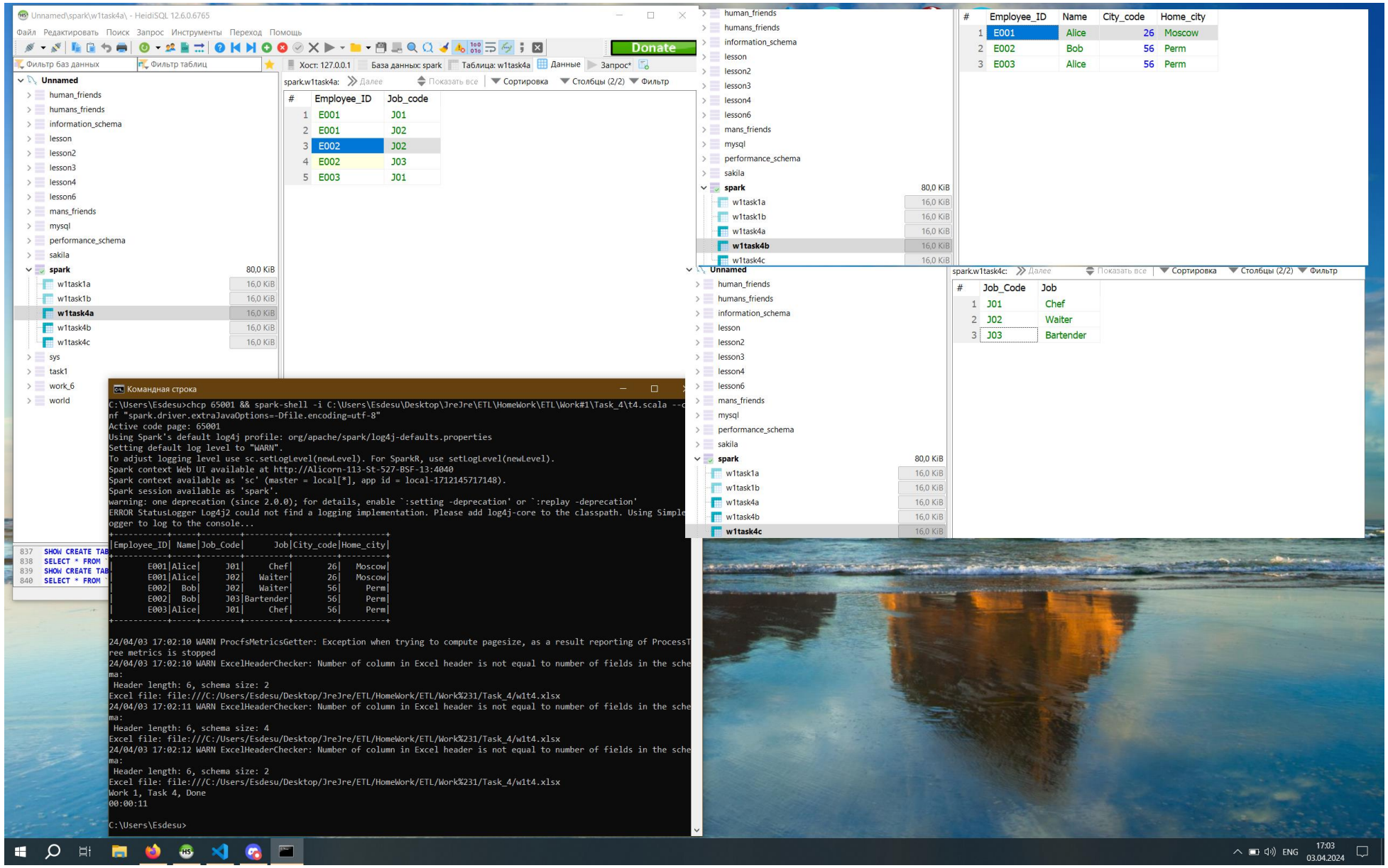
4. Определите в какой нормальной форме данная таблица, приведите ее ко 2 и 3 нормальным формам последовательно.
Нужно в apache spark создать таблицу сданными ниже. Можно импортировать из Excel либо с генерировать ее кодом.

