Manual de Consultas Útiles para la Base de Datos en Azure SQL

Este manual está diseñado para ayudarte a entender y practicar las consultas SQL utilizando una base de datos de ejemplo que gestiona asignaturas, alumnos, notas y labores extracurriculares. A través de ejemplos didácticos, aprenderás a manejar y extraer información relevante de la base de datos.

Estructura de la Base de Datos

Antes de comenzar con las consultas, es importante familiarizarse con la estructura de las tablas:

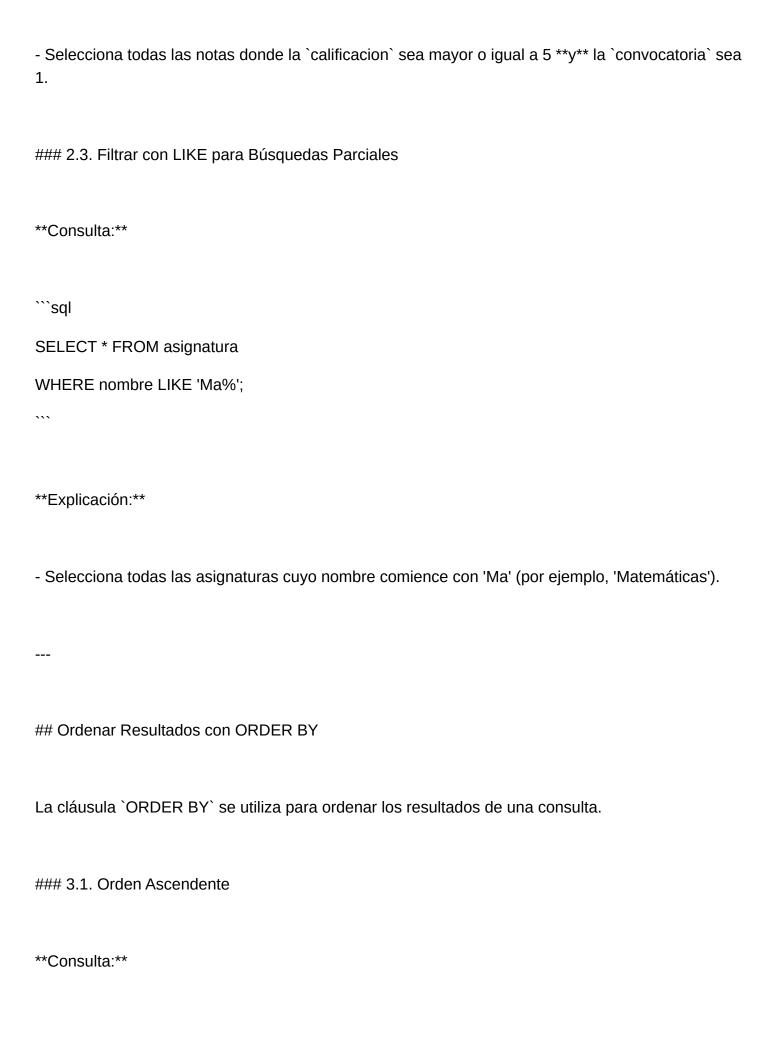
- **asignatura**: Contiene información sobre las asignaturas.
 - `id`: Identificador único de la asignatura.
 - `nombre`: Nombre de la asignatura.
- **alumno**: Contiene información sobre los alumnos.
- `id`: Identificador único del alumno.
- `nombre`: Nombre del alumno.
- `apellido`: Apellido del alumno.
- `fecha_nacimiento`: Fecha de nacimiento del alumno.
- **nota**: Registra las calificaciones de los alumnos en las asignaturas.
- `id`: Identificador único de la nota.
- `asignatura id`: Referencia a la asignatura.
- `calificacion`: Calificación obtenida.
- `fecha_examen`: Fecha del examen.
- `convocatoria`: Número de convocatoria.
- `alumno_id`: Referencia al alumno.

