

Loeng 10

Andmebaasi  
versioneerimine

# Kordamine

- Mis probleemi lahendab ORM (nt. JPA)

# Andmebaasi versioneerimine

- Skeemi versioneerimine
- Andmete versioneerimine

# Skeemi versioneerimine

- Esmane arendus
  - arenduse ajal pole vaja andmeid säilitada
- Hilisemad iteratsioonid
  - arenduse ajal on vaja andmeid säilitada

# Skeemi versioneerimine (häkk)

- Alustame skeemi failist
- Test baasi muutmiseks muudame skeemi faili
- Live baasis kasutame ALTER TABLE lauseid

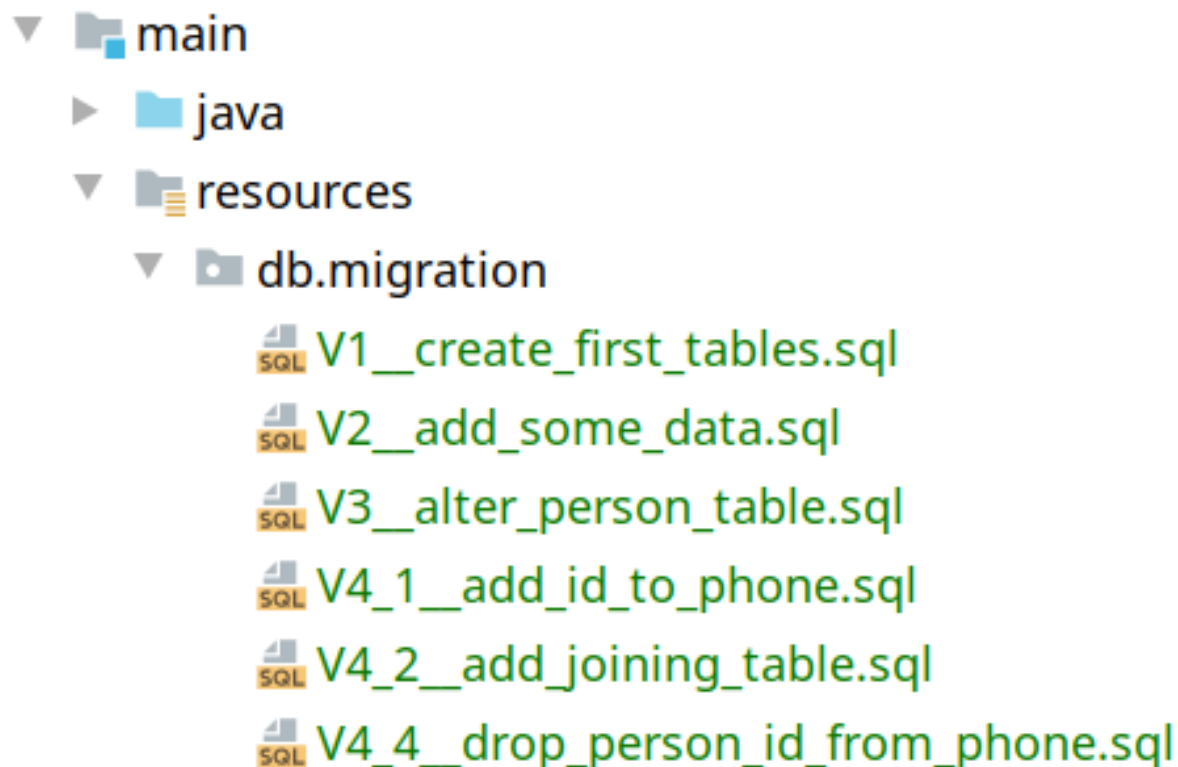
```
ALTER TABLE person  
    ADD COLUMN last_name VARCHAR(100);
```

# Probleemid

- Töömahukas ja habras, kui on palju keskondi
- Pole kindlat 1:1 vastavust arenduse baasi ja live baasi vahel

# Lahendus

- Kui live baasi ei saa uuest skeemist luua, siis võib arenduse keskonda muuta ka ALTER TABLE lausetega.



# Probleemid

- Eri keskkondades võivad olla erinevad versioonid
- Peab arvet pidama, millised ALTER laused on konkreetsetes baasis juba käima läinud
- Käsitsi arvepidamine on tülikas ja veaaldis



# Lahendus

- Rakendus hoiab baasis infot ka selle kohta, millised skriptid on juba käima läinud
- Rakenduse käivitamisel kontrollib rakendus, kas skripte on lisandunud ja kui on siis käivitab need

# Tööriistad

- Flyway
- Liquibase
- c5-db-migration
- dbdeploy
- ...

