Indian Institute of Technology Gandhinagar

Databases
CS 432

Report: Assignment 1

Authors

22110050	Birudugadda Srivibhav	20110101	Pramod Limbore
22110260	Srivathsa Vamsi Chaturvedula	20110106	Shriyash Mandavekar
22110124	Sriman reddy	20110071	Haikoo Khandor
22110162	Nikhilesh Myanapuri	20110104	Madhav Kanda

Under the Supervision of
Prof. Mayank Singh

January 29, 2024

Index

Responsibility of G1:

Description of Database

Introduction:

The objective of this assignment is to conduct a comprehensive study of the database requirements for the Food Delivery System at IIT Gandhinagar and design an efficient and responsive database system. The project's foundation involves creating an Entity-Relationship (E-R) Diagram and a relational database. The successful implementation of the food delivery system necessitates collaboration and coordination among various stakeholders, including students, faculty members, staff, restaurants, administrative staff, delivery personnel, and IT support.

Stakeholder Overview:

1) **Customers (Students, Faculty, and Staff):** Customers represent a crucial stakeholder group, influencing the system through preferences, order history, and feedback. Their engagement is vital for system improvements.

2) **Restaurants and Canteens:** Partnering eateries play a significant role in influencing menu items, pricing, and fulfilment processes, impacting the overall service quality.

3) **Administrative Staff:** Responsible for managing orders, coordinating with delivery personnel, and overseeing financial transactions, administrative staff are key stakeholders ensuring the smooth operation of the system.

4) **Delivery Personnel:** Pivotal for timely and accurate order deliveries, delivery personnel contribute to the overall customer satisfaction and the system's efficiency.

5) **Developers and IT Support Team:** Essential stakeholders tasked with system maintenance and upgrades, ensuring continuous functionality and improvement of the food delivery service.

Stakeholder Visits:

On 24th January 2024:

1) During the visit to Dwijendra, who manages Just Chill Restaurant, he introduced us to the "mony" app—a budget and expense tracker employed at his outlet. Dwijendra expressed enthusiasm and emphasised the absence of a pre-implemented food delivery system at IIT Gandhinagar. He envisioned introducing a new app to enhance organisational efficiency and eagerly anticipated the potential improvements it could bring to the restaurant's operations.

2) Visit to Mukundh (Amul Parlour): No information was obtained as Mukundh is an employee and couldn't share details.

3) Visit to Vadher Gautam (Tea Post Outlet): Similar to Mukundh, Gautam, an employee, needed help to provide information.

On 25th January 2024:

1) Visit to Prof Balagopal Komarath: Prof. Komarath and his family expressed satisfaction with using WhatsApp for grocery shopping.

2) Visit to Prof Nipun Batra: Prof. Batra and his family acknowledged the adequacy of WhatsApp for grocery shopping but highlighted the potential benefits of a new app for better organisation.

3) During our visit to the Brahmani Veg Solutions Outlet, the staff underscored their preference for using WhatsApp over alternative applications in their daily operations. They attributed this choice to the simplicity and ease of managing tasks, which is particularly noteworthy given the smaller size of their customer base.

These stakeholder visits provide valuable insights into their preferences, existing tools, and openness to adopting a new food delivery system app at IIT Gandhinagar. Such interactions will inform the database design, ensuring alignment with stakeholders' needs and preferences.

Major Requirements of the Food Delivery System Database:

1) **Customer Information Management:** The database should efficiently manage customer information, including students, faculty, and staff, as well as external users, providing a comprehensive record of their preferences, order history, and feedback.

2) **Order and Transaction History:** The system must maintain a detailed history of orders and financial transactions, allowing for easy tracking of purchases, deliveries, and payments.

3) **Menu and Pricing Configuration:** The database should support the dynamic configuration of menus and pricing for the partnered restaurants and canteens, enabling real-time updates based on changes in offerings and pricing strategies.

4) **Feedback and Ratings:** There should be provisions for collecting and storing customer feedback and ratings, facilitating continuous improvement in the quality of service the food delivery system provides.

5) **Restaurant and Canteen Information:** Information about the partnered restaurants and canteens, including menu items, pricing, and fulfilment processes, should be effectively managed to ensure accurate and up-to-date service.

6) **Order Processing and Fulfillment:** The database must support the efficient processing of orders, coordinating with restaurants, managing delivery personnel assignments, and ensuring timely and accurate order fulfilment.