- **labor_extra**: Registra las labores extracurriculares de los alumnos.
- `puesto`: Nombre del puesto o labor.
- `alumno_id`: Referencia al alumno.
Contenido del Manual
1. [Consultas Básicas con SELECT](#consultas-básicas-con-select)
2. [Filtrado de Datos con WHERE](#filtrado-de-datos-con-where)
3. [Ordenar Resultados con ORDER BY](#ordenar-resultados-con-order-by)
4. [Funciones de Agregación](#funciones-de-agregación)
5. [Uniones de Tablas con JOIN](#uniones-de-tablas-con-join)
6. [Subconsultas](#subconsultas)
7. [Inserción, Actualización y Eliminación de Datos](#inserción-actualización-y-eliminación-de-datos)
8. [Consultas Avanzadas](#consultas-avanzadas)
9. [Ejercicios Prácticos](#ejercicios-prácticos)
Consultas Básicas con SELECT
La sentencia `SELECT` se utiliza para seleccionar datos de una tabla.
1.1. Seleccionar Todos los Datos de una Tabla
Consulta:

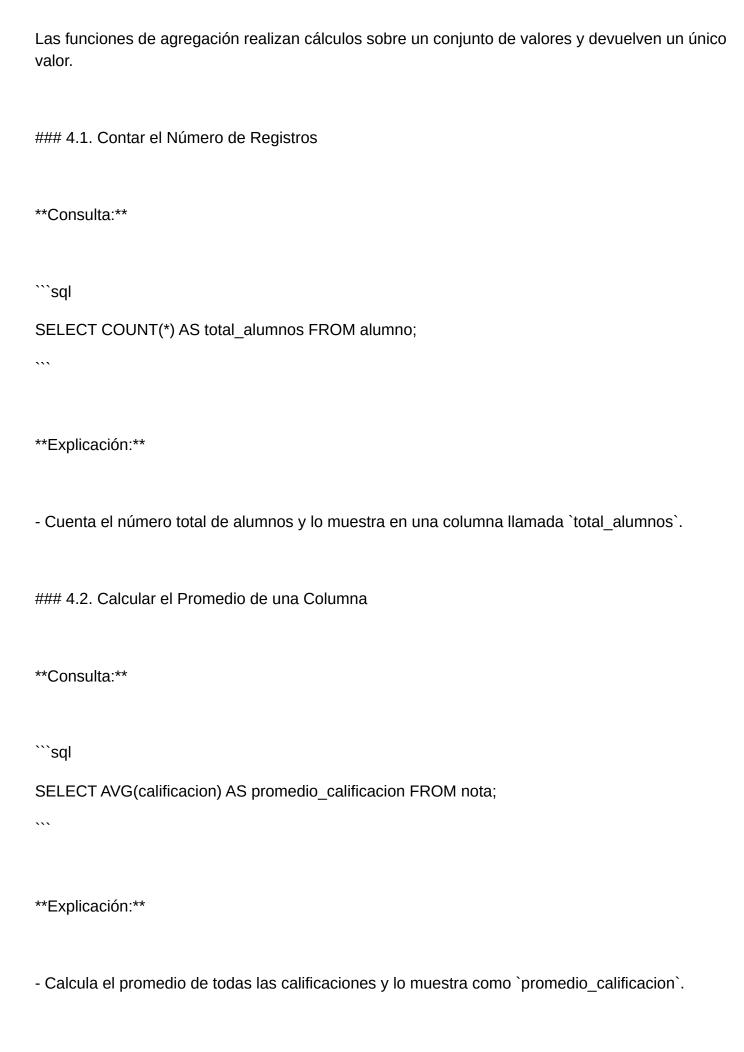
```
```sql
SELECT * FROM alumno;
Explicación:
- `SELECT *`: Selecciona todas las columnas.
- `FROM alumno`: Indica que la selección es de la tabla `alumno`.
Resultado:
Muestra todas las filas y columnas de la tabla `alumno`.
1.2. Seleccionar Columnas Específicas
Consulta:
```sql
SELECT nombre, apellido FROM alumno;
**Explicación:**
- Selecciona únicamente las columnas `nombre` y `apellido` de la tabla `alumno`.
```

Filtrado de Datos con WHERE La cláusula `WHERE` se utiliza para filtrar los registros que cumplen una condición específica. ### 2.1. Filtrar por una Condición Simple **Consulta:** ```sql SELECT * FROM alumno WHERE nombre = 'Juan'; **Explicación:** - Selecciona todas las columnas de `alumno` donde el `nombre` sea exactamente 'Juan'. ### 2.2. Filtrar con Operadores Lógicos **Consulta:** ```sql SELECT * FROM nota WHERE calificacion >= 5 AND convocatoria = 1;