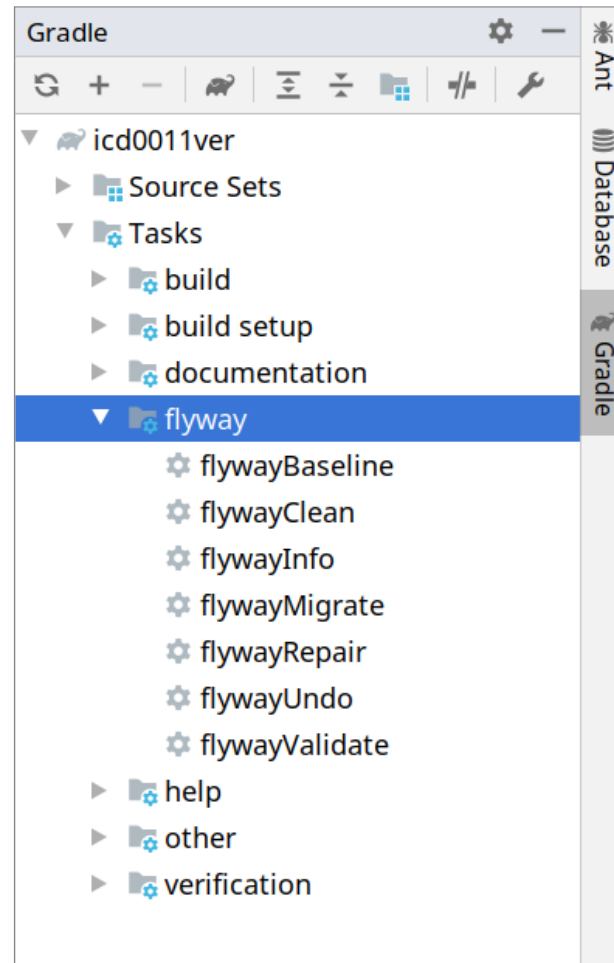
# Flyway

- Käsureda rakendus
- Gradle plugin
- Java teek

# Flyway Gradle plugin

```
buildscript {  
    dependencies {  
        classpath group: 'org.postgresql',  
                  name: 'postgresql',  
                  version: '42.2.18'  
    }  
}  
  
plugins {  
    id "org.flywaydb.flyway" version "7.1.1"  
}  
  
flyway {  
    url = 'jdbc:postgresql://db.mkalmo.xyz:5432/mkalmo'  
    user = 'mkalmo'  
    password = 's3cret'  
}
```

# Flyway Gradle plugin



# Flyway Gradle plugin

- > gradlew flywayInfo
- > gradlew flywayMigrate

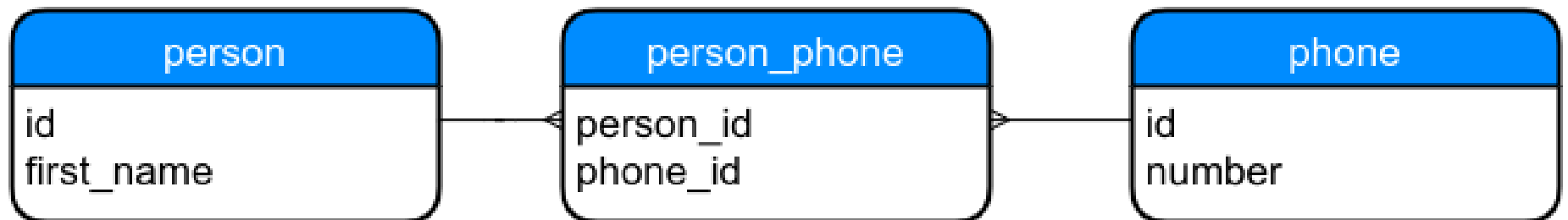
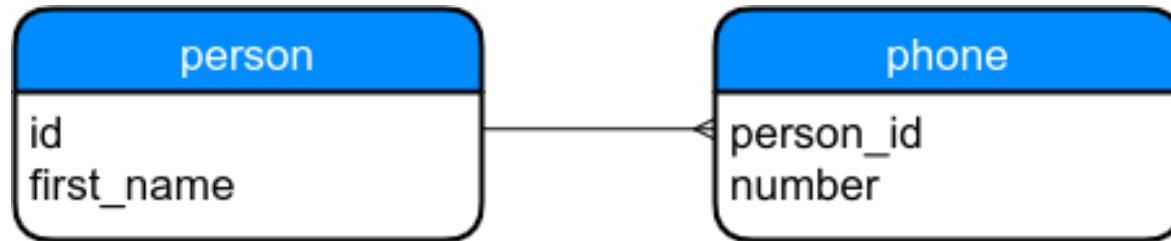
# Flyway Gradle plugin