7) **Delivery Personnel Management:** Maintain records of delivery personnel, including their schedules, performance metrics, and other relevant details to optimize delivery operations and ensure a high level of service.

8) **Administrative Functions:** Provide features for administrative staff to manage orders, coordinate with various stakeholders, and oversee financial transactions, ensuring the smooth functioning of the food delivery system.

9) **System Maintenance and Upgrades:** Support for IT personnel to manage and upgrade the system, ensuring its continuous functionality and incorporating enhancements based on changing requirements or technological advancements.

10) **Inventory Tracking for Restaurants:** Track the inventory of food items at partnered restaurants, providing insights into stock levels, expiry dates, and pricing to help maintain sufficient stock and avoid service disruptions.

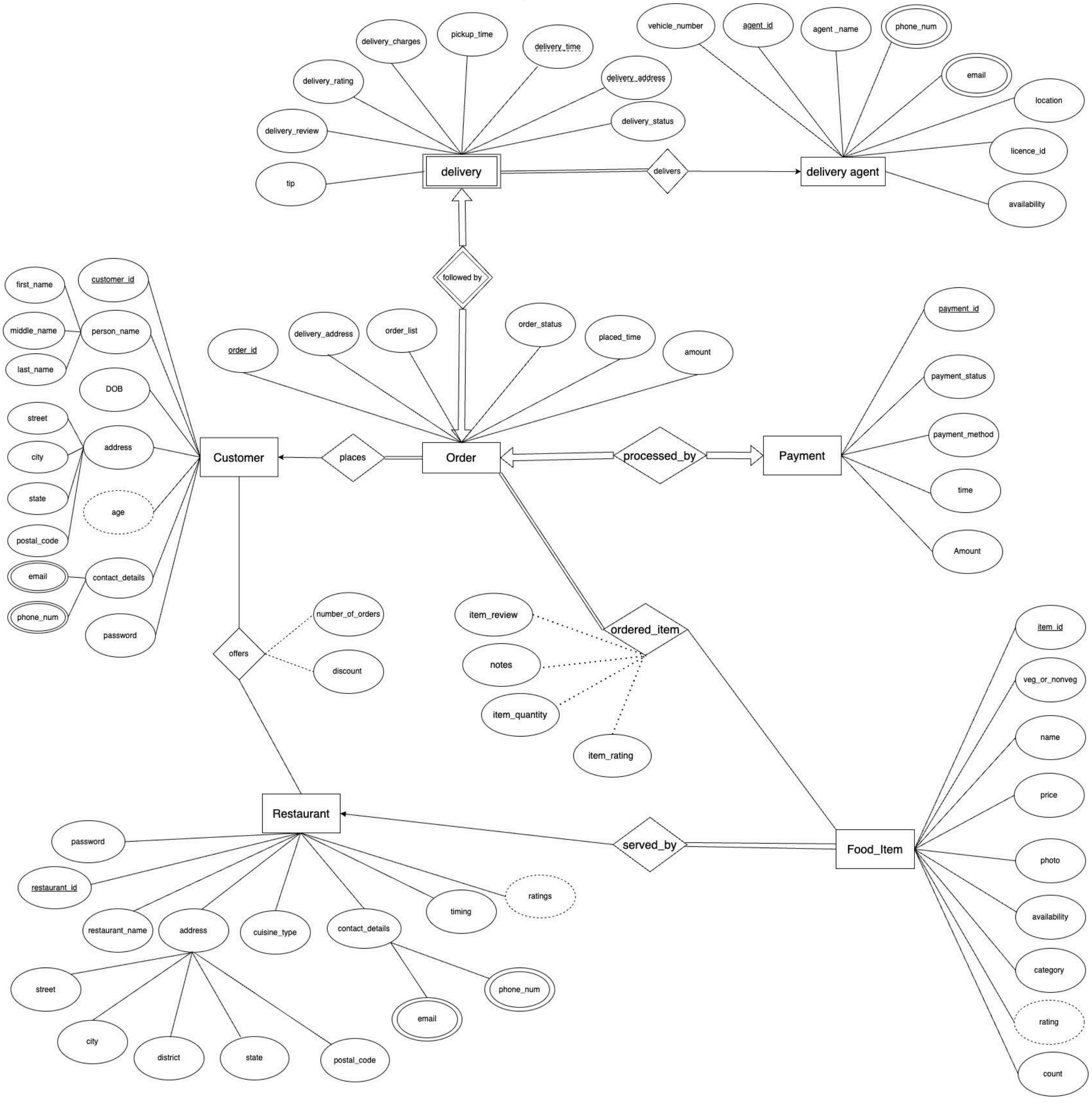
11) **Vendor Information:** Maintain a comprehensive list of vendors supplying food items to the partnered restaurants, record transactions and prices, and ensure transparency in the procurement process.

