UCS1404 Database Management Systems

MINI-PROJECT

Online Retail Store for Audio Devices

Database Design Presentation

Project by

Adithi Shankar	205001004	II Year	CSE-A
Devisri S R	205001035	II Year	CSE-A
Krithika Swaminathan	205001057	II Year	CSE-A

Project title:

Building an online retail store for audio devices by designing a web application and a supporting database.

Problem Statement:

The aim of this project is to allow users to shop for wearable audio devices online according to their requirements.

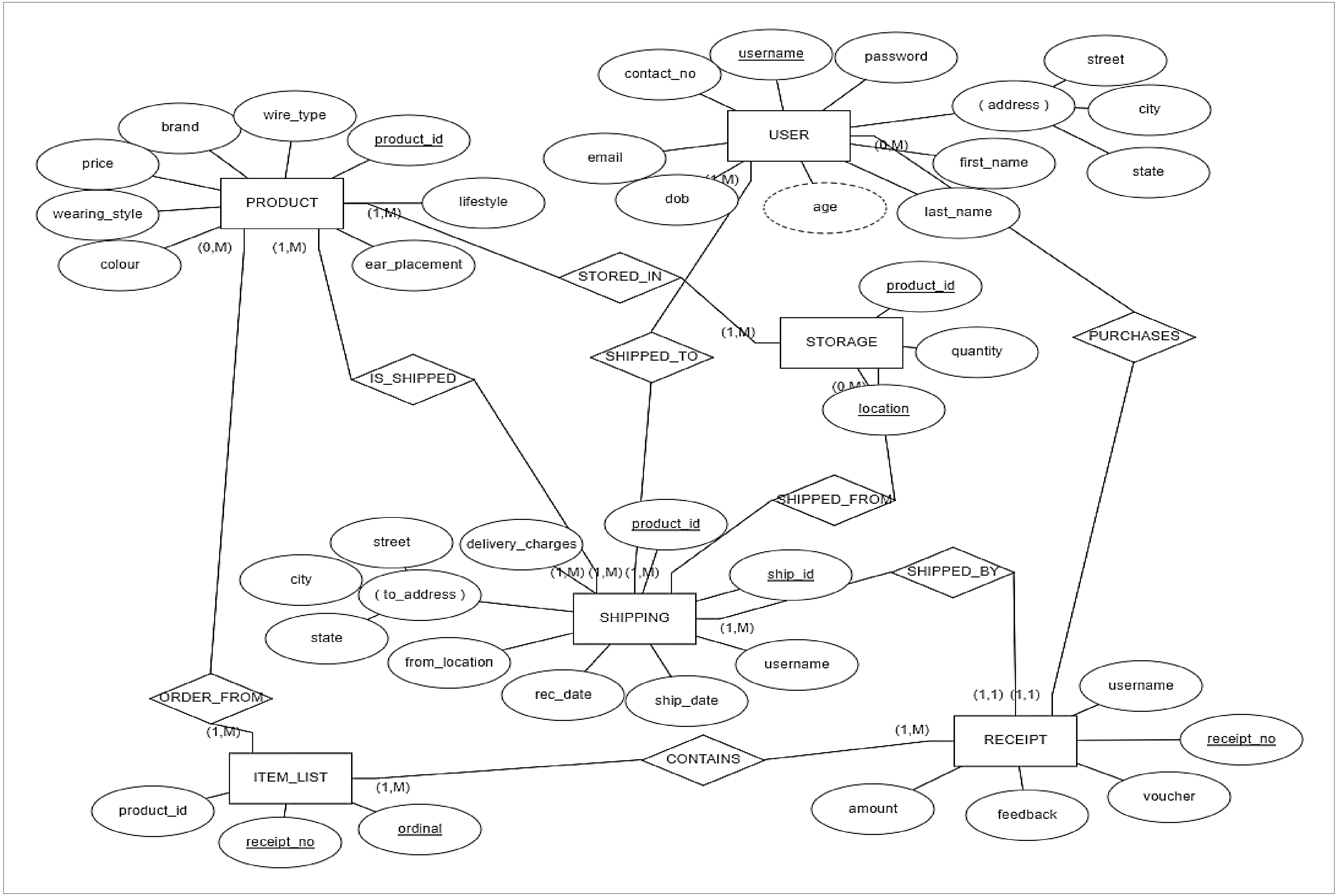
Definition of the problem and Project Summary:

- Audio devices are used by people from all walks of life. Most users order audio devices from online retailers like Croma, Amazon and Flipkart.
- This is largely due to the appealing offers and discounts as well as the ease of no contact buying.
- These stores also enable shoppers to browse through the wide range of products available and narrow their choice down according to their requirements.
- Our project aims to cover all possible specifications by constructing a well-designed database to supplement our web application.
- The following are the various specifications by which users can filter their choice:
  - Earphones, headphones, wireless, neckbands, wired, in ear, on ear, over ear, with mic, without mic, noise cancellation, lifestyle, brand, price range, colour.
- For the continued enhancement of our services, we aim to collect feedback from our users and recommend the best products to them.

Tentative Schema Model

- User: username, password, first_name, last_name, contact_no, email, dob, age (derived), address
- Product: product_id, brand, wearing_style, wire_type, ear_placement, lifestyle, colour, price, discount
- Receipt: receipt_no, username, voucher, feedback, amount
- Item_list: receipt_no, ordinal, product_id
- Shipping: ship_id, product_id, username, ship_date, from_location, rec_date, delivery_charges, to_address
- Storage: prod_id, location, quantity

Entity-Relationship (ER) Diagram



ER to Relational - Conversion Rules

1. **Mapping Regular Entities to Relations:**

- **Simple attributes:** ER attributes map directly onto the relation

  Most of the attributes in our ER model are simple attributes.

- **Composite attributes:** Use only their simple, component as attributes

  The address attribute of the User relation is a composite attribute, broken down into street, city and state.

- **Multivalued Attribute:** Becomes a separate relation with a foreign key taken from the superior entity

  No multivalued attributes are present in our ER model.

2. **Mapping of Weak Entity Types:**

- For each weak entity type W in the ER schema with owner entity type E, create a relation R & include all simple attributes (or simple components of composite attributes) of W as attributes of R.

- Include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).

- 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, if any.

  No weak entity types are present in our ER model.

ER to Relational - Conversion Rules

3. **<u>Mapping of Binary 1:1 Relation Types:</u>**

- **Foreign Key approach:** Choose one of the relations say S and include a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S.

Foreign Key approach is used to convert multiple entities in our ER model to their corresponding schemas in relational. There are no 1:1 relationships in our ER model.

4. **<u>Mapping of Binary 1:N Relationship Types:</u>**

- For each regular binary 1:N relationship type R, identify the relation S that represents the participating entity type at the N-side of the relationship type.
- Include as a foreign key in S (N side), the primary key of the relation (1 side) T, that represents the other entity type participating in R. Include any simple attributes of the 1:N relation type as attributes of S.

Foreign Key approach is used to convert multiple entities in our ER model to their corresponding schemas in relational. These include receipt_no in Item_List, username in Receipt, product_id in Item_List etc.

ER to Relational - Conversion Rules

5. **Mapping of Binary M:N Relationship Types:**

- For each regular binary M:N relationship type R, create a new relation S to represent R.
- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S.
- Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

