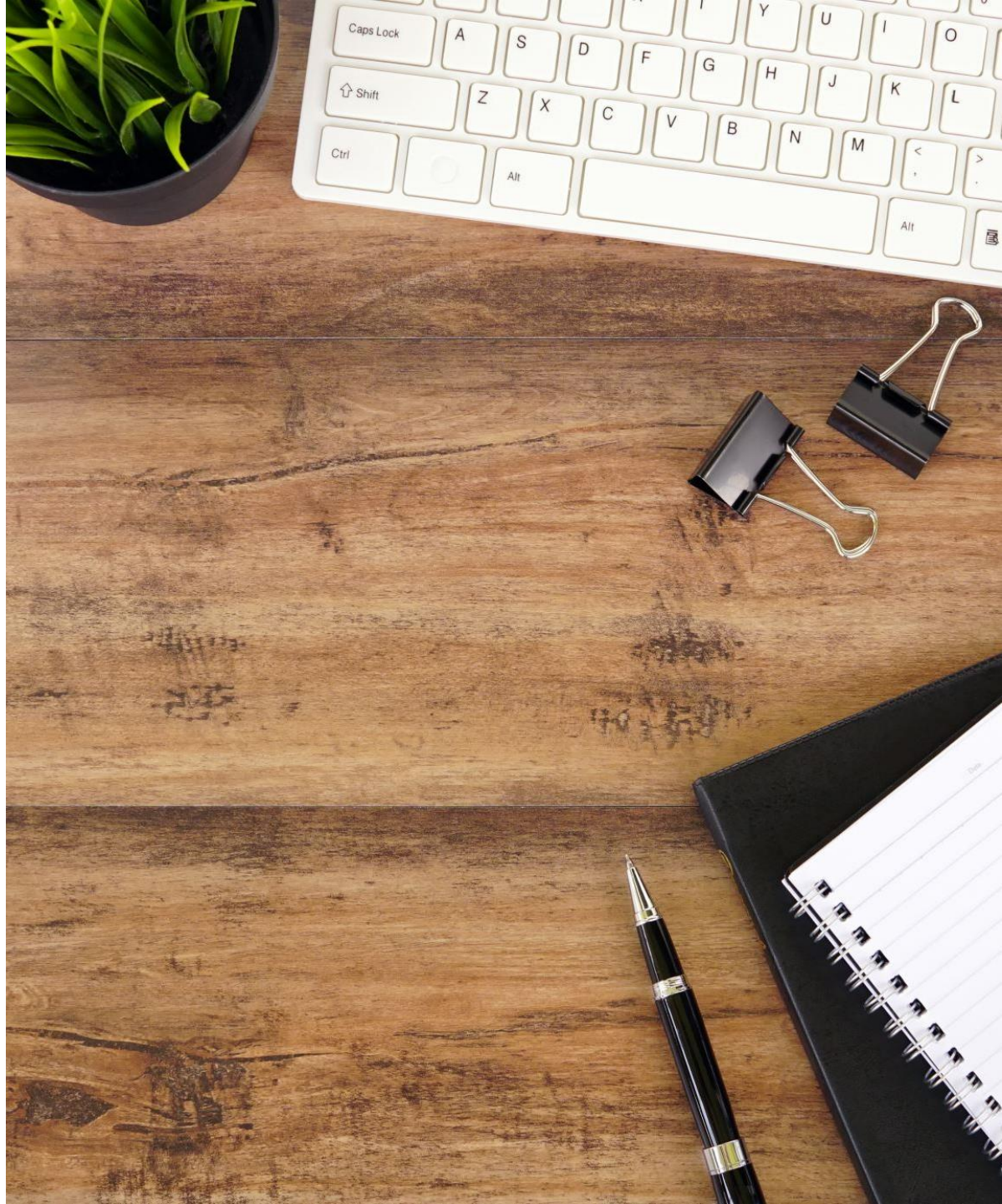
PRESENTACIÓN SEGURIDAD

Jorge Parra Hidalgo

ITIT 13104

Bases de datos II – Raul Toledo



USER ACCOUNTS

CREATE USER Y GRANT

```
CREATE USER ap_admin@localhost IDENTIFIED BY 'pa55word';
CREATE USER ap_user@localhost IDENTIFIED BY 'pa55word';

GRANT ALL
ON ap.*
TO ap_admin@localhost;

GRANT SELECT, INSERT, UPDATE, DELETE
ON ap.*
TO ap_user@localhost;
```

USER ACCOUNTS

CREATE USER Y GRANT

```
CREATE USER ap_admin@localhost IDENTIFIED BY 'pa55word';
CREATE USER ap_user@localhost IDENTIFIED BY 'pa55word';

GRANT ALL
ON ap.*
TO ap_admin@localhost;

GRANT SELECT, INSERT, UPDATE, DELETE
ON ap.*
TO ap_user@localhost;
```

THE GRANT TABLES IN THE MYSQL DATABASE

The four privilege levels

Level	Description
Global	All databases and all tables.
Database	All tables in the specified database.
Table	All columns in the specified table.
Column	Only the specified column or columns.

