



## Actividad

“Operadores y Forma de Trabajo del QBE”



**Departamento de Ciencias  
de la Computación**

**Asignatura:**

*“Bases de Datos”*

**Profesor:**

Cesar Eduardo Velázquez Amador

**Fecha:**

22 de noviembre de 2024

**Alumnos:**

- Salvador Alejandro Aguilera Macias
- Cesar Emmanuel Arriola Pizaña
- Luis Pablo Esparza Terrones
- Juan Francisco Gallo Ramírez
- Humberto Martínez López

**Ingeniería en Computación  
Inteligente**

5to Semestre

# Definición del QBE

---

Query By Example (QBE) es un método de consulta que permite a los usuarios recuperar información de una base de datos rellenando un formulario o una tabla con los datos deseados como ejemplo. Estos ejemplos actúan como un patrón o modelo, y el sistema traduce automáticamente la información proporcionada en una consulta estructurada (por ejemplo, en lenguaje SQL) que se ejecuta en la base de datos.

El sistema generalmente funciona en interfaces visuales donde los usuarios pueden:

- Seleccionar campos específicos de interés.
- Introducir valores o condiciones en columnas relacionadas.
- Combinar diferentes condiciones usando opciones de selección preconfiguradas (por ejemplo, operadores como AND y OR).

Este enfoque elimina la necesidad de escribir código directamente, lo que lo hace especialmente útil para usuarios que no tienen conocimientos avanzados de lenguajes de consulta.

## Características Principales de QBE

### 1. Simplicidad:

QBE permite formular consultas complejas sin necesidad de escribir código, lo que lo hace accesible incluso para usuarios sin experiencia técnica.

### 2. Interfaz Visual:

Utiliza un formulario o un entorno gráfico en el que los usuarios interactúan con los datos de manera intuitiva, seleccionando y definiendo condiciones de búsqueda.

### 3. Traducción Automática:

El sistema convierte automáticamente las entradas del usuario en sentencias de lenguajes de consulta como SQL.

### 4. Flexibilidad:

Soporta consultas básicas y avanzadas, como búsqueda por patrones, condiciones múltiples y combinaciones entre tablas.

## 5. Resultados Inmediatos:

Los usuarios pueden ver los resultados de sus consultas rápidamente, lo que mejora la eficiencia en la gestión y análisis de datos.

### Ventajas y Desventajas de QBE

#### Ventajas:

1. **Facilidad de Uso:** Ideal para usuarios no técnicos, ya que no requiere conocimientos en lenguajes como SQL.
2. **Rapidez:** Las consultas se formulan rápidamente usando formularios predefinidos.
3. **Aprendizaje Intuitivo:** La interfaz visual permite a los usuarios comprender cómo interactúan con los datos de forma más natural.

#### Desventajas:

- **Limitaciones en Consultas Complejas:** Aunque adecuado para tareas comunes, puede ser insuficiente para consultas altamente personalizadas o específicas.
- **Dependencia de la Interfaz:** Solo está disponible en sistemas que implementen herramientas gráficas compatibles.
- **Restricción de Funcionalidades:** Puede no soportar todas las capacidades avanzadas de lenguajes como SQL.

### Impacto y Aplicaciones Modernas de QBE

El enfoque de QBE ha influido en el diseño de herramientas modernas de bases de datos, especialmente aquellas orientadas al usuario final, como Microsoft Access, SAP HANA, y sistemas de gestión de datos en la nube. Actualmente, QBE sigue siendo relevante en entornos donde la accesibilidad y la facilidad de uso son prioritarias, como la generación de reportes, análisis de datos empresariales y la educación.

Aunque el dominio de QBE en el campo de las bases de datos ha sido superado por interfaces más avanzadas, su legado permanece en la evolución de tecnologías que priorizan la experiencia del usuario.

# Historia del QBE

---

Query By Example (QBE) fue desarrollado en 1977 por Moshé M. Zloof, un investigador de IBM, como parte de su trabajo en el proyecto del Sistema R, uno de los primeros prototipos de bases de datos relacionales. En ese momento, las bases de datos relacionales eran una innovación que permitía estructurar la información de forma tabular, pero aún presentaban barreras técnicas para los usuarios no especializados debido al uso de lenguajes de consulta complejos, como SQL.

El objetivo principal de Zloof al crear QBE era reducir estas barreras mediante un método de consulta más intuitivo. Este método permitió que los usuarios pudieran interactuar con los datos mediante una interfaz gráfica o un formulario en lugar de depender exclusivamente de comandos de texto. Zloof buscaba una forma de "mostrar con ejemplos" lo que el usuario quería buscar, en lugar de requerir que el usuario escribiera instrucciones específicas.