- > gradlew **clean build** flywayInfo
- > gradlew **clean build** flywayMigrate -i

# Demo (Flyway)



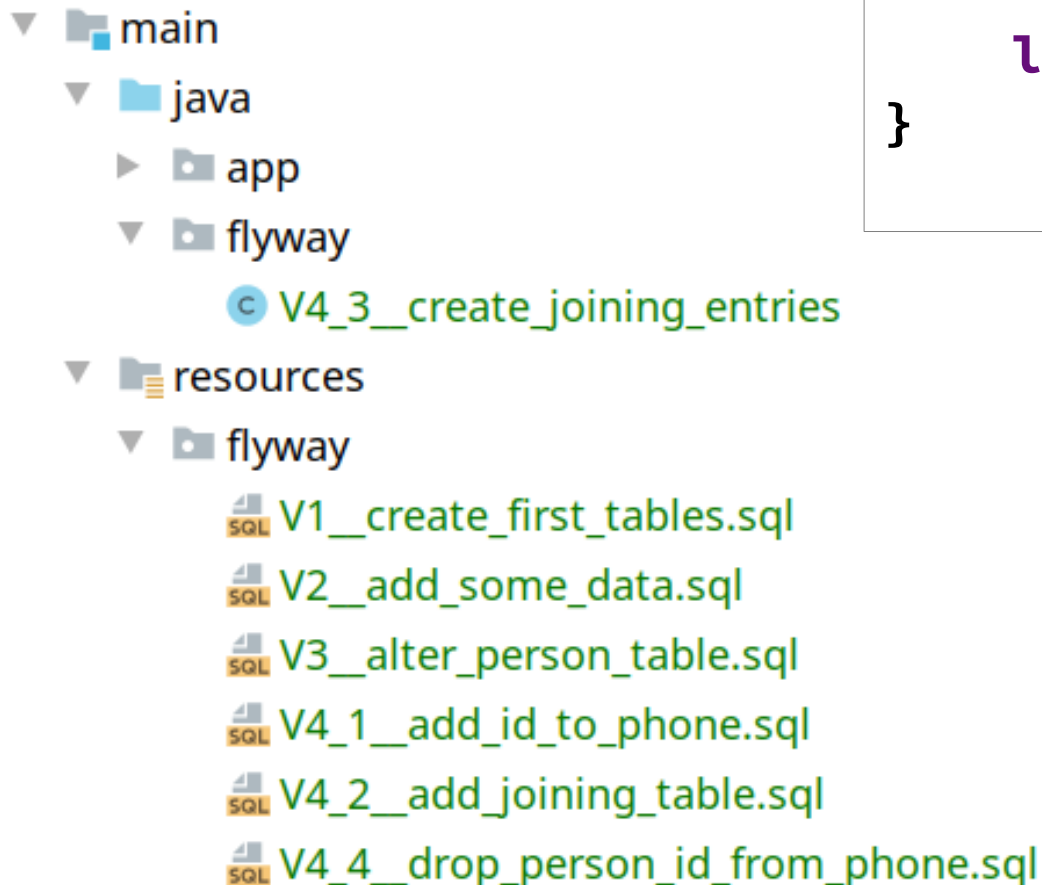
# Keerulisemad muudatused



# Keerulisemad muudatused

- Lisa id väli phone tabelisse
- Lisa ühendav tabel
- Lisa kirjed ühendavasse tabelisse
- Eemalda person\_id väli phone tabelist

# Koodi kasutamine



build.gradle

```
flyway {  
    ...  
    locations = ['classpath:flyway']  
}
```

# Koodi kasutamine

```
public class V4_3__create_joining_entries
    extends BaseJavaMigration {

    public void migrate(Context context) {

        var conn = context.getConnection();

        ...
    }
}
```

# JdbcTemplate-i kasutamine

...

```
public void migrate(Context context) {  
  
    var conn = context.getConnection();  
  
    var dataSource = new SingleConnectionDataSource(  
                                                                conn, true);  
  
    var template = new JdbcTemplate(dataSource);  
  
    ...  
}
```

...

