
Mòdul 0377 – Administració de sistemes gestors de bases de dades

Índex

1. Instal·lació i configuració d'un SGBD	1
1.1. Funcions del SGBD. Components. Tipus	1
Arquitectura ANSI/SPARC (tres nivells)	1
Tipus de SGBD	1
1.2. Instal·lació de MariaDB a Debian/Ubuntu	2
Verificar el funcionament	2
1.3. Fitxers de configuració principals	2
Paràmetres rellevants	3
1.4. El diccionari de dades	4
2. Accés a la informació: vistes, usuaris i permisos	5
2.1. Vistes	5
Vistes actualitzables	6
Sinònims	6
2.2. Gestió d'usuaris	6
Crear usuaris	6
Veure usuaris existents	7
2.3. Privilegis	7
Nivells de privilegis	7
Assignar privilegis (GRANT)	7
Revisar privilegis	8
Revocar privilegis (REVOKE)	8
2.4. Rols	8
2.5. Normativa legal: protecció de dades	9
3. Automatització de tasques: guions d'administració	9
3.1. Procediments emmagatzemats	9
3.2. Funcions emmagatzemades	10
3.3. Disparadors (triggers)	11
3.4. Esdeveniments planificats	12
3.5. Excepcions i gestió d'errors	13
4. Optimització del rendiment: monitoratge i índexs	14
4.1. Eines de monitoratge	14
Monitoratge des de la consola SQL	14
Eines externes	14
4.2. Índexs	15
Tipus d'índexs a MariaDB	15
Crear i gestionar índexs	15
4.3. Anàlisi de consultes: EXPLAIN	16
4.4. Optimització de l'estructura	16
4.5. Alertes de rendiment	17
4.6. Optimitzar el sistema operatiu per a MariaDB	18
5. Bases de dades distribuïdes i replicació	18
5.1. Bases de dades distribuïdes	18
Teorema CDP	18
5.2. Tècniques de fragmentació	19
Fragmentació horitzontal (sharding)	19

Fragmentació vertical	19
Fragmentació híbrida (mixta)	19
5.3. Replicació a MariaDB	19
Tipus de replicació	19
Formats de replicació	20
5.4. Configuració de replicació mestre-esclau	20
Configuració del servidor mestre	20
Configuració del servidor esclau	21
5.5. Replicació en cadena	22
5.6. Galera Cluster (replicació síncrona)	22
Instal·lació (Debian/Ubuntu)	22
Configuració (node 1)	22
5.7. Comprovació de fallades en sistemes distribuïts	23

Cicle formatiu: Administració de Sistemes Informàtics en Xarxa (ASIX)

Hores: 33 h

SGBD de referència: MariaDB

1. Instal·lació i configuració d'un SGBD

1.1. Funcions del SGBD. Components. Tipus

Un **SGBD** (Sistema Gestor de Bases de Dades) és el programari que gestiona l'emmagatzematge, la recuperació, la seguretat i la integritat de les dades. Les seves funcions principals:

- **Definició de dades** (DDL): CREATE, ALTER, DROP.
- **Manipulació de dades** (DML): SELECT, INSERT, UPDATE, DELETE.
- **Control d'accés** (DCL): GRANT, REVOKE.
- **Control de transaccions** (TCL): COMMIT, ROLLBACK.
- **Recuperació de fallades:** registres (*logs*), còpies de seguretat.
- **Gestió de la concurrència:** bloquejos, nivells d'aïllament.

Arquitectura ANSI/SPARC (tres nivells)

Nivell	Descripció	Qui l'utilitza
Extern (vistes)	Com veu les dades cada usuari o aplicació	Usuaris finals, aplicacions
Conceptual (lògic)	Estructura lògica global de la BD	Administrador de BD
Intern (físic)	Com s'emmagatzemen físicament les dades	SGBD, SO

Gràcies a la **independència de dades**, els canvis en un nivell no afecten els nivells superiors.

Tipus de SGBD

Categoria	Exemples
Relacionals lliures	MariaDB, PostgreSQL, SQLite
Relacionals propietaris	Oracle Database, Microsoft SQL Server, IBM DB2
NoSQL	MongoDB (documents), Redis (clau-valor), Cassandra (columnar), Neo4j (grafs)
NewSQL	CockroachDB, TiDB

1.2. Instal·lació de MariaDB a Debian/Ubuntu

```
# Actualitzar el sistema
sudo apt update && sudo apt upgrade -y

# Instal·lar MariaDB Server
sudo apt install mariadb-server mariadb-client -y

# Verificar l'estat del servei
sudo systemctl status mariadb

# Habilitar l'inici automàtic
sudo systemctl enable mariadb

# Executar l'assistent de seguretat inicial (imprescindible en
↳ producció)
sudo mysql_secure_installation
```

L'assistent `mysql_secure_installation` permet:

- Establir/canviar la contrasenya de root.
- Eliminar usuaris anònims.
- Desactivar l'accés remot de root.
- Eliminar la base de dades de test.