Explicación:



```
```sql
SELECT * FROM alumno
ORDER BY apellido ASC;
Explicación:
- Ordena los alumnos por el apellido en orden ascendente (A-Z).
3.2. Orden Descendente
Consulta:
```sql
SELECT * FROM nota
ORDER BY calificacion DESC;
**Explicación:**
- Ordena las notas por la calificación en orden descendente (de mayor a menor).
## Funciones de Agregación
```



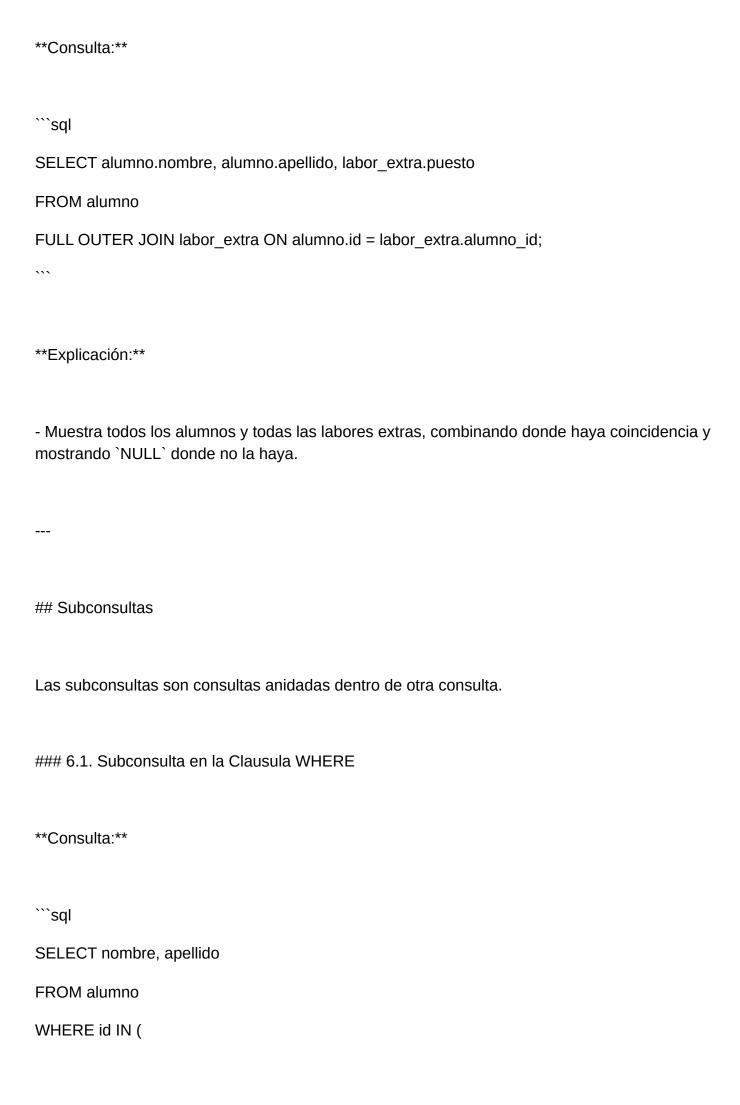
```
**Consulta:**
```sql
SELECT MIN(calificacion) AS menor_calificacion,
 MAX(calificacion) AS mayor_calificacion
FROM nota;
Explicación:
- Obtiene la calificación mínima y máxima de la tabla `nota`.
4.4. Agrupar Datos con GROUP BY
Consulta:
```sql
SELECT asignatura_id, AVG(calificacion) AS promedio_por_asignatura
FROM nota
GROUP BY asignatura_id;
**Explicación:**
```

- Agrupa las notas por `asignatura_id` y calcula el promedio de calificaciones para cada grupo.

