**FINAL DELIVERABLE:**

**SELLING ON AMAZON:**

**Structural Business Rules:**

* A Seller can sell many products.
* A product is sold by a single seller.
* Each product belongs to a category.
* A product belongs to a single category.
* A category can have many products.
* Product should have category ID, Name, Price, Quantity.
* An inventory belongs to a single seller.
* A seller can have a single inventory.
* A customer can place many orders.
* An order belongs to a single customer.
* Customer should have username, address, phone number and email address.
* An order can have many products.
* A product belongs to a single order.
* Order should belong to a particular shipping speed.
* A return request belongs to one and only customer.
* A customer can have zero too many return requests.
* Return request should have a Person ID and Product ID.
* A customer can have zero too many customers service request.
* A customer service request belongs to one and only one customer.
* Each order should have a tracking ID.
* A product list belongs to only one order.
* A order list has many order to product pairs.

**CONCEPTUAL ERD:**

Diagram

Description automatically generated

**LOGICAL ERD:**

Diagram, schematic

Description automatically generated

**TABLES CREATION:**

CREATE TABLE prod\_category (

category\_name VARCHAR(50) NOT NULL,

categ\_descript VARCHAR(30),

PRIMARY KEY (category\_name)

);

CREATE TABLE products (

product\_id DECIMAL(10) NOT NULL,

category\_name VARCHAR(50) NOT NULL,

prod\_name VARCHAR(30) NOT NULL,

prod\_descript VARCHAR(30),

price DECIMAL(10) NOT NULL,

quantity DECIMAL(10) NOT NULL,

PRIMARY KEY (product\_id),

FOREIGN KEY (category\_name) REFERENCES prod\_category(category\_name)

);

CREATE TABLE person (

person\_id DECIMAL(10) NOT NULL,

username VARCHAR(50) NOT NULL,

phone VARCHAR(30) NOT NULL,

email VARCHAR(30),

address VARCHAR(30),

PRIMARY KEY (person\_id)

);

CREATE TABLE seller (

person\_id DECIMAL(10) NOT NULL,

brand\_name VARCHAR(30),

rating DECIMAL(10),

PRIMARY KEY (person\_id),

FOREIGN KEY (person\_id) REFERENCES person(person\_id)

);

CREATE TABLE buyer (

person\_id DECIMAL(10) NOT NULL,

subscription VARCHAR(30),

PRIMARY KEY (person\_id),

FOREIGN KEY (person\_id) REFERENCES person(person\_id)

);

CREATE TABLE inventory (

person\_id DECIMAL(10) NOT NULL,

product\_id DECIMAL(10) NOT NULL,

product\_count DECIMAL(10),

FOREIGN KEY (person\_id) REFERENCES person(person\_id),

FOREIGN KEY (product\_id) REFERENCES products(product\_id)

);

CREATE TABLE shipping (

shipping\_id DECIMAL(10) NOT NULL,

status VARCHAR(30) NOT NULL,

service\_provider VARCHAR(30),

PRIMARY KEY (shipping\_id)

);

CREATE TABLE orders (

order\_id DECIMAL(10) NOT NULL,

person\_id DECIMAL(10) NOT NULL,

shipping\_id DECIMAL(10) NOT NULL,

order\_status VARCHAR(30),

PRIMARY KEY (order\_id),

FOREIGN KEY (shipping\_id) REFERENCES shipping(shipping\_id),

FOREIGN KEY (person\_id) REFERENCES buyer(person\_id)

);

CREATE TABLE returns (

return\_id DECIMAL(10) NOT NULL,

person\_id DECIMAL(10) NOT NULL,

product\_id DECIMAL(10) NOT NULL,

reason VARCHAR(30),

status VARCHAR(30),

PRIMARY KEY (return\_id),

FOREIGN KEY (person\_id) REFERENCES buyer(person\_id),

FOREIGN KEY (product\_id) REFERENCES products(product\_id)

);