Verificar el funcionament

```
# Connectar-se al servidor
sudo mysql -u root -p

# Des de la consola MariaDB
MariaDB [(none)]> SELECT VERSION();
MariaDB [(none)]> SHOW DATABASES;
MariaDB [(none)]> STATUS;
```

1.3. Fitxers de configuració principals

MariaDB a Debian/Ubuntu llegeix la configuració en ordre:

1. `/etc/mysql/mariadb.cnf` (fitxer principal, inclou els altres)
2. `/etc/mysql/conf.d/*.cnf`
3. `/etc/mysql/mariadb.conf.d/*.cnf` ← **principal en Debian/Ubuntu**
4. `~/.my.cnf` (configuració per usuari)

El fitxer clau és `/etc/mysql/mariadb.conf.d/50-server.cnf`.

Paràmetres rellevants

```
[mysqld]

# --- Identificació ---
user          = mysql
pid-file      = /run/mysqld/mysqld.pid

# --- Xarxa ---
bind-address  = 127.0.0.1      # Canviar a 0.0.0.0 per a accés extern
port          = 3306
socket        = /run/mysqld/mysqld.sock

# --- Directoris ---
datadir       = /var/lib/mysql
tmpdir        = /tmp

# --- Joc de caràcters ---
character-set-server = utf8mb4
collation-server   = utf8mb4_unicode_ci

# --- Connexions ---
max_connections    = 100
connect_timeout    = 10
wait_timeout       = 600
interactive_timeout = 600

# --- Motor per defecte ---
default_storage_engine = InnoDB

# --- Mida de la caché de consultes ---
query_cache_type      = 1
query_cache_size       = 16M

# --- InnoDB ---
innodb_buffer_pool_size = 256M  # ~70% de la RAM disponible per a BD
innodb_log_file_size    = 64M

# --- Logs ---
log_error             = /var/log/mysql/error.log
slow_query_log        = 1
slow_query_log_file    = /var/log/mysql/slow.log
long_query_time       = 2
log_bin               = /var/log/mysql/mariadb-bin
```

Després de modificar el fitxer, reiniciar el servei:

```
sudo systemctl restart mariadb

# Comprovar que no hi ha errors
sudo journalctl -u mariadb -n 50
```

1.4. El diccionari de dades

MariaDB emmagatzema les **metadades** (informació sobre l'estructura de la BD) a la base de dades `information_schema`.

```
-- Llistar totes les bases de dades
SELECT schema_name FROM information_schema.schemata;

-- Llistar les taules d'una BD
SELECT table_name, table_type, engine, table_rows
FROM   information_schema.tables
WHERE  table_schema = 'institut';

-- Informació de les columnes d'una taula
SELECT column_name, data_type, is_nullable, column_default, column_key
FROM   information_schema.columns
WHERE  table_schema = 'institut' AND table_name = 'alumne';

-- Restriccions d'una taula
SELECT constraint_name, constraint_type
FROM   information_schema.table_constraints
WHERE  table_schema = 'institut' AND table_name = 'matricula';

-- Claus foranes
SELECT constraint_name, column_name, referenced_table_name,
       ↪ referenced_column_name
FROM   information_schema.key_column_usage
WHERE  table_schema = 'institut' AND referenced_table_name IS NOT
       ↪ NULL;
```

2. Accés a la informació: vistes, usuaris i permisos

2.1. Vistes

Una **vista** és una consulta emmagatzemada que es comporta com una taula virtual. No emmagatzema dades, sinó la definició de la consulta.

Avantatges:

- Simplifica consultes complexes.
- Proporciona una capa d'abstracció: els usuaris no veuen les taules base.
- Permet control d'accés en l'àmbit de columna o fila.

```
-- Crear una vista
CREATE VIEW v_alumnes_actius AS
  SELECT id_alumne, nom, cognoms, email
  FROM   alumne
  WHERE  actiu = TRUE;

-- Consultar la vista
SELECT * FROM v_alumnes_actius WHERE cognoms LIKE 'Mart%';

-- Crear una vista de resum per a professors
CREATE VIEW v_resum_notes AS
  SELECT mo.nom_modul,
         COUNT(m.id_matricula)      AS total_alumnes,
         ROUND(AVG(m.nota_final), 2) AS nota_mitjana,
         SUM(m.nota_final >= 5)      AS aprovats,
         SUM(m.nota_final < 5)      AS suspesos
  FROM   matricula m
        INNER JOIN modul mo ON m.id_modul = mo.id_modul
  WHERE  m.nota_final IS NOT NULL
  GROUP BY mo.id_modul, mo.nom_modul;

-- Modificar una vista
CREATE OR REPLACE VIEW v_alumnes_actius AS
  SELECT id_alumne, nom, cognoms, email, data_naix
  FROM   alumne
  WHERE  actiu = TRUE;

-- Eliminar una vista
DROP VIEW IF EXISTS v_alumnes_actius;
```