# Liquibase

- Rollback toetus
- Diff toetus
- Konkreetsest baasist sõltumatuse võimalus

# Liquibase

- Konkreetsest baasist sõltumatuse võimalus

```
<changeSet id="1" author="bob">

  <createTable tableName="department">

    <column name="id" type="int">
      <constraints primaryKey="true" nullable="false"/>
    </column>

    <column name="name" type="varchar(50)">
      <constraints nullable="false"/>
    </column>

    <column name="active" type="boolean" defaultValueBoolean="true"/>

  </createTable>
</changeSet>
```

# Andmete versioneerimine



# Analoogia

- Git - lähtekoodi versioneerimine

# Turvalisus

- Ei saa teiste muudatusi märkamatult üle kirjutada

# Äriloogika

- Nt. arve koostamisel internetipaketi kasutuse algus
- Nt. klienditeenindaja soovib näha paketi muudatusi

# Primitiivne lahendus

- Muudatuste logimine
- Probleemid:
  - masinloetavus (nt. arve koostamine)
  - kasutajaliideses info näitamine

**BILLING ADRESS**

First name    Bojhan  
Last name    Henkes  
Phone number    +321-213-3124  
Company    Drupal

Street adress    Chambre 9th Ave  
City    Detriot  
Country    France  
Province    Bretagne  
Postal code    3124 TE

**SHIPPING ADRESS**

First name    Bojhan  
Last name    Henkes  
Phone number    +321-213-3124  
Company    Drupal

Street adress    Chambre 9th Ave  
City    Detriot  
Country    France  
Province    Bretagne  
Postal code    3124 TE

**► LOG**

TYPE	DATE	USER	CONTENT
User	Feb 24, 2010 - 19:42	<a href="#">Benjamin Douarie</a>	We contacted you by e-mail.
Admin	Feb 24, 2010 - 20:41	<a href="#">Benjamin Douarie</a>	Often forgets to pay because he is outside the country but will always pay so contact him by e-mail when needed.
Order	Feb 20, 2010 - 09:43	System	Order status changed from Payment recieved to completed.
Payment	Feb 20, 2010 - 09:42	System	Credit card payment for \$15.00 entered by Henk jan.

[Add new comment](#)

# Ajalootabelid

person
id
version
first_name
last_name

person_history
id
version
first_name
last_name

# Ajalootabelid (copy)

```
INSERT INTO orders_history(id, number, version)  
  SELECT id, number, version  
  FROM orders WHERE id = ?
```

# Ajalootabelid (lock)

```
SELECT * FROM orders  
WHERE id = ? FOR UPDATE
```

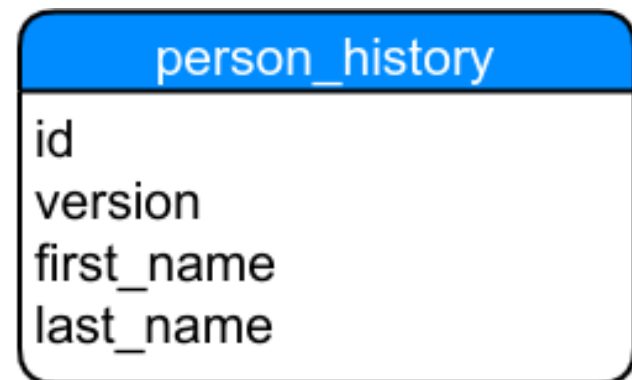
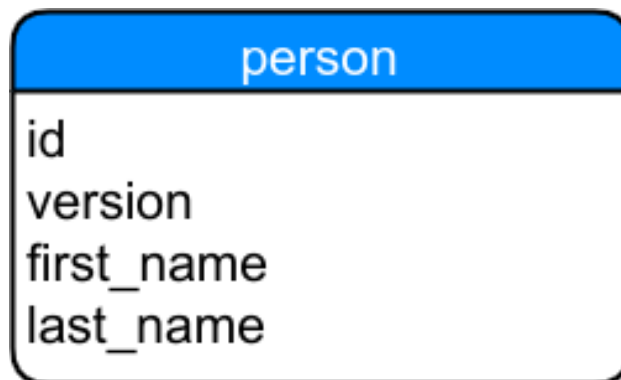


# Ajalootabelid (muutmine)

```
UPDATE orders  
    SET version = version + 1, ...  
WHERE id = ?
```

# Probleemid ajalootabelitega

- Välisvõtmete piirangud (constraint)
- Tabelite hulk saab kahekordne olema
- Skeemi muudatused tuleb teha mõlemasse tabelisse.
- Millisele kirjele viitavad seotud tabeli kirjed?



# Seotud tabeli kirjed

person	
id	name
1	Alice Smith
...	...

phones	
number	contact_id
123	1
456	1
789	9
...	...