Desde su introducción, QBE tuvo un impacto significativo en el diseño de sistemas de bases de datos modernos y se convirtió en una base para las herramientas de consulta visuales utilizadas en programas como Microsoft Access, DB2 de IBM y otras aplicaciones gráficas.

# Operadores usados en QBE

En QBE, los operadores se utilizan para definir las condiciones que se aplicarán a los campos de la consulta. Estos operadores permiten filtrar, comparar y establecer relaciones entre datos. Aunque pueden variar ligeramente según la implementación, los operadores comunes incluyen los siguientes:

## 1. Operadores Lógicos

Estos operadores se usan para combinar múltiples condiciones en una consulta.

- **AND**: Requiere que todas las condiciones especificadas sean verdaderas.

*Ejemplo: En un formulario, escribir >10 AND <50 en un campo seleccionará valores entre 11 y 49.*

- **OR**: Requiere que al menos una de las condiciones sea verdadera.

*Ejemplo: =20 OR =30 seleccionará filas donde el valor sea 20 o 30.*

- **NOT**: Excluye las filas que cumplen con la condición especificada.

*Ejemplo: NOT =100 excluirá los valores iguales a 100.*

## 2. Operadores de Comparación

Estos operadores permiten establecer condiciones específicas sobre los valores de los campos.

- **= (Igual a)**: Busca valores que coincidan exactamente.

*Ejemplo: =100 selecciona registros donde el campo es igual a 100.*

- **<> (Diferente de)**: Excluye registros con el valor especificado.

*Ejemplo: <>50 selecciona todos los valores excepto 50.*

- **> (Mayor que)**: Busca valores mayores que el especificado.

*Ejemplo: >25 selecciona valores mayores a 25.*

- **< (Menor que)**: Busca valores menores que el especificado.

*Ejemplo: <10 selecciona valores menores a 10.*

- **>= (Mayor o igual):** Incluye valores mayores o iguales al especificado.

*Ejemplo: >=15 selecciona valores mayores o iguales a 15.*

- **<= (Menor o igual):** Incluye valores menores o iguales al especificado.

*Ejemplo: <=20 selecciona valores menores o iguales a 20.*

### 3. Operadores de Patrones y Comodines

Estos operadores son útiles para buscar datos parcialmente coincidentes.

- **LIKE (Similares a):** Busca valores que coincidan con un patrón.

*Ejemplo: LIKE 'A%' seleccionará todos los valores que comiencen con la letra "A".*

- **\* (Comodín para cualquier número de caracteres):** Representa cero o más caracteres.

*Ejemplo: \*car seleccionará valores como "car", "scar", "racecar".*

- **? (Comodín para un solo carácter):** Representa exactamente un carácter.

*Ejemplo: ?ar seleccionará valores como "bar", "car", pero no "scar".*

### 4. Operadores Matemáticos

Estos operadores permiten realizar cálculos en las consultas.

- **+ (Suma):** Agrega valores.

*Ejemplo: Salario + Bono calcula el total de salario más bono.*

- **- (Resta):** Resta valores.

*Ejemplo: Cantidad - Descuento calcula la cantidad neta después del descuento.*

- **/ (División):** Divide valores.

*Ejemplo: Total / Número de Productos calcula el precio promedio.*

- **\* (Multiplicación):** Multiplica valores.

*Ejemplo: Precio \\* Cantidad calcula el costo total.*

## 5. Operadores de Agregación

Estos operadores permiten resumir o agrupar datos.

- **SUM:** Calcula la suma total de un campo.

*Ejemplo: SUM(Cantidad) da el total de todas las cantidades.*

- **AVG:** Calcula el promedio de un campo.

*Ejemplo: AVG(Salario) da el salario promedio.*

- **COUNT:** Cuenta el número de registros.

*Ejemplo: COUNT(\*) da el número total de filas.*

- **MAX:** Encuentra el valor máximo en un campo.

*Ejemplo: MAX(Edad) da la edad más alta.*

- **MIN:** Encuentra el valor mínimo en un campo.

*Ejemplo: MIN(Edad) da la edad más baja.*

## 6. Operadores Especiales

Algunos sistemas QBE ofrecen operadores específicos adicionales:

- **IS NULL:** Selecciona registros donde el campo no tiene valor.

*Ejemplo: IS NULL en un campo selecciona filas con valores vacíos.*

- **IS NOT NULL:** Selecciona registros donde el campo tiene algún valor.

*Ejemplo: IS NOT NULL selecciona filas con valores definidos.*

- **BETWEEN ... AND ... :** Selecciona valores dentro de un rango.