Vistes actualitzables

Una vista és **actualitzable** (permet INSERT/UPDATE/DELETE) si no conté: GROUP BY, HAVING, funcions d'agregació, DISTINCT, UNION, subconsultes al SELECT.

```
-- Vista actualitzable
CREATE VIEW v_contacte_alumne AS
    SELECT id_alumne, nom, cognoms, email, telefon
    FROM alumne;

-- Actualitzar a través de la vista
UPDATE v_contacte_alumne
SET     email = 'nou@exemple.cat'
WHERE   id_alumne = 7;
```

Sinònims

MariaDB no té sinònims nadius com Oracle. S'utilitzen vistes simples per simular-los:

```
CREATE VIEW clients AS SELECT * FROM alumne;
```

2.2. Gestió d'usuaris

Crear usuaris

```
-- Usuari local (només des de localhost)
CREATE USER 'professor'@'localhost' IDENTIFIED BY
    ↪ 'contrasenya_segura';

-- Usuari remot (des de qualsevol IP)
CREATE USER 'app_user'@'%' IDENTIFIED BY 'contrasenya_app';

-- Usuari des d'una IP específica
CREATE USER 'dba'@'192.168.1.100' IDENTIFIED BY 'contrasenya_dba';

-- Canviar la contrasenya
ALTER USER 'professor'@'localhost' IDENTIFIED BY 'nova_contrasenya';

-- Forçar canvi de contrasenya en el pròxim accés
ALTER USER 'professor'@'localhost' PASSWORD EXPIRE;

-- Eliminar un usuari
DROP USER IF EXISTS 'professor'@'localhost';
```

Veure usuaris existents

```
SELECT user, host, password_expired, account_locked
FROM   mysql.user;
```

2.3. Privilegis

Nivells de privilegis

Nivell	Àmbit
Global	Totes les BD del servidor
BD	Totes les taules d'una BD concreta
Taula	Una taula concreta
Columna	Una columna concreta
Rutina	Un procediment o funció

Assignar privilegis (GRANT)

```
-- Tots els privilegis sobre una BD (usuari administrador d'una BD)
GRANT ALL PRIVILEGES ON institut.* TO 'admin_inst'@'localhost';

-- Privilegis de lectura sobre tota una BD
GRANT SELECT ON institut.* TO 'professor'@'localhost';

-- Privilegis de lectura i escriptura sobre una taula
GRANT SELECT, INSERT, UPDATE ON institut.matricula TO
↳ 'secretaria'@'localhost';

-- Privilegis a nivell de columna
GRANT SELECT (nom, cognoms, email) ON institut.alumne TO
↳ 'consultor'@'localhost';

-- Permetre que un usuari pugui atorgar privilegis a altres
GRANT SELECT ON institut.* TO 'cap_estudi'@'localhost' WITH GRANT
↳ OPTION;

-- Privilegi global per crear usuaris (DBA)
GRANT CREATE USER ON *.* TO 'dba'@'localhost';

-- Aplicar els canvis immediatament
FLUSH PRIVILEGES;
```

Revisar privilegis

```
-- Veure els privilegis d'un usuari
SHOW GRANTS FOR 'professor'@'localhost';

-- Privilegis de l'usuari actual
SHOW GRANTS;
```

Revocar privilegis (REVOKE)

```
-- Revocar un privilegi concret
REVOKE INSERT ON institut.matricula FROM 'secretaria'@'localhost';

-- Revocar tots els privilegis sobre una BD
REVOKE ALL PRIVILEGES ON institut.* FROM 'professor'@'localhost';

FLUSH PRIVILEGES;
```

2.4. Rols

MariaDB 10.0.5+ suporta **rols** per agrupar privilegis.

```
-- Crear rols
CREATE ROLE rol_lectura;
CREATE ROLE rol_secretaria;
CREATE ROLE rol_professor;

-- Assignar privilegis als rols
GRANT SELECT ON institut.* TO rol_lectura;
GRANT SELECT, INSERT, UPDATE ON institut.matricula TO rol_secretaria;
GRANT SELECT ON institut.* TO rol_professor;
GRANT SELECT ON v_resum_notes TO rol_professor;

-- Assignar rols a usuaris
GRANT rol_lectura TO 'consultor'@'localhost';
GRANT rol_secretaria TO 'secretaria'@'localhost';
GRANT rol_professor TO 'professor'@'localhost';

-- Activar el rol per defecte en iniciar sessió
SET DEFAULT ROLE rol_professor FOR 'professor'@'localhost';

-- Eliminar un rol
DROP ROLE IF EXISTS rol_lectura;
```

2.5. Normativa legal: protecció de dades

A l'hora de gestionar bases de dades amb informació personal cal complir el **Reglament General de Protecció de Dades (RGPD)** (UE 2016/679) i la **Llei Orgànica de Protecció de Dades i Garantia dels Drets Digitals (LOPDGDD)** (Llei Orgànica 3/2018).

