**Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology**

**(Deemed to be University Estd. u/s 3 of UGC Act, 1956)**

**School of Computing**

**B.Tech. – Computer Science and Engineering**

**VTR UGE2021- (CBCS)**

Academic Year: 2025–2026

SUMMER SEMESTER - SS2526

Course Code : 10211CS207

Course Name: Database Management Systems

Slot NO : S2L5

DBMS TASK - 8 REPORT

# Title: Normalizing Databases Using Functional Dependencies up to BCNF

**Submitted by:**

|  |  |  |
| --- | --- | --- |
| **VTUNO** | **REGISTER NUMBER** | **STUDENT NAME** |
| VTU27661 | 24UECS1159 | C Venugopal reddy |

**Task 8**

# Lab Task 8: Normalizing Databases Using Functional Dependencies up to BCNF

**Objective:**

To normalize the database created in Task-2 using functional dependencies (FDs) and apply normalization techniques up to **BCNF (Boyce-Codd Normal Form)**.

**1. Apply the Functional Dependency and Normalize to 1NF**

## Step 1: Identify Functional Dependencies (FDs)

Consider the following relations and FDs:

**OrderTable (Order\_ID, Cust\_ID, Order\_Date, Order\_Total, Payment\_Status)**

* FD1: Order\_ID → Cust\_ID, Order\_Date, Order\_Total, Payment\_Status

**Customer (Cust\_ID, Cust\_Name, Cust\_Contact, Cust\_Email, Cust\_Address)**

* FD2: Cust\_ID → Cust\_Name, Cust\_Contact, Cust\_Email, Cust\_Address

**Menu\_Item (Item\_ID, Item\_Name, Price, Category, Rest\_ID)**

* FD3: Item\_ID → Item\_Name, Price, Category, Rest\_ID

## Normalization to 1NF (First Normal Form)

* Ensure that each column contains only atomic (indivisible) values.
* Remove any repeating groups.
* Example:

**Order\_ID Cust\_ID Order\_Date Order\_Total Payment\_Status**

1. 1 2025-01-20 800 Paid
2. 2 2025-01-21 500 Unpaid

# Normalize the Relations Using FD+ and α+

* Compute **FD+ (Closure of FDs)** using Armstrong’s Axioms.  Identify minimal keys and remove redundant FDs.

**Closure for OrderTable:**

* FD+: { Order\_ID → Cust\_ID, Order\_Date, Order\_Total, Payment\_Status }

**Closure for Customer:**

* FD+: { Cust\_ID → Cust\_Name, Cust\_Contact, Cust\_Email, Cust\_Address }

**Closure for Menu\_Item:**

* FD+: { Item\_ID → Item\_Name, Price, Category, Rest\_ID }

# Find the Minimal Cover and Canonical Cover

**Minimal Cover:**

* FD1: Order\_ID → Cust\_ID, Order\_Date, Order\_Total, Payment\_Status
* FD2: Cust\_ID → Cust\_Name, Cust\_Contact, Cust\_Email, Cust\_Address
* FD3: Item\_ID → Item\_Name, Price, Category, Rest\_ID

**Canonical Cover:**

* No redundancy detected.

# Normalize to 2NF

* A relation is in **2NF** if it is in 1NF and has no partial dependencies.  Remove partial dependencies by creating separate relations.

**Normalization to 2NF:**

* **OrderTable (Order\_ID, Order\_Date, Order\_Total, Payment\_Status)**
* **Customer (Cust\_ID, Cust\_Name, Cust\_Contact, Cust\_Email, Cust\_Address)**
* **Menu\_Item (Item\_ID, Item\_Name, Price, Category, Rest\_ID)**

# Normalize to 3NF

* A relation is in **3NF** if it is in 2NF and has no transitive dependencies.  Ensure non-prime attributes depend only on primary keys.

**Normalization to 3NF:**

* **Restaurant (Rest\_ID, Rest\_Name, Rest\_Location, Rest\_Contact)**
* **Menu\_Item (Item\_ID, Item\_Name, Price, Category, Rest\_ID)**

# Normalize to BCNF

* A relation is in **BCNF** if, for every functional dependency (X → Y), X is a super key.  Identify and remove transitive dependencies.

**Normalization to BCNF:**

* **OrderTable (Order\_ID, Cust\_ID, Order\_Date, Order\_Total, Payment\_Status)**
* **Customer (Cust\_ID, Cust\_Name, Cust\_Contact, Cust\_Email, Cust\_Address)**
* **Menu\_Item (Item\_ID, Item\_Name, Price, Category, Rest\_ID)**

**1. OrderTable**

| **Attribute** | **Description** | **Key** |
| --- | --- | --- |
| Order\_ID | Unique order number | **Primary Key** |
| Cust\_ID | Customer who placed the order | Foreign Key → Customer(Cust\_ID) |
| Order\_Date | Date of order |  |
| Order\_Total | Total amount |  |
| Payment\_Status | Paid / Unpaid |  |

**2. Customer**

| **Attribute** | **Description** | **Key** |
| --- | --- | --- |
| Cust\_ID | Unique customer ID | **Primary Key** |
| Cust\_Name | Name of the customer |  |
| Cust\_Contact | Contact number |  |
| Cust\_Email | Email address |  |
| Cust\_Address | Address |  |

**3. Menu\_Item**

| **Attribute** | **Description** | **Key** |
| --- | --- | --- |
| Item\_ID | Unique item ID | Primary Key |
| Item\_Name | Name of the food item |  |
| Price | Price of the item |  |
| Category | Food category (e.g., Starter, Main Course) |  |
| Rest\_ID | Restaurant ID | Foreign Key → Restaurant(Rest\_ID) |

**4. Restaurant**

| **Attribute** | **Description** | **Key** |
| --- | --- | --- |
| Rest\_ID | Unique restaurant ID | Primary Key |
| Rest\_Name | Restaurant name |  |
| Rest\_Location | Address/location |  |
| Rest\_Contact | Contact details |  |

RESULT:Thus the task has been executed and verified successfully.