
FIRST 8

Persistence

It’s the key to success

FIRST8

Even voorstellen

- Tobias Poll
- 33 jaar, Utrecht
- Cognitieve Kunstmatige Intelligentie
- EPD Systemen
- Java Ontwikkelaar bij First8
- Financiële systemen, OV-Chip
- Bier, toneel & yoga

FIRST8

Even voorstellen

- Frank de Jong
- 26 jaar
- Technische Informatica
- Java engineer @ First8
- Masters of Java 2016

FIRST8

Even voorstellen

- First8
- Conclusion

Waar gaat het over?

- JPA / Hibernate
- Relationele Databases
- Basis
- Best Practices
- Praktijkervaringen

Waar gaat het niet over?

- Spring Data
- Hippe nieuwe databases
- Alle SQL-smaken

Agenda

JPA

Entities

Create
Read
Update
Delete

Transacties

9:00 - 11:30
Workshop

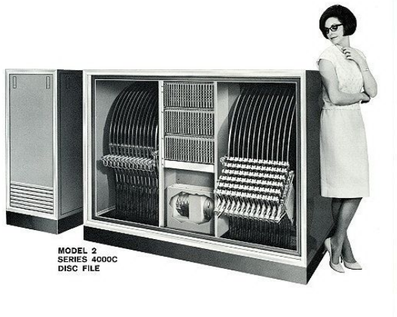
11:30
Nog meer persistency

Borrel?

Persistency

- Uitgevonden in 1540-1550
- Waarom?

Lang Leve de Data

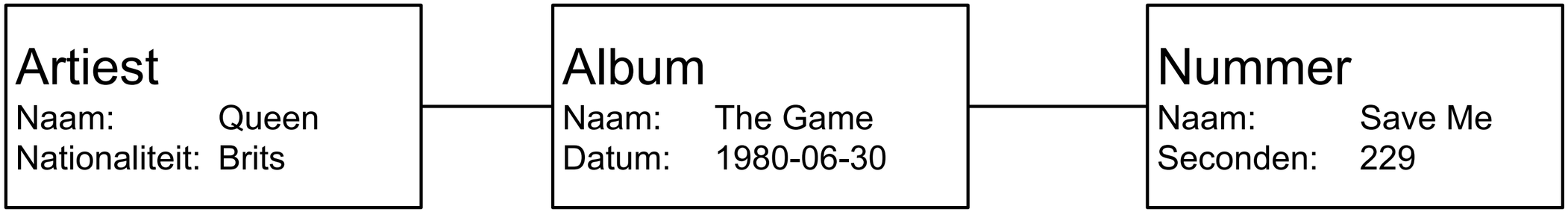


FIRST8

Databases

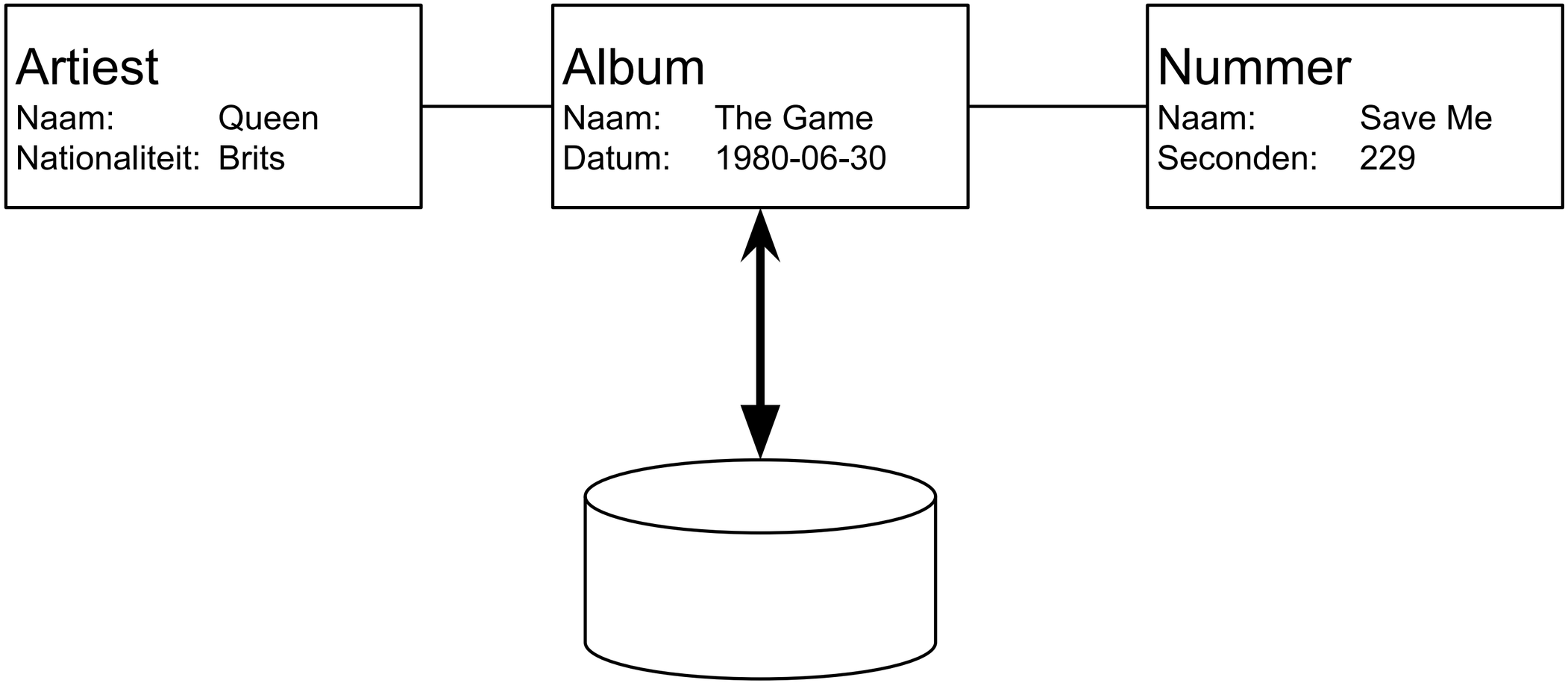
- Hierarchisch
- Netwerk
- Relationeel
- Object georiënteerd
- (en anderen)