CREATE TABLE customer\_service (

service\_id DECIMAL(10) NOT NULL,

person\_id DECIMAL(10) NOT NULL,

serv\_type VARCHAR(30),

discript VARCHAR(200) NOT NULL,

status VARCHAR(30),

PRIMARY KEY (service\_id),

FOREIGN KEY (person\_id) REFERENCES buyer(person\_id)

);

CREATE TABLE orders\_list (

order\_id DECIMAL(10) NOT NULL,

product\_id DECIMAL(10) NOT NULL,

quantity DECIMAL(10),

FOREIGN KEY (order\_id) REFERENCES orders(order\_id),

FOREIGN KEY (product\_id) REFERENCES products(product\_id)

);

**Stored procedures:**

CREATE OR REPLACE PROCEDURE ADD\_person(

person\_id\_arg IN DECIMAL,

phone\_arg IN VARCHAR,

email\_arg IN VARCHAR,

address\_arg IN VARCHAR,

firstname\_arg IN VARCHAR,

lastname\_arg IN VARCHAR)

IS

BEGIN

INSERT INTO person

(person\_id, phone, email, address, username)

VALUES

(person\_id\_arg, phone\_arg, email\_arg, address\_arg, firstname\_arg || ' ' || lastname\_arg);

END;

/

CREATE OR REPLACE PROCEDURE ADD\_seller(

person\_id IN DECIMAL,

brand\_name IN VARCHAR,

rating IN DECIMAL)

IS

BEGIN

INSERT INTO seller

(person\_id, brand\_name, rating)

VALUES

(person\_id, brand\_name, rating);

END;

create or replace PROCEDURE ADD\_inventory(

person\_id\_ARG IN DECIMAL,

product\_id\_arg IN decimal,

product\_count\_arg IN decimal)

IS

BEGIN

INSERT INTO inventory

(person\_id, product\_id, product\_count)

VALUES

(person\_id\_ARG, product\_id\_arg, product\_count\_arg);

END;

create or replace PROCEDURE ADD\_buyer(

person\_id\_arg DECIMAL,

subscription\_arg VARCHAR)

IS

BEGIN

INSERT INTO buyer

(person\_id, subscription)

VALUES

(person\_id\_arg, subscription\_arg);

END;

create or replace PROCEDURE ADD\_customer\_service(

service\_id\_arg DECIMAL,

person\_id\_arg DECIMAL,

serv\_type\_arg VARCHAR,

discript\_arg VARCHAR,

status\_arg VARCHAR)

IS

BEGIN

INSERT INTO customer\_service

(service\_id, person\_id, serv\_type, discript, status)

VALUES

(service\_id\_arg, person\_id\_arg, serv\_type\_arg, discript\_arg, status\_arg);

END;

create or replace PROCEDURE ADD\_SHIPPING(

shipping\_id\_arg DECIMAL,

status\_arg VARCHAR,

service\_provider\_arg VARCHAR)

IS

BEGIN

INSERT INTO shipping

(shipping\_id, status, service\_provider)

VALUES

(shipping\_id\_arg, status\_arg, service\_provider\_arg);

END;

create or replace PROCEDURE ADD\_orders\_list(

order\_id\_arg DECIMAL,

product\_id\_arg DECIMAL,

quantity\_arg decimal)

IS

BEGIN

INSERT INTO orders\_list

(order\_id, product\_id, quantity)

VALUES

(order\_id\_arg, product\_id\_arg, quantity\_arg);

END;

CREATE OR REPLACE PROCEDURE ADD\_PRODUCTS(

product\_id\_ARG IN DECIMAL,

category\_name\_arg IN VARCHAR,

prod\_name\_arg IN VARCHAR,

prod\_descript\_ARG IN VARCHAR,

price\_ARG IN DECIMAL,

quantity\_ARG IN DECIMAL)

IS

BEGIN

INSERT INTO products

(product\_id, category\_name, prod\_name, prod\_descript, price, quantity)

VALUES

(product\_id\_ARG, category\_name\_arg, prod\_name\_arg, prod\_descript\_ARG, price\_ARG, quantity\_ARG);

END;

/

**Inserting values:**

BEGIN

ADD\_CATEGORY('Electronics', 'related to electronics');

END;

/

BEGIN

ADD\_CATEGORY('Computers', 'gadgets related to computers');

