    /\*\*

     \* Finds persons by using the last name as a search criteria.

     \* @param lastName

     \* @return  A list of persons which last name is an exact match with the given last name.

     \*          If no persons is found, this method returns an empty list.

     \*/

    public List<Person> findByLastNameAndCity(String lastName, String city);

    /\*\*

     \* Returns the found todo entry whose title or description is given

     \* as a method parameter. If no todo entry is found, this method

     \* returns an empty list.

     \*/

    public List<Todo> findByTitleOrDescription(String title, String description);

    /\*\*

     \* Returns the number of todo entry whose title is given

     \* as a method parameter.

     \*/

    public long countByTitle(String title);

   /\*\*

     \* Returns the distinct todo entries whose title is given

     \* as a method parameter. If no todo entries is found, this

     \* method returns an empty list.

     \*/

    public List<Todo> findDistinctByTitle(String title);

    /\*\*

     \* Returns the first three todo entries whose title is given

     \* as a method parameter. If no todo entries is found, this

     \* method returns an empty list.

     \*/

    public List<Todo> findFirst3ByTitleOrderByTitleAsc(String title);

    /\*\*

     \* Returns the first three todo entries whose title is given

     \* as a method parameter. If no todo entries is found, this

     \* method returns an empty list.

     \*/

    public List<Todo> findTop3ByTitleOrderByTitleAsc(String title);

  List<Todo> findByDescriptionContainsOrTitleContainsAllIgnoreCase(String descriptionPart,

                                                                     String titlePart);

//JPQL query

    @Query("SELECT t FROM Todo t WHERE t.title = 'title'")

    public List<Todo> findById();

//JPQL Query

@Query(value="select U from User U where U.country=?1")

List<User> findAbc(String country);

//JPQL Named parameters

@Query("SELECT u FROM User u WHERE u.status = :pstatus and u.name = :pname")

User findUserByStatusAndNameNamedParams(

@Param("pstatus") Integer status,

@Param("pname") String name);

//Native query. below

    @Query(value = "SELECT \* FROM todos t WHERE t.title = 'title'",

            nativeQuery=true

    )

    public List<Todo> findByTitle();

    @Query("SELECT t FROM Todo t WHERE " +

            "LOWER(t.title) LIKE LOWER(CONCAT('%',:searchTerm, '%')) OR " +

            "LOWER(t.description) LIKE LOWER(CONCAT('%',:searchTerm, '%'))")

    List<Todo> findBySearchTerm(@Param("searchTerm") String searchTerm);

@Query(value="select U.country as country, U.name as name from user\_info U", nativeQuery=true)

List<Map<String, Object>> findMno();

@Query(value="select U.country as country, U.name as name from user\_info U", nativeQuery=true)

List<Object[]> findMno();