EJEMPLOS

- Global: GRANT ALL PRIVILEGES ON *.* TO 'usuario_global'@'localhost’;
- Database: GRANT SELECT, INSERT ON nacimientos_db.* TO 'usuario_bd'@'localhost’;
- Table: GRANT SELECT ON nacimientos_db.nacimientos_2022 TO 'usuario_tabla'@'localhost’;
- Column: GRANT SELECT (nombre_madre, fecha_nacimiento) ON nacimientos_db.nacimientos_2022 TO 'usuario_columna'@'localhost';

HOW TO CREATE, RENAME, AND DROP USERS

Acción	Comando clave	Detalles importantes
Crear usuario	`CREATE USER`	No otorga privilegios por defecto. Usa `GRANT` después.
Renombrar	`RENAME USER`	Mantiene privilegios existentes.
Eliminar	`DROP USER`	Borra usuario y sus privilegios. Usa `IF EXISTS` si es parte de un script.

EJEMPLOS

```
CREATE USER joel@localhost IDENTIFIED BY 'sesame';
```

```
RENAME USER usuario_antiguo@host TO usuario_nuevo@host;
```

```
DROP USER IF EXISTS jane;
```

HOW TO SPECIFY USER ACCOUNT NAMES

✅ **Reglas clave:**

- Si **no se especifica el host**, se asume `@'%'` (es decir, **desde cualquier host**).
- Usa **comillas** (simples `'`, dobles `"` o backticks `` ` ``) si el **usuario o host tiene caracteres especiales**, como guiones (`-`) o comodines (`%`).

EJEMPLOS

Ejemplo	Descripción
`john@localhost`	Usuario `john` puede conectarse solo desde el mismo servidor (`localhost`)
`'john'@'localhost'`	Igual que el anterior, con comillas (opcionales aquí)
`john@127.0.0.1`	Igual que `localhost` , pero usando la IP
`john`	Usuario puede conectarse desde cualquier host (`john@'%'`)
`john@'%'`	Forma explícita del anterior; `%` es comodín para cualquier host
`john@'%.murach.com'`	Solo puede conectarse desde hosts del dominio `murach.com`
`'quinn-the-mighty'@'%.murach.com'`	Usuario con guiones; requiere comillas en usuario y host

HOW TO GRANT PRIVILEGES

📌 **Reglas generales:**

- A partir de **MySQL 8.0**, **ya no se pueden crear usuarios** con `GRANT`; deben existir previamente (creados con `CREATE USER`).
- La cláusula `ON` determina el **nivel del privilegio**:
  - `*.*` → Global (todos los esquemas y tablas)
  - `base_datos.*` → Base de datos
  - `base_datos.tabla` → Tabla
  - `base_datos.tabla (columna1, ...)` → Columna
- `WITH GRANT OPTION` permite al usuario **otorgar sus propios privilegios** a otros.

EJEMPLOS

- GRANT privilegiosON [base_datos.]tablaTO usuario[WITH GRANT OPTION];

```
GRANT SELECT, INSERT, UPDATE ON ap.* TO joel@localhost;
```
Privilegios de base de datos (`ap`)

```
GRANT SELECT, INSERT, UPDATE ON ap.vendors TO joel@localhost;
```
Privilegios sobre la tabla `vendors`

HOW TO VIEW PRIVILEGES

```
SHOW GRANTS FOR 'usuario'@'host';
```

EJEMPLOS

Consulta	Descripción
`SHOW GRANTS FOR 'jim';`	Muestra los privilegios del usuario `jim` desde cualquier host (`%`)
`SHOW GRANTS FOR 'ap_user'@'localhost';`	Muestra los privilegios del usuario `ap_user` solo desde `localhost`
`SHOW GRANTS;`	Muestra los privilegios del **usuario actual conectado**

HOW TO REVOKE PRIVILEGES

```
REVOKE ALL, GRANT OPTION FROM 'usuario'@'host';
```

```
REVOKE ALL, GRANT OPTION FROM 'ap_user', 'anne'@'localhost';
```

```
REVOKE INSERT, UPDATE
ON base_datos.tabla
FROM 'usuario'@'host';
```

PUNTOS IMPORTANTES

Tema	Detalle
🔒 Permisos necesarios	Para revocar **todos** los privilegios necesitas tener el privilegio global `CREATE USER`.
🔄 Revocar específicos	Para revocar **privilegios específicos**, necesitas el privilegio `GRANT OPTION` y además **tener los privilegios que estás revocando.**
❌ No elimina al usuario	El comando `REVOKE` **no elimina** la cuenta del usuario, solo sus permisos. Para eliminar al usuario, usa `DROP USER`.

