**Контрольні запитання**

1. **Що таке реплікація?**

**Реплікація** (англ. Replication) - механізм синхронізації вмісту копій об'єкта (наприклад, вмісту баз даних). Реплікація - це процес, під яким розуміється копіювання даних з одного джерела на інший (або на кількість інших) і навпаки.

При рерлікації зміни, зроблені в одній копії об'єкта, можуть бути поширені в інших копіях.

1. **Поясніть реплікація Master-Slave.**

У цьому підході виділяється один основний сервер бази даних, який називається Master. На ньому відбуваються всі зміни в даних (будь-які запити MySQL INSERT / UPDATE / DELETE). Slave сервер постійно копіює всі зміни з Master. З додатка на Slave сервер відправляються запити читання даних (запити SELECT). Таким чином Майстер сервер відповідає за зміни даних, а Слейв за читання.

У додатку потрібно використовувати два з'єднання - одне для master, друге - для slave:

<?

$master = mysql\_connect('10.10.0.1', 'root', 'pwd');

$slave = mysql\_connect('10.10.0.2', 'root', 'pwd');

# ...

mysql\_query('INSERT INTO users ...', $master);

# ...

$q = mysql\_query('SELECT \* FROM photos ...', $slave);

При виході з ладу slave - сервера, досить просто перемкнути всі додатоки на роботу з master- сервером. Після відновити реплікацію на slave і знову його запустити.

Якщо виходить з ладу master, потрібно переключити всі операції (і читання і запису) на slave. Таким чином він стане новим master - сервером. Після відновлення старого master - серверу, налаштувати на ньому репліку, і він стане новим слейв.

Набагато частіше реплікацію Master-Slave використовують не для масштабування, а для резервування. В цьому випадку, Master - сервер обробляє всі запити від програми. Слейв сервер працює в пасивному режимі. Але в разі виходу з ладу Master - сервера, всі операції переключаються на Slave.

1. **Поясніть реплікація Master-Master.**

У цій схемі, будь-який з серверів може використовуватися як для читання так і для запису.

При використанні такого типу реплікації досить вибирати випадкове з'єднання з доступних Майстрів:

<?

$masters = [

'10.10.0.1',

'10.10.0.2',

'10.10.0.3',

];

$master = mysql\_connect($masters[array\_rand($masters)], 'root', 'pwd');

# ...

mysql\_query('INSERT INTO users ...', $master);

**Переваги Master-Master реплікації**:

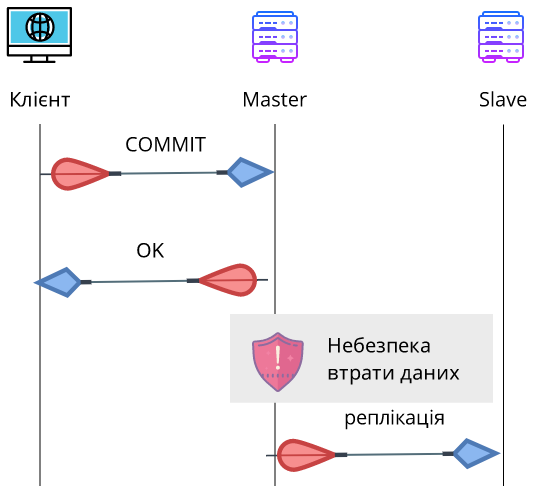
* Менше накладних росходів на перемикання
* Вище швидкість вирішення аварійних ситуацій
* Географічне розподілення дозволяє бути блище до клієнта

**Недоліки Master-Master реплікації:**:

* Щоб швидко працювати в умовах конкурентного доступу потрібні транзакції і ACID. У випадку Master-Master додається новий клас конфліктів.

1. **В чому відмінність між синхронною і асинхронною реплікацією?**

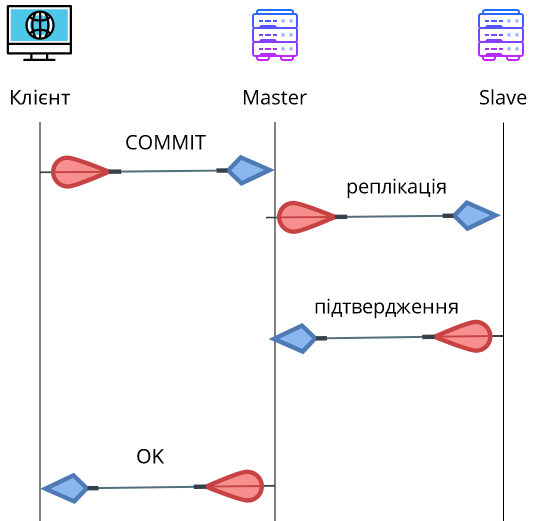
Ось приблизна схема того, як працює **асинхронна реплікація** в додатку.