Principis clau del RGPD aplicats a les BD:

- **Minimització de dades:** emmagatzemar únicament les dades estrictament necessàries.
- **Limitació del termini de conservació:** no mantenir dades personals més del necessari.
- **Integritat i confidencialitat:** protegir les dades contra accessos no autoritzats (control de permisos, xifrat).
- **Responsabilitat proactiva:** el responsable ha de demostrar el compliment.

Mesures tècniques recomanades:

- Crear usuaris amb el mínim privilegi necessari (*principi de mínim privilegi*).
- Xifrar les dades confidencials (contrasenyes amb PASSWORD() o millor hash extern; columnes amb AES_ENCRYPT()).
- Registrar els accessos i modificacions (logs d'auditoria).
- Realitzar còpies de seguretat xifrades.

3. Automatització de tasques: guions d'administració

3.1. Procediments emmagatzemats

```
DELIMITER //
```

```
-- Procediment per matricular un alumne
CREATE PROCEDURE sp_matricular(
    IN p_id_alumne INT,
    IN p_id_modul INT,
    IN p_curs YEAR,
    OUT p_resultat VARCHAR(100)
)
BEGIN
    DECLARE v_existeix INT DEFAULT 0;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        SET p_resultat = 'Error: no s''ha pogut realitzar la
↪ matrícula.';
    END;

    -- Comprovar si ja existeix la matrícula
    SELECT COUNT(*) INTO v_existeix
    FROM matricula
    WHERE id_alumne = p_id_alumne
    AND id_modul = p_id_modul
```

```

        AND curs = p_curs;

    IF v_existeix > 0 THEN
        SET p_resultat = 'Ja existeix aquesta matrícula.';
    ELSE
        START TRANSACTION;
        INSERT INTO matricula (id_alumne, id_modul, curs)
        VALUES (p_id_alumne, p_id_modul, p_curs);
        COMMIT;
        SET p_resultat = CONCAT('Matrícula registrada. ID: ',
        ↪ LAST_INSERT_ID());
    END IF;
END //

DELIMITER ;

-- Cridar el procediment
CALL sp_matricular(5, 2, 2024, @res);
SELECT @res;

```

3.2. Funcions emmagatzemades

```

DELIMITER //

CREATE FUNCTION fn_nota_literal(p_nota DECIMAL(4,2))
RETURNS VARCHAR(20)
DETERMINISTIC
BEGIN
    RETURN CASE
        WHEN p_nota >= 9 THEN 'Excel·lent'
        WHEN p_nota >= 7 THEN 'Notable'
        WHEN p_nota >= 6 THEN 'Bé'
        WHEN p_nota >= 5 THEN 'Suficient'
        WHEN p_nota IS NULL THEN 'Pendent'
        ELSE 'Suspès'
    END;
END //

DELIMITER ;

-- Ús de la funció
SELECT a.nom, a.cognoms, m.nota_final, fn_nota_literal(m.nota_final)
↪ AS qualificació
FROM matricula m
    INNER JOIN alumne a ON m.id_alumne = a.id_alumne
WHERE m.id_modul = 1;

```

3.3. Disparadors (triggers)

Un **disparador** s'executa automàticament quan es produeix un esdeveniment (INSERT, UPDATE, DELETE) sobre una taula, BEFORE o AFTER l'esdeveniment.

```
-- Taula d'auditoria
CREATE TABLE audit_notes (
    id_audit          INT AUTO_INCREMENT PRIMARY KEY,
    id_matricula      INT,
    nota_anterior     DECIMAL(4,2),
    nota_nova         DECIMAL(4,2),
    usuari            VARCHAR(100),
    data_canvi        DATETIME DEFAULT NOW()
);

DELIMITER //

-- Trigger BEFORE UPDATE: validar la nota
CREATE TRIGGER trg_valida_nota
BEFORE UPDATE ON matricula
FOR EACH ROW
BEGIN
    IF NEW.nota_final < 0 OR NEW.nota_final > 10 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La nota ha de ser entre 0 i 10.';
    END IF;
END //

-- Trigger AFTER UPDATE: registrar l'historial de canvis
CREATE TRIGGER trg_audit_nota
AFTER UPDATE ON matricula
FOR EACH ROW
BEGIN
    IF OLD.nota_final <> NEW.nota_final OR
       (OLD.nota_final IS NULL AND NEW.nota_final IS NOT NULL) THEN
        INSERT INTO audit_notes (id_matricula, nota_anterior,
↵ nota_nova, usuari)
        VALUES (NEW.id_matricula, OLD.nota_final, NEW.nota_final,
↵ USER());
    END IF;
END //

-- Trigger BEFORE DELETE: impedir l'eliminació de matrícula amb nota
CREATE TRIGGER trg_protegeix_nota
BEFORE DELETE ON matricula
FOR EACH ROW
BEGIN
    IF OLD.nota_final IS NOT NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'No es pot eliminar una matrícula amb nota
↵ registrada.';
    END IF;
END //
```

```

DELIMITER ;

-- Veure els triggers d'una BD
SHOW TRIGGERS FROM institut;
SHOW TRIGGERS FROM institut LIKE 'matricula';

-- Eliminar un trigger
DROP TRIGGER IF EXISTS trg_audit_nota;