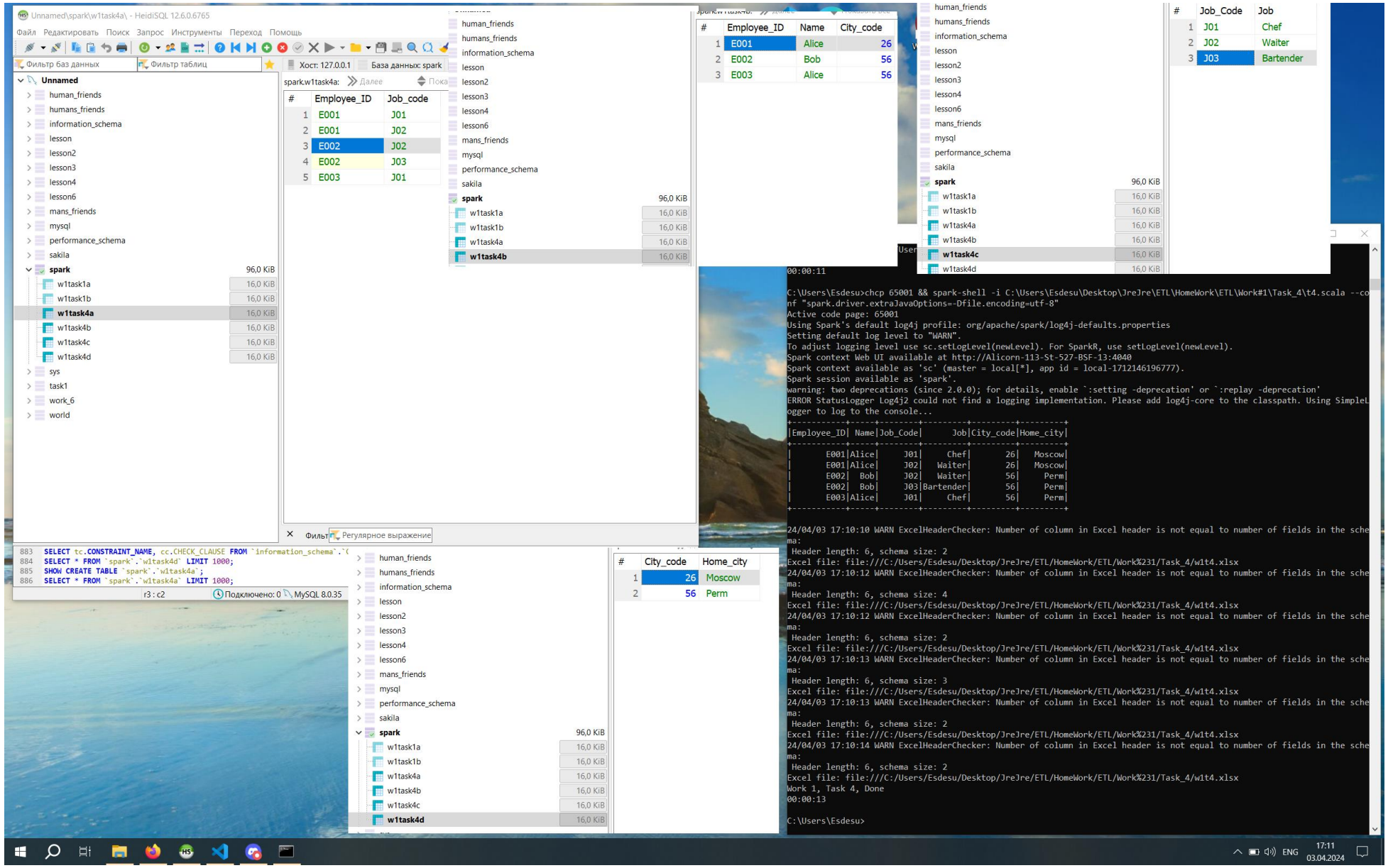
Сделать нормализацию по образцу
(например, как здесь https://ru.wikipedia.org/wiki/%D0%92%D1%82%D0%BE%D1%80%D0%B0%D1%8F_%D0%BD%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0).

Сохранить полученный результат в БД. На проверку соберите код, скриншоты отработки консоли спарка и полученный результат в БД в один pdf файл.

