*Deliverable 2*

*Task3*

**Restaurant Delivery System Database: Compact Entity Overview**

• **Customer Entity:**

* Represents individuals utilizing the system.
* Attributes: Unique ID, name, phone number, address, password, and email.

• **Delivery Personnel Entity:**

* Represents personnel responsible for order delivery.
* Attributes: Unique ID, name, operating city, and phone number and and operational status (Is\_fired).

• **Vehicle Entity:**

* Represents the delivery vehicles used in the system.
* Attributes: Unique ID, maximum meal capacity, speed, and operational status (Is\_working).

• **Delivery Entity:**

* Represents a set of customer orders bundled for delivery.
* Attributes: Unique ID, delivery time (DTime), and delivery status (Is\_delivered).

• **Orders Entity:**

* Represents individual customer orders.
* Attributes: Unique ID, and order city.

• **Meals Entity:**

* Represents various food items available in the system.
* Attributes: Restaurant ID, meal name, and price.

• **Restaurant Entity:**

* Represents individual restaurants in the delivery system.
* Attributes: Unique ID, restaurant name, and location city.

• **Payment Entity:**

* Represents payments made within the system.
* Attributes: Unique ID and payment date.

• **Payment-Cash Entity:**

* Represents cash transactions within the system.
* Attributes: Unique ID, transaction amount.

• **Payment-Online Entity:**

* Represents online payment transactions within the system.
* **Attributes:** Unique ID, transaction amount, credit card information (Credit\_CARD), and payment time (Ptime).

**Note:** The attributes presented are significant fields essential for understanding the database structure.

**Functional dependencies and Minimal Cover:**

**Delivery Personnel Entity:**

* Functional Dependencies (FDS):

**Functional Dependency 1:**

{Pid}→{PName, DpCity, DP\_Phone\_Number,Is\_fired}

* The delivery personnel's unique ID uniquely determines their name city, and phone number,and if he is working or not.

**Functional Dependency 2:**

{Vid}→{Pid}

* The vehicle's unique ID uniquely determines the associated delivery personnel Pid
* Minimal Cover:
* **Functional Dependency 1:**

{Pid}→{Name, DpCity, DPPhoneNumber,Is\_fired}

* The unique identifier Pid is the primary key, determining the delivery personnel's Name, DpCity, and DPPhoneNumber ons if he is working or not (IS\_fired). This dependency is minimal, as removing any attribute from the right-hand side and conducting a search to determine if we can reach it without considering that functional dependency would result in a false assertion, ultimately breaking the functional dependency.
* **Functional Dependency 2:**

{Vid}→{Pid}

* the unique identifier Vid determines the associated delivery personnel Pid. This dependency is minimal, as removing any attribute from the right-hand side would break the functional dependence.

**Summary:**

These functional dependencies represent essential relationships within the Delivery Personnel entity. They provide a minimal cover, ensuring there are no redundant dependencies, and each dependency is necessary for an accurate representation and normalization of the data. In summary, G (Minimal Cover) is equivalent to F (Set of Functional Dependencies)

**Vehicle Entity:**

* Functional Dependencies (FDS):

{Vid}→{Max Meals, Speed,Is\_Working}

* The unique identifier Vid uniquely determines the maximum meals and speed of the vehicle.
* Minimal Cover:
* **Functional Dependency:**

{Vid}→{Max Meals, Speed,Is\_working}

* The unique identifier Vid is the primary key, determining the Max Meals and Speed. This dependency is already minimal, as removing any attribute from the right-hand side and conducting a search to determine if we can reach it without considering that functional dependency would result in a false assertion, ultimately breaking the functional dependency

**Summary:**

These functional dependencies represent essential relationships within the Vehicle entity. They provide a minimal cover, ensuring there are no redundant dependencies, and each dependency is necessary for an accurate representation and normalization of the data. In summary, G (Minimal Cover) is equivalent to F (Set of Functional Dependencies)

**Delivery Entity:**

* Functional Dependencies (FDS):

{Did}→{Ddate, Vid,Dtime,Is\_delivered}

* The unique identifier Did uniquely determines the delivery date (Ddate) and the associated vehicle (Vid) and also the specific time that the delevery was delivered.
* Minimal Cover:
* **Functional Dependency:**

{Did}→{Ddate, Vid,Dtime}

* The unique identifier Did is the primary key, determining the Ddate and Vid. This dependency is already minimal, as removing any attribute from the right-hand side and conducting a search to determine if we can reach it without considering that functional dependency would result in a false assertion, ultimately breaking the functional dependency.