Data Doe-het-zelven



FIRST8

Data Doe-het-zelven



FIRST8

Data Doe-het-zelven

```
public void insert(Connection connection, Artist  artist) throws SQLException {
   try (statement = connection.createStatement()) {
      String baseQuery= "INSERT INTO ARTIST (‘NAME’, ‘COUNTRY’)" +
                  "VALUES(‘%s’, ‘%s’)";
      String query = String.format(baseQuery, artist.getName(),
                  artist.getCountry());
      statement.executeUpdate(query);
   }
};
```

FIRST8

Data Doe-het-zelven

```
public Artist retrieve(Connection connection, String name) throws SQLException {
   Artist artist = new Artist();
   try (statement = connection.createStatement()) {
       String baseQuery= "SELECT * FROM ARTIST WHERE NAME = ‘%s’";
       String query = String.format(baseQuery, name);
       ResultSet rs = statement.executeQuery(query);
       resultSet.next();
       artist.setName(resultSet.getString(“NAME”)));
       artist.setCountry(resultSet.getString(“COUNTRY”)));
   }
  return artist;
};
```

FIRST8

Maar…

- Data types…
- Relaties…
- Object = Tabel?
- Foutafhandeling...
- Security...
- Transacties…
- Herbruik / Onderhoud
- Verschillen per database…

Er moet een betere manier zijn!

???

Er moet een betere manier zijn!

- JPA
- MyBatis
- Spring JDBC Template
- jOOQ

JPA in het Kort

- JSR 220
- Features
  - Object Mapping
  - Query Language
  - Criteria API
  - DDL Generation
  - Validation
  - Caching
- Vendors
  - Hibernate
  - The rest