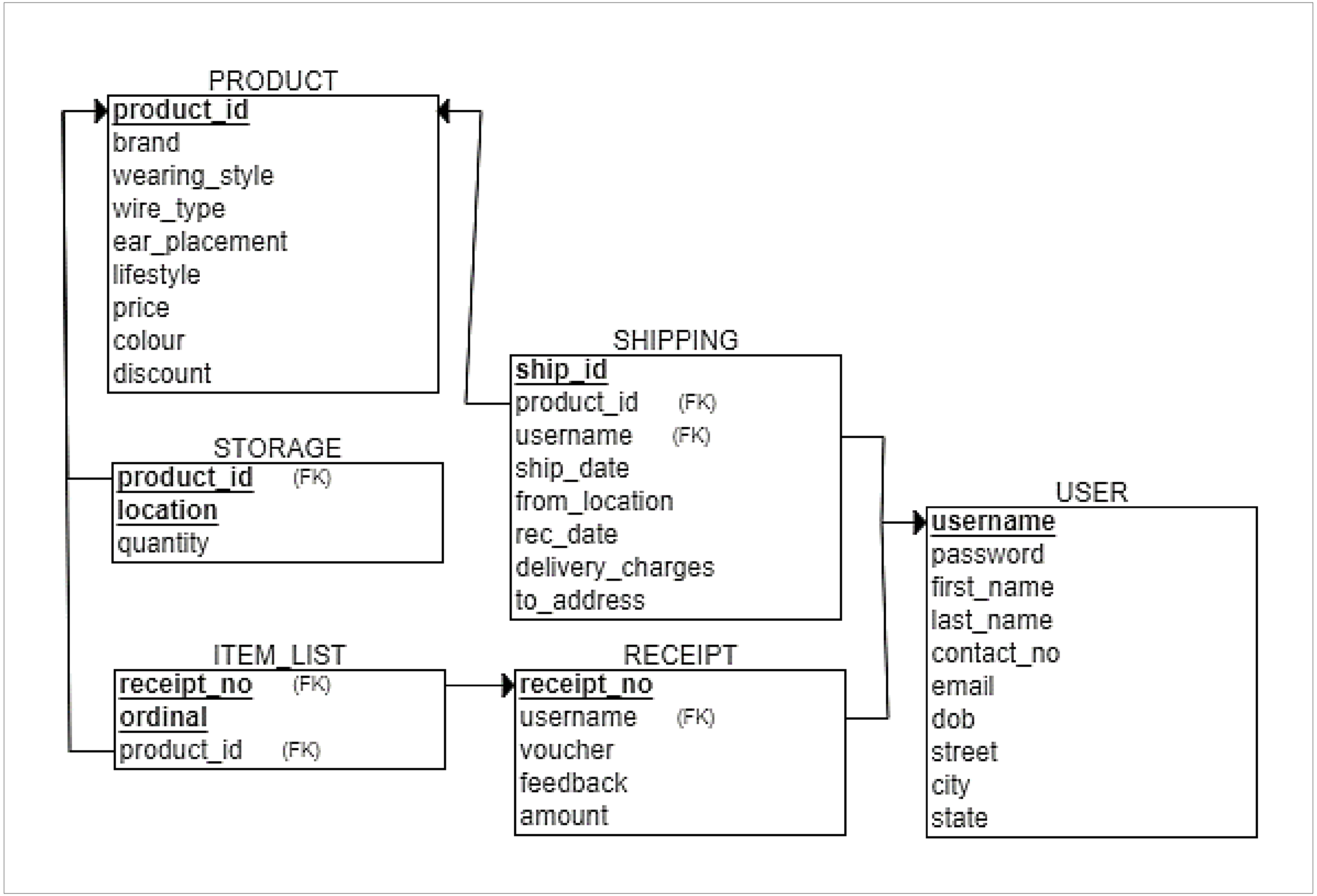
Foreign Key approach is used to convert multiple entities in our ER model to their corresponding schemas in relational. This includes product_id in Storage etc.

6. **Mapping of N-ary Relationship Types:**

- For each n-ary relationship type R, where n>2, create a new relationship S to represent R.
- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.

No N-ary relationship types are present in our ER model.

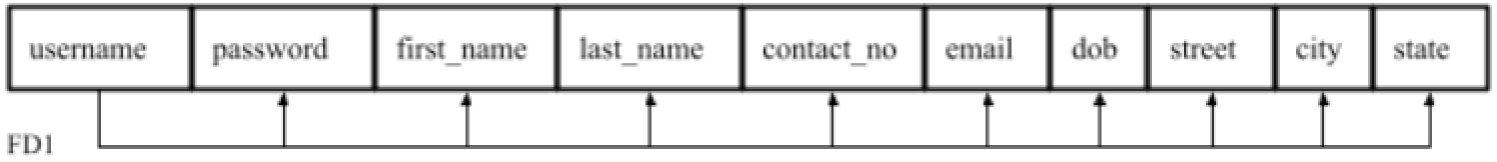
ER Model to Relational Model



Schema Diagram with Functional Dependencies

In each of these relations, each attribute is independent of every other attribute except the attributes acting as determinants.

