

1- Transaccions

a) Una transacció és un conjunt seqüencial d'ordre de manipulació de bases de dades que es tracten com si fosin una única ordre i per tant, una transacció no és completa si no es poden executar totes les ordres. Si falla només una ordre, la transacció falla i totes les accions anteriors seran revertides.

b) Les transaccions tenen les següents propietats:

- * Atomicitat: Totes les operacions per separat han de funcionar perquè funcioni la transacció. Si falla una, totes les accions anteriors seran revertides.
- * Consistència: Assegura que la BD només canvia si la consignació de la transacció té èxit.
- * Aïllament: Les transaccions són independents les unes de les altres.

c) L'avantatge principal de les transaccions és la seva reversibilitat.

d) És necessari deshabilitar la funció autocommit de MySQL per poder treballar amb transaccions. Després es pot tornar a habilitar. Es pot fer amb sentències SQL o de l'API utilitzada.

2- Concepte d'autocommit, commit i rollback de SQL

a) Documentació oficial: <https://dev.mysql.com/doc/refman/5.6/en/innodb-autocommit-commit-rollback.html>

b) Idees bàsiques:

- * Servidor en mode Autocommit → Si està ON cada cop que s'envia una sentència SQL al servidor s'executa automàticament. Cada sentència és una transacció a només una sentència SQL
- * Si el mode Autocommit està OFF llavors el servidor està en mode Commit.
- * Servidor en mode Commit → S'ha d'esperar a enviar l'ordre SQL COMMIT per executar les sentències SQL i fer els canvis permanents i visibles. Si no s'ha fet un COMMIT les sentències són reversibles amb l'ordre SQL ROLLBACK.
- * Algunes ordres SQL com per exemple **create table** implícitament fan un COMMIT. Veure: <https://dev.mysql.com/doc/refman/5.6/en/implicit-commit.html>.

3- Sentències múltiples

MySQL permet opcionalment tenir múltiples sentències en una cadena de sentències. L'enviament de múltiples sentències de cop redueix els viatges d'anada i tornada des del client al servidor, però requereix un maneig especial.

Les sentències múltiples o multiconsultes han de ser executades amb l'ordre **multi_query()**. Les sentències individuals de la cadena de sentències estan separades per un punt i coma. Altre mètodes que s'utilitzen amb sentències múltiples són: **store_results()**, **free()**, **more_results()** i **next_result()**.

La biblioteca MySQLi permet treballar amb sentències múltiples però PDO no sembla que tingui aquesta opció i no es troba a la documentació oficial de l'extensió PDO cap referència a cap mètode similar a **mysql_query()**.

4- Expressions regulars

Una **expressió regular** (o col·loquialment anomenades *regexp*, acrònim de l'anglès *regular expression*) és una representació, segons unes regles sintàctiques d'un llenguatge formal, d'una porció de text genèric a buscar dins d'un altre text, com per exemple uns caràcters, paraules o patrons de text concrets.

El text genèric de l'expressió regular pot representar patrons (en anglès *patterns*) amb determinats caràcters que tenen un significat especial (per exemple, en el cas del shell d'unix, el caràcter comodí "?" per representar un caràcter qualsevol, el caràcter comodí "*" per representar un nombre qualsevol de caràcters, o classes com "[abc]" per representar qualsevol dels caràcters 'a', 'b' o 'c').

Una expressió regular està escrita seguint les regles d'un llenguatge formal, que poden ser interpretades per un programa processador d'expressions regulars, capaç d'examinar un text i reconèixer-hi les parts que es corresponen (en anglès *match*) amb l'expressió regular especificada.

Molts processadors de textos i llenguatges de programació fan ús de les seves pròpies expressions regulars per a procediments de cerca o bé de cerca i substitució de textos.