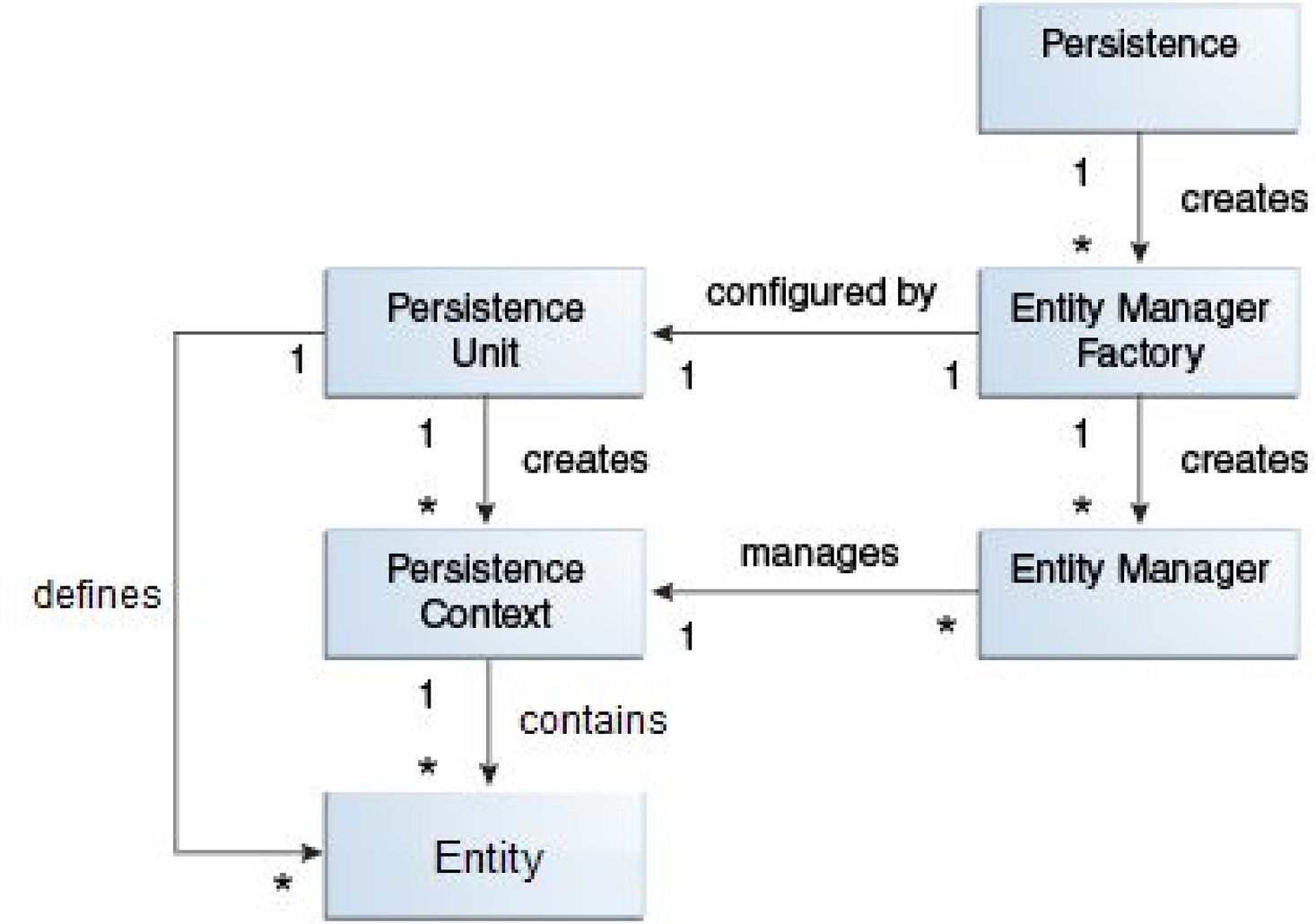
Annotaties vs XML

- Direct bij je code
- Korter
- Goede IDE ondersteuning

- Gescheiden van de code
- Makkelijker te hergebruiken
- Slechtere IDE ondersteuning

- Dus...

JPA - Persistence & Entities



Entity Manager

- Entiteitenbeheer (zoals de naam aangeeft)
- JPA Interface
- Managed Entities
- Wanneer naar de database?

Persist

- Zet in Persistence Context
- Maakt gegeven Entity Managed
- Naar database on commit/flush

Persist

- Zet in Persistence Context
- Maakt gegeven Entity Managed
- Naar database on commit/flush

Merge

- Kopieert naar Persistence Context
- Indien nieuw, maakt nieuw
- Geeft managed entity terug
- Naar database on commit/flush
- Zwaarder

Persist

- Zet in Persistence Context
- Maakt gegeven Entity Managed
- Naar database on commit/flush

