**K. J. Somaiya College of Engineering, Mumbai-77**
(Autonomous College Affiliated to University ofMumbai)

**Batch:B2 RollNo.:1711097**
**Experiment / assignment /tutorialNo.2**
**Grade: AA / AB / BB / BC / CC / CD/DD**

**Title:** Mapping ER and EER Modelto Relational Model

**Objective:** To apply mapping techniques to map ERdiagramand EER to its equivalent relational model

**Expected Outcome of Experiment:**

**CO 2:** Convert entity-relationship diagrams into relational tables, populate a relational database and formulate SQL queries on the data Use SQL for creation and query the database.

**Books/ Journals/ Websites referred:**

**G. K. Gupta :"*Database Management Systems*", McGraw –Hill**

1. Korth, Slberchatz, Sudarshan : "Database Systems Concept", 6th Edition , McGrawHill
2. Elmasri and Navathe, "Fundamentals *of* *Database Systems*", 5thEdition, PEARSONEducation.

.

## Relational Model

Relational Model represents the database as a collection of relations. Relational model can be thought of as table of values, each row in the table represents collection of related data values. In the relational model, each row in the table represents the fact that corresponds real world entity or relationship. The table name and column name are used to interpret the meanings of the values in eachrow.

In formal relational model terminology, a row is called tuple, a column header is called an attribute, and table is called a relation. The data type describing the types of values that can appear in each column is represented by a domain of possible values. Thus Relation is set of tuples.

## Procedure for doing the Relation Model (ER to Relational Mapping)

1. **Mapping of RegularEntity**

   - For each regular (strong) entity type in the ER schema, create a relation R that includes all the simple attributes ofE.
   - Choose one of the key attributes of E as the primary key for therelation

2. **Mapping of WeakEntity**

   - For each weak entity type W in the ER schema with owner entity typeE, create a relation R and include all attributes of the weak entity as attributes of the new relationR.
   - Then, include the primary key of the owner entity as foreign key attributes ofR
   - The primary key of R is the *combination of* the primary key(s) of the owner(s) and the partial key of the weak entity type W, ifany.

### 3. Mapping of Binary 1:1 RelationshipTypes

- For each 1:1 relationship type identify the entities participating in the relationship. There are two possible approachesbelow:
- a) Foreign Keyapproach:

  Choose one of the relations and include a foreign key in one relation (S) which is the primary key of the other relation (T). It is better to choose an entity type with *total participation* in the relationship in the role ofS.
- b) Merged relationoption:

  An alternate mapping of a 1:1 relationship type is possible by merging the two entity types and the relationship into a single relation. This may be appropriate when *both participations are total.*

### 4. Mapping of Binary 1:N RelationshipTypes

- For each regular 1:N relationship type R, identify the relation S, which is the entity on the N-side of therelationship.
- Include as foreign key in S the primary key of the relation which is on the 1 side of therelationship
- Include any simple attributes of the 1:N relation type as attributes ofS.

### 5. Mapping of Binary M:N RelationshipTypes

- For each M:N relationship type, *create a new relation* S to represent the relationship
- Include as foreign key attributes in S the primary keys of the entities on each side of the relationship; *the combination of the two primary keys will form the primary key* ofS

- Also include any simple attributes of the M:N relationshiptypeasattributes ofS.

**6. Mapping of MultivaluedAttributes.**

- For each multivalued attribute A, create a new relation. This relation will include an attribute corresponding to the multi-valued attribute, plus the primary key attribute of the relation that has the multi-valued attribute,K
- The primary key attribute of the relation is the foreign key representing the relationship between the entity and the multi-valuedrelation
- The primary key of R is the combination of A andK

**7. Mapping of N-ary RelationshipTypes**

- For each n-ary relationship type R, where n>2, create a new relation Sto represent therelationship.
- Include as foreign key attributes in S the primary keys of the relations that represent the participatingentities
- Also include any simple attributes of the n-ary relationship type as attributes ofS

**8. Options for Mapping Specialization or Generalization**

- Convert each specialization with m subclasses $\{S_1, S_2,\ldots,S_m\}$ and generalized superclass C, where the attributes of C are $\{k,a_1,\ldots a_n\}$ and k is the (primary) key, into relational schemas using one of the four following options:

**Option 8A: Multiple relations-Superclass and subclasses.**

**Option 8B: Multiple relations-Subclass relations only.**

**Option 8C: Single relation with one type attribute.**

**Option 8D: Single relation with multiple type attributes.**

## 9. Mapping of Union Types (Categories).

- For mapping a category whose defining superclass have different keys, it is customary to specify a new key attribute, called a surrogate key, when creating a relation to correspond to the category.
- In the example below, create a relation OWNER to correspond to the OWNER category and include any attributes of the category in this relation. The primary key of the OWNER relation is the surrogate key, which we called OwnerId.

