



UD 5. MODELO FÍSICO DML

Transacciones e índices

GUÍA DE
ESTUDIO

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

Bases de Datos (BD)

CFGS Desarrollo de Aplicaciones Web (DAW)

Abelardo Martínez

a.martinezserrano@edu.gva.es

Pau Miñana

p.minanacliment@edu.gva.es

Basado y modificado de Sergio Badal www.sergiobadal.com

a distancia



UD 5. MODELO FÍSICO DML

- ¿Qué veremos esta semana?

- UD 5.1 MODELO FÍSICO DML
 - Inserción de datos
 - Actualización de datos
 - Borrado de datos
- Actividades propuestas (no evaluables)
 - **Boletín A:** ejercicios sobre modelado físico DML

- ¿Qué veremos la semana que viene?

- Soluciones Boletín A
- UD 5.2 MODELO FÍSICO DML
 - Transacciones
 - Índices



UD 5. MODELO FÍSICO DML



UD 5 MODELO FÍSICO DML

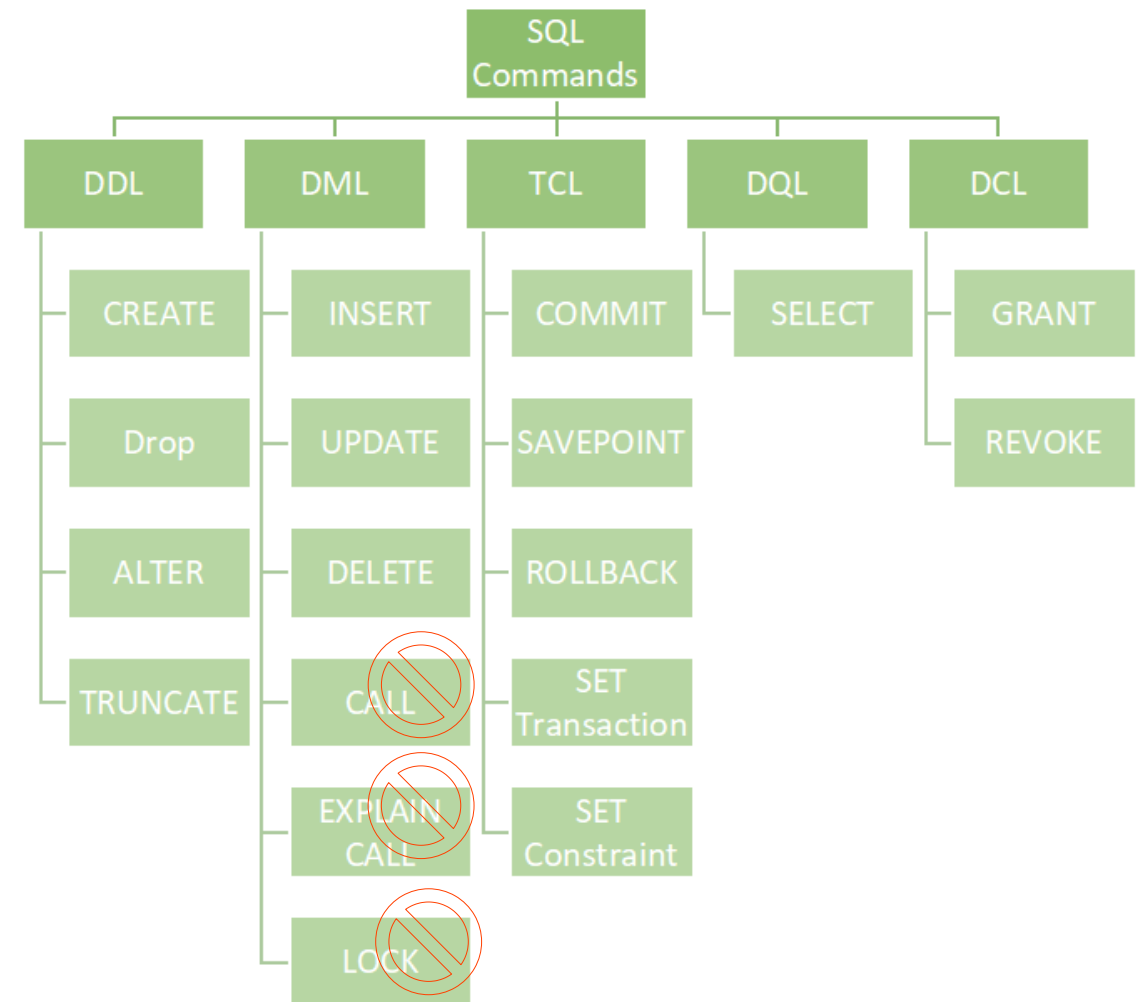
5.2.1 Transacciones

5.2.2 Índices

5.2.1 Transacciones

Hasta ahora hemos conseguido crear y modificar los metadatos (DDL). El próximo paso será incluir registros en las tablas que hemos creado.

- El DML (Data Manipulation Language) lo forman las instrucciones capaces de modificar los datos de las tablas.
- **Algunos autores incluyen un subgrupo dentro del DML al que llaman TCL (Transaction and Control Language) para tratar las transacciones que vemos ahora.**



5.2.1 Transacciones

Una transacción está formada por una serie de instrucciones DML (INSERT, DELETE, UPDATE) que pueden realizarse o no dependiendo de cómo tengamos configurado nuestro SGBD.