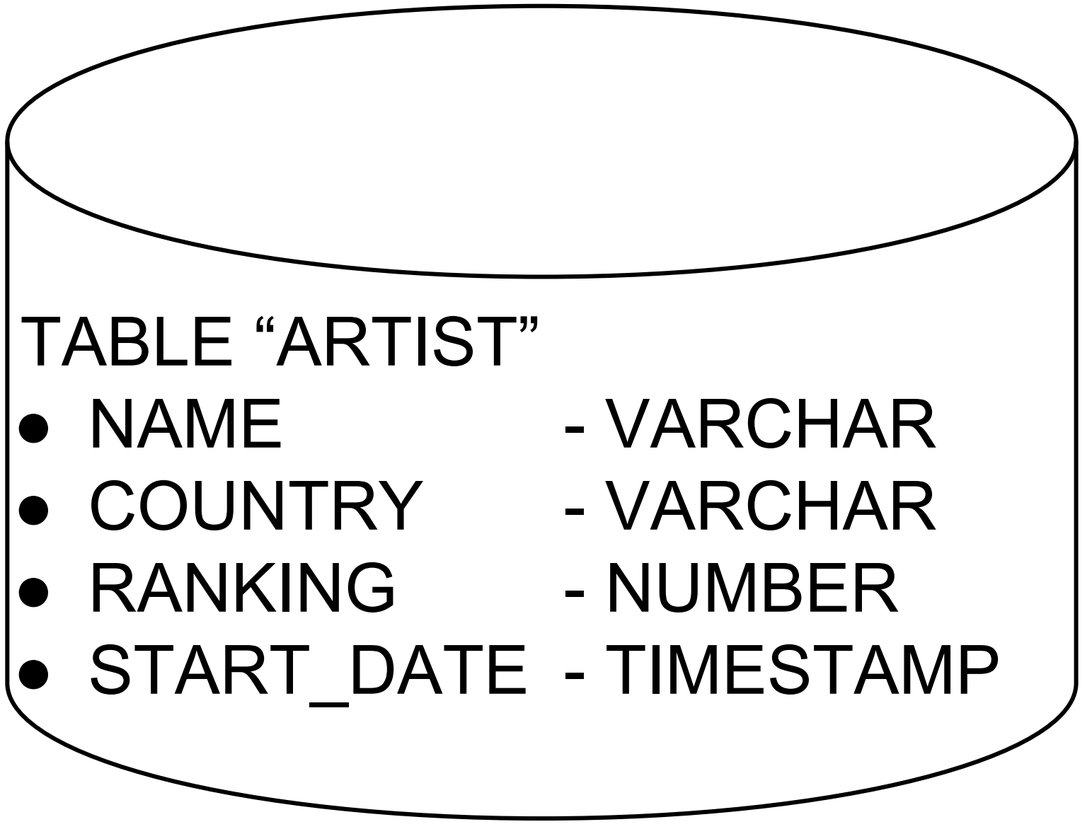
```
public void save(Entity e) {
    entityManager.persist(e);
    e.setName(“Nieuw”);
}
```

Merge

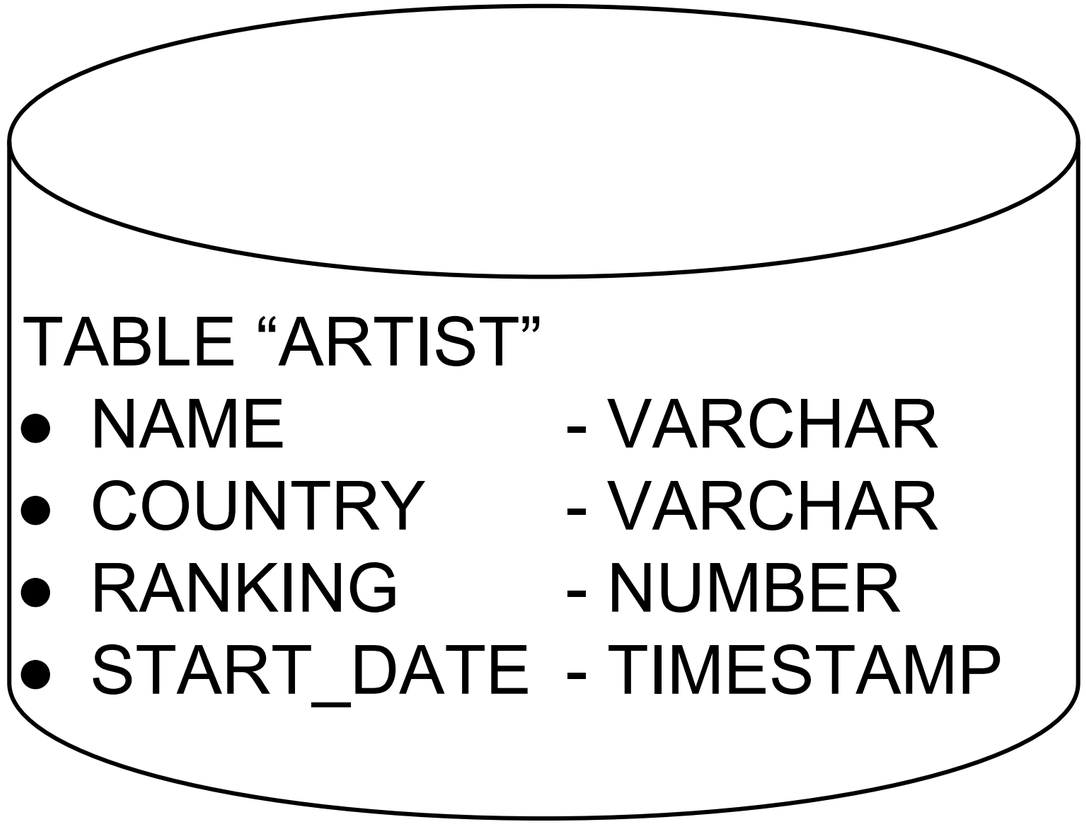
- Kopieert naar Persistence Context
- Indien nieuw, maakt nieuw
- Geeft managed entity terug
- Naar database on commit/flush
- Zwaarder

```
public void save(Entity e) {
    entityManager.merge(e);
    e.setName(“Nieuw”);
}
```