# Seotud tabeli kirjed

person_history		
id	ver	name
1	1	Alice Smith
1	2	Alice Jones
1	3	Alice Brown
...	...	...

phones_history		
number	ver	contact_id
123	1	1
456	1	1
789	3	9
...	...	...

# Seotud tabeli kirjed

person_history		
id	ver	name
1	1	Alice Smith
1	2	Alice Jones
1	3	Alice Brown
...	...	...

phones_history		
number	ver	contact_id
123	1	1
456	1	1
789	3	9
...		...

# Seotud tabeli kirjed (päringud)

```
SELECT * FROM orders_history oh  
  LEFT JOIN order_rows_history orh  
    ON oh.id = orh.order_id AND oh.version = orh.version  
WHERE oh.id = ? AND oh.version = ?;
```

# ORM (Hibernate)

```
compile group: 'org.hibernate',  
        name: 'hibernate-envers',  
        version: '5.4.23.Final'
```

```
@Entity  
@org.hibernate.envers.Audited  
public class Person {  
  
    @Id  
    @GeneratedValue  
    private Long id;
```

# Demo (Envers)



# Ajalugu samas tabelis

person
id
first_name
last_name
start_date
end_date

# Ajalugu samas tabelis

person			
id	name	start_date	end_date
1	Alice Smith	2020-01-01	2020-11-01
1	Alice Jones	2020-11-01	null
...	...		

```
... WHERE start_date <= ?  
      AND (end_date > ? OR end_date IS NULL);
```

# Ajalugu samas tabelis (seotud kirjed)

phones			
person_id	number	start_date	end_date
1	123	2020-01-01	2020-11-01
1	456	2020-11-01	null
1	789	2020-01-01	null
...	...	...	...

# Ajalugu samas tabelis (seotud kirjed)

```
SELECT c.id, c.name, p.number
FROM contacts c
    LEFT JOIN phones p
        ON c.id = p.contact_id
            AND p.start_date <= ?
            AND (p.end_date > ?
                OR p.end_date IS NULL)
WHERE start_date <= ?
    AND (end_date > ?
        OR end_date IS NULL);
```

# Ajalugu samas tabelis

- Kaob ära tabelite hulga probleem
- Piisab ühe tabeli muutmisest
- Lahendub sõltuvate kirjete mure
- Tõuseb andmete kvaliteet
- Võimaldab hoida tulevikus kehtima hakkavaid andmeid

# Ajalugu samas tabelis

- Jääb välisvõtmete piirangute probleem
- Lisandub päringute keerukuse probleem

# Päringute keerukuse leevendamine

```
CREATE VIEW contacts_current AS  
    SELECT id, name, start_date, end_date  
    FROM contacts c where end_date IS NULL;
```

```
SELECT * FROM contacts_current;
```

# Päringute keerukuse leevendamine

- Sql Builder



# Sql Builder

**Search**

Name

E-mail

Phone

Internal Phone

Department

(all) ▼

☐ my office

☐ online

Search

Cancel

# Sql Builder

```
String sql = "SELECT id, fullName FROM customers";

if (name != null) {
    sql += " WHERE name = ?";
}

if (email != null) {
    sql += " LEFT JOIN email ON ...";
    sql += " WHERE email = ?";
}

...

db.query(sql, ...);
```

# Sql Builder

```
SelectBuilder b = new SelectBuilder();  
b.columns("first_name", "last_name");  
b.from("customers c");  
b.eqIfNotNull("name", name);  
  
if (email != null) {  
    b.leftJoin("email e", "c.id = e.customer_id");  
    b.where("e.value = ?", email);  
}  
  
db.query(b.getSql(), b.getParameters());
```

# Sql Builder

```
SelectBuilder b = new SelectBuilder();  
b.columns("first_name", "last_name");  
b.from("customers c");  
b.eqIfNotNull("name", name);  
b.setQueryDate(date);
```

...

# Kirjandus

- Developing Time-Oriented Database Applications in SQL (1999)

# Ajalugu samas tabelis probleemid

- Andmete hulk
- Piirang alamtabeli kirjete ülekirjutamisele
- Ebamugav arenduse ajal (nt. käsitsi päringute tegemine)
- ORM-i kasutus on keeruline
- Mõjutab oluliselt rakenduse arhitektuuri

# Mõjutab oluliselt rakenduse arhitektuuri

```
policy.add(insuredObject);  
policy.setCustomer(customer);  
policy.setStartDate(date)  
policy.calculateInstallments();
```

```
String sql = "...";
```

```
...
```

```
...
```

# Turvalisuse tõstmine

- Krüptoahela kasutamine

person				
id	name	start_date	end_date	hash
1	Alice Smith	2020-01-01	2020-11-01	9a1abc48d3...
1	Alice Jones	2020-11-01	null	5cdbbfbfcaa5...
...	...	...	...	...



