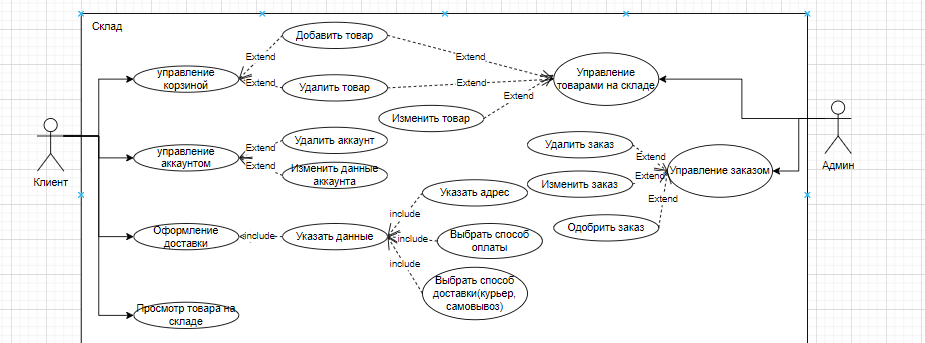
2. Описать целевую аудиторию и стратегии использования интернет-приложения.

Целевая аудитория интернет-приложения для управления складом может включать в себя различные организации и предприятия, занимающиеся складской деятельностью, такие как производственные компании, розничные сети, логистические компании и дистрибьюторы.

Стратегии использования интернет-приложения для управления складом могут включать:

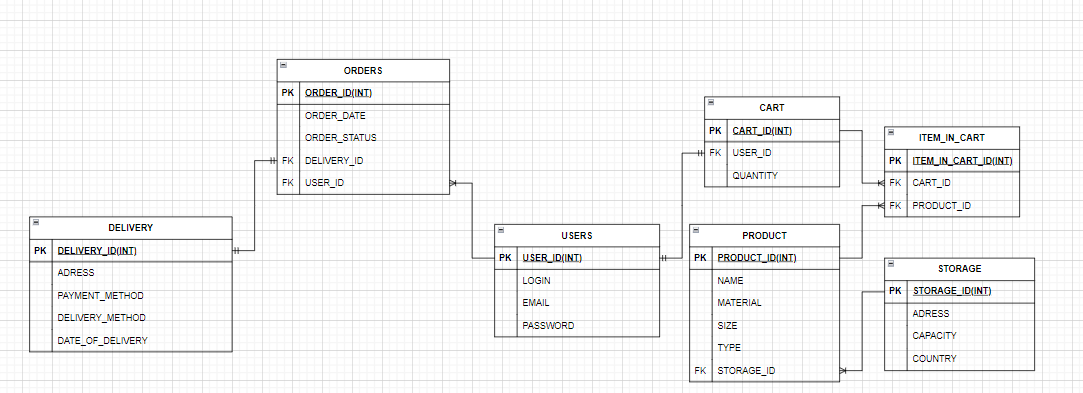
* Управление запасами
* Отслеживание и маршрутизация грузов
* Управление заказами и поставками

3. Разработать UML диаграммы для каждой роли приложения.

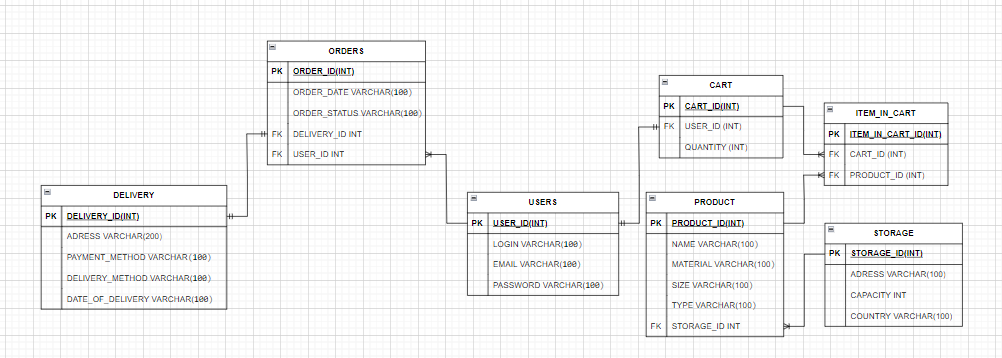


4. основные сущности

Логическая



Физическая схема



5. Преобразовать сущности в таблицы базы данных согласно выбранной модели.

