

ORACLE DATABASE DESIGN FOR SHOPPING CART APP

TABLE

USERS TABLE

```
CREATE TABLE users (  
  user_id    NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  
  username   VARCHAR2(50) NOT NULL  
);
```

USER_ID	USERNAME
1	user1
2	user2
3	user3

USER DETAILS TABLE

```
CREATE TABLE user_details (  
  user_id      NUMBER PRIMARY KEY,  
  password     VARCHAR2(100) NOT NULL,  
  email        VARCHAR2(100) NOT NULL,  
  first_name   VARCHAR2(50) NOT NULL,  
  last_name    VARCHAR2(50) NOT NULL,  
  address_id   NUMBER  
    REFERENCES address (address_id),  
  CONSTRAINT fk_user_details_users FOREIGN KEY (user_id) REFERENCES  
  users (user_id)  
);
```

USER_ID	PASSWORD	EMAIL	FIRST_NAME	LAST_NAME	ADDRESS_ID
1	92a2dbc0d5df071...	user1@example.com	John	Doe	1
2	6cf615d5bcaac77..	user2@example.com	Jane	Doe	2
3	5906ac361a137e2...	user3@example.com	Alice	Smith	3

ADDRESS TABLE

```
CREATE TABLE address (  
  address_id      NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY  
KEY,  
  street          VARCHAR2(100),  
  city            VARCHAR2(50),  
  state           VARCHAR2(50),  
  country         VARCHAR2(50),  
  postal_code     VARCHAR2(20)  
);
```

ADDRESS_ID	STREET	CITY	STATE	COUNTRY	POSTAL_CODE
1	123 Elm Street	Springfield	State1	Country1	12345
2	456 Maple Avenue	Springfield	State2	Country2	23456
3	789 Oak Lane	Springfield	State3	Country3	34567

PRODUCT CATEGORIES TABLE

```
CREATE TABLE product_categories (  
  category_id     NUMBER  
    GENERATED BY DEFAULT ON NULL AS IDENTITY,  
  category_name   VARCHAR2 (50),  
  description     VARCHAR2 (200)  
);
```

CATEGORY_ID	CATEGORY_NAME	DESCRIPTION
1	Electronics	Electronic products
2	Apparel	Clothing and fashion products
3	Footwear	Shoes and accessories
4	Home	Home appliances and decoration products
5	Cosmetics	Cosmetics and beauty products
6	Sports	Sports products
7	Toys	Toys and game materials
8	Books	Books and literature products
9	Music	Music instruments and equipment

CATEGORY_ID	CATEGORY_NAME	DESCRIPTION
10	Watch	Watches and accessories

PRODUCTS TABLE

```
CREATE TABLE products (
  product_id      NUMBER
    GENERATED BY DEFAULT ON NULL AS IDENTITY,
  product_name    VARCHAR2 (100) NOT NULL,
  category_id     NUMBER
    REFERENCES product_categories (category_id),
  unit_price      NUMBER,
  stock_quantity  NUMBER,
  product_details VARCHAR2 (200)
);
```

PRODUCT_ID	PRODUCT_NAME	CATEGORY_ID	UNIT_PRICE	STOCK_QUANTITY	PRODUCT_DETAILS
1	Laptop Model A	1	1200,5	50	High performance laptop
2	Smartphone	1	899,99	75	State-of-the-art smartphone
3	Wireless Headphones	1	79,95	100	Wireless headphones with Bluetooth connection
4	Tablet	1	499	45	Portable tablet device
5	Smart Watch	1	299,5	60	Smart watch that measures health
6	T-shirt	2	25,99	150	Cotton men's t-shirt
7	Shirt	2	35,5	120	Classic style shirt
8	Sweater	2	45,75	80	Warm winter sweater
9	Pants	2	39,95	100	Women's trousers
10	Jeans	2	49,99	90	Men's denim jeans

ORDERS TABLE

```
CREATE TABLE orders (
  order_id      NUMBER
```

```

        GENERATED BY DEFAULT ON NULL AS IDENTITY,
user_id      NUMBER
    REFERENCES users (user_id),
order_date   DATE NOT NULL,
address_id   NUMBER
    REFERENCES address (address_id),
order_status VARCHAR2 (20) NOT NULL
);

```

ORDER_ID	USER_ID	ORDER_DATE	ADDRESS_ID	ORDER_STATUS
1	2	09/01/2024	2	CANCELLED
2	2	09/01/2024	2	COMPLETE
3	2	09/01/2024	2	OPEN

