

DBMS ASSIGNMENT

Nikki Gautam

Question 2:

Part C:

We expect a write up of at-least ten sentences explaining your rationale behind the process. Why was a particular table normalized? What level of normalization did you use? If something is left denormalized, why was it left that way? Any query that could benefit from it?

If you are storing the phone number as a string instead of a number, what are the benefits and the drawbacks, and which one did you use?

Normalization Process:

- **1NF**: Ensured atomic values with no repeating groups, e.g., “**Order_MenuItem**” table links orders and menu items
- **2NF**: Non-key attributes are fully dependent on the primary key, e.g., price in the “**MenuItem**” table depends on “**itemid**”
- **3NF**: Removed transitive dependencies, e.g., “**Discount**” table has attributes directly related to discounts only

Denormalization Decisions:

- Some data, like “**totalamount**”, “**discountapplied**”, and “**paymentstatus**”, was kept denormalized for performance. This reduces the need for joins, making queries faster

Choice of Data Types:

- Phone numbers are stored as **strings** to maintain proper formatting (e.g., country codes, dashes). This avoids issues with truncation or incorrect formatting. However, storing them as strings increases storage size slightly

Indexes:

- Primary and foreign keys (e.g., **userid**, **restaurantid**, **orderid**) ensure data integrity and improve query performance through indexing

Transactions and Data Integrity:

- Transactions between “**Order**” and “**Payment**” tables ensure that both are updated together, maintaining consistency

Query Performance:

- Denormalizing fields like “**paymentstatus**” allows faster queries without joins, improving performance for common requests

Future Flexibility:

- The “**Discount**” table is normalized to allow multiple discounts per order, providing flexibility for future changes
- The proposed solution can easily support new features like promotions or additional payment methods without major changes