HyperionDev

Relational Databases

Model Answer Approach

Auto-graded task

1. What is normalisation?

Normalisation is a technique used to organise a database efficiently. The goal is to minimise data redundancy and reduce the potential for data anomalies. It involves breaking down a database into smaller, related tables and defining relationships between them. This process improves data consistency and integrity.

2. When is a table in 1NF?

A table is in the First Normal Form (1NF) if:

- Each cell contains only one value.
- Each row is unique and identifiable by a primary key.

3. When is a table in 2NF?

A table is in the Second Normal Form (2NF) if:

- It meets all the rules of 1NF.
- All non-key attributes are fully dependent on the entire primary key. This means no column depends only on a part of a composite primary key.

4. When is a table in 3NF?

A table is in the Third Normal Form (3NF) if:

- It meets all the rules of 2NF.
- All non-key attributes depend only on the primary key and not on other non-key attributes. This removes transitive dependencies.

5. INVOICE dependency diagram

The relational scheme for the `INVOICE` table structure is as shown below:

`INVOICE (INV_NUM, PROD_NUM, SALE_DATE, PROD_LABEL, VEND_CODE, VEND_NAME, QUANT_SOLD, PROD_PRICE)`

The primary key for the table `INVOICE` is `{INV_NUM, PROD_NUM}`.

The functional dependencies are as follows: 1NF

{INV_NUM, PROD_NUM} ⟶ {SALE_DATE, PROD_LABEL, VEND_CODE, VEND_NAME, QUANT_SOLD, PROD_PRICE}

{INV_NUM} ⟶ {SALE_DATE}

{PROD_NUM} ⟶ {PROD_LABEL, PROD_PRICE, VEND_CODE, VEND_NAME}

{VEN_CODE} ⟶ {VEN_NAME}

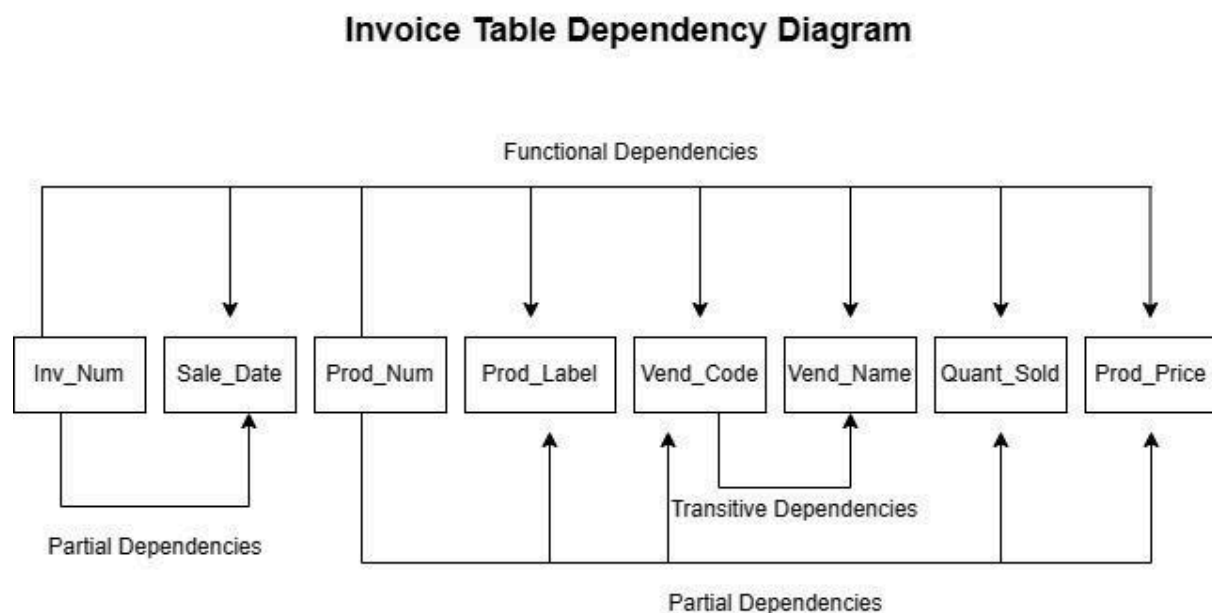Among the dependencies, the partial dependencies are as follows:

{INV_NUM} ⟶ {SALE_DATE}

{PROD_NUM} ⟶ { PROD_LABEL, PROD_PRICE, VEND_CODE, VEND_NAME}

Among the dependencies, the transitive dependencies are as follows:

{VEN_CODE} ⟶ {VEN_NAME}

The dependency diagram is as follows:

Invoice Table Dependency Diagram

6. Remove all partial dependencies and draw the new dependency diagrams

The functional dependencies are as follows: 2NF

{INV_NUM, PROD_NUM} ⟶ { QUANT_SOLD}

{INV_NUM} ⟶ {SALE_DATE}

{PROD_NUM} ⟶ { PROD_LABEL, PROD_PRICE, VEND_CODE, VEND_NAME}

Among the dependencies, the transitive dependencies are as follows:

{VEN_CODE} ⟶ {VEN_NAME}

Invoice Table Dependency Diagram

Functional Dependencies

| Prod_Num | Prod_Label | Prod_Price | Vend_Code | Vend_Name |

Transitive Dependencies

| Inv_Num | Prod_Num | Quant_Sold |

| Inv_Num | Sale_Date |

7. Remove all transitive dependencies and draw the new dependency diagrams

The functional dependencies are as follows: 3NF
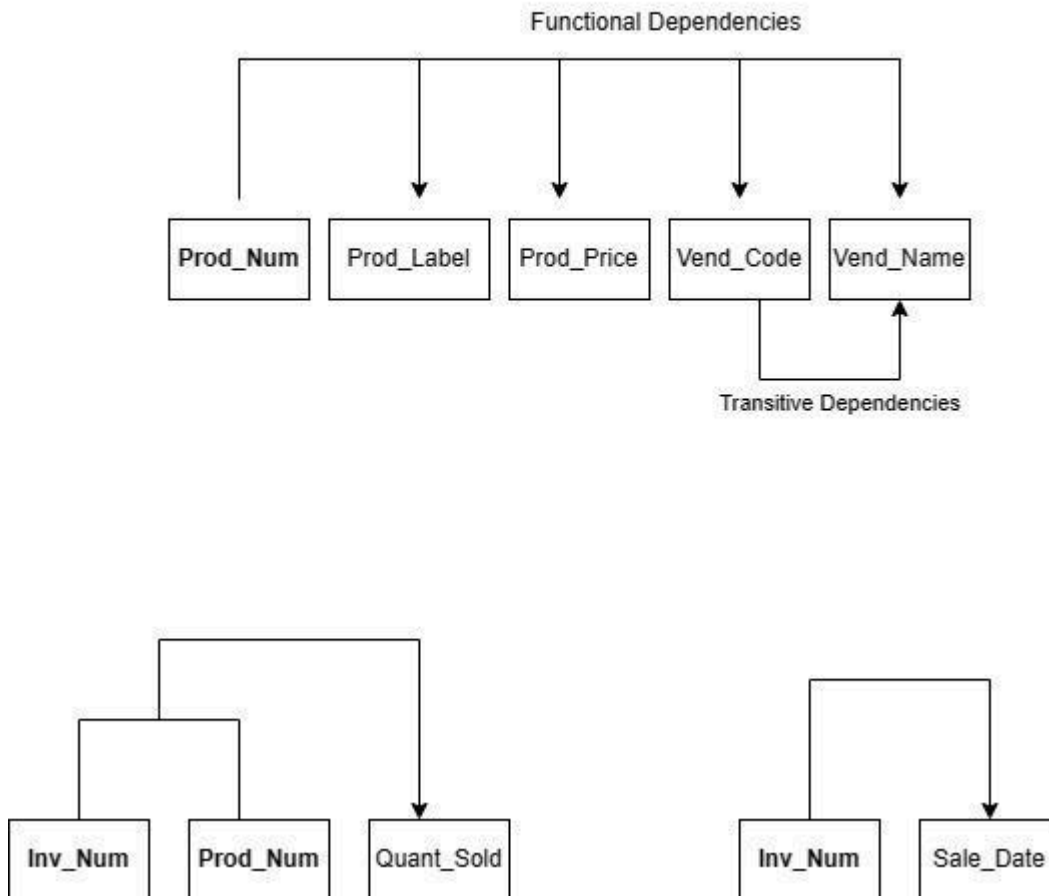
{INV_NUM, PROD_NUM} ⟶ { QUANT_SOLD}

{INV_NUM} ⟶ {SALE_DATE}

{PROD_NUM} ⟶ { PROD_LABEL, PROD_PRICE, VEND_CODE}

{VEN_CODE} ⟶ {VEN_NAME}

Invoice Table Dependency Diagram

Functional Dependencies

| Prod_Num | Prod_Label | Prod_Price | Vend_Code |

| Inv_Num | Prod_Num | Quant_Sold | | Inv_Num | Sale_Date |

| Vend_Code | Vend_Name |