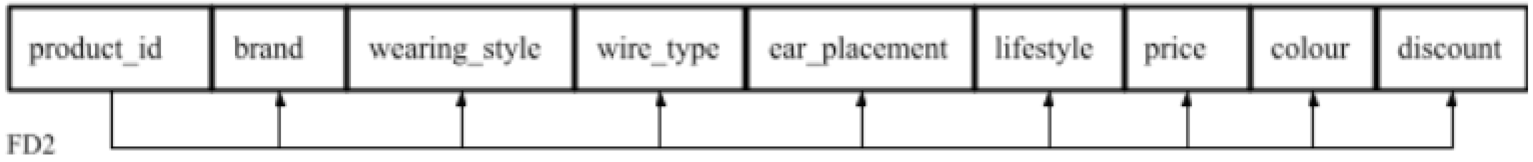
User:



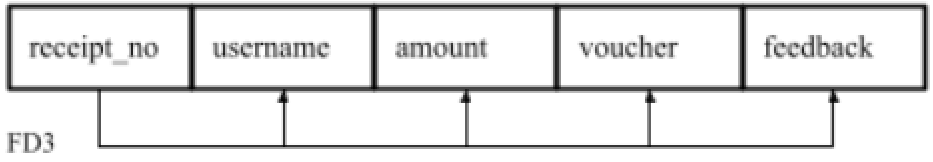
For example, a user's name, contact information and address are all independent of each other.

The product's brand does not determine the wearing style, wire-type, lifestyle or price of the product and the reverse is also not true.

Product:



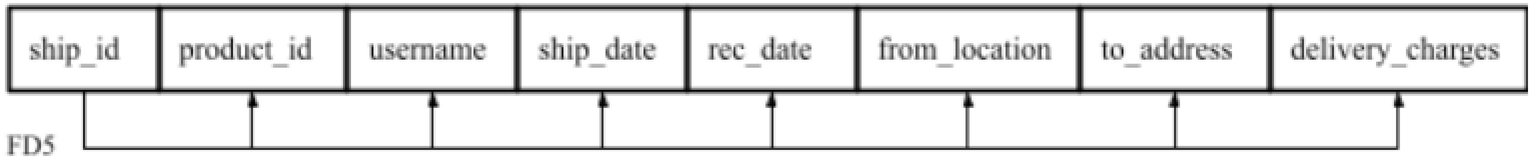
Receipt:



The account information, the amount to be paid and the user's feedback can only be determined from the receipt number

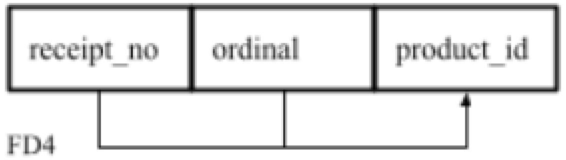
Schema Diagram with Functional Dependencies

Shipping:



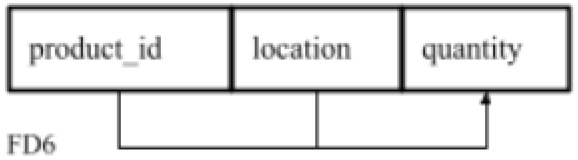
Each shipping can be uniquely identified only by the shipping-id.

Item_list:



Each product in the item-list is identified by the receipt number and its order number in the receipt.

Storage:



The product’s ID and storage location together determine the quantity of the product stored at that location.

Closure – Determining Keys

1. <u>User:</u>

FD1 determinant: {username}

$\{username\}^+$ = {username, password, first_name, last_name, contact_no, email, dob, street, city, state} = R

username is an irreducible superkey, i.e., the primary key

2. <u>Product:</u>

FD2 determinant: {product_id}

$\{product\_id\}^+$ = {product_id, brand, wearing_style, wire_type, ear_placement, lifestyle, price, colour, discount} = R

product_id is an irreducible superkey, i.e. the primary key

3. <u>Receipt:</u>

FD3 determinant: {receipt_no}

$\{receipt\_no\}^+$ = {receipt_no, username, amount, voucher, feedback} = R

receipt_no is an irreducible superkey, i.e., the primary key

Closure – Determining Keys (contd.)