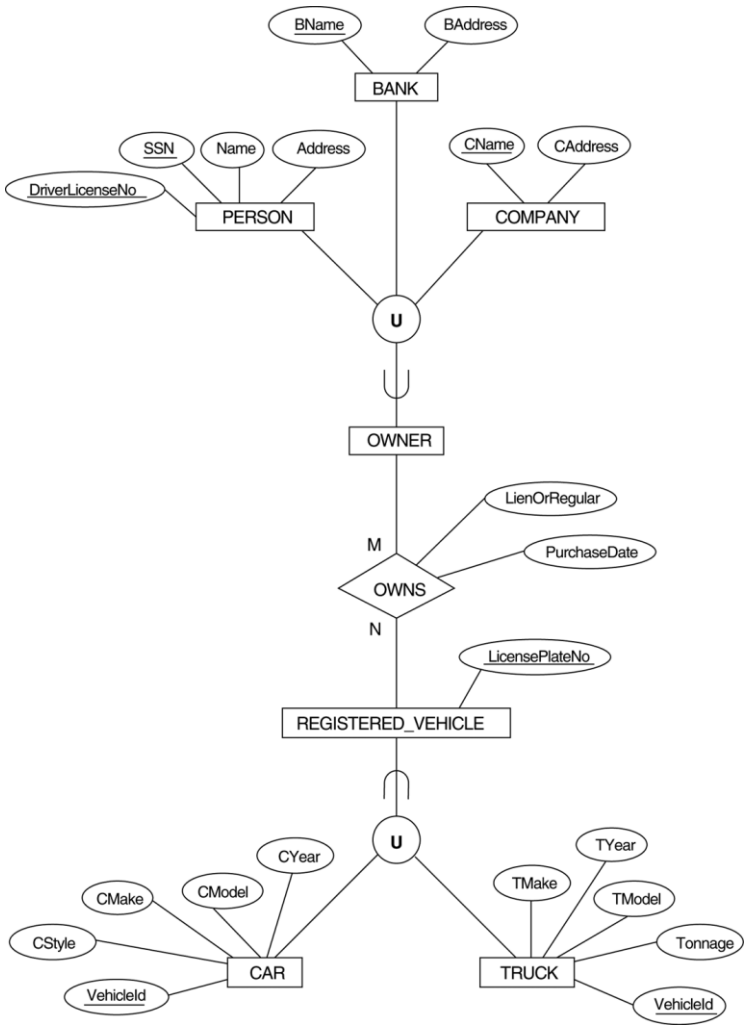



Figure 1:Two categories (union types): OWNER and REGISTERED_VEHICLE.

PERSON

| SSN | DriverLicenseNo | Name | Address | |
|---|---|---|---|---|

BANK

| BName | BAddress | OwnerId |
|---|---|---|

COMPANY

| CName | CAddress | OwnerId |
|---|---|---|

OWNER

| OwnerId |
|---|

REGISTERED_VEHICLE

| VehicleId | LicensePlateNumber |
|---|---|

CAR

| VehicleId | CStyle | CMake | CModel | |
|---|---|---|---|---|

TRUCK

| VehicleId | TMake | TModel | Tonnage | TYear |
|---|---|---|---|---|

OWNS

| OwnerId | VehicleId | PurchaseDate | LienOrRegular |
|---|---|---|---|

Figure 2: Mapping the EER categories (union types) in Figure 1 to relations.