# Millist lähenemist eelistada?

- Kui tähtis on ajalugu?
- Milleks ajalugu kasutatakse?
- Kui tihti ajalugu kasutatakse?
- Kas see peab olema kõigil tabelitel?
- Kui keeruline on rakenduse domeenimudel?
- Kui keeruline on rakenduse äriloogika?

# LAZY vs EAGER (JPA)

# LAZY probleemid

- Rohkem päringuid
- Probleemi ühenduse katkemisel

# EAGER probleemid

- Paindumatus
- Määramatus

**@Data**

**@Entity**

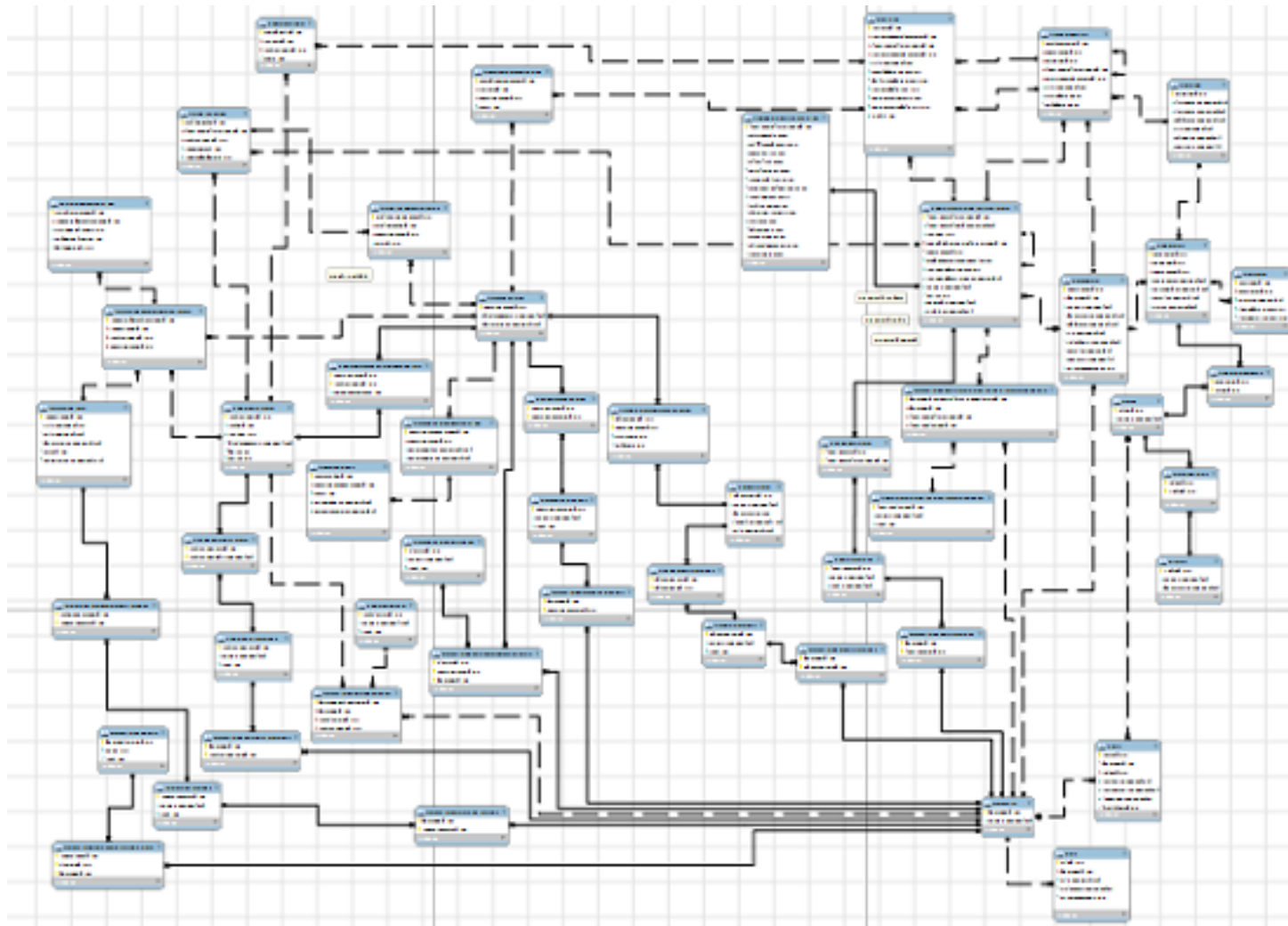
**public class** Person {

**@OneToMany**(fetch = FetchType.**EAGER**)

**private List**<Phone> **phones**;

```
List<Person> persons = em.createQuery(  
    "select p from Person p", Person.class)  
    .getResultList();
```