4. <u>Item_list:</u>

FD4 determinant: {receipt_no, ordinal}

{receipt_no, ordinal}$^+$ = {receipt_no, ordinal, product_id} = R (superkey)

{receipt_no}$^+$ = {receipt_no} ≠ R ; {ordinal}$^+$ = {ordinal} ≠ R

{receipt_no, ordinal} is an irreducible superkey, i.e., the primary key

5. <u>Shipping:</u>

FD5 determinant: {ship_id}

{ship_id}$^+$ = {ship_id, product_id, username, ship_date, rec_date, from_location, to_address, delivery_charges} = R

ship_id is an irreducible superkey, i.e., the primary key

6. <u>Storage:</u>

FD6 determinant: {product_id, location}

{product_id, location}+ = {product_id, location, quantity} = R (superkey)

{product_id}+ = {product_id} ≠ R ; {location}+ = {location} ≠ R

{product_id, location} is an irreducible superkey, i.e., the primary key

Normalisation

First Normal Form (1NF):

Definition: *disallows multivalued attributes, composite attributes, and their combinations.*

The only attribute values permitted by 1NF are single atomic (or indivisible) values.

In the User relation,

- There are no multivalued attributes.
- The composite attribute ‘address’ was decomposed into its component attributes, namely, ‘street’, ‘city’ and ‘state’.

User:

username	password	first_name	last_name	contact_no	email	dob	address

User:

username	password	first_name	last_name	contact_no	email	dob	street	city	state

No other relation has multivalued or composite attributes and hence, contain only single-valued attributes.

Every attribute in every relation in the database scheme is now indivisible.

Therefore, the relations USER, PRODUCT, RECEIPT, ITEM_LIST, SHIPPING and STORAGE are all in 1NF.

Normalisation

Second Normal Form (2NF):

Definition: *A relation schema R is in 2NF if every non-prime attribute A in R is fully functionally dependent on the primary key of R.*

Every relation in the database has only fully functional dependencies. Every non-prime attribute is fully functionally dependent on a prime attribute of the relation.

Therefore, the relations USER, PRODUCT, RECEIPT, ITEM_LIST, SHIPPING and STORAGE are all in 2NF.

Third Normal Form (3NF):

Definition: *A relation schema R is in 3NF if it satisfies 2NF and no non-prime attribute of R is transitively dependent on the primary key.*

There are no transitive dependencies in the relations. The determinants are all prime attributes of their respective relations.

Therefore, the relations USER, PRODUCT, RECEIPT, ITEM_LIST, SHIPPING and STORAGE are all in 3NF.

Normalisation

<u>**Boyce-Codd Normal Form (BCNF):**</u>

Definition: *A relation schema R is in BCNF if whenever a non-trivial functional dependency X → A holds in R, then X is a superkey of R.*

For every relation in the database, the determinant of each functional dependency is a superkey of the relation, as shown by the closure set of the primary key attributes.

Therefore, the relations USER, PRODUCT, RECEIPT, ITEM_LIST, SHIPPING and STORAGE are all in BCNF.

Database Schema - Online Retail Store for Audio Devices

The database schema after identification of the functional dependencies and primary keys in each relation, followed by normalisation, is as follows:

User:

username	password	first_name	last_name	contact_no	email	dob	street	city	state

FD1

Product:

product_id	brand	wearing_style	wire_type	ear_placement	lifestyle	price	colour	discount

FD2

Receipt:

receipt_no	username	amount	voucher	feedback

FD3

Storage:

product_id	**location**	quantity

FD6

Item_list:

receipt_no	**ordinal**	product_id

FD4

Shipping:

ship_id	product_id	username	ship_date	rec_date	from_location	to_address	delivery_charges

FD5

Thank You!