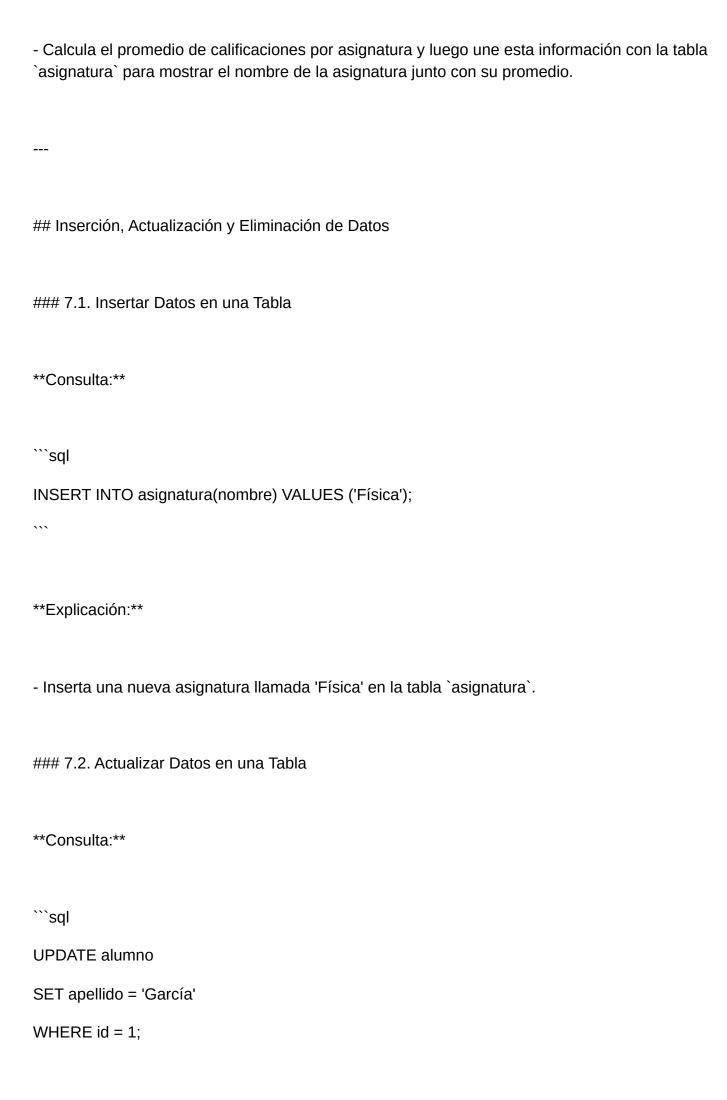
4.3. Obtener Mínimo y Máximo

Uniones de Tablas con JOIN
Las uniones (`JOIN`) permiten combinar filas de dos o más tablas basadas en una relación entre ellas.
5.1. INNER JOIN
Consulta:
```sql
SELECT alumno.nombre, alumno.apellido, asignatura.nombre AS asignatura, nota.calificacion
FROM nota
INNER JOIN alumno ON nota.alumno_id = alumno.id
INNER JOIN asignatura ON nota.asignatura_id = asignatura.id;
**Explicación:**
- Combina las tablas `nota`, `alumno` y `asignatura` para mostrar el nombre y apellido del alumno, el nombre de la asignatura y la calificación obtenida.
### 5.2. LEFT JOIN
**Consulta:**

sql
SELECT alumno.nombre, alumno.apellido, labor_extra.puesto
FROM alumno
LEFT JOIN labor_extra ON alumno.id = labor_extra.alumno_id;
**Explicación:**
- Muestra todos los alumnos y, si tienen una labor extra, muestra el puesto. Si no tienen, mostrará `NULL` en la columna `puesto`.
### 5.3. RIGHT JOIN
**Consulta:**
```sql
SELECT asignatura.nombre AS asignatura, nota.calificacion
FROM asignatura
RIGHT JOIN nota ON asignatura.id = nota.asignatura_id;
Explicación:
- Muestra todas las notas y, si la asignatura existe, muestra su nombre. Si no hay asignatura correspondiente, mostrará `NULL`.
5.4. FULL OUTER JOIN



```
SELECT alumno id
  FROM nota
  WHERE calificacion < 5
);
**Explicación:**
- Selecciona los nombres y apellidos de los alumnos que tienen al menos una calificación inferior a 5.
### 6.2. Subconsulta en la Clausula FROM
**Consulta:**
```sql
SELECT promedio_por_asignatura.asignatura_id, asignatura.nombre,
promedio_por_asignatura.promedio_calificacion
FROM (
 SELECT asignatura_id, AVG(calificacion) AS promedio_calificacion
 FROM nota
 GROUP BY asignatura_id
) AS promedio_por_asignatura
INNER JOIN asignatura ON promedio_por_asignatura.asignatura_id = asignatura.id;
Explicación:
```



**Explicación:**
- Actualiza el apellido del alumno con `id` 1 a 'García'.
### 7.3. Eliminar Datos de una Tabla
**Consulta:**
```sql
DELETE FROM nota
WHERE calificacion < 5;
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
Explicación:
- Elimina todas las notas donde la calificación sea inferior a 5.
Consultas Avanzadas
8.1. Uso de Funciones de Ventana
Consulta:

• • • •

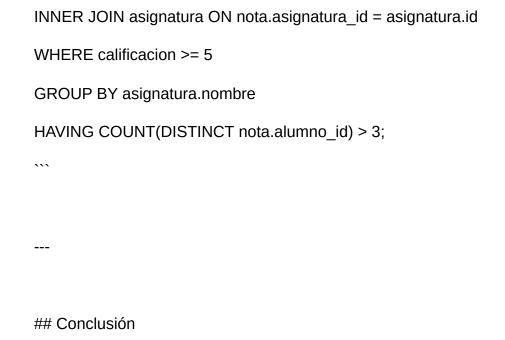
```
```sql
SELECT alumno.nombre, alumno.apellido, nota.calificacion,
 AVG(nota.calificacion) OVER (PARTITION BY nota.asignatura id) AS promedio asignatura
FROM nota
INNER JOIN alumno ON nota.alumno id = alumno.id;
Explicación:
- Para cada nota, calcula el promedio de calificaciones de la asignatura correspondiente sin agrupar
los resultados, permitiendo ver cómo se compara cada calificación individual con el promedio de su
asignatura.
8.2. Pivotar Datos
Consulta:
```sql
SELECT alumno.nombre, alumno.apellido,
    MAX(CASE WHEN asignatura.nombre = 'Matemáticas' THEN nota.calificacion END) AS
Matemáticas,
    MAX(CASE WHEN asignatura.nombre = 'Lengua' THEN nota.calificacion END) AS Lengua
FROM nota
INNER JOIN alumno ON nota.alumno_id = alumno.id
INNER JOIN asignatura ON nota.asignatura id = asignatura.id
GROUP BY alumno.nombre, alumno.apellido;
```

Explicación:
- Transforma las filas de calificaciones en columnas separadas para cada asignatura, facilitando la comparación directa de calificaciones por asignatura para cada alumno.

Ejercicios Prácticos
Para afianzar lo aprendido, realiza los siguientes ejercicios:
Ejercicio 1: Listar Alumnos con Más de una Nota
Objetivo:
Mostrar los alumnos que tienen más de una calificación registrada.
Consulta:
```sql
SELECT alumno.nombre, alumno.apellido, COUNT(nota.id) AS total_notas
FROM alumno
INNER JOIN nota ON alumno.id = nota.alumno_id
GROUP BY alumno.nombre, alumno.apellido
HAVING COUNT(nota.id) > 1;

```
Ejercicio 2: Obtener la Edad de los Alumnos
Objetivo:
Mostrar el nombre, apellido y edad actual de cada alumno.
Consulta:
```sql
SELECT nombre, apellido, DATEDIFF(YEAR, fecha nacimiento, GETDATE()) AS edad
FROM alumno;
### Ejercicio 3: Encontrar Alumnos sin Labor Extra
**Objetivo:**
Listar los alumnos que no tienen ninguna labor extracurricular.
**Consulta:**
```sql
SELECT nombre, apellido
FROM alumno
WHERE id NOT IN (
 SELECT alumno_id FROM labor_extra
);
```

```
Ejercicio 4: Calificaciones por Convocatoria
Objetivo:
Mostrar el promedio de calificaciones por convocatoria.
Consulta:
```sql
SELECT convocatoria, AVG(calificacion) AS promedio_calificacion
FROM nota
GROUP BY convocatoria;
### Ejercicio 5: Asignaturas con Más de 3 Alumnos Aprobaron
**Objetivo:**
Listar las asignaturas donde más de 3 alumnos han obtenido una calificación de 5 o más.
**Consulta:**
```sql
SELECT asignatura.nombre, COUNT(DISTINCT nota.alumno_id) AS alumnos_aprobados
FROM nota
```



Este manual ha cubierto una variedad de consultas SQL esenciales para interactuar y extraer información de una base de datos en Azure SQL. Practica estos ejemplos y experimenta con tus propias consultas para fortalecer tu comprensión y habilidades en SQL. Recuerda siempre verificar los resultados y comprender la lógica detrás de cada consulta para aplicarla eficazmente en diferentes escenarios.