

# Define Custom JPQL and Native SQL Queries

If your query uses more than 2-3 query parameters or needs to define multiple joins to other entities, it becomes too complex for a derived query. You should define a custom query instead.

## Define a Custom JPQL Query

You can define a custom JPQL query by annotating your repository method with a `@Query` annotation and providing a JPQL statement as its *value*.

```
@Query("SELECT p FROM ChessPlayer p JOIN p.tournaments t "
      + "WHERE p.lastName LIKE :lastName "
      + "AND p.firstName LIKE :firstName AND t.name = :tournamentName "
      + "ORDER BY p.lastName ASC, p.firstName ASC")
List<ChessPlayer> findByNameAndTournament(String lastName,
                                         String firstName,
                                         String tournamentName);
```

## Define a Custom Native SQL Query

The JPA API requires slightly different handling of native SQL queries. You, therefore, need to mark a query as a native statement. You do that by setting the *nativeQuery* attribute of the `@Query` annotation to *true*.

```
@Query(value = "SELECT * FROM Chess_Player p "
      + "LEFT JOIN Chess_Tournament_Players tp ON p.id = tp.players_id "
      + "LEFT JOIN Chess_Tournament t ON tp.tournaments_id = t.id "
      + "WHERE p.last_name LIKE %:lastName% "
      + "AND p.first_name LIKE %:firstName% AND t.name = :tournamentName "
      + "ORDER BY p.last_name ASC, p.first_name ASC",
      nativeQuery = true)
List<ChessPlayer> findByNameAndTournamentOrderedAdvancedLike(String lastName,
                                                             String firstName,
                                                             String tournamentName);
```

# Define Custom JPQL and Native SQL Queries

## Advanced Like Expressions

To make using *LIKE* expressions a little easier, Spring Data JPA allows you to add the required % characters to your query statement. It then removes them from the query and adds them to the bind parameter value.

```
@Query("SELECT p FROM ChessPlayer p "  
      + "WHERE p.lastName LIKE %:lastName% AND p.firstName LIKE %:firstName% "  
      + "ORDER BY p.lastName ASC, p.firstName ASC")  
List<ChessPlayer> findByNameAndTournamentOrderedAdvancedLike(String lastName,  
                                                             String firstName);
```

You can use this feature with custom JPQL and custom native SQL queries.