

## **Vistas como mecanismo de seguridad para bases de datos empresariales**

Aguado Axel Tomás, Román Gabriel Esteban, Cuzziol Juan.

Fac. de Ciencias Exactas Naturales y Agrimensura, Universidad Nacional del Nordeste,  
Corrientes.

Lic. en Sistemas de Información

Año 2023

[axelaguado17@gmail.com](mailto:axelaguado17@gmail.com), [gbrrielroman98@gmail.com](mailto:gbrrielroman98@gmail.com), [jcuzziol@gmail.com](mailto:jcuzziol@gmail.com)

## **CAPÍTULO I: INTRODUCCIÓN**

### **Tema**

"Implementación de estrategias de control de acceso y seguridad en bases de datos empresariales: Un enfoque en el uso de vistas para la limitación de acceso a datos sensibles."

### **Descripción del Problema**

Este trabajo práctico se centra en la implementación de estrategias efectivas de control de acceso y seguridad en entornos de bases de datos empresariales. El problema específico a investigar es cómo las vistas en bases de datos pueden ser utilizadas de manera óptima para limitar el acceso a datos sensibles, garantizando así la confidencialidad y cumplimiento normativo. La investigación abordará la creación de vistas que oculten columnas sensibles, filtren filas según criterios de seguridad y apliquen lógica de negocio específica. Además, se explorarán las mejores prácticas para asignar permisos a usuarios y roles, así como la auditoría de acceso a vistas para fortalecer las capas de seguridad en bases de datos empresariales. El trabajo se enfocará en casos de uso prácticos y considerará la integración de estas estrategias con otros mecanismos de seguridad, como la encriptación de datos sensibles.

### **Objetivo del Trabajo Práctico**

Analizar cómo las vistas pueden contribuir a mejorar la confidencialidad e integridad de los datos en bases de datos empresariales.

### **Objetivo específico**

- ❖ Determinar cómo la creación de vistas puede reducir la exposición directa de datos sensibles y proporcionar una capa adicional de seguridad.
- ❖ Estudiar cómo se pueden aplicar restricciones de integridad en vistas para garantizar la coherencia de los datos y prevenir posibles violaciones de integridad.

## **CAPÍTULO II: MARCO CONCEPTUAL O REFERENCIAL**

En esta investigación nos introducimos en el concepto de "Vistas" y "Vistas indexadas", donde buscamos no solo definir las y caracterizarlas, sino que también conocer sus aplicaciones en el mundo real, lo que nos lleva del campo teórico al práctico permitiéndonos un mejor entendimiento.

En base a la información recolectada decidimos utilizar las "Vistas" en el marco de la seguridad, implementada como estrategia de control de acceso en bases de datos

empresariales, poniendo el foco en la limitación de acceso a datos sensibles. Para el caso de las “Vistas indexadas” el objetivo se puso en la optimización de las consultas.

Vistas y Vistas Indexadas:

Las vistas son tablas virtuales que representan una porción de los datos almacenados en una base de datos. Decimos que son virtuales debido a que no contienen datos reales, su contenido está definido por consultas que muestran una selección de columnas y registros de una o varias tablas, como también de otras vistas. Las vistas se producen de forma dinámica cuando se hace referencia a una de ellas. [1]

Las vistas en bases de datos se utilizan para varios propósitos, entre ellos:

**Simplificación:** Las vistas pueden ocultar la complejidad de las relaciones y la estructura de la base de datos. Esto facilita la realización de consultas, que pueden ser reutilizadas en múltiples ocasiones, permitiéndonos manipular datos de forma más intuitiva y específica.

**Seguridad:** Ofrece un mecanismo de seguridad poderoso y flexible debido a que pueden usarse para restringir el acceso a datos sensibles, es decir, los usuarios pueden ser autorizados para acceder solo a la vista y no a las tablas subyacentes.

**Personalización:** Las vistas permiten a los usuarios crear sus propias vistas de los datos según sus necesidades específicas sin afectar a la estructura original de la base de datos. [2]

Al ser la seguridad uno de los motivos por lo cual se utilizan las vistas nos es primordial explicar a qué hacemos referencia cuando hablamos de ella. Hacemos referencia al hecho de asegurar que los datos se mantengan a salvo de la corrupción y que el acceso a estos sea controlado.