```

3.4. Esdeveniments planificats

MariaDB permet programar l'execució automàtica de tasques amb l'**Event Scheduler**.

```

-- Activar l'Event Scheduler
SET GLOBAL event_scheduler = ON;

-- Verificar
SHOW VARIABLES LIKE 'event_scheduler';

-- Esdeveniment que s'executa una vegada en una data futura
CREATE EVENT ev_netejar_proves
ON SCHEDULE AT '2025-07-01 03:00:00'
DO
    DELETE FROM alumne WHERE nom LIKE 'Nom_%' AND cognoms LIKE
    ↪ 'Cognom_%';

-- Esdeveniment recurrent: neteja diari dels logs antics
CREATE EVENT ev_neteja_audit
ON SCHEDULE EVERY 1 DAY
STARTS CURRENT_TIMESTAMP + INTERVAL 1 DAY
DO
    DELETE FROM audit_notes WHERE data_canvi < DATE_SUB(NOW(),
    ↪ INTERVAL 90 DAY);

-- Esdeveniment recurrent que s'atura en una data
CREATE EVENT ev_backup_semanal
ON SCHEDULE EVERY 1 WEEK
STARTS '2025-01-06 02:00:00'
ENDS '2025-12-31 02:00:00'
DO
    -- Aquí normalment es crida un procediment emmagatzemat
    CALL sp_backup_logic();

-- Veure els esdeveniments
SHOW EVENTS FROM institut;

-- Desactivar/activar un esdeveniment
ALTER EVENT ev_neteja_audit DISABLE;
ALTER EVENT ev_neteja_audit ENABLE;

```

```
-- Eliminar un esdeveniment
DROP EVENT IF EXISTS ev_netejar_proves;
```

3.5. Excepcions i gestió d'errors

```
DELIMITER //
```

```
CREATE PROCEDURE sp_exemple_errors(IN p_id INT)
BEGIN
    -- Gestió de l'error 1062 (entrada duplicada)
    DECLARE EXIT HANDLER FOR 1062
    BEGIN
        SELECT 'Error: ja existeix un registre amb aquest
↪ identificador.' AS missatge;
    END;

    -- Gestió de qualsevol error SQL
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        GET DIAGNOSTICS CONDITION 1
            @codi = MYSQL_ERRNO,
            @msg = MESSAGE_TEXT;
        SELECT CONCAT('Error ', @codi, ': ', @msg) AS missatge;
        ROLLBACK;
    END;

    START TRANSACTION;
    INSERT INTO alumne (id_alumne, nom, cognoms) VALUES (p_id, 'Test',
↪ 'Test');
    COMMIT;
    SELECT 'Registre inserit correctament.' AS missatge;
END //
```

```
DELIMITER ;
```


4. Optimització del rendiment: monitoratge i índexs

4.1. Eines de monitoratge

Monitoratge des de la consola SQL

```
-- Veure els processos actuals
SHOW PROCESSLIST;
SHOW FULL PROCESSLIST;

-- Matar un procés bloquejat
KILL QUERY 42; -- mata la consulta però manté la connexió
KILL 42;       -- tanca la connexió

-- Variables de sistema
SHOW GLOBAL STATUS;
SHOW GLOBAL STATUS LIKE 'Conn%';
SHOW GLOBAL STATUS LIKE 'Queries';
SHOW GLOBAL STATUS LIKE 'Innodb%';
SHOW GLOBAL STATUS LIKE 'Slow_queries';

-- Variables de configuració
SHOW GLOBAL VARIABLES LIKE 'max_connections';
SHOW GLOBAL VARIABLES LIKE 'innodb_buffer_pool_size';

-- Estadístiques de les taules
SHOW TABLE STATUS FROM institut;

-- Canviar paràmetres en calent (sense reiniciar)
SET GLOBAL max_connections = 200;
SET GLOBAL slow_query_log = ON;
```

Eines externes

- **mysqltuner**: script Perl que analitza la configuració i el rendiment i proposa millores.
- **Percona Monitoring and Management (PMM)**: plataforma completa de monitoratge.
- **Grafana + Prometheus**: amb l'exporter de MariaDB per a mètriques.

```
# Instal·lar i executar mysqltuner
wget https://raw.githubusercontent.com/major/MySQLTuner-perl/master/
↪ mysqltuner.pl
perl mysqltuner.pl --user root --pass secret
```

4.2. Índexs

Un **índex** és una estructura de dades addicional que accelera la recuperació de files sense haver d'escanejar tota la taula.