12) **Financial Transactions and Invoicing:** Record and manage financial transactions related to orders, generate restaurant invoices and maintain accurate financial records to facilitate transparent financial operations.

By addressing these major requirements, the food delivery system database will be a robust foundation for efficient and organised operations, benefiting stakeholders and end-users at IIT Gandhinagar.

Responsibility of G1 and G2:

ER Diagram



Link: https://drive.google.com/file/d/1WMHX3q_KzyOfyVp3YxpUGo3Ix6fWJlKo

Entities, Relationships, and Attributes used

Entities and attributes involved

1) **Order** - order_id, delivery_address, order_status, placed_time, amount, order_list
2) **Payment**- payment_id, payment_status, payment_method, time, amount
3) **Food_item**- item_id, veg_or_nonveg, name, price, photo, availability, category, rating, count
4) **Restaurant** - restaurant_id, restaurant_name, street, city, district, state, postal code, cuisine_type, timing, ratings, password, email, phone_num
5) **Customer**- customer_id, first_name, middle_name, last_name, DOB, street, city, state, postal_code, age, email, phone_num, password
6) **Delivery**- tip, delivery_review, delivery_rating, delivery_charges, pickup_time, delivery_time, delivery_address, delivery_status
7) **Delivery agent -** agent_id, agent_name, phone_num, email, location, licence_id, vehicle_number, availability

Relationships and their cardinalities, Participation Constraints

Sr. No.	Relationship	Relating Entities	Degree of relations	Cardinalities	Participation Constraints
1	served_by	Food_Item → Restaurant	Binary	Many to one	Food_item: Total Restaurant: Partial
2	offers	Restaurant → Customer	Binary	Many to many	Restaurant: Partial Customer: Partial
3	processed_by	Order → Payment	Binary	One to one	Order: Total Payment: Total
4	places	Customer → Order	Binary	One to Many	Customer: Partial Order: Total
5	ordered_item	Order → Food_Item	Binary	Many to Many	Order: Total Food_Item: Partial
6	followed_by	Order→ Delivery	Binary	One to one	Order: Total Delivery: Total
7	delivers	Delivery agent → Delivery	Binary	One to many	Delivery: Total Delivery agent: Partial

Relationships and Descriptive attributes:

Sr. No.	Relationship	Attributes
1	Ordered_item	item_review, notes, item_quantity, item_rating
2	Offers	number_of_orders, Discount

Justification

1) delivers:

a) ***Cardinality:*** Multiple deliveries can be assigned to a single delivery agent, but multiple delivery agents can't be assigned to a single delivery.

b) ***Participation Constraints:*** Not all delivery agents would be occupied with some delivery, but all orders are mapped to some delivery agent so delivery agent has a partial participation and delivery has a total participation.

2) served_by:

a) ***Cardinality:*** A restaurant can offer multiple food items, but each food item, distinguished by its unique food_id and price, cannot be served by multiple restaurants.

b) ***Participation Constraints:*** The restaurant offers food items. Food items can not exist without a restaurant, but a restaurant can exist without a food item. Hence, food_item has total participation in the relation.

3) offers:

a) ***Cardinality:*** Multiple restaurants can give offers to multiple customers. In other words, a restaurant can give an offer to multiple people, and one person can get multiple offers from different restaurants. Hence, the relation is many to many.

b) ***Participation Constraints:*** Every person may not get an offer, and every restaurant may not give offers hence, both the customer and restaurant partially participate in the "offers" relationship.

4) processed_by:

a) ***Cardinality:*** One order is paid by only one payment; hence, it is a one-to-one relationship.

b) ***Participation Constraints:*** Each order must be paid by a payment, and payment can only exist if the order is placed by a customer. Hence, both entities will have total participation in the relationship.

5) places:

a) ***Cardinality:*** A Customer can place many orders, but a unique order cannot be placed by multiple customers. Hence, the relationship is one to many.

b) ***Participation Constraints:*** Every order has to be placed by a customer. Hence, all orders are placed by customers, but not all customers place orders at a given time. Hence, "Order" has total participation, whereas "Customer" has partial participation.