**Seguridad:**

Medios para asegurar que los datos se mantengan a salvo de corrupción y que el acceso a ellos esté adecuadamente controlado. Proporcionar seguridad implica revelar exclusivamente información no sensible y rechazar cualquier solicitud que haga referencia a datos confidenciales. [3]

Por otro lado, continuando con vistas indexadas, para comprender su concepto, es necesario definir qué es un index. Los índices son estructuras de datos que se utilizan para mejorar el rendimiento de las consultas y acelerar la recuperación de información. Son creados en una o varias columnas de una tabla donde cada entrada al índice apunta a la ubicación física de los datos en la tabla subyacente. Tiene un funcionamiento similar al índice de un libro, reduciendo la necesidad de explorar todas las filas de una tabla. [4]

Continuando con nuestro marco conceptual, una vista indexada es una vista que tiene un índice asociado para acelerar el procesamiento de las consultas. Este índice se crea en la vista y no en las tablas base, ayudando al optimizador de consultas a encontrar los datos que necesita de forma eficiente cuando se consultan a través de la vista. [5] [6]

En SSMS (SQL Server Management Studio) una vista debe cumplir con ciertos requisitos para poder ser indexada, entre ellos:

- ❖ La vista debe estar vinculada con un esquema mediante la sentencia WITH SCHEMABINDING
- ❖ La definición de la vista indicada debe ser determinista: Una vista es determinista si todas las expresiones de la lista de selección y las cláusulas WHERE y GROUP BY son deterministas. Las expresiones deterministas siempre devuelven el mismo resultado cada vez que se evalúan con un conjunto específico de valores de entrada.
- ❖ Si GROUP BY está presente, la definición de vista debe contener COUNT\_BIG(\*).
- ❖ Si la definición de vista contiene una cláusula GROUP BY, la clave del índice clúster único solo puede hacer referencia a las columnas especificadas en la cláusula GROUP BY.
- ❖ No se acepta función SUM que hace referencia a una expresión que acepta valores NULL → esto se puede resolver de la siguiente manera: SUM(ISNULL(campo, valor por el cual reemplazar null)). [1]

Diferencias entre vistas y vistas indexadas:

Vistas	Vistas Indexadas
No tienen una estructura física en la base de datos. Son lógicas y se calculan en tiempo de ejecución.	Tienen una estructura física debido a la existencia de un índice. Los datos de la vista indexada se almacenan físicamente.
Son útiles para simplificar consultas, abstraer datos y proporcionar una capa de seguridad.	Se utilizan cuando se necesita mejorar el rendimiento de consultas específicas en vistas grandes o complejas.
Consumen menos espacio en disco, ya que no almacenan datos físicos.	Consumen más espacio debido al almacenamiento físico de datos y la presencia de índices.

### ¿Cuándo usar vistas o vistas indexadas?

Resulta útil la utilización de las vistas indexadas cuando se busca mejorar el rendimiento de consultas específicas en bases de datos que contienen vistas complejas o conjuntos de datos extensos, también en consultas que implican una unión (join).

Cuando hay consultas que se realizan con frecuencia y que involucran operaciones complejas en grandes conjuntos de datos, las vistas indexadas pueden acelerar significativamente el tiempo de respuesta. Por otro lado, si estas consultas son ocasionales o los conjuntos de datos son pequeños o medianos, donde la diferencia de rendimiento no sea significativa, se puede optar por la utilización de una vista estándar.

En situaciones donde las operaciones de búsqueda son críticas y deben realizarse de manera eficiente, las vistas indexadas pueden facilitar el acceso rápido a los datos necesarios. Se recomienda utilizar vistas indexadas teniendo en cuenta con qué regularidad los datos se actualizan ya que la creación de índices adicionales podría tener un impacto negativo en el

rendimiento de las actualizaciones y un aumento en la complejidad si estos se actualizan con frecuencia.

### CAPÍTULO III: METODOLOGÍA SEGUIDA

Iniciamos el proyecto charlando sobre el tema asignado, a medida que nos fuimos interiorizando en el tema comenzaron a surgir preguntas tales como: ¿Cómo se relacionan las vistas con la seguridad? ¿De qué forma limitan el acceso a datos sensibles? (Restricciones de integridad y visibilidad de datos), ¿Por qué surge la necesidad de limitar el acceso?, ¿Cómo se relacionan las “Vistas indexadas” con la optimización de las consultas? Luego siguiendo con una fase de investigación detallada sobre el mismo. Durante la conceptualización del problema, contextualizamos el tema y diagramamos los objetivos generales y específicos como parte del proceso de resolución.

