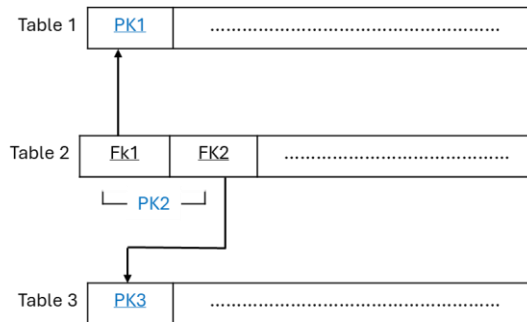**Assignment 3**

Design a database schema of your choice consisting of 3 relations. The schema diagram must show the attributes, primary, and foreign keys. The DB schema should match the following pattern:

Briefly describe your database. Show any potential considerations regarding the data in the tables that would impact fragmentation correctness, specifically the data in Table 2.

Q1: Apply a horizontal fragmentation on Table 1 and decide on predicates (exactly 2) for horizontal fragmentation. Show the completeness and disjointness of your fragments.

Q2: Apply a derived fragmentation on Table 2 based on the horizontal fragmentation of Table 1. Explain how you performed the derived fragmentation. Show the completeness and disjointness of your fragments.

Q3: Apply horizontal fragmentation to Table 3 and decide on predicates (exactly 2) for horizontal fragmentation. Show the completeness and disjointness of your fragments.

Q4: Show how Table 2 may be <u>further</u> fragmented from the fragments in Q2 by adding the predicates in Q3. Explain how you performed the derived fragmentation. Show the completeness and disjointness of your fragments.

Notes:

1. Provide a brief description of your schema.
2. Check which ER relationships could map to the given pattern. This will assist you in your schema choices.
3. This is a practical assignment. You need to create your DB schema. You may use an online SQL engine such as:
    programiz https://www.programiz.com/sql/online-compiler/
    SQLite https://sqliteonline.com/
    myCompiler https://www.mycompiler.io/new/sql

4. Note that not all online engines support referential integrity constraints. Hence, ensure that the data in your tables does not violate these constraints. Otherwise, your fragments will be incorrect.
5. Create your tables using the CREAT TABLE and the INSERT commands. Include a **snapshot** of your tables in the answer sheet.
6. The number of rows you insert in each table is subject to your judgment.
7. Save the fragments in new tables. Keep the original tables.
8. Apply appropriate SQL commands to show the completeness and disjointness of your fragments. Include a **snapshot** of the results in the answer sheet.

Rubric:

Questions will be graded based on a logical and correct application of fragmentation steps, the correctness of verification steps, and the provided results.

	Q1	Q2	Q3	Q4
	Mark	Mark	Mark	Mark
Fragmentation: SQL code accurately fragments the tables according to specified predicates. SQL code is correctly implemented, syntactically correct, and effectively fragments the tables. Results are present.	10%	15%	10%	20%
Disjointness and completeness are effectively demonstrated through the SQL code provided. Results are present.	10%	10%	10%	10%

	Mark
Validity of DB: Database design is logically sound	5%
Quality of solution	5%