Avantatges:

- Millora el rendiment de les consultes SELECT amb WHERE, JOIN, ORDER BY.

Inconvenients:

- Ocupa espai en disc.
- Alenteix les operacions INSERT, UPDATE, DELETE (cal actualitzar l'índex).

Tipus d'índexs a MariaDB

Tipus	Descripció
PRIMARY KEY	Índex únic implícit creat automàticament per a la PK.
UNIQUE	No permet valors duplicats.
INDEX / KEY	Índex ordinari (sí que permet duplicats).
FULLTEXT	Per a cerques de text complet.
SPATIAL	Per a dades geogràfiques.

Crear i gestionar índexs

```
-- Crear un índex en crear la taula
CREATE TABLE producte (
  id_producte INT PRIMARY KEY AUTO_INCREMENT,
  nom VARCHAR(100) NOT NULL,
  categoria VARCHAR(50),
  preu DECIMAL(8,2),
  INDEX idx_categoria (categoria),
  INDEX idx_preu (preu),
  UNIQUE INDEX uq_nom (nom)
);

-- Afegir un índex a una taula existent
CREATE INDEX idx_cognom ON alumne (cognoms);

-- Índex compost (per a consultes que filtren per múltiples columnes)
CREATE INDEX idx_modul_curs ON matricula (id_modul, curs);

-- Índex FULLTEXT (per a cerques de text)
ALTER TABLE producte ADD FULLTEXT INDEX ft_nom (nom);

-- Veure els índexs d'una taula
SHOW INDEX FROM alumne;
SHOW INDEX FROM matricula;
```

```
-- Eliminar un índex
DROP INDEX idx_cognom ON alumne;
ALTER TABLE alumne DROP INDEX idx_cognom;
```

4.3. Anàlisi de consultes: EXPLAIN

L'ordre EXPLAIN mostra com MariaDB executarà una consulta i és fonamental per a l'optimització.

```
EXPLAIN
SELECT a.nom, a.cognoms, m.nota_final
FROM   matricula m
       INNER JOIN alumne a ON m.id_alumne = a.id_alumne
WHERE  m.id_modul = 3 AND m.nota_final >= 5;
```

Columnes importants del resultat:

Columna	Descripció
type	Tipus d'accés: ALL (escaneig complet, dolent) → ref → eq_ref → const (molt bo)
key	Índex utilitzat (NULL = cap índex)
rows	Estimació de files examinades
Extra	Informació addicional: Using filesort, Using temporary (a evitar), Using index (bo)

```
-- Versió estesa
EXPLAIN FORMAT=JSON
SELECT * FROM alumne WHERE cognoms LIKE 'Puig%';
```

4.4. Optimització de l'estructura

```
-- Analitzar i actualitzar estadístiques (per a l'optimitzador)
ANALYZE TABLE alumne, matricula;

-- Comprovar i reparar taules MyISAM
CHECK TABLE alumne;
REPAIR TABLE alumne;

-- Defragmentar una taula InnoDB (reconstrueix l'espai lliure)
OPTIMIZE TABLE matricula;

-- Compactar el log d'InnoDB (allibera espai)
SET GLOBAL innodb_fast_shutdown = 0;
```

4.5. Alertes de rendiment

Es poden crear procediments i esdeveniments que monitorin el rendiment i alertin:

```
-- Taula per emmagatzemar alertes
CREATE TABLE alerta_rendiment (
    id_alerta    INT AUTO_INCREMENT PRIMARY KEY,
    tipus        VARCHAR(50),
    valor        DECIMAL(10,2),
    descripcio   VARCHAR(255),
    data_alerta  DATETIME DEFAULT NOW()
);

DELIMITER //

CREATE PROCEDURE sp_comprova_connexions()
BEGIN
    DECLARE v_connexions INT;
    DECLARE v_max_conn    INT;

    SELECT VARIABLE_VALUE INTO v_connexions
    FROM information_schema.GLOBAL_STATUS
    WHERE VARIABLE_NAME = 'Threads_connected';

    SELECT VARIABLE_VALUE INTO v_max_conn
    FROM information_schema.GLOBAL_VARIABLES
    WHERE VARIABLE_NAME = 'max_connections';

    IF v_connexions > (v_max_conn * 0.8) THEN
        INSERT INTO alerta_rendiment (tipus, valor, descripcio)
        VALUES ('CONNEXIONS', v_connexions,
            CONCAT('Connexions actives (', v_connexions, ')
        ↪ superen el 80% del màxim (', v_max_conn, ')'));
    END IF;
END //

DELIMITER ;

-- Esdeveniment que executa la comprovació cada 5 minuts
CREATE EVENT ev_monitoratge
ON SCHEDULE EVERY 5 MINUTE
DO CALL sp_comprova_connexions();
```

