueron captados por Facebook Ads.

**IT:** Inversión total en anuncio de Facebook Ads





# Churn rate



El **churn rate** es la tasa de cancelación o abandono registrada en la cartera de clientes.

**churn  
rate**

=

Clientes de  
baja del  
período

÷

Clientes al cierre del  
período — Nuevos  
clientes del período





# Funciones en SQL



# Funciones en SQL

Para construir KPI es importante relacionar **distintas variables** y obtener conclusiones a partir de los cálculos que realizamos. Comencemos ahora con un **primer acercamiento en SQL.**



# ORDER BY

---

--Ordena los registros por apellido de la A-Z

```
SELECT nombre, apellido, fechaIngreso  
FROM alumnos  
WHERE carrera = "Data Science"  
ORDER BY apellido ASC
```

--Ordena los registros por la fecha de ingreso de la más reciente a la más antigua y luego por apellido de la A-Z.

```
SELECT nombre, apellido, fechaIngreso  
FROM alumnos  
WHERE carrera = "Data Science"  
ORDER BY fechaIngreso DESC, apellido ASC
```



# LIMIT

--Devuelve los 10 primeros.

```
SELECT nombre, apellido, fechaIngreso  
FROM alumnos  
WHERE carrera = "Data Science"  
LIMIT 10;
```

--Devuelve del registro 6 al 15

```
SELECT nombre, apellido, fechaIngreso  
FROM alumnos  
WHERE carrera = "Data Science"  
LIMIT 5,10;
```



# COUNT

---

```
SELECT COUNT(*)  
FROM alumnos  
WHERE carrera = "Data Science"
```



# SUM

---

--Obtenemos el ingreso por pagos de AIC.

```
SELECT SUM(PagoAIC)
```

```
FROM alumnos
```

```
WHERE carrera = "Data Science" and graduado = "true"
```





# AVG

---

```
--Obtenemos el ingreso promedio por pagos de AIC.  
SELECT AVG(PagoAIC)  
FROM alumnos  
WHERE carrera = "Data Science" and graduado = "true"
```



# MAX - MIN

---

--Obtenemos cuando fue el último ingreso de los alumnos graduados.

```
SELECT MAX(fechaIngreso)
```

```
FROM alumnos
```

```
WHERE carrera = "Data Science" and graduado = "true"
```

--Obtenemos cuando fue el primer ingreso de los alumnos graduados.

```
SELECT MIN(fechaIngreso)
```

```
FROM alumnos
```

```
WHERE carrera = "Data Science" and graduado = "true"
```



# GROUP BY

---

--Cantidad de alumnos por carrera

```
SELECT carrera, COUNT(*)
```

```
FROM alumnos
```

```
GROUP BY carrera
```

--Cantidad de alumnos por carrera por fecha de ingreso

```
SELECT carreras, fechaIngreso, COUNT(*)
```

```
FROM alumnos
```

```
GROUP BY carreras, fechaIngreso
```



# HAVING

---

--Listado de cohortes con más de 50 alumnos.

```
SELECT cohortes, COUNT(*)
```

```
FROM alumnos
```

```
GROUP BY carrera
```

```
HAVING COUNT (*) > 50
```



# ORDEN DE LAS SENTENCIAS

01

**SELECT:** definir los campos a mostrar en la consulta, cuando utilizamos (\*) mostramos todos los campos.

02

**FROM:** establecer desde qué tabla se obtendrán los campos.

03

**WHERE:** Condiciones de filtro.



# ORDEN DE LAS SENTENCIAS

04

**GROUP BY:** Agrupar los campos en caso de tener funciones de agregación en SELECT.

05

**HAVING:** Se establecen filtros sobre los campos agregados, es importante agrupar antes de filtrar con HAVING.

06

**ORDER BY:** Por último, ordenamos según los criterios necesarios.

# HENRY