– TRANSACCIÓN 1

```
INSERT INTO ESTUDIANTES ('12', 'PEDRO', 'GARCÍA', 'LÓPEZ', '20/12/1978');  
INSERT INTO ESTUDIANTES ('13', 'MARTA', 'LÓPEZ', 'LÓPEZ', '22/10/1979');  
DELETE FROM ESTUDIANTES WHERE IDESTUDIANTE=13;
```

– TRANSACCIÓN 2

```
DELETE FROM ESTUDIANTES WHERE IDESTUDIANTE=12;  
INSERT INTO ESTUDIANTES ('14', 'LUCÍA', 'HERRERO', 'LÓPEZ', '02/11/1978');
```


5.2.1 Transacciones

Si tenemos el modo **autocommit** activado, cada instrucción **será definitiva** al momento de ejecutarla.

– TRANSACCIÓN 1

```
INSERT INTO ESTUDIANTES ('12', 'PEDRO', 'GARCÍA', 'LÓPEZ', '20/12/1978');  
INSERT INTO ESTUDIANTES ('13', 'MARTA', 'LÓPEZ', 'LÓPEZ', '22/10/1979');  
DELETE FROM ESTUDIANTES WHERE IDESTUDIANTE=13;
```

Si tenemos el modo **autocommit** desactivado, las instrucciones **serán definitivas** (**principalmente**) cuando hagamos un COMMIT (confirmar) o un ROLLBACK (descartar):

– TRANSACCIÓN 2

```
INSERT INTO ESTUDIANTES ('12', 'PEDRO', 'GARCÍA', 'LÓPEZ', '20/12/1978');  
INSERT INTO ESTUDIANTES ('13', 'MARTA', 'LÓPEZ', 'LÓPEZ', '22/10/1979');  
DELETE FROM ESTUDIANTES WHERE IDESTUDIANTE=13;  
COMMIT;
```

– TRANSACCIÓN 3

```
INSERT INTO ESTUDIANTES ('16', 'LAILA', 'KEIZ', 'LULA', '22/10/1979');  
DELETE FROM ESTUDIANTES WHERE IDESTUDIANTE=16;  
ROLLBACK;
```

5.2.1 Transacciones

Si tenemos el modo **autocommit** activado, cada instrucción **será definitiva** al momento de ejecutarla.