**Summary:**

These functional dependencies represent essential relationships within the Delivery entity. They provide a minimal cover, ensuring there are no redundant dependencies, and each dependency is necessary for an accurate representation and normalization of the data. In summary, G (Minimal Cover) is equivalent to F (Set of Functional Dependencies)

**Orders Entity:**

* Functional Dependencies (FDS):

{Oid}→{Cid, Order\_city }

* The unique identifier Oid uniquely determines the associated customer Cid.
* Minimal Cover:
* **Functional Dependency:**

{Oid}→{Cid, Order\_city }

* The unique identifier Oid is the primary key, determining the associated customer Cid. This dependency is already minimal, as removing any attribute from the right-hand side and conducting a search to determine if we can reach it without considering that functional dependency would result in a false assertion, ultimately breaking the functional dependency.

**Summary:**

These functional dependencies represent essential relationships within the Orders entity. They provide a minimal cover, ensuring there are no redundant dependencies, and each dependency is necessary for an accurate representation and normalization of the data. In summary, G (Minimal Cover) is equivalent to F (Set of Functional Dependencies)

**Meal Entity:**

* Functional Dependencies (FDS):

{Rid, Mname}→{Price}

* The combination of Rid and Mname uniquely determines the price of the meal.
* Minimal Cover:
* **Functional Dependency:**

{Rid, Mname}→{Price}

* The combination of Rid and Mname serves as a composite primary key, uniquely determining the price of the meal. This dependency is already minimal, as removing any attribute from the right-hand side and conducting a search to determine if we can reach it without considering that functional dependency would result in a false assertion, ultimately breaking the functional dependency..

**Summary:**

These functional dependencies represent essential relationships within the Meal entity. They provide a minimal cover, ensuring there are no redundant dependencies, and each dependency is necessary for an accurate representation and normalization of the data. In summary, G (Minimal Cover) is equivalent to F (Set of Functional Dependencies)

**Restaurant Entity:**

* Functional Dependencies (FDS):

{Rid}→{Rname, RCity}

* The unique identifier Rid uniquely determines the restaurant name (Rname) and city (RCity).
* Minimal Cover:
* **Functional Dependency:**

{Rid}→{Rname, RCity}

* The unique identifier Rid is the primary key, determining the restaurant name (Rname) and city (RCity). This dependency is already minimal, as removing any attribute from the right-hand side and conducting a search to determine if we can reach it without considering that functional dependency would result in a false assertion, ultimately breaking the functional dependency..

**Summary:**

These functional dependencies represent essential relationships within the Restaurant entity. They provide a minimal cover, ensuring there are no redundant dependencies, and each dependency is necessary for an accurate representation and normalization of the data. In summary, G (Minimal Cover) is equivalent to F (Set of Functional Dependencies)

**Customer Entity:**

* Functional Dependencies (FDS):

{Cid}→{Address, Cname, Email, Phone\_Number, Pass\_word}

* The unique identifier Cid uniquely determines the customer's address, name, email, phone number, and password.

{Email}→{Cid}

* The customer's email uniquely determines their Cid.

Minimal Cover**:**

* **Functional Dependency 1:**

{Cid}→{Address, Cname, Email, Phone\_Number, Pass\_word}

* The unique identifier Cid is the primary key, determining the customer's address, name, email, phone number, and password. This dependency is already minimal, as removing any attribute from the right-hand side would break the functional dependence.
* **Functional Dependency 2:**

{Email}→{Cid}

* The unique identifier Email is a candidate key, determining the customer's Cid. This dependency is already minimal, as removing any attribute from the right-hand side would break the functional dependence.

**Summary:**

These functional dependencies represent essential relationships within the Customer entity. They provide a minimal cover, ensuring there are no redundant dependencies, and each dependency is necessary for an accurate representation and normalization of the data. In summary, G (Minimal Cover) is equivalent to F (Set of Functional Dependencies)

**Payment Entity:**

* Functional Dependencies (FDS):

{PyID}→{PDate}

* The unique identifier PyID uniquely determines the payment date (PDate).
* Minimal Cover:
* **Functional Dependency:**

{PyID}→{Pdate}

* The unique identifier PyID is the primary key, determining the payment date (PDate). This dependency is already minimal, as removing any attribute from the right-hand side and conducting a search to determine if we can reach it without considering that functional dependency would result in a false assertion, ultimately breaking the functional dependency..