Entity Mapping



Entity Mapping



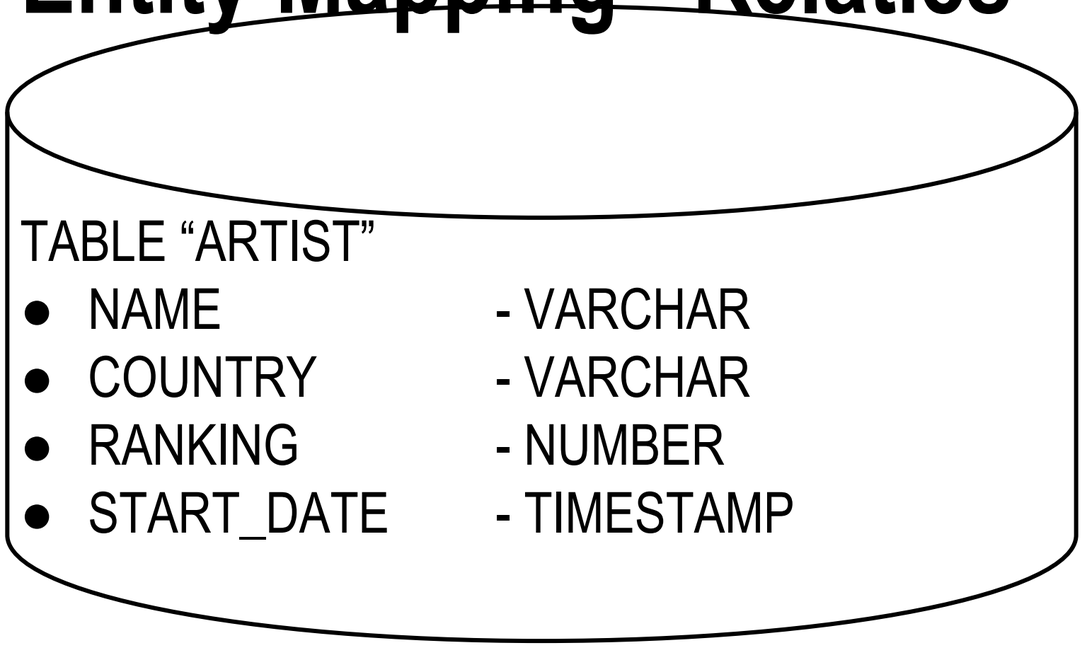

```
@Entity
@Table(name = “ARTIST”)
public class Artist {
    @Id
    @Column(name = “NAME”)
    private String name;

    @Column(name = “COUNTRY”)
    private String country;

    @Column(name = “RANKING”)
    private Integer ranking;

    @Column(name = “START_DATE”)
    private Date startDate;
}
```