4.6. Optimitzar el sistema operatiu per a MariaDB

```
# Verificar la mida del buffer pool d'InnoDB en calent
mysql -u root -p -e "SHOW VARIABLES LIKE 'innodb_buffer_pool_size';"

# Ajustar el límit de fitxers oberts (important per a servidors amb
  ↳ moltes taules)
# /etc/security/limits.conf
mysql soft nfile 65536
mysql hard nfile 65536

# Desactivar l'accés a l'hora als fitxers (millora el rendiment d'E/S)
# /etc/fstab: afegir l'opció noatime a la partició de dades
# /dev/sdb1 /var/lib/mysql ext4 defaults,noatime 0 2

# Veure l'ús de memòria de MariaDB
ps aux | grep mysql
cat /proc/$(pidof mysqld)/status | grep VmRSS
```

5. Bases de dades distribuïdes i replicació

5.1. Bases de dades distribuïdes

Una **base de dades distribuïda** és un conjunt de bases de dades emmagatzemades en nodes físicament separats, interconnectats per una xarxa, que es presenten als usuaris com una única BD lògica.

Avantatges:

- **Disponibilitat:** si un node cau, els altres continuen funcionant.
- **Rendiment:** les consultes es poden executar en paral·lel.
- **Escalabilitat:** és fàcil afegir nodes.
- **Localitat:** les dades es poden emmagatzemar a prop dels usuaris que les utilitzen.

Inconvenients:

- Complexitat de gestió.
- Dificultat per mantenir la consistència (teorema CDP).
- Latència de xarxa.

Teorema CDP

En un sistema distribuït, és impossible garantir simultàniament:

- **Consistència (C):** tots els nodes veuen les mateixes dades.
- **Disponibilitat (D):** totes les peticions reben resposta.
- **Tolerància a particions (P):** el sistema funciona malgrat fallades de xarxa.

Cada sistema ha d'escollir dues de les tres propietats.

5.2. Tècniques de fragmentació

La **fragmentació** divideix les dades entre els nodes.

Fragmentació horitzontal (sharding)

Cada node conté un subconjunt de **files** de la taula.

```
Node 1: alumnes amb id_alumne 1 a 10.000
Node 2: alumnes amb id_alumne 10.001 a 20.000
Node 3: alumnes amb id_alumne > 20.000
```

Criteris de fragmentació: per rang, per hash, per llista (zona geogràfica, etc.).

Fragmentació vertical

Cada node conté un subconjunt de **columnes** de la taula.

```
Node 1: id_alumne, nom, cognoms, data_naix (dades personals)
Node 2: id_alumne, email, telefon          (dades de contacte)
```

Fragmentació híbrida (mixta)

Combina fragmentació horitzontal i vertical.

5.3. Replicació a MariaDB

La **replicació** copia automàticament les dades d'un servidor (**mestre**) a un o més servidors (**esclaus**). S'utilitza per a:

- **Alta disponibilitat:** si el mestre cau, un esclau pot promocionar-se.
- **Escalabilitat de lectura:** les consultes SELECT es distribueixen entre els esclaus.
- **Còpies de seguretat:** es realitzen còpies des dels esclaus sense afectar el mestre.

Tipus de replicació

Tipus	Descripció
Asíncrona	El mestre no espera confirmació dels esclaus. Alta velocitat, petita finestra de pèrdua de dades.
Semisíncrona	El mestre espera confirmació d'almenys un esclau.
Síncrona	El mestre espera confirmació de tots els esclaus. Màxima consistència, però més lenta. Implementada per Galera Cluster.

Formats de replicació

- **STATEMENT:** replica les sentències SQL tal qual. Pot causar inconsistències amb funcions no deterministes.
- **ROW** (recomanat): replica les files canviades exactament.
- **MIXED:** utilitza STATEMENT per defecte i ROW quan és necessari.

5.4. Configuració de replicació mestre-esclau

Configuració del servidor mestre

Editar `/etc/mysql/mariadb.conf.d/50-server.cnf` al servidor mestre:

```
[mysqld]
server-id      = 1
log_bin        = /var/log/mysql/mariadb-bin
binlog_format  = ROW
binlog_do_db   = institut      # Opcional: replicar només aquesta BD
expire_logs_days = 10
```

```
sudo systemctl restart mariadb
```

```
-- Crear un usuari de replicació al mestre
CREATE USER 'replicacio'@'%' IDENTIFIED BY 'repl_password_segur';
GRANT REPLICATION SLAVE ON *.* TO 'replicacio'@'%';
FLUSH PRIVILEGES;

-- Obtenir la posició actual del log binari
FLUSH TABLES WITH READ LOCK;
SHOW MASTER STATUS;
-- Anotar: File i Position
-- Exemple: mariadb-bin.000003 | 456
```

```
# Fer una còpia de la BD per inicialitzar l'esclau
mysqldump -u root -p --all-databases --master-data >
↪ /tmp/dump_master.sql
# Alliberar el bloqueig
mysql -u root -p -e "UNLOCK TABLES;"

# Transferir el dump a l'esclau
scp /tmp/dump_master.sql usuari@esclau:/tmp/
```