Employee_ID	Name	Job_Code	Job	City_code	Home_city
E001	Alice	J01	Chef	26	Moscow
E001	Alice	J02	Waiter	26	Moscow
E002	Bob	J02	Waiter	56	Perm
E002	Bob	J03	Bartender	56	Perm
E003	Alice	J01	Chef	56	Perm

Представленная таблица находится в форме NF1, далее переводим её к NF2 и NF3 последовательно.


Unnamed\spark\w1task4a\ - HeidiSQL 12.6.0.6765
Файл Редактировать Поиск Запрос Инструменты Переход Помощь
Donate
Unnamed
human_friends
humans_friends
information_schema
lesson
lesson2
lesson3
lesson4
lesson6
mans_friends
mysql
performance_schema
sakila
spark 80,0 KiB
w1task1a 16,0 KiB
w1task1b 16,0 KiB
w1task4a 16,0 KiB
w1task4b 16,0 KiB
w1task4c 16,0 KiB
sys
task1
work_6
world
Employee_ID Job_code
E001 J01
E001 J02
E002 J02
E002 J03
E003 J01
Employee_ID Name City_code Home_city
E001 Alice 26 Moscow
E002 Bob 56 Perm
E003 Alice 56 Perm
Job_Code Job
J01 Chef
J02 Waiter
J03 Bartender
Командная строка
C:\Users\Esdesu>chcp 65001 && spark-shell -i C:\Users\Esdesu\Desktop\JreJre\ETL\HomeWork\ETL\Work#1\Task_4\t4.scala --conf "spark.driver.extraJavaOptions=-Dfile.encoding=utf-8"
Active code page: 65001
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://Alicorn-113-St-527-BSF-13:4040
Spark context available as 'sc' (master = local[*], app id = local-1712145717148).
Spark session available as 'spark'.
warning: one deprecation (since 2.0.0); for details, enable `:setting -deprecation' or `:replay -deprecation'
ERROR StatusLogger Log4j2 could not find a logging implementation. Please add log4j-core to the classpath. Using SimpleLogger to log to the console...
|Employee_ID| Name|Job_Code| Job|City_code|Home_city|
E001	Alice	J01	Chef	26	Moscow
E001	Alice	J02	Waiter	26	Moscow
E002	Bob	J02	Waiter	56	Perm
E002	Bob	J03	Bartender	56	Perm
E003	Alice	J01	Chef	56	Perm
24/04/03 17:02:10 WARN ProcfsMetricsGetter: Exception when trying to compute pagesize, as a result reporting of ProcessTree metrics is stopped
24/04/03 17:02:10 WARN ExcelHeaderChecker: Number of column in Excel header is not equal to number of fields in the schema:
Header length: 6, schema size: 2
Excel file: file:///C:/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work%231/Task_4/w1t4.xlsx
24/04/03 17:02:11 WARN ExcelHeaderChecker: Number of column in Excel header is not equal to number of fields in the schema:
Header length: 6, schema size: 4
Excel file: file:///C:/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work%231/Task_4/w1t4.xlsx
24/04/03 17:02:12 WARN ExcelHeaderChecker: Number of column in Excel header is not equal to number of fields in the schema:
Header length: 6, schema size: 2
Excel file: file:///C:/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work%231/Task_4/w1t4.xlsx
Work 1, Task 4, Done
00:00:11
C:\Users\Esdesu>
837 SHOW CREATE TAB
838 SELECT * FROM
839 SHOW CREATE TAB
840 SELECT * FROM
ENG 17:03 03.04.2024


Unnamed\spark\w1task4a\ - HeidiSQL 12.6.0.6765
Файл Редактировать Поиск Запрос Инструменты Переход Помощь
Фильтр баз данных
Фильтр таблиц
Unnamed
human_friends
humans_friends
information_schema
lesson
lesson2
lesson3
lesson4
lesson6
mans_friends
mysql
performance_schema
sakila
spark 96,0 KiB
w1task1a 16,0 KiB
w1task1b 16,0 KiB
w1task4a 16,0 KiB
w1task4b 16,0 KiB
w1task4c 16,0 KiB
w1task4d 16,0 KiB
sys
task1
work_6
world
Хост: 127.0.0.1
База данных: spark
spark.w1task4a:
Employee_ID Job_code
1 E001 J01
2 E001 J02
3 E002 J02
4 E002 J03
5 E003 J01
Employee_ID Name City_code
1 E001 Alice 26
2 E002 Bob 56
3 E003 Alice 56
Job_Code Job
1 J01 Chef
2 J02 Waiter
3 J03 Bartender
00:00:11
C:\Users\Esdesu>chcp 65001 && spark-shell -i C:\Users\Esdesu\Desktop\JreJre\ETL\HomeWork\ETL\Work#1\Task_4\t4.scala --conf "spark.driver.extraJavaOptions=-Dfile.encoding=utf-8"
Active code page: 65001
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://Alicorn-113-St-527-BSF-13:4040
Spark context available as 'sc' (master = local[*], app id = local-1712146196777).