FIRST8

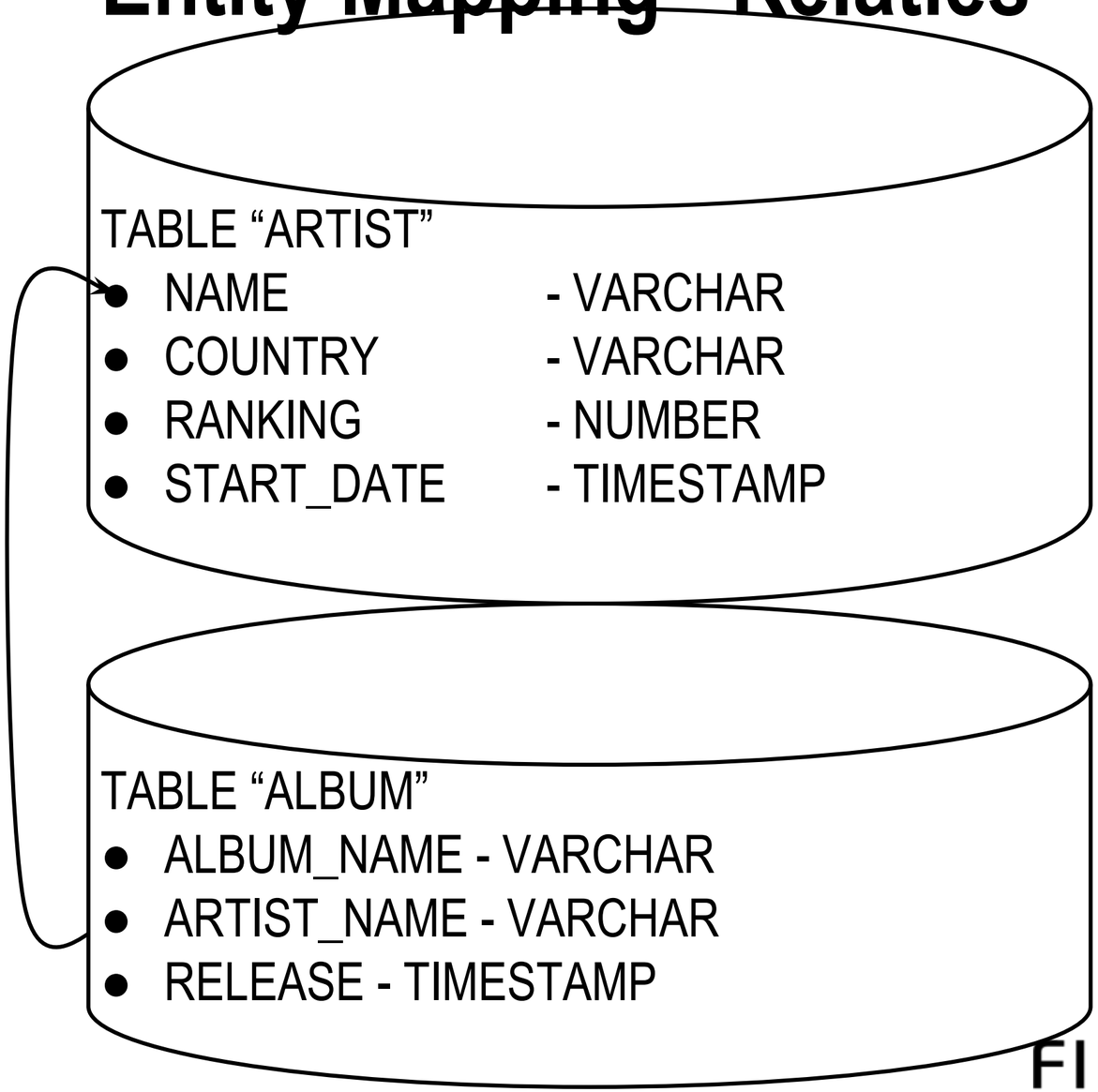
Entity Mapping - Relaties

- Concepts
  - Roles
  - Directionality
  - Cardinality
  - Ordinality
- Types
  - One to One
  - Many to One
  - One to Many
  - Many to Many