**Summary:**

These functional dependencies represent essential relationships within the Payment entity. They provide a minimal cover, ensuring there are no redundant dependencies, and each dependency is necessary for an accurate representation and normalization of the data. In summary, G (Minimal Cover) is equivalent to F (Set of Functional Dependencies)

**Payment\_cash Entity:**

* Functional Dependencies (FDS):

{PCid}→{Budget, PCPrice, PID,PYid}

* The unique identifier PCid uniquely determines the budget, payment price, and delivery personnel ID.
* Minimal Cover:
* **Functional Dependency:**

{PCid}→{Budget, PCPrice, PID,PYid}

The unique identifier PCid is the primary key, determining the budget, payment price, and delivery personnel ID. This dependency is already minimal, as removing any attribute from the right-hand side and conducting a search to determine if we can reach it without considering that functional dependency would result in a false assertion, ultimately breaking the functional dependency.

**Summary:**

These functional dependencies represent essential relationships within the Payment\_Cash entity. They provide a minimal cover, ensuring there are no redundant dependencies, and each dependency is necessary for an accurate representation and normalization of the data. In summary, G (Minimal Cover) is equivalent to F (Set of Functional Dependencies)

**Payment\_Online Entity:**

* Functional Dependencies (FDS):

{PoID}→{POPrice,Credit\_Card,Ptime,PYid}

The unique identifier PoID uniquely determines the online payment price (POPrice),the credit cards,and the payment id.

* Minimal Cover:
* **Functional Dependency:**

{PoID}→{POPrice,Credit\_Card,Ptime,PYid}

The unique identifier PoID is the primary key, determining the online payment price (POPrice), the credit cards,and the payment id. This dependency is already minimal, as removing any attribute from the right-hand side and conducting a search to determine if we can reach it without considering that functional dependency would result in a false assertion, ultimately breaking the functional dependency.

**Summary:**

These functional dependencies represent essential relationships within the Payment\_Online entity. They provide a minimal cover, ensuring there are no redundant dependencies, and each dependency is necessary for an accurate representation and normalization of the data. In summary, G (Minimal Cover) is equivalent to F (Set of Functional Dependencies)

**ComposedBy Relationship:**

* Functional Dependencies (FDS):
* **Functional Dependency:**

{Did, Order\_id} → {Did, Order\_id}

* The "ComposedBy" relationship represents a many-to-many association between the "Delivery" and "Orders" entities, governed by participation constraints. Specifically, it stipulates that each delivery must be associated with at least one order, and conversely, each order is linked to at least one delivery.
* **Functional Dependency Analysis:**
* In the given functional dependency {Did, Order\_id} → {Did, Order\_id}, this relationship involves the keys of both entities. Consequently, the functional dependency is considered trivial since it encompasses the entirety of both keys.
* Minimal Cover:
* As this relationship exhibits only one functional dependency, and that dependency is trivial, the minimal cover is effectively empty.

**Contains Relationship:**

* Functional Dependencies (FDS):
* **Functional Dependency:**

{Oid, {Rid, Mname}} -> {Oid, Rid, MName}

* The "Contains" relationship represents a many-to-many association between the "Meal" and "Orders" entities, governed by participation constraints. Specifically, it stipulates that each meal can be be associated with several orders, and conversely, each order contained at least one meal.
* **Functional Dependency Analysis:**
* In the given functional dependency {Oid, {Rid, Mname}} -> {Oid, Rid, MName} this relationship involves the keys of both entities. Consequently, the functional dependency is considered trivial since it encompasses the entirety of both keys.
* Minimal Cover:
* As this relationship exhibits only one functional dependency, and that dependency is trivial, the minimal cover is effectively empty.

**Pays Relationship:**

* Functional Dependencies (FDS):
* **Functional Dependency:**

{PyId, Oid} -> {PyId, Oid}

* The "Pays" relationship represents a many-to-many association between the "Payment" and "Orders" entities, governed by participation constraints. Specifically, it stipulates that each payment can be associated with at least one orders, and conversely, each orderis payed by at least one payment.
* **Functional Dependency Analysis:**

In the given functional dependency {PyId, Oid} -> {PyId, Oid}

* this relationship involves the keys of both entities. Consequently, the functional dependency is considered trivial since it encompasses the entirety of both keys.
* Minimal Cover:
* As this relationship exhibits only one functional dependency, and that dependency is trivial, the minimal cover is effectively empty.