|  |
| --- |
| -- Creating tables  CREATE TABLE DB\_USERS(  USER\_ID NUMBER PRIMARY KEY,  LOGIN VARCHAR2(100),  EMAIL VARCHAR2(100),  PASSWORD VARCHAR2(100)  );  GRANT CREATE TABLE TO admin;  SELECT username FROM dba\_users;  CREATE TABLE CART(  CART\_ID NUMBER PRIMARY KEY,  USER\_ID NUMBER UNIQUE,  QUANTITY NUMBER,  CONSTRAINT USER\_ID\_CONSTRAINT FOREIGN KEY (USER\_ID) REFERENCES DB\_USERS(USER\_ID)  );  drop table cart  drop table ITEM\_IN\_CART;  CREATE TABLE ITEM\_IN\_CART(  ITEM\_IN\_CART\_ID NUMBER PRIMARY KEY,  CART\_ID NUMBER,  PRODUCT\_ID NUMBER,  CONSTRAINT CART\_ID\_CONSTRAINT FOREIGN KEY (CART\_ID) REFERENCES CART(CART\_ID),  CONSTRAINT PRODUCT\_ID\_CONSTRAINT FOREIGN KEY (PRODUCT\_ID) REFERENCES PRODUCT(PRODUCT\_ID)  );  CREATE TABLE PRODUCT(  PRODUCT\_ID NUMBER PRIMARY KEY,  PRODUCT\_NAME VARCHAR2(100),  PRODUCT\_MATERIAL VARCHAR2(100),  PRODUCT\_SIZE VARCHAR2(100),  PRODUCT\_TYPE VARCHAR2(100),  STORAGE\_ID INT,  CONSTRAINT STORAGE\_ID\_CONSTRAINT FOREIGN KEY (STORAGE\_ID) REFERENCES STORAGE(STORAGE\_ID),  );  CREATE TABLE STORAGE(  STORAGE\_ID NUMBER PRIMARY KEY,  ADRESS VARCHAR2(100),  COUNTRY VARCHAR2(100),  CAPACITY NUMBER  );  DROP TABLE PRODUCT;  CREATE TABLE DELIVERY(  DELIVERY\_ID NUMBER PRIMARY KEY,  ADDRESS VARCHAR2(200),  PAYMENT\_METHOD VARCHAR2(100),  DELIVERY\_METHOD VARCHAR2(100),  DATE\_OF\_DELIVERY VARCHAR2(100)  );  CREATE TABLE ORDERS(  ORDER\_ID NUMBER PRIMARY KEY,  ORDER\_DATE VARCHAR2(100),  ORDER\_STATUS VARCHAR2(100),  DELIVERY\_ID NUMBER,  USER\_ID NUMBER,  CONSTRAINT USER\_ID\_CONSTRAINT1 FOREIGN KEY (USER\_ID) REFERENCES DB\_USERS(USER\_ID),  CONSTRAINT DELIVERY\_ID\_CONSTRAINT1 FOREIGN KEY (DELIVERY\_ID) REFERENCES DELIVERY(DELIVERY\_ID)  );  DROP TABLE ORDERS;  ---------------- DB\_USERS TABLE ---------------------  CREATE OR REPLACE PROCEDURE INSERT\_USER(  userId IN NUMBER,  login IN VARCHAR2,  email IN VARCHAR2,  password IN VARCHAR2  )  AS  BEGIN  INSERT INTO DB\_USERS (USER\_ID, LOGIN, EMAIL, PASSWORD)  VALUES (userId, login, email, password);  END;  /  BEGIN  INSERT\_USER(1, 'test\_user1', 'example@example.com', 'password123');  COMMIT;  END;  /  select \*from DB\_USERS;  CREATE OR REPLACE FUNCTION GET\_USERS  RETURN SYS\_REFCURSOR  AS  users\_cursor SYS\_REFCURSOR;  BEGIN  OPEN users\_cursor FOR  SELECT USER\_ID, LOGIN, EMAIL, PASSWORD  FROM DB\_USERS;    RETURN users\_cursor;  END;  DECLARE  users\_cursor SYS\_REFCURSOR;  BEGIN  users\_cursor := GET\_USERS();    DBMS\_SQL.RETURN\_RESULT(users\_cursor);  END;  /  ---------------- CART TABLE ---------------------  CREATE OR REPLACE PROCEDURE INSERT\_PRODUCT\_TO\_CART(  cart\_id IN NUMBER,  user\_id IN NUMBER,  quantity IN NUMBER  )  AS  BEGIN  INSERT INTO CART VALUES (cart\_id, user\_id, quantity);  END;  /  BEGIN  INSERT\_PRODUCT\_TO\_CART(2, 1, 180);  COMMIT;  END;  /  CREATE OR REPLACE FUNCTION GET\_CART\_INFO\_BY\_ID(  cart\_id IN NUMBER  )  RETURN SYS\_REFCURSOR  AS  cart\_cursor SYS\_REFCURSOR;  BEGIN  OPEN cart\_cursor