Entity Mapping - Relaties

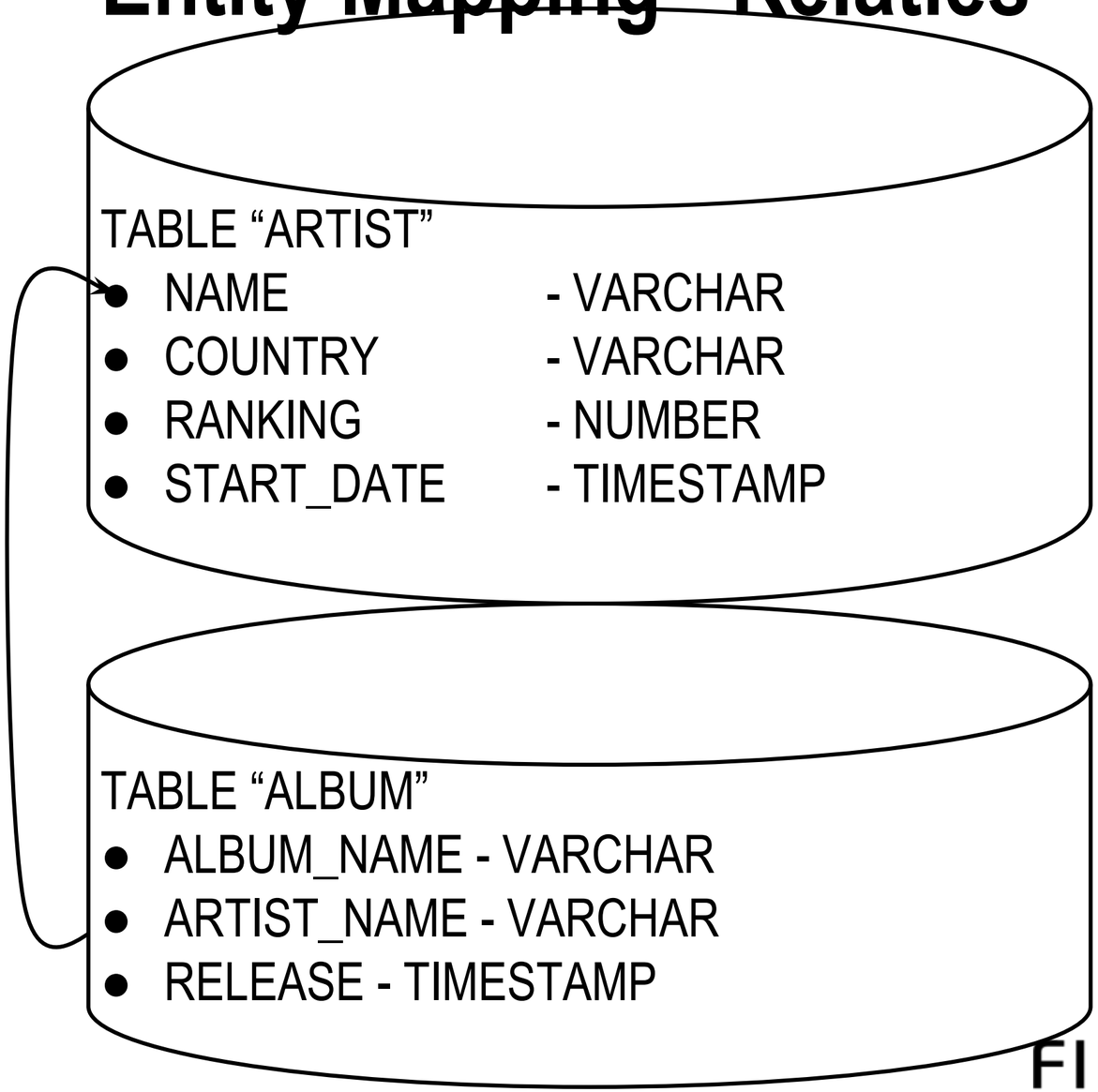


Entity Mapping - Relaties



FIRST8

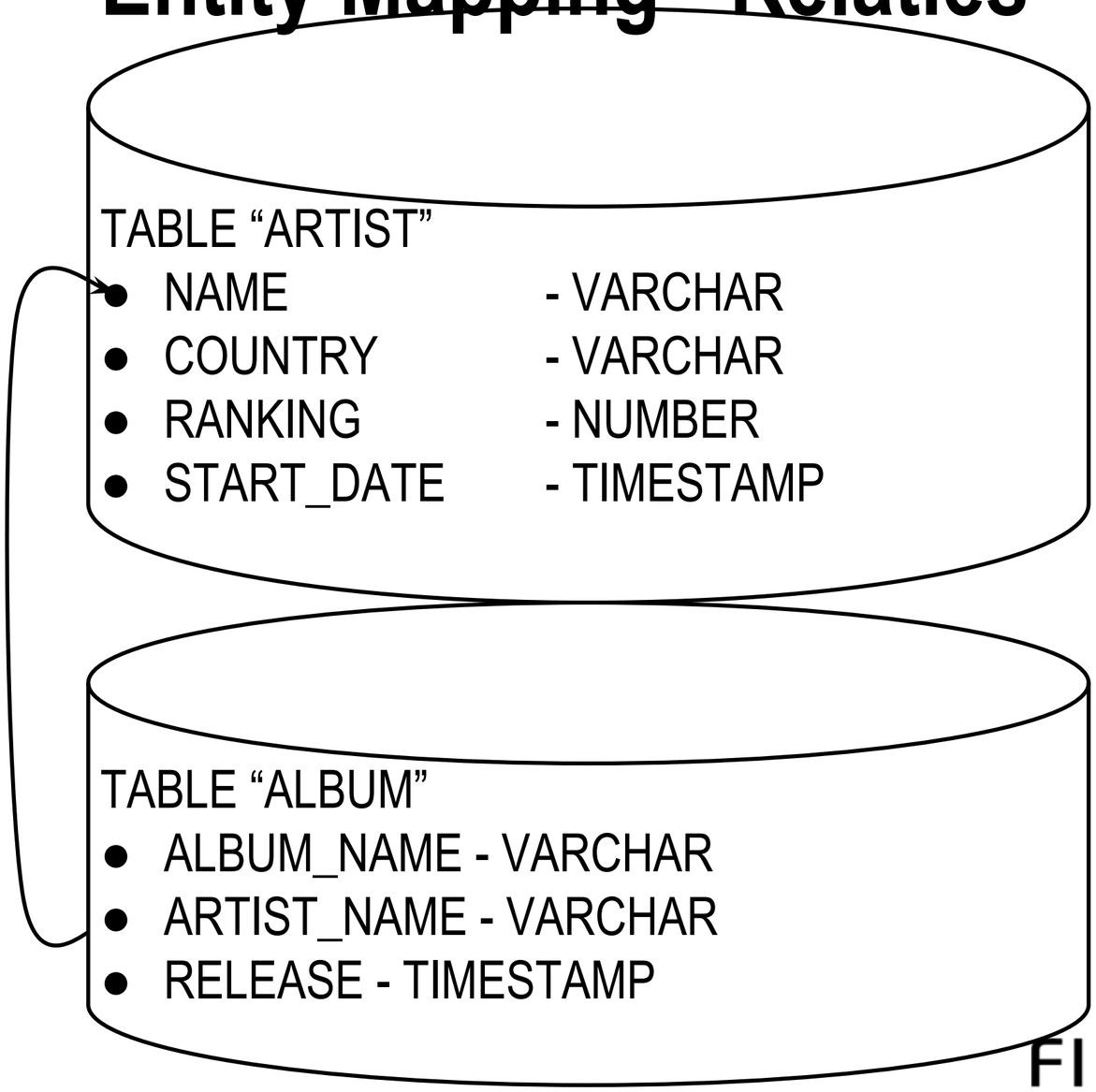
Entity Mapping - Relaties




```
@Entity
public class Album {
    @Id
    private String albumName;

    @JoinColumn(name = "ARTIST_NAME",
        referencedColumnName = "NAME")
    private Artist artist;

    private Date release;
}
```

FIRST8

Entity Mapping - Relaties




```
@Entity
public class Artist {
    @Id
    private String name;
    private String country;
    private Integer ranking
    private Date startDate;

    @OneToMany(mappedBy = "artist")
    private List<Album> albums;
}
```

```
@Entity
public class Album {
    @Id
    private String albumName;

    @JoinColumn(name = "ARTIST_NAME",
        referencedColumnName = "NAME")
    private Artist artist;

    private Date release;
}
```

FIRST8

Entity Mapping - Relatiebeheer

- Eager Fetching
- Lazy Fetching
- Cascades

Ophalen uit de database

- entityManager.find(Test.class, 1234);
- Query
  - JPQL
  - Native SQL
  - Direct of Named
- Criteria API

JPQL

- Query op entities
- SQL-esque
- Vertaald naar SQL in jouw smaak
- Typed
- Parameter binding

JPQL

- Query op entities
- SQL-esque
