DATA & APPLICATIONS
PROJECT PHASE 3

RELATIONAL MODEL

Team-32

Navdha Bansal-2022101055
Aryan Chandramania - 2021114004
Srija Mukhopadyay - 2021114002
Harinie Sivaramasethu - 2021114008
Aditi Singh- 2022101075

ER DIAGRAM TO RELATIONAL MODEL:

STEP 1: Mapping of strong entity types:

- For each strong entity in the ER schema we created a relation R that includes all the simple attributes of the strong entity.
- The entity types STAFF, CANTEEN, ORDER, CUSTOMER, PAYMENT were made into relations that contain all the simple attributes of the entity types.
- We separated composite attributes into their simple component attributes. This helps remove the 1-NF violation of composite attributes from the database by taking all the attributes of the composite attribute and adding them in the relation R.

STEP 2: Mapping of weak entity types:

- For each weak entity type (W) in the ER schema with the owner entity type (E), we created a relation R and included all simple attributes of W as attributes of R.
- We also included the primary key attributes of the owner entity type as foreign key attributes of R. These attributes along with its own partial key forms the primary key of the relation.
- For example - CanteenID is the foreign key attribute of MENU ITEMS entity type (which is the primary key of CANTEEN entity type) and CanteenID and CustomerID are foreign key attributes of FEEDBACK entity type (which are primary key attributes of CANTEEN and CUSTOMER entity type resp).

STEP 3: Mapping of binary 1:1 Relationship Types:

- As we do not have any binary 1:1 Relationship types in our ER model, there were no changes made to the relational model during this step.

STEP 4: Mapping of binary 1:N Relationship Types:

- For every binary 1:N relationship type R, we identify the relation S that represents the participating entity type at the N-side of the 1:N relationship type. We include the primary key of T (the relation that represents the participating entity type at the 1-side of the 1:N relationship type) as a foreign key in the relation S**.**
- For example- We included CustomerID (which is the primary key of CUSTOMER entity type) as a foreign key attribute in the PAYMENT relation as a customer can make N payments in a day.

- Similarly we included CanID (which refers to CanteenID which is the primary key of CANTEEN entity type) as a foreign key attribute in the STAFF relation as a canteen can employ N staff members.
- In the same way we included SuperStaffID (which refers to StaffID, the primary key of STAFF entity type) as a foreign key attribute in the STAFF relation itself for 1:N relationship type STAFF SUPERVISES STAFF.

STEP 5: Mapping of binary M:N Relationship Types:

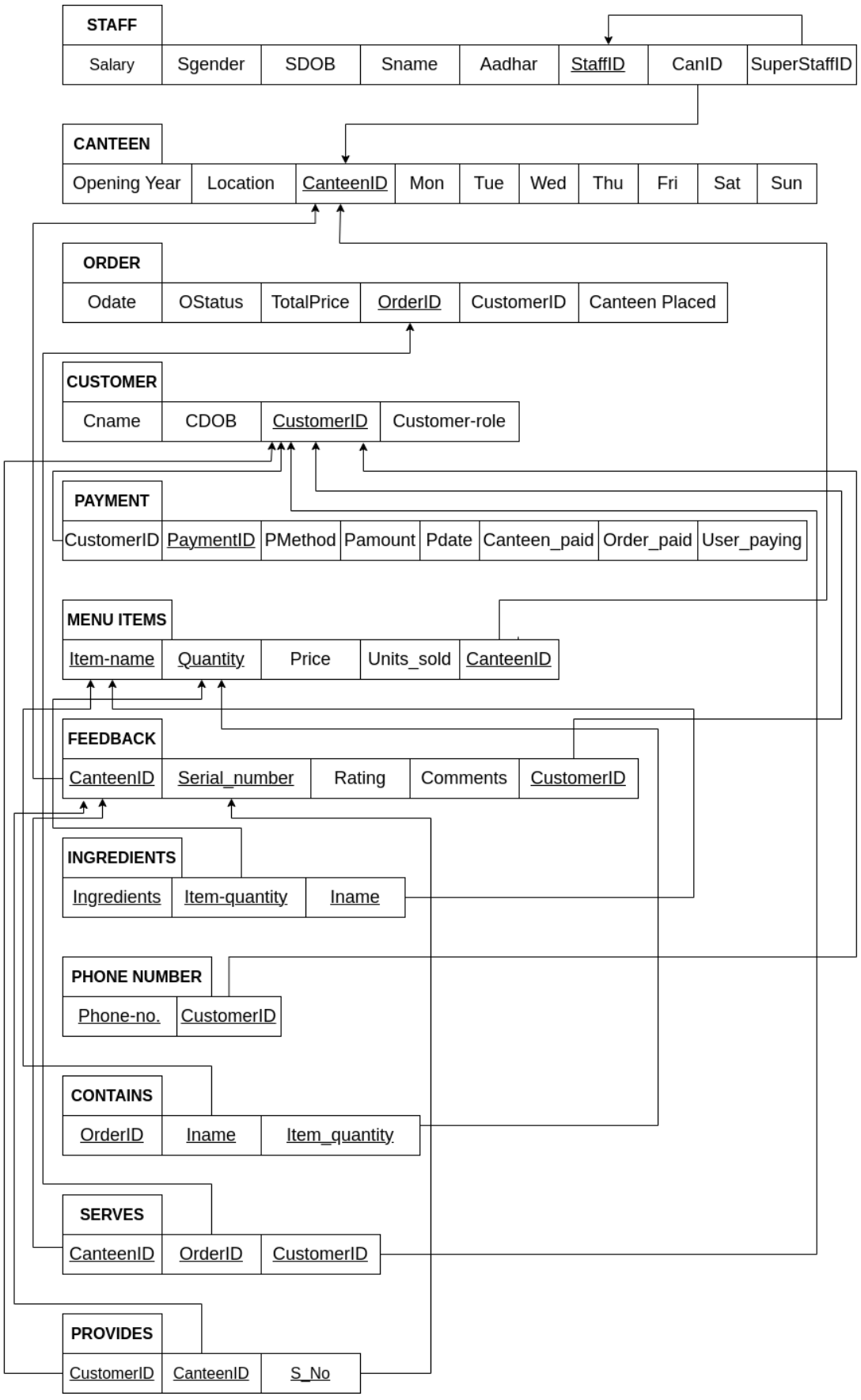
- For every binary M:N relationship type R, we created a new relation S to represent R. We include the primary keys of relations that represent the participating entity types as the foreign key attributes in S; the combination of all these attributes will be the primary key of the relation S**.**
- In our ER diagram, ORDER CONTAINS MENU ITEMS is a binary M:N relationship, so we created a new relation CONTAINS which contain **OrderID** (primary key of ORDER entity type), **Iname** (which refers to Item-name which is one of the primary key of MENU ITEMS) and **Item quantity** (referring to Quantity which is the other primary key of MENU ITEMS) as its foreign keys which are also it's primary keys.

STEP 6: Mapping of multivalued attributes:

- For every multivalued attribute A, we created a new relation R. This relation will include an attribute corresponding to A,as well as the primary key K (as the foreign key attribute of R) of the relation that represents the entity type that has A as the multivalued attribute. In this relation, the combination of A and K will be the primary key of the relation.
- Here we had two multivalued attributes: Ingredients (from MENU ITEMS), Phone Number (from CUSTOMER). We created two new relations: INGREDIENTS, PHONE NUMBER. We added the primary keys of the CUSTOMER, MENU ITEMS relations respectively as foreign keys in the newly added relations.
- This removed the violation of 1-NF (multivalued attribute) from our database.

STEP 7: Mapping of N-ary relationship types:

- For each degree > 2 relationship type, we create a new relationship S to represent the relation R. We include the primary keys of the relations that represent the participating entity types as foreign key attributes in S.
- For example, in the SERVES ternary relationship type, we included the primary keys CanteenID, OrderID,CustomerID from the CANTEEN, ORDER, CUSTOMER respectively, as foreign keys in the SERVES relation and will also act as primary keys for the relation.
- Similarly, in the PROVIDES ternary relationship type,we included CustomerID,CanteenID and S.No. from the CUSTOMER,CANTEEN and FEEDBACK resp, as foreign keys in the PROVIDES relation and will also act as primary keys for the relation.



https://drive.google.com/file/d/1aeRL1B8F3Cn9vQNKeSckzlNAAGC7xPuv/view

1-NF:

Our database does not need to be explicitly changed for it to be in 1-NF as there are no multivalued, composite or nested attributes remaining in the database after we map the ER diagram to a relational model.

We reduced composite attributes to their lowest atomic attributes and multivalued attributes were converted to relations and kept in separate tables.

2-NF:

Our database also does not need to be explicitly changed for it to be in 2-NF as every non prime attribute A in R is fully functionally dependent on the primary key of R.