Spark session available as 'spark'.
warning: two deprecations (since 2.0.0); for details, enable `:setting -deprecation' or `:replay -deprecation'
ERROR StatusLogger Log4j2 could not find a logging implementation. Please add log4j-core to the classpath. Using SimpleLogger to log to the console...
|Employee_ID| Name|Job_Code| Job|City_code|Home_city|
E001	Alice	J01	Chef	26	Moscow
E001	Alice	J02	Waiter	26	Moscow
E002	Bob	J02	Waiter	56	Perm
E002	Bob	J03	Bartender	56	Perm
E003	Alice	J01	Chef	56	Perm
24/04/03 17:10:10 WARN ExcelHeaderChecker: Number of column in Excel header is not equal to number of fields in the schema:
Header length: 6, schema size: 2
Excel file: file:///C:/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work%231/Task_4/w1t4.xlsx
24/04/03 17:10:12 WARN ExcelHeaderChecker: Number of column in Excel header is not equal to number of fields in the schema:
Header length: 6, schema size: 4
Excel file: file:///C:/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work%231/Task_4/w1t4.xlsx
24/04/03 17:10:12 WARN ExcelHeaderChecker: Number of column in Excel header is not equal to number of fields in the schema:
Header length: 6, schema size: 2
Excel file: file:///C:/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work%231/Task_4/w1t4.xlsx
24/04/03 17:10:13 WARN ExcelHeaderChecker: Number of column in Excel header is not equal to number of fields in the schema:
Header length: 6, schema size: 3
Excel file: file:///C:/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work%231/Task_4/w1t4.xlsx
24/04/03 17:10:13 WARN ExcelHeaderChecker: Number of column in Excel header is not equal to number of fields in the schema:
Header length: 6, schema size: 2
Excel file: file:///C:/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work%231/Task_4/w1t4.xlsx
24/04/03 17:10:14 WARN ExcelHeaderChecker: Number of column in Excel header is not equal to number of fields in the schema:
Header length: 6, schema size: 2
Excel file: file:///C:/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work%231/Task_4/w1t4.xlsx
Work 1, Task 4, Done
00:00:13
C:\Users\Esdesu>
SELECT tc.CONSTRAINT_NAME, cc.CHECK_CLAUSE FROM `information_schema`.`
SELECT * FROM `spark`.`w1task4d` LIMIT 1000;
SHOW CREATE TABLE `spark`.`w1task4a`;
SELECT * FROM `spark`.`w1task4a` LIMIT 1000;
r3 : c2
Подключено: 0
MySQL 8.0.35
Регулярное выражение
City_code Home_city
1 26 Moscow
2 56 Perm
ENG
17:11
03.04.2024

```scala
/*
chcp 65001 && spark-shell -i C:\Users\Esdesu\Desktop\JreJre\ETL\HomeWork\ETL\Work#1\Task_4\t4.scala --conf
"spark.driver.extraJavaOptions=-Dfile.encoding=utf-8"
*/

import org.apache.spark.internal.Logging
import org.apache.spark.sql.functions.{col, collect_list, concat_ws}
import org.apache.spark.sql.{DataFrame, SparkSession}
import org.apache.spark.sql.expressions.Window

val t1 = System.currentTimeMillis()
val sqlCoun = "jdbc:mysql://localhost:3306/spark?user=root&password="
val driver = "com.mysql.cj.jdbc.Driver"

if(1==1){
    var df1 = spark.read.format("com.crealytics.spark.excel")
            .option("sheetName", "Sheet1")
            .option("useHeader", "false")
            .option("treatEmptyValuesAsNulls", "false")
            .option("inferSchema", "true").option("addColorColumns", "true")
            .option("usePlainNumberFormat","true")
            .option("startColumn", 0)
            .option("endColumn", 99)
            .option("timestampFormat", "MM-dd-yyyy HH:mm:ss")