- Vertaald naar SQL in jouw smaak
- Typed
- Parameter binding

```
List<Test> results = entityManager.createQuery(
    "SELECT test FROM Test test WHERE test.price < :param”
      ).setParameter(“param”, 10).getResultList();
```

Criteria API

- Fluent interface
- Compile Safe
- Dynamische queries

Criteria API

- Fluent interface
- Compile Safe
- Dynamische queries

```
CriteriaBuilder criteriaBuilder = em.getCriteriaBuilder();
CriteriaQuery<Test> criteriaQuery = criteriaBuilder.createQuery(Test.class);
Root<Test> from = criteriaQuery.from(Test.class);
CriteriaQuery<Test> select = criteriaQuery.select(from);
TypedQuery<Test> query = em.createQuery(select);
List<Test> results = query.getResultList();
```

Update

- entityManager.merge
- entityManager.persist
- JPQL/SQL Query

Delete

- entityManager.remove
- JPQL/SQL Query

Transacties

- Unit of work
- Commits
- Rollbacks

- REQUIRED
- REQUIRES_NEW
- NEVER
- MANDATORY
- SUPPORTS
- NOT_SUPPORTED

Configuratie

- Java EE vs Java SE
- EntityManagerFactory
- persistence.xml

Validatie

- Constraints
- Validators
- Groups
- Lifecycle

Workshop!

- Optional (empty(), of(), ofNullable()
- Lombok (@Data, @NoArgsConstructor)
- Docker (docker-compose)

FIRST8


FIRST 8