[](https://github.com/endlesskwazar/distributed-databases-workbook/blob/master/resources/img/replication/img-2.png)

Від клієнта отримуємо "commit", коммітим транзакцію. Надсилаємо клієнту "ОК". А потім якось відсилаємо дані на slave - сервер. У цьому місці наші дані можуть бути втрачені, якщо slave раптом "впав". Ці дані можуть бути втрачені до відновлення сервера або назавжди. Якщо у вас «полетів» блок живлення, якщо у вас не було батарейки в диску чи сталася інша подібна річ, частинам даних практично завжди доводиться говорити "до побачення".

Для ряду випадків це абсолютно не принципово. Якщо загубиться який-небудь коментар на безкоштовному форумі, то це нікого не хвилює. А ось в фінансових або телекомунікаційних додатках втрата даних не повинна відбуватися ні в якому разі. Єдине рішення - це **синхронна реплікація**.

**Синхронна реплікація**

[](https://github.com/endlesskwazar/distributed-databases-workbook/blob/master/resources/img/replication/img-1.png)

Ми отримуємо "commit" від клієнта, відправляємо дані транзакції на slave - сервер. Чекаємо підтвердження від нього. Тільки після цього і говоримо клієнтові "ОК".

Тут у нас виникає якась затримка "спілкування" з slave - сервером в момент виконання операції "commit". Саме ця затримка є причиною уявлень про те, що все буде вкрай повільно.

1. **Поясніть процес налаштування реплікації MySQL.**

### Етап 1

На сервері, який буде виступати майстром, необхідно внести правки в my.cnf:

# Вибираємо ID сервера, довільне число, краще починати з 1

server-id = 1

# Шлях до бінарного лога

log\_bin = /var/log/mysql/mysql-bin.log

# Назву бази даних репліки

binlog\_do\_db = newdatabase

Перезапускаємо Mysql:

/etc/init.d/mysql restart

### Етап 2

Далі необхідно створити профіль користувача, з під якого відбуватиметься реплікація. Для цього запускаємо консоль:

mysql -u root -p

Далі створюємо і призначаємо права користувачеві для репліки:

GRANT REPLICATION SLAVE ON \*.\* TO 'slave\_user'@'%' IDENTIFIED BY 'password';

FLUSH PRIVILEGES;

Далі блокуємо всі таблиці в нашій базі даних:

USE newdatabase;

FLUSH TABLES WITH READ LOCK;

Перевіряємо статус Майстер-сервера:

SHOW MASTER STATUS;

Ми побачимо щось схоже на:

mysql> SHOW MASTER STATUS;

+------------------+----------+--------------+------------------+

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |

+------------------+----------+--------------+------------------+

| mysql-bin.000001 | 107 | newdatabase | |

+------------------+----------+--------------+------------------+

1 row in set (0.00 sec)

### Етап 3

Тепер необхідно зробити дамп бази даних:

mysqldump -u root -p newdatabase > newdatabase.sql

Розблокуємо таблиці в консолі mysql:

UNLOCK TABLES;

### Етап 4

В консолі mysql на слейв створюємо базу з таким же ім'ям, як і на Майстрі:

CREATE DATABASE newdatabase;

Після цього завантажуємо дамп (з bash):

mysql -u root -p newdatabase < newdatabase.sql

### Етап 5

В налаштуваннях my.cnf на слейв необхідно вказати такі параметри:

# ID слейв, зручно вибирати таким числом після Майстри

server-id = 2

# Шлях до relay

relay-log = /var/log/mysql/mysql-relay-bin.log

# Шлях до bin на Майстрі

log\_bin = /var/log/mysql/mysql-bin.log

# База даних для реплікації

binlog\_do\_db = newdatabase

### Етап 6

Нам залишилося включити реплікацію, для цього необхідно вказати параметри підключення до майстра. В консолі mysql на слейв необхідно виконати запит:

CHANGE MASTER TO MASTER\_HOST='10.10.0.1', MASTER\_USER='slave\_user', MASTER\_PASSWORD='password',

MASTER\_LOG\_FILE = 'mysql-bin.000001', MASTER\_LOG\_POS = 107;

Після цього запускаємо реплікацію на слейв:

START SLAVE;