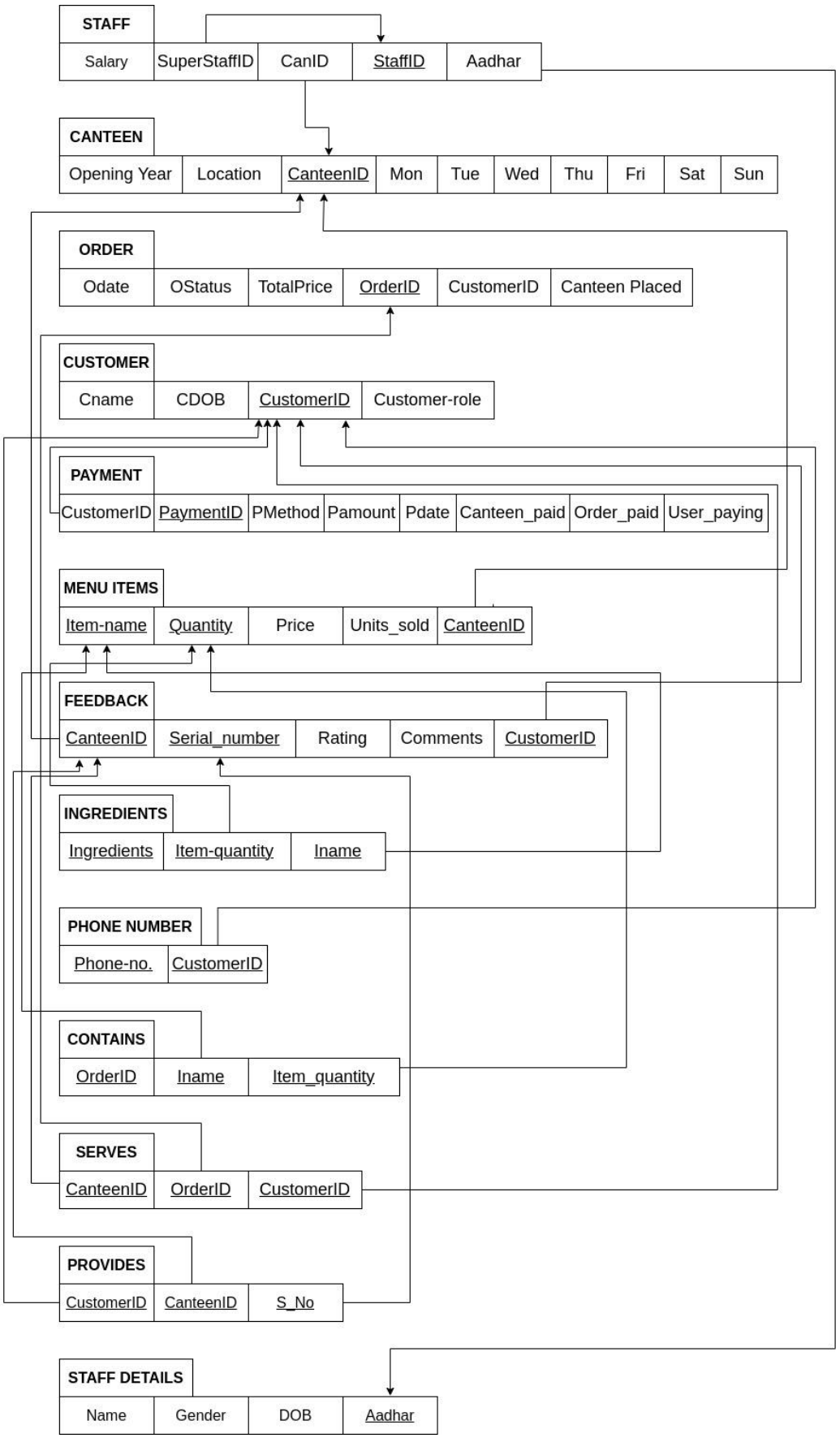
3-NF:

In the relation STAFF, the attributes Sgender, SDOB and Sname are transitively dependent on the primary key, StaffID.
{Aadhar} -> {Sname, Sgender, SDOB} forms an FD and thus violates 3NF. It has to thus be normalized.

We do this by creating a new table STAFF DETAILS which has Aadhar as the primary key and {Sname,Sgender and SDOB} as the other non prime attributes.

The original STAFF table now has StaffID as primary key and Aadhar as a foreign key which references Aadhar from STAFF DETAILS table. Now Aadhar, CanID, Salary, SuperStaffID are the non-prime attributes in the STAFF table.

The diagram for 3-NF can be seen in the next page:



https://drive.google.com/file/d/1JrAlp46pL-8KKr2Bcz52COnnJnBWhi9F/view