**Dificultades encontradas:** Al ser uno de nuestros primeros acercamientos a la investigación, nos encontramos con algunos retos que superamos debatiendo y buscando información sobre cómo escribir un informe, citar correctamente, también cuáles son los pasos de la investigación y cómo encontrar fuentes confiables. Además de las dificultades mencionadas anteriormente, también tuvimos problemas al crear un índice sobre una vista ya que el lote de datos proporcionado, contaba con registros con valores duplicados.

**Herramientas:** La obtención de datos se llevó a cabo mediante una exploración de la bibliografía especializada en el ámbito de las bases de datos y vistas en particular. Se recurrió a fuentes académicas, libros relevantes y artículos científicos encontrados a través de plataformas como Google Scholar. Además, se consultó la documentación oficial de Microsoft SQL Server. Este enfoque permitió recopilar información clave sobre el manejo de vistas e índices en bases de datos relacionales, garantizando la calidad y fiabilidad de los datos utilizados en el desarrollo del trabajo.

### CAPÍTULO IV: DESARROLLO DEL TEMA / PRESENTACIÓN DE RESULTADOS

Las vistas pueden desempeñar un papel crucial en la limitación del acceso a datos sensibles en sistemas empresariales mediante la implementación de políticas de seguridad. A continuación, presentamos algunas estrategias:

#### Ocultar Columnas Sensibles:

```
CREATE VIEW dbo.PagoConsortio
WITH SCHEMABINDING
AS
SELECT A.apeynom, C.nombre, G.periodo, G.fechapago, TG.descripcion
FROM dbo.administrador A
JOIN dbo.consortio C ON A.idadmin = C.idadmin
JOIN dbo.gasto G ON C.idconsorcio = G.idconsorcio
JOIN dbo.tipogasto TG ON G.idtipogasto = TG.idtipogasto;
go
```

La creación de la vista PagoConsortio puede excluir columnas sensibles de tablas bases como la columna 'importe' de la tabla *gasto* ya que podría ser información personal o financiera confidencial. En la tabla *consorcio* la columna 'dirección' también podría

considerarse sensible si proporciona detalles específicos de la ubicación. La columna 'tel' contiene información de contacto, que muchas veces se considera sensible.

Resultados		Mensajes			
	apeynom	nombre	periodo	fechapago	descripcion
1	Perez Juan Manuel	EDIFICIO-111	6	2013-06-16 00:00:00.000	OTROS
2	SEGOVIA ALEJANDRO H.	EDIFICIO-2481	6	2013-06-16 00:00:00.000	OTROS
3	NAHMIAS DE K. NIDIA	EDIFICIO-3161	6	2013-06-16 00:00:00.000	OTROS
4	GOMEZ MATIAS GABRIEL	EDIFICIO-4211	6	2013-06-16 00:00:00.000	OTROS
5	CARDOZO MAXIMA	EDIFICIO-531	6	2013-06-16 00:00:00.000	OTROS
6	RATTI JUAN E.	EDIFICIO-6331	6	2013-06-16 00:00:00.000	OTROS
7	MARTINEZ EDUARDO	EDIFICIO-2011	6	2013-06-16 00:00:00.000	OTROS

Consulta ejecutada correctamente.

Las vistas también permiten que los usuarios solo vean los datos que son relevantes para su contexto.

### Filtrar Filas por Criterios de Seguridad:

```
-- Vista para Filtrar Datos por Zona Específica
CREATE VIEW VistaConsortiosPorZonaEspecifica AS
SELECT Z.descripcion AS nombre_zona, C.*
FROM Zona Z
JOIN Consorcio C ON Z.idzona = C.idzona
WHERE Z.idzona = '5';

SELECT * FROM VistaConsortiosPorZonaEspecifica;
```