ORDER DETAILS TABLE

```

CREATE TABLE order_details (
    detail_id  NUMBER
        GENERATED BY DEFAULT ON NULL AS IDENTITY,
    order_id   NUMBER
        REFERENCES orders (order_id)
    NOT NULL,
    product_id NUMBER
        REFERENCES products (product_id)
    NOT NULL,
    quantity   NUMBER NOT NULL,
    unit_price  NUMBER NOT NULL
);

```

DETAIL_ID	ORDER_ID	PRODUCT_ID	QUANTITY	UNIT_PRICE
1	1	3	2	79,95
2	1	5	3	60,95

SHOPPING CART TABLE

```

CREATE TABLE shopping_cart (
    cart_id    NUMBER
        GENERATED BY DEFAULT ON NULL AS IDENTITY,
    user_id    NUMBER
        REFERENCES users (user_id),
    created_at  DATE NOT NULL,

```

```
    expires_at DATE
);
```

CART_ID	USER_ID	CREATED_AT	EXPIRES_AT
10	2	09/01/2024	16/01/2024
11	3	09/01/2024	16/01/2024

PRODUCT SHOPPING CART TABLE

```
CREATE TABLE product_shoppingcart (
    cart_id    NUMBER
        REFERENCES shopping_cart (cart_id),
    product_id NUMBER
        REFERENCES products (product_id),
    quantity   NUMBER NOT NULL
);
```

CART_ID	PRODUCT_ID	QUANTITY
10	3	2
11	5	2

USER FAVORITES TABLE

```
CREATE TABLE user_favorites (
    favorite_id    NUMBER
        GENERATED BY DEFAULT ON NULL AS IDENTITY,
    user_id        NUMBER
        REFERENCES users (user_id),
    product_id     NUMBER
        REFERENCES products (product_id),
    favorite_date  DATE NOT NULL
);
```

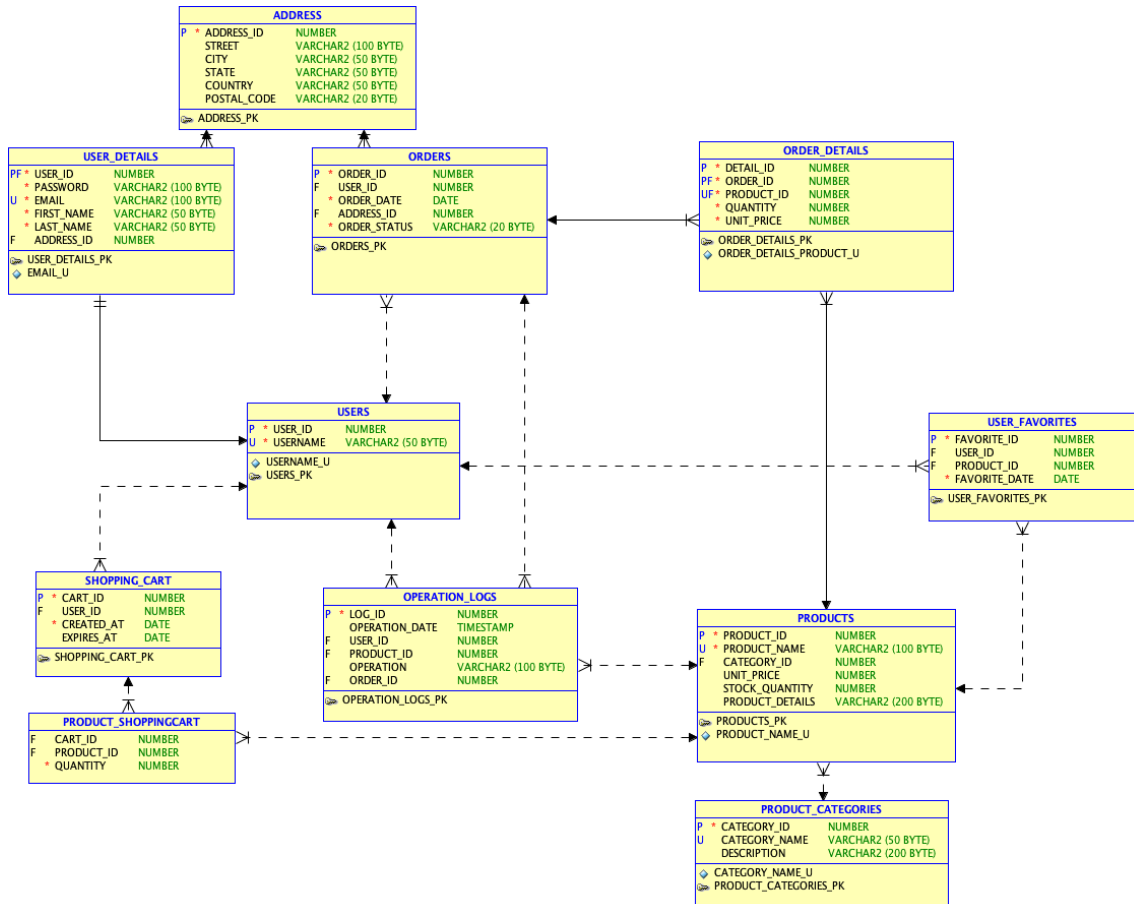
FAVORITE_ID	USER_ID	PRODUCT_ID	FAVORITE_DATE
2	1	15	09/01/2024
3	2	3	09/01/2024