END;

/

BEGIN

ADD\_CATEGORY('Furniture', 'related to comfort');

END;

/

BEGIN

ADD\_PRODUCTS(1, 'Computers', 'acer', 'nitro 5', 600, 10);

END;

/

BEGIN

ADD\_PRODUCTS(2, 'Electronics', 'mouse', 'gaming mouse', 20, 15);

END;

/

BEGIN

ADD\_PRODUCTS(3, 'Electronics', 'cable', 'ethernet cable', 15, 20);

END;

/

BEGIN

ADD\_PRODUCTS(4, 'Electronics', 'cpu', 'with high-end processor', 500, 5);

END;

/

BEGIN

ADD\_PRODUCTS(5, 'Computers', 'hp', '2gb ram', 20, 12);

END;

/

BEGIN

ADD\_PRODUCTS(6, 'Furniture', 'table', 'glossy finish', 25, 8);

END;

/

BEGIN

ADD\_PRODUCTS(7, 'Furniture', 'closet', 'mobile closet', 30, 5);

END;

/

BEGIN

ADD\_PRODUCTS(8, 'Electronics', 'self-driving video camera', 'follows a subject', 180, 3);

END;

/

BEGIN

ADD\_PRODUCTS(9, 'Computers', 'holographic keyboard', 'tracks virtual key presses', 30, 6);

END;

/

BEGIN

UPDATE prod\_category SET categ\_descript = 'related to electronics' WHERE category\_name = 'Electronics';

UPDATE prod\_category SET categ\_descript = 'gadgets related to computers' WHERE category\_name = 'Computers';

UPDATE prod\_category SET categ\_descript = 'related to comfort' WHERE category\_name = 'Furniture';

END;

/

BEGIN

ADD\_person(1, '7745197610', 'AChirra@clarku.edu', '600 main st', 'Adharsh');

END;

/

BEGIN

ADD\_person(2, '7749945426', 'PWyner@clarku.edu', '600 main st', 'Peter', 'Wyner');

END;

/

BEGIN

ADD\_person(3, '7749945326', 'RJulia@clarku.edu', '600 main st', 'Julia', 'Roberts');

END;

/

BEGIN

ADD\_person(4, '7749946326', 'RAlex@clarku.edu', '340 main st', 'Alex', 'Roberts');

END;

/

BEGIN

ADD\_person(5, '7749946316', 'RKatie@clarku.edu', '345 main st', 'Katie', 'Roberts');

END;

/

BEGIN

ADD\_person(6, '7749946315', 'Rjhon@clarku.edu', '15 main st', 'Jhon', 'Roberts');

END;

/

BEGIN

ADD\_person(7, '7749946338', 'jmaverick@clarku.edu', '10 main st', 'Maverick', 'J');

END;

/

BEGIN

ADD\_person(8, '7749946356', 'ctom@clarku.edu', '36 main st', 'Tom', 'Cruise');

END;

/

BEGIN

ADD\_person(9, '7749946389', 'mtobey@clarku.edu', '79 gem st', 'Tobey', 'Maguire');

END;

/

BEGIN

ADD\_person(10, '7749946320', 'jdwayne@clarku.edu', '10 gem st', 'Dwayne', 'Johnson');

END;

/

BEGIN

ADD\_seller(1, 'Adharsh', 5);

END;

/

BEGIN

ADD\_seller(2, 'PW', 5);

END;

/

BEGIN

ADD\_seller(3, 'RJ', 4);

END;

/

BEGIN

ADD\_buyer(1, 'prime');

END;

/

BEGIN

ADD\_buyer(2, 'prime');

END;

/

BEGIN

ADD\_buyer(3, 'prime');

END;

/

BEGIN

ADD\_buyer(4, '');

END;

/

BEGIN

ADD\_buyer(5, '');

END;

/

BEGIN

ADD\_buyer(6, '');

END;

/

BEGIN

ADD\_buyer(7, 'prime');

END;

/

BEGIN

ADD\_buyer(8, 'prime');

END;

/

BEGIN

ADD\_buyer(9, '');

END;