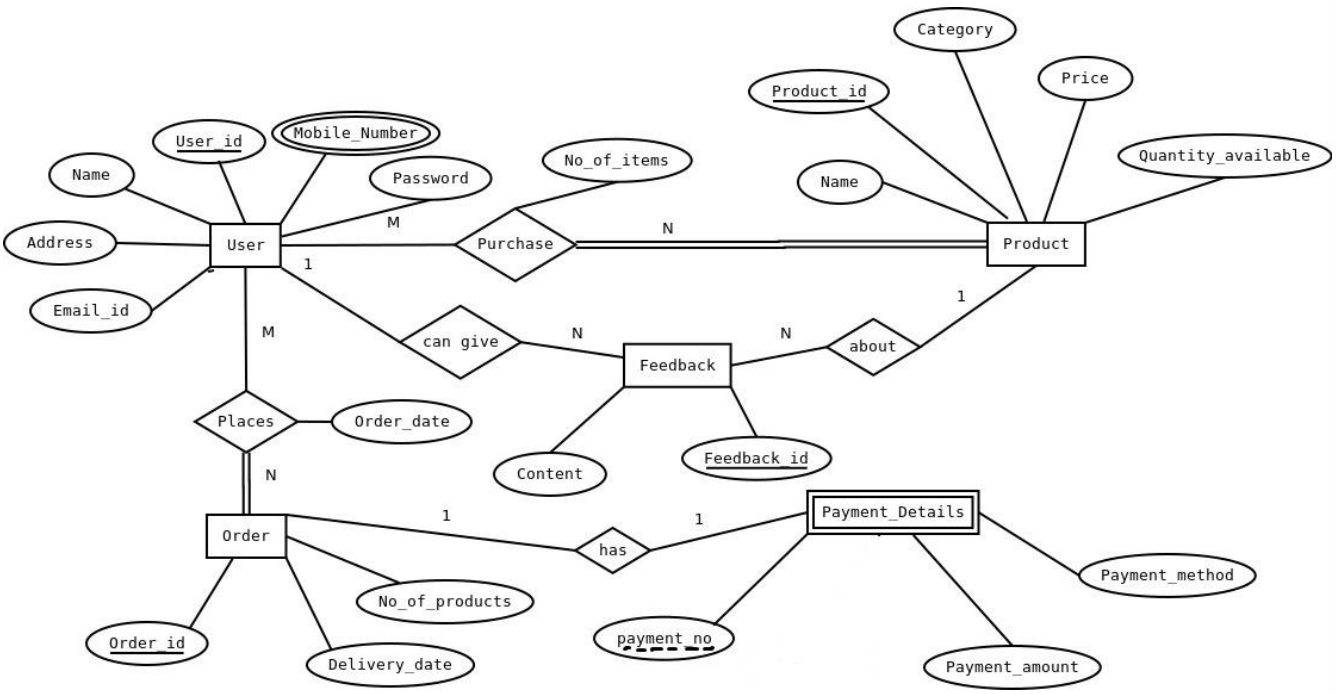
**Case Study considered for Database Design**

**Online SuperMarket Facility**

**Problem Statement:**

Supermarket facility sells items to customers. So we create an online service for customers to place orders for the items available. A space for the new or already registered customers to browse the items like home products, food items, accessories, etc available. The customers have to register in order to place an order. The user information is stored in a database consisting of the user id, name, address, mobile number, email id and password. The user can purchase any number of products and the product information is stored in the database consisting of product id, name, price, category and its available quantity. A user, after selecting products to purchase, places an order where the order date is stored. The order information like the number of products purchased, id of order and delivery date is stored. The order has payment details consisting of payment id, payment number, payment amount and the method of payment. A user can also give many feedbacks about any number of products. The feedback has content and a feedback id.

**ER Diagram**



**K. J. Somaiya College of Engineering, Mumbai-77**
(Autonomous College Affiliated to University ofMumbai)

**Mapping of ER to Relational Model:**

1. **Mapping of regular entity types**:

   Create the relations USER, PRODUCT, FEEDBACK and ORDER in the relational schema corresponding to the regular entities in the ER diagram.

   The attributes Name, Mobile_no, Address, Email_id, User_id and Password are created in User Relation.

   The attributes Name, Category, Price, Quantity_available and Product_id are created in Product Relation.

   The attributes Order_id, No_of_products and Delivery_date are created in Order Relation.

   The attributes Feedback_id and content are created for Feedback Relation.

   User_id, Product_id, feedback_id and Order_id are the primary keys for the relations USER, PRODUCT, FEEDBACK and ORDER.

2. **Mapping of weak Entity types**:

   Create the relation PAYMENT_DETAILS to correspond to the weak entity type PAYMENT_DETAILS.

   Include the primary key Order_id of the ORDER relation as a foreign key attribute of PAYMENT_DETAILS.

   The primary key of the PAYMENT_DETAILS relation is the combination {Order_id, Payment_no} because Payment_no is the partial key of PAYMENT_DETAILS.

3. **Mapping of M:N Relationship types**:

   The primary keys of the ORDER and USER relations are included as foreign keys in PLACES and renamed O_id and U_id, respectively.

   The primary keys of the PRODUCT and USER relations are included as foreign keys in PURCHASE and renamed P_id and U_id, respectively.

