TRANSACTIONS

So far we have considered single independent SQL statements. Sometimes we have several SQL queries that need to be executed together: as a single logical unit of work. This is why transactions came to be.

In the 1970s, Jim Gray defined the so-called **ACID** principles in order to make transactions reliable. All the single characters stand for a single principle:

A: „atomicity”

Atomicity requires that every single transaction can be all or nothing. So if one part of the transactions fails, the entire transaction fails. So aborted transactions do not happen.

C: „consistency”

Consistency property makes sure that any transaction will bring the database from one valid state to another.

I: „isolation”

Isolation property ensures that the current excution of transactions results in a system state that would be obtained if transactions were executed serially.

D: „durability”

Durability ensures that once a transaction has been comitted, it will remain so.

/* this is how we commit a transaction

```
SET AUTOCOMMIT = 0;
START TRANSACTION;
UPDATE person SET person_name = 'Kevin B' WHERE person_id = 1;
COMMIT;
```

/* this is how we roll back the whole transaction

```
SET AUTOCOMMIT = 0;
START TRANSACTION;
UPDATE person SET person_name = 'Kevin B' WHERE person_id = 1;
ROLLBACK;
```

/* this is how we roll back to a given save point in the transaction

```
SET AUTOCOMMIT = 0;
START TRANSACTION;
SELECT * FROM person;
SAVEPOINT save1;
UPDATE person SET person_name = 'Kevin B' WHERE person_id = 1;
SAVEPOINT save2;
UPDATE person SET university = 2 WHERE person_id = 1;
ROLLBACK TO save1;
RELEASE SAVEPOINT save1;
```