# EAGER



# Lahendus

- Kirjutada päringud ise

```
SELECT p FROM Person p LEFT JOIN FETCH p.phones
```

```
SELECT * FROM person  
LEFT JOIN phone on person.id = phone.person_id
```

# DISTINCT

```
SELECT DISTINCT p FROM Person p  
LEFT JOIN FETCH p.phones
```

```
SELECT * FROM person  
LEFT JOIN phone on person.id = phone.person_id  
WHERE person.id = 1;
```

# Kasu ORM-ist

- Kui kirjutatakse päringud käsitsi, kas siis üldse on asjal mõtet?



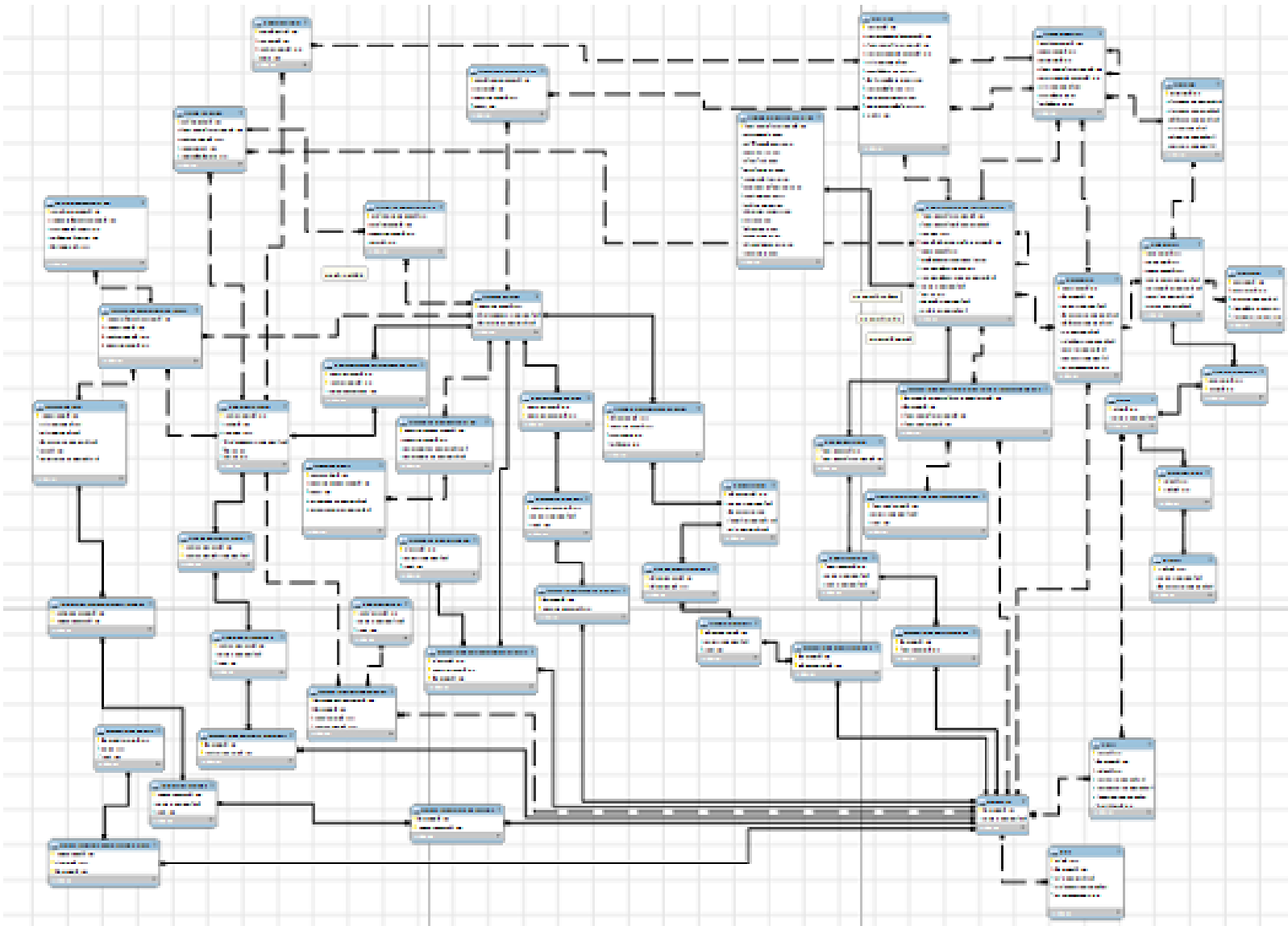
# Kasu ORM-ist

```
Policy policy = repository.load(policyId);
```

```
policy.add(insuredObject);  
policy.setCustomer(customer);  
policy.setStartDate(date)  
policy.calculateInstallments();  
...
```

```
repository.save(policy);
```