6) followed by:

a) ***Cardinality:*** One order should be assigned to a single delivery, and a single delivery should be assigned to a single order.

b) ***Participation Constraints:*** All orders should be mapped to a delivery so they can be delivered. All deliveries should be associated with an order so that no delivery is empty. Thus, it is total from both sides.

7) ordered_item:

a) ***Cardinality:*** A single order can have multiple food items, and a single item can be in multiple orders. Thus, the relation is many to many.

b) ***Participation Constraints:*** A food item does not need to be in an order. For example, if there is food A, B and C, an order can include Food A and B but not C. However, an order must have at least one food item because only then it is defined as an order. Hence, the participation is total from "Order" whereas partial from the "Food_item

Design requirements

1. **c.** At least one primary key and one foreign key: *order_id* is a primary key and foreign key for the *delivery* entity set as it is a weak entity.
2. **d.** At least one one-to-one relationship: *processed_by* is a one-to-one relationship between *order* and *payment* entity sets.
3. **e.** At least any/both of (one-to-many, many-to-one) relationships: *served_by* is a many-to-one relationship from *food_item* to *restaurant* entity sets.
4. **f.** At least one many-to-many relationship: *offers* is a many-to-many relationship from *customers* to *restaurant* entity sets.
5. **g.** At least one of each (total & partial) participation constraint: *customers* entity set has partial participation in *places* relationship where as *order* entity set has total participation in the same relationship.

Responsibility of G2:

Relational Schema

For Relationship sets, the following rules were followed for the relational schema:

- For an ER diagram with a binary many-to-many relationship set, the union of the primary-key attributes from the participating entity sets is taken as the primary key.
- For a binary one-to-one relationship set, the primary key of any one of the entity sets can be chosen as the primary key.
- For a binary many-to-one or one-to-many relationship set, the primary key of the entity set on the "many" side of the relationship is considered the primary key

Relationship sets:

Note: FK stands for Foreign key

A. ***delivers(agent_id, order_id (FK))*** is a binary many-to-one relationship, so the primary key of this relation would be the primary key of the '*delivery*' entity set. Since '*delivery*' is a weak entity set, *order_id* will serve as the primary key.

B. ***served_by(restaurant_id, item_id)*** is a binary one-to-many relationship. So, the primary key of the relation is the item_id which is the primary key of the entity set on many sides of the relation. Therefore, *item_id* is the primary key.

C. ***offers(customer_id, restaurant_id, number, number_of_orders, discount)*** is a binary many-to-many relationship. The primary key of both entity sets will be chosen as the primary key of the relation. Therefore, *customer_id* and *restaurant_id* are the primary keys.

D. ***processed_by(payment_id, order_id)*** is a binary one-to-one relationship. The primary key for this relation can be any of the primary keys of these entity sets. Hence, we choose *payment_id*.

E. ***places(order_id, customer_id)*** is a binary one-to-many relationship, so the primary key for this relation would be the primary key of the entity set on the many side, which means *order_id* will serve as the primary key

F. ***followed_by(order_id)*** is a binary one-to-one relationship. So, the primary key of this relationship can be any of the primary keys of the participating entities. Since, '*delivery*' is a weak entity set, *order_id* will serve as the primary key.

G. ***ordered_item(order_id, item_id, item_review, notes, item_quantity, item_rating)*** is a binary many-to-many relationship. So, the union of the primary key attributes from the participating entity sets becomes the primary key. Therefore *order_id* and *item_id* are primary keys

Entity Sets

Note: FK stands for Foreign key. We do not include any foreign keys in relational schemas of strong entity sets as discussed in the lecture slides.

A. ***Order(order_id, delivery_address, order_list, order_status, placed_time, amount)***

1. In the order entity set, order_id ensures that each record in order is uniquely identified.
2. Therefore, it is a primary key. Order_list contains the list of food items ordered by the customer. All the items except placed_time are NOT_NULL.

B. ***Payment(payment_id , payment_status, payment_method, amount)***

1. For each payment, payment_id ensures that each payment is unique. Hence, it is a primary key.
2. All the attributes are NOT NULL.

C. ***Food_item(item_id, restaurant_id, veg_or_nonveg, name, price, photo, availability, category, rating, count)***

1. Each item is uniquely identified by its item_id. Hence, it is a primary key.
2. Except for photo, category and rating, all the attributes are NOT NULL.
3. The count is the number of times the food item has been previously ordered. It should be a non-negative integer. Rating is a derived attribute which is a mean of all ratings.