– TRANSACCIÓN 1

```
INSERT INTO ESTUDIANTES ('12', 'PEDRO', 'GARCÍA', 'LÓPEZ', '20/12/1978');  
INSERT INTO ESTUDIANTES ('13', 'MARTA', 'LÓPEZ', 'LÓPEZ', '22/10/1979');  
DELETE FROM ESTUDIANTES WHERE IDESTUDIANTE=13;
```



AUTOCOMMIT ON
(por defecto)

Si tenemos el modo **autocommit** desactivado, las instrucciones **serán definitivas** (**principalmente**) cuando hagamos un COMMIT (confirmar) o un ROLLBACK (descartar):

– TRANSACCIÓN 2

```
INSERT INTO ESTUDIANTES ('12', 'PEDRO', 'GARCÍA', 'LÓPEZ', '20/12/1978');  
INSERT INTO ESTUDIANTES ('13', 'MARTA', 'LÓPEZ', 'LÓPEZ', '22/10/1979');  
DELETE FROM ESTUDIANTES WHERE IDESTUDIANTE=13;  
COMMIT;
```



AUTOCOMMIT OFF
(por defecto)

– TRANSACCIÓN 3

```
INSERT INTO ESTUDIANTES ('16', 'LAILA', 'KEIZ', 'LULA', '22/10/1979');  
DELETE FROM ESTUDIANTES WHERE IDESTUDIANTE=16;  
ROLLBACK;
```

5.2.1 Transacciones

Si tenemos el modo **autocommit** activado, cada instrucción **será definitiva** al momento de ejecutarla.

– TRANSACCIÓN 1

```
INSERT INTO ESTUDIANTES ('12', 'PEDRO', 'GARCÍA', 'LÓPEZ', '20/12/1978');  
INSERT INTO ESTUDIANTES ('13', 'MARTA', 'LÓPEZ', 'LÓPEZ', '22/10/1979');  
DELETE FROM ESTUDIANTES WHERE IDESTUDIANTE=13;
```


Si tenemos el modo **autocommit** desactivado, las instrucciones **serán definitivas** (principalmente) cuando hagamos un COMMIT (confirmar) o un ROLLBACK (descartar):

– TRANSACCIÓN 2

```
INSERT INTO ESTUDIANTES ('12', 'PEDRO', 'GARCÍA', 'LÓPEZ', '20/12/1978');  
INSERT INTO ESTUDIANTES ('13', 'MARTA', 'LÓPEZ', 'LÓPEZ', '22/10/1979');  
DELETE FROM ESTUDIANTES WHERE IDESTUDIANTE=13;  
COMMIT;
```

– TRANSACCIÓN 3

```
INSERT INTO ESTUDIANTES ('16', 'LAILA', 'LÓPEZ', 'LÓPEZ', '22/10/1979');  
DELETE FROM ESTUDIANTES WHERE IDESTUDIANTE=16;  
ROLLBACK;
```



Una transacción comienza con la primera instrucción DML (INSERT, DELETE, UPDATE) que se ejecute y finaliza con alguna de estas circunstancias:

- Una operación COMMIT o ROLLBACK
- Una instrucción DDL (como ALTER TABLE por ejemplo)
- Una instrucción DCL (como GRANT)
- El usuario abandona la sesión
- Caída del sistema

5.2.1 Transacciones

```
mysql> -- By default, in MySQL autocommit is turned ON
mysql> CREATE TABLE ESTUDIANTES (IDEST VARCHAR(2), NOMBRECOMP VARCHAR (20));
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO ESTUDIANTES VALUES (12, 'Heikki');
Query OK, 1 row affected (0.00 sec)
mysql> -- Do another transaction with autocommit turned OFF
mysql> SET autocommit=0;
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO ESTUDIANTES VALUES (15, 'John');
Query OK, 1 row affected (0.00 sec)
mysql> INSERT INTO ESTUDIANTES VALUES (20, 'Paul');
Query OK, 1 row affected (0.00 sec)
mysql> DELETE FROM ESTUDIANTES WHERE NOMBRECOMP = 'Heikki';
Query OK, 1 row affected (0.00 sec)
mysql> -- Now we undo those last 2 inserts and the delete.
mysql> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> SELECT * FROM ESTUDIANTES;
+-----+-----+
| IDEST | NOMBRECOMP |
|       |             |
+-----+-----+
| 12 | Heikki |
+-----+-----+
```

5.2.1 Transacciones

El concepto de transacción está muy relacionado con el mundo Oracle, por lo que solo lo veremos desde un punto de vista teórico al menos en esta unidad.

