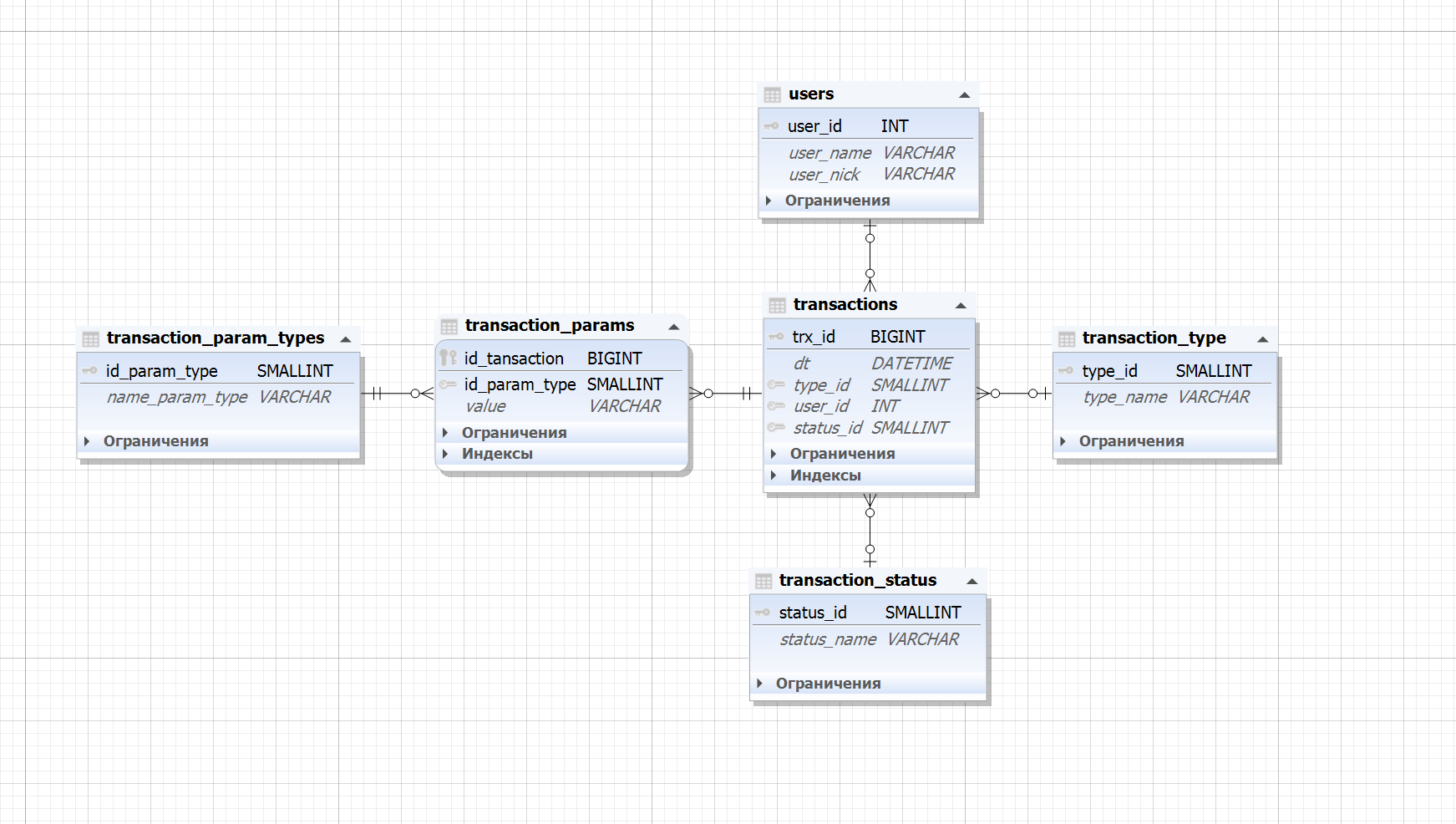
Тестовый вариант синхронизации данных.

Дана небольшая бд транзакций - transaction\_base. Её DDL в приложении.



В которой таблицы:

1. transactions – основная и самая крупная таблица операций
2. transaction\_params – её доп параметры. Один-ко многим. ВАЖНО(!) считаем, что в обе таблицы запись идёт одной транзакцией и они обе будут в одной операции бинлога
3. справочник типов этих параметров
4. справочники статусов и типов транзакций
5. Список пользователей.

Необходимо настроить синхронизацию данных на другой аналогичный сервер двумя путями. Оба пути должны себя включать чекпоинты на получателе на случай, если например получатель умер и его восстанавливали из бекапа.