D. ***Restaurant (restaurant_id, restaurant_name, street, city, district, state, postal code, cuisine_type, timing, ratings, password, email, phone_num)***

1. Each restaurant has restaurant_id as its primary key.
2. Restaurant_address is a composite attribute which consists of street, city, district, state, and postal_code.
3. It also contains contact_details as another composite attribute, which consists of email and phone number. email and phone_num are multivalued attributes, as they can change later.

E. ***Customer(customer_id, first_name, middle_name, last_name, DOB, street, city, state, postal_code, age, email, phone_num, password)***

1. Each customer is uniquely identified by its customer_id.
2. Customer entity has two composite attributes namely person_name(first_name, middle_name, last_name), address(street, city, district, state, and postal_code) and contact_details(email, phone_num).
3. Email and phone_num are multivalued attributes. Age is a derived attribute

which is obtained from DOB. Age and DOB can be NULL.

F. ***Delivery(order_id(FK), agent_id(FK), tip, delivery_review, delivery_rating, delivery_charges, pickup_time, delivery_time, delivery_address, delivery_status)***

1. Delivery is a weak entity set hence it is identified by the primary key of the order entity set, which works as the foreign key here.
2. It can also be uniquely identified by the primary key of the delivery_agent.
3. The delivery agent is an identifying entity set hence the participation of delivery is total.
4. We have also identified two partial keys or discriminators, namely delivery_time and delivery_address.

G. ***Delivery agent (agent_id, agent_name, phone_num, email, location, licence_id, vehicle_number)***

1. Each delivery agent is assigned a unique agent_id. email and phone_num are multivalued attributes due to the reasons described above.

Note: For multivalued attributes, when forming relational schema, the convention is to form separate relational schema for the entity set containing multivalued attributes, with the primary key equal to the primary key of the original entity set and attribute having the multivalued characteristic.

delivery_agent_phone_num(agent_id(FK), phone_num)

delivery_agent_email(agent_id(FK), email)

customer_phone_num(customer_id(FK), phone_num)

customer_email(customer_id(FK), email)

restaurant_phone_num(restaurant_id(FK), phone_num)

restaurant_email(restaurant_id(FK), email)

Work Distribution

Team Member	Contribution
1. Birudugadda Srivibhav(G1)	Stakeholder visits and enquiry details, Identification of entity sets, attributes and relationships for ER diagram, helped in ER diagram, Report compilation
2. Srivathsa Vamsi(G1)	Stakeholder visits and enquiry details, Identification of entity sets, attributes and relationships for ER diagram, helped in ER diagram, Report compilation
3. Sriman Reddy(G1)	Stakeholder visits and enquiry details, Identification of entity sets, attributes and relationships for ER diagram, helped in ER diagram, Report compilation
4. Nikhlesh Myanapuri(G1)	Stakeholder visits and enquiry details, Identification of entity sets, attributes and relationships for ER diagram, helped in ER diagram, Report compilation
5. Haikoo Khandor(G2)	Helped in the ER diagram process, Identified primary and foreign keys, added constraints on the listed mapping cardinalities, added relationship_sets, entity sets and justified them
6. Madhav Kanda(G2)	Helped in the ER diagram process, Identified primary and foreign keys, added constraints on the listed mapping cardinalities, added relationship_sets, entity sets and justified them
7. Pramod Limbore(G2)	Helped in the ER diagram process, Identified primary and foreign keys, added constraints on the listed mapping cardinalities, added relationship_sets, entity sets and justified them
8. Shriyash Mandavekar(G2)	Helped in the ER diagram process, Identified primary and foreign keys, added constraints on the, listed mapping cardinalities, added relationship_sets, entity_sets and justified them

Acknowledgements

We want to express our thanks to Prof Mayank Singh and TA Ritesh Patidar for their support in the assignment. We are grateful to Prof Balagopal Komarath and Prof Nipun Batra for their valuable suggestions. A big thank you to all the stakeholders for their help during the interviews; their contributions were crucial to the assignment's success.

References

- https://www.geeksforgeeks.org/cardinality-in-dbms/?ref=ml_lbp
- https://www.geeksforgeeks.org/design-restaurant-management-system-system-design/#database-design
- https://www.gleek.io/blog/relational-schema
- Classroom Slides