/

BEGIN

ADD\_buyer(10, 'prime');

END;

/

BEGIN

ADD\_inventory(1, 9, 7);

END;

/

BEGIN

ADD\_inventory(2, 5, 6);

END;

/

BEGIN

ADD\_inventory(1, 8, 5);

END;

/

BEGIN

ADD\_SHIPPING(0, 'not shipped', '');

END;

/

BEGIN

ADD\_SHIPPING(1, 'shipping confirmed', 'DHL');

END;

/

BEGIN

ADD\_SHIPPING(2, 'shipping confirmed', 'UPS');

END;

/

BEGIN

ADD\_SHIPPING(3, 'in process', 'DHL');

END;

/

BEGIN

ADD\_orders(1, 1, 0, 'order confirmed');

END;

/

BEGIN

ADD\_orders(2, 4, 2, 'shipped');

END;

/

BEGIN

ADD\_orders(3, 2, 0, 'order confirmed');

END;

/

BEGIN

ADD\_orders(4, 6, 3, 'shipping');

END;

/

BEGIN

ADD\_orders(5, 3, 2, 'shipping');

END;

/

BEGIN

ADD\_orders(6, 5, 3, 'shipping');

END;

/

BEGIN

ADD\_orders(7, 10, 0, 'order confirmed');

END;

/

BEGIN

ADD\_orders(8, 7, 0, 'order confirmed');

END;

/

BEGIN

ADD\_orders(9, 9, 1, 'shipping');

END;

/

BEGIN

ADD\_orders(10, 8, 2, 'shipping');

END;

/

BEGIN

ADD\_orders\_list(1, 8, 1);

END;

/

BEGIN

ADD\_orders\_list(2, 2, 1);

END;

/

BEGIN

ADD\_orders\_list(2, 5, 2);

END;

/

BEGIN

ADD\_orders\_list(2, 1, 5);

END;

/

BEGIN

ADD\_orders\_list(3, 9, 3);

END;

/

BEGIN

ADD\_orders\_list(4, 9, 1);

END;

/

BEGIN

ADD\_orders\_list(5, 2, 1);

END;

/

BEGIN

ADD\_orders\_list(5, 9, 2);

END;

/

BEGIN

ADD\_orders\_list(6, 2, 1);

END;

/

BEGIN

ADD\_orders\_list(6, 8, 2);

END;

/

BEGIN

ADD\_orders\_list(7, 3, 1);

END;

/

BEGIN

ADD\_orders\_list(8, 4, 3);

END;

/

BEGIN

ADD\_orders\_list(8, 6, 1);

END;

/

BEGIN

ADD\_orders\_list(9, 5, 7);

END;

/

BEGIN

ADD\_orders\_list(10, 7, 1);

END;

/

BEGIN

ADD\_returns(1, 1, 5, 'poor performance', 'approved');

END;

/

BEGIN

ADD\_returns(2, 3, 4, 'defective product', 'pending');

END;

/

BEGIN

ADD\_returns(3, 5, 2, 'wrong item received', 'approved');

END;

/

BEGIN

ADD\_customer\_service(1, 2, 'missing items', 'mouse pad missing in the package delivered', 'waiting for approval');

END;

/

BEGIN

ADD\_customer\_service(2, 3, 'damaged product', 'product arrived with visible damages', 'in process');

END;

/

BEGIN

ADD\_customer\_service(3, 5, 'return request', 'customer wants to return the product', 'approved');

END;

/

BEGIN

ADD\_customer\_service(4, 7, 'refund request', 'customer requests a refund for the order', 'pending');

END;

/

**Updated Values Inserting:**

CREATE OR REPLACE PROCEDURE update\_person(

person\_id\_arg IN DECIMAL,

firstname\_arg IN VARCHAR,

lastname\_arg IN VARCHAR)

IS

BEGIN

UPDATE person

SET firstname = firstname\_arg,

lastname = lastname\_arg

WHERE person\_id = person\_id\_arg;

END;

/

BEGIN

update\_person(1, 'Adharsh', 'Chirra');

END;

/

BEGIN

update\_person(2, 'Peter', 'Wyner');