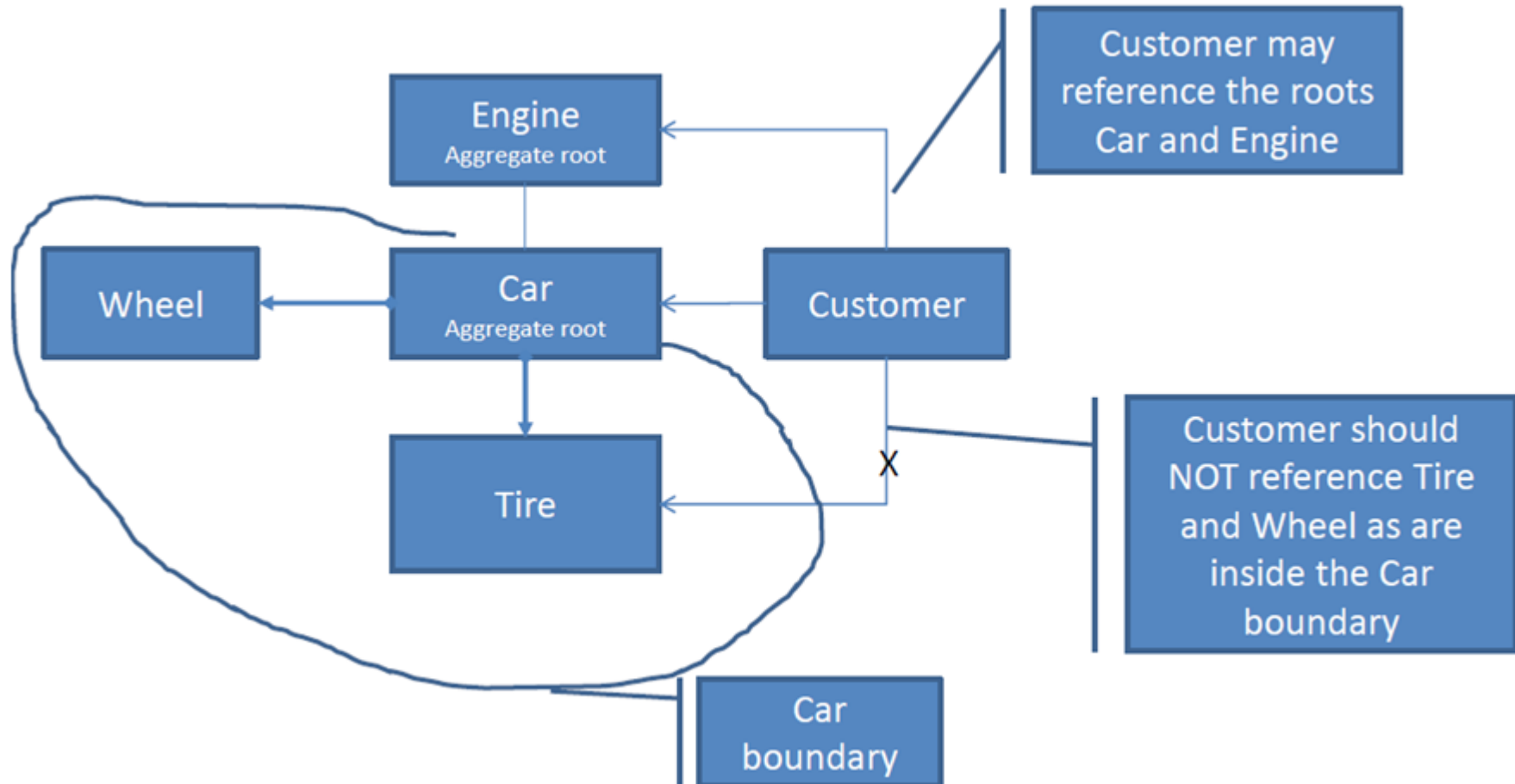
# ORM ja baasi disain



# Domain-Driven Design

- Domain-Driven Design: Tackling Complexity in the Heart of Software (2003)

# Aggregate Root



# Bounded Context