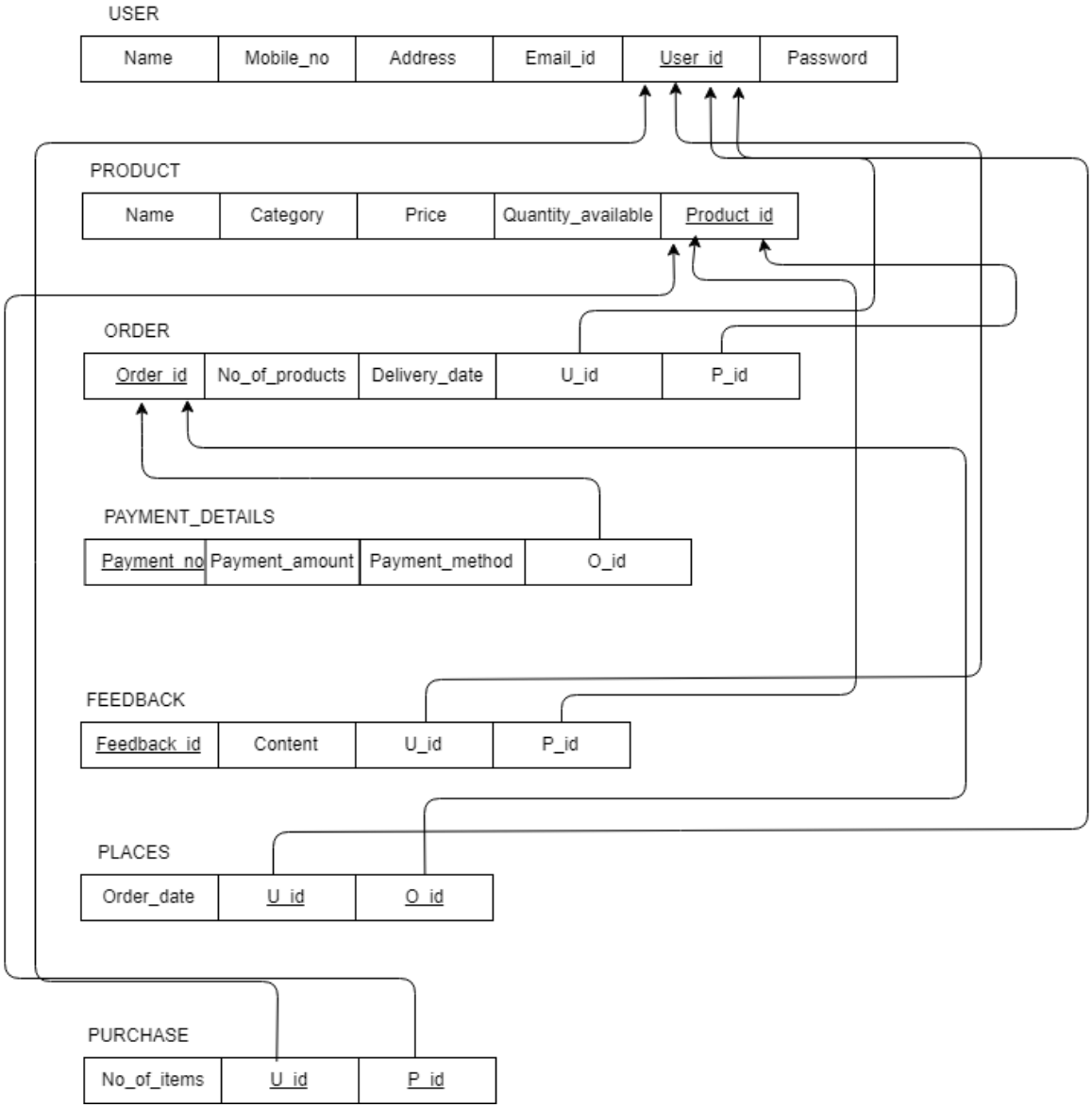
4. **Mapping of 1:N Relationship types**:

   For 1:N relationship types CAN_GIVE and ABOUT.

   For CAN_GIVE we include the primary key User_id of the User relation as foreign key in the FEEDBACK relation and call it U_id.

   For ABOUT we include the primary key Product_id of the Product relation as foreign key in the FEEDBACK relation and call it P_id.

## Relational Model for Project



**Conclusion:** Thus, by following the proper mapping techniques, we have successfully mapped our ER diagram to the Relational Model.