UD 5. MODELO FÍSICO DML



UD 5 MODELO FÍSICO DML

5.2.1 Transacciones

5.2.2 Índices

5.2.2 Índices

Los **índices** son un **objeto más de las bases de datos**, como pueden ser las tablas, que hacen que las bases de datos aceleren las operaciones de consulta y ordenación de los datos (DQL o SELECT).

De cada tabla, se puede crear uno o varios índices siendo cada uno una copia de esa tabla ordenada en función de un campo o criterio concreto.

Hay índices que se crean AUTOMÁTICAMENTE (implícitos) e índices que podemos crear nosotros MANUALMENTE (explícitos).

5.2.2 Índices

Por ejemplo, si creamos la tabla ESTUDIANTES con el **DNI como PK**, el **id de ciudad de residencia** y el **id de país de residencia** como FKs y el **numSS** (número de la seguridad social) como UK estamos, implícitamente, creando estos objetos en la base de datos:

- 1) La tabla ESTUDIANTES
- 2) Una copia de esa tabla ordenada por la PK (DNI)
- 3) Una copia de esa tabla ordenada por la FK id_ciudad
- 4) Una copia de esa tabla ordenada por la FK id_pais
- 5) Una copia de esa tabla ordenada por la UK numSS

De esta manera, cada vez que hagamos un DML (INSERT, UPDATE, DELETE) sobre esa tabla tendremos que reconstruir todos esos objetos por lo que **los índices son recomendados solo cuando las operaciones DML son infinitamente menores que las de consulta (DQL o SELECT).**

5.2.2 Índices



¿Cuántos índices implícitos (automáticos) ves aquí?

```
mysql> show create table alpinistas;
+-----+-----+
| Table          | Create Table                               |
+-----+-----+
| alpinistas     | CREATE TABLE `alpinistas` (
  `nif` char(9) NOT NULL,
  `nombre` varchar(30) DEFAULT NULL,
  `fecha_nacimiento` date DEFAULT NULL,
  `codigo` mediumint DEFAULT NULL,
  `fecha_ingreso` date DEFAULT NULL,
  PRIMARY KEY (`nif`),
  KEY `codigo_asociacion_fk` (`codigo`),
  CONSTRAINT `codigo_asociacion_fk` FOREIGN KEY (`codigo`) REFERENCES `asociacion` (`codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0,00 sec)
```

5.2.2 Índices

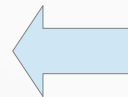


¿Cuántos índices implícitos (automáticos) ves aquí?

```
mysql> show create table alpinistas;
+-----+
| Table          | Create Table
+-----+
| alpinistas     | CREATE TABLE `alpinistas` (
  `nif` char(9) NOT NULL,
  `nombre` varchar(30) DEFAULT NULL,
  `fecha_nacimiento` date DEFAULT NULL,
  `codigo` mediumint DEFAULT NULL,
  `fecha_ingreso` date DEFAULT NULL,
  PRIMARY KEY (`nif`),
  KEY `codigo_asociacion_fk` (`codigo`),
  CONSTRAINT `codigo_asociacion_fk` FOREIGN KEY (`codigo`) REFERENCES `asociacion` (`codigo`)
  ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
```

1 row in set (0,00 sec)

Por cada FOREIGN KEY se crea una KEY (clave interna), por lo que tenemos **DOS ÍNDICES IMPLÍCITOS**



5.2.2 Índices



¿Cuántos índices implícitos (automáticos) ves aquí?

```
mysql> show create table asociacion;
+-----+-----+
| Table          | Create Table                               |
+-----+-----+
| asociacion     | CREATE TABLE `asociacion` (
  `codigo` mediumint NOT NULL,
  `nombre` varchar(30) DEFAULT NULL,
  `ubicacion` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0,00 sec)
```