HOW TO CHANGE PASSWORD

✅ **Opción recomendada:** `ALTER USER`

El comando `ALTER USER` es **la forma más moderna y preferida** para cambiar contraseñas. Está disponible desde **MySQL 5.6** y permite establecer políticas de seguridad adicionales.

```
ALTER USER 'usuario'@'host' IDENTIFIED BY 'nueva_contraseña';
```

EXTRA

```
ALTER USER USER() IDENTIFIED BY 'nueva_contraseña';
```

```
ALTER USER IF EXISTS 'usuario' PASSWORD EXPIRE INTERVAL 90 DAY;
```

```
PASSWORD HISTORY 5
```

```
PASSWORD REUSE INTERVAL 180 DAY
```

```
SET PASSWORD FOR 'usuario'@'host' = 'nueva_contraseña';
```

BUENAS PRACTICAS

✅ Buenas prácticas

Recomendación	Por qué
Usa `ALTER USER` en lugar de `SET PASSWORD`	Es más moderno, compatible con otras bases de datos, y permite políticas avanzadas
No uses la función `PASSWORD()`	Fue eliminada en MySQL 8.0
Verifica que todos los usuarios tengan contraseñas	Mejora la seguridad general del servidor
Usa `IF EXISTS` para evitar errores	Evita fallos si el usuario no existe

A SCRIPT THAT CREATES USERS

A script that sets up the users and privileges for a database

```
-- drop the users (remove IF EXISTS for MySQL 5.6 and earlier)
DROP USER IF EXISTS john;
DROP USER IF EXISTS jane;
DROP USER IF EXISTS jim;
DROP USER IF EXISTS joel@localhost;

-- create the users
CREATE USER john IDENTIFIED BY 'sesame';
CREATE USER jane IDENTIFIED BY 'sesame';
CREATE USER jim IDENTIFIED BY 'sesame';
CREATE USER joel@localhost IDENTIFIED BY 'sesame';

-- grant privileges to a developer (joel)
GRANT ALL ON *.* TO joel@localhost WITH GRANT OPTION;

-- grant privileges to the ap manager (jim)
GRANT SELECT, INSERT, UPDATE, DELETE ON ap.* TO jim WITH GRANT OPTION;

-- grant privileges to ap users (john, jane)
GRANT SELECT, INSERT, UPDATE, DELETE ON ap.vendors TO john, jane;
GRANT SELECT, INSERT, UPDATE, DELETE ON ap.invoices TO john, jane;
GRANT SELECT, INSERT, UPDATE, DELETE ON ap.invoice_line_items TO john, jane;
GRANT SELECT ON ap.general_ledger_accounts TO john, jane;
GRANT SELECT ON ap.terms TO john, jane;
```

HOW TO CREATE, MANAGE, AND DROP ROLES

Un rol en una base de datos (como MySQL) es un conjunto de privilegios que puedes asignar a uno o más usuarios. Es una forma de gestionar permisos de manera más sencilla, organizada y reutilizable.

EJEMPLO PRACTICO

1. Creas un rol llamado `facturacion`.
2. Le das permisos de `INSERT`, `UPDATE`, `SELECT` sobre las tablas `facturas` y `clientes`.
3. Asignas ese rol a los 10 usuarios.

Así:

- Si un día necesitas dar más permisos, solo modificas el rol.
- Si alguien deja el equipo, simplemente le quitas el rol.

EJEMPLO COMANDO

```
CREATE ROLE IF NOT EXISTS invoice_entry;
```

```
GRANT INSERT, UPDATE ON ap.invoices TO invoice_entry;
GRANT INSERT, UPDATE ON ap.invoice_line_items TO invoice_entry;
```

```
GRANT invoice_entry TO john, jane;
```

```
-- Ver roles activos en la sesión
SELECT CURRENT_ROLE();

-- Ver privilegios asignados a un rol
SHOW GRANTS FOR invoice_entry;
```

REVOKE/DROP PRIVILEGES AND ROLES

```
REVOKE UPDATE ON ap.invoice_line_items FROM invoice_entry;
```

```
REVOKE invoice_entry FROM john;
```

```
DROP ROLE IF EXISTS invoice_entry;
```