OPERATION LOGS TABLE

```
CREATE TABLE operation_logs (  
  log_id          NUMBER  
    GENERATED BY DEFAULT ON NULL AS IDENTITY,  
  operation_date  TIMESTAMP DEFAULT current_timestamp,  
  user_id         NUMBER  
    REFERENCES users (user_id),  
  product_id      NUMBER  
    REFERENCES products (product_id),  
  operation       VARCHAR2 (100),  
  order_id        NUMBER  
    REFERENCES orders (order_id)  
);
```

LOG_ID	OPERATION_DATE	USER_ID	PRODUCT_ID	OPERATION	ORDER_ID
44	09/01/2024 22:11:35,844580000	2		User confirmed order.	1
45	09/01/2024 22:11:49,394769000	2		User listed their orders.	
46	09/01/2024 22:12:16,890789000	2		User listed their OPEN orders.	
47	09/01/2024 22:13:09,386946000	2		User cancelled their order. Order id: 1	1
48	09/01/2024 22:13:18,527752000	2		User listed their orders.	
49	09/01/2024 22:16:46,749906000	2		User listed their CANCELLED orders.	
50	09/01/2024 22:21:19,497163000	1	15	User added the product to favorites.	
51	09/01/2024 22:21:33,167272000	1		User listed their favorites.	
52	09/01/2024 22:21:55,388016000	1	15	User removed the product from favorites.	
53	09/01/2024 22:22:07,697081000	1	15	User added the product to favorites.	

SCHEME DIAGRAM



TRIGGERS

```
CREATE SEQUENCE users_sequence;
```

```
CREATE OR REPLACE TRIGGER users_on_insert BEFORE
  INSERT ON users
  FOR EACH ROW
BEGIN
  SELECT users_sequence.NEXTVAL
  INTO :new.user_id
  FROM dual;
END;
```

```
CREATE SEQUENCE product_categories_sequence;  
CREATE OR REPLACE TRIGGER product_categories_on_insert
```

```
CREATE SEQUENCE product_sequence;  
CREATE OR REPLACE TRIGGER product_on_insert
```

```
CREATE SEQUENCE shopping_cart_sequence;  
CREATE OR REPLACE TRIGGER shopping_cart_on_insert
```

```
CREATE SEQUENCE order_details_sequence;  
CREATE OR REPLACE TRIGGER order_details_on_insert
```

```
CREATE SEQUENCE orders_sequence;  
CREATE OR REPLACE TRIGGER orders_on_insert
```

```
CREATE SEQUENCE user_favorites_sequence;  
CREATE OR REPLACE TRIGGER user_favorites_on_insert
```

```
CREATE SEQUENCE operation_logs_sequence;  
CREATE OR REPLACE TRIGGER operation_logs_on_insert
```