*Ejemplo: BETWEEN 10 AND 20 selecciona valores de 10 a 20 inclusive.*

# Forma de trabajo de QBE

El funcionamiento de QBE se basa en un enfoque visual y simplificado que permite a los usuarios interactuar con bases de datos relacionales mediante formularios o interfaces gráficas, en lugar de escribir código directamente. A continuación, se describe paso a paso cómo opera QBE:

## 1. Interfaz Gráfica Basada en Tablas

En QBE, los usuarios trabajan con una interfaz gráfica que representa las tablas de la base de datos como formularios. Cada formulario:

- Muestra los nombres de los campos (columnas) de la tabla.
- Proporciona espacios en blanco donde los usuarios pueden ingresar valores, condiciones o patrones.

Por ejemplo, si se tiene una tabla de "Empleados" con las columnas Nombre, Departamento y Salario, la interfaz mostrará estas columnas en un formato amigable.

## 2. Provisión de Ejemplos

Los usuarios introducen ejemplos en los campos de interés para especificar qué datos desean recuperar. Estos ejemplos actúan como condiciones para filtrar los datos. Los ejemplos pueden incluir:

- Valores específicos (e.g., "Ventas" en el campo Departamento).
- Condiciones lógicas (e.g., >30000 en el campo Salario).
- Patrones de búsqueda (e.g., J\* para nombres que comienzan con "J").

El sistema entiende estas entradas como una guía para construir la consulta.

## 3. Traducción Automática

Una vez que el usuario proporciona los ejemplos:

- El sistema de QBE traduce automáticamente la información en una consulta estructurada, generalmente en SQL.



- Por ejemplo, si el usuario introduce >50000 en el campo "Salario", el sistema genera una consulta SQL como:

**SELECT \* FROM Empleados WHERE Salario > 50000;**

Este proceso es completamente transparente para el usuario.

#### 4. Ejecución de la Consulta

El sistema ejecuta la consulta generada en la base de datos y recupera los registros que cumplen las condiciones especificadas. Estos resultados son presentados al usuario en un formato tabular, similar a cómo se muestran las tablas en la base de datos.

#### 5. Refinamiento Iterativo

El usuario puede refinar la consulta ajustando o agregando ejemplos en el formulario. Cada ajuste genera una nueva consulta, lo que facilita un enfoque iterativo y experimental para encontrar los datos deseados.

##### Ejemplo Práctico

Supongamos que un usuario desea buscar empleados del departamento "Ventas" que ganen más de 30,000. El flujo sería:

##### 1. Formulario QBE:

El sistema muestra una tabla con los campos:

*Nombre | Departamento | Salario*

##### 2. Introducción de Ejemplos:

El usuario introduce:

- "Ventas" en el campo Departamento.
- >30000 en el campo Salario.

##### 3. Traducción a SQL:

El sistema genera automáticamente la consulta:

*SELECT \* FROM Empleados WHERE Departamento = 'Ventas' AND Salario > 30000;*

## Resultados:

Se muestran los registros que cumplen las condiciones:

Nombre	Departamento	Salario
Ana Pérez	Ventas	45000
Luis García	Ventas	35000

## Ventajas del Modo de Trabajo de QBE

- **Accesibilidad:** Los usuarios no necesitan saber SQL ni otros lenguajes de consulta.
- **Intuitivo:** La entrada de ejemplos en formularios es más natural para usuarios no técnicos.
- **Iterativo:** Permite ajustar consultas rápidamente para obtener resultados más precisos.
- **Visualización Directa:** Los resultados se presentan de inmediato en un formato reconocible.

QBE trabaja como un intermediario entre el usuario y la base de datos, traduciendo acciones simples y visuales en consultas avanzadas. Su facilidad de uso lo convierte en una herramienta poderosa para aquellos que necesitan recuperar y analizar datos sin tener un conocimiento profundo de programación o bases de datos.

# Ejemplos de consultas y operaciones

---

En QBE, los operadores se utilizan para definir las condiciones que se aplicarán a los campos de la consulta. Estos operadores permiten filtrar, comparar y establecer relaciones entre datos. Aunque pueden

---

## 1. Seleccionar Todos los Registros de una Tabla

Descripción:

Recuperar todos los registros de una tabla sin condiciones.

Entrada en QBE:

Dejar todos los campos vacíos en el formulario

Consulta SQL:

```
SELECT * FROM Empleados;
```

---

## 2. Filtrar Registros por un Valor Exacto

Descripción:

Obtener empleados que pertenezcan al departamento "Ventas".

Entrada en QBE:

- En el campo *Departamento*, escribir: Ventas

Consulta SQL:

```
SELECT * FROM Empleados WHERE Departamento = 'Ventas';
```

---

## 3. Filtrar por una Condición Numérica

Descripción:

Recuperar empleados con salario mayor a 30,000.

Entrada en QBE:

- En el campo *Salario*, seleccionar: mayor, y escribir: 30000

Consulta SQL:

```
SELECT * FROM Empleados WHERE Salario > 30000;
```

---

#### 4. Filtrar por un Rango de Valores

Descripción:

Obtener empleados con edades entre 25 y 35 años.