Resultados		Mensajes							
	nombre_zona	idprovincia	idlocalidad	idconsorcio	nombre	direccion	idzona	idconserje	idadmin
3	OESTE	3	29	5	EDIFICIO-3295	MZ1 PC6 C26 B° SANTA BARBARA	5	92	9
4	OESTE	5	14	5	EDIFICIO-5145	ESPA&AMP;AMP;AMP;NTILDE;A N° 825	5	84	17
5	OESTE	8	31	6	EDIFICIO-8316	SAN MARTIN N° 1226, Dpto 9	5	73	28
6	OESTE	8	35	7	EDIFICIO-8357	CATAMARCA N° 1858	5	72	29
7	OESTE	8	42	10	EDIFICIO-84210	SAN LORENZO N° 1628, 3º piso, Dpto B	5	69	32
8	OESTE	11	26	4	EDIFICIO-11264	PASAJE DODERO N° 2440	5	59	42
9	OESTE	13	2	1	EDIFICIO-1331	ENTRE RIOS N° 3165, 1º y 2º piso	5	51	50

Consulta ejecutada correctamente. DESKTOP-H2D4K7C

Filtrar por una zona específica en una vista se puede utilizar como una medida de seguridad eficaz para restringir el acceso a datos sensibles. Esto garantiza que solo usuarios autorizados puedan ver información relacionada con esa zona, siguiendo el principio de "necesidad de saber", donde los usuarios solo pueden acceder y ver la información esencial para realizar sus funciones asignadas.

Creemos una vista donde la consulta nos devolverá datos del administrador del consorcio (idAdministrador, nombre\_administrador, nombre\_consorcio) y el monto total de los gastos realizados(gasto\_total) acompañado del número total de gastos realizados(row\_count). Esta vista es para entender cómo se crea un índice. Ya que es una vista con pocos registros.

```
CREATE VIEW vista_ejemplo_index WITH SCHEMABINDING AS
SELECT
    a.idadmin as idAdministrador,
    a.apeynom as nombre_administrador,
    c.nombre as nombre_consorcio,
    SUM(ISNULL(g.importe, 0)) as gasto_total,
    COUNT_BIG(*) as row_count
FROM
    dbo.administrador a
    JOIN dbo.consorcio c ON a.idadmin = c.idadmin
    JOIN dbo.gasto g ON c.idconsorcio = g.idconsorcio
GROUP BY a.idadmin, a.apeynom, c.nombre;
go
```

Se crea un índice cluster sobre la columna idAdministrador, nombre\_administrador, nombre\_consorcio.

```
CREATE UNIQUE CLUSTERED INDEX IDX_ejemplo_id
ON vista_ejemplo_index(idAdministrador, nombre_administrador, nombre_consorcio);
GO
```

Con la siguiente consulta se puede ver la estadística de la vista:

DBCC SHOW\_STATISTICS('dbo.vista\_ejemplo\_index', 'IDX\_ejemplo\_id');

## IMPLEMENTACIÓN:

### Tareas:

1. Crear un documento de acuerdo al modelo existente en el aula virtual.
2. Crear una vista sobre la tabla administrador que solo muestre los campos apeynom, sexo y fecha de nacimiento.
3. Realizar insert de un lote de datos sobre la vista recién creada. Verificar el resultado en la tabla administrador.
4. Realizar update sobre algunos de los registros creados y volver a verificar el resultado en la tabla.
5. Borrar todos los registros insertados a través de la vista.
6. Crear una vista que muestre los datos de las columnas de las siguientes tablas: (Administrador->Apeynom, consorcio->Nombre, gasto->periodo, gasto->fechaPago, tipoGasto->descripcion).
7. Crear un índice sobre la columna fechaPago sobre la vista recién creada.
8. Expresar las conclusiones en base a las pruebas realizadas.

Resultado de la consulta 'SELECT \* FROM administrador' antes del UPDATE:

	apeynom	sexo	fechnac
1	Perez Juan Manuel	M	1985-02-18 00:00:00.000
2	BASUALDO DELMIRA	F	1980-10-09 00:00:00.000
3	SEGOVIA ALEJANDRO H.	M	1974-06-02 00:00:00.000
4	ROMERO ELEUTERIO	M	1972-08-19 00:00:00.000
5	NAHMIAS DE K. NIDIA	F	1971-11-28 00:00:00.000
6	CORREA DE M. MARIA G.	F	1990-01-16 00:00:00.000
7	NAHMIAS JOSE	M	1974-09-02 00:00:00.000
8	NAHMIAS DE R. REBECA J.	F	1989-03-07 00:00:00.000
9	LOVATO CERENTINI ISABEL	F	1973-10-15 00:00:00.000
10	GOMEZ MATIAS GABRIEL	M	1974-03-20 00:00:00.000
11	CORREA HUGO E.	M	1993-08-11 00:00:00.000