END;

/

BEGIN

update\_person(3, 'Julia', 'Roberts');

END;

/

BEGIN

update\_person(4, 'Alex', 'Roberts');

END;

/

BEGIN

update\_person(5, 'Katie', 'Roberts');

END;

/

BEGIN

update\_person(6, 'Jhon', 'Roberts');

END;

/

BEGIN

update\_person(7, 'Maverick', 'J');

END;

/

BEGIN

update\_person(8, 'Tom', 'Cruise');

END;

/

BEGIN

update\_person(9, 'Tobey', 'Maguire');

END;

/

BEGIN

update\_person(10, 'Dwayne', 'Johnson');

END;

/

**Aspect-1:**

***Get those 'Computers' or 'Electronics' goods that cost less than thirty.***

SELECT \* FROM PRODUCTS

WHERE PRICE < 30 AND (CATEGORY\_NAME='Computers' or CATEGORY\_NAME='Electronics');

A screenshot of a computer

Description automatically generated

**Aspect-2:**

***When the number of products in the inventory is <=  to  11, retrieve the names of those products.***

SELECT prod\_name FROM products

JOIN inventory ON products.product\_id = inventory.product\_id AND inventory.product\_count <= 11;

A screenshot of a computer

Description automatically generated

**Aspect-3:**

***Using the person table, extract last names & the number of occurrences. Sort by last name and remove entries with a count higher than three.***

SELECT LASTNAME, COUNT(\*) FROM PERSON

GROUP BY LASTNAME

HAVING COUNT(\*) > 3;

A screenshot of a computer

Description automatically generated

**Aspect-4:**

***Getting the address, first and last names, & person ID for purchases containing three or more goods.***

SELECT orders.person\_id, person.firstname, person.lastname, person.address

FROM orders

INNER JOIN person ON orders.person\_id = person.person\_id

WHERE order\_id IN (

SELECT order\_id FROM orders\_list

WHERE product\_id IN (

SELECT product\_id FROM orders\_list

GROUP BY product\_id

HAVING COUNT(\*) >= 3

)

);

A screenshot of a computer

Description automatically generated

**Aspect-5:**

***Obtain shipping details about orders placed by customers with Prime Membership.***

SELECT shipping\_id, status, service\_provider FROM shipping

WHERE shipping\_id IN (

SELECT shipping\_id FROM orders

WHERE person\_id IN (

SELECT person\_id FROM buyer

WHERE subscription = 'prime'

)

);

A screenshot of a computer

Description automatically generated

**INDEX CREATION:**

***Query to Optimize:***

Imagine a situation in which we routinely run queries based on the person's table to obtain data using the username. This query is often used, particularly in activities involving users.

***Justification:***

Setting an index on the username column may greatly improve retrieval time and increase the efficiency of queries utilizing this column. Without doing a full table search, indexing enables the database system to quickly find and retrieve the rows connected to a certain username.

CREATE INDEX idx\_username ON person(username);

A screenshot of a computer

Description automatically generated

***explanation:***

Faster access to records using the username field will be possible thanks to the newly built index, idx\_username. By using the index, the database engine may find the rows that belong to a specific username immediately, saving it from having to scan the whole person table sequentially. Consequently, query efficiency is enhanced, particularly for SELECT queries that contain criteria pertaining to the username field.

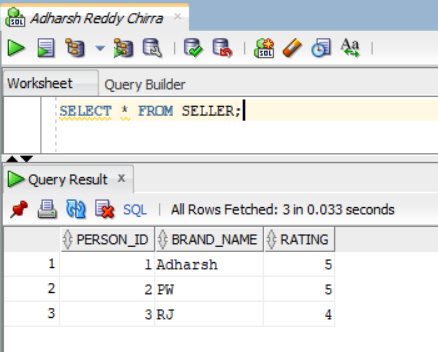
**Screenshots:**

SELECT \* FROM PERSON;

A screenshot of a computer

Description automatically generated

SELECT \* FROM SELLER;



SELECT \* FROM INVENTORY;

A screenshot of a computer

Description automatically generated

SELECT \* FROM BUYER;