Entrada en QBE:

- En el campo *Edad*, escribir el rango: 25 - 35

Consulta SQL:

```
SELECT * FROM Empleados WHERE Edad BETWEEN 25 AND 35;
```

---

#### 5. Filtrar por Valores Nulos

Descripción:

Buscar empleados cuyo campo "Teléfono" esté vacío.

Entrada en QBE:

- En el campo *Teléfono*, escribir: IS NULL

Consulta SQL:

```
SELECT * FROM Empleados WHERE Teléfono IS NULL;
```

---

#### 6. Filtrar Usando Comodines

Descripción:

Recuperar empleados cuyos nombres comiencen con "A".

Entrada en QBE:

- En el campo *Nombre*, escribir: A\*

Consulta SQL:

```
SELECT * FROM Empleados WHERE Nombre LIKE 'A%';
```

---

## 7. Filtrar por Múltiples Condiciones

### Descripción:

Buscar empleados del departamento "Ventas" con salarios mayores a 40,000.

### Entrada en QBE:

- En el campo *Departamento*, escribir: Ventas
- En el campo *Salario*, seleccionar: mayor, y escribir: 40000

### Consulta SQL:

```
SELECT * FROM Empleados  
WHERE Departamento = 'Ventas' AND Salario > 40000;
```

---

## 8. Ordenar Registros

### Descripción:

Ordenar los empleados por salario en orden descendente.

### Entrada en QBE:

- En el campo *Salario*, escribir: DESC

### Consulta SQL:

```
SELECT * FROM Empleados ORDER BY Salario DESC;
```

---

## 9. Contar Registros

### Descripción:

Contar cuántos empleados hay en la base de datos.

### Entrada en QBE:

- En cualquier campo, escribir: COUNT(\*)

### Consulta SQL:

```
SELECT COUNT(*) FROM Empleados;
```

---

## 10. Operación de Unión entre Tablas

### Descripción:

Obtener una lista de productos junto con los nombres de sus proveedores.

### Entrada en QBE:

Relacionar las tablas Productos y Proveedores a través del campo común (e.g., ID\_Proveedor), y seleccionar los campos deseados:

- En Productos.Nombre: Marcar como "Seleccionar".
- En Proveedores.Nombre: Marcar como "Seleccionar".

### Consulta SQL:

```
SELECT Productos.Nombre, Proveedores.Nombre  
FROM Productos INNER JOIN Proveedores  
ON Productos.ID_Proveedor = Proveedores.ID;
```

---

## 11. Agrupar y Filtrar por Agregados

### Descripción:

Obtener el salario promedio de los empleados por departamento y filtrar los departamentos con un salario promedio mayor a 35,000.

### Entrada en QBE:

- En el campo *Departamento*, marcar "Seleccionar"
- En el campo *Salario*, escribir: AVG(Salario)
- En el campo *Departamento*, aplicar el filtro: > 35000

### Consulta SQL:

```
SELECT Departamento, AVG(Salario) AS SalarioPromedio  
FROM Empleados  
GROUP BY Departamento HAVING AVG(Salario) > 35000;
```

---

## 12. Filtrar por Múltiples Valores (IN)

### Descripción:

Obtener el salario promedio de los empleados por departamento y filtrar los departamentos con un salario promedio mayor a 35,000.

#### Entrada en QBE:

- En el campo *Departamento*, marcar "Seleccionar".
- En el campo *Salario*, escribir: AVG(Salario)
- En el campo *Departamento*, aplicar el filtro: > 35000

#### Consulta SQL:

```
SELECT Departamento, AVG(Salario) AS SalarioPromedio  
FROM Empleados  
GROUP BY Departamento HAVING AVG(Salario) > 35000;
```

# Bibliografía

---

Zloof, M. M. (1977). Query-by-Example: A Data Base Language. IBM Systems Journal, 16(4), 324-343. <https://doi.org/10.1147/sj.164.0324>

Date, C. J. (2003). An Introduction to Database Systems (8.<sup>a</sup> ed.). Addison-Wesley.

Ramakrishnan, R., & Gehrke, J. (2000). Database Management Systems (2.<sup>a</sup> ed.). McGraw-Hill.

Wikipedia contributors. (n.d.). Microsoft Query by Example. Wikipedia. Recuperado el 22 de noviembre de 2024, de [https://en.wikipedia.org/wiki/Microsoft\\_Query](https://en.wikipedia.org/wiki/Microsoft_Query)

Wikipedia contributors. (n.d.). Query by Example. Wikipedia. Recuperado el 22 de noviembre de 2024, de [https://en.wikipedia.org/wiki/Query\\_by\\_Example](https://en.wikipedia.org/wiki/Query_by_Example)