3. Realizar insert de un lote de datos sobre la vista recién creada. Verificar el resultado en la tabla administrador.

```
INSERT INTO VistAdministrador (apeynom, sexo, fechnac) VALUES ('ROMAN GABRIEL ESTEBAN', 'M', '19981222');
```

```
INSERT INTO VistAdministrador (apeynom, sexo, fechnac) VALUES ('GARCETTE FERNANDO', 'M', '19971018');
```

```
SELECT * FROM VistAdministrador;
```

```
go
```

	apeynom	sexo	fechnac
166	LUGO DE R. RAMONA	F	1975-10-18 00:00:00.000
167	LOPEZ ELOYSA	F	1989-06-08 00:00:00.000
168	LOPEZ CONRADO	M	1983-10-26 00:00:00.000
169	TALAVERA CONSTANCIA	F	1986-01-02 00:00:00.000
170	VARGAS MIGUEL	M	1986-03-15 00:00:00.000
171	MACIEL PAULINA	F	1970-03-18 00:00:00.000
172	SAUCEDO FAUSTINA	F	1987-07-23 00:00:00.000
173	GONZALEZ LIDIA ESTER	F	1984-11-20 00:00:00.000
174	ARAUJO GUILLERMO JOSE	M	1985-10-13 00:00:00.000
175	ROMAN GABRIEL ESTEBAN	M	1998-12-22 00:00:00.000
176	GARCETTE FERNANDO	M	1997-10-18 00:00:00.000

Realizar un INSERT directamente en una vista puede ser aceptable en algunos casos, pero hay consideraciones importantes a tener en cuenta, especialmente si la vista está basada en una tabla que tiene más columnas que las especificadas en la vista. Aquí hay algunas consideraciones:

Cuando se hace un INSERT en la vista, las columnas adicionales de la tabla base quedarán con sus valores predeterminados o valores NULL si no tienen un valor predeterminado.

Restricciones y Valores Predeterminados.

En general, es preferible realizar operaciones de inserción directamente en la tabla base para evitar posibles ambigüedades y para garantizar que todas las restricciones y valores predeterminados se apliquen correctamente.

4.

```
--Realizar update sobre algunos de los registros creados y volver a verificar el resultado en la tabla.  
UPDATE VistAdministrador  
SET apeynom = 'ROMAN GABRIEL'  
WHERE apeynom = 'ROMAN GABRIEL ESTEBAN'  
go
```

175	ROMAN GABRIEL	M	1998-12-22 00:00:00.000
176	GARCETTE FERNANDO	M	1997-10-18 00:00:00.000

5.

```
delete from administrador  
where idadmin = '175'
```

6.

```
/*  
Crear una vista que muestre los datos de las columnas de las siguientes tablas:  
(Administrador->Apeynom, consorcio->Nombre, gasto->periodo, gasto->fechaPago,  
tipoGasto->descripcion) .  
*/
```

```
CREATE VIEW vista_adm_gasto_consortio WITH SCHEMABINDING AS  
SELECT  
    a.apeynom as nombre_admin,  
    c.nombre as nombre_consortio,  
    g.idgasto as idGasto,  
    g.periodo as periodo,  
    g.fechapago as fecha_pago,  
    tp.descripcion as tipo_gasto  
FROM  
    dbo.administrador a  
    JOIN dbo.consortio c ON a.idadmin = c.idadmin  
    JOIN dbo.gasto g ON c.idadmin = g.idconsorcio  
    JOIN dbo.tipogasto tp ON g.idtipogasto = tp.idtipogasto  
GO
```