A screenshot of a computer

Description automatically generated

SELECT \* FROM CUSTOMER\_SERVICE;

A screenshot of a computer

Description automatically generated

SELECT \* FROM SHIPPING;

A screenshot of a computer

Description automatically generated

SELECT \* FROM ORDERS\_LIST;

A screenshot of a computer

Description automatically generated

SELECT \* FROM PROD\_CATEGORY;

A screenshot of a computer

Description automatically generated

SELECT \* FROM PRODUCTS;

A screenshot of a computer

Description automatically generated

SELECT \* FROM RETURNS;

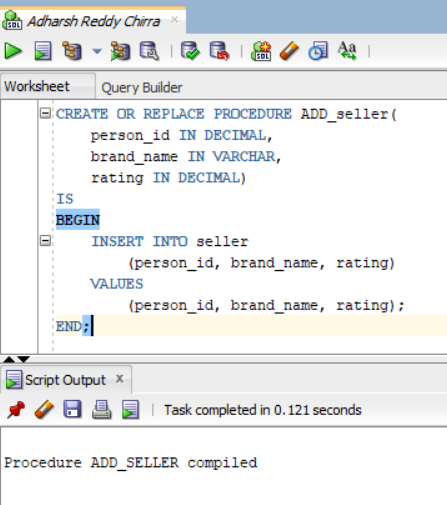
A screenshot of a computer

Description automatically generated

***STORED PROCEDURES SCREENSHOTS:***

***A screenshot of a computer

Description automatically generated***

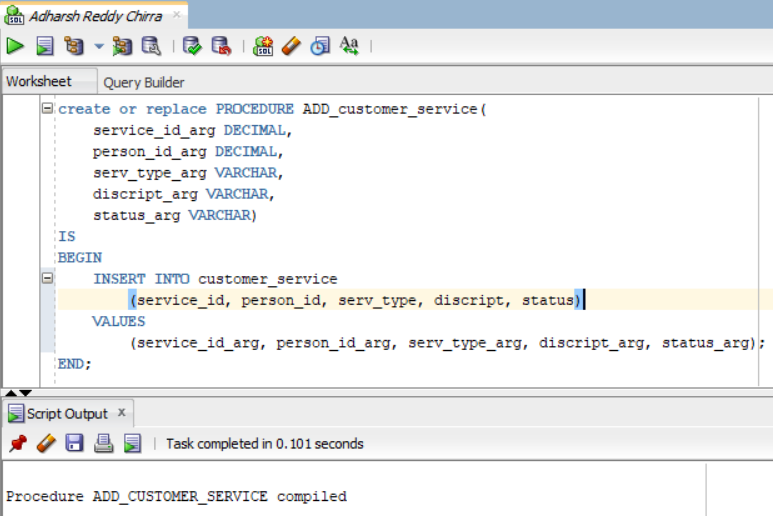
******

***A screenshot of a computer

Description automatically generated***

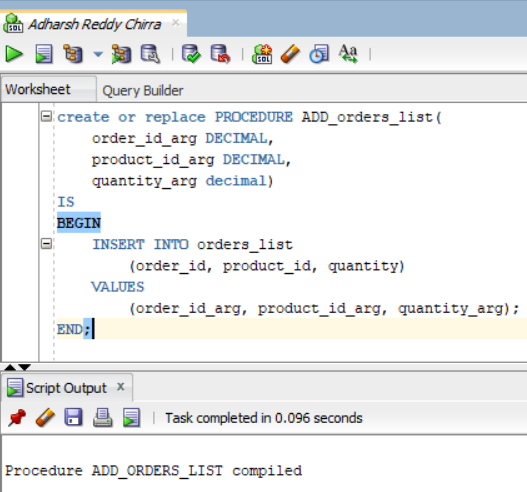
***A screenshot of a computer

Description automatically generated***

******

***A screenshot of a computer

Description automatically generated***

******

***A screenshot of a computer

Description automatically generated***

***Updated stored procedures:***

***A screenshot of a computer

Description automatically generated***

***Updated values inserting:***

A screenshot of a computer program

Description automatically generated