Configuració del servidor esclau

Editar `/etc/mysql/mariadb.conf.d/50-server.cnf` a l'esclau:

```
[mysqld]
server-id      = 2
relay-log      = /var/log/mysql/mariadb-relay
log_bin        = /var/log/mysql/mariadb-bin
read_only      = 1    # Prevenir escriptures accidentals a l'esclau
```

```
sudo systemctl restart mariadb

# Importar el dump del mestre
mysql -u root -p < /tmp/dump_master.sql
```

```
-- Configurar la connexió al mestre
CHANGE MASTER TO
  MASTER_HOST      = '192.168.1.10',
  MASTER_USER      = 'replicacio',
  MASTER_PASSWORD  = 'repl_password_segur',
  MASTER_LOG_FILE  = 'mariadb-bin.000003',
  MASTER_LOG_POS   = 456;

-- Iniciar la replicació
START SLAVE;

-- Verificar l'estat
SHOW SLAVE STATUS\G
```

Camps clau de `SHOW SLAVE STATUS\G`:

Camp	Significat
Slave_IO_Running	Yes si el fil d'E/S (llegeix el log binari del mestre) funciona.
Slave_SQL_Running	Yes si el fil SQL (aplica els esdeveniments) funciona.
Seconds_Behind_Master	Retard en segons respecte al mestre (0 = al dia).
Last_Error	Darrer error ocorregut.

5.5. Replicació en cadena

En la replicació en cadena, un esclau actua alhora com a mestre d'altres esclaus.

```
Mestre (server-id=1)
├─ Esclau/Mestre intermediari (server-id=2)
│   └─ Esclau final (server-id=3)
│       └─ Esclau final (server-id=4)
```

Per a l'esclau intermediari cal habilitar:

```
[mysqld]
server-id      = 2
log_bin        = /var/log/mysql/mariadb-bin
log_slave_updates = 1 # Escriu al seu propi binlog els
↳ esdeveniments rebuts del mestre
read_only      = 1
```

5.6. Galera Cluster (replicació síncrona)

Galera Cluster és una solució de clúster multi-mestre per a MariaDB que proporciona:

- **Multi-mestre actiu:** tots els nodes accepten escriptures.
- **Replicació síncrona:** no hi ha pèrdua de dades.
- **Alta disponibilitat:** tolerant a la fallada de nodes.

Instal·lació (Debian/Ubuntu)

```
sudo apt install mariadb-server galera-4 -y
```

Configuració (node 1)

```
# /etc/mysql/mariadb.conf.d/60-galera.cnf
[mysqld]
binlog_format      = ROW
default-storage-engine = innodb
innodb_autoinc_lock_mode = 2
bind-address       = 0.0.0.0

# Galera Provider
wsrep_on           = ON
wsrep_provider      = /usr/lib/galera/libgalera_smm.so
wsrep_cluster_name  = "cluster_institut"
wsrep_cluster_address =
↳ "gcomm://192.168.1.10,192.168.1.11,192.168.1.12"
wsrep_node_address  = "192.168.1.10"
wsrep_node_name     = "node1"
```

```
wsrep_sst_method          = rsync
```

```
# Inicialitzar el clúster (només al primer node, una única vegada)
sudo galera_new_cluster
```

```
# Als altres nodes, iniciar normalment
sudo systemctl start mariadb
```

```
-- Verificar l'estat del clúster
SHOW STATUS LIKE 'wsrep%';
-- wsrep_cluster_size: nombre de nodes
-- wsrep_local_state_comment: Synced = OK
-- wsrep_connected: ON
-- wsrep_ready: ON
```

5.7. Comprovació de fallades en sistemes distribuïts

```
# Simular la caiguda d'un node
sudo systemctl stop mariadb # En un dels nodes esclaus

# Verificar des del mestre que la replicació detecta la fallada
mysql -u root -p -e "SHOW SLAVE HOSTS;"
mysql -u root -p -e "SHOW MASTER STATUS;"

# Reiniciar el node aturat
sudo systemctl start mariadb

# Verificar que l'esclau es re-sincronitza
mysql -u root -p -e "SHOW SLAVE STATUS\G"
# Seconds_Behind_Master hauria de tornar a 0 progressivament

# En Galera: comprovar que el node tornat a unir-se es re-sincronitza
↪ (SST/IST)
mysql -u root -p -e "SHOW STATUS LIKE 'wsrep_local_state_comment';"
# Ha de passar de Joiner → Joined → Synced
```

Versions d'aquest document

- [HTML](#)
- [PDF](#)
- [ODT](#)
- [MD](#)

[Domini Públic \(CC0\)](#)