## Вариант 1. Запросы к БД.

## Необходимо разработать консольное приложение, которое будет синхронизировать данные из базы-источника в базу-приемник с аналогичной схемой хранения. Данные из основных таблиц базы-источника должны подтягиваться пачками (например, по 1000 штук) в таблицы базы-приемника. Приложение должно сохранять место последней успешной синхронизации в служебной таблице базы-приемника, что бы в случае перезапуска или отказа приложения продолжить синхронизацию с последней успешной итерации. После полной синхронизации основных таблиц приложение ждет новых данных в базе-источнике и синхронизирует их при появлении. Считаем, что таблицы-справочники меняются редко и целиком синхронизируем их по таймеру. В случае конфликта используем ON DUPLICATE KEY UPDATE. Так же периодический очищаем устаревшие записи в таблице транзакции базе-приемнике (старше n-дней, задается в конфиге приложения)

## Вариант 2. Синхронизация с помощью бинлога.

Дано: сервер MySQL не ниже 5,7. На нём включен log-bin и gtid-mode для сохранения бинарных логов в формате GTID. Подключиться как слейв к серверу и вытягивать бинлоги при помощи либы <https://github.com/siddontang/go-mysql> или аналогичной. Обязательно должна быть возможность конфигурирования списка таблиц для синхронизации, в т.ч. и из разных схем. Например transaction\_base.transactions, test.test\_table

Приложение. DDL БД

*CREATE DATABASE transaction\_base*

*CHARACTER SET utf8*

*COLLATE utf8\_general\_ci;*

*/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;*

*/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;*

*SET NAMES 'utf8';*

*USE transaction\_base;*

*CREATE TABLE users (*

*user\_id int(11) UNSIGNED NOT NULL AUTO\_INCREMENT,*

*user\_name varchar(50) DEFAULT NULL,*

*user\_nick varchar(255) DEFAULT NULL,*

*PRIMARY KEY (user\_id)*

*)*

*ENGINE = INNODB,*

*CHARACTER SET utf8,*

*COLLATE utf8\_general\_ci;*

*CREATE TABLE transaction\_type (*

*type\_id smallint(6) NOT NULL AUTO\_INCREMENT,*

*type\_name varchar(50) DEFAULT NULL,*

*PRIMARY KEY (type\_id)*

*)*

*ENGINE = INNODB,*

*CHARACTER SET utf8,*

*COLLATE utf8\_general\_ci;*

*CREATE TABLE transaction\_status (*

*status\_id smallint(6) NOT NULL AUTO\_INCREMENT,*

*status\_name varchar(50) DEFAULT NULL,*

*PRIMARY KEY (status\_id)*

*)*

*ENGINE = INNODB,*

*CHARACTER SET utf8,*

*COLLATE utf8\_general\_ci;*

*CREATE TABLE transactions (*

*trx\_id bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT,*

*dt datetime DEFAULT NULL,*

*type\_id smallint(6) DEFAULT NULL,*

*user\_id int(11) UNSIGNED DEFAULT NULL,*

*status\_id smallint(6) DEFAULT NULL,*

*PRIMARY KEY (trx\_id)*

*)*

*ENGINE = INNODB,*

*CHARACTER SET utf8,*

*COLLATE utf8\_general\_ci;*

*ALTER TABLE transactions*

*ADD INDEX IDX\_transactions\_dt (dt);*

*ALTER TABLE transactions*

*ADD CONSTRAINT FK\_transactions\_status\_id FOREIGN KEY (status\_id)*

*REFERENCES transaction\_status (status\_id);*

*ALTER TABLE transactions*

*ADD CONSTRAINT FK\_transactions\_type\_id FOREIGN KEY (type\_id)*

*REFERENCES transaction\_type (type\_id);*

*ALTER TABLE transactions*

*ADD CONSTRAINT FK\_transactions\_user\_id FOREIGN KEY (user\_id)*

*REFERENCES users (user\_id);*

*CREATE TABLE transaction\_param\_types (*

*id\_param\_type smallint(6) UNSIGNED NOT NULL,*

*name\_param\_type varchar(255) DEFAULT NULL,*

*PRIMARY KEY (id\_param\_type)*

*)*

*ENGINE = INNODB,*

*CHARACTER SET utf8,*

*COLLATE utf8\_general\_ci;*

*CREATE TABLE transaction\_params (*

*id\_tansaction bigint(20) UNSIGNED NOT NULL,*

*id\_param\_type smallint(6) UNSIGNED NOT NULL,*

*value varchar(255) DEFAULT NULL,*

*PRIMARY KEY (id\_tansaction, id\_param\_type)*

*)*

*ENGINE = INNODB,*

*CHARACTER SET utf8,*

*COLLATE utf8\_general\_ci;*

*ALTER TABLE transaction\_params*

*ADD CONSTRAINT FK\_transaction\_params\_id\_param FOREIGN KEY (id\_param\_type)*

*REFERENCES transaction\_param\_types (id\_param\_type);*

*ALTER TABLE transaction\_params*

*ADD CONSTRAINT FK\_transaction\_params\_id\_tansa FOREIGN KEY (id\_tansaction)*

*REFERENCES transactions (trx\_id) ON DELETE CASCADE ON UPDATE CASCADE;*

*/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;*

*/\*!40014 SET FOREIGN\_KEY\_CHECKS = @OLD\_FOREIGN\_KEY\_CHECKS \*/;*