FUNCTIONS

```
CREATE OR REPLACE FUNCTION get_sha256_hash (  
    p_input IN VARCHAR2  
) RETURN VARCHAR2 AS  
    v_hash_raw RAW (2000);  
    v_hash_hex VARCHAR2 (4000);  
BEGIN  
    v_hash_raw := dbms_crypto.hash (utl_i18n.string_to_raw (p_input,  
'AL32UTF8'), dbms_crypto.hash_sh256);  
  
    v_hash_hex := lower (rawtohex (v_hash_raw));  
    RETURN v_hash_hex;  
END get_sha256_hash;
```

```
CREATE OR REPLACE FUNCTION calculate_cart_total (  
    p_user_id IN NUMBER  
) RETURN NUMBER IS  
    v_total_amount NUMBER := 0;  
    v_cart_count NUMBER := 0;  
BEGIN  
    /* Kullanıcının sepetindeki ürünlerin toplam tutarını hesapla*/  
    SELECT COUNT (*)  
    INTO v_cart_count  
    FROM shopping_cart sc
```



```

WHERE sc.user_id = p_user_id;
    /* Eğer kullanıcıya ait bir sepet yoksa veya sepet boşsa, hata
mesajı ver*/
    IF v_cart_count != 0 THEN
        SELECT SUM (ps.quantity * p.unit_price)
        INTO v_total_amount
        FROM product_shoppingcart ps
        JOIN products p
        ON ps.product_id = p.product_id
        JOIN shopping_cart sc
        ON ps.cart_id = sc.cart_id
        WHERE sc.user_id = p_user_id;
        RETURN v_total_amount;
    ELSE
        dbms_output.put_line ('The user''s shopping cart is empty or does
not exist. ');
        RETURN NULL;
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        /* Hata durumunda burada gerekli işlemler yapılabilir,
hata mesajı yazdırılabilir.*/
        dbms_output.put_line ('Hata: ' || sqlerrm);
        RETURN NULL; /* Hata durumunda NULL döndür*/
END calculate_cart_total;

```

PACKAGES

USER OPERATION PACKAGE

```

FUNCTION login_user (p_username IN VARCHAR2, p_password IN VARCHAR2)
RETURN BOOLEAN;

FUNCTION get_user_id (p_username IN VARCHAR2, p_password IN VARCHAR2)
RETURN NUMBER;

PROCEDURE change_password (p_username IN VARCHAR2, p_new_password IN
VARCHAR2);

PROCEDURE register_user (p_username IN VARCHAR2, p_password IN
VARCHAR2, p_email IN VARCHAR2, p_first_name IN VARCHAR2, p_last_name IN
VARCHAR2, p_street IN VARCHAR2, p_city IN VARCHAR2, p_state IN VARCHAR2,
p_country IN VARCHAR2 , p_postal_code IN VARCHAR2);

```

SHOPPING CART OPERATION PACKAGE

```
PROCEDURE add_product (p_user_id IN NUMBER, p_product_id IN NUMBER,  
p_quantity IN NUMBER);
```

```
PROCEDURE remove_product (p_user_id IN NUMBER, p_product_id IN  
NUMBER);
```

```
PROCEDURE clear_cart (p_user_id IN NUMBER);
```

```
PROCEDURE print_user_cart (p_user_id IN NUMBER);
```

ORDER OPERATION PACKAGE

```
PROCEDURE confirm_order (p_user_id IN NUMBER);
```

```
PROCEDURE print_user_orders (p_user_id IN NUMBER);
```

```
PROCEDURE cancel_order (p_order_id IN NUMBER);
```

```
PROCEDURE print_order_by_status (p_user_id IN NUMBER, p_order_status  
VARCHAR2);
```

```
PROCEDURE complete_order (p_order_id IN NUMBER);
```

FAVORITE OPERATION PACKAGE

```
PROCEDURE add_favorites (p_user_id IN NUMBER, p_product_id IN NUMBER);
```

```
PROCEDURE remove_favorites (p_user_id IN NUMBER, p_product_id IN  
NUMBER);
```

```
PROCEDURE get_user_favorites (p_user_id IN NUMBER);
```

PRODUCT FILTERS PACKAGE

```
PROCEDURE list_top_sold_products;
```

```
PROCEDURE list_top_favorited_products;
```

```
PROCEDURE list_products_by_category;
```

```
PROCEDURE list_category;
```

```
PROCEDURE list_products_by_price (order_by_price VARCHAR2);
```

```
PROCEDURE list_products_by_alphabet (order_by_name VARCHAR2);
```

```
PROCEDURE list_products_by_stock (order_by_stock VARCHAR2);
```

```
PROCEDURE list_products_by_category (category_name VARCHAR2);
```

PRINT PACKAGE

```
PROCEDURE print_user_info (p_user_id IN NUMBER, p_header IN VARCHAR2);
```

```
PROCEDURE print_error (p_header IN VARCHAR2);
```

ORDER HISTORY OPERATION PACKAGE

```
PROCEDURE save_order_history (p_user_id IN NUMBER);
```

OPERATION LOGGING PACKAGE

```
PROCEDURE logoperation (p_user_id IN NUMBER, p_product_id IN NUMBER,  
p_operation IN VARCHAR2, p_order_id IN NUMBER);
```