FOR  SELECT \*  FROM CART  WHERE CART.CART\_ID = cart\_id  FETCH FIRST 1 ROW ONLY;    RETURN cart\_cursor;  END;  /  DECLARE  cart\_cursor SYS\_REFCURSOR;  BEGIN  cart\_cursor := GET\_CART\_INFO\_BY\_ID(1);    DBMS\_SQL.RETURN\_RESULT(cart\_cursor);  END;  /  ---------------- ITEM\_IN\_CART TABLE ---------------------  CREATE OR REPLACE PROCEDURE INSERT\_ITEM\_IN\_CART(  item\_in\_cart\_id IN NUMBER,  cart\_id IN NUMBER,  product\_id IN NUMBER  )  AS  BEGIN  INSERT INTO ITEM\_IN\_CART VALUES (item\_in\_cart\_id, cart\_id, product\_id);  END;  /  BEGIN  INSERT\_ITEM\_IN\_CART(2, 1, 1);  COMMIT;  END;  CREATE OR REPLACE FUNCTION GET\_ITEMS\_IN\_CART  RETURN SYS\_REFCURSOR  AS  items\_cursor SYS\_REFCURSOR;  BEGIN  OPEN items\_cursor FOR  SELECT \*  FROM ITEM\_IN\_CART;    RETURN items\_cursor;  END;  /  DECLARE  items\_cursor SYS\_REFCURSOR;  BEGIN  items\_cursor := GET\_ITEMS\_IN\_CART();    DBMS\_SQL.RETURN\_RESULT(items\_cursor);  END;  /  ---------------- PRODUCT TABLE ---------------------  CREATE OR REPLACE PROCEDURE INSERT\_PRODUCT(  product\_id IN NUMBER,  product\_name IN VARCHAR2,  product\_material IN VARCHAR2,  product\_size IN VARCHAR2,  product\_type IN VARCHAR2  )  AS  BEGIN  INSERT INTO PRODUCT VALUES (product\_id, product\_name, product\_material, product\_size, product\_type);  END;  /  -- Вызов хранимой процедуры INSERT\_PRODUCT  BEGIN  INSERT\_PRODUCT(2, 'TESTNAME2', 'TESTMATERIAL3', '10M^2', 'ШКАФ');  COMMIT;  END;  /  SELECT \* FROM PRODUCT;  ---------------- DELIVERY TABLE ---------------------  CREATE OR REPLACE PROCEDURE INSERT\_DELIVERY(  delivery\_id IN NUMBER,  address IN VARCHAR2,  payment\_method IN VARCHAR2,  delivery\_method IN VARCHAR2,  date\_of\_delivery IN VARCHAR2  )  AS  BEGIN  INSERT INTO DELIVERY VALUES (delivery\_id, address, payment\_method, delivery\_method, date\_of\_delivery);  END;  /  BEGIN  INSERT\_DELIVERY(2, 'TESTADRESS3', 'CARD', 'КУРЬЕР', '2024-04-29');  COMMIT;  END;  /  SELECT\* FROM DELIVERY;  ---------------- ORDERS TABLE ---------------------  CREATE OR REPLACE PROCEDURE INSERT\_ORDERS(  order\_id IN NUMBER,  order\_date IN VARCHAR2,  order\_status IN VARCHAR2,  delivery\_id IN NUMBER,  user\_id IN NUMBER  )  AS  BEGIN  INSERT INTO ORDERS VALUES (order\_id, order\_date, order\_status, delivery\_id, user\_id);  END;  /  BEGIN  INSERT\_ORDERS(5, '2024-04-29', 'ДОСТАВЛЕНО', 1, 2);  COMMIT;  END;  /  SELECT \* FROM DB\_USERS;  SELECT \*FROM DELIVERY;  select \*from ORDERS;  ---------------- VIEWS ---------------------  CREATE OR REPLACE VIEW INFORMATION\_ABOUT\_ORDER AS  SELECT  ORDERS.ORDER\_ID,  ORDERS.ORDER\_DATE,  ORDERS.ORDER\_STATUS,  DELIVERY.ADRESS,  DELIVERY.PAYMENT\_METHOD,  DELIVERY.DELIVERY\_METHOD,  DELIVERY.DATE\_OF\_DELIVERY,  DB\_USERS.LOGIN,  DB\_USERS.EMAIL,  DB\_USERS.PASSWORD  FROM ORDERS  INNER JOIN DELIVERY ON DELIVERY.DELIVERY\_ID = ORDERS.DELIVERY\_ID  INNER JOIN DB\_USERS ON ORDERS.USER\_ID = DB\_USERS.USER\_ID;  SELECT \* FROM INFORMATION\_ABOUT\_ORDER; |