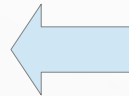
5.2.2 Índices



¿Cuántos índices implícitos (automáticos) ves aquí?

```
mysql> show create table asociacion;
+-----+-----+
| Table          | Create Table                               |
+-----+-----+
| asociacion     | CREATE TABLE `asociacion` (             |
|   `codigo`     | mediumint NOT NULL,                       |
|   `nombre`     | varchar(30) DEFAULT NULL,                |
|   `ubicacion`  | varchar(30) DEFAULT NULL,                |
|   PRIMARY KEY  | (`codigo`)                                |
| ) ENGINE=InnoDB | DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0,00 sec)
```

Tenemos **UN ÍNDICE IMPLÍCITO**



5.2.2 Índices

Hay índices que se crean AUTOMÁTICAMENTE (implícitos) e **índices que podemos crear nosotros MANUALMENTE (explícitos)**.

Por ejemplo, puedo tener una tabla de ESTUDIANTES con los campos DNI, nombre, apellidos y fecha de nacimiento con un índice sobre el campo fecha de nacimiento que me permita consultar rápidamente todos los estudiantes ordenados por ese campo.

Se almacenan aparte de la tabla a la que hace referencia, lo que permite crearlos y borrarlos en cualquier momento.

```
CREATE INDEX ix_fecha  
ON ESTUDIANTES (fecha_nacimiento);
```


5.2.2 Índices

Otro índice EXPLÍCITO (manual) podría ser el siguiente:

- Para los campos apellido1, apellido2 y nombre.
- Esto no es lo mismo que crear un índice para cada campo. Este índice es efectivo solo cuando se buscan u ordenan clientes usando los tres campos (apellido1, apellido2 y nombre) a la vez.

```
CREATE INDEX ix_nombre_completo  
ON ESTUDIANTES (apellido1, apellido2, nombre);
```

5.2.2 Índices

```
CREATE INDEX nombre_completo  
ON ESTUDIANTES (apellido1, apellido2, nombre);
```

Si realizamos una consulta sobre esa tabla ordenada por esos campos y en ese orden, la respuesta será de milisegundos **aunque la tabla tenga millones de registros.**

Se aconseja crear índices en campos que:

- Contengan una gran cantidad de valores
- Contengan una gran cantidad de nulos
- Sean parte habitual de cláusulas WHERE, GROUP BY u ORDER BY (que veremos más adelante)
- Sean parte de consultas muy frecuentes

No se aconseja en campos que:

- Pertenezcan a tablas pequeñas, ya que no aportan una mejora significativa
- No se usen a menudo en las consultas
- Pertenezcan a tablas que se actualicen frecuentemente (DML)* → aún así puede que se necesite porque una consulta utilice mucho ciertos campos (por ejemplo en una tabla de pedidos)

5.2.2 Índices

Para mostrar los índices de una tabla usamos el comando **SHOW INDEXES FROM ESTUDIANTES;**

Para eliminar un índice concreto usamos el comando **DROP INDEX IX_FECHA ON ESTUDIANTES;**

5.2.2 Índices

Veamos un caso concreto:

```
mysql> create index ix_fecha on alpinistas(fecha_nacimiento desc);
Query OK, 0 rows affected (0,03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show create table alpinistas;
+-----+
| Table          | Create Table
+-----+
| alpinistas | CREATE TABLE `alpinistas` (
  `nif` char(9) NOT NULL,
  `nombre` varchar(30) DEFAULT NULL,
  `fecha_nacimiento` date DEFAULT NULL,
  `codigo` mediumint DEFAULT NULL,
  `fecha_ingreso` date DEFAULT NULL,
  PRIMARY KEY (`nif`),
  KEY `codigo_asociacion_fk` (`codigo`),
  KEY `ix_fecha` (`fecha_nacimiento` DESC),
  CONSTRAINT `codigo_asociacion_fk` FOREIGN KEY (`codigo`) REFERENCES `asociacion` (`codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
+-----+
1 row in set (0,00 sec)

mysql> select * from alpinistas order by fecha_nacimiento desc;
```

Creo un índice
sobre un campo
por el que voy a
ordenar muy
frecuentemente.