            .option("maxRowsInMemory", 20)
            .option("excerptSize", 10)
            .option("header", "true")
            .format("excel")
            .load("C:/Users/Esdesu/Desktop/JreJre/ETL/HomeWork/ETL/Work#1/Task_4/w1t4.xlsx")
            df1.show()

            df1.filter(col("Employee_ID").isNotNull).select("Employee_ID", "Job_code")
```

```
            .write.format("jdbc").option("url", sqlCoun)
            .option("driver", driver).option("dbtable", "w1task4a")
            .mode("overwrite").save()

            val nf2 = Window.partitionBy(lit(1)).orderBy(("id")).rowsBetween(Window.unboundedPreceding, Window.currentRow)

            val df2 = df1.withColumn("id", monotonicallyIncreasingId())

            df2.withColumn("Employee_ID", when(col("Employee_ID").isNull, last("Employee_ID", ignoreNulls =
true).over(nf2)).otherwise(col("Employee_ID")))
            .withColumn("table", lit("w1task4b"))

            .orderBy("id").drop("id", "Job_Code", "Job")
            .filter(col("table") === "w1task4b")
            .dropDuplicates()
            .drop("table")
            .write.format("jdbc").option("url", sqlCoun)
            .option("driver", driver).option("dbtable", "w1task4b")
            .mode("overwrite").save()

            df2.withColumn("table", lit("w1task4c"))
            .orderBy("id").drop("id", "Employee_ID", "Name", "City_code", "Home_city")
            .filter(col("table") === "w1task4c")
            .dropDuplicates()
            .drop("table")
            .write.format("jdbc").option("url", sqlCoun)
            .option("driver", driver).option("dbtable", "w1task4c")
            .mode("overwrite").save()

            val nf3 = Window.partitionBy(lit(1)).orderBy(("id")).rowsBetween(Window.unboundedPreceding, Window.currentRow)

            val df3 = df1.withColumn("id", monotonicallyIncreasingId())
```

```
            df3.withColumn("Employee_ID", when(col("Employee_ID").isNull, last("Employee_ID", ignoreNulls =
true).over(nf2)).otherwise(col("Employee_ID")))
            .withColumn("table", lit("w1task4b"))

            .orderBy("id").drop("id", "Job_Code", "Job", "Home_city")
            .filter(col("table") === "w1task4b")
            .dropDuplicates()
            .drop("table")
            .write.format("jdbc").option("url", sqlCoun)
            .option("driver", driver).option("dbtable", "w1task4b")
            .mode("overwrite").save()

            df3.withColumn("table", lit("w1task4c"))
            .orderBy("id").drop("id", "Employee_ID", "Name", "City_code", "Home_city")
            .filter(col("table") === "w1task4c")
            .dropDuplicates()
            .drop("table")
            .write.format("jdbc").option("url", sqlCoun)
            .option("driver", driver).option("dbtable", "w1task4c")
            .mode("overwrite").save()

            df3.withColumn("table", lit("w1task4d"))
            .orderBy("id").drop("id", "Employee_ID", "Name", "Job_Code", "Job")
            .filter(col("table") === "w1task4d")
            .dropDuplicates()
            .drop("table")
            .write.format("jdbc").option("url", sqlCoun)
            .option("driver", driver).option("dbtable", "w1task4d")
            .mode("overwrite").save()

            println("Work 1, Task 4, Done")
```

```
}

val s0 = (System.currentTimeMillis() - t1)/1000
val s = s0 % 60
val m = (s0/60) % 60
val h = (s0/60/60) % 24
println("%02d:%02d:%02d".format(h, m, s))
System.exit(0)
```