Respuesta a la consulta `select * from vista_adm_gasto_consortio;`

	nombre_admin	nombre_consortio	idGasto	periodo	fecha_pago	tipo_gasto
1	Perez Juan Manuel	EDIFICIO-111	1	6	2013-06-16 00:00:00.000	OTROS
2	Perez Juan Manuel	EDIFICIO-111	2	3	2013-03-11 00:00:00.000	Sueldos
3	Perez Juan Manuel	EDIFICIO-111	3	7	2013-07-09 00:00:00.000	Sueldos
4	Perez Juan Manuel	EDIFICIO-111	4	7	2013-07-08 00:00:00.000	Sueldos
5	Perez Juan Manuel	EDIFICIO-111	5	8	2013-08-14 00:00:00.000	Limpieza
6	Perez Juan Manuel	EDIFICIO-111	6	1	2013-01-25 00:00:00.000	Aportes
7	Perez Juan Manuel	EDIFICIO-111	7	7	2013-07-26 00:00:00.000	Aportes
8	Perez Juan Manuel	EDIFICIO-111	8	8	2013-08-03 00:00:00.000	Sueldos
9	Perez Juan Manuel	EDIFICIO-111	9	8	2013-08-22 00:00:00.000	Sueldos
10	Perez Juan Manuel	EDIFICIO-111	10	3	2013-03-26 00:00:00.000	Sueldos
11	BASUALDO DELMIRA	EDIFICIO-122	11	3	2013-03-28 00:00:00.000	Limpieza

Consulta ejecutada correctamente.

7.

`CREATE UNIQUE CLUSTERED INDEX UX_aplicacion ON vista_adm_gasto_consortio(nombre_admin, nombre_consortio, idGasto);`

`CREATE NONCLUSTERED INDEX IDX_fechapago ON vista_adm_gasto_consortio(fecha_pago);`

Respuesta a la consulta `DBCC SHOW_STATISTICS('dbo.vista_adm_gasto_consortio', 'IDX_fechapago');`

100 %

Resultados

Mensajes

	Name	Updated	Rows	Rows Sampled	Steps	Density	Average key length	String Index	Filter Expression	Unfiltered Rows
1	IDX_fecha_pago	Nov 1 2023 10:47AM	8000	8000	195	0.1842185	42.22	NO	NULL	8000

	All density	Average Length	Columns
1	0.0007097232	8	fecha_pago
2	0.0001976675	25.68	fecha_pago, nombre_admin
3	0.0001976675	38.22	fecha_pago, nombre_admin, nombre_consortio
4	0.000125	42.22	fecha_pago, nombre_admin, nombre_consortio, idGa...

	RANGE_HI_KEY	RANGE_ROWS	EQ_ROWS	DISTINCT_RANGE_ROWS	AVG_RANGE_ROWS
1	2013-01-01 00:00:00.000	0	3	0	1
2	2013-01-12 00:00:00.000	35	6	10	3.5
3	2013-01-24 00:00:00.000	41	8	10	4.1
4	2013-02-01 00:00:00.000	20	6	10	2.0

Consulta ejecutada correctamente.

DESKTOP-H2D4K7C\SQLEXPRES

El índice `IDX_fechapago` puede ser útil para optimizar el rendimiento de la vista. Puede acelerar las consultas que se basan en la fecha de pago, el nombre del administrador o el nombre del consorcio. Por ejemplo, una consulta que devuelve todas las filas de la vista `dbo.vista_aplicacion` donde la fecha de pago es mayor o igual al 2023-01-24 00:00:00.000.

## **CAPÍTULO V: CONCLUSIONES**

Concluimos con que las vistas son una herramienta poderosa en bases de datos empresariales para implementar estrategias de seguridad. Pueden limitar el acceso a datos sensibles al ocultar columnas delicadas. Además, el uso de vistas indexadas mejora el rendimiento en consultas sobre datos complejos. En resumen, este trabajo práctico destaca la eficacia de las vistas para fortalecer la seguridad en bases de datos empresariales.

Pero también es necesario mencionar que no son suficientes. Las organizaciones deben implementar otras medidas de seguridad, como la autenticación y la autorización, para proteger sus datos sensibles. Lo cual nos gustaría profundizar en investigaciones futuras.

# Bibliografía

- [1] stevestein, «MicroSoft,» 13 Junio 2017. [En línea]. Available:  
<https://learn.microsoft.com/es-es/previous-versions/sql/2014/relational-databases/views/views?view=sql-server-2014>.
- [2] T. C. y. C. O. Begg, Database Systems: A Practical Approach to Design, Implementation, and Management, Pearson, 2015.
- [3] R. E. y. S. B. Navathe, Fundamentals of Database Systems, Pearson, 2016.
- [4] R. E. y. S. B. Navathe, Fundamentals of Database Systems, Pearson, 2016.
- [5] G.-M. y. U. Widom, Database Systems The Complete Book, Pearson, 2009.
- [6] R. R. y. J. Gehrke, DATABASE MANAGEMENT SYSTEMS, McGraw-Hill Education, 2003.